

ASTRONOMY 460: INTRODUCTION TO UNIX AND COMPUTING AT THE UW ASTRONOMY DEPARTMENT¹

Snežana Stanimirović, September 29, 2014

1. Introduction

The purpose of this document is to introduce you to the Unix operating system and basic computing resources of the Astronomy Department. Our goal is to be able to login and write observing scripts and reports, and do data processing using computers in our classroom.

The best way to learn Unix is to sit in front of a computer for an hour or so and try it out yourself. Once you master some basic commands, you will be amazed at the many useful things you can do.

2. Logging In

We have two types of computers in the classroom: iMacs we used to run the SRT Simulator, and several Dell computers (called Stripe and Jelen) used for undergraduate research. iMacs run under the OS X operating system, which is Unix-based with additional graphical features developed by Apple. Dell computers run Linux operating system, again Unix-based with some additional features. So working with both types of computers will be the same in terms of commands executed etc. The only difference is that Dells are located on the local Astronomy network, therefore you can share general network features available and developed over the years in the Department (e.g. printers, various functions and procedures). iMacs are only used for classes so they have a limited functionality. Therefore, we will use them only as a platform to log in to one of network computers called “leffe”.

So, if you are using one of several Dell computers (please try to use only Stripe and Jelen): use your provided user name and password to login, open an xterm window (located on the bottom left corner, the icon looks the same as for Terminal), and you are ready to practice Unix commands. Please remember to log out fully when you are done.

¹This document was modified from Prof. Carl Heiles’ Unix introduction for the UC Berkeley Undergraduate Lab.

If you are using an iMac (any one except OBSERVER): open a terminal window and using your user account/password log in to “leffe”:

ssh -X username@leffe Once on “leffe” you are ready to play with Unix.

If you haven’t had a user account before your password will look very strange. Please change it to something reasonable so you can remember it. To do this please run **passwd** .

3. Getting Unix Help

To read a help page on how to use particular Unix command use:

leffe% man lpr , where **lpr** is the printing command. Press **q** to exit from the help file.

Unix is a very powerful operating system and has also a scripting component that is useful for searching through various files and directories. If you want to know more, a useful handbook is “Unix in a Nutshell” by Arnold Robbins, available from Amazon. We also have it in the Astronomy (Woodman) Library. I have a copy in my office you are welcome to borrow.

4. Directory Structure

The directory structure of Unix is organized in a hierarchical system, or by levels. Once you log in you will be in your home directory called /usr/users/sstanimi (as an example I am using here username sstanimi). When you move throughout the directory system you are moving from level to level, with sub-directories being nested inside directories (this is of course very similar to Microsoft Windows operating system that most of you are familiar with).

We have a special class directory where all of you should work.

This is /d/leffe/astro_460/Class_2014 .

Each of you should make an individual directory (e.g. Birenbaum) and this is the place where you will be keeping your data/processing/observing files this semester. Feel free to make your own sub-directories to organize your files (your homework is actually asking you to do this).

5. Working with Directories

First, check to see in which directory you are. At the Unix prompt, type the following:
leffe% pwd

This command stands for *present working directory*. The computer should return something along these lines: `/d/leffe/astro_460/Class_2014/sstanimi`. From this you know that you are in your login directory.

To see a listing of what already exists in your directory use:

leffe% ls

this will list files and sub-directories you already created, or have some system-based directories (like Desktop, Public, Music etc).

Something I find particularly useful if I have many, many files in my directory is:

leffe% ls -lt

This will list all files in the chronological order, with the newest files being at the top. You will also see when exactly were the files created, who made them, and how large they are.

Another super useful thing in Unix are wildcards! Let's say you know you made a file that starts with *a* but you don't remember exactly how you called it. Simply do:

leffe% ls a*

and you will get a listing of all files in your directory starting with *a*. Same works for numbers.

To create a sub-directory called *lab1* in your home directory:

leffe% mkdir lab1

Notice how will this sub-directory appear when you do **ls**.

To move in/out of a directory use the **cd** command. To move to a particular directory, say *lab1*:

leffe% cd lab1

To move from your current directory one level up:

leffe% cd ..

You can move to directories by specifying their full path, e.g.:

leffe% cd /d/leffe/astro_460/lab1

You can delete (remove) a directory by first deleting (removing) all the files in the directory, and then typing:

leffe% rmdir lab1

6. Creating (Emacs) and Viewing Files

We will be creating files (e.g. observing scripts or processing files) using a Unix-based text editor called *Emacs*. To open a new file for editing:

```
leffe% xemacs myfile.cmd &
```

A new window will pop up with a bar on the top that has menu for driven commands. You can easily type in the window and move up/down to where you want to go. Please experiment. The menu bar is very handy (this is similar to Microsoft Word), but you can also learn the basic Emacs keystrokes which can save you time.

For a quick look at a file without editing:

```
leffe% more myfile.cmd
```

The first page of your file will appear, and you can either view the rest of the file by hitting the space bar or using the Return button (which will go line-by-line). To stop looking at the file, hit **q** and you will return to the Unix prompt.

7. Moving, Copying, and Removing Files

To copy a file *lab1* to another (new) file called *lab2*:

```
leffe% cp lab1 lab2
```

This is handy if you want to make a new modified version. This way you simply start by editing *lab2*. If you want to copy a file to a new version located in a different directory just give the full file path:

```
leffe% cp lab1 /d/leffe/astro_460/Class_2014/lab2/lab2.cmd
```

To move a file from one location to another:

```
leffe% mv lab1 /d/leffe/astro_460/Class_2014/lab3/junk.cmd
```

This way *lab1* is essentially being moved to a new location also is given a new name *junk.cmd*

To remove a file:

```
leffe% rm lab1
```

but a better and safer version to use is:

```
leffe% rm -i lab1
```

this way you will be asked if you really want to remove a file. Once removed, the file is gone for ever, so please be careful. Especially be careful with using wildcards as things like **rm *** will remove everything in your directory!!!!

8. Permissions

As each of you has individual user account you will be able to edit/remove only the files you made. This is great as otherwise we would need to worry about accidentally changing/deleting files made by other people. Each file or directory will have some permissions set so it's handy to know how to understand this. For example,

ls -l problem2.txt

on one of my files produced the following:

```
-rw-r--r-- 1 sstanimi users 3666 Sep 10 2008 problem2.txt
```

This means that user *sstanimi* created this particular file on Sep 10 2008, and 3666 refers to the size of the file in kilobytes. The very first thing with *r* and *w* tells us about file permissions. These 9 characters correspond to three groups of permissions, first three characters are for the user (you or *sstanimi* in this case), the second group of three refers to permissions given to the group of users *sstanimi* belongs to (called *users* in this case), and the last set of three refers to the world or anyone else.

For each character there are 3 possible options: *x* (execute), *r*(read), *w*(write). So, user *sstanimi* has the read and write permission for this file, while everyone in the group called *users* or anyone else on the astronomy network is allowed to only read this file.

This example generally applies to most cases, however you can change permission for your files using command **chmod** if you really want this (e.g. sometimes is handy to allow others to edit a particular document). We will not be needing this in this class.

9. Printing

Once logged in you can print your files using the *floor3* printer:
leffe% lp -dfloor3 myfile.txt