UNIX/LINUX Top100 Commands

Version 1.13 — 28.3.2008

© 2003–2008 T. Birnthaler, OSTC GmbH

17. Dezember 2014

Inhaltsverzeichnis

1 The most important UNIX/LINUX commands

`man CMD`	Display description for `CMD` [**manual**]
`whatis CMD`	Display one line of description for `CMD` (man-*index!*)
`apropos TEXT`	Display one line of description for topic `TEXT` (man-*index!*)
`info CMD`	Display description for `CMD` (*GNU programs!*)
`help CMD`	Display description for builtin shell `CMD` (`bash`)
`CMD --help`	Display usage message for `CMD` (*GNU programs!*)
`CMD -h  -?  -.`	Display usage message for `CMD` (*all other programs!*) [**help**]
`type [-a] CMD`	Search for `CMD` type/place (builtin/alias/function/binary, `bash`, all)
`which/whence CMD`	Search for `CMD` type/place ((`t`)`csh/ksh`)
`who`	Display all users logged in
`whoami  who am i`	Display currently (in terminal) logged in user
`id [-a]`	Display user name + all his group names + IDs [**identity, all**]
`su [-] USER`	Switch to account `USER` (-=*incl. environment!*) [**switch user**]
`su    su -`	Become superuser (*without/with environment change!*)
`exit   logout`	Switch back to former account resp. log out
`passwd`	Change password (1x old, 2x new one) [**password**]
`more/less FILE...`	Display `FILE` page by page (also `pg`, `page`)
`CMD	more/less`	Display `CMD` output page by page
`lp(r) FILE...`	Print out `FILE` (`lpr`=BSD, `lp`=SysV) [**line print**]
`CMD	lp(r)`	Print out `CMD` output
`pwd`	Display current directory [**print working directory**]
`cd PATH`	Switch to directory `PATH` [**change directory**]
`cd`	Switch to own home directory (e.g. `/home/tom`)
`cd /   cd ..`	Switch to root resp. parent directory
`ls [DIR]`	Display file names of current directory sorted alphabetically
`ls -l [DIR]`	Display file names + attributes ... [**long**]
`ls -a [DIR]`	Display all file names (*hidden ones too!*) [**all**]
`ls -d [DIR]`	Display only directory name, not content [**directory**]
`ls -F [DIR]`	Append file type postfix to name (*=exec, /=dir, @=link) [**Flag**]
`ls -R [DIR]`	Display whole directory tree from `DIR` on [**Recursive**]
`ls -t [DIR]`	Display sorted by modification time [**time**]
`touch FILE...`	Update change date of `FILE` or create new file
`*`	Matches all files in current directory
`mv FILE NAME`	Rename `FILE` to `NAME` [**move**]
`mv FILE... DIR`	Move `FILE`... to directory `DIR` [**move**]
`mv FILE DIR/NAME`	Move `FILE` to directory `DIR` and rename it [**move**]
`cp FILE NAME`	Copy `FILE` to `NAME` (*overwrite!*) [**copy**]
`cp FILE... DIR`	Copy `FILE`... to directory `DIR` [**copy**]
`ln FILE NAME`	Create hard link `NAME` to `FILE` [**link**]
`ln -s FILE NAME`	Create symbolic link `NAME` to `FILE` [**symbolic**]
`rm FILE...`	Delete `FILE` [**remove**]
`rm -r DIR...`	Delete complete directory tree starting at `DIR` [**recursive**]
`mkdir DIR...`	Create directory `DIR` [**make directory**]
`rmdir DIR...`	Delete directory `DIR` (*has to be empty!*) [**remove directory**]
`cat FILE1 FILE2...`	Concatenate files and output them to `stdout` [**concatenate**]
`chmod MODE FILE...`	Change file perm. `MODE=[ugoa][+-=][rwxst]` [**change mode**]
`umask [NNN]`	Display/set standard permission mask `NNN` for new files/dirs
	(*defines permissions to be removed!*) [**usage mask**]
`umask`	Normally displays `022` (i.e. remove write perm. for `g+o`)

2   Other important UNIX/LINUX commands

`<STRG-C>`	Cancel currently running command (`SIGINTR` = 2)
`ps`	Display own processes [**process status**]
`ps aux` `ps -ef`	Display all p. (BSD/SysV) [**all/user/extended/every/full**]
`kill PID`	Stop process `PID` (standard signal `SIGTERM` = 15)
`kill -9 PID`	Stop process `PID` unconditionally (`SIGKILL` = 9)
`CMD &`	Run `CMD` in background
`nohup CMD &`	Analogous, no stop on logout [**no hangup**]
`<STRG-Z> bg`	Switch current command to background [**background**]
`jobs`	Display all background processes of shell
`CMD < FILE`	Read input for `CMD` from `FILE`
`CMD > FILE`	Write output of `CMD` to `FILE`
`CMD >> FILE`	Append output of `CMD` to `FILE`
`CMD 2> FILE`	Write error messages of `CMD` to `FILE` (*not* `(t)csh`*!*)
`CMD 2>> FILE`	Append error messages of `CMD` to `FILE` (*not* `(t)csh`*!*)
`CMD >& FILE`	Write output + error messages of `CMD` to `FILE` (`(t)csh`*!*)
`CMD1	CMD2`	Connect output of `CMD1` with input of `CMD2` [**pipe**]
`VAR="TEXT"`	Set shell variable `VAR` to value `TEXT` (*no space around* =*!*)
`set VAR "TEXT"`	Set shell variable for `(t)csh` (*no* = *but space!*)
`export VAR`	Convert shell variable `VAR` to environment variable
`setenv VAR="TEXT"`	Set environment variable for `(t)csh` (=*!*)
`echo $VAR`	Display content of variable `VAR`
`HOME`	Standard directory for `cd` (home directory)
`LANG`	Language settings (e.g. `de_DE.UTF-8`, `fr_FR`, `es_ES`) [**language**]
`PATH`	Search path for binary programs (dirs separated by `:`)
`PS1` `prompt`	Define shell prompt (`sh`/`(t)csh`) [**prompt string**]
`TERM`	Terminal type (editors, `more`, `curses` library)
`USER/LOGNAME`	Name of logged in user
`grep "TEXT" FILE...`	Display all lines of `FILE` that contain `TEXT`
`CMD	grep "TEXT"`	Display all lines of `CMD` output that contain `TEXT`
`wc -l FILE...`	Display number of lines in `FILE` [**word count/lines**]
`CMD	wc -l`	Display number of lines of `CMD` output
`head FILE...`	Display first 10 lines of `FILE`
`CMD	head -50`	Display first 50 lines of `CMD` output
`tail FILE...`	Display last 10 lines of `FILE`
`CMD	tail -50`	Display last 50 lines of `CMD` output
`tail -f FILE`	Show new lines at end of `FILE` continuously [**follow**]
`echo "TEXT"`	Display `TEXT` on terminal
`clear`	Erase terminal content (blank screen)
`date [+FORMAT]`	Display date + time (according to `FORMAT`, e.g. `%T`, `%D`)
`more/less FILE...`	Page throug `FILE` with `more`/`less` [**visual**]
`<SPACE> <RETURN>`	Skip to next page/one line further
`b`	Skip one page back (*only* `less`*!*) [**backward**]
`/TEXT <RETURN>`	Search `TEXT`
`n  N`	Repeat last search forward/backward (*only* `less`*!*)
`q`	Leave `more`/`less` [**quit**]
`h`	Display help [**help**]
`vi FILE...`	Edit `FILE` with Vi [**visual**]
`i  a  I  A  o  O`	Switch into edit mode [**insert/append/open**]
`ESC`	Leave edit mode (back to command mode)
`:wq <RETURN>`	Exit Vi with saving edited file [**write/quit**]
`ESC :q! <RETURN>`	Exit Vi anyway (*without saving edited file!*) [**quit**]