

параметров при вызове процедуры. Наименование характеристики говорит само за себя - Windows Application Programming Interface - применяются соглашения о передаче параметров, принятые в системах Windows NT и Windows'95. Если вы не планируете писать приложения на ассемблере, вам нужно это просто запомнить.

А вот переменные hInstance и hPrevInstance заслуживают более подробного обсуждения. Так как Windows NT и Windows'95 являются многозадачными системами, то очевидно, что одна и та же программа может быть запущена несколько раз. Для того чтобы различать экземпляры программ, каждому экземпляру присваивается условный номер - хэндл (handle). Справедливо ради, надо отметить, что в Win32 присваиваются хэндлы чему угодно - окну, меню, курсору, иконке и т. д. Фактически хэндл - это указатель на блок памяти, в котором размещен тот или иной объект. В заголовочных файлах тип HANDLE определен как void*, а тип HINSTANCE как HANDLE. Согласно вентерской нотации, идентификаторы переменных типа HANDLE должны начинаться с буквы h.

Уважаемый читатель! Обратите внимание на вытекающее из этого положения следствие. Раз уж объект имеет хэндл, который является УКАЗАТЕЛЕМ, то, значит, этот объект сам расположен в памяти! Другими словами, в тех случаях, когда мы должны получить хэндл того или иного объекта, фактически мы должны получить адрес загруженного в память объекта!

Но вернемся к hInstance. Когда вызывается WinMain(), Windows через эту переменную сообщает программе хэндл экземпляра программы. В Windows 3.1 hPrevInstance являлся хэндлом предыдущего экземпляра программы. Если запускается первый экземпляр программы, то параметр hPrevInstance был равен нулю. Этот факт можно использовать для того, чтобы не позволять системе запускать более одного экземпляра программы. В Win32 hPrevInstance оставлен ИСКЛЮЧИТЕЛЬНО для совместимости с предыдущими версиями Windows, он не несет никакой нагрузки и постоянно равен нулю. Так просто, как в более ранних версиях Windows, определить наличие ранее запущенного экземпляра программы не удастся. Придется нам и этот вопрос оставить на потом, до изучения основ многозадачности Windows.

Следующий параметр - pszCmdLine - представляет собой указатель на строку, ту командную строку, которая набирается после имени запускаемой программы. При необходимости программа может проанализировать этот аргумент и выполнить те или иные действия.

И последний параметр - nCmdShow - определяет, в каком виде создаваемое окно будет появляться на экране. Окно может появляться в максимизированном виде либо в виде иконки (минимизированном),

может иметь произвольный размер, определяемый программой и другие характеристики. В Win32 API определяются десять возможных значений этого параметра. Их идентификаторы начинаются с SW (вероятно, от названия функции ShowWindow, которая использует эти значения). Наиболее часто используются значения SW_SHOWNORMAL и SW_SHOWMINNOACTIVE. Возможные значения этого параметра приведены в табл. 2. Большинство идентификаторов являются самоопределенными (вряд ли, скажем, SW_SHOWMAXIMIZED приводит к отображению окна в виде иконки!). Вы можете поэкспериментировать с ними. Их полное описание можно найти в файлах системы помоши. Теперь вспомним, что перед созданием окна мы должны сначала определить его класс, поэтому у нас на очереди

Регистрация класса окна

Сразу после входа в WinMain() нам необходимо создать класс окна и сообщить о нем системе. Класс создается и регистрируется функцией RegisterClass(). Единственным аргументом этой функции является указатель на структуру типа WNDCLASS, в которой хранятся характеристики создаваемого класса. Из этого следует, что у нас добавилось головной боли - перед регистрацией класса заполнить процедуру типа WNDCLASS. В приведенной выше программе структура была определена следующим образом:

WNDCLASS WndClass;

Таблица 2. Возможные значения второго параметра функции ShowWindow()

Параметр	Значение	Параметр	Значение
SW_HIDE	0	SW_SHOWNOACTIVE	4
SW_HIDEWINDOW	0	SHOW_OPENNOACTIVE	4
SW_SHOWNORMAL	1	SW_SHOW	5
SW_NORMAL	1	SW_MINIMIZE	6
SHOW_OPENWINDOW	1	SW_SHOWMINNOACTIVE	7
SW_SHOWMINIMIZED	2	SW_SHOWNA	8
SHOW_ICONWINDOW	2	SW_RESTORE	9
SHOW_MAXIMIZED	3	SW_SHOWDEFAULT	10
SHOW_FULLSCREEN	3	SW_SHOWMAXIMIZED	10
SW_MAXIMIZE	3	SW_MAX	10