Introduction to Unix

Basics

- You enter commands to the computer at the command prompt, and they are interpreted by the shell, a special helper program. Common shells are the Bash shell, the Bourne shell, and the tcsh shell. For beginners, they are all the same.

- To run a program, just type its name at the command prompt and hit enter. If it is a graphical program, a new window will appear.

- If a command takes arguments, type them after the command, separated with spaces.

- Some arguments are the names of files; others are options, which are usually a dash character followed by a letter, or two dash characters followed by a word. As the name implies their use is optional, and usually modifies the operation of the program.

- Previous commands are remembered. You may retrieve them by hitting the up arrow key.

- Commands may be edited (with the right and left arrow keys) before you hit enter.

- If you want the command prompt to reappear before the program finishes, add an ampersand character (&) to the end of the command. This runs the program in the background. This is very useful when running programs that open new windows, because otherwise your shell would be disabled until the other program exits.

Redirection and Pipes

The input to or output from a program may be redirected to files or piped to other programs. This involves using special characters on the command line.

prog < file Use the contents of file as the input to prog, instead of the keyboard.

prog > file Redirect the output of program to file. If file exists, overwrite it.

prog >> file Append the output of program to the end of file.

prog1 | prog2 Use the output of prog1 as the input to prog2. This is called a pipe.

Wildcard Characters

It is often necessary to specify as arguments to a command several files that have very similar names. To avoid excessive typing, wildcard characters may be used to build up a pattern. The shell looks at your pattern and replaces it with any files that match it, as if you had typed them yourself.

? Matches exactly one character; no more, no fewer.

* Matches any number of characters, including none at all.

Examples:

AB?DE Matches ABCDE, but not ABDE or ABCDDE.

AB*DE Matches ABDE, ABCDE, and ABCDDE.

AB*DE* Also matches ABDE, ABCDE, and ABCDDE.

AB*DE? Does not match any of ABDE, ABCDE, or ABCDDE.
Unix Command Cheatsheet

Getting In and Getting Out

ssh user@machine [command]

Makes a connection from your current machine to a remote one. Also provides encryption so that
others on the net can’t see what you’re doing. If command is supplied, it runs command on the remote
machine and exits. exit

Terminates a login session.

Looking at What Is Available

ls [-l] [-R] [file...] [directory...]

Lists the files in the current directory, or in the directory(s) you specify, and/or the files you specify.
The -l option adds file sizes and dates, among other things, to the list. The -R option shows not
only files in the current directory (or the one you specify), but also files in all directories below that.

cd [directory]

Changes the current directory to your home directory, or to the directory you specify.

pwd

Prints the current working directory.

Examining What You Find

cat file...
Types the contents of a file to the screen.

less file...
Displays the contents of a file on the screen, pausing at each screenful, and allowing you to scroll
forward or backward.

head [-n #] file...

Shows the first 10 lines of a file, or the number of lines you specify.

tail [-n #] file...

Shows the last 10 lines of a file, or the number of lines you specify.

Making Changes

cp source... destination
Copies a file or group of files to a destination. If source is more than one file, then destination must
be a directory, which will then contain copies of the files listed in source.

mv source... destination
Moves or renames a file or directory. If source and destination are both in the same directory
(or no directory is specified), the file in source is renamed to destination. If destination is a
directory, then the file(s) listed in source are moved there. If both source and destination are
both directories, the directory in source is moved or renamed to destination.

rm [-r] file... [directory...]

Removes a file or list of files. If -r is specified, then you may also specify entire directory trees to
be removed. Be careful with this one.

scp [-C] [user@[machine:]][file] [user@[machine:]][file]

Copies a file from one computer to another. Usually one or more parts of the source and destination
are not necessary. For example, ssh file machine: copies a file from the current directory of the
computer you are using to your home directory on another machine. If your username is different
on the remote machine, use user@machine: for the second argument.

Getting Help

man program

Pulls up the “manual page” listing how to use program.
More Unix Commands

Creating and Deleting Directories
mkdir [path]directory...
Creates a subdirectory in the current directory, or in one you specify.
rmdir directory...
Removes a directory. If the directory is not empty, an error occurs.

Wrestling with files
find directory [-name file]
Looks in directory and all those below it for files named file. If file has wildcards in it (* or ?), be sure to put quotes around it, or you won’t get what you want.
grep [-i] [-n] [-l] pattern file...
Scans each file specified for lines matching pattern. If -i is specified, the search is case insensitive.
By default, any matching line is printed. -n forces the line number of the match also to be printed,
-l, useful if you specify more than one file, only prints the name of the file(s) that have the matches.
awk [-Fchar] ‘patternaction ...’ file...
Extends the functionality of grep with a programming language that is an easier dialect of C. Each
line of each file specified is read, and each pattern specified is applied to it. If a given pattern
matches that line, the corresponding action is performed. Before the patterns are applied, the line
is broken down into fields, which are portions of text separated by blank space (or by the character
specified in the optional -F option). The names of the fields are a dollar sign followed by a number
specifying which field, from left to right, you want, starting with 1. For example, $3 would be the
third field. The special field $0 contains the entire line.
pattern can be many things:

/text/ Lines containing text match. The period (.) is a wild card that matches any character.

$1=="text" This would test just the contents of a single field.

$2==3.2 If you’re testing for a number, leave the quotes off.

$1=="text" && $2==4 This requires both tests to be true.
$1=="text" || $2==4 This requires either the first or the second test to be true.

sqrt($2)+sin($5)>exp(2.3) Complicated equations can be used also.
a==5 User-defined variables can also be tested.

NF==3 The special variable NF equals the number of fields on the current line.
NR==20 The special variable NR equals the number of lines processed so far.
BEGIN This special pattern matches once before any lines are read.
END This special pattern matches once after all lines are read.

In addition, action can be many things:

print $1,"is equal to",$2 Outputs parts of the line and user-supplied text.
a=5 User-defined variables can be set.
if(NR==3) print $2 Several control structures are available.
for(i=1;i<10;i++) print i,$1*i Here’s another example.
a=5; print $3 Multiple commands are separated by a semicolon.
The Emacs Editor

Emacs is a text editor with a long history in the Unix world. Because of this, it has an incredible number of features. There are modes for editing plain text files, editing Fortran or C source code files, html documents, LaTeX files, etc. There are also modes for compiling programs, reading mail, scanning directories, surfing the web, etc. Some people believe that all work you can do on a computer can be done in Emacs.

Beginners sometimes find commands hard to use and remember. There are basically two types of commands in Emacs. The first kind involves hitting a key in combination with either the Control or Alt keys. These are abbreviated with C- and M- (the M stands for Meta, another name for the Alt key). For example C-x means hit Control, while holding it down hit x, then release both. C-x C-f means do the above sequence, and then do the same procedure with the f key.

The second type of command involves hitting the special sequence M-x (Alt key and x together), and then typing the name of a command. This is usually only for less frequently used commands.

File Commands
C-x C-f Open a new file. You’ll be prompted for the name on the bottom line.
C-x C-s Save the current file to disk.
C-x C-c Exit Emacs. You will be prompted to save any unsaved changes.

Editing Commands
C-a Jump to the beginning of a line.
C-e Jump to the end of a line.
C-d Delete a single character. The del key will also work on proper terminals.
C-g Cancel any operation. If Emacs was asking you something, it’ll stop.

Searching
C-s Search forward. As you type each letter of your search text on the bottom line, the cursor will jump to the first spot that matches what you’ve typed so far. While in search mode, you may find the next match with another C-s. When you’re happy with where the cursor is, hit enter to leave search mode.
C-r Search backward.
M-% Search and replace. You’ll be asked for the search text and text with which to replace it. Hit y for each match you want to replace, or ! to replace all matches immediately.
C-x C-x Return to the “previous spot”, also known as the “mark”. This is useful to return to where you were before performing a search.
M-> Jump to the end of the file. The previous spot will be remembered.
M-< Jump to the beginning of the file.

Window Commands
C-x 3 Split the current window horizontally into two smaller windows, one beside the other.
C-x 2 Split the current window vertically into two smaller windows, one atop the other.
C-x 1 Kill all but the current window. Does NOT close any files, just your view of them.
C-x o Jump the cursor to the next window. This is the letter o for “other”, not zero.

Other Commands
M-x bury-buffer Cycle through the list of files you’ve opened.
C-x k Kill the current file from Emacs’ memory.
M-x tex-buffer Run contents of current file through TeX.
M-x revert-buffer Update buffer to reflect changes on disk made by another program.
M-x display-time Show current time on “modeline” at bottom of screen.
M-x column-number-mode Show column number of cursor in modeline.