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

~°~°~°~°~°~°~°~°~°~°~°~
### SHELL & misc  ###
~°~°~°~°~°~°~°~°~°~°~°~
<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
```

```
 73
 74  dig +short TXT google-public-dns-a.google.com # check without 'TXT'
 75  dig +short TXT istheinternetonfire.com
 76  traceroute -m 60 216.81.59.173; telnet towel.blinkenlights.nl # Star Wars
 77
 78  echo "You can simulate on-screen typing just like in the movies" | pv -qL 10
 79
 80  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=$(( $a + 1 )) ; sleep 0.001 ;done; } # FROM: http://www.climagic.org/co
     olstuff/cursor-tricks.html
 81
 82  ( play -q -n synth sine F2 sine C3 remix - fade 0 4 .1 norm -4 bend 0.5,2399,2 fade 0 4.0 0
     .5 & )
 83  echo 'main(t){for(;;t++)putchar(((t<<1)^((t<<1)+(t>>7)&t>>12))|t>>(4-(1^7&(t>>19)))|t>>7);}
     ' | cc -x c - -o crowd && ./crowd | aplay
 84
 85
 86  ###################
 87    Bash scripting
 88  ###################
 89  figlist | sed '1,/Figlet fonts/d;/:/,$d' | xargs -I{} figlet -f {} Hello # ASCII banner fon
     ts
 90  xmessage -center "$(figlet ERRORMSG 42)", notify-send (libnotify), bar, dialog, gdialog==ze
     nity # GUI: error windows, selection dialog, progress bars...
 91  mooz/percol | peco/peco | moreutils/vipe # interactive filtering through pipes
 92
 93  set -o pipefail -o errexit -o nounset -o xtrace # can be read / exported to subshells using
      $SHELLOPTS
 94  export PS4='+ ${FUNCNAME[0]:+${FUNCNAME[0]}():}line ${LINENO}: '
 95
 96  bash -n $script # Check syntax without executing
 97  bash --debugger $script
 98
 99  parent_func=$(caller 0 | cut -d' ' -f2) # "$line $subroutine $filename"
100  source ~/sctrace.sh # FROM: http://stackoverflow.com/questions/685435/bash-stacktrace/68609
     2
101
102  readonly EXEC_DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )" # Script file parent
     dir
103  readonly LOG_FILE="$EXEC_DIR/logs/$(basename $0).log.$(date +%Y-%m-%d-%H)"
104  exec > >(tee -a $LOG_FILE); exec 2>&1
105  date "+%F %T,%N" | cut -c-23 # Standard logs date
106  date -u +%s # Seconds since EPOCH
107  date -d @$seconds_since_epoch "+%F" # under OSX: date -jf "%s" $secs "+%F"
108
109  # !! aliases used in functions definitions are immediately substituted,
110  # NOT resolved dynamically !
111  alias foo='echo A'
112  bar () { foo; }
113  alias foo='echo B'
114  bar # echo A
115
116  # Set positional parameters $0 $1 ...
117  set - A B C
118
119  : ${1:?'Missing or empty parameter'}
120  : ${var:="new value set if empty"}
121  local var=${1:-"default value"}
122  foo () { local x=$(false); echo $?; }; foo # -> 0 !!GOTCHA!! 'local' is also a command, and
      its return code shadows the one of the cmd invoked
123
124  echo ${PWD//\//-} # Variables substitutions (http://tldp.org/LDP/abs/html/parameter-substit
     ution.html)
125  ${var%?} # Remove the final character of var
126
127  for pair in $whatever; do key=${pair%:*}; value=${pair#*:}; ...
128  for f in ./*.txt; do; [[ -f "$f" ]] || continue # Safe 'for' loop - http://bash.cumulonim.b
     iz/BashPitfalls.html
129
130  readonly CONST=42 # works with arrays & functions too - Beautiful hack to unset: http://sta
     ckoverflow.com/a/21294582
131
132  # Q: Can we find a function 'identity' that satisfies the following 2 properties ? stackove
     rflow.com/q/21635301
133  identity () { for arg in "$@"; do echo "$arg"; done; }
134  identity "$(identity a\ b c\ d)"
135  # a b
136  # c d # expected output: OK
137  argv_count () { echo "argv_count($@):$#" >&2; }
138  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] ]] ; } ; }
```

```
214
215 is_file_open () { lsof | grep $(readlink -f "$1") ; }
216
217 cat <<EOF
218 EOF
219
220 exec 8<>filename # Open file descriptors #8 for reading and writing
221 echo BlaBlaBla
222 exec 8>&- # Close file descriptor
223
224 /var/tmp is better than /tmp # as filling it is less system impacting
225 tdir="$(mktemp -d ${TMPDIR:-/tmp}/$0_XXXXXX)" # mktemp dir & default value
226 /dev/shmi # Use RAM for tmp files - monitor usage with ipcs -m
227
228 for i in {0..255}; do printf "\x1b[38;5;${i}mcolor${i}\x1b[0m\n"; done # display all 256 co
    lors
229 for i in {1..8}; do echo "$(tput setaf $i)color_$i$(tput sgr0)"; done # colored terminal ou
    tput
230 # + colors can be set like this: tput initc 2 500 900 100 # RGB values between 0 & 1000
231 # Other tput: setab [1-7], setf [1-7], setb [1-7], bold, dim, smul, rev... cf. man terminfo
232 tput sc;tput cup 0 $(($(tput cols)-29));date;tput rc # put a clock in the top right corner
233
234 select value in choice1 choice2; do break; done # multiple choices
235 read -s password # 'silent' user input, no characters are displayed
236 strings /dev/urandom | grep -o '[[:alnum:]]' | head -n 30 | tr -d '\n' # 30 characters pass
    word generation
237 stty -echo # disable TTY output
238 stty -echo -icanon time 0 min 0 # non-blocking read trick FROM: http://stackoverflow.com/a/
    5297780
239 # ask a yes or no question, with a default of no.
240 echo -n "Do you ...? [y/N]: "
241 read answer
242 if expr "$answer" : ' *[yY].*' > /dev/null; then
243    echo OK
244 else
245    echo KO
246 fi
247
248 set -o noglob # disable wildcard expansion
249 # Extended bash globbing
250 shopt -s extglob # http://www.linuxjournal.com/content/bash-extended-globbing
251 shopt [-o] # list options values. Alt: $- E.g. check if shell is interactive: [[ $- =~ i ]]
     - Also, is stdin open in a terminal: [ -t 0 ]
252
253 ( set -o posix; set ) # List all defined variables
254 foo=bar; foo () { :; } ; unset foo # !!GOTCHA!! the variable is unset first, then the funct
    ion if called a 2nd time
255 # Get all commands prefixed by (useful for unit tests)
256 compgen -abck unit_test_ # control readline auto-completion (help complete), can be enable
    by '-e' flag of 'read'
257 complete -f -X '!*.ext' command # exclude files using a filter
258 complete -F _compfunc command
259 _compfunc() {
260     local cmd="${1##*/}"
261     local word=${COMP_WORDS[COMP_CWORD]}
262     local line=${COMP_LINE}
263     local xpat='!*.foo'
264
265     COMPREPLY=($(compgen -f -X "$xpat" -- "${word}"))
266 }
267 hash # frequently used commands cache
268
269 syslogd -m 0 -r -SS # port: 514
270 logger -is -t SCRIPT_NAME -p user.warn "Message"
271 echo "<15>My logline" | nc -u -w 1 $HOSTNAME 514 # <15> means 'user.debug', see RFC3164: Fa
    cility*8 + Severity, default:13 <-> user.notice
272 time tcpdump udp and dst port 514 | awk '{print $3" "$7}' | sed 's/\.syslog//' > noisy_devi
    ces
273 logcheck, logtail
274 petit --hash /var/log/messages # Cmdline log analyze, also --wordcount. Alt: lnav ; sysdig
    -c spy_syslog
275
276 #   --reject doesn't apply to the whole path, only to the filename/query
277 mv $file ${file%.*}.bak # Change extension
278 mv --backup=numbered new target # !! --suffix/SIMPLE_BACKUP_SUFFIX can be broken on some di
    stros
279 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/
280 # !! $@ not supported if < v.7.5
281
282 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, us
```

```
    e 644 permissions owned by root, and note that the $USER arg is not present in /etc/crontab
     files
283 sudo grep crontask /var/log/cron.log
284 flock -n /pathi/to/lockfile -c cmd # run cmd only if lock acquired, useful for cron jobs
285 lockfile-create/remove/check # file locks manipulation
286 while true do inotifywait -r -e modify -e create -e delete -e move_self . ./run.sh done # i
    notify-tools based keep-alive trick - Alt: ayancey/dirmon
287 huptime --exec $cmd # zero downtime restarts of unmodified (networking) programs, intercept
     bind(2) and accept(2) calls
288
289 # Launch command at a specified time or when load average is under 0.8
290 echo $cmd | at midnight
291 echo $cmd | batch
292
293 nice / ionice / renice # Control process priority (useful in cron job)
294 # control the resources available to the shell and to processes it starts
295 ulimit -v # max virtual memory
296 ulimit -s # max stack size
297 ulimit -t # max of cpu time
298 ulimit -u # max number of processes
299 ulimit -a # ALL - Also: /proc/$pid/limits
300
301 mkfifo /tmp/myfifo; exec 3<> /tmp/myfifo # Über trick: dummy FD => non-blocking named-pipe
302 python -c "from fcntl import ioctl ; from termios import FIONREAD ; from ctypes import c_in
    t ; from sys import argv ; size_int = c_int() ; fd = open(argv[1]) ; ioctl(fd, FIONREAD, si
    ze_int) ; fd.close() ; print size_int.value" $fifo # readble bytes in a fifo -> NOT RELIABL
    E, e.g. always return 0 with non-blocking named-pipe
303 ulimit -p # should get max pipe size, but WRONG : defined in pipe_fs_i.h
304 fcntl(fd, F_SETPIPE_SZ, size) # to change max size, if Linux > 2.6.35 (/proc/sys/fs/pipe-ma
    x-size)
305
306 gnuplot -e "set term dumb; plot '<seq 1 9'" # ASCII graph - Alt, with UTF8 & colors: https:
    //github.com/tehmaze/diagram
307 gnuplot -e "set term dumb size 200,50; plot [-5:6.5] sin(x) with impulse"
308 loop_cfg_file=/tmp/gnuplot_loop.cfg
309 in_data_file=/tmp/gnuplot_in.data
310 echo <<EOF >$loop_cfg_file
311 set term dumb size 200,50
312 set title 'Traffic-In (bytes/s)'
313 # Sampling 30 data points each time
314 plot '<tail -n 30 $in_data_file'
315 pause 1
316 replot
317 # Loop by rereading this file, doesn't work with -e on the command-line
318 reread
319 EOF
320 tail -F $log_file | grep $keyword | pv --line-mode --numeric >/dev/null 2>$in_data_file & #
     Alt: petit --sgraph
321 gnuplot $loop_cfg_file # real-time ASCII graphing !
322
323
324 +++++++++++++++
325 | Text stream |
326 +++++++++++++++
327
328 cat -vET # shows non-printing characters as ascii escapes.
329 printf "\177\n" # echo non-ascii, here 'DEL' in octal. echo $'\177' is equivalent, BUT:
330 # echo $'A\0B' -> A
331 # printf 'A\0B\n' -> AB
332
333 aha # convert ANSI colors into HTML tags
334 make 2>&1 | colout -t cmake | colout -t g++ # from nojhan github: "Color Up Arbitrary Comma
    nd Output"
335
336 # grep Alt: ack, grin, ag (ggreer/the_silver_searcher), pt (monochromegane/the_platinum_sea
    rcher)
337 grep -a # if "Binary file (standard input) matches"
338 grep -q # silent, !! FAIL with SIGPIPE if 'pipefail' is used: http://stackoverflow.com/a/19
    120674/636849
339 grep '\<word\>' # match word-boundaries
340 grep -I # ignore binary files
341 grep -R --include='*.py' --exclude-dir='build/'
342 grep -o # output only matching parts
343 grep -C3 # output 3 lines of context, see also -B/-A
344 grep -H/-h # output with/without filename
345 grep -L $pattern $files # Get only filenames where PATTERN is not present
346 grep -P '^((?!b).)*a((?!b).)*$' # Grep 'a' but not 'b' -> PCRE ;  awk '/a/ && !/b/'
347 grep -P -n "[\x80-\xFF]" $file # Find non-ASCII characters
348 LANG=C grep -F # faster grep : fixed strings + no UTF8 multibyte, ASCII only (significantly
     better if v < 2.7)
349 sed -n '/FOO/,/BAR/p' # Print lines starting with one containing FOO and ending with one co
    ntaining BAR.
```

```
350 sed -e "9r rabbit.ascii" -e "6iTITLE" template.html # insert a file + a specific text line
    in another file
351 perl -ne '/(error|warn)(?!negative-look-ahead-string-to-not-match-just-after)/i'
352 perl -ne '/r[eg](ex)p+/ && print "$1\n"' # print only matching groups
353 grep | cut -c1-200 # truncate results to 200 characters
354 pyp # pip install --user pyp : alternative to sed, awk, cut & perl - Alt: pyped, Russell91/
    pythonpy
355
356 pdftotext $file.pdf - | grep # from xpdf-utils - Alt: euske/pdfminer pdf2txt.py OR pdftk OR
     LibreOffice Draw
357 gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite [-dPDFSETTINGS=/screen|/ebook|/printer|/prepress]
     -sOutputFile=$out.pdf $in.pdf # reduce pdf size with ghostscript - Also: -dFirstPage=X -dL
    astPage=Y - Alt: http://compress.smallpdf.com
358 pdfjam file1.pdf file2.pdf 1, 3- `# optional selector` --nup 2x1 --landscape --outfile out.
    pdf # printer-friendly version - Also: pdf290 to rotate
359
360 tr -c '[:alnum:]' _
361
362 # filter outpout : not lines 1-3 and last one
363 type ssh_setup | sed -n '1,3!p' | sed '$d'| sed 's/local //g'
364 # this is also a crazy hack : put the output in ORIG_CMD, then redefine ssh_setup () { eval
     $ORIG_CMD $@; ... }
365
366 perl -ne 'if (length > $w) { $w = length; print $ARGV.":".$_ };  END {print "$w\n"}' *.py #
     Longest line of code
367 $cmd | awk '{print length, $0}' | sort -rn # sort by line length
368 cloc # count lines of code
369
370 comm -12 #or uniq -d - Sets intersec - See also: "Set Operations in the Unix Shell" - Alt:
    moreutils/combine
371 join # join lines of two files on a common field
372
373 tee -a $file # display input to stdout + append to end of $file
374 echo ECHO | sed s/$/.ext/ # Append at the end of stdout (or beginning with ^)
375 sed -i "1i$content" $file # append at the beginning of $file
376
377 sed ':a;N;$!ba;s/PATTERN\n/PATTERN/g' # remove newlines after PATTERN - How it works : N me
    ans 'pattern_space+=\n+nextline' and we use branching to :a - Alt: just '1!N; s/.../'
378 seq 1 10 | paste -s -d+ | bc # Replace newlines by a separator, aka 'join' - Also, for arra
    ys: OLD_IFS=$IFS; IFS=+; echo "${argv[*]}"; IFS=$OLD_IFS
379 sed "0,/$pattern/d" $file # print only lines after $pattern
380 # paste is also useful to interlace files: paste $file1 $file2
381 tac # reverse lines
382
383 perl -pe 's/\s+/\n/g' # Break on word per line
384 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 ""}'
385 mawk # faster awk
386 fold # breaks lines to proper width
387 fmt # reformat lines into paragraphs
388 printf "%-8s\n" "${value}" # 8 spaces output formatting
389 | xargs -n 1 sh -c 'echo ${0:0:3}' # 3 first characters of $string
390
391 csv{cut,look,stat,grep,sort,clean,format,join,stack,py,sql} {in,sql}2csv # pip install csvk
    it
392
393 jq -r '..|objects|.name//empty' # JSON syntax highlighting + sed-like processing - Basic al
    t: python -mjson.tool
394 echo '{"A1":"a1","A2":"b2","B1":"b2"}' | jq '"A." as $regex | del(.[keys[]|select(match($re
    gex))])'
395 echo '{"A0":["a1","a2","a3"], "B0":["b1","b2","b3"], "c3":[]}' | jq '".[^3]" as $regex|to_e
    ntries|map(select(.key|match($regex)))|map(.value|=map(select(match($regex))))|from_entries
    '
396 source <(jq -r 'to_entries|.[]|"SAUCE_\(.key|ascii_upcase)=\(.value)"' .saucelabs_auth.json
     )
397 pup, html-xml-utils, xml2, 2xml, html2, 2html # convert XML/HTML to "grepable" stream - Als
    o: xmlstarlet & http://stackoverflow.com/a/91801
398
399 zcat /usr/share/man/man1/man.1.gz | groff -mandoc -Thtml > man.1.html # also -Tascii
400 txt2man -h 2>&1 | txt2man -T # make 'man' page from txt file
401 pandoc --standalone --smart --table-of-contents --include-in-header $css -f markdown -t htm
    l $f > ${f%%.md}.html # -s -S --toc
402 pandoc -s -f markdown -t man foo.md | man -l - # md2man : man pandoc_markdown
403 stmd foo.md | lynx -stdin # standard replacement for original 'markdown' command
404
405
406 =#=#=#=#=#=#
407    FILES
408 #=#=#=#=#=#
409
410 sleuthkit/scalpel # > foremost, file carving tool, cf. http://www.forensicswiki.org/wiki/To
    ols:Data_Recovery
```

```
411
412 ls | cut -d . -f 1 | funiq # Sum up kind of files without ext
413
414 find -D rates ... # details success rates of each match logic term
415 find / -xdev -size +100M -exec ls -lh {} \; # find big/largest files IGNORING other partiti
    ons - One can safely ignore /proc/kcore - Alt: man agedu (-s $dir then -w / -t) ; + all too
    ls listed in http://dev.yorhel.nl/ncdu
416 find . -type d -name .git -prune -o -type f -print # Ignore .git
417 find -regex 'pat\|tern' # >>>way>more>efficient>than>>> \( -path ./pat -o -path ./tern \) -
    prune -o -print
418 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
419 find . -mtime +730 -print0 | xargs -0 --max-args 150 rm -f # to avoid 'Argument List Too Lo
    ng' - Alt to mtime: -newer $than_this_file
420 fdupes -r $dir # find duplicate files: size then MD5 then byte-by-byte - Also: findimagedup
    es
421
422 rename \  _ * # Replace whitespaces by underscores
423
424 # To see all files open in a directory structure:
425 lsof +D /some/dir
426 # To see all files jeff has open:
427 sudo lsof -u jeff
428 # Additional useful option : -r $t : repeat the listing every $t second
429 fuser $dir # identify processes using files or sockets
430
431 namei / readlink -f # Shows Where a File/Directory Comes From (links, etc.)
432
433 killall -HUP $process_name # To tell a process to reload its file descriptors, e.g. when de
    leting a log file
434
435 sudo dd if=/dev/urandom of=FAKE-2012Oct23-000000.rdb bs=1M count=6000 # Create fake file
436 truncate -s $size_in_bytes $file # from coreutils
437
438 # setuid: When an executable file has been given the setuid attribute, normal users on the
    system who have permission to execute this file gain the privileges of the user who owns th
    e file within the created process.
439 # setgid: Setting the setgid permission on a directory (chmod g+s) causes new files and sub
    directories created within it to inherit its group ID
440 setcap # man capabilities
441 umask # Control the permissions a process will give by default to files it creates; useful
    to avoid temporarily having world-readable files before 'chmoding' them
442
443 setfacl -Rm u:"$user":rwx "$HOME/$dir" && setfacl -Rm d:u:"$user":rwx "$HOME/$dir" # Select
    ively gives access to another user - Also: getfacl
444 sudo chattr +i [-R] $file # Forbid file deletion - To check a file attributes: lsattr. Also
    : getfattr/setfattr
445
446 tune2fs # control extX file system parameters, e.g. reclaim disk space reserved to root
447 debugfs -R "stat <$(ls -i $file | awk '{print $1}')>" $(df $file | tail -n 1 | awk '{print
    $1}') # Get $file creation time ('crtime') on ext4 filesystems
448
449 # Bring back deleted file from limbo (ONLY if still in use in another process)
450 lsof | grep myfile # get pid
451 ls -l /proc/$pid/fd
452 cp /proc/$pid/fd/4 myfile.saved
453
454 # http://www.cyberciti.biz/tips/linux-audit-files-to-see-who-made-changes-to-a-file.html
455 auditctl -w $file -p wax -k $tag
456 ausearch -k $tag [-ts today -ui 506 -x cat]
457
458 Xfennec/cv # show progress & throughput of all running cp, mv, tar, gzip, cat...
459 rsync -v --progress --dry-run --compress $src_dir/ $dst_dir # Alt: rdiff-backup
460 --cvs-exclude --exclude=".*"
461 --archive # recursive + preserve mtime, permissions...
462 --delete # remove extra remote files
463 --backup --backup-dir=/var/tmp/rsync # keep a copy of the dst file
464
465 tar -czvf "$archive.tgz" "$dir_without_trailing_slash" # Extract: tar -xzvf $archive
466 tar -J... # instead of -z, .xz compression format support
467 pax > cpio > tar # http://dpk.io/pax
468 zipinfo $file.zip
469 pigz # paralell gzip, do not compress folders
470 yum install p7zip # for .7z files
471 lzop, lz4 # faster, use less CPU
472
473 sha{1,224,256,384,512}sum, md5sum, cksum
474
475
476 |°|°|°|°|°|°|°|°
477 == NETWORKING
478 |°|°|°|°|°|°|°|°
```

```
479
480 mtr $host > ping / traceroute
481 paris-traceroute > traceroute
482
483 socat > nc (netcat) > telnet # prefix with rlwrap ! Alt: stone -> a TCP/IP packet repeater in the application layer, avoid forking for each packet received
484 socat - udp4-listen:5000,fork # create server redirecting listening on port 5000 output to terminal
485 nc -l -u -k -w 1 5000
486 echo hello | socat - udp4:127.0.0.1:5000 # send msg to server
487 echo hello | nc -u -w 1 127.0.0.1 5000
488
489 # Port scanning
490 nmap -sS -O 127.0.0.1 # Guess OS !! Also try -A - Alt: p0f
491 nmap $host -p $port --reason [-sT|-sU] # TCP/UDP scanning ; -Pn => no host ping, only scanning
492 nmap 192.168.1.* # Or 192.168.1.0/24, scan entire subnet
493 nmap -DdecoyIP1,decoyIP2 ... # cloak your scan
494
495 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
496 lsof -i -n | grep ssh # list SSH connections/tunnels
497
498 netstat -ntap # To find which processes are sending packets
499 netstat --statistics --udp # global network statistics - 'ss' is the replacement for deprecated 'netstat', but this has no equivalent
500 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
501 ss -lp [-t|-u] # list only listening TCP/UDP sockets/ports
502
503 /proc/net/{snmp, netstat, ...} # network counters
504 dropwatch # to find out where are packets dropped
505 hping # packets crafting
506 mitmproxy --host # interactive examination and modification of HTTP traffic - cf. blog.philippheckel.com but no need for -T - Alt: CharlesProxy, BurpProxy, Fiddler on Windows
507 mitmdump # tcpdump-like: view, record, and programmatically transform HTTP traffic
508
509 # Dump all tcp transmission to a specific IP :
510 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
511 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
512 time tcpdump udp and dst port 514 -w /dev/null -c 1000 # Alt solution to estimate the rate
513
514 ip n[eighbour] # ARP or NDISC cache entries - replace deprecated 'arp'
515 ip a[ddr] [show|add $ip] dev eth0 # replace deprecated 'ifconfig'
516 ip link set eth0 [up|down] # enable/disable the[interface specified
517 ip tunnel list # list ssh stunnels replace deprecated 'iptunnel'
518 ip route # host routing tables - replace deprecated 'route'
519 iw # details about wireless interfaces - replace deprecated 'iwconfig'
520 iwlist wlan0 scan | grep GHz # get congestion of Wifi channels
521 MACADDR=$(ip address show eth0 | grep link/ether | awk '{print $2 }') # can be used to get a unique machine id number instead of using $RANDOM:
522 echo $((  16#$(echo $MACADDR | sed 's/://g') % 10000 )) # use base16 - ALT: use md5sum
523
524 # On RedHat / CentOS / Fedora
525 $EDITOR /etc/sysconfig/network-scripts/ifcfg-eth0
526 $EDITOR /etc/sysconfig/network
527 /etc/init.d/network restart
528 ifup, ifdown # bring a network interface up
529
530 ls /var/lib/dhc* # check what DHCP client is used
531 # Query DNS cmds > deprecated 'nslookup'
532 host [-t txt] $hostname # -a (all records) -v
533 dig +short NS $hostname # find authoritative nameservers
534 dig @$dns_server $hostname
535 dig +short -x $ip # Reverse DNS
536 dig +trace +norecurse txt $dns_server
537 avahi-resolve -n $USER.local # Multicast DNS == mDNS - from avahi-tools pkg
538 # Caching
539 /etc/hosts /etc/{host,resolv,nsswitch}.conf /etc/dhcp*/*.conf # manual / basic
540 bind / dnsmasq / lwresd / unbound # DNS daemon
541 nsscache / nss_db / nscd (broken: ignore TTL) # Cache /etc/{passwd,group,shadow,...} - Notes: nscd-aggstats, nscd -g
542 getent ahostsv4 www.google.com # whole query through NSS
543 rndc # display various DNS cache control commands, part of Bind9 tools suite
544 rndc -p 954 dumpdb -cache # dump the cache in $(find /var -name named_dump.db) ; lwresd $port can be figured out with lsof/nmap
545 # View queries bypassing lwresd
546 /usr/sbin/tcpdump -pnl -s0 -c150 udp and dst port 53 and src port not \
547     $(/usr/sbin/lsof -n -i4udp | awk '$1 == "lwresd" && $NF !~ /:921$/ { print substr($NF,3); exit }')
```

```
548
549 iptables -A INPUT -s $host -j DROP
550 iptables -A INPUT -p tcp -m tcp --dport 8888 -j ACCEPT
551 iptables -nvL --line-numbers # Also: iptables-save
552 iptables -D INPUT $rule_number
553 # Logging: connexion attempts will be traced in dmesg and, depending on syslog config, /var
    /log/kern.log
554 iptables -j LOG --log-level debug --log-prefix='[iptableslog] [dropped] ' -m limit --limit
    1/sec --log-prefix -A INPUT/OUTPUT
555 iptables -j LOG --log-level debug --log-prefix='[iptableslog] [new] ' -m state --state NEW
    -I INPUT/OUTPUT 1
556 watch --color 'dmesg --notime | xargs -IX printf "[$(date -u)] %s\n" X >> /var/log/dmesg.lo
    g; dmesg --clear; grcat conf.proftpd </var/log/dmesg.log | tail -n 20'
557
558 snmpget -v2c -c "$community_string" $device sysDescr.0 # or sysUpTime.0, sysName.0 - Alt: s
    nmpbulkwalk -> gets all OOIDs
559 # SNMP port : 161
560 # LAG == Link Aggregation
561
562 nc -l -p 7777 > /dev/null # on receiver machine
563 pv -btr /dev/zero | nc $host 7777 # show live throughput between two machines
564 yes | pv -btr | ssh $host 'cat > /dev/null' # same through SSH
565
566 grep -Eo '[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}' # grep an IP
567
568 # Find wireless driver
569 lspci -vv -s $(lspci | grep -i wireless | awk '{print $1}')
570
571 # Non portable tools
572 slurm, iptraf, ntop, iftop, nethogs # this last one can show per-process bandwidth used
573 iperf # measure throughput between 2 points / saturate a network connection -> useful for t
    esting
574 mininet # realistic virtual network, running real kernel, switch and application code, on a
     single machine
575
576 ipcalc < cidr $ip/X # get netmask, network address - FROM http://fossies.org/linux/privat/c
    idr-2.3.2.tar.gz/
577
578 /etc/ssmtp/{revaliases,ssmtp.conf} # Configure 'mail' command - Alt: mutt -> fake FROM with
     EMAIL en var : http://stackoverflow.com/a/12158550
579
580 w3m > elinks > links > lynx # http://askubuntu.com/questions/15988/browse-internet-inside-t
    erminal
581 lynx -dump -stdin # convert HTML to text
582 wget --random-wait -r -p -e robots=off -U mozilla http://www.example.com # Alt: axel.alioth
    .debian.org - can use multiple connections (and mirrors) to download one file
583   -p --page-requisites : download all the files necessary to properly display a page: inlin
    ed images, sounds, CSS...
584   -k --convert-links : convert the links in the document to make them suitable for local vi
    ewing
585   --no-parent : do not ever ascend to the parent directory when retrieving recursively
586   -A --accept acclist -R --reject rejlist : comma-separated list of filename suffixes or pa
    tterns to accept or reject
587   -l --level=depth : default = 5
588   -c --continue : continue getting a partially-downloaded file
589   --spider : do not download pages, only check they exist. Useful e.g. with --input-file bo
    okmarks.html
590 curl --fail --insecure --request POST --header "$(< $headers_file)" -d @data_file # --trace
    -ascii - - http://curl.haxx.se/docs/httpscripting.html - Alt: jakubroztocil/httpie
591 # Web scrapping:
592 httrack
593 Xdummy > Xvfb # in-memory X11 display server that doesn't render to screen
594 pjscrape, PhantomJS, SlimerJS, CasperJS
595 GreaseMonkey/TamperMonkey, ChickenFoot, Scrapbook, iMacros, DejaClick # FF extensions
596 Selenium, Scrapy, RoboBrowser, FlexGet, ghost.py, splinter, binux/pyspider # python crawlin
    g libs
597 kimono, import.io
598 parklemotion/nokogiri # Ruby gem
599
600 python -m webbrowser -t "$url"
601 urlwatch --urls=urls-list.txt | ifne mutt -s "Page change detected" $email_address
602
603
604 _'_"_'_"_'_"_'_"_'_"_'_"_'_"_
605 #    ssh@  SSL  :ssh    #
606 -"-'-"-'-"-'-"-'-"-'-"-'-"-'-
607 keithw/mosh # faster 'ssh' replacement that allows the client and server to "roam" and chan
    ge IP addresses, while keeping the connection alive
608 liftoff/GateOne # HTML5-powered terminal emulator and SSH client - Also: http://en.wikipedi
    a.org/wiki/Comparison_of_SSH_clients#Platform
609 cat ~/.ssh/id_rsa.pub | ssh $user@$host "mkdir -p ~/.ssh && cat >>  ~/.ssh/authorized_keys"
     # Alt: ssh-copy-id $user@$host
```

```
610 ssh $host "$(printf "%q" $(cat script.sh))" # %q adds escapes to any string
611 ssh $host "$cmds ; /bin/bash -i" # Keep ssh session open after executing commands
612 ssh -f $host -L 2034:$host:34 -N # port forwarding
613 [ENTER] ~. # Exit a hung SSH session
614 # Force a user (based on its pub key) to only run one command one a host (e.g. tail -f) usi
    ng ~/.ssh/authorized_keys : cf. tmux example
615 # How to change your login on a specified acces: http://orgmode.org/worg/worg-git-ssh-key.p
    hp
616 /etc/ssh/sshd_config # SSH daemon config to allow UNIX user/pswd auth: PasswordAuthenticati
    on yes, UsePAM yes OR AllowGroups sshusers
617 /etc/pam.d/* # use pam_unix.so
618 knockd # port knocking server
619 cat $file.key $file.crt > $file.pem
620 openssl s_client -CApath $ca -cert $pem -key $key -connect $host:443 -ssl3 # bare SSL clien
    t
621 openssl x509 -text -noout -in $cert.pem # get certs details
622 openssl x509 -inform der -in $cert.cer -out $cert.pem # convert .cer to .pem
623 keytool -printcert -file $cert.pem # get certs details
624 sshfs # && fusermount -u
625 Russell91/sshrc # bring your .*rc with you
626 mussh \ # MUltihost SSH Wrapper - Also: fabfile.org
627  -l $USER \
628  -m 2 \ # run on two hosts concurrently
629  -h rpi-1 rpi-2 \ # hostnames
630  -c "$cmd"
631
632
633 </-/--------------\-\>
634 <!<! Apapapapache !>!>
635 <\-\--------------/-/>
636 source /etc/apache2/envvars && apache2 -V # -l -L -M
637 sudo bash -c 'source /etc/apache2/envvars && apache2 -t && apache2ctl -S' # check config
638 vim /etc/apache2/sites-available/default-ssl
639 service apache2 restart
640 tail -F /var/log/apache2/*.log
641 a2enmod / a2dismod $modname  # enable / disable std modules
642 ab -n5000 -c50 "http://path/to/app?params" # Apache benchmarking - Alt: tarekziade/boom
643
644
645 =cCcCcCc=
646 # Cisco #
647 =cCcCcCc=
648 # Remote-cmd & monitor device config: RANCID/clogin
649
650 enable # unlock more comnmands
651 show version
652 exit
653
654 show logging [buffered]
655
656 sh run
657 sh int
658 sh ip int [brief]
659 sh ip rou 1.2.3.4
660
661 # for Fastpath, e.g. QuantaLB:
662 show logging hosts
663 show logging buffered
664 traceroute $ip
665
666
667 -%-%-%-%-%-
668  =SYSTEM=
669 -%-%-%-%-%-
670
671 powertop # diagnose issues with power consumption
672 sysctl
673
674 cat /etc/*-release
675 lsb_release -a
676 uname -a
677 cat /etc/issue*
678
679 /proc/version
680 /proc/cpuinfo # Number of cores, cache size & alignement...
681 watch -d 'cat /proc/meminfo' # Watch system stats
682 /proc/sys/fs/file-nr # allocated/free file descriptors
683 /proc/loadavg : # graph in TTY: tload - Alt: uptime
684 - 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
685 - 4th field : number of currently executing kernel scheduling entities (processes, threads)
     / number of existing kernel scheduling entities
```

```
686  - 5th field : PID of last process created
687
688  echo 1 > /sys/module/printk/parameters/printk_time # Enable dmesg timestamps
689  dmesg -s 500000 | grep -i -C 1 "fail\|error\|fatal\|warn\|oom" # In case of OOM, Linux kern
     el 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
690
691  watch -d -n 1 "cat /proc/$pid/status | grep ctxt_switches" # mostly nonvoluntary context sw
     itches => CPU bound / else IO bound - FROM: https://blogs.oracle.com/ksplice/entry/solving_
     problems_with_proc
692
693  # Monitoring
694  iostat # ! '%util' & 'svctm' are misleading + iotop, non portable + brendangregg/perf-tools
     /blob/master/iosnoop
695  mpstat 5 # cpu usage stats every 5sec
696  monit # monitor processes, network stats, files & filesystem. Has an HTTP(s) interface, cus
     tom alerts
697  dstat
698  pt-summary, glances, psdash, conky
699  collectd, perfwatcher
700  hdparm -tT /dev/sda # Check a disk read/write speed
701
702  stap # SystemTap
703  perf # aka perf_events, needs a version of linux-tools-* matching the kernel - More: http:/
     /www.pixelbeat.org/programming/profiling/
704      top -G
705      stat -e cycles,instructions,cache-misses,dTLB-load-misses -p $PID
706  tobert/pcstat # page cache stats for files
707
708  # Checking Swap Space Size and Usage
709  free -m # how much free ram I really have ? -> look at the row that says "-/+ buffers/cache
     "
710  vmstat 2
711  sar # provides history data
712
713  w / who # users currently logged
714  last [-f /var/log/wtmp.1] # previous logged users
715  dump-utmp /var/run/utmp # or /var/log/wtmp
716  lastcomm # or dump-acct pacct : list last executed commands. From acct pkg, must be turned
     on with /etc/init.d/psacct start
717  # Alt (very resource consuming): auditctl -a task,always; ausearch -i -sc execve
718
719  /etc/motd # Message of the day, can be combined from multiple files: man update-motd
720
721  # Get uid / groups infos
722  id $USER # for primary group, use -ng flag
723  adduser / usermod -a -G # DO NOT FORGET THE -a !!!
724  useradd -m -G sudo,sshusers -p $(openssl passwd ******)
725
726  # Add a Linux secondary group without logging out
727  newgroup $new_secondary_group
728  newgroup $original_primary_group
729
730  awk -F":" '{ print "username: " $1 "\t\tuid:" $3 }' /etc/passwd # List system users
731  /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
732
733  sudo su -l # login as user root
734  sudo -K # Remove sudo time stamp => no more sudo rights
735  fakeroot # runs a command in an environment wherein it appears to have root privileges for
     file manipulation
736  chroot $path_to_fake_root $cmd # 'chroot jail' => changes the apparent root directory
737
738  faketime $time_spec $cmd
739
740  fdisk -l # has flaws, better use bitbucket.org/skypher/fdisk
741  testdisk # disk data recovery
742  lsusb # Alt: usb-devices
743  lspci -v # list devices
744  lshw -C disk # list disks : ata, cdrom, dvdrom
745  blkid # list UUIDs
746  dmidecode
747  /sbin/mdadm --examine --scan --verbose # need root - RAID config
748  shutdown -r -F now # force FCSK disk check - Or: touch /forcefsck - Alt:
749  smartctl -a /dev/sdb2 # scan a device - Alt: gsmartcontrol or above
750
751  # Find what package a command belong to:
752  apt-file search /path/to/anyfile
753  yum provides $cmd
754  dpkg -S /path/to/cmd
755  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

```

```
833 /var/log/kern.log EMPTY # needs $ModLoad imklog in /etc/rsyslog.conf + service rsyslog rest
    art (thx: http://serverfault.com/a/405244 ): BUT:
834 # -> "imklog: error reading kernel log - shutting down: Bad file descriptor" + CPU maxing o
    ut. Web search => looks like a known issue solved with more recent versions of rsyslog
835
836 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
837
838
839 FFFFFFFFFFFFFFF
840 F i r e f o x
841 FFFFFFFFFFFFFFF
842 ~/.mozilla/firefox/*.default/mimeTypes.rdf # FIREFOX 'open with' mapping
843 find Cache/ -type f -exec file {} \; | grep image | cut -d':' -f1 # all cached images
844 about:cache # Firefox cache infos: location, size, number of entries
845 about:memory # Firefox memory allocation details
846 about:about # all the about: pages e.g. :crashes :healthreport :permissions :plugins :sessi
    onrestore
847 $ff_profile_dir/.parentlock # fix "Firefox is already running but is not responding" error
848 cp sessionstore.bak sessionstore.js # Restore previous session tabs
849 <CTRL>+F5 # refresh page bypassing the cache
850 MAJ+F2: screenshot --fullpage $filename # PNG screenshot of the webpage - Alt: http://freze
    .it
851
852 https://developer.mozilla.org/en-US/docs/Tools/Web_Console
853 - inspect(), pprint()
854 - console.time(name) .timeEnd(name) .profile(name) .profileEnd(name)
855 - cd("#frame1"); # get into a specific iframe
856 - $("css selector") or $$() for ALL matches; $x("xpath expression")
857 //div[contains(concat(' ',normalize-space(@class),' '),' foo ')] # http://pivotallabs.com/x
    path-css-class-matching/
858
859 =\/=/\=\/=/\=\/=
860 =  Virtualbox
861 =\/=/\=\/=/\=\/=
862 sudo adduser $USER vboxusers # then logout
863 VBoxManage list vms
864 VBoxManage controlvm $name poweroff
865 VBoxClient --clipboard
866 $HOME/VirtualBox VMs/{machinename}/Logs
867
868 # Cool features : remote display (VRDS), shared folders & clipboard, seamless mode
869
870
871 () () () () () () () () () ()
872 () Synthèse vocale
873 () () () () () () () () () ()
874 espeak -s 180 -p 40 "Hey ! Look behind you"
875 espeak -s 180 -p 40 -ven+12 "Hi ! My name is Colossus."
876 espeak -s 150 -p 20 -vfr "Je vais te péter la gueule"
877 espeak -v mb/mb-fr1 -s 50 'Je peux parler plus lentement' | mbrola /usr/share/mbrola/voices
    /fr1 - -.au | aplay
878 #FROM:   http://doc.ubuntu-fr.org/synthese_vocale
879 #        http://linux.byexamples.com/archives/303/text-to-speech-synthesizer/
880 #        http://cookerspot.tuxfamily.org/wikka.php?wakka=SyntheseVocaleEspeak
881
882
883 ::=::=::=::
884 : MAC OSX :
885 ::=::=::=::
886
887 curl http://google.com/ | base64 | say # FUN
888
889 dns-sd -Q $USER.local # mDNS query
890
891 sudo softwareupdate -i -a # Manual software update
892
893 Finder > Applications > Utilities > Disk Utility # Repair permissions
894
895 system_profiler # list system components, ports...
896 pmset -g # power management settings
897
898 pbpaste | pbcopy # clipboard
899
900 textutil -convert txt # or -info : convert / get infos on files
901
902 xattr -l $file # File listed with '@' => extended attributes
903
904 sudo dseditgroup -o edit -a $USER -t user $GROUP # Add user to group
905
906 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") $file; done | less # illustrate how to repl
    ace find -printf + timestamp conversion + find non-hidden files only ; GOAL: list files by
    modification date
907
908 # DTrace scripts: man -k dtrace
909 iosnoop # or better hfsslower.d from the DTrace book, available online
910 execsnoop # trace processes created
911 opensnoop -ve # trace open files, also maclife.d from DTrace book to trace files creation/d
    eletion
912 dtruss -d # strace
913 soconnect_mac.d # trace TCP connections, from DTrace book
914 errinfo # trace system call fail
915 bitesize.d # trace I/O
916 iotop
917
918 # C#
919 NUNITLIB=/Library/Frameworks/Mono.framework/Versions/2.10.11/lib/mono/2.0/nunit.framework.d
    ll
920 gmcs -debug -t:library -r:$NUNITLIB *.cs
921 nunit-console *.dll
922 mono *.exe
923
924 # AppleScript
925 #!/usr/bin/osascript
926 on log(msg)
927   set log_line to (do shell script "date  +'%Y-%m-%d %H:%M:%S'" as string) & " " & msg
928   do shell script "echo " & quoted form of log_line
929 end log
930 log "HELLO WORLD !"
931
932
933 ]_]_]_]_]_]_]_]
934 ] ImageMagick
935 ]_]_]_]_]_]_]_]
936 # Compile IM with HDRI:
937 # - http://www.imagemagick.org/script/install-source.php
938 # - sudo aptitude install libmagickcore-dev liblcms2-dev libtiff4-dev libfreetype6-dev libj
    peg8-dev liblqr-1-0-dev libglib2.0-dev libfontconfig-dev libxext-dev libz-dev libbz2-dev
939 # - ./configure --enable-hdri
940 # - identify -version # to check HDRI is enabled
941 # Scripts: http://www.fmwconcepts.com/imagemagick/
942 display $img_file
943 convert img.png -adaptive-resize 800x600 -auto-orient -crop 50x100+10+20 img.jpg
944 mogrify ... *.jpg # for f in *.jpg; do convert $f ... ; done
945 identify -v $img_file # get PPI: -format "%w x %h %x x %y"
946 import -display :0.0 -window root screenshot.png # Alt: gnome-screenshot --interactive # Or
     Gimp
947 animate -delay 5 *.png
948 compare img1 img2
949 composite # merge images
950
951 gifsicle "$gif" -I | sed -ne 's/.* \([0-9]\+\) images/\1/p' # frames count + cf. stopmo_log
    o/gen_anim.sh
952 convert $(for f in *.png; do echo -delay 5 $f; done; ) -rotate -90 -resize 50% -loop 0 out.
    gif
953 tesseract-ocr # Google OCR / text extraction - http://askubuntu.com/a/280713/185582
954 qrencode -o $png $url && zbarimg -q $png # from zbar-tools - Can generate ASCII ! - Alt: Py
    thon qrcode
955 barcode -b "Hello World !" -o out.ps && convert out.ps -trim out.png
956 pngquant ## 70% lossy compression
957 jpegtran -optimize -progressive -grayscale -outfile $out_file $in_file # FROM: libjpeg-turb
    o-progs
958 identify -verbose $jpg | grep -Fq 'Interlace: JPEG' # is JPEG progressive ? Alt: grep -Fq "
    $(echo -en "\xff\xc2")" $jpg
959 mat # Metadata Anonymisation Toolkit, removes e.g. images hermful metadata
960 feh -F -d -D 3 --cycle-once * # fast image viewer: fullscreen slideshow with 3s delay - Alt
    : gpicviw
961
962
963 $$$$$$$$$$$$$$
964 $ Google APIs $
965 $$$$$$$$$$$$$$
966 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 !
967
968 # Search tips&tricks : https://support.google.com/websearch/answer/136861
969 site:$base_url "exact match" OR "a * saved is a * earned" -term # basics
970 inurl:gouv.fr # Also: intitle:
971 filetype:pdf
972 cache:$url
973 define:$term
974 related:$url
```

```
975  link:$url # Search for pages that link to a URL
976  https://www.google.fr/search?q=5%2B(-sqrt(1-x^2-(y-abs(x))^2))*sin(100*((10-x^2-(y-abs(x))^
     2))),+x+is+from+-1+to+1,+y+is+from+-1+to+1.5,+z+is+from+1+to+6 # 3D heart surface
977
978  youtube-dl --ignore-errors --extract-audio FLF8xTv55ZmwikWWmWLPEAZQ # download playlist as
     .m4a files - in case of HTTP error 500, try -f18
979
980  # Snippet-search
981  cse_id=003799500572498885021:6zbuscnifvi
982  curl -s "https://www.googleapis.com/customsearch/v1?key=${api_key}&cx=${cse_id}&fields=item
     s(snippet)&q=define%20${term}"
983  # DOCS: https://developers.google.com/custom-search/json-api/v1/using_rest https://develope
     rs.google.com/custom-search/json-api/v1/performance#partial
984
985
986  {[{[{[{[{[{[{[{
987   'AWS': "cli"
988  }]}]}]}]}]}]}]}
989  aws configure # eu-west-1
990  aws iam list-user-policies --user-name lucas # Also: aws iam list-roles
991  aws s3 cp $file s3://lucas-pail/ # Other cmds: mb rb ls rm mv
992  # AWS Lambda - mostly from http://alestic.com/mt/mt-search.cgi?blog_id=1&tag=AWS%20Lambda
993  aws lambda list-functions
994  aws lambda invoke-async --function-name $function --region us-east-1 --invoke-args inputfil
     e.json --debug
995  aws logs describe-log-groups --region us-east-1
996  log_group_name=/aws/lambda/$function
997  log_stream_names=$(aws logs describe-log-streams --region us-east-1 --log-group-name "$log_
     group_name" --output text --query 'logStreams[*].logStreamName')
998  for stream in $log_stream_names; do
999      aws logs get-log-events --region us-east-1 --log-group-name "$log_group_name" --log-str
     eam-name "$stream" --output text --query 'events[*].message'
1000 done | less
1001
1002
1003 /./../
1004  /irc
1005 /./../
1006 http://fr.wikipedia.org/wiki/Aide:IRC/commandes
1007 http://www.ircbeginner.com/ircinfo/ircc-commands.html
1008
1009 # Weechat - Alt: irssi bitlbee or mcabber for Jabber only
1010 weechat --run-command '/set;/quit' > ~/dump-weechat-config
1011 tF /home/lucas/.weechat/weechat.log /home/lucas/.weechat/logs/*
1012 http://weechat.org/files/temp/scripts/hdata.py # install with '/python load hdata.py' - Als
     o: hdata_update.py
1013 /script search iset # then 'i' to install
1014
1015 /server list[full] [server]
1016 /connect freenode
1017 /nick dr_max_kurt
1018 /join #laquadrature
1019 /set irc.server.freenode.autoconnect on
1020 /set irc.server.freenode.nicks "dr_max_kurt"
1021 /set irc.server.freenode.sasl_ ...
1022 <ALT>+<ARROW> # switch window
1023
1024
1025 ~~~~~~~~~~~~~
1026 -> Freeplane
1027 ~~~~~~~~~~~~~
1028 # SHORTCUTS, cf. Help > Key Reference - To add new ones: <CTRL>+<click> a menu item
1029 <Enter> / <Insert> : new sibling / child node
1030 <CTRL>+<double click> : new free node
1031
1032 <CTRL>+L : link selected nodes
1033 <CTRL>+<SHIFT>+K : add hyperlink, <CTRL>+K to modify
1034
1035 <Space> : Toggle folding
1036 <ALT>+<left> : 'back', go to previously visited node
1037
1038 # Customized shortcuts
1039 <CTRL>+F? : apply style "Level ?"
1040 <CTRL>+1 : add image (from 'Node extensions' menu, so that they cab resized)
1041 <CTRL>+<SHIFT>+1 : add icon
1042 <CTRL>+3 : node color
1043 <CTRL>+<SHIFT>+3 : node background color
```