

Лабораторная работа №5

Защита от копирования. Привязка к аппаратному обеспечению. Использование реестра

Цель работы: ознакомиться с возможностями «привязки» к характеристикам компьютера.

Теоретические сведения

В качестве анализируемых характеристик компьютера могут использоваться:

1. Информация об используемой операционной системе
2. Имя пользователя;
3. Имя компьютера;
4. Наличие звуковой карты;
5. Наличие подключенных принтера, сканера и т.д;
6. Дата создания BIOS;
7. Серийный номер диска;
8. Характеристики процессора.

Для получения подобных характеристик в операционной системе Windows используются API-функции и информация из реестра.

API-функции

API сокращенно Application Programming Interface (интерфейс прикладного программирования). API – набор функций, которые операционная система предоставляет программисту. API обеспечивает относительно простой путь для программистов для использования полных функциональных возможностей аппаратных средств или операционной системы.

32-разрядные версии Windows обычно используют один и тот же набор функций API, хотя имеются некоторые различия между платформами.

Почти все функции, которые составляют Windows API, находятся внутри DLL (Dynamic Link Library). Эти dll-файлы находятся в системной папке Windows. Существует свыше 1000 функций API, которые условно делятся на четыре основные категории:

- 1) работа с приложениями – запуск и закрытие приложений, обработка команд меню, перемещения и изменения размера окон;
- 2) графика – создание изображений;
- 3) системная информация – определение текущего диска, объема памяти, имя текущего пользователя и т.д.
- 4) работа с реестром – манипуляции с реестром Windows.

Реестр Windows

Реестр – база данных операционной системы, содержащая конфигурационные сведения. По замыслу Microsoft реестр должен был полностью заменить файлы ini,

которые были оставлены только для совместимости со старыми программами, ориентированными на более ранние версии операционной системы.

Переход от ini файлов к реестру произошел по той причине, что на эти файлы накладывается ряд серьезных ограничений, и главное из них состоит в том, что предельный размер такого файла составляет 64Кб.

Предупреждение: никогда не удаляйте или не меняйте информацию в реестре, если Вы не уверены что это именно то, что нужно. В противном случае некорректное изменение данных может привести к сбоям в работе Windows и, в лучшем случае, информацию придется восстанавливать [из резервной копии](#).

Реестр имеет следующую структуру:

1) HKEY_CLASSES_ROOT. В этом разделе содержится информация о зарегистрированных в Windows типах файлов, что позволяет открывать их по двойному щелчку мыши, а также информация для OLE и операций drag-and-drop;

2) HKEY_CURRENT_USER. Здесь содержатся настройки оболочки пользователя (например, Рабочего стола, меню "Пуск", ...), вошедшего в Windows. Они дублируют содержимое подраздела HKEY_USER\name, где name – имя пользователя, вошедшего в Windows. Если на компьютере работает один пользователь и используется обычный вход в Windows, то значения раздела берутся из подраздела HKEY_USERS\DEFAULT;

3) HKEY_LOCAL_MACHINE. Этот раздел содержит информацию, относящуюся к компьютеру: драйверы, установленное программное обеспечение и его настройки;

4) HKEY_USERS. Содержит настройки оболочки Windows для всех пользователей. Как было сказано выше, именно из этого раздела информация копируется в раздел HKEY_CURRENT_USER. Все изменения в HKCU (сокращенное название раздела HKEY_CURRENT_USER) автоматически переносятся в HKU;

5) HKEY_CURRENT_CONFIG. В этом разделе содержится информация о конфигурации устройств Plug&Play и сведения о конфигурации компьютера с переменным составом аппаратных средств;

6) HKEY_DYN_DATA. Здесь хранятся динамические данные о состоянии различных устройств, установленных на компьютере пользователя. Именно сведения этой ветви отображаются в окне "Свойства: Система" на вкладке "Устройства", вызываемого из Панели управления. Данные этого раздела изменяются самой операционной системой, так что редактировать что-либо вручную не рекомендуется.

Примеры процедур и функций, определяющих параметры компьютера

Определение версии операционной системы

```
BOOL DisplaySystemVersion()
{
    OSVERSIONINFOEX osv;
    BOOL bOsVersionInfoEx;
    ZeroMemory(&osv, sizeof(OSVERSIONINFOEX));
    osv.dwOSVersionInfoSize = sizeof(OSVERSIONINFOEX);
    if( !(bOsVersionInfoEx = GetVersionEx ((OSVERSIONINFO *) &osv)) )
    {
        osv.dwOSVersionInfoSize = sizeof (OSVERSIONINFO);
```

```

    if (! GetVersionEx ( (OSVERSIONINFO *) &osvi) )
        return FALSE;
}

switch (osvi.dwPlatformId)
{
    case VER_PLATFORM_WIN32_NT:
        if ( osvi.dwMajorVersion <= 4 )
            printf("Microsoft Windows NT ");
        if ( osvi.dwMajorVersion == 5 && osvi.dwMinorVersion == 0 )
            printf ("Microsoft Windows 2000 ");
        if( bOsVersionInfoEx )
        {
            if ( osvi.wProductType == VER_NT_WORKSTATION )
            {
                if ( osvi.dwMajorVersion == 5 && osvi.dwMinorVersion == 1 )
                    printf ("Microsoft Windows XP ");

                if( osvi.wSuiteMask & VER_SUITE_PERSONAL )
                    printf ( "Home Edition " );
                else
                    printf ( "Professional " );
            }
            else if ( osvi.wProductType == VER_NT_SERVER )
            {
                if ( osvi.dwMajorVersion == 5 && osvi.dwMinorVersion == 2 )
                    printf ("Microsoft Windows .NET ");

                if( osvi.wSuiteMask & VER_SUITE_DATACENTER )
                    printf ( "DataCenter Server " );
                else if( osvi.wSuiteMask & VER_SUITE_ENTERPRISE )
                {
                    if( osvi.dwMajorVersion == 4 )
                        printf ("Advanced Server " );
                    else
                        printf ( "Enterprise Server " );
                }
                else if ( osvi.wSuiteMask == VER_SUITE_BLADE )
                    printf ( "Web Server " );
                else
                    printf ( "Server " );
            }
        }
    else
    {
        HKEY hKey;
        char szProductType[BUFSIZE];
        DWORD dwBufLen=BUFSIZE;
        LONG lRet;
        lRet = RegOpenKeyEx( HKEY_LOCAL_MACHINE,
            "SYSTEM\\CurrentControlSet\\Control\\ProductOptions",
            0, KEY_QUERY_VALUE, &hKey );
        if( lRet != ERROR_SUCCESS )
            return FALSE;
        lRet = RegQueryValueEx( hKey, "ProductType", NULL, NULL,
            (LPBYTE) szProductType, &dwBufLen);
        if( (lRet != ERROR_SUCCESS) || (dwBufLen > BUFSIZE) )
            return FALSE;
        RegCloseKey( hKey );
        if ( lstrcmpi( "WINNT", szProductType) == 0 )
            printf( "Professional " );
        if ( lstrcmpi( "LANMANNT", szProductType) == 0 )
            printf( "Server " );
    }
}

```

```

        if ( lstrcmpi( "SERVERNT", szProductType) == 0 )
            printf( "Advanced Server " );
    }
    if ( osv. dwMajorVersion <= 4 )
    {
        printf ("version %d.%d %s (Build %d)\n",
            osv. dwMajorVersion,
            osv. dwMinorVersion,
            osv. szCSDVersion,
            osv. dwBuildNumber & 0xFFFF);
    }
    else
    {
        printf ("%s (Build %d)\n",
            osv. szCSDVersion,
            osv. dwBuildNumber & 0xFFFF);
    }
    break;
case VER_PLATFORM_WIN32_WINDOWS:
    if (osv. dwMajorVersion == 4 && osv. dwMinorVersion == 0)
    {
        printf ("Microsoft Windows 95 ");
        if ( osv. szCSDVersion[1] == 'C' || osv. szCSDVersion[1] == 'B' )
            printf("OSR2 ");
    }
    if (osv. dwMajorVersion == 4 && osv. dwMinorVersion == 10)
    {
        printf ("Microsoft Windows 98 ");
        if ( osv. szCSDVersion[1] == 'A' )
            printf("SE ");
    }
    if (osv. dwMajorVersion == 4 && osv. dwMinorVersion == 90)
    {
        printf ("Microsoft Windows Millennium Edition ");
    }
    break;
}
return TRUE;
}

```

Определение серийного номера раздела диска

```

TCHAR    szVolName[256];
DWORD    dwNum;
DWORD    dwMaxComSize;
DWORD    dwFlags;
TCHAR    szFS[256];
BOOL     bRes;
bRes = GetVolumeInformation ( "c:\\", szVolName, sizeof(szVolName), &dwNum,
&dwMaxComSize, &dwFlags, szFS, sizeof(szFS));

```

Определение имени компьютера

```

const int WSVer = 0x101;
WSADATA wsaData;
char Buf[128];
if (WSAStartup(WSVer, &wsaData) == 0)
{
    gethostname(&Buf[0], 128);
    MessageBox(0, Buf, 0, 0);
    WSACleanup;
}

```

Определение имени пользователя

```

char buffer[UNLEN+1];

```

```

DWORD size;
size=sizeof(buffer);
GetUserName(buffer,&size);

```

Определение версии BIOS

```

LPSTR GetSystemBiosVersion()
{
    HKEY hKey;
    LONG Res1, Res2;
    DWORD cData=255;
    TCHAR SystemBiosVersion[255]={'\0'};
    Res1=RegOpenKeyEx(HKEY_LOCAL_MACHINE,"HARDWARE\\DESCRIPTION\\System",NULL,
KEY_QUERY_VALUE, &hKey);
    if(Res1==ERROR_SUCCESS)
    {

        Res2=RegQueryValueEx(hKey,"SystemBiosVersion",NULL,NULL,...
(LPBYTE)SystemBiosVersion,&cData);
        if(Res2==ERROR_SUCCESS)
        {
            for (const char* p = SystemBiosVersion; *p; p += strlen(p)+1)
            {
                printf("%s\n", p);
            }

            return SystemBiosVersion;
        }
        else
        {
            MessageBox(NULL,"RegQueryValueEx: SystemBiosVesion","ERROR",MB_OK);
            return NULL;
        }
    }
    else
    {
        MessageBox(NULL,"RegOpenKeyEx: SystemBiosVersion","ERROR",MB_OK);
        return NULL;
    }
    RegCloseKey(hKey);
}

```

Определение частоты процессора (способ №1)

```

double CPUSpeed(void)
{
    DWORD dwTimerHi, dwTimerLo;
    asm
    {
        DW 0x310F
        mov dwTimerLo, EAX
        mov dwTimerHi, EDX
    }
    Sleep (500);
    asm
    {
        DW 0x310F
        sub EAX, dwTimerLo
        sub EDX, dwTimerHi
        mov dwTimerLo, EAX
        mov dwTimerHi, EDX
    }
    return dwTimerLo/(1000.0*500);
}

```

Задание на лабораторную работу

Разработать программу, реализующую привязку к компьютеру, используя совокупность характеристик согласно варианту задания. Добиться того, чтобы программа не запускалась на другом компьютере.

Таблица 4. Варианты заданий

№ варианта	Характеристики
1	Серийный номер раздела жесткого диска, MAC-адрес сетевой карты
2	Информация из реестра, тактовая частота процессора
3	Версия операционной системы, MAC-адрес сетевой карты
4	Имя пользователя, серийный номер раздела жесткого диска
5	Название компьютера, информация из реестра
6	Версия БИОС, имя пользователя
7	Серийный номер раздела жесткого диска, имя пользователя
8	Имя пользователя, тактовая частота процессора
9	MAC-адрес сетевой карты, тактовая частота процессора

Контрольные вопросы

1. Что понимается под «привязкой» к компьютеру?
2. Какие характеристики обычно используются для идентификации компьютера?
3. Перечислите основные API-функции для определения индивидуальных характеристик компьютера.
4. Что представляет собой реестр Windows?
5. Какую структуру имеет реестр?