UNIX COMMAND SUMMARY

SUNY Poly Computer Science Department

Spring 2015 **www.cs.sunyit.edu**

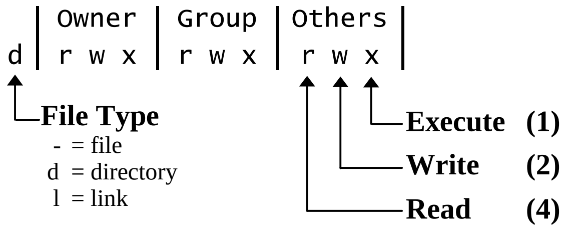
File Manipulation

ls	List files in current directory
ls -l	List files using long format
ls -al	List all files using long format
ls -al _string_*	Long list all files beginning with *string*
cp *filename* /tmp	Copy a file to another directory
mv *filename* /tmp	Move a file to another directory
mv *file1 file2*	Rename *file1* to *file2*
rm *filename*	Remove a file
vi *filename*	Open *filename* in the vi editor
vim *filename*	Open *filename* in the vi improved editor
touch *filename*	Create the empty file named *file*
sort *filename*	Sort the lines of *file*
uniq *filename*	Only show unique lines in sorted file *file*
wc *filename*	Word count
wc -l *filename*	Line count
diff *file1 file2*	Show the differences between 2 files
file *filename*	Describe the contents of a file
cat *filename*	Display the contents of a file
cat *f1 f2* > *f3*	Combine contents of *f1* and *f2*, save it in *f3*
less *filename*	Display the contents with paging
head *filename*	Display the first 10 lines
head -*x filename*	Display the first *x* lines
tail *filename*	Display the last 10 lines
tail -*x filename*	Display the last *x* lines
tail -f *filename*	Show the last 10 lines and wait for additional data to be appended to the file. ctrl-c to quit.

Directory Manipulation

ls -l | grep ^d	List all directories
cd *dirname*	Change to the directory *dirname*
cd ..	Change to the parent of current directory
cd -	Return to the last directory you were in
cd	Change to your home directory
pwd	Show the path of the present working directory.
mkdir *dirname*	Create the directory *dirname*
rmdir *dirname*	Remove the empty directory *dirname*
rm -rf *dirname*	Remove an entire directory and all of its contents, including files and subdirectories
du -h *dirname*	Show disk usage for *dirname* and all subdirectories in an easier to read format.

File & Directory Permissions



```
  | Owner | Group | Others |
d | r w x | r w x | r w x  |
```

File Type
- = file
d = directory
l = link

Execute (1)
Write (2)
Read (4)

Detailed file information is displayed if the command ls -l is given. The first column in this list details the permissions applied to the file. If a permission is missing, it is represented by a **-** **,** ie: drwxr-xr-x

Read (r) = 4
Write (w) = 2
Execute (x) = 1

File permissions are modified with the **chmod** command and the appropriate octal code for each of the 3 user types (owner, group, or others). To determine each of the 3 octal codes, add the digits that correspond to the desired permission level.

Owner:	4+2+1	= 7
Group:	4+2	= 6
Other:	4	= 4

chmod 764 *filename* will make the file R+W+X for the owner, R+W for the group, and R for everyone else.

chmod 755 *file*	Full permissions for the owner, read and execute for the group and everyone else.
chmod 644 *file*	Read/write permissions for the owner, read access for the group and everyone else.
chmod u+x *file*	Make the file executable for the owner.
chmod +x *file*	Make the file executable for everyone.
umask	Display the current file creation mask
umask *num*	Set default permissions for new files. *Num* is the permission bits which are turned off. For example, a **umask** of **027** would turn off write access for Group and turn off all for Others.

Input/Output Redirection

>	Output from a process is sent to another place instead of to the screen, overwriting the destination file if it already exists. Example: ls > filelist.txt
<	Input to a program is taken from another source instead of from the keyboard. Example: a.out < datafile
< >	Redirect both input and output at the same time. Example: program < input > output
>>	The output from a process is appended to a file that already exists. Example: cat paper >> finaldoc
|	Output from one command is sent as input to the next. Example: ls -al | less or a.out | lpr
<<*end*	Read input from keyboard until the text *end* is typed on a new line. Example: cat <<end
(ls *dir* ; cat *file*) | lpr	Combine output and pipe to lpr
set noclobber	Don't overwrite files with output redirection.

Command Line Wildcards

*	Match all characters
?	Match any single character
[abc]	Match a single character from the list abc, ex: ls A[def]*
[a-z]	Match a single character from the range a to z
[a-z]*	Match a string that begins with a letter in the range

Examples of valid ranges: a-z A-Z 0-9 d-q A-z

Searching for files and text

which *file*	Search your path to find the location of the executable named *file*
locate *file*	Search for a file by using a preexisting system database.
find /usr -name *file*	Search for a file named *file*, starting in the directory /usr.
find . -name "**file**"	Search for filenames containing the string *file*, starting in the current directory.
grep *pattern file*	Search within *file* for lines containing *pattern*
grep *pattern* *	Search files within the current directory and display all lines that contain *pattern*

Regular Expression Pattern Matching

.	Match any single character
*	Match zero or more occurrences of preceding character
?	Match zero or 1 occurrences of preceding character
+	Match 1 or more occurrences of preceding character
[...]	Match any character in the set (or range)
[^...]	Match any character not in the set (or range)
^ , $	Match characters at the beginning, end of a line
\< , \>	Match characters at the beginning, end of word
\(... \)	Save this pattern as a grouping to be recalled later
\n	Contents of the n^{th} grouping when replacing text
\c	Turn off the special meaning of the character *c*

Printing

lpr *filename*	Print *filename* to the default printer
lpr -P*name filename*	Print *filename* to the printer *name*
command | lpr	Print the output of *command*
a2ps *file* | lpr	Format and print text file *file* for easier reading
lpq	Display jobs in the default print queue
lpq -P*name*	Display jobs in the specified queue
lprm *jobnum*	Remove *jobnum* from the queue. Use lpq to obtain job numbers.
lprm -P*name jobnum*	Remove *jobnum* from the specified queue.

CS Printers

Name	Room	Model
c012lpr	C012	HP LaserJet 8150N
c014lpr	C014	HP LaserJet 8150DN
c014clr	C014	HP Color LaserJet 3800
c107lpr	C109	HP LaserJet 4250
c122lpr	C122	HP LaserJet 4250
c122clr	C122	Lexmark C720 Color
b118lpr	KH B118	HP LaserJet 4350

Command Line Substitution

\`*command*\`	Command substitution. Replace the command with its output on the command line. Ex: mail \`whoami\`@fang
$*variable*	Variable substitution. Replace the variable with its value on the command line. Ex: echo $HOME
? * [...]	Filename Substitution. Insert the wildcard expansion into the command line. Ex: ls *.c or ls [abcd]*
!! or !*n*	History Substitution. Replace the history command with its value. !! = last command
~*username*	Replace with the full path of a user's home directory. A ~ by itself means your home directory. Ex: cd ~*username* or cd ~

History

history	List all commands typed so far
!!	Repeat the last command
!*n*	Repeat command *n* from the history list
!*pattern*	Repeat last command beginning with *pattern*
^*pattern1*^*pattern2*	Repeat last command but replace *pattern1* with *pattern2*
Up Arrow	Recall previous command in stack
Down Arrow	Recall next command in stack

Job Control

command &	Run *command* in the background.
CTRL-C	Cancel the current foreground job
CTRL-Z	Suspend the current foreground job
CTRL-T	Show the status of the running job (freebsd)
bg	Resume suspended job in the background
fg	Bring suspended or background job to the foreground
stop	Suspend a background job
jobs	List all running and suspended jobs
%*jobnum*	Resume job *jobnum* in the foreground
%*jobnum* &	Resume job *jobnum* in the background
stop %*jobnum*	Suspend job *jobnum*
kill %*jobnum*	Kill job *jobnum*
set notify	Immediate notification of job state changes
cmd1 ; *cmd2*	Execute multiple commands on one line

Process Management

top	Display processes using the most CPU time
ps	Display processes running for the current login session
ps aux	Display all running processes on the system
ps auxU *user*	Display all processes belonging to *user*
kill *pidnum*	Kill a process using the process id number from the ps or top command
pkill *name*	Kill a process by name

Environment

id *username*	Display info about the current logged in user or the user *username*
whoami	Return your username
uname -a	Display OS information
env	Dump the environment settings
setenv *var value*	Set the environment variable *var* to *value*
unset *var*	Unset the environment variable *var*
echo $*variable*	Print the value of *variable* to the screen
alias	Display a listing of current aliases
alias *short long*	Create an alias *short* for a *long* command
source	Reread a shell configuration file
exit	Exit from the current login session
.alias.user	User defined aliases stored in this file are initialized for each login. DogNET only.
.login.user	Shell configurations stored in this file are initialized for each login. DogNET only.

Communication & Remote Access

hostname	Display the hostname of your workstation
mail *username*	Send an email to another user
pine	Program for sending and receiving mail.
finger *user@fang*	Query host *fang* for information on *user*
w	Who is logged in and what they are doing
mesg y | n	Turn on or off incoming messages
talk *username*	Interactive communication with *user*
write *username*	Write to another user's terminal window
ssh *user@rhost*	Initiate a login session to *rhost* as *user*
sftp *user@rhost*	Initiate a secure FTP connection
BitchX	IRC client
lynx	Web browser

Filesystem

df -h	Display filesystem utilization
du -sk *directory*	Calculate total disk usage for *directory*
quota	Display your total disk usage and quotas
ln -s *source link*	Create a symbolic *link* pointing to *source*
gzip *file*	Compress a file (or use bzip2)
gunzip *file*	Decompress a file (or use bunzip2)
tar -cf *tarfile*.tar *files*	Archive *files* into the tarball *tarfile*.tar
tar -xf *tarfile*.tar	Extract all files from the tarball *tarfile*.tar
dos2unix *filename*	Convert DOS end of line characters to UNIX format
unix2dos *filename*	Convert UNIX end of line characters to DOS format

GUI Applications

acroread / xpdf	PDF viewers
firefox / chromium	Web browser
pidgin	Instant Messenger
ghostview / gv	Postscript viewers
gimp	Graphics editor
mplayer / xine	Video players
netbeans	Source code editor
rdesktop	Windows remote desktop client
libreoffice	Word Processing and spreadsheet
xchat	IRC client
xdvi	DVI Previewer
xine	MP3 player
xsane	Page scanning program
xterm	Terminal window
xv	Graphics Viewer

Miscellaneous Commands

awk	Pattern scanning and processing language
bc	Command line calculator
cal	Show the calendar for the current month
clear	Clear the screen
cut	Select portions of each line
cvs	Concurrent Version System
date	Display the current date and time
echo *string*	Echo *string* to the screen
figlet *string*	Display *string* in larger characters
strings *file*	Search for strings in binary *file*
screen	A window manager for the command line. Good for use with unstable connections.
script *file*	Record terminal session to *file*
sed	Stream editor
sleep *seconds*	Pause for a specified number of seconds
svn	Subversion concurrent versioning system
fossil	Fossil revision control system
tee	Pipe to another command and to the screen
man *command* man -k *keyword*	Learn more about *command* or *keyword* from the system manual

Vi Quick Reference

MOVEMENT

By Line

*n*G or :*n*	Go to line *n*
0 , $	first, last position on line
^ or _	first non-whitespace character on line
+ , -	first character on next, previous line

By Screen

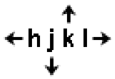
ctrl-f , ctrl-b	scroll forward, back one full screen
ctrl-d , ctrl-u	scroll forward, back half a screen
ctrl-e , ctrl-y	show one more line at bottom, top
H , M , L	go to top, middle, or bottom of the screen
z , z. , z-	position line with cursor at top, middle, bottom

Marking Position on Screen

m*p*	mark current position as *p* (a..z)
`*p*	move to mark position *p*
'*p*	move to first non-whitespace on line with mark *p*

Miscellaneous Movement

:*num*	jump to line *num*
h, j, k, l	move left, down, up, right
f*m* , F*m*	forward, backward to character *m*
t*m*	forward to character before *m*
T*m*	backward to character after *m*
w	move to next word (stops at punctuation)
W	move to next word (skips punctuation)
b	move to previous word (stops at punctuation)
B	move to previous word (skips punctuation)
e	end of word (punctuation not part of word)
E	end of word (punctuation part of word)
G	move to end of file
) , (	next, previous sentence
} , {	next, previous paragraph
]] , [[	next, previous section
%	goto matching parentheseis ( ) { } [ ]



EDITING TEXT

Entering Text

a , A	append after cursor, at end of line
i , I	insert before cursor, at beginning of line
o , O	open line below, above cursor
c*m*	change text (*m* is a movement sequence)
c*nm*	change text (*n* number of movement characters)
C	change rest of line

Cut, Copy, Paste (Working with Buffers)

d*m*	delete (*m* is a movement sequence)
dd	delete line
D , d0	delete to end, beginning of line
x , X	delete character under, before cursor
y*m*	yank to buffer (*m* is a movement sequence)
yy or Y	yank current line to buffer
p , P	paste from buffer after, before cursor
"*b*dd , "*b*yy	cut, copy line into named buffer *b* (a..z)
"*B*dd , "*B*yy	append to buffer *B* (A..Z)
"*b*p	paste from named buffer *b*

Miscellaneous

n>*m*	indent *n* lines (*m* is movement)
n<*m*	un-indent left *n* lines (*m* is movement)
.	repeat last command
u , U	undo last change, all changes on the line
J	join end of line with next line
ctrl-g	show status
~	toggle upper/lower case
:set nu , nonu	turn on, off line numbers
:set ic , noic	turn on, off ignoring case for searches
:set autowrite	Automatically save the file upon CTRL-Z
:set autotab	Automatically indent new lines
:set tabstop=#	Set distance for indentation to # lines

Searching and Replacing

/*string*	search forward for *string*
?*string*	search backward for *string*
/*string*/+*n*	search forward for *string*, move down *n* lines
n , N	repeat search forward, backward
:s/*old*/*new*/	replace next occurrence of *old* with *new*
:s/*old*/*new*/g	replace all occurrences on the line
:*x*,*y*s/*old*/*new*/g	replace all occurrences from line *x* to *y*
:.,+*y*s/*old*/*new*/	replace all matches on current and next *y* lines
:%s/*old*/*new*/g	replace all occurrences in the file
:%s/*old*/*new*/gc	replace all occurrences in file with confirmation
:%g/*pattern*/*c*	run command *c* on all matching lines
&	repeat last search and replace command

File Commands

:w	write changes
:w!	write changes, overriding permission errors.
:w *file*	write to file *file*
:wq	write changes and quit
:r *file*	read and insert text from external file *file*
:e *file*	Open *file* in the current editor window
:e!	reopen current file, discarding changes
:q!	quit without saving
:n , :prev	edit next, previous file in argument list
:.!*program*	replace line with the output of *program*

Examples

c5w	Change 5 words of text
y3 <enter>	Yank 3 lines into the buffer. Paste with p.
:5,10d	Delete lines 5 through 10
:%g/*mystring*/d	Delete all lines which contain the text *mystring*
:%g/mystring/s/old/new/	On all lines which contain *mystring*, Replace the text *old* with *new*

2/23/06 - merantn