Linux Command Summary

Commands

Command	Meaning	Options
`exit`	log out	
`passwd`	change your password	
`clear`	clear screen	
`man` *command*	shows the manual page for *command*	`man -k` *word* shows a list of man pages that mention *word*
`history`	shows all previously-issued commands	
`!!`	executes most recently-issued command	
`!c`	executes most recently-issued command starting with `c`	
`whoami`	displays your login name	
`date`	displays current date and time	
`pwd`	displays current directory	
`ls`	lists contents of current directory	`ls -a` shows all files, including hidden files `ls -l` shows in long format
`cp` *file1 file2*	copies *file1* to *file2*	`cp -r` *dir1 dir2* recursively copies *dir1* to *dir2*
`mv` *file1 file2*	moves *file1* to *file2* (also used to rename)	
`rm` *file*	removes *file*	can be used to recursively remove a directory, if `-r` option is used
`touch` *file*	updates *file*'s last modified time to current time	can be used to create an empty file if *file* does not exist
`cd` *dir*	changes directory to *dir*	`cd -` returns to most recently visited directory
`mkdir` *dir*	creates new directory *dir* in current directory	can specify more than one directory at once
`rmdir` *dir*	removes directory *dir*	only works if *dir* is empty; if not empty, use `rm -r` *dir*; can specify more than directory at once
`echo` *string*	displays *string* to screen	
`chmod` *perms file*	sets permissions on *file* to *perms*	
`chfn`	changes personal info (name, address, etc.) on Linux system	
`ps`	displays current processes	`ps -a` show all users' processes `ps -A` show ALL processes (incl. system processes)
`kill` *pid*	kills process with number *pid*	`kill -9` *pid* more forceful kill, for stubborn processes
`who`	show who is logged into this machine	
`finger` *username*	shows personal info for *username*	
`time` *command*	shows amount of time taken executing *command*	
`fg`	brings background job to the foreground	useful if you accidentally ran vi or emacs with an `&`
`find` *dir* -name "*pattern*"	find all files whose names match *pattern* in *dir* and its subdirectories	

Tools

Tool	Purpose	Options
`cat` *f1 f2 ...*	displays files *f1*, *f2*, ... one after the other	`cat -n` *f1 f2 ...* attaches line numbers
`more` *file*	displays *file* one screen at a time	
`diff` *f1 f2*	compares files *f1* and *f2*; outputs instructions for converting *f1* to *f2*	`diff -w` *f1 f2* ignores whitespace
`cmp` *f1 f2*	compares files *f1* and *f2*; outputs the first position where they differ	
`wc` *file*	counts the number of words, lines, and characters in *file*	`wc -c` *file* shows just the number of characters `wc -l` *file* shows just the number of lines `wc -w` *file* shows just the number of words
`egrep` *pat file*	prints all lines in *file* that contain pattern *pat*	`egrep -n` *pat file* print matching lines with line numbers `egrep -v` *pat file* prints lines that do *not* match *pat*
`head` *file*	prints first 10 lines of *file*	*-num* prints *num* lines (e.g. `head -5` *file*)
`tail` *file*	like head, but prints last 10 lines of *file*	
`sort` *file*	sorts the lines of *file*	`sort -n` *file* sorts strings of digits in numerical order
`uniq` *file*	removes consecutive duplicate lines from *file*	removes all duplicates if *file* is sorted

Programs

Program	Purpose	Options
`vi` *file*	invokes vi text editor on *file*	
`emacs` *file*	invokes emacs text editor on *file*	
`nano` *file*	invokes nano text editor on *file*	
`pine` (or `alpine`)	read email	
`wget` *url*	fetches file from the web at *url*	
`xpdf` *file*	displays pdf file	
`lpr` *file*	prints *file* to printer	`lpr -Pljp_3016` *file* specifies the printers in MC3016
`lpq`	checks the print queue	
`lprm` *jobno*	removes job *jobno* (must belong to you!) from print queue	
`ssh` *machine*	makes SSH connection to *machine*; opens a secure shell on remote *machine*; type `exit` to end SSH connection	`ssh -Y` (or `-X`) *machine* enables X-forwarding (must have X server running on local machine)
`scp` *mach1*:*file1 mach2*:*file2*	securely copy *file1* on *mach1* to *file2* on *mach2*	can omit *mach1* if it is the local machine; similarly for *mach2*

Variables

Variable	Meaning
`${PWD}`	present working directory (equivalent to executing `pwd`)
`${HOME}`	your home directory (equivalent to ∼)
`${SHELL}`	your default shell
`${PRINTER}`	your default printer
`${PATH}`	your default search path for commands
`${$}`	current script's process ID
`${0}`	name of currently-running script
`${1}`, `${2}`, ...	arguments 1, 2, ... of current script/function
`${#}`	number of args supplied to currently-running script/function (not including script name)
`${@}`	all args supplied to currently-running script/function as separate strings (not including script name)
`${?}`	return code of most recently-executed command/function

Permissions

Symbol	Meaning
`u`	file's owner
`g`	members of the file's group, other than the owner
`o`	other users
`a`	all users (equivalent to ugo)
`+`	add permission bit
`-`	revoke permission bit
`=`	set permission bits exactly
`r`	read permission. for files—file's contents can be read. for directories—directory's contents can be listed
`w`	write permission. for files—file's contents can be modified. for directories—files can be added/renamed/removed in the directory
`x`	execute permission. for files—file may be executed as a program or script. for directories—directory can be traversed (i.e. can cd into the directory)

Script Conditional Operators

Operator	Meaning
`=`	string equality
`!=`	string inequality
`-eq`	integer equality
`-ne`	integer inequality
`-gt`	integer greater than
`-ge`	integer greater than or equal to
`-lt`	integer less than
`-le`	integer less than or equal to
`-a`	and
`-o`	or
`!`	not
`( )`	parentheses for grouping
`-d`	file exists and is a directory
`-e`	file exists
`-f`	file exists and is a regular file
`-r`	file exists and is readable
`-w`	file exists and is writable
`-x`	file exists and is executable

Globbing Patterns

Operator	Meaning
`*`	matches 0 or more characters
`?`	matches one character
`[abxy]`	matches exactly one of the characters in brackets
`[!abxy]`	matches any character except the ones in the brackets
`[a-z]`	matches any character in the given range
{*pat1*,*pat2*}	matches either *pat1* or *pat2* (technically not a glob; note no spaces)

Directories

Directory	**Meaning**
.	Current directory
. .	Parent of current directory
~	Your home directory
/	Root directory
Starts with / or ~	Absolute path
Does not start with / or ~	Relative path