

**Name of the project:** Customised Virtual File System

**Technology used:** System Programming using C, C++

**User interface:** Command User Interface

**Platform required:** We prefer Windows NT operating System platform OR Linux Distributions.

**Hardware requirements:** Minimum 512 MB of RAM Necessary.

**Description of the project:** This project is used create customised file System which can be work on Windows NT platforms we are going to implement UFS (UNIX File System) File system.

### **About Virtual File System**

- 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 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 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 UNNIX operating system. • By using this project we can get overview of UFS (UNIX File System) on any platform

### **Data structures used in the project:**

```
1. struct SuperBlock
{
    int TotalInodes;
    int FreeInode;
```

```
    }Obj_Super;
```

## 2. struct inode

```
{    char FileName[50];  
    int Inode_Number;  
    int File_Size;  
    int File_Type;  
    int Actual_File_Size;  
    int Link_Count;  
    int Reference_Count;  
    char *Data;  
    struct inode *pNext;  
};
```

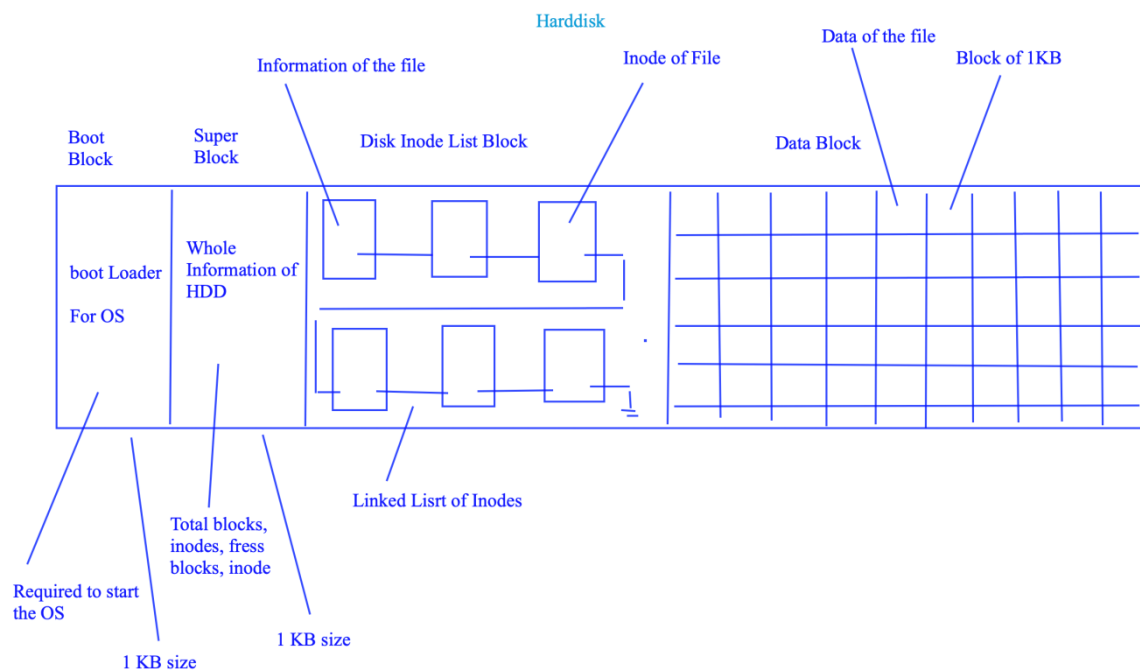
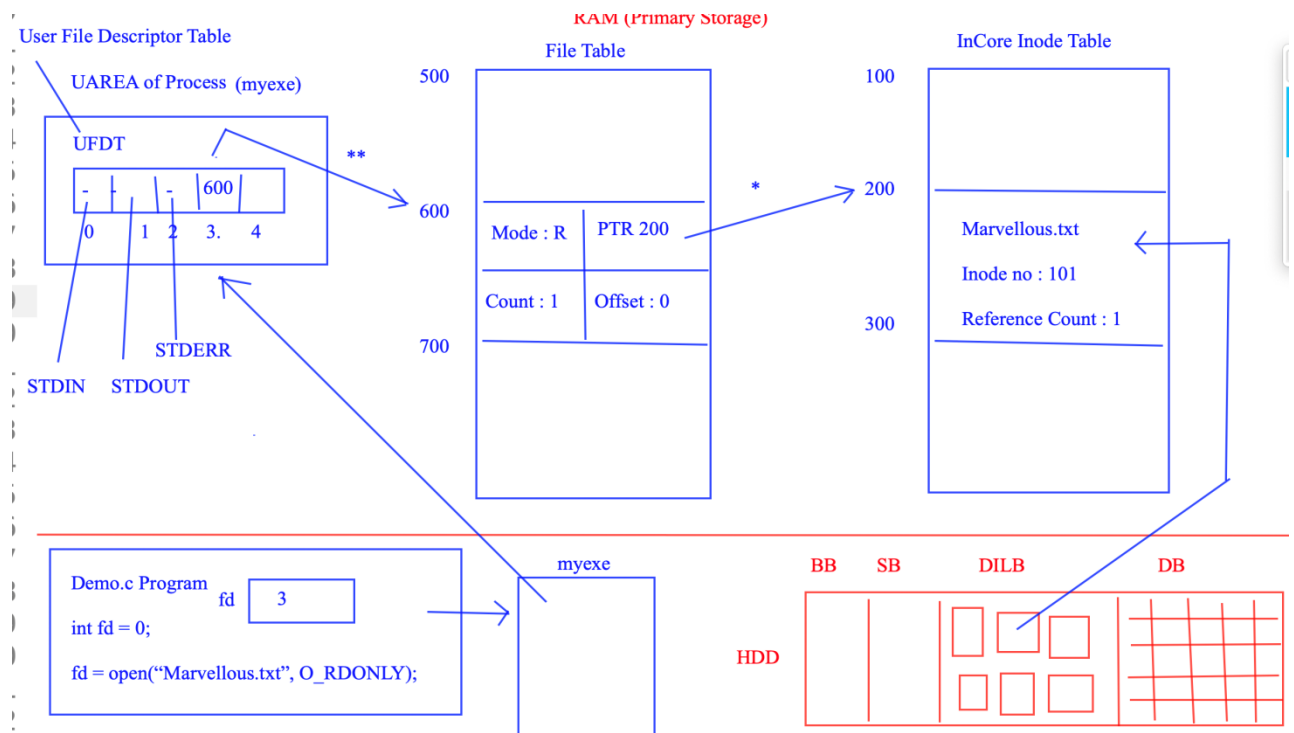
## 3. struct FileTable

```
{  
    int ReadOffSet;  
    int WriteOffset;  
    int Count;  
    PINODE ptr;  
    int Mode;  
};
```

## 4. struct UFDT

```
{  
    PFILETABLE ufdt[MAXFILES]; }UFDTOBJ;
```

## Diagram of data structures used in the project:



### **1. What is mean by file system?**

**Ans:** File System is way of Sorting and retrieved the Files into the Hard disk.

### **2. Which file systems are used by Linux and Windows operating systems?**

**Ans:** Windows Use NTFS (New Technology File System).

Linux Use FAT 32 /64 (File Allocation Table).

### **3. What are the parts of the file system?**

Ans : Name of That File ,Type of That File , Location ,Size of File ,Size on Disk, Created Date File ,Modified Date File ,Accessed Date of File ,Attributes Also like Read-Only.

### **1. Explain UAREA and its contents.**

**Ans :** Every Process has an Entry in the kernel process table, and each process is Allocated a UArea that Conations private data Manipulated only by the kernel.

UFDT (User File Descriptor Table) there is one Array that first three position are fixed.

0. STDIN (input for keyword)

1. STDOUT (output Display on Monitor)

2. STDERR (Error Display on Monitor)

This Three contain all process to use At Run time.

3. FD (File Descriptor) which files u open that files FD store thred position to So on.

### **2. Explain the use of the File Table and its contents.**

File table is information about that file.

**Contents are:-**

1. Read and write offset

2. Count

3. Pointer

4. Mode of that file.

### **3. Explain the use of InCore inode Table and its use.**

Incore inode is opened file all information there available and what is that inode number reference count and DILD information will be used .it used find to our data to use.

### **4. What is mean by inode?**

An inode is a record in a disk table, which contains information about a file or folder such as its size, owner, device node, socket, pipe, etc. except data content and file name. The number of inodes on your account equals the number of files and folders you have on it.

### **5. What are the contents of Superblock?**

Whole Information of HDD (Hard Disk) .

**Contents are:-**

1. Total blocks Available
- 2 .Total inodes Available
3. Free inodes .
4. Use inodes

### **5. What are the types of files?**

There are four types of files in the UNIX file system.

#### **1. Ordinary Files**

An ordinary file may contain text, a program, or other data. It can be either an ASCII file, with each of its bytes being in the numerical range 0 to 127, i.e. in the 7-bit range, or a binary file, whose bytes can be of all .possible values 0 to 255, in the 8-bit range.

#### **2. Directory Files**

Suppose that in the directory x I have a, b and c, and that b is a directory, containing files u and v. Then b can be viewed not only as a directory, containing further files, but also as a file itself. The file b consists of

information about the directory b; i.e. the file b has information stating that the directory b has files u and v, how large they are, when they were last modified.

### 3. Device Files

In UNIX, physical devices (printers, terminals etc.) are represented as “files.” This seems odd at first, but it really makes sense: This way, the same read() and write() functions used to read and write real files can also be used to read from and write to these devices.

### 4. Link Files

Suppose we have a file X, and type  
ln X Y

If we then run **ls**, it will appear that a new file, Y, has been created, as a copy of X, as if we had typed

cp X Y

However, the difference is the **cp** does create a new file, while **ln** merely gives an alternate name to an old file. If we make Y using **ln**, then Y is merely a new name for the same physical file X.

### 6. What are the contents of the inode?

**Contents are:-**

1. InodeNumber
2. Size of File
3. Actual Size
4. TimeDateStamp
5. User ID
6. Link-Count
7. Block Number .

### 7. What is the use of a directory file?

A directory is defined as an organizational unit, or container, used to organize folders and files into a hierarchical structure. A directory contains bookkeeping information about files that are, figuratively speaking, beneath them in the hierarchy.

## **8. How the operating system maintains security for files?**

Security refers to providing a protection system to computer system resources such as CPU, memory, disk, software programs and most importantly data/information stored in the computer system. So a computer system must be protected against unauthorized access, malicious access to system memory, viruses, worms etc.

## **9. What happens when a user wants to open the file?**

It depends on the operating system what exactly happens when you open a file each file (and Directory) contains a file name entry and the inode number for lookup when you open a file assuming you have the relevant permissions a file descriptor created using the unique inode number associated with file name.

## **10. What happens when a user calls lseek system call?**

## **11. What is the difference between library function and system call?**

A Library Function is linked to the user program and executes in user space while a system call is not linked to user program and executes in kernel space library function can be debugged easily a debugger while system call cannot be debugged as they are executed by the kernel side.

## **12.What is the use of this project?**

its used to understand the data structure and also system call internal work.

## **13.Is there any improvement needed in this project?**

Yes this file system only Customised we will try physically proper .

## **Internal working of below system calls**

### **1. Open**

Open system call used to already create file to opened this proper mode of file.

### **3. Close**

Close System call used to close the open file .

### **3. Read**

Read System call is used to open file start reading the first bytes off set.

### **4. Write**

Write system call Is used to write the Data into file.

### **5. Lseek**

Lseek System call is used to change the offset of file.

### **6. Stat**

Stat System call Is used to display the information of opened file

### **7. Chmod**

Chmod System call is used to change the mode particular file .

### **8. Unlink**

Unlink System call is used to delete the particular file.