Custom File System for Linux

1 PROBLEM STATEMENT

Design and implement a file system for Linux operating systems that provides efficient storage, organization, and management of files and directories on disk, ensuring data integrity and reliability.

2 NEED

File system implementation in an operating system provides several advantages, including efficient data storage, data security, data recovery, improved performance, scalability, flexibility, and cross-platform compatibility.

- File System Structure
- File Allocation
- Data Retrieval
- Security and Permissions

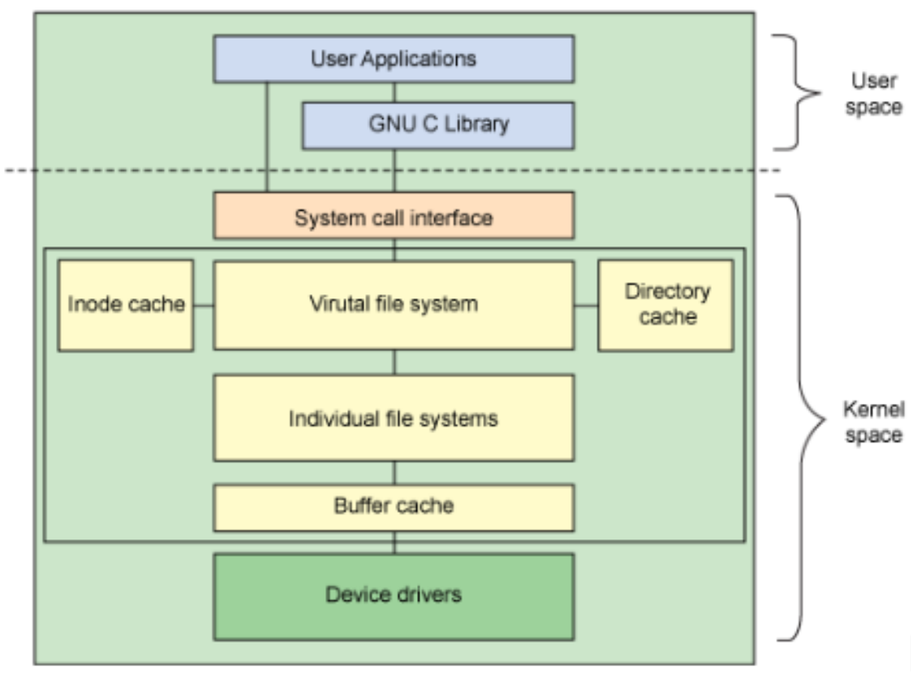
3 METHODOLOGY

1. Basic understanding of fundamental file system concepts
2. Environment Setup and configurations for linux
3. File System Functionality Implementation- designing core structures like inodes and superblock
4. Block Allocation Strategy
5. Testing and Validation using make file and mount

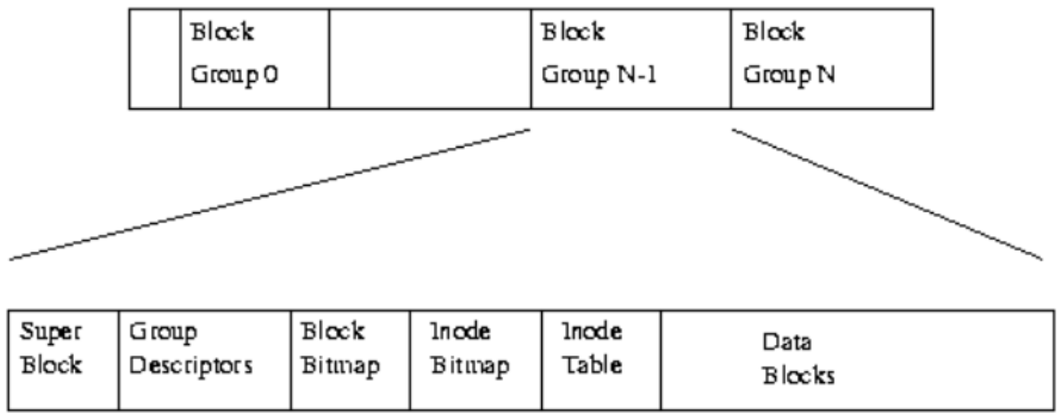
4 SYSTEM CALLS AND APIS

- Dynamic memory allocation functions- malloc() and free()
- Data structures **bitmaps** are used to represent file system structures such as inodes
- System calls like memcpy() and memset()
- **APIs** provide a layer of abstraction that allows developers to interact with the operating system and other software components in a more standardized and consistent manner.

5 DESIGN



6 IMPLEMENTATION



S SAHANA (1RV22CS166)
PRAJWAL M (1RV22CS140)