

## Introduction

In this first lab you will be introduced to the computing environment in the Baldy 21 lab. If you are familiar with Unix or Linux you may know how to do some or all of the following tasks. **Do them all the same.** If you are new to Linux and Unix, these exercises will get you up and running with most of the tools you need to successfully complete the lab assignments of this course. Some further tools will be introduced as we need them in later labs.

Read through the entire lab before you start working, so that you know what to expect.

---

## Note

If you do not have your UB Card or a UBIT account yet, visit the UB Card office in the UB Commons, room 211 on the North Campus, or 104 Harriman Hall on the South Campus. To find out your UBIT username, you can either visit [www.buffalo.edu](http://www.buffalo.edu) and enter your name in the "FIND PEOPLE" search box, or take your UB Card to a UB Card Swipe Station to find your UBIT name and initial password; these are located in the Capen Library (first floor Cybrary), the Lockwood Library (2nd floor Cybrary), the Music Library, and UB Micro (in UB Commons). Your initial password should be the first four digits of your birthday (MMDD) followed by the last six digits of your UB card number (not your person number).

---

## Lab Tasks

You must now carry out each of the following tasks during the recitation. If you run into problems, put up your hand and ask your TA for help.

### Logging in - Connecting to the system

Your first lab task is to log in to your CSE computer account. Your CSE account is valid on the computers in Baldy 21. This account is separate from your UBUx account.

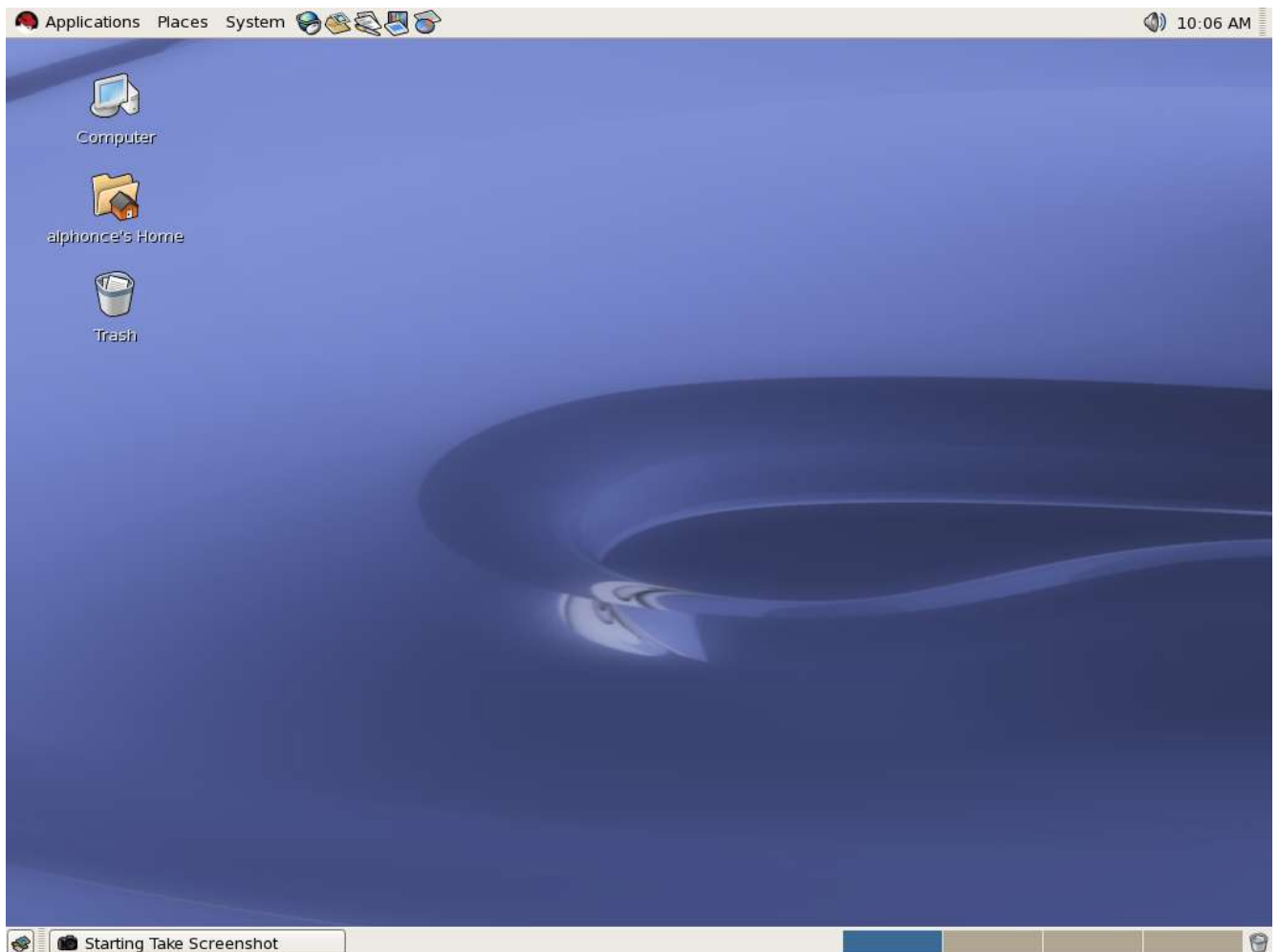
- If you have not had a CSE account previously, your login name is the same as your UBIT name. Your password is the same as your UBIT password was at the time when your CSE account was created.
- If you have previously had a CSE account (e.g. if you took any other CSE course last year where you received an account), your login name and password should not have changed.

Linux (a Unix variant) is the name of the *operating system* that is running on the machines in the Baldy 21 lab. An operating system (or OS) provides basic services to both programs (applications) running on the machine and to users who are working on the machine. Examples of services provided by an OS are a facility to read the contents of a file from a disk, or run multiple programs at one time (multi-task).

### To log in:

- You should see a welcome screen with a place for your username. Enter your username (UBIT name), and then press OK.
- You will then be asked to enter your password. Enter your password, and then press OK.

When the login process is finished, you will be presented with a desktop that looks something like this (your background may be slightly different):



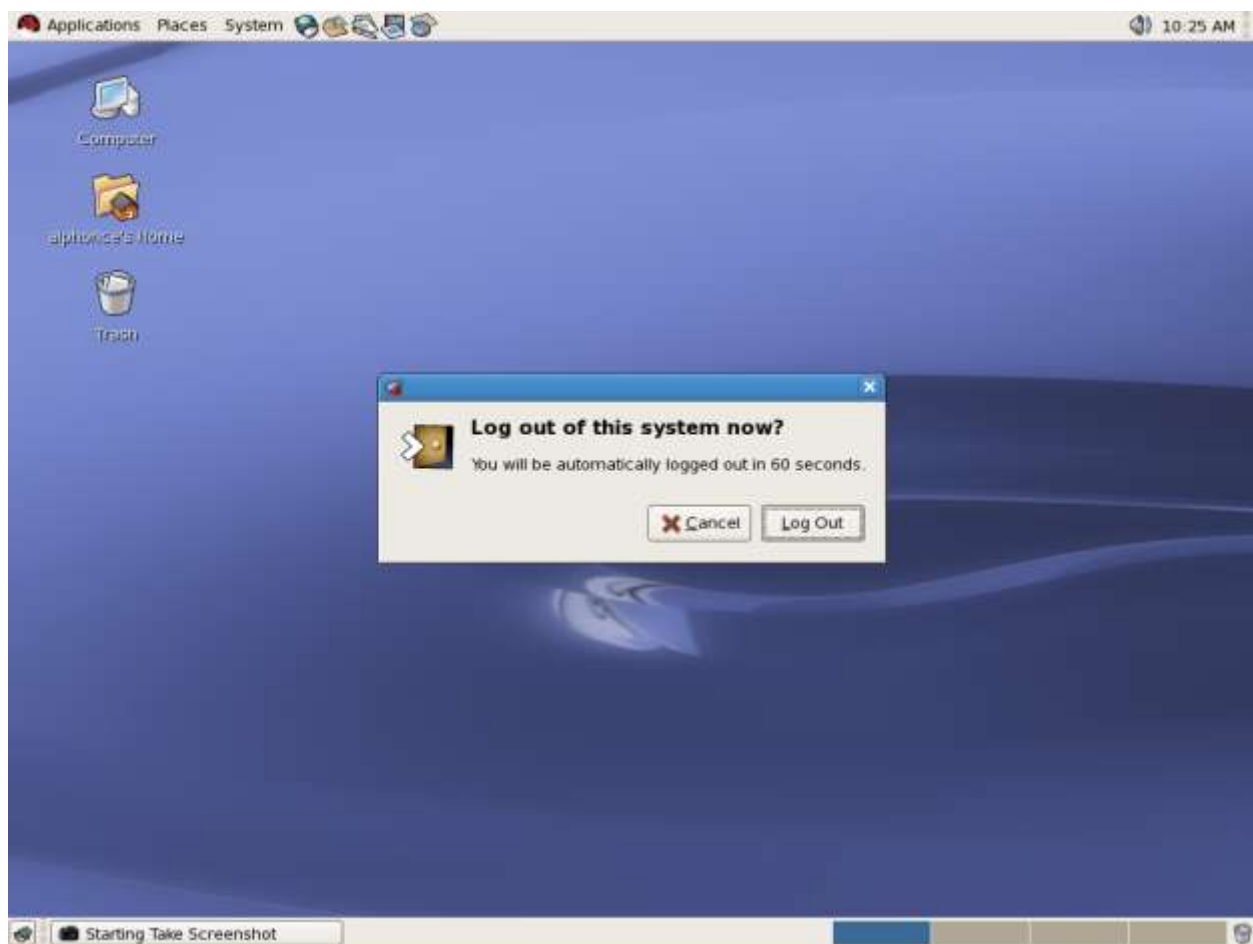
There is a panel across the top of the screen which has several menus on it. This is

similar to the Start Menu feature of Microsoft Windows. There is also a panel along the bottom of the screen. If you want to know what anything does, hover your mouse over it and a little description will appear.

In case of emergency where you suddenly need to leave, such as your grandmother dying (it happens a lot in this class), let's discuss how to log out of the system before we go any further.

- Just click on the "System" on the panel at the top of the screen, then select "Log Out".

You will see a pop-up window asking if you're sure you want to log out:



Click on "Log Out" and you will be logged out. **Remember to never leave the lab without logging out, even if it is "just for a minute"!**




Spend a moment getting comfortable. First, double click on 'Computer' to



to



To move a window with a title bar, click and hold the left mouse button in the middle of the title bar and drag it to a new position. If you click on the small buttons in the title bars of the windows, different things will happen:

- The rightmost button on the title bar:  Left-clicking this will close the window. Try it: open a window and close it.
- The button to the left of that (in the middle of the three):  Clicking this will toggle the window between maximized and unmaximized modes. In maximized mode, the window will take up the full space of the screen. In normal mode, the window is not necessarily the full size of the screen and you have the ability to move it around and see other things on the screen as well. Try maximizing and unmaximizing some window.
- The button to the left of that:  Left-clicking on this will minimize the window. It keeps the program running but hides it from view. Now we need to get it back. Just click on the corresponding item on the panel along the bottom of the screen.
- The button on the far left of the title bar (this button may look different depending on the window opened):

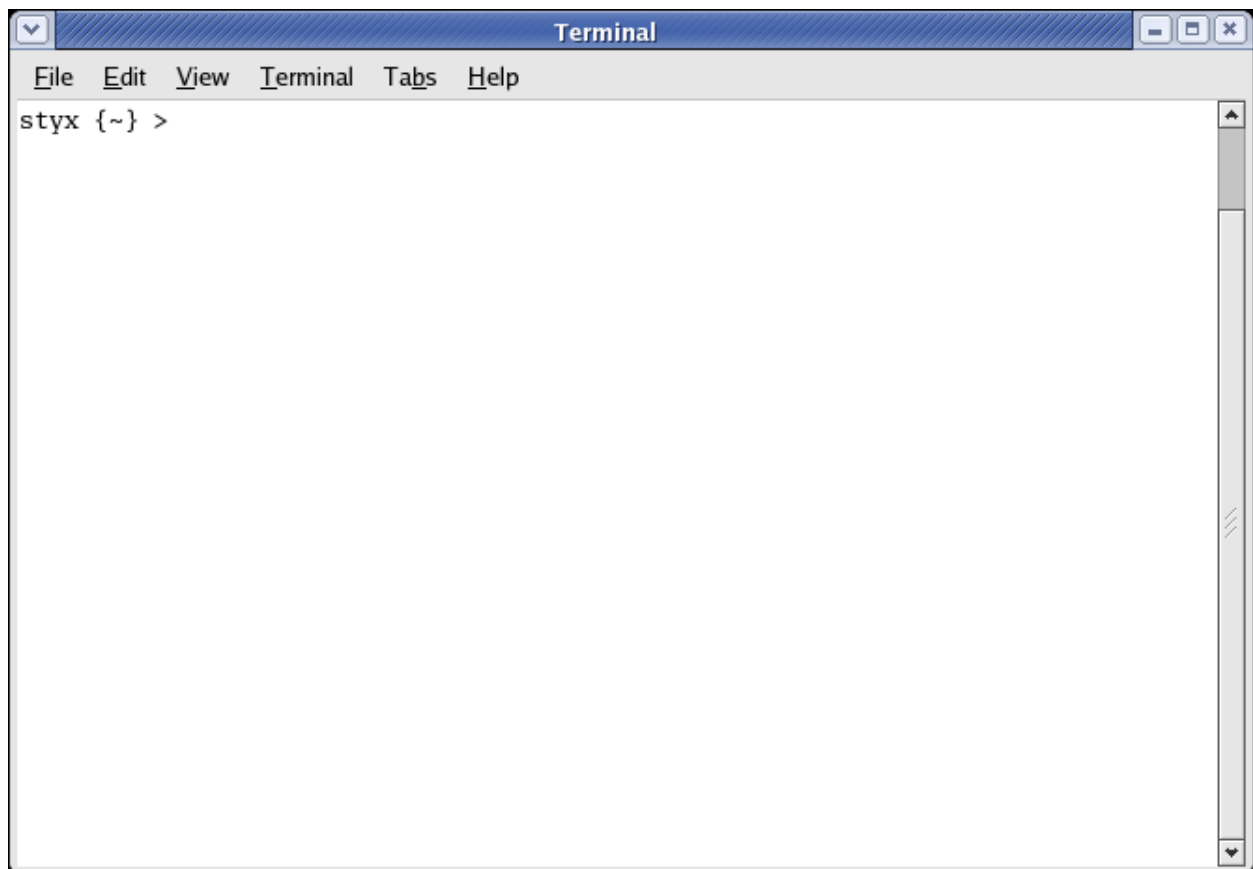


Clicking on this will bring up a menu of different things you can do to the window. Try it out and see what can be done.

To interact directly with Linux, we need what is called a *terminal*. Here are two ways to bring up a terminal:

1. Right click on an empty space on your desktop and select "Open Terminal" from the menu.
2. Click on the Applications menu on the panel at the top of the screen. Click on "System Tools", then click on "Terminal".

The terminal window will look like something this:



The text "styx{~} > " is called a prompt (yours might look a little different). In this example the prompt includes the name of the computer which you are logged in to (styx). You will type commands at the prompt, after the ">" symbol, to get the operating system to perform various tasks for you.

The files which you create are stored on a machine called a *file server*, which is accessible from all of the workstations in the lab. In other words, it doesn't matter which workstation you log in at, you will be able to access exactly the same files. The computer which you will use in the Baldy 21 lab is called styx.cse.buffalo.edu, or just

styx for short. This computer shares its files with several systems, one of which is called timberlake.cse.buffalo.edu (timberlake for short), which you can log into from other places around campus or from your house or dorm or apartment. Remote connections should always go through timberlake.

### Changing your Password

It is a good idea to change your password from your default password to something else. You should change your passwords often to help keep your account secure. You must never share your account password with anyone (including your friends and classmates). You can choose to change your password now or come back to these instructions and change it at a later time.

To change your password on the CSE systems, see the "Password" section of this page of the on-line CSE help pages:

<https://wiki.cse.buffalo.edu/services/content/cse-unixlinux-accounts>

Note that you must log in to hadar.cse.buffalo.edu to change your password. Your TA will tell you how to do this. Also keep in mind that because your CSE account is separate from your UBUlinux account, changing the password on one account does not automatically change the password on the other. You should change passwords on both accounts on a regular basis.

### E-mail

Since you have a UBIT account you also have an e-mail address *@buffalo.edu*. You can read about the e-mail system at <http://ubit.buffalo.edu/ubmail/>. To make sure that you can send and receive e-mail, send an e-mail to your TA right now (go to <https://ubmail.buffalo.edu/>). Your TA will reply to your e-mail - if you do not receive a reply, ask your TA for help. Depending on e-mail load, it may take a minute or two for the e-mails to be transmitted.

### Some Basic Linux

In this section, you will experiment with some basic Linux commands, and learn a little bit about the way in which Linux organizes its file system.

#### Basic Linux directory structure

Like most file systems, Linux's is tree-structured. The root of the file system is named `/`. As a user on the system you have a *home directory*, where all your files are stored. Linux has a "shortcut" to represent your home directory. This shortcut is the tilde (`~`). Whenever you want your home directory, you can just type `~`. To find out where your home directory is, type

echo ~

When I do this, I find out that my home directory is `/home/csefaculty/alphonc`. This means that my home directory `alphonc` is located in the subdirectory `csefaculty`, which is a subdirectory of `home`, which is an immediate subdirectory of the root directory (which is, as mentioned above, designated by `/`).

`"/home/csefaculty/alphonc"` is a *path* which describes how to get from one directory (in this case the root directory) to another (in this case my home directory). A path can either be *absolute* (starting at `/`) or *relative* (starting at the current directory).

Every directory has two special entries, `.` and `..`. The entry `.` refers to the directory itself, and `..` and refers to the directory's *parent* directory (i.e. the directory of which this one is an immediate subdirectory). The `.` and the `..` are "shortcuts" as well. No matter where you are, if you want to use your parent directory, then you just use `..`

The root directory is the only directory in the file system which does not have a parent directory. (Actually, this is not quite accurate. If you inspect the root directory you will find that there is a `..` entry. The parent of the root directory turns out to be the root directory itself!)

### Some Linux Commands

Below I have listed some basic Linux commands. Read the description of each, and carry out any exercises given. Note that many of these commands are equivalent to what you can do in a graphical environment, and there is a file explorer available to you on your desktop, but these can serve as a reference in case you are in a Linux environment without a graphical system on top of it (such as when you type commands directly in a terminal window).

- **ls** - This command gives a *listing* of the files in a directory. Without any arguments, it provides a listing of the files in the current directory, and is therefore equivalent to

```
ls .
```

If you specify a directory using a path, `ls` will respond with a listing of the contents of the specified directory. Try typing `ls` at the Linux prompt.

- **cd** - This command changes the current directory. Without any arguments, it changes the current directory to your home directory, and is therefore equivalent to

```
cd ~
```

If you specify a directory using a path, `cd` will set the current directory to the specified directory. One neat thing is that using `..` as an argument to `cd` moves you to the parent directory.

- **man** - The `man` command brings up a manual page (help screen) for a specified command. Let's get the man page for the `ls` command by typing `man ls`. Read what `ls -l` is supposed to do, and then try it.
- **mkdir** - This command creates the specified subdirectory.
- **cp** - This command *copies* a file to a new location, leaving the original unchanged. Making sure you are in your home directory, typing `cd ~` first if necessary, try the following: `cp .cshrc letter.txt`, then use the `ls` command to look at your directory contents.
- **mv** - This command *moves* a file from one location to another. If the destination location is a directory, then the file is just moved there. If the destination location is a filename, then not only is the file moved there, its name is also changed. You can use this to rename a file without moving it; assuming you had a file named `letter.txt`, you could rename it to `MyCoverLetter.txt`, type the command `mv letter.txt MyCoverLetter.txt`; if you created a file named `letter.txt` in the previous step, try this out and see what the result is using the `ls` command.
- **rm** - The command `rm` removes a file. This command is not (easily) reversible, and it will not ask you (by default) to confirm your delete action. Hourly snapshots are taken of the files on the system that can help you restore something if it is accidentally deleted. These snapshots are located in `/home/.snapshot` and then are kept by the hour (0 being the most recent snapshot). When you get into the hour, you use your path to navigate to your directory and then you can copy the files from the snapshot back to your home directory. If you've been following along with the exercises, type `rm MyCoverLetter.txt`, then check the effect using `ls`. For more information on the backup snapshots, see: <https://wiki.cse.buffalo.edu/services/content/netapp-filer-snapshots>
- **rmdir** - This command removes the specified directory.

There is nothing you need to hand in this week. Be sure, in case you did not do everything in these exercises, to make time to come in during open lab hours. Open hours are any times listed at:

<http://www.cse.buffalo.edu/faculty/alphonse/Courses/Fall2009/cse115/Labs/>

when the lab is not scheduled for use by a class.