

Lab06:文件系统模拟

提交要求及时间

请参考作业要求。

1.1 实验简介

本实验要求在假设的 I/O 系统之上开发一个简单的文件系统,这样做既能让实验者对文件系统有整体了解,又避免了涉及过多细节。用户通过 `create`, `open`, `read` 等命令与文件系统交互。文件系统把磁盘视为顺序编号的逻辑块序列,逻辑块的编号为 0 至 $L-1$ 。I/O 系统利用内存中的数组模拟磁盘。

1.2 I/O 系统

实际物理磁盘的结构是多维的:有柱面、磁道、扇区等概念。I/O 系统的任务是隐藏磁盘的结构细节,把磁盘以逻辑块的形式呈现给文件系统。逻辑块顺序编号,编号取值范围为 0 至 $L-1$,其中 L 表示磁盘的存储块总数。实验中,可用类似于字符数组 `ldisk[L][B]` 构建磁盘模型,其中 B 表示每个存储块的长度,一般为 512 字节。I/O 系统从文件系统接收命令,根据命令指定的逻辑块号把磁盘块的内容读入命令指定的内存区域,或者把命令指定的内存区域内容写入磁盘块。文件系统和 I/O 系统之间的接口由如下两个函数定义:

- `int read_block(int i, char *p);`

该函数把逻辑块 i 的内容读入到指针 p 指向的内存位置,拷贝的字符个数为存储块的长度 B ,返回实际读取的字节数。

- `int write_block(int i, char *p);`

该函数把指针 p 指向的内容写入逻辑块 i ,拷贝的字符个数为存储块的长度 B ,返回实际写入的字节数。

此外,为方便测试,我们还需要实现另外两个函数:一个用来把数组 `ldisk` 存储到文件;另一个用来把文件内容恢复到数组。

1.3 文件系统

文件系统位于 I/O 系统之上。

1.3.1 用户与文件系统之间的接口

文件系统需提供如下函数：create, destroy, open, read, write。

- create(filename): 根据给定的文件名创建新文件。
- destroy(filename): 删除指定文件。
- open(filename): 打开文件。该函数返回的索引号可用于后续的 read, write, lseek 或 close 操作。
- close(index): 关闭指定的文件。
- read(index, mem_area, count): 从指定文件顺序读入 count 个字节 mem_area 指定的内存位置。读操作从文件的读写指针指示的位置开始。
- write(index, mem_area, count): 把 memarea 指定的内存位置开始的 count 个字节顺序写入指定文件。写操作从文件的读写指针指示的位置开始。
- lseek(index, pos): 把文件的读写指针移动到 pos 指定的位置。pos 是一个整数，表示从文件开始位置的偏移量。文件打开时，读写指针自动设置为 0。每次读写操作之后，它指向最后被访问的字节的下一个位置。lseek 能够在不进行读写操作的情况下改变读写指针位置。
- directory: 列表显示所有文件及其长度。

1.3.2 文件系统的组织

磁盘的前 k 个块是保留区，其中包含如下信息：位示图 and 文件描述符。位示图用来描述磁盘块的分配情况。位示图中每一位对应一个逻辑块。创建或者删除文件，以及文件的长度发生变化时，文件系统都需要进行位示图操作。前 k 个块的剩余部分包含一组文件描述符（思考：可以有多少个文件？）。每个文件描述符包含如下信息：

- 文件长度，单位字节
- 文件分配到的磁盘块号数组。该数组的长度是一个系统参数。在实验中我们可以把它设置为一个比较小的数，例如 3（请思考，这个数对文件的大小意味着什么）。

1.3.3 目录

我们的文件系统中仅设置一个目录，该目录包含文件系统中所有的文件。除了不需要显示地创建和删除之外，目录在很多方面和普通文件相像。目录对应 0 号文件描述符。初始状态下，目录中没有文件，所有，目录对应的描述符中记录的长度应为 0，而且也没有分配磁盘块。每创建一个文件，目录文件的长度便增加一分。目录文件的内容由一系列的目录项组成，其中每个目录项由如下内容组成：

- 文件名
- 文件描述符序号

1.3.4 文件的创建与删除

创建文件时需要进行如下操作：

- 找一个空闲文件描述符（扫描 ldisk[0]~ldisk[k-1]）
- 在文件目录里为新创建的文件分配一个目录项（可能需要为目录文件分配新的磁盘块）
- 在分配到的目录项里记录文件名及描述符编号

- 返回状态信息（如有无错误发生等）

删除文件时需要进行如下操作（假设文件没有被打开）

- 在目录里搜索该文件的描述符编号
- 删除该文件对应的目录项并更新位示图
- 释放文件描述符
- 返回状态信息

1.3.5 文件的打开与关闭

文件系统维护一张打开文件表，打开文件表的长度固定，其表目包含如下信息：

- 读写缓冲区
- 读写指针
- 文件描述符号

文件被打开时，便在打开文件表中为其分配一个表目；文件被关闭时，其对应的表目被释放。读写缓冲区的大小等于一个磁盘存储块。打开文件时需要进行的操作如下：

- 搜索目录找到文件对应的描述符编号
- 在打开文件表中分配一个表目
- 在分配到的表目中把读写指针置为 0，并记录描述符编号
- 读入文件的第一块到读写缓冲区中
- 返回分配到的表目

在打开文件表中的索引号关闭文件时需要进行的操作如下：

- 把缓冲区的内容写入磁盘
- 释放该文件在打开文件表中对应的表目
- 返回状态信息

1.3.6 读写

文件打开之后才能进行读写操作。读操作需要完成的任务如下：

1. 计算读写指针对应的位置在读写缓冲区中的偏移
2. 把缓冲区中的内容拷贝到指定的内存位置，直到发生下列事件之一：
 - 到达文件尾或者已经拷贝了指定的字节数。这时，更新读写指针并返回相应信息
 - 到达缓冲区末尾。这时，把缓冲区内容写入磁盘，然后把文件下一块的内容读入磁盘。最后返回第 2 步。

其他操作请同学们自己考虑。

1.4 测试

为了能够对我们的模拟系统进行测试，请编写一个操纵文件系统的外壳程序或者一个菜单驱动系统。

1.5 参考文献

1. 文件系统-维基百科
2. Linux 文件系统剖析
3. FAT32 File System Specification