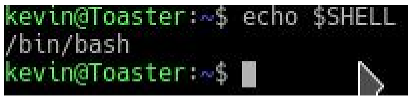
Kevin Kuan
CSE 337
Homework 3

Part 1: Simple UNIX Commands

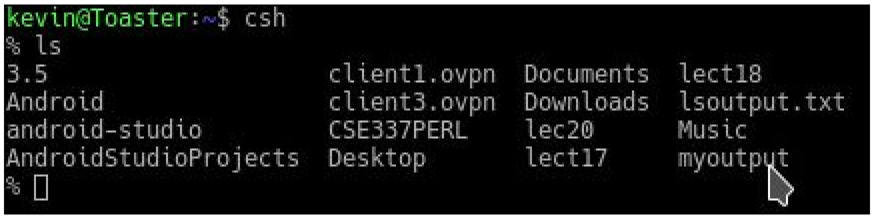
1)

a) My default login shell on my personal computer is bash



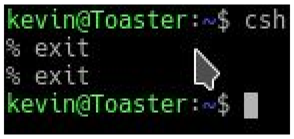
```
kevin@Toaster:~$ echo $SHELL
/bin/bash
kevin@Toaster:~$
```

b) I had to first use [sudo apt-get install csh] The command to invoke csh from my personal computer is to simply type the shell name. Alternatively I could type [/bin/csh].



```
kevin@Toaster:~$ csh
% ls
3.5                    client1.ovpn  Documents  lect18
Android                client3.ovpn  Downloads  lsoutput.txt
android-studio         CSE337PERL    lec20      Music
AndroidStudioProjects  Desktop       lect17     myoutput
%
```

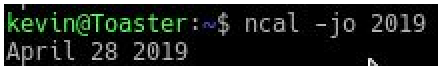
c) To terminate it, simply typed [exit]



```
kevin@Toaster:~$ csh
% exit
% exit
kevin@Toaster:~$
```

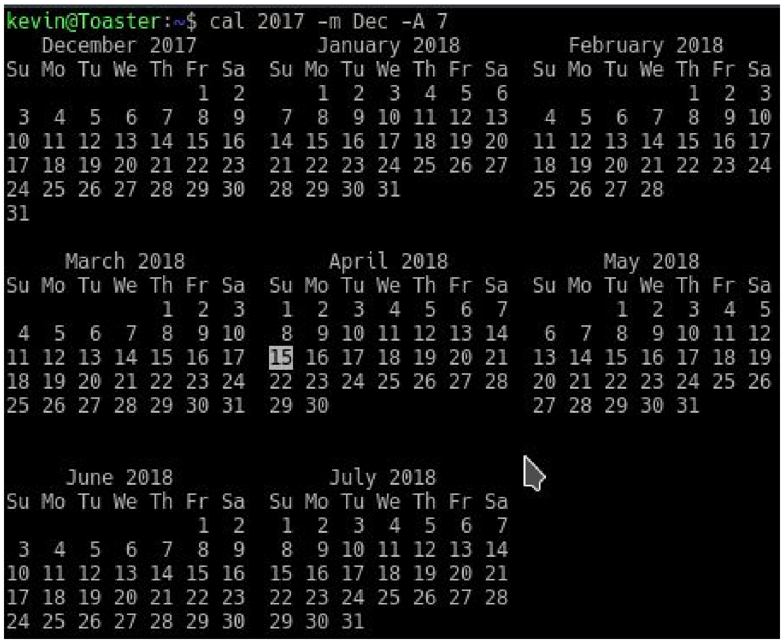
2)

a) [ncal -jo 2019]



```
kevin@Toaster:~$ ncal -jo 2019
April 28 2019
```

b) [cal 2017 -m DEc -A 7]

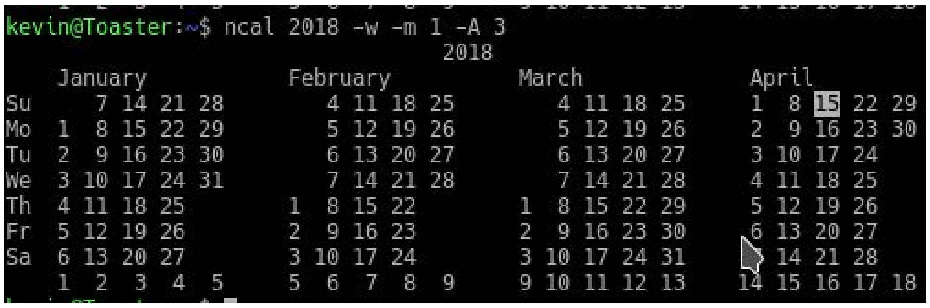


```
kevin@Toaster:~$ cal 2017 -m Dec -A 7
   December 2017          January 2018          February 2018
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2      1  2  3  4  5  6               1  2  3
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   4  5  6  7  8  9 10
10 11 12 13 14 15 16  14 15 16 17 18 19 20  11 12 13 14 15 16 17
17 18 19 20 21 22 23  21 22 23 24 25 26 27  18 19 20 21 22 23 24
24 25 26 27 28 29 30  28 29 30 31           25 26 27 28
31

     March 2018            April 2018            May 2018
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
             1  2  3   1  2  3  4  5  6  7         1  2  3  4  5
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   6  7  8  9 10 11 12
11 12 13 14 15 16 17  15 16 17 18 19 20 21  13 14 15 16 17 18 19
18 19 20 21 22 23 24  22 23 24 25 26 27 28  20 21 22 23 24 25 26
25 26 27 28 29 30 31  29 30                 27 28 29 30 31

     June 2018             July 2018
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1  2   1  2  3  4  5  6  7
 3  4  5  6  7  8  9   8  9 10 11 12 13 14
10 11 12 13 14 15 16  15 16 17 18 19 20 21
17 18 19 20 21 22 23  22 23 24 25 26 27 28
24 25 26 27 28 29 30  29 30 31
```

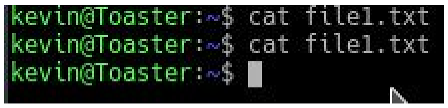
c) [ncal 2018 -w -m 1 -A 3]



```
kevin@Toaster:~$ ncal 2018 -w -m 1 -A 3
                                  2018
    January            February           March              April
Su     7 14 21 28         4 11 18 25         4 11 18 25      1  8 15 22 29
Mo  1  8 15 22 29         5 12 19 26         5 12 19 26      2  9 16 23 30
Tu  2  9 16 23 30         6 13 20 27         6 13 20 27      3 10 17 24
We  3 10 17 24 31         7 14 21 28         7 14 21 28      4 11 18 25
Th  4 11 18 25         1  8 15 22         1  8 15 22 29      5 12 19 26
Fr  5 12 19 26         2  9 16 23         2  9 16 23 30      6 13 20 27
Sa  6 13 20 27         3 10 17 24         3 10 17 24 31      7 14 21 28
    1  2  3  4  5      5  6  7  8  9      9 10 11 12 13     14 15 16 17 18
```

3)

a) One way is [touch ~/file1.txt]. The second way is [cat > ~/file2.txt] but then hit ctrl+c. I then use cat to display if anything was written in the file.



```
kevin@Toaster:~$ cat file1.txt
kevin@Toaster:~$ cat file1.txt
kevin@Toaster:~$ 
```

b) [wget -O ~/googleindex.html http://google.com] and I verified by checking the file in vim[vim ~/googleindex.html] i can either use the command [ls] to check that file was created and use cat [cat ~/googleindex.html] to show the contents.

4)

a) The absolute paths are [/home/kevin/file1.txt] and [/home/kevin/file2.txt]. This can be found by [readlink -f file1.txt].

b) It is the same, it would be [/home/kevin/file1.txt] and [/home/kevin/file2.txt] for both the files.

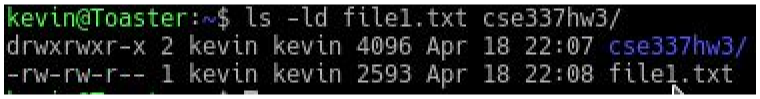
Part 2: UNIX File System

1)

a) To change into my home directory, [cd ~/].

b) I made a directory with [mkdir cse337hw3] and then made a file with the commands [ls -l > file1.txt]

c) To check the permissions [ls -ld file1.txt cse337hw3/]



```
kevin@Toaster:~$ ls -ld file1.txt cse337hw3/
drwxrwxr-x 2 kevin kevin 4096 Apr 18 22:07 cse337hw3/
-rw-rw-r-- 1 kevin kevin 2593 Apr 18 22:08 file1.txt
```

This means that the directory cse337hw3 can be read,written, and executable by the owner and groups. However, everyone else can only read or write. For the file1.txt, it can only be read and written by the owner and group. It can only be read by everyoneelse.

d) For a directory with rw-r-----, it means that the owner of the directory can read and write. The group can only read. Everyone else cannot do anything. For a file rwxrw-r-- it means that the owner can read,write, and execute. The group can only read and write. Everyone else can only read.

2)

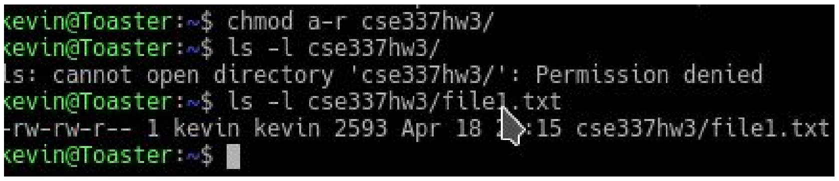
a) [cd cse337hw3]

b) [cp ~/file1.txt cse337hw3/]

c) [rm ~/file.txt]
d) [mkdir /cse337hw3/sub_dir]

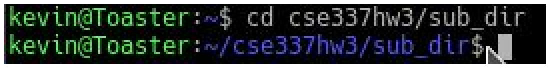
3)

a) [cd ~/]
b) [chmod a-r cse337hw3] I then checked with the commands below:



```
kevin@Toaster:~$ chmod a-r cse337hw3/
kevin@Toaster:~$ ls -l cse337hw3/
ls: cannot open directory 'cse337hw3/': Permission denied
kevin@Toaster:~$ ls -l cse337hw3/file1.txt
-rw-rw-r-- 1 kevin kevin 2593 Apr 18 1 :15 cse337hw3/file1.txt
kevin@Toaster:~$
```

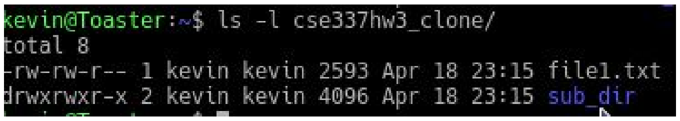
c) Yes i am by [cd cse337hw3/sub_dir]



```
kevin@Toaster:~$ cd cse337hw3/sub_dir
kevin@Toaster:~/cse337hw3/sub_dir$
```

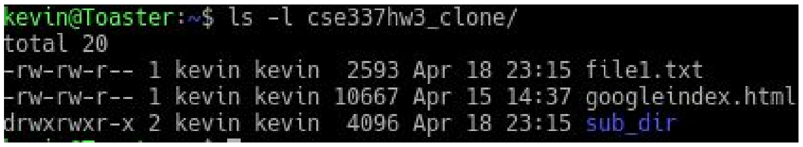
4)

a) I added the permissions back by [chmod a+r cse337] and clone it by [cp cse337hw3 cse337hw3_clone]. I use [ls -l cse337hw3_clone] to check the contents.



```
kevin@Toaster:~$ ls -l cse337hw3_clone/
total 8
-rw-rw-r-- 1 kevin kevin 2593 Apr 18 23:15 file1.txt
drwxrwxr-x 2 kevin kevin 4096 Apr 18 23:15 sub_dir
```

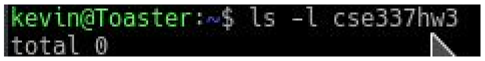
b) [mv ~/googleindex.html cse337hw3_clone]



```
kevin@Toaster:~$ ls -l cse337hw3_clone/
total 20
-rw-rw-r-- 1 kevin kevin  2593 Apr 18 23:15 file1.txt
-rw-rw-r-- 1 kevin kevin 10667 Apr 15 14:37 googleindex.html
drwxrwxr-x 2 kevin kevin  4096 Apr 18 23:15 sub_dir
```

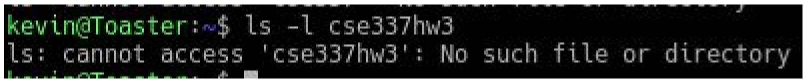
5)

a) [rm ~/cse337hw3/*]



```
kevin@Toaster:~$ ls -l cse337hw3
total 0
```

b) [rmdir ~/cse337hw3]



```
kevin@Toaster:~$ ls -l cse337hw3
ls: cannot access 'cse337hw3': No such file or directory
```

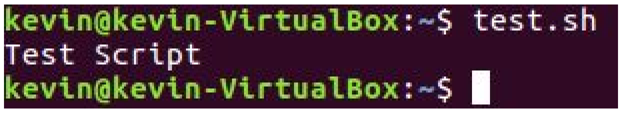
Part 3: UNIX Shell Utilities

1) [sed -n '10,$p' input|sed '/cat/s/dog/fly/'] means to return lines 10 to the last one inclusive from the file called input and then from there if a line contains cat, replace the first dog after it to fly and output it.
 I created a text file and filled lines 10 to 20 with combinations for cat and dogs to tes tthis.

2) [head -n 17 <filename> | tail -1] . I tested this by creating a text file numbered each line to 20. It outputted 17. From this command.

3) To add a directory to my $PATH directory, I do [PATH=~/cse337hw3:$PATH] where ~/cse337hw3 is the directory I want to add to the path. I also made a simple bash script that echoes text.I then tested it by using [echo $PATH | grep cse337] this highlights the cse337 from

echoing the $PATH variable. Next I go to my home directory (one level up from cse337hw3) and type [test.sh]. This is useful as it allows one to run a script no matter what the current working directory is by simply typing the name of the script like below.



```
kevin@kevin-VirtualBox:~$ test.sh
Test Script
kevin@kevin-VirtualBox:~$
```

4) [ls -lp ~/ | grep -v /] Here, ls -lp displays everything as a list from the home directory and if it is a directory, it adds a "/" to the end of it and then passes it into grep -v where it omits anything with a "/".

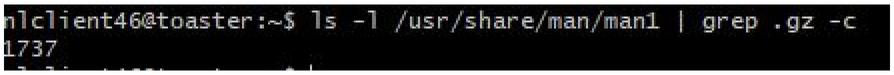
By adding using the -l command, i can check if a file has a "d" in front of the permissions to designate a directory, a file has a "-" before the permission bits.

5) using [crontab -e] it opens the configurations. I then can add a job specified in the instructions with this syntax [32   6   7,14,21,28   *   *   touch ~/update.txt]

I checked if there is a file called update.txt made the next day with ls -l update.txt. This also shows when it was made.

6)
   a) [mkdir ~/A4tmp]
   b) Find /var/log -type f -size +2000c 2> ~/A4tmp/logerror.txt
      I then used [cat ~/A4tmp/logerro.txt] and it displays the error messages when using the find as it cant access some directories without root access

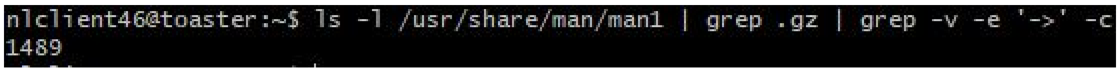
7)
   a) using [ls -l /usr/share/man/man1 | grep .gz -c], I get **1738**. Here I redirected the output of ls -l <directory> into grep to matching ".gz" and the -c means to count



```
nlclient46@toaster:~$ ls -l /usr/share/man/man1 | grep .gz -c
1737
```

      I can manually remove the -c flag to see the results

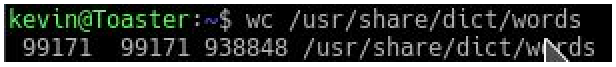
   b) I used [ls -l /usr/share/man/man1 | grep .gz | grep -v -e '->' -c] and got 1489. Here I piped the ones with .gz extension into another grep where the -v does matches without the '->' to show symbolic link and the -c counts it.



```
nlclient46@toaster:~$ ls -l /usr/share/man/man1 | grep .gz | grep -v -e '->' -c
1489
```

      I can remove the "-c" to manually see the results

8)
   a) Using [wc /usr/share/dict/words] I get the following below:



```
kevin@Toaster:~$ wc /usr/share/dict/words
 99171  99171 938848 /usr/share/dict/words
```

This means that there are 99171 new lines, 99171 words, and 938848 characters

b) There is one word per line, hence the same number for words and new lines
c) [sed -n ‘100,200p’ /usr/share/dict/words > ~/A4tmp/100-200.txt] and [sed -n ‘400,600p’ /usr/share/dict/words > ~/A4tmp/400-600.txt]

   I can verify this buy manually check the 100th, 200th, 400th, and 600th lines and compare it to my txt files. I can also use [wc -l <file>] to count how many lines there are. 100-200.txt has 101 lines as it is inclusive and 400-600.txt has 201.