



编号：\_\_\_\_39061416\_\_\_\_

版本：\_\_\_\_1.0\_\_\_\_

**北京航空航天大学**  
B E I H A N G U N I V E R S I T Y

**Project : 文件系统**

编写	黄建宇	2012. 05. 24
复查	黄建宇	2012. 05. 24
批准	黄建宇	2012. 05. 24



## 目录

1 小组成员.....	1
2 实验目的与总体结构.....	1
2.1 实验目的.....	1
2.2 需求说明.....	1
2.3 设计说明.....	2
2.3.1 结构设计.....	2
2.3.2 功能设计.....	3
3 实验内容.....	6
3.1 实现方法.....	6
3.2 测试和使用说明.....	8
3.2.1 程序开发环境: .....	8
3.2.2 运行环境.....	8
3.2.3 安装说明.....	8
3.2.4 测试用例和运行结果分析: .....	9
4 每个人的工作与会议记录.....	19
4.1 会议时间表与会议记录.....	19
5 其他说明.....	19
5.1 组内成员任务分工说明.....	19
5.2 实验完成部分说明.....	19
6 程序清单.....	20
7 实验心得.....	20
7.1 心得.....	20
7.2 建议.....	20



# 1 小组成员

每个人的贡献大小以此排名为依据，靠前的为贡献较大者

39061416 黄建宇

# 2 实验目的与总体结构

了解实验的目的，对于作业的整体设计说明，要求思路清晰，表达明确

## 2.1 实验目的

- 了解文件管理系统的作用和工作方式。
- 学习 Linux VFS 的实现机制及 inode、超级块等相关概念。
- 了解 FAT 文件系统的结构。
- 学习文件管理系统的一般开发方法。

## 2.2 需求说明

准备一张FAT16 格式的U 盘（可以将U 盘在Windows 下直接格式化为FAT16 格式），在Linux 下编写一个文件系统管理程序，对U 盘上的文件进行管理。具体要求如下：

### 1.基本要求：

（1）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 ud_ls();
```

（2）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 ud_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。

（3）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 ud_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。

（4）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 ud_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ud\_ls()函数查询到创建的文件。



## 2.扩展要求:

- (1) 增加删除目录的功能。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。
- (2) 增加对绝对路径和多级目录的支持。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。
- (3) 对ud\_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。
- (4) 对ud\_ls()函数进行完善，增加对全部非根目录信息的读取（本试验中只读取了一个扇区的非根目录信息）。

## 2.3 设计说明

### 2.3.1 结构设计

#### ➤ 程序逻辑设计

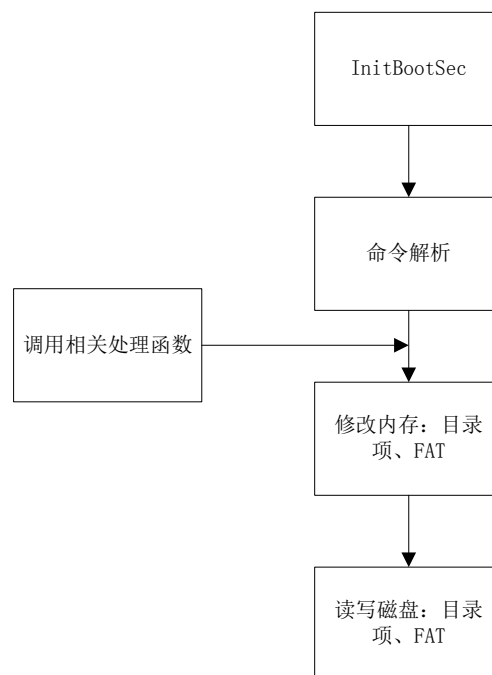


图2.1程序逻辑流程图

#### ➤ 命令解析流程图

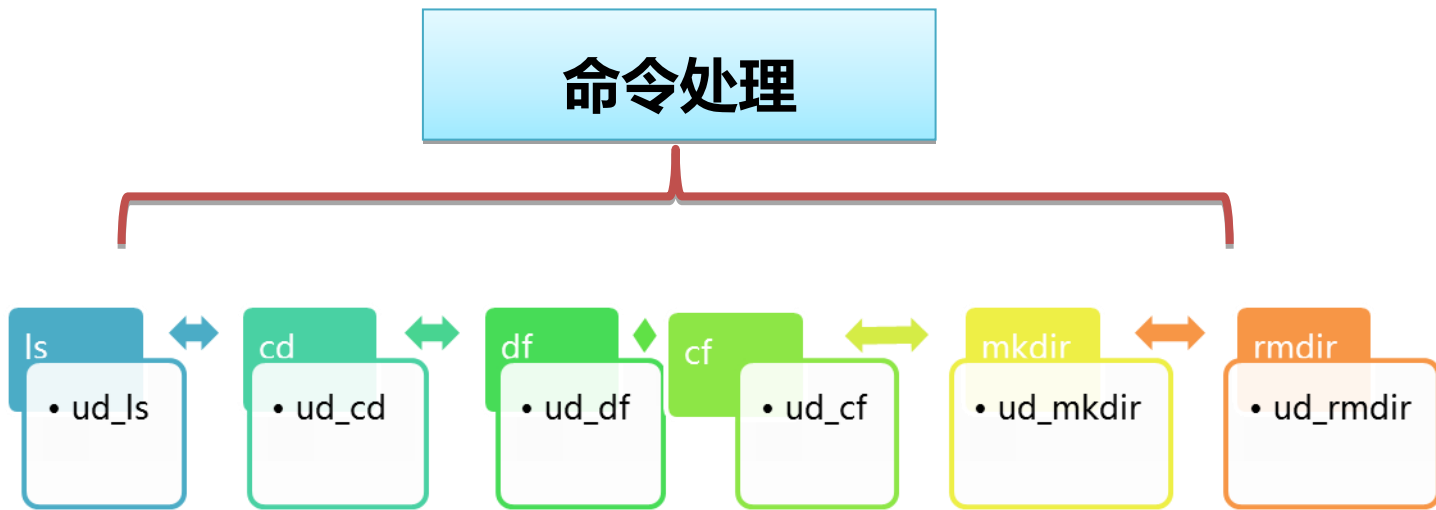


图2. 2命令解析流程图

### 2.3.2 功能设计

#### 1.数据结构:

##### ➤ 全局变量或常量

```
#define DEVNAME "/dev/ram1"
```

```
#define DIR_ENTRY_SIZE 32
```

```
#define SECTOR_SIZE 512
```

```
#define CLUSTER_SIZE 512*4
```

```
#define FAT_ONE_OFFSET 512
```

```
#define FAT_TWO_OFFSET 512+250*512
```

```
#define ROOTDIR_OFFSET 512+250*512+250*512+512
```

```
#define DATA_OFFSET 512+250*512+250*512+512*32
```

```
#define CombineByte(low,high) ((high)<<8|(low))
```

```
#define CombineWord(lowest,lower,higher,highest) ((highest)<< 24|(higher)<<16|(lower)<<8|lowest)
```

```
//attribute mask
```

```
#define ATTR_READONLY 0x01
```

```
#define ATTR_HIDDEN 0x02
```

```
#define ATTR_SYSTEM 0x04
```

```
#define ATTR_VLABEL 0x08
```



```
#define ATTR_SUBDIR 0x10
#define ATTR_ARCHIVE 0x20
```

```
//time mask
#define MASK_HOUR 0xf800
#define MASK_MIN 0x07e0
#define MASK_SEC 0x001f
```

```
//date mask
#define MASK_YEAR 0xfe00
#define MASK_MONTH 0x01e0
#define MASK_DAY 0x001f
```

```
int fd;
struct Boot_Des bdptor;
struct Entry *curdir = NULL;
int dirno = 0;
struct Entry* fatherdir[10];
```

```
unsigned char fatbuf[512*250];
```

### ➤ 启动记录

```
struct Boot_Des{
    unsigned char 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
};
```

### ➤ 根目录项

```
struct Entry{
    unsigned char short_name[12]; //0-10,11
    unsigned char long_name[27];  //26
    unsigned short year,month,day; //22-23
    unsigned short hour,min,sec;   //24-25
```



```
unsigned short FirstCluster;    //26-27
unsigned int size;              //28-31
//attribute    11 bytes
//7  6  5  4  3  2  1  0
//N  N  A  D  V  S  H  R      N not in use

unsigned char readonly:1;
unsigned char hidden:1;
unsigned char system:1;
unsigned char vlabel:1;
unsigned char subdir:1;
unsigned char archive:1;
};
```

## 2.主要函数与接口说明

```
void welcome(); //welcome information
void changeTimeDate(unsigned char Timeinfo[2],unsigned char Dateinfo[2]); //change time and date
void transformDate(unsigned short *year, unsigned short *month, unsigned short *day, unsigned char info[2]); //transform the date
void transformTime(unsigned short *hour, unsigned short *min, unsigned short *sec, unsigned char info[2]); //transform the time
void TransformFileName(unsigned char *name); //filename format transform
void initBootSec(); //scan the boot sector:initial part
int GetEntry(struct Entry *pentry); //get the directory entry
int ScanEntry (char *entryname,struct Entry *pentry,int mode); //scan the entry
unsigned short GetFatCluster(unsigned short prev); //get the fat cluster
void ClearFatCluster(unsigned short cluster); //clear the fat cluster
int WriteFat(); //write fat
int ReadFat(); //read fat
int ud_ls(); //ls command
int ud_cd(char *dir); //cd command
int ud_rmdir(char *filename); //rmdir command
int ud_df(char *filename); //df command
int ud_mkdir(char *filename); //mkdir command
int ud_cf(char *filename,int size); //cf command
void help(); //help information
```

## 3.接口与调用关系设计

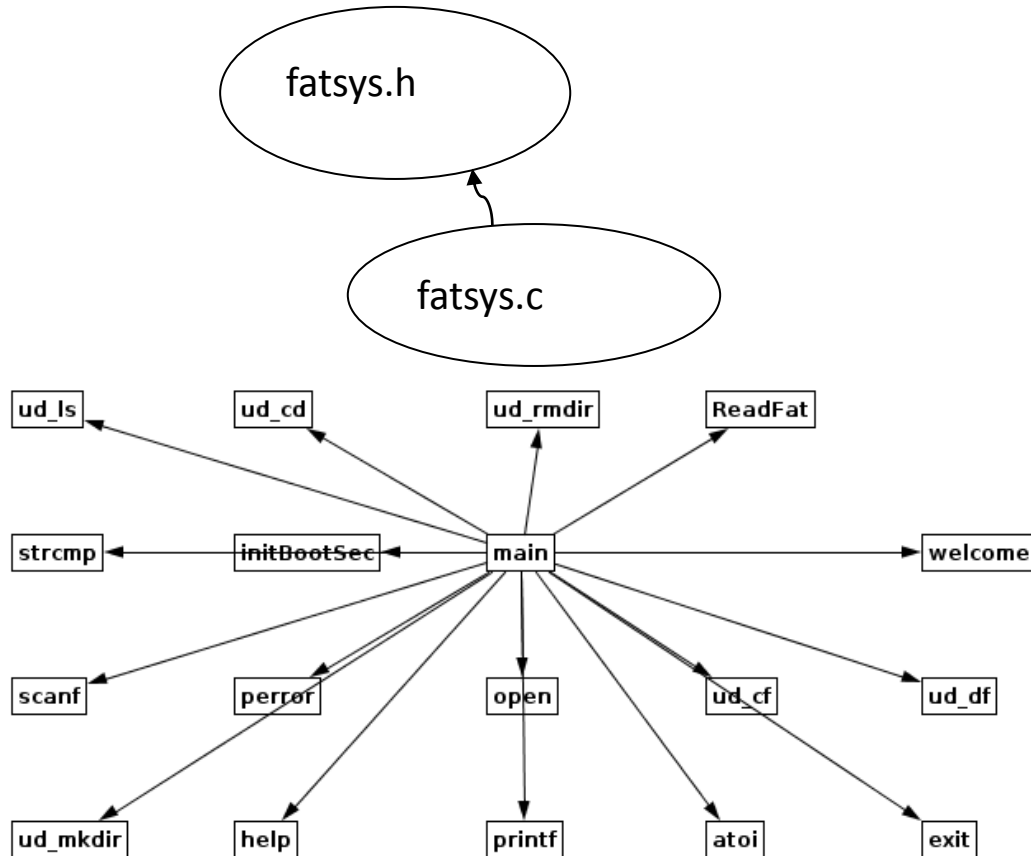


图2.3 接口与调用关系图

#### 4. 核心逻辑或功能代码

请见 3 实验内容中的 3.1 实现方法以及详细注释。

## 3 实验内容

详细说明实验的过程，实现方法及遇到的问题

### 3.1 实现方法

#### ➤ ram1 挂载

请见 ram1mount.sh

```
sudo dd if=/dev/zero of=/dev/ram1 bs=1M count=1
```

```
sudo mkfs.vfat /dev/ram1
```

#### ➤ 启动记录

初始化时扫描 U 盘基本信息，使用 void initBootSec()函数实现。

Byte Offset(in hex)	Field Length	Sample Value	Meaning
0x00	3	EB 3C 90	跳转指令
0x03	8	MSDOS5.0	DOS 版本信息





0x0b	2	512	每个扇区数
0x0d	1	64	每簇扇区数
0x0e	2	2	保留扇区数
0x10	1	2	文件分配表数目
0x11	2	512	根目录个数
0x13	2	0	逻辑扇区总数
0x15	1	248	介质描述符
0x16	2	239	每个文件分配表所占扇区数
0x18	2	63	每柱面扇区数目
0x1a	2	255	磁盘每个盘面的磁头数
0x1c	4	32	隐藏扇区数目
0x24	26		扩展 BIOS 参数区
0x3E	448		启动代码
1FE	2	0x55AA	扇区结束标志

➤ **ud\_ls 函数的实现**

该函数要求列出当前目录下的文件信息。如果是根目录，那么可以从根目录区获得目录项的信息。如果不是根目录，还需要根据当前目录 **curdir** 中保存的起始簇号 **FirstCluster** 计算出存储在数据区的扇区的地址，然后读该地址后面的内容，就是当前目录下的文件信息，将它们以用户理解的方式打印出来。

➤ **ud\_cd 函数的实现**

该函数要求实现切换到当前目录至上一层目录或子目录或当前目录。函数的参数是目录的名称，这里可以是全路径名，也可以是切换多级目录。

首先判断一些特殊情况，如果当前目录是根目录且要求切换到上层目录，则直接返回；如果切换到当前目录，则什么也不用做，直接放回。然后在当前目录下查找与参数名相吻合的子目录项，如果找到，那么释放原来的 **curdir**，使它指向这个子目录。

这里特别加入对多级目录的解析，实现代码如下：

```
nextdir = dir;
while( (curcurdir = strsep(&nextdir, "/"))
{
    pentry = (struct Entry*)malloc(sizeof(struct Entry));
    ret = ScanEntry(curcurdir, pentry, 1);
    if(ret < 0)
    {
        printf("no such dir\n");
        free(pentry);
        return -1;
    }
    dirno ++;
    fatherdir[dirno] = curcurdir;
```



```
    curdir = pentry;  
}
```

#### ➤ ud\_df

该函数删除当前目录下的文件。首先查找当前目录下是否存在删除的文件。如果存在，需要把文件对应的目录项内容删除，同时在两张 **fat** 表中，依次找到该文件占用的簇号，把簇号对应的 **fat** 表项的内容标为未使用，达到删除文件的目的。开始对 **fat** 表的修改在内存中进行，最后把整个 **fat** 表写回 U 盘（磁盘），减少了 I/O 次数，提高速度。

#### ➤ ud\_cf

该函数在当前目录下创建文件。首先查找当前目录下是否存在同名文件。如果存在，则显示相应信息，并提示创建失败。否则，依次遍历 **fat** 表，找到空白簇，然后将簇信息依次写入，之后在当前目录下为创建的文件建立目录项。将这些信息写回 U 盘，文件创建成功。

#### ➤ ud\_mkdir

该函数在当前目录下创建子目录。首先查找当前目录下是否存在同名子目录。如果存在，则显示相应信息，并提示创建失败。否则，依次遍历 **fat** 表，找到空白簇，然后将簇信息依次写入，之后在当前目录下为创建的子目录建立目录项。将这些信息写回 U 盘，子目录创建成功。

#### ➤ ud\_rmdir

该函数删除当前目录下的子目录。首先查找当前目录下是否存在删除的子目录。如果存在，需要把子目录对应的目录项内容删除，同时在两张 **fat** 表中，依次找到该子目录占用的簇号，把簇号对应的 **fat** 表项的内容标为未使用，达到删除子目录的目的。开始对 **fat** 表的修改在内存中进行，最后把整个 **fat** 表写回 U 盘（磁盘），减少了 I/O 次数，提高速度。

## 3.2 测试和使用说明

### 3.2.1 程序开发环境：

gcc、gdb

### 3.2.2 运行环境

ubuntu 9.04, linux 内核 2.6 以上版本

### 3.2.3 安装说明

1. 打开源码所在文件夹，以 root 权限 make 编译。
2. 以 root 权限运行 ./fatsys。

### 3.2.4 测试用例和运行结果分析:

#### ➤ 输入描述

运行./fatsys 后，输入 ls,cf,mkdir,df,rmdir 等命令综合测试。

#### ➤ 输出描述

在下面的截图中有显示。

#### ➤ 运行结果

## *fatsys* 程序

- 1 makefile 编译程序：注意中间有命令需要最高 root 权限（root）。

```
james@james-desktop:~/Desktop/fatsys$ make
sudo sh ram1mount.sh
1+0 records in
1+0 records out
1048576 bytes (1.0 MB) copied, 0.00817203 s, 128 MB/s
mkfs.vfat 3.0.7 (24 Dec 2009)
unable to get drive geometry, using default 255/63
gcc -Wall -O2 -o fatsys fatsys.c
fatsys.c: In function 'main':
fatsys.c:942: warning: unused variable 'ret'
fatsys.c:939: warning: unused variable 'pentry'
fatsys.c:973: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
fatsys.c:981: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
fatsys.c:986: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
fatsys.c:991: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
fatsys.c:992: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
fatsys.c:998: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
fatsys.c:1003: warning: ignoring return value of 'scanf', declared with attribute warn_unused_result
james@james-desktop:~/Desktop/fatsys$
```

图3.1 makefile编译截图

- 2 sudo ./fatsys 运行程序，读出模拟 U 盘（/dev/ram1）启动扇区信息，输出 help 信息。

```
james@james-desktop:~/Desktop/fatsys$ sudo ./fatsys

////////////////////////////////////
Welcome to File System version 1.0
Author:39061416 Huang Jianyu
Website:www.huangjy.info
Mobile Phone:15210965935
Email:hjyahead@gmail.com
////////////////////////////////////

-----
|      Oem name      |      mkdosfs |
| BytesPerSector    |      512     |
| SectorsPerCluster |      4       |
| ReservedSector    |      1       |
| FATS              |      2       |
| RootDirEntries    |     512      |
| LogicSectors      |      0       |
| MedioType         |     248      |
| SectorPerFAT      |     128      |
| SectorPerTrack    |      63      |
| Heads            |     255      |
| HiddenSectors     |      0       |
|-----|-----|

please input a command, including followings:
ls                list all files
cd <dir>          change direcotry
cf <filename> <size> create a file
df <file>         delete a file
mkdir <dirname>   create a directory
rmdir <dir>       delete a directory
exit             exit this system
> 
```

图3. 2 fatsys程序运行开始截图

### 3 输入 ls 命令测试



```
Author:39061416 Huang Jianyu
Website:www.huangjy.info
Mobile Phone:15210965935
Email:hjyahead@gmail.com
////////////////////////////////////

-----
| Oem_name          | mkdosfs |
| BytesPerSector    | 512     |
| SectorsPerCluster | 4       |
| ReservedSector    | 1       |
| FATs              | 2       |
| RootDirEntries    | 512     |
| LogicSectors      | 0       |
| MedioType         | 248     |
| SectorPerFAT      | 128     |
| SectorPerTrack    | 63      |
| Heads             | 255     |
| HiddenSectors     | 0       |
|-----|-----|

please input a command, including followings:
ls          list all files
cd <dir>    change direcotry
cf <filename> <size> create a file
df <file>    delete a file
mkdir <dirname> create a directory
rmdir <dir>  delete a directory
exit        exit this system

>ls
Root_dir
-----
| NAME      | DATE      | TIME      | CLUSTER | SIZE | ATTR |
|-----|-----|-----|-----|-----|-----|
>
```

图3.3 输入ls运行截图

4 输入 cf test.txt 1500

```
>ls
Root_dir
-----
| NAME      | DATE      | TIME      | CLUSTER | SIZE | ATTR |
|-----|-----|-----|-----|-----|-----|
>cf test.txt 1500
>ls
Root_dir
-----
| NAME      | DATE      | TIME      | CLUSTER | SIZE | ATTR |
|-----|-----|-----|-----|-----|-----|
| TEST.TXT  | 2012/5 /24 | 20:24:14 | 2       | 1500 | file |
|-----|-----|-----|-----|-----|-----|
>
```

图3.4 输入cf test.txt 1500运行截图

5 输入 mkdir testdir

```
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
>cf test.txt 1500
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| TEST.TXT      | 2012/5 /24    | 20:24:14     | 2        | 1500 | file |
-----
>mkdir testdir
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| TEST.TXT      | 2012/5 /24    | 20:24:14     | 2        | 1500 | file |
| TESTDIR      | 2012/5 /24    | 20:24:22     | 3        | 0    | dir  |
-----
>
```

图3.5 输入mkdir testdir运行截图

## 6 输入 cd testdir 命令

```
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
>cf test.txt 1500
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| TEST.TXT      | 2012/5 /24    | 20:26:14     | 2        | 1500 | file |
-----
>mkdir testdir
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| TEST.TXT      | 2012/5 /24    | 20:26:14     | 2        | 1500 | file |
| TESTDIR      | 2012/5 /24    | 20:26:18     | 3        | 0    | dir  |
-----
>cd testdir
>ls
TESTDIR_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| .             | 2012/5 /24    | 20:26:18     | 4        | 0    | dir  |
| ..            | 2012/5 /24    | 20:26:18     | 5        | 0    | dir  |
-----
>
```

图3.6 输入cd testdir运行截图

## 7 输入 cd .测试



```
>cd testdir
>ls
TESTDIR_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| .             | 2012/5 /24 | 20:26:18   | 4       | 0    | dir  |
| ..            | 2012/5 /24 | 20:26:18   | 5       | 0    | dir  |
-----

>cd .
>ls
TESTDIR_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| .             | 2012/5 /24 | 20:26:18   | 4       | 0    | dir  |
| ..            | 2012/5 /24 | 20:26:18   | 5       | 0    | dir  |
-----

>
```

图3.7输入cd .测试

## 8 输入 cd ..测试

```
>cd testdir
>ls
TESTDIR_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| .             | 2012/5 /24 | 20:26:18   | 4       | 0    | dir  |
| ..            | 2012/5 /24 | 20:26:18   | 5       | 0    | dir  |
-----

>cd .
>ls
TESTDIR_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| .             | 2012/5 /24 | 20:26:18   | 4       | 0    | dir  |
| ..            | 2012/5 /24 | 20:26:18   | 5       | 0    | dir  |
-----

>cd ..
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| TEST.TXT      | 2012/5 /24 | 20:26:14   | 2       | 1500 | file  |
| TESTDIR       | 2012/5 /24 | 20:26:18   | 3       | 0    | dir   |
-----

>
```

图3.8 输入cd ..测试

## 9 输入 cf test.txt 120（文件名重复）测试

```
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| TEST.TXT      | 2012/5 /24 | 20:26:14   | 2       | 1500 | file  |
| TESTDIR       | 2012/5 /24 | 20:26:18   | 3       | 0    | dir   |
-----

>cf test.txt 120
This filename is exist
```



图3.9 输入cf test.txt 120（文件名重复）测试

10 输入 df test.txt 测试

```
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| TEST.TXT      | 2012/5 /24 | 20:26:14   | 2       | 1500 | file |
| TESTDIR      | 2012/5 /24 | 20:26:18   | 3       | 0    | dir  |
-----
>cf test.txt 120
This filename is exist
>df test.txt
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| TESTDIR      | 2012/5 /24 | 20:26:18   | 3       | 0    | dir  |
-----
>
```

图3.10 输入df test.txt测试

11 rmdir testdir 测试

```
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| TEST.TXT      | 2012/5 /24 | 20:26:14   | 2       | 1500 | file |
| TESTDIR      | 2012/5 /24 | 20:26:18   | 3       | 0    | dir  |
-----
>cf test.txt 120
This filename is exist
>df test.txt
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
| TESTDIR      | 2012/5 /24 | 20:26:18   | 3       | 0    | dir  |
-----
>rmdir testdir
>ls
Root_dir
-----
| NAME          | DATE       | TIME       | CLUSTER | SIZE | ATTR |
-----
>
```

图3.11 输入rmdir testdir测试

12 绝对路径和多级目录测试，能实现逐级查找目录。





```
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
>mkdir parentdir
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| PARENTDIR     | 2012/5 /24    | 20:28:26     | 2        | 0    | dir  |
-----
>cd parentdir
>mkdir childdir
>ls
PARENTDIR_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| .             | 2012/5 /24    | 20:28:26     | 3        | 0    | dir  |
| ..            | 2012/5 /24    | 20:28:26     | 6        | 0    | dir  |
| CHILDDIR      | 2012/5 /24    | 20:29:8      | 7        | 0    | dir  |
-----
>cd ..
>cd ..
>ls
Root_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
| PARENTDIR     | 2012/5 /24    | 20:28:26     | 2        | 0    | dir  |
-----
>cd parentdir/childdir
>ls
CHILDDIR_dir
-----
| NAME          | DATE          | TIME          | CLUSTER | SIZE | ATTR |
-----
>
```

图3. 12 输入多级目录路径测试

## ***fatsys2*** 程序

- 1 输入 `sudo./filesys /dev/ram1`

```
james@james-desktop:~/Desktop/fatsys2$ sudo ./fileys /dev/ram1

////////////////////////
Welcome to File System version 1.0
Author:39061416 Huang Jianyu
Website:www.huangjy.info
Mobile Phone:15210965935
Email:hjyahead@gmail.com
////////////////////////

=====Boot sector=====
oem_name: mkdosfs
byts_per_sec: 512
sec_per_clus: 4
resvd_sec_cnt: 1
=====
count_of_clusters = 32695
fat_type:2
>sec fat:1
sec_root_dir:257
sec_first_data:289
file system mount OK.
/:>
```

图3.13 输入sudo./fileys /dev/ram1

## 2 输入 ls

```
=====
count_of_clusters = 32695
fat_type:2
>sec fat:1
sec_root_dir:257
sec_first_data:289
file system mount OK.
/:>ls

Total 0 files.
/:>cf test.txt
huangjianyu
hello
test

=====
wrote 23 bytes to file test.txt ok.
/:>ls

    23 byte    2012/05/24 - 23:19  test.txt

Total 1 files.
/:>
```

图3.14 输入ls

3 输入 cf test.txt，然后在后面输入文件内容。最后按下：Ctrl+D 表示文件结束。输入 ls，显示 test.txt 在目录中。再输入 mkdir testdir 创建目录，输入 ls，显示多了 testdir 目录。



```
sec_root_dir:257
sec_first_data:289
file system mount OK.
/:>ls

Total 0 files.
/:>cf test.txt
huangjianyu
hello
test

=====
wrote 23 bytes to file test.txt ok.
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt

Total 1 files.
/:>mkdir testdir
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
    0 byte   2012/05/24 - 23:20  testdir

Total 2 files.
/:>
```

图3.15 输入cf test.txt,再输入具体内容

## 4 输入 cd testdir

```
/:>cf test.txt
huangjianyu
hello
test

=====
wrote 23 bytes to file test.txt ok.
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt

Total 1 files.
/:>mkdir testdir
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
    0 byte   2012/05/24 - 23:20  testdir

Total 2 files.
/:>cd testdir
/testdir/:>ls
    0 byte   2012/05/24 - 23:20  .
    0 byte   2012/05/24 - 23:20  ..

Total 2 files.
/testdir/:>
```

图3.16 输入cd testdir

## 5 输入 cd .与 cd ..测试

```
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
Total 1 files.
/:>mkdir testdir
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
    0 byte   2012/05/24 - 23:20  testdir
Total 2 files.
/:>cd testdir
/testdir/>ls
    0 byte   2012/05/24 - 23:20  .
    0 byte   2012/05/24 - 23:20  ..
Total 2 files.
/testdir/>cd .
/testdir/>cd ..
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
    0 byte   2012/05/24 - 23:20  testdir
Total 2 files.
/:>
```

图3.17 输入cd .与cd ..

#### 6 输入 rm test.txt 测试

```
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
    0 byte   2012/05/24 - 23:20  testdir
Total 2 files.
/:>cd testdir
/testdir/>ls
    0 byte   2012/05/24 - 23:20  .
    0 byte   2012/05/24 - 23:20  ..
Total 2 files.
/testdir/>cd .
/testdir/>cd ..
/:>ls
    23 byte   2012/05/24 - 23:19  test.txt
    0 byte   2012/05/24 - 23:20  testdir
Total 2 files.
/:>rm test.txt
/:>ls
    0 byte   2012/05/24 - 23:20  testdir
Total 1 files.
/:>
```

图3.18 输入rm test.txt

#### ➤ 结果分析



1. 从**错误!未找到引用源**。到图 3. 12 可以看出，本程序实现了基本要求：实现 ls、cd、df、cf 等相关函数与命令。
2. 从图 3. 5、图 3. 11 可以看出，本程序实现扩展要求：删除目录的功能。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。
3. 从图 3. 12 可以看出，本实验程序实现扩展要求：增加对绝对路径和多级目录的支持。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。
4. 从图 3. 15 可以看出，本实验程序实现对 ud\_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。

## 4 每个人的工作与会议记录

每次实验至少要开 4 次小组会议,每次会议都要记录以下内容:

- a 说明每一位组员前一阶段完成的具体工作,是否按时按量完成任务。
- b 说明每一位组员下一阶段需要完成的工作。
- c 其他会议内容(如讨论的问题以及解决方案等)

### 4.1 会议时间表与会议记录

2012 年 5 月 5 日	文件系统设计	黄建宇
2012 年 5 月 10 日	文件系统文档书写	黄建宇
2012 年 5 月 12 日	文件系统文档审查, 修改	黄建宇
2012 年 5 月 18-20 日	文件系统程序编写	黄建宇
2012 年 5 月 21-24 日	调试, debug, 完善文档	黄建宇

每次会议都基本完成上次会议所计划内容，并规划下次完成进度。

## 5 其他说明

### 5.1 组内成员任务分工说明

(注明各个组员的工作量比例)

由于本组只有 1 人，工作量 100%。

### 5.2 实验完成部分说明

基本都已完成。

实现扩展要求：

- 增加删除目录的功能。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。（图 3.5、图 3.11）

- 对 `ud_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。（图 3.15）
- 增加对绝对路径和多级目录的支持。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。（图 3.12）
- 对 `ud_ls()` 函数进行完善，增加对全部非根目录信息的读取（本试验中只读取了一个扇区的非根目录信息）。

## 6 程序清单

<code>makefile</code>	makefile 编译文件
<code>fatsys.h</code>	全局数据结构保存
<code>fatsys.c</code>	文件系统实现主程序
<code>ram1mount.sh</code>	挂载/dev/ram1

## 7 实验心得

### 7.1 心得

我感受到理论与实践之间的差距，在课堂上我感觉作业调度很简单，但到了 OS 课程设计实际模拟时，却遇到一个个困难。通过解决这一个个困难，我感觉我对课堂上的理论部分有了更深入的理解。

特别感谢王雷老师精彩的授课，感谢助教辛苦的讲解。最终我高效率高质量的完成的实验的要求。

### 7.2 建议

建议老师多给一些相关的资料，比如初步介绍一下本实验要用到的相关函数。我看了《Unix 高级环境编程》才了解时间函数的相关知识。