```
<CTRL>+<SUPER>+<LEFT>/<RIGHT> # half-maximize a window on one side, in Ubuntu

~Â°~Â°~Â°~Â°~Â°~Â°~Â°~Â°~Â°~Â°~
### SHELL & misc  ###
~Â°~Â°~Â°~Â°~Â°~Â°~Â°~Â°~Â°~Â°~
<CTRL>+e : perform shell-"Expansnon" on cmd-line - cf. .inputrc
<CTRL>+u : remove part of current line "a gaUche"
<CTRL>+k : remove "Klosing" part of current line
<CTRL>+w : cut the "Word" before the cursor to the clipboard
<CTRL>+y : "Yank" (paste) the last thing cut
<CTRL>+r : search bash history, powerful to use with cmd #tags
<ALT>+r : cancel the changes made on a line and "Revert" it as it was in the history
!$ # the last word of the last cmd-line
!!:n # the nth word of the last cmd-line
^command^user^ # repeat last command with word substitution - Alt: alias r='fc -s'; r command=user

less + /search + SHIFT-F # LIKE grep|tail -f !!
cmd1 <(cmd2) >(cmd3) # make cmd2 output & cmd3 input look like a file for cmd1 - Alt:
cmd2 | cmd1 /dev/stdin # aka /proc/self/fd/0 aka /dev/fd/0 - Useful when a command needs a file as parameter, use a symlink if extension matters
>| # '>' that overrides 'set -o noclobber' == existing regular files cannot be overwritten by redirection of output

time read # chrono
man ascii # display ASCII table
cal # quick calendar - Also: calcurse, wyrd
look # find English words (or lines in a file) beginning with a string

# 'top' < 'htop'
* display full command path of processes : "c"
* killing : Press "k", then pid, then signal (15, 9...)
* sorting : press "O" and select the column
* display /cores stats : "1"
* colors : 'Z'; then save config: 'W'
VIRT: how much memory the program is able to access at the present moment
RES: resident size, how much actual physical memory a process is consuming (heap memory that is currently in RAM) + (non heap memory in RAM) + (thread stack memory * number of threads) + (direct/mapped buffer memory)
DATA is the amount of VIRT used that isn't shared and that isn't code-text; i.e., it is the virtual stack and heap of the process
SHR: how much of the VIRT size is actually sharable memory or libraries
SWAP: bogus
pstree -p [$OPT_PID] # hierarchy of processes

pgrep -f $procname_pattern | xargs ps -fp # or kill - Alt: pidof $procname
/proc/$pid/{cmdline,comm,exe} # get process name - Alt: ps -o comm= -p $pid
file /proc/$pid/cwd # get process working directory - Alt: pwdx $pid

nohup $cmd # detach command
disown -h $pid # detach running process - To redirect its stdout/err: dupx / https://gist.github.com/zaius/782263

kill -15 # (TERM) then wait 2sec, then -2 (INT) then -1 (HUP) THEN -9 (KILL)
# Because then: no sockets shutdown / no tmp files cleanup / children not informed / no terminal reset
kill -l # list signals

xkill # kill window by clicking
xprop # get window infos by cliking
xdpyinfo / xwininfo -children -id $ID # get X11 windows infos

xclip [-selection clipboard] # copy & paste clipboard
ranger # text-based file manager written in Python with vi key bindings

write / mesg # 2nd control write access
wall # broadcast message

ttyrec, ipbt, ttygif, playitagainsam, KeyboardFire/mkcast # record & playback terminal sessions - Last one provides a JS player
licecap # record any screen interaction as GIF - cf. http://superuser.com/a/657800/255048

export -f bash_func; xargs -P 0 -i sh -c 'bash_func "$@"' _ {} # Alt: GNU parallel, mfisk/filemap 'fm' Map-Reduce command

cd // # broken PWD

: () { : | : & } ; : # Fork bomb

perl -wle 'exit 1 if (1 x shift) !~ /^1?$|^(11+?)\1+$/' # Primality testing with a REGEX !

a(){ echo $2 \\$1 $1 $2 $1 ;};a \' ' a(){ echo $2 \\$1 $1 $2 $1 ;};a ' # Quine
```

```
dig +short TXT google-public-dns-a.google.com # check without 'TXT'
dig +short TXT istheinternetonfire.com
traceroute -m 60 216.81.59.173; telnet towel.blinkenlights.nl # Star Wars

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

rainbow_cursor_worm () { a=1;x=1;y=1;xd=1;yd=1;while true;do if [[ $x == $LINES || $x == 0 ]]; then xd=$(( $xd *-1 )) ; fi ; if [[ $y == $COLUMNS || $y == 0 ]]; then yd=$(( $yd * -1 )) ; fi ; x=$(( $x + $xd )); y=$(( $y + $yd )); printf "\33[%s;%sH\33[48;5;%sm \33[0m" $x $y $(($a%199+16)) ;a=$(( $a + 1 )) ; sleep 0.001 ;done; } # FROM: http://www.climagic.org/coolstuff/cursor-tricks.html

( play -q -n synth sine F2 sine C3 remix - fade 0 4 .1 norm -4 bend 0.5,2399,2 fade 0 4.0 0.5 & )
echo 'main(t){for(;;t++)putchar(((t<<1)^((t<<1)+(t>>7)&t>>12))|t>>(4-(1^7&(t>>19)))|t>>7);}' | cc -x c - -o crowd && ./crowd | aplay


##################
  Bash scripting
##################
figlist | sed '1,/Figlet fonts/d;/:/,$d' | xargs -I{} figlet -f {} Hello # ASCII banner fonts
xmessage -center "$(figlet ERRORMSG 42)", notify-send (libnotify), bar, dialog, gdialog==zenity # GUI: error windows, selection dialog, progress bars...
mooz/percol | peco/peco | moreutils/vipe # interactive filtering through pipes

set -o pipefail -o errexit -o nounset -o xtrace # can be read / exported to subshells using $SHELLOPTS
export PS4='+ ${FUNCNAME[0]:+${FUNCNAME[0]}():}line ${LINENO}: '

bash -n $script # Check syntax without executing
bash --debugger $script

parent_func=$(caller 0 | cut -d' ' -f2) # "$line $subroutine $filename"
source ~/sctrace.sh # FROM: http://stackoverflow.com/questions/685435/bash-stacktrace/686092

readonly EXEC_DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )" # Script file parent dir
readonly LOG_FILE="$EXEC_DIR/logs/$(basename $0).log.$(date +%Y-%m-%d-%H)"
exec > >(tee -a $LOG_FILE); exec 2>&1
date "+%F %T,%N" | cut -c-23 # Standard logs date
date -u +%s # Seconds since EPOCH
date -d @$seconds_since_epoch "+%F" # under OSX: date -jf "%s" $secs "+%F"

# !! aliases used in functions definitions are immediately substituted,
# NOT resolved dynamically !
alias foo='echo A'
bar () { foo; }
alias foo='echo B'
bar # echo A

# Set positional parameters $0 $1 ...
set - A B C

: ${1:?'Missing or empty parameter'}
: ${var:="new value set if empty"}
local var=${1:-"default value"}
foo () { local x=$(false); echo $?; }; foo # -> 0 !!GOTCHA!! 'local' is also a command, and its return code shadows the one of the cmd invoked

echo ${PWD//\//-} # Variables substitutions (http://tldp.org/LDP/abs/html/parameter-substitution.html)
${var%?} # Remove the final character of var

for pair in $whatever; do key=${pair%:*}; value=${pair#*:}; ...
for f in ./*.txt; do; [[ -f "$f" ]] || continue # Safe 'for' loop - http://bash.cumulonim.biz/BashPitfalls.html

readonly CONST=42 # works with arrays & functions too - Beautiful hack to unset: http://stackoverflow.com/a/21294582

# Q: Can we find a function 'identity' that satisfies the following 2 properties ? stackoverflow.com/q/21635301
identity () { for arg in "$@"; do echo "$arg"; done; }
identity "$(identity a\ b c\ d)"
# a b
# c d # expected output: OK
argv_count () { echo "argv_count($@):$#" >&2; }
argv_count $(identity a\ b c\ d)
# 4 # NOT 2 : KO
# ANSWER: NO, because $() mangle the output in one string
# => use | over $() for list of strings containing spaces

# Q: How to store the output of a command in a variable without spawning a subshell ? stackoverflow.com/q/21632126
bar () { echo "$BASH_SUBSHELL $BASHPID"; }
mapfile -t bar_output < <(foo) # STILL creates a new process + only available since bash 4
# -> use a non-blocking FIFO !

local argv=("$@") # Convert to array
"${argv[*]}" # expands to a single word with the value of each array member separated by the first character
of the IFS variable
"${name[@]}" # expands each element of name to a separate word
${#argv[@]} != ${#argv} # array size VS char-length of 1st elem
${argv[@]:(-1)} # last element
echo ${argv[@]:1:2} # Array slice
unset argv[0] # remove element, WITHOUT-INDEX-SHIFTING

# Parsing *=* args (unsecure) by pushing elements in an array
declare -a argFiles # optional
for arg in "$@"; do
    case $arg in
        *=*) eval $argi ;;
        *) argFiles[${#argFiles[*]}]="$arg" ;;
    esac
done
# http://wiki.bash-hackers.org/howto/getopts_tutorial
while getopts ":ab:" opt; do
    case $opt in
    a) echo "-a was triggered." >&2 ;;
    b) echo "-b was triggered. Parameter: $OPTARG" >&2 ;;
    \?) echo "Invalid option: -$OPTARG" >&2 ; exit 1 ;;
    :) echo "Option -$OPTARG requires an argument." >&2 ; exit 1 ;;
    esac
done

# CGI scripts
$(</dev/stdin) # POST data
saveIFS=$IFS; IFS='=&'; qparams_array=($QUERY_STRING); IFS=$saveIFS # ?foo=bar&x=42 => (foo bar x 42)
declare -A qparams; for ((i=0; i<${#qparams_array[@]}; i+=2)); do
qparams["${qparams_array[i]}"]="${qparams_array[i+1]}"; done # Alt: bashlib
echo -ne "Content-type: text/html\n\nCGI Bash Example: $(for k in "${!qparams[@]}"; do echo $k:${qparams[$k]};
done)"

declare -A hash_table # Bash 4+ associative arrays
# Or, with built-in arrays and cksum-based hashing function (FROM:
http://stackoverflow.com/questions/1494178/how-to-define-hash-tables-in-bash)
hf () { local h=$(echo "$*" |cksum); echo "${h//[!0-9]}"; } # hashing function
table[$(hf foo bar)]="x42"
echo ${table[$(hf foo bar)]}
echo ${table[@]}
# With /dev/shm in-memory files (+persistent)
hinit() { rm -rf "/dev/shm/hashmap.$1" ; mkdir -p "/dev/shm/hashmap.$1" ; }
hput() { echo "$3" > "/dev/shm/hashmap.$1/$2" ; } # or printf to avoid \n
hget() { cat "/dev/shm/hashmap.$1/$2" ; }
hkeys() { ls -1 "/dev/shm/hashmap.$1" ; }
hvalues() { cat "/dev/shm/hashmap.$1/*" ; }
hcount() { hkeys $1 | wc -l ; }
hdestroy() { rm -rf "/dev/shm/hashmap.$1" ; }

# Powerful regex
[[ "some string" =~ "$regex" ]]
group1="${BASH_REMATCH[1]}"

# Simulating 'pipefail', from gzip:zgrep source code
r=$(
    exec 4>&1
    (eval "$cmd1" 4>&-; echo $? >&4) | sed "$cmd2" 4>&-
) && exit $r

# Create and set permissions
install -o ${SUDO_USER:-$USER} -m 644 $src $dst
install -d -m 777 $directory

# Floating point arithmetic
echo "$((RANDOM%6+1)) + 1/3" | bc -l # or specify "scale=X;" instead of flag - Also: qalc
factor $really_long_int # decompose in factors

is_true () { ! { [ -z "$1" ] || [[ "$1" =~ 0+ ]] || [[ "$1" =~ [Ff][Aa][Ll][Ss][Ee] ]] ; } ; }

is_file_open () { lsof | grep $(readlink -f "$1") ; }

cat <<EOF
EOF

exec 8<>filename # Open file descriptors #8 for reading and writing
echo BlaBlaBla
```

```
exec 8>&- # Close file descriptor

/var/tmp is better than /tmp # as filling it is less system impacting
tdir="$(mktemp -d ${TMPDIR:-/tmp}/$0_XXXXXX)" # mktemp dir & default value
/dev/shmi # Use RAM for tmp files - monitor usage with ipcs -m

for i in {0..255}; do printf "\x1b[38;5;${i}mcolor${i}\x1b[0m\n"; done # display all 256 colors
for i in {1..8}; do echo "$(tput setaf $i)color_$i$(tput sgr0)"; done # colored terminal output
# + colors can be set like this: tput initc 2 500 900 100 # RGB values between 0 & 1000
# Other tput: setab [1-7], setf [1-7], setb [1-7], bold, dim, smul, rev... cf. man terminfo
tput sc;tput cup 0 $(($(tput cols)-29));date;tput rc # put a clock in the top right corner

select value in choice1 choice2; do break; done # multiple choices
read -s password # 'silent' user input, no characters are displayed
strings /dev/urandom | grep -o '[[:alnum:]]' | head -n 30 | tr -d '\n' # 30 characters password generation
stty -echo # disable TTY output
stty -echo -icanon time 0 min 0 # non-blocking read trick FROM: http://stackoverflow.com/a/5297780
# ask a yes or no question, with a default of no.
echo -n "Do you ...? [y/N]: "
read answer
if expr "$answer" : ' *[yY].*' > /dev/null; then
   echo OK
else
   echo KO
fi

set -o noglob # disable wildcard expansion
# Extended bash globbing
shopt -s extglob # http://www.linuxjournal.com/content/bash-extended-globbing
shopt [-o] # list options values. Alt: $- E.g. check if shell is interactive: [[ $- =~ i ]] - Also, is stdin open in a terminal: [ -t 0 ]

( set -o posix; set ) # List all defined variables
foo=bar; foo () { :; } ; unset foo # !!GOTCHA!! the variable is unset first, then the function if called a 2nd time
# Get all commands prefixed by (useful for unit tests)
compgen -abck unit_test_ # control readline auto-completion (help complete), can be enable by '-e' flag of 'read'
complete -f -X '!*.ext' command # exclude files using a filter
complete -F _compfunc command
_compfunc() {
    local cmd="${1##*/}"
    local word=${COMP_WORDS[COMP_CWORD]}
    local line=${COMP_LINE}
    local xpat='!*.foo'

    COMPREPLY=($(compgen -f -X "$xpat" -- "${word}"))
}
hash # frequently used commands cache

syslogd -m 0 -r -SS # port: 514
logger -is -t SCRIPT_NAME -p user.warn "Message"
echo "<15>My logline" | nc -u -w 1 $HOSTNAME 514 # <15> means 'user.debug', see RFC3164: Facility*8 + Severity, default:13 <-> user.notice
time tcpdump udp and dst port 514 | awk '{print $3" "$7}' | sed 's/\.syslog//' > noisy_devices
logcheck, logtail
petit --hash /var/log/messages # Cmdline log analyze, also --wordcount. Alt: lnav ; sysdig -c spy_syslog

#   --reject doesn't apply to the whole path, only to the filename/query
mv $file ${file%.*}.bak # Change extension
mv --backup=numbered new target # !! --suffix/SIMPLE_BACKUP_SUFFIX can be broken on some distros
logrotate -s /var/log/logstatus /etc/logrotate.conf [-d -f] # Logrotate (to call in a cron job) Examples: http://www.thegeekstuff.com/2010/07/logrotate-examples/
# !! $@ not supported if < v.7.5

echo -e "00 00 * * * $USER cmd >> cmd.log 2&>1\n" | sudo tee /etc/crond.d/crontask # don't forget the newline at the end, don't use % symbols, don't put a dot '.' in its filename, use 644 permissions owned by root, and note that the $USER arg is not present in /etc/crontab files
sudo grep crontask /var/log/cron.log
flock -n /pathi/to/lockfile -c cmd # run cmd only if lock acquired, useful for cron jobs
lockfile-create/remove/check # file locks manipulation
while true do inotifywait -r -e modify -e create -e delete -e move_self . ./run.sh done # inotify-tools based keep-alive trick - Alt: ayancey/dirmon
huptime --exec $cmd # zero downtime restarts of unmodified (networking) programs, intercept bind(2) and accept(2) calls

# Launch command at a specified time or when load average is under 0.8
echo $cmd | at midnight
echo $cmd | batch

nice / ionice / renice # Control process priority (useful in cron job)
```

```
# control the resources available to the shell and to processes it starts
ulimit -v # max virtual memory
ulimit -s # max stack size
ulimit -t # max of cpu time
ulimit -u # max number of processes
ulimit -a # ALL - Also: /proc/$pid/limits

mkfifo /tmp/myfifo; exec 3<> /tmp/myfifo # Ãœber trick: dummy FD => non-blocking named-pipe
python -c "from fcntl import ioctl ; from termios import FIONREAD ; from ctypes import c_int ; from sys import argv ; size_int = c_int() ; fd = open(argv[1]) ; ioctl(fd, FIONREAD, size_int) ; fd.close() ; print size_int.value" $fifo # readable bytes in a fifo -> NOT RELIABLE, e.g. always return 0 with non-blocking named-pipe
ulimit -p # should get max pipe size, but WRONG : defined in pipe_fs_i.h
fcntl(fd, F_SETPIPE_SZ, size) # to change max size, if Linux > 2.6.35 (/proc/sys/fs/pipe-max-size)

gnuplot -e "set term dumb; plot '<seq 1 9'" # ASCII graph - Alt, with UTF8 & colors: https://github.com/tehmaze/diagram
gnuplot -e "set term dumb size 200,50; plot [-5:6.5] sin(x) with impulse"
loop_cfg_file=/tmp/gnuplot_loop.cfg
in_data_file=/tmp/gnuplot_in.data
echo <<EOF >$loop_cfg_file
set term dumb size 200,50
set title 'Traffic-In (bytes/s)'
# Sampling 30 data points each time
plot '<tail -n 30 $in_data_file'
pause 1
replot
# Loop by rereading this file, doesn't work with -e on the command-line
reread
EOF
tail -F $log_file | grep $keyword | pv --line-mode --numeric >/dev/null 2>$in_data_file & # Alt: petit --sgraph
gnuplot $loop_cfg_file # real-time ASCII graphing !


+++++++++++++++
| Text stream |
+++++++++++++++

cat -vET # shows non-printing characters as ascii escapes.
printf "\177\n" # echo non-ascii, here 'DEL' in octal. echo $'\177' is equivalent, BUT:
# echo $'A\0B' -> A
# printf 'A\0B\n' -> AB

aha # convert ANSI colors into HTML tags
make 2>&1 | colout -t cmake | colout -t g++ # from nojhan github: "Color Up Arbitrary Command Output"

# grep Alt: ack, grin, ag (ggreer/the_silver_searcher), pt (monochromegane/the_platinum_searcher)
grep -a # if "Binary file (standard input) matches"
grep -q # silent, !! FAIL with SIGPIPE if 'pipefail' is used: http://stackoverflow.com/a/19120674/636849
grep '\<word\>' # match word-boundaries
grep -I # ignore binary files
grep -R --include='*.py' --exclude-dir='build/'
grep -o # output only matching parts
grep -C3 # output 3 lines of context, see also -B/-A
grep -H/-h # output with/without filename
grep -L $pattern $files # Get only filenames where PATTERN is not present
grep -P '^((?!b).)*a((?!b).)*$' # Grep 'a' but not 'b' -> PCRE ;  awk '/a/ && !/b/'
grep -P -n "[\x80-\xFF]" $file # Find non-ASCII characters
LANG=C grep -F # faster grep : fixed strings + no UTF8 multibyte, ASCII only (significantly better if v < 2.7)
sed -n '/FOO/,/BAR/p' # Print lines starting with one containing FOO and ending with one containing BAR.
sed -e "9r rabbit.ascii" -e "6iTITLE" template.html # insert a file + a specific text line in another file
perl -ne '/(error|warn)(?!negative-look-ahead-string-to-not-match-just-after)/i'
perl -ne '/r[eg](ex)p+/ && print "$1\n"' # print only matching groups
grep | cut -c1-200 # truncate results to 200 characters
pyp # pip install --user pyp : alternative to sed, awk, cut & perl - Alt: pyped, Russell91/pythonpy

pdftotext $file.pdf - | grep # from xpdf-utils - Alt: euske/pdfminer pdf2txt.py OR pdftk OR LibreOffice Draw
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite [-dPDFSETTINGS=/screen|/ebook|/printer|/prepress] -sOutputFile=$out.pdf $in.pdf # reduce pdf size with ghostscript - Also: -dFirstPage=X -dLastPage=Y - Alt: http://compress.smallpdf.com
pdfjam file1.pdf file2.pdf 1, 3- `# optional selector` --nup 2x1 --landscape --outfile out.pdf # printer-friendly version - Also: pdf290 to rotate

tr -c '[:alnum:]' _

# filter outpout : not lines 1-3 and last one
type ssh_setup | sed -n '1,3!p' | sed '$d'| sed 's/local //g'
# this is also a crazy hack : put the output in ORIG_CMD, then redefine ssh_setup () { eval $ORIG_CMD $@; ... }
```

```
perl -ne 'if (length > $w) { $w = length; print $ARGV.":".$_ };  END {print "$w\n"}' *.py # Longest line of
code
$cmd | awk '{print length, $0}' | sort -rn # sort by line length
cloc # count lines of code

comm -12 #or uniq -d - Sets intersec - See also: "Set Operations in the Unix Shell" - Alt: moreutils/combine
join # join lines of two files on a common field

tee -a $file # display input to stdout + append to end of $file
echo ECHO | sed s/$/.ext/ # Append at the end of stdout (or beginning with ^)
sed -i "1i$content" $file # append at the beginning of $file

sed ':a;N;$!ba;s/PATTERN\n/PATTERN/g' # remove newlines after PATTERN - How it works : N means
'pattern_space+=\n+nextline' and we use branching to :a - Alt: just '1!N; s/...//'
seq 1 10 | paste -s -d+ | bc # Replace newlines by a separator, aka 'join' - Also, for arrays: OLD_IFS=$IFS;
IFS=+; echo "${argv[*]}"; IFS=$OLD_IFS
sed "0,/$pattern/d" $file # print only lines after $pattern
# paste is also useful to interlace files: paste $file1 $file2
tac # reverse lines

perl -pe 's/\s+/\n/g' # Break on word per line
awk [-F":|="] '{ print $NF }' # Print last column. Opposite: awk '{$NF=""; print $0}'. Only last elems: awk -
F' ' '{for (i = 3; i <= NF; i++) printf "%s ",$i; print ""}'
mawk # faster awk
fold # breaks lines to proper width
fmt # reformat lines into paragraphs
printf "%-8s\n" "${value}" # 8 spaces output formatting
| xargs -n 1 sh -c 'echo ${0:0:3}' # 3 first characters of $string

csv{cut,look,stat,grep,sort,clean,format,join,stack,py,sql} {in,sql}2csv # pip install csvkit

jq -r '..|objects|.name//empty' # JSON syntax highlighting + sed-like processing - Basic alt: python -
mjson.tool
echo '{"A1":"a1","A2":"b2","B1":"b2"}' | jq '"A." as $regex | del(.[keys[]|select(match($regex))])'
echo '{"A0":["a1","a2","a3"], "B0":["b1","b2","b3"], "c3":[]}' | jq '".[^3]" as
$regex|to_entries|map(select(.key|match($regex)))|map(.value|=map(select(match($regex))))|from_entries'
source <(jq -r 'to_entries|.[]|"SAUCE_\(.key|ascii_upcase)=\(.value)"' .saucelabs_auth.json )
pup, html-xml-utils, xml2, 2xml, html2, 2html # convert XML/HTML to "grepable" stream - Also: xmlstarlet &
http://stackoverflow.com/a/91801

zcat /usr/share/man/man1/man.1.gz | groff -mandoc -Thtml > man.1.html # also -Tascii
txt2man -h 2>&1 | txt2man -T # make 'man' page from txt file
pandoc --standalone --smart --table-of-contents --include-in-header $css -f markdown -t html $f >
${f%%.md}.html # -s -S --toc
pandoc -s -f markdown -t man foo.md | man -l - # md2man : man pandoc_markdown
stmd foo.md | lynx -stdin # standard replacement for original 'markdown' command


=#=#=#=#=#=
   FILES
#=#=#=#=#=#

sleuthkit/scalpel # > foremost, file carving tool, cf. http://www.forensicswiki.org/wiki/Tools:Data_Recovery

ls | cut -d . -f 1 | funiq # Sum up kind of files without ext

find -D rates ... # details success rates of each match logic term
find / -xdev -size +100M -exec ls -lh {} \; # find big/largest files IGNORING other partitions - One can
safely ignore /proc/kcore - Alt: man agedu (-s $dir then -w / -t) ; + all tools listed in
http://dev.yorhel.nl/ncdu
find . -type d -name .git -prune -o -type f -print # Ignore .git
find -regex 'pat\|tern' # >>>way>more>efficient>than>>> \( -path ./pat -o -path ./tern \) -prune -o -print
find . \( ! -path '*/.*' \) -type f -printf '%T@ %p\n' | sort -k 1nr | sed -e 's/^[^ ]* //' -e "s/'/\\\\'/" |
xargs -I{} -n 1 ls -l "{}" # list files by modification time
find . -mtime +730 -print0 | xargs -0 --max-args 150 rm -f # to avoid 'Argument List Too Long' - Alt to mtime:
-newer $than_this_file
fdupes -r $dir # find duplicate files: size then MD5 then byte-by-byte - Also: findimagedupes

rename \  _ * # Replace whitespaces by underscores

# To see all files open in a directory structure:
lsof +D /some/dir
# To see all files jeff has open:
sudo lsof -u jeff
# Additional useful option : -r $t : repeat the listing every $t second
fuser $dir # identify processes using files or sockets

namei / readlink -f # Shows Where a File/Directory Comes From (links, etc.)

killall -HUP $process_name # To tell a process to reload its file descriptors, e.g. when deleting a log file
```

```
sudo dd if=/dev/urandom of=FAKE-2012Oct23-000000.rdb bs=1M count=6000 # Create fake file
truncate -s $size_in_bytes $file # from coreutils

# setuid: When an executable file has been given the setuid attribute, normal users on the system who have
permission to execute this file gain the privileges of the user who owns the file within the created process.
# setgid: Setting the setgid permission on a directory (chmod g+s) causes new files and subdirectories created
within it to inherit its group ID
setcap # man capabilities
umask # Control the permissions a process will give by default to files it creates; useful to avoid
temporarily having world-readable files before 'chmoding' them

setfacl -Rm u:"$user":rwx "$HOME/$dir" && setfacl -Rm d:u:"$user":rwx "$HOME/$dir" # Selectively gives access
to another user - Also: getfacl
sudo chattr +i [-R] $file # Forbid file deletion - To check a file attributes: lsattr. Also: getfattr/setfattr

tune2fs # control extX file system parameters, e.g. reclaim disk space reserved to root
debugfs -R "stat <$(ls -i $file | awk '{print $1}')>" $(df $file | tail -n 1 | awk '{print $1}') # Get $file
creation time ('crtime') on ext4 filesystems

# Bring back deleted file from limbo (ONLY if still in use in another process)
lsof | grep myfile # get pid
ls -l /proc/$pid/fd
cp /proc/$pid/fd/4 myfile.saved

# http://www.cyberciti.biz/tips/linux-audit-files-to-see-who-made-changes-to-a-file.html
auditctl -w $file -p wax -k $tag
ausearch -k $tag [-ts today -ui 506 -x cat]

Xfennec/cv # show progress & throughput of all running cp, mv, tar, gzip, cat...
rsync -v --progress --dry-run --compress $src_dir/ $dst_dir # Alt: rdiff-backup
--cvs-exclude --exclude=".*"
--archive # recursive + preserve mtime, permissions...
--delete # remove extra remote files
--backup --backup-dir=/var/tmp/rsync # keep a copy of the dst file

tar -czvf "$archive.tgz" "$dir_without_trailing_slash" # Extract: tar -xzvf $archive
tar -J... # instead of -z, .xz compression format support
pax > cpio > tar # http://dpk.io/pax
zipinfo $file.zip
pigz # paralell gzip, do not compress folders
yum install p7zip # for .7z files
lzop, lz4 # faster, use less CPU

sha{1,224,256,384,512}sum, md5sum, cksum


|Â°|Â°|Â°|Â°|Â°|Â°|Â°|Â°
== NETWORKING
|Â°|Â°|Â°|Â°|Â°|Â°|Â°|Â°

mtr $host > ping / traceroute
paris-traceroute > traceroute

socat > nc (netcat) > telnet # prefix with rlwrap ! Alt: stone -> a TCP/IP packet repeater in the application
layer, avoid forking for each packet received
socat - udp4-listen:5000,fork # create server redirecting listening on port 5000 output to terminal
nc -l -u -k -w 1 5000
echo hello | socat - udp4:127.0.0.1:5000 # send msg to server
echo hello | nc -u -w 1 127.0.0.1 5000

# Port scanning
nmap -sS -O 127.0.0.1 # Guess OS !! Also try -A - Alt: p0f
nmap $host -p $port --reason [-sT|-sU] # TCP/UDP scanning ; -Pn => no host ping, only scanning
nmap 192.168.1.* # Or 192.168.1.0/24, scan entire subnet
nmap -DdecoyIP1,decoyIP2 ... # cloak your scan

lsof -i -P -p $pid # -i => list all Internet network files ; -P => no conversion of port numbers to port names
for network files ; -n => no IP->hostname resolution
lsof -i -n | grep ssh # list SSH connections/tunnels

netstat -ntap # To find which processes are sending packets
netstat --statistics --udp # global network statistics - 'ss' is the replacement for deprecated 'netstat', but
this has no equivalent
ss -nap # -a => list both listening and non-listening sockets/ports ; -n => no DNS resolution for addresses,
use IPs ; -p => get pid & name of process owning the socket
ss -lp [-t|-u] # list only listening TCP/UDP sockets/ports

/proc/net/{snmp, netstat, ...} # network counters
dropwatch # to find out where are packets dropped
hping # packets crafting
mitmproxy --host # interactive examination and modification of HTTP traffic - cf. blog.philippheckel.com but
```

```
no need for -T - Alt: CharlesProxy, BurpProxy, Fiddler on Windows
mitmdump # tcpdump-like: view, record, and programmatically transform HTTP traffic

# Dump all tcp transmission to a specific IP :
sudo tcpdump -i $interface host $IP [ip proto icmp|udp|tcp] -A -s 0 # last flag remove the limit on the captured packet size | Use -X for hex-dump | -n to disable dns resolution
tcpdump udp and dst port 514 -w - | pv -btr >/dev/null # Incoming syslog UDP packets rate -> can be used for TCP or all network traffic too
time tcpdump udp and dst port 514 -w /dev/null -c 1000 # Alt solution to estimate the rate

ip n[eighbour] # ARP or NDISC cache entries - replace deprecated 'arp'
ip a[ddr] [show|add $ip] dev eth0 # replace deprecated 'ifconfig'
ip link set eth0 [up|down] # enable/disable the[interface specified
ip tunnel list # list ssh stunnels replace deprecated 'iptunnel'
ip route # host routing tables - replace deprecated 'route'
iw # details about wireless interfaces - replace deprecated 'iwconfig'
iwlist wlan0 scan | grep GHz # get congestion of Wifi channels
MACADDR=$(ip address show eth0 | grep link/ether | awk '{print $2 }') # can be used to get a unique machine id number instead of using $RANDOM:
echo $((  16#$(echo $MACADDR | sed 's/://g') % 10000 )) # use base16 - ALT: use md5sum

# On RedHat / CentOS / Fedora
$EDITOR /etc/sysconfig/network-scripts/ifcfg-eth0
$EDITOR /etc/sysconfig/network
/etc/init.d/network restart
ifup, ifdown # bring a network interface up

ls /var/lib/dhc* # check what DHCP client is used
# Query DNS cmds > deprecated 'nslookup'
host [-t txt] $hostname # -a (all records) -v
dig +short NS $hostname # find authoritative nameservers
dig @$dns_server $hostname
dig +short -x $ip # Reverse DNS
dig +trace +norecurse txt $dns_server
avahi-resolve -n $USER.local # Multicast DNS == mDNS - from avahi-tools pkg
# Caching
/etc/hosts /etc/{host,resolv,nsswitch}.conf /etc/dhcp*/*.conf # manual / basic
bind / dnsmasq / lwresd / unbound # DNS daemon
nsscache / nss_db / nscd (broken: ignore TTL) # Cache /etc/{passwd,group,shadow,...} - Notes: nscd-aggstats, nscd -g
getent ahostsv4 www.google.com # whole query through NSS
rndc # display various DNS cache control commands, part of Bind9 tools suite
rndc -p 954 dumpdb -cache # dump the cache in $(find /var -name named_dump.db) ; lwresd $port can be figured out with lsof/nmap
# View queries bypassing lwresd
/usr/sbin/tcpdump -pnl -s0 -c150 udp and dst port 53 and src port not \
    $(/usr/sbin/lsof -n -i4udp | awk '$1 == "lwresd" && $NF !~ /:921$/ { print substr($NF,3); exit }')

iptables -A INPUT -s $host -j DROP
iptables -A INPUT -p tcp -m tcp --dport 8888 -j ACCEPT
iptables -nvL --line-numbers # Also: iptables-save
iptables -D INPUT $rule_number
# Logging: connexion attempts will be traced in dmesg and, depending on syslog config, /var/log/kern.log
iptables -j LOG --log-level debug --log-prefix='[iptableslog] [dropped] ' -m limit --limit 1/sec --log-prefix -A INPUT/OUTPUT
iptables -j LOG --log-level debug --log-prefix='[iptableslog] [new] ' -m state --state NEW -I INPUT/OUTPUT 1
watch --color 'dmesg --notime | xargs -IX printf "[$(date -u)] %s\n" X >> /var/log/dmesg.log; dmesg --clear; grcat conf.proftpd </var/log/dmesg.log | tail -n 20'

snmpget -v2c -c "$community_string" $device sysDescr.0 # or sysUpTime.0, sysName.0 - Alt: snmpbulkwalk -> gets all OOIDs
# SNMP port : 161
# LAG == Link Aggregation

nc -l -p 7777 > /dev/null # on receiver machine
pv -btr /dev/zero | nc $host 7777 # show live throughput between two machines
yes | pv -btr | ssh $host 'cat > /dev/null' # same through SSH

grep -Eo '[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}' # grep an IP

# Find wireless driver
lspci -vv -s $(lspci | grep -i wireless | awk '{print $1}')

# Non portable tools
slurm, iptraf, ntop, iftop, nethogs # this last one can show per-process bandwidth used
iperf # measure throughput between 2 points / saturate a network connection -> useful for testing
mininet # realistic virtual network, running real kernel, switch and application code, on a single machine

ipcalc < cidr $ip/X # get netmask, network address - FROM http://fossies.org/linux/privat/cidr-2.3.2.tar.gz/

/etc/ssmtp/{revaliases,ssmtp.conf} # Configure 'mail' command - Alt: mutt -> fake FROM with EMAIL en var :
```

```
http://stackoverflow.com/a/12158550

w3m > elinks > links > lynx # http://askubuntu.com/questions/15988/browse-internet-inside-terminal
lynx -dump -stdin # convert HTML to text
wget --random-wait -r -p -e robots=off -U mozilla http://www.example.com # Alt: axel.alioth.debian.org - can use multiple connections (and mirrors) to download one file
  -p --page-requisites : download all the files necessary to properly display a page: inlined images, sounds, CSS...
  -k --convert-links : convert the links in the document to make them suitable for local viewing
  --no-parent : do not ever ascend to the parent directory when retrieving recursively
  -A --accept acclist -R --reject rejlist : comma-separated list of filename suffixes or patterns to accept or reject
  -l --level=depth : default = 5
  -c --continue : continue getting a partially-downloaded file
  --spider : do not download pages, only check they exist. Useful e.g. with --input-file bookmarks.html
curl --fail --insecure --request POST --header "$(< $headers_file)" -d @data_file # --trace-ascii - - http://curl.haxx.se/docs/httpscripting.html - Alt: jakubroztocil/httpie
# Web scrapping:
httrack
Xdummy > Xvfb # in-memory X11 display server that doesn't render to screen
pjscrape, PhantomJS, SlimerJS, CasperJS
GreaseMonkey/TamperMonkey, ChickenFoot, Scrapbook, iMacros, DejaClick # FF extensions
Selenium, Scrapy, RoboBrowser, FlexGet, ghost.py, splinter, binux/pyspider # python crawling libs
kimono, import.io
parklemotion/nokogiri # Ruby gem

python -m webbrowser -t "$url"
urlwatch --urls=urls-list.txt | ifne mutt -s "Page change detected" $email_address


_'_"_'_"_'_"_'_"_'_"_'_"_
#    ssh@  SSL  :ssh    #
_"_'_"_'_"_'_"_'_"_'_"_'_
keithw/mosh # faster 'ssh' replacement that allows the client and server to "roam" and change IP addresses, while keeping the connection alive
liftoff/GateOne # HTML5-powered terminal emulator and SSH client - Also: http://en.wikipedia.org/wiki/Comparison_of_SSH_clients#Platform
cat ~/.ssh/id_rsa.pub | ssh $user@$host "mkdir -p ~/.ssh && cat >>  ~/.ssh/authorized_keys" # Alt: ssh-copy-id $user@$host
ssh $host "$(printf "%q" $(cat script.sh))" # %q adds escapes to any string
ssh $host "$cmds ; /bin/bash -i" # Keep ssh session open after executing commands
ssh -f $host -L 2034:$host:34 -N # port forwarding
[ENTER] ~. # Exit a hung SSH session
# Force a user (based on its pub key) to only run one command one a host (e.g. tail -f) using ~/.ssh/authorized_keys : cf. tmux example
# How to change your login on a specified acces: http://orgmode.org/worg/worg-git-ssh-key.php
/etc/ssh/sshd_config # SSH daemon config to allow UNIX user/pswd auth: PasswordAuthentication yes, UsePAM yes OR AllowGroups sshusers
/etc/pam.d/* # use pam_unix.so
knockd # port knocking server
cat $file.key $file.crt > $file.pem
openssl s_client -CApath $ca -cert $pem -key $key -connect $host:443 -ssl3 # bare SSL client
openssl x509 -text -noout -in $cert.pem # get certs details
openssl x509 -inform der -in $cert.cer -out $cert.pem # convert .cer to .pem
keytool -printcert -file $cert.pem # get certs details
sshfs # && fusermount -u
Russell91/sshrc # bring your .*rc with you
mussh \ # MUltihost SSH Wrapper - Also: fabfile.org
 -l $USER \
 -m 2 \ # run on two hosts concurrently
 -h rpi-1 rpi-2 \ # hostnames
 -c "$cmd"


</-/--------------\-\>
<!<! Apapapapache !>!>
<\-\--------------/-/>
source /etc/apache2/envvars && apache2 -V # -l -L -M
sudo bash -c 'source /etc/apache2/envvars && apache2 -t && apache2ctl -S' # check config
vim /etc/apache2/sites-available/default-ssl
service apache2 restart
tail -F /var/log/apache2/*.log
a2enmod / a2dismod $modname  # enable / disable std modules
ab -n5000 -c50 "http://path/to/app?params" # Apache benchmarking - Alt: tarekziade/boom


=cCcCcCc=
# Cisco #
=cCcCcCc=
# Remote-cmd & monitor device config: RANCID/clogin
```

```
enable # unlock more comnmands
show version
exit

show logging [buffered]

sh run
sh int
sh ip int [brief]
sh ip rou 1.2.3.4

# for Fastpath, e.g. QuantaLB:
show logging hosts
show logging buffered
traceroute $ip


-%-%-%-%-%-
 =SYSTEM=
-%-%-%-%-%-

powertop # diagnose issues with power consumption
sysctl

cat /etc/*-release
lsb_release -a
uname -a
cat /etc/issue*

/proc/version
/proc/cpuinfo # Number of cores, cache size & alignement...
watch -d 'cat /proc/meminfo' # Watch system stats
/proc/sys/fs/file-nr # allocated/free file descriptors
/proc/loadavg : # graph in TTY: tload - Alt: uptime
- first 3 fields : number of jobs in the run queue (state R) or waiting for disk I/O (state D) averaged over
1, 5, and 15 minutes
- 4th field : number of currently executing kernel scheduling entities (processes, threads) / number of
existing kernel scheduling entities
- 5th field : PID of last process created

echo 1 > /sys/module/printk/parameters/printk_time # Enable dmesg timestamps
dmesg -s 500000 | grep -i -C 1 "fail\|error\|fatal\|warn\|oom" # In case of OOM, Linux kernel will kill the
process with highest /proc/$pid/oom_score - To exclude a process from the OOM killer list: echo "-17">
/proc/$pid/oom_adj

watch -d -n 1 "cat /proc/$pid/status | grep ctxt_switches" # mostly nonvoluntary context switches => CPU bound
/ else IO bound - FROM: https://blogs.oracle.com/ksplice/entry/solving_problems_with_proc

# Monitoring
iostat # ! '%util' & 'svctm' are misleading + iotop, non portable + brendangregg/perf-
tools/blob/master/iosnoop
mpstat 5 # cpu usage stats every 5sec
monit # monitor processes, network stats, files & filesystem. Has an HTTP(s) interface, custom alerts
dstat
pt-summary, glances, psdash, conky
collectd, perfwatcher
hdparm -tT /dev/sda # Check a disk read/write speed

stap # SystemTap
perf # aka perf_events, needs a version of linux-tools-* matching the kernel - More:
http://www.pixelbeat.org/programming/profiling/
    top -G
    stat -e cycles,instructions,cache-misses,dTLB-load-misses -p $PID
tobert/pcstat # page cache stats for files

# Checking Swap Space Size and Usage
free -m # how much free ram I really have ? -> look at the row that says "-/+ buffers/cache"
vmstat 2
sar # provides history data

w / who # users currently logged
last [-f /var/log/wtmp.1] # previous logged users
dump-utmp /var/run/utmp # or /var/log/wtmp
lastcomm # or dump-acct pacct : list last executed commands. From acct pkg, must be turned on with
/etc/init.d/psacct start
# Alt (very resource consuming): auditctl -a task,always; ausearch -i -sc execve

/etc/motd # Message of the day, can be combined from multiple files: man update-motd

# Get uid / groups infos
id $USER # for primary group, use -ng flag
```

```
adduser / usermod -a -G # DO NOT FORGET THE -a !!!
useradd -m -G sudo,sshusers -p $(openssl passwd ******)

# Add a Linux secondary group without logging out
newgroup $new_secondary_group
newgroup $original_primary_group

awk -F":" '{ print "username: " $1 "\t\tuid:" $3 }' /etc/passwd # List system users
/etc/shadow # $encryption_id$salt$encrypted - can be checked with mkpasswd -$encryption_id $salt $password (or
'openssl passwd') - To check a user/sudo password, cf. http://askubuntu.com/a/276182

sudo su -l # login as user root
sudo -K # Remove sudo time stamp => no more sudo rights
fakeroot # runs a command in an environment wherein it appears to have root privileges for file manipulation
chroot $path_to_fake_root $cmd # 'chroot jail' => changes the apparent root directory

faketime $time_spec $cmd

fdisk -l # has flaws, better use bitbucket.org/skypher/fdisk
testdisk # disk data recovery
lsusb # Alt: usb-devices
lspci -v # list devices
lshw -C disk # list disks : ata, cdrom, dvdrom
blkid # list UUIDs
dmidecode
/sbin/mdadm --examine --scan --verbose # need root - RAID config
shutdown -r -F now # force FCSK disk check - Or: touch /forcefsck - Alt:
smartctl -a /dev/sdb2 # scan a device - Alt: gsmartcontrol or above

# Find what package a command belong to:
apt-file search /path/to/anyfile
yum provides $cmd
dpkg -S /path/to/cmd
rpm -qif $(which cmd)
rpm -Uvh pkg.rpm # upgrade RPM
apt-get source $pkg
apt-cache search $keyword
apt-cache rdepends $pkg # list dependencies
rpm -q --whatrequires $pkg # list dependencies

apt-key fingerprint # display imported keys fingerprints
sudo dpkg -D1 -i *.deb

rpm --qf "%{INSTALLTIME:date} %{NAME}-%{VERSION}-%{RELEASE}.%{ARCH}.rpm\n" -qa *regex* # list rpm
rpmbuild file.spec
alien # transformer un .rpm en .deb

init q # Reload upstart config : /etc/inittab, /etc/init.d, /etc/init/*.conf -> can be really simple & useful
init-checkconf # check upstart script syntax
initctl list # list active upstart services
chkconfig, service # control & check upstart scripts
# Alt & init.d example: http://support.ghost.org/deploying-ghost/
# /etc/init/ script example:
start on startup
script
    set -o errexit -o nounset -o xtrace # NOT -o pipefail or script won't start
    cd $dir
    exec >> etherpad-upstart.log
    exec 2>> etherpad-upstart.log
    date
    exec start-stop-daemon --start -c $user --exec /path/to/exec
end script

xev # Listen to keyboard events
loadkeys fr # Change keyboard to FR
setxkbmap -print # print keyboard config
numlockx # Toggle numpad key locking

mplayer -identify -vo null -ao null -frames 0 $file | grep "Video stream found" # Identify video
mencoder vid.wmv -o vid.avi -ofps 25 -ni -ovc lavc -oac mp3lame # Convert .wmv to .avi
avconv -i vid%02d.mp4 -vcodec copy -acodec copy vid.avi # .mp4 to .avi - Replacement for ffmpeg - GUI: Adapter
avconv -i $video_file -r 1 -an "videoframe%03d.png" # extract images from a video with FPS=1

winetricks $dll # install one of: winetricks list dlls
wine uninstaller # real files are in ~/.wine/


&*&*&*&*&*&*&*&*
~= Issues fixes =~
&*&*&*&*&*&*&*&*
# Resurect computer : http://en.wikipedia.org/wiki/Magic_SysRq_key
```

```
echo <ctrl-v><ctrl-o> # or 'reset', fix terminal frenzy

sudo ldconfig

install myspell-fr # LibreOffice SpellCheck

killall gnome-settings-daemon # Fix crazy numpad (no '-')
sudo service lightdm restart # restart Gnome session / useful in case of a frozen X server
killall gnome-panel
killall unity-panel-service # restore displaying clock in Ubuntu, hidden when buggy

gsettings set org.gnome.desktop.media-handling automount false # disable automount

rm ~/.config/user-dirs.locale # can fix broken locale

# Audio/mike issues
pulseaudio -D
pavucontrol
alsamixer
gstreamer-properties

sudo /usr/share/doc/libdvdread4/install-css.sh # Install libdvdcss

sudo su -c 'echo 1 > /sys/bus/pci/rescan' # Rescan for memory card

xhost +local:root # Xlib: connection to ":0.0" refused by server

xdg-mime default lighttable.desktop text/x-markdown # Also: mimetype $file

/var/log/kern.log EMPTY # needs $ModLoad imklog in /etc/rsyslog.conf + service rsyslog restart (thx:
http://serverfault.com/a/405244 ): BUT:
# -> "imklog: error reading kernel log - shutting down: Bad file descriptor" + CPU maxing out. Web search =>
looks like a known issue solved with more recent versions of rsyslog

sudo lsof -s | grep deleted | grep -Ev '/dev/|/run/' | awk '$5 == "REG"' | sort -n -r -k 7,7 # find deleted
files that are still using space on disk


FFFFFFFFFFFFFF
F i r e f o x
FFFFFFFFFFFFFF
~/.mozilla/firefox/*.default/mimeTypes.rdf # FIREFOX 'open with' mapping
find Cache/ -type f -exec file {} \; | grep image | cut -d':' -f1 # all cached images
about:cache # Firefox cache infos: location, size, number of entries
about:memory # Firefox memory allocation details
about:about # all the about: pages e.g. :crashes :healthreport :permissions :plugins :sessionrestore
$ff_profile_dir/.parentlock # fix "Firefox is already running but is not responding" error
cp sessionstore.bak sessionstore.js # Restore previous session tabs
<CTRL>+F5 # refresh page bypassing the cache
MAJ+F2: screenshot --fullpage $filename # PNG screenshot of the webpage - Alt: http://freze.it

https://developer.mozilla.org/en-US/docs/Tools/Web_Console
- inspect(), pprint()
- console.time(name) .timeEnd(name) .profile(name) .profileEnd(name)
- cd("#frame1"); # get into a specific iframe
- $("css selector") or $$() for ALL matches; $x("xpath expression")
//div[contains(concat(' ',normalize-space(@class),' '),' foo ')] # http://pivotallabs.com/xpath-css-class-
matching/

=\/=/\=\/=/\=\/=
=  Virtualbox
=\/=/\=\/=/\=\/=
sudo adduser $USER vboxusers # then logout
VBoxManage list vms
VBoxManage controlvm $name poweroff
VBoxClient --clipboard
$HOME/VirtualBox VMs/{machinename}/Logs

# Cool features : remote display (VRDS), shared folders & clipboard, seamless mode


()()()()()()()()()()
() SynthÃ¨se vocale
()()()()()()()()()()
espeak -s 180 -p 40 "Hey ! Look behind you"
espeak -s 180 -p 40 -ven+12 "Hi ! My name is Colossus."
espeak -s 150 -p 20 -vfr "Je vais te pÃ©ter la gueule"
espeak -v mb/mb-fr1 -s 50 'Je peux parler plus lentement' | mbrola /usr/share/mbrola/voices/fr1 - -.au | aplay
#FROM:   http://doc.ubuntu-fr.org/synthese_vocale
#        http://linux.byexamples.com/archives/303/text-to-speech-synthesizer/
#        http://cookerspot.tuxfamily.org/wikka.php?wakka=SyntheseVocaleEspeak


::=::=::=::
: MAC OSX :
::=::=::=::

curl http://google.com/ | base64 | say # FUN

dns-sd -Q $USER.local # mDNS query

sudo softwareupdate -i -a # Manual software update

Finder > Applications > Utilities > Disk Utility # Repair permissions

system_profiler # list system components, ports...
pmset -g # power management settings

pbpaste | pbcopy # clipboard

textutil -convert txt # or -info : convert / get infos on files

xattr -l $file # File listed with '@' => extended attributes

sudo dseditgroup -o edit -a $USER -t user $GROUP # Add user to group

find $(ls | grep -Ev 'Library|Documents|Downloads|httrack|phantomjs|vitavermis') \( ! -path '*/.*' \) -type f
-print0 | xargs -0 stat -f '%m %N' | sort -k 1nr | while read timestamp file; do echo $(date -jf "%s"
$timestamp "+%F") $file; done | less # illustrate how to replace find -printf + timestamp conversion + find
non-hidden files only ; GOAL: list files by modification date

# DTrace scripts: man -k dtrace
iosnoop # or better hfsslower.d from the DTrace book, available online
execsnoop # trace processes created
opensnoop -ve # trace open files, also maclife.d from DTrace book to trace files creation/deletion
dtruss -d # strace
soconnect_mac.d # trace TCP connections, from DTrace book
errinfo # trace system call fail
bitesize.d # trace I/O
iotop

# C#
NUNITLIB=/Library/Frameworks/Mono.framework/Versions/2.10.11/lib/mono/2.0/nunit.framework.dll
gmcs -debug -t:library -r:$NUNITLIB *.cs
nunit-console *.dll
mono *.exe

# AppleScript
#!/usr/bin/osascript
on log(msg)
  set log_line to (do shell script "date  +'%Y-%m-%d %H:%M:%S'" as string) & " " & msg
  do shell script "echo " & quoted form of log_line
end log
log "HELLO WORLD !"


]_]_]_]_]_]_]_]
] ImageMagick
]_]_]_]_]_]_]_]
# Compile IM with HDRI:
# - http://www.imagemagick.org/script/install-source.php
# - sudo aptitude install libmagickcore-dev liblcms2-dev libtiff4-dev libfreetype6-dev libjpeg8-dev liblqr-1-
0-dev libglib2.0-dev libfontconfig-dev libxext-dev libz-dev libbz2-dev
# - ./configure --enable-hdri
# - identify -version # to check HDRI is enabled
# Scripts: http://www.fmwconcepts.com/imagemagick/
display $img_file
convert img.png -adaptive-resize 800x600 -auto-orient -crop 50x100+10+20 img.jpg
mogrify ... *.jpg # for f in *.jpg; do convert $f ... ; done
identify -v $img_file # get PPI: -format "%w x %h %x x %y"
import -display :0.0 -window root screenshot.png # Alt: gnome-screenshot --interactive # Or Gimp
animate -delay 5 *.png
compare img1 img2
composite # merge images

gifsicle "$gif" -I | sed -ne 's/.* \([0-9]\+\) images/\1/p' # frames count + cf. stopmo_logo/gen_anim.sh
convert $(for f in *.png; do echo -delay 5 $f; done; ) -rotate -90 -resize 50% -loop 0 out.gif
tesseract-ocr # Google OCR / text extraction - http://askubuntu.com/a/280713/185582
qrencode -o $png $url && zbarimg -q $png # from zbar-tools - Can generate ASCII ! - Alt: Python qrcode
barcode -b "Hello World !" -o out.ps && convert out.ps -trim out.png
pngquant ## 70% lossy compression
```

```
jpegtran -optimize -progressive -grayscale -outfile $out_file $in_file # FROM: libjpeg-turbo-progs
identify -verbose $jpg | grep -Fq 'Interlace: JPEG' # is JPEG progressive ? Alt: grep -Fq "$(echo -en
"\xff\xc2")" $jpg
mat # Metadata Anonymisation Toolkit, removes e.g. images hermful metadata
feh -F -d -D 3 --cycle-once * # fast image viewer: fullscreen slideshow with 3s delay - Alt: gpicviw


$$$$$$$$$$$$$$
$ Google APIs $
$$$$$$$$$$$$$$
sqrt(cos(x))*cos(200 x) + sqrt(abs(x))-0.7)*(4-x*x)^0.01, sqrt(9-x^2), -sqrt(9-x^2) from -4.5 to 4.5 # Google
it & profit !

# Search tips&tricks : https://support.google.com/websearch/answer/136861
site:$base_url "exact match" OR "a * saved is a * earned" -term # basics
inurl:gouv.fr # Also: intitle:
filetype:pdf
cache:$url
define:$term
related:$url
link:$url # Search for pages that link to a URL
https://www.google.fr/search?q=5%2B(-sqrt(1-x^2-(y-abs(x))^2))*sin(100*((10-x^2-(y-
abs(x))^2))),+x+is+from+-1+to+1,+y+is+from+-1+to+1.5,+z+is+from+1+to+6 # 3D heart surface

youtube-dl --ignore-errors --extract-audio FLF8xTv55ZmwikWWmWLPEAZQ # download playlist as .m4a files - in
case of HTTP error 500, try -f18

# Snippet-search
cse_id=003799500572498885021:6zbuscnifvi
curl -s "https://www.googleapis.com/customsearch/v1?
key=${api_key}&cx=${cse_id}&fields=items(snippet)&q=define%20${term}"
# DOCS: https://developers.google.com/custom-search/json-api/v1/using_rest
https://developers.google.com/custom-search/json-api/v1/performance#partial


{[{[{[{[{[{[{[{[
 'AWS': "cli"
}]}]}]}]}]}]}]}]}
aws configure # eu-west-1
aws iam list-user-policies --user-name lucas # Also: aws iam list-roles
aws s3 cp $file s3://lucas-pail/ # Other cmds: mb rb ls rm mv
# AWS Lambda - mostly from http://alestic.com/mt/mt-search.cgi?blog_id=1&tag=AWS%20Lambda
aws lambda list-functions
aws lambda invoke-async --function-name $function --region us-east-1 --invoke-args inputfile.json --debug
aws logs describe-log-groups --region us-east-1
log_group_name=/aws/lambda/$function
log_stream_names=$(aws logs describe-log-streams --region us-east-1 --log-group-name "$log_group_name" --
output text --query 'logStreams[*].logStreamName')
for stream in $log_stream_names; do
    aws logs get-log-events --region us-east-1 --log-group-name "$log_group_name" --log-stream-name "$stream"
--output text --query 'events[*].message'
done | less


/./././
 /irc
/./././
http://fr.wikipedia.org/wiki/Aide:IRC/commandes
http://www.ircbeginner.com/ircinfo/ircc-commands.html

# Weechat - Alt: irssi bitlbee or mcabber for Jabber only
weechat --run-command '/set;/quit' > ~/dump-weechat-config
tF /home/lucas/.weechat/weechat.log /home/lucas/.weechat/logs/*
http://weechat.org/files/temp/scripts/hdata.py # install with '/python load hdata.py' - Also: hdata_update.py
/script search iset # then 'i' to install

/server list[full] [server]
/connect freenode
/nick dr_max_kurt
/join #laquadrature
/set irc.server.freenode.autoconnect on
/set irc.server.freenode.nicks "dr_max_kurt"
/set irc.server.freenode.sasl_ ...
<ALT>+<ARROW> # switch window


~~~~~~~~~~~~~
-> Freeplane
~~~~~~~~~~~~~
# SHORTCUTS, cf. Help > Key Reference - To add new ones: <CTRL>+<click> a menu item
<Enter> / <Insert> : new sibling / child node
```

```
<CTRL>+<double click> : new free node

<CTRL>+L : link selected nodes
<CTRL>+<SHIFT>+K : add hyperlink, <CTRL>+K to modify

<Space> : Toggle folding
<ALT>+<left> : 'back', go to previously visited node

# Customized shortcuts
<CTRL>+F? : apply style "Level ?"
<CTRL>+1 : add image (from 'Node extensions' menu, so that they cab resized)
<CTRL>+<SHIFT>+1 : add icon
<CTRL>+3 : node color
<CTRL>+<SHIFT>+3 : node background color
```