

# 操作系统实验报告

## 实验四 文件系统

学号	姓名
10231016	童浩（组长）
10231008	解佳琦
10231024	陈宇宁
38230213	邱伟豪

## 1 需求说明

### 1.1 基本需求

#### 1.1.1 显示目录列表

设计并实现一个目录列表函数（类似于 Linux 的 Shell 命令 `ls`）。该函数只需要包含基本命令即可，不必支持许多选项。函数格式为：

```
int fd_ls();
```

#### 1.1.2 改变当前目录

设计并实现一个改变目录函数，即把当前目录切换到上一层目录或当前目录的子目录中（无需处理路径名），函数格式为：

```
int fd_cd(char *directory);
```

本函数假设U 盘上已存在子目录。需要在文件系统中使用一个静态变量来代表当前目录。本函数要对当前目录变量进行操作，并且可以返回上一级目录，需要在文件系统中使用一个静态变量来代表当前目录的父目录。

### 1.1.3 删除文件

设计并实现一个删除文件函数，该函数使用要删除的文件名（在当前目录中）作为参数，函数格式为：

```
int fd_df(char *name);
```

该函数需要查找文件，遍历FAT 中的链接，设置FAT 中的每个簇项并将其标志为未使用，更新目录项。在删除的情况中，要注意文件的隐藏、只读和系统属性。任何具有这些设置的文件都不能删除。

### 1.1.4 创建文件

设计并实现一个创建文件函数，该函数使用要创建的文件名和文件大小为参数，函数格式为：

```
int fd_cf(char *filename , int size);
```

该函数需要遍历FAT 表中的链接，找出FAT 表中的能存放下文件大小的空簇，并更新目录项，可以用fd\_ls()函数查询到创建的文件。

## 1.2 进阶需求

### 1.2.1 删除目录

通常需要先判断目录是否为空目录，若目录不为空，则需给出提示，并删除其包含的所有子目录和文件；若是空目录则可直接删除。本实验通过更改 fd\_df 实现，命令保持不变。

### 1.2.2 绝对路径和多级目录

这里需要对输入的目录路径字符串进行解析，然后逐级查找目录。

### 1.2.3 创建文件时写入实际内容

对 fd\_cf()函数进行改进，使其可以向文件中写入实际内容，并根据写入的内容计算文件实际大小。

## 1.2.4 非根目录信息读取

对 `fd_ls()` 函数进行完善，增加对全部非根目录信息的读取。

## 1.3 自行改进

### 1.3.1 创建目录

增加 `mkdir` 命令，可以创建目录，但是不可以于当前目录下其他文件或文件名重名。

### 1.3.2 更改文件名显示方式

原程序所有文件名均显示为大写，并且创建的文件也会变成大写。本实验改进为用实际输入的文件名。

### 1.3.3 绝对路径和相对路径的支持

在所有命令中都以用绝对路径和相对路径

### 1.3.4 删除非空文件夹提示内容

删除非空文件夹时会给出提示，提示文件夹中包含的文件和文件夹数目。

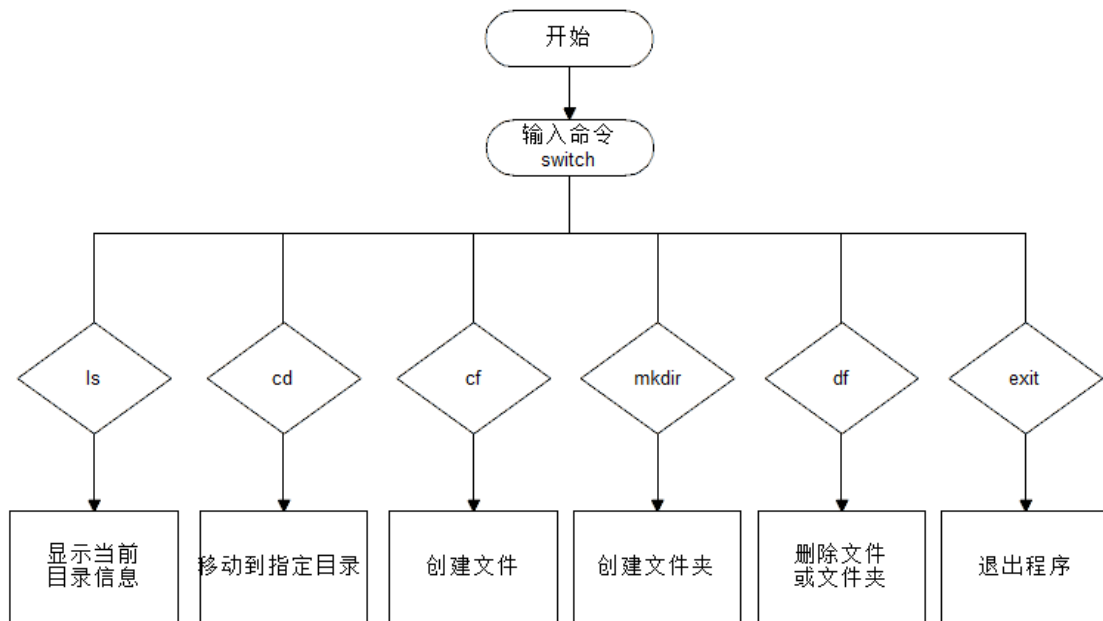
### 1.3.5 增加命令提示符

增加命令提示符，给出当前绝对路径。

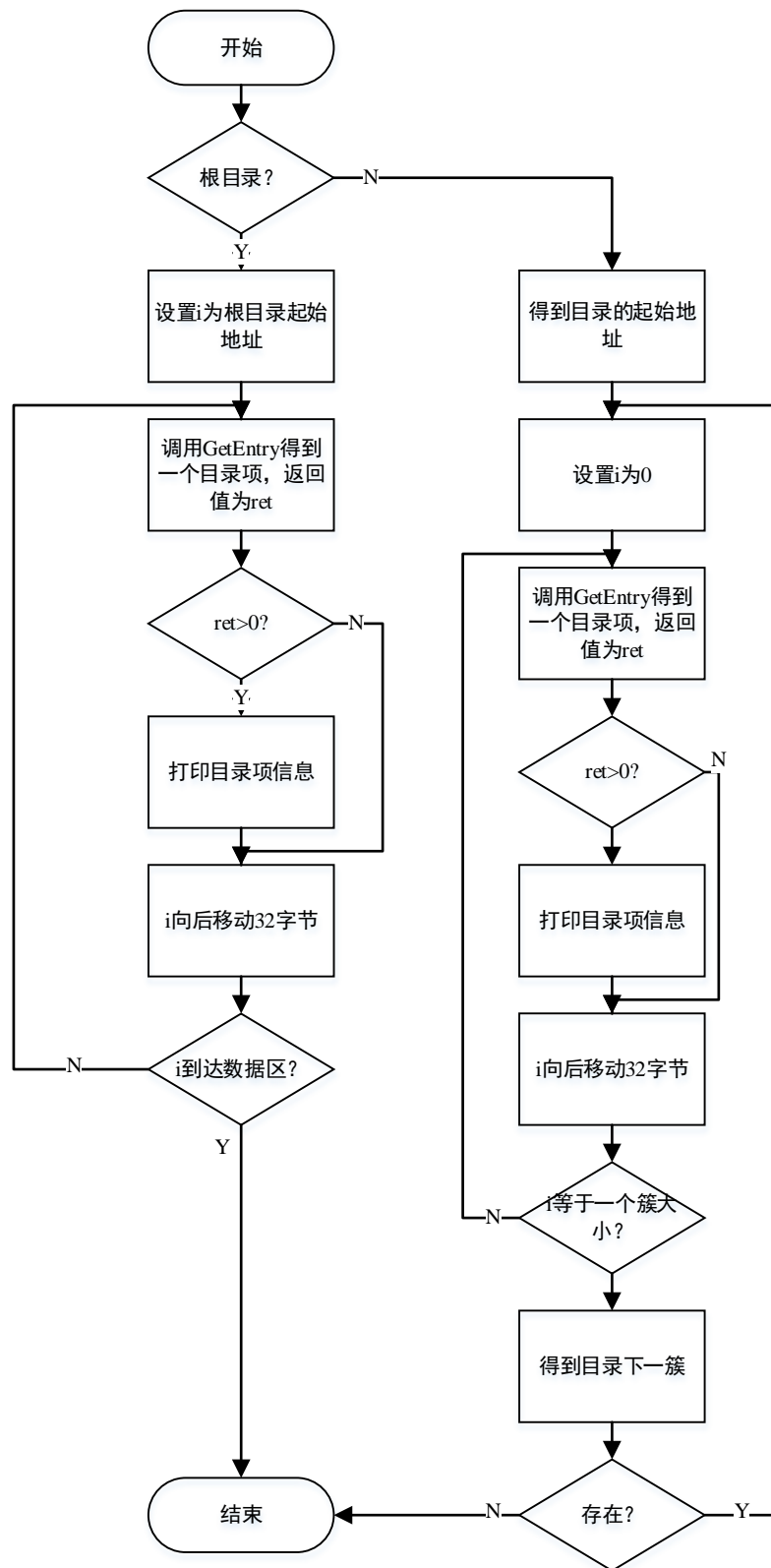
## 2 设计说明

### 2.1 结构设计

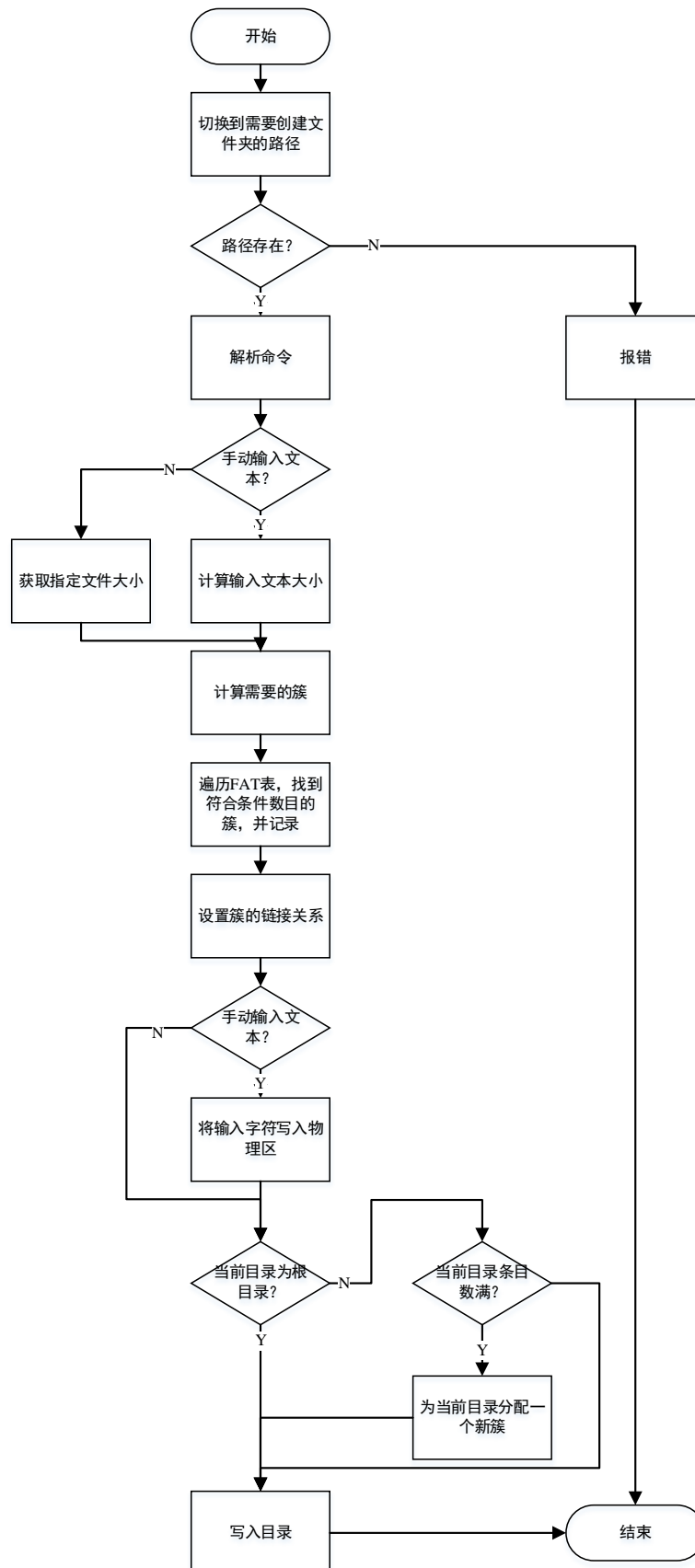
#### 2.1.1 程序整体结构图



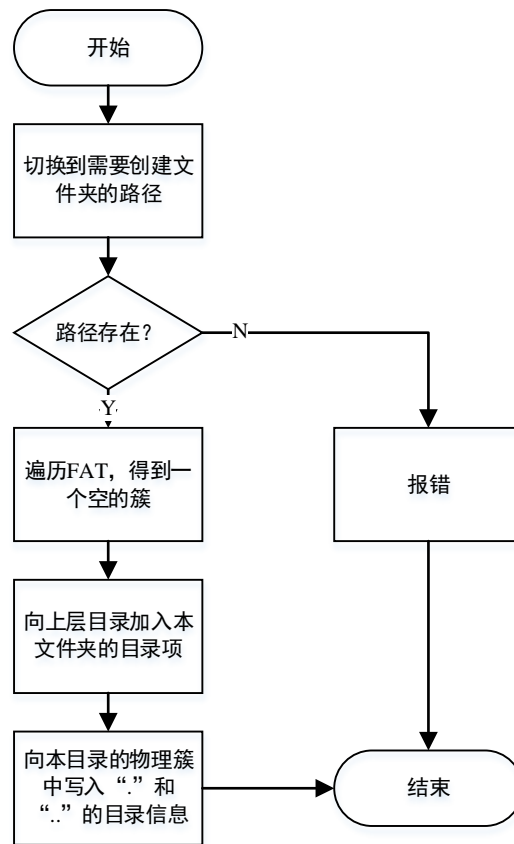
### 2.1.2 ls 列出目录信息命令流程图



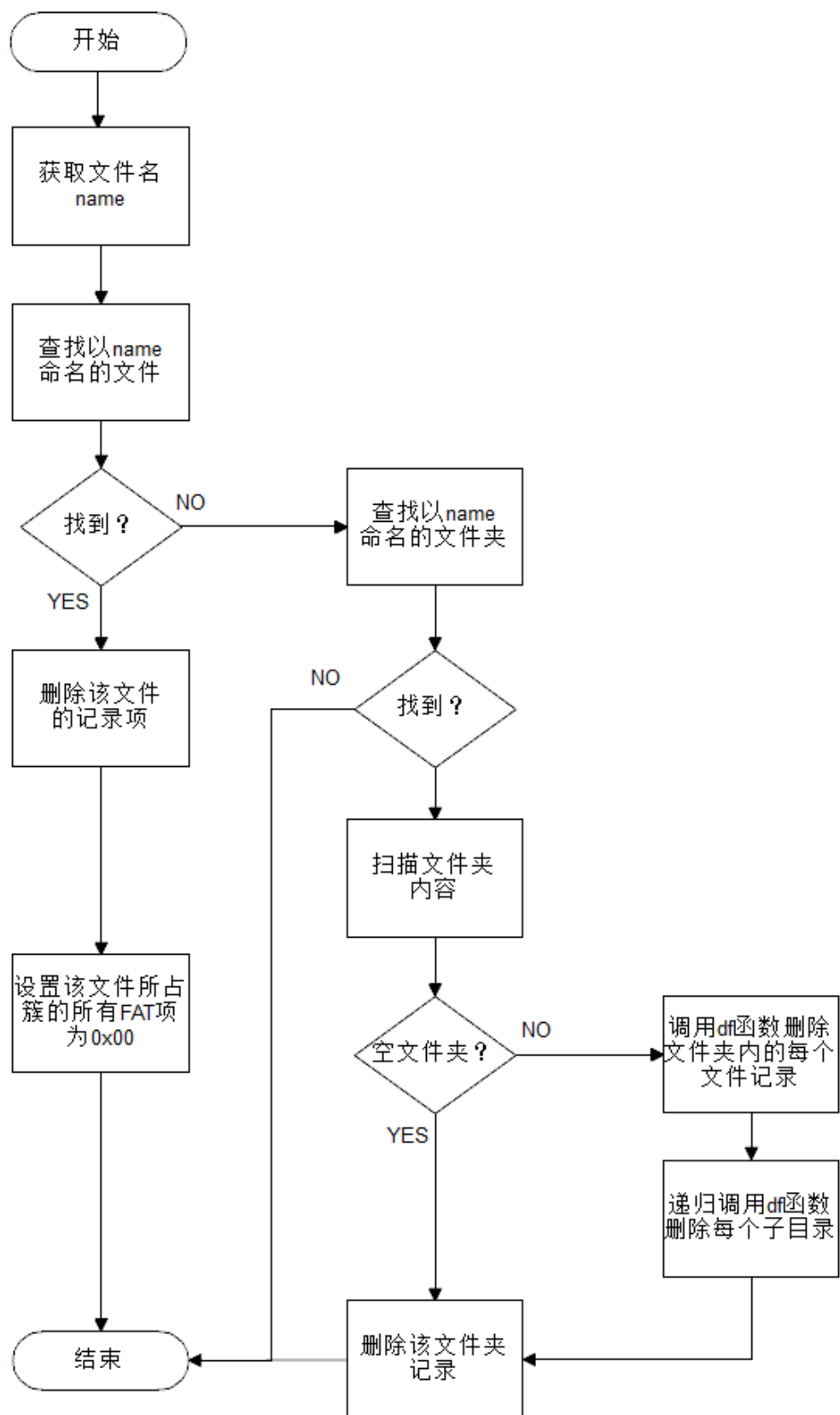
### 2.1.3 cf 创建文件命令流程图



## 2.1.4 mkdir 创建文件夹命令流程图

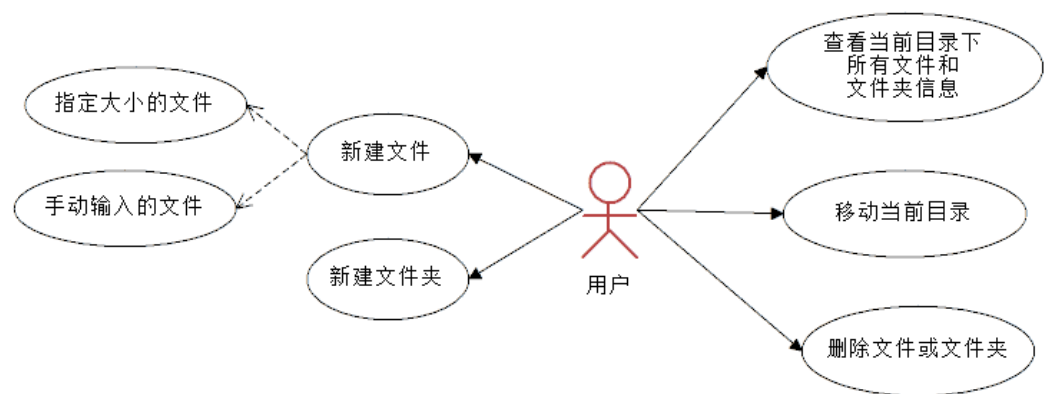


### 2.1.5 df 删除文件或文件夹命令流程图





## 2.2 用例图



## 2.2 功能设计

这里给出各功能模块的设计方案或实现方法。

### 2.2.1 重要的数据结构设计

#### (1) 文件系统的启动项显示信息

```
struct BootDescriptor_t{  
  
    unsignedchar Oem_name[9]; /*0x03-0x0a*/  
  
    int BytesPerSector;      /*0x0b-0x0c*/  
  
    int SectorsPerCluster;   /*0x0d*/  
  
    int ReservedSectors;     /*0x0e-0x0f*/  
  
    int FATs;                /*0x10*/  
  
    int RootDirEntries;       /*0x11-0x12*/  
  
    int LogicSectors;         /*0x13-0x14*/  
  
    int MediaType;           /*0x15*/  
  
    int SectorsPerFAT;        /*0x16-0x17*/  
  
    int SectorsPerTrack;      /*0x18-0x19*/  
  
    int Heads;               /*0x1a-0x1b*/  
  
    int HiddenSectors;        /*0x1c-0x1d*/  
  
};
```

## (2) 目录项信息

```
struct Entry{

    unsignedchar short_name[12];    /*字节0-10， 11字节的短文件名*/

    unsignedchar long_name[27];     /*未使用， 26字节的长文件名*/

    unsignedshort year,month,day;    /*22-23字节*/

    unsignedshort hour,min,sec;      /*24-25字节*/

    unsignedshort FirstCluster;     /*26-27字节*/

    unsignedint size;                /*28-31字节*/

    /*属性值                                11字节

    *7  6  5  4  3  2  1  0

    *N  N  A  D  V  S  H  R          N未使用

    */

    unsignedchar readonly:1;

    unsignedchar hidden:1;

    unsignedchar system:1;

    unsignedchar vlabel:1;

    unsignedchar subdir:1;

    unsignedchar archive:1;

};
```

## (3) 请求的结构设计

```
typedef struct
{
    pid_t pro;                //发出请求的进程的进程号
    MemoryAccessRequestType reqType;    //访存请求类型
    unsigned long virAddr;    //外存地址
    BYTE value;                //写请求的值
} MemoryAccessRequest, *Ptr_MemoryAccessRequest;
```

## 2.2.2 主要函数或接口设计

这里给出主要函数或接口的功能说明、实现方法和调用关系。

### 2.2.2.1 函数功能说明

- (1) 函数原型: void clearfile()

功能说明: 清除当前分区上的文件索引及数据信息。

参数说明: 无

- (2) 函数原型: void ScanBootSector()

功能说明: 打印文件系统命令程序的启动项记录信息。

参数说明: 无参数

- (3) 函数原型: void FileNameFormat(unsignedchar \*name)

功能说明: 格式化文件名, 便于索引时的比较。

参数说明:

unsignedchar \*name: 当前需要格式化的文件的名称

- (4) 函数原型: int GetEntry(struct Entry \*pentry)

功能说明: 从根目录或文件簇中得到文件表项。成功则返回偏移值; 失败返回负值。

参数说明:

struct Entry \*pentry: 储存结果的变量

- (5) 函数原型: int ScanEntry(char \*entryname, struct Entry \*pentry, int mode)

功能说明: 搜索当前目录, 查找文件或目录项。若返回偏移量大于 0 则成功; 是-1 则失败。

参数说明:

char \*entryname: 要查找的目录项文件名

struct Entry \*pentry: 储存结果的变量

int mode: 查找模式。mode=1, 为目录表项; mode=0, 为文件。

- (6) 函数原型: int fd\_ls()

功能说明: ls 命令的功能实现

参数说明:

无

(7) 函数原型: `int ls(int echo,int* dircount,int* filecount)`

功能说明: 对当前目录表项进行遍历, 得到表项信息以及该目录下文件和文件夹个数

参数说明:

`int echo`: 是否显示表项信息。为 1 显示, 用于 `fd_ls()` 中; 否则, 不显示, 用于删除目录。

`int* dircount`: 子目录计数

`int* filecount`: 文件计数

(8) 函数原型: `int fd_cd(char *dir)`

功能说明: 改变目录。返回值: 正数, 子目录嵌套层数增加数目; 负数, 子目录嵌套层数减少数目; -10000, 失败。

参数说明:

`char *dir`: 需要改变的路径

(9) 函数原型: `int cd(char *dir)`

功能说明: 改变目录到父目录或子目录。返回值: 1 是子目录; -1 是父目录; -2 是失败。

参数说明:

`char *dir`: 需要改变的路径

(10) 函数原型: `unsigned short GetFatCluster(unsigned short prev)`

功能说明: 在 FAT 表中获得下一簇的位置。返回值为下一簇

参数说明:

`unsigned short prev`: 当前簇的位置

(11) 函数原型: `void ClearFatCluster(unsignedshort cluster)`

功能说明: 清除 FAT 表中的簇信息。无返回值。

参数说明:

`unsigned short cluster`: 要清除的簇的位置

(12) 函数原型: `int WriteFat()`

功能说明: 将改变的 FAT 表值写回 FAT 表。

参数说明:

无

(13) 函数原型: `int ReadFat()`

功能说明: 读 FAT 表的信息, 存入 `FATBuf[]` 中。

参数说明:

无

(14) 函数原型: `int fd_df(char *dirname)`

功能说明: 删除某路径下的文件或文件夹

参数说明:

`char *dirname`: 要删除的路径

(15) 函数原型: `int df(char *filename)`

功能说明: 删除当前目录下的文件或文件夹

参数说明:

`char *filename`: 要删除的文件或文件夹名

(16) 函数原型: `int delete_file(struct Entry *pentry, int lastdir)`

功能说明: 删除一个文件

参数说明:

`struct Entry *pentry`: 该文件的目录信息

`int lastdir`: 文件所在目录层数

(17) 函数原型: `int delete_dir(struct Entry *pentry, int lastdir)`

功能说明: 删除一个目录

参数说明:

`struct Entry *pentry`: 该子目录的目录信息

int lastdir: 该子目录所在目录层数

(18) 函数原型: int fd\_mkdir(char \*dirname)

功能说明: 创建一个目录

参数说明:

char \*dirname: 创建目录的路径

(19) 函数原型: int fd\_cf(char \*dirname,int size)

功能说明: 创建一个文件

参数说明:

char \*dirname: 创建文件的路径

int size: 文件大小

(20) 函数原型: int cf(char \*filename,int size,int mode)

功能说明: 在当前目录下创建一个文件或文件夹

参数说明:

char \*filename: 创建文件或文件夹的名

int size: 文件大小, 若为文件夹, size 为 0

int mode: 1 为创建目录, 0 为创建文件

(21) 函数原型: int fd\_cf\_str(char \*file\_name,char \*str)

功能说明: 创建一个文件并写入自定义内容

参数说明:

char \* file\_name: 创建文件的路径

char \*str: 写入文件的文本

(22) 函数原型: do\_usage()

功能说明: 显示命令以及使用方法。

参数说明:

无

(23) 函数原型: void getcurpath(char\* temp)

功能说明: 获取当前路径的信息的字符串

参数说明:

char \* temp: 储存当前路径的字符串

(24) 函数原型: void printpath()

功能说明: 打印当前目录

参数说明:

无

(25) 函数原型: int main()

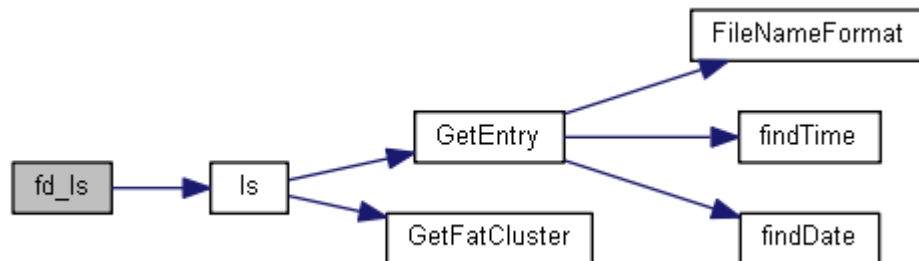
功能说明: 主函数

参数说明:

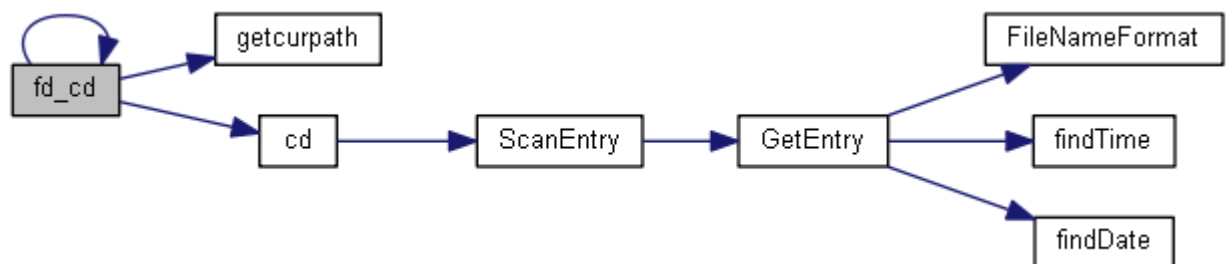
无

### 2.2.2.2 函数调用关系

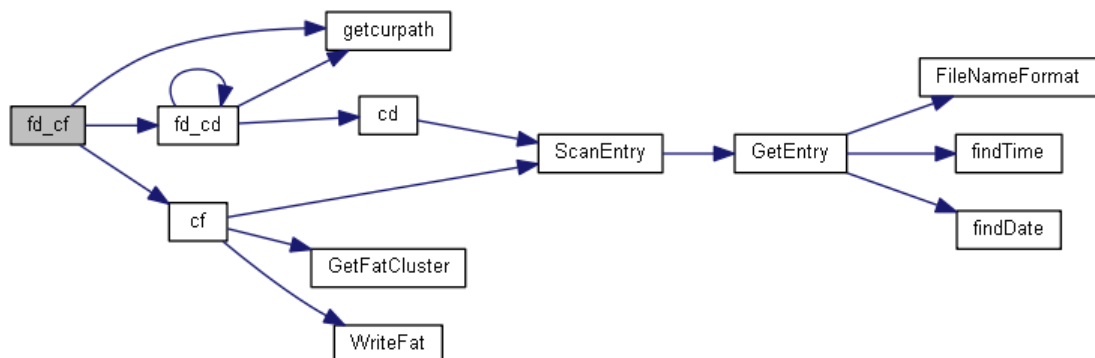
(1) ls 命令处理函数 fd\_ls 调用关系图



(2) cd 命令处理函数 fd\_cd 调用关系图

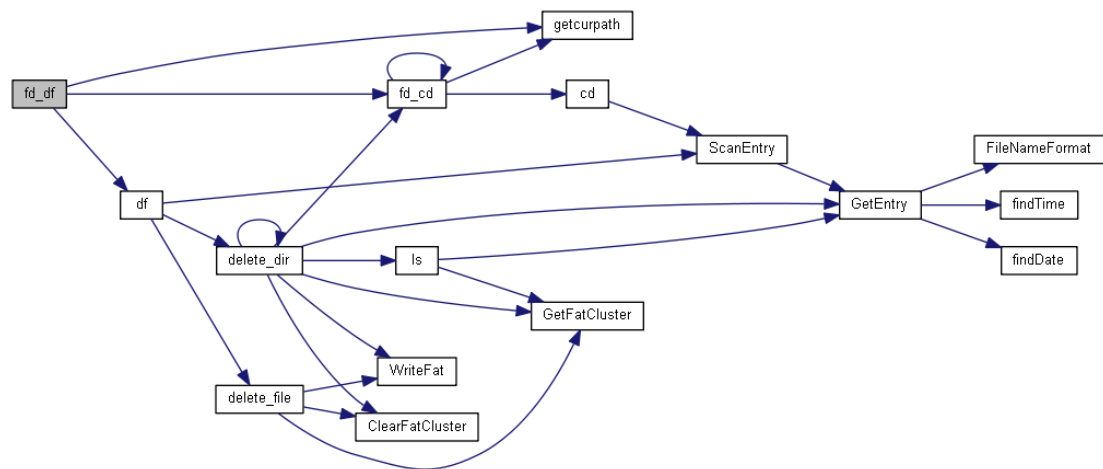


(3) cf 命令处理函数 fd\_cf 调用关系图

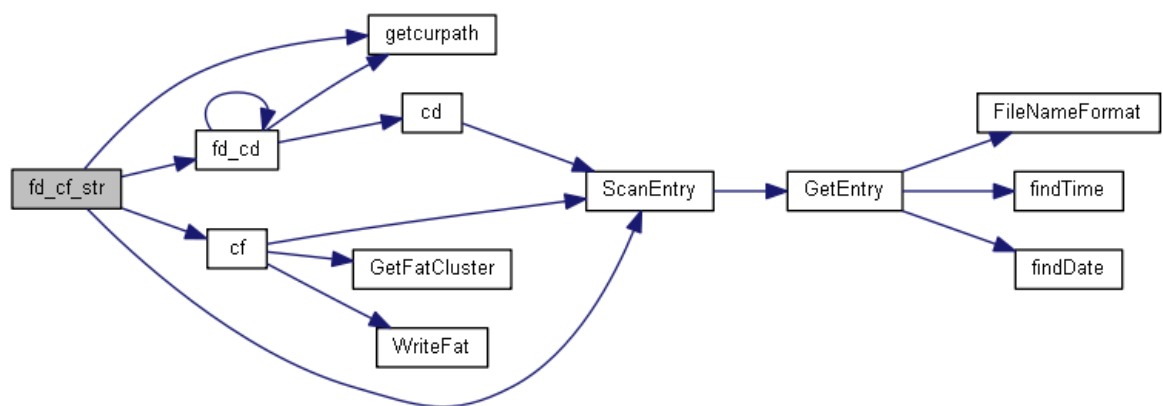




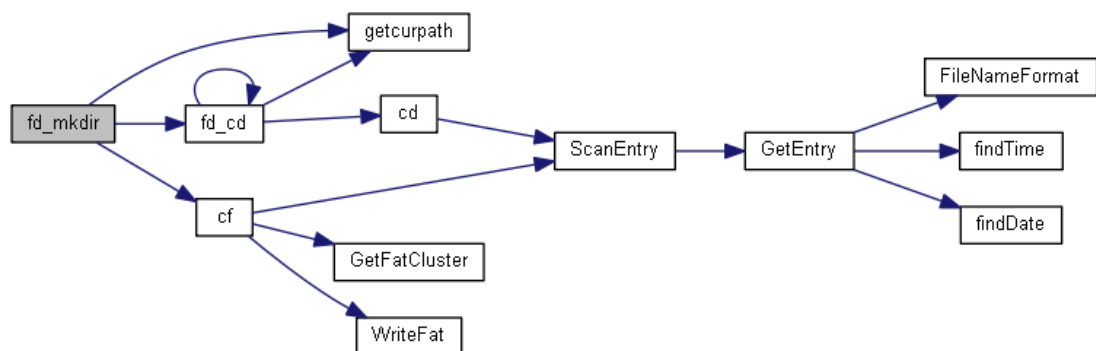
(4) df 命令处理函数 fd\_df 调用关系图



(5) cf -m 命令处理函数 fd\_cf\_str 调用关系图



(6) mkdir 命令处理函数 fd\_mkdir 调用关系图



## 3 测试和使用说明

### 3.1 使用说明

列出程序的开发环境，如操作系统、使用的编程语言、开发工具和组件等。

列出程序的运行环境，如操作系统、必要的运行库等。

开发环境：

操作系统：Ubuntu12.04

编程语言：C 语言

开发工具：CodeBlocks

运行环境：

操作系统：Ubuntu 12.04

运行库： Gcc

### 3.2 测试说明

在源文件目录，打开 filesys.h, 更改 DEV\_NAME 为使用的设备路径。在终端输入 make, 生成可执行文件。

(1) 进入文件系统：

```
root@ubuntu:~/Desktop/filesys# ./filesys
Oem_name          mkdosfs
BytesPerSector    512
SectorsPerCluster 32
ReservedSector    32
FATs              2
RootDirEntries    512
LogicSectors      0
MedioType         248
SectorPerFAT      256
SectorPerTrack    63
Heads             255
HiddenSectors     0
please input a command, including followings:
      ls          list all files
      cd <dir>    change directory
      cf <filename> <size> create a file
      cf <filename> -m put in the content right away
      mkdir <dirname> create a directory
      df <file>   delete a file
      exit       exit this system
/>
```

(2) mkdir 命令创建一个文件夹:

```
/>mkdir testDir
/>ls
Root_dir
      name      date      time      cluster size      attr
      testDir   2013:6:23   20:44:0    2        16384      dir
/>
```

创建了一个 testDir 的文件夹

(3) cd 命令进入和退出目录:

```
/>cd testDir
/testDir>ls
testDir_dir
      name      date      time      cluster size      attr
      .         2013:6:23   20:44:0    2          0      dir
      ..        2013:6:23   20:44:0    0        16384      dir
/testDir>cd ..
/>a
```

图中可以看出, 目录改变后, 命令提示符中的路径也相应改变

(4) 使用绝对路径和相对路径进入目录:

```
/>cd testDir
/testDir>mkdir level2
/testDir>ls
testDir_dir
      name      date      time      cluster size      attr
      .         2013:6:23   20:44:0    2          0      dir
      ..        2013:6:23   20:44:0    0        16384      dir
      level2    2013:6:23   20:46:32    3        16384      dir
/testDir>cd ..
/>mkdir level1
/>cd level1
/level1>cd /testDir/level2
/testDir/level2>cd ../../level1
/level1>cd ../testDir
/testDir>
```

在 testDir 下创建了一个名为 level2 的文件夹, 在根目录下创建了 level1 文件夹。

在 level1 下使用绝对路径/testDir/level2 进入 level2

在 level2 下使用相对路径../../level1 进入 level1

(5) 创建文件, 参数为文件名和大小:

```
/testDir>cf testfile1 10
/testDir>ls
testDir_dir
      name      date      time      cluster size      attr
      .         2013:6:23   20:44:0    2          0      dir
      ..        2013:6:23   20:44:0    0        16384      dir
      level2    2013:6:23   20:46:32    3        16384      dir
      testfile1 2013:6:23   20:50:8    5         10      file
```

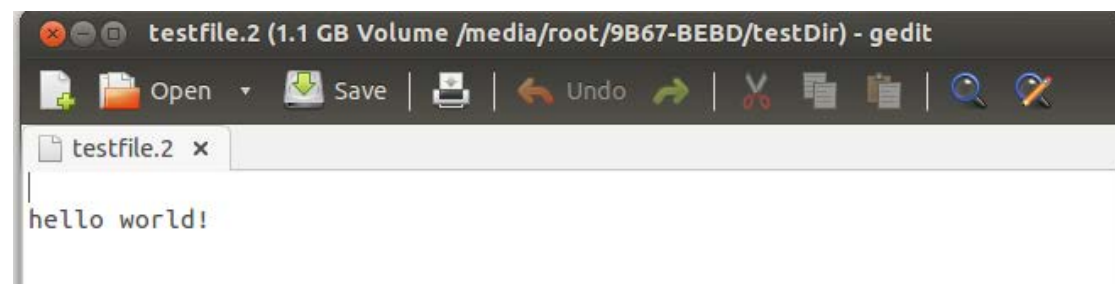
在 testDir 下创建文件 testfile1，大小为 10B。

(6) 创建文件，参数为文件名和文件内容：

```
/testDir>cf testfile2 -m
please put in the content ending with Ctrl+D
hello world!
/testDir>ls
testDir_dir
  name      date       time        cluster size    attr
  .         2013:6:23  20:44:0     2         0      dir
  ..        2013:6:23  20:44:0     0        16384   dir
  level2    2013:6:23  20:46:32    3        16384   dir
  testfile1 2013:6:23  20:50:8     5         10     file
  testfile2 2013:6:23  20:51:36    6         14     file
/testDir>
```

创建 testfile2，文件内容为 hello world!

打开该文件后看到其内容与输入相同：



(7) 删除文件和文件夹：

```
/>df testDir/testfile1
/>cd testDir
/testDir>ls
testDir_dir
  name      date       time        cluster size    attr
  .         2013:6:23  20:44:0     2         0      dir
  ..        2013:6:23  20:44:0     0        16384   dir
  level2    2013:6:23  20:46:32    3        16384   dir
  testfile2 2013:6:23  20:51:36    6         14     file
/testDir>
```

这里在根目录使用相对路径删除了 testfile1 文件。

```
/>df testDir
There exists 1 dir(s) and 1 file(s), are you sure to delete the dir?
(Y)es or (N)o:
y
```

在根目录删除文件夹 testDir，因为文件夹非空，会显示内容，并提示是否要删除。

```
/>ls
Root_dir
      name      date      time      cluster size      attr
      level1    2013:6:23    20:46:48      4      16384      dir
/>
```

上图可以看出，文件夹已经删除。

(8) 使用绝对路径和相对路径创建文件或文件夹：

```
/>cf level1/testfile3 20
/>mkdir /level1/dirdir
/>cd level1
/level1>ls
level1_dir
      name      date      time      cluster size      attr
      .         2013:6:23    20:46:48      4         0      dir
      ..        2013:6:23    20:46:48      0      16384      dir
      testfile3 2013:6:23    20:59:18      2         20      file
      dirdir    2013:6:23    20:59:30      3      16384      dir
/level1>
```

如图，在根目录使用相对路径创建了 testfile3 文件，用绝对路径创建了 dirdir 文件夹。

# 4 会议记录

会议次数	任务	子任务	本次内容	后期计划	会议日期
1	调试工程源代码	配置环境, 调试成功, 了解实验目的	讨论基础知识, 了解实验基本流程	添加创建目录功能	2013/6/1
2	添加创建目录功能、手动输入文件内容	查阅资料, 更改源程序以添加创建目录功能	添加创建目录功能、手动输入文件内容, 并调试测试	删除文件夹功能、绝对路径相对路径支持	2013/6/7
3	添加删除文件夹功能、绝对路径相对路径支持	从 fd_df 基础上更改, 支持删除文件夹	添加删除文件夹功能、绝对路径相对路径支持	ls 扩展	2013/6/10
4	ls 扩展	动态分配目录文件大小	实现任务, 整体测试程序	继续调试, 排除 bug, 准备检查	2013/6/13

## 5 其它说明

学号	姓名	分工情况	工作量比例
10231016	童浩（组长）	总体程序框架，df 命令，以及删除、创建文件夹，总体调试	40%
10231008	解佳琦	cd 命令的实现。	30%
10231024	陈宇宁	cd 命令的路径拆分，绝对路径与相对路径的实现	15%
38230213	邱伟豪	ls 命令的实现	15%

## 6 程序清单

### 6.1 源代码

（路径：~\源代码\filesys）

#### 6.1.1 程序源文件

filesys.c  
filesys.h

#### 6.1.2 其他文件

makefile

### 6.2 可执行程序

（路径：~\可执行程序\）  
filesys