One-liners

Remove all spaces from all files in current folder

```
>_ rename 's/ //g' *
```

A function to find the fastest free DNS server

```
>_ timeDNS() { parallel -j0 --tag dig @{} "$*" ::: 208.67.222.222 208.67.220.220 198.153.192.1 198.153.194.1 156.154.70.1 156.154.71.1 8.8.8.8 8.8.4.4 | grep Query | sort -nk5; }
```

New Maintainer for CommandLineFu

```
>_ mail tech@commandlinefu.com
```

Bare Metal IRC Client

```
>_ nik=clf$RANDOM;sr=irc.efnet.org; expect -c "set timeout -1;spawn nc $sr 6666;set send_human {.1 .2 1 .2 1};expect AUTH*\n ;send -h \"user $nik * * :$nik commandlinefu\nnick $nik\n\"; interact -o -re (PING.:)(.*\$) {send \"PONG :\$interact_out(2,string)\"}"
```

Get your Firefox history

```
>_ sqlite3 ~/.mozilla/firefox/*.[dD]efault/places.sqlite "SELECT strftime('%d.%m.%Y %H:%M:%S', visit_date/1000000, 'unixepoch', 'localtime'),url FROM moz_places, moz_historyvisits WHERE moz_places.id = moz_historyvisits.place_id ORDER BY visit_date;"
```

Submit command & rewrite orginal command

```
>_ <ctrl>+o
```

Check all bash scripts in current dir for syntax errors

```
>_ find . -name '*.sh' -exec bash -n {} \;
```

Encrypted chat with netcat and openssl (one-liner)

```
>_ server$ while true; do read -n30 ui; echo $ui |openssl enc -aes-256-ctr -a -k PaSSw; done | nc -l -p 8877 | while read so; do decoded_so=` echo "$so"| openssl enc -d -a -aes-256-ctr -k PaSSw`; echo -e "Incoming: $decoded_so"; done
```

Symlink all files from a base directory to a target directory

```
>_ ln -s /BASE/* /TARGET/
```

List directories only

```
>_ ls -d */
```

Show running services (using systemctl)

```
>_ command systemctl --no-page --no-legend --plain -t service --state=running
```

Binary clock

```
>_ perl -e 'for(;;sleep 1){printf"\r"."%.4b "x6,split"",`date +%H%M%S`}'
```

Pulsed terminal clock

```
>_ clear;while true;sleep 1;do for((a=1; a<=$( tput cols)/3;a++));do tput cup 0 $a;echo " " $(date);done;sleep 1;for((a;a>=1;a--));do tput cup 0 $a;echo $(date) " ";done;done
```

Removes characters from cursor to the end of line

```
>_ Ctrl+k
```

All what exists in dir B and not in dir A will be copied from dir B to new or existing dir C

```
>_ rsync -v -r --size-only --compare-dest=../A/ B/ C/
```

Find previously entered commands (requires configuring .inputrc)

```
>_ <Meta-p> (aka <ALT+P>)
```

Add directory to $PATH if it's not already there

```
>_ if [[ ":$PATH:" != *":$dir:"* ]]; then PATH=${PATH}:$dir; fi
```

Debug how files are being accessed by a process

```
>_ inotifywait -m -r .
```

Retry the previous command until it exits successfully

```
>_ !!; while [ $? -ne 0 ]; do !!; done
```

Convert JSON to YAML

```
>_ python -c 'import sys, yaml, json;
yaml.safe_dump(json.load(sys.stdin),
sys.stdout, default_flow_style=False)' <
file.json > file.yaml
```

Search google.com on your terminal

```
>_ Q="YOURSEARCH";
GOOG_URL="http://www.google.com
/search?q="; AGENT="Mozilla/4.0";
stream=$(curl -A "$AGENT" -skLm 10
"${GOOG_URL}\"${Q/\ /+}\"" | grep
-oP '\/url\?q=.+?&amp' | sed 's/\/url?q=
//;s/&amp//'); echo -e "${stream//\%/\x}"
```

Play awesome rythmic noise using aplay

```
>_ echo "main(i){for(i=0;;
i++)putchar(((i*(i>>8|i>>9)&46&
i>>8))^(i&i>>13|i>>6));}" | gcc -x c -
&& ./a.out | aplay
```

Turns red the stderr output

```
>_ color()(set -o pipefail;"$@"
2>&1>&3|sed $'s,.*,\e[31m&
\e[m,'>&2)3>&1
```

Recursively unrar into dir containing archive

```
>_ find . -name '*.rar' -execdir unrar e
{} \;
```

True Random Dice Roll

```
>_ tr -cd '1-6' < /dev/urandom | head -c
1; echo
```

Find biggest 10 files in current and subdirectories and sort by file size

```
>_ find . -type f -print0 | xargs -0 du -h |
sort -hr | head -10
```

Get your external IP and Network Info

```
>_ curl ifconfig.me/all
```

List all commands present on system

```
>_ ls ${PATH//:/ }
```

A line across the entire width of the terminal

```
>_ printf "%$(tput cols)s\n"|tr ' ' '='
```

Recursively find top 20 largest files (> 1MB) sort human readable format

```
>_ find . -type f -print0 | xargs -0 du -h | sort -hr | head -20
```

Create the four oauth keys required for a Twitter stream feed

```
>_ step1() { k1="Consumer key" ; k2="Consumer secret" ; k3="Access token" ; k4="Access token secret" ; once=$RANDOM ; ts=$(date +%s) ; hmac="$k2&$k4" ; id="19258798" ; }
```

Check if a machine is online

```
>_ ping -c 1 -q MACHINE_IP_OR_NAME >/dev/null 2>&1 && echo ONLINE || echo OFFLINE
```

Extract element of xml

```
>_ xmlstarlet sel -t -c "/path/to/element" file.xml
```

Generate a random password 30 characters long

```
>_ cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 32
```

Remove the first and the latest caracter of a string

```
>_ var=:foobar:; echo ${var:1:-1}
```

Load file into RAM (cache) for faster accessing for repeated usage

```
>_ cat <file> > /dev/null
```

Show IP Address in prompt --> PS1 var

```
>_ export PS1="[\u@`hostname -I` \W]$ "
```

Log a command to console and to 2 files separately stdout and stderr

```
>_ command > >(tee stdout.log) 2> >(tee stderr.log >&2)
```

Join lines split with backslash at the end

```
>_ sed -e '/\\$/{:0;N;s/\\\n//;t0}'
```

Convert text to lowercase

```
>_ lower() { echo ${@,,}; }
```

Find builtin in bash v4+

```
>_ ls -l /etc/**/*killall
```

Make image semi-transparent

```
>_ convert input.png -alpha set -channel A -fx 0.5 output.png
```

Compare directories via diff

```
>_ diff -rq dirA dirB
```

Execute a shell with netcat without -e

```
>_ mknod backpipe p && nc remote_server 1337 0<backpipe | /bin/bash 1>backpipe
```

Save your terminal commands in bash history in real time

```
>_ shopt -s histappend ; PROMPT_COMMAND="history -a;$PROMPT_COMMAND"
```

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 }'`
```

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 Unavailable";'
```

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'
```

Tricky implementation of two-dimensional array in Bash.

```
>_ arr[i*100+j]="whatever"
```

Terrorist threat level text

```
>_ echo "Terrorist threat level: `od -An -N1 -i /dev/random`"
```

Bash logger

```
>_ script /tmp/log.txt
```

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>
```

Convert a string to "Title Case"

```
>_ echo "this is a test" | sed 's/.*/\L&/; s/[a-z]*/\u&/g'
```

RTFM function

```
>_ rtfm() { help $@ || info $@ || man $@ || $BROWSER "http://www.google.com/search?q=$@"; }
```

Back ssh from firewalled hosts

```
>_ ssh -R 5497:127.0.0.1:22 -p 62220 user@public.ip
```

Rename HTML files according to their title tag

```
>_ perl -wlne'/title>([^<]+)/i&&rename$ARGV,"$1.html"' *.html
```

Rename files according to file with colums of corresponding names

```
>_ xargs -n 2 mv < file_with_colums_of_names
```

Create a new file

```
>_ > file
```

Monitor a file with tail with timestamps added

```
>_ tail -f file | awk '{now=strftime("%F %T%z\t");sub(/^/, now);print}'
```

Check disk for bad sectors

```
>_ badblocks -n -s -b 2048 /dev/sdX
```

Create a file of a given size in linux

```
>_ truncate -s 1M file
```

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"
```

Extended man command

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

Show which process is blocking umount (Device or resource is busy)

```
>_ lsof /folder
```

Show the UUID of a filesystem or partition

```
>_ blkid /dev/sda7
```

Run command on a group of nodes

```
>_ mussh -h host1 host2 host3 -c uptime
```

Setting global redirection of STDERR to STDOUT in a script

```
>_ exec 2>&1
```

Outgoing IP of server

```
>_ dig +short @resolver1.opendns.com myip.opendns.com
```

Isolate file name from full path/find output

```
>_ echo ${fullpath##*/}
```

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"
```

Find all symlinks that link to directories

```
>_ find -type l -xtype d
```

Use Kernighan & Ritchie coding style in C program

```
>_ indent -kr hello.c
```

Delay execution until load average falls under 1.5

```
>_ echo 'some command' | batch
```

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 -pmysqlpassword dbname | ssh -C user@ sshserver 'cd /path/to /backup/dir; cat > dbname.sql'
```

Tail -f at your own pace

```
>_ tail -fs 1 somefile
```

Watch the progress of 'dd'

```
>_ dd if=/dev/zero | pv | dd of=/dev/null
```

Realtime lines per second in a log file

```
>_ tail -f access.log | pv -l -i10 -r >/dev/null
```

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
$(dirname "$1"); else command cd "$1";
fi; fi; }
```

Temporarily ignore known SSH hosts

```
>_ ssh -o
UserKnownHostsFile=/dev/null
root@192.168.1.1
```

Dd with progress bar

```
>_ dd if=/dev/nst0 |pv| dd
of=restored_file.tar
```

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

```
>_ curl http://www.commandlinefu.com
/commands/browse/sort-by-
votes/plaintext/[0-2500:25] | grep -v _
curl_ > comfu.txt
```

Print a row of 50 hyphens

```
>_ python -c 'print "-"*50'
```

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}"
```

Grep tab chars

```
>_ grep "^V<TAB>" your_file
```

Send a message to a windows machine in a popup

```
>_ echo "message" | smbclient -M
NAME_OF_THE_COMPUTER
```

Using mplayer to play the audio only but suppress the video

```
>_ mplayer -novideo something.mpg
```

Using mplayer to play the audio only but suppress the video

```
>_ mplayer -vo null something.mpg
```

Create a temporary file in a command line call

```
>_ any_script.sh < <(some command)
```

Shell function to make gnu info act like man.

```
>_ my info() { info --subnodes -o - $1 | less; }
```

Replace spaces in filenames with underscores

```
>_ for f in *;do mv "$f" "${f// /_}";done
```

Insert the last argument of the previous command

```
>_ !$
```

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'
```

Remote control for Rhythmbox on an Ubuntu Media PC

```
>_ alias rc='ssh ${MEDIAPCHOSTNAME} env DISPLAY=:0.0 rhythmbox-client --no-start'
```

Use colordiff in side-by-side mode, and with automatic column widths.

```
>_ colordiff -yW"`tput cols`" /path/to/file1 /path/to/file2
```

Extract email adresses from some file (or any other pattern)

```
>_ grep -Eio '([[:alnum:]_.-]+@[[:alnum:]_.-]+?\.[[:alpha:].]{2,6})'
```

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
```

Bash screensaver off

```
>_ setterm -powersave off -blank 0
```

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
```

Smart renaming

```
>_ mmv 'banana_*_*.asc' 'banana_#2_#1.asc'
```

Is today the end of the month?

```
>_ [ `date --date='next day' +'%B'` == `date +'%B'` ] || echo 'end of month'
```

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
```

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 &
```

Show Directories in the PATH Which does NOT Exist

```
>_ (IFS=:;for p in $PATH; do test -d $p || echo $p; done)
```

Backup files older than 1 day on /home/dir, gzip them, moving old file to a dated file.

```
>_ find /home/dir -mtime +1 -print -exec gzip -9 {} \; -exec mv {}.gz {}_`date +%F`.gz \;
```

Alias to close terminal with :q

```
>_ alias ':q'='exit'
```

Display an embeded help message from bash script header

```
>_ [ "$1" == "--help" ] && { sed -n -e '/^# Usage:/,/^$/ s/^# \?//p' < $0; exit; }
```

Sort file greater than a specified size in human readeable format including their path and typed by color, running from current directory

```
>_ find ./ -size +10M -type f -print0 | xargs -0 ls -Ssh1 --color
```

Print file without duplicated lines using awk

```
>_ awk '!a[$0]++' file
```

Get the 10 biggest files/folders for the current direcotry

```
>_ du -sk * |sort -rn |head
```

Get your external IP address

```
>_ curl -s 'http://checkip.dyndns.org' | sed 's/.*Current IP Address: \([0-9\.]*\).*/\1/g'
```

Send email with one or more binary attachments

```
>_ echo "Body goes here" | mutt -s "A subject" -a /path/to/file.tar.gz recipient@example.com
```

Convert a web page into a png

```
>_ touch $2;firefox -print $1 -printmode PNG -printfile $2
```

Uncomment the lines where the word DEBUG is found

```
>_ sed '/^#.*DEBUG.*/ s/^#//' $FILE
```

This toggles mute on the Master channel of an alsa soundcard

```
>_ amixer sset Master toggle
```

Submit data to a HTML form with POST method and save the response

```
>_ curl -sd 'rid=value&submit=SUBMIT' <URL> > out.html
```

Count how many times a string appears in a (source code) tree

```
>_ $ grep -or string path/ | wc -l
```

Search commandlinefu from the CLI

```
>_ curl -sd q=Network http://www.commandlinefu.com/search/autocomplete |html2text -width 100
```

Compare a remote file with a local file

```
>_ vimdiff <file> scp://[<user>@]<host>/<file>
```

Using netcat to copy files between servers

```
>_ On target: "nc -l 4000 | tar xvf -" On source: " tar -cf - . | nc target_ip 4000"
```

Search command history on bash

```
>_ ctrl + r
```

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 bash inline

```
>_ <<<"k=1024; m=k*k; g=k*m; g" bc
```

Alt + 1 .

```
>_ alt + 1 .
```

Create several copies of a file

```
>_ for i in {1..5}; do cp test{,$i};done
```

Search for a single file and go to it

```
>_ cd $(dirname $(find ~ -name
emails.txt))
```

Nice info browser

```
>_ pinfo
```

Sends your internal IP by email

```
>_ ifconfig en1 | awk '/inet / {print $2}'
| mail -s "hello world" email@email.com
```

Download schedule

```
>_ echo 'wget url' | at 12:00
```

Use mtr to create a text file report

```
>_ mtr --report --report-cycles 10
www.google.com >
google_net_report.txt
```

Quick screenshot

```
>_ import -pause 5 -window root
desktop_screenshot.jpg
```

Edit the last or previous command line in an editor then execute

```
>_ fc [history-number]
```

Quicker move to parent directory

```
>_ alias ..='cd ..'
```

Show webcam output

```
>_ mplayer tv:// -tv
driver=v4l:width=352:height=288
```

Open the last file you edited in Vim.

```
>_ alias lvim="vim -c \"normal '0\""
```

Keep from having to adjust your volume constantly

```
>_ find . -iname \*.mp3 -print0 | xargs
-0 mp3gain -krd 6 && vorbisgain -rfs .
```

[re]verify a disc with very friendly output

```
>_ dd if=/dev/cdrom | pv -s 700m |
md5sum | tee test.md5
```

Processes by CPU usage

```
>_ ps -e -o pcpu,cpu,
nice,state,cputime,args --sort pcpu | sed
"/^ 0.0 /d"
```

Rotate a set of photos matching their EXIF data.

```
>_ jhead -autorot *.jpg
```

Awk using multiple field separators

```
>_ awk -F "=| "
```

Shows size of dirs and files, hidden or not, sorted.

```
>_ du -cs * .[^\.]* | sort -n
```

Hard disk information - Model/serial no.

```
>_ hdparm -i[I] /dev/sda
```

Use mplayer to save video streams to a file

```
>_ mplayer -dumpstream -dumpfile
"yourfile" -playlist "URL"
```

Wrap text files on the command-line for easy reading

```
>_ fold -s <filename>
```

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

```
>_ blockdev --rereadpt /dev/sda
```

Get your mac to talk to you

```
>_ say -v Vicki "Hi, I'm a mac"
```

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.txt
```

Recursively change file name from uppercase to lowercase (or viceversa)

```
>_ find . -type f|while read f; do mv $f
`echo $f |tr '[:upper:]' '[ :lower:]'`; done
```

Check the age of the filesystem

```
>_ df / | awk '{print $1}' | grep dev |
xargs tune2fs -l | grep create
```

Backup all MySQL Databases to individual files

```
>_ for I in `echo "show databases;" |
mysql | grep -v Database`; do
mysqldump $I > "$I.sql"; done
```

Remove all subversion files from a project recursively

```
>_ rm -rf `find . -type d -name .svn`
```

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
```

Print indepth hardware info

```
>_ sudo dmidecode | more
```

Concatenate (join) video files

```
>_ mencoder -forceidx -ovc copy -oac copy -o output.avi video1.avi video2.avi
```

Convert the contents of a directory listing into a colon-separated environment variable

```
>_ find . -name '*.jar' -printf '%f:'
```

Make a log of a terminal session

```
>_ script
```

Remove invalid host keys from ~/.ssh/known_hosts

```
>_ ssh-keygen -R \[localhost\]:8022
```

Optimal way of deleting huge numbers of files

```
>_ rsync -a --delete empty-dir/ target-dir/
```

Drop first column of output by piping to this

```
>_ awk '{ $1="";print}'
```

Create a bunch of dummy text files

```
>_ base64 /dev/urandom | head -c 33554432 | split -b 8192 -da 4 - dummy.
```

Get the Nth argument of the last command (handling spaces correctly)

```
>_ !:n
```

Temporarily ignore mismatched SSH host key

```
>_ ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no username@host
```

Remove all zero size files from current directory (not recursive)

```
>_ find . -maxdepth 1 -size 0c -delete
```

Watch the progress of 'dd'

```
>_ pv -tpreb /dev/urandom | dd of=file.img
```

Get a stream feed from a Twitter user

```
>_ step1 ; step2 ; step3 ; step4 ; curl -o-
--get 'https://stream.twitter.com
/1/statuses/filter.json' --header
"$oauth_header" --data "follow=$id"
```

Create the authorization header required for a Twitter stream feed

```
>_ step4() {
oauth_header="Authorization: OAuth
oauth_consumer_key=\"$k1\",
oauth_nonce=\"$once\",
oauth_signature=\"$signature\",
oauth_signature_method=\"HMAC-
SHA1\", oauth_timestamp=\"$ts\",
oauth_token=\"$k3\",
oauth_version=\"1.0\"" ; }
```

Create the signature base string required for a Twitter stream feed

```
>_ step2(){ b="GET&https%3A
%2F%2Fstream.twitter.com%2F1%2Fst
atuses%2Ffilter.json&
follow%3D${id}%26oauth_consumer_k
ey%3D${k1}%26oauth_nonce%3D${on
ce}%26oauth_signature_method%3DH
MAC-
SHA1%26oauth_timestamp%3D${ts}%
26oauth_token%3D${k3}%26oauth_ver
sion%3D1.0";}
```

Create the oauth token required for a Twitter stream feed

```
>_ step3() { s=$( echo -n $b | openssl
dgst -sha1 -hmac $hmac -binary |
openssl base64); signature=`for((i=0;
i<${#s};i++)); do case ${s:i:1} in +) e
%2B;; /) e %2F;; =) e %3D;; *) e
${s:i:1};; esac ; done` ; } ; e() { echo -n
$1; }
```

Open a screenshot of a remote desktop via ssh

```
>_ xloadimage <(ssh
USER@HOSTNAME DISPLAY=:0.0
import -window root png:-)
```

Check your spelling

```
>_ aspell -a <<< '<WORDS>'
```

Find all files larger than 500M and less than 1GB

```
>_ find / -type f -size +500M -size -1G
```

Show bandwidth use oneliner

```
>_ while true; do cat /proc/net/dev;
sleep 1; done | awk -v dc="date
\"+%T\"" '/eth0/{i = $2 - oi; o = $10 -
oo; oi = $2; oo = $10; dc|getline d;
close(dc); if (a++) printf "%s %8.2f
KiB/s in %8.2f KiB/s out\n", d, i/1024,
o/1024}'
```

Change/Modify timestamp interactively

```
>_ touch -d $(zenity --calendar --date-format=%F) filename
```

Sort the contents of a text file in place.

```
>_ sort -g -o list.txt{,}
```

Instantly load bash history of one shell into another running shell

```
>_ $ history -a #in one shell , and $ history -r #in another running shell
```

Copy with progress bar - rsync

```
>_ rsync -rv <src> <dst> --progress
```

Multiple variable assignments from command output in BASH

```
>_ read day month year < <(date +'%d %m %y')
```

Generate a random password 30 characters long

```
>_ gpg --gen-random --armor 1 30
```

Run a ext4 file system check and badblocks scan with progress info

```
>_ fsck.ext4 -cDfty -C 0 /dev/sdxx
```

List your MACs address

```
>_ cat /sys/class/net/eth0/address
```

Rename all files which contain the sub-string 'foo', replacing it with 'bar'

```
>_ for i in ./*foo*;do mv -- "$i" "${i//foo/bar}";done
```

Create a 5 MB blank file via a seek hole

```
>_ dd if=/dev/zero of=testfile.seek seek=5242879 bs=1 count=1
```

ROT13 using the tr command

```
>_ alias rot13="tr a-zA-Z n-za-mN-ZA-M"
```

Make vim open in tabs by default (save to .profile)

```
>_ alias vim="vim -p"
```

Secure copy from one server to another without rsync and preserve users, etc

```
>_ tar -czvf - /src/dir | ssh remotehost
"(cd /dst/dir ; tar -xzvf -)"
```

The same as [Esc] in vim

```
>_ Ctrl + [
```

Binary clock

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

Function to split a string into an array

```
>_ read -a ARR <<<'world domination
now!'; echo ${ARR[2]};
```

Generate MD5 hash for a string

```
>_ md5sum <<<"test"
```

Recompress all .gz files in current directory using bzip2 running 1 job per CPU core in parallel

```
>_ parallel -j+0 "zcat {} | bzip2
>{.}.bz2 && rm {}" ::: *.gz
```

Phpinfo from the command line

```
>_ php -i
```

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 }
```

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 ;}
```

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
```

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 "[$?] $(history|tail -n1|sed
-e '\''s/^\s*[0-9]\+\s*//;s/;\s*alert$//'\'')"'
```

Runs a X session within your X session

```
>_ ssh -C -Y -l$USER xserver.mynet.xx
'Xnest -geometry 1900x1150 -query
localhost'
```

Use screen as a terminal emulator to connect to serial consoles

```
>_ screen /dev/tty<device> 9600
```

Have a random "cow" say a random thing

```
>_ fortune | cowsay -f $(ls /usr/share/
cowsay/cows/ | shuf -n1)
```

Sends a postscript file to a postscript printer using netcat

```
>_ cat my.ps | nc -q 1 hp4550.mynet.xx
9100
```

Lines in file2 that are not in file1

```
>_ grep -Fxv -f file1 file2
```

Display which distro is installed

```
>_ cat /etc/*release
```

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}"|sed -n 's/.*ge=\([0-9]
\{2\}\).*/\1/p'`;for((p=1;p<=l;p++));do
for u in `curl -s "${d}page=${p}&
lettre=${c}"|egrep -o "http\S*.com/dl
/\?f=\w*"`;do aria2c
"${u}";done;done;done
```

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
```

Send DD a signal to print its progress

```
>_ while :;do killall -USR1 dd;sleep
1;done
```

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
$@ &
```

See your current RAM frequency

```
>_ dmidecode -t 17 | awk -F":" '/Speed/
{ print $2 }'
```

Scp with compression.

```
>_ scp -C 10.0.0.4:/tmp/backup.sql /path/to/backup.sql
```

Perl Command Line Interpreter

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

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
```

Find and delete empty dirs, start in current working dir

```
>_ find . -type d -empty -delete
```

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" }'
```

Optimal way of deleting huge numbers of files

```
>_ find /path/to/dir -type f -delete
```

Geoip information

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

What is my ip?

```
>_ curl http://www.whatismyip.org/
```

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 -m1 "$S" FF|xargs -P5 -iFF sed -i -e "s%${S}%${R}%g" FF'
```

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 -identify 2>/dev/null | perl -nle '/ID_LENGTH=([0-9\.]+)/ && ($t +=$1) && printf "%02d:%02d:%02d\n",$t/3600,$t/60%60,$t%60' | tail -n 1
```

Send data securly over the net.

```
>_ cat /etc/passwd | openssl aes-256-cbc -a -e -pass pass:password | net cat -l -p 8080
```

When was your OS installed?

```
>_ ls -lct /etc | tail -1 | awk '{print $6,
$7}'
```

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
```

Replace spaces in filenames with underscorees

```
>_ ls | while read f; do mv "$f" "${f//
/_}";done
```

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
```

Grep certain file types recursively

```
>_ grep -r --include="*.[ch]" pattern .
```

Change proccess affinity.

```
>_ taskset -cp <core> <pid>
```

Back up your commandlinefu contributed commands

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

Currently mounted filesystems in nice layout

```
>_ column -t /proc/mounts
```

Benchmark web server with apache benchmarking tool

```
>_ ab -n 9000 -c 900
localhost:8080/index.php
```

Salvage a borked terminal

```
>_ echo <ctrl-v><esc>c<enter>
```

Multiple SSH Tunnels

```
>_ ssh -L :: -L :: @
```

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)'
```

Easily scp a file back to the host you're connecting from

```
>_ mecp () { scp "$@" ${SSH_CLIENT%% *}:Desktop/; }
```

Super Speedy Hexadecimal or Octal Calculations and Conversions to Decimal.

```
>_ echo "$(( 0x10 )) - $(( 010 )) = $(( 0x10 - 010 ))"
```

Find and replace tabs for spaces within files recursively

```
>_ find ./ -type f -exec sed -i 's/\t/ /g' {} \;
```

Delete all files found in directory A from directory B

```
>_ for file in <directory A>/*; do rm <directory B>/`basename $ file`; done
```

Synchronicity

```
>_ cal 09 1752
```

Terminal redirection

```
>_ script /dev/null | tee /dev/pts/3
```

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
```

Rot13 using the tr command

```
>_ alias rot13="tr '[A-Za-z]' '[N-ZA-Mn-za-m]'"
```

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
```

Clean swap area after using a memory hogging application

```
>_ swapoff -a ; swapon -a
```

Loop over a set of items that contain spaces

```
>_ ls | while read ITEM; do echo "$ITEM"; done
```

Traceroute w/TCP to get through firewalls.

```
>_ tcptraceroute www.google.com
```

Launch a command from a manpage

```
>_ !date
```

Split File in parts

```
>_ split -b 19m file Nameforpart
```

Speak the top 6 lines of your twitter timeline every 5 minutes.....

```
>_ while [ 1 ]; do curl -s -u
username:password http://twitter.com
/statuses/friends_timeline.rss|grep
title|sed -ne 's/<\/*title>//gp' | head -n 6
|festival --tts; sleep 300;done
```

Create an SSH SOCKS proxy server on localhost:8000 that will re-start itself if something breaks the connection temporarily

```
>_ autossh -f -M 20000 -D 8000
somehost -N
```

To get you started!

```
>_ vimtutor
```

Exclude grep from your grepped output of ps (alias included in description)

```
>_ ps aux | grep [h]ttpd
```

Renames multiple files that match the pattern

```
>_ rename 's/foo/bar/g' *
```

Block an IP address from connecting to a server

```
>_ iptables -A INPUT -s
222.35.138.25/32 -j DROP
```

Infile search and replace on N files (including backup of the files)

```
>_ perl -pi.bk -e's/foo/bar/g' file1 file2
fileN
```

Du with colored bar graph

```
>_ du -x --max-depth=1|sort -rn|awk -F /
-v c=$COLUMNS 'NR==1{t=$1}
NR>1{r=int($1/t*c+.5);
b="\033[1;31m"; for (i=0; i<r; i++)
b=b"#"; printf " %5.2f%% %s\033[0m
%s\n", $1/t*100, b, $2}'|tac
```

Mount a VMware virtual disk (.vmdk) file on a Linux box

```
>_ kpartx -av <image-flat.vmdk>; mount -o /dev/mapper/loop0p1 /mnt/vmdk
```

Quick copy

```
>_ cp foo{,bak}
```

Sending a file over icmp with hping

```
>_ hping3 10.0.2.254 --icmp --sign MSGID1 -d 50 -c 1 --file a_file
```

Fetch the current human population of Earth

```
>_ curl -s http://www.census.gov/popclock/data/population/world | python -c 'import json,sys;obj=json.load(sys.stdin);print obj["world"]["population"]'
```

Kill a broken ssh connection

```
>_ <Return>~.
```

Convert JSON to YAML

```
>_ ruby -ryaml -rjson -e 'puts YAML.dump(JSON.parse(STDIN.read))' < file.json > file.yaml
```

Convert Shell Text to Upper/Lower Case

```
>_ ALT-U / ALT-L
```

The fastest remote directory rsync over ssh archival I can muster (40MB/s over 1gb NICs)

```
>_ rsync -aHAXxv --numeric-ids --delete --progress -e "ssh -T -c arcfour -o Compression=no -x" user@<source>:<source_dir> <dest_dir>
```

Binary digits Matrix effect

```
>_ perl -e '$|++; while (1) { print " " x (rand(35) + 1), int(rand(2)) }'
```

Open two files side by side in vim (one window, two panes)

```
>_ vim -O file1 file2
```

Repeat a command every one second

```
>_ watch -n 1 "do foo"
```

Update all packages installed via homebrew

```
>_ brew update && brew upgrade `brew outdated`
```

Daemonize nc - Transmit a file like a http server

```
>_ while ( nc -l 80 < /file.htm > : ) ; do : ; done &
```

Find ulimit values of currently running process

```
>_ cat /proc/PID/limits
```

Diff current vi buffer edits against original file

```
>_ :w !diff -u % -
```

Redirect tar extract to another directory

```
>_ tar xfz filename. tar.gz -C
PathToDirectory
```

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")}'
```

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 ;
done
```

Watch several log files of different machines in a single multitail window on your own machine

```
>_ multitail -l 'ssh machine1 "tail -f
/var/log/apache2/error.log"' -l 'ssh
machine2 "tail -f /var/log/apache2
/error.log"'
```

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=1913b4ac473c692372d10820995
8fd15" http://api.imgur.com
/2/upload.xml| grep -Eo "<original>
(.)*</original>" | grep -Eo
"http://i.imgur.com/[^<]*";}
```

All out

```
>_ pkill -KILL -u username
```

Numeric zero padding file rename

```
>_ rename 's/\d+/sprintf("%04d",$&)/e'
*.jpg
```

Find out the starting directory of a script

```
>_ echo "${0%/*}"
```

Selecting a random file/folder of a folder

```
>_ shuf -n1 -e *
```

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
```

List all file extensions in a directory

```
>_ find . -type f | awk -F'.' '{print $NF}'
| sort| uniq -c | sort -g
```

Send an http HEAD request w/curl

```
>_ curl -I http://localhost
```

View hex mode in vim

```
>_ :%!xxd
```

View ~/.ssh/known_hosts key information

```
>_ ssh-keygen -l -f ~/.ssh/known_hosts
```

Kill all Zombie processes (Guaranteed!)

```
>_ kill -9 `ps -xaw -o state -o ppid | grep
Z | grep -v PID | awk '{print $2}'`
```

Prevent accidents and test your command with echo

```
>_ echo rm *.txt
```

Exclude a column with awk

```
>_ awk '{ $5=""; print }' file
```

Get My Public IP Address

```
>_ curl ifconfig.me
```

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
```

Empty a file

```
>_ :> file
```

Better way to use notify-send with at or cron

```
>_ DISPLAY=:0.0 XAUTHORITY=~/.Xauthority notify-send test
```

Emptying a text file in one shot

```
>_ :%d
```

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
```

Countdown Clock

```
>_ MIN=1 && for i in $(seq $(($MIN*60)) -1 1); do echo -n "$i, "; sleep 1; done; echo -e "\n\nBOOOM! Time to start."
```

Print info about your real user.

```
>_ who loves mum
```

Serve current directory tree at http://$HOSTNAME:8080/

```
>_ twistd -n web --path .
```

JSON processing with Python

```
>_ curl -s "http://feeds.delicious.com/v2/json?count=5" | python -m json.tool | less -R
```

Move items from subdirectories to current directory

```
>_ find -type f -exec mv {} . \;
```

Restrict the bandwidth for the SCP command

```
>_ scp -l10 pippo@serverciccio:/home/zutaniddu/* .
```

Optimize PDF documents

```
>_ gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dPDFSETTINGS=/screen -dNOPAUSE -dQUIET -dBATCH -sOutputFile=output.pdf input.pdf
```

Live ssh network throughput test

```
>_ pv /dev/zero|ssh $host 'cat > /dev/null'
```

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
```

Upgrade all perl modules via CPAN

```
>_ cpan -r
```

Remove lines that contain a specific pattern($1) from file($2).

```
>_ sed -i '/myexpression/d' /path/to
/file.txt
```

List your largest installed packages (on Debian/Ubuntu)

```
>_ dpigs
```

Rsync + find

```
>_ find . -name "whatever.*" -print0 |
rsync -av --files-from=- --from0 ./
./destination/
```

Autossh + ssh + screen = super rad perma-sessions

```
>_ AUTOSSH_POLL=1 autossh -M
21010 hostname -t 'screen -Dr'
```

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
```

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
```

Cycle through a 256 colour palette

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

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
```

Move a lot of files over ssh

```
>_ tar -cf - /home/user/test | gzip -c |
ssh user@ sshServer 'cd /tmp; tar xfz -'
```

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)
```

Vim easter egg

```
>_ $ vim ... :help 42
```

Find the process you are looking for minus the grepped one

```
>_ pgrep command_name
```

Stream YouTube URL directly to mplayer

```
>_ id="dMH0bHeiRNg";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')
```

Update twitter via curl

```
>_ curl -u user -d status="Tweeting
from the shell" http://twitter.com
/statuses/update.xml
```

Ssh autocomplete

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

Get all IPs via ifconfig

```
>_ ifconfig | perl -nle'/dr:(\S+)/ &&
print $1'
```

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
```

Get the canonical, absolute path given a relative and/or noncanonical path

```
>_ readlink -f ../super/symlink_bon/ahoy
```

Create a Multi-Part Archive Without Proprietary Junkware

```
>_ tar czv Pictures | split -d -a 3 -b 16M
- pics. tar.gz.
```

List and delete files older than one year

```
>_ find <directory path> -mtime +365 -and -not -type d -delete
```

Display last exit status of a command

```
>_ echo $?
```

Enable ** to expand files recursively (>=bash-4.0)

```
>_ shopt -s globstar
```

Command Line to Get the Stock Quote via Yahoo

```
>_ curl -s 'http://download.finance.yahoo.com/d/quotes.csv?s=csco&f=l1'
```

Plays Music from SomaFM

```
>_ read -p "Which station? "; mplayer --reallyquiet -vo none -ao sdl http://somafm.com/startstream=${REPLY}.pls
```

Convert camelCase to underscores (camel_case)

```
>_ sed -r 's/([a-z]+)([A-Z][a-z]+)/\1_\l\2/g' file.txt
```

Set your profile so that you resume or start a screen session on login

```
>_ echo "screen -DR" >> ~/.bash_profile
```

Play high-res video files on a slow processor

```
>_ mplayer -framedrop -vfm ffmpeg -lavdopts lowres=1:fast:skiploopfilter=all
```

Create directory named after current date

```
>_ mkdir $(date +%Y%m%d)
```

Monitor dynamic changes in the dmesg log.

```
>_ watch "dmesg |tail -15"
```

Generate a list of installed packages on Debian-based systems

```
>_ dpkg --get-selections > LIST_FILE
```

Find the process that is using a certain port e.g. port 3000

```
>_ lsof -P | grep ':3000'
```

Pause Current Thread

```
>_ ctrl-z
```

Converts to PDF all the OpenOffice.org files in the directory

```
>_ for i in $(ls *.od{tp}); do unoconv -f pdf $i; done
```

Create a bunch of dummy files for testing

```
>_ touch {1..10}.txt
```

Convert .wma files to .ogg with ffmpeg

```
>_ find -name '*wma' -exec ffmpeg -i {} -acodec vorbis -ab 128k {}.ogg \;
```

Find distro name and/or version/release

```
>_ cat /etc/*-release
```

Generate Random Passwords

```
>_ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c6
```

Bash shortcut: !$!^ !* !:3 !:h and !:t

```
>_ echo foo bar foobar barfoo && echo !$ !^ !:3 !* && echo /usr/bin/foobar&& echo !$:h !$:t
```

Grep without having it show its own process in the results

```
>_ ps aux | grep "[s]ome_text"
```

Find recursively, from current directory down, files and directories whose names contain single or multiple whitespaces and replace each such occurrence with a single underscore.

```
>_ find ./ -name '*' -exec rename 's/\s+/_/g' {} \;
```

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'
```

Quickly get summary of sizes for files and folders

```
>_ du -sh *
```

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"} '
```

Disco lights in the terminal

```
>_ while true; do printf "\e[38;5;$(($(od -d -N 2 -A n /dev/urandom)%$(tput colors)))m.\e[0m"; done
```

Select rectangular screen area

```
>_ Ctrl + Alt
```

Show how old your linux OS installtion is

```
>_ sudo tune2fs -l $(df -h / |(read; awk '{print $1; exit}')) | grep -i created
```

Discover the process start time

```
>_ ps -eo pid,lstart,cmd
```

Automatically find and re-attach to a detached screen session

```
>_ screen -D -R
```

This is how you should push a string in a command's stdin.

```
>_ command <<< word
```

Ping as traceroute

```
>_ for i in {1..30}; do ping -t $i -c 1 google.com; done | grep "Time to live exceeded"
```

What model of computer I'm using?

```
>_ sudo dmidecode | grep Product
```

Extract tar content without leading parent directory

```
>_ tar -xaf archive. tar.gz --strip-components=1
```

Target a specific column for pattern substitution

```
>_ awk '{gsub("foo","bar",$5)}1' file
```

Run a command when a file is changed

```
>_ while inotifywait -e modify /tmp/myfile; do firefox; done
```

Substrings a variable

```
>_ var='123456789'; echo ${var:<start_pos>:<offset>}
```

One command line web server on port 80 using nc (netcat)

```
>_ while true ; do nc -l 80 < index.html ; done
```

RDP through SSH tunnel

```
>_ ssh -f -L3389:<RDP_HOST>:3389 <SSH_PROXY> "sleep 10" && rdesktop -T'<WINDOW_TITLE>' -uAdministrator -g800x600 -a8 -rsound:off -rclipboard:PRIMARYCLIPBOARD -5 localhost
```

"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 '{print $2}'`
```

Remote screenshot

```
>_ ssh user@remote-host "DISPLAY=:0.0 import -window root -format png -"|display -format png -
```

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 ; }
```

Add the result of a command into vi

```
>_ !!command
```

Place the NUM-th argument of the most recent command on the shell

```
>_ <ALT>+<.> or <ALT>+<NUM>+<.> or <ALT>+<NUM>, <ALT>+<.>
```

Find the package a command belongs to on Debian

```
>_ dpkg -S $( which ls )
```

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
```

Start dd and show progress every X seconds

```
>_ dd if=/path/inputfile | pv | dd of=/path/outpufile
```

Ask user to confirm

```
>_ Confirm() { read -sn 1 -p "$1 [Y/N]? "; [[ $REPLY = [Yy] ]]; }
```

Random unsigned integer

```
>_ echo $(openssl rand 4 | od -DAn)
```

Kill all process that belongs to you

```
>_ kill -9 -1
```

Make a file not writable / immutable by root

```
>_ sudo chattr +i <file>
```

Analyze traffic remotely over ssh w/ wireshark

```
>_ ssh root@HOST tcpdump -U -s0 -w - 'not port 22' | wireshark -k -i -
```

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
```

Translates acronyms for you

```
>_ wtf is <acronym>
```

Delete DOS Characters via VIM (^M)

```
>_ :set ff=unix
```

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
```

Print just line 4 from a textfile

```
>_ sed -n '4{p;q}'
```

Print just line 4 from a textfile

```
>_ sed -n '4p'
```

Quick HTML image gallery from folder contents

```
>_ find . -iname '*.jpg' -exec echo '<img src="{}">' \; > gallery.html
```

Get your external IP address without curl

```
>_ wget -qO- icanhazip.com
```

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

```
>_ screen -raAd
```

Start a HTTP server which serves Python docs

```
>_ pydoc -p 8888 & gnome-open
http://localhost:8888
```

Do some learning...

```
>_ ls /usr/bin | xargs whatis | grep -v
nothing | less
```

Insert the last argument of the previous command

```
>_ <ALT> .
```

Carriage return for reprinting on the same line

```
>_ while true; do echo -ne "$(date)\r";
sleep 1; done
```

Copy a folder tree through ssh using compression (no temporary files)

```
>_ ssh <host> ' tar -cz /<folder>
/<subfolder>' | tar -xvz
```

Command line calculator

```
>_ calc(){ awk "BEGIN{ print $* }" ;}
```

Backup a local drive into a file on the remote host via ssh

```
>_ dd if=/dev/sda | ssh user@server '
dd of=sda.img'
```

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
```

List Network Tools in Linux

```
>_ apropos network |more
```

Print text string vertically, one character per line.

```
>_ echo "vertical text" | grep -o '.'
```

Add prefix onto filenames

```
>_ rename 's/^/prefix/' *
```

Salvage a borked terminal

```
>_ <ctrl+j>stty sane<ctrl+j>
```

Pick a random line from a file

```
>_ shuf -n1 file.txt
```

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
```

Stripping ^M at end of each line for files

```
>_ dos2unix <filenames>
```

Backup files incremental with rsync to a NTFS-Partition

```
>_ rsync -rtvu --modify-window=1 --progress /media/SOURCE/ /media/TARGET/
```

Find removed files still in use via /proc

```
>_ find -L /proc/*/fd -links 0 2>/dev/null
```

Connect to google talk through ssh by setting your IM client to use the localhost 5432 port

```
>_ ssh -f -N -L 5432:talk.google.com:5222 user@home.network.com
```

Urldecoding

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

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
```

Wrap long lines of a text

```
>_ fold -s -w 90 file.txt
```

Validate and pretty-print JSON expressions.

```
>_ echo '{"json":"obj"}' | python -m simplejson.tool
```

Grep colorized

```
>_ grep -i --color=auto
```

From within vi, pipe a chunk of lines to a command line and replace the chunk with the result

```
>_ !}sort
```

Grep for word in directory (recursive)

```
>_ grep --color=auto -iRnH "$search_word" $directory
```

Unix alias for date command that lets you create timestamps in ISO 8601 format

```
>_ alias timestamp='date "+%Y%m%dT%H%M%S"'
```

Another Curl your IP command

```
>_ curl -s http://checkip.dyndns.org | sed 's/[a-zA-Z<>/ :]//g'
```

Count IPv4 connections per IP

```
>_ netstat -anp |grep 'tcp\|udp' | awk '{print $5}' | sed s/::ffff:// | cut -d: -f1 | sort | uniq -c | sort -n
```

Files extension change

```
>_ rename .oldextension .newextension *.oldextension
```

Archive all files containing local changes (svn)

```
>_ svn st | cut -c 8- | sed 's/^/\"/;s/$/\"/' | xargs tar -czvf ../backup.tgz
```

Just run it ;)

```
>_ echo SSBMb3ZlIFlvdQo= | base64 -d
```

Pattern match in awk - no grep

```
>_ awk '/pattern1/ && /pattern2/ && !/pattern3/ {print}'
```

Optimal way of deleting huge numbers of files

```
>_ find /path/to/dir -type f -print0 | xargs -0 rm
```

Google text-to-speech in mp3 format

```
>_ t2s() { wget -q -U Mozilla -O $(tr ' ' _ <<< "$1"| cut -b 1-15).mp3 "http://translate.google.com/translate_tts?ie=UTF-8&tl=en&q=$(tr ' ' + <<< "$1")"; }
```

Google verbatim search on your terminal

```
>_ function google { Q="$@"; GOOG_URL='https://www.google.de/search?tbs=li:1&q='; AGENT="Mozilla/4.0"; stream=$(curl -A "$AGENT" -skLm 10 "${GOOG_URL}${Q//\ /+}" | grep -oP '\/url\?q=.+?&amp' | sed 's|/url?q=||; s|&amp||'); echo -e "${stream//\%/\x}"; }
```

Grep processes list avoiding the grep itself

```
>_ ps axu | grep [a]pache2
```

Swap the two last arguments of the current command line

```
>_ <ctrl+e> <esc+t>
```

Transfer a file to multiple hosts over ssh

```
>_ pscp -h hosts.txt -l username
/etc/hosts /tmp/hosts
```

Tail a log over ssh

```
>_ ssh -t remotebox "tail -f /var/log
/remote.log"
```

Du with colored bar graph

```
>_ t=$(df| awk
'NR!=1{sum+=$2}END{print
sum}');sudo du / --max-depth=1|sed
'$d'|sort -rn -k1 | awk -v t=$t
'OFMT="%d" {M=64; for (a=0;a<$1;
a++){if (a>c){c=a}}br=a/c;b=M*br;
for(x=0;x<b;x++){printf "\033[1;31m"
"|" "\033[0m"}print " "$2" "(a/t*100)"%
total"}'
```

Sort the output of the 'du' command by largest first, using human readable output.

```
>_ du -h --max-depth=1 |sort -rh
```

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
```

Query well known ports list

```
>_ getent services <<service>>
```

Create .pdf from .doc

```
>_ oowriter -pt pdf your_word_file.doc
```

Diff XML files

```
>_ diffxml() { diff -wb <(xmllint --format
"$1") <(xmllint --format "$2"); }
```

Discovering all open files/dirs underneath a directory

```
>_ lsof +D <dirname>
```

Clear screen, keep prompt at eye-level (faster than clear(1), tput cl, etc.)

```
>_ cls(){ printf "\33[2J";} or, if no
printf, cat >cls;<ctrl-v><ctrl+
[>[2J<enter><ctrl+d> cls(){ cat cls;}
```

Down for everyone or just me?

```
>_ down4me() { wget -qO -
"http://www.downforeveryoneorjustme.
com/$1" | sed '/just you/!d;s
/<[^>]*>//g' ; }
```

A formatting test for David Winterbottom: improving commandlinefu for submitters

```
>_ echo "?????, these are the umlauted vowels I sing to you. Oh, and sometimes ?, but I don't sing that one cause it doesn't rhyme."
```

Run a program transparently, but print a stack trace if it fails

```
>_ gdb -batch -ex "run" -ex "bt" ${my_program} 2>&1 | grep -v ^"No stack."$
```

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'
```

Continue a current job in the background

```
>_ <ctrl+z> bg
```

Get all links of a website

```
>_ lynx -dump http://www.domain.com | awk '/http/{print $2}'
```

Find all active ip's in a subnet

```
>_ sudo arp-scan -I eth0 192.168.1.0/24
```

Display BIOS Information

```
>_ dmidecode -t bios
```

Disassemble some shell code

```
>_ echo -ne "<shellcode>" | x86dis -e 0 -s intel
```

List bash functions defined in .bash_profile or .bashrc

```
>_ compgen -A function
```

Ignore the .svn directory in filename completion

```
>_ export FIGNORE=.svn
```

Working random fact generator

```
>_ wget randomfunfacts.com -O - 2>/dev/null | grep \<strong\> | sed "s;^.*<i>\(.*\)</i>.*$;\1;"
```

Remote backups with tar over ssh

```
>_ tar jcpf - [sourceDirs] |ssh
user@host "cat > /path/to/backup
/backupfile. tar.bz2"
```

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 -; }
```

Change pidgin status

```
>_ purple-remote
"setstatus?status=away&
message=AFK"
```

Grep by paragraph instead of by line.

```
>_ grepp() { [ $# -eq 1 ] && perl -00ne
"print if /$1/i" || perl -00ne "print if
/$1/i" < "$2";}
```

Vim: Switch from Horizontal split to Vertical split

```
>_ ^W-L
```

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";
```

Sort all running processes by their memory & CPU usage

```
>_ ps aux --sort=%mem,%cpu
```

Change user, assume environment, stay in current dir

```
>_ su -- user
```

Save current layout of top

```
>_ <Shift + W>
```

Function that outputs dots every second until command completes

```
>_ sleeper(){ while `ps -p $1 &>/dev
/null`; do echo -n "${2:-.}"; sleep
${3:-1}; done; }; export -f sleeper
```

Testing hard disk reading speed

```
>_ hdparm -t /dev/sda
```

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
```

An easter egg built into python to give you the Zen of Python

```
>_ python -c 'import this'
```

Get all IPs via ifconfig

```
>_ ifconfig -a | perl -nle'/(\d+\.\d+\.
\d+\.\d+)/ && print $1'
```

Use Cygwin to talk to the Windows clipboard

```
>_ cat /dev/clipboard; $(somecommand)
> /dev/clipboard
```

List programs with open ports and connections

```
>_ lsof -i
```

Find corrupted jpeg image files

```
>_ find . -name "*jpg" -exec jpeginfo -c
{} \; | grep -E "WARNING|ERROR"
```

'Fix' a typescript file created by the 'script' program to remove control characters

```
>_ cat typescript | perl -pe 's/\e([^\[\]]|
\[.*?[a-zA-Z]|\].*?\a)//g' | col -b >
typescript-processed
```

Share a 'screen'-session

```
>_ screen -x
```

Purge configuration files of removed packages on debian based systems

```
>_ sudo aptitude purge `dpkg --get-
selections | grep deinstall | awk '{print
$1}'`
```

Show all detected mountable Drives/Partitions /BlockDevices

```
>_ hwinfo --block --short
```

Merge *.pdf files

```
>_ gs -q -sPAPERSIZE=letter
-dNOPAUSE -dBATCH
-sDEVICE=pdfwrite
-sOutputFile=out.pdf `ls *.pdf`
```

Append stdout and stderr to a file, and print stderr to the screen [bash]

```
>_ somecommand 2>&1 >> logfile | tee -a logfile
```

Quickly change all .html extensions on files in folder to .htm

```
>_ for i in *.html ; do mv $i ${i%.html}.htm ; done
```

Delete the specified line

```
>_ sed -i 8d ~/.ssh/known_hosts
```

Read the output of a command into the buffer in vim

```
>_ :r !command
```

Find broken symlinks

```
>_ find . -type l ! -exec test -e {} \; -print
```

Ssh tunnel with auto reconnect ability

```
>_ while [ ! -f /tmp/stop ]; do ssh -o ExitOnForwardFailure=yes -R 2222:localhost:22 target "while nc -zv localhost 2222; do sleep 5; done"; sleep 5;done
```

Find process associated with a port

```
>_ fuser [portnumber]/[proto]
```

Scping files with streamlines compression (tar gzip)

```
>_ tar czv file1 file2 folder1 | ssh user@server tar zxv -C /destination
```

Add all files not under version control to repository

```
>_ svn status |grep '\?' |awk '{print $2}'| xargs svn add
```

Determine if a port is open with bash

```
>_ : </dev/tcp/127.0.0.1/80
```

Search for a process by name

```
>_ ps -fC PROCESSNAME
```

Download all mp3's listed in an html page

```
>_ wget -r -l1 -H -t1 -nd -N -np -A.mp3 -erobots=off [url of website]
```

Run the built in PHP-server in current folder

```
>_ php -S 127.0.0.1:8080
```

Press enter and take a WebCam picture.

```
>_ read && ffmpeg -y -r 1 -t 3 -f
video4linux2 -vframes 1 -s sxga -i
/dev/video0 ~/webcam-$(date
+%m_%d_%Y_%H_%M).jpeg
```

Commandline document conversion with Libreoffice

```
>_ soffice --headless -convert-to
odt:"writer8" somefile.docx
```

HourGlass

```
>_ hourglass(){ trap 'tput cnorm' 0 1 2
15 RETURN;local s=$(($SECONDS
+$1));(tput civis;while
(($SECONDS<$s));do for f in '|' '\' '-'
'/';do echo -n "$f";sleep .2s; echo -n
$'\b';done;done;);}
```

Gets the english pronunciation of a phrase

```
>_ say() { mplayer
"http://translate.google.com
/translate_tts?q=$1"; }
```

Check syntax for all PHP files in the current directory and all subdirectories

```
>_ find . -name \*.php -exec php -l "{}"
\;
```

Ask for a password, the passwd-style

```
>_ read -s -p"Password: "
USER_PASSWORD_VARIABLE; echo
```

Monitor memory usage

```
>_ watch vmstat -sSM
```

Content search.

```
>_ ff() { local IFS='|'; grep -rinE "$*" . ;
}
```

Timer with sound alarm

```
>_ sleep 3s && espeak "wake up, you
bastard" 2>/dev/null
```

Rapidly invoke an editor to write a long, complex, or tricky command

```
>_ <ESC> v
```

Run command on a group of nodes in parallel

```
>_ echo "uptime" | tee >(ssh host1)
>(ssh host2) >(ssh host3)
```

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')
```

Kill process by name

```
>_ pkill -x firefox
```

Change prompt to MS-DOS one (joke)

```
>_ export PS1="C:\$( pwd | sed
's:/:\\\\\\:g' )\\> "
```

Make ISO image of a folder

```
>_ mkisofs -J -allow-lowercase -R -V
"OpenCD8806" -iso-level 4 -o
OpenCD.iso ~/OpenCD
```

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; }
```

Typing the current date (or any string) via a shortcut as if the keys had been actually typed with the hardware keyboard in any application.

```
>_ xvkbd -xsendevent -text $(date
+%Y%m%d)
```

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
```

Quickest (i blv) way to get the current program name minus the path (BASH)

```
>_
path_stripped_programname="${0##*/
}"
```

A trash function for bash

```
>_ trash <file>
```

Clean your broken terminal

```
>_ stty sane
```

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 | sha1sum | sha224sum | sha256sum | sha384sum | sha512sum | gzip - | strings -n 1 | tr -d "[:space:]" | tr -s '[:print:]' | tr '!-~' 'P-~!-O' | rev | cut -b 2-11; history -d $(($HISTCMD-1)); }
```

Search back through previous commands

```
>_ Ctrl-R <search-text>
```

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')
```

Replace spaces in filenames with underscores

```
>_ rename -v 's/ /_/g' *
```

Show directories in the PATH, one per line

```
>_ echo "${PATH//:/$'\n'}"
```

Move a lot of files over ssh

```
>_ rsync -az /home/user/test user@sshServer:/tmp/
```

Find and delete empty dirs, start in current working dir

```
>_ find . -empty -type d -exec rmdir {} +
```

List your largest installed packages.

```
>_ wajig large
```

Monitor TCP opened connections

```
>_ watch -n 1 "netstat -tpanl | grep ESTABLISHED"
```

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
```

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
```

Fibonacci numbers with awk

```
>_ seq 50| awk 'BEGIN {a=1; b=1}
{print a; c=a+b; a=b; b=c}'
```

Create a favicon

```
>_ convert -colors 256 -resize 16x16
face.jpg face.ppm && ppmtowinicon
-output favicon.ico face.ppm
```

Check Ram Speed and Type in Linux

```
>_ sudo dmidecode --type 17 | more
```

Run the Firefox Profile Manager

```
>_ firefox -no-remote -P
```

Add forgotten changes to the last git commit

```
>_ git commit --amend
```

Sort dotted quads

```
>_ sort -nt . -k 1,1 -k 2,2 -k 3,3 -k 4,4
```

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
```

Add your public SSH key to a server in one command

```
>_ cat . ssh/id_rsa.pub | ssh hostname '
cat >> . ssh/authorized_keys'
```

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
```

Read and write to TCP or UDP sockets with common bash tools

```
>_ exec 5<>/dev/tcp/time.nist.gov/13;
cat <&5 & cat >&5; exec 5>&-
```

Use top to monitor only all processes with the same name fragment 'foo'

```
>_ top -p $(pgrep -d , foo)
```

Delete command line last word

```
>_ ctrl+w
```

Draw kernel module dependancy graph.

```
>_ lsmod | awk 'BEGIN{ print
"digraph{"}{split($4, a, ","); for (i in a)
print $1, "->", a[i]}END{ print
"}"}'|display
```

Color man pages

```
>_ echo "export
LESS_TERMCAP_mb=$'\E[01;31m'" >>
~/.bashrc
```

Print without executing the last command that starts with...

```
>_ !ssh:p
```

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
```

Protect directory from an overzealous rm -rf *

```
>_ cd <directory>; touch ./-i
```

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
```

List 10 largest directories in current directory

```
>_ du -hs */ | sort -hr | head
```

Reuse last parameter

```
>_ !$
```

Blink LED Port of NIC Card

```
>_ ethtool -p eth0
```

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;
done | dialog --gauge "Install part $j :
`sed $(perl -e "print int
rand(99999)")"q;d" /usr/share
/dict/words`" 6 40;done
```

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")
```

Generate a Random MAC address

```
>_ openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/.$//'
```

Running scripts after a reboot for non-root users .

```
>_ @reboot <yourscript.sh>
```

Run command on a group of nodes in parallel

```
>_ echo "uptime" | pee "ssh host1" "ssh host2" "ssh host3"
```

Make, or run a script, everytime a file in a directory is modified

```
>_ while true; do inotifywait -r -e MODIFY dir/ && make; done;
```

Choose from a nice graphical menu which DI.FM radio station to play

```
>_ zenity --list --width 500 --height 500 --column 'radio' --column 'url' --print-column 2 $(curl -s http://www.di.fm/ | awk -F '"' '/href="http:.*\.pls.*96k/ {print $2}' | sort | awk -F '/|\.' '{print $(NF-1) " " $0}') | xargs mplayer
```

Check the status of 'dd' in progress (OS X)

```
>_ killall -INFO dd
```

Ping scanning without nmap

```
>_ for i in {1..254}; do ping -c 1 -W 1 10.1.1.$i | grep 'from'; done
```

Convert a Nero Image File to ISO

```
>_ dd bs=1k if=image.nrg of=image.iso skip=300
```

View network activity of any application or user in realtime

```
>_ lsof -r 2 -p PID -i -a
```

Copy with progress

```
>_ rsync --progress file1 file2
```

Print a list of standard error codes and descriptions.

```
>_ perl -le 'print $!+0, "\t", $!++ for 0..127'
```

Recursive search and replace old with new string, inside files

```
>_ find . -type f -exec sed -i s/oldstring/newstring/g {} +
```

Generate random password

```
>_ pwgen -Bs 10 1
```

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; }
```

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;done
```

Show directories in the PATH, one per line

```
>_ echo $PATH | tr \: \\n
```

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
```

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

```
>_ tail -F file
```

Convert "man page" to text file

```
>_ man ls | col -b > ~/Desktop/man_ls.txt
```

Get your public ip using dyndns

```
>_ curl -s http://checkip.dyndns.org/ | grep -o "[[:digit:].]\+"
```

Display current bandwidth statistics

```
>_ ifstat -nt
```

Given process ID print its environment variables

```
>_ sed 's/\o0/\n/g' /proc/INSERT_PID_HERE/environ
```

Restoring some data from a corrupted text file

```
>_ ( cat badfile.log ; tac badfile.log | tac ) > goodfile.log
```

View the system console remotely

```
>_ sudo cat /dev/vcs1 | fold -w 80
```

Look up the definition of a word

```
>_ curl dict://dict.org/d:something
```

Diff files on two remote hosts.

```
>_ diff <(ssh alice cat /etc/apt/sources.list) <(ssh bob cat /etc/apt/sources.list)
```

Ctrl+S Ctrl+Q terminal output lock and unlock

```
>_ Ctrl+S Ctrl+Q
```

Redirect STDIN

```
>_ < /path/to/file.txt grep foo
```

I hate `echo X | Y`

```
>_ base64 -d <<< aHR0cDovL3d3dy50d2l0dGVyc2hlZXAuY29tL3Jlc3VsdHMucGhwP3U9Y29tbWFuZGxpbmVmdQo=
```

Send keypresses to an X application

```
>_ xvkbd -xsendevent -text "Hello world"
```

Add calendar to desktop wallpaper

```
>_ convert -font -misc-fixed-*-*-*-*-*-*-*-*-*-*-*-* -fill black -draw "text 270,260 \" `cal` \"" testpic.jpg newtestpic.jpg
```

Browse system RAM in a human readable form

```
>_ sudo cat /proc/kcore | strings | awk 'length > 20' | less
```

Backup your hard drive with dd

```
>_ sudo dd if=/dev/sda of=/media/disk/backup/sda.backup
```

Calculates the date 2 weeks ago from Saturday the specified format.

```
>_ date -d '2 weeks ago Saturday' +%Y-%m-%d
```

Get Cisco network information

```
>_ tcpdump -nn -v -i eth0 -s 1500 -c 1 'ether[20:2] == 0x2000'
```

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
```

Extract audio from a video

```
>_ ffmpeg -i video.avi -f mp3 audio.mp3
```

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
```

Use last argument of last command

```
>_ file !$
```

Show which programs are listening on TCP and UDP ports

```
>_ netstat -plunt
```

Random git-commit message

```
>_ git commit -m "$(curl -s http://whatthecommit.com/index.txt)";
```

Generate QR code for a WiFi hotspot

```
>_ qrencode -s 7 -o qr-wifi.png "WIFI:S:$(zenity --entry --text="Network name (SSID)" --title="Create WiFi QR");T:WPA;P:$(zenity --password --title="Wifi Password");;"
```

Check if system is 32bit or 64bit

```
>_ arch
```

Vi a remote file

```
>_ vi scp://username@host//path/to/somefile
```

Show permissions of current directory and all directories upwards to /

```
>_ namei -m $(pwd)
```

Move you up one directory quickly

```
>_ shopt -s autocd
```

Hide the name of a process listed in the `ps` output

```
>_ exec -a "/sbin/getty 38400 tty7" your_cmd -erase_all_files
```

Remove a line from a file using sed (useful for updating known SSH server keys when they change)

```
>_ ssh-keygen -R <thehost>
```

Show what PID is listening on port 80 on Linux

```
>_ fuser -v 80/tcp
```

List by size all of the directories in a given tree.

```
>_ du -h /path | sort -h
```

Find the most recently changed files (recursively)

```
>_ find . -type f -printf '%TY-%Tm-%Td
%TT %p\n' | sort
```

All IP connected to my host

```
>_ netstat -lantp | grep ESTABLISHED |
awk '{print $5}' | awk -F: '{print $1}' |
sort -u
```

Figure out what shell you're running

```
>_ echo $0
```

Download a file and uncompress it while it downloads

```
>_ wget http://URL/FILE. tar.gz -O - |
tar xfz -
```

Rename all .jpeg and .JPG files to have .jpg extension

```
>_ rename 's/\.jpe?g$/.jpg/i' *
```

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
```

Display BIOS Information

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

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
```

Use the previous commands params in the current command

```
>_ !!:[position]
```

Mirror a directory structure from websites with an Apache-generated file indexes

```
>_ lftp -e "mirror -c" http://example.com /foobar/
```

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 /samedir/tocopyto/ && tar -xf -"
```

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
```

Limit bandwidth usage by apt-get

```
>_ sudo apt-get -o Acquire::http::Dl-Limit=30 upgrade
```

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
```

Grep -v with multiple patterns.

```
>_ grep 'test' somefile | grep -vE '(error|critical|warning)'
```

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/stdoutlog
```

Identify long lines in a file

```
>_ awk 'length>72' file
```

Use all the cores or CPUs when compiling

```
>_ make -j 4
```

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
```

Analyze awk fields

```
>_ awk '{print NR": "$0; for(i=1;
i<=NF;++i)print "\t"i": "$i}'
```

Colored SVN diff

```
>_ svn diff <file> | vim -R -
```

Run a command, store the output in a pastebin on the internet and place the URL on the xclipboard

```
>_ ls | curl -F 'sprunge=<-'
http://sprunge.us | xclip
```

Show git branches by date - useful for showing active branches

```
>_ for k in `git branch|perl -pe
s/^..//`;do echo -e `git show
--pretty=format:"%Cgreen%ci %Cblue
%cr%Creset" $k|head -n
1`\\t$k;done|sort -r
```

Find if the command has an alias

```
>_ type -all command
```

Find last reboot time

```
>_ who -b
```

Show a config file without comments

```
>_ egrep -v "^$|^[[:space:]]*#"
/etc/some/file
```

Start screen in detached mode

```
>_ screen -d -m [<command>]
```

Iso-8859-1 to utf-8 safe recursive rename

```
>_ detox -r -s utf_8 /path/to/old/win
/files/dir
```

Create dir tree

```
>_ mkdir -p doc/{text/,img/{wallpaper
/,photos/}}
```

Run any GUI program remotely

```
>_ ssh -fX <user>@<host> <program>
```

Delete all empty lines from a file with vim

```
>_ :g!/\S/d
```

List the files any process is using

```
>_ lsof +p xxxx
```

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}'
```

Change directory to actual path instead of symlink path

```
>_ cd `pwd -P`
```

Download an entire static website to your local machine

```
>_ wget --recursive --page-requisites
--convert-links www.moyagraphix.co.za
```

Recursively remove .svn directories from the current location

```
>_ find . -type d -name '.svn' -print0 |
xargs -0 rm -rdf
```

Count files beneath current directory (including subfolders)

```
>_ find . -type f | wc -l
```

Convert from hexidecimal or octal to decimal

```
>_ echo $((0x1fe)) $((033))
```

Save man-page as pdf

```
>_ man -t awk | ps2pdf - awk.pdf
```

Convert seconds into minutes and seconds

```
>_ echo 'obase=60;299' | bc
```

List files accessed by a command

```
>_ strace -ff -e trace=file my_command
2>&1 | perl -ne 's/^[^"]+"(([^\\"]|
\\[\\"nt])*)".*/$1/ && print'
```

Short and elegant way to backup a single file before you change it.

```
>_ cp httpd.conf{,.bk}
```

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 -i lname
"/tmp/Flash*" 2>/dev/null`; do ln -s "$h"
`readlink "$h" | cut -d' ' -f1`; done
```

Rsync instead of scp

```
>_ rsync --progress --partial --rsh="ssh
-p 8322" --bwlimit=100 --ipv4
user@domain.com:~/file.tgz .
```

Send a circular

```
>_ echo "dear admin, please ban
eribsskog" | wall
```

Visit wikileaks.com

```
>_ echo 213.251.145.96 wikileaks.com
>>/etc/hosts
```

Make sudo forget password instantly

```
>_ sudo -K
```

Print a row of characters across the terminal

```
>_ printf "%`tput cols`s"|tr ' ' '#'
```

Download and unpack tarball without leaving it sitting on your hard drive

```
>_ wget -qO - 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
```

Split a tarball into multiple parts

```
>_ tar cf - <dir>|split -b<max_size>M -
<name>. tar.
```

Remove executable bit from all files in the current directory recursively, excluding other directories

```
>_ chmod -R -x+X *
```

More precise BASH debugging

```
>_ env PS4='
${BASH_SOURCE}:${LINENO}
(${FUNCNAME[0]}) ' sh -x /etc/profile
```

Get all pdf and zips from a website using wget

```
>_ wget --reject html,htm --accept
pdf,zip -rl1 url
```

Mount the first NTFS partition inside a VDI file (VirtualBox Disk Image)

```
>_ mount -t ntfs-3g -o
ro,loop,uid=user,gid=group,umask=00
07,fmask=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
```

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 |
awk '{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
```

Create a list of binary numbers

```
>_ echo {0..1}{0..1}{0..1}{0..1}
```

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'
```

Disable history for current shell session

```
>_ unset HISTFILE
```

Search for commands from the command line

```
>_ clfu-seach <search words>
```

How to run X without any graphics hardware

```
>_ startx -- `which Xvfb` :1 -screen 0
800x600x24 && DISPLAY=:1 x11vnc
```

Install a Firefox add-on/theme to all users

```
>_ sudo firefox -install-global-extension
/path/to/add-on
```

Clear current line

```
>_ CTRL+u
```

Quickly generate an MD5 hash for a text string using OpenSSL

```
>_ echo -n 'text to be encrypted' |
openssl md5
```

Terminate a frozen SSH-session

```
>_ RETURN~.
```

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
```

Generat a Random MAC address

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

Batch convert files to utf-8

```
>_ find . -name "*.php" -exec iconv -f ISO-8859-1 -t UTF-8 {} -o ../newdir_utf8/{} \;
```

Get http headers for an url

```
>_ curl -I www.commandlinefu.com
```

Return external ip

```
>_ curl ipinfo.io
```

Simple multi-user encrypted chat server for 5 users

```
>_ ncat -vlm 5 --ssl --chat 9876
```

Check if your ISP is intercepting DNS queries

```
>_ dig +short which.opendns.com txt @208.67.220.220
```

Remove a range of lines from a file

```
>_ sed -i <file> -re '<start>,<end>d'
```

Extract audio from Flash video (*.flv) as mp3 file

```
>_ ffmpeg -i video.flv -vn -ar 44100 -ac 2 -ab 192k -f mp3 audio.mp3
```

Cat a bunch of small files with file indication

```
>_ grep . *
```

Stop Flash from tracking everything you do.

```
>_ for i in ~/.adobe ~/.macromedia ; do ( rm $i/ -rf ; ln -s /dev/null $i ) ; done
```

List all open ports and their owning executables

```
>_ lsof -i -P | grep -i "listen"
```

Purge configuration files of removed packages on debian based systems

```
>_ aptitude purge '~c'
```

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"
```

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
-bordercolor white -border 3 text.gif
```

Remote screenshot

```
>_ DISPLAY=":0.0" import -window
root screenshot.png
```

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;}
```

List all authors of a particular git project

```
>_ git log --format='%aN' | sort -u
```

Track flights from the command line

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

Harder, Faster, Stronger SSH clients

```
>_ ssh -4 -C -c blowfish-cbc
```

Clean up poorly named TV shows.

```
>_ rename -v 's/.*[s,S](\d{2}).*
[e,E](\d{2}).*\.avi/SHOWNAME\
S$1E$2.avi/'
poorly.named.file.s01e01.avi
```

Pretty Print a simple csv in the command line

```
>_ column -s, -t <tmp.csv
```

Cleanup firefox's database.

```
>_ find ~/.mozilla/firefox/ -type f -name
"*.sqlite" -exec sqlite3 {} VACUUM \;
```

Check open ports

```
>_ lsof -Pni4 | grep LISTEN
```

Find Duplicate Files (based on MD5 hash)

```
>_ find -type f -exec md5sum '{}' ';' |
sort | uniq --all-repeated=separate -w
33 | cut -c 35-
```

Triple monitoring in screen

```
>_ tmpfile=$(mktemp) && echo -e
'startup_message off\nscreen -t top h
top\nsplit\nfocus\nscreen -t nethogs
nethogs wlan0\nsplit\nfocus\nscreen -t
io top io top' > $tmpfile && sudo
screen -c $tmpfile
```

Smiley Face Bash Prompt

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

Save an HTML page, and covert it to a .pdf file

```
>_ wget $URL | htmldoc --webpage -f
"$URL".pdf - ; xpdf "$URL".pdf &
```

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; sudo swapon
/swapfile
```

Relocate a file or directory, but keep it accessible on the old location throug a simlink.

```
>_ mv $1 $2 && ln -s $2/$(basename
$1) $(dirname $1)
```

A short counter

```
>_ yes '' | cat -n
```

Rsync remote data as root using sudo

```
>_ rsync -- rsync-path 'sudo rsync'
username@source:/folder/ /local/
```

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=dbname &
```

Pipe STDOUT to vim

```
>_ tail -1000 /some/file | vim -
```

Copy stdin to your X11 buffer

```
>_ ssh user@host cat /path/to/some/file
| xclip
```

Get info about remote host ports and OS detection

```
>_ nmap -sS -P0 -sV -O <target>
```

Copy a file structure without files

```
>_ find * -type d -exec mkdir /where /you/wantem/\{\} \;
```

Start a new command in a new screen window

```
>_ alias s='screen -X screen'; s top; s vi; s man ls;
```

Notepad in a browser (type this in the URL bar)

```
>_ data:text/html, <html contenteditable>
```

Display current time in requested time zones.

```
>_ zdump Japan America/New_York
```

Dmesg with colored human-readable dates

```
>_ dmesg -T|sed -e 's|\(^.*'`date +%Y`']\)\(.*\)|\x1b[0;34m\1\x1b[0m - \2|g'
```

Stamp a text line on top of the pdf pages.

```
>_ echo "This text gets stamped on the top of the pdf pages." | enscript -B -f Courier-Bold16 -o- | ps2pdf - | pdftk input.pdf stamp - output output.pdf
```

Print diagram of user/groups

```
>_ awk 'BEGIN{FS=":"; print "digraph{"}{split($4, a, ","); for (i in a) printf "\"%s\" [shape=box]\n\"%s\" -> \"%s\"\n", $1, a[i], $1}END{ print "}"}' /etc/group|display
```

Create a file server, listening in port 7000

```
>_ while true; do nc -l 7000 | tar -xvf -; done
```

Bypass any aliases and functions for the command

```
>_ \foo
```

Share your terminal session real-time

```
>_ mkfifo foo; script -f foo
```

Df without line wrap on long FS name

```
>_ df -P | column -t
```

Stderr in color

```
>_ mycommand 2> >(while read line; do echo -e "\e[01;31m$line\e[0m"; done)
```

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 !chmod +x <afile> | endif | endif
```

Create a single PDF from multiple images with ImageMagick

```
>_ convert *.jpg output.pdf
```

Dd with progress bar and statistics

```
>_ sudo dd if=/dev/sdc bs=4096 | pv -s 2G | sudo dd bs=4096 of=~/USB_BLACK_BACKUP.IMG
```

Delete all empty lines from a file with vim

```
>_ :g/^$/d
```

Perl one-liner to get the current week number

```
>_ date +%V
```

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
```

Ultimate current directory usage command

```
>_ ncdu
```

Terminal - Show directories in the PATH, one per line with sed and bash3.X `here string'

```
>_ tr : '\n' <<<$PATH
```

Have an ssh session open forever

```
>_ autossh -M50000 -t server.example.com 'screen -raAd mysession'
```

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"."\x00" if ($h eq $ph)'|xargs -0 rm -v --
```

Find broken symlinks and delete them

```
>_ find -L /path/to/check -type l -delete
```

List recorded formular fields of Firefox

```
>_ cd ~/.mozilla/firefox/ && sqlite3 `cat
profiles.ini | grep Path | awk -F= '{print
$2}'`/formhistory.sqlite "select * from
moz_formhistory" && cd - > /dev/null
```

Using awk to sum/count a column of numbers.

```
>_ cat count.txt | awk '{ sum+=$1}
END {print sum}'
```

Get all the keyboard shortcuts in screen

```
>_ ^A ?
```

Get list of servers with a specific port open

```
>_ nmap -sT -p 80 -oG - 192.168.1.* |
grep open
```

Silently ensures that a FS is mounted on the given mount point (checks if it's OK, otherwise unmount, create dir and mount)

```
>_ ( mountpoint -q "/media/mpdr1" &&
df /media/mpdr1/* > /dev/null 2>&1) ||
(( sudo u mount "/media/mpdr1" >
/dev/null 2>&1 || true) && ( sudo mkdir
"/media/mpdr1" > /dev/null 2>&1 ||
true) && sudo mount "/dev/sdd1"
"/media/mpdr1")
```

Sniff network traffic on a given interface and displays the IP addresses of the machines communicating with the current host (one IP per line)

```
>_ sudo tcpdump -i wlan0 -n ip | awk '{
print gensub(/(.*)\..*/,"\\1","g",$3), $4,
gensub(/(.*)\..*/,"\\1","g",$5) }' | awk -F
" > " '{ print $1"\n"$2}'
```

Check if system is 32bit or 64bit

```
>_ getconf LONG_BIT
```

Create a local compressed tarball from remote host directory

```
>_ ssh user@host "tar -zcf -
/path/to/dir" > dir.tar.gz
```

Rapidly invoke an editor to write a long, complex, or tricky command

```
>_ <ESC> v or ctrl-x ctrl-e
```

Send a circular

```
>_ wall <<< "Broadcast This"
```

The BOFH Excuse Server

```
>_ telnet towel.blinkenlights.nl 666
```

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
```

Bash: hotkey to put current commandline to text-editor

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

Show current working directory of a process

```
>_ pwdx pid
```

Base conversions with bc

```
>_ echo "obase=2; 27" | bc -l
```

Put readline into vi mode

```
>_ set -o vi
```

Transfer SSH public key to another machine in one step

```
>_ ssh-keygen; ssh-copy-id user@host; ssh user@host
```

Start a command on only one CPU core

```
>_ taskset -c 0 your_command
```

Convert uppercase files to lowercase files

```
>_ rename 'y/A-Z/a-z/' *
```

List of commands you use most often

```
>_ history | awk '{print $2}' | sort | uniq -c | sort -rn | head
```

Prevent large files from being cached in memory (backups!)

```
>_ nocache <I/O-heavy-command>
```

Convert single digit to double digits

```
>_ for i in ?.ogg; do mv $i 0$i; done
```

Limit the cpu usage of a process

```
>_ sudo cpulimit -p pid -l 50
```

Generate an XKCD #936 style 4 word password

```
>_ shuf -n4 /usr/share/dict/words | tr -d
'\n'
```

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
```

Diff remote webpages using wget

```
>_ diff <(wget -q -O - URL1) <(wget -q
-O - URL2)
```

List alive hosts in specific subnet

```
>_ nmap -sP 192.168.1.0/24
```

Processes per user counter

```
>_ ps hax -o user | sort | uniq -c
```

View all date formats, Quick Reference Help Alias

```
>_ alias dateh=' date --help|sed -n "/^
*%%/,/^ *%Z/p"|while read l;do
F=${l/% */}; date +%$F:"|'"'"'${F//%n/
}'"'"'|${l#* }";done|sed "s/\ *|\ */|/g"
|column -s "|" -t'
```

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 140
```

Make sure a script is run in a terminal.

```
>_ [ -t 0 ] || exit 1
```

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="";
done
```

Quickly (soft-)reboot skipping hardware checks

```
>_ /sbin/kexec -l /boot/$KERNEL
--append="$KERNELPARAMTERS"
--initrd=/boot/$INITRD; sync;
/sbin/kexec -e
```

Pipe output of a command to your clipboard

```
>_ some command|xsel --clipboard
```

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

```
>_ ls -hog
```

Git remove files which have been deleted

```
>_ git rm $( git ls-files --deleted)
```

Efficiently print a line deep in a huge log file

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

Annotate tail -f with timestamps

```
>_ tail -f file | while read; do echo "$(date +%T.%N) $REPLY"; done
```

Fast, built-in pipe-based data sink

```
>_ <COMMAND> |:
```

Repoint an existing symlink to a new location

```
>_ ln -nsf <TARGET> <LINK>
```

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
```

Find Duplicate Files (based on size first, then MD5 hash)

```
>_ fdupes -r .
```

Close a hanging ssh session

```
>_ ~.
```

Recover a deleted file

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

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'
```

Convert filenames in current directory to lowercase

```
>_ rename 'y/A-Z/a-z/' *
```

Find files that were modified by a given command

```
>_ touch /tmp/file ; $EXECUTECOMMAND ; find /path -newer /tmp/file
```

Quickly graph a list of numbers

```
>_ gnuplot -persist <(echo "plot '<(sort -n listOfNumbers.txt)' with lines")
```

Cut out a piece of film from a file. Choose an arbitrary length and starting time.

```
>_ ffmpeg -vcodec copy -acodec copy -i orginalfile -ss 00:01:30 -t 0:0:20 newfile
```

Analyse an Apache access log for the most common IP addresses

```
>_ tail -10000 access_log | awk '{print $1}' | sort | uniq -c | sort -n | tail
```

Format txt as table not joining empty columns

```
>_ column -tns: /etc/passwd
```

Shell recorder with replay

```
>_ script -t /tmp/mylog.out 2>/tmp/mylog.time; <do your work>; <CTRL-D>; scriptreplay /tmp/mylog.time /tmp/mylog.out
```

Bind a key with a command

```
>_ bind '"\C-l":"ls -l\n"'
```

Exclude a column with cut

```
>_ cut -f5 --complement
```

List files with quotes around each filename

```
>_ ls -Q
```

Duplicate several drives concurrently

```
>_ dd if=/dev/sda | tee >( dd of=/dev/sdb) | dd of=/dev/sdc
```

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
```

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";}
```

Get your external IP address

```
>_ curl ip.appspot.com
```

List installed deb packages by size

```
>_ dpkg-query -Wf '${Installed-Size}\t${Package}\n' | sort -n
```

Monitor bandwidth by pid

```
>_ nethogs -p eth0
```

Execute a command with a timeout

```
>_ timeout 10 sleep 11
```

Send echo to socket network

```
>_ echo "foo" > /dev/tcp/192.168.1.2/25
```

Use vim to get colorful diff output

```
>_ svn diff | view -
```

Find files containing text

```
>_ grep -lir "some text" *
```

Perform a branching conditional

```
>_ true && { echo success;} || { echo failed; }
```

Resume scp of a big file

```
>_ rsync --partial --progress --rsh=ssh $file_source $user@$host:$destination_file
```

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"}'
```

Use tee + process substitution to split STDOUT to multiple commands

```
>_ some_command | tee >(command1) >(command2) >(command3) ... | command4
```

Encrypted archive with openssl and tar

```
>_ tar --create --file - --posix --gzip -- <dir> | openssl enc -e -aes256 -out <file>
```

Create strong, but easy to remember password

```
>_ read -s pass; echo $pass | md5sum | base64 | cut -c -16
```

Use file(1) to view device information

```
>_ file -s /dev/sd*
```

Prevent accidents while using wildcards

```
>_ rm *.txt <TAB> <TAB>
```

Insert the last argument of the previous command

```
>_ <ESC> .
```

View the newest xkcd comic.

```
>_ xkcd(){ wget -qO- http://xkcd.com/|tee >(feh $(grep -Po '(?<=")http://imgs[^/]+/comics/[^"]+\.\w{3}'))|grep -Po '(?<=(\w{3})" title=").*(?=" alt)';}
```

Press Any Key to Continue

```
>_ read -sn 1 -p "Press any key to continue..."
```

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;}
```

Remove color codes (special characters) with sed

```
>_ sed -r "s/\x1B\[([0-9]{1,2}(;[0-9]{1,2})?)?[m|K]//g"
```

Throttle bandwidth with cstream

```
>_ tar -cj /backup | cstream -t 777k | ssh host ' tar -xj -C /backup'
```

Run a long job and notify me when it's finished

```
>_ ./my-really-long-job.sh && notify-send "Job finished"
```

When feeling down, this command helps

```
>_ sl
```

Nicely display permissions in octal format with filename

```
>_ stat -c '%A %a %n' *
```

Create a single-use TCP (or UDP) proxy

```
>_ nc -l -p 2000 -c "nc example.org 3000"
```

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
```

Instead of writing a multiline if/then/else/fi construct you can do that by one line

```
>_ [[ test_condition ]] && if_true_do_this || otherwise_do_that
```

Switch 2 characters on a command line.

```
>_ ctrl-t
```

Speed up launch of firefox

```
>_ find ~ -name '*.sqlite' -exec sqlite3 '{}' 'VACUUM;' \;
```

Check the status of 'dd' in progress (OS X)

```
>_ CTRL + T
```

Convert seconds into minutes and seconds

```
>_ bc <<< 'obase=60;299'
```

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

```
>_ ( command & )
```

Find usb device

```
>_ diff <(lsusb) <(sleep 3s && lsusb)
```

Take screenshot through SSH

```
>_ DISPLAY=:0.0 import -window root /tmp/shot.png
```

Opens vi/vim at pattern in file

```
>_ vi +/pattern [file]
```

April Fools' Day Prank

```
>_ PROMPT_COMMAND='if [ $RANDOM -le 3200 ]; then printf "\0337\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
```

Pipe stdout and stderr, etc., to separate commands

```
>_ some_command > >(/bin/cmd_for_stdout) 2> >(/bin/cmd_for_stderr)
```

Schedule a script or command in x num hours, silently run in the background even if logged out

```
>_ ( ( sleep 2h; your-command your-args ) & )
```

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
```

Make anything more awesome

```
>_ command | figlet
```

List all files opened by a particular command

```
>_ lsof -c dhcpd
```

Gets a random Futurama quote from /.

```
>_ curl -Is slashdot.org | egrep '^X-(F|B|L)' | cut -d \- -f 2
```

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
```

Runs a bash script in debugging mode

```
>_ bash -x ./post_to_commandlinefu.sh
```

List the number and type of active network connections

```
>_ netstat -ant | awk '{print $NF}' | grep -v '[a-z]' | sort | uniq -c
```

Tell local Debian machine to install packages used by remote Debian machine

```
>_ ssh remotehost ' dpkg --get-selections' | dpkg --set-selections && dselect install
```

Lists all listening ports together with the PID of the associated process

```
>_ lsof -Pan -i tcp -i udp
```

Intersection between two files

```
>_ grep -Fx -f file1 file2
```

Alias HEAD for automatic smart output

```
>_ alias head='head -n $((${LINES:-`tput lines 2>/dev/null||echo -n 12`} - 2))'
```

Colorful man

```
>_ apt-get install most && update-alternatives --set pager /usr/bin/most
```

Prints line numbers

```
>_ nl
```

Diff on two variables

```
>_ diff <(echo "$a") <(echo "$b")
```

Manually Pause/Unpause Firefox Process with POSIX-Signals

```
>_ killall -STOP -m firefox
```

Use lynx to run repeating website actions

```
>_ lynx -accept_all_cookies -cmd_script=/your/keystroke-file
```

Display a list of committers sorted by the frequency of commits

```
>_ svn log -q|grep "|"|awk "{print \$3}"|sort|uniq -c|sort -nr
```

Bind a key with a command

```
>_ bind -x '"\C-l":ls -l'
```

Find all file larger than 500M

```
>_ find / -type f -size +500M
```

GREP a PDF file.

```
>_ pdftotext [file] - | grep 'YourPattern'
```

Live ssh network throughput test

```
>_ yes | pv | ssh $host "cat > /dev/null"
```

Convert unixtime to human-readable

```
>_ date -d @1234567890
```

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

```
>_ strings /dev/mem
```

Prettify an XML file

```
>_ tidy -xml -i -m [file]
```

Get the IP of the host your coming from when logged in remotely

```
>_ echo ${SSH_CLIENT%% *}
```

Random Number Between 1 And X

```
>_ echo $[RANDOM%X+1]
```

Notify yourself when a long-running command which has ALREADY STARTED is finished

```
>_ <ctrl+z> fg; notify_me
```

Create colorized html file from Vim or Vimdiff

```
>_ :TOhtml
```

Output your microphone to a remote computer's speaker

```
>_ arecord -f dat | ssh -C user@host
aplay -f dat
```

Create a nifty overview of the hardware in your computer

```
>_ lshw -html > hardware.html
```

Save your sessions in vim to resume later

```
>_ :mksession! <filename>
```

Convert PDF to JPG

```
>_ for file in `ls *.pdf`; do convert
-verbose -colorspace RGB -resize 800
-interlace none -density 300 -quality 80
$ file `echo $ file | sed 's/\.pdf$/\.jpg/'`;
done
```

Exclude multiple columns using AWK

```
>_ awk '{$1=$3=""}1' file
```

Easily find megabyte eating files or directories

```
>_ alias dush="du -sm *|sort -n|tail"
```

Exclude .svn, .git and other VCS junk for a pristine tarball

```
>_ tar --exclude-vcs -cf src. tar src/
```

Remove blank lines from a file using grep and save output to new file

```
>_ grep . filename > newfilename
```

Delete a line from your shell history

```
>_ history -d
```

Exit without saving history

```
>_ kill -9 $$
```

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"
```

Given a file path, unplug the USB device on which the file is located (the file must be on an USB device !)

```
>_ echo $( sudo lshw -businfo | grep -B 1 -m 1 $(df "/path/to/file" | tail -1 | awk '{print $1}' | cut -c 6-8) | head -n 1 | awk '{print $1}' | cut -c 5- | tr ":" "-") | sudo tee /sys/bus/usb/drivers/usb/unbind
```

Remove a line in a text file. Useful to fix "ssh host key change" warnings

```
>_ sed -i 8d ~/.ssh/known_hosts
```

Save a file you edited in vim without the needed permissions (no echo)

```
>_ :w !sudo tee > /dev/null %
```

Analyze traffic remotely over ssh w/ wireshark

```
>_ ssh root@server.com 'tshark -f "port !22" -w -' | wireshark -k -i -
```

Create a directory and change into it at the same time

```
>_ md () { mkdir -p "$@" && cd "$@"; }
```

Control ssh connection

```
>_ [enter]~?
```

Python version 3: Serve current directory tree at http://$HOSTNAME:8000/

```
>_ python -m http.server
```

Ls not pattern

```
>_ ls !(*.gz)
```

Show apps that use internet connection at the moment.

```
>_ lsof -P -i -n | cut -f 1 -d " "| uniq | tail -n +2
```

Colorized grep in less

```
>_ grep --color=always | less -R
```

Quick access to ASCII code of a key

```
>_ showkey -a
```

Find files in a date range

```
>_ find . -type f -newermt "2010-01-01" ! -newermt "2010-06-01"
```

Run complex remote shell cmds over ssh, without escaping quotes

```
>_ ssh host -l user $(<cmd.txt)
```

A robust, modular log coloriser

```
>_ ccze
```

Intercept, monitor and manipulate a TCP connection.

```
>_ mkfifo /tmp/fifo; cat /tmp/fifo | nc -l -p 1234 | tee -a to.log | nc machine port | tee -a from.log > /tmp/fifo
```

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"}'
```

Check site ssl certificate dates

```
>_ echo | openssl s_client -connect www.google.com:443 2>/dev/null |openssl x509 -dates -noout
```

Convert Youtube videos to MP3

```
>_ youtube-dl -t --extract-audio --audio-format mp3 YOUTUBE_URL_HERE
```

Using `!#$' to referance backward-word

```
>_ cp /work/host/phone/ui/main. cpp !#$:s/host/target
```

Use tee to process a pipe with two or more processes

```
>_ echo "tee can split a pipe in two"|tee >(rev) >(tr ' ' '_')
```

Get your outgoing IP address

```
>_ dig +short myip.opendns.com @resolver1.opendns.com
```

Port Knocking!

```
>_ knock <host> 3000 4000 5000 && ssh -p <port> user@host && knock <host> 5000 4000 3000
```

Compare two directory trees.

```
>_ diff <(cd dir1 && find | sort) <(cd dir2 && find | sort)
```

Add timestamp to history

```
>_ export HISTTIMEFORMAT="%F %T "
```

Search recursively to find a word or phrase in certain file types, such as C code

```
>_ find . -name "*.[ch]" -exec grep -i -H "search pharse" {} \;
```

Recursively change permissions on files, leave directories alone.

```
>_ find ./ -type f -exec chmod 644 {} \;
```

Find files that have been modified on your system in the past 60 minutes

```
>_ find / -mmin 60 -type f
```

Find out how much data is waiting to be written to disk

```
>_ grep ^Dirty /proc/meminfo
```

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
```

Bring the word under the cursor on the :ex line in Vim

```
>_ :<C-R><C-W>
```

Make directory tree

```
>_ mkdir -p work/{d1,d2}
/{src,bin,bak}
```

Duplicate installed packages from one machine to the other (RPM-based systems)

```
>_ ssh root@remote.host "rpm -qa" |
xargs yum -y install
```

Draw kernel module dependancy graph.

```
>_ lsmod | perl -e ' print "digraph \"
lsmod\" {";<>;while(<>)
{@_=split/\s+/; print "\"$_[0]\" ->
\"$_\"\n" for split/,/,$_[3]} print "}"' |
dot -Tpng | display -
```

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
```

Binary Clock

```
>_ watch -n 1 'echo "obase=2;`date
+%s`" | bc'
```

Remind yourself to leave in 15 minutes

```
>_ leave +15
```

Download all images from a site

```
>_ wget -r -l1 --no-parent -nH -nd
-P/tmp -A".gif,.jpg" http://example.com
/images
```

What is my public IP-address?

```
>_ curl ifconfig.me
```

Which program is this port belongs to ?

```
>_ lsof -i tcp:80
```

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'
```

Nice weather forecast on your shell

```
>_ curl wttr.in/seville
```

Retry the previous command until it exits successfully

```
>_ until !!; do :; done
```

Synchronize date and time with a server over ssh

```
>_ date --set="$(ssh user@server date)"
```

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
```

List only the directories

```
>_ ls -d */
```

RTFM function

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

Broadcast your shell thru ports 5000, 5001, 5002 ...

```
>_ script -qf | tee >(nc -kl 5000) >(nc -kl 5001) >(nc -kl 5002)
```

Edit a google doc with vim

```
>_ google docs edit --title "To-Do List" --editor vim
```

Eavesdrop on your system

```
>_ diff <( lsof -p 1234) <(sleep 10; lsof -p 1234)
```

Run a file system check on your next boot.

```
>_ touch /forcefsck
```

To print a specific line from a file

```
>_ sed -n 5p <file>
```

Directly ssh to host B that is only accessible through host A

```
>_ ssh -t hostA ssh hostB
```

Remove all files previously extracted from a tar(.gz) file.

```
>_ tar -tf <file. tar.gz> | xargs rm -r
```

Share a terminal screen with others

```
>_ % screen -r someuser/
```

List all bash shortcuts

```
>_ bind -P
```

Start COMMAND, and kill it if still running after 5 seconds

```
>_ timeout 5s COMMAND
```

Pretend to be busy in office to enjoy a cup of coffee

```
>_ cat /dev/urandom | hexdump -C | grep "ca fe"
```

Recursively remove all empty directories

```
>_ find . -type d -empty -delete
```

Create a quick back-up copy of a file

```
>_ cp file.txt{,.bak}
```

Attach screen over ssh

```
>_ ssh -t remote_host screen -r
```

Run a command only when load average is below a certain threshold

```
>_ echo "rm -rf /unwanted-but-large/folder" | batch
```

Makes the permissions of file2 the same as file1

```
>_ chmod --reference file1 file2
```

Show a 4-way scrollable process tree with full details.

```
>_ ps awwfux | less -S
```

Remove a line in a text file. Useful to fix

```
>_ ssh-keygen -R <the_offending_host>
```

Open Finder from the current Terminal location

```
>_ open .
```

Create a persistent connection to a machine

```
>_ ssh -MNf <user>@<host>
```

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
```

Remove security limitations from PDF documents using ghostscript

```
>_ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=OUTPUT.pdf -c .setpdfwrite -f INPUT.pdf
```

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 '<en try>'
'{for (i=2; i<=NF; i++) {print $i}}' |
sed -n "s/<title>\(.*\)<\/title.*name>
\(.*\)<\/name>.*/\2 - \1/p"
```

Search commandlinefu.com from the command line using the API

```
>_ cmdfu(){ curl
"http://www.commandlinefu.com
/commands/matching/$@/$(echo -n $@
| openssl base64)/plaintext"; }
```

Display a cool clock on your terminal

```
>_ watch -t -n1 "date +%T|figlet"
```

(Debian/Ubuntu) Discover what package a file belongs to

```
>_ dpkg -S /usr/bin/ls
```

Processor / memory bandwidthd? in GB/s

```
>_ dd if=/dev/zero of=/dev/null bs=1M
count=32768
```

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
```

Send pop-up notifications on Gnome

```
>_ notify-send ["<title>"] "<body>"
```

Search for a <pattern> string inside all files in the current directory

```
>_ grep -RnisI <pattern> *
```

Record a screencast and convert it to an mpeg

```
>_ ffmpeg -f x11grab -r 25 -s 800x600 -i
:0.0 /tmp/outputFile.mpg
```

Convert seconds to human-readable format

```
>_ date -d@1234567890
```

Monitor the queries being run by MySQL

```
>_ watch -n 1 mysqladmin --user=
<user> --password=<password>
processlist
```

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 "" }'
```

Job Control

```
>_ ^Z $bg $disown
```

Generate a random password 30 characters long

```
>_ s trings /dev/urandom | grep -o
'[[:alnum:]]' | head -n 30 | tr -d '\n'; echo
```

Monitor progress of a command

```
>_ pv access.log | gzip > access.log.gz
```

Multiple variable assignments from command output in BASH

```
>_ read day month year <<< $(date
+'%d %m %y')
```

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

```
>_ ss -p
```

Edit a file on a remote host using vim

```
>_ vim scp://username@host//path
/to/somefile
```

Mkdir & cd into it as single command

```
>_ mkdir /home/foo/doc/bar && cd $_
```

Create a pdf version of a manpage

```
>_ man -t manpage | ps2pdf -
filename.pdf
```

Remove all but one specific file

```
>_ rm -f !(survivior.txt)
```

Make directory including intermediate directories

```
>_ mkdir -p a/long/directory/path
```

Rapidly invoke an editor to write a long, complex, or tricky command

```
>_ fc
```

Git remove files which have been deleted

```
>_ git add -u
```

Copy a file using pv and watch its progress

```
>_ pv sourcefile > destfile
```

Graphical tree of sub-directories

```
>_ ls -R | grep ":$" | sed -e 's/:$//' -e
's/[^-][^\/]*\/--/g' -e 's/^/ /' -e 's/-/|/'
```

Easily search running processes (alias).

```
>_ alias 'ps?'='ps ax | grep '
```

Intercept stdout/stderr of another process

```
>_ strace -ff -e trace=write -e write=1,2
-p SOME_PID
```

Define a quick calculator function

```
>_ ? () { echo "$*" | bc -l; }
```

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/.*\(v[0-9]\.lscache.*
\)/http:\/\/\1/g') | grep -oP '^[^|,]*')
```

Create a CD/DVD ISO image from disk.

```
>_ readom dev=/dev/scd0 f=/path
/to/image.iso
```

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

```
>_ dd if=/dev/mem | cat | strings
```

Replace spaces in filenames with underscores

```
>_ rename 'y/ /_/' *
```

Inserts the results of an autocompletion in the command line

```
>_ ESC *
```

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"
```

Matrix Style

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

Rip audio from a video file.

```
>_ mplayer -ao pcm -vo null -vc dummy
-dumpaudio -dumpfile <output-file>
<input-file>
```

Google Translate

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

Find the process you are looking for minus the grepped one

```
>_ ps aux | grep [p]rocess-name
```

Extract tarball from internet without local saving

```
>_ wget -qO - "http://www. tarball.com/
tarball.gz" | tar zxvf -
```

Copy your SSH public key on a remote machine for passwordless login - the easy way

```
>_ ssh-copy-id username@hostname
```

Insert the last command without the last argument (bash)

```
>_ !:-
```

Display which distro is installed

```
>_ cat /etc/issue
```

Python smtp server

```
>_ python -m smtpd -n -c
DebuggingServer localhost:1025
```

Remove duplicate entries in a file without sorting.

```
>_ awk '!x[$0]++' <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
```

Kills a process that is locking a file.

```
>_ fuser -k filename
```

Show File System Hierarchy

```
>_ man hier
```

Add Password Protection to a file your editing in vim.

```
>_ vim -x <FILENAME>
```

Save command output to image

```
>_ ifconfig | convert label:@- ip.png
```

Display a block of text with AWK

```
>_ awk '/start_pattern/,/stop_pattern/' file.txt
```

Set CDPATH to ease navigation

```
>_ CDPATH=:..:~:~/projects
```

Backticks are evil

```
>_ echo "The date is: $( date +%D)"
```

Reuse all parameter of the previous command line

```
>_ !*
```

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

```
>_ lsof -P -i -n
```

Sharing file through http 80 port

```
>_ nc -v -l 80 < file.ext
```

Delete all files in a folder that don't match a certain file extension

```
>_ rm !(*.foo|*.bar|*.baz)
```

Create a script of the last executed command

```
>_ echo "!!" > foo.sh
```

Easy and fast access to often executed commands that are very long and complex.

```
>_ some_very_long_and_complex_command # label
```

Watch Network Service Activity in Real-time

```
>_ lsof -i
```

Diff two unsorted files without creating temporary files

```
>_ diff <(sort file1) <(sort file2)
```

Escape any command aliases

```
>_ \[command]
```

Push your present working directory to a stack that you can pop later

```
>_ pushd /tmp
```

Quickly rename a file

```
>_ mv filename.{old,new}
```

Display the top ten running processes - sorted by memory usage

```
>_ ps aux | sort -nk +4 | tail
```

Simulate typing

```
>_ echo "You can simulate on-screen typing just like in the movies" | pv -qL 10
```

Reboot machine when everything is hanging

```
>_ <alt> + <print screen/sys rq> + <R> - <S> - <E> - <I> - <U> - <B>
```

32 bits or 64 bits?

```
>_ getconf LONG_BIT
```

Set audible alarm when an IP address comes online

```
>_ ping -i 60 -a IP_address
```

Shutdown a Windows machine from Linux

```
>_ net rpc shutdown -I ipAddressOfWindowsPC -U username%password
```

List of commands you use most often

```
>_ history | awk '{a[$2]++}END{for(i in a){print a[i] " " i}}' | sort -rn | head
```

Put a console clock in top right corner

```
>_ while sleep 1;do tput sc; tput cup 0 $(($( tput cols)-29));date; tput rc;done &
```

Close shell keeping all subprocess running

```
>_ disown -a && exit
```

Watch Star Wars via telnet

```
>_ telnet towel.blinkenlights.nl
```

Make 'less' behave like 'tail -f'.

```
>_ less +F somelogfile
```

SSH connection through host in the middle

```
>_ ssh -t reachable_host ssh unreachable_host
```

A very simple and useful stopwatch

```
>_ time read (ctrl-d to stop)
```

Update twitter via curl

```
>_ curl -u user:pass -d status="Tweeting from the shell" http://twitter.com/statuses/update.xml
```

Compare a remote file with a local file

```
>_ ssh user@host cat /path/to /remotefile | diff /path/to/localfile -
```

Clear the terminal screen

```
>_ ctrl-l
```

Download an entire website

```
>_ wget --random-wait -r -p -e robots=off -U mozilla http://www.example.com
```

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
```

Type partial command, kill this command, check something you forgot, yank the command, resume typing.

```
>_ <ctrl+u> [...] <ctrl+y>
```

Output your microphone to a remote computer's speaker

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

Query Wikipedia via console over DNS

```
>_ dig +short txt <keyword>.wp.dg.cx
```

Quick access to the ascii table.

```
>_ man ascii
```

Get your external IP address

```
>_ curl ifconfig.me
```

Execute a command at a given time

```
>_ echo "ls -l" | at midnight
```

Currently mounted filesystems in nice layout

```
>_ mount | column -t
```

Salvage a borked terminal

```
>_ reset
```

Place the argument of the most recent command on the shell

```
>_ 'ALT+.' or '<ESC> .'
```

Rapidly invoke an editor to write a long, complex, or tricky command

```
>_ ctrl-x e
```

Runs previous command but replacing

```
>_ ^foo^bar
```

Serve current directory tree at http://$HOSTNAME:8000/

```
>_ python -m SimpleHTTPServer
```

Run the last command as root

```
>_ sudo !!
```