

Архитектура операционных систем

Курс АОС.

Тема

«Введение»

Понятие сетевой ОС

Требования к системе

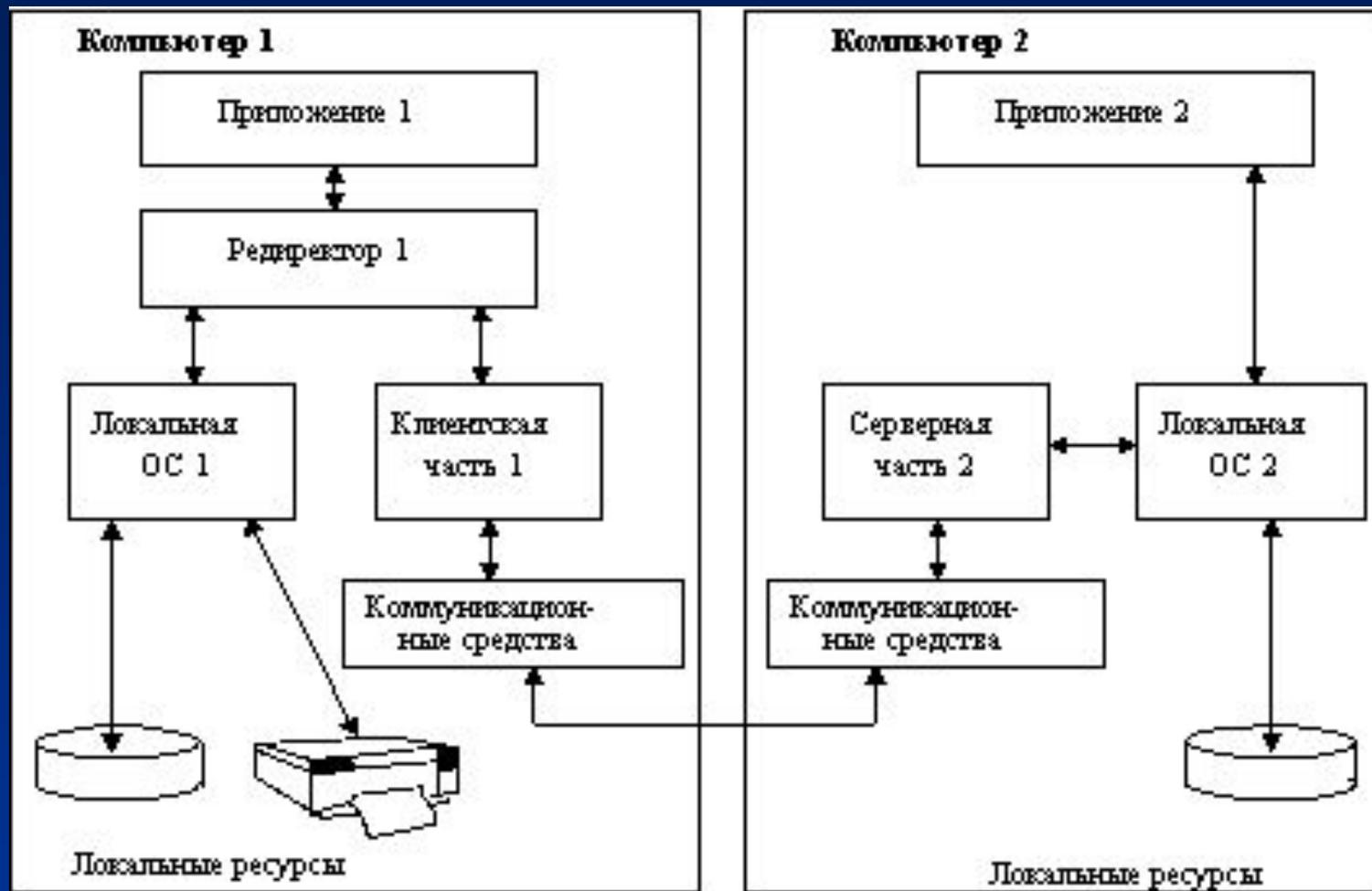
Москалев А.А.



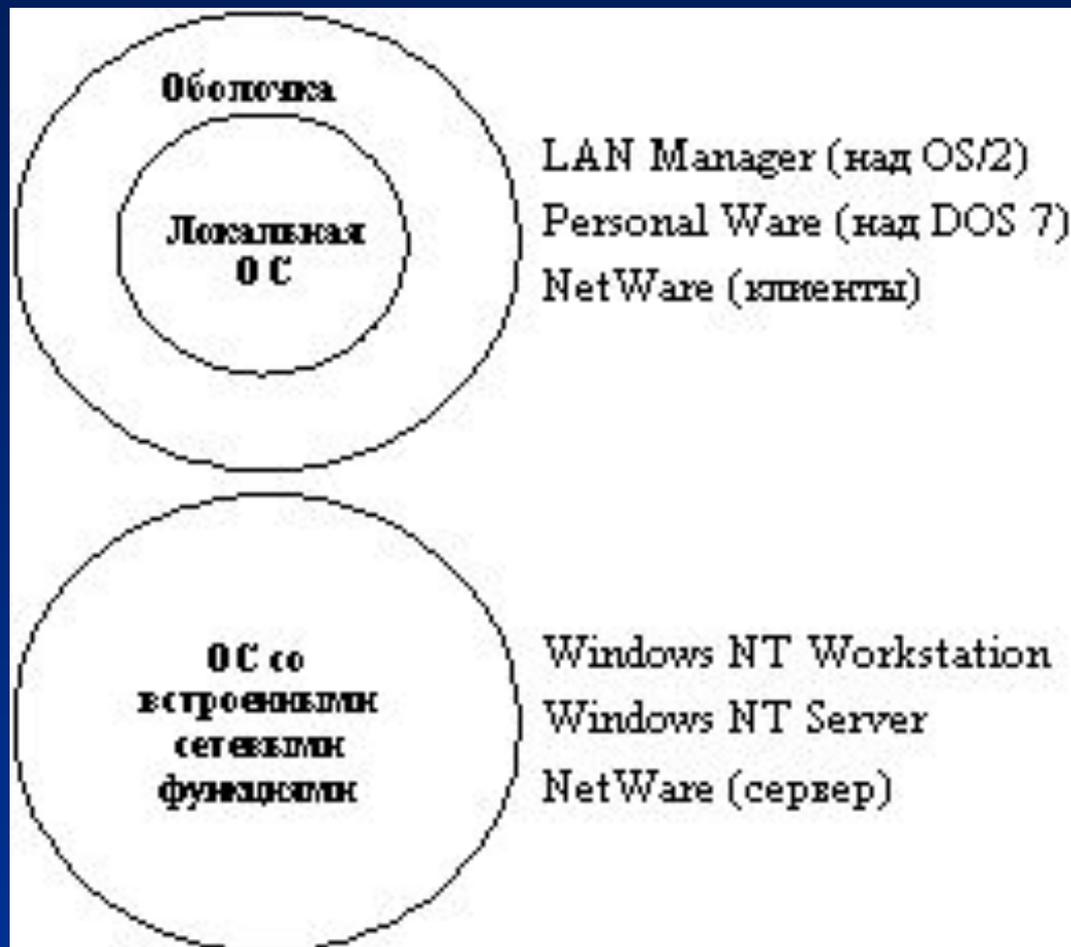
Сетевая операционная система



Сетевая операционная система (2)



Сетевая операционная система (3)



Основные требования (Goals)

- **Расширяемость (Extensibility)**
 - Приспосабливаемость к возрастающим требованиям и возможностям.
- **Переносимость (Portability)**
 - Возможность использования на разных аппаратных платформах и архитектурах.
- **Достоверность и надежность (Reliability and Robustness)**
 - Защита от внутренних ошибок.
 - Приложение не должно зависеть от других приложений или системных задач.
- **Совместимость (Compatibility)**
 - User interface и APIs должны быть совместимы со старыми версиями Windows и другими старыми системами.
 - Возможность взаимодействия с UNIX, OS/2, и NetWare.
- **Производительность (Performance)**
 - Система должна работать максимально быстро.

Переносимость (portability) Windows NT

- HAL (Hardware Abstraction Layer):
 - Поддержка для x86, MIPS, Alpha AXP(изначально), PowerPC (NT 3.51), Itanium (Windows XP/2003)
 - Машинно-зависимые функции локализованы в HAL
- Многослойная архитектура (Layered design):
 - Специфичные для данной архитектуры функции локализуются в ядре (Kernel)
- Компоненты ядра Windows пишутся преимущественно на C:
 - Исполнительная система, утилиты, драйверы
 - UI и подсистема GDI – написаны на C++
 - HW-зависимые и определяющие производительность части пишутся на ассемблере:
 - обработчики прерываний (int trap handler)
 - переключение контекста (context switching)

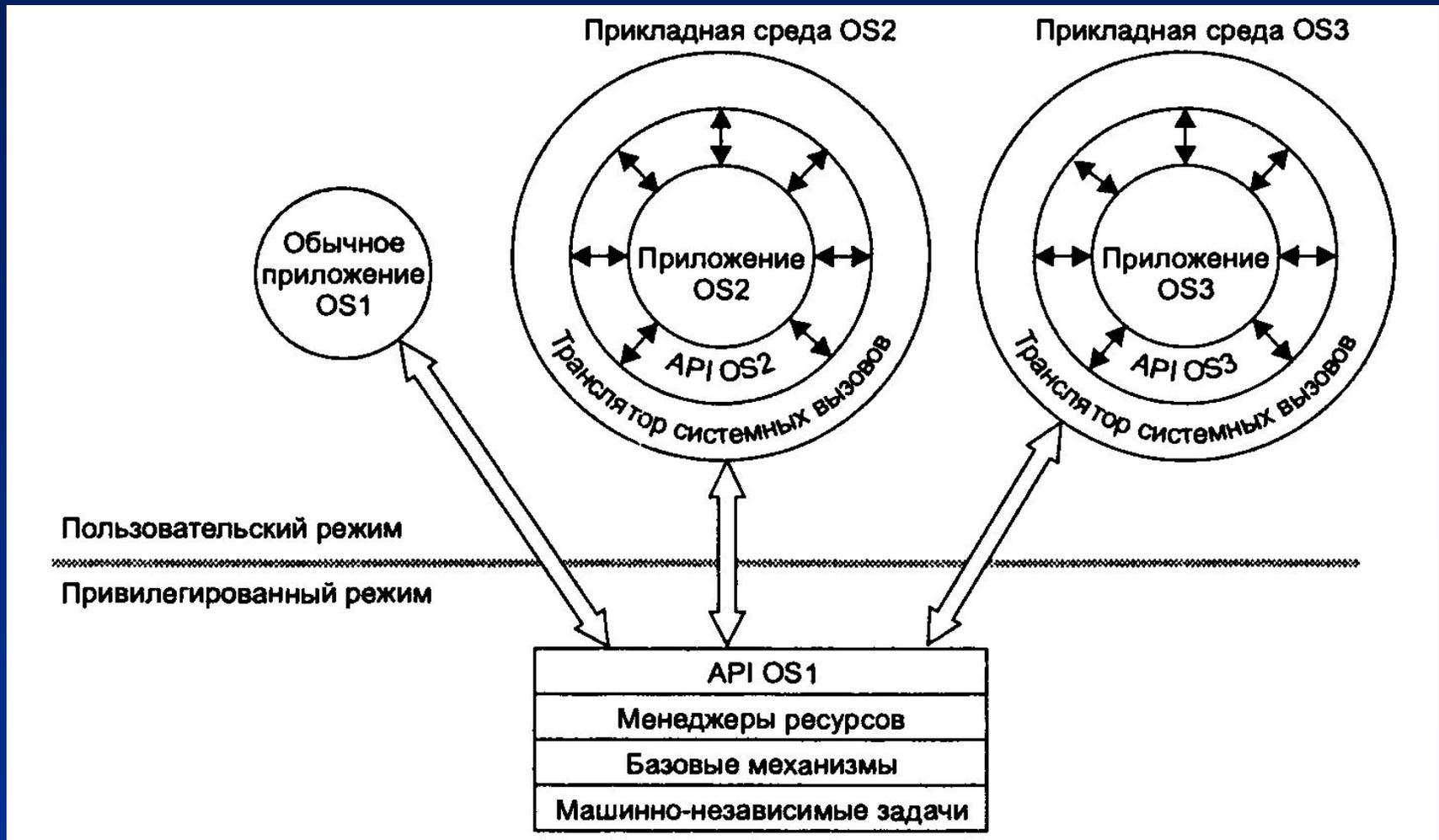
Переносимость (portability)



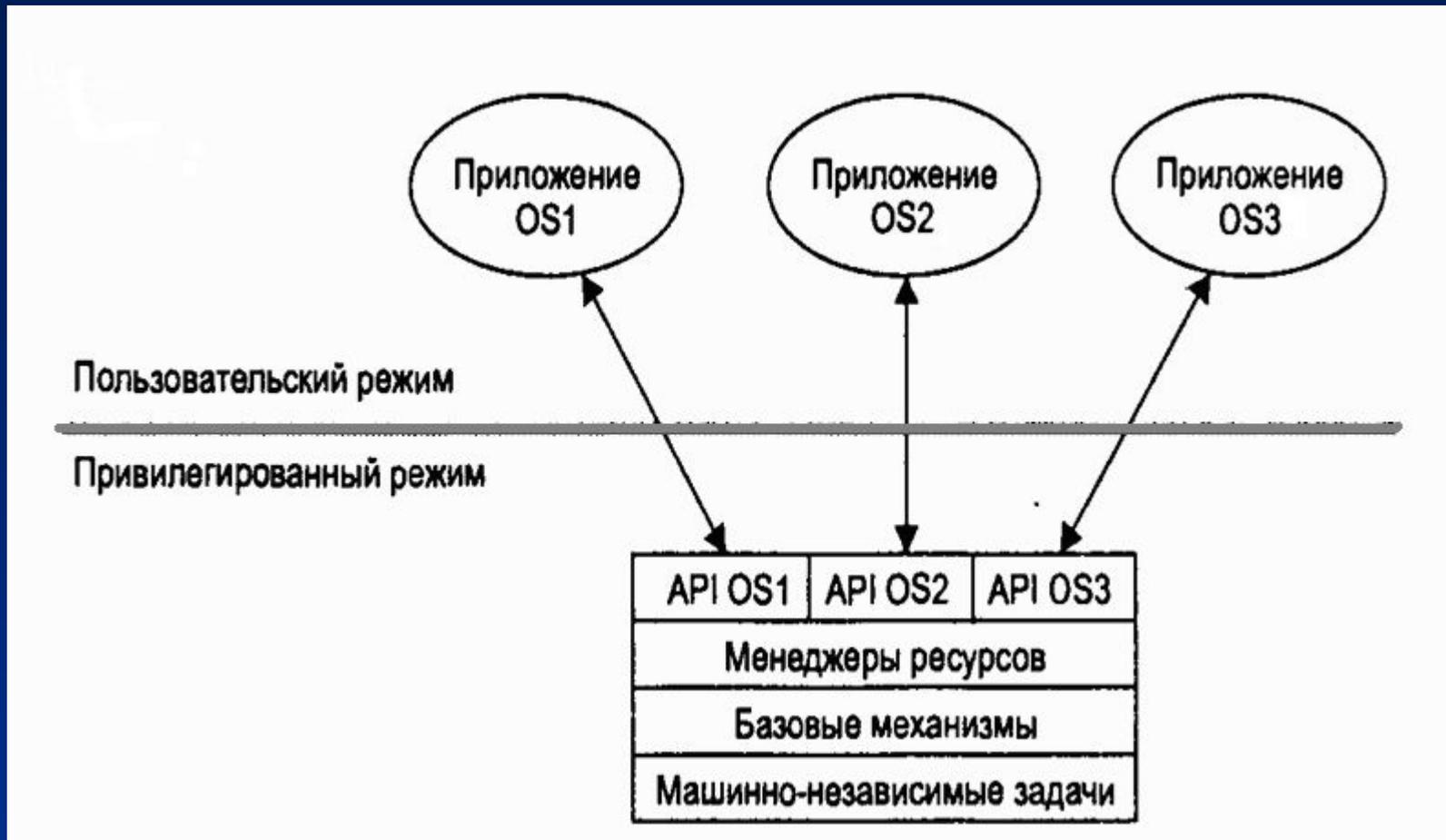
Совместимость (Compatibility)

- Операционная система ОС1 будет совместима с операционной системой ОС2 если программу, написанную для ОС2, можно будет выполнять в среде ОС1

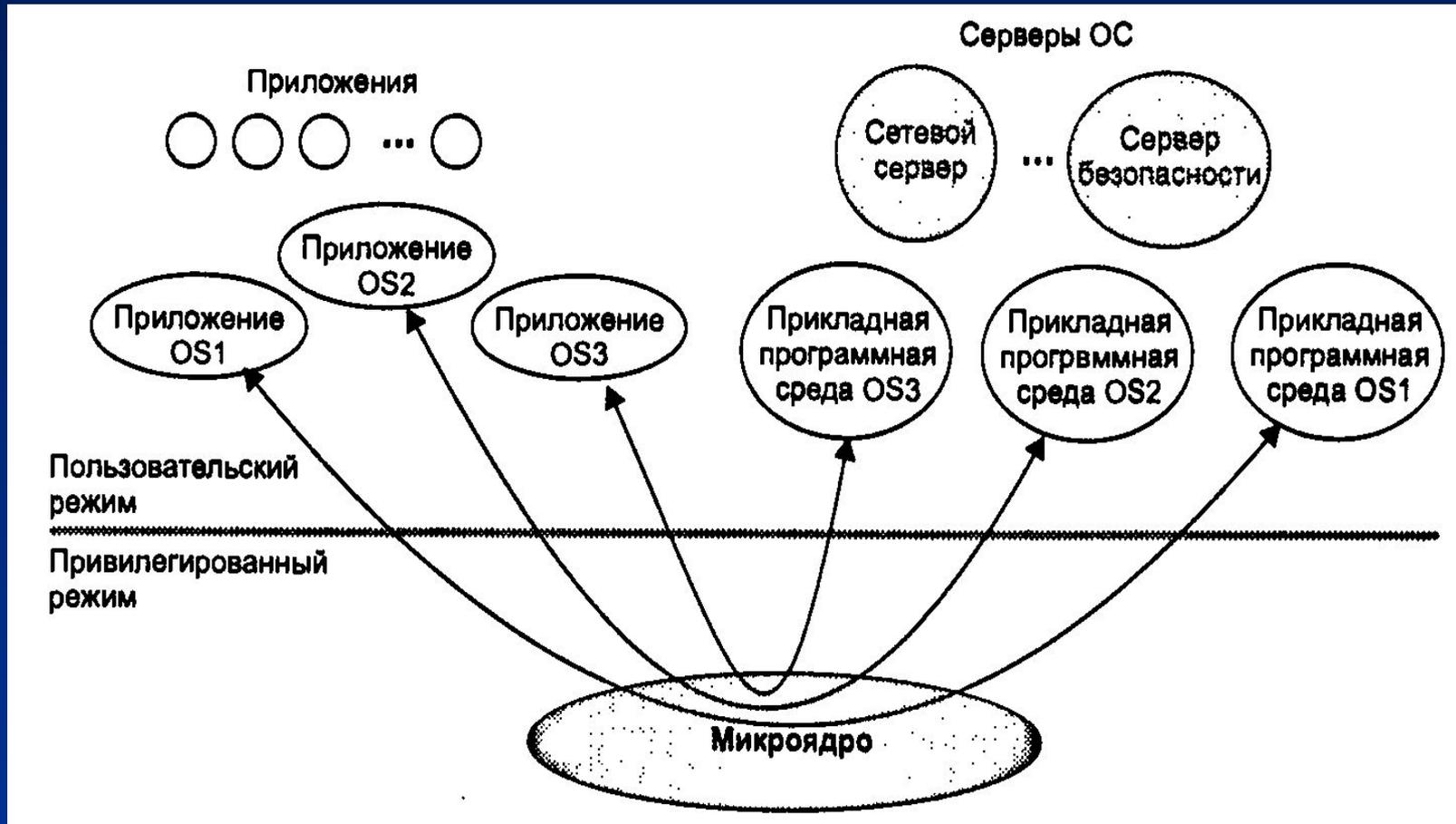
Прикладные программные среды(1)



Прикладные программные среды(2)



Прикладные программные среды(3)



Windows API и прикладные среды (subsystems)

- Windows API (application programming interface):
 - Общий прикладной интерфейс для Windows NT/2000/XP/2003, Windows 95/98/ME and Windows CE
 - ОС реализуют (разные) подмножества API
 - MSDN: <http://msdn.microsoft.com>
- Windows поддерживают несколько прикладных сред (подсистем) с разными API:
 - Windows (primary), POSIX, OS/2
 - Приложения пользовательского режима получают доступ к функциональности ОС через подсистемы
- **Подсистема(прикладная программная среда) определяет API, особенности организации процессов и семантику файловой системы**

64-bit & 32-bit Windows APIs

- 64 разрядными являются указатели и типы данных, производные от указателей, например дескрипторы (handles)
 - 64 – разр и некоторые другие, WPARAM, LPARAM, LRESULT, SIZE_T
 - Остальные остаются 32-разряда(INT, DWRD, LONG и т.д.)
- Только пять замен в API !
 - Четыре для данных класса окна (Window/Class Data)
 - (_ptr) версии вызовов
 - Обновления констант для ЭТИХ вызовов
 - Одна (_ptr) версия для свойств полосы прокрутки

API	Data Model	int	long	pointer
Win32	ILP32	32	32	32
Win64	LLP64	32	32	64
UNIXes	LP64	32	64	64

Win32 и Win64
вместе
называются
Windows API

Сервисы, функции и процедуры

- API функции Windows :
 - Документированные вызываемые из пользовательского режима процедуры
 - *CreateProcess, CreateFile, GetMessage*
- Системные сервисы Windows :
 - Недокументированные функции, вызываемые из режима пользователя
 - *NtCreateProcess* используется в *CreateProcess* (Windows) и в *fork()* (POSIX) как внутренний сервис
- Внутренние процедуры Windows :
 - Процедуры, включенные в Windows Executive, Kernel или HAL
 - Вызываемые только из режима ядра(драйверы устройств, NT OS компоненты)
 - *ExAllocatePool* выделяет память из системной кучи Windows