Report

File System

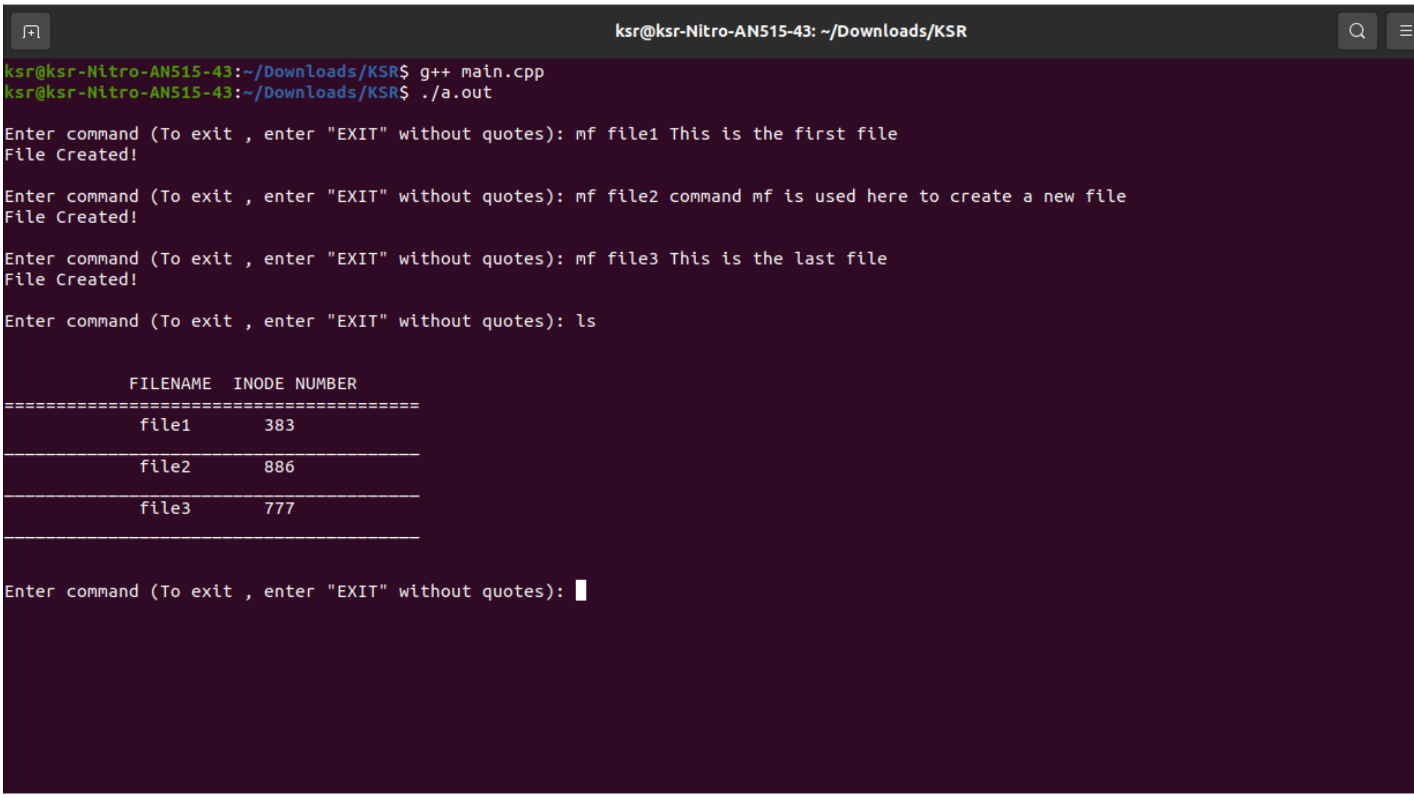
Designed a simple File System having defined constraints using C++. Implemented multiple functions such as create a file, list all files, delete the file, read and write the file.

Input

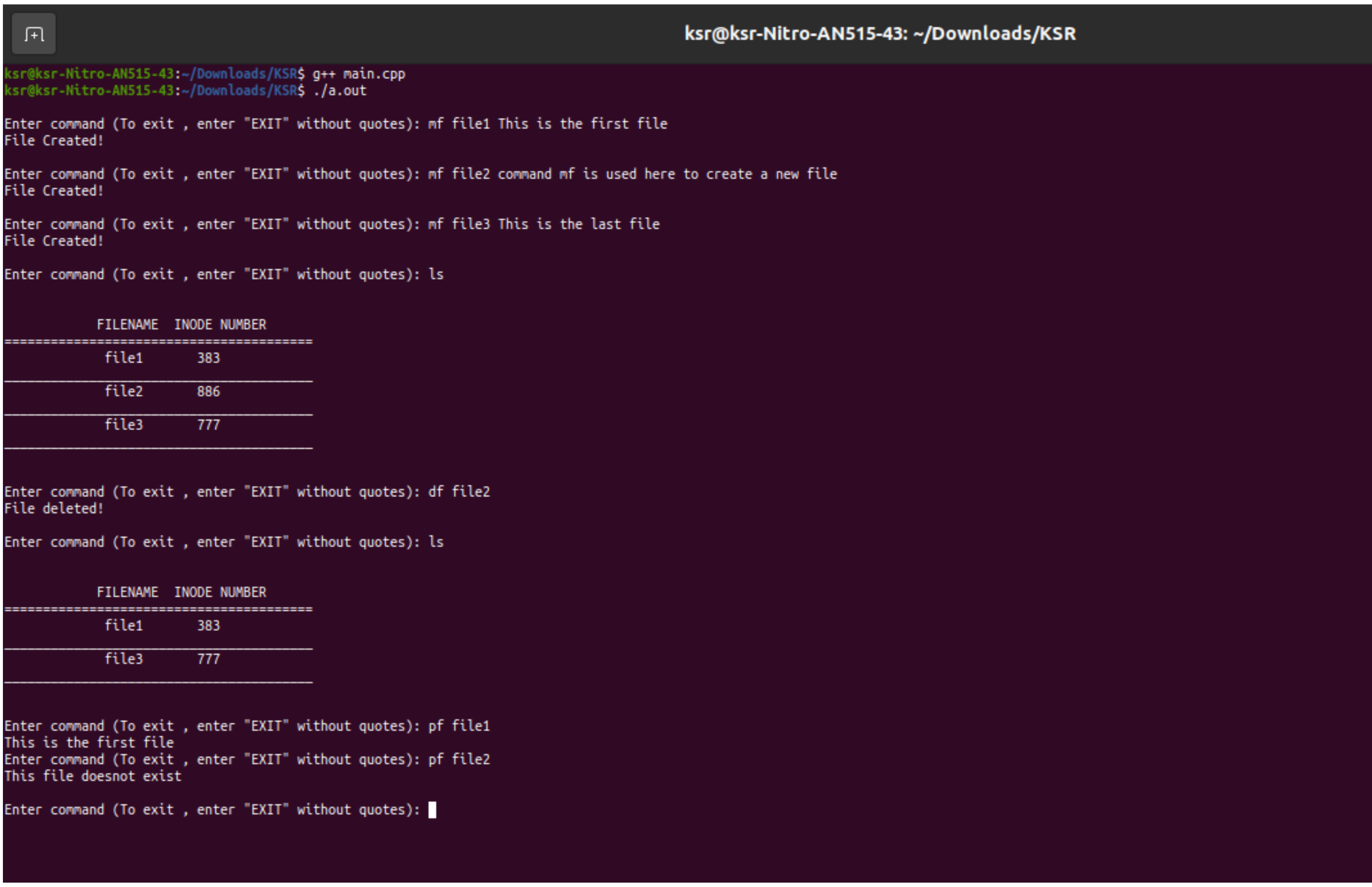
```
    Input is taken through Console.

    Input includes the following commands:
        mf file-name filecontents
        df file-name
        rf file-name1 file-name2
            pf file-name
            ls
```

Commands

```
    mf file-name filecontents
        first check for the uniqueness of the filename
        create a unique inode for file
        insert filename and corresponding inode into the meta data.
        insert the inode and filename into the sets for checking the uniqueness further.
        calculate the number of disk files required and create disk block files with inodeno_blockno

    df file-name
        check if the file with given name is present in the file system
        get the inode of the file to be deleted
        remove the file entry from the meta data
        retrieve the number of disk files created for the file to be deleted
        remove the disk block entry from the file
        delete all the disk files.
        remove the file from the list of files
        remove the inode from the inode list.

    rf file-name1 file-name2
        check whether the new filename is unique
        check if the file with given name is present in the file system
        rename the filename in the metadata
        remove the file from the list of files
        insert the new filename into the list of files

    pf file-name
        check if the file with given name is present in the file system
        get the inode of the file to be displayed
        retrieve the number of disk files created for the file to be displayed
        display all the disk file contents

    ls
        list all the files from the meta_data
```

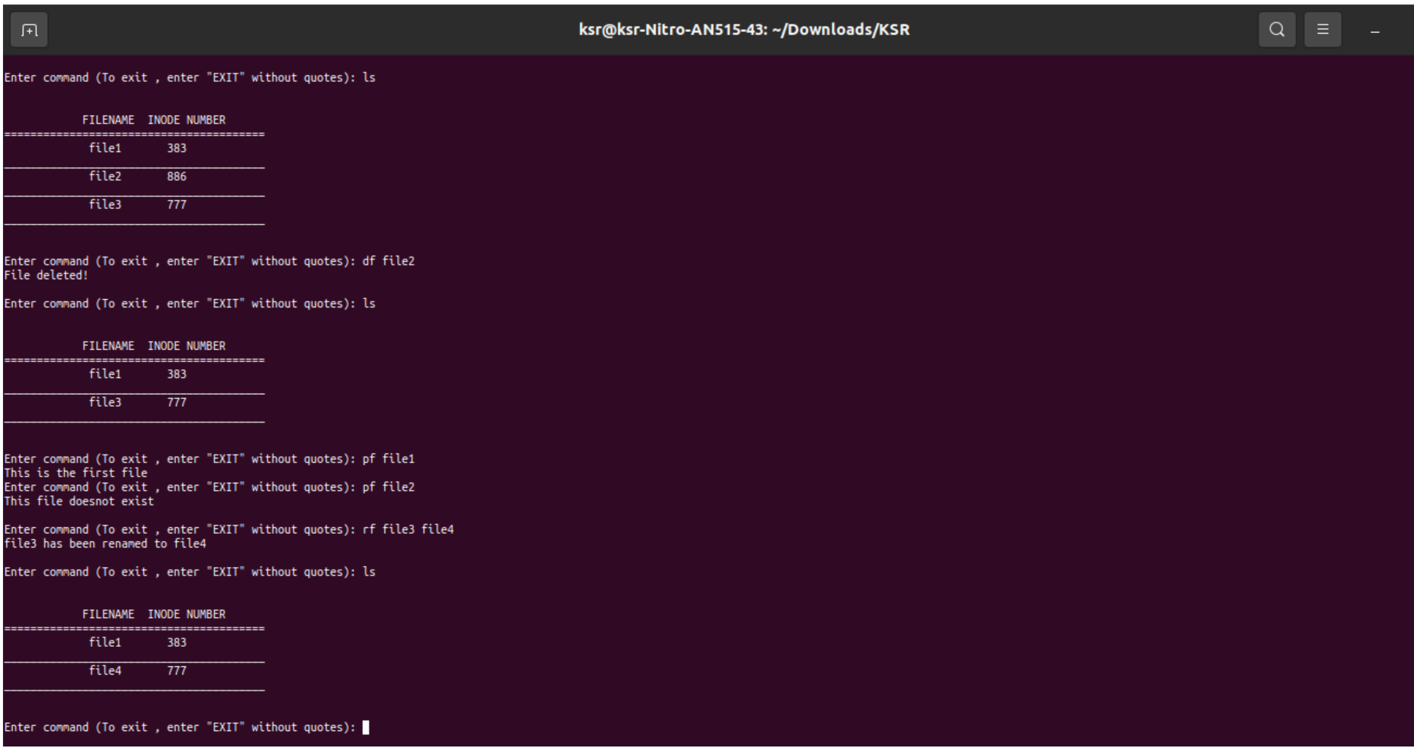


```
ksr@ksr-Nitro-AN515-43:~/Downloads/KSR$ g++ main.cpp
ksr@ksr-Nitro-AN515-43:~/Downloads/KSR$ ./a.out

Enter command (To exit , enter "EXIT" without quotes): mf file1 This is the first file
File Created!

Enter command (To exit , enter "EXIT" without quotes): mf file2 command mf is used here to create a new file
File Created!

Enter command (To exit , enter "EXIT" without quotes): mf file3 This is the last file
File Created!

Enter command (To exit , enter "EXIT" without quotes): ls


            FILENAME  INODE NUMBER
========================================
             file1       383
________________________________________
             file2       886
________________________________________
             file3       777
________________________________________


Enter command (To exit , enter "EXIT" without quotes): 
```



```
ksr@ksr-Nitro-AN515-43:~/Downloads/KSR$ g++ main.cpp
ksr@ksr-Nitro-AN515-43:~/Downloads/KSR$ ./a.out

Enter command (To exit , enter "EXIT" without quotes): mf file1 This is the first file
File Created!

Enter command (To exit , enter "EXIT" without quotes): mf file2 command mf is used here to create a new file
File Created!

Enter command (To exit , enter "EXIT" without quotes): mf file3 This is the last file
File Created!

Enter command (To exit , enter "EXIT" without quotes): ls


            FILENAME  INODE NUMBER
========================================
             file1       383
________________________________________
             file2       886
________________________________________
             file3       777
________________________________________
```

```

Enter command (To exit , enter "EXIT" without quotes): df file2
File deleted!

Enter command (To exit , enter "EXIT" without quotes): ls


            FILENAME  INODE NUMBER
========================================
             file1       383
________________________________________
             file3       777
________________________________________


Enter command (To exit , enter "EXIT" without quotes): pf file1
This is the first file
Enter command (To exit , enter "EXIT" without quotes): pf file2
This file doesnot exist

Enter command (To exit , enter "EXIT" without quotes): 
```



```
Enter command (To exit , enter "EXIT" without quotes): ls


          FILENAME  INODE NUMBER
========================================
           file1      383
________________________________________
           file2      886
________________________________________
           file3      777
________________________________________


Enter command (To exit , enter "EXIT" without quotes): df file2
File deleted!

Enter command (To exit , enter "EXIT" without quotes): ls


          FILENAME  INODE NUMBER
========================================
           file1      383
________________________________________
           file3      777
________________________________________


Enter command (To exit , enter "EXIT" without quotes): pf file1
This is the first file
Enter command (To exit , enter "EXIT" without quotes): pf file2
This file doesnot exist

Enter command (To exit , enter "EXIT" without quotes): rf file3 file4
file3 has been renamed to file4

Enter command (To exit , enter "EXIT" without quotes): ls


          FILENAME  INODE NUMBER
========================================
           file1      383
________________________________________
           file4      777
________________________________________


Enter command (To exit , enter "EXIT" without quotes): █
```

Page Replacement Algorithm Simulator

Build Status

Program to accept number of physical frames, list of page accesses, and the page replacement algorithm and output the number of faults and whether each access was a fault or not. Supports FIFO, LRU, and OPTIMAL algorithms.

Setup

Prerequisites

A C++ compiler, such as g++, is required to build the program.

Building the program

Input file specifications

There must be an input file named PageAccessSequence.txt in the directory. The first line in the file should contain the number of physical frames. Each subsequent line represents one page access, and contains exactly one integer, which represents the page number being accessed.

All values must be non negative and fit in the int data type of the system.

Running the program

Three modes are supported: 1. FIFO - First In First Out Algorithm 2. LRU - Least Recently Used Algorithm 3. OPTIMAL - Optimal Algorithm

To execute in a given mode, say MODE, use

```
$ ./simulator MODE
```

Output

Based on the algorithm, the output will be stored in an output file, in the same directory. If the mode was MODE, the output file will be CS15BTECH11036_MODE.out.

The first line of the output file contains the number of page faults. Each subsequent line corresponds to the page access from the input file. The output is FAULT if there was a page fault, and NOFAULT otherwise.