Name : Aneesha Rose
Roll No : CS19B1068
Course : B.Tech CSE
Semester : 5
Subject : CS305
Purpose: Assignment

UNIX PROGRAMMING

DATE: 18/10/2021

AIM : *Create a program which does when given input as:*

1 - create a new directory

2 - read files and sub directories recursively of a given directory

3 - create a directory which cannot be removed

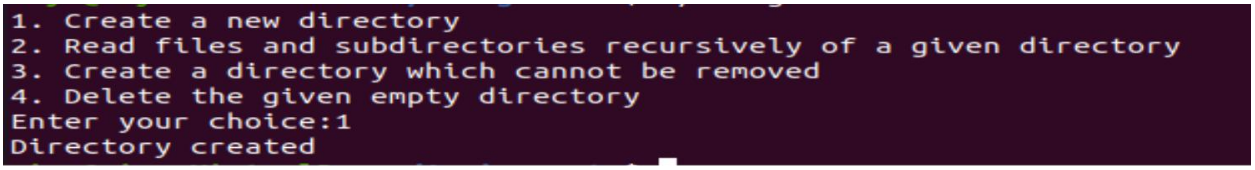
4 - delete the given empty directory

PROGRAM:

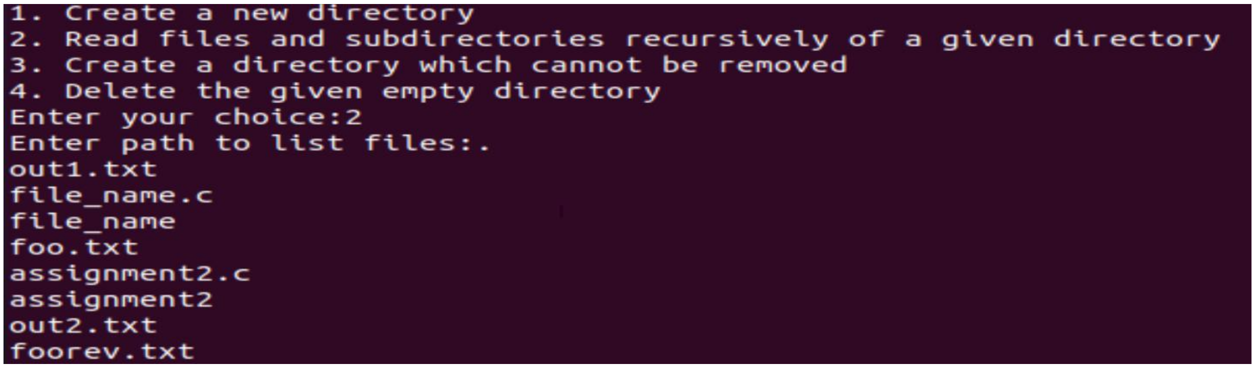
```
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include <errno.h>

void listdir(char *path);

int main()
{
   int choice=0;
   char dir2[100];
   char dir3[10];
   char path2[20]="./";
   printf("1. Create a new directory\n2. Read files and subdirectories recursively of a given
directory\n3. Create a directory which cannot be removed\n4. Delete the given empty
```

```
directory\n");
   printf("Enter your choice:");
   scanf("%d",&choice);
   int check=0;
   int ch=0;
   char dirname[20] = "riyaroshan";
   switch(choice)
   {    while(1)
      {
      case 1:
      {
         check = mkdir(dirname,0777);
         if(!check)
            printf("Directory created\n");
         else
         {
            printf("Unable to create directory\n");
            exit(1);
         }
      break;
      }
      case 2:
      {
         char path[100];
         printf("Enter path to list files:");
         scanf("%s",&path);
         listdir(path);
      break;
      }
      case 3:
      {
         ch = mkdir(dir2,0777);
         printf("Enter name of directory to be created: ");
         scanf("%s",&dir2);
         if(!ch)
            printf("Directory created\n");
         printf("Proceed to attempt to delete file\n");
         if(rmdir(dir2)!=0)
            perror("rmdir() error");
         else
         {
            strcat(path2,dir2);
            strcat(path2,"/file.txt");
            }
        }
        case 4:
        {
            printf("\nEnter directory to be deleted:");
            scanf("%s",&dir3);
            if(rmdir(dir3)!=0)
            {
                printf("Directory cannot be deleted");
                perror("rmdir() error");
            }
            else
                printf("Directory deleted\n");
        break;
        }
        default:
            printf("Invalid choice");
        break;
        }
    }
}
void listdir(char *basePath)
{
    char path[1000];
    struct dirent *dp;
    DIR *dir = opendir(basePath);
    if(!dir)
        return;

    while((dp = readdir(dir)) != NULL)
    {
        if(strcmp(dp->d_name, ".")!=0 && strcmp(dp->d_name, "..")!=0)
        {
            printf("%s\n",dp->d_name);
            strcpy(path, basePath);
            strcat(path,"/");
            strcat(path, dp->d_name);

            listdir(path);
        }
    }
    closedir(dir);
}
```

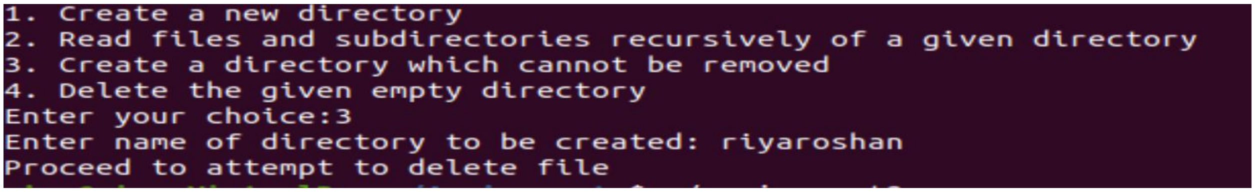
OUTPUT:



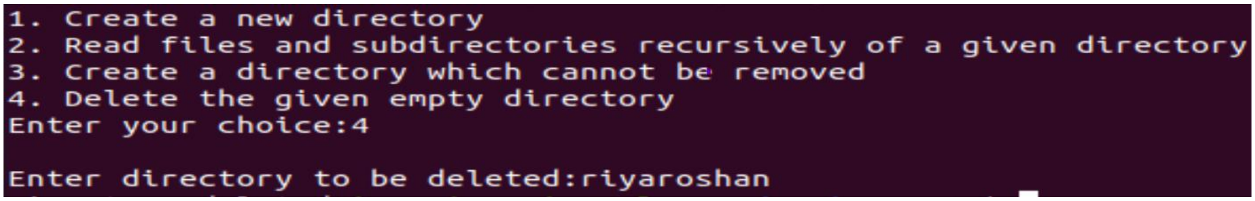
```
1. Create a new directory
2. Read files and subdirectories recursively of a given directory
3. Create a directory which cannot be removed
4. Delete the given empty directory
Enter your choice:1
Directory created
```



```
1. Create a new directory
2. Read files and subdirectories recursively of a given directory
3. Create a directory which cannot be removed
4. Delete the given empty directory
Enter your choice:2
Enter path to list files:.
out1.txt
file_name.c
file_name
foo.txt
assignment2.c
assignment2
out2.txt
foorev.txt
```



```
1. Create a new directory
2. Read files and subdirectories recursively of a given directory
3. Create a directory which cannot be removed
4. Delete the given empty directory
Enter your choice:3
Enter name of directory to be created: riyaroshan
Proceed to attempt to delete file
```



```
1. Create a new directory
2. Read files and subdirectories recursively of a given directory
3. Create a directory which cannot be removed
4. Delete the given empty directory
Enter your choice:4

Enter directory to be deleted:riyaroshan
```

RESULT:

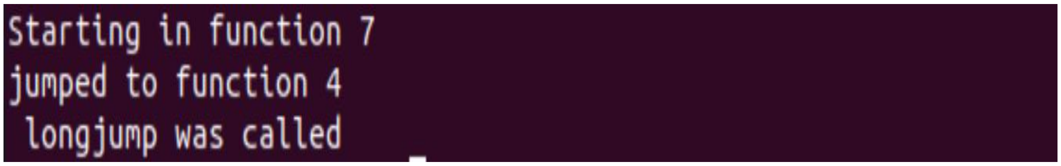
The program is successful

AIM : *Write a program to: Implement a program using non-local jump functions. Jump from function 7 to function 4*

PROGRAM:

```
#include <stdio.h>
#include <setjmp.h>

static jmp_buf env;
void func1()
{
     printf("In function 1 \n");
}
void func2()
{
    printf("In function 2 \n");
}
void func3()
{
    printf("In function 3 \n");
}
void func4()
{
    printf("jumped to function 4\n longjump was called \n");
    longjmp(env, 1);
}
void func5()
{
    printf("In function 5 \n");
}
void func6()
{
     printf("In function 6 \n");
}
void func7()
{
    printf("Starting in function 7\n");
    if(!setjmp(env))
    func4();
}

int main()
{
     func7();
}
```

OUTPUT:



```
Starting in function 7
jumped to function 4
 longjump was called
```

RESULT:

The program is successful