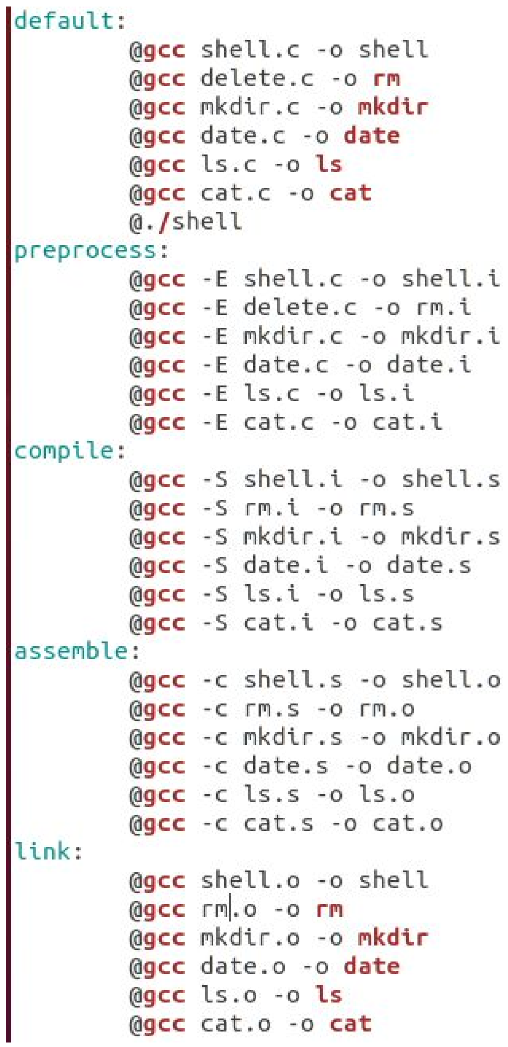
OS Assignment 1 (Ques1)

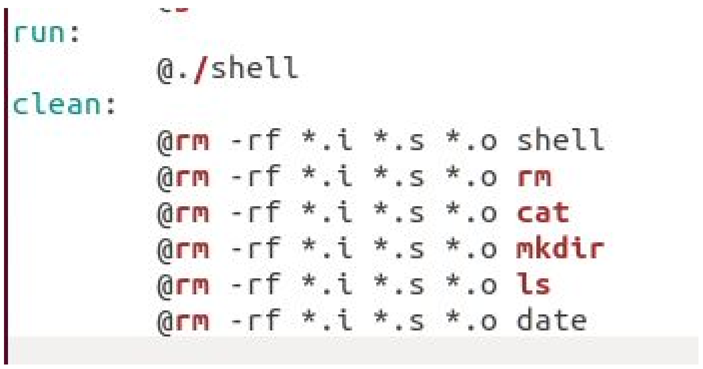
Gitansh Raj Satija 2019241 CSE

Documentation:

<u>Makefile:</u>



```
default:
	@gcc shell.c -o shell
	@gcc delete.c -o rm
	@gcc mkdir.c -o mkdir
	@gcc date.c -o date
	@gcc ls.c -o ls
	@gcc cat.c -o cat
	@./shell
preprocess:
	@gcc -E shell.c -o shell.i
	@gcc -E delete.c -o rm.i
	@gcc -E mkdir.c -o mkdir.i
	@gcc -E date.c -o date.i
	@gcc -E ls.c -o ls.i
	@gcc -E cat.c -o cat.i
compile:
	@gcc -S shell.i -o shell.s
	@gcc -S rm.i -o rm.s
	@gcc -S mkdir.i -o mkdir.s
	@gcc -S date.i -o date.s
	@gcc -S ls.i -o ls.s
	@gcc -S cat.i -o cat.s
assemble:
	@gcc -c shell.s -o shell.o
	@gcc -c rm.s -o rm.o
	@gcc -c mkdir.s -o mkdir.o
	@gcc -c date.s -o date.o
	@gcc -c ls.s -o ls.o
	@gcc -c cat.s -o cat.o
link:
	@gcc shell.o -o shell
	@gcc rm.o -o rm
	@gcc mkdir.o -o mkdir
	@gcc date.o -o date
	@gcc ls.o -o ls
	@gcc cat.o -o cat
```



```
run:
	@./shell
clean:
	@rm -rf *.i *.s *.o shell
	@rm -rf *.i *.s *.o rm
	@rm -rf *.i *.s *.o cat
	@rm -rf *.i *.s *.o mkdir
	@rm -rf *.i *.s *.o ls
	@rm -rf *.i *.s *.o date
```

Commands and the flags implemented:

- cd (-L -P)
  cd Changes the directory

- echo (-n -E)
  echo prints the message after the command
  -n prevents generating a newline after executing

  -E prevents the interpretation of \ special characters
- pwd (-L -P)
  pwd gives the directory
  -P avoids symlinks
  -L also works for symlinks
- history (-c -d)
  history tells us the history of the shell
  -c clears the history
  -d takes an integer argument and deletes the history present there.
- exit
  exit terminates the shell
- ls (-a -l)
  ls shows files stored in a directory
  -a also shows hidden files (which start with '.')
  -l shows file using long listing format
- mkdir (-m -v)
  mkdir makes the directory
  -m or -m= sets the mode of the file to be created
  -v prompts a message after creating every directory
- cat (-T -E)
  Cat displays the content of the file
  -E puts a $ sign at the end of every line
  -T replaces each tab with ^I
- rm (-d -v)
  Rm deletes the file/directory
  -d deletes empty directories
  -v prompts a message for every removal
- Date (-u -R)
  Date displays the directory
  -u outputs date in UTC format
  -R outputs the date in RFC 5322 format

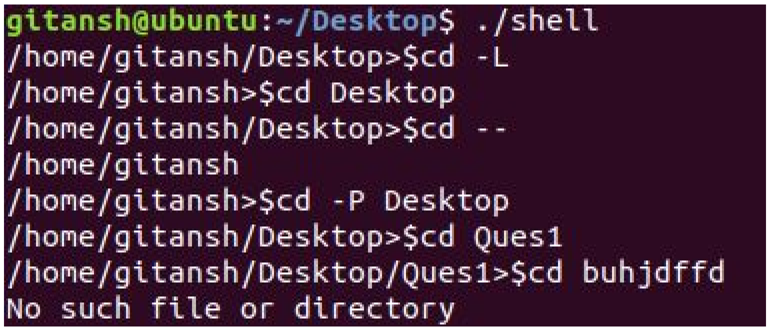
The shell contains 6 programs, 1 main program which contains the main function and takes the input. All the internal commands are within the main program. The 5 external commands are made in separate program files. For handling the external commands we use execvp(), fork() and wait() commands. A sample code for it is

```
pid_t id=fork();
        if (id == 0)
        {
           char *args[] = {"./date", tt, NULL};
           execvp("./date", args);
           exit(0);
        }
        else if (id>0){
              int st;
              pid_t cpid;
           cpid = wait(&st);
           if(cpid<0)
              perror("wait");
        }
        else
        {
           perror("fork");
        }
```

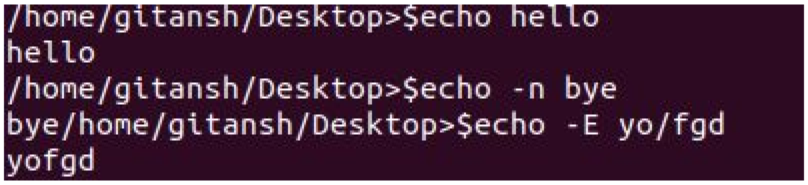
We use fork() and wait() to create a child process which helps in executing the external command with the help of execvp() function which passes the executable file and arguments to the program from external command which are received in the *argv[]. We then process accordingly.

Here are some sample input/output of the shell:

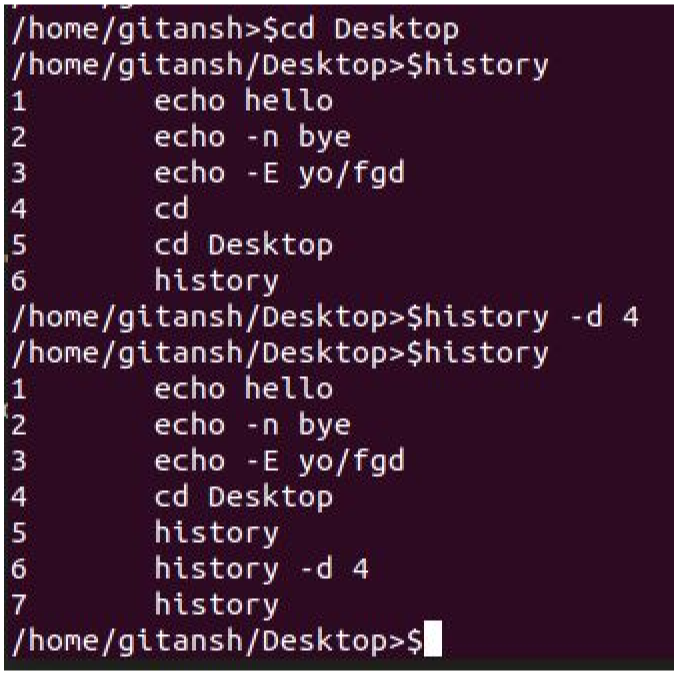
Assumptions: integer followed by -d is more than 0
Single enter is used to terminate the program
Command line options should be placed correctly



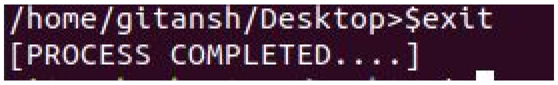
```
gitansh@ubuntu:~/Desktop$ ./shell
/home/gitansh/Desktop>$cd -L
/home/gitansh>$cd Desktop
/home/gitansh/Desktop>$cd --
/home/gitansh
/home/gitansh>$cd -P Desktop
/home/gitansh/Desktop>$cd Ques1
/home/gitansh/Desktop/Ques1>$cd buhjdffd
No such file or directory
```



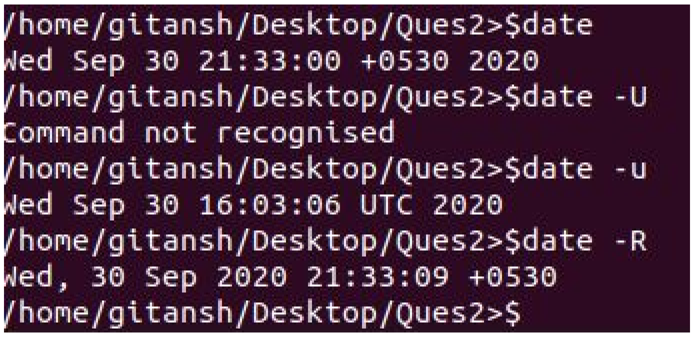
```
/home/gitansh/Desktop>$echo hello
hello
/home/gitansh/Desktop>$echo -n bye
bye/home/gitansh/Desktop>$echo -E yo/fgd
yofgd
```



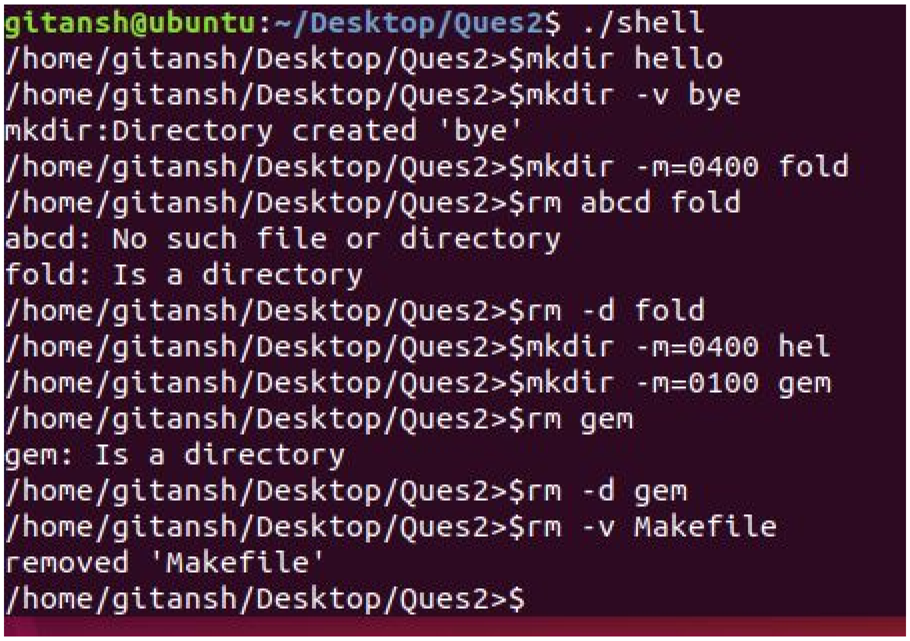
```
/home/gitansh>$cd Desktop
/home/gitansh/Desktop>$history
1       echo hello
2       echo -n bye
3       echo -E yo/fgd
4       cd
5       cd Desktop
6       history
/home/gitansh/Desktop>$history -d 4
/home/gitansh/Desktop>$history
1       echo hello
2       echo -n bye
3       echo -E yo/fgd
4       cd Desktop
5       history
6       history -d 4
7       history
/home/gitansh/Desktop>$
```



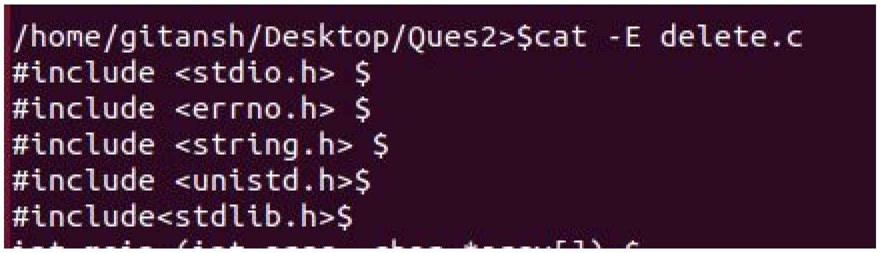
```
/home/gitansh/Desktop>$exit
[PROCESS COMPLETED....]
```



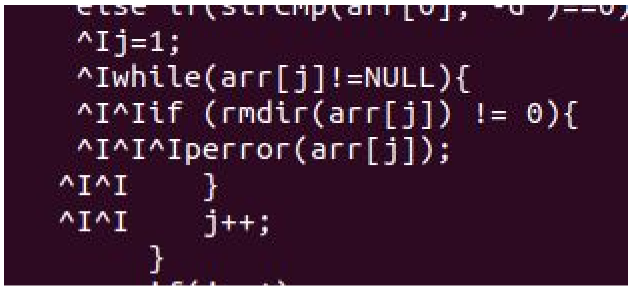
```
/home/gitansh/Desktop/Ques2>$date
Wed Sep 30 21:33:00 +0530 2020
/home/gitansh/Desktop/Ques2>$date -U
Command not recognised
/home/gitansh/Desktop/Ques2>$date -u
Wed Sep 30 16:03:06 UTC 2020
/home/gitansh/Desktop/Ques2>$date -R
Wed, 30 Sep 2020 21:33:09 +0530
/home/gitansh/Desktop/Ques2>$
```



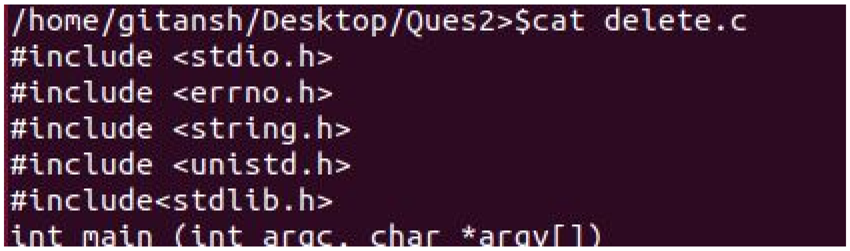
```
gitansh@ubuntu:~/Desktop/Ques2$ ./shell
/home/gitansh/Desktop/Ques2>$mkdir hello
/home/gitansh/Desktop/Ques2>$mkdir -v bye
mkdir:Directory created 'bye'
/home/gitansh/Desktop/Ques2>$mkdir -m=0400 fold
/home/gitansh/Desktop/Ques2>$rm abcd fold
abcd: No such file or directory
fold: Is a directory
/home/gitansh/Desktop/Ques2>$rm -d fold
/home/gitansh/Desktop/Ques2>$mkdir -m=0400 hel
/home/gitansh/Desktop/Ques2>$mkdir -m=0100 gem
/home/gitansh/Desktop/Ques2>$rm gem
gem: Is a directory
/home/gitansh/Desktop/Ques2>$rm -d gem
/home/gitansh/Desktop/Ques2>$rm -v Makefile
removed 'Makefile'
/home/gitansh/Desktop/Ques2>$
```



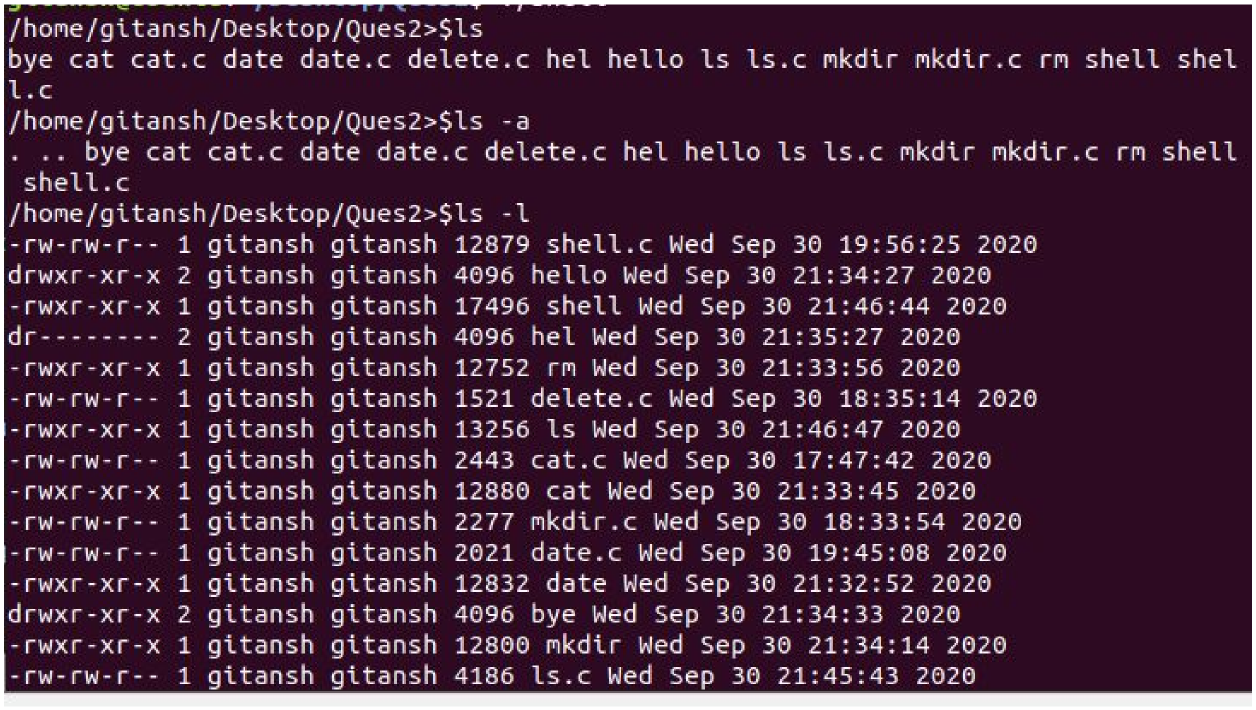
```
/home/gitansh/Desktop/Ques2>$cat -E delete.c
#include <stdio.h> $
#include <errno.h> $
#include <string.h> $
#include <unistd.h>$
#include<stdlib.h>$
int main (int argc, char *argv[]) {
```



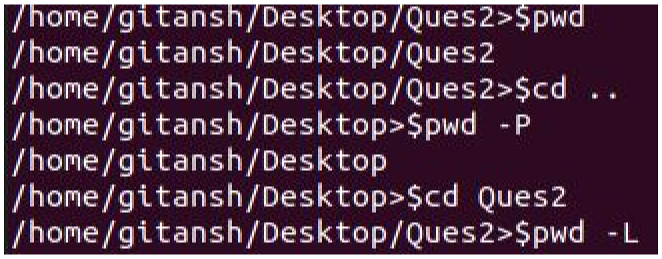
```
    else if(strcmp(arr[0], "-d")==0)
    ^Ij=1;
    ^Iwhile(arr[j]!=NULL){
    ^I^Iif (rmdir(arr[j]) != 0){
    ^I^I^Iperror(arr[j]);
   ^I^I    }
   ^I^I    j++;
        }
```



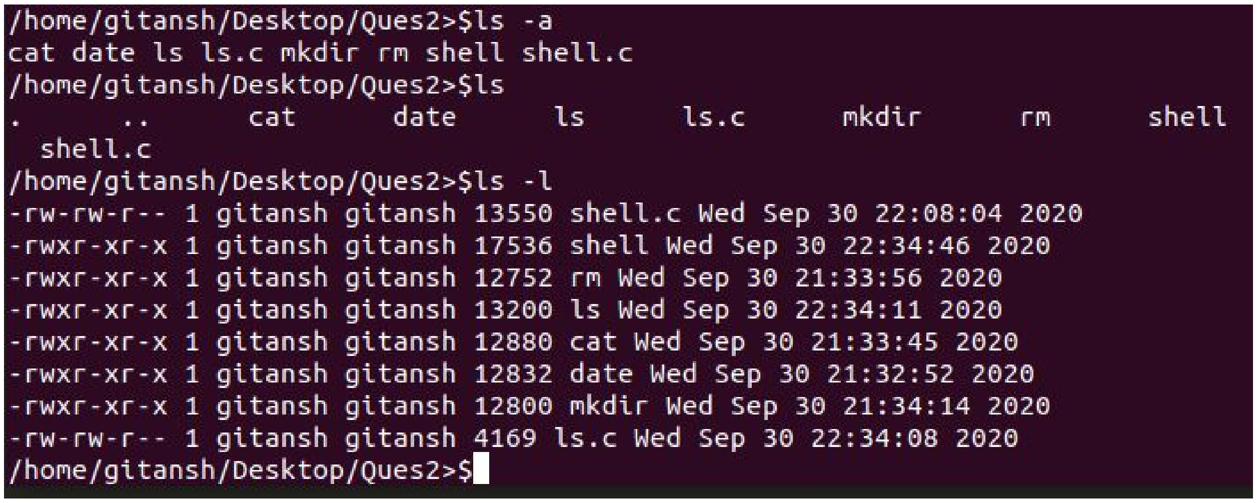
```
/home/gitansh/Desktop/Ques2>$cat delete.c
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <unistd.h>
#include<stdlib.h>
int main (int argc, char *argv[])
```



```
/home/gitansh/Desktop/Ques2>$ls
bye cat cat.c date date.c delete.c hel hello ls ls.c mkdir mkdir.c rm shell shel
l.c
/home/gitansh/Desktop/Ques2>$ls -a
. .. bye cat cat.c date date.c delete.c hel hello ls ls.c mkdir mkdir.c rm shell
 shell.c
/home/gitansh/Desktop/Ques2>$ls -l
-rw-rw-r-- 1 gitansh gitansh 12879 shell.c Wed Sep 30 19:56:25 2020
drwxr-xr-x 2 gitansh gitansh 4096 hello Wed Sep 30 21:34:27 2020
-rwxr-xr-x 1 gitansh gitansh 17496 shell Wed Sep 30 21:46:44 2020
dr-------- 2 gitansh gitansh 4096 hel Wed Sep 30 21:35:27 2020
-rwxr-xr-x 1 gitansh gitansh 12752 rm Wed Sep 30 21:33:56 2020
-rw-rw-r-- 1 gitansh gitansh 1521 delete.c Wed Sep 30 18:35:14 2020
-rwxr-xr-x 1 gitansh gitansh 13256 ls Wed Sep 30 21:46:47 2020
-rw-rw-r-- 1 gitansh gitansh 2443 cat.c Wed Sep 30 17:47:42 2020
-rwxr-xr-x 1 gitansh gitansh 12880 cat Wed Sep 30 21:33:45 2020
-rw-rw-r-- 1 gitansh gitansh 2277 mkdir.c Wed Sep 30 18:33:54 2020
-rw-rw-r-- 1 gitansh gitansh 2021 date.c Wed Sep 30 19:45:08 2020
-rwxr-xr-x 1 gitansh gitansh 12832 date Wed Sep 30 21:32:52 2020
drwxr-xr-x 2 gitansh gitansh 4096 bye Wed Sep 30 21:34:33 2020
-rwxr-xr-x 1 gitansh gitansh 12800 mkdir Wed Sep 30 21:34:14 2020
-rw-rw-r-- 1 gitansh gitansh 4186 ls.c Wed Sep 30 21:45:43 2020
```



```
/home/gitansh/Desktop/Ques2>$pwd
/home/gitansh/Desktop/Ques2
/home/gitansh/Desktop/Ques2>$cd ..
/home/gitansh/Desktop>$pwd -P
/home/gitansh/Desktop
/home/gitansh/Desktop>$cd Ques2
/home/gitansh/Desktop/Ques2>$pwd -L
```



```
/home/gitansh/Desktop/Ques2>$ls -a
cat date ls ls.c mkdir rm shell shell.c
/home/gitansh/Desktop/Ques2>$ls
.      ..      cat      date      ls      ls.c      mkdir      rm      shell
  shell.c
/home/gitansh/Desktop/Ques2>$ls -l
-rw-rw-r-- 1 gitansh gitansh 13550 shell.c Wed Sep 30 22:08:04 2020
-rwxr-xr-x 1 gitansh gitansh 17536 shell Wed Sep 30 22:34:46 2020
-rwxr-xr-x 1 gitansh gitansh 12752 rm Wed Sep 30 21:33:56 2020
-rwxr-xr-x 1 gitansh gitansh 13200 ls Wed Sep 30 22:34:11 2020
-rwxr-xr-x 1 gitansh gitansh 12880 cat Wed Sep 30 21:33:45 2020
-rwxr-xr-x 1 gitansh gitansh 12832 date Wed Sep 30 21:32:52 2020
-rwxr-xr-x 1 gitansh gitansh 12800 mkdir Wed Sep 30 21:34:14 2020
-rw-rw-r-- 1 gitansh gitansh 4169 ls.c Wed Sep 30 22:34:08 2020
/home/gitansh/Desktop/Ques2>$
```

Error Handling:
We make use of many system calls in our program and more often than not the calls are unsuccessful. Hence, we need to handle these errors.

For this we make use of the perror() function which is equivalent to printf("%s", strerror(errno)) and handles any errors in system calls.

Corner Cases Covered :-
The code is prepared to defend against all types of errors that might generate through unsuccessful system calls or invalid input types. Below I have mentioned 2 errors for each command (as mentioned in the question statement)

1. cd (chdir system call)
   Search permission is denied for any component of the pathname.
   A component of the pathname is not a directory.

2. echo (printing system calls)
   The write operation was terminated due to the receipt of a signal, and no data was transferred.
   A wide-character code that does not correspond to a valid character has been detected.

3. history
   Insufficient storage space is available.
   An attempt is made to write to a pipe or FIFO that is not open for reading by any process. A SIGPIPE signal will also be sent to the thread.

4. pwd (getcwd and getenv system calls)
   Search permission was denied for the current directory, or read or search permission was denied for a directory above the current directory in the file hierarchy.
   The *size* argument is 0.
   Insufficient storage space is available.

5. exit (exit system call)
   No errors are defined

6. ls (scandir system call)
   A loop exists in symbolic links encountered during resolution of the *dir* argument.
   A component of *dir* names an existing file that is neither a directory nor a symbolic link to a directory.

7. Cat (open system call)
   Search permission is denied on a component of the path prefix, or the file exists and the permissions specified by *oflag* are denied
   The implementation does not support synchronized I/O for this file.

8. date (time system call)
   The result cannot be represented [EOVERFLOW]
   NULL value as UTC time not found

9. rm (rmdir and unlink system calls)
   The *path* argument names a directory that is not an empty directory, or there are hard links to the directory other than dot or a single entry in dot-dot.
   The directory entry to be unlinked is part of a read-only file system.

10. mkdir (mkdir system call)
   Search permission is denied on a component of the path prefix, or write permission is denied on the parent directory of the directory to be created.
   The named file exists.

Apart from this, we also handle errors for fork(), waitp() and execvp() system calls

- Insufficient storage space is available.(fork)
- The system lacked the necessary resources to create another process, or the system-imposed limit on the total number of processes under execution system-wide or by a single user {CHILD_MAX} would be exceeded.(fork)
- The calling process has no existing unwaited-for child processes.(wait)
- The function was interrupted by a signal. The value of the location pointed to by *stat_loc* is undefined. (wait)
- The number of bytes used by the new process image's argument list and environment list is greater than the system-imposed limit of {ARG_MAX} bytes. (execvp)
- The new process image file has the appropriate access permission but has an unrecognized format.(execvp)