

Customised Virtual File System

Project documentation

1. Name of project :-

- + Customised Virtual File System is the name of project where we build file system arrangement by UNIX and implement its data structure.

2. Technology used :-

- + Technology use in the project is -
C programming for system programming.

3. User Interface used :-

- + Command User Interface

4. Platform required :-

- + windows NT platform OR linux Distribution.

5. Hardware requirement :-

- + Intel 32 bit processor.

6. Description of project :-

- + In this project we emulate all data structures used by file management system oriented file of OS.
- + As it is virtual it maintain all records on RAM.
- + In this project we create data structure required by file subsystem as Inode table, File table, UARGA, user file descriptor table, super block, Disk inode list block, Data block, Boot Block

Customised Virtual File System

Project documentation

1. Name of project :-

+ Customised virtual file system is the name of project where we build file system management by UNIX and implement its data structure.

2. Technology used :-

+ Technology use in the project is -
C programming for system programming.

3. User Interface used :-

+ Command User Interface.

4. Platform required :-

+ windows NT platform OR linux Distribution.

5. Hardware requirement :-

+ Intel 32 bit processor.

6. Description of project :-

+ In this project we emulate all data structures used by file management system oriented type of OS.
+ As it is virtual it maintain all records on RAM.
+ In this project we create data structure required by file subsystem as Inode Inode table, File table, UARGA, User file descriptor table, super block, Disk inode list block, Data block, Boot Block

7. Data structure used in project -

+ ① UFDTAOR:-

- At first we create UFDTAOR which is array which stores address i.e. array of pointers, which points to filetable.

② FileTable:-

- It is structure which stores value about its inode and contain address of its inode table.
- It contains readoffset, writeoffset, count, mode, ptrinode which stores address of it.

③ Inode Inode Table:-

- It contains all detail information about file from which we can fetch actual data.
- This structure contains name of file, size of file, Actual file size, permissions, Inode numbers, file type, Buffer pointer, Link count, Reference count and address of next inode in pointer.

④ File system partitions:-

- As partitions on physical Hard disk are as follows.

① Boot block:- BB

- code requires to start operating system is stored in this block which is of size 1 kb
- Our system does not need to start any OS as its stored on primary storage device i.e. RAM therefore OS is there to run system.

① Super Block :- (SB)

- In OS it contains all information about hard disk i.e. metadata of hard disk.
- In my application it also contain info about number of free inodes and used inodes.

② Disk Inode List block :- (DILB)

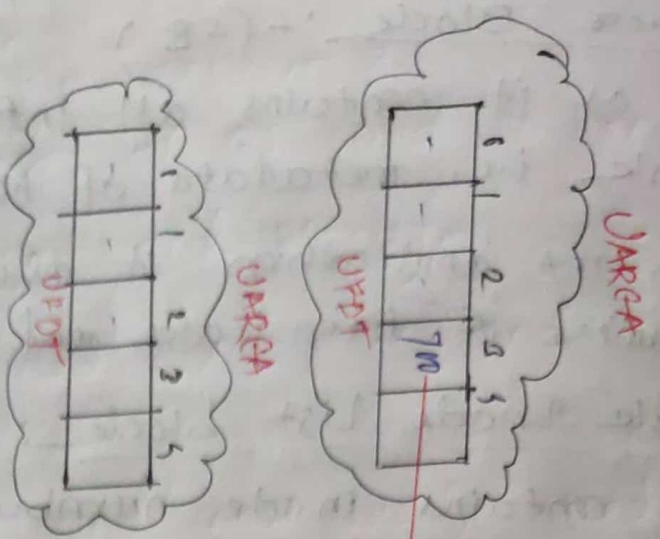
- It contains inode numbers in linked list of inodes

④ Data Block :- (DB)

- This part contain actual data of file stored and each one of size is 1kb.

8. Diagram of data structure and in project.



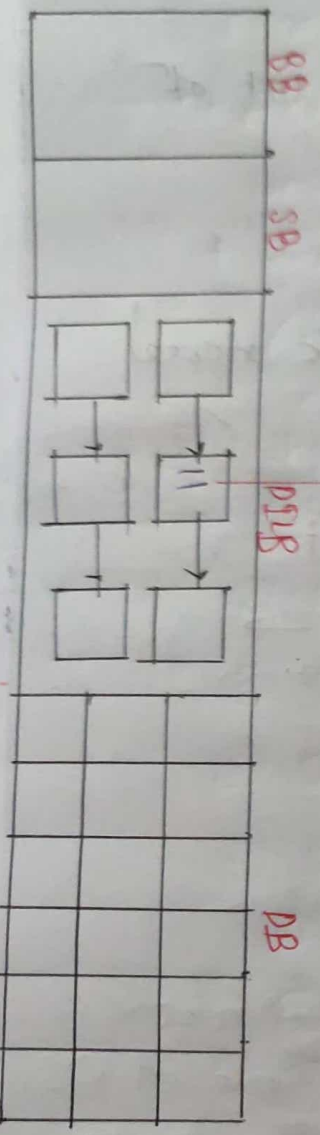


File Table

600			
700	mode: R	ptr: 100	
800	count: 1	offset: 0	

Inode Table

100	inode in RAM
200	inode no: 11
300	Block no: 121
400	permissions: 0777
500	owner: root
600	file size: 1024



g. Flow of the project.

- + At first we create space to take command as array of fixed row size 80
- + Then while creating any function & call it we have to create superblock which stores metadata of our project and another is to create DIB which contain InvertHash
- + after that we take string and break it into several command line array, which also known as tokenisation.
- + Then as per command line count we call the function with its several parameters.
- + For all this we implement above shown diagram data structure of file system in linux.
- + At first when we create any file it get file descriptor in array of VFDT in user space which stored address of file table in it
- + in file table there is data about our file like mode, offset and pointer pointing toward inode.
- + Inode Inode table contain all information written into the file.
- + This is how we can create data structure of Unix file system and customized as we want.

Q① What is mean by file system?

→ It is built in layer of OS used to handle the data mangement and storage.

Q② Which file system used by linux and window operating system?

→ + EXT4 is preferred and most widely used linux file system (fourth extended file system)

+ Window used default file system NTFS.

(New Technology File system)

Q③ What are parts of file system?

→ File system divided as —

① User Area

② File Table

③ Inode Table

④ In file system it is divided as —

i) Boot Block

ii) Super Block

iii) Disk Inode List Block

iv) Data Block.

Q4) Explain UAREA and its content.

→ + UAREA in file table used to store specific information so it known as user area

+ In our project it contain one array known as UFDTAB which is array of pointers which is User File Descriptor Table Array

+ This array contain address of its respective file table which stored data about file which is created.

Q5) Explain use of File Table and its content.

→ + It is used to stored the value about its inode and contain address of its inode table.

+ It contain read offset, write offset, count, mode, ptr to inode

+ File table contains to store all the information regarding file.

Q6) Explain the use of Inode Table and its use.

→ + It contains all detail information about file from which we can fetch actual data.

+ It contain name of file, size of file, actual file size, permission, Inode number, filetype, Buffer pointer, Link count, Reference count and address of next inode in pointer.

Q7) What is mean by inode ?

→ + Inode is index node.

+ It serves as a unique identifier for specific piece of metadata on a given filesystem.

+ Each inode describe about file

+ inode operates with Inode Table to store data in data block.

Q8) What are content of superblock ?

→ + superblock contain all metadata of hard disk

+ In our project superblock contains Total Inodes and Free Inodes

Q9) What are types of files ?

→

File Item

Description.

- regular

stores data

- directory

contain information used to access other files.

- special

Defines a FIFO pipe file or a physical device.

Q ⑪ What is the use of directory file?

→ + It contains information used to access other files

+ It is used to store, organize and separate files and directories on a computer.

Q ⑫ How the OS maintains security for files?

→ OS maintains security for files -

① classification of user as -

+ Owner → user who created

+ Group → group of members

+ Universe → all other users.

② classification for file as -

+ Read - Reading for file

+ Write - Writing or rewriting the file.

+ Append - editing at the end.

+ Delete - Deleting the file.

Q ⑬ What happens when a user wants to open a file?

→ + When we open the file we request the operating system to load the desired file from the secondary storage to RAM.

+ While open we pass parameters as filename, and its mode. After checking with permissions and mode we have to update read offset and write offset. to update in it.