A

Project Report

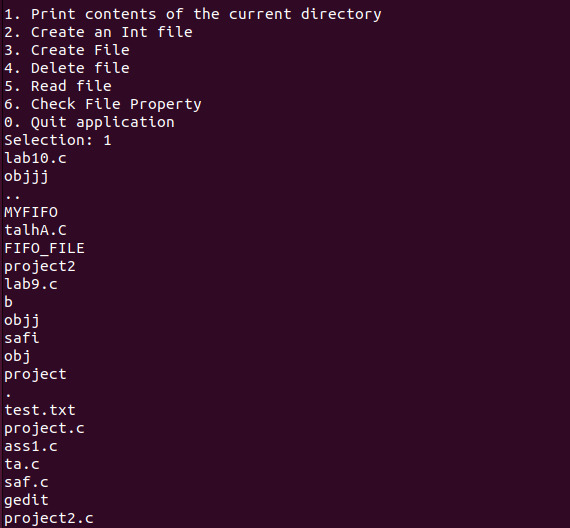
On

**“File Management System”**

For The Course

**“Operating Systems”**

**MAIN SCREEN:**

****

**DESCRIPTION:**

This management system allows a user to:

1. Display contents of the current directory.
2. Create file to store data of type [Int].
3. Create an empty file.
4. Delete an existing file.
5. Read data of an existing file.
6. Check file properties.

**CODE:**

include <stdio.h>

#include <stdlib.h>

#include <dirent.h>

#include <unistd.h>

#include <sys/stat.h>

#include <time.h>

#include <errno.h>

#include <stdbool.h>

#include <string.h>

void readfc()

{

FILE \*fh;

char filen[20];

printf("Enter the name of the file you wish to Read: \n");

scanf("%s", filen);

fh = fopen(filen, "r");

if (fh != NULL) {

char c;

while ( (c = fgetc(fh)) != EOF )

putchar(c);

fclose(fh);

}

else {

printf("Error opening file.\n");

}

}

void Delete()

{

char filename[100];

printf("File: ");

scanf("%s", filename);

if (remove(filename) != 0)

{

fprintf(stderr, "Errno: %d\n", errno);

perror("Error msg");

} else printf("%s deleted.\n", filename);

}

void createfile() {

int test\_num;

char filename[50];

FILE \*test\_ptr;

printf("Enter the name of the file you wish to create: \n");

scanf("%s", filename);

test\_ptr = fopen(filename,"w");

if(test\_ptr == NULL){

printf("Error!");

exit(1);

}

printf("Enter an integer to write to the file as a test: ");

scanf("%d",&test\_num);

fprintf(test\_ptr,"%d",test\_num);

fclose(test\_ptr);

return;

}

void ls(void){

struct dirent \*ptr\_dir;

DIR \*dr = opendir(".");

if (dr == NULL){

printf("Could not open current directory\n");

return;

}

while ((ptr\_dir = readdir(dr)) != NULL) {

printf("%s \n", ptr\_dir->d\_name);

}

closedir(dr);

}

void create\_e\_file(){

char ch, filename[50];

FILE \*my\_fp;

printf("Enter the name of file \n");

scanf("%s",filename);

my\_fp = fopen(filename,"a");

if ( my\_fp == NULL ) {

perror("Error while opening the file:\n" );

exit(EXIT\_FAILURE) ;

}

fclose(my\_fp);

}

void printFileProperties(struct stat stats);

int fileproperties()

{

char path[100];

struct stat stats;

printf("Enter file name: ");

scanf("%s", path);

if (stat(path, &stats) == 0)

{

printFileProperties(stats);

}

else

{

printf("Unable to get file properties.\n");

printf("Please check whether '%s' file exists.\n", path);

}

return 0;

}

void printFileProperties(struct stat stats)

{

printf("\nFile access: ");

if (stats.st\_mode & R\_OK)

printf("read ");

if (stats.st\_mode & W\_OK)

printf("write ");

if (stats.st\_mode & X\_OK)

printf("execute");

printf("\nFile size: %ld \n", stats.st\_size);

}

void main\_m(){

int input;

printf( "\n" );

printf( "1. Print contents of the current directory \n" );

printf( "2. Create an Int file \n" );

printf( "3. Create File \n" );

printf( "4. Delete file \n" );

printf( "5. Read file \n" );

printf( "6. Check File Property \n" );

printf( "0. Quit application \n" );

printf( "Selection: " );

scanf( "%d", &input );

switch ( input ) {

case 1:

ls();

break;

case 2:

createfile();

break;

case 3:

create\_e\_file();

break;

case 4:

Delete();

break;

case 5:

readfc();

break;

case 6:

fileproperties();

break;

case 0:

printf( "Quitting application..! \n");

break;

default:

printf( "Bad input, quitting! \n");

}

getchar();

}

int main(){

main\_m();

}