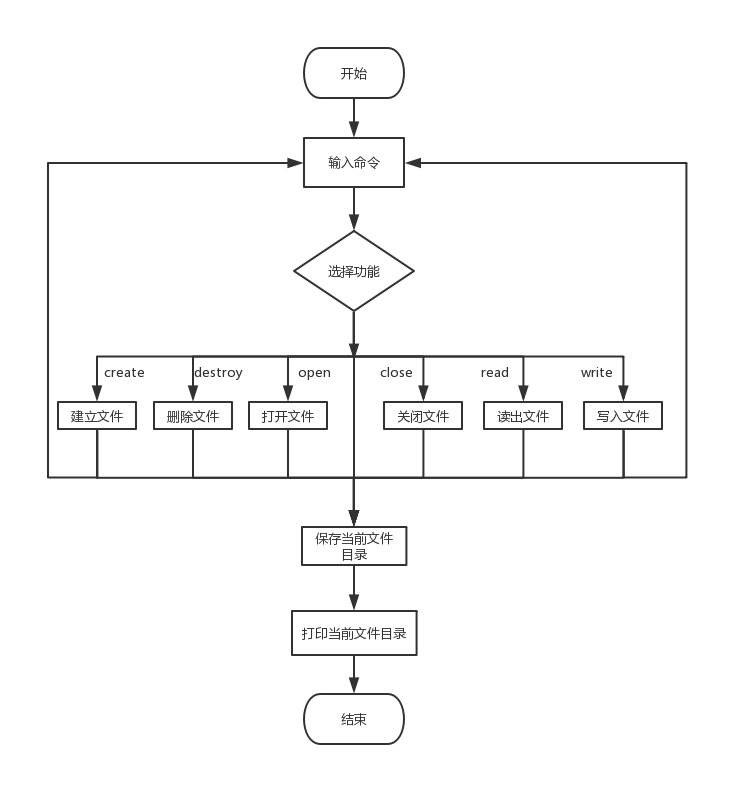
**实验五 ：文件系统**

**安全1601 16281221 邓子轩**

1. **实验目的**

本实验要求在模拟的I/O系统之上开发一个简单的文件系统。用户通过create, open, read等命令与文件系统交互。文件系统把磁盘视为顺序编号的逻辑块序列，逻辑块的编号为0至L − 1。I/O系统利用内存中的数组模拟磁盘。

1. **概要设计**



1. **程序设计**

|  |
| --- |
| 磁盘初始化代码 |
| char s[L][B];  void init(){  FILE \*fp = fopen("filesystem.txt","w+");  memset(s,0,sizeof(s));  for(int i = 0 ; i < 8 ; i++){  s[0][i] = 0xff;  }  for(int i = 1 ; i < L ; i++){  for(int j = 0 ; j < B ; j++){  if(i==1&&j<4)continue;  s[i][j] = 0xff;  }  }  for(int i = 0 ; i < L ; i++){  fwrite(s[i],sizeof(char),B,fp);  }  fclose(fp);  } |

|  |
| --- |
| 文件描述符 |
| struct file\_des{  int len; //文件长度  int num[7]; //文件分配到的磁盘块号数组  file\_des(){  len=0;  for(int i = 0 ; i < 7 ; i++)  num[i] = -1;  }  }; |

|  |
| --- |
| 目录项 |
| //目录项，每个占32字节  struct dir{  char name[28]; //文件名  int index; //文件描述符序号  }; |

|  |
| --- |
| 打开文件表 |
| //打开文件表  struct file\_open{  char buf[B]; //文件缓冲区  int p; //文件读写指针  int index; //文件描述符序号  }; |

**算法及关键数据结构设计**

• create(filename): 根据指定的文件名创建新文件。

• destroy(filename): 删除指定文件。

• open(filename): 打开文件。该函数返回的索引号可用于后续的read, write, lseek, 或close 操作。

• close(index): 关闭制定文件。

• read(index, mem\_area, count): 从指定文件顺序读入 count 个字 节 mem\_area 指定的内存位置。读操作从文件的读写指针指示的位置开始。

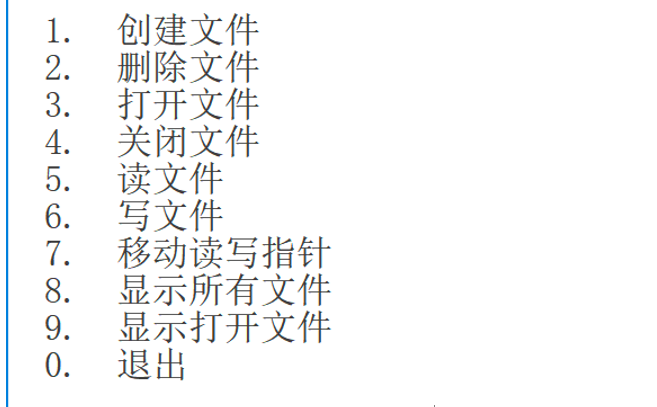
• write(index, mem\_area, count): 把 mem\_area 指定的内存位置开始的 count 个字节顺序写入指定文件。写操作从文件的读写指针指示的位置开始。

• lseek(index, pos):把文件的读写指针移动到 pos 指定的位置。pos 是一个整数，表示从文件开始位置的偏移量。文件打开时，读写指针自动设置为 0。每次读写操作之后，它指向最后被访问的字节的下一个位置。 lseek 能够在不进行读写操作的情况下改变读写指针能位置。

• directory: 列表显示所有文件及其长度。

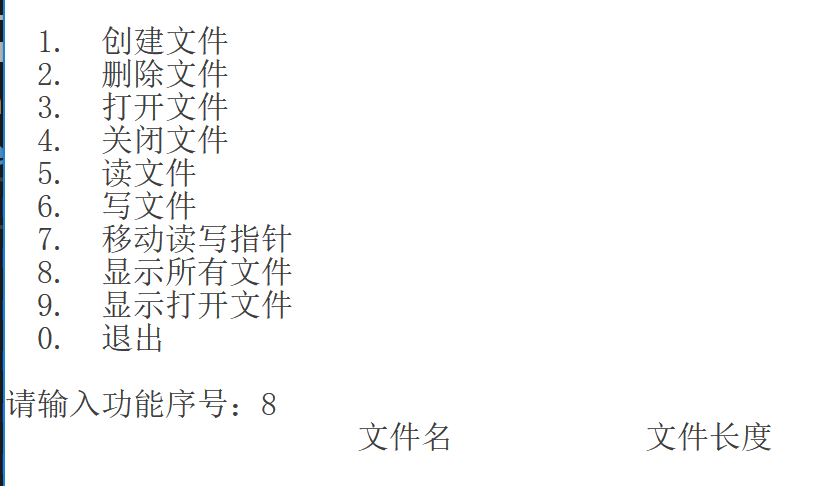
1. **实验结果**

文件系统功能列表：

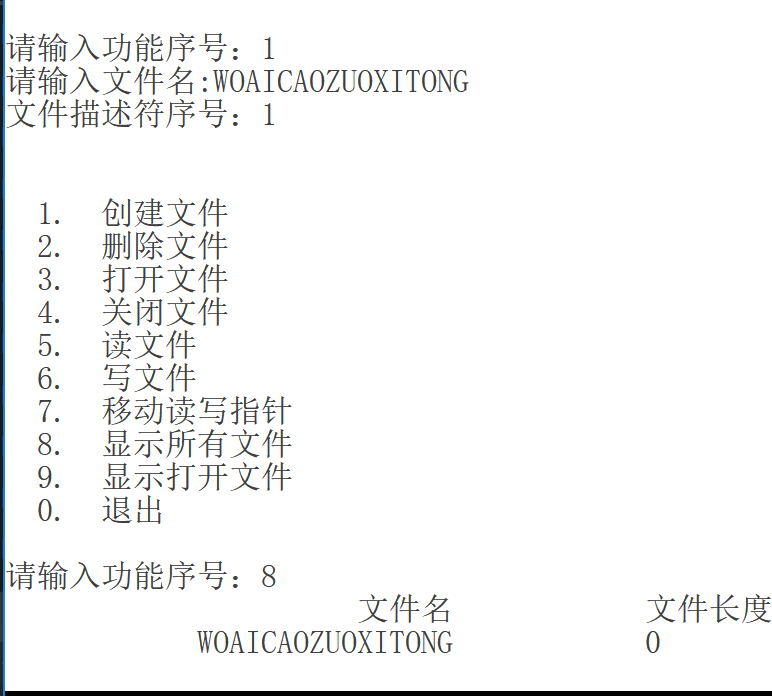


1、创建文件

先查看已有文件：



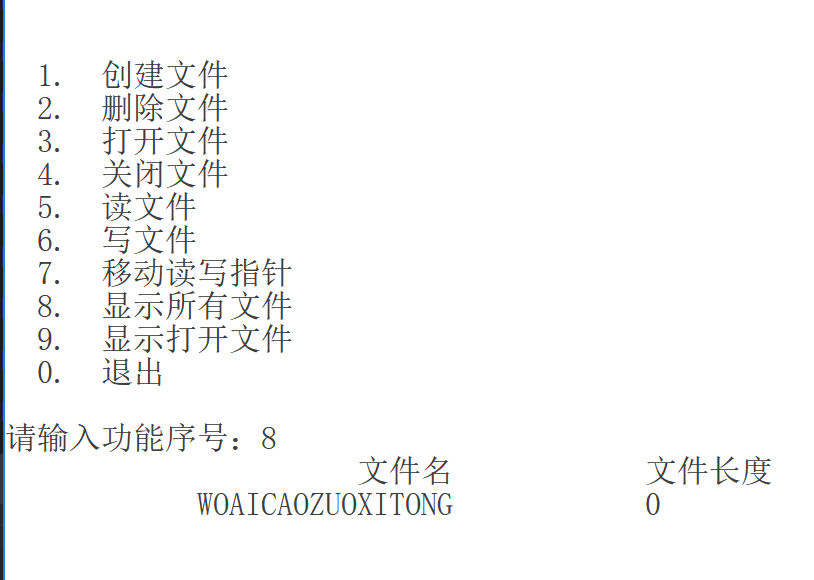
创建文件：



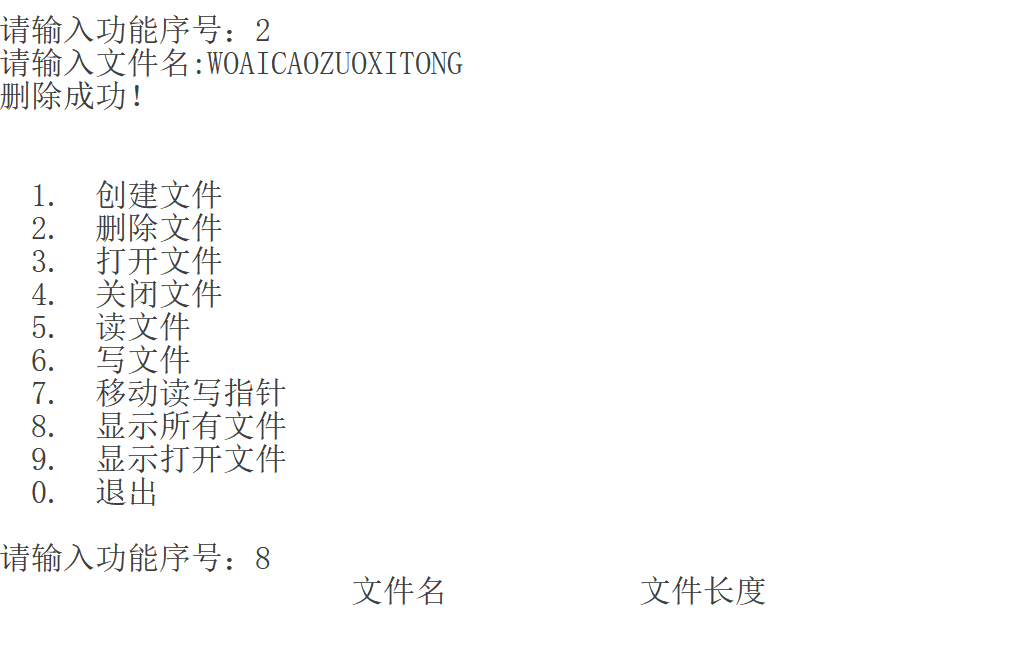
创建成功！

2、删除文件

先查看已有文件：

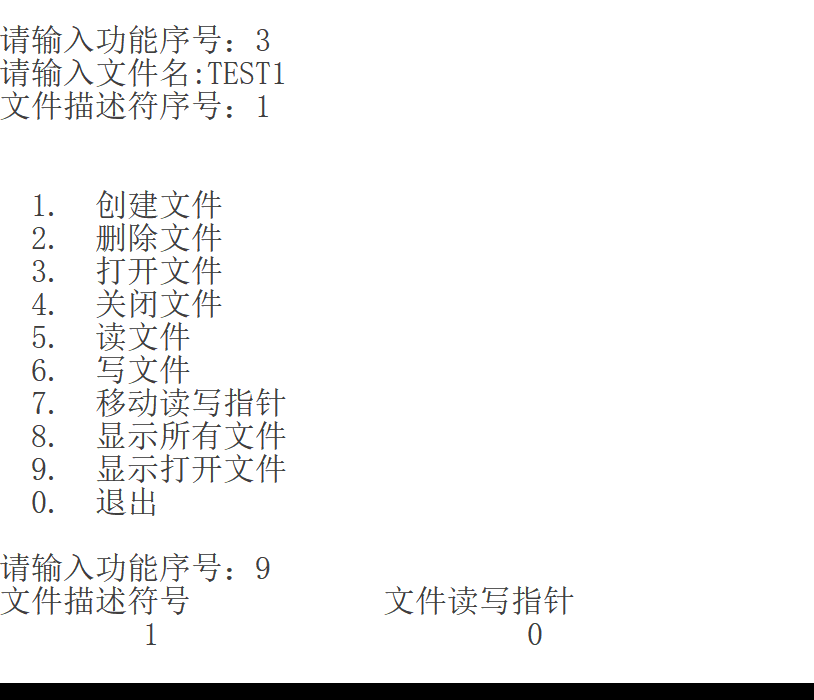


删除文件：



删除成功！

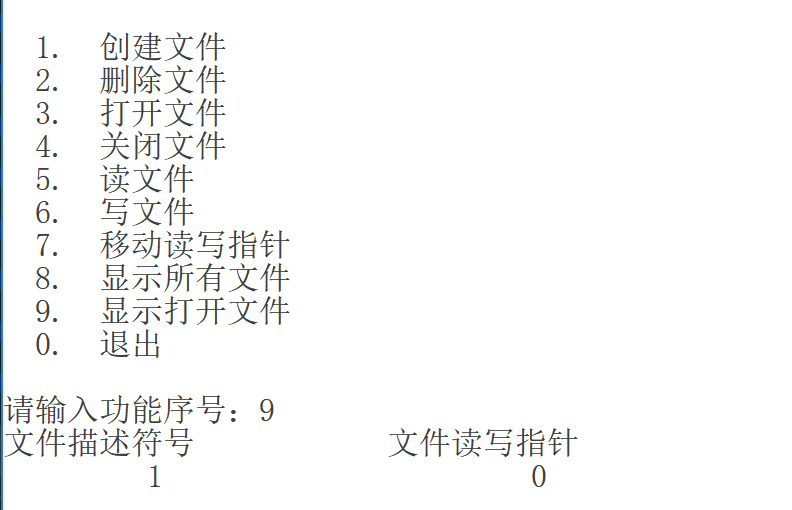
3、打开文件



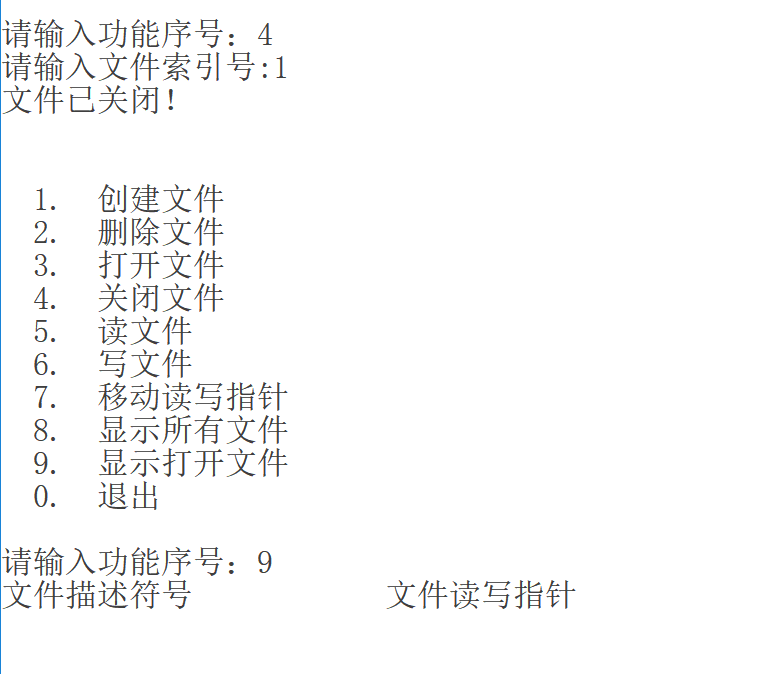
使用打开文件功能后，再使用显示打开文件功能确认是否打开，文件打开成功！

4、关闭文件

先查看已打开的文件：



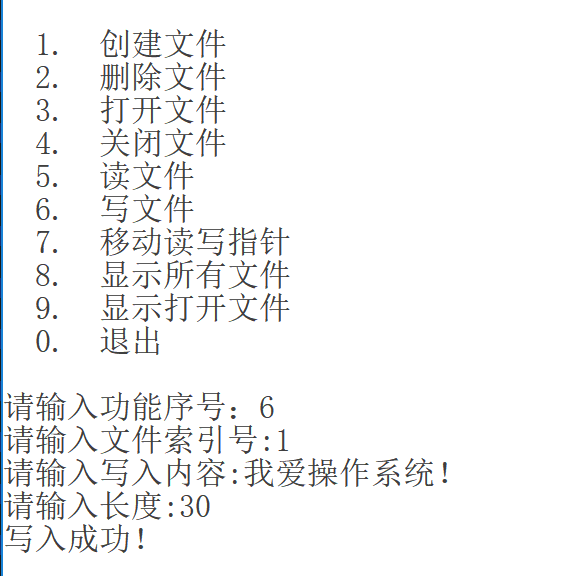
关闭文件：



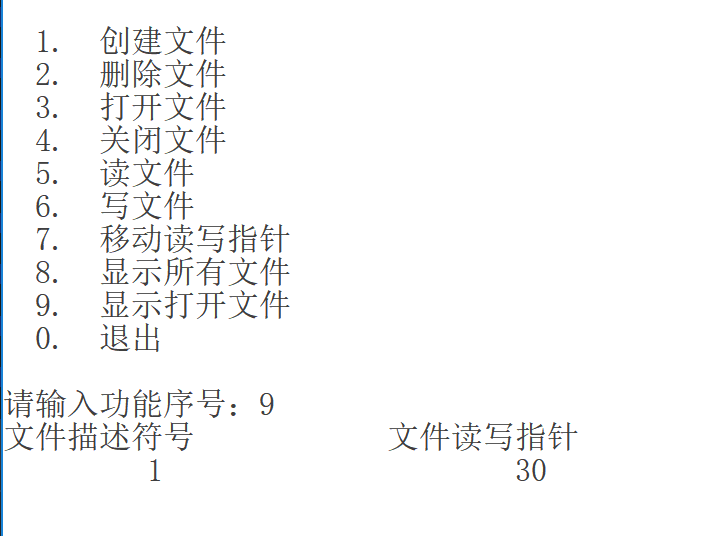
使用关闭文件功能后，再使用显示打开文件功能确认是否关闭，文件关闭成功！

5、读写文件

先打开文件，再写文件，写入成功！

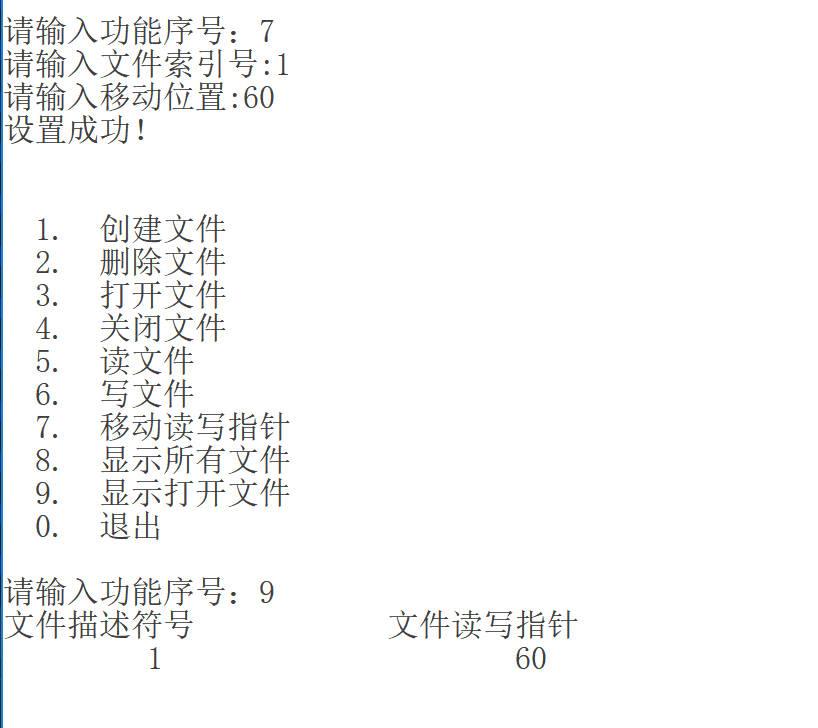


查看文件长度以及读写指针：

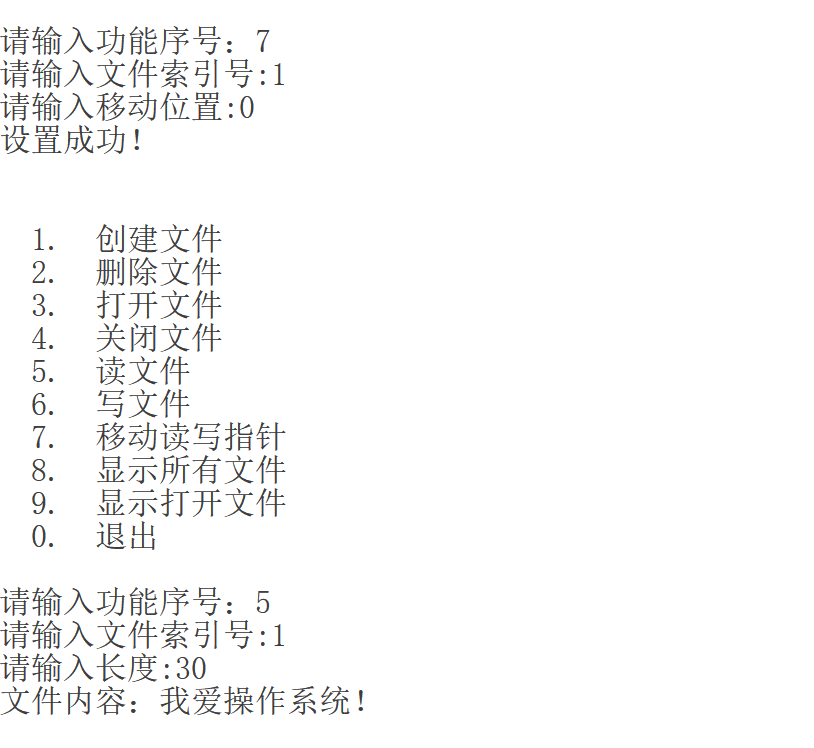




移动读写指针：



读文件：



在读文件之前，把读写指针移动到位置0，读出所有写入内容。