

1 项目概要

本项目构造了一个单进程类 Unix 二级文件系统，使用一个普通的大文件（mcwDisk.img，称之为一级文件）来模拟 Unix V6++ 的一张磁盘，磁盘中存储的信息以 512 字节的数据块为单位。

该文件系统的设计思路参考 Unix V6++，主要实现了以下功能：

- 实现对该逻辑磁盘的基本读写操作，并在该逻辑磁盘上定义了二级文件系统结构。包括存储资源管理信息块 Superblock、外存索引结点 DiskInode 区和文件数据区。
- 实现了对文件存储资源的有效管理。DiskInode 结构通过“多级混合索引方式”实现了文件索引结构，Superblock 通过“栈方式”管理一定数量的空闲数据块和 DiskInode 结点，并采用“成组链接法”管理多余的数据块。
- 实现了“树形带交叉勾连”的目录结构，目录检索算法和目录的增、删、改的设计与实现。
- 实现了内存的文件打开结构，提高文件操作效率。其中，利用内存 Inode 结构和内存索引结点表管理近期使用的文件；利用打开文件控制块 File 和系统打开文件表实现文件的动态管理；为进程设置进程打开文件描述符表 OpenFiles，并完成对前述数据结构的勾连。
- 实现了高速缓存结构。由于只考虑单用户单进程单设备，故缓存队列做了合理的简化，只保留了一个自由队列，但功能上保证了：按照 LRU 方式管理缓存、缓存尽可能重用、缓存延迟写回，利用高速缓存块提高了文件读取的效率。

为用户提供基本文件操作接口，具体说明见需求分析。通过文件操作接口调用相应的系统调用函数，进而调用文件系统的各功能函数和结构实现相应功能，包括但不限于：文件的创建、打开、读写、关闭和删除，目录的创建、结构展示和跳转、文件读写指针的移动等。

本实验报告的行文结构安排如下：

- 需求分析：介绍本次课程设计的具体需求，包括程序功能需求、程序交互需求，以及用户其余需求和程序可行性说明。
- 概要分析：说明文件系统的整体设计思路，包括子模块的划分、数据结构的设计、主程序流程和模块间调用关系等。
- 详细设计：进一步说明各子模块的具体设计思路，包括对重点变量和函数的说明，对重点功能的流程和函数调用说明。
- 调试分析及正确性验证：展示运行结果，测试各条指令，并说明运行中遇到的主要问题和解决方案。

- 用户使用说明：用户使用方法的介绍，展示各指令的具体使用方法和可能产生的错误等。

2 需求分析

2.1 程序功能需求

使用一个普通的大文件来模拟 Unix V6++的一张磁盘，磁盘中存储的信息以 512 字节的数据块为单位。至少应当包含以下结构和功能的设计与实现：

1) 磁盘文件结构

- 在一级文件系统上定义二级文件系统结构
- 定义 SuperBlock 结构，完成 Inode 节点的分配与回收算法设计与实现、文件数据区的分配与回收算法设计与实现。
- 定义磁盘 Inode 节点结构，实现 Inode 区的组织，定义多级索引结构，以及索引结构的生成和检索

2) 文件目录结构

- 定义目录文件的结构
- 目录结构的检索算法的设计与实现
- 目录结构增、删、改的设计与实现

3) 文件打开结构

- 文件打开结构的设计：内存 Inode 节点，File 结构，进程打开文件表等
- 内存 Inode 节点的分配与回收
- 文件打开、关闭过程

4) 磁盘高速缓存

- 模拟缓存，实现与硬盘数据的交互
- 实现延迟写入的功能
- 程序退出时将缓存内容更新到磁盘，防止数据丢失

5) 文件操作接口

本次实验需要为用户提供的文件操作接口包括：

fformat、ls、mkdir、fcreat、fopen、fclose、fread、fwrite、fseek、fdelete...

表 2.1 展示了各指令的具体设计和功能说明，为方便用户使用，学生加入了 help、autotest、shmtree、exit 等四条指令。