# Lab #1

**Physics 91SI   Spring 2013**

**Objective:** In this lab, you will become familiar with basic UNIX commands and tools.

This lab does not involve any actual programming - we'll start with that on Thursday. You will need to have an SSH client set up and be able to log in to the `corn` cluster, which we will be using for the remainder of the course. If you have not already, please follow the instructions on the course website, then connect your client to `corn.stanford.edu`.

Throughout this exercise, feel free to use any resources you want to accomplish the job. In fact, we highly encourage you to use the internet to find out what certain commands do or which command is the right one for your job -- that is a core part of this class. The resources on the course website are a good starting point, but don't be afraid to just google something.

You won't need an X-server for this lab, but if you haven't gotten one working we will come around and help you get set up - you'll need it for Thursday.

## **Part 1:** Basic UNIX

First, log on to `corn` using your ssh client.

Use `ls` and `cd` to look around a bit. What is the directory structure? What is in the directories in your home directory? Remember that . and .. refer to the current directory and the current parent directory, respectively. Try using `man ls` to find other options, and see if you can find any "hidden files."

Now, `cd` into a directory of your choice. Use `cd` again to go into a directory, but this time use the full path starting at / , not a relative path. *Hint: use* `pwd` *("print working directory") to find the current full directory path.*

Go back to your home directory ( ~ ). Go ahead and make a Physics 91SI directory - this will be your "working directory" for the quarter. You can call it whatever you want, and put it directly into your home directory or in a subdirectory of your choosing. *Hint: UNIX is case sensitive. Also, you can't put simple spaces in names on the command line. If you want to use names with spaces in them, you can enclose your entire statement in double quotes:*

```
"physics 91 si"
```

*Alternatively, you can add the backslash-space character instead of a space:*

```
physics\ 91\ si
```

Because dealing with spaces is annoying, directory and file names in UNIX typically use hyphens instead of spaces. Use `mv` to rename your directory to something more convenient. Now go into your working directory and make a `lab1` directory. (We encourage you to do this for every lab, from now on - and we'll talk about a better way to do it on Thursday)

Now, *copy* the `lab1` directory into another directory, named -- say -- "`lab1-copy`" (*Hint: try* `man cp`). Remaining in the directory that you are in, use a text-editor of your choice to make a text file (with ending `.txt`) with some content, e.g. *"Hello World!"*, *inside* the lab 1 directory. Without changing directories, copy that text file into your `lab1-copy` directory. Still without changing directories, read the copy of the text file on the command line to assure it has the contents that you would expect it to.

## **Part 2:** Challenge

Make a new text file in your original `lab1` directory. Put in it -- in order, and on separate lines -- all the commands necessary to

1. Print a message for each of the following steps, explaining what the commands are doing
2. Print the contents of the `lab1-copy` directory
3. Print the contents of any `.txt` file in that directory
4. Delete the lab 1 copy directory and all its contents

Make sure to use the correct full or relative paths while writing this. Also note that relative paths are always taken relative to the current directory, which changes every time you use `cd`.

Now, figure out how to change the permissions of this text file to "executable" and do it.

Actually execute the file and verify that it did what it was supposed to. This is a first example of what is called a "shell script", a file that executes a fixed series of commands so that you don't have to worry about typing them over and over again. We'll talk more about these on Thursday.

If you weren't able to complete the challenge, go ahead and delete the `lab1-copy` directory directly from the command line.

## **Part 3:** Aliases and Links

Now, open the file `~/.cshrc` in gedit or another editor. Where is this file located? What does the file do?

Find the relevant section, and add an `alias` that changes the current directory to your `lab1` directory. Try the command you created - does it work? It turns out that `cshrc` only gets invoked on login, which means that you either have to type out the command again on the command line or log out and log back in for it to work.

Now try to do the same thing with an environment variable, setting its value to the full path of your `physics91si` directory. Look up the command necessary; it should enable you to type something like "`cd $SI`" in your shell, as a shortcut of sorts.

Now try to make a symbolic link to your `physics91si` directory, and put it in your home directory. It might also be useful to make one to the course directory in `/afs/ir/class/physics91si` - we'll be posting starter code and other goodies on there throughout the quarter. *Hint: try "`man ln`"*

You'll notice that the alias, variable, and link do more or less the same thing in this example. If you have time, look at the other aliases and variables in your `cshrc` file, and see how the link you made behaves with filesystem commands (`ls`, `cd`, etc.). Why is the link useful? Why might you use an environment variable or alias instead?

## **Part 4:** More Commands

The attached sheet contains a list of common UNIX commands. Use the resources at your disposal, including -h (or --help), man pages, and the internet to fill in the blanks.

Also, there's an easter egg in one of the instructor's AFS directories. Using what you've learned about the file system and UNIX commands, find the egg and open it...