Lab 9 – File System Management

OBJECTIVE

Learn to use open, read, write, close system for file management.

TIME REQUIRED : 3 hrs

PROGRAMMING LANGUAGE : C/C++

SOFTWARE REQUIRED : Ubuntu/Fedora, gcc/gc, Text Editor, Terminal, Windows, Dev

HARDWARE REQUIRED : Core i5 in Computer Labs

FILE SYSTEM MANAGEMENT IN LINUX

File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc. The Linux System calls under this are **open(), read(), write(), close().**

- **open():**
  - It is the system call to open a file.
  - This system call just opens the file, to perform operations such as read and write, we need to execute different system call to perform the operations.
  - **Syntax:**

```
fd = open (file_name, mode, permission);
Example:
fd = open ("file", O_CREAT | O_RDWR, 0777);
```

  Here,
  - file_name is the name to the file to open.
  - mode is used to define the file opening modes such as create, read, write modes.
  - permission is used to define the file permissions.

Return value: Function returns the file descriptor.

- **read():**
  - This system call opens the file in reading mode
  - We can not edit the files with this system call.
  - Multiple processes can execute the read() system call on the same file simultaneously.

Syntax:

```
length = read(file_descriptor , buffer, max_len);
Example:
n = read(0, buff, 50);
```

Here,

- file_descriptor is the file descriptor of the file.
- buffer is the name of the buffer where data is to be stored.
- max_len is the number specifying the maximum amount of that data can be read.

Return value: If successful read returns the number of bytes actually read.

- **write():**
  - This system call opens the file in writing mode
  - We can edit the files with this system call.
  - Multiple processes can not execute the write() system call on the same file simultaneously.

  Syntax:

  ```
  length = write(file_descriptor , buffer, len);
  Example:
  n = write(fd, "Hello world!", 12);
  ```

  Here,

  - file_descriptor is the file descriptor of the file.
  - buffer is the name of the buffer to be stored.
  - len is the length of the data to be written.

  Return value: If successful write() returns the number of bytes actually written.

- **close():**
  - This system call closes the opened file.

  Syntax:

  ```
  int close(int fd);
  ```

  Here,

- fd is the file descriptor of the file to be closed.

Return value: If file closed successfully it returns 0, else it returns -1.

C code to demonstrate example of System call:

Run the following code and write down the outcome of the programs.

Activity 1

Write a program that creates a file with a 4K bytes free space. [2].

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
char buf1[]="LAB ";
char buf2[]="OS Linux";

int main( void)
{
        int fd;
        if ((fd=creat("file.gol", 0666)) < 0) {
                printf("Creation error");
            exit (1);
        }

        if (write(fd, buf1, sizeof(buf1)) < 0)
       printf("Writing error");
       exit(2);
        }

        if (lseek(fd, 4096, SEEK_SET) < 0)
       printf("Positioning error");
       exit(3);
        }

        if (write(fd, buf2, sizeof(buf2)) < 0)
       printf("Writing error");
       exit(2);
        }
}
```

Trace the execution of the program with the help of the following commands:

ls -l
stat file.gol
od -c file.gol

Answer:

```
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ gcc new.c -o tstnew
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ ./tstnew
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ ls -l
total 136
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq     0 00:20 18   مارچ diff.c
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq     9 23:20 17   مارچ File1.txt
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq  4105 01:11 18   مارچ file.gol
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq   620 01:10 18   مارچ new.c
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq    48 00:13 18   مارچ program.bin
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq   479 00:15 18   مارچ readbin.c
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq   262 23:25 17   مارچ readfile.c
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq     0 23:19 17   مارچ test.txt
-rwxrwxr-x 1 yasirsiddiq yasirsiddiq 16960 23:20 17   مارچ tst
-rwxrwxr-x 1 yasirsiddiq yasirsiddiq 16944 01:10 18   مارچ tstnew
-rwxrwxr-x 1 yasirsiddiq yasirsiddiq 16936 23:25 17   مارچ tstr
-rwxrwxr-x 1 yasirsiddiq yasirsiddiq 16928 00:13 18   مارچ tstrb
-rwxrwxr-x 1 yasirsiddiq yasirsiddiq 16928 00:13 18   مارچ tstwb
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq   258 23:20 17   مارچ witefile.c
-rw-rw-r-- 1 yasirsiddiq yasirsiddiq   486 00:20 18   مارچ writebin.c
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ stat fil.gol
stat: cannot stat 'fil.gol': No such file or directory
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ stat file.gol
  File: file.gol
  Size: 4105            Blocks: 16         IO Block: 4096   regular file
Device: 805h/2053d      Inode: 563345      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/yasirsiddiq)   Gid: ( 1000/yasirsiddiq)
Access: 2022-03-18 01:11:03.799134598 +0500
Modify: 2022-03-18 01:11:03.803134727 +0500
Change: 2022-03-18 01:11:03.803134727 +0500
```

```
Access: (0664/-rw-rw-r--)  Uid: ( 1000/yasirsiddiq)   Gid: ( 1000/yasirsiddiq)
Access: 2022-03-18 01:11:03.799134598 +0500
Modify: 2022-03-18 01:11:03.803134727 +0500
Change: 2022-03-18 01:11:03.803134727 +0500
 Birth: -
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ od -c file.gol
0000000   L   A   B      \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000020  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
*
0010000   O   S       L   i   n   u   x  \0
0010011
```

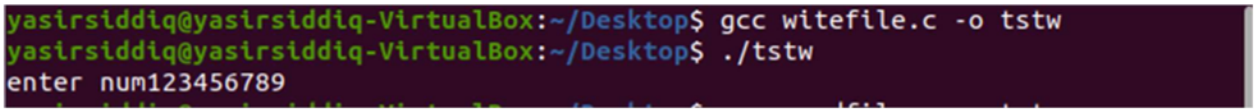
Activity 2:

The following code will open a file named "File1.txt" in writing mode. This will ask user to enter numbers, it will read input from the terminal and save it in the file. Write down the outcome of this code.

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

```c
#include <stdlib.h>

int main()
{
      int num;
      FILE *fptr;
      fptr = fopen("File1.txt","w");

      if(fptr == NULL)
      {
            printf("Error");
            exit(1);
      }

      printf("enter num");
      scanf("%d",&num);

      fprintf(fptr,"%d", num);

      fclose(fptr);

}
```

Answer:



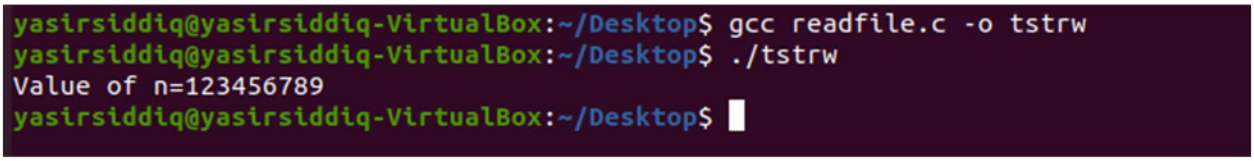
Activity 3:

The following code will open a file named "File1.txt" in read mode. Write down the outcome of this code.

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

int main()
{
      int num;
      FILE *fptr;


      if((fptr = fopen("File1.txt","r")) == NULL)
      {
            printf("Error in opening file!");
            exit(1);
      }


      fscanf(fptr,"%d",&num);

      printf("Value of n=%d \n",num);
      fclose(fptr);
```

```
}
```

Answer:


yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ gcc readfile.c -o tstrw
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$ ./tstrw
Value of n=123456789
yasirsiddiq@yasirsiddiq-VirtualBox:~/Desktop$