**PRACTICAL-3**

**Aim:**

Write a program for ‘ls’ command using ‘opendir()’ and ‘readdir()’ system call.

**Program Code:**

#include<stdio.h>

#include<dirent.h>

#include<stdlib.h>

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

struct dirent \*dp;

DIR \*dir;

int count = 0;

dir = opendir(argv[1]);

if(dir == NULL) {

printf("Error\n");

exit(1);

}

while((dp = readdir(dir)) != NULL) {

if(dp->d\_type == DT\_DIR)

printf("Directory name: %s\n", dp->d\_name);

else

printf("File name: %s\n", dp->d\_name);

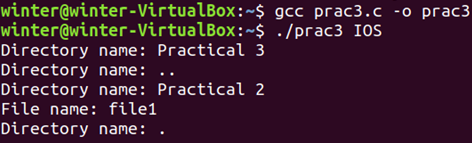
}

printf("\n");

return 0;

}

**Output:**

****

**ASSIGNMENT-2**

**Aim:**

Create a program that can be used to implement following specification for 'ls' command.

ls, ls  -a (already implemented as a part of practical 3) , ls  -A , ls  -R

**Program Code:**

#include<stdio.h>

#include<dirent.h>

#include<stdlib.h>

#include<string.h>

void diropen(char \*path);

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

struct dirent \*dp;

DIR \*dir;

int count = 0;

dir = opendir(argv[1]);

if(dir == NULL) {

printf("Error\n");

exit(1);

}

else {

if(strcmp(argv[2], "-a") == 0) {

while((dp = readdir(dir)) != NULL) {

if(dp->d\_type == DT\_DIR)

printf("Directory name: %s\n", dp->d\_name);

else

printf("File name: %s\n", dp->d\_name);

}

}

if(strcmp(argv[2], "-A") == 0){

while((dp = readdir(dir)) != NULL) {

if(strcmp(dp->d\_name, ".") == 0 || strcmp(dp->d\_name, "..") ==0)

continue;

else {

if(dp->d\_type == DT\_DIR)

printf("Directory name: %s\n", dp->d\_name);

else

printf("File name: %s\n", dp->d\_name);

}

}

}

if(strcmp(argv[2], "-R") == 0) {

diropen(argv[1]);

}

}

closedir(dir);

printf("\n");

return 0;

}

void diropen(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) {

if(dp->d\_type == DT\_DIR)

printf("Directory name: %s\n", dp->d\_name);

else

printf("File name: %s\n", dp->d\_name);

// Construct new path from our base path

strcpy(path, basePath);

strcat(path, "/");

strcat(path, dp->d\_name);

diropen(path);

}

}

closedir(dir);

}

**Output:**

