Донецкий Национальный Технический Университет

Кафедра прикладной математики информатики

Лабораторная работа № 2

по предмету «Исследование схемы сегментации MBR»

Выполнил:

Лысенко А. С.

Проверила:

Шевченко О. Г.

Покровск 2017

Листинг программы

#include<stdio.h>

#include<iostream>

#include<conio.h>

#include<Windows.h>

using namespace std;

BYTE mbr[512];

BYTE ebr[512];

BYTE mbr1[512];

BYTE mbr2[512];

DWORD dwRead;

DISK\_GEOMETRY dg;

HANDLE hDevice;

DWORD ReturnSize;

DWORD DiskSize;

HANDLE hDevice1;

\_\_int64 z = 0x0FFFFFFFF;

int i = 0;

int d = 0;

int a = 0;

int a1 = 0;

int a2 = 0;

\_\_int64 ladd = 0;

\_\_int64 badd = 0;

\_\_int64 madd = 0;

int z1 = 0x048;

//int l = \*(int\*)&mbr[0] >> z1;

\_\_int64 l = 0;

int y = 0;

int y2 = 0;

int b = 0;

int e = 0;

int f = 0;

int k1 = 0;

void showEbr(int a)

{

i = 0;

ladd = a; //

a1 = a;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

//printf("%d\n", a);

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &ebr, 512, &dwRead, NULL);

int d = 0;

//int a = 0;

int b = 0;

while (d < 2)

{

//cout << hex << (int)ebr[446 + i] << endl;

e = (unsigned char)ebr[446 + i];

/\*if (e == 128)

cout << "Загрузочный" << endl;

else

cout << "Не загрузочный" << endl;\*/

//cout << hex << (int)ebr[446 + 4 + i] << endl;

f = (unsigned char)ebr[446 + 4 + i];

if (f == 7 || f == 17)

{

//cout << "Основной раздел" << endl;

cout << "ФС : NTFS\n";

ladd = y; // add func

y += a;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[z1];

cout << "Серийный номер: " << dec << l << endl;

}

else if (f == 11 || f == 12)

{

y = \*(int\*)&ebr[446 + 8 + i];

//cout << "Основной раздел" << endl;

cout << "ФС : FAT32\n";

ladd = y; // add func

y += a;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[67];

cout << "Серийный номер: " << dec << l << endl;

}

else if (f == 1 || f == 4 || f == 13 || f == 14 || f == 20 || f == 22 || f == 23 || f == 27)

{

cout << "Основной раздел" << endl;

cout << "ФС : ?\n";

}

else if (f == 5 || f == 15)

cout << "Расширенный раздел" << endl;

else if (f == 6)

{

cout << "ФС : FAT16" << endl;

y = \*(int\*)&ebr[446 + 8 + i];

ladd = y; // add func

y += a;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[39];

cout << "Серийный номер: " << dec << l << endl;

}

else

cout << "ФС не опознана" << endl;

y = \*(int\*)&ebr[446 + 8 + i];

//a = a / 2 / 1024;

printf("Адрес начала раздела: %d в секторах\n", y);

b = \*(int\*)&ebr[446 + 12 + i];

b = b / 2 / 1024;

printf("Размер диска: %d Мб\n", b);

cout << endl;

i += 16;

d++;

}

cout << endl;

a += y;

ladd = a; //

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &ebr, 512, &dwRead, NULL);

i = 0;

d = 0;

while (d < 2)

{

//cout << hex << (int)ebr[446 + i] << endl;

e = (int)ebr[446 + i];

if (e == 128)

cout << "Загрузочный" << endl;

else

cout << "Не загрузочный" << endl;

//cout << hex << (int)ebr[446 + 4 + i] << endl;

f = (int)ebr[446 + 4 + i];

if (f == 7 || f == 17)

{

//cout << "Основной раздел" << endl;

cout << "ФС : NTFS\n";

}

else if (f == 11 || f == 12)

{

//cout << "Основной раздел" << endl;

cout << "ФС : FAT32\n";

ladd = y; // add func

y += a;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[67];

}

else if (f == 1 || f == 4 || f == 13 || f == 14 || f == 20 || f == 22 || f == 23 || f == 27)

{

cout << "Основной раздел" << endl;

cout << "ФС : ?\n";

}

else if (f == 5 || f == 15)

cout << "Расширенный раздел" << endl;

else if (f == 6)

{

y = \*(int\*)&ebr[446 + 8 + i];

cout << "ФС : FAT16" << endl;

y += a;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[39];

cout << "Серийный номер: " << dec << l << endl;

}

else

cout << "ФС не опознана" << endl;

y = \*(int\*)&ebr[446 + 8 + i];

//a = a / 2 / 1024;

printf("Адрес начала раздела: %d в секторах\n", y);

b = \*(int\*)&ebr[446 + 12 + i];

b = b / 2 / 1024;

printf("Размер диска: %d Мб\n", b);

cout << endl;

i += 16;

d++;

}

cout << endl;

a1 += y;

ladd = a1; //

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &ebr, 512, &dwRead, NULL);

i = 0;

d = 0;

while (d < 2)

{

//cout << hex << (int)ebr[446 + i] << endl;

e = (int)ebr[446 + i];

if (e == 128)

cout << "Загрузочный" << endl;

else

cout << "Не загрузочный" << endl;

//cout << hex << (int)ebr[446 + 4 + i] << endl;

f = (int)ebr[446 + 4 + i];

if (f == 7 || f == 17)

{

y = \*(int\*)&ebr[446 + 8 + i];

//cout << "Основной раздел" << endl;

cout << "ФС : NTFS\n";

ladd = y; // add func

y += a1;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[z1];

cout << "Серийный номер: " << dec << l << endl;

}

else if (f == 11 || f == 12)

{

//cout << "Основной раздел" << endl;

cout << "ФС : FAT32\n";

ladd = y; // add func

y += a1;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[67];

}

else if (f == 1 || f == 4 || f == 13 || f == 14 || f == 20 || f == 22 || f == 23 || f == 27)

{

cout << "Основной раздел" << endl;

cout << "ФС : ?\n";

}

else if (f == 5 || f == 15)

cout << "Расширенный раздел" << endl;

else if (f == 6)

{

y = \*(int\*)&ebr[446 + 8 + i];

cout << "ФС : FAT16" << endl;

y += a1;

ladd = y;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[39];

cout << "Серийный номер: " << dec << l << endl;

}

else

cout << "ФС не опознана" << endl;

y = \*(int\*)&ebr[446 + 8 + i];

//a = a / 2 / 1024;

printf("Адрес начала раздела: %d в секторах\n", y);

b = \*(int\*)&ebr[446 + 12 + i];

b = b / 2 / 1024;

printf("Размер диска: %d Мб\n", b);

cout << endl;

i += 16;

d++;

}

cout << endl;

}

void showDisk()

{

int i = 0;

int d = 0;

int b = 0;

int e = 0;

int f = 0;

int a = 0;

char s[20] = "\\\\.\\PhysicalDrive0";

hDevice = CreateFile(s, GENERIC\_READ | GENERIC\_WRITE, FILE\_SHARE\_WRITE | FILE\_SHARE\_READ,

NULL, OPEN\_EXISTING, 0, NULL);

if (hDevice == INVALID\_HANDLE\_VALUE)

printf("Невозможно получить хэндл - ого жесткого диска.\n");

else

{

dg.MediaType = FixedMedia;

DeviceIoControl(hDevice, IOCTL\_DISK\_GET\_DRIVE\_GEOMETRY, NULL, 0,

&dg, sizeof(DISK\_GEOMETRY), &ReturnSize, (LPOVERLAPPED)NULL);

printf("Цилиндров : %d\n", dg.Cylinders);

printf("Дорожек на цилиндр : %d\n", dg.TracksPerCylinder);

printf("Секторов на дорожку : %d\n", dg.SectorsPerTrack);

printf("Размер сектора : %d байт\n", dg.BytesPerSector);

DiskSize = (dg.Cylinders.LowPart \* dg.TracksPerCylinder \*

dg.SectorsPerTrack) / 1024 \* dg.BytesPerSector;

printf("Размер HDD : %d Mb\n", DiskSize / 1024);

cout << endl;

if (!ReadFile(hDevice, &mbr, 512, &dwRead, NULL))

{

cout << "Ошибка не может прочитать MBR" << endl;

CloseHandle(hDevice);

}

while (d < 4)

{

cout << endl;

e = (unsigned char)mbr[446 + i];

//printf("Код загрузочного раздела: %X\n", e);

if (e == 128)

cout << "Загрузочный" << endl;

else

cout << "Не загрузочный" << endl;

f = (unsigned char)mbr[446 + 4 + i];

//printf("Код типа раздела: %X\n", f);

if (f == 7 || f == 17)

{

//unsigned char

cout << "Основной раздел" << endl;

cout << "ФС : NTFS\n";

a = \*(int\*)&mbr[446 + 8 + i];

printf("Адрес начала раздела: %d в секторах\n", a);

b = \*(int\*)&mbr[446 + 12 + i];

b = b / 2 / 1024;

printf("Размер диска: %d Мб\n", b);

ladd = a; //

if(k1 == 0)

a1 = a;

else if (k1 > 0)

a1 = +a;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice, &mbr, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr[z1];

;

cout << "Серийный номер: " << dec << l << endl;

a = \*(int\*)&mbr[446 + 8 + i];

}

else if (f == 11 || f == 12)

{

cout << "Основной раздел" << endl;

cout << "ФС : FAT32\n";

}

else if (f == 1 || f == 4 || f == 6 || f == 13 || f == 14 || f == 20 || f == 22 || f == 23 || f == 27)

{

cout << "Основной раздел" << endl;

cout << "ФС : ?\n";

}

else if (f == 5 || f == 15)

cout << "Расширенный раздел" << endl;

else

{

cout << "ФС не опознана" << endl;

a = \*(int\*)&mbr[446 + 8 + i];

printf("Адрес начала раздела: %d в секторах\n", a);

b = \*(int\*)&mbr[446 + 12 + i];

b = b / 2 / 1024;

printf("Размер диска: %d Мб\n", b);

}

a1 += a;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice, &mbr, 512, &dwRead, NULL);

i += 16;

d++;

cout << endl;

}

cout << endl;

//}

i = 0;

d = 0;

b = 0;

e = 0;

f = 0;

char s1[20] = "\\\\.\\PhysicalDrive1";

hDevice1 = CreateFile(s1, GENERIC\_READ | GENERIC\_WRITE, FILE\_SHARE\_WRITE | FILE\_SHARE\_READ,

NULL, OPEN\_EXISTING, 0, NULL);

if (hDevice1 == INVALID\_HANDLE\_VALUE)

printf("Невозможно получить хэндл - ого жесткого диска.\n");

else

{

dg.MediaType = FixedMedia;

DeviceIoControl(hDevice1, IOCTL\_DISK\_GET\_DRIVE\_GEOMETRY, NULL, 0,

&dg, sizeof(DISK\_GEOMETRY), &ReturnSize, (LPOVERLAPPED)NULL);

//CloseHandle(hDevice);

printf("Цилиндров : %d\n", dg.Cylinders);

printf("Дорожек на цилиндр : %d\n", dg.TracksPerCylinder);

printf("Секторов на дорожку : %d\n", dg.SectorsPerTrack);

printf("Размер сектора : %d байт\n", dg.BytesPerSector);

DiskSize = (dg.Cylinders.LowPart \* dg.TracksPerCylinder \*

dg.SectorsPerTrack) / 1024 \* dg.BytesPerSector;

printf("Размер HDD : %d Mb\n", DiskSize / 1024);

cout << endl;

if (!ReadFile(hDevice1, &mbr, 512, &dwRead, NULL))

{

cout << "Ошибка не может прочитать MBR" << endl;

CloseHandle(hDevice1);

}

}

int p = 0;

a1 = 0;

int k = 0;

while (d < 4)

{

a = \*(\_\_int64\*)&mbr[446 + 8 + i];

e = (unsigned char)mbr[446 + i];

//printf("Код загрузочного раздела: %X\n", e);

if (e == 128)

cout << "Загрузочный" << endl;

else

cout << "Не загрузочный" << endl;

f = (unsigned char)mbr[446 + 4 + i];

b = \*(int\*)&mbr[446 + 12 + i];

b = b / 2 / 1024;

if (f == 7 || f == 17)

{

cout << "Основной раздел" << endl;

cout << "ФС : NTFS\n";

printf("Размер диска: %d Мб\n", b);

//if (k1 > 0)

printf("Адрес начала раздела: %d в секторах\n", a);

ladd = a; //

//a1 = a;

ladd \*= 512;

badd = ladd & z;

ladd = ladd >> 32;

ladd = ladd & z;

SetFilePointer(hDevice1, (int)badd, (PLONG)&ladd, 0);

ReadFile(hDevice1, &mbr1, 512, &dwRead, NULL);

l = \*(DWORD\*)&mbr1[z1];

cout << "Серийный номер: " << dec << l << endl;

}

else if (f == 11 || f == 12)

{

cout << "Основной раздел" << endl;

cout << "ФС : FAT32\n";

}

else if (f == 1 || f == 4 || f == 13 || f == 14 || f == 20 || f == 22 || f == 23 || f == 27)

{

cout << "Основной раздел" << endl;

cout << "ФС : ?\n";

}

else if (f == 5 || f == 15)

{

cout << "Расширенный раздел" << endl;

showEbr(a);

}

else

{

cout << "Расширенный раздел" << endl;

}

i += 16;

d++;

}

cout << endl;

}

}

int main()

{

setlocale(LC\_CTYPE, "Rus");

showDisk();

char VolumeNameBuffer[100];

char FileSystemNameBuffer[100];

DWORD VolumeSerialNumber;

BOOL GetVolumeInformationFlag = GetVolumeInformation(

"c:\\", (LPSTR)&VolumeNameBuffer,

100,

(LPDWORD)&VolumeSerialNumber,

NULL, //&MaximumComponentLength,

NULL, //&FileSystemFlags,

(LPTSTR) FileSystemNameBuffer,

100

);

if (GetVolumeInformationFlag != 0)

{

cout << "c:\\" << endl;

//cout << "Volume Name is " << &VolumeNameBuffer << endl;

cout << "Серийный номер: " << dec << VolumeSerialNumber << endl;

cout << "Файловая система: " << FileSystemNameBuffer << endl;

}

else cout << "Не представляет (GetVolumeInformation)" << endl;

GetVolumeInformationFlag = GetVolumeInformation(

"d:\\", VolumeNameBuffer,

100,

(LPDWORD)&VolumeSerialNumber,

NULL, //&MaximumComponentLength,

NULL, //&FileSystemFlags,

(LPTSTR)FileSystemNameBuffer,

100

);

if (GetVolumeInformationFlag != 0)

{

cout << "d:\\" << endl;

//cout << "Volume Name is " << &VolumeNameBuffer << endl;

cout << "Серийный номер: " << dec << VolumeSerialNumber << endl;

cout << "Файловая система: " << FileSystemNameBuffer << endl;

}

else cout << "Не представляет (GetVolumeInformation)" << endl;

GetVolumeInformationFlag = GetVolumeInformation(

"f:\\", VolumeNameBuffer,

100,

&VolumeSerialNumber,

NULL, //&MaximumComponentLength,

NULL, //&FileSystemFlags,

(LPTSTR)FileSystemNameBuffer,

100

);

if (GetVolumeInformationFlag != 0)

{

cout << "f:\\" << endl;

//cout << "Volume Name is " << &VolumeNameBuffer << endl;

cout << "Серийный номер: " << dec << VolumeSerialNumber << endl;

cout << "Файловая система: " << FileSystemNameBuffer << endl;

}

else cout << "Не представляет (GetVolumeInformation)" << endl;

GetVolumeInformationFlag = GetVolumeInformation(

"g:\\", VolumeNameBuffer,

100,

(LPDWORD)&VolumeSerialNumber,

NULL, //&MaximumComponentLength,

NULL, //&FileSystemFlags,

(LPTSTR)FileSystemNameBuffer,

100

);

if (GetVolumeInformationFlag != 0)

{

cout << "g:\\" << endl;

//cout << "Volume Name is " << &VolumeNameBuffer << endl;

cout << "Серийный номер: " << dec << VolumeSerialNumber << endl;

cout << "Файловая система: " << FileSystemNameBuffer << endl;

}

else cout << "Не представляет (GetVolumeInformation)" << endl;

GetVolumeInformationFlag = GetVolumeInformation(

"h:\\", VolumeNameBuffer,

100,

(LPDWORD)&VolumeSerialNumber,

NULL, //&MaximumComponentLength,

NULL, //&FileSystemFlags,

(LPTSTR)FileSystemNameBuffer,

100

);

if (GetVolumeInformationFlag != 0)

{

cout << "h:\\" << endl;

//cout << "Volume Name is " << &VolumeNameBuffer << endl;

cout << "Серийный номер: " << dec << VolumeSerialNumber << endl;

cout << "Файловая система: " << FileSystemNameBuffer << endl;

}

else cout << "Не представляет (GetVolumeInformation)" << endl;

GetVolumeInformationFlag = GetVolumeInformation(

"i:\\", VolumeNameBuffer,

100,

(LPDWORD)&VolumeSerialNumber,

NULL, //&MaximumComponentLength,

NULL, //&FileSystemFlags,

(LPTSTR)FileSystemNameBuffer,

100

);

if (GetVolumeInformationFlag != 0)

{

cout << "i:\\" << endl;

//cout << "Volume Name is " << &VolumeNameBuffer << endl;

cout << "Серийный номер: " << dec << VolumeSerialNumber << endl;

cout << "Файловая система: " << FileSystemNameBuffer << endl;

}

else cout << "Не представляет (GetVolumeInformation)" << endl;

\_getch();

}