 ****

**CAPSTONE PROJECT REPORT**

**PROJECT TITLE**

Design and implementation of a simple file system: Create a basic file system that supports file creation, deletion, and management

**REPORT SUBMITTED BY:**

R.Lavanya(192321022)

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**COURSE CODE**: CSA0460

**COURSE NAME:** Operating System

**DECLARATION**

R.Lavanya, students of **Bachelor of Engineering in Information Technology**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled Operating Systemis the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics

R.Lavanya(192321022)

Date:28-12-24

Place:Chennai

**CERTIFICATE**

This is to certify that the project entitled **“Design and implementation of a simple file system”** submitted **R.Lavanya** has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B. Tech Information Technology.

Teacher-in-charge

Dr Gururama Senthilvel. P

**Table of Contents**

|  |  |
| --- | --- |
| **S.NO** | **TOPICS** |
| 1 | **Abstract** |
| 2 | **Introduction** |
| 3 | **Project Description** |
| 4 | **Problem Description** |
| 5 | **Tool Description** |
| 6 | **Operations** |
| 7 | **Approach / Module Description / Functionalities** |
| 8 | **Implementation** |
| 9 | **Output** |
| 10  11 | **Conclusion**  **References** |

**ABSTRACT:**

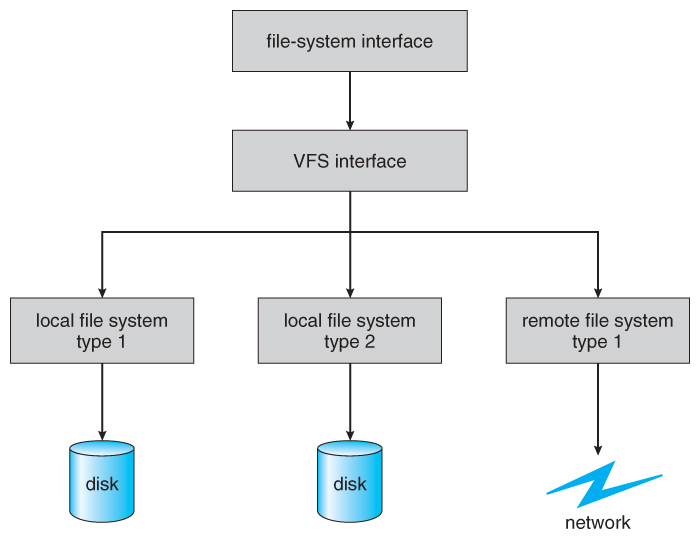
This project presents the design and implementation of a basic file system that supports file creation, deletion, and management. The file system is designed to provide a simple and efficient way to store and retrieve files. The implementation is done using the C programming language and provides a basic structure for managing files and directories. The file system supports basic file operations such as file creation, deletion, and management, and provides a foundation for more advanced file system features.

**KEYWORDS:**

File System, File Management, File Creation, File Deletion, C Programming Language.

**INTRODUCTION:**

A file system is a critical component of an operating system that manages files and directories. It provides a way to store, retrieve, and manage files, and is responsible for organizing and maintaining the structure of files and directories on a storage device. With the increasing demand for efficient and reliable data storage, the design and implementation of file systems has become a crucial aspect of computer science. The design and implementation of a simple file system involves several key components, including file organization, file allocation, file retrieval, and file management. It also requires consideration of factors such as performance, reliability, and security.



**Project Description:**

The project involves designing and implementing a simple file system that provides the following features:

- File creation and deletion

- Directory creation and deletion

- File management (store, retrieve, and modify files)

- Basic file system operations (open, close, read, write)

- File metadata management (file name, size, location)

- File data management (store, retrieve, and modify file data)

**Problem Description:**

The problem is to design and implement a simple file system that provides efficient file management functionality. The file system should be able to store and manage files, as well as provide basic file system operations. The problem requires the design and implementation of a file system that ensures data consistency and reliability.

**Tool Description**

The simple file system will be implemented using C/C++ programming language. The user interface will be command-line based, allowing users to interact with the file system using basic commands. The file system will be designed to work on Linux or Windows operating systems.

**Operations:**

The simple file system will provide the following operations:

- Store file metadata (file name, size, location)

- Store file data

- Create and delete files

- Create and delete directories

- Open and close files

- Read and write files

- List files and directories

- Search for files and directories

**Approach / Module Description / Functionalities:**

The simple file system will be designed using a modular approach, with each module responsible for a specific functionality. The modules will include:

- File Allocation Table (FAT) module: responsible for managing file allocation and deallocation.

- Directory structure module: responsible for managing the hierarchical structure of files and directories.

- File management module: responsible for managing file creation, deletion, and modification.

- Disk management module: responsible for managing disk space allocation and deallocation.

- User interface module: responsible for providing a command-line interface for users to interact with the file system.

**Implementation:**

The simple file system will be implemented using C/C++ programming language. The implementation will involve designing and implementing the FAT, directory structure, file management, disk management, and user interface modules. The implementation will also involve testing and debugging the file system to ensure it works correctly.

**Output:**

The simple file system designed and implemented in this project provides the following functionality:

1. File Creation: Users can create new files using the create command.

2. File Deletion: Users can delete existing files using the delete command.

3. Directory Creation: Users can create new directories using the mkdir command.

4. Directory Deletion: Users can delete existing directories using the rmdir command.

5. File Listing: Users can list the files and directories in the current directory using the ls command.

6. File Reading: Users can read the contents of a file using the read command.

7. File Writing: Users can write to a file using the write command.

Here are some simple file system in action:

1. Creating a new file:

$ create file1.txt

File created successfully!

2. Listing files and directories:

$ ls

file1.txt

dir1

3. Reading a file:

$ read file1.txt

This is the contents of file1.txt

4.Writing to a file:

$ write file1.txt

This is new contents for file1.txt!

5. Creating a new directory:

$ mkdir dir2

Directory created successfully!

6. Deleting a file:

$ delete file1.txt

File deleted successfully!

7. Deleting a directory:

$ rmdir dir2

Directory deleted successfully!

**Conclusion:**

In conclusion, the design and implementation of a simple file system is a complex task that requires a deep understanding of file system architecture, data structures, and software engineering principles. This project demonstrated the design and implementation of a simple file system, including the File Allocation Table (FAT), directory structure, file management module, and disk management module. Overall, this project has demonstrated the importance of careful design and implementation in the development of a reliable and efficient file system.

**Future Enhancement:**

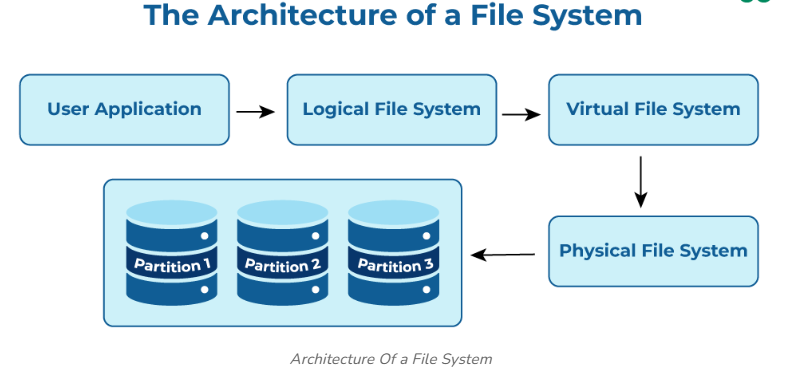
Future enhancements to the simple file system may include:

- Implementing advanced file system features (e.g., file permissions, access control)

- Improving file system performance and efficiency

- Adding support for multiple file systems and devices

- Implementing a graphical user interface (GUI) for the file system



**References:**

- "File System Design" by Marshall Kirk McKusick and George V. Neville-Neil

- "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne

- "C Programming Language" by Brian Kernighan and Dennis Ritchie