Day 2: More Bash

Review

Unix System: Navigation

- **ls (path) – lists current directory contents**
  - **-a – show hidden files (names start with .)**
  - ***Optionally* takes one argument – a file path**

- **pwd – prints path to current location**
- **man (command) - prints displays command manual**
- **cd (path) – change directory to *path***
- **mv (object) (destination/name) – move object to destination, call it “name”**
- **cp (object) (destination/name) – copy object to destination, call it “name”**
- **mkdir (name) – make a directory called name**
- **rmdir (name) – remove a directory called name (must be empty or need -r)**
- **rm (name) – remove a file called name**

VIM – Viewing and Editing

- VI Ships with all UNIX systems as stock
- Activate it from the command line:
  - vim (filename) – opens filename with vim

- VI is a MODE based text editor – different modes do different things
- the escape key can be used to back out to the default mode at any time
- the i key activates insert mode – use to insert text in a location
- the arrow keys can be used to move around in either mode

- From the ESCAPE mode, [:w] saves whatever changes you have made, and [:q] quits vim. You can combine these with [:wq].

A simple BASH script

```
[bbm617@quser11 summerstudents]$ ls
[bbm617@quser11 summerstudents]$ vim myscript.sh
```

```
echo "hello world!"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

:wq to save and quit!

Run:

```
[bbm617@quser11 summerstudents]$ ls
[bbm617@quser11 summerstudents]$ vim myscript.sh
[bbm617@quser11 summerstudents]$ ls
myscript.sh
[bbm617@quser11 summerstudents]$ source myscript.sh
hello world!
[bbm617@quser11 summerstudents]$
```

Interacting w/ Files using Bash

cat (filename) – Display all contents of filename
tail -n number filename – Display last number lines of filename
head -n number filename – Display first number lines of filename

Appending output to files:

- **Use >> to append**
- **Use > to write new**

```
[bbm617@quser13 summerstudents]$ ls
[bbm617@quser13 summerstudents]$ echo "cd ../;" > myscript.sh
[bbm617@quser13 summerstudents]$ echo "pwd;" >> myscript.sh
[bbm617@quser13 summerstudents]$ cat myscript.sh
cd ../;
pwd;
[bbm617@quser13 summerstudents]$ pwd
/home/bbm617/simulations/summerstudents
[bbm617@quser13 summerstudents]$ source myscript.sh
/home/bbm617/simulations
[bbm617@quser13 simulations]$ pwd
/home/bbm617/simulations
[bbm617@quser13 simulations]$
```

```
cd ../;
pwd;
~
~
~
~
~
```

Control Structures

Suppose you need to make 10 directories labeled 0 through 9

```
[bbm617@quser13 summerstudents]$ mkdir 1
[bbm617@quser13 summerstudents]$ ls
1
[bbm617@quser13 summerstudents]$ mkdir 2
[bbm617@quser13 summerstudents]$ ls
1  2
[bbm617@quser13 summerstudents]$ mkdir 3
[bbm617@quser13 summerstudents]$ ls
1  2  3
[bbm617@quser13 summerstudents]$
```

This is clumsy and inefficient – but bash provides a way to make it fast

```
[bbm617@quser13 summerstudents]$ for number in {0..9};
do mkdir $number; done
[bbm617@quser13 summerstudents]$ ls
0  1  2  3  4  5  6  7  8  9
[bbm617@quser13 summerstudents]$
```