LAB 5 : Linux Command Line Utilities Simulator in C Programming

Nirajan Bekoju
PUL076BCT039
076bct039.nirajan@pcampus.edu.np

Nishant Luitel
PUL076BCT041
076bct041.nishant@pcampus.edu.np

Nabin Da Shrestha
PUL076BCT037
076bct037.nabin@pcampus.edu.np

Prakash Chaulagain
PUL076BCT045
076bct045.prakash@pcampus.edu.np

I. Introduction

A Linux Command Line Utilities Simulator in C Programming is a collection of programs that simulate the behavior of various Linux commands in a command line environment. The purpose of this simulator is to provide a way for users to learn and practice using Linux commands without having to install Linux or access a Linux terminal.

Each program in the simulator is written in the C programming language and emulates the functionality of a specific Linux command. For example, one program may simulate the behavior of the ”ls” command, while another program may simulate the behavior of the ”cd” command.

The programs in the simulator typically use standard input and output to communicate with the user, allowing the user to input command line arguments and see the output of the simulated command. The programs may also use system calls and libraries to access the file system and perform other operations required by the simulated command.

The Linux Command Line Utilities Simulator in C Programming can be a useful tool for individuals who are new to Linux and want to learn how to use Linux commands, as well as for experienced Linux users who want to test out different command options or troubleshoot issues in a safe and controlled environment. It can also be used by educators as a teaching tool in computer science or information technology courses.

II. Implementation of ls Command

Aim : To write a C program to simulate the operation of “ls” command in Unix.

Algorithm :

```
STEP 1 : Check if the number of command line arguments is less than 2. If yes, Print error and exit.
STEP 2 : Check if the second argument (i.e. directory to be listed) is valid or not. If not then exit.
STEP 3 : Print the content of the directory
STEP 4 : Close the directory entry file.
```

Program :

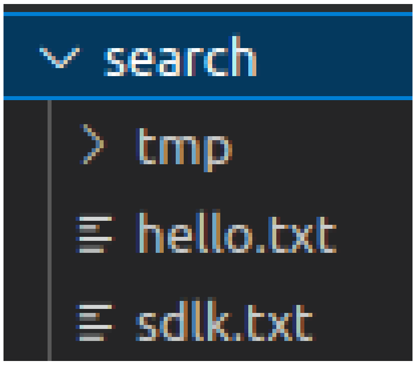
```
#include <stdio.h>
#include <sys/types.h>
#include <dirent.h>

// struct dirent {
//     ino_t          d_ino;       /* Inode number */
//     off_t          d_off;       /* Not an offset; see below */
//     unsigned short d_reclen;    /* Length of this record */
//     unsigned char  d_type;      /* Type of file; not supported
//                                    by all filesystem types */
//     char           d_name[256]; /* Null-terminated filename */
// };

int main(int argc, char *argv[]){
    DIR *dp;

    struct dirent *dirp;
    if(argc < 2){
        printf("\n You have provided only 1 argument \n");
        return -1;
    }
    dp = opendir(argv[1]);
    if(dp == NULL){
        printf("\n Cannot open %s file!\n", argv[1]);
        return -1;
    }
    dirp = readdir(dp);
    while (dirp != NULL)
    {
        printf("%s\n", dirp->d_name);
        dirp = readdir(dp);
    }
    closedir(dp);
}
```

The folder structure for ”search” folder is as follow:



Ouput :

```
$ ./ls search

tmp
hello.txt
sdlk.txt
.
..
```

III. Conclusion

In conclusion, the Linux Command Line Utilities Simulator in C Programming is a valuable tool for those who wish to learn or practice using Linux commands. With this simulator, users can experiment with different commands and options without the risk of damaging their own system. The implementation of the ”ls” command in this lab report demonstrates the functionality and potential of the simulator. As a teaching tool, the simulator can aid educators in providing hands-on learning experiences for students in computer science or information technology courses. Overall, the Linux Command Line Utilities Simulator in C Programming is a useful resource for individuals who wish to improve their Linux command line skills.