Introduction to UNIX

Brian Keeney

January 18, 2005

UNIX is an operating system (OS) just like Windows or MacOS; it is the interface between you and the computer. The majority of the scientific community (especially in astrophysics) uses UNIX (or a similar OS called Linux) because they are designed to facilitate multiple users and can run on almost any computer architecture (Intel, Sun, Apple, etc.). Therefore, files and programs are easily exchanged amongst collaborators at various locations and there are many scientific programs in UNIX/Linux that do not exist for other operating systems.

This week I'll give you a run-down of essential UNIX commands and give you a chance to practice using them. Next week, I'll teach you how to perform some basic tasks from within IRAF and get you started on the data reduction for Lab 1 (hopefully you'll all have some data by then!).

What is a "Terminal"?

UNIX is a primarily a command line OS, meaning that your primary interface with the computer will be through the keyboard, typing commands, rather than a graphical, mouse-based interface. UNIX does have GUI (graphical user interface) components, but the command line is what makes it powerful. After logging on to cosmos, you should always make sure that a terminal (a window which contains the command line interface) is open. Instructions for how to open a terminal can be found in the handout from last week.

The terminal window has a "command prompt" that can vary depending on the system, but on cosmos it should look like `cosmos:~>` with a cursor next to it. When you see a command prompt, it means that the terminal is ready to accept a command.

NOTE: The commands you enter into the terminal are case sensitive, meaning that there is a difference between uppercase letters and lowercase letters, so be sure to type in the appropriate case (usually lower case).

Directory Structure

Directories in UNIX are similar to folders in Windows. There is a directory "tree" and each directory can contain files or more directories, or both. When you're in a terminal with a command prompt, you are in one of these directories. Whenever you log in, you start in your home directory. This is your space on the computer system and you have the ability to create, edit, and delete files here (you are restricted from doing so in most other directories). You may create as many subdirectories as you'd like within your home directory tree to organize your files.

To find out the directory that you're currently in, use the `pwd` (print working directory) command. For example, this is the result of running `pwd` from my home directory:

```
cosmos:~>pwd
/raid/home/grad/keeney
```

Thus, my home directory (my username is `keeney`) is a subdirectory of `/raid/home/grad`. Your path to your home directory should look similar (although it will be a subdirectory of `/raid/home/ugrad`). The "path" is the roadmap to a file or directory. For example, the `FIT2fits` file in my home directory has a path of `/raid/home/grad/keeney`. We could completely specify the location and name of the file by specifying the path and filename: `/raid/home/grad/keeney/FIT2fits`.

To examine the contents of a directory, use the `ls` command. For example, in my home directory `ls` returns the following output:

```
cosmos:~>ls
3520/      FIT2fits*  Mail/      iraf/
```

To create a directory, use the `mkdir` command. For example, if I wanted to make a directory to contain image files from the SBO telescopes, I would type the following:

```
cosmos:~>mkdir sbodata
```

Now, when I type `ls`, the directory `sbodata` exists:

```
cosmos:~>ls
3520/      FIT2fits*  Mail/      iraf/      sbodata/
```

To move into the `sbodata` directory, we use the `cd` (change directory) command.

```
cosmos:~>cd sbodata
```

Now, when we do a `pwd` command:

```
cosmos:~/sbodata>pwd
/raid/home/grad/keeney/sbodata
```

To move up in the directory tree (i.e., from the directory `/raid/home/grad/keeney/sbodata` to the directory `/raid/home/grad/keeney`), we use `..` to represent one directory higher in the directory tree than your current location. For example, from the new `sbodata` directory, we can move up in the tree by using:

```
cosmos:~/sbodata>pwd
/raid/home/grad/keeney/sbodata
cosmos:~/sbodata>cd ..
cosmos:~>pwd
/raid/home/grad/keeney
cosmos:~>cd ..
cosmos:/home/grad>pwd
/raid/home/grad
```

Note that you can use `cd ..` to keep moving all the way up the directory tree until you get to the "root directory", `/`, but you won't be able to modify the contents of any directories that are outside of your home directory.

You can also use the `~` symbol to represent your home directory. You can go back to your home directory from **anywhere** by typing `cd ~`. To indicate someone else's home directory, you can use the shortcut `~`*username*. For example, if you wanted to `cd` into my home directory you could either type `cd /raid/home/grad/keeney` or `cd ~keeney`.

Finally, you can delete a directory (but only if it's empty) using the `rmdir` command. To delete the `sbodata` directory, go back to the home directory and then use the `rmdir` command.

```
cosmos:~/sbodata>pwd
/raid/home/grad/keeney/sbodata
cosmos:~/sbodata>cd ..
cosmos:~>pwd
/raid/home/grad/keeney
cosmos:~>ls
3520/      FIT2fits*  Mail/      iraf/      sbodata/
cosmos:~>rmdir sbodata
cosmos:~>ls
3520/      FIT2fits*  Mail/      iraf/
```

Files

At this point we've made some directories, but they're not very useful if they're empty. We use files to store information or commands, and they are what we use to perform scientific analysis. As in Windows, files might contain a text document, a PDF document (like this handout), an image in one of many formats (FITS, jpg, gif, tiff, etc.), or just about anything else that you can imagine.

Let's start by learning how to move and copy files into directories. You can copy files using the `cp` command. The syntax for this command is `cp` *original-filename new-filename*. For example, to copy a FITS image from my `sbodata` directory to yours:

```
cosmos:~>cp ~keeney/sbodata/Orion_Ha_60s.FIT ~/sbodata
```

This will copy the file named `Orion_Ha_60s.FIT` contained in my sbodata directory (under my home directory) to the directory `sbodata` under your home directory. If you wanted to copy a file but change its name, you could do so by typing something like:

```
cosmos:~>cp ~keeney/sbodata/Orion_Ha_60s.FIT ~/sbodata/Orion.FIT
```

The new file in your directory would then be called `Orion.FIT` instead of `Orion_Ha_60s.FIT`.

You can move files using the `mv` command. Moving a file is just like copying it, except that the original file will be deleted. The syntax for `mv` is exactly the same as for `cp`. For example, you can use `mv` to rename the file `Orion_Ha_60s.FIT` to `Orion.FIT`:

```
cosmos:~/sbodata>pwd
/raid/home/grad/keeney/sbodata
cosmos:~/sbodata>ls
Orion_Ha_60s.FIT
cosmos:~/sbodata>mv Orion_Ha_60s.FIT Orion.FIT
cosmos:~/sbodata>ls
Orion.FIT
```

To delete a file, use the `rm` (remove) command. The syntax for this command is just `rm` *filename*. For example, after copying `Orion_Ha_60s.FIT` and renaming it `Orion.FIT`, you might want to delete it altogether by typing `rm Orion.FIT`. Note that you might have to confirm the deletion by responding `y` or `yes` at the query that comes up.

You can also use "wildcards" in filenames. If you wanted to list only the files that end in `.FIT` in a current directory, you could do so using `ls *.FIT`. The `*` is a wildcard and represents any set of characters of any length. Try this by typing `ls ~keeney/sbodata` and `ls ~keeney/sbodata/*.FIT`. Notice that the second command only lists the file that ends in `.FIT`.

Similarly, if you wanted to copy all of the `.FIT` files in a given directory, you could do so using:

```
cosmos:~/sbodata>cp ~keeney/sbodata/*.FIT .
```

Here I've used another handy shortcut, the `.` character. In UNIX, "`.`" = "the current directory" (just like "`..`" = "one directory up from here"), so if you're already in the directory that you want your files to be placed in, you can use the syntax above. Of course, you can always type it the long way (`cp ~keeney/sbodata/*.FIT ~/sbodata`) if you'd like.

Finally, if you'd like to remove all of the `.FIT` files in a directory, you can do so using `rm *.FIT`. Note that you can also delete all of the files in a given directory using `rm *`, but **this is not**

recommended unless you know **exactly** what you are doing. You don't want to lose all of your data the night before a lab is due!

UNIX Syntax and Help

UNIX commands like those we've been using have a set of options that can be set by "flags". Flags tell the command to run with some option set and are specified by a dash (`-`) followed by a letter. For example, try typing `ls -l` instead of `ls`. You should notice that `ls` now lists more information than it did before (this is the so-called "list long" command). Most, if not all UNIX commands have options that you can set.

Of course, this doesn't do you much good if you don't know what flags do what or which options are available. UNIX has information on individual commands on a "manual" page for that command. To see the manual page, type `man` *command-name*. Try typing `man ls` and look for the description of the `-l` flag.

Text Editors

Text editors are programs which allow you to edit a text file. It's somewhat like a word processor (i.e., Microsoft Word) without text formatting for bold, underline, italics, etc. We'll use text editors for things like making data files to pass to an IRAF task.

Probably the easiest text editor to use is `nedit`. It has a graphical interface and is reasonably intuitive to use. I don't often use it myself, but I suspect most of you will opt to use it. To use `nedit` you can type `nedit` at the command line. You can then open files, save files, exit `nedit`, etc. from the "File" menu. Alternatively, you can type `nedit` *filename* at the command line to open the file *filename* from the start. Other text editors that you may wish to try are `emacs`, `xemacs`, and `vi`; I generally use `emacs`, but it is not as easy to pick up as `nedit`.

If you want to view the contents of a text file (rather than edit it), you can use one of two UNIX commands: `more` or `less`. These two commands operate the same way but allow you to navigate the contents differently. The `more` command allows you to page down one or 1/2 page at a time, whereas the `less` command allows you to move line by line. To view a text file, use either `more` *filename* or `less` *filename*. You can try them out with your IRAF `login.cl` file.

```
cosmos:~>cd ~/iraf
cosmos:~>more login.cl
```

You've probably noticed by now that `nedit` (and `emacs` for that matter) belong to a class of programs that operate by opening a new window outside of the terminal. When you're running a program from the terminal, you lose the command line until that program has finished running. When a program opens its own window, however, it would be nice to still be able to use the

terminal. You can do this by running these commands "in the background" by appending a `&` at the end of the line. This will allow you to keep the terminal active (i.e., keep the command prompt) while the program called with the `&` is running. For example, you can run `nedit` while keeping the terminal active by typing:

```
cosmos:~>nedit &
```

Now you can use `nedit` and the terminal at the same time. Note that you do not want to use this for commands like `ls` or `more` because you want to use the terminal to interact with those commands. In general, you only want to use the `&` for commands that run in their own window.

Shortcuts

UNIX has two convenient shortcuts when working with the command line. The first is tab completion. While you're typing a filename or command, if you type enough of that filename such that the rest of it is unique (i.e., there are no other files in that directory to which you could possibly be referring), hit the `<tab>` key and UNIX will fill in the rest of the filename for you.

The second shortcut is to copy and paste filenames. Use `ls` to see what's in the directory, then click and drag with the left mouse button to highlight the filename that you want and click the middle button to copy the highlighted text where you want it to go. You can also copy and paste highlighted text using the "Copy" and "Paste" buttons on the leftmost part of the keyboard. These shortcuts make working with long filenames **much** easier!

An Incomplete List of Other Useful Commands

netscape	This brings up a Netscape web browser window. You can also open Netscape by clicking on the globe with the clock on it in the Front Panel of CDE.
passwd	This allows you to change your password for logging onto the system.
xv	This is a general image viewer which supports many image formats.
gv	This is a postscript file viewer.
acroread	This will display a PDF document.
gimp	This is the "GNU Image Manipulation Program", sort of like Photoshop for UNIX.

For more useful commands, tips and tricks, and other general UNIX techniques, there are *tons* of introductory UNIX books available, as well as some good websites, so look around. Also, I am willing to answer any questions that you have.

Let's See What You Learned...

Now we'll see if any of this sunk in. Try to use the information in this handout to do the following:

1. Open a Netscape browser window and go to the class homepage:
   `http://casa.colorado.edu/~keeney/classes/astr3510/`
2. Verify that you're in your home directory. Your home directory should be `/raid/home/astr/ugrad/`*username*, where *username* is your username for logging onto cosmos.
3. From your home directory, create a directory named `sbodata`. Verify that it was created.
4. Copy the file `Orion_Ha_60s.FIT` from the `sbodata` subdirectory of my home directory to the directory that you just created.
5. After copying the image, rename it `orion.fits`.
6. Use one of the programs in the "incomplete list" above to view this Hα image of the Orion Nebula. I will show you how to use some image viewers designed specifically for FITS files next week.