**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 | 12 | 题目 | 读/写磁盘指定位置信息 | | |

一、实验目的和要求

了解磁盘的物理组织；

掌握Windows系统提供的有关对磁盘操作的API函数；

根据输入的扇区号读/写指定扇区

二、实验环境

Windows 10

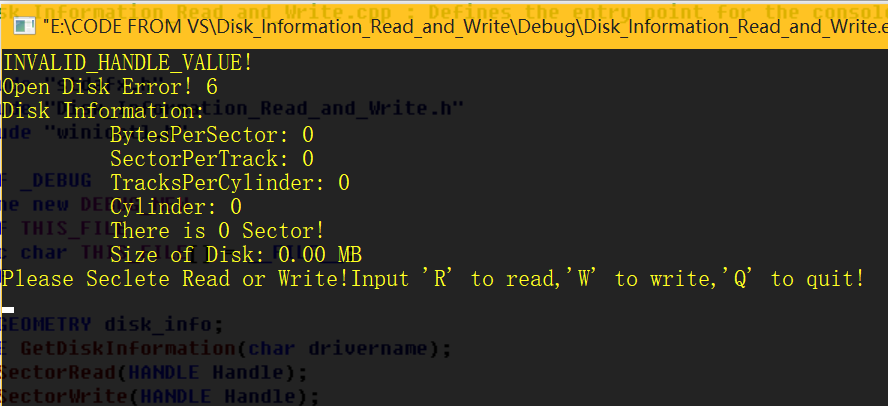
VC++6.0

三、实验内容及实施

在实验11的基础上，继续完成该实验。编写两个函数，分别完成如下功能。

1. 对给定的扇区号读取该扇区的内容。
2. 将用户输入的数据写入指定扇区。

四、实验结果



五、实验分析

在主程序中让用户选择，分别完成不同的操作。R执行SectorRead()，完成读给定扇区的功能，W执行写操作SectorWrite()，完成对给定扇区的写入。

六、实验讨论（可选）

和实验11一样，实验12对于磁盘的读取还是出了错—GetLastError()—6—句柄无效错误

七、附录 （可选）

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

2 *//*

3

4 #include "stdafx.h"

5 #include "Disk\_Information\_Read\_and\_Write.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 DISK\_GEOMETRY disk\_info;

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

16 BOOL **SectorRead**(HANDLE Handle);

17 BOOL **SectorWrite**(HANDLE Handle);

18

19 */////////////////////////////////////////////////////////////////////////////*

20 *// The one and only application object*

21

22 CWinApp theApp;

23 **using namespace** std;

24

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

26 {

27 **int** nRetCode = 0;

28 HANDLE Handle;

29 **char** Choice;

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

31

32 **while**(TRUE)

33 {

34

35 **printf**("Please Seclete Read or Write!Input 'R' to read,'W' to write,'Q' to quit!\n");

36 Choice=**getchar**();

37 **printf**("\n");

38 **switch**(Choice)

39 {

40

41 **case** 'W':

42 {

43 **if** (!**SectorWrite**(Handle)) **printf**("Write Sector Fail!\n");

44 **getchar**();

45 **break**;

46 }

47 **case** 'R':

48 {

49

50 **if**(!**SectorRead**(Handle)) **printf**("Read Sector Fail!\n");

51 **getchar**();

52 **break**;

53 }

54 **case** 'Q':

55 {

56 **exit**(0);

57 **break**;

58 }

59 **default**:

60 {

61 **printf**("Input Error! Try again please!\n");

62 **getchar**();

63 }

64 }

65 }

66 **return** nRetCode;

67 }

68

69

70 HANDLE **GetDiskInformation**(**char** drivername)

71 {

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

73 device[4]=drivername;

74 HANDLE FloopyDisk;

75 DWORD ReturnSize;

76 DWORD Sector;

77 **double** DiskSize;

78 FloopyDisk=**CreateFile**(device,

79 GENERIC\_READ|GENERIC\_WRITE,

80 FILE\_SHARE\_READ|FILE\_SHARE\_WRITE,

81 NULL,

82 OPEN\_EXISTING,

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

84 NULL);

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

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

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

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

89 **if**(!**DeviceIoControl**(FloopyDisk,

90 IOCTL\_DISK\_GET\_DRIVE\_GEOMETRY,

91 NULL,

92 0,

93 &disk\_info,

94 50,

95 &ReturnSize,

96 (LPOVERLAPPED)NULL))

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

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

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

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

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

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

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

104 disk\_info.TracksPerCylinder\*

105 disk\_info.SectorsPerTrack;

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

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

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

109 **return** FloopyDisk;

110 }

111

112 BOOL **SectorRead**(HANDLE Handle)

113 {

114 **char** ReadBuffer[1024\*16];

115 DWORD SectorNumber;

116 DWORD BytestoRead;

117 DWORD Sector;

118 DWORD rc;

119 **int** i;

120 **if**(Handle==NULL)

121 {

122 **printf**("There is No Disk!\n");

123 **return** FALSE;

124 }

125 **printf**("Please Input the Sector Number to Read From: \n");

126 **scanf**("%d",&SectorNumber);

127 **printf**("\n");

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

129 disk\_info.TracksPerCylinder\*

130 disk\_info.SectorsPerTrack;

131 **if**(SectorNumber>Sector) **printf**("There is not this Sector!\n");

132 **printf**("Content:\n");

133 BytestoRead=SectorNumber\*(disk\_info.BytesPerSector);

134 rc=**SetFilePointer**(Handle,BytestoRead,NULL,FILE\_BEGIN);

135 **if**(!**ReadFile**(Handle,ReadBuffer,BytestoRead,&BytestoRead,NULL))

136 {

137 **printf**("Read File Error: %d\n",**GetLastError**());

138 **return** FALSE;

139 }

140 **printf**("\t Text Content: \n");

141 **for**(i=0;i<512;i++)

142 {

143 **printf**("%c",ReadBuffer[i]);

144 }

145 **printf**("\n");

146 **printf**("\t Hex Text Content: \n");

147 **for**(i=0;i<512;i++)

148 {

149 **printf**("%x",ReadBuffer[i]);

150 **printf**("");

151 }

152 **printf**("\n");

153 **return** TRUE;

154 }

155

156 BOOL **SectorWrite**(HANDLE Handle)

157 {

158 **char** WriteBuffer[1024];

159 DWORD SectorNumber,SectorMove;

160 DWORD BytestoWrite;

161 DWORD Sector;

162 DWORD rc;

163

164 **if**(Handle==NULL)

165 {

166 **printf**("There is No disk!\n");

167 **return** FALSE;

168 }

169 **printf**("Please Input the Sector Number to Write to: \n");

170 **scanf**("%d",&SectorNumber);

171 **printf**("\n");

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

173 disk\_info.TracksPerCylinder\*

174 disk\_info.SectorsPerTrack;

175 **if**(SectorNumber>Sector) **printf**("There is not this Sector!\n");

176 **printf**("Please Input the Content to Write to Disk A: \n");

177 **scanf**("%s",&WriteBuffer);

178 SectorMove=SectorNumber\*(disk\_info.BytesPerSector);

179 rc=**SetFilePointer**(Handle,SectorMove,NULL,FILE\_BEGIN);

180 **if**(!**WriteFile**(Handle,WriteBuffer,512,&BytestoWrite,NULL))

181 {

182 **printf**("Read File Error: %d\n",**GetLastError**());

183 **return** FALSE;

184 }

185 **printf**("Write Complete!\n");

186 **return** TRUE;

187 }