ss#include <windows.h>

#include <winioctl.h>

#include <stdio.h>

BOOL GetDriveGeometry(DISK\_GEOMETRY\* pdg)

{

HANDLE hDevice; // дескриптор проверяемого устройства

BOOL bResult; // флажок результата

DWORD junk; // сбрасываем результаты

LPCTSTR f\_name = L"\\\\.\\PhysicalDrive0";

hDevice = CreateFile(f\_name, // открываемое устройство

0, // нет доступа к устройству

FILE\_SHARE\_READ | // режим совместного использования

FILE\_SHARE\_WRITE,

NULL, // атрибуты безопасности по умолчанию

OPEN\_EXISTING, // расположение

0, // атрибуты файла

NULL); // не копировать атрибуты файла

if (hDevice == INVALID\_HANDLE\_VALUE) // невозможно открыть устройство

{

return (FALSE);

}

bResult = DeviceIoControl(hDevice, // запрошенное устройство

IOCTL\_DISK\_GET\_DRIVE\_GEOMETRY, // выполняемая операция

NULL, 0, // буфера ввода нет

pdg, sizeof(\*pdg), // буфер вывода

&junk, // # возвращено байтов

(LPOVERLAPPED)NULL); // синхронизация ввода/вывода (I/O)

CloseHandle(hDevice);

return (bResult);

}

int main(int argc, char\* argv[])

{

DISK\_GEOMETRY pdg; // геометрическая структура дискового устройства

BOOL bResult; // флажок общих результатов

ULONGLONG DiskSize; // размер диска, в байтах

bResult = GetDriveGeometry(&pdg);

if (bResult)

{

printf("Cylinders = %I64d\n", pdg.Cylinders);

printf("Tracks per cylinder = %ld\n", (ULONG)pdg.TracksPerCylinder);

printf("Sectors per track = %ld\n", (ULONG)pdg.SectorsPerTrack);

printf("Bytes per sector = %ld\n", (ULONG)pdg.BytesPerSector);

DiskSize = pdg.Cylinders.QuadPart \* (ULONG)pdg.TracksPerCylinder \*

(ULONG)pdg.SectorsPerTrack \* (ULONG)pdg.BytesPerSector;

printf("Disk size = %I64d (Bytes) = %I64d (Gb)\n", DiskSize,

DiskSize / (1024 \* 1024 \* 1024));

}

else

{

printf("GetDriveGeometry failed. Error %ld.\n", GetLastError());

}

return ((int)bResult);

}