**山东师范大学**

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 院系部所 | 信息科学与工程学院 | | 课程名称 | | 计算机操作系统 |
| 所在班级 | 计工本1702 | 学号 | 201711010202 | 姓名 | 王汝芸 |
| 实验编号 | 11 | 题目 | 获取磁盘基本信息 | | |

一、实验目的和要求

**实验目的：**

1. 了解磁盘的物理组织。
2. 熟悉Windows系统如何查看磁盘相关参数。
3. 掌握Windows系统提供的有关对磁盘操作API函数。

**实验要求：**

了解MSDN Library Visual Studio 6.0中提供的磁盘主要数据结构DISK\_GEOMETRY中每个成员的含义，深入理解操作系统将设备当做文件处理的特性，理解函数CreateFile( )及DeviceloControl()中每个参数的实际意义并能在本实验中正确使用。

二、实验环境

Windows 10

Visual C++ 6.0

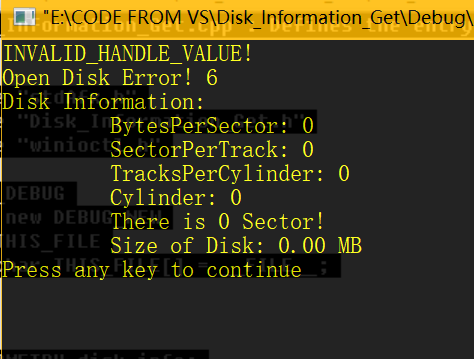
三、实验内容及实施

**实验内容：**

编写一个函数，根据给出的驱动器号读取磁盘基本信息，包括磁盘的大小、该磁盘包括多少个扇区、该磁盘有多少个柱面，以及每个柱面的磁道数、每个磁道的扇区数、每个扇区包含的字节数。

本实验要使用的主要数据结构DISK\_GEOMETRY是由系统提供的，其声明在“#include “winioctl.h””中，因此要将其加入到实验程序的头文件说明中，否则程序编译时系统将无法识别DISK\_GEOMETRY结构

四、实验结果



五、实验分析

分析实验指导书代码及结果得出结论如下：

磁盘上有一部分空间是存储磁盘的物理信息的，这部分空间系统是不能够直接存取的，因此没有编入逻辑扇区。也就是说逻辑扇区比磁盘的实际扇区数要小，因此计算出的磁盘大小是磁盘可以空间的大小，比磁盘的物理大小要小。

六、实验讨论（可选）

该实验结果与实验指导书结果（下图）



并不相同，GetLastError()返回值为6—查表知为句柄错误；

排查掉代码语法错误后，怀疑是因为电脑固态硬盘的原因，不存在机械硬盘的物理构造，然后用舍友既有机械硬盘又有固态硬盘的电脑，在机械硬盘区建立工程，运行相同的代码，得到相同的错误GetLastError()—6；

在网上查询无果，至此，实验中止，未得到目标结果。

七、附录 （可选）

1 *// Disk\_Information\_Get.cpp : Defines the entry point for the console application.*

2 *//*

3

4 #include "stdAfx.h"

5 #include "Disk\_Information\_Get.h"

6 #include "winioctl.h"

7

8 #ifdef \_DEBUG

9 #define new DEBUG\_NEW

10 #undef THIS\_FILE

11 **static char** THIS\_FILE[] = \_\_FILE\_\_;

12 #endif

13

14

15 DISK\_GEOMETRY disk\_info;

16 HANDLE **GetDiskInformation**(**char** drivername);

17 */////////////////////////////////////////////////////////////////////////////*

18 *// The one and only application object*

19

20 CWinApp theApp;

21 **using namespace** std;

22

23 **int** **\_tmain**(**int** argc, TCHAR\* argv[], TCHAR\* envp[])

24 {

25 **int** nRetCode = 0;

26 HANDLE Handle;

27 Handle = **GetDiskInformation**('A');

28 **return** nRetCode;

29 }

30

31 HANDLE **GetDiskInformation**(**char** drivername)

32 {

33 **char** device[]="\\\\.\\: ";

34 device[4]=drivername;

35 HANDLE FloopyDisk;

36 DWORD ReturnSize;

37 DWORD Sector;

38 **double** DiskSize;

39 FloopyDisk=**CreateFile**(device,

40 GENERIC\_READ|GENERIC\_WRITE,

41 FILE\_SHARE\_READ|FILE\_SHARE\_WRITE,

42 NULL,

43 OPEN\_EXISTING,

44 FILE\_FLAG\_RANDOM\_ACCESS|FILE\_FLAG\_NO\_BUFFERING,

45 NULL);

46 **if**(FloopyDisk==INVALID\_HANDLE\_VALUE)

47 **printf**("INVALID\_HANDLE\_VALUE!\n");

48 **if**(**GetLastError**()==ERROR\_ALREADY\_EXISTS)

49 **printf**("Can not Open Disk! %d\n",**GetLastError**());

50 **if**(!**DeviceIoControl**(FloopyDisk,

51 IOCTL\_DISK\_GET\_DRIVE\_GEOMETRY,

52 NULL,

53 0,

54 &disk\_info,

55 50,

56 &ReturnSize,

57 (LPOVERLAPPED)NULL))

58 **printf**("Open Disk Error! %d\n",**GetLastError**());

59 **printf**("Disk Information:\n");

60 **printf**("\t BytesPerSector: %d\n",disk\_info.BytesPerSector);

61 **printf**("\t SectorPerTrack: %d\n",disk\_info.SectorsPerTrack);

62 **printf**("\t TracksPerCylinder: %d\n",disk\_info.TracksPerCylinder);

63 **printf**("\t Cylinder: %d\n",disk\_info.Cylinders);

64 Sector=disk\_info.Cylinders.QuadPart\*

65 disk\_info.TracksPerCylinder\*

66 disk\_info.SectorsPerTrack;

67 **printf**("\t There is %d Sector!\n",Sector);

68 DiskSize=Sector\*disk\_info.BytesPerSector;

69 **printf**("\t Size of Disk: %4.2f MB\n",(DiskSize)/(1024\*1024));

70 **return** FloopyDisk;

71 }