山东大学__计算机科学与技术__学院

操作系统 课程实验报告

学号：202200101007	姓名：张祎乾	班级：22.3 班

实验题目：实验 5：扩展 Nachos 的文件系统

实验学时：2	实验日期：2025/2/24

实验目的：

理解文件系统中文件操作的实现方法，如文件打开、读、写、扩展、定位、关闭等；理解如何管理硬盘空闲块；

创建文件时，如何为文件分配目录项及文件头(FCB)；

理解文件扩展时，如何为要扩展的数据查找并分配空闲块；

理解文件扩展后，文件大小是如何记录与保存的；

文件被删除后，如何回收为其分配的资源，如文件头、目录项、硬盘块等；

拓展：有精力的同学可进一步尝试多级目录(目录树)的设计与实现方法。

根据上述工作，总结操作系统(如 Nachos)读写文件，以及对文件追加数据的过程与步骤(如操作系统需要操作哪些文件系统的控制信息，如何操作的等)。

实验环境：WSL、Ubuntu

先言：我在进行实验的过程中，主要有三部分记录，可以点击跳转：

1、 调试命令及图片：我将运行指南相关命令时的调试截图放于此

2、 任务及收获：实验中提到的任务，实验目的中的部分内容

3、 README.md，我将一些指南的重要信息记录在其中，我会以图片的形式贴于此文件，若可能，我将会把 README.md 一起提交

特殊部分(根据实验进行选择，可能没有)：

拓展：多级目录

调试命令及图片

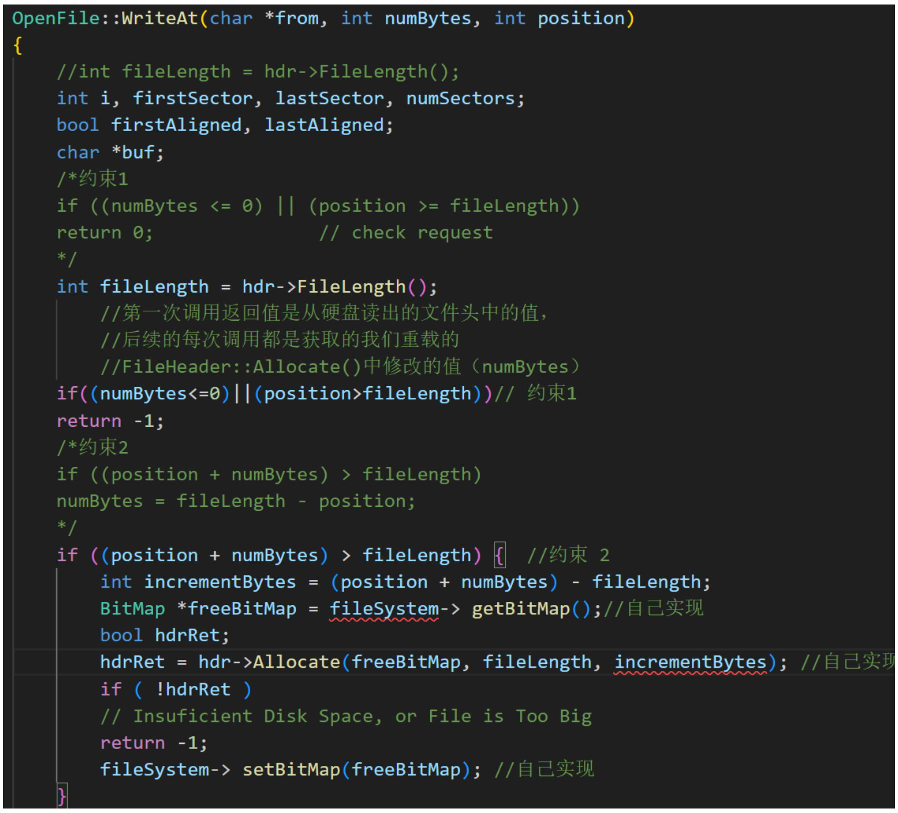
6.2.1

(1)需要修改哪些模块，需要使用哪些不需要修改的模块；
需要修改的模块：
需要使用但不需要修改的模块：
(2)在那些需要修改的模块中，哪些函数需要修改，如何修改；
中需要修改的函数：
(3)在那些需要修改的模块中，是否需要添加函数与变量；
中需要添加的函数
中需要添加的变量
(4)是否需要在修改的模块中移动变量，或者从一个模块移动到另一个模块；

6.4

6.4.2 修改代码

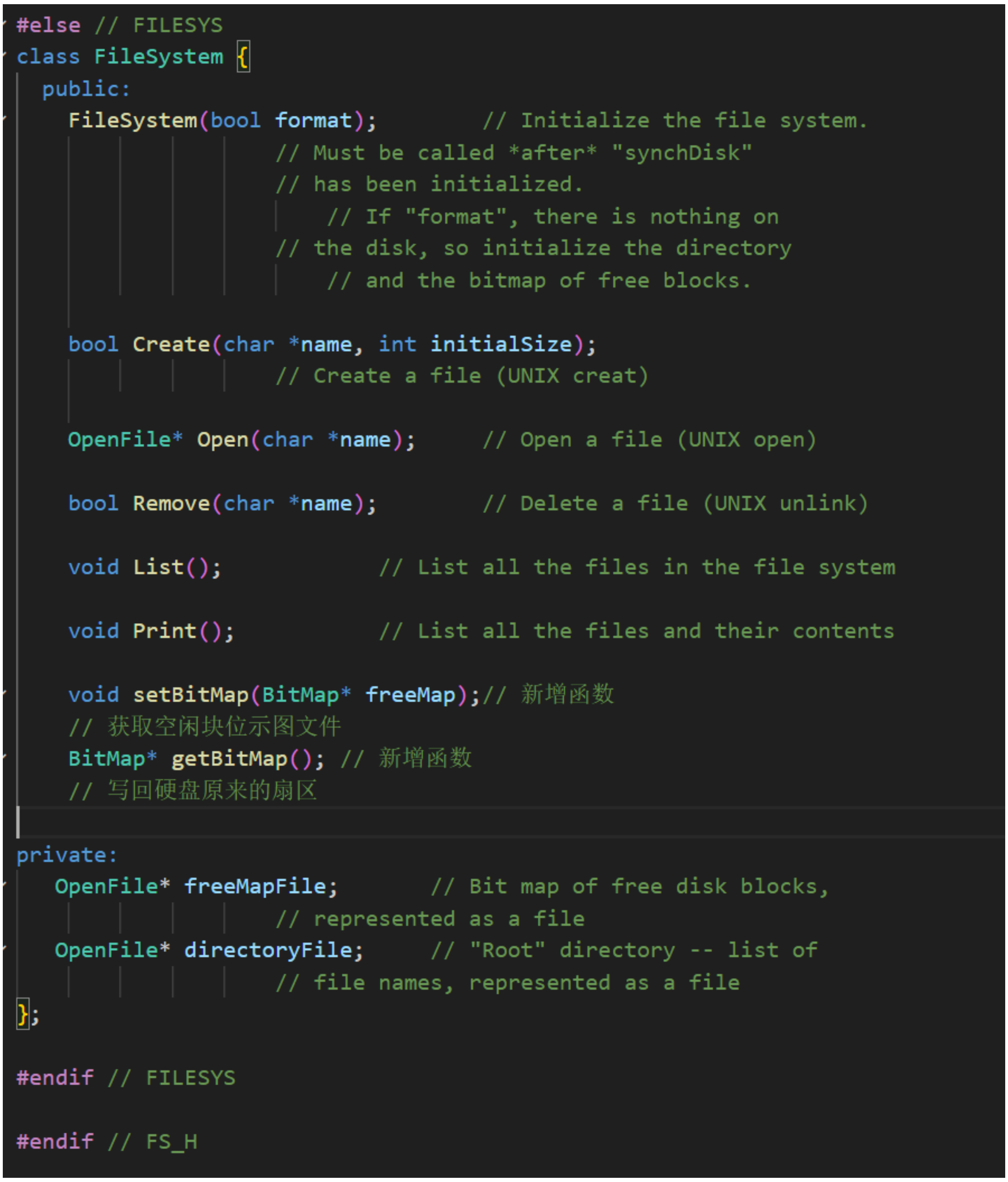
Openfile.cc 中的 OpenFile::WriteAt 修改为如下所示：



```
OpenFile::WriteAt(char *from, int numBytes, int position)
{
    //int fileLength = hdr->FileLength();
    int i, firstSector, lastSector, numSectors;
    bool firstAligned, lastAligned;
    char *buf;
    /*约束1
    if ((numBytes <= 0) || (position >= fileLength))
    return 0;                // check request
    */
    int fileLength = hdr->FileLength();
        //第一次调用返回值是从硬盘读出的文件头中的值，
        //后续的每次调用都是获取的我们重载的
        //FileHeader::Allocate()中修改的值（numBytes）
    if((numBytes<=0)||(position>fileLength))// 约束1
    return -1;
    /*约束2
    if ((position + numBytes) > fileLength)
    numBytes = fileLength - position;
    */
    if ((position + numBytes) > fileLength) {  //约束 2
        int incrementBytes = (position + numBytes) - fileLength;
        BitMap *freeBitMap = fileSystem-> getBitMap();//自己实现
        bool hdrRet;
        hdrRet = hdr->Allocate(freeBitMap, fileLength, incrementBytes); //自己实现
        if ( !hdrRet )
        // Insuficient Disk Space, or File is Too Big
        return -1;
        fileSystem-> setBitMap(freeBitMap); //自己实现
    }
```

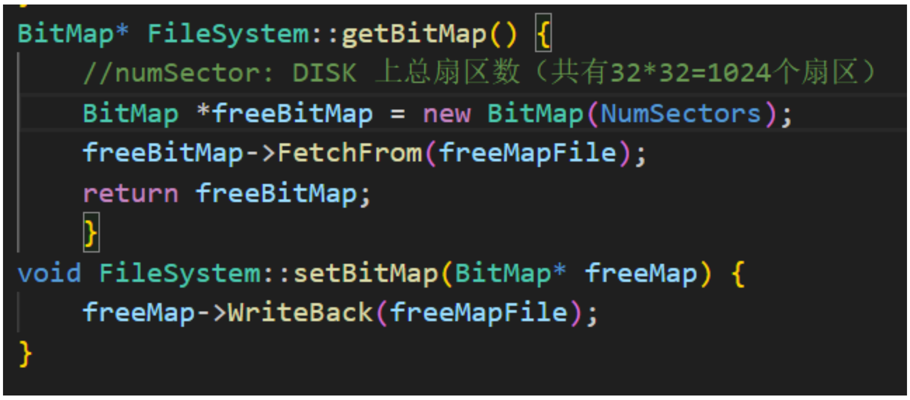
filesys.h 中的 FileSystem 类修改如下：

新增 void setBitMap(BitMap* freeMap); 和 BitMap* getBitMap();



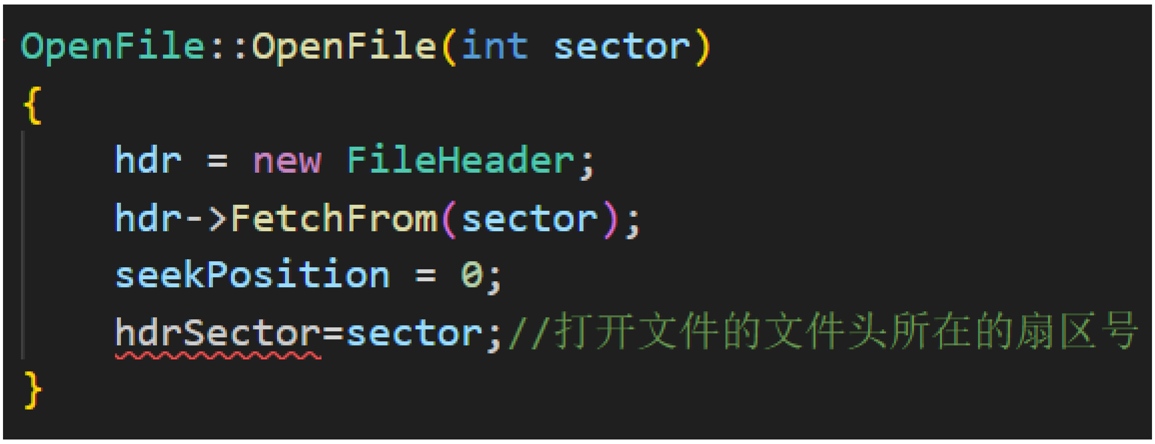
```
#else // FILESYS
class FileSystem {
  public:
    FileSystem(bool format);        // Initialize the file system.
                    // Must be called *after* "synchDisk" 
                    // has been initialized.
                        // If "format", there is nothing on
                    // the disk, so initialize the directory
                        // and the bitmap of free blocks.

    bool Create(char *name, int initialSize);  
                    // Create a file (UNIX creat)

    OpenFile* Open(char *name);     // Open a file (UNIX open)

    bool Remove(char *name);        // Delete a file (UNIX unlink)

    void List();            // List all the files in the file system

    void Print();           // List all the files and their contents

    void setBitMap(BitMap* freeMap);// 新增函数
    // 获取空闲块位示图文件
    BitMap* getBitMap(); // 新增函数
    // 写回硬盘原来的扇区

private:
   OpenFile* freeMapFile;       // Bit map of free disk blocks,
                    // represented as a file
   OpenFile* directoryFile;     // "Root" directory -- list of 
                    // file names, represented as a file
};

#endif // FILESYS

#endif // FS_H
```

filesys.cc 中实现上述两个新增函数：



```
BitMap* FileSystem::getBitMap() {
    //numSector: DISK 上总扇区数（共有32*32=1024个扇区）
    BitMap *freeBitMap = new BitMap(NumSectors);
    freeBitMap->FetchFrom(freeMapFile);
    return freeBitMap;
    }
void FileSystem::setBitMap(BitMap* freeMap) {
    freeMap->WriteBack(freeMapFile);
}
```

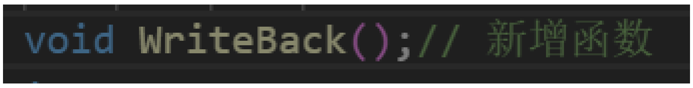
openfile.cc 中的 OpenFile::OpenFile 函数修改如下：



```
OpenFile::OpenFile(int sector)
{
    hdr = new FileHeader;
    hdr->FetchFrom(sector);
    seekPosition = 0;
    hdrSector=sector;//打开文件的文件头所在的扇区号
}
```

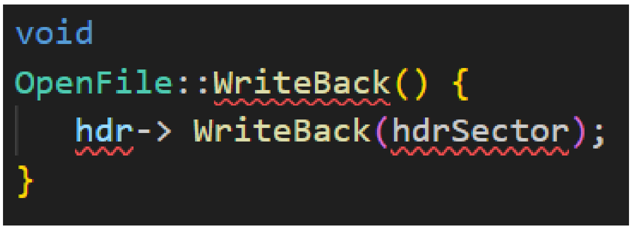
openfile.h 中的 OpenFile 类修改如下：

新增函数 void WriteBack();



```
void WriteBack();// 新增函数
```

Openfile.h 中实现 WirteBack 函数：

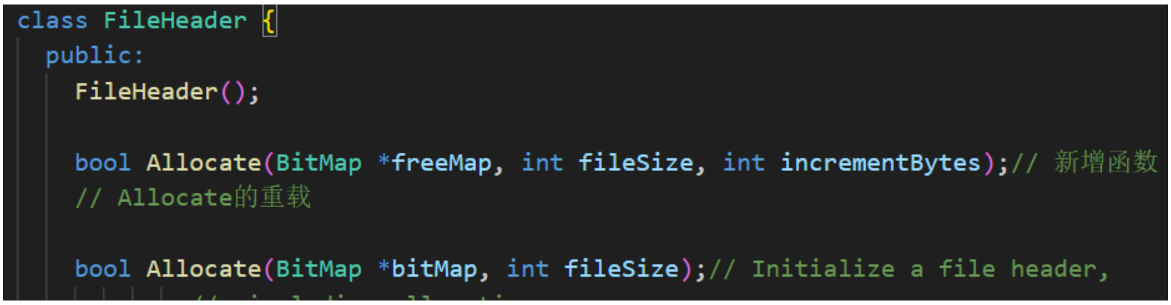


```
void
OpenFile::WriteBack() {
    hdr-> WriteBack(hdrSector);
}
```

filehdr.h 中的 OpenFile 类修改如下：

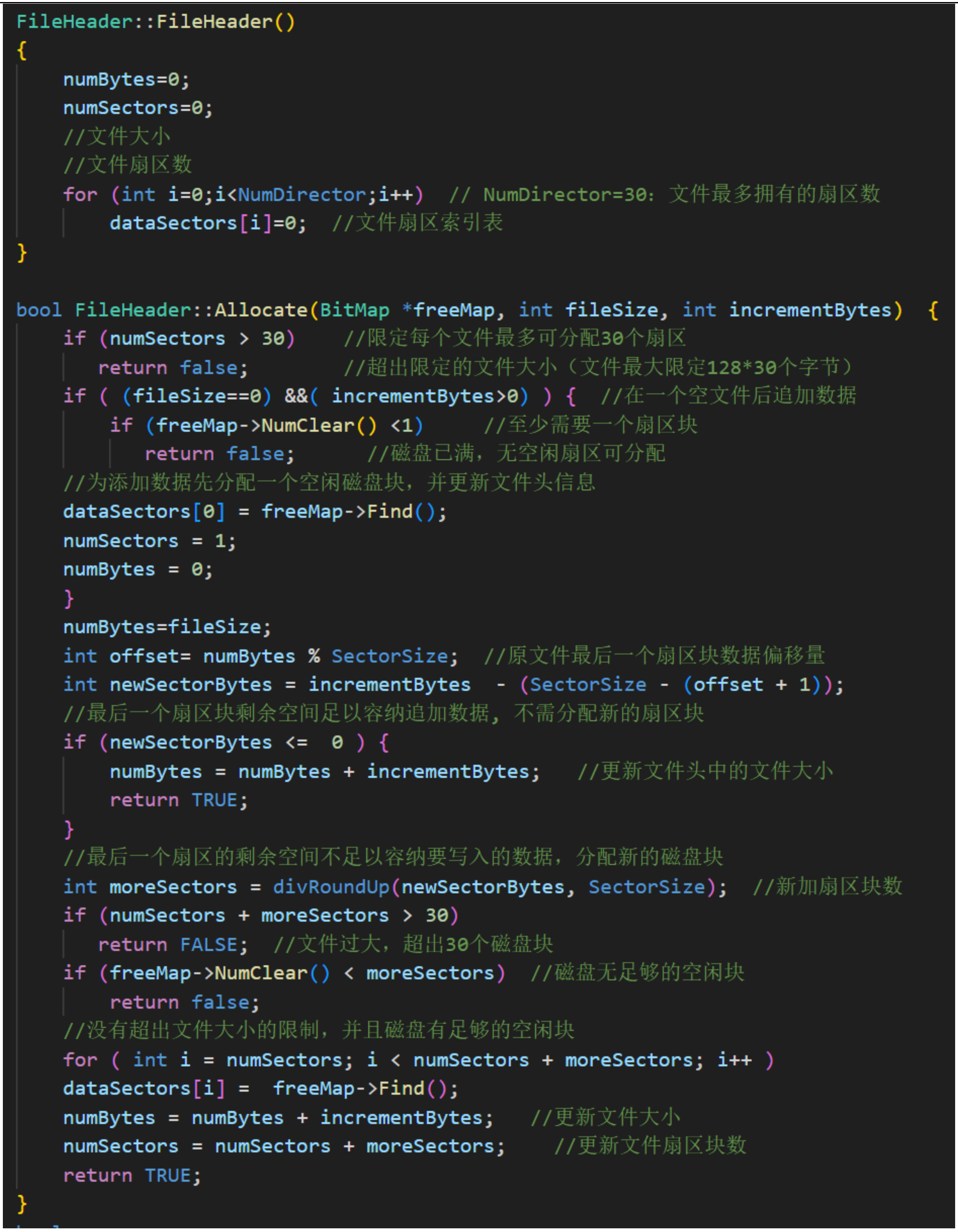
新增构造函数 FileHeader();

新增重载函数 bool Allocate(BitMap *freeMap, int fileSize, int incrementBytes);



```
class FileHeader {
  public:
    FileHeader();

    bool Allocate(BitMap *freeMap, int fileSize, int incrementBytes);// 新增函数
    // Allocate的重载

    bool Allocate(BitMap *bitMap, int fileSize);// Initialize a file header,
```

filehdr.h 中的实现两个函数：

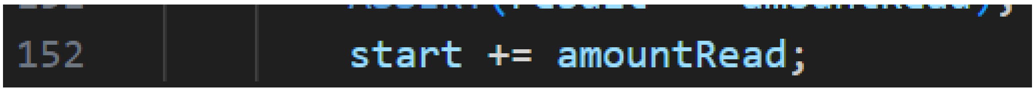


```
FileHeader::FileHeader()
{
    numBytes=0;
    numSectors=0;
    //文件大小
    //文件扇区数
    for (int i=0;i<NumDirector;i++)  // NumDirector=30: 文件最多拥有的扇区数
        dataSectors[i]=0;  //文件扇区索引表
}

bool FileHeader::Allocate(BitMap *freeMap, int fileSize, int incrementBytes)  {
    if (numSectors > 30)     //限定每个文件最多可分配30个扇区
       return false;         //超出限定的文件大小（文件最大限定128*30个字节）
    if ( (fileSize==0) &&( incrementBytes>0) ) {  //在一个空文件后追加数据
        if (freeMap->NumClear() <1)      //至少需要一个扇区块
            return false;       //磁盘已满，无空闲扇区可分配
    //为添加数据先分配一个空闲磁盘块，并更新文件头信息
    dataSectors[0] = freeMap->Find();
    numSectors = 1;
    numBytes = 0;
    }
    numBytes=fileSize;
    int offset= numBytes % SectorSize;  //原文件最后一个扇区块数据偏移量
    int newSectorBytes = incrementBytes  - (SectorSize - (offset + 1));
    //最后一个扇区块剩余空间足以容纳追加数据, 不需分配新的扇区块
    if (newSectorBytes <=  0 ) {
        numBytes = numBytes + incrementBytes;   //更新文件头中的文件大小
        return TRUE;
    }
    //最后一个扇区的剩余空间不足以容纳要写入的数据, 分配新的磁盘块
    int moreSectors = divRoundUp(newSectorBytes, SectorSize);  //新加扇区块数
    if (numSectors + moreSectors > 30)
       return FALSE;  //文件过大，超出30个磁盘块
    if (freeMap->NumClear() < moreSectors)  //磁盘无足够的空闲块
        return false;
    //没有超出文件大小的限制，并且磁盘有足够的空闲块
    for ( int i = numSectors; i < numSectors + moreSectors; i++ )
    dataSectors[i] =  freeMap->Find();
    numBytes = numBytes + incrementBytes;   //更新文件大小
    numSectors = numSectors + moreSectors;    //更新文件扇区块数
    return TRUE;
}
```

fstest.cc 中修改如下：

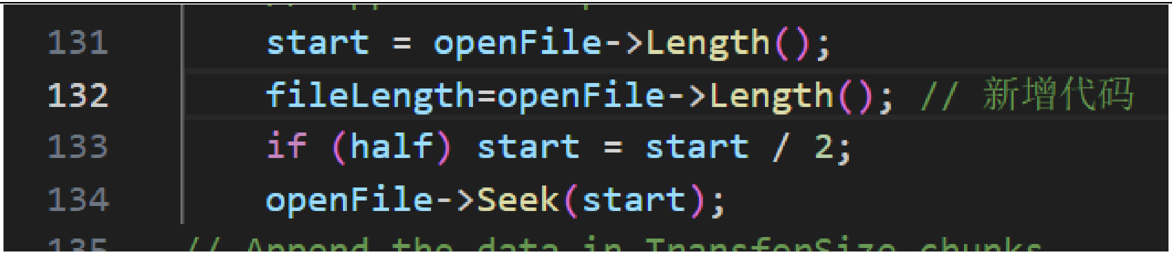
Append()(应该是以下三处，做完实验补的报告，可能有遗漏)：

解除 start += amountRead;的注释(约莫 152 行)



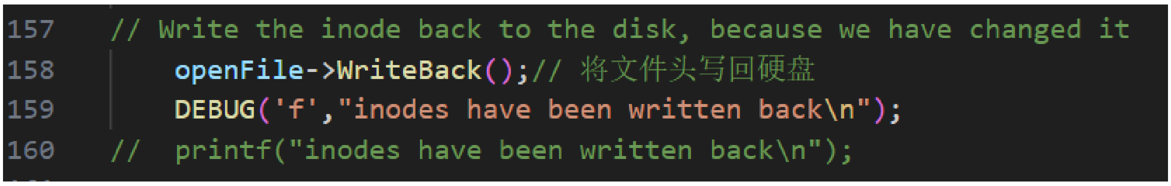
```
152        start += amountRead;
```

新增一行代码(约莫 132 行)



```
    start = openFile->Length();
    fileLength=openFile->Length(); // 新增代码
    if (half) start = start / 2;
    openFile->Seek(start);
// Append the data in TransferSize chunks
```

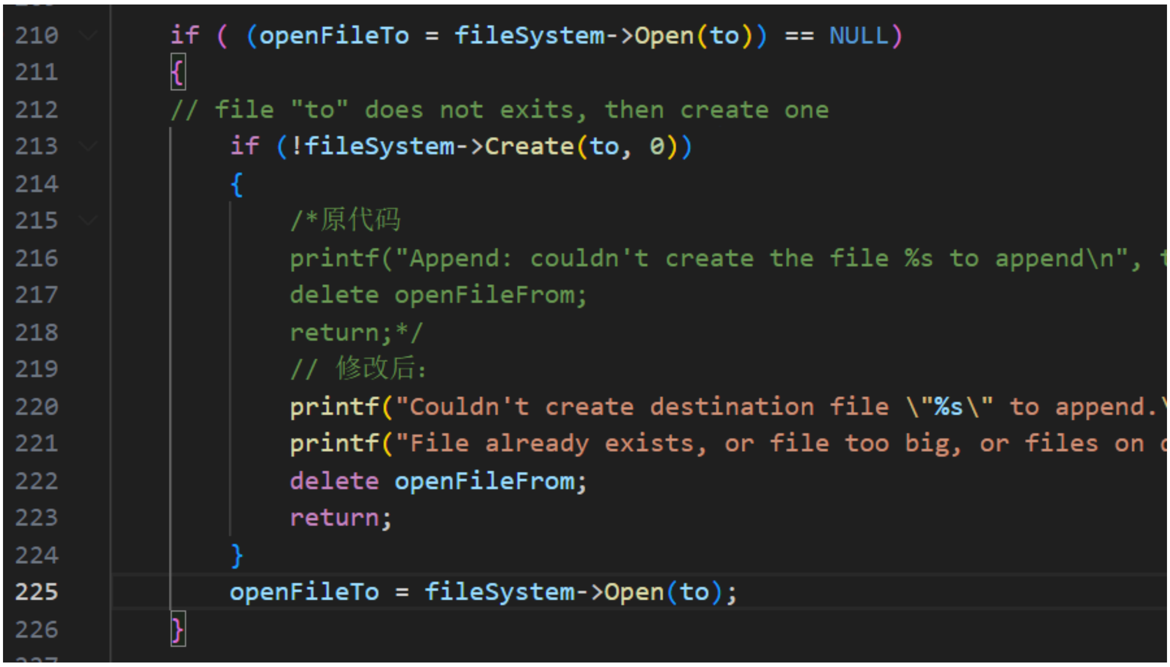
取消 openFileTo->WriteBack();注释并新增打印信息



```
// Write the inode back to the disk, because we have changed it
    openFile->WriteBack();// 将文件头写回硬盘
    DEBUG('f',"inodes have been written back\n");
//  printf("inodes have been written back\n");
```

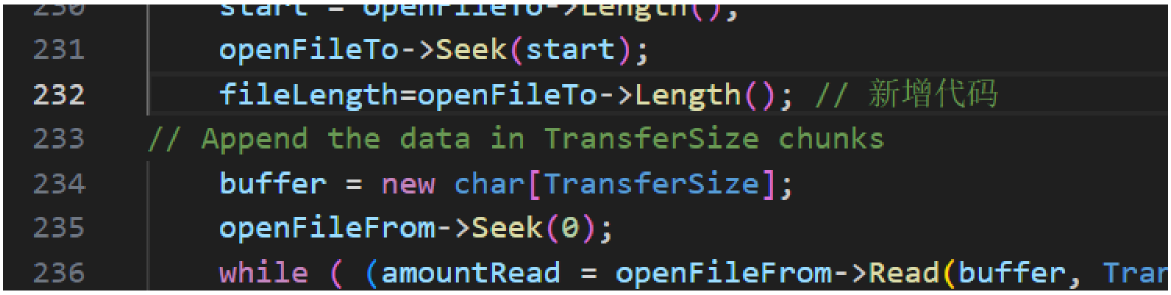
Nappend():

修改 if ((openFileTo = fileSystem->Open(to)) == NULL)中的内容(不是很有必要，此处修改是因为原代码和指南给出的修改后代码不一样，所以我同步一下)



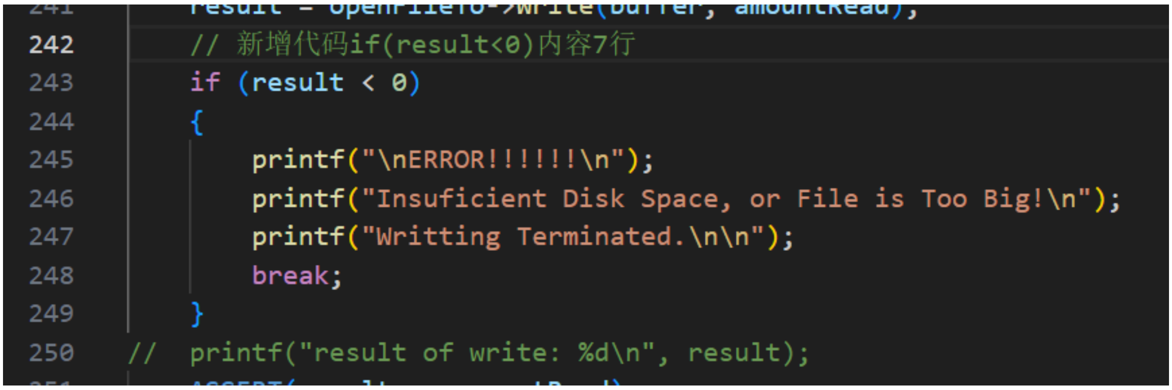
```
    if ( (openFileTo = fileSystem->Open(to)) == NULL)
    {
    // file "to" does not exits, then create one
        if (!fileSystem->Create(to, 0))
        {
            /*原代码
            printf("Append: couldn't create the file %s to append\n", t
            delete openFileFrom;
            return;*/
            // 修改后:
            printf("Couldn't create destination file \"%s\" to append.\
            printf("File already exists, or file too big, or files on d
            delete openFileFrom;
            return;
        }
        openFileTo = fileSystem->Open(to);
    }
```

新增代码 fileLength=openFileTo->Length();(约莫 232 行，我前边加了几行注释，实际应该少几行)



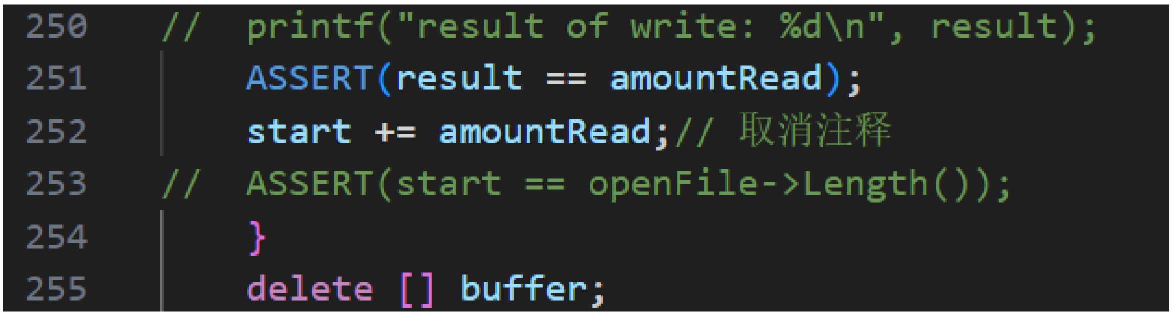
```
230     start = openFileTo->Length();
231     openFileTo->Seek(start);
232     fileLength=openFileTo->Length(); // 新增代码
233 // Append the data in TransferSize chunks
234     buffer = new char[TransferSize];
235     openFileFrom->Seek(0);
236     while ( (amountRead = openFileFrom->Read(buffer, Trar
```

新增代码 if(result<0)内容 7 行



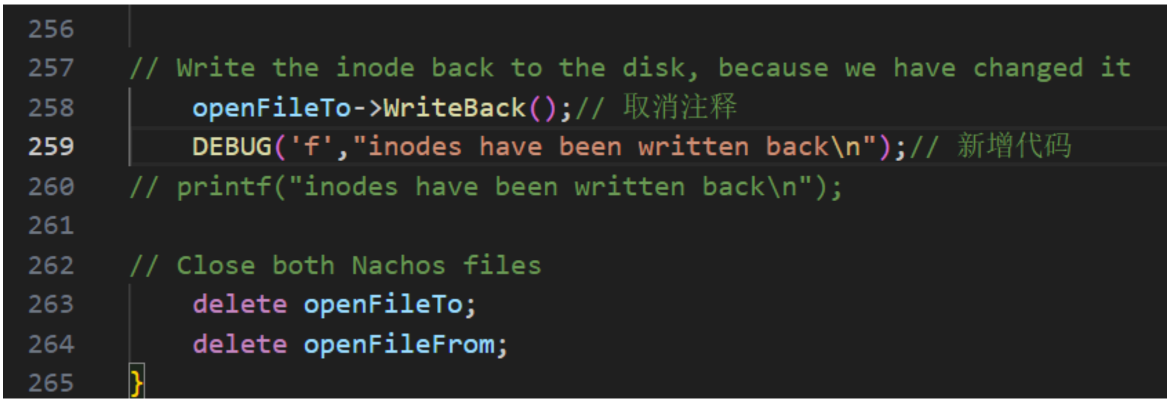
```
        result = openFileTo->Write(buffer, amountRead);
        // 新增代码if(result<0)内容7行
        if (result < 0)
        {
            printf("\nERROR!!!!!!\n");
            printf("Insuficient Disk Space, or File is Too Big!\n");
            printf("Writting Terminated.\n\n");
            break;
        }
    //  printf("result of write: %d\n", result);
```

取消 start += amountRead;的注释



```
    //  printf("result of write: %d\n", result);
        ASSERT(result == amountRead);
        start += amountRead;// 取消注释
    //  ASSERT(start == openFile->Length());
        }
        delete [] buffer;
```

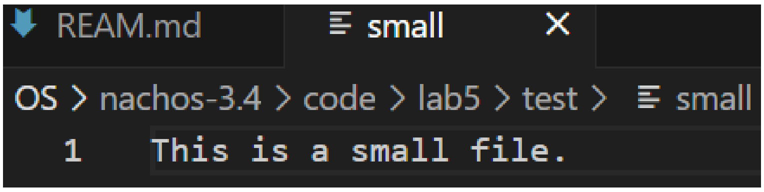
取消 openFileTo->WriteBack();注释并新增打印信息



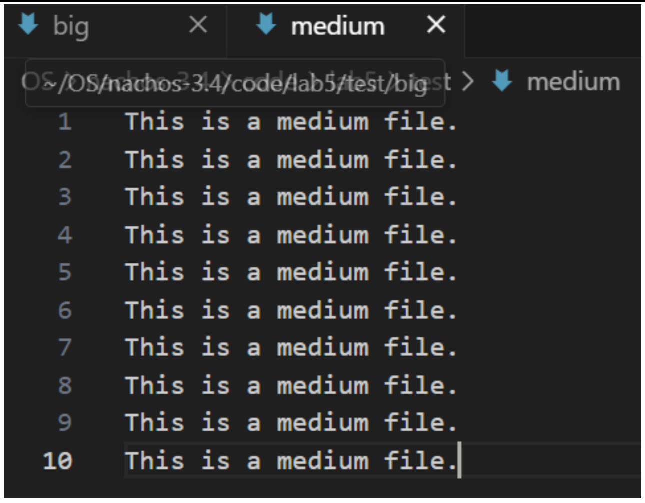
```

    // Write the inode back to the disk, because we have changed it
        openFileTo->WriteBack();// 取消注释
        DEBUG('f',"inodes have been written back\n");// 新增代码
    // printf("inodes have been written back\n");

    // Close both Nachos files
        delete openFileTo;
        delete openFileFrom;
    }
```

6.4.3 nachos 文件系统测试

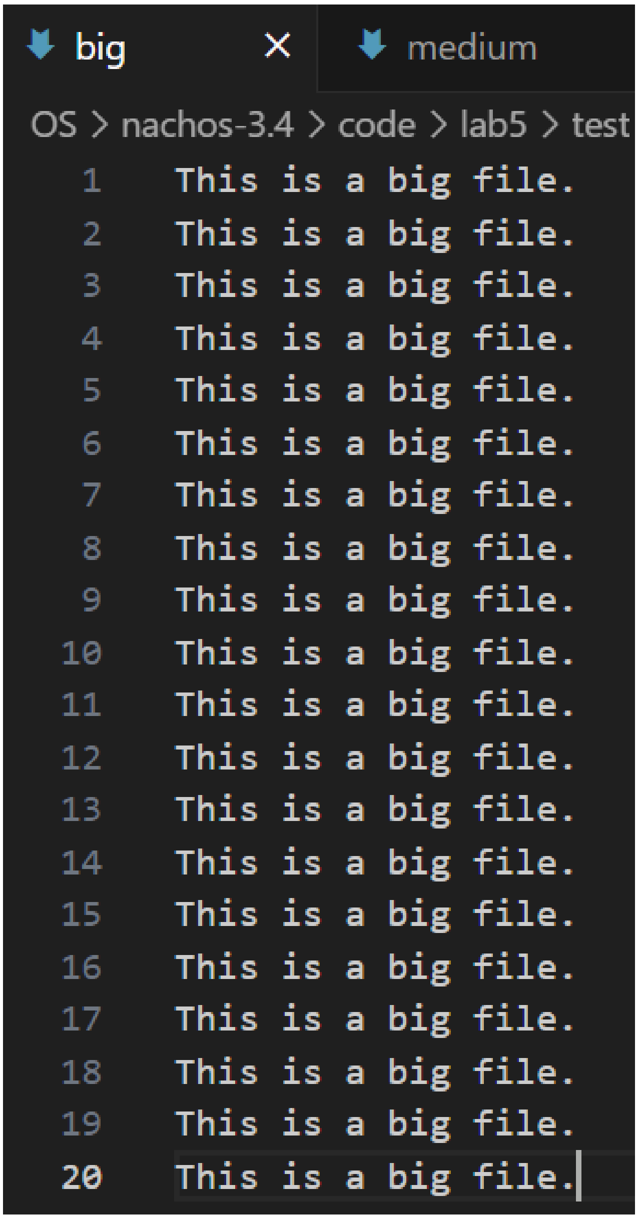
(1) 将../file/test/下的 small、medium、big 文件的内容修改。



```
This is a small file.
```



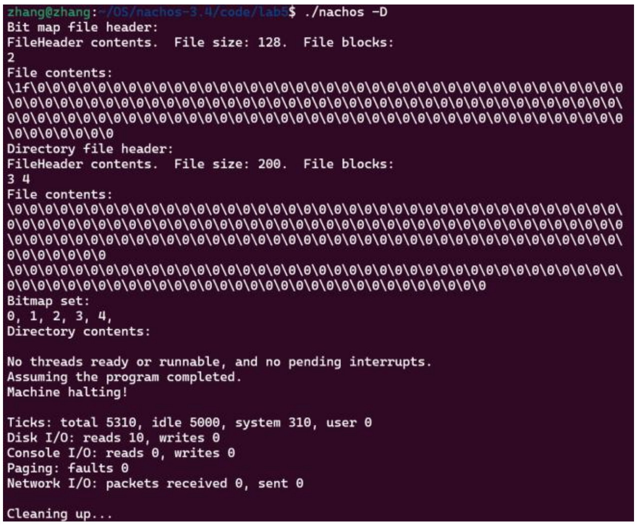
```
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
This is a medium file.
```



```
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
This is a big file.
```

(2)./nachos -f：初始化 DISK

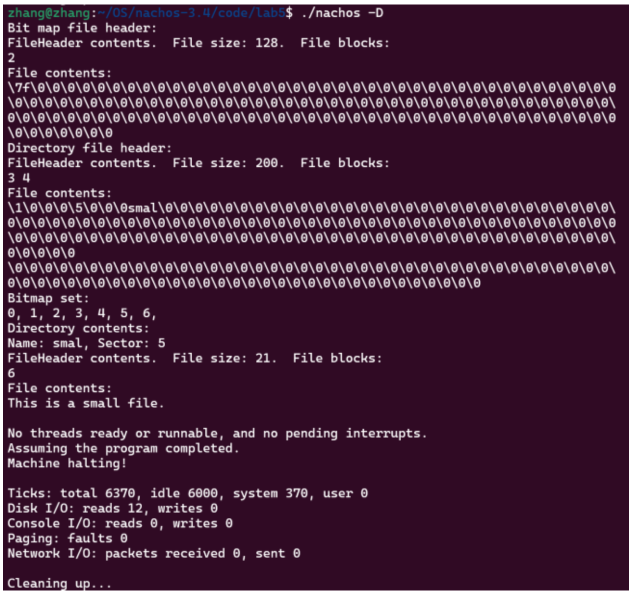
(3)./nachos -D：获得原始 DISK 数据



```
zhang@zhang:~/OS/nachos-3.4/code/lab5$ ./nachos -D
Bit map file header:
FileHeader contents.  File size: 128.  File blocks:
2
File contents:
\1f\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0
Directory file header:
FileHeader contents.  File size: 200.  File blocks:
3 4
File contents:
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
Bitmap set:
0, 1, 2, 3, 4,
Directory contents:

No threads ready or runnable, and no pending interrupts.
Assuming the program completed.
Machine halting!

Ticks: total 5310, idle 5000, system 310, user 0
Disk I/O: reads 10, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
```

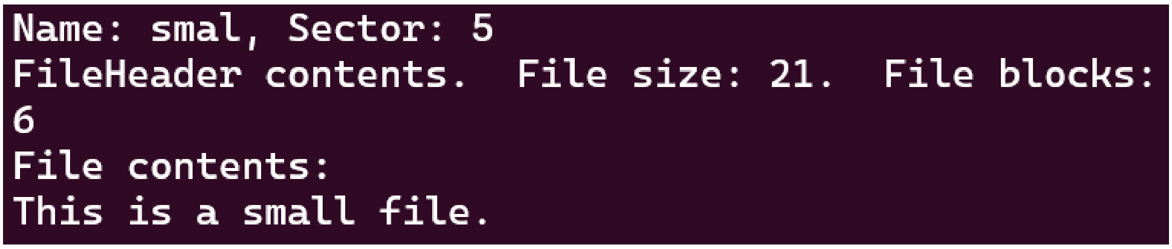
(4) ./nachos -cp test/small smal：复制 test 目录下的 UNIX 文件 small 到 DISK 中

(5)./ nachos -D：查看拥有 small 的 DISK



```
zhang@zhang:~/OS/nachos-3.4/code/lab5$ ./nachos -D
Bit map file header:
FileHeader contents.  File size: 128.  File blocks:
2
File contents:
\7f\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0
Directory file header:
FileHeader contents.  File size: 200.  File blocks:
3 4
File contents:
\1\0\0\0\5\0\0\0smal\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
Bitmap set:
0, 1, 2, 3, 4, 5, 6,
Directory contents:
Name: smal, Sector: 5
FileHeader contents.  File size: 21.  File blocks:
6
File contents:
This is a small file.

No threads ready or runnable, and no pending interrupts.
Assuming the program completed.
Machine halting!

Ticks: total 6370, idle 6000, system 370, user 0
Disk I/O: reads 12, writes 0
Console I/O: reads 0, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0

Cleaning up...
```

可以看到，文件 small 的文件头处于 5 号磁盘，文件内容处于 6 号磁盘：

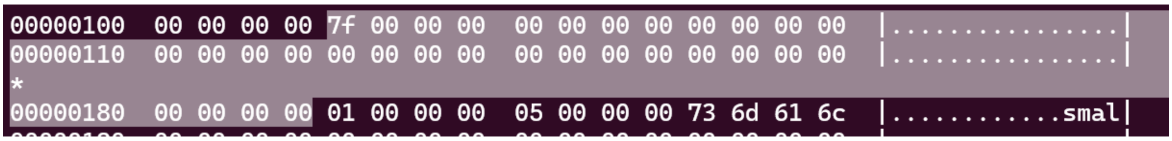


```
Name: smal, Sector: 5
FileHeader contents.  File size: 21.  File blocks:
6
File contents:
This is a small file.
```

hexdump -C DISK

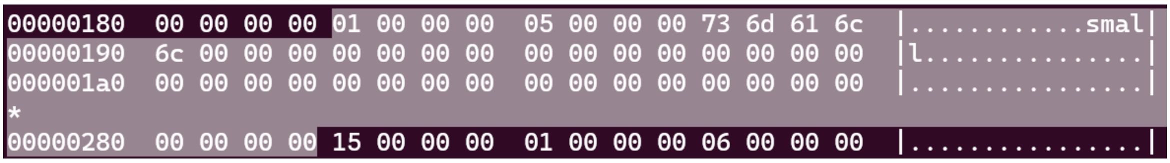
0 号扇区 1 号扇区均未改变

2 号扇区：位示图，启动了 5、6 号扇区



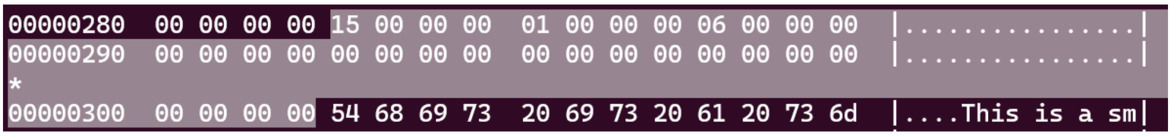
```
00000100  00 00 00 00 7f 00 00 00  00 00 00 00 00 00 00 00  |................|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000180  00 00 00 00 01 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
```

3、4 号扇区：多了 small 文件的目录项：1-5-small



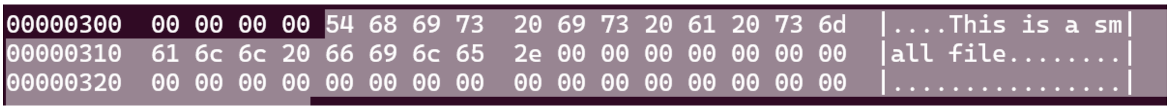
```
00000180  00 00 00 00 01 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
00000190  6c 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |l...............|
000001a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000280  00 00 00 00 15 00 00 00  01 00 00 00 06 00 00 00  |................|
```

5 号扇区：small 文件头，表示 small 文件有：15Bytes-1 个扇区-6 号扇区



```
00000280  00 00 00 00 15 00 00 00  01 00 00 00 06 00 00 00  |................|
00000290  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000300  00 00 00 00 54 68 69 73  20 69 73 20 61 20 73 6d  |....This is a sm|
```

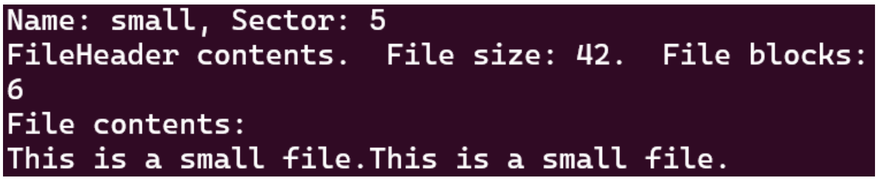
6 号扇区：small 文件内容



```
00000300  00 00 00 00 54 68 69 73  20 69 73 20 61 20 73 6d  |....This is a sm|
00000310  61 6c 6c 20 66 69 6c 65  2e 00 00 00 00 00 00 00  |all file........|
00000320  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

(6) ./nachos -ap test/small small：将 test 目录下的 UNIX 文件 samll 附加到 Nachos 文件 small 中

(7) ./nachos -D

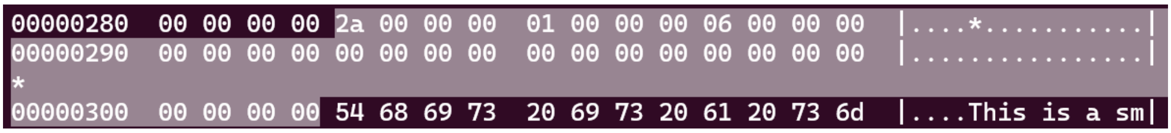


```
Name: small, Sector: 5
FileHeader contents.  File size: 42.  File blocks:
6
File contents:
This is a small file.This is a small file.
```

hexdump -C DISK

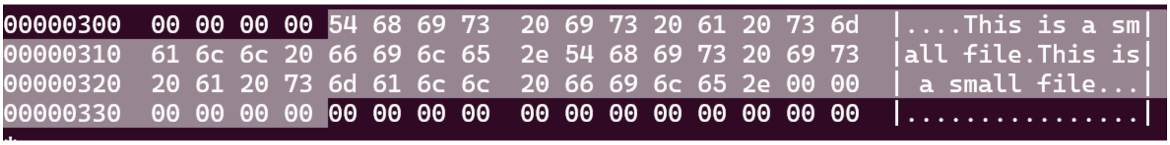
0、1、2、3、4 号扇区均未改变

5 号扇区：表明该文件的字节数增加了，并且是翻倍的



```
00000280  00 00 00 00 2a 00 00 00  01 00 00 00 06 00 00 00  |....*...........|
00000290  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000300  00 00 00 00 54 68 69 73  20 69 73 20 61 20 73 6d  |....This is a sm|
```

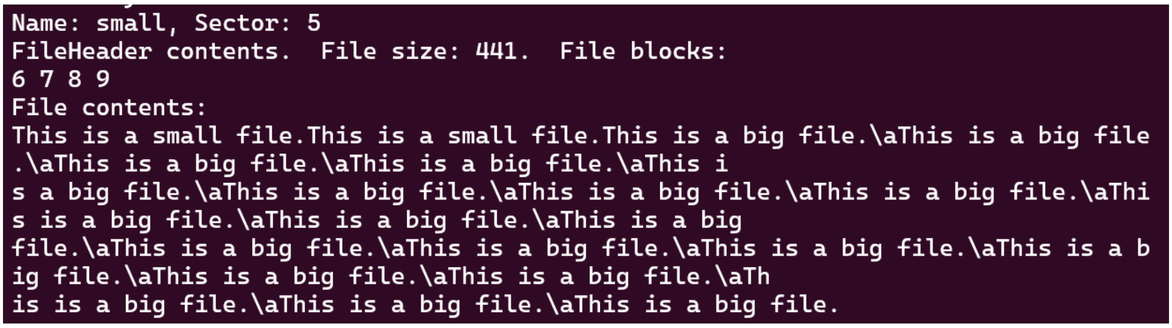
6 号扇区：文件内容多重复了一遍



```
00000300  00 00 00 00 54 68 69 73  20 69 73 20 61 20 73 6d  |....This is a sm|
00000310  61 6c 6c 20 66 69 6c 65  2e 54 68 69 73 20 69 73  |all file.This is|
00000320  20 61 20 73 6d 61 6c 6c  20 66 69 6c 65 2e 00 00  | a small file...|
00000330  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

(8) ./nachos -ap test/big small，将 UNIX 文件 big 附加到一个 small 中

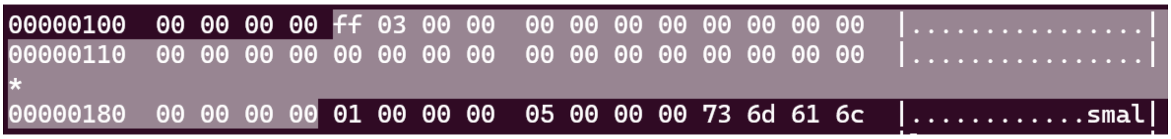
(9)./nachos -D



```
Name: small, Sector: 5
FileHeader contents.  File size: 441.  File blocks:
6 7 8 9
File contents:
This is a small file.This is a small file.This is a big file.\aThis is a big file
.\aThis is a big file.\aThis is a big file.\aThis i
s a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThi
s is a big file.\aThis is a big file.\aThis is a big
file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThis is a b
ig file.\aThis is a big file.\aThis is a big file.\aTh
is is a big file.\aThis is a big file.\aThis is a big file.
```

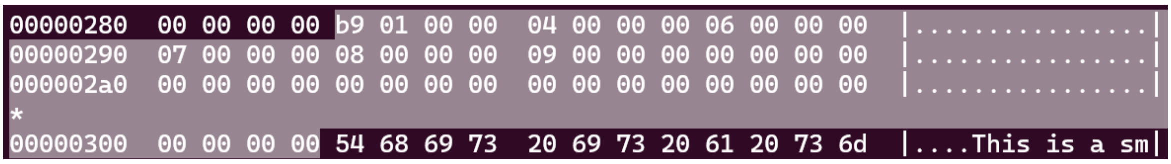
hexdump -C DISK

2 号扇区，表明 7、8、9 号扇区也被启用了



```
00000100  00 00 00 00 ff 03 00 00  00 00 00 00 00 00 00 00  |................|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000180  00 00 00 00 01 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
```

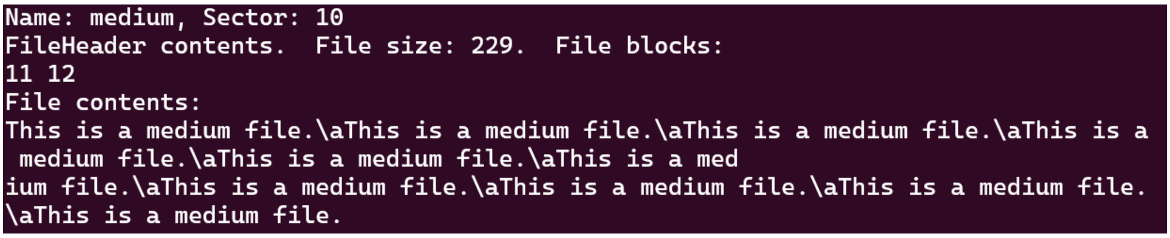
5 号扇区：文件大小多了，用了 4 个扇区，用了 6、7、8、9 号扇区



```
00000280  00 00 00 00 b9 01 00 00  04 00 00 00 06 00 00 00  |................|
00000290  07 00 00 00 08 00 00 00  09 00 00 00 00 00 00 00  |................|
000002a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000300  00 00 00 00 54 68 69 73  20 69 73 20 61 20 73 6d  |....This is a sm|
```

(10) ./nachos -ap test/medium medium

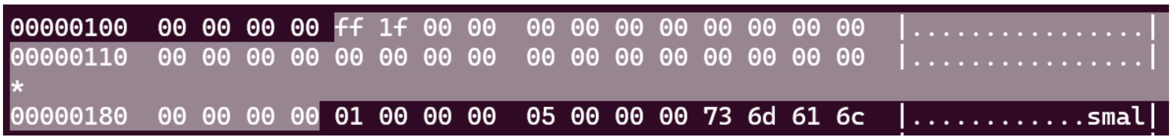
(11)./nachos -D：发现多了 medium 文件



```
Name: medium, Sector: 10
FileHeader contents.  File size: 229.  File blocks:
11 12
File contents:
This is a medium file.\aThis is a medium file.\aThis is a medium file.\aThis is a
 medium file.\aThis is a medium file.\aThis is a med
ium file.\aThis is a medium file.\aThis is a medium file.\aThis is a medium file.
\aThis is a medium file.
```

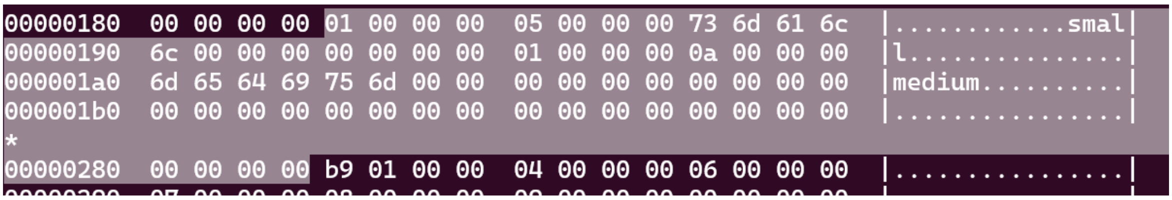
hexdump -C DISK

2 号扇区：表明多启用了 10、11、12 号扇区



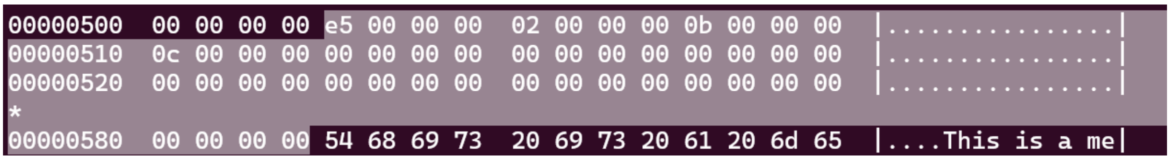
```
00000100  00 00 00 00 ff 1f 00 00  00 00 00 00 00 00 00 00  |................|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000180  00 00 00 00 01 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
```

3、4 号扇区：多了一条目录项：1-a-medium



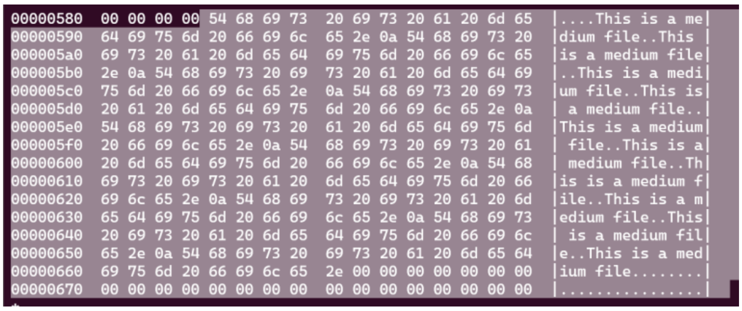
```
00000180  00 00 00 00 01 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
00000190  6c 00 00 00 00 00 00 00  01 00 00 00 0a 00 00 00  |l...............|
000001a0  6d 65 64 69 75 6d 00 00  00 00 00 00 00 00 00 00  |medium..........|
000001b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000280  00 00 00 00 b9 01 00 00  04 00 00 00 06 00 00 00  |................|
```

10 号扇区：medium 有 0xe5Bytes，2 个扇区，11 和 12 号扇区



```
00000500  00 00 00 00 e5 00 00 00  02 00 00 00 0b 00 00 00  |................|
00000510  0c 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000520  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000580  00 00 00 00 54 68 69 73  20 69 73 20 61 20 6d 65  |....This is a me|
```

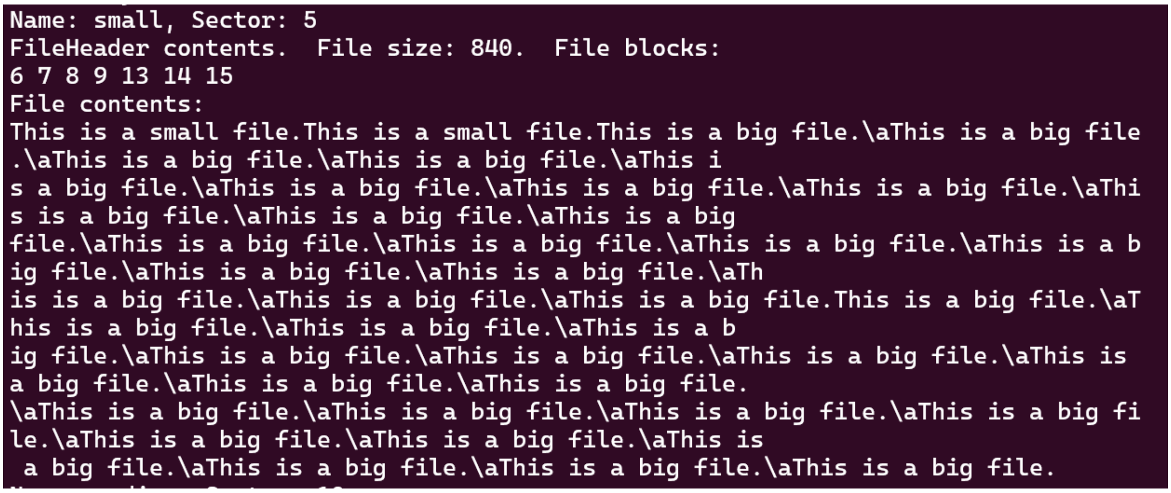
11、12 号扇区：文件内容



```
00000580  00 00 00 00 54 68 69 73  20 69 73 20 61 20 6d 65  |....This is a me|
00000590  64 69 75 6d 20 66 69 6c  65 2e 0a 54 68 69 73 20  |dium file..This |
000005a0  69 73 20 61 20 6d 65 64  69 75 6d 20 66 69 6c 65  |is a medium file|
000005b0  2e 0a 54 68 69 73 20 69  73 20 61 20 6d 65 64 69  |..This is a medi|
000005c0  75 6d 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |um file..This is|
000005d0  20 61 20 6d 65 64 69 75  6d 20 66 69 6c 65 2e 0a  | a medium file..|
000005e0  54 68 69 73 20 69 73 20  61 20 6d 65 64 69 75 6d  |This is a medium|
000005f0  20 66 69 6c 65 2e 0a 54  68 69 73 20 69 73 20 61  | file..This is a|
00000600  20 6d 65 64 69 75 6d 20  66 69 6c 65 2e 0a 54 68  | medium file..Th|
00000610  69 73 20 69 73 20 61 20  6d 65 64 69 75 6d 20 66  |is is a medium f|
00000620  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 6d  |ile..This is a m|
00000630  65 64 69 75 6d 20 66 69  6c 65 2e 0a 54 68 69 73  |edium file..This|
00000640  20 69 73 20 61 20 6d 65  64 69 75 6d 20 66 69 6c  | is a medium fil|
00000650  65 2e 0a 54 68 69 73 20  69 73 20 61 20 6d 65 64  |e..This is a med|
00000660  69 75 6d 20 66 69 6c 65  2e 00 00 00 00 00 00 00  |ium file........|
00000670  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

(12) ./nachos -ap test/big small

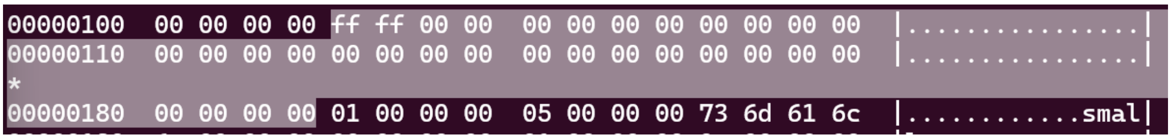
(13) ./nachos -D



```
Name: small, Sector: 5
FileHeader contents.  File size: 840.  File blocks:
6 7 8 9 13 14 15
File contents:
This is a small file.This is a small file.This is a big file.\aThis is a big file
.\aThis is a big file.\aThis is a big file.\aThis i
s a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThi
s is a big file.\aThis is a big file.\aThis is a big
file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThis is a b
ig file.\aThis is a big file.\aThis is a big file.\aTh
is is a big file.\aThis is a big file.\aThis is a big file.This is a big file.\aT
his is a big file.\aThis is a big file.\aThis is a b
ig file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThis is
a big file.\aThis is a big file.\aThis is a big file.
\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThis is a big fi
le.\aThis is a big file.\aThis is a big file.\aThis is
 a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.
```

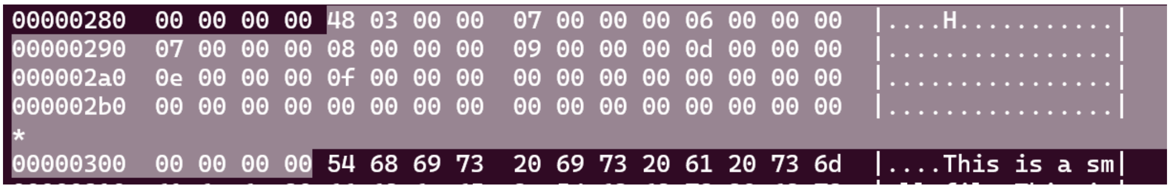
hexdump -C DISK

2 号扇区：表明多启用了 13、14、15 号扇区



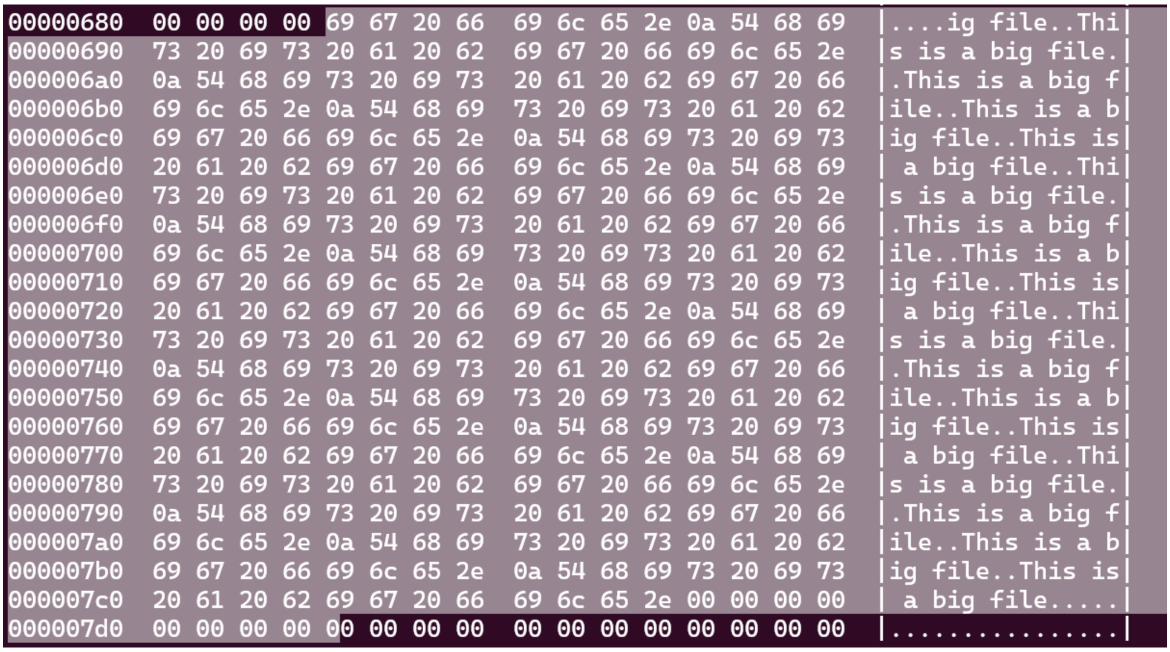
```
00000100  00 00 00 00 ff ff 00 00  00 00 00 00 00 00 00 00  |................|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000180  00 00 00 00 01 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
```

5 号扇区：medium 有 0x0348Bytes，7 个扇区，6、7、8、9、13、14、15 号扇区



```
00000280  00 00 00 00 48 03 00 00  07 00 00 00 06 00 00 00  |....H...........|
00000290  07 00 00 00 08 00 00 00  09 00 00 00 0d 00 00 00  |................|
000002a0  0e 00 00 00 0f 00 00 00  00 00 00 00 00 00 00 00  |................|
000002b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000300  00 00 00 00 54 68 69 73  20 69 73 20 61 20 73 6d  |....This is a sm|
```

13、14、15 号扇区：文件内容

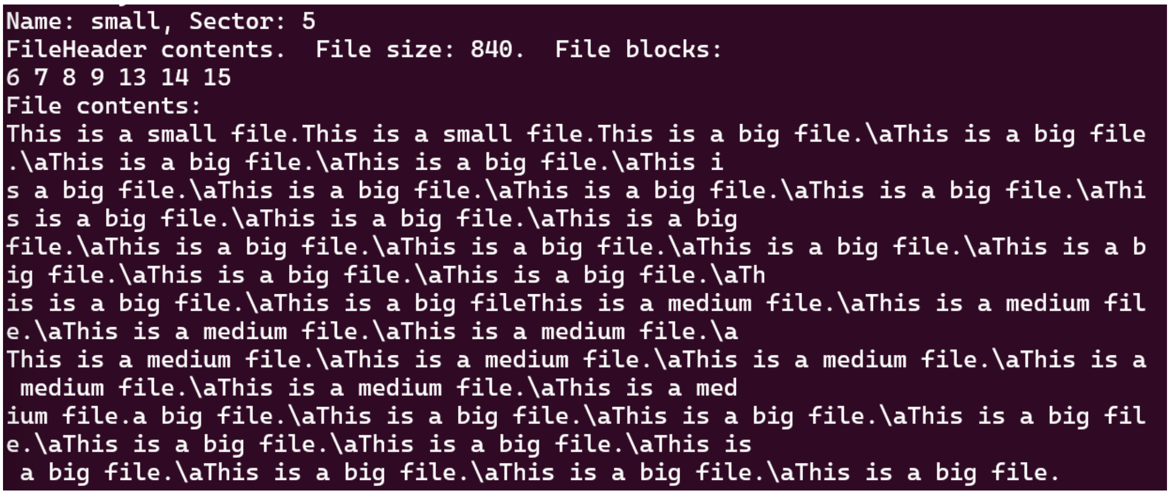


```
00000680  00 00 00 00 69 67 20 66  69 6c 65 2e 0a 54 68 69  |....ig file..Thi|
00000690  73 20 69 73 20 61 20 62  69 67 20 66 69 6c 65 2e  |s is a big file.|
000006a0  0a 54 68 69 73 20 69 73  20 61 20 62 69 67 20 66  |.This is a big f|
000006b0  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 62  |ile..This is a b|
000006c0  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
000006d0  20 61 20 62 69 67 20 66  69 6c 65 2e 0a 54 68 69  | a big file..Thi|
000006e0  73 20 69 73 20 61 20 62  69 67 20 66 69 6c 65 2e  |s is a big file.|
000006f0  0a 54 68 69 73 20 69 73  20 61 20 62 69 67 20 66  |.This is a big f|
00000700  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 62  |ile..This is a b|
00000710  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
00000720  20 61 20 62 69 67 20 66  69 6c 65 2e 0a 54 68 69  | a big file..Thi|
00000730  73 20 69 73 20 61 20 62  69 67 20 66 69 6c 65 2e  |s is a big file.|
00000740  0a 54 68 69 73 20 69 73  20 61 20 62 69 67 20 66  |.This is a big f|
00000750  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 62  |ile..This is a b|
00000760  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
00000770  20 61 20 62 69 67 20 66  69 6c 65 2e 0a 54 68 69  | a big file..Thi|
00000780  73 20 69 73 20 61 20 62  69 67 20 66 69 6c 65 2e  |s is a big file.|
00000790  0a 54 68 69 73 20 69 73  20 61 20 62 69 67 20 66  |.This is a big f|
000007a0  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 62  |ile..This is a b|
000007b0  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
000007c0  20 61 20 62 69 67 20 66  69 6c 65 2e 00 00 00 00  | a big file.....|
000007d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

(14) ./nachos -hap test/medium small

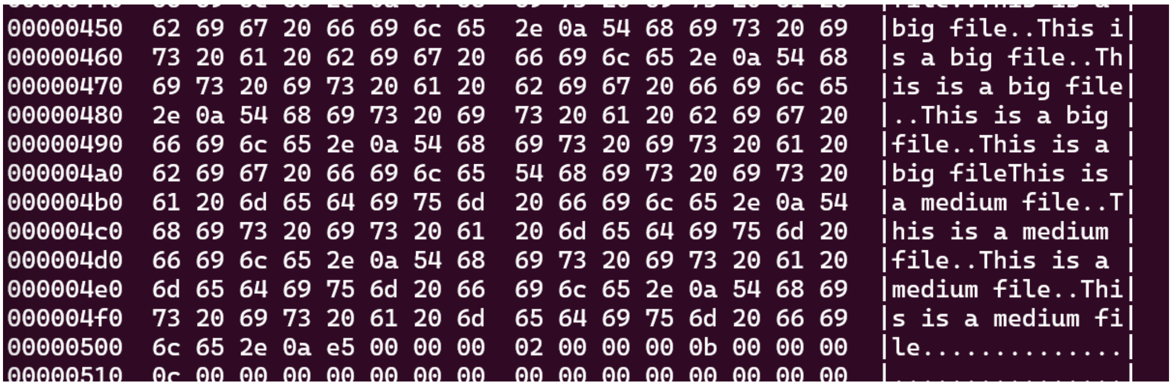
(15) ./nachos -D

确认系统是在 small 的中间写入文件 test/medium 的内容



```
Name: small, Sector: 5
FileHeader contents.  File size: 840.  File blocks:
6 7 8 9 13 14 15
File contents:
This is a small file.This is a small file.This is a big file.\aThis is a big file
.\aThis is a big file.\aThis is a big file.\aThis i
s a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThi
s is a big file.\aThis is a big file.\aThis is a big
file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThis is a b
ig file.\aThis is a big file.\aThis is a big file.\aTh
is is a big file.\aThis is a big fileThis is a medium file.\aThis is a medium fil
e.\aThis is a medium file.\aThis is a medium file.\a
This is a medium file.\aThis is a medium file.\aThis is a medium file.\aThis is a
 medium file.\aThis is a medium file.\aThis is a med
ium file.a big file.\aThis is a big file.\aThis is a big file.\aThis is a big fil
e.\aThis is a big file.\aThis is a big file.\aThis is
 a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.
```

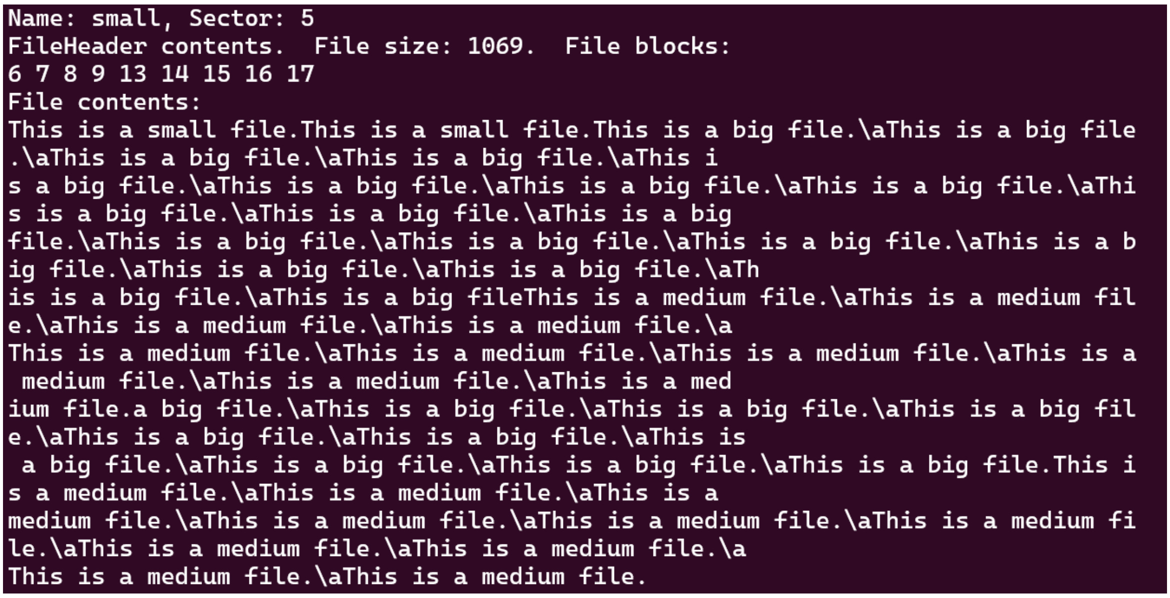
hexdump -C DISK



```
00000450  62 69 67 20 66 69 6c 65  2e 0a 54 68 69 73 20 69  |big file..This i|
00000460  73 20 61 20 62 69 67 20  66 69 6c 65 2e 0a 54 68  |s a big file..Th|
00000470  69 73 20 69 73 20 61 20  62 69 67 20 66 69 6c 65  |is is a big file|
00000480  2e 0a 54 68 69 73 20 69  73 20 61 20 62 69 67 20  |..This is a big |
00000490  66 69 6c 65 2e 0a 54 68  69 73 20 69 73 20 61 20  |file..This is a |
000004a0  62 69 67 20 66 69 6c 65  54 68 69 73 20 69 73 20  |big fileThis is |
000004b0  61 20 6d 65 64 69 75 6d  20 66 69 6c 65 2e 0a 54  |a medium file..T|
000004c0  68 69 73 20 69 73 20 61  20 6d 65 64 69 75 6d 20  |his is a medium |
000004d0  66 69 6c 65 2e 0a 54 68  69 73 20 69 73 20 61 20  |file..This is a |
000004e0  6d 65 64 69 75 6d 20 66  69 6c 65 2e 0a 54 68 69  |medium file..Thi|
000004f0  73 20 69 73 20 61 20 6d  65 64 69 75 6d 20 66 69  |s is a medium fi|
00000500  6c 65 2e 0a e5 00 00 00  02 00 00 00 0b 00 00 00  |le..............|
00000510  0c 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

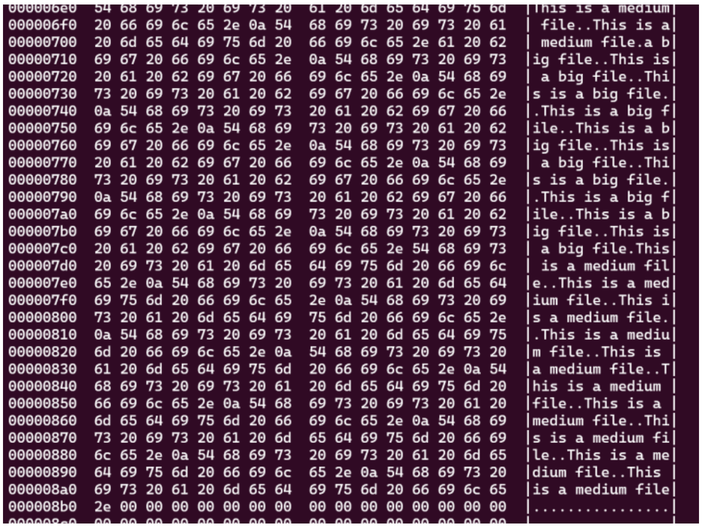
(16) ./nachos -nap medium small 测试将一个 nachos 文件附加到另一个 nachos 文件的功能

(17) ./nachos -D



```
Name: small, Sector: 5
FileHeader contents.  File size: 1069.  File blocks:
6 7 8 9 13 14 15 16 17
File contents:
This is a small file.This is a small file.This is a big file.\aThis is a big file
.\aThis is a big file.\aThis is a big file.\aThis i
s a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThi
s is a big file.\aThis is a big file.\aThis is a big
file.\aThis is a big file.\aThis is a big file.\aThis is a big file.\aThis is a b
ig file.\aThis is a big file.\aThis is a big file.\aTh
is is a big file.\aThis is a big fileThis is a medium file.\aThis is a medium fil
e.\aThis is a medium file.\aThis is a medium file.\a
This is a medium file.\aThis is a medium file.\aThis is a medium file.\aThis is a
 medium file.\aThis is a medium file.\aThis is a med
ium file.a big file.\aThis is a big file.\aThis is a big file.\aThis is a big fil
e.\aThis is a big file.\aThis is a big file.\aThis is
 a big file.\aThis is a big file.\aThis is a big file.\aThis is a big file.This i
s a medium file.\aThis is a medium file.\aThis is a
medium file.\aThis is a medium file.\aThis is a medium file.\aThis is a medium fi
le.\aThis is a medium file.\aThis is a medium file.\a
This is a medium file.\aThis is a medium file.
```

hexdump -C DISK

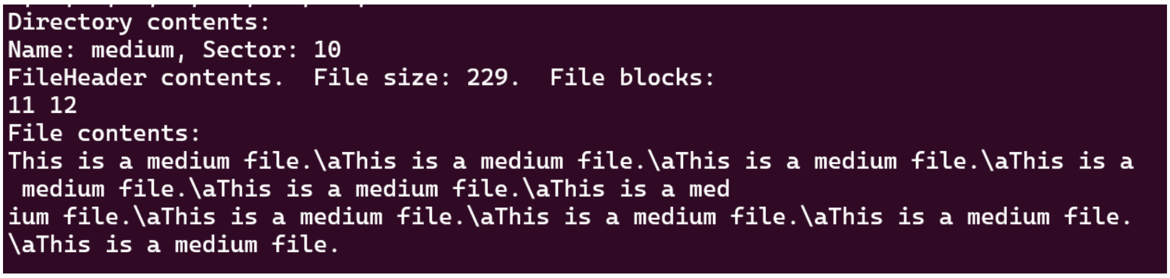


```
000006e0  54 68 69 73 20 69 73 20  61 20 6d 65 64 69 75 6d  |This is a medium|
000006f0  20 66 69 6c 65 2e 0a 54  68 69 73 20 69 73 20 61  | file..This is a|
00000700  20 6d 65 64 69 75 6d 20  66 69 6c 65 2e 61 20 62  | medium file.a b|
00000710  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
00000720  20 61 20 62 69 67 20 66  69 6c 65 2e 0a 54 68 69  | a big file..Thi|
00000730  73 20 69 73 20 61 20 62  69 67 20 66 69 6c 65 2e  |s is a big file.|
00000740  0a 54 68 69 73 20 69 73  20 61 20 62 69 67 20 66  |.This is a big f|
00000750  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 62  |ile..This is a b|
00000760  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
00000770  20 61 20 62 69 67 20 66  69 6c 65 2e 0a 54 68 69  | a big file..Thi|
00000780  73 20 69 73 20 61 20 62  69 67 20 66 69 6c 65 2e  |s is a big file.|
00000790  0a 54 68 69 73 20 69 73  20 61 20 62 69 67 20 66  |.This is a big f|
000007a0  69 6c 65 2e 0a 54 68 69  73 20 69 73 20 61 20 62  |ile..This is a b|
000007b0  69 67 20 66 69 6c 65 2e  0a 54 68 69 73 20 69 73  |ig file..This is|
000007c0  20 61 20 62 69 67 20 66  69 6c 65 2e 54 68 69 73  | a big file.This|
000007d0  20 69 73 20 61 20 6d 65  64 69 75 6d 20 66 69 6c  | is a medium fil|
000007e0  65 2e 0a 54 68 69 73 20  69 73 20 61 20 6d 65 64  |e..This is a med|
000007f0  69 75 6d 20 66 69 6c 65  2e 0a 54 68 69 73 20 69  |ium file..This i|
00000800  73 20 61 20 6d 65 64 69  75 6d 20 66 69 6c 65 2e  |s a medium file.|
00000810  0a 54 68 69 73 20 69 73  20 61 20 6d 65 64 69 75  |.This is a mediu|
00000820  6d 20 66 69 6c 65 2e 0a  54 68 69 73 20 69 73 20  |m file..This is |
00000830  61 20 6d 65 64 69 75 6d  20 66 69 6c 65 2e 0a 54  |a medium file..T|
00000840  68 69 73 20 69 73 20 61  20 6d 65 64 69 75 6d 20  |his is a medium |
00000850  66 69 6c 65 2e 0a 54 68  69 73 20 69 73 20 61 20  |file..This is a |
00000860  6d 65 64 69 75 6d 20 66  69 6c 65 2e 0a 54 68 69  |medium file..Thi|
00000870  73 20 69 73 20 61 20 6d  65 64 69 75 6d 20 66 69  |s is a medium fi|
00000880  6c 65 2e 0a 54 68 69 73  20 69 73 20 61 20 6d 65  |le..This is a me|
00000890  64 69 75 6d 20 66 69 6c  65 2e 0a 54 68 69 73 20  |dium file..This |
000008a0  69 73 20 61 20 6d 65 64  69 75 6d 20 66 69 6c 65  |is a medium file|
000008b0  2e 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

Small 多用了扇区 16、17，确认追加成功

(18) ./nachos -r small

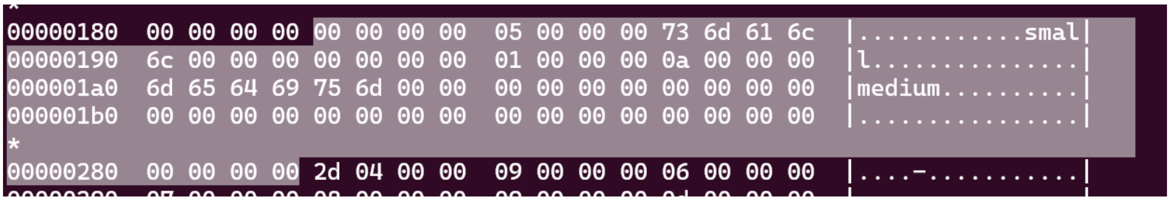
(19) ./nachos -D



```
Directory contents:
Name: medium, Sector: 10
FileHeader contents.  File size: 229.  File blocks:
11 12
File contents:
This is a medium file.\aThis is a medium file.\aThis is a medium file.\aThis is a
 medium file.\aThis is a medium file.\aThis is a med
ium file.\aThis is a medium file.\aThis is a medium file.\aThis is a medium file.
\aThis is a medium file.
```

发现 small 文件被删除

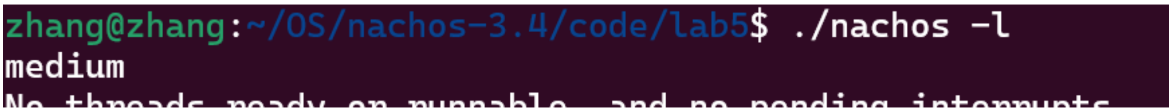
hexdump -C DISK



```
00000180  00 00 00 00 00 00 00 00  05 00 00 00 73 6d 61 6c  |............smal|
00000190  6c 00 00 00 00 00 00 00  01 00 00 00 0a 00 00 00  |l...............|
000001a0  6d 65 64 69 75 6d 00 00  00 00 00 00 00 00 00 00  |medium..........|
000001b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000280  00 00 00 00 2d 04 00 00  09 00 00 00 06 00 00 00  |....-...........|
```

发现，关于 small 文件的目录项和相关文件内容仍然存储在 DISK 中，但是 small 文件目录项中的 inUser 变为了 0，表明没有该文件，相关扇区在位示图中也被置 0

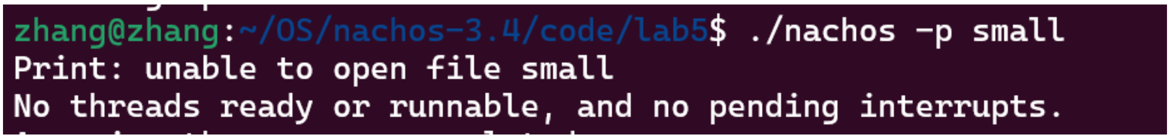
(20) ./nachos -l



```
zhang@zhang:~/OS/nachos-3.4/code/lab5$ ./nachos -l
medium
No threads ready or runnable, and no pending interrupts
```

发现只能找到文件 medium

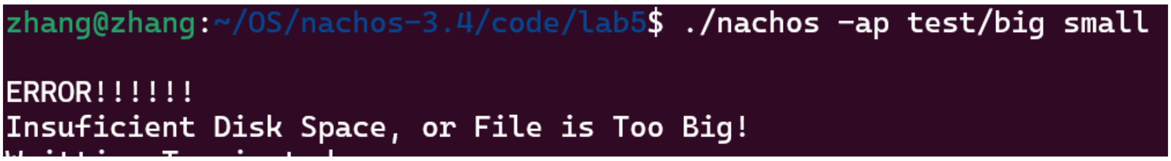
./nachos -p small



```
zhang@zhang:~/OS/nachos-3.4/code/lab5$ ./nachos -p small
Print: unable to open file small
No threads ready or runnable, and no pending interrupts.
```

发现找不到文件 small，符合预期

(21)反复运行./nachos -ap test/big small



```
zhang@zhang:~/OS/nachos-3.4/code/lab5$ ./nachos -ap test/big small

ERROR!!!!!!
Insuficient Disk Space, or File is Too Big!
```

运行到第 10 遍时，系统提示“Insuficient Disk Space, or File is Too Big!Writting Terminated.”表明文件过大，已经超出了 Nachos 所允许的文件大小

(22)反复运行 nachos -ap 或 nachos -cp 在硬盘 DISK 上新建文件，测试 nachos 文件系统中最多可创建多少个文件

答：当我初始化磁盘，并反复运行：

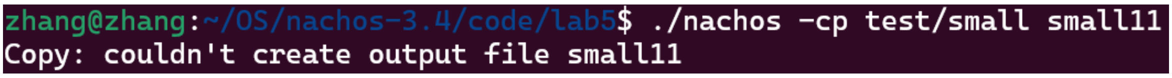
./nachos -cp test/small small1

./nachos -cp test/small small2

./nachos -cp test/small small3

……

发现运行到./nachos -cp test/small small11 时系统报错：



```
zhang@zhang:~/OS/nachos-3.4/code/lab5$ ./nachos -cp test/small small11
Copy: couldn't create output file small11
```

证明 nachos 确实只能容纳 10 个文件

任务及收获

目前 Nachos 实现的文件系统存在诸多限制，其中之一是文件大小不能扩展，即无法在已经存在的文件尾部追加数据；

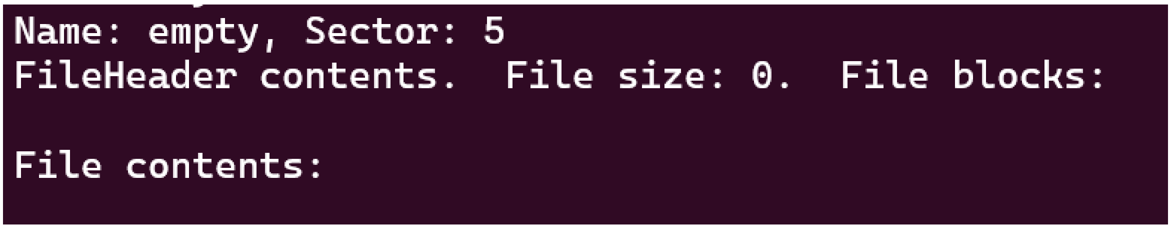
该实验的任务就是让你修改 Nachos 的文件系统，以满足：

(1)文件创建时，其大小可初始化为 0；

(2)当一个文件写入更多的数据时，其大小可随之增大；

(3)要求能够在从一个文件的任何位置开始写入数据，即能够正确处理命令行参数-ap,-hap,及-nap；

(1)如下，我已可以复制大小为 0 的 empty 文件



```
Name: empty, Sector: 5
FileHeader contents.  File size: 0.  File blocks:

File contents:
```

(2)(3)在 6.4.3 nachos 文件系统测试中已得到验证

收获：

读懂了 Nachos 的文件系统，包括文件的创造，扩展，复制等操作

实现了 nachos -ap，nachos -hap，nachos -nap 的作用

修改了

openfile.h 中的 OpenFile 类

filehdr.h 中的 FileHeader 类

filesystem.h 中的 FileSystem 类

fstest.cc 中的 Append()函数及 NAppend()函数

进行了如下修改：

①修改 openfile.cc 中的 OpenFile::WriteAt()允许从文件尾部开始写数据，并可为要写入的数据分配新的扇区；

②修改 FileSystem 类，添加空闲块位示图文件的硬盘读写操作；

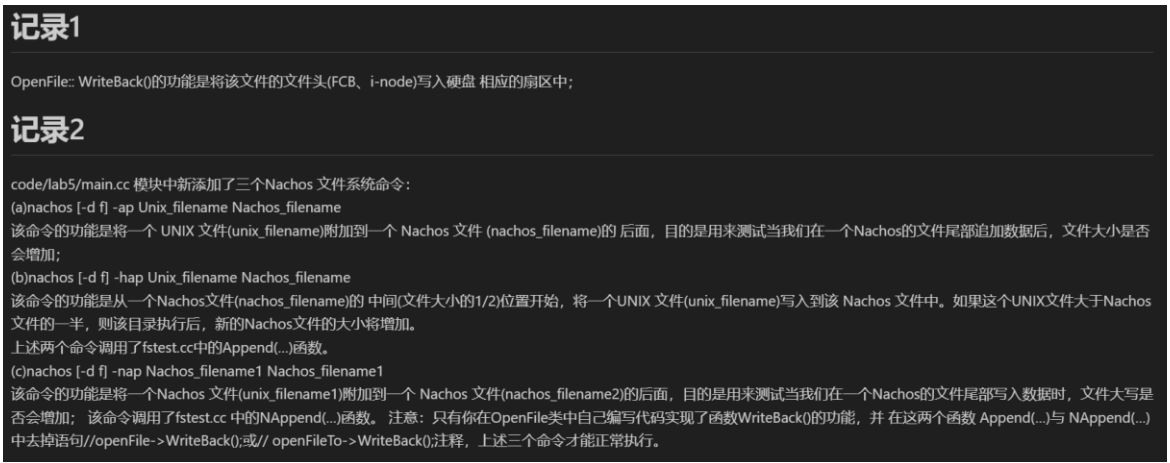
③修改 OpenFile::OpenFile()及 OpenFile::WriteBack()，实现文件头的硬盘读写；

④修改 FileHeader::Allocate()，为添加的数据分配硬盘块（扇区）；

⑤修改 fstest.cc 的 Append()函数，使下次的写指针指向新写入数据的尾部，并在扩展操作结束后调用 OpenFile::WriteBack()将修改后的文件头写入硬盘；

(3)在那些需要修改的模块中，是否需要添加函数与变量；

README.md

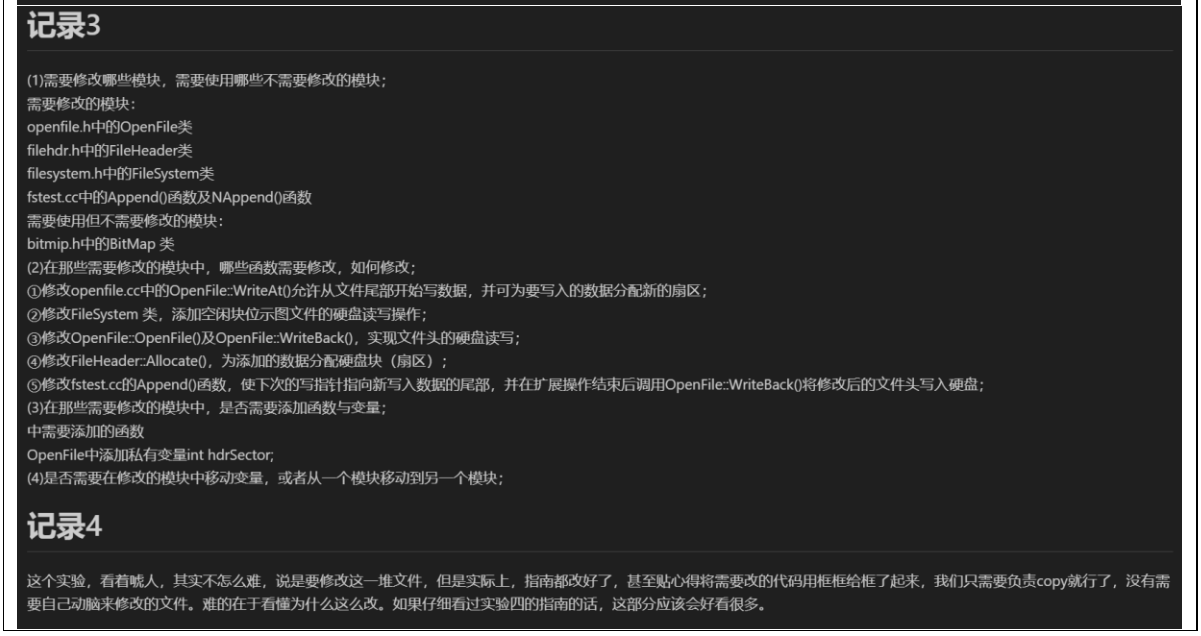


记录1

OpenFile:: WriteBack()的功能是将该文件的文件头(FCB、i-node)写入硬盘 相应的扇区中；

记录2

code/lab5/main.cc 模块中新添加了三个Nachos 文件系统命令：
(a)nachos [-d f] -ap Unix_filename Nachos_filename
该命令的功能是将一个 UNIX 文件(unix_filename)附加到一个 Nachos 文件 (nachos_filename)的 后面，目的是用来测试当我们在一个Nachos的文件尾部追加数据后，文件大小是否会增加；
(b)nachos [-d f] -hap Unix_filename Nachos_filename
该命令的功能是从一个Nachos文件(nachos_filename)的 中间(文件大小的1/2)位置开始，将一个UNIX 文件(unix_filename)写入到该 Nachos 文件中。如果这个UNIX文件大于Nachos文件的一半，则该目录执行后，新的Nachos文件的大小将增加。
上述两个命令调用了fstest.cc中的Append(...)函数。
(c)nachos [-d f] -nap Nachos_filename1 Nachos_filename1
该命令的功能是将一个Nachos 文件(unix_filename1)附加到一个 Nachos 文件(nachos_filename2)的后面，目的是用来测试当我们在一个Nachos的文件尾部写入数据时，文件大写是否会增加； 该命令调用了fstest.cc 中的NAppend(...)函数。 注意：只有你在OpenFile类中自己编写代码实现了函数WriteBack()的功能，并 在这两个函数 Append(...)与 NAppend(...)中去掉语句//openFile->WriteBack();或// openFileTo->WriteBack();注释，上述三个命令才能正常执行。



记录3

(1)需要修改哪些模块，需要使用哪些不需要修改的模块；
需要修改的模块：
openfile.h中的OpenFile类
filehdr.h中的FileHeader类
filesystem.h中的FileSystem类
fstest.cc中的Append()函数及NAppend()函数
需要使用但不需要修改的模块：
bitmip.h中的BitMap 类
(2)在那些需要修改的模块中，哪些函数需要修改，如何修改；
①修改openfile.cc中的OpenFile::WriteAt()允许从文件尾部开始写数据，并可为要写入的数据分配新的扇区；
②修改FileSystem 类，添加空闲块位示图文件的硬盘读写操作；
③修改OpenFile::OpenFile()及OpenFile::WriteBack()，实现文件头的硬盘读写；
④修改FileHeader::Allocate()，为添加的数据分配硬盘块（扇区）；
⑤修改fstest.cc的Append()函数，使下次的写指针指向新写入数据的尾部，并在扩展操作结束后调用OpenFile::WriteBack()将修改后的文件头写入硬盘；
(3)在那些需要修改的模块中，是否需要添加函数与变量；
中需要添加的函数
OpenFile中添加私有变量int hdrSector;
(4)是否需要在修改的模块中移动变量，或者从一个模块移动到另一个模块；

记录4

这个实验，看着唬人，其实不怎么难，说是要修改这一堆文件，但是实际上，指南都改好了，甚至贴心得将需要改的代码用框框给框了起来，我们只需要负责copy就行了，没有需要自己动脑来修改的文件。难的在于看懂为什么这么改。如果仔细看过实验四的指南的话，这部分应该会好看很多。

拓展：多级目录

拓展(选做)：目前 Nachos 文件系统仅仅实现了单级目录结构，只有一个根目录。可以尝试采用目录树对文件进行管理。
答：暂时未实现