Customised Virtual File System (CVFS)

Technology: C Programming

Project Overview

This project is a **custom implementation of a Virtual File System (VFS)** that simulates the core functionality of the **Linux file system**. It is built entirely in **C**, with its own **custom shell** to interact with the virtual environment. The project provides a practical understanding of **system calls, file handling, memory management, and OS internals**.

Key Features

- **Custom Shell Interface**
  - Provides Linux-like commands for file operations (create, open, read, write, delete, ls, etc.).
- **System Call Simulation**
  - Implements core Linux file system system calls (open, read, write, lseek, close, rm, etc.) using C.
- **File System Data Structures**
  - Incore Inode Table
  - File Table
  - UAREA (User Area)
  - User File Descriptor Table
- **Platform Independent**
  - Allows system-level file handling functionalities of Linux to be used on **any operating system platform**.
- **Database-like Functionality**
  - Provides a **customised database management layer** with structured file handling.

Learning Outcomes

- Deep understanding of **Linux File System internals**.
- Practical knowledge of **data structures** used in OS (inode, file tables, UAREA).
- Strong grasp of **system programming in C**.
- Hands-on with **shell design & command interpreter**.
- Application of **low-level logic building** for OS-like environments.

GitHub Repository

🔗 Customised Virtual File System

Example Usage

```
$./Myexe
```

Marvellous CVFS> create Demo.txt

Marvellous CVFS> write Demo.txt "Jay Ganesh"

Marvellous CVFS> read Demo.txt

Jay Ganesh

Marvellous CVFS> ls

Demo.txt

Marvellous CVFS> rm Demo.txt

Marvellous CVFS> exit
