Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

SSN College of Engineering

Department of Computer Science and Engineering

UCS2412 – Operating Systems Laboratory

II Year CSE - IV Semester A & B Sections

Exercise – 2- Simulation of system commands using system calls

Objective:

1. To develop a C program to implement the cp, ls, grep commands (with some options) using system calls.

Sample Learning Outcome:

1. Implement the various system commands like cp, grep, ls, head, tail, wc using system calls

2. Learn to process command line arguments and error handling mechanism

3. Understand the relation between the system calls and commands

Best Practices:

1. Algorithm design

2. Naming convention – for file names, variables

3. Comment usage at proper places

4. Prompt messages during reading input and displaying output

5. Error handling mechanisms for failures in system calls

6. Incremental program development

7. Modularity

8. All possible test cases in output

Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

AIM:
To develop a C program to implement the cp, ls, grep commands (with some options) using system calls.

cp command: basic cp, -i
To copy a file into another
ls command: basic ls, -1, -R
To list all files in the directory
grep command: basic grep, -c, -v, -n
To search the given pattern in the file

Procedure for cp:

1. The arguments should be obtained in command line and error messages should be printed if they are not sufficient.

2. Use open, read, write, creat ,close system calls to do the following.

mycp sourcefilename destinationfilename

-copies source file to destination file

3. The failure messages for opening a file, creating a file should be intimated.

Note: mycp is the user programs implementing cp.

Procedure for ls:
1. To view the files in a directory include dirent.h that helps for opening, reading, closing a directory.

2. Open the user named directory giving specific path using opendir system call. This returns a pointer to a DIR data structure that represents a directory.

3. Can even use "." to represent the current working directory.

4. Traverse the directory entries using readdir system call. readdir () returns a pointer to a dirent structure whose member d_name contains the name of the current file.

5. Output the entries of directory.

6. Close the directory pointer

NOTE: Use open, read, write, creat ,close, opendir, readdir, closedir system calls wherever necessary.

Procedure for grep:
1. Open the command line specified file using the required system call.

2. Read the contents iteratively till the end of the file and compare it with the pattern you are searching for.

3. If word found print the line on to the display.

4. Count the number of occurrences and display it finally.

5. Close the file descriptor.

NOTE: Use open, read, write, creat ,close system calls wherever necessary.

SAMPLE INPUT/OUTPUT:

cp:

Source.txt:
SSN COLLEGE OF ENGINEERING

target.txt:
SSN NAGAR
KALAVAKKAM

```
$ ./mycp source.txt target.txt
FILE COPIED!
```

ls:

```
$ ./myls lab
```

OUTPUT:

```
.
..
diros
diros.zip
Ex-3-cp-cat.doc
Ex-3-cp-cat.pdf
Ex-4-ls-grep.doc
fork.pdf
grep.doc
prgs.doc
sys-call prgs.doc
```

grep:

```
$ ./mygrep pattern  filename
```

OUTPUT:
Display the contents of the file that has the pattern in it

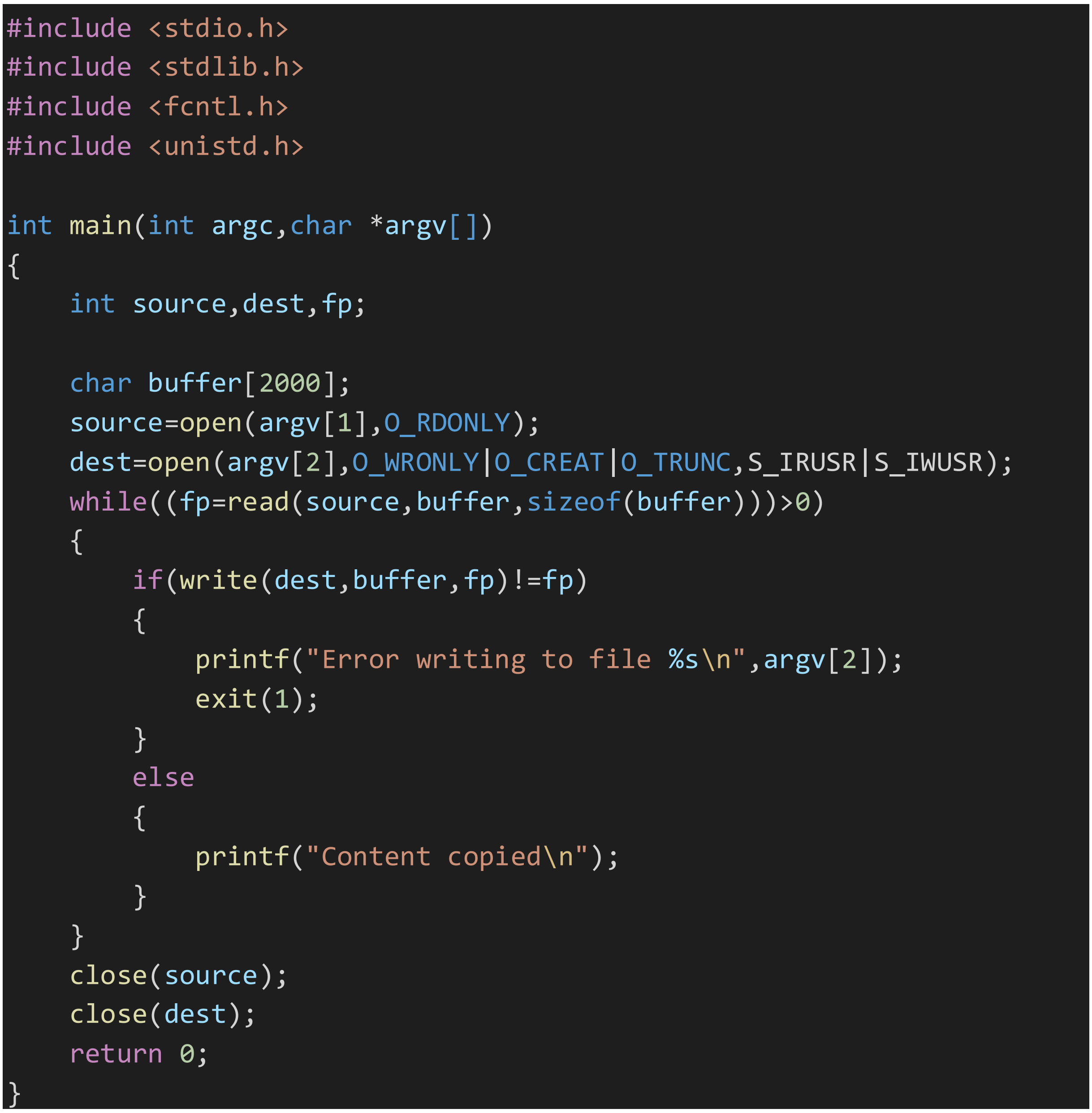
Ex-no: 2
Date:08-03-2023

Name: M.Rohith
3122 21 5001 085

1. C Program to implement cp and cp -i commands using system call

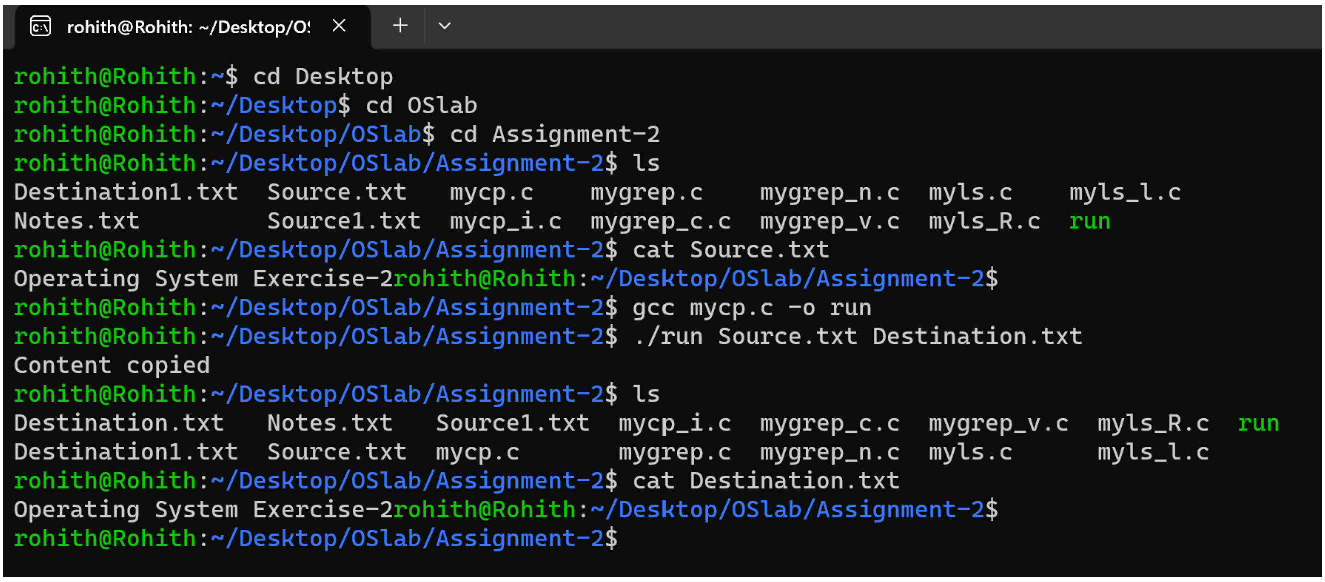
mycp.c

Program code:



```c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

int main(int argc,char *argv[])
{
    int source,dest,fp;

    char buffer[2000];
    source=open(argv[1],O_RDONLY);
    dest=open(argv[2],O_WRONLY|O_CREAT|O_TRUNC,S_IRUSR|S_IWUSR);
    while((fp=read(source,buffer,sizeof(buffer)))>0)
    {
        if(write(dest,buffer,fp)!=fp)
        {
            printf("Error writing to file %s\n",argv[2]);
            exit(1);
        }
        else
        {
            printf("Content copied\n");
        }
    }
    close(source);
    close(dest);
    return 0;
}
```

Output:



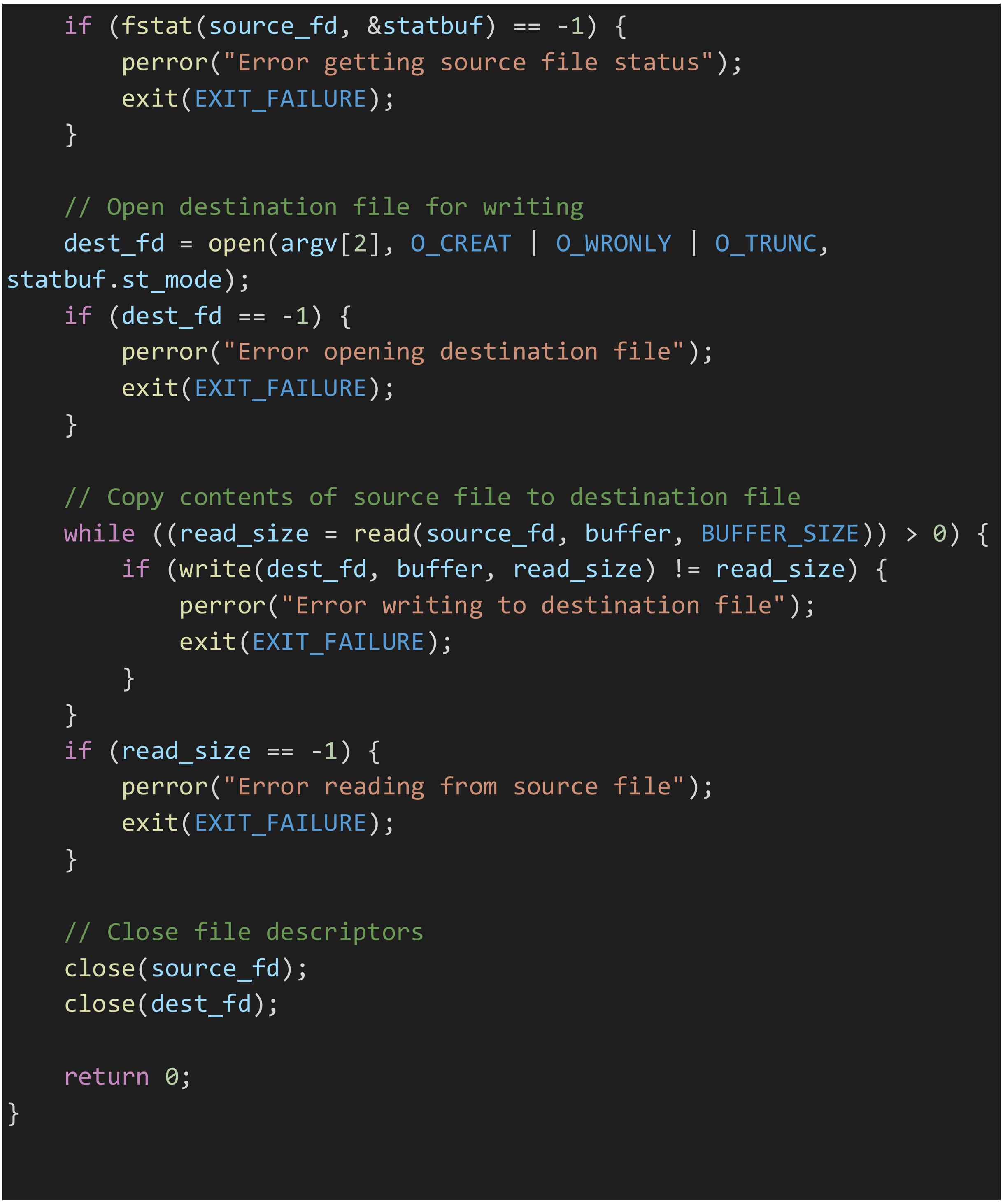
```
rohith@Rohith:~$ cd Desktop
rohith@Rohith:~/Desktop$ cd OSlab
rohith@Rohith:~/Desktop/OSlab$ cd Assignment-2
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination1.txt  Source.txt   mycp.c    mygrep.c    mygrep_n.c  myls.c    myls_l.c
Notes.txt         Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Source.txt
Operating System Exercise-2rohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mycp.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Source.txt Destination.txt
Content copied
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Destination.txt
Operating System Exercise-2rohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

https://www.geeksforgeeks.org/what-is-open-api-in-unix/

https://www.geeksforgeeks.org/input-output-system-calls-c-create-open-close-read-write/

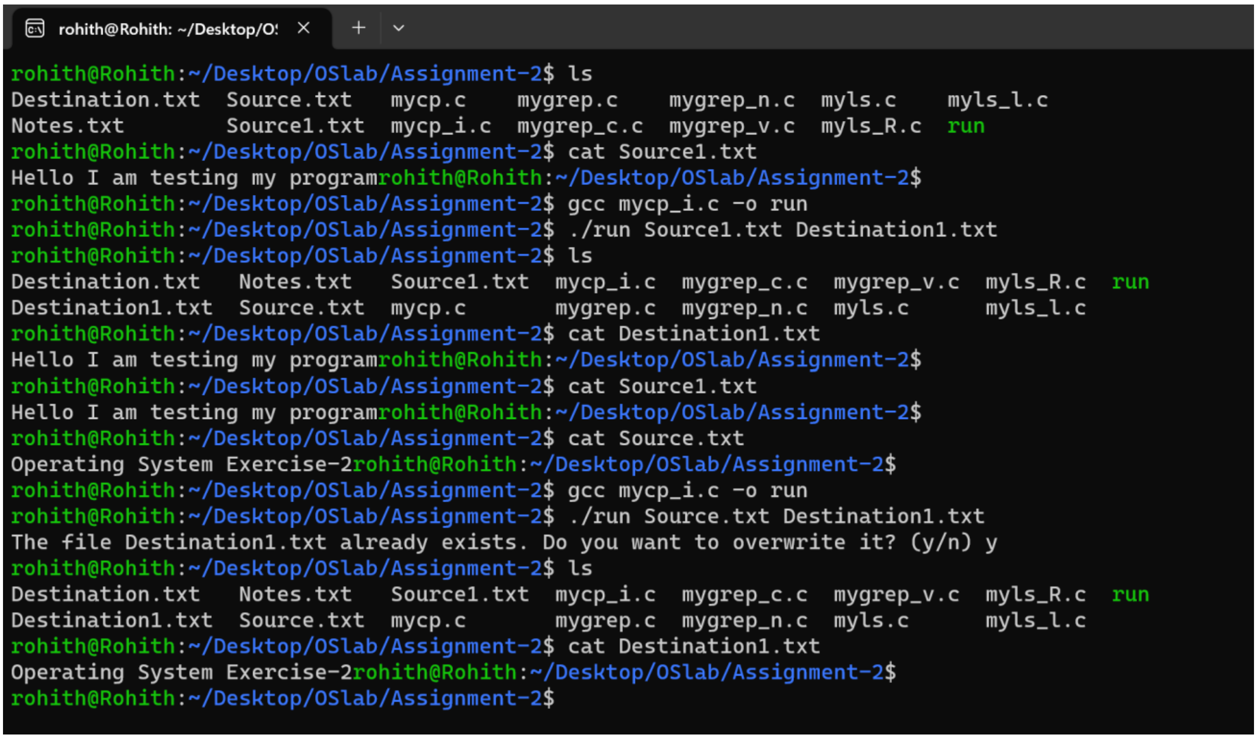
Program code for mycp-i:

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

#define BUFFER_SIZE 4096

int main(int argc, char *argv[]) {
    int source_fd, dest_fd, read_size;
    char buffer[BUFFER_SIZE];
    struct stat statbuf;

    if (argc != 3) {
        printf("Usage: mycp-i source_file dest_file\n");
        exit(EXIT_FAILURE);
    }

    // Check if destination file already exists and prompt for
confirmation
    if (access(argv[2], F_OK) == 0) {
        printf("The file %s already exists. Do you want to overwrite
it? (y/n) ", argv[2]);
        char response;
        scanf("%c", &response);
        if (response != 'y' && response != 'Y') {
            exit(EXIT_SUCCESS);
        }
    }

    // Open source file for reading
    source_fd = open(argv[1], O_RDONLY);
    if (source_fd == -1) {
        perror("Error opening source file");
        exit(EXIT_FAILURE);
    }

    // Get file status to determine permissions of source file
```



```
    if (fstat(source_fd, &statbuf) == -1) {
        perror("Error getting source file status");
        exit(EXIT_FAILURE);
    }

    // Open destination file for writing
    dest_fd = open(argv[2], O_CREAT | O_WRONLY | O_TRUNC,
statbuf.st_mode);
    if (dest_fd == -1) {
        perror("Error opening destination file");
        exit(EXIT_FAILURE);
    }

    // Copy contents of source file to destination file
    while ((read_size = read(source_fd, buffer, BUFFER_SIZE)) > 0) {
        if (write(dest_fd, buffer, read_size) != read_size) {
            perror("Error writing to destination file");
            exit(EXIT_FAILURE);
        }
    }
    if (read_size == -1) {
        perror("Error reading from source file");
        exit(EXIT_FAILURE);
    }

    // Close file descriptors
    close(source_fd);
    close(dest_fd);

    return 0;
}
```

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt  Source.txt   mycp.c    mygrep.c    mygrep_n.c  myls.c    myls_l.c
Notes.txt        Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Source1.txt
Hello I am testing my programrohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mycp_i.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Source1.txt Destination1.txt
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Destination1.txt
Hello I am testing my programrohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Source1.txt
Hello I am testing my programrohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Source.txt
Operating System Exercise-2rohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mycp_i.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Source.txt Destination1.txt
The file Destination1.txt already exists. Do you want to overwrite it? (y/n) y
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Destination1.txt
Operating System Exercise-2rohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

https://www.geeksforgeeks.org/access-command-in-linux-with-examples/

https://pubs.opengroup.org/onlinepubs/009696699/functions/fstat.html

https://www.tutorialspoint.com/c_standard_library/c_function_perror.htm

Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

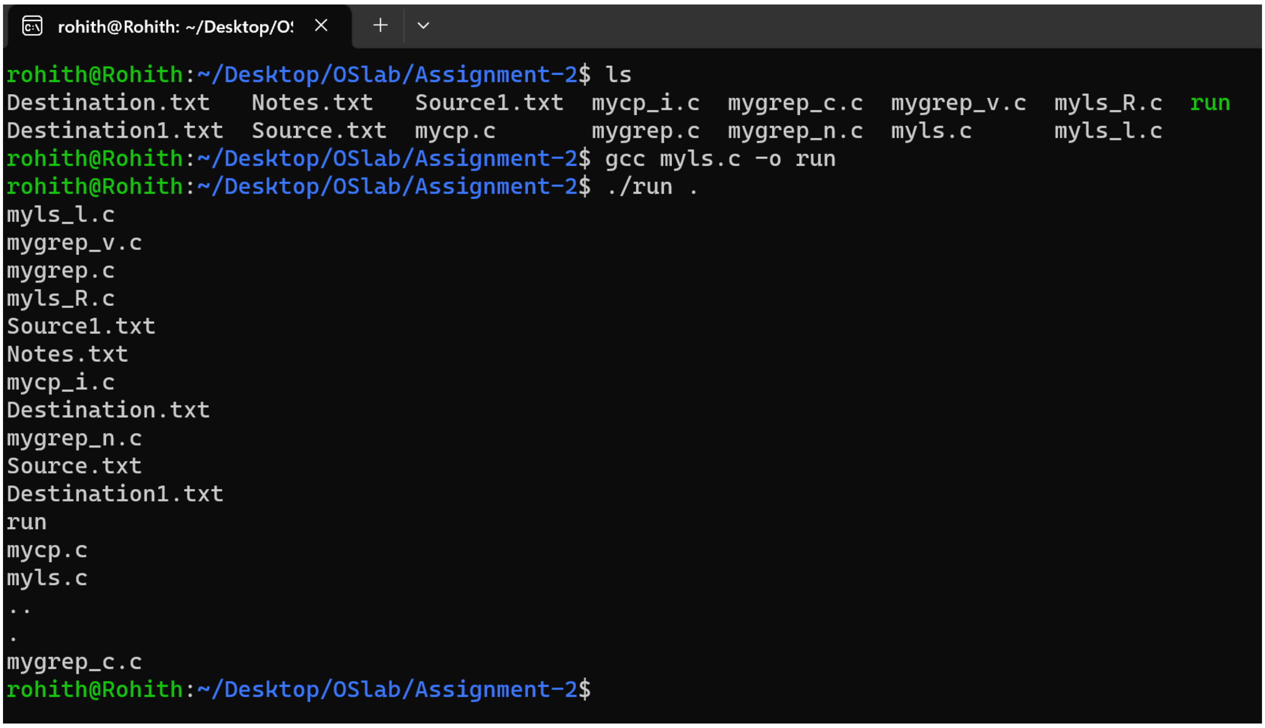
2. C program to implement ls, ls -l and ls -R commands using system call

Program code:

myls.c

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

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

    if(argc != 2) {
        printf("Usage: %s <directory>\n", argv[0]);
        exit(1);
    }

    if((dir = opendir(argv[1])) == NULL) {
        perror("opendir error");
        exit(1);
    }

    while((dp = readdir(dir)) != NULL) {
        printf("%s\n", dp->d_name);
    }

    closedir(dir);
    exit(0);
}
```

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc myls.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run .
myls_l.c
mygrep_v.c
mygrep.c
myls_R.c
Source1.txt
Notes.txt
mycp_i.c
Destination.txt
mygrep_n.c
Source.txt
Destination1.txt
run
mycp.c
myls.c
..
.
mygrep_c.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

https://pubs.opengroup.org/onlinepubs/7908799/xsh/dirent.h.html

Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

myls _l.c

Program code:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <dirent.h>
#include <sys/stat.h>
#include <unistd.h>
#include <pwd.h>
#include <grp.h>
#include <time.h>

int main(int argc, char *argv[]) {
    struct dirent *dp;
    DIR *dir;
    struct stat fileStat;
    struct passwd *pw;
    struct group *gr;
    char dateTime[80];

    if(argc != 2) {
        printf("Usage: %s <directory>\n", argv[0]);
        exit(1);
    }

    if((dir = opendir(argv[1])) == NULL) {
        perror("opendir error");
        exit(1);
    }

    while((dp = readdir(dir)) != NULL) {
        char path[1024];
        sprintf(path, "%s/%s", argv[1], dp->d_name);

        if(stat(path, &fileStat) < 0) {
            perror("stat error");
            exit(1);
        }

        pw = getpwuid(fileStat.st_uid);
        gr = getgrgid(fileStat.st_gid);
        strftime(dateTime, 80, "%b %d %H:%M", 
localtime(&fileStat.st_mtime));

        printf("%s %ld %s %s %ld %s %s\n",
               (S_ISDIR(fileStat.st_mode)) ? "d" : "-",
               (long) fileStat.st_nlink,
               (pw == NULL) ? "" : pw->pw_name,
               (gr == NULL) ? "" : gr->gr_name,
               (long) fileStat.st_size,
               dateTime,
               dp->d_name);
    }

    closedir(dir);
    exit(0);
}
```

https://pubs.opengroup.org/onlinepubs/7908799/xsh/dirent.h.html

https://pubs.opengroup.org/onlinepubs/009695399/basedefs/pwd.h.html

https://pubs.opengroup.org/onlinepubs/009695399/basedefs/grp.h.html

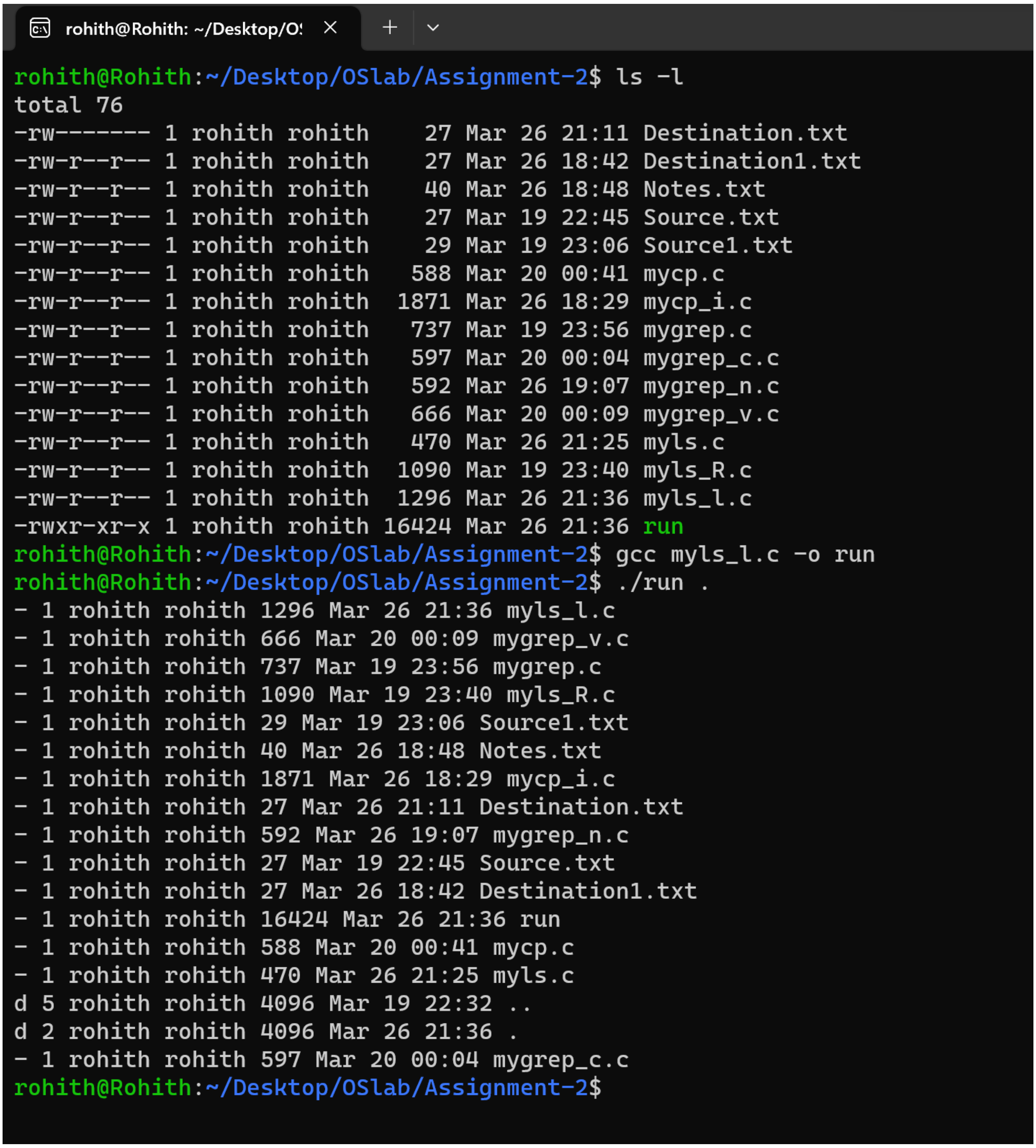
https://www.geeksforgeeks.org/sprintf-in-c/

https://www.geeksforgeeks.org/strftime-function-in-c/

https://www.gnu.org/software/libc/manual/html_node/Testing-File-Type.html

https://linuxhint.com/stat-system-call-linux/

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls -l
total 76
-rw------- 1 rohith rohith    27 Mar 26 21:11 Destination.txt
-rw-r--r-- 1 rohith rohith    27 Mar 26 18:42 Destination1.txt
-rw-r--r-- 1 rohith rohith    40 Mar 26 18:48 Notes.txt
-rw-r--r-- 1 rohith rohith    27 Mar 19 22:45 Source.txt
-rw-r--r-- 1 rohith rohith    29 Mar 19 23:06 Source1.txt
-rw-r--r-- 1 rohith rohith   588 Mar 20 00:41 mycp.c
-rw-r--r-- 1 rohith rohith  1871 Mar 26 18:29 mycp_i.c
-rw-r--r-- 1 rohith rohith   737 Mar 19 23:56 mygrep.c
-rw-r--r-- 1 rohith rohith   597 Mar 20 00:04 mygrep_c.c
-rw-r--r-- 1 rohith rohith   592 Mar 26 19:07 mygrep_n.c
-rw-r--r-- 1 rohith rohith   666 Mar 20 00:09 mygrep_v.c
-rw-r--r-- 1 rohith rohith   470 Mar 26 21:25 myls.c
-rw-r--r-- 1 rohith rohith  1090 Mar 19 23:40 myls_R.c
-rw-r--r-- 1 rohith rohith  1296 Mar 26 21:36 myls_l.c
-rwxr-xr-x 1 rohith rohith 16424 Mar 26 21:36 run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc myls_l.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run .
- 1 rohith rohith 1296 Mar 26 21:36 myls_l.c
- 1 rohith rohith 666 Mar 20 00:09 mygrep_v.c
- 1 rohith rohith 737 Mar 19 23:56 mygrep.c
- 1 rohith rohith 1090 Mar 19 23:40 myls_R.c
- 1 rohith rohith 29 Mar 19 23:06 Source1.txt
- 1 rohith rohith 40 Mar 26 18:48 Notes.txt
- 1 rohith rohith 1871 Mar 26 18:29 mycp_i.c
- 1 rohith rohith 27 Mar 26 21:11 Destination.txt
- 1 rohith rohith 592 Mar 26 19:07 mygrep_n.c
- 1 rohith rohith 27 Mar 19 22:45 Source.txt
- 1 rohith rohith 27 Mar 26 18:42 Destination1.txt
- 1 rohith rohith 16424 Mar 26 21:36 run
- 1 rohith rohith 588 Mar 20 00:41 mycp.c
- 1 rohith rohith 470 Mar 26 21:25 myls.c
d 5 rohith rohith 4096 Mar 19 22:32 ..
d 2 rohith rohith 4096 Mar 26 21:36 .
- 1 rohith rohith 597 Mar 20 00:04 mygrep_c.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

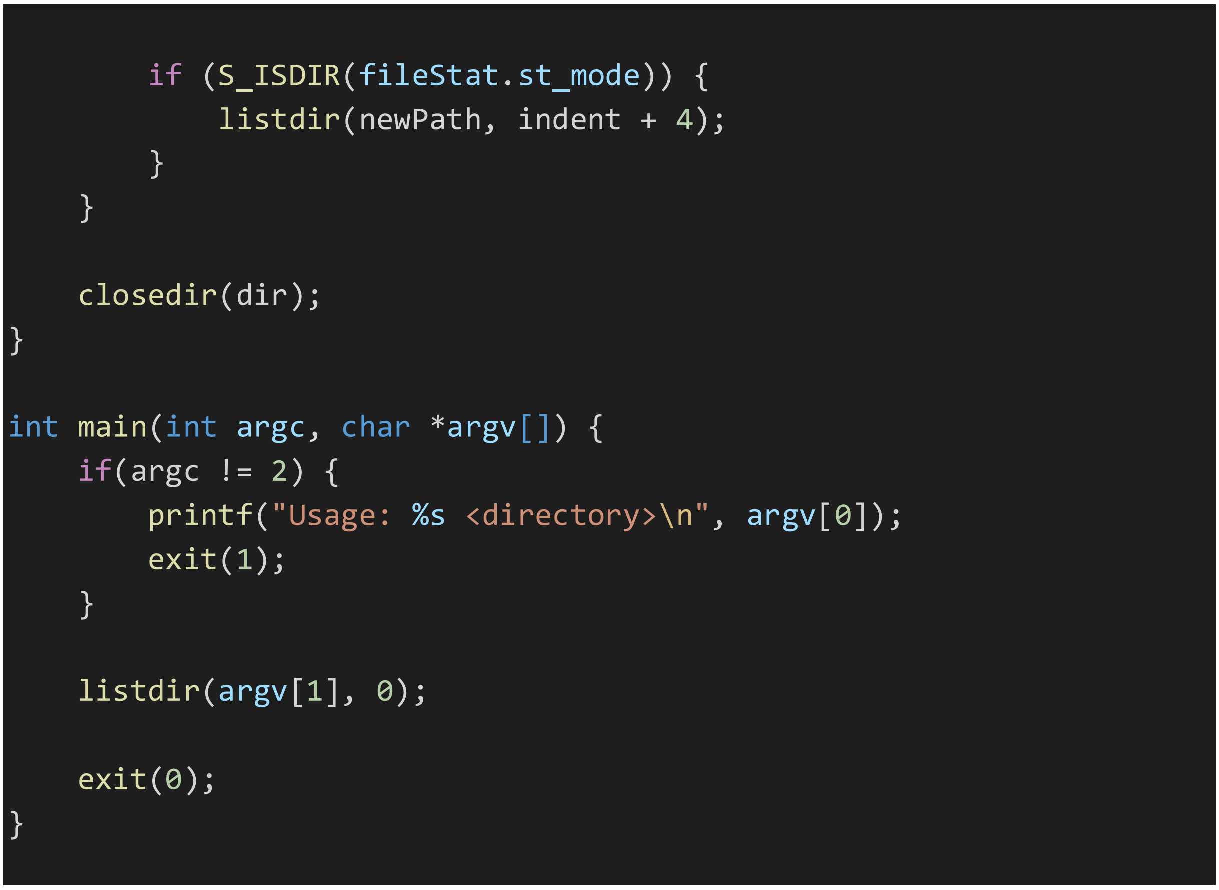
Ex-no: 2
Date:08-03-2023

Name: M.Rohith
3122 21 5001 085

myls_R.c

Program code:

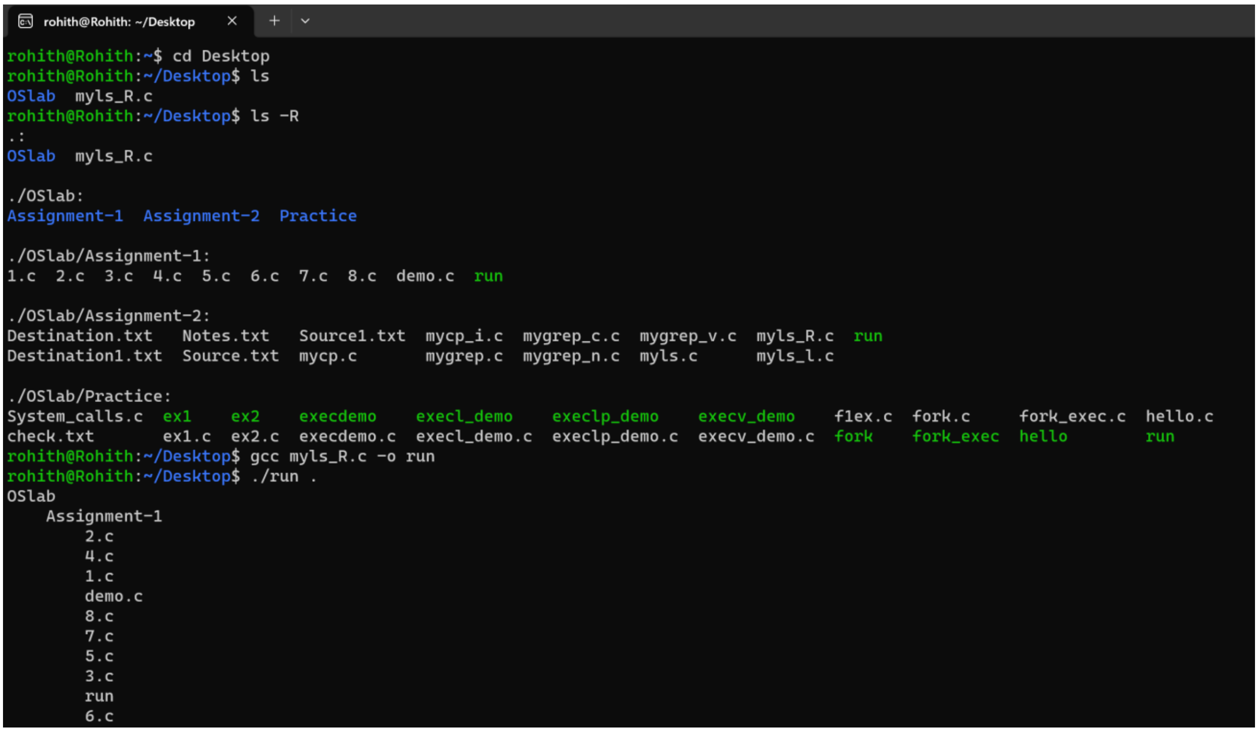
```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <dirent.h>
#include <sys/stat.h>
#include <unistd.h>
#include <pwd.h>
#include <grp.h>
#include <time.h>
#include <string.h>

void listdir(const char *path, int indent) {
    struct dirent *dp;
    DIR *dir;
    struct stat fileStat;
    struct passwd *pw;
    struct group *gr;
    char dateTime[80];
    char newPath[1024];

    if((dir = opendir(path)) == NULL) {
        perror("opendir error");
        exit(1);
    }

    while((dp = readdir(dir)) != NULL) {
        if (strcmp(dp->d_name, ".") == 0 || strcmp(dp->d_name, "..") == 0) {
            continue;
        }

        printf("%*s%s\n", indent, "", dp->d_name);

        sprintf(newPath, "%s/%s", path, dp->d_name);

        if(stat(newPath, &fileStat) < 0) {
            perror("stat error");
            exit(1);
        }
```



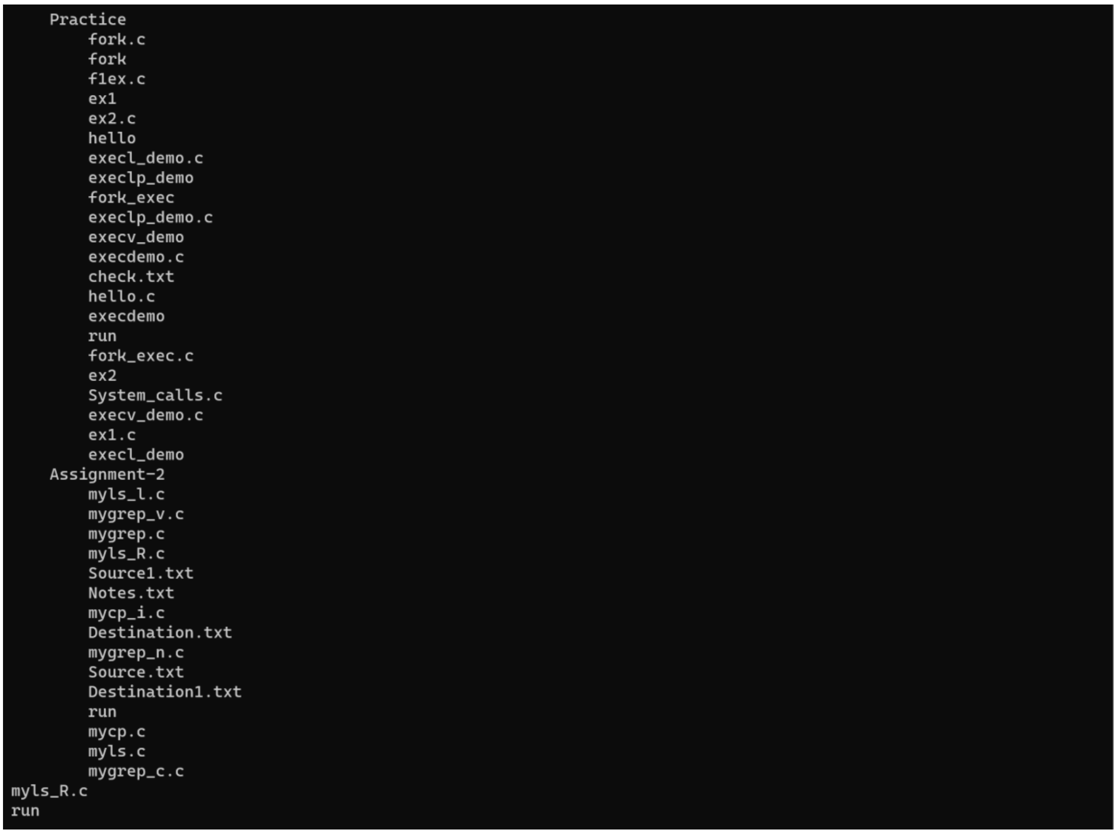
```
        if (S_ISDIR(fileStat.st_mode)) {
            listdir(newPath, indent + 4);
        }
    }

    closedir(dir);
}

int main(int argc, char *argv[]) {
    if(argc != 2) {
        printf("Usage: %s <directory>\n", argv[0]);
        exit(1);
    }

    listdir(argv[1], 0);

    exit(0);
}
```

https://stackoverflow.com/questions/7899119/what-does-s-mean-in-printf

Output:



```
rohith@Rohith:~$ cd Desktop
rohith@Rohith:~/Desktop$ ls
OSlab  myls_R.c
rohith@Rohith:~/Desktop$ ls -R
.:
OSlab  myls_R.c

./OSlab:
Assignment-1  Assignment-2  Practice

./OSlab/Assignment-1:
1.c  2.c  3.c  4.c  5.c  6.c  7.c  8.c  demo.c  run

./OSlab/Assignment-2:
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c

./OSlab/Practice:
System_calls.c  ex1    ex2    execdemo    execl_demo    execlp_demo    execv_demo    f1ex.c  fork.c     fork_exec.c  hello.c
check.txt       ex1.c  ex2.c  execdemo.c  execl_demo.c  execlp_demo.c  execv_demo.c  fork    fork_exec  hello        run
rohith@Rohith:~/Desktop$ gcc myls_R.c -o run
rohith@Rohith:~/Desktop$ ./run .
OSlab
    Assignment-1
        2.c
        4.c
        1.c
        demo.c
        8.c
        7.c
        5.c
        3.c
        run
        6.c
```



```
    Practice
        fork.c
        fork
        f1ex.c
        ex1
        ex2.c
        hello
        execl_demo.c
        execlp_demo
        fork_exec
        execlp_demo.c
        execv_demo
        execdemo.c
        check.txt
        hello.c
        execdemo
        run
        fork_exec.c
        ex2
        System_calls.c
        execv_demo.c
        ex1.c
        execl_demo
    Assignment-2
        myls_l.c
        mygrep_v.c
        mygrep.c
        myls_R.c
        Source1.txt
        Notes.txt
        mycp_i.c
        Destination.txt
        mygrep_n.c
        Source.txt
        Destination1.txt
        run
        mycp.c
        myls.c
        mygrep_c.c
myls_R.c
run
```

Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

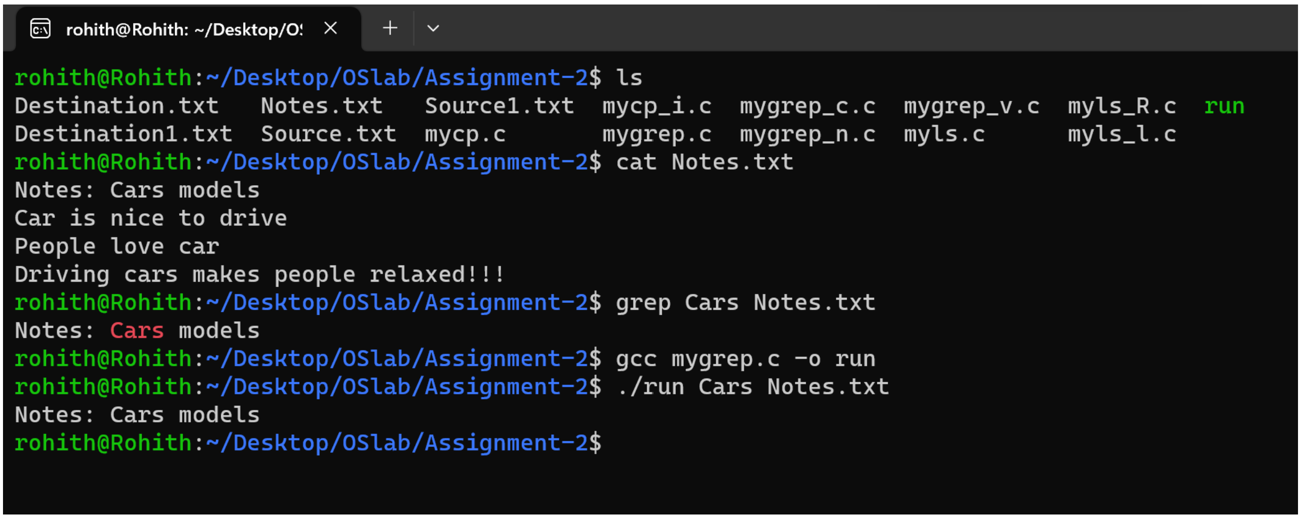
3. C program to implement the grep, grep -c, grep -v, grep -n commands using system call

mygrep.c

Program code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

int main(int argc,char *argv[])
{
    int fd,n;
    char buffer[1024];
    char *pattern=argv[1];
    char *filename=argv[2];
    off_t offset=0;

    fd=open(filename,O_RDONLY);

    while((n=read(fd,buffer,sizeof(buffer)))>0)
    {
        for(int i=0;i<n;i++)
        {
            if(buffer[i]=='\n')
            {
                if(strstr(buffer+offset,pattern)!=NULL)
                {
                    printf("%.*s\n",(int)(i-offset),buffer+offset);
                }
                offset=i+1;
            }
        }
        offset=n-offset;
        memmove(buffer,buffer+offset,offset);
    }
    close(fd);
    return 0;
}
```

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Notes.txt
Notes: Cars models
Car is nice to drive
People love car
Driving cars makes people relaxed!!!
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ grep Cars Notes.txt
Notes: Cars models
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mygrep.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Cars Notes.txt
Notes: Cars models
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

https://www.ibm.com/docs/hu/aix/7.2?topic=files-typesh-file

https://stackoverflow.com/questions/7899119/what-does-s-mean-in-printf

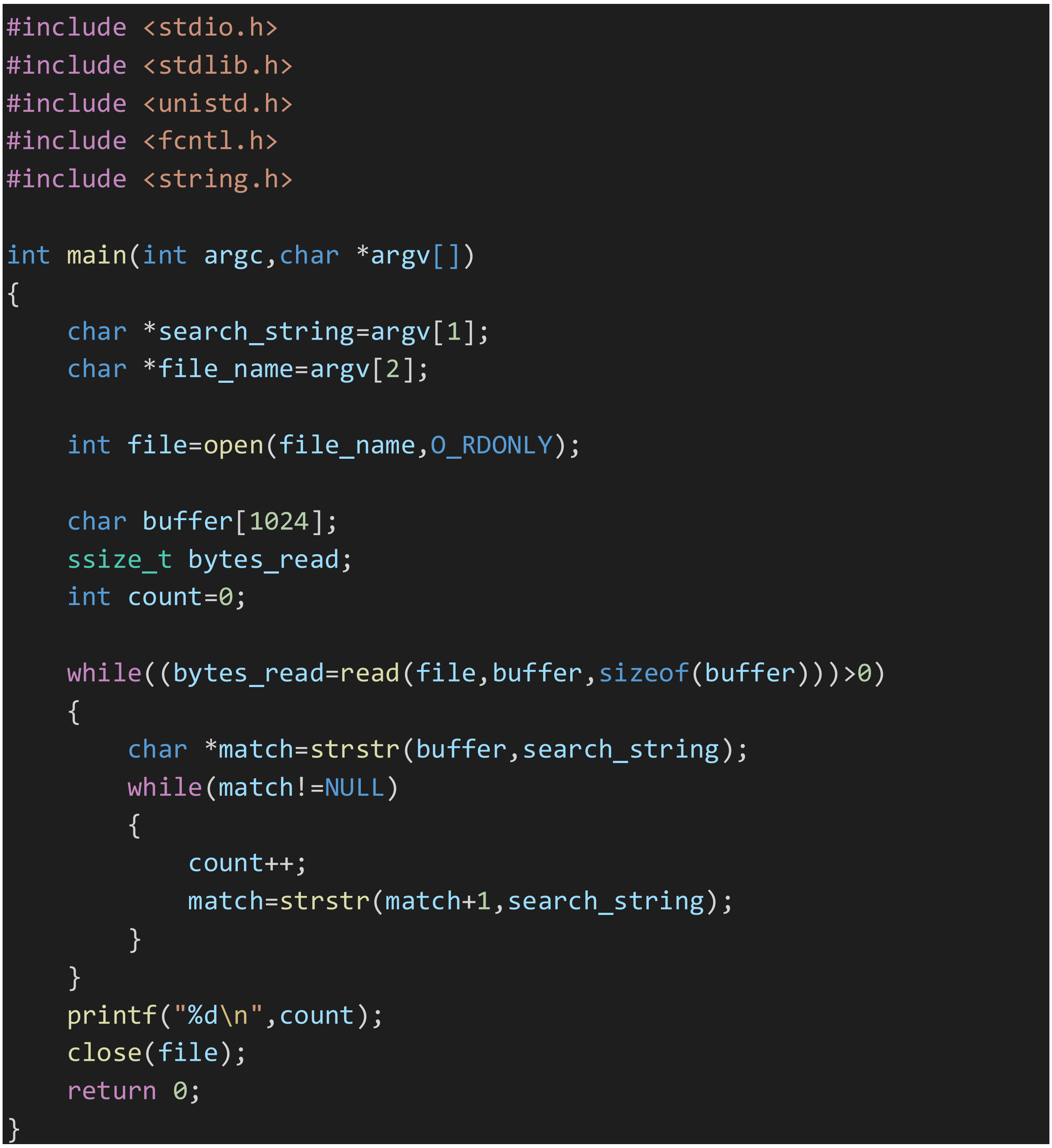
https://www.geeksforgeeks.org/strstr-in-ccpp/

https://www.geeksforgeeks.org/memmove-in-cc/

Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

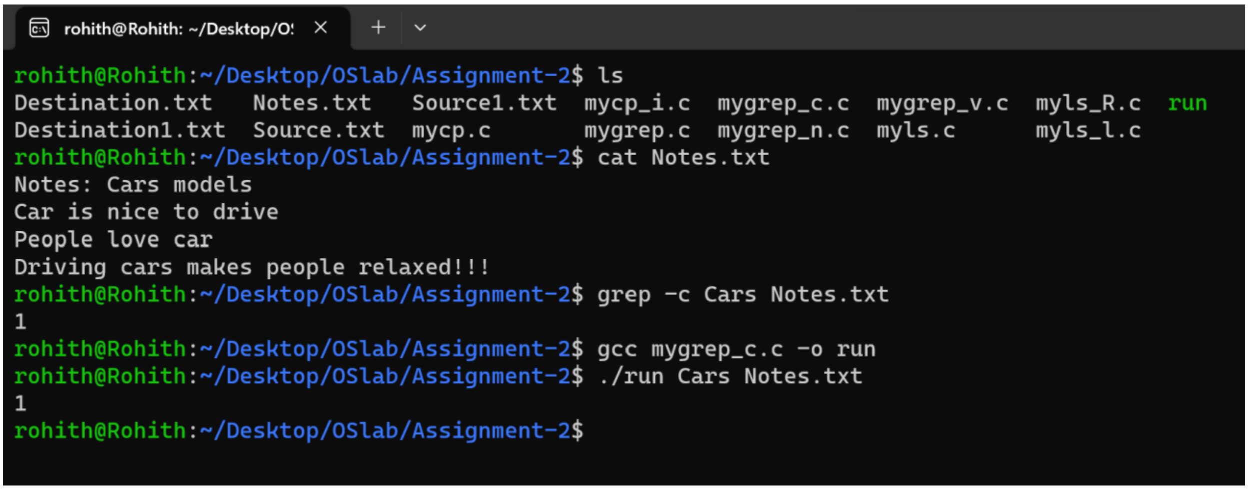
mygrep_c.c

Program code:



```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

int main(int argc,char *argv[])
{
    char *search_string=argv[1];
    char *file_name=argv[2];

    int file=open(file_name,O_RDONLY);

    char buffer[1024];
    ssize_t bytes_read;
    int count=0;

    while((bytes_read=read(file,buffer,sizeof(buffer)))>0)
    {
        char *match=strstr(buffer,search_string);
        while(match!=NULL)
        {
            count++;
            match=strstr(match+1,search_string);
        }
    }
    printf("%d\n",count);
    close(file);
    return 0;
}
```

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Notes.txt
Notes: Cars models
Car is nice to drive
People love car
Driving cars makes people relaxed!!!
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ grep -c Cars Notes.txt
1
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mygrep_c.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Cars Notes.txt
1
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

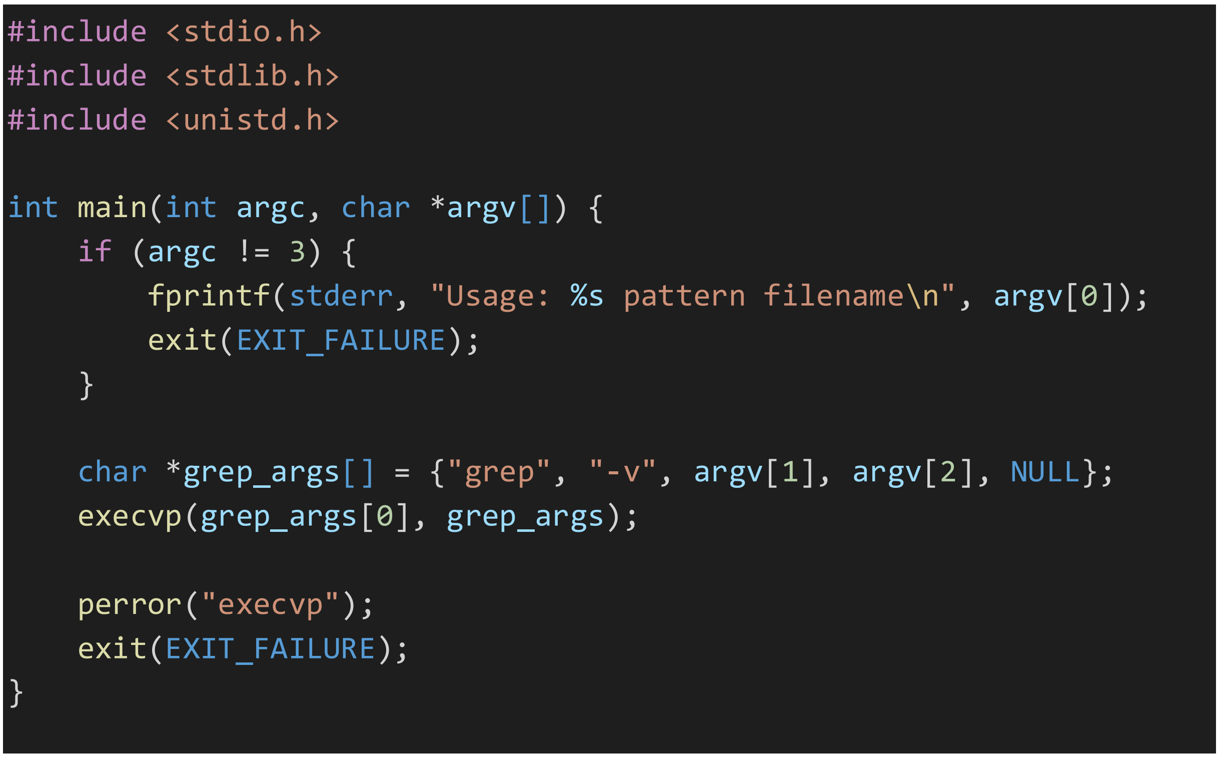
https://www.ibm.com/docs/hu/aix/7.2?topic=files-typesh-file

https://www.geeksforgeeks.org/strstr-in-ccpp/

Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

mygrep_v.c

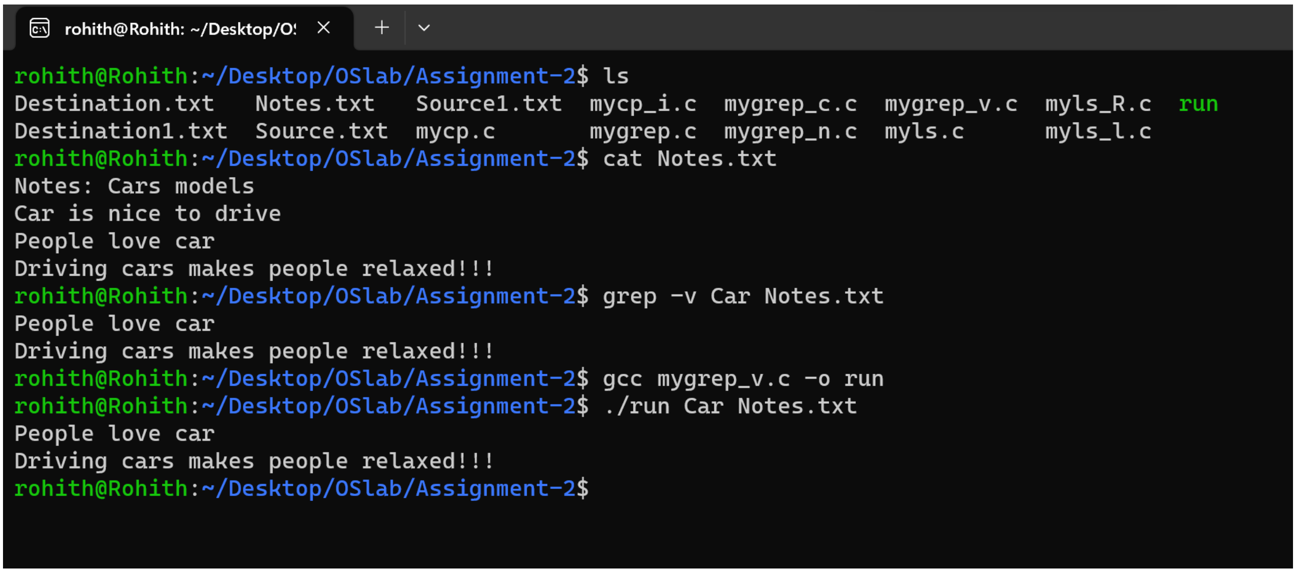
Program code:



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

int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: %s pattern filename\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    char *grep_args[] = {"grep", "-v", argv[1], argv[2], NULL};
    execvp(grep_args[0], grep_args);

    perror("execvp");
    exit(EXIT_FAILURE);
}
```

https://www.geeksforgeeks.org/exec-family-of-functions-in-c/

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Notes.txt
Notes: Cars models
Car is nice to drive
People love car
Driving cars makes people relaxed!!!
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ grep -v Car Notes.txt
People love car
Driving cars makes people relaxed!!!
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mygrep_v.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Car Notes.txt
People love car
Driving cars makes people relaxed!!!
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

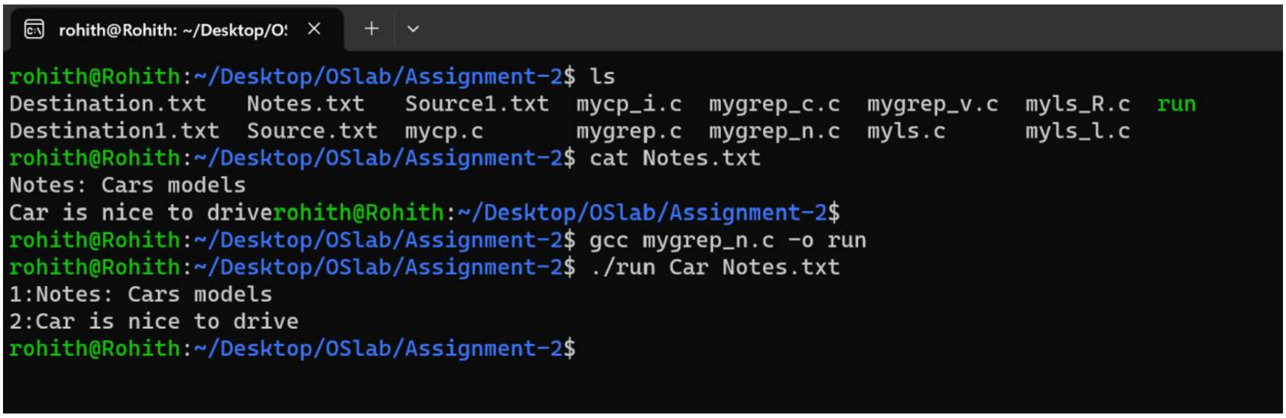
Ex-no: 2 Name: M.Rohith
Date:08-03-2023 3122 21 5001 085

mygrep_n.c

Program code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Usage: %s pattern filename\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    int filedesc = open(argv[2], O_RDONLY);
    if (filedesc == -1) {
        perror("open");
        exit(EXIT_FAILURE);
    }

    dup2(filedesc, STDIN_FILENO);
    close(filedesc);

    char *args[] = {"grep", "-n", argv[1], NULL};
    execvp(args[0], args);

    perror("execvp");
    exit(EXIT_FAILURE);
}
```

Output:



```
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ls
Destination.txt   Notes.txt   Source1.txt  mycp_i.c  mygrep_c.c  mygrep_v.c  myls_R.c  run
Destination1.txt  Source.txt  mycp.c       mygrep.c  mygrep_n.c  myls.c      myls_l.c
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ cat Notes.txt
Notes: Cars models
Car is nice to driverohith@Rohith:~/Desktop/OSlab/Assignment-2$
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ gcc mygrep_n.c -o run
rohith@Rohith:~/Desktop/OSlab/Assignment-2$ ./run Car Notes.txt
1:Notes: Cars models
2:Car is nice to drive
rohith@Rohith:~/Desktop/OSlab/Assignment-2$
```

https://www.geeksforgeeks.org/dup-dup2-linux-system-call/

https://stackoverflow.com/questions/15102992/what-is-the-difference-between-stdin-and-stdin-fileno