

Загрузка ресурсов из DLL

Помимо функций, динамические библиотеки могут содержать и ресурсы - строки, иконки, рисунки, диалоги и т. д. Хранение ресурсов в DLL очень удобно; в частности - при создании приложений с многоязычным интерфейсом: заменив одну DLL на другую, мы заменяем все надписи в программе, скажем, с английского на русский - и, заметьте, без всяких "хирургических вмешательств" в код приложения! Аналогично можно менять иконки, внешний вид диалогов и т. д.

Создание DLL, содержащей ресурсы, ничем не отличается от создания исполняемого приложения с ресурсами; сначала необходимо создать сам файл ресурсов - например, так:

```
// MyResDll.rc
#pragma code_page(1251)

STRINGTABLE DISCARDABLE
BEGIN
    1                "Hello, World!"
END
```

Листинг Создание DLL, содержащей одни лишь ресурсы

Файл ресурсов надо скомпилировать - **"rc MyResDll.rc"** - и преобразовать линкером в DLL, обязательно указав флаг **"/NOENTRY"**, т. к. эта динамическая библиотека содержит исключительно одни ресурсы и ни строки кода:
"link MyRedDll.res /DLL /NOENTRY".

В Visual Studio это сделать еще проще - достаточно кликнуть по папке *"Resources"* окна *"File View"* и добавить новый файл ресурса, который затем можно будет модифицировать визуальным редактором по своему усмотрению.

Для загрузки ресурса из DLL - в принципе, можно воспользоваться уже знакомой нам функцией **LoadLibrary**, и передавать возвращенный ею дескриптор **LoadString** или другой функции, работающей с ресурсами. Однако загрузку динамической библиотеки можно значительно ускорить, если "объяснить" системе, что эта DLL не содержит ничего, кроме ресурсов, и нам достаточно лишь спроецировать ее на адресное пространство процесса, а обо всем остальном мы сумеем позаботиться и самостоятельно.

Здесь пригодится функция **LoadLibraryEx**: ее первый аргумент, как и у **LoadLibrary**, задает имя динамической библиотеки для загрузки, второй - зарезервирован и должен быть равен нулю, а третий, будучи равным **LOAD_LIBRARY_AS_DATAFILE**, заставляет функцию делать именно то, что нам нужно - загружать DLL как базу данных (если динамическая библиотека содержит помимо ресурсов еще и код, то загрузка с этим ключом проходит все равно успешно, но функции загруженной DLL не будут доступны - только ресурсы):

```
// DllLoadRes.c
#include <windows.h>
#include <stdio.h>
```

```

main()
{
    HINSTANCE h;
    char buf[100];
    h=LoadLibraryEx("MyResDll.dll",0,
        LOAD_LIBRARY_AS_DATAFILE);

    LoadString(h,1,&buf[0],99);
    printf("%s\n",&buf[0]);
}

```

Листинг Демонстрация загрузки DLL, содержащей только ресурсы

Эта программа компилируется точно так же, как и предыдущие примеры явной компоновки - и после запуска победно выводит на экране *"Hello, Word!"*, подтверждая, что ресурс "строка" из динамической библиотеки был успешно загружен! Аналогичным способом можно загружать ресурсы из исполняемых файлов; с этой точки зрения они ничем не отличаются от динамических библиотек.