



Операционные системы и их интерфейсы

Назначение операционной системы

Физическими или аппаратными ресурсами компьютера называются физические устройства, из которых состоит компьютер. К таким устройствам относятся центральный процессор, оперативная память, внешняя память, шины передачи данных и различные устройства ввода-вывода информации.

Логическими или информационными ресурсами компьютера называются данные и программы, которые хранятся в памяти компьютера. Когда говорят обо всех ресурсах компьютера, включая как физические, так и логические ресурсы, то обычно используют термины *ресурсы компьютера* или *системные ресурсы*.

Типы операционных систем

Все программы, которые работают на компьютере под управлением операционной системы, называются **пользовательскими программами**. Совокупность пользовательских программ, которая предназначена для решения определенной задачи, называется **приложением**. Если операционная система одновременно может исполнять только одну пользовательскую программу, то она называется **однопрограммной** или **однопользовательской**. Если же под управлением операционной системы могут одновременно выполняться несколько пользовательских программ, то такая операционная система называется **мультипрограммной** или **многопользовательской**.

- ▶ Операционная система, предназначенная для работы в режиме реального времени, называется *операционной системой реального времени*. Главное отличие операционных систем реального времени заключается в их быстром реагировании на внешние события и надежности функционирования. Если пользователь, сидя у компьютера, будет только раздражен медленной или ненадежной работой операционной системы, то медленная или ненадежная работа операционной системы реального времени может вызвать поломку оборудования и аварию.

Интерфейс программирования приложений Win32 API

Интерфейс программирования приложений Win32 API представляет собой набор функций и классов, которые используются для программирования приложений, работающих под управлением операционных систем фирмы Microsoft.

Функционально Win32 API подразделяется на следующие категории:

- Base Services (базовые сервисы);
- Common Control Library (библиотека общих элементов управления);
- Graphics Device Interface (интерфейс графических устройств);
- Network Services (сетевые сервисы);
- User Interface (интерфейс пользователя);
- Windows NT Access Control (управление доступом для Windows NT);
- Windows Shell (оболочка Windows);
- Windows System Information (информация о системе Windows).

Типы данных в Win32 API

Новые простые типы данных определены как синонимы простых типов данных языка программирования C. Чтобы отличать эти типы от других типов, их имена определены прописными буквами. Общее количество простых типов данных, определенных в интерфейсе Win32 API, довольно велико. Поэтому ниже приведены определения только тех простых типов данных из этого интерфейса, которые очевидным образом переименовывают простые типы данных языка программирования C.



```
typedef char CHAR;  
typedef unsigned char UCHAR;  
typedef UCHAR *PUCHAR;  
typedef int INT;  
typedef unsigned int UINT;  
typedef void *LPVOID;  
typedef CONST void *LPCVOID;
```

Объекты и их дескрипторы в Windows

- ▶ **Объектом** в Windows называется структура данных, которая представляет системный ресурс. Таким ресурсом может быть, например, файл, канал, графический рисунок.
- ▶ Операционные системы Windows предоставляют приложению объекты трех категорий:
 - User (объекты интерфейса пользователя);
 - Graphics Device Interface (объекты интерфейса графических устройств);
 - Kernel (объекты ядра).

- ▶ Под доступом к объектам понимается возможность приложения выполнять над объектом некоторые функции. Приложение не имеет прямого доступа к объектам, а обращается к ним косвенно. Для этого в операционных системах Windows каждому объекту ставится в соответствие **дескриптор** (handle). В Win32 API дескриптор имеет тип **HANDLE**. Дескриптор объекта представляет собой запись в таблице, которая поддерживается системой и содержит адрес объекта и средства для идентификации типа объекта.

- ▶ После завершения работы с объектом его дескриптор нужно закрыть, используя функцию **closeHandle**, которая имеет следующий прототип:

```
BOOL CloseHandle(  
HANDLE hObject // дескриптор объекта  
);
```