Customized Virtual File System

This project is used to emulate all functionalities provided by File systems.

Platform required - Windows NT platform OR Linux Distributions

Architectural requirement - Intel 32 bit processor

User Interface - Command User Interface

SDK used - None

Technology used - System Programming using C

About Virtual File System

- In this project we emulate all data structures which are used by the operating system to manage File system oriented tasks.
- As the name suggests it's virtual because we maintain all records in Primary storage.
- In this project we create all data structures which are required for File Subsystems as Inode Inode Table, File Table, UAREA, User File Descriptor Table, Super block, Disk Inode List Block, Data Block, Boot Block etc.
- We provide all implementations of necessary system calls and commands of the 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 to Algorithms of UNIX operating systems.
- By using this project we can get an overview of UFS (UNIX File System) on any platform.