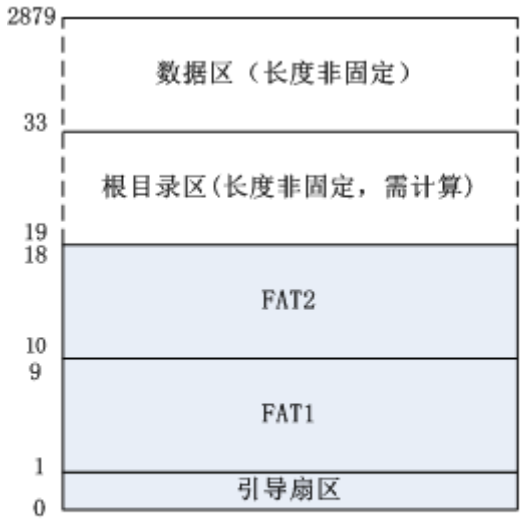
操作系统实验二

> 作者：rubisco
>
> 更新时间：2021-11-3

日期	完成情况	后续任务
2021-10-31	完成了对img文件引导扇区、FAT、根目录区的分析	实现数据区读取
2021-11-2	实现了对数据区的读取	发现问题：存在文件名奇怪的文件，不知道是否应该去掉
2021-11-3	完成了用C++的实现	用nasm替换输出函数

认识FAT12文件系统的镜像

每一个1.44M的软盘镜像，都是由2880个512B的扇区组成的，且可以分成5个部分。



软盘(1.44MB,FAT12)

其中，第1个扇区是引导扇区，1-9扇区是FAT1、10-18扇区是FAT2；FAT1和FAT2互为备份。

引导扇区

操作系统标志FAT12文件系统的方法。下表为每个位置的含义。可供对照自己的读取是否正确。

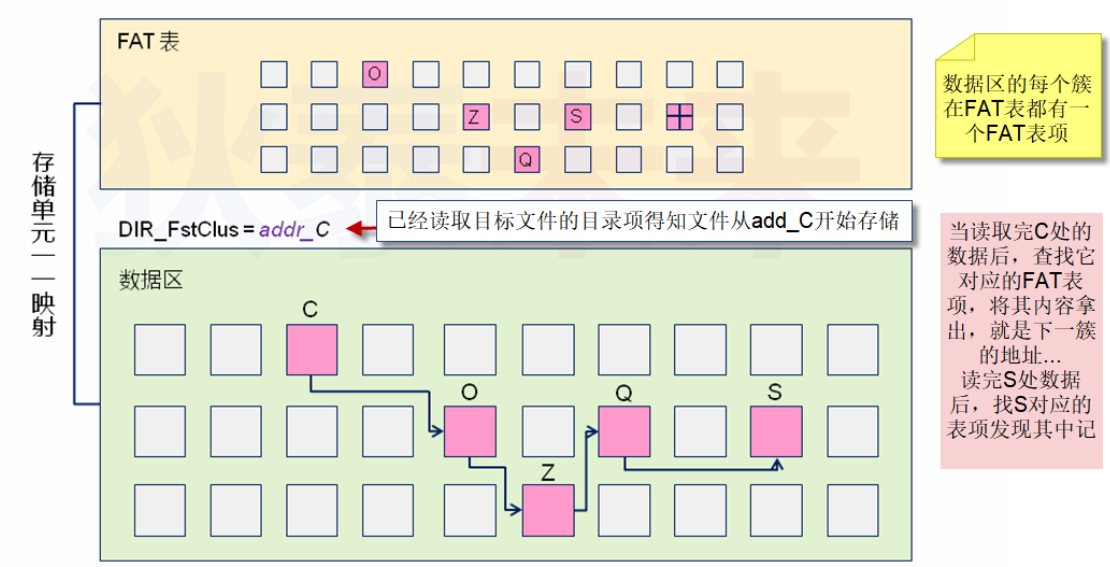
名称	开始字节	长度	内容	参考值
BS_jmpBOOT	0	3	一个短跳转指令	jmp short LABEL_STARTnop
BS_OEMName	3	8	厂商名	'ZGH'
BPB_BytesPerSec	11	2	每扇区字节数（Bytes/Sector）	0x200
BPB_SecPerClus	13	1	每簇扇区数（Sector/Cluster）	0x1
BPB_ResvdSecCnt	14	2	Boot记录占用多少扇区	ox1
BPB_NumFATs	16	1	共有多少FAT表	0x2
BPB_RootEntCnt	17	2	根目录区文件最大数	0xE0
BPB_TotSec16	19	2	扇区总数	0xB40
BPB_Media	21	1	介质描述符	0xF0
BPB_FATSz16	22	2	每个FAT表所占扇区数	0x9
BPB_SecPerTrk	24	2	每磁道扇区数（Sector/track）	0x12
BPB_NumHeads	26	2	磁头数（面数）	0x2
BPB_HiddSec	28	4	隐藏扇区数	0
BPB_TotSec32	32	4	如果BPB_TotSec16=0,则由这里给出扇区数	0
BS_DrvNum	36	1	INT 13H的驱动器号	0
BS_Reserved1	37	1	保留，未使用	0
BS_BootSig	38	1	扩展引导标记(29h)	0x29
BS_VolID	39	4	卷序列号	0
BS_VolLab	43	11	卷标	'ZGH'
BS_FileSysType	54	8	文件系统类型	'FAT12'
引导代码及其他内容	62	448	引导代码及其他数据	引导代码（剩余空间用0填充）
结束标志0xAA55	510	2	第510字节为0x55，第511字节为0xAA	0xAA55

FAT1、FAT2

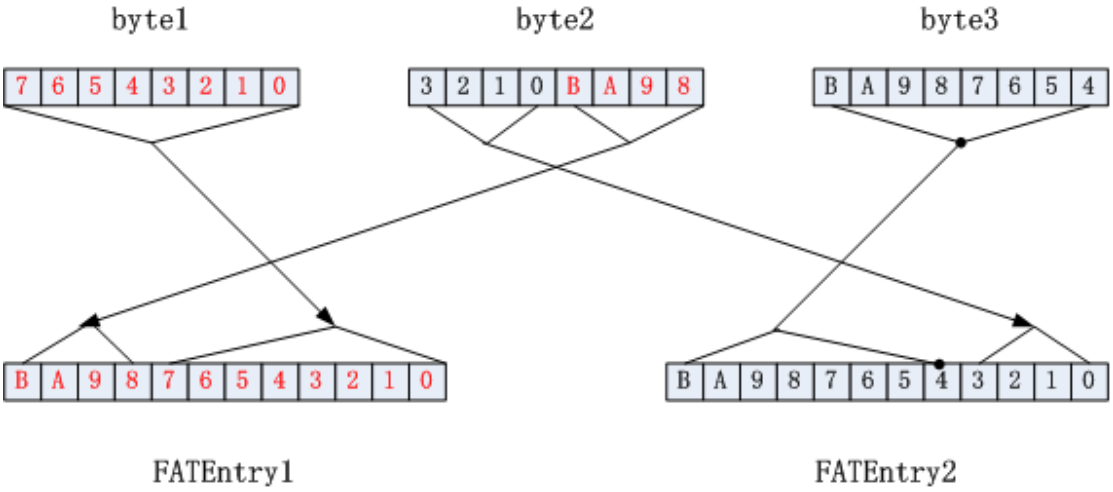
FAT1和FAT2互为备份。在分析时只要分析一个即可。

FAT表每12位（bit）称为一个FAT项，对应着数据区的一个**簇**，FAT表和数据区的簇一一对应，起到了两个作用：

1. 标志着目录/文件在数据区的位置
2. FAT项本身的值指向了该文件剩余部分所在的FAT项。（本质上行为像一个链表）



（选做加分项）如何从12位中获取需要的值?



从FAT表中得到的一个FAT项的值

- 大端党人视角：~~注意到FAT项的12位并不是按顺序存储的。而是经过了换序~~
- 小端党人视角：将上图byte1、byte3位置互换，即可读出正确结果

根目录

根目录区的大小不确定，它是由计算得出的。计算所需要的数已经定义在引导扇区中。公式如下:记扇区数为num，则有：

```
int size = (int)((unsigned char)(head.BPB_RootEntCnt[0])+(unsigned char)
(head.BPB_RootEntCnt[1])*256); //根目录最大文件数
```

每个文件占32字节的表项，故有

```
int num = (int)(32*size+511)/512;
```

文件表项信息

如图所示，每一个文件信息项由

- 11位的name（8位名+3位拓展名）
  - 更长的文件名不用考虑
- **1位的文件属性**
  - 16表示目录
  - 32表示文件
  - 此外还有表示系统的、隐藏的、只读的等，没有涉及。不做考虑
- 10位保留位
- 更改时间、日期各两位
- **两位的簇号，对应着这个文件存储的FAT表项号**
  - 小端存储（簇号 = 【0】+【1】*256）
- 文件大小

数据区处理

每一个数据扇区都和一个FAT表项一一对应。根据根目录区的信息，在数据区递归读取即可。

NASM输出

NASM输出函数

因为要实现红色和非红色两个版本，我们需要实现两个函数。

设置颜色

> 参考链接

设置ecx = 颜色代码，edx = 颜色代码长度。输出即可完成颜色设置。

数值	颜色
30	黑
31	红
32	绿
33	黄
34	蓝色
35	紫色
36	深绿
37	白色

数值	样式
0m	关闭所有属性
1m	高亮
4m	下划线
5m	闪烁
7m	反显
8m	消隐

参数获取

在32位的系统中，每4个字节代表一个寻址单元。

ESP用于指向栈的栈顶（下一个压入栈的活动记录的顶部），而栈由高地址向低地址成长，函数调用是用入栈的方式传递参数，故在函数处理参数时，ESP+4就是最后一个入栈的参数的地址，ESP+8就是再前一个参数的地址。

即参数1是【esp+4】,参数2是【esp+8】。以此类推

nasm函数链接到C

在C++中使用关键字`extern`表明这个函数是在另一个文件定义的。

```
extern "C"{
    void myPrint_asm(const char* str,int len);
    void myPrintRed_asm(const char* str,int len);
}
```

在nsam中使用`GLOBAL`关键字声明这是一个全局函数。

```
section .text
    GLOBAL myPrint_asm
    GLOBAL myPrintRed_asm
```

makefile入门

本质上就是一个shell脚本，指定了make指令发生时，shell进行的操作

```
build: myPrint.o
    @echo "going to compile main.cpp"
    @g++ -std=c++11 -o fat main.cpp myPrint.o
    @echo "finish compile main.cpp"

myPrint.o: myPrint.asm
    @echo "going to compile myPrint.asm"
    @nasm -f elf -o myPrint.o myPrint.asm
    @echo "going to compile myPrint.asm"

run: build
    @./fat

clean:
    @rm -f *.o
    @rm -f fat
```

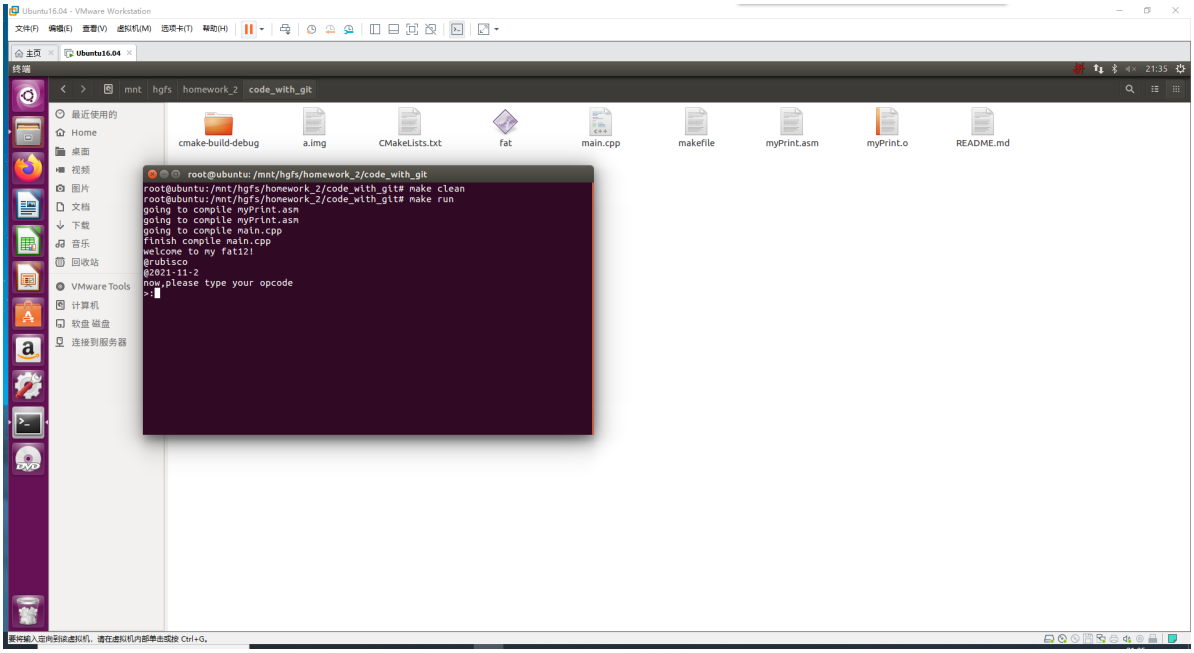
其中，冒号前面的是行为，后面的是依赖。如果在运行某一个行为时缺少依赖，就会先创建依赖的行为。

操作

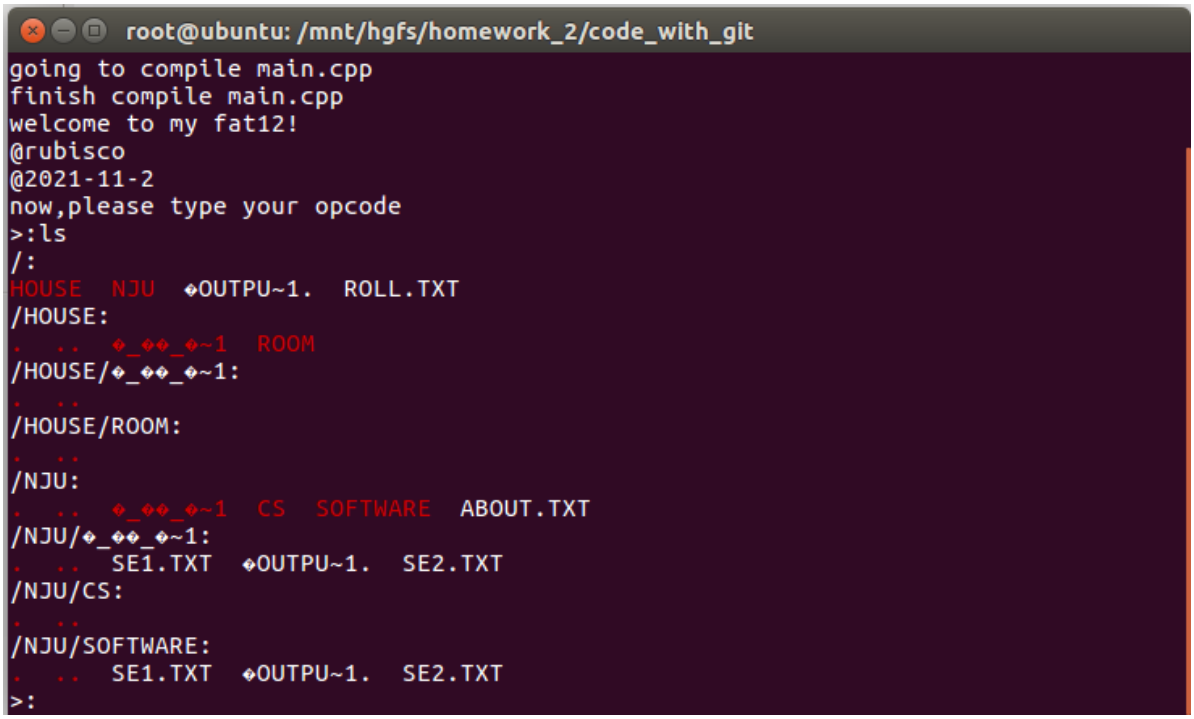
- 编译：make build
- 运行：make run
- 删除：make clean

运行截图

使用makefile


root@ubuntu: /mnt/hgfs/homework_2/code_with_git
root@ubuntu:/mnt/hgfs/homework_2/code_with_git# make clean
root@ubuntu:/mnt/hgfs/homework_2/code_with_git# make run
going to compile myPrint.asm
going to compile myPrint.asm
going to compile main.cpp
finish compile main.cpp
welcome to my fat12!
@rubisco
@2021-11-2
now,please type your opcode
>:
cmake-build-debug
a.img
CMakeLists.txt
fat
main.cpp
makefile
myPrint.asm
myPrint.o
README.md

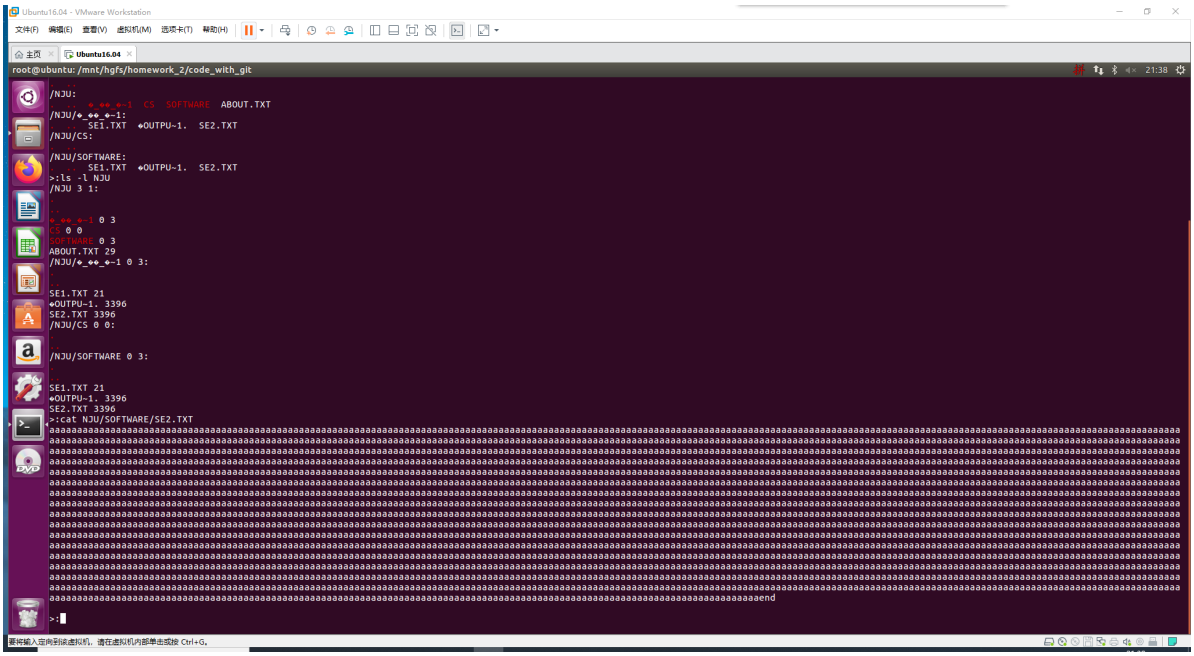
ls


root@ubuntu: /mnt/hgfs/homework_2/code_with_git
going to compile main.cpp
finish compile main.cpp
welcome to my fat12!
@rubisco
@2021-11-2
now,please type your opcode
>:ls
/:
HOUSE NJU ◆OUTPU~1. ROLL.TXT
/HOUSE:
. .. ◆_◆◆_◆~1 ROOM
/HOUSE/◆_◆◆_◆~1:
. ..
/HOUSE/ROOM:
. ..
/NJU:
. .. ◆_◆◆_◆~1 CS SOFTWARE ABOUT.TXT
/NJU/◆_◆◆_◆~1:
. .. SE1.TXT ◆OUTPU~1. SE2.TXT
/NJU/CS:
. ..
/NJU/SOFTWARE:
. .. SE1.TXT ◆OUTPU~1. SE2.TXT
>:

ls -l NJU

```
root@ubuntu: /mnt/hgfs/homework_2/code_with_git
>:ls -l NJU
/NJU 3 1:
.
..
◆_◆◆_◆~1 0 3
CS 0 0
SOFTWARE 0 3
ABOUT.TXT 29
/NJU/◆_◆◆_◆~1 0 3:
.
..
SE1.TXT 21
◆OUTPU~1. 3396
SE2.TXT 3396
/NJU/CS 0 0:
.
..
/NJU/SOFTWARE 0 3:
.
..
SE1.TXT 21
◆OUTPU~1. 3396
SE2.TXT 3396
>:
```

cat 长文件


root@ubuntu: /mnt/hgfs/homework_2/code_with_git
>:cat NJU/SOFTWARE/SE2.TXT
aaaa

```
root@ubuntu: /mnt/hgfs/homework_2/code_with_git
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaend

>:exit
thank you for using! see you!
root@ubuntu:/mnt/hgfs/homework_2/code_with_git#
```

报错信息

```
root@ubuntu: /mnt/hgfs/homework_2/code_with_git
>:exit
thank you for using! see you!
root@ubuntu:/mnt/hgfs/homework_2/code_with_git# make run
going to compile main.cpp
finish compile main.cpp
welcome to my fat12!
@rubisco
@2021-11-2
now,please type your opcode
>:lsb
there is no such opcode!please type the right one.
>:ls -l NJU NJU
you have typed path more than once
>:ls -l -- NJU
you have typed wrong parameter
>:ls NJ
there is no such dir:NJ    please check your path
>:cat RO.TXT
there is no such file! please check you path
>:cat NJ/TR.TXT
there is no such dir:NJ    please check your path
>:exit  1298631
there is no such opcode!please type the right one.
>:
```

源代码

👉我的GitLab：http://49.235.227.136:8099/