UNIX Commands Tutorial

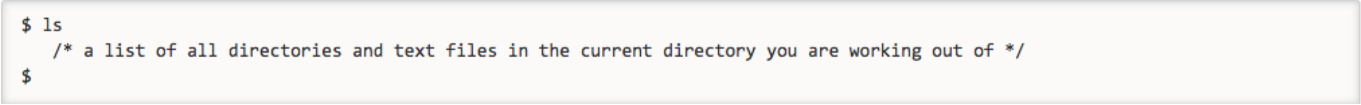
Let's learn a little bit about the **UNIX Command Line**:

1. The UNIX Command Line has many capabilities, far more than what you are given with the Graphical User Interface(GUI) of your computer
2. The Command Line allows you to go through directories, manipulate files and folders, and work with text files.
3. The Command Line interfaces are often preferred by more advanced computer users, because they provide a more concise and powerful way of controlling a program or operating system.
4. You will find that as you go throughout college and beyond, you will be using the Command Line quite a bit, and we will start this journey now.

To start this tutorial, we are going to give you a list of commands to follow. We are going to create your lab directory, and your lab01 directory.

Open the computer terminal emulator on one of the Linux-based lab machines by clicking **Applications** on the top left corner of the screen. Then choose **System Tools → MATE Terminal** We have now opened up the Command Line: We will start going through some commonly used UNIX Commands.

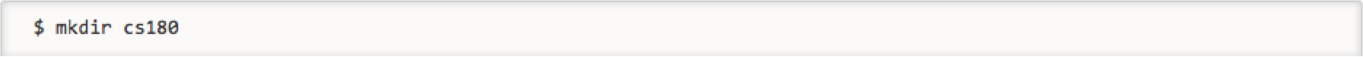
First we want to see what other directories are in our home directory. We will do this by using the **ls** command:



```
$ ls
   /* a list of all directories and text files in the current directory you are working out of */
$
```

NOTE: We use the ls **command to simply list the files in the current working directory. Other parameters can be added after the ls, but for now, we won't bother with that.**

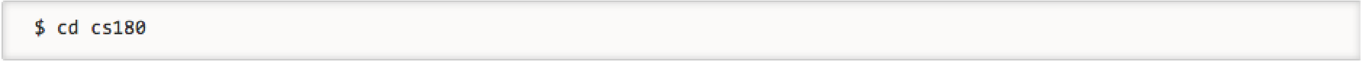
We now want to create a new folder called “labs”. This is where you will hold all of your labs for the semester. In order to create a new folder (directory) called “cs180” on the desktop, we will use the **mkdir** command:



```
$ mkdir cs180
```

NOTE: Now keep in mind that when you use the mkdir **command, the command line will not automatically change to the created directory. So if you want to change to the directory you just created, you must use the** cd **command again.**

We now want to change or step into that CS180 directory. We will use the **cd** command to step into that desktop directory:



```
$ cd cs180
```

NOTE: We use the cd **command to navigate through your computer. You will find that you will be using this command more than any other. It allows you to navigate to subdirectories, parent directories, and even the root directory**

Now that we have moved into our project directory, we now want to create a directory specifically for this first lab. Repeat the last two steps by making a directory called “lab1” and then change into that directory:

```
$ mkdir lab1
$ cd lab1
```

For this “lab1” directory, we now want to create a file called “HelloWorld.java”. To do this, we will use the **touch** command, which simply allows you to create new files:

```
$ touch HelloWorld.java
```

If you run the **ls** command next, then you will see that “HelloWorld.java” is a text file listed in the directory “lab1”:

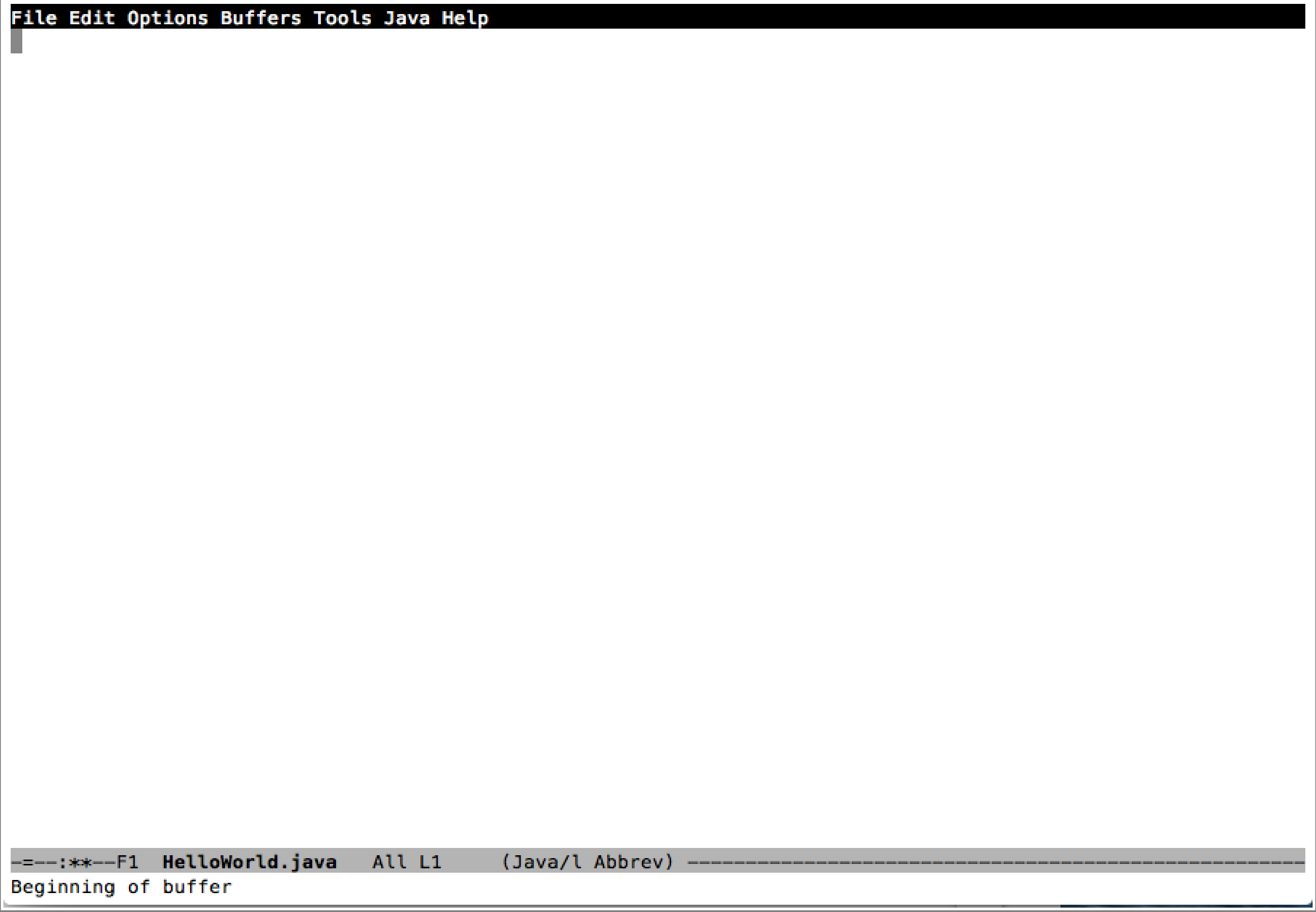
```
$ ls
HelloWorld.java
$
```

Now that the file has been created, we now want to open it in a text editor. We are using emacs as our text editor. We will use the command **emacs** to open the file in emacs:

```
$ emacs HelloWorld.java
```

The following window should pop up. This is what the emacs editor looks like:

Emacs Text Editor



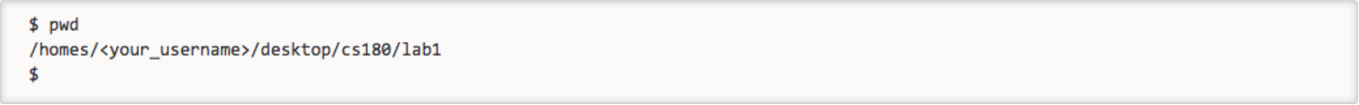
```
-=--:**--F1  HelloWorld.java    All L1     (Java/l Abbrev) ------------------------------------------------------
Beginning of buffer
```

NOTE: To exit out of the file, press Command+C, and then Command+x. After that, hit y for “yes”, and it will take you back to the command line. More emacs commands will be talked about after this UNIX walkthrough

For now go ahead and close this window and go back to the command line. There are some other useful commands that you can use for terminal.

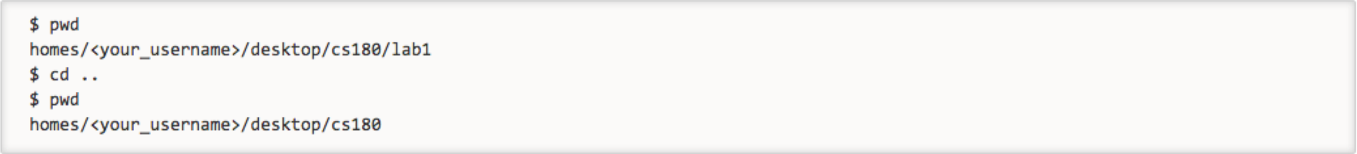
NOTE: If you are creating a file, and if you want to open it in emacs **right after it's been created, you actually don't have to use the** touch **command. If you want to open up a nonexistent file in emacs, emacs will automatically create that file and put you in the editor.**

The **pwd** command allows you to simply see which directory you are currently in. If you run this command in your lab1 directory, then the following will appear:



```
$ pwd
/homes/<your_username>/desktop/cs180/lab1
$
```

When using the cd command, you can use **cd ..** to go to the parent directory. If you are in lab1, then this command should put you in it's parent directory (cs180):



```
$ pwd
homes/<your_username>/desktop/cs180/lab1
$ cd ..
$ pwd
homes/<your_username>/desktop/cs180
```

NOTE: If you want to go all the way back to the root directory, simply run “cd ~**“**

USEFUL TIPS

- UNIX has a **built-in manual**, so you can read a full list of the command's functionality, arguments, and options with the **man** command:
  `$ man ls`
  Exit reading of a **man** page by pressing the **q** key. Press the space bar to keep scrolling through the page.
- Press the **tab** key part way through a file name or directory name and the command line interface will fill in the rest. Pressing the **tab** key twice will print the names of all files or directories that begin with what you typed.
- Reuse commands you've already used by pressing the up and down arrow keys.
- The terminal is **case sensitive** — **desktop** is different from **Desktop**.

Below is a list of the commands that can be useful to you:

Command	Description
ls	The ls command is used to view files, folders, and directories in a given working directory
cd <directory>	The cd command will allow the user to change between file directories
mkdir <new_directory>	The mkdir command allows the user to make a new directory.
touch <new_file>	The touch command allows users to make new files. Eq. to the mkdir command for directories

Command	Description
emacs <file>	The emacs command allows the user to open up a file in the emacs text editor.
javac <Example.java>	The javac command allows the user to compile a Java program
java Example	The java command is used to run a Java Program
pwd	The pwd command is used when the user wants to see what their current working directory is.
rm <file>	The rm command allows you to remove a file from the wherever you are. You cannot recover the file once the command is used.
mv <old_location> <new_location>	The mv command allows a user to move a file to another folder or directory.
man <command>	The man command is used to show the manual of the inputted command.
clear	The clear command is used if you want a clean terminal.
cat <file>	Allows you to see the contents in a file

It is very important that you practice these commands in your free time You will need these commands not only for this semester, but for future classes as well