杭州电子科技大学

——操作系统课程设计

课程设计题目:文件系统

姓名： 卢文亮

学号： 14108219

班级： 14108412

学院： 计算机学院

专业： 软件工程

负责老师：周旭

# 简单文件系统的实现

一、实现内容

(1) 在内存中开辟一个虚拟磁盘空间作为文件存储分区，在其上实现一个简单的基于多级目录的单用户单任务系统中的文件系统。在退出该文件系统的使用时，应将该虚拟文件系统以一个Windows 文件的方式保存到磁盘上，以便下次可以再将它恢复到内存的虚拟磁盘空间中。

(2) 文件存储空间的分配可采用显式链接分配或其他的办法。

(3) 空闲磁盘空间的管理可选择位示图或其他的办法。如果采用位示图来管理文件存储空间，并采用显式链接分配方式，那么可以将位示图合并到FAT中。

(4) 文件目录结构采用多级目录结构。为了简单起见，可以不使用索引结点，其中的每个目录项应包含文件名、物理地址、长度等信息，还可以通过目录项实现对文件的读和写的保护。

(5) 要求提供以下操作命令：

my\_format：对文件存储器进行格式化，即按照文件系统的结构对虚拟磁盘空间进行布局，并在其上创建根目录以及用于管理文件存储空间等的数据结构。

my\_mkdir：用于创建子目录。

my\_rmdir：用于删除子目录。

my\_ls：用于显示目录中的内容。

my\_cd：用于更改当前目录。

my\_create：用于创建文件。

my\_open：用于打开文件。

my\_close：用于关闭文件。

my\_write：用于写文件。

my\_read：用于读文件。

my\_rm：用于删除文件。

my\_exitsys：用于退出文件系统。

二、设计思路（主要算法描述、程序流程图等）：

1．系统主函数main()

2. 进入文件系统函数startsys()

① 申请虚拟磁盘空间；

② 使用c语言的库函数fopen()打开myfsys文件：若文件存在，则转③；若文件不存在，则创建之，转⑤

③ 使用c语言的库函数fread()读入myfsys文件内容到用户空间中的一个缓冲区中，并判断其开始的8个字节内容是否为“10101010”（文件系统魔数），如果是，则转④；否则转⑤；

④ 将上述缓冲区中的内容复制到内存中的虚拟磁盘空间中；转⑦

⑤ 在屏幕上显示“myfsys文件系统不存在，现在开始创建文件系统”信息，并调用my\_format()对①中申请到的虚拟磁盘空间进行格式化操作。转⑥；

⑥ 将虚拟磁盘中的内容保存到myfsys文件中；转⑦

⑦ 使用c语言的库函数fclose()关闭myfsys文件；

⑧ 初始化用户打开文件表，将表项0分配给根目录文件使用，并填写根目录文件的相关信息，由于根目录没有上级目录，所以表项中的dirno和diroff分别置为5（根目录所在起始块号）和0；并将ptrcurdir指针指向该用户打开文件表项。

⑨ 将当前目录设置为根目录。

3．磁盘格式化函数my\_format()

4．更改当前目录函数my\_cd()

5．创建子目录函数my\_mkdir()

6．删除子目录函数rmdir()

7．显示目录函数my\_ls()

8．创建文件函数my\_create()

① 为新文件分配一个空闲打开文件表项，如果没有空闲表项则返回-1，并显示错误信息； ② 若新文件的父目录文件还没有打开，则调用my\_open()打开；若打开失败，则释放①中为新建文件分配的空闲文件打开表项，返回-1，并显示错误信息；

③ 调用do\_read()读出该父目录文件内容到内存，检查该目录下新文件是否重名，若重名则释放①中分配的打开文件表项，并调用my\_close()关闭②中打开的目录文件；然后返回-1，并显示错误信息；

④ 检查FAT是否有空闲的盘块，如有则为新文件分配一个盘块，否则释放①中分配的打开文件表项，并调用my\_close()关闭②中打开的目录文件；返回-1，并显示错误信息；

⑤ 在父目录中为新文件寻找一个空闲的目录项或为其追加一个新的目录项;需修改该目录文件的长度信息，并将该目录文件的用户打开文件表项中的fcbstate置为1；

⑥ 准备好新文件的FCB的内容，文件的属性为数据文件，长度为0，以覆盖写方式调用do\_write()将其填写到⑤中分配到的空目录项中；

⑦ 为新文件填写①中分配到的空闲打开文件表项，fcbstate字段值为0，读写指针值为0；

⑧ 调用my\_close()关闭②中打开的父目录文件；

⑨ 将新文件的打开文件表项序号作为其文件描述符返回。

9．删除文件函数my\_rm()

10．打开文件函数my\_open()

11．关闭文件函数my\_close()

12．写文件函数my\_write()

① 检查fd的有效性（fd不能超出用户打开文件表所在数组的最大下标），如果无效则返回-1，并显示出错信息；

② 提示并等待用户输入写方式：（1：截断写；2：覆盖写；3：追加写）

③ 如果用户要求的写方式是截断写，则释放文件除第一块外的其他磁盘空间内容（查找并修改FAT表），将内存用户打开文件表项中文件长度修改为0，将读写指针置为0并转④；如果用户要求的写方式是追加写，则修改文件的当前读写指针位置到文件的末尾，并转④；如果写方式是覆盖写，则直接转④；

④ 提示用户：整个输入内容通过CTR+Z键（或其他设定的键）结束；用户可分多次输入写入内容，每次用回车结束；

⑤ 等待用户从键盘输入文件内容，并将用户的本次输入内容保存到一临时变量text[]中，要求每次输入以回车结束，全部结束用CTR+Z键（或其他设定的键）；

⑥ 调用do\_write()函数将通过键盘键入的内容写到文件中。

⑦ 如果do\_write()函数的返回值为非负值，则将实际写入字节数增加do\_write()函数返回值，否则显示出错信息，并转⑨；

⑧ 如果text[]中最后一个字符不是结束字符 CTR+Z， 则转⑦继续进行写操作；否则转⑨； ⑨ 如果当前读写指针位置大于用户打开文件表项中的文件长度，则修改打开文件表项中的文件长度信息，并将fcbstate置1；

⑩ 返回实际写入的字节数。

13．实际写文件函数do\_write()

14．读文件函数my\_read()

15．实际读文件函数do\_read()

16. 退出文件系统函数my\_exitsys()

四、总结：

这次实验的系统是文件系统，对于文件系统系统的格式要求感觉挺复杂的，刚开始写的时候没有一点头绪，对于FAT表和FCB以及输入输出流并不是很了解，概念经常混杂在一起，在经过读、写、改代码的过程中，逐渐了解掌握了这些概念，知道了它们对应的作用，并对于文件系统的格式有了大体的掌握。

程序有存在着许多的问题，写文件的时候只实现了覆盖写，并且在写满之后不会去寻找下一个块，只是截取块大小的字节。

经过了这次文件系统的实验，我对于操作系统有了进一步的认识，明白了平常所谓的文件夹就是一个索引，便于用户使用。最重要的是渐渐掌握一种编写程序时严谨的逻辑思维能力，考虑的要周全一些。