Tutorial 2

1 Shell Command Review

- **echo** - display a line of text
- **cd** - change directory
  - with no directory or ˜ returns to your home directory
  - with **-** returns to your previous directory
- **ls** - view files in the current/specified directory
  - with **-l** returns a long form listing of the files
  - with **-a** returns all files(including hidden files)
  - can combine multiple options, e.g. **ls -al**
- **pwd** - print the absolute path to the current directory
- **mkdir** - create a new directory
- **cp, mv, rm** - copy, move/rename, remove
  - with **-i** asks if you are sure
  - with **-fr** for **rm** forces the shell to remove a file/directory
- **cat/less** - print file contents to the terminal
- **diff** - compare two files
- **find** - search for files and directories
  - with **name pattern** restricts file names to globbing pattern
- **chgrp** - change the froup ownership of a file or directory
  - with **-R** recursively modifies the group of a directory
- **chmod** - change permissions of a file
  - with **-R** recursively modifies the group of a directory
  - **u** - user
  - **g** - group
  - **o** - other

  - **a** - all
  - **w** - write
  - **r** - read
  - **x** - execute
  - **+** - adds permissions
  - **-** - removes permissions

- **man** - prints information about commands

2 Alias

Aliases give convenient names to commands you use a lot.
Example:

```
alias mar138="marmoset_submit cs138"
```

3 Globbing

- ***** - matches 0 or more characters
- **?** - matches 1 character
- **{. . . }** - matches any alternative in the set
- [. . .] - matches 1 character in the set
- [**!**. . .] - matches 1 character not in the set

4 Input/Output redirection

- $<$ - reads input form a file reather than keyboard
- $>, 1 >, 2 >$ - (over)writes output/error to a file rather than screen
- $>>, 1 >>, 2 >>$ - appends output/error to a file rather than screen

5 Piplining

Pipes allow us to combine commands.
Example: suppose we want to count how many loops in all .cc files in **~/cs138/a1** directory. commands that we will be using:

-
  ```
  cat ~/cs138/a1/*.cc
  ```
  cats all .cc files’ contents

-
  ```
  egrep ’for|while’ ~/cs138/a1/*.cc
  ```
  finds all lines with for or while

-
  ```
  wc -l
  ```
  prints the number of lines

Put them together using pipeline,

```
cat ~/cs138/a1/*.cc | egrep ’for|while’ | wc -l
```

.
Or simplier,

```
egrep ’for|while’ ~/cs138/a1/*.cc | wc -l
```

6 Types of Quotes

6.1 Double Quotes

Protects everything except doublequote, backquote, and $VARS(suppresses globbing)

6.2 Single Quotes

Protects everything

7 Exercise

1. Sort all files(including hidden files) lexicographically

2.
   ```
   echo *
   echo ’*’
   echo "*"
   echo ’${HOME}’
   echo "${HOME}"
   echo ’"${HOME}"’
   echo "’${HOME}’"
   ```

3. Store all the names of .cc files to assignlist.txt
4. Find all .cc and .cpp files in current directory

8 Interesting commands

```
cowsay
cmatrix
```