**课 程 设 计 报 告**

设计题目：模拟DOS的FAT文件系统

班 级：物联网2002班

组长学号：20206087

组长姓名：赵熙龙

指导教师：鲍玉斌

设计时间：2023年6月

设计分工

组长学号及姓名：赵熙龙 20206087

分工：

组员1学号及姓名：马文轩 20206016

分工：

组员2学号及姓名：刘振铭 20206053

分工：

组员3学号及姓名：蹇栎 20206066

分工：

**摘 要**

模拟DOS的FAT文件系统，结合位图来管理磁盘的空闲空间和空闲的inode节点。该系统实现了包括文件读写、打开、关闭、删除、创建等操作，目录的创建，跳转等功能，还支持多用户系统。

**关键词**：操作系统，文件系统

目录

**[设计题目：模拟DOS的FAT文件系统 1](#_Toc22400)**

**[关键词：操作系统，文件系统，…… 2](#_Toc32135)**

**[1 问题定义和分析 4](#_Toc11472)**

[1.1 设计任务 4](#_Toc6988)

[1.2 问题定义 4](#_Toc31105)

[1.3 项目约束 4](#_Toc17879)

[1.4 开发人员 4](#_Toc16416)

**[2 需求分析 4](#_Toc14039)**

[2.1 功能需求 4](#_Toc22446)

[2.2 非功能需求 4](#_Toc1320)

**[3 设计 5](#_Toc12268)**

[3.1 系统层次设计 5](#_Toc11382)

[3.2 系统流程设计 5](#_Toc15619)

[3.3 数据结构设计 6](#_Toc17103)

[3.4 系统基础属性设计 7](#_Toc27970)

[3.5 通用模块接口设计 8](#_Toc21460)

[3.6 功能模块接口设计 8](#_Toc25105)

**1 问题定义和分析**

确定要完成的任务以及对其进行可行性分析。

**1.1 设计任务**

模拟DOS的FAT文件系统。

**1.2 问题定义**

模拟DOS的FAT文件系统，其中使用位图管理空闲inode，用FAT管理簇的使用。该文件系统要实现文件的创建，删除，打开，关闭，读写，修改等操作，还有目录的跳转，创建等功能。该系统还应允许创建多个用户。

**1.3 项目约束**

开发时间：2023年5月20日至2023年6月5日

开发环境：Windows

**1.4 开发人员**

组长：赵熙龙

组员：马文轩，刘振铭，蹇栎

**2 需求分析**

确定要完成的功能及其他需求

**2.1 功能需求**

多用户 ：允许在一个文件系统当中有多个用户存在。

多级目录：允许创建多级目录。

具有login (用户登录)：用户提供用户名以及密码即可登录。

系统初始化（建文件卷、提供登录模块）：在文件系统创建时要初始化分区。

文件的创建： 创建文件。

文件的打开：打开文件。

文件的读：读取文件。

文件的写：写入文件。

文件关闭：关闭打开的文件。

删除文件：删除选定的文件。

创建目录：创建目录。

改变目录：修改目录信息。

列出文件目录：列出当前目录的文件信息。

退出：用户退出文件系统。

格式化：删除文件系统当中所有的数据。

**2.2 非功能需求**

**3 设计**

**3.1 系统层次设计**

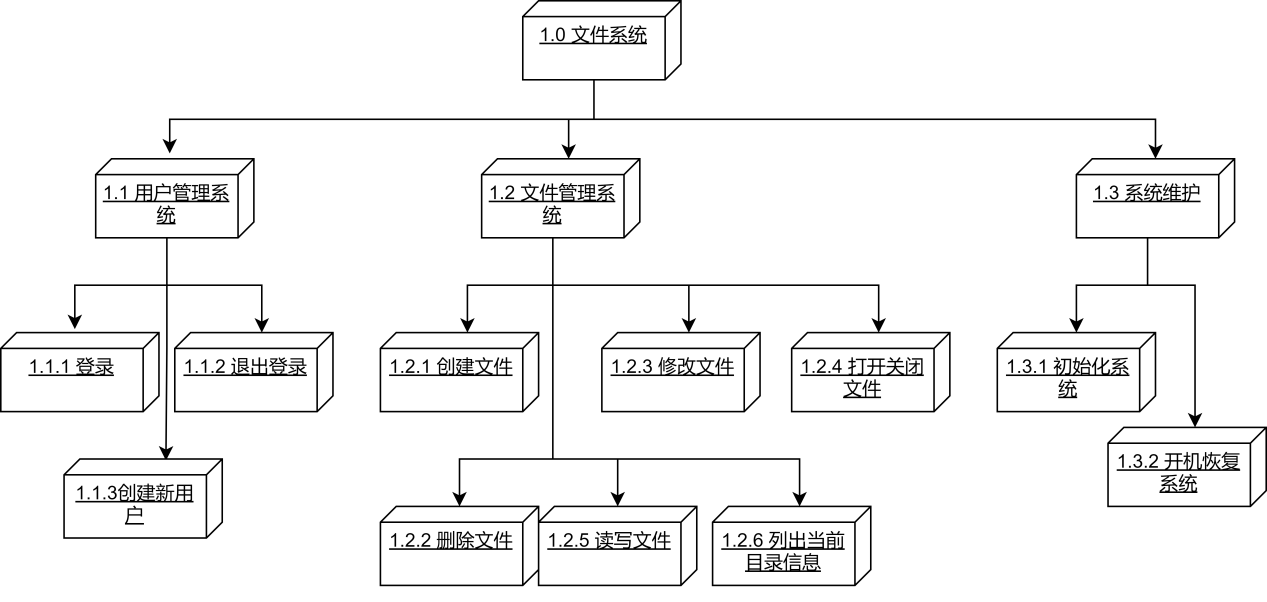
****

图 1 系统层次图

**3.2 系统流程设计**

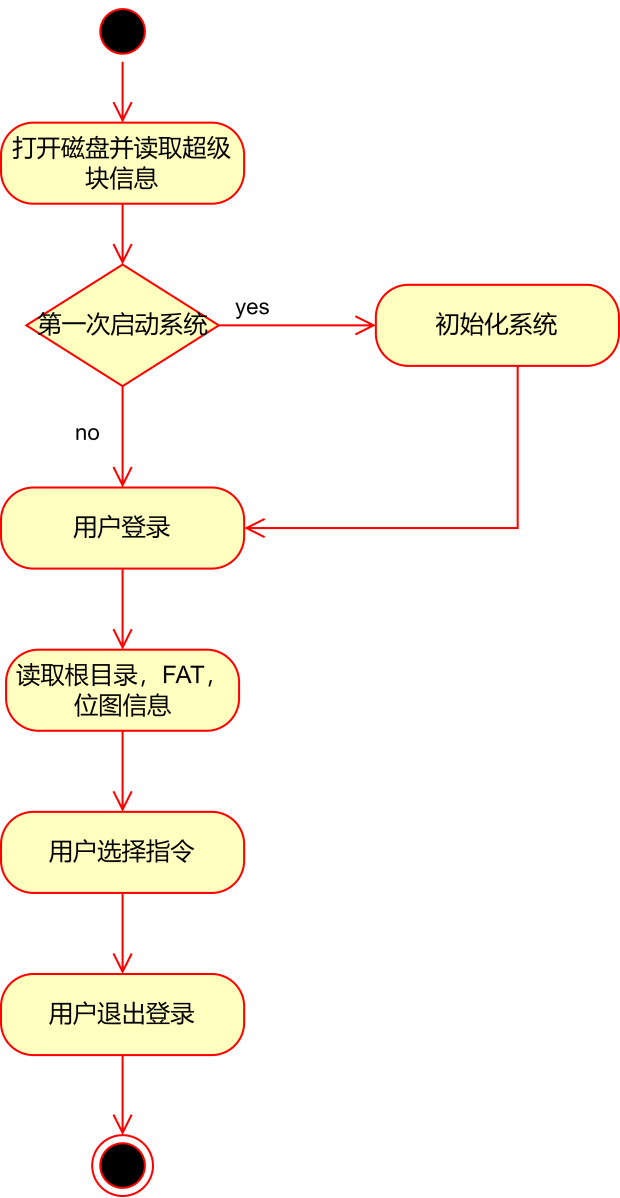
****

图 2 系统流程图

**3.3 数据结构设计**

表3.2.1 索引节点(inode)结构

|  |  |  |
| --- | --- | --- |
| 数据类型 | 标识符 | 中文描述 |
| unsigned short | i\_ino | inode序号 |
| uid\_t | i\_uid | 文件所属用户序号 |
| gid\_t | i\_gid | 文件所在组号 |
| unsigned short | i\_addr | FAT表中的第一个块序号 |
| type\_t | i\_type | 文件类型 |
| unsigned short | i\_acl | 文件权限 |
| unsigned short | i\_size | 文件大小 |
| unsigned short | i\_blkCnt | 块数 |
| time\_t | i\_atime | 最近访问时间 |
| time\_t | i\_mtime | 最近修改时间 |

表3.2.2 超级块（superBlk）结构

|  |  |  |
| --- | --- | --- |
| 数据类型 | 标识符 | 中文描述 |
| unsigned int | disk\_size | 磁盘大小 |
| unsigned short | free\_blk | 剩余空闲块数量 |
| unsigned int | free\_disk | 剩余磁盘大小 |
| struct inode\* | root | 根目录地址 |
| unsigned short[] | bitmap | 空闲块位图，0表示空闲，1表示占用 |
| unsigned short[] | bitmap\_inode | 空闲inode节点位图，0表示空闲，1表示占用 |
| unsigned short | inode\_count | 已用inode节点的数量 |
| unsigned short | inode\_free | 空闲inode节点数量 |
| bool | sys\_Status | 系统状态 |
| User[] | Users | 记录系统用户 |
| Unsigned short[] | FAT | FAT表 |

表3.2.3 用户（user）结构

|  |  |  |
| --- | --- | --- |
| 数据类型 | 标识符 | 中文描述 |
| uid\_t | uid | 用户id |
| unsigned short | u\_mode | 用户的类别，如管理员与普通用户 |
| Char[] | u\_name | 用户名 |
| Char[] | u\_pwd | 密码 |

表3.2.4 文件（file）结构

|  |  |  |
| --- | --- | --- |
| 数据类型 | 标识符 | 中文描述 |
| Char\* | f\_name | 文件名 |
| unsigned short | f\_ino | 文件inode序号 |

**3.4 系统基础属性设计**

|  |  |  |
| --- | --- | --- |
| 属性名 | 属性值 | 描述 |
| BLOCKSIZE | 256 | 块大小,单位是BYTE |
| INODESIZE | 32 | inode大小 |
| FILENAMESIZE | 14 | 文件名大小 |
| CLUSTERSIZE | (4\*BLOCKSIZE) | 簇大小 |
| CLUSTERNUM | 512 | 簇的数量 |
| FATSIZE | 512 | FAT表的大小 |
| BITMAPSIZE | (FATSIZE/16) | 位图的大小 |
| MAXUSERS | 8 | 用户数量限定 |
| SUBFILENUM | (CLUSTERSIZE/(FILENAMESIZE+2)) | 一个目录下文件数量限制 |
| SUPERBLKSIZE | 2 | 超级块占用两个簇 |
| INODEBLKSIZE | (INODESIZE\*CLUSTERNUM/CLUSTERSIZE) | inode区占用的簇大小 |
| SYSCLUSTERSIZE | (SUPERBLKSIZE+INODEBLKSIZE) | 系统保留区所占用的磁盘簇大小 |
| USERNAMESIZE | 38 | 用户名最大值 |
| PWDSIZE | 20 | 密码最大值 |
| ADMIN | 0x00 | 管理员 |
| USER | 0x01 | 普通用户 |

**3.5 通用模块接口设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 输入 | 输出 | 处理 | 功能描述 |
| GetInode | 超级块结构体的指针，  所属的用户id，  所属组id，  文件类型， | 分配好的inode节点 | 初始化新inode的各种属性，为inode分配空闲块，为inode分配inode序号 | 申请一个新的inode节点 |
| GetFATList | 要访问的inode指针，  超级块指针 | 该inode的地址块序列 | 遍历该inode占用的FAT条目，记录inode占用的数据块并返回一个数组 | 找到文件占用的块序列 |
| GetIno | 文件名，  当前目录子文件序列 | 文件的inode序号 | 遍历当前目录子文件列表，如果存在该文件则返回ino，否则返回-1 | 通过文件名获取inode序号 |
| FreeInode | 超级块指针，  文件ino | 删除状态信息 | 通过ino编号找到inode的地址并删除inode信息，将FAT和位图相应位置为0 | 释放一个inode节点 |
| SetCurPath | 超级块指针，  磁盘指针，  当前目录，  当前目录子文件列表，  新目录ino | 新的当前目录和当前目录文件列表 | 保存旧目录信息，跳转到新目录 | 目录跳转 |

**3.6 功能模块接口设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 输入 | 输出 | 处理 | 功能描述 |
| sysInit | 待初始化的超级块指针，  打开的磁盘 | 初始化完毕的超级块， | 初始化超级块各个属性，申请一个新inode作为根目录，创建二级目录并为其申请新的inode，最后将inode信息和根目录信息写入到磁盘当中 | 初始化文件系统 |
| Mkdir | 超级块指针，  当前目录子文件列表，  所属用户id，  所属组id，  打开的磁盘 | 创建完毕的新目录 | 为新目录申请一个inode，并将文件名和ino绑定写入到当前目录的文件列表内 | 在当前目录下创建一个新目录 |
| CreatFile | 超级块指针，  当前目录子文件列表，  所属用户id，  所属组id，  文件类型，  打开的磁盘 | 新创建的文件 | 为新文件申请一个inode，并将文件名和ino绑定写入到当前目录的文件列表内 | 在当前目录下创建一个新文件 |
| Login | 超级块指针，  输入的用户名和密码，  当前用户信息指针 | 登录状态信息，当前用户信息 | 对比各个用户的用户名和密码，若是全部符合则返回登陆成功，并将符合的用户置为当前用户 | 用户登录 |
| Logout | 无 | 无 | 保存系统更改 | 用户登出 |
| Halt | 无 | 无 | 保存超级块信息和inode信息以及当前目录的文件信息，保存完毕后退出程序 | 停机，即文件系统关闭 |
| openFile | 文件名，  系统打开文件列表 | 文件打开状态，  文件inode序号 | 根据文件名获取文件ino，判断其是否在系统打开文件列表中，如果不在将其打包放入系统打开文件列表。 | 打开文件 |
| closeFile | 文件名，  系统打开文件列表 | 文件打开状态，  系统打开文件列表 | 查找打开文件列表，如果存在该文件则将其从系统打开文件列表中删除 | 关闭文件 |
| deleteFile | 文件名，  当前目录文件列表 | 文件删除状态，  当前目录文件列表 | 查找当前目录文件列表，如果存在，则删除并删除inode信息 | 删除文件 |
| readFile | 文件名ino，  读取的基地址，  读取的文件偏移地址，  读取的块大小，  读取的块数量，  存放数据的空间地址 | 读取状态 | 通过ino找到inode位置，再依据inode信息找到数据块，读取要求的数据 | 读文件 |
| writeFile | 文件ino，  要写入的数据指针，  写入的基地址。  写入的文件偏移。  写入的块大小。  写入的数量 | 写入状态 | 通过ino找到inode位置，再依据inode信息找到数据块，写入要求的数据 | 写文件 |
| chDir | 要转移的目录名，  当前目录指针，  当前目录文件列表 | 新的当前目录，  新的当前目录文件列表 | 查找当前目录文件列表，如果存在名字一样的子目录则先将当前目录子文件信息保存再磁盘中，然后当前目录指针转移到下一个目录，并更新当前目录文件列表。  特殊情况：文件名为”..”,则转移到当前目录的父目录。 | 改变目录 |
| ls | 无 | 无 | 列出当前目录子文件列表中的所有有效项。 | 列出文件列表 |