МАТЕРИАЛЫ САЙТА

Написание драйвера в подробностях №4

Журнал «Хакер» , 31.10.2005 36 мин на чтение 0 13423

Мобильная версия статьи

Цикл подошел к концу. В этой статье я расскажу об еще нескольких способах инсталляции драйверов
(в прошлый раз мы рассмотрели инсталляцию с помощью простого изменения реесстра (вручную или с использованием .reg — файла) и с использованием программы Monitor из состава Driver Studio) — с помощью inf — файла и с использованием возможностей SCM — менеджера. А последний пункт вплотную подводит нас к написанию собственной консольной проги для тестирования

возможностей нашего драйвера.

Ну а в качестве маленького бонуса я расскажу, как обещала, о правке солюшенов для визуальной студии, для того, чтобы ты смог писать в ней драйвера (а правка нужна для корректной подсветки синтаксиса кода и компиляции). Приступим.

Инсталляция драйвера (продолжение)

В данной статье мы рассмотрим еще два способа инсталляции драйвера: с помощью inf — файла и с использованием возможностей SCM — менеджера. Начнем с первого.

Для того, чтобы мы могли корректно установить наш драйвер с использованием виндошного Мастера Установка, нам необходимо создать inf — файл с правильной структурой. Пример оного вместе с комментариями читай ниже:

```
; Primer.Inf
[Version]
Signature=»$Hacker$»
Class=Unknown
Provider=%HackerMagazine%
DriverVer=27/10/2005,0.0.0.1

[Manufacturer]
%HackerMagazine%=Hacker.Magazine

[Hacker.Magazine]
%Primer%=Primer.Install, *hackermagazine\Primer
[DestinationDirs]
Primer.Files.Driver=10,System32\Drivers ; путь для копирования драйвера под Windows 98
Primer.Files.Driver.NTx86=10,System32\Drivers ; путь для копирования драйвера под Windows NT

[SourceDisksNames]
1=»Primer build directory»,,,
[SourceDisksFiles]
Primer.sys=1,drv\w98 ; путь, где находится версия драйвера под Windows 98

[SourceDisksFiles.x86]
Primer.sys=1,drv\nt ; путь, где находится версия драйвера под Windows NT
; в нашем случае обе версии драйверов, конечно, одинаковые
```

```
; Windows 98
; секция, используемая в случае установки драйвера под Windows 98
[Primer.Install]
CopyFiles=Primer.Files.Driver
AddReg=Primer.AddReg

[Primer.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,Primer.sys
[Primer.Files.Driver]
Primer.sys
; Windows 2000, XP, Server 2003
; секция, используемая в случае установки драйвера под Windows NT
[Primer.Install.NTx86]
CopyFiles=Primer.Files.Driver.NTx86

[Primer.Files.Driver.NTx86]
Primer.sys,,,%COPYFLG_NOSKIP%

[Primer.Install.NTx86.Services]
AddService = Primer, %SPSVCINST_ASSOCSERVICE%, Primer.Service

[Primer.Service]
DisplayName = %Primer.ServiceName%
ServiceType = %SERVICE_KERNEL_DRIVER%
StartType = %SERVICE_AUTO_START%
ErrorControl = %SERVICE_ERROR_NORMAL%
ServiceBinary = %10%\System32\Drivers\Primer.sys
; Строки
[Strings]
HackerMagazine=»Napisanie draiverov v podrobnostjah №4"
Primer=»Primer driver: checked build»
Primer.ServiceName=»Primer NTDDK driver (V.001)»

SPSVCINST_ASSOCSERVICE=0x00000002
COPYFLG_NOSKIP=2
SERVICE_KERNEL_DRIVER=1
SERVICE_AUTO_START=2
SERVICE_DEMAND_START=3
SERVICE_ERROR_NORMAL=1
```

Сохраняй этот файлик с расширением inf. Ну а теперь можешь запускать Мастер Установки, в нем указывай способ выбора устройства вручную

(без автоматического поиска и определения), путь к каталогу с inf — файлом… Ну, думаю, тебя не нужно учить общению с Мастером 🙂 После завершения установки, если все пройдет нормально, то драйвер успешно встанет в систему и ты сможешь лицезреть его в списке устройств. Все, теперь его можно тестировать.

Перейдем к SCM — менеджеру. SCM — менеджер — это сервис Windows NT, предоставляющий удобную возможность работать с драйвером без использования Мастера Установки — с помощью функций, вызываемых из пользовательского приложения. Для того, чтобы начать работать с SCM — мененджером, необходимо вызвать функцию OpenSCManager, а для завершения работы — CloseServiceHandle. Но, к сожалению, работать с помощью SCM — менеджера можно не со всеми типами драйверов.

Ну а теперь мы приступим к написанию своего консольного приложения, тестирующего драйвер и использующего SCM — менеджер.

Написание консольного приложения для тестирования драйвера

Во-первых, нужно создать файл Ioctl.h. Его содержимое
(которое, надо думать, тебе знакомо по файлу Driver.h :)):

```
#ifndef _IOCTL_H_05703_ BASHBD_1UIWQ1_4763_ 1NJKDH256_801_
#define _IOCTL_H_05703_ BASHBD_1UIWQ1_4763_ 1NJKDH256_801_
#define IOCTL_PRINT_DEBUG_ MESS CTL_CODE( \FILE_DEVICE_UNKNOWN, 0x701,
METHOD_BUFFERED, FILE_ANY_ACCESS)

#define IOCTL_CHANGE_IRQL CTL_CODE(\
FILE_DEVICE_UNKNOWN, 0x702, METHOD_BUFFERED, FILE_ANY_ACCESS)

#define IOCTL_MAKE_SYSTEM_CRASH CTL_CODE(\
FILE_DEVICE_UNKNOWN, 0x703, METHOD_BUFFERED, FILE_ANY_ACCESS)

#define IOCTL_TOUCH_PORT_378H CTL_CODE(\
FILE_DEVICE_UNKNOWN, 0x704, METHOD_BUFFERED, FILE_ANY_ACCESS)

#define IOCTL_SEND_BYTE_TO_USER CTL_CODE(\
FILE_DEVICE_UNKNOWN, 0x705, METHOD_BUFFERED, FILE_ANY_ACCESS)
#endif
```

Вот содержимое файла resource.h:

```
#ifdef APSTUDIO_INVOKED
```

```
#ifndef APSTUDIO_ READONLY_SYMBOLS
#define _APS_NEXT_ RESOURCE_VALUE 101
#define _APS_NEXT_ COMMAND_VALUE 40001
#define _APS_NEXT_ CONTROL_VALUE 1000
#define _APS_NEXT_ SYMED_VALUE 101
#endif
#endif
```

… и файла ресурсов .rc:

```
//Microsoft Developer Studio generated resource script.
//
#include «resource.h»

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include «afxres.h»

////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////////////////////////////////////////////
// Russian resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_RUS)
#ifdef _WIN32
LANGUAGE LANG_RUSSIAN, SUBLANG_DEFAULT
#pragma code_page(1251)
#endif //_WIN32

#ifndef _MAC
////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,0
PRODUCTVERSION 1,0,0,0
FILEFLAGSMASK 0x3fL
```

```
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
BLOCK «StringFileInfo»
BEGIN
BLOCK «040904b0»
BEGIN
VALUE «Comments», «Tester for Primer.sys driver\0»
VALUE «CompanyName», «Hacker\0»
VALUE «FileDescription», «TestPrim\0»
VALUE «FileVersion», «1, 0, 0, 0\0»
VALUE «InternalName», «TestPrim\0»
VALUE «LegalCopyright», «Copyright © 2005 Hacker\0»
VALUE «LegalTrademarks», «\0»
VALUE «OriginalFilename», «TestPrim.exe\0»
VALUE «PrivateBuild», «\0»
VALUE «ProductName», «Primer.sys tester\0»
VALUE «ProductVersion», «1, 0, 0, 0\0»
VALUE «SpecialBuild», «\0»
END
END
BLOCK «VarFileInfo»
BEGIN
VALUE «Translation», 0x409, 1200
END
END

#endif // !_MAC

#endif // Russian resources
/////////////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////////
// English (U.K.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENG)
#ifdef _WIN32
```

```
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_UK
#pragma code_page(1252)
#endif //_WIN32

#ifdef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE MOVEABLE PURE
BEGIN
«resource.h\0»
END

2 TEXTINCLUDE MOVEABLE PURE
BEGIN
«#include «»afxres.h»»\r\n»
«\0»
END

3 TEXTINCLUDE MOVEABLE PURE
BEGIN
«\r\n»
«\0»
END

#endif // APSTUDIO_INVOKED

#endif // English (U.K.) resources
//////////////////////////////////////////////////////////////////////////////

#ifndef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

//////////////////////////////////////////////////////////////////////////////
#endif // not APSTUDIO_INVOKED
```

Ну а теперь головной .cpp — шник программы с комментариями:

```
//////////////////////////////////////////////////////////////////////
// Файл PrimerTest.cpp
// Консольное приложение для тестирования драйвера Primer.sys
//////////////////////////////////////////////////////////////////////

// Заголовочные файлы, которые необходимы данной проге:
#include <windows.h>
#include <stdio.h>
#include <winioctl.h>
#include <tchar.h>

#include «Ioctl.h»
// Имя объекта драйвера и местоположение загружаемого файла
#define DRIVERNAME _T(«Primer»)
//#define DRIVERBINARY _T(«C:\\Primer\\Primer.sys»)
//#define DRIVERBINARY _T(«C:\\Ex\\objchk_w2k\\i386\\Primer.sys»)
#define DRIVERBINARY _T(«C:\\Ex\\tester\\Primer.sys»)

// Начинаем знакомиться с SCM — менеджером. Далее — функция установки
драйвера на основе его вызовов.
BOOL InstallDriver(SC_HANDLE scm, LPCTSTR DriverName, LPCTSTR driverExec)
{
SC_HANDLE Service =
CreateService (scm, // открытый дескриптор к SCManager
DriverName, // имя сервиса — Primer
DriverName, // для вывода на экран
SERVICE_ALL_ACCESS, // желаемый доступ
SERVICE_KERNEL_DRIVER, // тип сервиса
SERVICE_DEMAND_START, // тип запуска
SERVICE_ERROR_NORMAL, // указывает, как обрабатывается ошибка
driverExec, // путь к бинарному файлу
// Остальные параметры не используются – укажем NULL
NULL, // Не определяем группу загрузки
NULL, NULL, NULL, NULL);
if (Service == NULL) // неудача
{
DWORD err = GetLastError();
if (err == ERROR_SERVICE_EXISTS) {/* уже установлен
*/}
// более серьезная ошибка:
else printf («ERR: Can’t create service. Err=%d\n»,err);
return FALSE;
}
CloseServiceHandle (Service);
```

Checl

```
return TRUE;
}

// Функция удаления драйвера на основе SCM вызовов
BOOL RemoveDriver(SC_HANDLE scm, LPCTSTR DriverName)
{
SC_HANDLE Service =
OpenService (scm, DriverName, SERVICE_ALL_ACCESS);
if (Service == NULL) return FALSE;
BOOL ret = DeleteService (Service);
if (!ret) { /* неудача при удалении драйвера
*/ }

CloseServiceHandle (Service);
return ret;
}
// Функция запуска драйвера на основе SCM вызовов
BOOL StartDriver(SC_HANDLE scm, LPCTSTR DriverName)
{
SC_HANDLE Service =
OpenService(scm, DriverName, SERVICE_ALL_ACCESS);
if (Service == NULL) return FALSE; /* open failed */
BOOL ret =
StartService(Service, // дескриптор
0, // число аргументов
NULL); // указатель на аргументы
if (!ret) // неудача
{
DWORD err = GetLastError();
if (err == ERROR_SERVICE_ALREADY_RUNNING)
ret = TRUE; // драйвер уже работает
else { /* другие проблемы */}
}

CloseServiceHandle (Service);
return ret;
}
// Функция остановки драйвера на основе SCM вызовов
BOOL StopDriver(SC_HANDLE scm, LPCTSTR DriverName)
{
SC_HANDLE Service =
OpenService (scm, DriverName, SERVICE_ALL_ACCESS);
if (Service == NULL) // невозможно выполнить остановку драйвера
{
```

```
DWORD err = GetLastError();
return FALSE;
}
SERVICE_STATUS serviceStatus;
BOOL ret =
ControlService(Service, SERVICE_CONTROL_STOP, &serviceStatus);
if (!ret)
{
DWORD err = GetLastError();

}

CloseServiceHandle (Service);
return ret;
}
// Соберем вместе действия по установке, запуску, останову
// и удалению драйвера (для учебных целей).
// Пользоваться этой функцией в данном примере нам не придется,
// поэтому закомментируем ее.
/*
void Test_SCM_Installation(void)
{
SC_HANDLE scm = OpenSCManager(NULL,NULL,SC_MANAGER_ALL_ACCESS);
if(scm == NULL) // неудача
{
// получаем код ошибки и ее текстовый эквивалент
unsigned long err = GetLastError();
PrintErrorMessage(err);
return;
}
BOOL res;
res = InstallDriver(scm, DRIVERNAME, DRIVERBINARY);
// ошибка может быть не фатальной. Продолжаем:
res = StartDriver (scm, DRIVERNAME);
if(res)
{
// Здесь следует разместить функции работы с драйвером
res = StopDriver (scm, DRIVERNAME);
if(res) res = RemoveDriver (scm, DRIVERNAME);
}
CloseServiceHandle(scm);
return;
}
*/
```

```
#define SCM_SERVICE ; // см. ниже
// вводим элемент условной компиляции, при помощи
// которого можно отключать использование SCM установки драйвера
// в тексте данного приложения.

// Основная функция тестирующего приложения.
// Здесь диагностике ошибочных ситуаций уделено очень мало внимания.
// В реальных приложениях так делать
// нельзя!

int __cdecl main(int argc, char* argv[])
{
#ifdef SCM_SERVICE
// используем сервис SCM для запуска драйвера.
BOOL res; // получаем доступ к SCM :
SC_HANDLE scm = OpenSCManager(NULL,NULL,SC_MANAGER_ALL_ACCESS);
if(scm == NULL) return -1; // неудача

// делаем попытку установки драйвера
res = InstallDriver(scm, DRIVERNAME, DRIVERBINARY);
if(!res) // неудача, но, быть может, драйвер уже установлен ?
printf(«Cannot install service»);

res = StartDriver (scm, DRIVERNAME);
if(!res)
{
printf(«Cannot start driver!»);
res = RemoveDriver (scm, DRIVERNAME);
if(!res)
{
printf(«Cannot remove driver!»);
}
CloseServiceHandle(scm); // отключаемся от SCM
return -1;
}
#endif

HANDLE hHandle = // Получаем доступ к драйверу
CreateFile("\\\\.\\Primer",
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
OPEN_EXISTING,
```

```
FILE_ATTRIBUTE_NORMAL,
NULL);
if(hHandle==INVALID_HANDLE_VALUE)
{
printf(«ERR: can not access driver Primer.sys !\n»);
return (-1);
}
DWORD BytesReturned; // Переменная для хранения числа
// переданных байт
// Приступаем к тестированию драйвера. Последовательно выполняем обращения к
драйверу
// с различными кодами IOCTL:

unsigned long ioctlCode=IOCTL_PRINT_DEBUG_MESS;
if(!DeviceIoControl(hHandle,
ioctlCode,
NULL, 0,
NULL, 0,
&BytesReturned,
NULL))
{
printf("Error in IOCTL_PRINT_DEBUG_MESS!");
return(-1);
}

ioctlCode=IOCTL_CHANGE_IRQL;
if(!DeviceIoControl(hHandle,
ioctlCode,
NULL, 0,
NULL, 0,
&BytesReturned,
NULL))
{
printf("Error in IOCTL_CHANGE_IRQL!");
return(-1);
}

ioctlCode=IOCTL_TOUCH_PORT_378H;
if(!DeviceIoControl(hHandle,
ioctlCode,
NULL, 0,
NULL, 0,
&BytesReturned,
NULL))
```

```
{
printf("Error in IOCTL_TOUCH_PORT_378H!»);
return(-1);
}
// Продолжаем тестирование. Получаем 1 байт данных из драйвера.
// По окончании данного вызова переменная xdata должна
// содержать значение 33:
unsigned char xdata = 0x88;
ioctlCode=IOCTL_SEND_BYTE_TO_USER;
if(!DeviceIoControl(hHandle,
ioctlCode,
NULL, 0,
&xdata, sizeof(xdata),
&BytesReturned,
NULL))
{
printf("Error in IOCTL_SEND_BYTE_TO_USER!");
return(-1);
}

// Выводим мессагу в дебаговую консоль:
printf(«IOCTL_SEND_BYTE_TO_USER: BytesReturned=%d xdata=%d»,
BytesReturned, xdata);

// Выполнение следующего теста в Windows NT приведет к
// краху ОС`и
/*
ioctlCode=IOCTL_MAKE_SYSTEM_CRASH;
if(!DeviceIoControl(hHandle,
ioctlCode,
NULL, 0,
NULL, 0,
&BytesReturned,
NULL))
{
printf("Error in IOCTL_MAKE_SYSTEM_CRASH!");
return(-1);
}
*/
// Закрываем дескриптор доступа к драйверу:
CloseHandle(hHandle);

#ifdef SCM_SERVICE
// Останавливаем драйвер, удаляем его и отключаемся от
```

```
SCM.
res = StopDriver (scm, DRIVERNAME);
if(!res)
{
printf(«Cannot stop driver!»);
CloseServiceHandle(scm);
return -1;
}

res = RemoveDriver (scm, DRIVERNAME);
if(!res)
{
printf(«Cannot remove driver!»);
CloseServiceHandle(scm);
return -1;
}

CloseServiceHandle(scm);
#endif
return 0;
}
```

Вот и все, компиль и тестируй. Напоминаю еще раз: запусти вьювер дебаг-информации из Monitor
(например) и посмотри все дебаговые сообщения драйвера!

Правка солюшенов визуальной студии

Теперь я расскажу тебе, как исправить солюшены визуальной студии, чтобы ты смог писать драйвера в ней. Тем не менее, код драйвера, откомпилированный с помощью DDK
(а лучше всего использовать именно его),
считается эталонным. Ниже я привожу текст файла Primer.sln
(Visual Studio 7) (для компиляции не-WDM драйвера):

```
Microsoft Visual Studio Solution File, Format Version 7.00
Project (» {8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942} «) = «Primer»,
EndProject
Global
GlobalSection (SolutionConfiguration) = preSolution
ConfigName.0 = Checked
EndGlobalSection
GlobalSection (ProjectDependencies) = postSolution
```

```
EndGlobalSection
GlobalSection (ProjectConfiguration) = postSolution
{ E524BA09-7993-4528-91A9-7E27FAA3565F }.Checked.ActiveCfg =
Checked | Win32
{ E524BA09-7993-4528-91A9-7E27FAA3565F }.Checked.Build.0 =
Checked | Win32
EndGlobalSection
GlobalSection (ExtensibilityGlobals) = postSolution
EndGlobalSection
GlobalSection (ExtensibilityAddIns) = postSolution
EndGlobalSection
EndGlobal
```

А теперь — .vcproj:

```
<?xml version = «1.0» encoding = «windows-1251» ?>
<VisualStudioProject ProjectType = «Visual C++»
Version = «7.00»
Name = «Primer»
SccProjectName = » «
SccLocalPath = » » >
<Platforms><Platform Name = «Win32» /> </Platforms>
<Configurations> <Configuration Name = «Checked | Win32 «
OutputDirectory = «.\checked»
IntermeaditeDirectory = «.\checked»
ConfigurationType = «2»
UseOfMFC = «0»
ATLMinimizesCRunTimeLibraryUsage = «FALSE»
CharacterSet = «1» >
<Tool Name = «VCCLCompilerTool»
AdditionalOptions = «/Zel — cbstring /QIfdiv- /QIf /Gi- /Gm- /GX»
Optimization = «0»
EnableIntrinsicFunctions = «FALSE»
OmitFramePointers = «TRUE»
OptimizeForProcessor = «2»
AdditionalIncludeDirectories = «C:\WinDDK\2600\inc\ddk\w2k;
C:\WinDDK\2600\inc\w2k;C:\WinDDK\2600\inc\crt»
PreprocessorDefinitions = «_X86_=1;i386=1;
CONDITION_HANDLING=1;NT_UP=1;
NT_INST=0; WIN32=100;_NT1X_= 100;WINNT=1;
_WIN32_WINNT=0x0400; WIN32_LEAN_AND_MEAN=1;
DEVL=1; DBG=1; FPO=0"
IgnoreStandartIncludePath=»TRUE»
StringPooling = «TRUE»
```

ExceptionHandling = «TRUE»
RuntimeLibrary=»0"
StructMemberAlignment=»4"
BufferSecurityCheck = «FALSE»
EnableFunctionLevelLinking = «TRUE»
PrecompiledHeaderFile = ".\checked/Primer.pch"
AssemblerListingLocation = «.\checked/»
ObjectFile=».\checked/»
ProgramDataBaseFileName= ".\checked\Primer.pdb"
WarningLevel=»3"
SuppressStartupBanner=»TRUE»
DebugInformationFormat=»1"
CallingConvention=»2"
CompileAs = «0»
ForcedIncludeFiles= "warning.h"/>
<Tool Name = «VCCustomerBuildTool»/>
<Tool Name = «VCLinkerTool» AdditionalOptions = » «
AdditionalDependencies = «hal.lib ntoskrnl.lib int64.lib
msvcrt.lib»
OutputFile = «.\checked\Primer.sys»
Version = «5.0»
LinkIncremental = «1'
SuppressStartupBanner = «TRUE»
AdditionalLibraryDirectories = «C:\WinDDK\2600\lib\w2k\i386»
IgnoreAllDefaultsLibraries = «TRUE»
ProgramDatabaseFile = «.\checked/Primer.pdb»
GenerateMapFile = «TRUE»
MapFileName = «Primer.map»
StackReserveSize = «262144»
StackCommitSize = «4096»
OptimizeReferences = «2»
EnableCOMDATFolding = «2»
EntryPointSymbol = «DriverEntry»
SetChecksum = «TRUE»
BaseAddress = «0x10000»
ImportLibrary = » «
MergeSections = «.rdata = .text»
TargetMachine = «1» />
<Tool Name = «VCMIDILTool»
MkTypeLibCompatible = «TRUE»
SuppressStartupBanner = «TRUE»
TargetEnvironment = «1»
TypeLibraryName = «.\checked/Primer.tlb»/>
<Tool Name = «VCPostBuildEventTool» />

```
<Tool Name = «VCPreBuildEventTool»/>
<Tool Name = «VCPreLinkEventTool» />
<Tool Name = «VCResourceCompilerTool»/>
<Tool Name = «VCWebServiceProxyGeneratorTool» />
<Tool Name = «VCWebDeploymentTool» />
</Configuration>
</Configurations>
<Files> <Filter Name = «Header Files» Filter = «.h»>
<File RelativePath = «.\Driver.h»> </File>
</Filter>
<Filter Name = «Source Files» Filter = «.c;.cpp»>
<File RelativePath = «.\main.cpp»> </File>
</Filter>
</Files>
<Globals></Globals>
</VisualStudioProject>
```

Комментарии тут излишни — все параметры понятны без слов. Это, конечно, примерная структура, в твоем проекте многое может отличаться. Но для нашего тестового драйвера такие два файла вполне подойдут.

Заключение

Ну вот и все. Желаю удачи и интересных находок в плавании по увлекательному океану драйверов и железа! Да не облысеют твои пятки!

кер»

Теги: Софт Статьи

ДАЛЕЕ ПО ЭТОЙ ТЕМЕ РАНЕЕ ПО ЭТОЙ ТЕМЕ

Службы Windows изнутри

24.04.2006

Зомби-сети: дом мертвецов

15.08.2006

Переполнение буфера в драйверах NVIDIA для Linux

17.10.2006

Переполнение буфера в службе iGateway в продуктах CA

25.01.2006

Написание собственной Операционной Системы №2

23.11.2005

Очень Опытный Пользователь: повышение привилегий до Админа

26.05.2006