

## **2. Семейство ОС MS Windows**

### **2.1 Общая характеристика семейства ОС Windows**

Современные версии – многозадачные (с вытесняющей многозадачностью) многопоточные ОС.

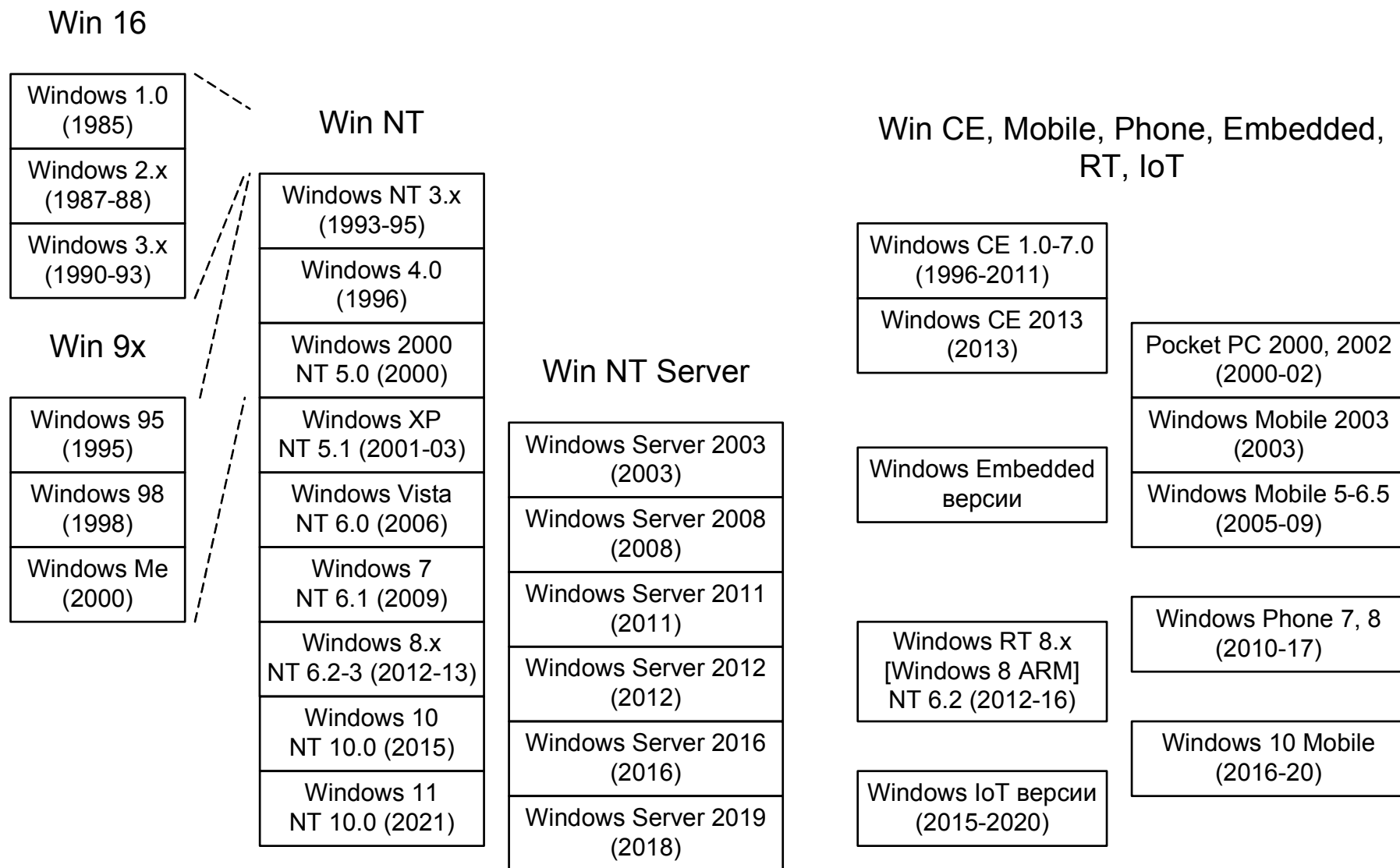
Характеристики:

- Наличие 32- и 64-разрядных версий (ранние версии были 16-разрядными, затем сочетали 16- и 32-разрядный код)
- Архитектура – «модифицированное микроядро» (сочетание микро- и макроядра)
- Симметричная многопроцессорная обработка
- Масштабируемость (серверные версии)
- Плоская (линейная) модель памяти

## Системное программирование: Семейство ОС MS Windows

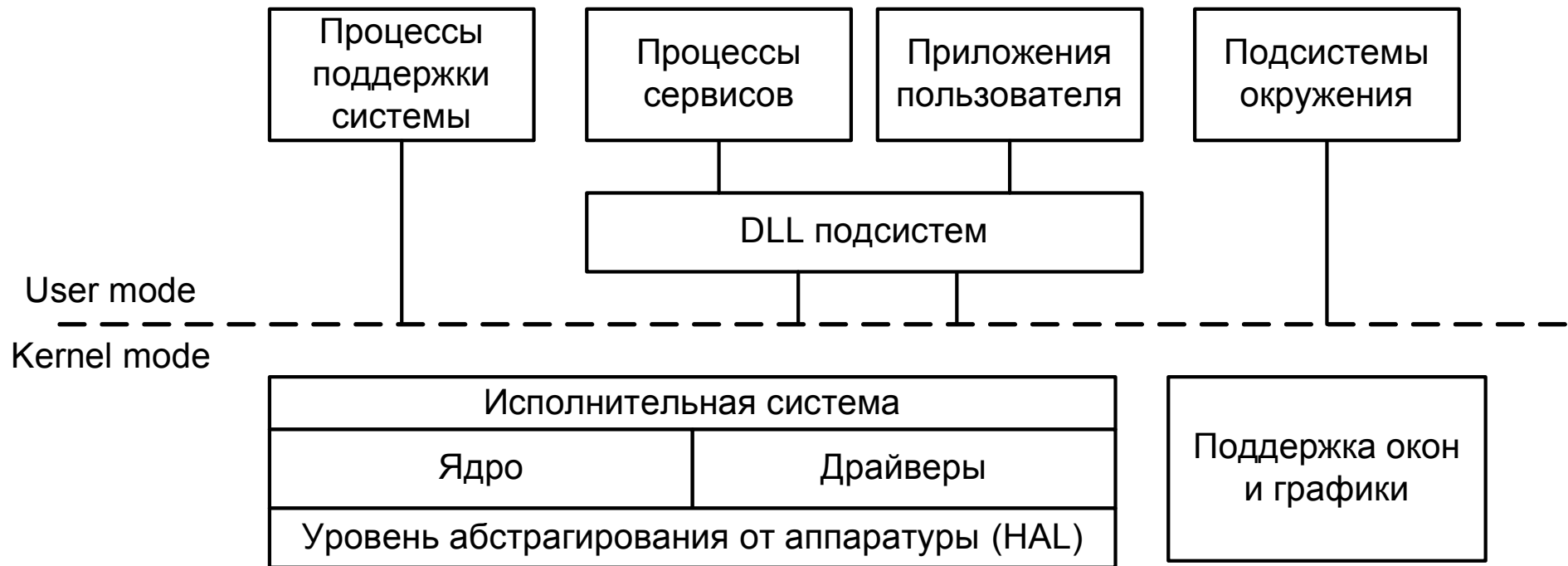
- Встроенные средства безопасности (Win NT: пользователи и группы, разграничение прав доступа)
- Централизованное управление и настройка
- Поддержка чужеродных приложений (16-разрядные DOS/Windows с ограничениями, частично POSIX и OS/2)
- Частичная переносимость (DEC Alpha, ARM)
- Совместимость или преемственность программных интерфейсов

# Системное программирование: Семейство ОС MS Windows



## Семейство ОС Windows

## 2.2 Архитектура Windows (Win NT)



Архитектура Windows NT (упрощенно)

**Исполнительная система** – базовые сервисы ОС для управления задачами, ресурсами, вводом-выводом и др.

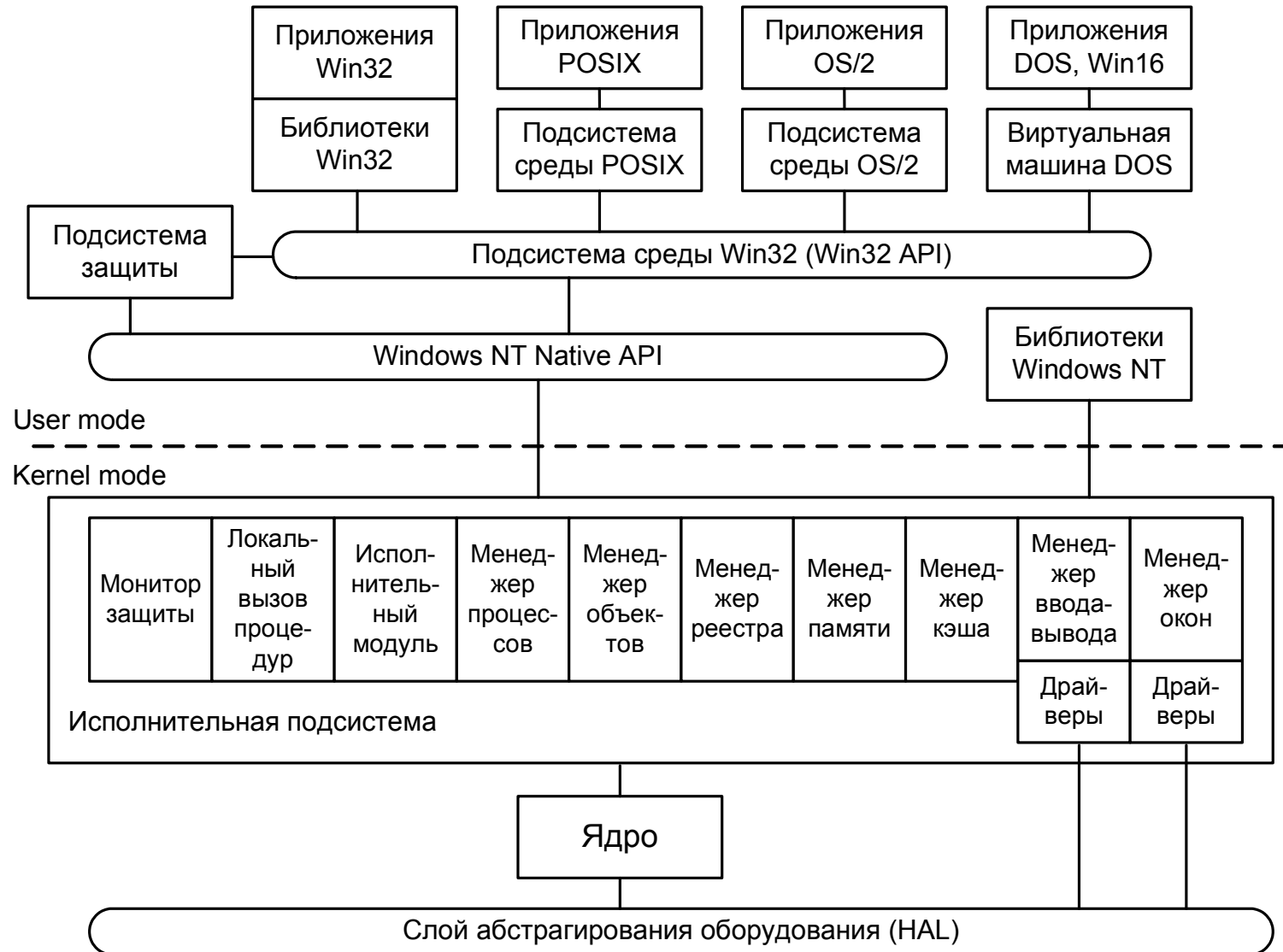
**Ядро** – реализация низкоуровневых функций ОС

***Драйверы*** – унифицированное взаимодействие с устройствами, файловой системой и т.д.

Подсистема поддержки ***окон и графики*** (реализация отделена от остальной архитектуры по соображениям производительности)

***Уровень абстрагирования от оборудования (HAL)*** – изоляция вышестоящих уровней от технической платформы

## Системное программирование: Семейство ОС MS Windows



Архитектура Windows NT (более подробно)

**Защищенные** (protected) подсистемы – подсистемы среды: Win 32 (64) и прочие («вторичные»)

Программный интерфейс Win 32 (64) API

**Неотъемлемые** (integrated) подсистемы: Win NT Native API, подсистема безопасности, некоторые другие подсистемы (например, сетевые)

Программный интерфейс Win NT Native API

Переключение User Mode – Kernel Mode

**Исполнительная** (executive) подсистема: «менеджеры» (основные функции ядра ОС)

Постепенный отход от поддержки «неродных» приложений; включение новых типов «родных» (.NET, UWP и т.д.)

**Ядро** («микроядро»).

Основные объекты, экспортируемые ядром:

- **объект-диспетчер (*dispatcher object*)** – используются для планирования и синхронизации операций: события, семафоры, мьютексы и т.д.
- **управляющий объект (*control object*)** – управление системными операциями: прерывания, отложенные и асинхронные вызовы и т.д.

Слой абстрагирования оборудования (HAL)



## 2.3 Основные особенности, концепции, объекты

### Многозадачность

Процесс, поток – основные объекты:

**Процесс (*Process*)** – соответствует выполняющейся программе, обладатель (получатель) ресурсов (от планировщика ресурсов) и контейнер потоков.

**Поток (*Thread*)** – участник планирования выполнения программ, получатель времени ЦП (от планировщика задач).

Планирование времени с разделением (квантованием) времени, вытесняющее, приоритетное.

**Нить (*Fiber*)** и «**задание**» (***Job***) – дополнительные объекты.

Иерархия объектов.

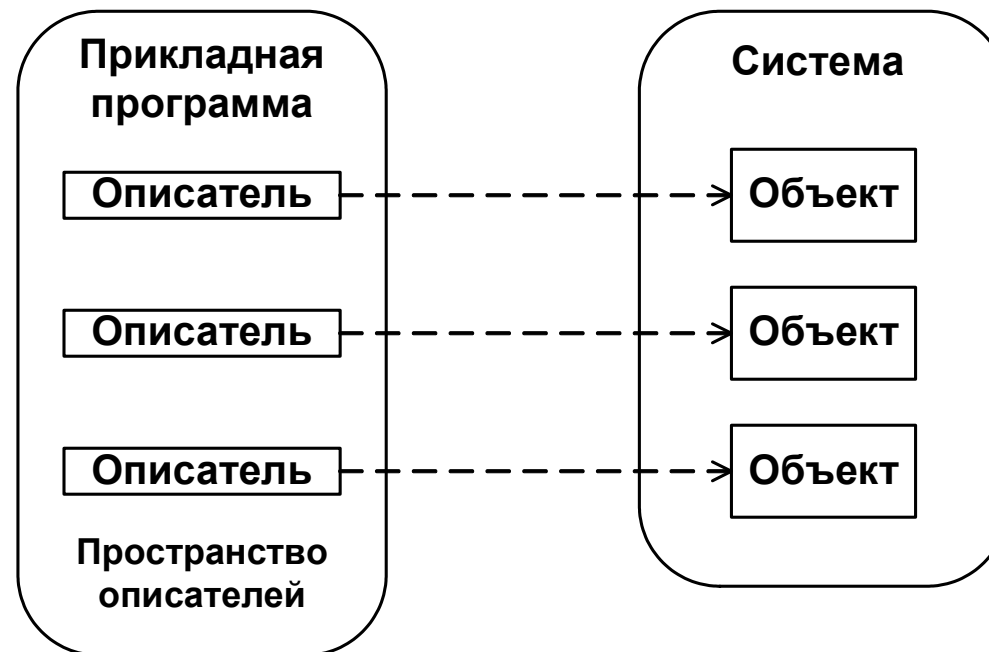
Виды процессов (программ):

- Оконные (windowed)
- Консольные (console)
- Службы/сервисы (service)
- Системные

Основной формат исполняемого двоичного файла («образа на диске») – **PE** (Portable Executable). Маркер/сигнатура “**MZP**” в заголовке.

## Программный интерфейс

Обычно определяется как **объектный** (не имеет прямого отношения к ООП): основан на использовании **объектов** и **системных вызовов** (функции). Также включает соглашения о типах и форматах данных, predetermined константы.



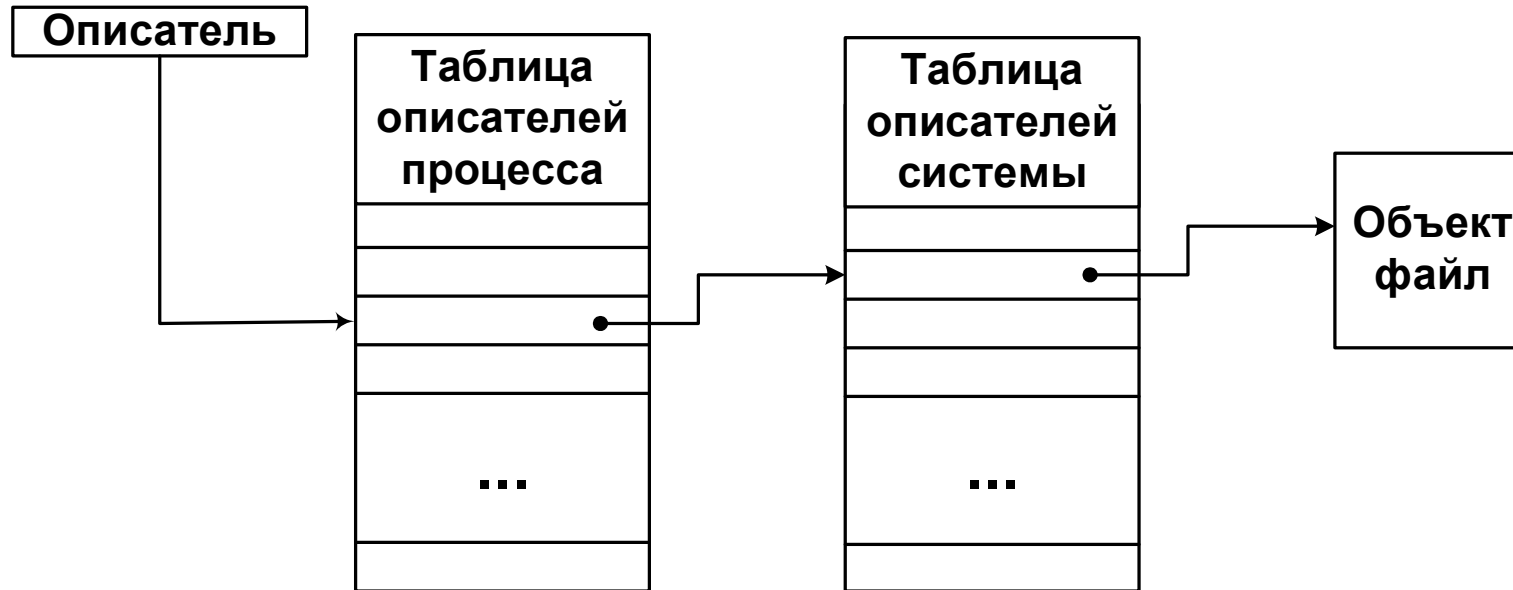
Описатели и объекты

**Объект** – программное отображение реального ресурса системы различного рода. Внутренняя структура объекта скрыта от пользователей, и прямого доступа к объекту нет.

**Описатель** (дескриптор, «хендл») – идентификатор (ссылка) объекта. Описатели входят в пространство описателей конкретного процесса. Может существовать множество описателей для одного и того же объекта, в т.ч. и в пространствах описателей разных процессов.

Примеры реализации описателей:

- тип HANDLE – синтаксически совместим с указателем (двойным указателем), но не допускает операций «взятия содержимого по адресу» (Windows)
- дескриптор (номер) файла, fd – обычно целое число (Unix и Unix-подобные)



### Трансляция описателя (дескриптора файла)

Альтернатива – поддержка структуры, описывающей реальный объект/ресурс, непосредственно в пользовательском адресном пространстве и управление им непосредственно прикладной программой. Пример – FCB (File Control Block) или MCB (Memory Control Block) в MS DOS.

Обычно принято, что объект, для которого нет ни одного описателя, удаляется – механизм **сборки «мусора»**. На уровне системы (а не, например, среды .NET!) освобождение неиспользуемых объектов происходит сразу же после утраты последнего его описателя.

Пример: функция `CloseHandle()` прекращает действие указанного `Handle`, но если этот описатель был единственным, то также происходит и удаление объекта (Windows).

В некотором роде исключением являются объекты «процесс» и «поток»: они сами владеют экземпляром своего `Handle` («локальный» `Handle`).

Объектная модель обеспечивает:

- Удобную унифицированную идентификацию системных ресурсов
- Совместное использование объектов процессами
- Защита ресурсов (содержимого объектов) от несанкционированного доступа
- Учет используемых и неиспользуемых объектов

Модель объектов и описателей (дескрипторный доступ к объектам) типична для современных ОС. Она в целом аналогична модели структур inode и имен файлов в файловых системах Unix.