**Kamala Education Society’s**

**Pratibha Institute of Business Management**

**Chinchwad, Pune-411019**

**2021-2022**

**PROJECT REPORT**

**ON**

**File System Development**

**BY**

**Jay Padmakar Bhavsar**

**Course: MCA II**

**Roll No.:21041**

**Guide Name: Mr. Shubham Nagure**

**Index**

**CHAPTER 1: INTRODUCTION**

**Introduction**

**Existing System and Need for System**

**Limitations of existing system**

**CHAPTER 2: PROPOSED SYSTEM**

**2.1 Problem statement/s**

**2.2 Objectives of proposed system**

**2.3 Scope of the system**

**2.4 Module specifications**

**2.5 Detail Description of Technology Used**

**2.6 Operating environment**

**CHAPTER 3: SYSTEM ANALYSIS & DESIGN**

**3.1 On Disk File System Organization**

**3.2 Overview of the File System**

**3.3 Flow Chart**

**Conclusion**

**Bibliography**

**ANNEXURE 1 SAMPLE PROGRAM CODE / Project Demo (which will**

**prove sufficient development is done by the student)**

**1.Introduction**

Open-source local file systems, such as Linux Ext4 ,XFS, and Btrfs ,remain a critical component in the world of modern storage. For example, many recent distributed file systems, such as Google GFS and Hadoop DFS, all replicate data objects (and associated metadata) across local file systems. On smart phones, most user data is managed by a local file system; for example, Google Android phones use Ext4 and Apple’s iOS devices use HFSX. Finally, many desktop users still do not backup their data regularly in this case, the local file system clearly plays a critical role as sole manager of user data.

We introduce a simple file system implementation, known as vsfs (the Very Simple File System). This file system is a simplified version of a typical UNIX file system and thus serves to introduce some of the basic on-disk structures, access methods, and various policies that

you will find in many file systems today.

**1.2 Existing System and Need for System**

There are few drawbacks like Power Management, not every hardware will be compatible - older hardware may not have the drivers and some little used hardware may not have also, or have fewer resources compared to what it have in other OS,not every software and webAPPS are compatible - the software part is almost non-existant nowadays (except for games, Linux still have only a few games), but webAPPS (the ones you use in a browser) may not work, specially those that use much security like the ones used by banks.

All of these file systems have different data structures and do some things better or worse than their peers. Thus, the way we will be learning about file systems is through case studies: first, a simple file system (vsfs) in this chapter to introduce most concepts, and then a series of studies of real file systems to understand how they can differ in practice.

**CHAPTER 2: PROPOSED SYSTEM**

**2.1Problem statement/s**

To understand the existing File System which are stable in market to solve the issues regrading file/data accessing accuracy, Data Redundancy , Data Security and user can easily friendly with existing Os with help our File System.

**2.2Objectives of proposed system**

Our objective is to:

* to make FS which can manage memory efficiently.
* Job schedule -Round Robin Scheduling.
* Files Handling-Reading file, Writing Files and Sharing files.

Memory management technique-Fragments, Swapping.

Our aim is to be to create a simple fs so that new developers can easily onboard to the project and get understanding of FS easily.

In order to creatively think the new ideas to build it differently with more options to explore

**2.3Scope of the system**

As per study we develop and File System code that do basic functional operations with proper algorithm for each operation

We are going to read the File with help of suitable Algorithm to access the file data

Work on Security of the file how we can secure the Files in File System

Work on the Kernel Development.

**2.4Module specifications**

To build an File System we used Progamming Language- C

Debugger- GCC

1) fs.h -

* It is an Header File to declare the Structure of the File System.
* It Consist of Structure of SuperBlock, Inode, Disk Block
* Function declaration to access the File create,allocate,read,write Files
* Function declaration to create, mount, synchronize File System.

2) fs.c-

* It consist of the actual implementation or body of the Function which are declare in header file fs.h
* Where we code the logic to for File system Structure and File Structure and its access methods.

**2.5Detail Description of Technology Used**

We used multiple technologies based on the requirement.

**Vscode**: It is an IDE where we code file system.

**Docker**: We implement an containerization concept to containerized the development environment so that multiple developer can work together on File System project. We need ***docker*** so developer doesn’t need to install any extension or no need to setup environment. They simply need to pull the **Container Image**.

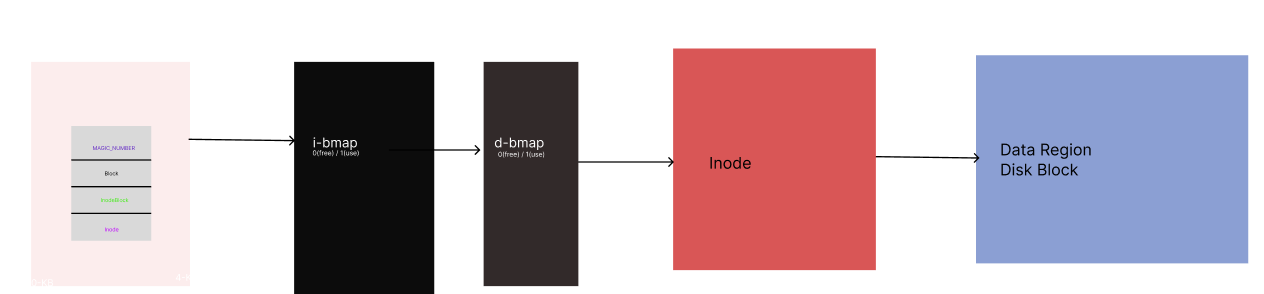
**Git and Github:**It is used to push the code to the virtual directory or we can say repository for **Version Control** and Multiple developer can contribute to our project.

**2.6Operating environment**

We used an Clean OS like linux or its different types to mount File System to test the its all Functionality.

**3.System Analysis & Design**

**1.On disk File System Organization**

****

**Diagram

Description automatically generated with low confidence**

1. **Overview of File System Code**

**Graphical user interface, application, Word

Description automatically generated**

**3.Flow Chart**

**Chart

Description automatically generated**

**Conclusion**

A file system is an essential component of any computer or computing system, responsible for managing files and directories and providing a structured way to store and access data. A file system provides an abstraction layer that hides the details of storage devices and makes it easy for users and applications to access and manage data.

File systems have evolved over time to support various use cases and requirements, from local file systems that manage files on a single computer or device, to distributed file systems that manage files across multiple devices and networks. The scope of a file system depends on the specific requirements of the system and the needs of the users, and may include features such as performance, reliability, scalability, security, and usability.

Designing and implementing a file system requires careful consideration of both functional and non-functional requirements, as well as an understanding of the underlying storage devices and operating system. With the rapid growth of data and computing needs, file systems continue to play a crucial role in managing data and providing efficient and reliable access to information.

**Bibliography**

**ANNEXURES 1:**

**SAMPLE PROGRAM CODE / Project Demo (which will**

**prove sufficient development is done by the student)**

**Github link:** https://github.com/jaybhavsar555/vsfs