

COMP11 Lab 0: Linux or Bust

Say you have a Mac, and your friend e-mails you a Word document that they wrote on their PC. You open it, only to discover that it's...a mess. The fonts are wrong, the formatting is off, the whole thing is a disaster.

Operating systems, amirite?!

In COMP11, you will be writing programs for every homework assignment. We want to make sure that you're writing your programs using the same operating system (and the same configuration of that operating system) that *we're* going to use to test your programs. That way, programs that look correct to you will look correct to us.

Learning Objectives

This lab is designed to give you practice with the following skills:

- How to log into a Halligan computer directly and remotely
- How to issue commands to a Linux operating system via the **Terminal** program
- How to navigate and manipulate the file system
- How to obtain starter code that we provide to you for labs and homeworks
- How to use Atom to edit files on your laptop
- How to sync the COMP11 files between your laptop and the Halligan computers

Reread these objectives after your lab. If, at that point, you don't feel like you could successfully use one or more of these skills, please attend office hours so you can get some in-person help from a TA or a Professor!

Leave the Graphics Behind (2 mins)

In COMP11, you will become familiar with the Linux operating system. All of the computers (we'll also call them "machines") in Halligan 116, 118, and 120 are running Linux. You can also log into our Linux machines remotely using either a Mac or a PC (we'll do this later in lab). You will use the Linux operating system to do all of your programming work for COMP11.

- Log into the lab computer using your CS login and CS password. Use [this link](#) if you need to set your password.

You'll find yourself looking at an unfamiliar but otherwise seemingly normal graphical desktop. You can move your mouse around, click on icons and menu bars - everything that you're used to doing.

Boring.

- Go to the **Applications** menu, choose **System Tools**, and then click on **Terminal**. (You may have to scroll down to find **Terminal**.)

The window you're now looking at allows you to control the computer using only **text-based commands**. These text-based commands can accomplish everything that can be done by moving and clicking a mouse,

and much more. They also allow you to control the system faster, as it is much easier to transmit a line of text than an entire pixel-by-pixel graphical display. Many programmers use **Terminal** to do *everything* on their machine! But it does take some getting used to.

Welcome to your new home.

But where are we? What are we doing here? To start, we'll walk you through some of the most common and useful commands and tricks that you'll be using during the semester. You should take notes on these commands so you can quickly reference them as you work on your homeworks and future labs.

Sanity Check (15 mins)

Let's take a minute to prove that we can really control the machine with text-based commands, and do some of the basic things that you might typically do with a mouse. Specifically, we're going to move around the file system a bit and create/remove a folder (known in Linux as a "directory").

- ▶ Type "**pwd**" and hit enter. **pwd** will list the current directory that you are in.
- ▶ Type "**ls**" and hit enter. **ls** will list all the folders and files in the directory that you are in.

You'll notice that your current directory is `/h/{CS_login}` (where `{CS_login}` is your personal CS login). When you log into a Halligan machine and open **Terminal**, or log into one of our Linux machines remotely, you will be immediately taken to your own personal space on our servers - your "home directory" (this is what the 'h' stands for). This is where you will keep all of your C++ files for COMP11.

You probably also noticed that one of the directories listed in your home directory is **Desktop**. This indeed refers to the actual, graphical desktop that is sitting behind your **Terminal** window. Let's run some text-based commands to interact with this directory:

- ▶ Type "**cd Desktop**" and hit enter. **cd {directory_name}** will take you into the specified directory (**cd** is short for "change directory").
- ▶ Enter **pwd** again. You should now be in `/h/{CS_login}/Desktop`.
- ▶ Enter **ls** again. The contents listed should match what you currently see on your graphical desktop.
- ▶ Type "**mkdir sanitycheck**" and hit enter. **mkdir** creates a new directory with the specified name.
- ▶ Enter **ls**. The new directory that you created, **sanitycheck**, should now appear in the **Terminal** directory listing. A new folder entitled **sanitycheck** should have also appeared on your graphical desktop. Voilà!
- ▶ Type "**mv sanitycheck sanitycheck.1**" and hit enter. **mv** renames a file or directory. You can confirm that this command was successful both by entering **ls** and checking the updated folder name on the graphical desktop.
- ▶ Using your mouse, right-click the **sanitycheck.1** folder on your desktop and delete it.
- ▶ Enter **ls** in **Terminal**. No more **sanitycheck.1**.
- ▶ Using your mouse, right-click your desktop and create a new folder called **sanitycheck.2**.
- ▶ In **Terminal**, enter **ls**. **sanitycheck.2** should now be listed.
- ▶ Type "**rm -r sanitycheck.2**" and hit enter. **rm** deletes a file. The **-r**, which is known as a "flag", allows you to delete a directory by indicating that it is okay to also delete everything within that directory. (Fun fact: the **rm** command is a permanent decision! Use it carefully.)
- ▶ Type "**cd ..**" and hit enter. The **..** moves you up a directory. You should now be back in `/h/{CS_login}`, which you can confirm with **pwd**.

Hopefully you're feeling convinced that you can navigate around the system using these text-based commands. Let's do something more interesting.

Running COMP11 Courseware (10 mins)

Throughout the semester, you'll be asked to run many programs that were written by our course staff. To see a list of these programs, type:

```
► ls /comp/11/bin
```

First we're going to have you run the `welcome` program. To do this, run the following two commands:

- `use comp11` (this command gives you access to all the scripts in `/comp/11/bin` for the duration of your `Terminal` session; follow the link after you've completed the lab to see how to have this command run automatically)
- `welcome`

I know, groundbreaking! Want to run it again? Just **hit the up arrow on your keyboard**. `Terminal` will retrieve the last command you typed. The more times that you hit the up arrow, the further back `Terminal` will go to retrieve previous commands. Want to run it a third time? Type "`welc`" and **hit the tab key**. `Terminal` will attempt to auto-complete what you're trying to type.

↑↑↑ These are two of `Terminal`'s most critical shortcuts! Remember them! ↑↑↑

One program of ours that you'll be running throughout the semester, `pull-code11`, will copy code from our private directories into the directory of your choosing. Run the following series of commands:

- `cd` (this command will return you to your home directory no matter where you are)
- `mkdir comp11` (now you have a directory where you can keep all of your comp11 work)
- `cd comp11` (change your current directory to the one you just made)
- `pull-code11 lab00`
- `ls` (there should now be a directory called `lab00`)
- `cd lab00` (change your current directory to `lab00`, which is inside of the directory `comp11`)
- `ls`

As you'll see, the file `welcome.cpp` is now in your possession. This is the source code of the exhilarating program you just ran. We'll show you two ways to look at it's contents. First, to take a quick peek, you can print it's contents out in `Terminal` with the following command:

```
► cat welcome.cpp
```

Most of this likely looks like gibberish, but you should be able to spot some of the text you saw when you ran the program. Next, you'll open the code with the editor we'll be using throughout the semester, `Atom`. Enter the following:

```
► atom welcome.cpp &
```

Atom is a text editor specifically designed for writing code, just like Microsoft Word is a text editor designed for writing tortured poetry. In the same way that Microsoft Word will try to draw your attention to potential misspellings and bad grammar, **Atom** has a lot of visual features that makes coding easier.

For now you can close **Atom** on the lab computer and fire up your personal laptop (if you brought one).

Get Synced (45 mins)

Halligan is great, and you can always use the lab computers to work on your assignments, but you'll also want to work on assignments from your own computer. Thankfully you can remotely log into your Halligan account from any Mac or PC with an internet connection. If you're a Mac user, this is done using the Mac version of **Terminal** (found in Applications/Utilities/Terminal). PC users can use the program **puTTY**, which can be downloaded for free.

Mac users, open your **Terminal** program and connect to Halligan using the following command:

```
► ssh -X {CS_login}@homework.cs.tufts.edu
```

PC users, download and install **puTTY**. You can log into Halligan using hostname **homework.cs.tufts.edu** and port 22. Click "yes" if you get a security alert and then finish logging in with your CS login and password.

You have now created a connection over the internet between your laptop and your Halligan file system. Cool! You'll be able to navigate through your Halligan space using the same commands that you practiced above. For example, the following command will take you right back to where we left off on the lab computer:

```
► cd comp11/lab00 (the '/' indicates that the lab00 directory is within the comp11 directory)
```

The only thing that you cannot do via your remote login to Halligan is to write or edit code by launching **Atom** in a separate window. For this, you will need to download **Atom** onto your personal computer and configure its **remote-sync-pro** feature. This will allow you to link a folder on your personal computer with a folder on Halligan. You'll be able to edit files in your personal computer, and save the changes both to your personal computer and to Halligan.

To install **Atom** and set up **remote-sync-pro** please follow the instructions on our course tech guide:

```
► https://www.cs.tufts.edu/comp/11/docs/techguide.html#sec-atom
```

If you follow these instructions all the way through to "Remote Sync PRO -> Download Folder", the **lab00** folder and **welcome.cpp** code from your Halligan space should now exist in your laptop's **comp11** folder. You should be able to save a change to **welcome.cpp** on your laptop and see it reflected on Halligan. Let's try it.

The **welcome.cpp** program ends by printing "Welcome to COMP11!" on line 24 (ask a TA for help getting line numbers turned on in Atom if needed). On your laptop, use **Atom** to change that message to something else - anything else. Save it. You should see some elaborate messaging at the bottom of your **Atom** window as the syncing happens. Finally, use **Terminal** (either on your laptop or on the lab computer) to check the updated contents of **welcome.cpp**:

```
► cat welcome.cpp
```

The change you just made should appear in the outputted source code.

Submit your work

To get credit for this lab, you will submit your altered version of `welcome.cpp` via our [submit11](#) system. Enter the following command into your **Terminal** window:

```
► submit11-lab00
```

Linux Summary

This lab introduced the following Linux commands, which you will need throughout the semester:

`pwd`, `ls`, `cd`, `mkdir`, `mv`, `rm`, and `cat`

You can find detailed descriptions and examples of these commands, and many more, online. In fact, Linux provides it's own manual for all its built-in commands. You can access it via the `man` command. For example:

```
► man rm
```

will open the Linux usage manual for the `rm` command (you can quit the manual by typing “q”).