CMPSC 102
Discrete Structures
Fall 2018

Reading Supplement
Tips on Using Linux and the Command Line Interface

Directories, Folders, CLIs, GUIs

A "folder" is the same thing as a "directory"—a place to organize and save files.

You can navigate between folders in one of two ways—by typing commands through the terminal window (this is sometimes called a "CLI," or "command line interface") or by the more familiar method of clicking and double-clicking on images of folders and icons representing files (this is an example of a "GUI," or "graphical user interface").

In this class we concentrate primarily on the command-line interface using the terminal window., since this material is valuable to people who go on to become "power users"—professional developers, systems administrators, etc., who want to work as efficiently as possible. You are free to use the GUI for your own programming, but instructions will almost always be given in CLI form.

Home Directory vs. Desktop

You have a "home directory" that has the same name as your account, i.e., *Buffalo Bull* has a home directory named "`buffaloBill`." Within this directory is a subdirectory named "`Desktop`." When you see a file icon or folder on your screen, that is actually saved in your `Desktop` subdirectory.

When you right-click on the screen and choose "Open in Terminal," commands typed in your terminal window will be relative to the `Desktop` directory since that is where you clicked to open the window. ***The desktop is* not *your home directory***—it is one level below it.

It is okay to store things on the `Desktop`. Just be aware that certain commands described in Table 1, such as the command "`cd`," will automatically return you to your *home directory* and *not* to the `Desktop`.

Using the Terminal Window and Keyboard

If you would like the terminal window application to appear in your list of applications on the left (the application "launcher"), do the following:

- Right-click on the screen and select the "Open in Terminal" command. The icon for the terminal application should appear on the left, resembling the following graphic:  . ***Note: since you are not in your home directory, you must change the directory to the home.*** To get to the home folder, type the command "`cd`".

- Right-click on the terminal icon on the left and select "Lock to Favorites". This will permanently add the terminal window to the docker on the Ubuntu desktop.

Remember—the next time you launch the Terminal application from this new icon, it will open in your home directory, not the desktop.

Prompts

Terminal window commands are typed at the *prompt*. The prompt is usually some combination of your username, your computer (i.e., "`prairie`") and the current folder or directory. For instance, in the text below, the prompt "`buffaloBill@prairie\$`" means that user `buffaloBill`, working at machine `prairie`, is in the home directory. The command "`cd cs102`" changes to directory `cs102`, and the new prompt indicates this. Typing the command "`cd lab1`" changes to the `lab1` subdirectory, and the prompt now shows this as well:

```
buffaloBill@prairie:~$ cd cs102              [in home directory]
buffaloBill@prairie:~/cs102$ cd lab1         [now in cs102 subdirectory]
buffaloBill@prairie:~/cs102/lab1$            [now in lab1 sub-subdirectory]
 ... etc. ...
```

If you ever see a prompt like "`buffaloBill@prairie:~/Desktop`" then you know that you are in the `Desktop` subdirectory of your home directory.

Arrow Keys

The "arrow keys" at the lower right of your keyboard enable you to move back through the "history" of commands you have typed. For instance, it is often the case, when developing a program, that you will have to repeatedly type certain commands as you debug your program. There is no need to keep retyping these commands! Using the up-arrow key (" ↑ ") will let you rapidly move back to an earlier command so all you need to do is press the "Enter" or "Return" key. You can also use the left-arrow and right-arrow keys ("←" and "→") to move back and forth in a typed command to correct typing errors or to use a modified version of an earlier command.

Special Keys

The "Ctrl" key (bottom row, below the "shift" key) is used in conjunction with other keys for special commands. For instance, the key combination Ctrl-C will "kill" a process that is running in the terminal window. (Why would you ever want to do that? We will see that if you write a program that goes into an "infinite loop," you must terminate it with a Ctrl-C command.)

The key combination Ctrl-Z will *suspend* a process that is running in the terminal window; it can later be *resumed* in a number of ways. This is occasionally useful and you can resume suspended processes by just typing a percent sign (%) followed by the job number in the terminal. Note: you can locate the job number at the terminal while the program is running by typing, `ps -aux | grep <yourUserName>`.

If you press the key labeled "PrtScn", the computer will take a "snapshot" of the screen and save it in folder named `Pictures` (another subdirectory of your home directory). If you hold down the "Shift" key while pressing the "PrtScn" button, a plus-shaped cursor will appear and you can use it and the left mouse button to select a rectangular area of the screen. When you release the mouse button, you will have a snapshot of the selected area.

Linux Commands

Table 1 shows some common Linux commands. You will note that many of them are just two letters ("`rm`" for "remove," "`cp`" for "copy," etc.).

Command	Meaning	Example
`ls`	List files in current directory	`buffaloBill@prairie:~/cs102/lab1$ ls` `Lab1.class Lab1.java`
`mkdir` *name*	Make a new directory called *name* in the current directory	`buffaloBill@prairie:~/cs102$ mkdir lab2`
`cd` *name*	Change to directory *name* in the current directory	`buffaloBill@prairie:~/cs102$ cd lab2` `buffaloBill@prairie ~/cs102/lab2$`
`cd`	Change to home directory	`buffaloBill@prairie:~/cs102$ cd` `buffaloBill@prairie:~$`
`cd ..`	Change to directory one level up	`buffaloBill@prairie:~/cs102/lab1$ cd ..` `buffaloBill@prairie:~/cs102$`

`cp` *name1 name2*	Copy file *name1* to *name2*	`buffaloBill@prairie:~$ ls`
		`Fun.class Fun.java`
		`buffaloBill@prairie:~$ cp Fun.java Lab2.java`
		`buffaloBill@prairie:~$ ls`
		`Fun.class Fun.java Lab2.java`

`mv` *name1 name2*	Rename file *name1* as *name2*	`buffaloBill@prairie:~$ ls`
		`Fun.class Fun.java`
		`buffaloBill@prairie:~$ mv Fun.java Lab2.java`
		`buffaloBill@prairie:~$ ls`
		`Fun.class Lab2.java`

`rm` *name*	Remove file *name*	`buffaloBill@prairie:~$ ls`
		`Fun.class Lab2.java`
		`buffaloBill@prairie:~$ rm Fun.class`
		`buffaloBill@prairie:~$ ls`
		`Lab2.java`

`pwd`	Shows which directory you are in	`buffaloBill@prairie:~$ pwd` `/home/b/buffaloBill/Desktop`
`exit`	Close the terminal	

Table 1: Some Common Linux Commands