APS 105 — Computer Fundamentals

Lab 0: Introduction to Your TA, Linux and CodeLite

Winter 2021

This goal of this first laboratory is to set the stage for the many labs to come in this course. Your first objective is to "meet" your TA for the year, and get to know him/her by name, and to give that TA a sense of your background knowledge in programming. (You will be asked in the lectures what your TA's name is). The second objective is to learn how to use a Linux-based computer using text entry, as opposed to the mouse-based interaction that you're used to. The final objective is to learn the basics of creating, compiling and running a simple program with CodeLite, the software tool we'll be using throughout this course.

The labs in 2021 in APS105 will be "online". What this means is:

- You will have an assigned lab period and a lab "room, but you do not have to appear in that room. The room and computing facilities are available to you at that time, should you need them. But as you will see, it is possible and practical to do your work at a computer convenient to you.
- Similarly, you will be assigned a TA. Your TA will be available to you online for one hour a week, the timing of which will be arranged between you, the others handled by the TA and the TA. The "default" time will be the time of your lab assignment, but you, the other students and the TA may find other times more convenient.
- A piazza discussion group and a tutorial time will also be available to you for asking questions and seeking answers from TAs, instructors and your peers.

Preparation

Every lab you'll be doing in this course will require significant amounts of preparation - in general, you'd like to have most of your program working as well as you can prior to the time your TA is available. For this lab, do the following preparation:

These instructions may assume that you might have access to the ECF computing facilities in SF. At the time of writing (Dec 2020), this is not possible, and you will have to do a home installation. Perhaps sometime later in the term these labs could open up.

1. Read all of this document, paying particular attention to **Getting Started with Linux and ECF**, as that is part of what you'll be doing in this lab. If you don't have your own computer, you also can use the University computers located in the Engineering Computing Facility (ECF) Labs, in the Sandford Fleming building, room 1012, 1013 or 1106.

2. If you do have your own computer and you wish to install CodeLite[1] on it, you will find a second document titled **Installing & Using the APS 105 Virtual Machine** that in the same place on Blackboard that you found this document. On your own home computer or laptop, you can install *either* CodeLite

[1] Codelite is the development software used in this course.

as an IDE (a programming environment) *or* install "The APS 105 Virtual Machine" software as described in the document. While the former will install *just* CodeLite as an IDE, the latter will set up a Linux environment and the CodeLite software that is identical to that in our labs. You can do both, but if you have one, you won't really need the other. One important thing: *do not attempt to install either of these on the lab computers*.

In the Lab Session with the TA

1. You will get information about your specific TA. This TA should contact you before the lab (check your email, including "junk" to be sure). The TA will "visit" you one at a time in your session to get acquainted and get some information from you to help manage the labs.
2. To start, do the work described on the next page under the title **GettingStartedwithLinuxandECF** to the end. Ask your TA about anything you don't understand.
3. Read and do the exercises in the document **GettingStartedwiththeCodeLiteEnvironment**, which is available in the same place on the Quercus portal as this document, making sure you can do this on the University ECF Linux computers or on a virtual link.
4. Once you're comfortable with this material, ask your TA to be graded - we want to check that you have understood these basics. Good luck!

Getting Started with Linux and ECF

---

1 Introduction

Welcome! This part of Lab 0 will teach you how to use the computer system you'll be using during labs for APS 105F, Computer Fundamentals. In the labs if you use them, and to submit your code, we will be using a *Linux*-based computer, which is somewhat different from a Windows or a Macintosh-based system that you may be familiar with. If you have never used a Linux system before, it is essential that you read and learn this information - the marking and submission of all of the assignments of this course *must* be done through the ECF Linux system, and you'll need to know how to use Linux to do that. Regardless, Linux is a good system to be at least a bit familiar with if you use computers at all, and essential to those that want to continue on in a career in software with the faculty.

2 Before You Start: Finding Our Computers and Obtaining “Login” ID

To try out Linux you can create your own environment (see the InstallingUsingVM document) or link to a machine over the internet or using one of the Engineering Computing Facility (ECF) Linux computers. Our discussion will be about using these ECF computers, but the most of the information is applicable to all situations.

The ECF machines are mainly located in The Sandford Fleming (SF) building Rooms 1012 and 1013.

The first thing you must do when you begin your work on the computer is to “log in” to let the system know that it is you who is using the system. Your ECF login name will be the same as your ’utorid’ that you were given (hopefully!) last term. If you have not yet logged into an ECF machine go to this link to use your utorid and password to activate your ECF account and password.

https://ssl.ecf.utoronto.ca/ecf/weblogin/services/agree

If for some reason this doesn’t work you can also reset your password when you are at an ECF Linux computer: in the username field, type account (all lower case). This will bring up a window in which you can select ‘I do not know my ECF password, and/or I want to reset it.’ This will take you to it. Note that, if you change your password, it can take up to 20 minutes to actually take effect.

3 Begin: First Login

With your ECF login name and password, go ahead and login to the Linux computer. After logging in, you will be presented with a sequence of windows. After reading each one, click on the okay button in the lower left corner of the window to make it go away (or just wait and it will eventually disappear on its own). Linux will then start up the *X Window System*.

4 Next: Learn about the Linux Terminal

One of the things that takes getting used to in a Linux computer is the ‘terminal’ (sometimes called the command line) through you’ll do operations that you might be more used to doing with a mouse. While it might not seem like it, many professional software engineers prefer to interact with their computers using this kind of typing interface. The terminal is a powerful tool for navigating the filesystem, manipulating files, and creating and running programs. While it can do all the actions that you normally perform using a mouse and the graphical user interface (GUI), it can also do much more, and an experienced user can do some things much faster! The rest of this document will show you the basics of how to use the terminal. This is absolutely necessary for you to learn, as you will need to use the terminal to test if your lab assignments are working correctly and to electronically submit them for marking.

To open a new terminal window, right-click on any empty portion of the linux desktop (i.e., right on the wallpaper and not on any icons or menu items) and select the “Open Terminal” option.

Leave the terminal open for now and continue reading.

5 The Linux filesystem and Paths

Data and programs that you will be using in this course are stored as files on a disk. There are many types of files: some are software programs that can be executed, other files contain text that can be read, while other files contain images that can be displayed on the screen. Two things that all files have in common are that they can be stored, and that they all have a name.

As we add more files to computer storage the number of files grows to the point where it is hard to keep track of them all. For this purpose, Linux files are organized into *directories*, which are the same as folders in Microsoft Windows and Mac OS.

All of the files and directories on the entire Linux system, if made into a picture, would look like a tree as shown in the diagram below. At the top is the *root directory*. It is represented by a slash character (/), which is its name. The root directory may contain files, and it will also contain other directories. The files and directories within the root directory are shown as items below (and connected) to the root directory. These directories may have files and other directories contained within them and these are shown as nodes below and attached to the parent directory. This pattern can be repeated many times.

In Figure 3, chen is a *subdirectory* of prof, and u is the *parent* directory of prof.

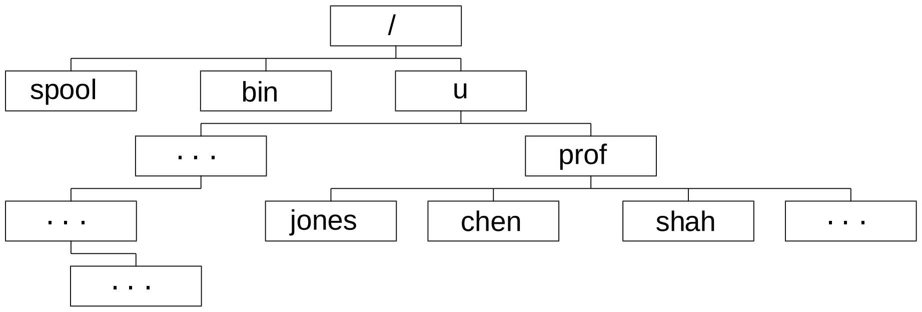


Figure 1: An example filesystem structure.

5.1 Paths

A **path** is like an address to a specific file or directory within a filesystem. It tells one how to navigate, starting from the root of the filesystem, to a specific file or directory. Continuing with the example shown in Figure 3, the path to the directory “chen” would be /u/prof/chen.

6 Navigating the filesystem using the terminal

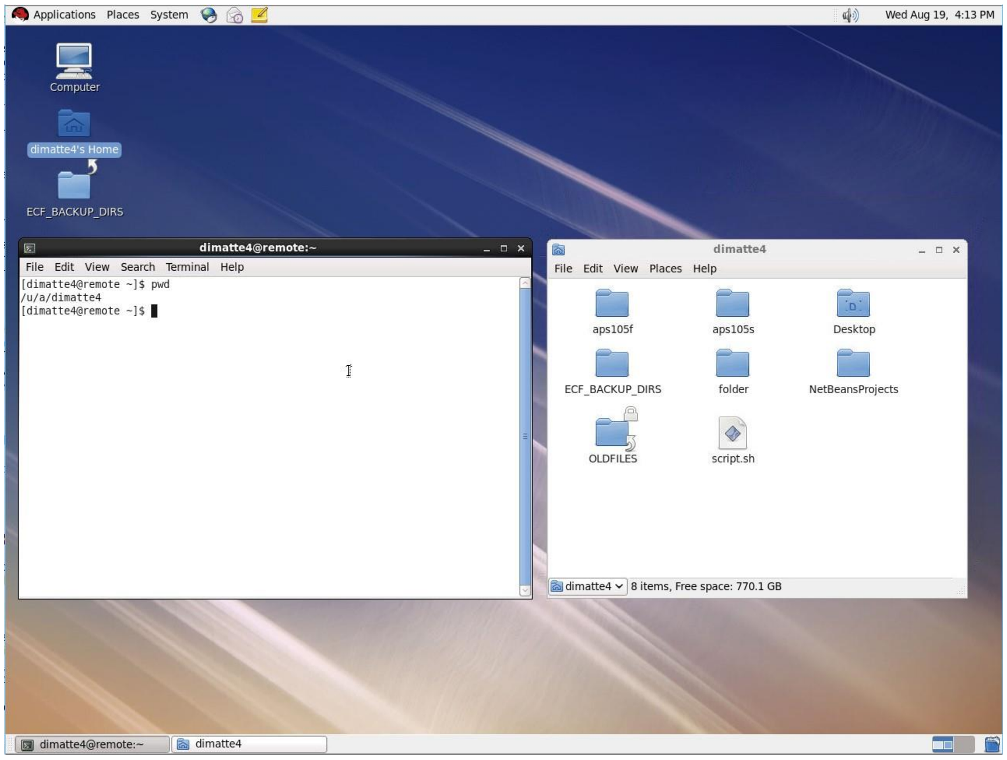
You already have experience with navigating through the filesystem of your own computer by clicking through folders in the graphical user interface (GUI). In a Linux computer you can also do this graphically or by using the text-based terminal. In this section we show you how to do this with a terminal. Return to your terminal window and follow along with the sections below.

6.1 Printing your Working Directory - the ‘pwd’ command

When you want to work with files in the terminal, you generally have immediate access to the files in one directory only. This directory is called the *working directory*. To find out what the present working directory is, type in the following command (press the Enter key after the command to have the terminal execute it): pwd

Linux will respond with the path to your working directory. The default working directory when you first open a terminal is your “home directory”, and has the same name as your ECF username. This is also the same folder that you can open graphically using the shortcut on the desktop called “hyour ECF username herei’s Home” (the folder icon has a house on it). Open that folder now and move it to the right of your terminal window, so that you can graphically see the changes to the filesystem as you interact with it using the terminal. See Figure 2 for an example of this, being done by a user with the username dimatte4.

Now that you know the path to your home directory from the pwd command, close the folder window (not the terminal) and try navigating to it again without using the desktop shortcut. To do so, we have to start at the root of the filesystem. To get to the root directory of the Linux filesystem



```
[dimatte4@remote ~]$ pwd
/u/a/dimatte4
[dimatte4@remote ~]$
```

Figure 2: An example of the pwd command in a user’s home directory, with the home directory visible in a separate window.

graphically, open the “Computer” icon on the desktop, and then open “Filesystem” icon in the window that appears. The window that is opened should be titled “/”. Recall from Section 5 that this is the root of the filesystem. See Figure 3 for an image of what the root directory should look like in a graphical window.

Now that you have access to the root folder, try navigating through the folders to your home folder by traversing the folders in the order that they appear in the path reported by the pwd command. For example, if you were the user dimatte4 that is shown in the example in Figure 2, from the root directory you would open the directory u, then a, and then finally dimatte4. Note that the contents of some of these folders take a while to appear in the graphical windows because they contain directories for thousands of users (one reason why using the terminal is better).

Once you navigate back to your home directory, close all windows other than your terminal and home directory, and arrange them side-by-side again and continue following along with this guide.

6.2 Listing files and directories - the 'ls' command

The ls command is used to **list** the files and directories within your working directory. To see the contents of your working directory, type in the following command (and follow it with the enter/return key, as you will with all text commands):

```
ls
```

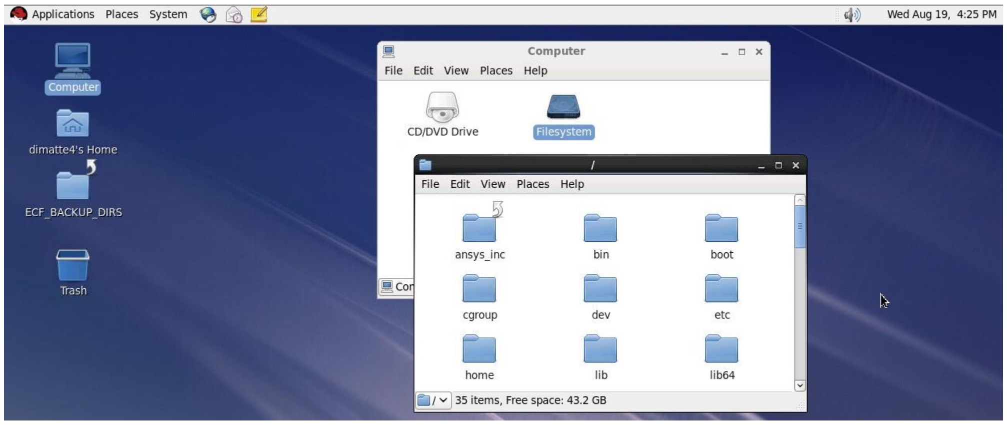
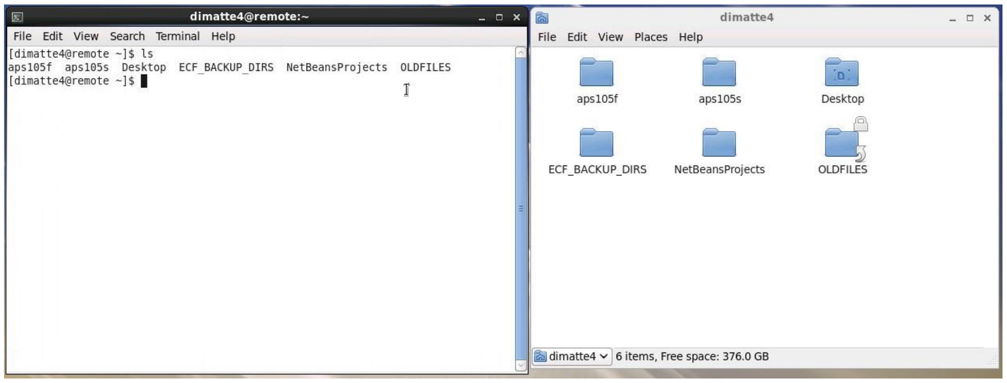


Figure 3: An graphical window showing the root of the Linux filesystem.

Linux will respond with the names of all the files and directories contained in the working directory. Since we are still in our home directory, the command will show us what is in our home directory. Note that the graphical home folder contains the same files and directories that the ls command reported! See Figure 4 for an example.



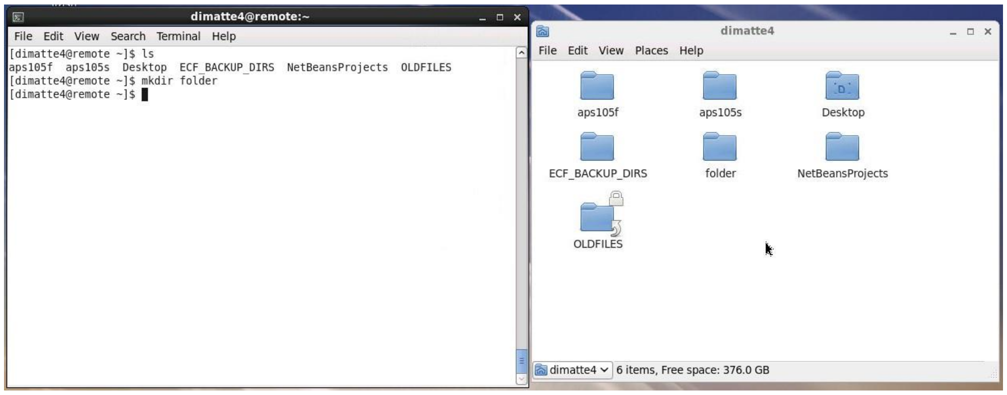
```
[dimatte4@remote ~]$ ls
aps105f  aps105s  Desktop  ECF_BACKUP_DIRS  NetBeansProjects  OLDFILES
[dimatte4@remote ~]$ 
```

Figure 4: An example of the ls command in a user's home directory.

6.3 Making Directories - the 'mkdir' command

The mkdir command is used to create a new directory. The new directory will be created within your working directory. This command is unlike the previous commands because it requires you to also give it some input, called an **argument**. The argument is the name of the directory you would like to create. For example, to create a directorty named folder, type the following into the terminal (notice how the argument follows the name of the command): mkdir folder

Figure 5 shows an example of making a directory called folder, with a graphical window open showing it created in the user's home directory. Perform another ls command to check to see if the new directory shows up in the terminal.



```
[dimatte4@remote ~]$ ls
aps105f  aps105s  Desktop  ECF_BACKUP_DIRS  NetBeansProjects  OLDFILES
[dimatte4@remote ~]$ mkdir folder
[dimatte4@remote ~]$ 
```

Figure 5: An example of the mkdir command in a user's home directory.

6.4 Change to a different directory - the 'cd' command

The cd command is used to change your working directory. The argument to cd is the name of the directory (inside your current working directory) that you would like to change to. For example, to change directories into the directory you created in the last step, use the following command:

```
cd folder
```

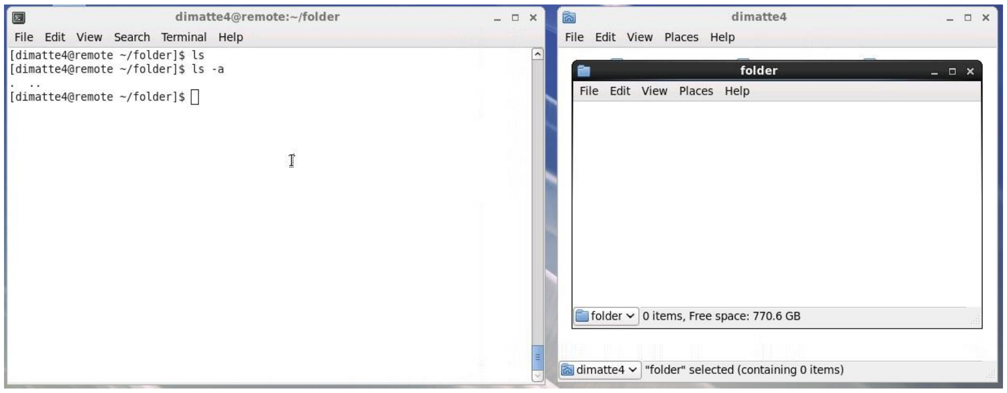
The command does not give any output if it executes properly. You'll know that it worked properly since your working directory will have changed. Execute a pwd command to print your working directory to the terminal and confirm that it has changed. You should notice how the path reported by pwd is the path to your home directory, followed by "/folder" - this is because the director folder is a subdirectory of your home directory. Stay in this directory for the following step.

6.5 Showing hidden files using the 'ls' command

Inside the directory folder, perform an ls command to show what's inside it. It should be empty, since we just created it (confirm this). In fact, it's not empty, but it contains two special hidden directories.
In order to show hidden files and directories - their names begin with a dot (.) - you must supply an option to the 'ls' command. Many commands take options, and these options change the behaviour of the command. To show hidden files, type in the following command (note the option is supplied with a dash):

```
ls -a
```

See Figure 6 for an example of this, shown for a user with the username dimatte4. Notice how two items appear, called (.) and (..), that don't appear in the GUI when you open folder. This is because these are hidden files. Read on to learn more about these two special hidden directories.


dimatte4@remote:~/folder
File Edit View Search Terminal Help
dimatte4
File Edit View Places Help
folder
File Edit View Places Help
folder
dimatte4

```
[dimatte4@remote ~/folder]$ ls
[dimatte4@remote ~/folder]$ ls -a
.  ..
[dimatte4@remote ~/folder]$ 
```

0 items, Free space: 770.6 GB

"folder" selected (containing 0 items)

Figure 6: An example of the ls -a command in a user's home directory.

6.6 The directories '.' and '..'

In Linux, **all directories** contain two hidden directories called (.) and (..). They are both links, like an "alias" in Mac OS or a "shortcut" in Windows. The (.) directory is a link to the current working directory. This is useful as a shortcut for typing out the path to the working directory, as you can simply substitute it with a (.). This will come in handy later.
The (..) directory is a link to the parent directory of the current working directory (i.e., one level up the tree of the filesystem). So, if we are still inside our folder directory, we can navigate back to our home directory by executing the following command:

```
cd ..
```

since the parent directory of folder is our home directory. Try this, and then confirm that you are back in your home directory by executing a pwd command.

6.7 Removing directories - the 'rmdir' command

The rmdir command is used to remove an empty directory (i.e., a directory that contains not other directories or files). To delete the folder that we created, type in the following command: rmdir folder

Use the ls command to confirm that the directory no longer exists. You should also see that it no longer exists in your graphical window either.

6.8 Your home directory - the '~' shortcut

There's one last shortcut that you should know - the shortcut to your home directory. Your home directory can be referred to by the tilde ~ character. For example, if you want to change directories to your home directory, regardless of what your current working directory is, you can simply do:

```
cd ~
```

This shortcut will also come in handy later when you want to type in long paths that begin with your home directory.

7 Commands to manipulate files

This section builds upon what you learnt in Section 6, allowing you to do some basic operations on files.

7.1 Copying files - the 'cp' command

The 'cp' command is used to copy files. Two arguments must be supplied: the first is the path to the file that you wish to copy, and the second is the path to the file you wish to create. The file will be named

whatever name you specify at the end of the second path. If you do not specify a file name at the end of the path, but just a directory, it will automatically be named with the same name as the original file.
For example, you can copy the C program hello.c, which is located in the directory /share/copy/aps105s, to your home directory with the following command (try it):

```
cp /share/copy/aps105s/hello.c ~
```

Note that we used the tilde shortcut here for supplying the path to our home directory. Perform an ls within your home directory to confirm that the copy exists, and also check in the GUI window. If you wanted to copy the same file (into your home directory), but name the copy goodbye.c, you would use the following command:

```
cp /share/copy/aps105s/hello.c ~/goodbye.c
```

7.2 Moving files and directories - the 'mv' command

The mv command is used to move files. Two arguments must be supplied: the first is the path to the file that you wish to move, and the second is the path to the directory you wish to move it into. Try moving the file hello.c from your home directory into your Desktop by using the following command:

```
mv ~/hello.c ~/Desktop
```

The directory called Desktop within your home directory is the actual graphical desktop of the Linux machine, so you should see the file hello.c now appear in the Desktop.

7.3 Renaming files and folders with 'mv'

The mv command can also be used to move files. In the previous step, the second argument to the command was a path to a directory. If, instead, we supply a path to a folder and then append a file name to the end of that path, the original file will be moved to that location and it's name will be changed to whatever you specified. So, for example, if we wanted to rename the file hello.c (that now exists on the Desktop), we can use the following command to rename it to hello2.c:

```
mv ~/Desktop/hello.c ~/Desktop/hello2.c
```

7.4 Removing (deleting) Files with the rm command

The rm command is used to remove (i.e., delete) files. The single argument to rm is the path to the file that you wish to delete. To delete the hello2.c file on your Desktop (created in the previous step), you would issue the following command: rm ~/Desktop/hello2.c

8 Summary of Linux Commands

Command	Meaning
pwd	display the path of the current working directory
ls	list files and directories in the working directory
ls -a	same as above, plus listing hidden
mkdir hnamei	create a directory called hnamei
rmdir hnamei	delete a directory called hnamei
cd hdiri	change working directory to hdiri
cd ~	change working directory to home directory
cd ..	change working directory to parent directory
cp hfile1i hfile2i	copy hfile1i and name it hfile2i
mv hfile1i hfile2i	move or rename hfile1i to hfile2i
rm hfilei	delete hfile1i

Table 1: Summary of useful Linux commands.

9 Logging Out

To log out, use the *Actions* menu on the top, and then choose Log out.

Never forget to log off when you are done! If you do not log out, someone will be able to get unauthorized access to your programs and other files after you leave the terminal. Once in your account, they can easily copy your solutions to another computer.

10 Next Steps: Learning the CodeLite Environment

Once you are comfortable with using Linux, it is time to move on to learning the software we'll use to enter, compile, run and debug your programs: CodeLite. See the next document: **Getting Started with the CodeLite Environment**.