`man`: manual `man -f [command name]`: find instruction page in manual

```
man echo
man ls
man cd
```

`echo`: displays a line of text `pwd`: print working directory `cd`: change directory `touch`: new file `mkdir`: make directory `rm`: remove file `rmdir`: remove directory `sleep`: suspend execution for an interval of time `ls`: list directory contents `ls -l`: verbose `cat`: dumps entire file to standard output, good for displaying short and simple files `less`: displays a file, allowing movement inside of it `head`: displays first ten lines of file `tail`: displays last ten lines of file `ps`: shows all current processes `top`: shows all users connected to server `kill` `cp`: copy a file

```
cp data.dat data2.dat
```

`mv`: move a file from one directory to another `wc`: counts lines, words, characters in file. `ssh`: used to securely log in to remote systems `scp`: used to copy files to or from remote systems to your system.

Permission levels

`r` means read only `w`: write permission `x`: execute permission in case of directory, `x` grants permission to list directory contents `a`: all `u`: users `(111)` `g`: group `(101)` `o`: owners `(000)` `chmod`: change access

```
chmod a+x hello_world.pl #gives access to all (users, group, others)
chmod u = rwx, g = rx, o = file.txt
chmod a-x file.txt
chmod g-x file.txt
```

`grep`: search files in a directory for a specific string

```
grep "hello world" *.txt
```

`diff`: compares two files

```
diff /dev/null hello.txt
```

- `/dev/null` is a special address -- it's always empty, and anything moved there is deleted.

Binary Permissions

r	w	x	#
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

```
chmod 764 # chmod(user, group, others)
          # user access: read, write, and execute
          # group access: read and write only
          # others access: read only
```

Running a job

- Make sure the program gas executable permissions
- use `./` to run the program

Changing file name when compiling

```
gcc hello_world.c -o hello_world
./hello_world
```

I/O Redirection ("piping")

- Programs can output to tother programs
- `program_a | program_b`
    - prog a's output becomes prog b's output

```
ls | wc # shows word count for what displays after running ls
ls -l | wc # shows word count for what displays after runnign ls -l
```

- `program_a > file.txt`
    - output of prog a gets written to file.txt
- `program_a < input.txt`
    - prog a gets its input from a input.txt

```
./aa_sequence.pl > sequence.txt
```

Environment Variables

`who`: shows currently connect users

```
who | wc -l # shows # of lines, aka # of users
```

`hostname`: displays hostname

```
hostname
hostname -d
hostname -i
```

Combining Commands

#linux #unix #cs3377 #CS #utd