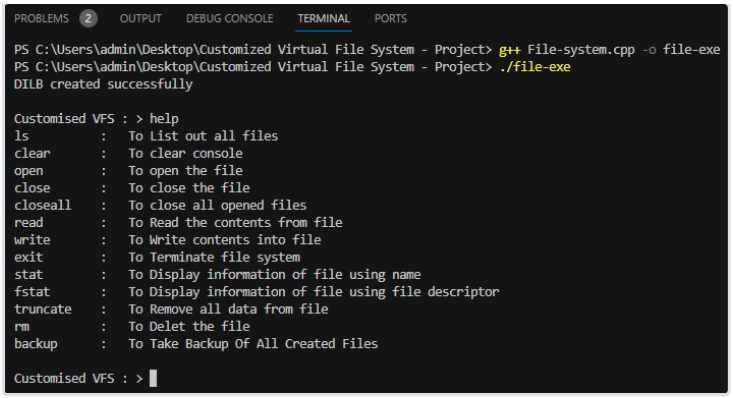
Customized Virtual File System Documentation

- Name of the Project: Customized Virtual File System
- Technology Used: System Programming using c++
- User Interface: Command User Interface
- Platform Required: Windows NT Platform OR Linux Distributions
- Architectural requirement: Intel 32 bit processor
- SDK used : None
- Description of the Project
  - In this project we emulate all data structures which are used by operating system to manage File system oriented tasks.
  - As the name suggest its virtual because we maintain all records in Primary storage.
  - In this project we create all data structures which required for File Subsystems.
  - We provide all implementations of necessary system calls and commands of File subsystem as Open, Close, Read, Write, Lseek, Create, RM, LS, Stat, Fstat etc.
  - While providing the implementations of all above functionality we use our own data structures by referring Algorithms of UNIX operating system.
  - By using this project we can get overview of UFS (UNIX File System) on any platform.
- Data Structures Used in the Project
  - Inode Table,
  - File Table
  - UAREA
  - User File Descriptor Table
  - Super block
  - Disk Inode List Block
  - Data Block
  - Boot Block etc.
- Diagram of Data Structures Used: (Diagram here)
  -
- How to used this Project
  - Commands to start this project
    - ```
      cd Customized virtual File System- project
      g++ File-system.cpp -o file-exe
      ./file-exe
      ```
  - Now you are in Project Customized VFS : >
    - help command will give all list of commands
      - ```
        Customized VFS: > help
        ```
      - Code snippet of terminal after project started



- command to used Customized VFS:
  - man : man command will give description of how to used different commands link read, write, create etc
    - ```
      Customized VFS: > man create
      ```

  - clear : clear command will used to clear the console
    - ```
      Customized VFS: > clear
      ```
  - create: create command is used to create new file with permission 1 (read only), 2(write only) and 3(read and write).
    - ```
      Customised VFS : > create myfile.txt 3
      File is successfully created with file descriptor : 0
      ```
      - ```
        Customized VFS : > create myfile.txt 1 // read only file
        ```
      - ```
        Customized VFS : > create myfile.txt 2 // write only file
        ```
      - ```
        Customized VFS : > create myfile.txt 3 // read and write only file
        ```

- write: write command is used to write in file

```
Customised VFS : > write myfile.txt
Enter the data :
Hello world
Sucessfully : 11 bytes written
Customized VFS: >
```

- read : read command is used to read file and we have to specified how many bytes we want to read.

```
Customized VFS: > read myfile.txt
Hello world
Customized VFS : >
```

- open : open command is used to open file in mode 1 (read mode), 2(write mode), 3 (read and write mode).

```
Customized VFS: > open myfile.txt 3
```

- close : close command is used to close file

```
Customized VFS: > close myfile.txt
```

- ls : ls command will give list of all command which are available.

```
Customised VFS : > ls

File Name       Inode number    File size       Link count
--------------------------------------------------
myfile.txt              1               0               1
--------------------------------------------------

Customised VFS : >
```

- closeall : closeall command is used to close all open file.

```
Customized VFS: > closeall
```

- stat: stat command give statistical information about file that is properties.

```
Customised VFS : > stat myfile.txt

---------Statistical Information about file-----------
File name : myfile.txt
Inode Number 1
File size : 1024
Actual File size : 0
Link count : 1
Reference count : 3
File Permission : Read & Write
 ------------------------------------------------------


Customised VFS : >

```

- truncate: truncate command is used to delete data from file

```
Customised VFS : > truncate myfile.txt
Data Succesfully Removed
```

- rm: rm command used to remove the file.

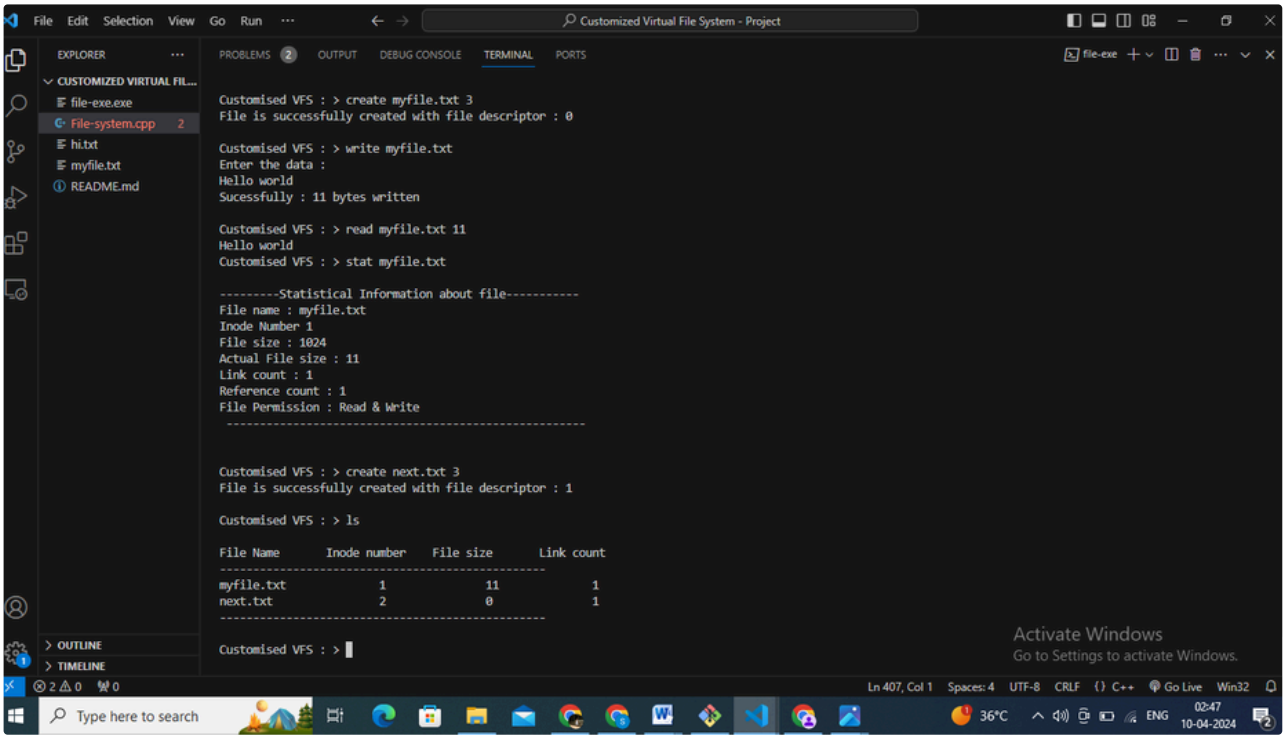
```
Customised VFS : > rm giraff.txt
File Successfully Deleted

```

- exit: exit command is used to terminate file system

```
Customized VFS : > exit
```

- Code snippet of terminal



- Thank you for using this documentation.