# 操作系统课程设计报告

**实验五：简单文件系统的实现**

**专业： 理工类实验班**

**班级： 19185312**

**姓名： 郑凯心**

**学号： 19063140**

**指导老师： 任彧**

**上机时间：周五下午6、7、8**

**杭州电子科技大学 卓越学院**

**2021.10**

## 一、实验目的

通过具体的文件存储空间的管理、文件物理结构、目录结构和文件操作的实现，加深对文件系统内部数据结构、功能以及实现过程的理解。

### 二、实验内容

（1）在内存中开辟一个虚拟磁盘空间作为文件存储分区，在其上实现一个简单的基于多级目录的单用户单任务系统中的文件系统。在退出该文件系统的使用时，应将虚拟磁盘上的内容以一个文件的方式保存到磁盘上，以便下次可以再将它恢复到内存的虚拟磁盘中。

（2）文件物理结构可采用显式链接或其他结构。

（3）空闲磁盘空间的管理可选择 FAT 表、位示图或其他办法。

（4）文件目录结构采用多级目录结构。为简单起见，可以不使用索引结点，每个目录项应包含文件名、物理地址、长度等信息，还可以通过目录项实现对文件的读和写的保护。

（5）要求提供以下操作命令：

my\_format：对文件存储器进行格式化，即按照文件系统的结构对虚拟磁盘空间进行布局，并在其上创建根目录以及用于管理文件存储空间等的数据结构。

my\_mkdir：用于创建子目录。

my\_rmdir：用于删除子目录。

my\_ls：用于显示目录中的内容。

my\_cd：用于更改当前目录。

my\_create：用于创建文件。

my\_open：用于打开文件。

my\_close：用于关闭文件。1

my\_write：用于写文件。

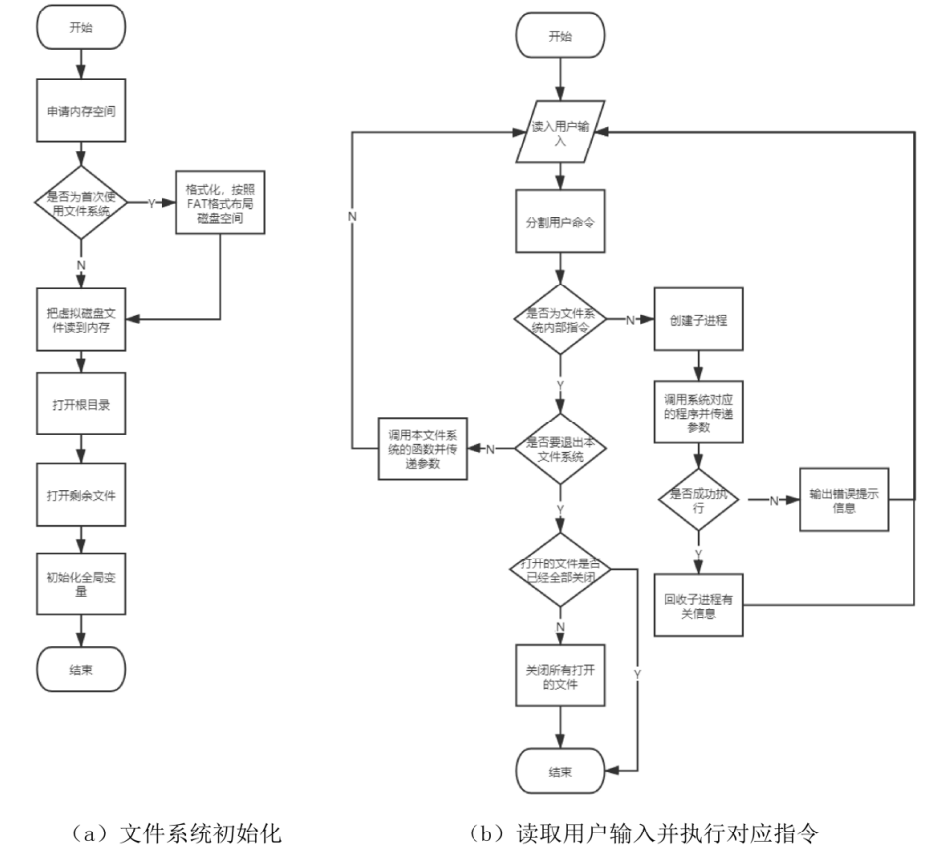
my\_read：用于读文件。

my\_rm：用于删除文件。

 my\_exitsys：用于退出文件系统。

### 三、实验方法

实验流程图



###### 文件系统初始化函数

如流程图 (a）所示，系统初始化函数start\_sys(void)

(1) 对应命令:无

(2) 命令调用格式:无

(3) 函数设计格式: int start\_sys(void)

(4) 功能:系统初始化主函数

(5) 输入:无

(6) 输出:无

(7) 函数需完成的工作

①申请虚拟磁盘内存空间

②打开fsfile文件:若文件存在，则跳转

③;若文件不存在，则创建之，跳转⑤③读入fsfile文件内容到缓冲区中，跳转④

④关闭myfsys文件;跳转⑥

⑤输出"文件系统不存在"提示信息，并调用my\_format()对虚拟磁盘空间进行格式化操作:跳转⑥

⑥打开根日录，将当前日录设置为根日录，并填写根日录文件的用户打开文件表项和FCB

⑦类似地打开剩余文件并填写对应用户打开文件表项和FCB;⑧初始化全局变量

###### 更改当前目录函数以及实际更改当前目录函数

更改当前目录函数my\_cd()

(1) 对应命令:cd

(2) 命令调用格式: cd dirname

(3) 函致设计格式: void my\_ed(char \*dirname)

(4) 功能:改变当前目录到指定的名为dirname 的目录

(5) 输入:dirname:新的当前目录的目录名

(6) 输出:无

(7) 函数需完成的工作:

①检查dirname参数是否缺少并将dirname转换为绝对路径

②检查新的当前目录名是否存在，如果存在则转③，否则返回，并显示出错信息

③检查当前目录是否打开，如果新的当前目录文件没有打开，则打开该目录文件;并将ptrcurdir指向该打开文件表项;调用实际更改当前目录函数do\_chdir()函数设置当前目录为该目录

###### 创建子目录函数以及实际创建子目录函数

创建子目录函致my\_mkdir()

(1) 对应命令: mkdir

(2) 命令调用格式: mkdir dirname

(3) 函数设计格式: void my\_mkdir(char \*dirname)

(4) 功能:在当前目录下创建名为dirname 的子目录

(5) 输入:dirname:新建目录的目录名

(6) 输出:无

(7) 函数需完成的工作:

①my\_mkdir()主要负责检查创建子目录命令参数是否正确，同时检查当前目录下新建目录文件是否重名，若重名则返回，并显示错误信息;以下将详细说明实际创建子目录函数do\_mkdir()

②检查该目录下是否有空闲FCB，若没有则显示错误信息，否则继续

③检查FAT是否有空闲的盘块，如有则为新建目录文件分配一个盘块，否则返回并显示错误信息

④准备好新建目录文件的FCB的内容，文件的属性为目录文件在新建目录文件所分配到的磁盘块中建立两个特殊的目录项“."和”.."目录项

###### 删除子目录函数以及实际删除子目录函数

删除子目录函数rmdir()

(1) 对应命令:rmdir

(2) 命令调用格式: rmdir dirname

(3) 函数设计格式: void my\_rmdir(char \*dirname)

(4) 功能:在当前目录下删除名为dirname的子目录

(5) 输入 dirname:欲删除目录的目录名

(6) 输出:无

(7) 函数需完成的工作:

①my\_rmdir()主要负责检查删除子目录命令参数是否正确，同时检查当前目录下欲删除目录文件是否存在,若不存在则返回并显示错误信息;然后检查该目录文件是否已经打开，若打开则返回并提示用户文件已经打开，否则继续;并作其他类似检查（如是否为目录，是否为根目录等,处理方式类似,故不再赘述);以下将说明实际删除子目录函数do\_rmdir()

②将其所属的两个特殊的目录项“.”和“..”目录项都设为空;

③回收该目录文件所占据的磁盘块，并修改FAT;

###### 显示目录函数以及实际显示目录函数

显示目录函数my\_ls0

(1) 对应命令:ls

(2) 命令调用格式: ls [-1]

(3) 函数设计格式: void my\_ls(void)

(4) 功能:显示当前目录的内容并用颜色区分子目录和普通文件，若为-1模式则显示更加详细的信息

(5) 输入:无

(6) 输出:无

(7) 函数需完成的工作:

①my\_ls()主要负责检查显示目录命令参数是否正确,正确则继续，否则将返回并显示错误信息

②检查命令参数是否包含-1，若包含-1参数则将mode 设为-1模式，否则mode为默认模式

③将路径转换为绝对路径后调用do\_ls()，显示当前目录的内容并用颜色区分子目录和普通文件，若为-1模式则显示更加详细的信息

###### 创建文件函数以及实际创建文件函数

创建文件函数my\_create()

(1) 对应命令:create

(2) 命令调用格式: create filename

(3) 函数设计格式: int my\_create(char \*filename)

(4) 功能:创建名为filename的新文件或者创建归属下级目录的filename文件

(5) 输入: filename:新建文件的文件名，可能包含路径。

(6) 输出:若创建成功，返回该文件的文件描述符（文件打开表中的数组下标)﹔否则返回-1

(7) 函数需完成的工作:

①my\_create()主要负责检查创建文件命令参数是否正确,同时检查当前目录下新建文件是否重名，若重名则返回，并显示错误信息;类似地分割出父目录项并检查父目录是否存在，若存在则在该目录下新建文件，否则仍在当前目录下新建文件;以下将详细说明实际创建文件函数do\_create ()

②检查该目录下是否有空闲FCB，若没有则显示错误信息，否则继续

③检查FAT是否有空闲的盘块，如有则为新建文件分配一个盘块，否则返回并显示错误信息分割出文件名和后缀名，然后补充新建文件的FCB 的内容，文件的属性为文件

###### 删除文件函数以及实际删除文件函数

删除文件函数my\_rm()

(1) 对应命令: rm

(2) 命令调用格式: rm filename

(3) 函数设计格式: void my\_rm(char \*filename)

(4) 功能:删除名为filename的文件

(5) 输入: filename: 欲删除文件的文件名，可能还包含路径

(6) 输出:无

(7) 函数需完成的工作:

①my\_rm()主要负责检查删除文件命令参数是否正确，同时检查当前目录下欲删除文件是否存在，若不存在则返回并显示错误信息:然后检查该文件是否已经打开，若打开则返回并提示用户文件已经打开,否则继续:并作其他类似检查(如是否为文件等,处理方式类似，故不再赘述) ;以下将说明实际删除文件函数do\_rm(file)

②回收该文件所占据的磁盘块

③修改FAT

###### 退出文件系统函数

退出文件系统函数my\_exitsys()

(1) 对应命令: exit

(2) 命令调用格式: exit

(3) 函数设计格式: void my\_exitsys()

(4) 功能:退出文件系统

(5) 输入:无

(6) 输出:无

(7) 函数需完成的工作:

①关闭所有已经打开的文件

②将虛拟磁盘空间中的所有内容保存到磁盘上的fsfile 文件中

③关闭fsfile文件

④释放内存空间以及虛拟磁盘空间

###### 读取用户输入并执行对应指令函数

如流程图(b)所示，系统初始化函数cmd\_loop(void)

(1) 对应命令:无

(2) 命令调用格式:无

(3) 函数设计格式: void cmd\_loop(void)

(4) 功能:系统主函数

(5) 输入:用户指令

(6) 输出:无

(7) 函数需完成的工作:

①输出命令行提示符

②读入用户输入

③分割用户命令，分为指令和参数两个部分﹔若为文件系统内部指令跳转④，否则跳转⑧

④创建子进程并传递参数后调用系统对应的程序

⑤程序成功执行，跳转⑥;否则跳转⑦:

⑥回收子进程有关信息;跳转②

⑦输出错误提示信息，跳转②

⑧若命令为退出文件系统，跳转⑨:否则跳转⑩

⑨若有打开的文件则关闭所有打开的文件后退出，否则直接退出:传递参数后调用本文件系统的函数，完成后跳转②

###### 格式化函数以及实际格式化函数

磁盘格式化函数my\_format()

(1) 对应命令:format

(2) 命令调用格式: format

(3) 函数设计格式: void my\_format()

(4) 功能:对虚拟磁盘进行格式化，布局虚拟磁盘，建立根目录文件(或根目录区)

(5) 输入:无

(6) 输出:无

(7) 函数需完成的工作:

①my\_format()主要负责检查格式化命令参数是否正确，以下将说明实际格式化函数do\_format()

②写入文件系统的描述信息，如FAT表大小及位置、根目录大小及位置、盘块大小、盘块数量、数据区开始位置等信息，然后将虚拟磁盘第一个块作为引导块

③在引导块后建立两张完全一样的FAT表,用于记录文件所占据的磁盘块及管理虚拟磁盘块的分配，每个FAT占据两个磁盘块:对于每个FAT中，前面5个块设置为已分配，后面995个块设置为空闲

④在第二张FAT后创建根目录文件 root，将数据区的第1块分配给根目录文件，在该磁盘上创建两个特殊的目录项“.”和"..”

###### 打开文件函数以及实际打开文件函数

打开文件函数my\_open()

(1) 对应命令: open

(2) 命令调用格式: open [-1] filename

(3) 函数设计格式: int my\_open(char \*filename)

(4) 功能:打开当前目录下名为filename的文件或显示所以已经打开的文件

(5) 输入: filename: 欲打开文件的文件名

(6) 输出:若打开成功，返回该文件的描述符(在用户打开文件表中表项序号) ;否则返回-1

(7) 函数需完成的工作:

①检查打开文件命令参数是否正确:若命令参数是否包含-1，则输出所有已经打开文件的相关信息(如文件名字等)，否则继续

②检查该目录下欲打开文件是否存在，若不存在则返回，并显示错误信息

③检查该文件是否已经打开，若已打开则返回，并显示错误信息

④调用do\_open()，为该文件填写用户打开文件表表项内容，目录和普通文件作相同处理

###### 关闭文件函数

关闭文件函数my\_close()

(1) 对应命令: close

(2) 命令调用格式: close [-a] fd

(3) 函数设计格式: void my\_close(int fd)

(4) 功能:关闭前面由my\_open() 打开的文件描述符为fd的文件或者一键关闭所有己打开的文件

(5) 输入: fd:文件描述符

(6) 输出:无

(7) 函数需完成的工作:

①检查关闭文件命令参数是否正确:若命令参数是否包含-a，则关闭所有已经打开文件，否则继续;

②检查该目录下欲关闭文件是否存在，若不存在则返回，并显示错误信息;

③检查该文件是否已经打开，若已打开则关闭，否则忽略:

④调用do\_close()安全关闭文件(对没有修改的文件先执行保存) ;

###### 写文件函数

写文件函数my\_write()

(1) 对应命令: write

(2) 命令调用格式: write [-a]/[-r]/[-w] fd

(3) 函数设计格式: int my\_write(int fd)

(4) 功能:写文件前需要打开文件，否则不能完成写操作。将用户通过键盘输入的内容写到fd所指定的文件中写操作常有三种方式:截断写、覆盖写和追加写。截断写是放弃原来文件的内容，重新写文件:覆盖写是修改文件在当前读写指针所指的位置开始的部分内容:追加写是在原文件的最后添加新的内容。用户可根据自己的需要选用不同的参数选择其中的一种写方式，并将随后键盘输入的内容按照所选的方式写到文件中，最后通过双击回车结束

(5) 输入:文件的描述符

(6) 输出:实际写入的字节数

(7) 函数需完成的工作:

①检查模式、文件描述符等参数的有效性，如果无效则返回-1,并显示出错信息;

②类似地，检查文件是否为目录文件，是否已经打开等，处理方式类似，故不再赘述;

③将用户的本次输入内容保存到临时变量str[]中，等待用户双击回车结束输入

④利用打开文件表表项中的首块号查找FAT表，找到该逻辑块所在的磁盘块块号;将该磁盘块块号转化为虚拟磁盘上的内存位置

⑤根据用户所选用的不同模式对应调用do\_write() 函数将通过键盘键入的内容写到文件中

###### 实际写文件函数

实际写文件函数do\_write()

(1) 对应命令:无

(2) 命令调用格式:无

(3) 函数设计格式: int my\_write(int fd, char \*text, int len, char wstyle)

(4) 功能:被写文件函数my\_write()调用，用来将键盘输入的内容写到相应的文件中

去。磁盘文件的读写操作都必须以完整的数据块为单位进行，在写操作时，先将数据写在缓

冲区中，缓冲区的大小与磁盘块的大小相同,然后再将缓冲区中的数据一次性写到磁盘块中:读出时先将一个磁盘块中的内容读到缓冲区中，然后再传送到用户区。

(5) 输入: fd: open()函数的返回值，文件的描述符: .

text:指向要写入的内容的指针

len:本次要求写入字节数

wstyle:写方式

(6) 输出:实际写入的字节数

(7) 函数需完成的工作:

①将文件原有的内容和输入内容进行拼接，然后根据用户的不同需要进行裁剪。

②将.上述得到的文件分块写入到文件系统中;

③返回实际写入的字节数:

###### 读文件函数

读文件函数my\_read()

(1) 对应命令: read

(2) 命令调用格式: read [-r]/[-a] fd len

(3) 函数设计格式: int myread (int fd, int len)

(4) 功能:读文件前需要打开文件，否则不能完成读操作。读出指定文件中从读写指针开始的长度为len的内容到用户空间中

(5) 输入:

fd: open()函数的返回值，文件的描述符

len:要从文件中读出的字节数

(6) 输出:实际读出的字节数

(7) 函数需完成的工作:

①检查模式、文件描述符等参数的有效性，如果无效则返回-1，并显示出错信息

②类似地，检查文件是否为目录文件、是否已经打开等，处理方式类似，故不再赘述

③利用打开文件表表项中的首块号查找FAT表，找到该逻辑块所在的磁盘块块号:将该磁盘块块号转化为虚拟磁盘上的内存位置

④若用户选择读指定位置，则将指针移动到对应位置，否则默认为文件的起始位置。根据用户所选用的不同模式对应调用do. read()函數显示文件内容

###### 实际读文件函数

实际读文件函数do\_read()

(1) 对应命令:无

(2) 命令调用格式:无

(3) 函数设计格式: int do\_read (int fd, int len, char \*text)

(4) 功能:被my\_read()调用，读出指定文件中从读写指针开始的长度为len的内容到用户空间的text中

(5) 输入:

fd: open() 函数的返回值，文件的描述符

len:要求从文件中读出的字节数

text:指向存放读出数据的用户区地址

(6) 输出:实际读出的字节数

(7) 函数需完成的工作:

①确定文件的长度并计算对应的文件块数

②将该内存位置开始的一个磁盘块的内容读入buf[]中

③比较buf[]中从偏移量off开始的剩余字节数是否大于等于应读读的字节数len,如果是，则将从off开始的buf中的len长度的内容读入到text[]中;否则，将从off开始的buf中的剩余内容读入到text[]中

④直至读完所有内容后返回实际读出的字节数

###### 打印fat表函数

打印fat表函数my\_fat()

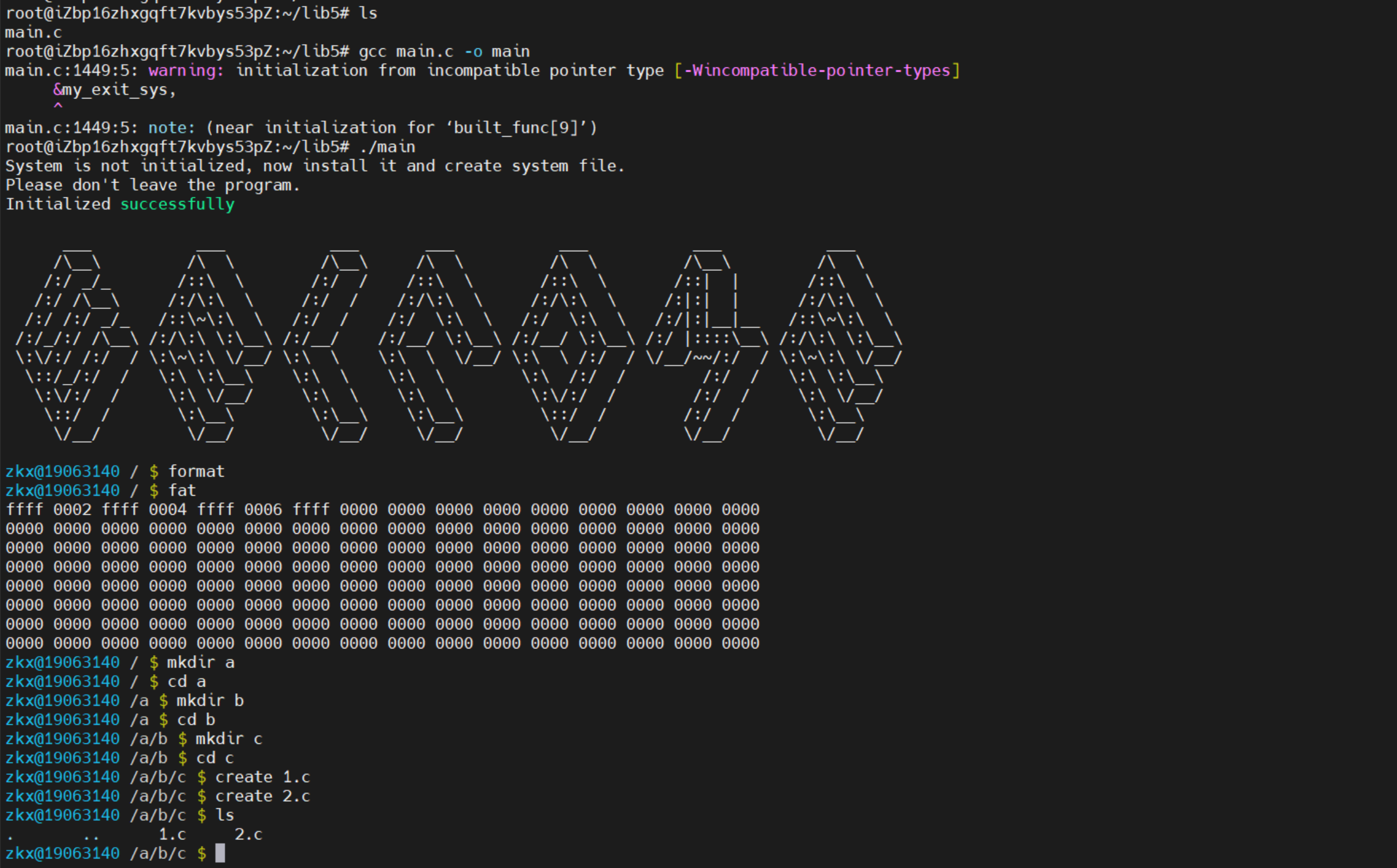
(1) 对应命令: fat

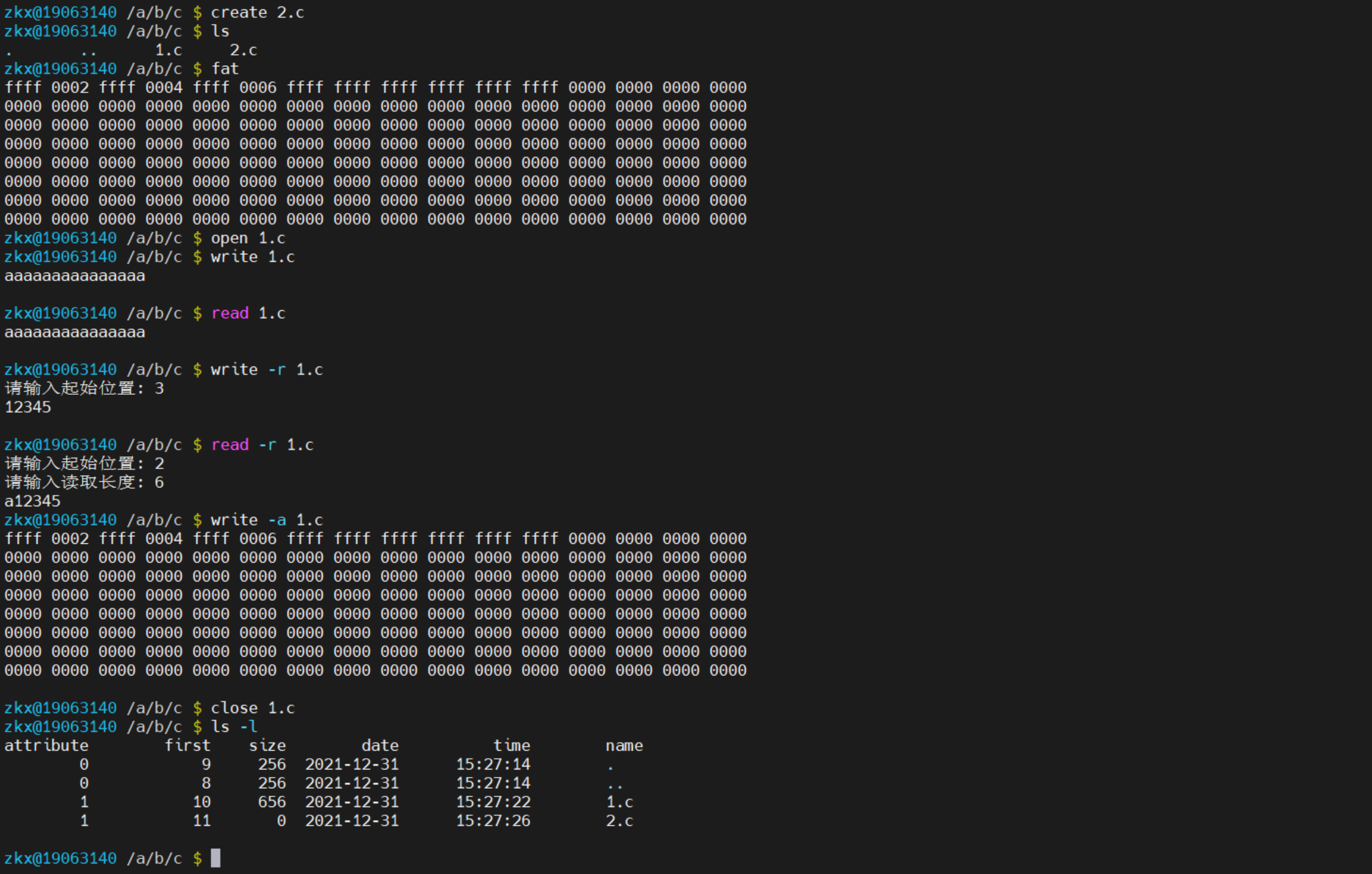
(2) 命令调用格式: fat

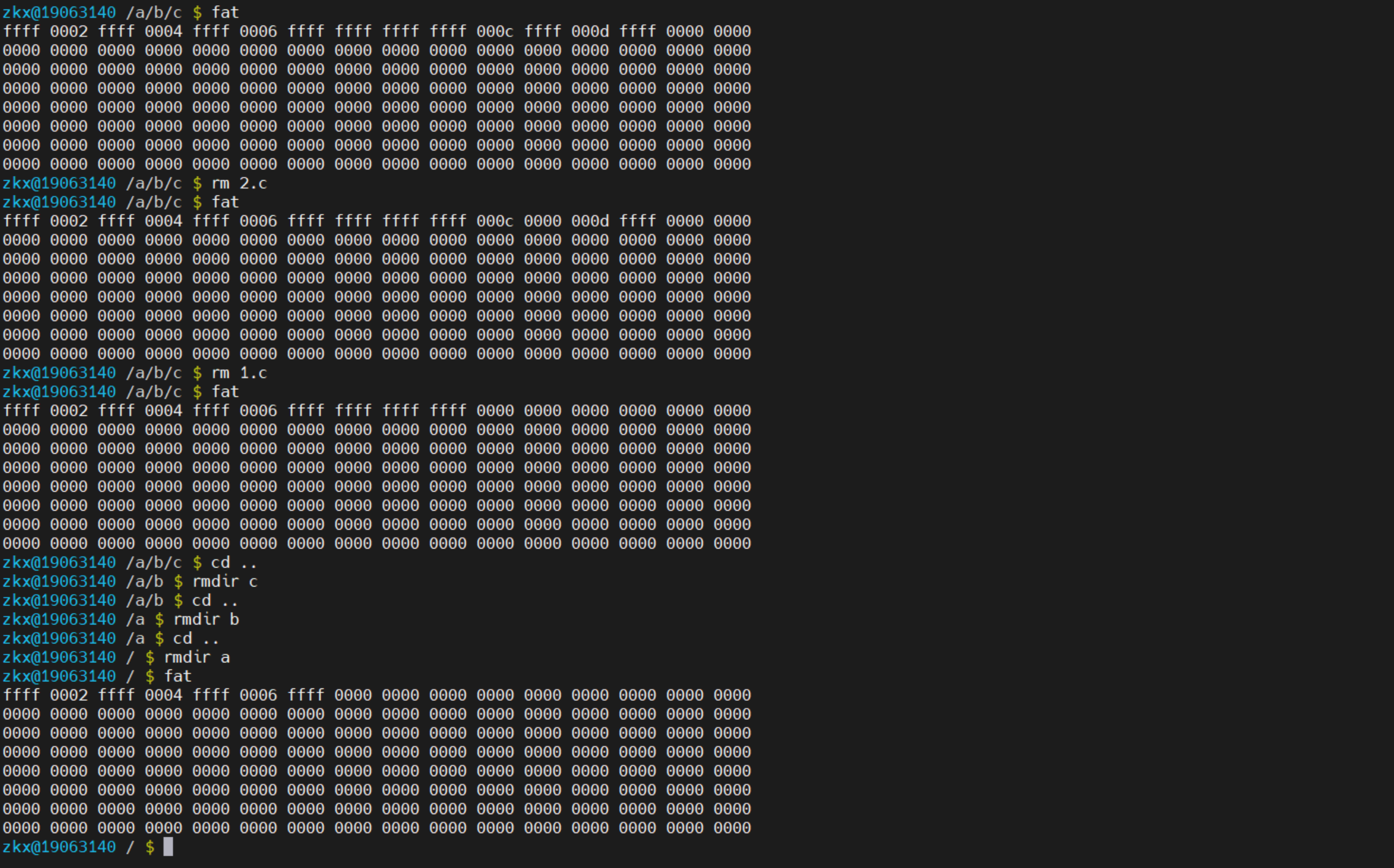
(3) 函数设计格式: int fat (char \*\*args)

(4) 功能:以十六进制打印fat表每行16个

### 四、实验过程和结果







### 五、实验体会

文件系统相对于前面几个实验是一个浩大的工程，需要对FAT表、FCB、文件的读写十分了解和熟悉。在读、写、改代码的过程中，不仅对课内所学到的文件系统的基本原理理解得更加深刻。在实践中确实能感受到文件系统的知识联系紧密，有了扎实的理论基础，方能在应用中灵活应用。

### 六、参考文献

计算机操作系统(杭电编)