

OPERATING SYSTEMS LAB - PRACTICAL 2

Name - Sakshi Soni

Roll No - 13

AIM -

Write a C program to simulate Directory management system calls of Linux

- a) Create directory (mkdir comand)
- b) List the number of files in current directory(ls command)
- c) Navigate the directory (cd command)
- d) Remove directory (rmdir command)

PROGRAM AND OUTPUT -

1.

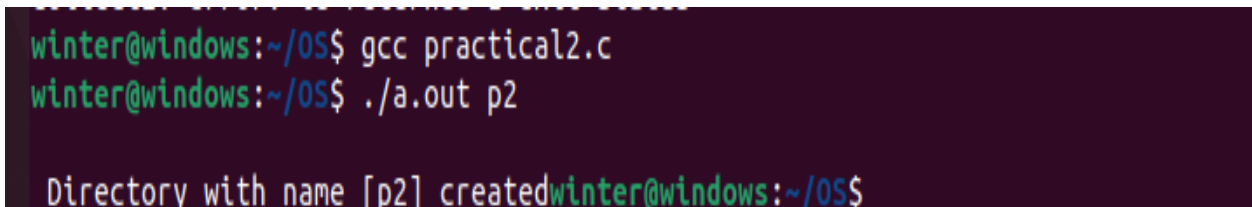
```
#include<stdio.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<stdlib.h>
#include<unistd.h>
#include<dirent.h>
```

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

```

f=mkdir(argv[1],0777);
if(f==-1){
printf("\nCannot create directory\n");
exit(-1);
}
else{
printf("\n Directory with name [%s] created",argv[1]);
}
return 0;
}

```



A terminal window with a dark purple background. The prompt is 'winter@windows:~/OS\$'. The first command is 'gcc practical2.c' and the second is './a.out p2'. The output is 'Directory with name [p2] created' followed by the prompt 'winter@windows:~/OS\$'.

```

winter@windows:~/OS$ gcc practical2.c
winter@windows:~/OS$ ./a.out p2

Directory with name [p2] createdwinter@windows:~/OS$

```

2.

```

#include<stdio.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<stdlib.h>
#include<unistd.h>
#include<dirent.h>
int main(int argc, char *argv[]){
    char cwd[1024];
    char dirname[10];
    DIR *p;

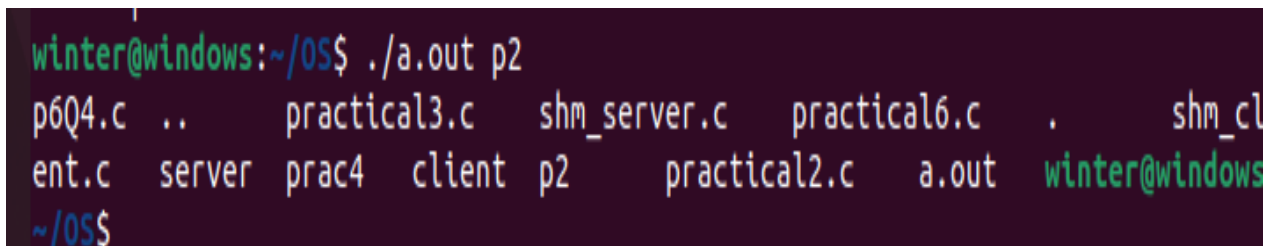
```

```

    struct dirent *d;
    if(getcwd(cwd,sizeof(cwd))!=NULL)
        fprintf("\nCurrent working directory is: %s\n",cwd);
    else
        perror("getcwd() error");
    p=opendir(cwd);
    if(p==NULL){
        perror("\n Cannot find directoy\n");
        exit(-1);
    }
    while(d=readdir(p)){
        printf("%s\t",d->d_name);

    }
    return 0;
}

```



A terminal window screenshot showing the output of a command. The prompt is 'winter@windows:~/OS\$'. The command executed is './a.out p2'. The output is a directory listing: 'p6Q4.c .. practical3.c shm_server.c practical6.c . shm_cl', 'ent.c server prac4 client p2 practical2.c a.out winter@windows', and '~ /OS\$'.

3.

```

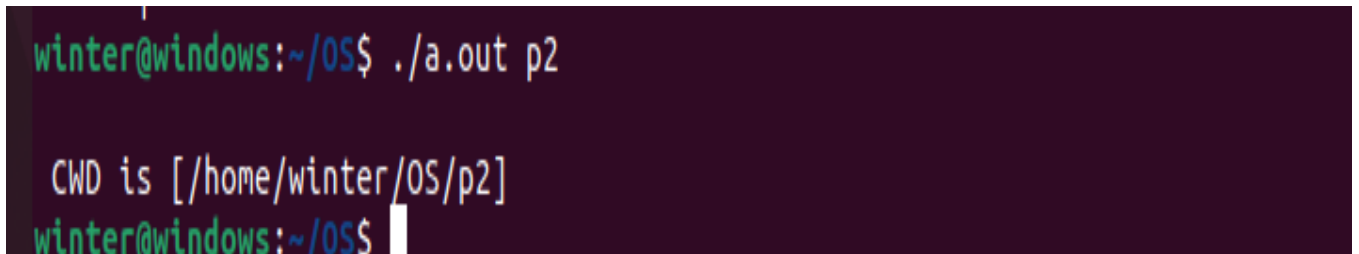
#include<stdio.h>
#include<sys/stat.h>
#include<sys/types.h>
#include<stdlib.h>

```

```

#include<unistd.h>
#include<dirent.h>
#define NAME_MAX 100
int main(int argc, char **argv){
    char buf[NAME_MAX];
    char *path=argv[1];
    if (chdir(path)==-1){
        fprintf(stderr,"\nerror: could not change to dir
[%s]\n",path);
        return 1;
    }
    getcwd(buf,NAME_MAX);
    printf("\n CWD is [%s]\n",buf);
    return 0;
}

```



A terminal window with a dark purple background. The prompt is 'winter@windows:~/OS\$'. The user enters './a.out p2'. The output is 'CWD is [/home/winter/OS/p2]'. The prompt returns to 'winter@windows:~/OS\$'.

```

winter@windows:~/OS$ ./a.out p2

CWD is [/home/winter/OS/p2]
winter@windows:~/OS$

```

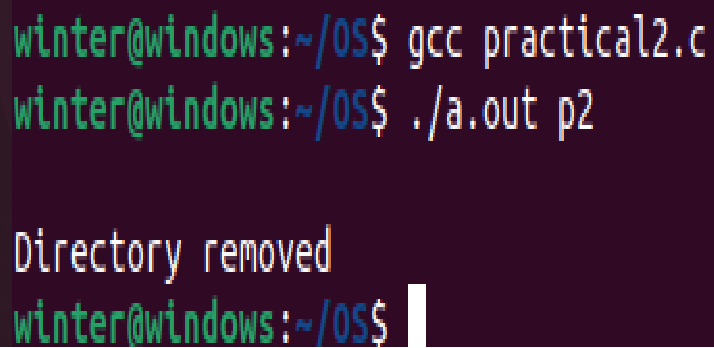
4.

```

int main(int argc, char *argv[]){
    int f;
    f=rmdir(argv[1]);
    if(f==-1){
        printf("\nCannot remove directory\n");
    }
}

```

```
    exit(-1);  
}  
else{  
    printf("\nDirectory removed\n");  
}  
    return 0;  
}
```

A terminal window with a dark purple background. The prompt is 'winter@windows:~/OS\$'. The first command is 'gcc practical2.c' and the second is './a.out p2'. The output is 'Directory removed'. The prompt is now 'winter@windows:~/OS\$' with a cursor.

```
winter@windows:~/OS$ gcc practical2.c  
winter@windows:~/OS$ ./a.out p2  
  
Directory removed  
winter@windows:~/OS$
```

RESULT -

Directory management system calls of Linux has been studied and C programs on them has been implemented.