1 Set-up for the Very First Session Only (CS192)

We will be working in the GNU/Linux environment in this class.

The first time you sit down at a CS student lab computer you will need to **log in, and change your password.**

To get started:

(Note if you have done steps 1-3 for another CS course, just login)

1. Your CS department login name will normally be your CSU EID (all lower case), and your initial password will be your nine digit student identification number. If you try to log in and this does not work, ask the instructor for help.
2. Type your login followed by the **Enter** key. (Remember, case matters in GNU/Linux!)
3. Your initial password is your student ID number. Type in your student ID number followed by the **Enter** key. No dashes or spaces, please, and do not use the number pad.
4. This should begin the process to log you into the computer (it may take a bit of time to do this).
5. There may be a welcome/setup in GNOME (this is the default desktop manager). You may click "next" through all three prompts that appear upon first login.

1.1 Logging Out (for the end of the session)

*Important: Please be sure you **always** log out when you leave a system.*

When you are ready to log out, click the icon with the down arrow in the right hand top corner, select your login name, then select the **Log out** option from the list, and select the **Log out** button (then confirm the log out request) from the End session window that appears. (Note: Do not select **Shut Down** as this will reboot the system and log-off anyone else who is currently logged-in. In fact, **never** shut down **any** CS department computer.)

2 Getting Started in GNU/Linux

You are now in the Fedora Linux distribution, using the GNOME desktop environment. Notice that the screen looks different from a Microsoft Windows, or Mac OS X computer. There are several icons across the top of the screen.

The first thing to do is to change your password to something more secure. First you will need to get a terminal window that lets you talk more directly with the operating system, then you will type a command to let you change your password.

2.1 Opening a New Terminal Window

Terminal windows let you talk directly to the GNU/Linux operating system. You will need terminal windows frequently when you use the GNU/Linux environment. In this class we expect you to use terminal windows to interact with the GNU/Linux operating system instead of the folder displays with which you may be more familiar. Both are useful, however, terminals tend to be more efficient and powerful once learned. To open a terminal window:

1. Move the mouse to the upper left-hand corner.
2. Left-click on Activities.
3. Click in search box near the top of the screen.
4. Select **Search Box** (usually found in the upper right-hand corner).
5. Type **Terminal** and select the terminal of your choice, or drag it to your tool bar at the left (the list is alphabetical; do not select the one with the `$` symbol).
6. A new general-purpose window should appear on your screen.

2.2 Changing Your Password

We recommend that you change your CS department password to something other than your nine digit student ID number. You may use the same password as your CSU account or may create a different one.

A good password includes letters, digits, and special characters (like a comma) and has a length of at least ten characters (fifteen if you don't use special characters). It is one you can remember easily, but not one other people can easily guess. Choose a new password and type it twice, each time followed by the **Enter** key, as explained below. To change a password, type (in the terminal window) exactly this command:

```
passwd
```

followed by the **Enter** key. Follow the directions you see. There will be no "echo" from the display, but type it anyway (do not use backspaces). Presently you will see a prompt that says:

```
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
```

3 Using Mozilla Firefox Browser and Canvas

Most of you are likely experienced with web browsers. We will begin this seminar using Mozilla Firefox. Start it up by selecting **Activities** and type **Firefox** in the search box. It should look similar to other browsers you may have used. Check out a web page, such as the CSU CS Department web page, by typing the URL into the address bar (the white banner high up and near the center).

```
http://www.cs.colostate.edu
```

Right-click anywhere in the window and select **View Page Source** from the menu that pops up. A new window will pop up showing you the HTML markup that is used to build the page. Look at it. See if you can find similarities between HTML and the web pages it displays. Close the page source window.

We will be using the course website http://www.cs.colostate.edu/~cs192 and Canvas for assignments in this class. Go to:

```
http://www.cs.colostate.edu/~cs192
```

Go to the Progress & Assignments tab, and you will find this assignment. It is titled, "Account Setup; Linux & Web Pages"

Go ahead and explore Mozilla Firefox browser further for a bit if you wish. When you are ready, move on to learn more about the GNU/Linux environment.

4 About the GNU/Linux Environment

4.1 GNU/Linux is fussy!

GNU/Linux is a very precise environment. Case matters on all commands and filenames. Always assume lower case. To "execute" a command, type the command followed by the **Enter** key.

In most instructions, a lower-case 'L' (l) may look exactly like a numeral one (1). Most commands are letters; in all seminar instructions, assume that this character is a lower-case "L" unless the instructions explicitly say otherwise.

Commands are generally in the form of: *`verb adjective[s] object[s]`*

The verb is the command and is required. The adjective[s] (modifiers) and object[s] may be optional. Let's take a look at some of them to see the patterns.

Computer programming wisdom says that you make the computer do as much of the work for you as possible. Ways to make the computer do more work will be introduced as we move along. We encourage you to use them!

4.2 First GNU/Linux Commands

Select your terminal window. You can find it by minimizing your browser, or by selecting the **Activities** menu. If you need to do so, start a new terminal window (same as before).

Tell the computer to tell you who it thinks you are by typing the command:

```
whoami
```

followed by the **Enter** key. The computer should print back your login name. Now that you know you and the computer agree about who you are, take a look at *where* the computer thinks you are.

GNU/Linux by default starts you in your *home directory.* A directory in GNU/Linux is essentially the same as a folder in a Windows environment. It can hold other directories or files, such as text files, documents, and so on. GNU/Linux maintains an extensive tree of lots of folders within folders, and, in general, folders for users are kept way down in the tree. To see where you presently are in the GNU/Linux structure, enter the command that will display the **p**resent **w**orking **d**irectory:

```
pwd
```

You will see a list of directory names separated by forward slashes; the last item listed should be the same as you login name. GNU/Linux knows about you as a user (someone who may log in) and has reserved a directory in your name, in which you may store your work.

To see what is in this directory already you need to **lis**t it by typing the command:

```
ls
```

You will probably see a file named `README`. You do not need to know what is in this file, but it gives you a chance to learn "more" – a useful command that will print out the contents of a text file. To see what is in this file, enter the command:

```
more README
```

The contents of this file will be displayed in your terminal window, screenful by screenful. Don't bother reading what it says unless you are really interested. Display the next screenful by striking the **spacebar** or **q**uit the command by typing a **q**.

Here is a chance to show you your first "shortcut." Start to enter the command again. Type "more R" and then type the **Tab** key. GNU/Linux will try to complete the command for you instead of making you type the "EADME". You will still need to type the **Enter**, though.

4.3 Creating New Files

You are likely accustomed to using a document processor, such as Microsoft Word, to create documents. GNU/Linux has document processors, also. But, since we are going to mostly be writing computer programs in our labs, instead we will use a program called a *text editor*. A text editor is similar to a document processor, but it creates files that are very simple text, with no special formatting or fonts. The text editor we will use for this class is a program called `gedit`. You can start `gedit` from a terminal window by typing:

```
gedit myfile &
```

This command means "run the program named `gedit` to work in a file named 'myfile' but let me continue to use this terminal window for other work." (The "&" is optional; it's that part that let's you continue to use the terminal for other work.) After you enter this command you will see some text in your terminal window showing the job was done, and a new window will pop up. Your cursor should be ready to go in the new window. Type something – maybe a short poem or a line of text – in the new window. When you are happy with what you typed, select **File** from the top of the `gedit` window and select **Save** from the pull down menu list that appears.

4.4 Copying Files

First make sure your new file is really there. Select your terminal window again, and enter the list command:

```
ls
```

You should now see a file named "myfile" in the list of files in your home directory. See what is in this file by entering:

```
more myfile
```

Does this look like what you typed?

Now let's make a copy of this file. Enter the command:

```
cp myfile mynewfile
```

The computer will not show you it did anything, but will simply display the command prompt again. Do another `ls` command to make sure the file is still there, and a `more` command to see what is in it again. Now in your `gedit` window select **File**, then **Close**. Still in the **File** menu select **Open File**. From the right hand box, select:

```
mynewfile
```

Now you can go change the file. Add another line, or change something about it. Then save the file again (**File** then **Save**). Then exit `gedit` (**File** then **Quit gedit**). Do a `more` command on your new file in your terminal window, and see if the changes you just made are there (try "more myn" and then the **Tab** key; GNU/Linux will finish the line for you!).

4.5 Removing Files

It is a good idea to clean up as you go along. The GNU/Linux `rm` command lets you remove files. Before you remove files, it is a good idea to use the `pwd` (present working directory) command and/or the `ls` (list) command to make sure you are really in the directory you think you are. Once you are sure you are where you want to be and that you no longer need these files enter the command:

```
rm myfile mynewfile
```

This command just quietly returns a prompt. Do one more `ls` command so you can make sure that the files are gone.

4.6 New Directories and Permissions

Now you are ready to build a basic, made-by-hand, web page. The CSU CS Department automatically looks for web pages in a text file named `index.html` in a directory named `public_html`. To make the new directory enter the command:

```
mkdir public_html
```

GNU/Linux is notoriously extremely liberal, by default. It requires that you explicitly set what it calls *permissions* so that everything works the way it is supposed to. We can explain this if you are interested; for now, just enter the command:

```
chmod 711 public_html
```

Do a long listing:

```
ls -l
```

You should see the following at the beginning of the line for your newly created `public_html` directory:

```
drwx--x--x
```

The ‘d’ means `public_html` is a directory, ‘rwx’ (111 in binary, 7 in octal and decimal) means that you as the owner can read, write, and search within the directory (worth 4, 2, and 1 points respectively); the next ‘–x’ (001 in binary, 1 in octal and decimal) means that members of your group can search within this directory, but cannot read or write in it; the next ‘–x’ means that all others on the system can search within the directory, but cannot read or write in it.

Now **c**hange **d**irectory so you are looking inside this new directory:

```
cd public_html
```

(you could also type “cd pu” then a **Tab** to have the computer type the rest!). Do a `pwd` (present working directory) to see where you are, and list the contents of this directory; it should be empty. **Stop and verify:** Are you really in the directory `public_html`?

5 Basic HTML Tutorial

Web pages are generally written in a simple markup language called HTML (for HyperText Markup Language). HTML uses *tags* (commands embedded in angle brackets) to describe page structures/semantics. Most commands have a start tag (a word or command enclosed in *angle brackets* `<html>`) and an end, which is a slash (/) in front of the same word or command. For instance, the first entry in the file you will use to make your web page will be `<html>` and the last entry will be `</html>`. Or, in this example, the word “bold” will be in boldfaced on the web page:

```
<b>bold</bold> type word!
```

Most, but not all, commands require end commands. For instance, to start a new line of text use `<br>` (break) or `<p>` (new paragraph) (HTML will not start new lines unless instructed to do so!).

5.1 Your First Web Page

Open a terminal window if you need one. From terminal window, enter:

```
gedit index.html &
```

In the new `gedit` window you are going to type exactly the following lines:

```
<html>
<head>
</head>
<body>
Hello World!  (or whatever you want, but be sure to include your name!)
</body>
</html>
```

Save this into the file `index.html` by selecting **Save** from the **File** menu.

Go back to you terminal (you may need to left-click in the window, or select it from the box at the bottom of the screen).

5.2 Setting Permissions for Web Files

One last bit of important information to know is that you must specially set the permissions for any file you want to use from a web browser. The command to do this for the `index.html` file is:

```
chmod 644 index.html
```

Do a long listing.

```
ls -l
```

You should see the following at the beginning of the line for your newly created `index.html`:

```
-rw-r--r--
```

The '-' at the beginning means this is a regular file, 'rw-' (110 in binary, 6 in octal and deciman) means that you as the owner can read and write this file (but not execute it, something programs would need to have set); the next 'r--' (100 in binary, 4 in octal and decimal) means that members of your group can read the file but cannot write it; the next 'r--' means that all others on the system can read this file but cannot write to it. The 'r' is important if anyone (except yourself) is going to be able to view your web page.

Now start up the Firefox browser, if it isn't already running. In the address bar type:

```
http://www.cs.colostate.edu/~your_login_name
```

then hit **Enter**.

(For instance, I type `http://www.cs.colostate.edu/~bartlett`) Your very first web page should appear. Go back to `gedit`. Modify or add text lines near “Hello!” (you may change it entirely). Save the file again and reload the page to see your changes.