Project Documentation

1.Name of the project : Customized Virtual File System

2.Technology used : C, C++

3.User interface used : Command Line Interface

4.platform required : GNU compiler, Windows, Mac, Linux

5.Hardware requirements : Minimum 2GB RAM

6.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 as Inode, Inode Table, File Table, UAREA, User File Descriptor Table, Superblock, 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 UNIX operating system. By using this project, we can get overview of UFS (UNIX File System) on any platform.

7.Data structure used in the project : Singly Linear Linked Lists

Some frequently Asked Questions

1. What is meant by file system?

file system or filesystem (often abbreviated to fs) is a method and data structure that the operating system uses to control how data is stored and retrieved

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

Windows uses FAT and NTFS as file systems, while Linux uses a variety (Ext4, Ext3) of file systems. Unlike Windows, Linux is bootable from a network drive.

3. What are the parts of the file system?

Space management, Filenames, Directories, Metadata, File system as an abstract user interface, Utilities, Restricting and permitting access, maintaining integrity, User data, Using a file system, Multiple file systems within a single system, Design limitations

4. Explain UAREA and its contents.

UAREA is “user area” inside the RAM. this user area provided by the operating system to the user. inside the UAREA file descriptor is an index number. using array data structure with index of file descriptors memory address is stored inside UFDT. UFDT stands “user file descriptor table”.

5. Explain the use of the File Table and its contents.

File table used to store metadata about actual file. File table contains readoffset, writeoffset, count, mode, address of next pointing inode etc.

6 Explain the use of Incore inode Table and its use.

Incore inode table is used to store information about the inode. inode contains file name, inode number, file size, actual file size, file type, buffer, link count, reference count, permission, next to be point address etc.

7 What is meant by inode?

The inode (index node) is a data structure in a Unix-style file system that describes a file-system object such as a file or a directory. Each inode stores the attributes and disk block locations of the object's data. A directory is a list of inodes with their assigned names.

8.What are the contents of Superblock?

Super block has whole information about hard disk. Total blocks, Total inodes, Free Blocks, Free inodes, size of file system.

9 What are the types of files?

Ordinary Files, Directories, Special Files, Pipes, Sockets, Symbolic Links.

10 What are the contents of the inode?

Inode contains file name, inode number, file size, actual file size, file type, buffer, link count, reference count, permission, next to be point address etc.

11.What is the use of a directory file?

A directory is used to store, organize, and separate files and directories on a computer. For example, you could have a directory to store pictures and another directory to store all your documents. By storing specific types of files in a folder, you could quickly get to the type of file you wanted to view.

12.How the operating system maintains security for files?

File system security within UNIX and Unix-like systems is based on 9 permission bits, set user and group ID bits, and the sticky bit, for a total of 12 bits. These permissions apply almost equally to all filesystem objects such as files, directories and devices.

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

When you open a file, assuming you have the relevant permissions, a file descriptor is created using the unique inode number associated with file name. As many processes/applications can point to the same file, inode has a link field that maintains the total count of links to the file.

14 What happens when a user calls lseek system call?

lseek() system call repositions the read/write file offset i.e., it changes the positions of the read/write pointer within the file. In every file any read or write operations happen at the position pointed to by the pointer. lseek() system call helps us to manage the position of this pointer within a file

15.What is the difference between library function and system

1. System Call:

There are two modes in the computer system one is user mode and another is kernel mode. In computer system there are different types of processes that are running on a computer system. When a user runs an application, it is said to be in user mode or computer is in user mode. When there is a requirement of hardware resource, the process sends a request to the kernel to get the process access and then computer enters in kernel mode. These requests are sent by using system call. The computer switches between these two modes frequently. Whenever the task is completed, the computer goes back to the user mode from kernel mode. This mode of transition is called context switching.

2. Library Call:

A Library call is a kind of request to use a specific piece of function that is defined in a programming library. A library file contains code data file, object file that are packed into one file for use. To make a library call, library should be imported first. The library call may be depended on the system call to complete the task.

16 What is the use of this project?

This project is research oriented and system level design project, this is not business purpose. This file system is derived from Unix file system. my project works like Unix file system. To reason behind to make this project to inhance my skill and looks how system file works.

17. What are the difficulties that you faced in this project? is there any important elements needed in this project?

At the time of designing of this project thinking is difficult about how unix/linux file system is implemented. how unix/linux file system is internally works. and how I make this project virtual. This project works in RAM only.

18. Is there any improvement needed in this project

Depends on developer thinking, developer can implement few additional system level functionalities in this project.

internal working of below system calls

open

Actually open() system call only creates a file descriptor which then may be used by either mmap() or read(). Both memory mapped I/O and standard I/O internally access files on disk through page cache, a buffer in which files are cached in order to reduce number of I/O operations.

2. close

A close system call is a system call used to close a file descriptor by the kernel. For most file systems, a program terminates access to a file in a filesystem using the close system calls. Some operating systems will invoke the close syscall as part of an operating system recovery as a result of a system failure.

3. read

read command in Linux system is used to read from a file descriptor. Basically, this command read up the total number of bytes from the specified file descriptor into the buffer. If the number or count is zero then this command may detect the errors. But on success, it returns the number of bytes read

4. Write

It writes data from a buffer declared by the user to a given device, such as a file. The data to be written, for instance a piece of text, is defined by a pointer and a size, given in number of bytes.

5. lseek

lseek () system call repositions the read/write file offset i.e., it changes the positions of the read/write pointer within the file. In every file any read or write operations happen at the position pointed to by the pointer. lseek () system call helps us to manage the position of this pointer within a file

6 stat

Stat system call is to check the status of a file such as to check when the file was accessed. The stat () system call actually returns file attributes. The file attributes of an inode are basically returned by Stat () function. ... Whenever you create a file an inode number for that file is created

7 chmod

chmod changes the file permission bits which is stored in the filesystem. The file permission bits are stored as per filesystem (ext, hfs, etc) but passed on to kernel to ensure it can enforce the file ACL accordingly. As for your question, where is it exactly stored, this depends on filesystem. Different filesystem stores it slightly differently but the interfacing is pretty standardized for kernel to access it.

8. unlink

In Unix-like operating systems, unlink is a system call and a command line utility to delete files. The program directly interfaces the system call, which removes the file name and (but not on GNU systems) directories like rm and rmdir.

Explanation of below commands

1. ls

ls is a command to list computer files in Unix and Unix-like operating systems. ls is specified by POSIX and the Single UNIX Specification. When invoked without any arguments, ls lists the files in the current working directory.

2. ls - l

By default, the ls command will print the name of all the files and directories only. To get the additional information and a cleaner view, use the flag "-l".

3. ls - a

ls -a will list all files including hidden files (files with names beginning with a dot).

4.rm

Use the rm command to remove files you no longer need. The rm command removes the entries for a specified file, group of files, or certain select files from a list within a directory.

5.cat

Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files.

6.cd

The cd command, also known as chdir (change directory), is a command-line shell command used to change the current working directory in various operating systems. It can be used in shell scripts and batch files

7.chmod

chmod is the command and system call used to change the access permissions of file system objects (files and directories) sometimes known as modes. It is also used to change special mode flags such as setuid and setgid flags and a 'sticky' bit.

8.cp

cp stands for copy. This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name

9.df

the df command to display information about total space and available space on a file system. If a file or directory is specified, then the df command displays information for the file system on which it resides. Normally, the df command uses free counts contained in the superblock

10.find

The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions.

11.grep

grep is a command-line utility for searching plain-text data sets for lines that match a regular expression. Its name comes from the ed command g/re/p (globally search for a regular expression and print matching lines), which has the same effect.

12.ln

The ln command is a standard Unix command utility used to create a hard link or a symbolic link (symlink) to an existing file or directory. The use of a hard link allows multiple filenames to be associated with the same file since a hard link point to the inode of a given file, the data of which is stored on disk

13.mkdir

mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories

14.pwd

The pwd command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.

15.touch

The touch command is a standard command used in UNIX/Linux operating system which is used to create, change and modify timestamps of a file. ... touch command: It is used to create a file without any content. The file created using touch command is empty.

16.uname

The uname command reports basic information about a computer's software and hardware. When used without any options, uname reports the name, but not the version number, of the kernel (i.e., the core of the operating system)

17.stat

The stat is a command which gives information about the file and filesystem. Stat command gives information such as the size of the file, access permissions and the user ID and group ID, birth time access time of the file. Stat command has another feature, by which it can also provide the file system information.

18.man

man command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

19.mkfs

The mkfs command stands for “make file system” is utilized to make a file system (which is, a system for organizing a hierarchy of directories, subdirectories, and files) on a formatted storage device usually, a partition on a hard disk drive (HDD) or it can also be a USB drive, etc