Assignment 1
Write up describing working of the shell

Internal Commands:

pwd
Prints the name of the current working directory.
Test Case

```
  arnav@rogstrix in ~/Assignment1 on  master [?] via C v1
  λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell
Welcome to Arnav's shell
>> pwd
/home/arnav/Assignment1
>> pwd -P
/home/arnav/Assignment1
>> pwd -L
/home/arnav/Assignment1
>>
```

Options:
-P:(default) avoids all symlinks
-L:use Pwd from the environment even if it contains   symlinks
If no option is specified then -P is taken as the default.

cd

Changes the current working directory.

Test Case

```
 ┌arnav@rogstrix in ~/Assignment1 on  master [?] via C v12.2.0-gcc took 13ms
○└λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell2 && "/home/arnav/As
 Welcome to Arnav's shell
 >> pwd
 /home/arnav/Assignment1
 >> cd ..

 >> pwd
 /home/arnav
 >> cd Assignment1

 >> cd a
 Specify a valid directory: No such file or directory

 >> cd

 >> pwd
 /home/arnav
 >>
```

Options:

.. : This is used to take the user one directory back.

--help: Prints out information regarding the cd command and options supported by the shell.

Takes to the home directory in case no directory has been specified.

Errors handled:

Incorrect Directory Specified

echo

To display a line of text.

Test Case

```
 ┌arnav@rogstrix in ~/Assignment1 on  master [?] via C v12.2.0-gcc took 18ms
 └λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell2 && "/home/arnav/Assignment1/"shell2
Welcome to Arnav's shell
>> echo this shell was tough
this shell was tough

>> echo this assignment was tough
this assignment was tough

>> echo -n this assignment
this assignment
>> echo --help
echo - display a line of text
-n     do not output the trailing newline
--help display this help and exit
>> █
```

Options:

--help: To display information regarding the echo command and options that the shell supports.

-n: Using this flag does not put a newline at the end of the line.

Note: Backslash escapes are disabled.

exit

Simply exits the shell/ends the program.

```
 ┌arnav@rogstrix in ~/Assignment1 on  master [?] via C v12.2.0-gcc took 17ms
 └λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell2 && "/home/arnav/Assignment1/"shell2
Welcome to Arnav's shell
>> exit
```

External Commands

cat

Concatenate files and print those files on stdout. Multiple files can be specified at once.

Test Case

```
>> cat a.c
A file of this name does not exist in current working directory
>> cat -n -E ls.c
1 #include <stdio.h> $
2 #include <stdlib.h> $
3 #include <dirent.h> $
4 #include <errno.h> $
5 #include <string.h> $
6 #include <inttypes.h> $
7  $
8 int main(int argc,char **argv){ $
9     struct dirent *d; $
10     int a_flag=0; $
11     int A_flag=0; $
12     DIR *dir; $
13     char *X=argv[1]; $
14     if(argc==1){ $
15         dir=opendir("."); $
16     } $
17     else if(argc>=2 && X[0]=='-'){ $
18         dir=opendir("."); $
19         int i=1; $
20         while(i<argc){ $
21             char *q=argv[i]; $
22             if(q[1]=='a'){ $
23                 a_flag=1; $
```

Options:

-n: Prints out line numbers of each file specified under cat.

-E: Prints out a dollar sign ($) at the end of each line.

Errors handled:

File specified does not exist.

File specified is actually a directory.

File does not have read permissions.

rm

To remove files or directories. Multiple files can be specified.

Test Case

```
  ┌arnav@rogstrix in ~/Assignment1 on  master [?] via C v1
○ └λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell
  Welcome to Arnav's shell
  >> rm cat ls mkdir date
  /home/arnav/Assignment1/cat
  The given file has been removed
  /home/arnav/Assignment1/ls
  The given file has been removed
  /home/arnav/Assignment1/mkdir
  The given file has been removed
  /home/arnav/Assignment1/date
  The given file has been removed
  >> rm -d arnav
  /home/arnav/Assignment1/arnav
  The given directory has been removed
  >> rm a
  /home/arnav/Assignment1/a
  File does not exist
  >> ▐
```

Options

*-f:*forced delete; delete nonexistent files and directories without prompt.

-d: remove directories that are empty

Errors handled

File specified does exist.

Directory specified instead of file. -d flag is not set.

Directory specified is not empty.

<u>date</u>

Print system date and time.

Test Case

```
  arnav@rogstrix in ~/Assignment1 on  master [?] via C v12.2.0-gcc
  λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell2 && "/home
Welcome to Arnav's shell
>> date
Tue Oct 25 18:17:56 2022
>> date -u
Tue Oct 25 12:48:00 2022
>> date -rls.c
Tue Oct 25 17:55:37 2022

>> date -r=l.c
No need to add = sign
>> date -rl.c
l.c
No such file or directory
>>
```

Options

-u: Prints out the UTC(Coordinated) time.

-r[file_name]: prints the last date a file was modified.

Errors handled

Equal to sign added between -r and file name.

File name specified in -r option does not exist.

mkdir

Make directories.

Test Case

```
 ┌ arnav@rogstrix in ~/Assignment1 on  master [?] via C v12.2.0-gcc t
○└λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell2 && "/home
  Welcome to Arnav's shell
  >> mkdir arnav
  >> mkdir -v arn
  Directory created  /home/arnav/Assignment1/arn
  >> mkdir -m=r arna
  >> []
```

Options

-v: Prints the message that the directory has created.

-m: Creates the directory with certain permissions.Permissions like read(r),write(w) and execute(x) are supported.

Note: In -m r/w/x would set up permissions for each user,group and owner. Numbers like 0400 for read/0751/0521 not accepted. Moreover only one mode bit allowed at a time.

Errors handled

Directory already exists.

Incorrect mode under -r option. (Neither r/x/w)

Name of directory to be created not specified.

Ls

List all contents of the current directory or directory specified.

Test Case

```
 ┌─arnav@rogstrix in ~/Assignment1 on  master [?] via C v12.2.0-gcc took 14ms
○└─λ cd "/home/arnav/Assignment1/" && gcc shell2.c -o shell2 && "/home/arnav/Assignment1/"shell2
Welcome to Arnav's shell
>> cd ..

>> ls Assignment1
 shell.c  ls.c  rm.c  shell2.c  threads.c  mkdir.c  cat.c  date.c  Makefile  threads  rm  cat  date  arnav  arn  arna
 mkdir  ls  shell2
>> ls -a
 .  ..  .bash_logout  .bash_profile  .config  .kde4  .local  .profile  .xinitrc  Desktop  Downloads  Templates  Publi
c  Documents  Music  Pictures  Videos  .cache  .bashrc  .wget-hsts  .vscode-oss  .pki  .firedragon  .java  IdeaProjec
ts  VirtualBox VMs  .android  .git  a.asm  a.o  a  r.asm  Makefile  assignment.asm  chaitanya.asm  chaitanya.o  chait
anya  assignment  r.o  r  backup.o  backup  assignment1.asm  assignment.1  assignment1.o  assignment1  h_loop(void)
;  tempCodeRunnerFile.c  a.out  test  ls  shell  Assignment1  b  .viminfo  .Xauthority  .gtkrc-2.0  .bash_history
>> ls -A
 .bash_logout  .bash_profile  .config  .kde4  .local  .profile  .xinitrc  Desktop  Downloads  Templates  Public  Docu
ments  Music  Pictures  Videos  .cache  .bashrc  .wget-hsts  .vscode-oss  .pki  .firedragon  .java  IdeaProjects  Vir
tualBox VMs  .android  .git  a.asm  a.o  a  r.asm  Makefile  assignment.asm  chaitanya.asm  chaitanya.o  chaitanya  a
ssignment  r.o  r  backup.o  backup  assignment1.asm  assignment.1  assignment1.o  assignment1  h_loop(void)  ;  temp
CodeRunnerFile.c  a.out  test  ls  shell  Assignment1  b  .viminfo  .Xauthority  .gtkrc-2.0  .bash_history
>>
```

Options:

-a: also list out directories and files starting with '.'(hidden files).

-A: list out all directories and files including those hidden except for '.' and '..'.

Directory_name: List out all the files and directories in the specified directory.

Errors handled

Directory specified does not exist.

Using threads

External Commands like cat,ls,date,mkdir and rm can be executed using threads by adding &t at the end of the command. &t must be the last option in the command, if it isn't the shell throws an error.

```
>> mkdir -v arnav &t
Directory created  /home/arnav/Assignment1/arnav
>> rm -d arnav &t
/home/arnav/Assignment1/arnav
The given directory has been removed
>> date -u &t
Wed Oct 26 00:23:30 2022
>>
```