

Name: K.SaiKrishna

Reg-No: 192311106

27. Develop a C program for simulating the function of ls UNIX Command.

Aim:

The aim of this C program is to simulate the functionality of the `ls` command in UNIX. The program lists the files and directories in the current directory or specified directory.

Algorithm:

1. Take an optional directory path as input. If no directory is specified, use the current directory.
2. Use the `opendir` function to open the directory.
3. Use the `readdir` function to read each entry in the directory.
4. Print the names of the files and directories.
5. Close the directory using `closedir`.

Procedure:

1. Include necessary headers (`stdio.h`, `dirent.h`, `stdlib.h`).
2. Open the directory using `opendir`.
3. Read the directory entries using `readdir`.
4. Print the filenames of the directory entries.
5. Close the directory after reading.

Code:

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

```
#include <stdlib.h>
```

```
#include <dirent.h>
```

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

```
    DIR *dir;
```

```
    struct dirent *entry;
```

```
    char *path = (argc > 1) ? argv[1] : ".";
```

```

dir = opendir(path);

if (dir == NULL) {

    perror("opendir");

    return 1;

}

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

    printf("%s\n", entry->d_name);

}

closedir(dir);

return 0;

}

```

Output:

The screenshot shows a code editor with a sidebar on the left containing navigation links: Welcome, K Sai Krishna, Create New Project, My Projects, Classroom (new), Learn Programming, Programming Questions, Upgrade, and Logout. The main editor area displays C code for directory traversal. The code includes comments and uses `opendir`, `readdir`, `printf`, `perror`, and `closedir`. The output window at the bottom shows the execution results: `main.c`, `testfile.txt`, and `a.out`.

```

8
9 // Check if directory path is provided as an argument
10 const char *path = (argc > 1) ? argv[1] : "."; // Default to current d
11
12 // Open the directory
13 dir = opendir(path);
14 if (dir == NULL) {
15     perror("Error opening directory");
16     exit(1);
17 }
18
19 printf("Contents of directory: %s\n", path);
20
21 // Read and display directory contents
22 while ((entry = readdir(dir)) != NULL) {
23     printf("%s\n", entry->d_name);
24 }
25
26 // Close the directory
27 if (closedir(dir) == -1) {
28     perror("Error closing directory");
29     exit(1);
30 }
31
32 return 0;
33 }
34

```

input

main.c
testfile.txt
a.out

Program finished with exit code 0

Result:

Running the program would output a list of filenames and directories in the current directory or specified directory.