

# 文件系统的构建思路

## 总体架构

文件系统由三个主要组件组成：

1. **磁盘服务器 (Disk Server)**：模拟物理磁盘，提供读写操作接口。
2. **文件系统服务器 (File System Server)**：管理文件系统的结构和元数据，提供文件操作接口。
3. **文件系统客户端 (File System Client)**：作为用户界面，发送文件操作命令给文件系统服务器。

## 磁盘服务器

### 功能

- 模拟磁盘存储，管理柱面和扇区。
- 提供基本的读写操作，包含：
  - `I`: 获取磁盘几何信息（柱面数和每柱面扇区数）。
  - `R C S`: 读取指定柱面和扇区的数据。
  - `W C S l data`: 写入指定柱面和扇区的数据。

### 实现步骤

1. **初始化**：通过命令行参数接收磁盘文件名、柱面数、每柱面扇区数、寻道时间和端口号。
2. **数据存储**：使用 `mmap` 或文件 API 进行实际数据存储。
3. **命令处理**：通过套接字接受客户端命令并进行处理，读取或写入数据。

## 文件系统服务器

### 功能

- 管理文件和目录结构，提供文件操作接口。
- 支持基本文件操作，包括：
  - 格式化文件系统。
  - 创建、删除文件和目录。
  - 读写文件内容。
  - 修改文件内容（插入、删除数据）。
  - 列出目录内容。

### 实现步骤

1. **初始化**：通过命令行参数接收磁盘服务器地址和端口号。
2. **数据结构**：
  - **iNode**: 管理文件元数据，包括文件大小、直接块和间接块指针。
  - **目录结构**: 管理文件和目录的层次结构。
  - **空闲块管理**: 使用位图或链表管理磁盘的空闲块。
1. **文件操作**：

- **格式化**：初始化文件系统的元数据结构。
  - **文件创建**：分配新的 iNode 和目录项。
  - **文件读写**：根据文件偏移和长度，读写指定的文件内容。
  - **文件修改**：支持文件内容的插入和删除操作。
1. **命令处理**：通过套接字接受客户端命令并进行处理，调用相应的文件操作函数。

## 文件系统客户端

### 功能

- 提供用户界面，发送文件操作命令给文件系统服务器。
- 解析用户输入的命令，格式化后发送给文件系统服务器。
- 显示服务器返回的结果。

### 实现步骤

1. **初始化**：通过命令行参数接收文件系统服务器地址和端口号。
2. **命令处理**：
  - 循环读取用户输入的命令。
  - 根据命令类型，格式化为文件系统服务器接受的格式。
  - 发送命令到文件系统服务器并接收响应。
  - 显示响应结果给用户。

## 文件系统实现中的关键点

1. **二级 iNode 结构**：
  - **直接块**：存储文件的直接数据块指针。
  - **一级间接块**：存储指向数据块指针的块。
2. **空闲块管理**：
  - **位图**：使用位图管理空闲块，每一位表示一个块是否空闲。
  - **链表**：使用链表管理空闲块，链表中的每个节点表示一个空闲块。
3. **命令协议**：
  - 文件系统服务器和磁盘服务器之间的命令协议需要精确定义，以确保两者间的通信顺畅。
  - 文件系统客户端和文件系统服务器之间的命令协议需要确保用户命令能够正确解析和处理。
4. **多用户支持**：
  - 设计用户数据结构，存储用户信息并保存在文件系统中。
  - 实现用户登录、用户创建和删除功能。
  - 实现文件访问控制，支持文件共享和权限管理。

### 详细实现步骤

1. **实现磁盘服务器**：
  - 编写磁盘服务器代码，处理基本的读写命令。
  - 测试磁盘服务器的基本功能，确保数据存储和读取正确。

## 2. 实现文件系统服务器：

- 编写文件系统服务器的初始化和数据结构管理代码。
- 实现基本的文件操作命令，并测试其功能。
- 添加高级文件操作（如插入和删除数据）并测试。

## 3. 实现文件系统客户端：

- 编写客户端代码，处理用户输入的命令。
- 测试客户端与文件系统服务器的交互，确保命令能够正确解析和执行。

## 4. 支持多用户：

- 1. 用户数据结构设计
  - 定义一个用户结构体，包含用户ID、用户名、密码等信息。
  - 可以使用哈希表或链表等数据结构来管理用户信息。
- 2. 用户登录和认证
  - 实现用户登录功能，用户需提供用户名和密码进行认证。
  - 认证成功后，分配一个会话ID给用户，用于标识用户会话。
- 3. 多用户文件管理
  - 每个用户拥有自己的家目录（Home Directory），用于存储个人文件和目录。
  - 在文件系统中为每个用户创建一个目录，以用户名作为目录名，存放用户的个人文件。
  - 文件系统应提供创建用户目录、删除用户目录等操作。
- 4. 文件权限控制
  - 设计文件权限控制机制，限制用户对文件的访问权限。
  - 文件拥有者可以设置文件的读、写、执行权限，其他用户根据权限进行访问。
- 5. 用户间文件共享
  - 实现用户间文件共享功能，允许用户分享文件给其他用户。
  - 文件拥有者可以设置共享权限，其他用户根据权限访问共享文件。
- 6. 安全性考虑
  - 对用户输入进行验证，防止恶意用户进行非法操作。
  - 加密用户密码存储，确保用户密码安全。
- 7. 会话管理
  - 管理用户会话，确保会话的安全性和有效性。
  - 实现会话过期机制，自动注销长时间未活动的会话。
- 8. 日志记录
  - 记录用户操作日志，用于追踪用户操作和审计。
  - 记录登录、登出、文件操作等关键事件，并保存到日志文件中。
- 9. 测试和调试
  - 编写测试用例，验证多用户功能的正确性和稳定性。
  - 进行调试，排查可能存在的问题和异常情况。

通过以上步骤，可以构建一个基本的文件系统，包括磁盘模拟、文件和目录管理、用户交互和多用户支持。