Introduction to Linux and Computing at the CfA

Taken from the *RSI 2012 Athena Introduction*
with edits by Anjali Tripathi

Astro 189
January 2013

The Shell

In Unix systems, the shell is the interface between the user and the computer. When you type a command into the shell at the prompt and press the "Return" key, the computer will do it. For example, you run programs from the shell by typing the name of the program. Try typing the following commands in your xterm window:

```
cfa0> whoami
cfa0> xclock
```

Directories
Directories are folder that contain other files and folders.

When you first log in, your shell starts out in a special directory called your home directory. Your home directory is similar to "My Documents." All of your files will be stored here.

The directory that your shell is in at any time is called the "current directory." If your forget what directory you're in, use the command `pwd` to find out where you are.

Files and their names You can name files almost anything you want with a few restrictions:

- 255-character maximum
- Letters, numbers, hyphens, periods, and underscores are allowed
- Spaces are allowed but try not to use them because file names with spaces require special handling on the shell.
- All names are CaSe sEnSItiVe

The Shell Seven

- `ls` - Lists the contents of the current directory
- `cd` - Change directory
- `rm` - Removes (deletes) one or more files or directories
- `mv` - Move/rename a file or directory
- `cp` - Copy a file or directory

- `mkdir` - Create a new directory
- `rmdir` - Remove a directory

Manipulating Directories

- To make a new directory:

```
cfa0> mkdir foo
```

- To remove an empty directory:

```
cfa0> rmdir foo
```

- To list the contents of the current directory:

```
cfa0> ls
```


Moving Within Directories

The command to change directories is `cd <directoryname>`. Simply typing `cd` will return you to your home directory.

Let's say `stewieG` has the directory `bar` in his `foo` directory.

- Within `foo`, he could type
  `cfa0> cd bar`

- From anywhere, he could type
  `cfa0> cd /home/stewieG/foo/bar`

- `pwd` prints your current directory.

```
cfa0> pwd
/home/stewieG
```

In addition, whenever you're working with files and directories that already exist, you can press the TAB key after typing the first few characters of the filename and it will add the rest of any matching filenames.

Directories Redux

The location of a file or directory is specified by its path. In the path, different objects are separated by a forward slash (/).

Absolute paths start with a forward slash. Example:

`/home/USER/foo.tex`

Relative paths do not start with a slash and specify a location relative to the current directory. Example:

If you are in `/home/stewieG`, then `Public/foo.tex`
refers to `/home/stewieG/Public/foo.tex`.

~, ., ..

If you leave your home directory, you may need to refer to it as the starting location of a path. You can do this with a tilde.

`/foo/bar.tex` is the same as `/home/USER/foo/bar.tex`

Every directory contains two special directories: `.` and `..`

- `.` refers to the current directory
- `..` refers to the directory one level up.

Makin' Copies

The command to copy files is `cp`. It takes two paths as arguments. There are two main usages:

- Copying one file to a new location with a new name:

```
cfa0> cp foo.tex ~/bar/baz.tex
```

- Copying files in a directory to another directory:

```
cfa0> cp * ../foo/
```

Movin' on up...

The `mv` command also has two main uses:

- Moving without keeping a copy in the original location:

```
cfa0> mv foo.c bar/
```

- Renaming files:

```
cfa0> mv foo.c baz.c
```


Get It Outta Here!

- `rm` is the standard UNIX command. This permanently gets rid of the file and can't be reversed.

- To get it to ask you if you're sure you want to delete before it deletes, use `rm -i`

Your turn

- Figure out where you are by using the `pwd` command
- List the contents of your current directory using `ls`
- Create a new directory called *foo* using `mkdir`
- Change your current directory to the directory *foo* using `cd`
- Create a new file called *bar.txt* using the `touch` command

Your turn, continued...

- Make a new file, *blank.txt*, and another directory called *ps1*
- List the contents of your current directory, then move the file *bar.txt* to the *ps1* directory with the `mv` command
- Copy the file *blank.txt* to the *ps1* directory using `cp`
- Remove *blank.txt* and *bar.txt* from the *ps1* directory with `rm`
- Remove the *ps1* directory

- Change your current directory back to your home directory

Options and Arguments

Let's use the `ls` command to learn about options and arguments.

- **Options** modify the action of the program. For example, you can use `ls -a` to view all files (including normally hidden ones). `ls -l` shows a long form of the information.

- **Arguments** tell a program what it should act on. For `ls` the arguments are the names of the files or directories to list. If no argument is given, the default is the current directory. Example:

  `ls /Paper` shows the files in your own `Paper` directory

It Started with a "C" . . .

If you want to specify several files at once there are two kinds of "wildcards" you can use:

- `*` matches 0 or more characters. Example:

```
cfa0> ls P*
```

- `?` matches any one character. Example:

```
cfa0> ls /home/rsi/200?
```

- These two modifiers can be used together.

I Want less!

`less` is a program that displays text files one page at a time.

```
cfa0> less ~/.anyone
```

- `[space]` scrolls forward one screen
- `b` goes back one screen
- Arrow keys move text forward or backward line by line
- `q` quits

Gedit: Text Editor

CfA Linux comes with a graphical text editor called Gedit. You can access Gedit using:

1. At the terminal, type `gedit &`.

Emacs

`emacs` is another text editor. It is not a word processor. It is a text editor. You can run `emacs` in two ways:

- `cfa0> emacs foo.tex &`

  opens the file `foo.tex` if it exists and creates `foo.tex` if it doesn't exist.

- `cfa0> emacs &`

Emacs: What you see

- Menu Bars
- Buttons
- White space
- Minibuffer

You can do a lot of stuff through the menus and with the buttons and your mouse, but there are a lot more commands you can

access with just the keyboard, which, once you get used to, will make using emacs much faster. And, of course, you'll have to use the minibuffer whether you use the mouse or the keyboard. . .

Minibuffer? But I Just Met Her! The minibuffer is the bottom line of the emacs window. It displays commands and messages. It is a source of many helpful things, such as confirmation that your file is saved or a prompt to finish out a command.

When you are typing emacs commands, you're using the minibuffer.

Emacs Command Syntax

A brief description of our syntax:

- `C-x` means hold down the `[Control]` key and press `x`
- `C-x u` means hold down the `[Control]` key and press `x`, then release the `[Control]` key and hit `u`
- `C-x C-s` means hold down the `[Control]` key and press `x` followed by `s` before letting go of `[Control]`
- `M-x` means to hold down the `[Meta]` (also called the `[Alt]` key on Linux machines) and press `x`

Help! I'm stuck in a minibuffer!
Many key combinations have special meanings in `emacs`, and it's easy to invoke a weird command by mistake. When this happens, don't panic: there are ways out.

If you get stuck in the minibuffer typing to a prompt you don't recognize, press `C-g` a few times. This should get you out of whatever trouble you were in (you'll see the word "Quit" in the minibuffer).

The **undo** command in `emacs` has two key sequences: `C-x u` and `C-_` (control-underscore). If you suddenly delete a bunch of your paper by mistake, using either of these commands should bring it back. You can undo several consecutive actions by using the undo command repeatedly.

Manipulating Files

- `C-x C-f` opens an existing file or creates and opens a new file. You will be prompted in the minibuffer for the name of the file to open.

- `C-x C-s` saves the file you are working on. Use this often.

- `C-x C-c` quits `emacs`.

Oh, and `C-h t` opens the `emacs` tutorial file. It might be well worth an hour of your time to work through that.

Manipulating Text

- `C-w` Cut ("wipe")
- `M-w` Copy
- `C-y` pastes ("yanks") what was most recently cut or copied
- `C-k` cuts the line to the right of the cursor
- `M-q` reformats current paragraph

Search and Replace

- To find the next occurrence of a string in your file, press `C-s` and begin typing the string. `emacs` will search as you type. You may press return to leave the search or press `C-s` repeatedly to find the second, third, and following occurrences of the string.
- `C-r` is like `C-s` but searches backwards.
- To replace all occurrences of `foo` with `bar` in your file, *go to the beginning of the file*, then press `C-M-%`. Yes, thats `Ctrl-Meta-Shift-5`!! Enter `foo` and press return; then enter `bar` and press return.

Getting Help

- Reference Manual entries are available for most commands. To get more information about a command type

  ```
  athena$ man <command>
  ```

- Google it

- Ask us! Email the course staff - Ruth, Matt, and Anjali are here to help!