

# Linux Command Summary

## Commands

Command	Meaning	Options
exit	log out	
passwd	change your password	
clear	clear screen	
man <i>command</i>	shows the manual page for <i>command</i>	man -k <i>word</i> shows a list of man pages that mention <i>word</i>
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 <i>file1 file2</i>	copies <i>file1</i> to <i>file2</i>	cp -r <i>dir1 dir2</i> recursively copies <i>dir1</i> to <i>dir2</i>
mv <i>file1 file2</i>	moves <i>file1</i> to <i>file2</i> (also used to rename)	
rm <i>file</i>	removes <i>file</i>	can be used to recursively remove a directory, if -r option is used
touch <i>file</i>	updates <i>file</i> 's last modified time to current time	can be used to create an empty file if <i>file</i> does not exist
cd <i>dir</i>	changes directory to <i>dir</i>	cd - returns to most recently visited directory
mkdir <i>dir</i>	creates new directory <i>dir</i> in current directory	can specify more than one directory at once
rmdir <i>dir</i>	removes directory <i>dir</i>	only works if <i>dir</i> is empty; if not empty, use rm -r <i>dir</i> ; can specify more than one directory at once
echo <i>string</i>	displays <i>string</i> to screen	
chmod <i>perms file</i>	sets permissions on <i>file</i> to <i>perms</i>	
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 <i>pid</i>	kills process with number <i>pid</i>	kill -9 <i>pid</i> more forceful kill, for stubborn processes
who	show who is logged into this machine	
finger <i>username</i>	shows personal info for <i>username</i>	
time <i>command</i>	shows amount of time taken executing <i>command</i>	
fg	brings background job to the foreground	useful if you accidentally ran vi or emacs with an &
find <i>dir</i> -name " <i>pattern</i> "	find all files whose names match <i>pattern</i> in <i>dir</i> and its subdirectories	

## Tools

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

## Programs

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

## Variables

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

## Globbering 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 <i>pat1</i> or <i>pat2</i> (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