OS Lab1

Q1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Should write programs for at least six commands.

LS command

code:

```
#include <stdio.h>
#include <dirent.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

void _ls(const char *dir,char op);


int main(int argc,char **argv){
   //printf("%s",argv[1]);
   if (argc == 1){
      _ls(".",'e');
   }
   else if(argc == 2){
       //printf("sd\n");
       if(strcmp( argv[1], "-R") == 0)
           _ls(".", 'R');
       else if(strcmp( argv[1], "-a") == 0 || strcmp( argv[1], "-l") == 0)
           _ls(".", argv[1][1]);
       else
           _ls(argv[1], 'e');
   }
  exit(EXIT_SUCCESS);
}


void _ls(const char *dir,char op){
   struct dirent *d;
   DIR *dh = opendir(dir);
```

```
   if (!dh)
   {
       if (errno == ENOENT)
       {
           //If the directory is not found
           perror("Directory doesn't exist");
       }
       else
       {
           //If the directory is not readable then throw error and exit
           perror("Unable to read directory");
       }
       exit(EXIT_FAILURE);
   }

   if( op == 'R'){
       while ((d = readdir(dh)) != NULL){
           printf("%s\n",d->d_name);

           if(d->d_type==DT_DIR && ((strcmp(".",d->d_name)!=0) &&
(strcmp("..",d->d_name)!=0))){
               char path[100]={0};
               strcat(path,dir);
               strcat(path,"/");
               strcat(path,d->d_name);
               printf("sub directory of %s is\t",d->d_name);
               _ls(path,'R');
           }
       }
   } else{
       while ((d = readdir(dh)) != NULL)
       {
           //If hidden files are found we continue
           if (!(op == 'a') && d->d_name[0] == '.')
           continue;
           printf("%s  ", d->d_name);
           if(op == 'l') printf("\n");
       }
       if(!(op == 'l'))
       printf("\n");
   }
```

```
}
```

Output:

Ls
output:

```
Ishans-MacBook-Air:Lab1 ishan$ ./ls
cat  ls.c  ps.c  js  rm.c  cat.c  wc.c  wc  states.txt  kill  ps  a.out  kill.c  ls  rm  capitals.txt
Ishans-MacBook-Air:Lab1 ishan$
```

Ls -a
output:

```
Ishans-MacBook-Air:Lab1 ishan$ ./ls -a
.  ..  cat  .DS_Store  ls.c  ps.c  js  rm.c  cat.c  wc.c  wc  states.txt  kill  ps  a.out  kill.c  ls  rm  capitals.txt
```

Ls -l

```
Ishans-MacBook-Air:Lab1 ishan$ ./ls -l
cat
ls.c
ps.c
js
rm.c
cat.c
wc.c
wc
states.txt
kill
ps
a.out
kill.c
ls
rm
capitals.txt
```

Ls -R

```
Ishans-MacBook-Air:Lab1 ishan$ ./ls -R
.
..
cat
.DS_Store
ls.c
ps.c
js
sub directory of js is  .
..
Lab1
sub directory of Lab1 is        .
..
.DS_Store
stackLinkedList.h
Stack.h
.vscode
sub directory of .vscode is     .
..
settings.json
.DS_Store
rm.c
cat.c
wc.c
wc
states.txt
kill
ps
a.out
kill.c
ls
rm
capitals.txt
```

ls <dirname>

```
Ishans-MacBook-Air:Lab1 ishan$ ./ls js
Lab1  Lab2  Lab1_22CSM1S01_Ishan_Garg.zip  Lab1&2_22CSM1S01_Ishan_Garg.zip  Lab2-1
```

Rm command

Code:

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
 int main(int argc, char *argv[])
{

   if(strcmp( argv[1], "-i") == 0){
```

```
        int i = 2;
        while(i < argc){
            printf("remove %s?", argv[i]);
            char ans[3];
            scanf("%[^\n]%*c", ans);
            if(ans[0] == 'y'){
                remove(argv[i]);
                i++;
            }
            else{
                i++;
                continue;
            }
        }

    } else{
        int i = 1;
        while(i < argc){
            remove(argv[i]);
            i++;
        }
    }

    return 0;

}
```

Output:
Folder before rm:

```
Ishans-MacBook-Air:Lab1 ishan$ ls
a.out           cat.c           kill.c          ps              rm.c            wc.c
capitals.txt    js              ls              ps.c            states.txt
cat             kill            ls.c            rm              wc
```

rm <filename>

```
Ishans-MacBook-Air:Lab1 ishan$ rm cat.c
Ishans-MacBook-Air:Lab1 ishan$ ls
a.out           js              kill.c          ls.c            ps.c            rm.c            wc
capitals.txt    kill            ls              ps              rm              states.txt      wc.c
```

rm -i <filename1> <filename2>…

```
Ishans-MacBook-Air:Lab1 ishan$ rm -i kill.c kill
remove kill.c? y
remove kill? y
Ishans-MacBook-Air:Lab1 ishan$ ls
a.out           js              ls.c            ps.c            rm.c            wc
capitals.txt    ls              ps              rm              states.txt      wc.c
```

rm <file1> <file2> ...

```
Ishans-MacBook-Air:Lab1 ishan$ rm ps ps.c
Ishans-MacBook-Air:Lab1 ishan$ ls
a.out           js              ls.c            rm.c            wc
capitals.txt    ls              rm              states.txt      wc.c
Ishans-MacBook-Air:Lab1 ishan$
```

Cat Command

Code:

```c
#include <stdio.h>
#include <sys/syscall.h>
#include <fcntl.h>

#define MAX_LIMIT 20

void printFile(char *fileName, int argc){

   int fd,i;
   char buf[2];

   fd=open(fileName,O_RDONLY,0777);

   if(fd==-argc)
   {
       printf("file open error");
   }
   else
   {
       while((i=read(fd,buf,1))>0)
       {
           printf("%c",buf[0]);
       }
```

```
        close(fd);
    }
}

int main( int argc,char *argv[3] )
{

    if(argc == 1){
        while(1){
            char str[MAX_LIMIT];
            scanf("%[^\n]%*c", str);
            printf("%s\n", str);
        }
    } else if(argc == 2){
        printFile(argv[1], argc);
    } else{
        int i = 1;
        while(i < argc){
            printFile(argv[i], argc);
            i++;
        }
    }

    return 0;
}
```

Output:

cat <file>

```
Ishans-MacBook-Air:Lab1 ishan$ cat states.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
ChhattisgarhIshans-MacBook-Air:Lab1 ishan$
```

cat <file1> <file2> …

```
Ishans-MacBook-Air:Lab1 ishan$ cat states.txt capitals.txt
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
ChhattisgarhHyderabad
Itanagar
Dispur
Patna
RaipurIshans-MacBook-Air:Lab1 ishan$
```

Cat

```
Ishans-MacBook-Air:Lab1 ishan$ cat
wrting only cat command
wrting only cat command
displays the exactly the same this that you write
displays the exactly the same this that you write
it echos whatever you write
it echos whatever you write
bye
bye
```

Kill command

code:

```
#include<stdio.h>

#include<unistd.h>

#include<sys/types.h>

#include<signal.h>
```

```c
int main(int argc,char *argv[])

{

   if(argv[1][0] == '-'){

       argv[1] += 1;

       for (int i = 2; i < argc; i++)

       {

           kill(atoi(argv[i]),atoi(argv[1]));

       }

   } else{

       for (int i = 1; i < argc; i++)

       {

           kill(atoi(argv[i]),SIGKILL);

       }

   }

   return 0;

}
```

Output:

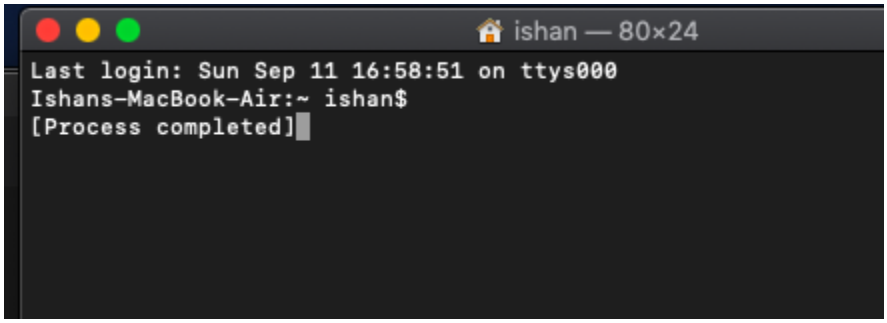
Top command to see process id:

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	%CPU_ME
18587	screencaptur	0.8	00:00.14	4	2	152	4000K	0B	0B	18587	1	sleeping	*0[342+]	0.00000
18586	screencaptur	6.0	00:00.75	2	1	54	3000K+	620K	0B	272	272	sleeping	*0[1]	0.15325
18585	top	4.1	00:02.13	1/1	0	27	3340K+	0B	0B	18585	17892	running	*0[1]	0.00000
18579	bash	0.0	00:00.01	1	0	21	768K	0B	0B	18579	18578	sleeping	*0[1]	0.00000
18578	login	0.0	00:00.03	2	1	31	1100K	0B	0B	18578	884	sleeping	*0[9]	0.00000
18574	corecaptured	0.0	00:00.03	9	1	70	17M	0B	0B	18574	1	sleeping	0[0]	0.00000

Kill pid
Killing bash pid 18579

```
Ishans-MacBook-Air:Lab1 ishan$ ./kill 18579
Ishans-MacBook-Air:Lab1 ishan$
```

Bash process that got killed:



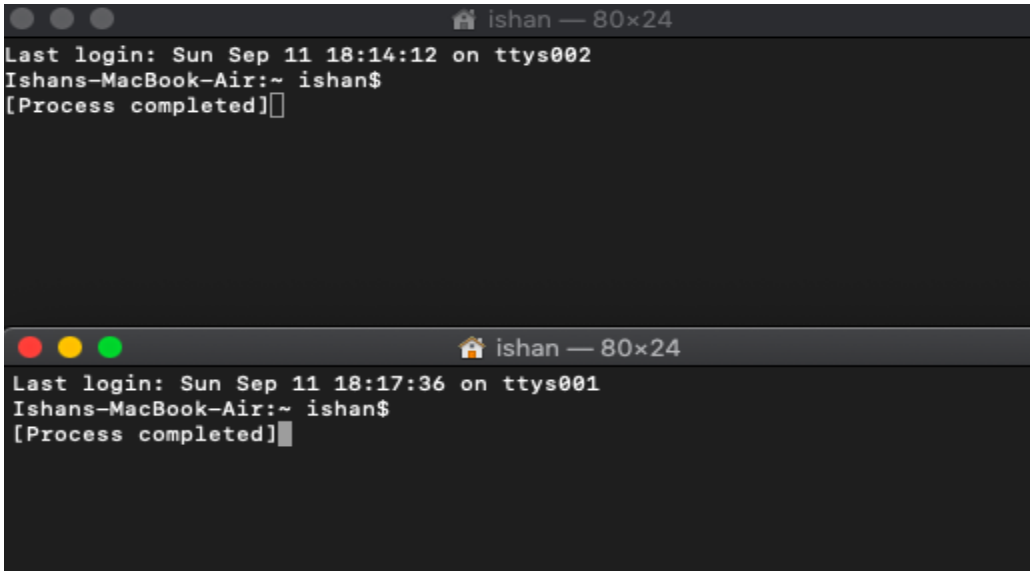
Kill -<signo> <pid1> <pid2> …

Top command:

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID	STATE	BOOSTS	%CPU_ME
18692	screencaptur	0.1	00:00.13	4	2	151	3920K	0B	0B	18692	1	sleeping	*0[276+]	0.00000
18691	screencaptur	0.8	00:00.67	2	1	54	2920K	620K	0B	272	272	sleeping	*0[1]	0.02085
18690	top	4.3	00:01.21	1/1	0	27	3384K+	0B	0B	18690	17892	running	*0[1]	0.00000
18689	mdworker_sha	0.0	00:00.14	4	1	58	6304K	0B	0B	18689	1	sleeping	*0[1]	0.00000
18688	mdworker_sha	0.0	00:00.13	4	1	58	6052K	0B	0B	18688	1	sleeping	*0[1]	0.00000
18682	bash	0.0	00:00.01	1	0	21	812K	0B	0B	18682	18681	sleeping	*0[1]	0.00000
18681	login	0.0	00:00.02	2	1	32	1092K	0B	0B	18681	884	sleeping	*0[9]	0.00000
18675	bash	0.0	00:00.01	1	0	21	800K	0B	0B	18675	18674	sleeping	*0[1]	0.00000

Killing bash 18682 and 18675 with signal no 9

Killed bash programs:



Wc command

Code:

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>



void countFile(char* arg, char op){
   FILE * file;
   char ch;
   int characters=0, words=0, lines=0;
   file = fopen(arg, "r");
   while ((ch = fgetc(file)) != EOF)
   {
   characters++;
   if (ch == '\n')
       lines++;
   if (ch == ' ' || ch == '\n')
       words++;
   }
   if (characters > 0)
   {
   words++;
   lines++;
   }

   if(op == 'w')
       printf("%d %s\n", words-1, arg);
   else if(op == 'c')
       printf("%d %s\n", characters, arg);
   else if(op == 'l')
       printf("%d %s\n", lines-1, arg);
   else
       printf("%d %d %d %s\n", lines-1,words-1,characters, arg);


   fclose(file);
}

int main(int c, char *argv[])
{
   char op = '0';
```

```
    if(argv[1][0] == '-'){
        if(strcmp(argv[1],"-w") ==0 ){
            op = 'w';
        } else if(strcmp(argv[1],"-c") ==0){
            op = 'c';
        } else{
            op = 'l';
        }

        for (int i = 2; i < c; i++)
        {
            countFile(argv[i], op);
        }

        return 0;
    }

    for (int i = 1; i < c; i++)
    {
        countFile(argv[i], op);
    }

    return 0;
}
```

Output:
wc <file1> <file2> …

```
Ishans-MacBook-Air:Lab1 ishan$ ./wc states.txt capitals.txt
4 6 57 states.txt
4 4 38 capitals.txt
```

wc -c <file1> <file2> ..

```
Ishans-MacBook-Air:Lab1 ishan$ ./wc -c states.txt capitals.txt
57 states.txt
38 capitals.txt
```

wc -l <file1> <file2> ...

```
Ishans-MacBook-Air:Lab1 ishan$ ./wc -l states.txt capitals.txt
4 states.txt
4 capitals.txt
```

wc -w <file1> <file2> ...

```
Ishans-MacBook-Air:Lab1 ishan$ ./wc -w states.txt capitals.txt
6 states.txt
4 capitals.txt
```

Mkdir command

Code:

```c
#include<stdio.h>
#include<string.h>
#include<unistd.h>

int main(int argc, char* argv[]){



if(argc!=2 || strcmp(argv[1], "--help") == 0)
 {
   printf("\nusage: mkdir creates a directory\n");
   return 0;
  // break;
 }
 char *cmd = "mkdir";
 char *args[3];
 args[0] = "mkdir";
 args[1] = argv[1];
 args[2] = NULL;

 execvp(cmd, argv);

 return 0;
}
```

Output:

```
Ishans-MacBook-Air:Lab1 ishan$ ls
-Wno-atomic-implicit-seq-cst    kill                            rm
a.out                           kill.c                          rm.c
capitals.txt                    ls                              states.txt
cat                             ls.c                            wc
cat.c                           mkdir                           wc.c
js                              mkdir.c
Ishans-MacBook-Air:Lab1 ishan$ ./mkdir ishan
Ishans-MacBook-Air:Lab1 ishan$ ls
-Wno-atomic-implicit-seq-cst    js                              mkdir.c
a.out                           kill                            rm
capitals.txt                    kill.c                          rm.c
cat                             ls                              states.txt
cat.c                           ls.c                            wc
ishan                           mkdir                           wc.c
Ishans-MacBook-Air:Lab1 ishan$
```