

# CS246—Linux Command Summary

Fall 2016

## Commands

Command	Meaning	Options
exit	log out	
passwd	change your password	
clear	clear screen	
man <i>command</i>	show the manual page for <i>command</i>	man -k <i>word</i> show a list of man pages that mention <i>word</i>
history	show all previously-issued commands	
!!	execute most recently-issued command	
!c	execute most recently-issued command starting with c	
whoami	display your login name	
date	display current date and time	
pwd	display current directory	
ls	list contents of current directory	ls -a show all files, including hidden files ls -l show in long format
cp <i>file1 file2</i>	copy <i>file1</i> to <i>file2</i>	cp -r <i>dir1 dir2</i> recursively copy <i>dir1</i> to <i>dir2</i>
mv <i>file1 file2</i>	move <i>file1</i> to <i>file2</i> (also use to rename)	
rm <i>file</i>	remove <i>file</i>	can be used to recursively remove a directory, if -r option is used
touch <i>file</i>	update <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>	change directory to <i>dir</i>	cd - return to most recently visited directory
mkdir <i>dir</i>	create new directory <i>dir</i> in current directory	can specify more than one directory at once
rmdir <i>dir</i>	remove 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 directory at once
echo <i>string</i>	display <i>string</i> to screen	
chmod <i>perms file</i>	set permissions on <i>file</i> to <i>perms</i>	
chfn	change personal info (name, address, etc.) on Unix system	
chsh	change your login shell	
ps	display current processes	ps -a show all users' processes ps -A show ALL processes (incl. system processes)
kill <i>pid</i>	kill 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>	show personal info for <i>username</i>	
time <i>command</i>	show amount of time taken executing <i>command</i>	
fg	bring 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
cat <i>f1 f2</i> ...	display files <i>f1</i> , <i>f2</i> , ... one after the other	cat -n <i>f1 f2</i> ... attaches line numbers
more <i>file</i>	display <i>file</i> one screen at a time	
diff <i>f1 f2</i>	compare files <i>f1</i> and <i>f2</i> ; outputs instructions for converting <i>f1</i> to <i>f2</i>	diff -w <i>f1 f2</i> ignores whitespace
cmp <i>f1 f2</i>	compare files <i>f1</i> and <i>f2</i> ; outputs the first position where they differ	
wc <i>file</i>	count the number of words, lines, and characters in <i>file</i>	wc -c <i>file</i> show just the number of characters wc -l <i>file</i> show just the number of lines wc -w <i>file</i> show just the number of words
egrep <i>pat file</i>	print all lines in <i>file</i> that contain pattern <i>pat</i>	egrep -n <i>pat file</i> print matching lines with line numbers egrep -v <i>pat file</i> print lines that do <i>not</i> match <i>pat</i>
head <i>file</i>	print first 10 lines of <i>file</i>	- <i>num</i> prints <i>num</i> lines (e.g. head -5 <i>file</i> )
tail <i>file</i>	like head, but prints last 10 lines of <i>file</i>	
sort <i>file</i>	sorts the lines of <i>file</i>	sort -n <i>file</i> sorts strings of digits in numerical order
uniq <i>file</i>	removes consecutive duplicate lines from <i>file</i>	removes all duplicates if <i>file</i> is sorted

## Programs

Program	Purpose	Options
vi <i>file</i>	invoke vi text editor on <i>file</i>	
emacs <i>file</i>	invoke emacs text editor on <i>file</i>	
pico <i>file</i>	invoke pico text editor on <i>file</i>	
pine (or alpine)	read email	
wget <i>url</i>	fetch file from the web at <i>url</i>	
xpdf <i>file</i>	display pdf file	
lpr <i>file</i>	print <i>file</i> to printer	lpr -Pljp_3016 <i>file</i> specifies the printers in MC3016
lpq	checks the print queue	
lprm <i>jobno</i>	removes job <i>jobno</i> (must belong to you!) from print queue	
ssh <i>machine</i>	make SSH connection to <i>machine</i> ; opens a secure shell on remote <i>machine</i> ; type exit to end SSH connection	ssh -Y (or -X) <i>machine</i> enable X forwarding (must have X server running on local machine)
scp <i>mach1:file1 mach2:file2</i>	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
\${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

## 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 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