题目五　模拟磁盘文件系统实现

一、课程设计目的

了解磁盘文件系统的结构、功能和实现。并可练习合作完成系统的团队精神和提高程序设计能力。

二、小组人数

建议 3～5 人一组共同完成模拟磁盘文件系统的实现。

选择题目“模拟磁盘文件系统实现”的小组在最终提交时须公开演示及讲解。由于这个题目较复杂，难度和工作量远大于前面几个题目，故小组成员最后得分也酌情高于选择前面四个题目的同学的分数（高 5~10 分）。

三、编程语言

建议使用一些 Windows 环境下的程序设计语言如 VC、Java，以借助这些语言的多线程来模拟并行发生的行为。要求图形界面。

四、课程设计内容

设计一个简单的文件系统，用文件模拟磁盘，用数组模拟缓冲区，要求：

（1）　支持多级目录结构，支持文件的绝对读路径；

（2）　文件的逻辑结构采用流式结构，物理结构采用链接结构中的显式链接方式；

（3）　采用文件分配表 FAT；

（4）　实现的命令包括建立目录、列目录、删除空目录、建立文件、删除文件、显示文件内容、打开文件、读文件、写文件、关闭文件、改变文件属性。可以采用命令行界面执行这些命令，也可以采用“右击快捷菜单选择”方式执行命令。

（5）　最后编写主函数对所作工作进行测试。

五、课程设计具体内容和要求

为了正确地实现文件的存取，文件系统设计了一组与存取文件有关的功能模块，用户可以用“访管指令”调用这些功能模块，以实现文件的存取要求。我们把文件系统设计的这一组功能模块称为“文件操作”，实验就是要模拟实现一些文件操作。文件操作不是独立的，它和文件系统的其它部分密切相关，若要实现文件操作就离不开文件的目录结构、文件的组织结构和磁盘空间的管理。因此，这个实验虽然是文件操作的模拟实现，但是还必须模拟一部分文件的组织结构、目录结构和磁盘空间管理的实现。

（1）　文件的组织结构

文件的逻辑结构有两种形式：流式文件和记录式文件。实验中只支持流式文件，采用称为显式链接的物理文件结构，把磁盘中每一块的指针部分提出来，组织在一起，形成文件分配表（FAT）。文件分配表的作用不仅如此，其它的作用下面将提到。

磁盘有多少块，文件分配表就有多少项，若某文件的一个磁盘块号为 i，则这个文件的下一个磁盘块号应该记录在文件分配表的第 i 项。例如，某系统文件分配表的前几项值如下图所示。某个文件的起始盘块号为 3，则该文件的磁盘块号依次为 3、4、9、12、13。

项	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
内容	-1	-1	-1	4	9	0	7	8	-1	12	11	-1	13	-1	0	0	

（2） 磁盘空间的管理

首先要模拟一个磁盘。因为是实验，不使用真正的磁盘，所以实验中用一个文件模拟一个小磁盘。假设模拟磁盘有 128 个物理块，每个物理块大小为 64 字节。盘块的块号从 0 编起。

将前面所讲的文件分配表放在磁盘的开始处，因为盘块有 128 块，所以文件分配表有 128 项，每项占用一个字节。这样文件分配表占用了磁盘的 0 块和 1 块，这两块就不能作其它用处。若一个盘块是某个文件的最后一块，填写“-1”表示文件结束。

文件的建立和删除就需要对磁盘的空间进行分配和回收，所以要建立一定的数据表格来记录磁盘的使用情况。用文件分配表的第 i 项表示第 i 个盘块的使用情况。磁盘的第 0 块一定会被系统数据占用，所以任何一个文件的某个盘块块号都不可能是“0”，因而可以用“0”表示磁盘盘块空闲，若这个盘块已经分配出去，即是某个文件的一块，由上面我们知道文件分配表中对应项记录的是文件下一块的块号或结束标志都不是“0”。这样非“0”值表示盘块已分。像前图那张文件分配表中，块号为 5、14 和 15 的盘块是空闲的，其余是已分配的。在文件分配表中可以用一个超过盘块编号的正整数表示文件结束，在此实验中采用 255 代替 -1 表示文件结束。

如果磁盘中某些部分损坏，只要不是系统区（引导扇区、文件分配表或根目录等），不分配那些坏的盘块，磁盘可以继续使用。在文件分配表对应坏盘块的项不能是“0”，一般需要特定的数值表示（这个数值应该是盘块编号以外并且不是结束标志的数值，例如，实验中采用 128～254 之间的某个数值）。假设实验中模拟磁盘的第 23 块和第 49 块已经损坏，不能使用，则在文件分配表的第 23 项和第 49 项写入“254”表示该盘块损坏不能使用。

由于磁盘分配时，有时不能预定文件的大小，例如建立文件时并不知道文件的大小。因

而磁盘的分配有时是一块一块申请的。磁盘空间回收时，整个文件删除时回收很多，但有时文件修改时可能会删除某些内容，造成归还磁盘块，这时是一块一块回收的。注意：分配一个磁盘块时，不应该从文件分配表第一项查起，因为磁盘中最开始的几块为系统数据区，可假定系统区域占用了 x 个盘块，则应该从第 x 块（0～x-1 块为系统区域）开始查询。回收一个磁盘块很简单，比如回收磁盘块的块号为 x，只要找到文件分配表中第 x 项，将第 x 项的值改为 0 即可。

（3） 目录结构

文件目录是用于检索文件的，它是文件系统实现按名存取的主要手段。文件目录由若干目录项组成，每一个目录记录一个文件的有关信息。一般的说，目录项应该包括如下内容：

①有关文件的控制信息。例如，用户名、文件名。文件类型、文件属性。实验模拟个人计算机上的文件操作，这部分内容仅包括文件名、文件类型和文件属性；

②有关文件结构的信息。例如，文件的逻辑结构、文件的物理结构、记录个数、文件在存储介质的位置等。实验中，仅仅支持流式文件，不支持记录式文件，所以这部分内容仅仅包括文件在存储介质的位置（分给文件第一个盘块的块号，即起始盘块号）、文件的长度；

③有关文件管理的信息。例如，文件的建立日期、文件被修改的日期、文件保留期限和记帐信息等。实验中为了简单起见，这部分内容都不采用。

因此，实验中文件的目录项包括：文件名、文件类型、文件属性、文件的起始盘块号、文件的长度，每个目录项占用 8 个字节，具体结构如下所示：

文件名：3 个字节（实验中合法文件名仅可以使用字母、数字和除“$”、“.”、“/”以外的字符，第一个字节的值为“$”时表示该目录为空目录项，文件名和类型名之间用“.”分割，用“/”作为路径名中目录间分隔符）；

文件类型名：2 个字节；

文件属性：1 个字节；

起始盘块号：1 个字节；

文件长度：1 个字节（为了实验的简单，假设文件长度单位为盘块）

有了文件目录后，当用户要求使用某个文件时，文件系统可以顺序查找目录项，并比较文件名，就可以找到指定文件的目录项，根据目录行中有关内容核对使用权限、并读出文件供用户使用。因此文件目录的组织和管理要便于检索和防止冲突。

在操作系统中目录就有根目录和子目录两种。除了文件需要登记形成目录外，还要登记子目录的情况。实验中，根目录固定位置、固定大小（可以登记有限个文件或子目录项），

子目录像文件一样，可使用任何一个空闲磁盘块。为了实验简单，实验中根目录占用一个盘块，子目录的长度不采用可以任意长的方法，而是采用定长的方法，每个子目录的长度也是一个盘块，只能放 8 个目录项。文件和目录的登记项是混在一起的，登记项的结构应该和文件目录一样，每个目录项占用 8 个字节，结构如下：

目录名：3 个字节（实验中合法目录名仅可以使用字母、数字和除“$”、“.”、“/”以外的字符，第一个字节的值为“$”时表示该目录为空目录项）；

保留 2 字节未使用（在实验中填写空格）；

目录属性：1 个字节；

起始盘块号：1 个字节；

保留 1 字节未使用（在实验中填写“0”）。

在目录登记项中，系统为目录名后 2 个字节（对应文件类型名位置）填写空格，目录起始盘块号后 1 字节（对应文件长度位置）填写 0。目录属性和文件属性占用同一个字节，为了区别目录和文件，该字节每一位代表不同的含义（为“1”表示“是”，为“0”表示否），如下图所示，第 0 位表示文件为只读文件，第 1 位表示文件为系统文件，第 2 位表示文件为可读、可写的普通文件，第 3 位表示该登记项不是文件的登记项，而是目录的登记项，其余几位闲置未用。如该字节为 8（00001000），表示该目录是一个目录的登记项，该字节为 3（00000011），表示该目录是一个只读系统文件的登记项，该字节为 4（000000100），表示该目录是一个可读可写的普通文件。

第 7 位	第 6 位	第 5 位	第 4 位	第 3 位	第 2 位	第 1 位	第 0 位
未使用	未使用	未使用	未使用	目录属性	普通文件	系统文件	只读文件

目录检索的方法常用的是顺序检索，根据绝对路径名查找文件的方法一般如下：先找到根目录的起始盘块，一般根目录位置是固定的，实验中就是模拟磁盘的第 2 块，将该盘块读出；取出路径名中根目录后的目录名或文件名，和根目录中目录项依次比较，比较完一块，再根据文件分配表找到下一块，再读入比较，直到找到名字一致的目录项或根目录登记项均已查完为止；若没有找到，则查找失败，结束；若查找的是文件，结束；若查找的是目录，从查找到的目录项中，取出目录的起始盘块号，读入此盘块，然后用上述相同的查找方法继续查找，直到找到该文件（或目录）或查找失败结束。

查找文件除了绝对路径名外，还可以使用相对路径名。相对路径名是从当前目录出发到指定文件的路径。如果文件（或目录）在当前目录下，使用相对路径名查找速度比较快。和

绝对路径的查找方法一样，只是查找的起点是当前目录，不是根目录。实验中只使用绝对路径名。

（4） 文件操作

确定文件组织结构、目录结构和磁盘空间管理的方法后，就可以模拟文件操作的实现。实验中文件操作包括建立文件、打开文件、关闭文件、读文件、写文件、删除文件、显示文件内容和改变文件属性，目录命令包括建立目录、显示目录内容和删除空目录。在实验中没有程序调用这些命令，为了看到它们的模拟情况，从键盘输入选择指令来模拟用户程序的调用。

首先要建立一个“已打开文件表”，用来记录打开或建立文件的相关内容，结构如下图所示：

文件路径名	文件属性	起始盘块号	文件长度	操作类型	读指针		写指针	
					块号	块内地址	块号	块内地址

用数组模拟已打开文件表，数据结构定义如下：

```
#define   n   5          //实验中系统允许打开文件的最大数量
typedef struct
{ int dnum;              //磁盘盘块号
  int bnum;              //磁盘盘块内第几个字节
}pointer;                //已打开文件表中读、写指针的结构
typedef struct
{ char name[20];         //文件绝对路径名
  char attribute;;       //文件的属性，用 1 个字节表示，所以此用 char 类型
  int number;            //文件起始盘块号
  int length;            //文件长度，文件占用的字节数
  int flag;              //操作类型，用“0”表示以读操作方式打开文件，用“1”表示以写操作方式打开文件
  pointer read;      //读文件的位置，文件打开时 dnum 为文件起始盘块号，bnum 为“0”
  pointer write;     //写文件的位置，文件刚建立时 dnum 为文件起始盘块号，bnum 为“0 ，打开文件时 dnum 和 bnum 为文件的末尾位置
```

```
}OFTLE;        //已打开文件表项类型定义
struct
{ OFILE file[n];   //已打开文件登记表
   int length;        //已打开文件登记表中登记的文件数量
}openfile;            //已打开文件登记表定义
```

无论上述哪种文件操作都会涉及到已打开文件表，对于已打开文件表主要是查找、删除和插入操作。下面分析所有常用的文件操作。

①建立文件（create_file）

用户要把一个新文件放到存储介质上前，首先调用文件系统的“建立”操作。

建立文件的主要工作就是检查文件目录，确认无重名文件后，寻找空闲登记项进行登记；寻找空闲存储块（至少一块）以备存储文件信息或存放索引表，最后应该填写已打开文件表。

实验中需要的参数比较少，只要有文件名和文件属性就可以，create_file（文件名，文件属性）。

实验中，建立文件时给出文件名和文件属性，文件属性如果是只读性质则不能建立；文件建立时根据给定的文件路径名进行查找，如果父目录不存在，建立文件失败；如果存在，查看有无重名文件，如果有，则提示该文件已存在，建立文件失败；如无重名文件，则为该文件建立文件目录，并分配给它一个磁盘块，最后填写目录和已打开文件表。

②打开文件（open_file）

用户要求使用一个已经存在的文件时，首先执行“打开文件”操作。可以在读或写一个文件时，默认打开该文件，将“打开”和“读或写”合二为一，不必非得先“打开”再“读或写”。

实验中，所需参数有文件名、操作类型（读或写），open_file（文件名，操作类型）。

实验中，打开文件首先要检查该文件是否存在，不存在，打开失败；如果文件存在，还要检查打开方式，确保不能以写方式打开只读文件；最后填写已打开文件表，若文件已经打开则不需要填写已打开文件表。

③读文件（read_file）

用户要求读文件信息时调用文件系统的“读文件”操作。

实验中，读文件的参数只需要文件名和读取长度，read_file（文件名，读取长度）。因为采用的是流式文件结构，所以读的长度用字节表示。

实验中，读文件操作的主要工作是查找已打开文件表中是否存在该文件；如果不存在，

则打开后再读；然后检查是否是以读方式打开文件，如果是以写方式打开文件，则不允许读；最后从已打开文件表中读出读指针，从这个位置上读出所需要长度，若所需长度没有读完已经遇到文件结束符，就终止操作。实验中用“#”表示文件结束。

④写文件（write_file）

用户要求存取文件信息时调用文件系统的“写文件”操作。实验中，写文件的参数只需要文件名、存放准备写入磁盘信息的缓冲和写的长度，write_file（文件名，缓冲，写长度）。因为采用流式文件结构，所以写长度用字节表示。

实验中，写文件操作的主要工作是查找已打开文件表中是否存在该文件，如果不存在，则打开后再写；如果存在，还要检查是否以写方式打开文件；如果不是写方式打开文件，不能写；最后从已打开文件表中读出写指针，从这个位置上写入缓冲中的数据。

写文件有两种情况，一种情况是建立文件后的写入，这种写比较简单，一边写一边申请空间即可完成；另一种情况是文件打开后的写入，这个比较复杂，存在着文件中间修改的问题。实验中，第二种情况只要求完成从文件末尾向后追加的功能。

⑤关闭文件（close_file）

用户对文件读写完毕后需要调用文件系统的“关闭文件”操作。

实验中，关闭文件的参数只需要文件名，close_file（文件名）。

实验中关闭文件，首先要看该文件是否打开，如果没有打开，就不用关闭；如果已经打开，则检查打开方式，如果是写方式打开的，要追加文件结束符，修改目录项；最后从已打开文件表中删除对应项。

⑥删除文件（delete_file）

用户认为文件没有必要保存时需要调用文件系统的“删除文件”操作。实验中，删除文件时参数只要文件名，delete_file（文件名）。

实验中，删除文件操作的主要工作是检查文件是否存在；不存在，操作失败；如存在，查找该文件是否打开，如果打开不能删除；如果没有打开，则删除文件目录项并归还文件所占磁盘空间。

⑦显示文件内容（typefile）

显示文件内容首先要找到该文件的目录登记项，如果文件不存在，指令执行失败；如果存在，查看文件是否打开，打开则不能显示文件内容；若没有打开，从目录中取出文件的起始盘块号，一块一块显示文件内容。

⑧改变文件属性（change）

改变文件属性，首先查找该文件，如果不存在，结束；如果存在，检查文件是否打开，打开不能改变属性；没有打开，根据要求改变目录项中属性值。

实验中，首先要系统初始化，包括建立文件模拟磁盘、初始化磁盘、初始化根目录为空目录项；然后，可以选择一项功能执行。

目录的操作命令：

①建立目录（md）

建立目录首先要找到建立目录的位置（父目录），然后查找该目录是否存在，如果父目录不存在，不能建立；如果存在，查找是否存在同名目录，存在，不能建立；不存在，则查找一个空目录项，为该目录申请一个盘块，并填写目录内容。

②显示目录内容（dir）

显示目录内容首先要找到该目录，如果目录不存在，指令执行失败；如果存在，一项一项显示目录内容。

③删除空目录（rd）

删除空目录首先要找到该目录，如果目录不存在，指令执行失败；如果存在，但是根目录或非空目录，显示不能删除，操作失败；若是非空子目录，则删除其目录项并回收对应空间。删除空目录的过程和删除文件的过程相似。

另外注意，对磁盘文件进行读操作时，需要磁盘的一个盘块读入主存后才能进行处理，对磁盘文件进行写操作时，要写满缓冲后才写入磁盘。所以模拟文件操作时，不能将整个模拟磁盘的内容同时读入主存，应该当需要模拟磁盘的某个盘块内容时，从对应文件中读出；修改后需要写回模拟磁盘。实验中就是用这种方法模拟磁盘的输入输出。

实验中需要定义两个数组 buffer1 和 buffer2 模拟缓冲。

实验中首先系统初始化，包括建立文件模拟磁盘，初始化磁盘和根目录初始化为空目录项，然后选择各个命令进行测试。

有能力的同学可在上述基础上，将磁盘文件系统改进为支持多级树型目录，支持相对路径，子目录可以任意长的文件系统。