Windows操作系统

C/C++ 程序实验

首都师范大学 信息工程学院

系统分析与管理实验室

## 实验十 Windows 磁盘I/O

### 一、实验目的

  本练习着重于了解磁盘的物理组织，以及如何通过用户态的程序直接读写磁盘上的信息，并具体针对磁盘进行练习，练习的具体过程：

1、判定逻辑驱动器C中磁盘的基本信息。

2、根据给定的物理扇区号读取磁盘的扇区。

3、把磁盘上得到的信息输出到标准输出流中。

要求：

编写程序调用physicalDisk、sectorRead和segmentDump三个函数，并验证其正确性。

HANDLE physicalDisk(char deviceLetter)；

int sectorRead(HANDLE hdevice，unsigned int logicalSectorNumber，void \*buffer)；

void sectorDump(HANDLE hdevice，unsigned int logicalSectorNumber)；

a)根据输入的驱动器号读取磁盘基本信息。

b)根据输入的扇区号读扇区内容。

c)输出所读扇区的内容。

也就是要实现查看磁盘信息、查看指定扇区内容的作用。

### 二、工具/准备工作

在开始本实验之前，请回顾教科书的相关内容。

需要做以下准备：

1) 一台运行Windows 操作系统的计算机。

2) 计算机中需安装Microsoft Visual Studio Code。

### 三、实验内容与步骤

#### 1. 磁盘I/O

**步骤1**：登录进入Windows 。

**步骤2**：在“开始”菜单中单击“程序”-“Microsoft Visual Studio Code”。

**步骤3：**新建项目名为“10-1”，并且新建项“10-1.cpp”。

**清单10-1 磁盘I/O**

#include <windows.h>

#include <iostream>

#include <winioctl.h>

#include <string.h>

using namespace std;

struct Disk //关于Disk结构的定义

{

HANDLE handle;

DISK\_GEOMETRY disk\_info;

};

Disk disk;

HANDLE Floppy;

static \_int64 sector;

bool flag;

Disk physicDisk(char driverLetter); //函数声明

bool sectorRead(Disk theDisk, unsigned logicSectorNumber, char\* buffer); //函数声明

void sectorDump(Disk theDisk, int logicSectorNumber); //函数声明

void main(void)

{

int secNum;

char choise;

disk = physicDisk('c'); //选择要查看的磁盘

if (flag)

{

while (1)

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\* 0. 查看指定目录信息: \*" << endl;

cout << "\* 1. 退出: \*" << endl;

cin >> choise;

switch (choise)

{

case '0': //查看指定的扇区

cout << "总共有" << (long)sector << "个扇区!" << endl;

cout << "请输入要查看的扇区号码:" << endl;

cin >> secNum;

if (secNum<sector && secNum >= 0)

sectorDump(disk, secNum);

else

cout << "没有这个扇区!!!" << endl;

//exit

break;

case '1':

exit(0);

break;

default:

cout << "输入错误!!!请重新输入!!!" << endl;

cout << endl;

}

}

}

}

Disk physicDisk(char driverLetter) //创建磁盘句柄，并打开磁盘，返回disk

{

flag = true;

DISK\_GEOMETRY\* temp = new DISK\_GEOMETRY;

char device[9] = "\\\\.\\C:";

device[4] = driverLetter;

Floppy = CreateFile(device, //将要打开的驱动器名

GENERIC\_READ, //存取的权限

FILE\_SHARE\_READ | FILE\_SHARE\_WRITE,//共享的权限

NULL, //默认的属性位

OPEN\_EXISTING, //创建驱动器方式

0, //所创建的驱动器类型

NULL); //指向模板文件的类型

if (GetLastError() == ERROR\_ALREADY\_EXISTS) //打开失败，返回错误代码

{

cout << "不能打开磁盘" << endl;

cout << GetLastError() << endl;

flag = false;

return disk;

}

DWORD bytereturned;

BOOL Result;

disk.handle = Floppy;

//获得物理磁盘的几何信息

Result = DeviceIoControl(Floppy,

IOCTL\_DISK\_GET\_DRIVE\_GEOMETRY,

NULL,

0,

temp,

sizeof(\*temp),

&bytereturned,

(LPOVERLAPPED)NULL);

if (!Result) //如果失败，返回错误代码

{

cout << "打开失败" << endl;

cout << "错误代码位:" << GetLastError() << endl;

flag = false;

return disk;

}

disk.disk\_info = \*temp;

//输出的是整个物理磁盘的信息

cout << "柱面数为:" << (unsigned long)disk.disk\_info.Cylinders.QuadPart << endl << endl;

cout << "每柱面的磁道数为:" << disk.disk\_info.TracksPerCylinder << endl << endl;

cout << "没磁道的扇区数为:" << disk.disk\_info.SectorsPerTrack << endl << endl;

cout << "每扇区的字节数为:" << disk.disk\_info.BytesPerSector << endl << endl;

sector = disk.disk\_info.Cylinders.QuadPart\* (disk.disk\_info.TracksPerCylinder)\*

(disk.disk\_info.SectorsPerTrack);

double DiskSize = (double)disk.disk\_info.Cylinders.QuadPart\* //相乘获得磁盘大小

(disk.disk\_info.TracksPerCylinder)\*

(disk.disk\_info.SectorsPerTrack)\*

(disk.disk\_info.BytesPerSector);

cout << "磁盘大小为:" << DiskSize / (1024 \* 1024) << "MB." << endl;

cout << driverLetter << "盘所在的磁盘共有" << (long)sector << "个扇区" << endl;

delete temp;

return disk;

}

void sectorDump(Disk theDisk, int logicSectorNumber) //如果读成功，输出读的结果

{

BOOL sDresult;

char\* buffer = new char[theDisk.disk\_info.BytesPerSector];

sDresult = sectorRead(disk, logicSectorNumber, buffer);

if (sDresult)

{

for (int i = 0; i<int(theDisk.disk\_info.BytesPerSector); i++)

cout << buffer[i];

cout << endl; //换行

cout << endl;

cout << endl;

}

delete buffer;

}

bool sectorRead(Disk theDisk, unsigned logicSectorNumber, char\* buffer) //读指定的扇区

{

HANDLE handle1 = theDisk.handle;

BOOL sDresult, Rresult;

DWORD numberofread;

//定位读扇区的位置

sDresult = SetFilePointer(handle1,

theDisk.disk\_info.BytesPerSector\* logicSectorNumber,

NULL,

FILE\_BEGIN);

//读扇区，并返回读的结果

Rresult = ReadFile(handle1,

buffer,

theDisk.disk\_info.BytesPerSector,

&numberofread,

NULL);

if (Rresult) //如果读扇区成功

return true;

else //如果失败,返回错误代码

{

cout << "读指定的扇区失败" << endl;

cout << "错误代码为:" << GetLastError() << endl;

return false;

}

}

**步骤4：**按“F5”开始调试，注意路径里不要含有中文。

**步骤5：**按暂停按钮可暂停程序的执行，按终止按钮可终止程序的执行。



操作能否正常进行？如果不行，则可能的原因是什么？

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

运行结果是：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

根据结果分析磁盘物理结构：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**选作**：

1、判定逻辑驱动器D盘的基本信息，根据结果分析与实际分区属性是否一致，为什么？

2、尝试修改程序实现输出内容的有效显示。