**Bash/Unix Command Cheat Sheet**

Leaning how to utilize the command line in Unix is crucial for navigating your computer

as well as making, deleting, and searching your files. Many of these commands will

come in handy on a very regular basis, so they should become second nature rather

quickly. Until then, feel free to use this sheet!

**Some General Notes and Definitions**

I will be using the term **“directory”**. Think of these as files, or places on your computer.

For example, your desktop is a directory. A folder on your desktop is a directory within

the desktop directory. etc, etc....

Your **“home”** directory is your starting place. Whenever you first open up the terminal,

you start from here. Often, this is denoted by a “~”. To go back to your home directory:

cd ~

An object’s **“path”** means its location on the computer (think of it as an address that tells

you exactly where something is). For example, a file named “file.txt” in the directory

“text\_files” in the folder “MyStuff” on your Desktop would be:

~/Desktop/MyStuff/text\_files/file.txt

Think of your **path** like a stream. Your home directory is the highest point and

everything flows out from there (i.e. everything is “down stream”). Everything within a

directory is “down stream” from that directory.

Any command that takes a file or directory name as input can also take a path as input.

For example, the following lines do the same thing (they remove the file called “file.txt”)

rm file.txt

rm ~/Desktop/MyStuff/text\_files/file.txt

When you want to perform an action on multiple files in a directory, using “ \* ” is useful.

rm \* removes all files in a directory (dangerous!)

rm \*.txt removes all files with names that end in ‘.txt’

rm test.\* removes all files with names that start with “test.”

rm test\*.txt removes files with names that start with “test” and end with “.txt”

Many of the commands discussed below have options connected with them. Options

are usually denoted by upper or lowercase letters (and sometimes numbers) after the

command, usually preceded by a ‘-’. For example,

<command> -[option1][option2]

ls -lf

If you forget the exact syntax to use with a command, or need to find what options are

available, use the man command. For example:

man ls

This will give you the manual page for the ls command, including a list of the available

options and their actions.

**Tab completion** is very useful. When you are typing in a command or the name of a

file, hitting tab will have the computer attempt to auto complete what you were typing

based on the available options. If there is more than one option, it will give you a list of

the possible commands/files. For example, type “ip[TAB]”. The computer gives you a

list of all files, directories, and commands that start with “ip”. Type in “ipy[TAB]” and

the computer auto completes the command to “ipython”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Listing files in a directory

ls Gives a list of (almost) everything

in your current directory

ls -a Lists everything in the directory

(including files that start with a ‘.’)

ls -l Lists Things in an up-down list

with some information, like when

the file was last edited and the

size of the file

ls -la Does both options!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Navigating directories

cd [directory path] Moves you to a new directory

cd .. Moves you “up” one directory

cd ../.. Moves you “up” two directories

pwd Shows you where you are (current path)

cd ~ Moves you to your home directory

which [command] Shows you the path to the command that you’re running

# Making and moving files and directories

mkdir [directory name] Make a new directory within your

current directory

rmdir [directory name] Remove a directory with the

specified name. This will only

work on empty directories

rm [file name] Remove a file with the specified

name. Will not work on

Directories

rm -rf [file name] Removes a file with the specified

name. This WILL work on

directories that are not empty.

cp [file1] [file2] copy file1 to file2

mv [file1] [file2] moves file1 to file2. This can be

used to move the location of a file

or it can be used to rename files.

wget [webaddress of file] Download a file from the internet

into your current directory

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Writing and Reading Files

echo hello prints out “hello” to the screen

echo hello > [file] opens a file and prints “hello” to

it. This will overwrite the file and

delete whatever was in it at first

echo hello >> [file] Same as above, but it will append (add

to the end of) the file rather than overwrite it

cat [file] Print out the contents of a file

cat [file1] > [file2] Print the contents of file1 to file2

(overwriting file2)

cat [file1] >> [file2] Append file2 with the contents of

File1

grep [something] [file] Print out the lines in file that

include the string “something” in

the line

more [file] View the contents of a file. Better

than cat if the file is long. Push

spacebar to scroll down

tail -[number] [file] View the last [number] of lines of

a file

head -[number] [file] View the first [number] of lines of

a file

wc [file] Prints number of lines, words,

and size of file

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Sorting Streams

sort –k number [file] | more Alphabetical sort, where number

refers to the column.

sort –n –k number [file] | more String numerical sort.

sort –g –k number [file] | more General numerical sort.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Making “tar balls” (compressed files)

tar cfz filename.tar.gz [list of files] Compress files into

filename.tar.gz. (note the

lack of a “-” before the cfz option here....)

tar xfz filename.tar.gz Decompress filename.tar.gz

And put all the files in it into your current directory

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Some examples using “**piping**” ( “|”)

cat [file] | grep word Print the lines in file that include

“word” in the line

grep word [file] | wc Gives the number of lines, words,

and size of the portion of file that

grep returns as having “word” in

the line