

## 98-172 : Great Practical Ideas for Computer Scientists

---

### How I UNIX

#### Welcome to the first ever lab-itation!

To re-iterate, the lab-itations are a series of sessions where we help you get familiar with the topic of the week. This week, we want to make sure you are familiar with how to get access to a UNIX machine (`unix.andrew.cmu.edu`), use it, transfer files, and use AFS.

#### SSH

- (a) **What is SSH?** `ssh` is a program for logging into a remote machine and for executing commands on a remote machine. Here at CMU, you will probably make extensive use of it as a way to do your programming homework.
- (b) **Windows** We suggest that you use the program PuTTY (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>). It is by far the best terminal client for Windows. You should get started by downloading and installing PuTTY on your Windows machine.

In the **Host Name** box, you should type `unix.andrew.cmu.edu`, then click the “Data” tab under “Connection.” Type your andrewid into the **Auto-Login Username** box. Click session all the way at the top to bring you back to the main screen. Type a name like “Andrew UNIX” into the **Saved Sessions** box, and click the Save button.

For all future times running PuTTY, you can just double-click on “Andrew UNIX” in the saved sessions box, and it will auto-fill in all of the information.

When you log on, you will be asked for a password to connect, this is the same password you use to log onto WebISO.

To copy text in PuTTY, all you have to do is select it, and to paste, just right click.

- (c) **Mac OS X** Congratulations! Your OS has a built-in terminal (“Terminal.app”). You might consider looking at one of the other Terminal emulators for OS X like `iTerm2` if you don’t like Terminal.app. Open up your terminal emulator (it’s just a normal program). To facilitate quick sshing, you should edit the `ssh` config file. Open the file `.ssh/config` in your home directory (creating it, if it doesn’t exist). (So, for instance, on my machine, the file path would be `/Users/adamblan/.ssh/config`). Add the following (replacing `ANDREWID` with your andrewid) to this file:

```
Host andrew
  Hostname unix.andrew.cmu.edu
  User ANDREWID
  GSSAPIAuthentication yes
  GSSAPIDelegateCredentials yes
```

Normally, without the above incantation, you would need to type `ssh ANDREWID@unix.andrew.cmu.edu` to ssh, but now you can just type `ssh andrew` and the rest will be auto-detected. If you prefer

a different name than `andrew`, you can change that line in the `ssh` config file. When you type `ssh andrew`, you will be prompted for a password—it is the same password you use to log onto WebISO.

- (d) **Linux** Although you are probably already familiar with `ssh`ing, you may not have edited your `ssh` config file. You might consider reading the OS X section!

## SCP

- (a) **What is SCP?** `scp` copies files between hosts on a network. It uses `ssh` for data transfer, and uses the same authentication and provides the same security as `ssh`. `scp` will ask for passwords or passphrases if they are needed for authentication.

- (b) **Windows** The program we suggest you use is WinSCP (<http://winscp.net/eng/index.php>). Get started by downloading and installing it. Open it up, and use the same information as for `ssh`; if you want it to remember your password, you can enter it on the first screen, otherwise, it will just ask you for it later.

If you click the “Save” button instead of the “Login” one, you won’t have to enter the information again. In the future, you can just double-click on the entry.

When you login, you will see two panels: the left one is your machine and the right one is the machine you connected to. You can drag files between the two machines.

To test it out, create a file called “interesting” which contains an interesting (or boring!) fact about yourself and copy it to the directory `/afs/andrew.cmu.edu/course/98/172/handin/ANDREWID/scratch`

- (c) **Mac OS X, Linux** Again, you already have the `scp` program.

To copy a file, `source/hello.txt` from your machine to the directory destination on `unix.andrew.cmu.edu`, you should use the command:

```
scp source/hello.txt andrew:destination
```

To copy a file, `source/world.txt` from `unix.andrew.cmu.edu` to the directory destination on your machine, you should use the command:

```
scp andrew:source/world.txt destination
```

Each of those commands will only copy one file. You will find that sometimes you want to copy an entire directory. You can do that by adding the “r” flag to your command. To copy the directory `important` from your machine to `inside folder` on `unix.andrew.cmu.edu`, you would use

```
scp -r important andrew:folder
```

To test it out, create a file called “interesting” which contains an interesting (or boring!) fact about yourself and copy it to the directory `/afs/andrew.cmu.edu/course/98/172/handin/ANDREWID/scratch`

## AFS

- (a) **What is AFS?** AFS is a distributed file system that was invented at CMU. As is customary for things invented here, we use it extensively. You have a “quota” of space and a home directory where you (and only you!) can put files. Your OS should be irrelevant at this point, as you should be using `unix.andrew.cmu.edu`
- (b) **Your Home Directory** When you `ssh` or `scp` to `unix.andrew.cmu.edu`, the directory you are put in will look something like `/afs/andrew.cmu.edu/usr8/adamblan`
- If you are `ssh`d, you can show the directory you are in by typing `pwd` (“Print Working Directory”) and hitting `enter`. The terminal will respond back with the name of the current directory.
- The directory you are in is important, because you can put things there that nobody else can look at; we will discuss AFS permissions in the next lecture. You will almost certainly do homework in your `afs` directory at some point.

## Basic Terminal Usage

- (a) `cd` (“change directory”) is a program that will let you move between directories. All of these examples are executed in the `scratch` directory that was mentioned before. If you would like to follow along, you should start by doing:

```
[adamblan@unix38 ~]$ cd /afs/andrew.cmu.edu/course/98/172/handin/adamblan/scratch
[adamblan@unix38 scratch]$
```

(Make sure you use your `andrewid` instead of “adamblan”)

- (b) `ls` (“list”) is a program that will list the files in the current directory. It takes several different “flags” or “options” which will give you different output. Here is an example of `ls` with no flags:

```
[adamblan@unix38 scratch]$ ls
i_am_a_file  move_it_here
```

The `l` flag makes `ls` list more information about the files. Take a look:

```
[adamblan@unix38 scratch]$ ls -l
total 3
-rw-r--r-- 1 adamblan games 22 Sep 3 11:25 i_am_a_file
drwxrwxrwx 2 adamblan games 2048 Sep 3 11:30 move_it_here
```

A few other flags to try out are `a` (`ls -a`) which makes `ls` also show hidden files and `R` (`ls -R`) which displays all sub-directories as well (“recursively”). You can combine flags too; so, `ls -laR` would make `ls` list all files recursively including hidden files.

- (c) **pwd** (“print working directory”) is a program that will tell you the name of the current directory, as we’ve already seen. Here’s a simple example:

```
[adamblan@unix38 scratch]$ pwd
/afs/andrew.cmu.edu/course/98/172/handin/adamblan/scratch
```

- (d) **echo** <what you want to echo>

is a program that will write whatever you want to your terminal. Take a look:

```
[adamblan@unix38 scratch]$ echo I am a practical computer scientist!
I am a practical computer scientist!
```

- (e) **cat** <filename> is a program that you should use to list the contents of a file. For instance:

```
[adamblan@unix38 scratch]$ cat i_am_a_file
These are my contents
```

You can also use it to “concatenate” two files together, but you will use it to see contents of files far more often.

- (f) **cp** <source file> <destination file> is a program that will “copy” files. It also supports the R flag (**cp -R** <source file> <destination file>) to copy a directory.

```
[adamblan@unix38 scratch]$ cp i_am_a_file im_the_same_file
[adamblan@unix38 scratch]$ ls
i_am_a_file  im_the_same_file  move_it_here
[adamblan@unix38 scratch]$ cat im_the_same_file
These are my contents
```

- (g) **mv** <source> <destination> is a program that you can use for two purposes. The first is to “move” a file from one place to another. The second is to rename a file. Here’s an example of both:

```
[adamblan@unix38 scratch]$ mv im_the_same_file move_it_here
[adamblan@unix38 scratch]$ ls
i_am_a_file  move_it_here
[adamblan@unix38 scratch]$ cd move_it_here/
[adamblan@unix38 move_it_here]$ ls
im_the_same_file
[adamblan@unix38 move_it_here]$ mv im_the_same_file new_name
[adamblan@unix38 move_it_here]$ cat new_name
These are my contents
```