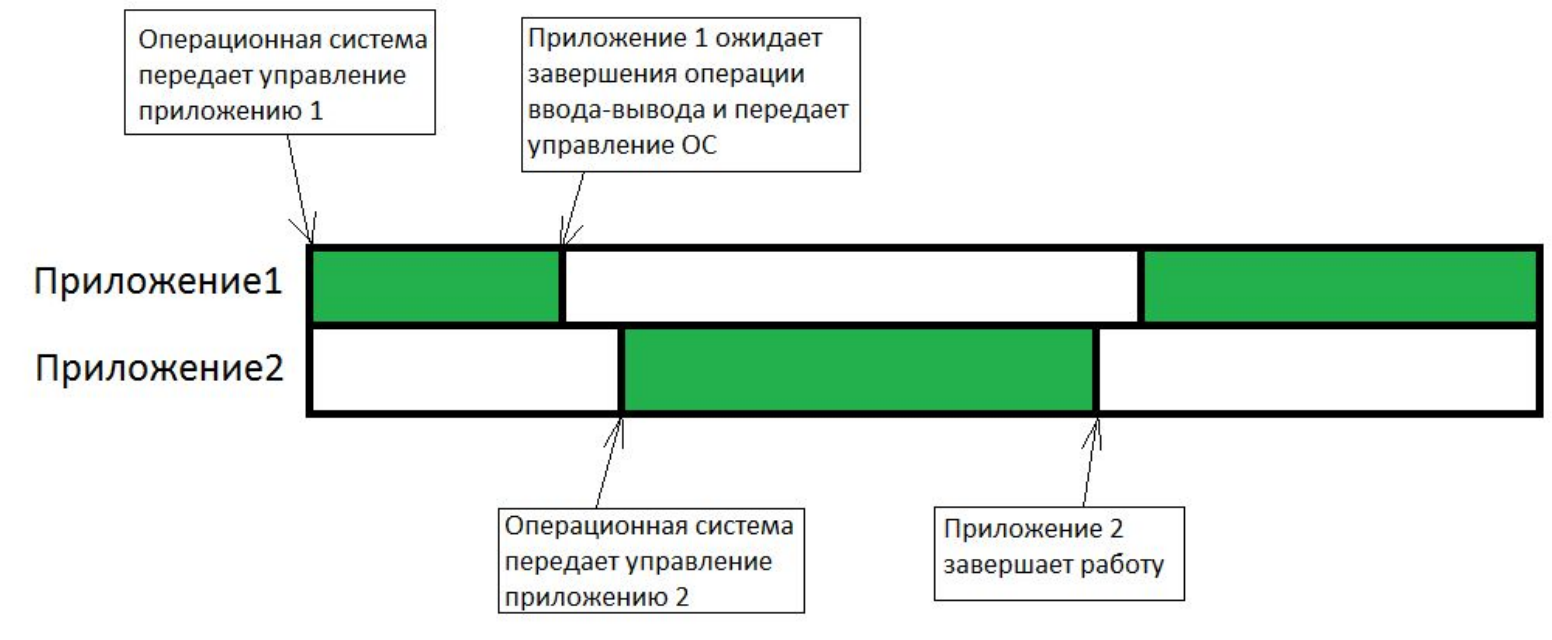
Архитектура Windows

Основные понятия

Многозадачность

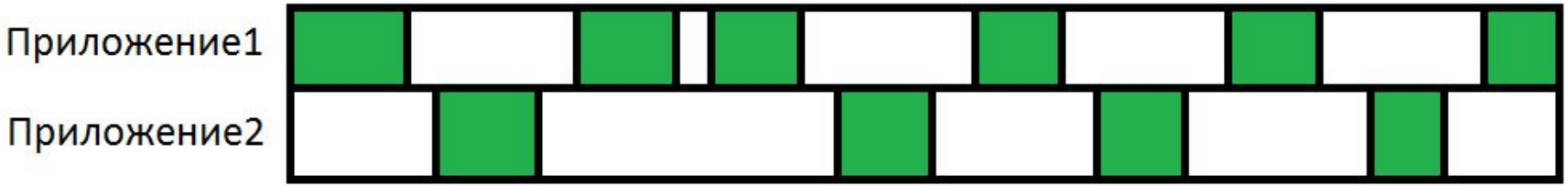
Невытесняющая многозадачность. Операционная система одновременно загружает в память два или более приложений, процессорное время предоставляется одному выбранному приложению. Выбранное приложение работает до тех пор, пока самостоятельно не освободит процессор.



Многозадачность

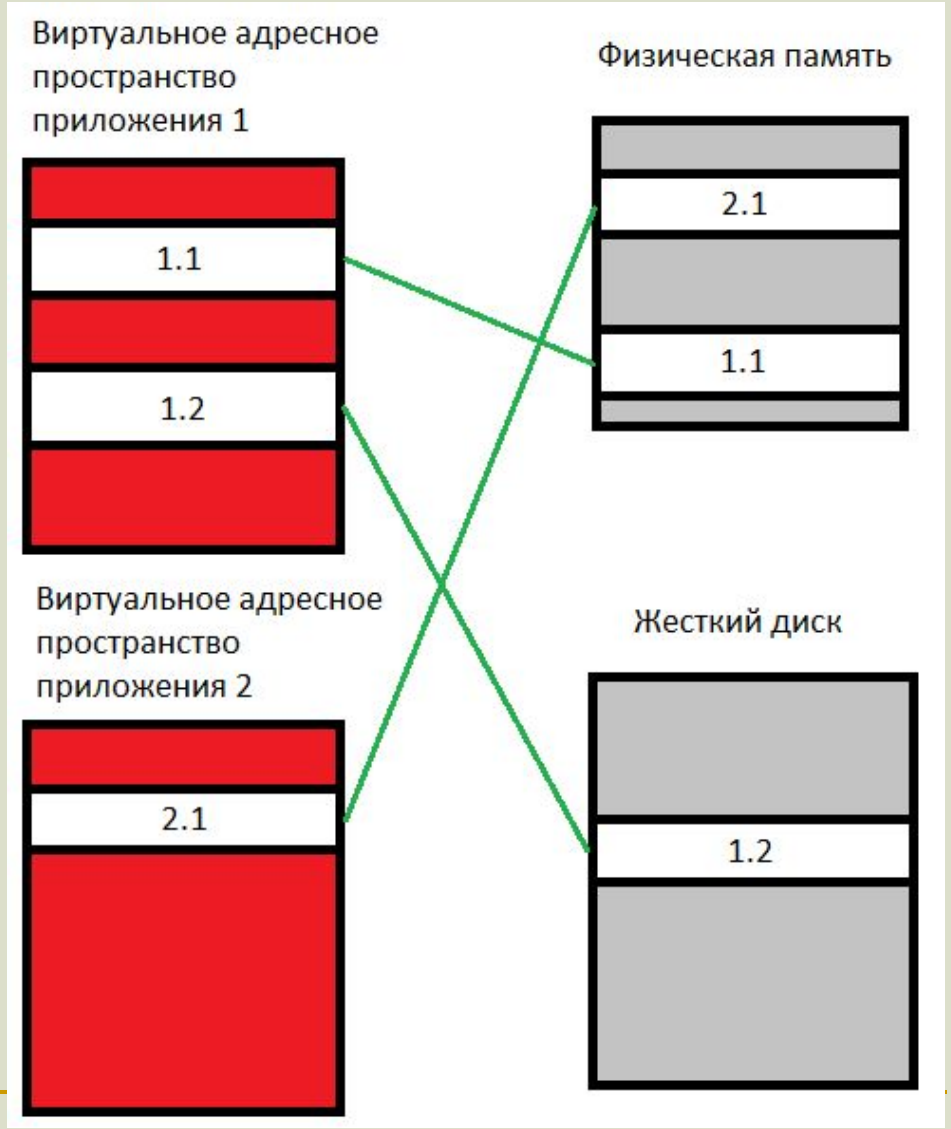
Вытесняющая многозадачность. Операционная система сама передает управление от одной выполняемой программы другой в случае завершения операций ввода-вывода, возникновения событий в аппаратуре компьютера, истечения таймеров и квантов времени, или же поступлений тех или иных сигналов от одной программы к другой.

В этом виде многозадачности процессор может быть переключен с исполнения одной программы на исполнение другой без всякого пожелания первой программы и буквально между любыми двумя инструкциями в её коде.



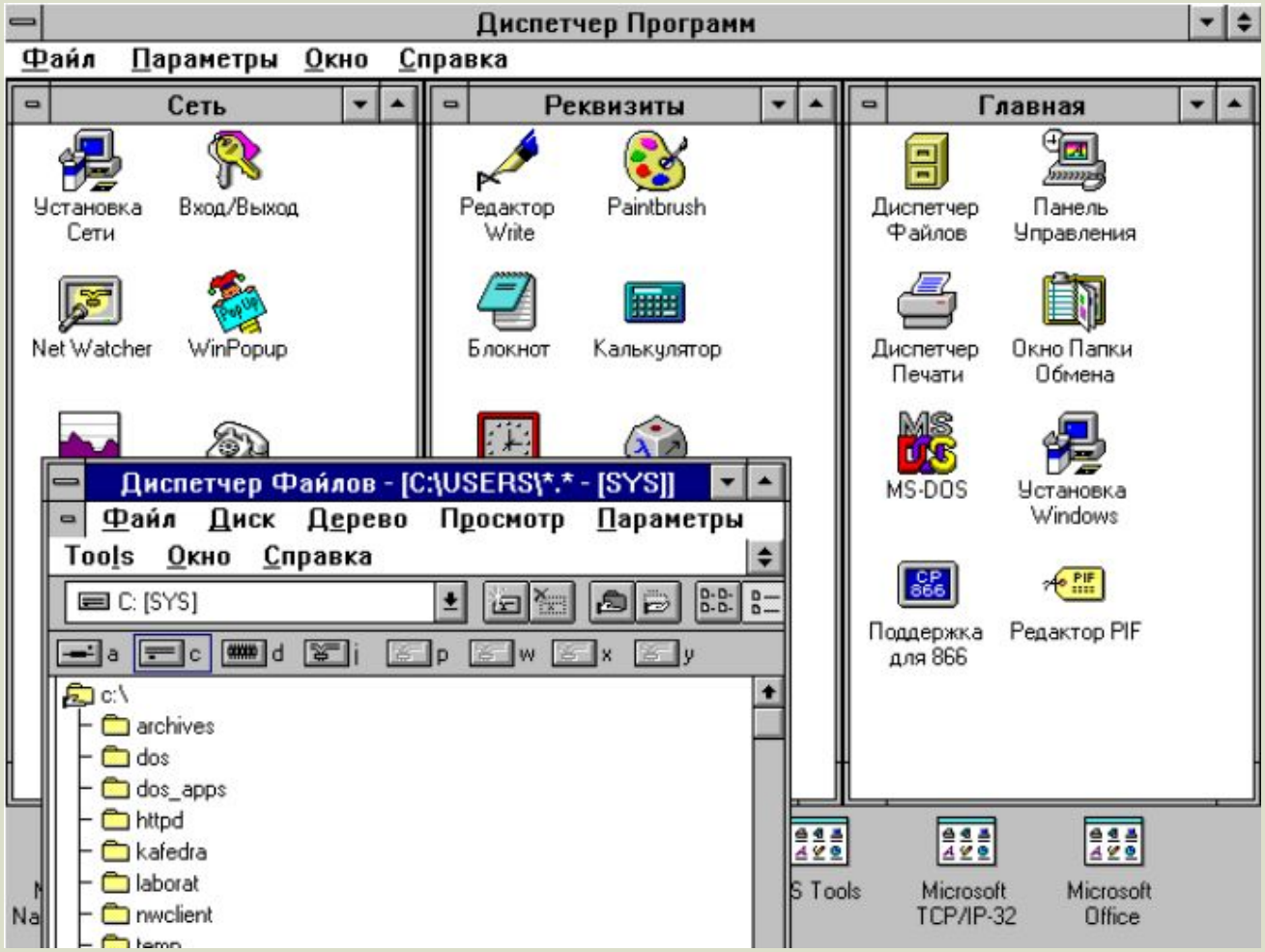
Виртуальная память

Виртуальная память — технология управления памятью ЭВМ, разработанная для многозадачных операционных систем. При использовании данной технологии для каждой программы используются независимые схемы адресации памяти, отображающиеся тем или иным способом на физические адреса в памяти ЭВМ.



Версии ОС Windows

Оболочки над DOS



Windows 1.0 (1985)
Windows 2.0 (1987)
Windows 2.1 (1987)
Windows 3.0 (1990)
Windows 3.1 (1992)
Windows 3.1/3.11(1993)

не были полноценными операционными системами

Последние версии поддерживали
- новые режимы работы процессора
- многозадачность (невытесняющую)
- работу с сетью

Версии ОС Windows

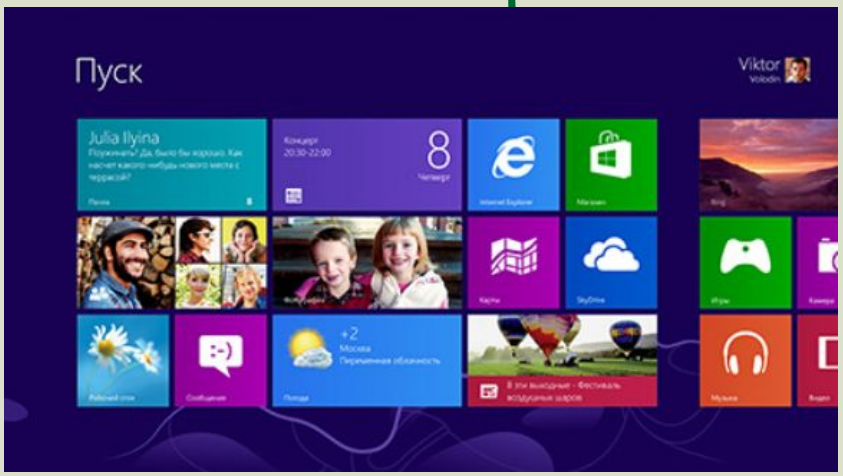
Windows 9x

Windows 95
Windows 98
Windows ME.

Практически отсутствуют средства обеспечения безопасности и отказоустойчивости

- Вытесняющая многозадачность
- Поддержка виртуальной памяти
- Поддержка TCP/IP
- Plug and Play

Версии ОС Windows



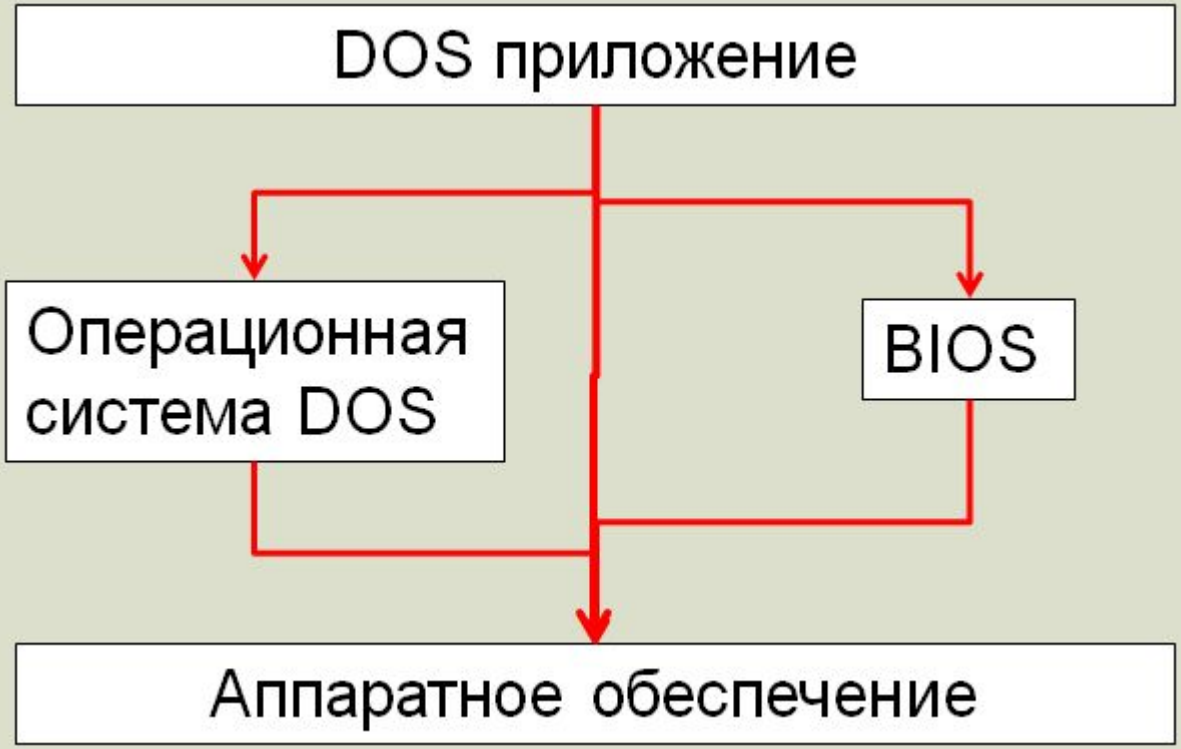
- Механизмы обеспечения безопасности
- Квоты
- Журналируемая файловая система
- Шифрование
- Большая стабильность по сравнению с Windows 9x

Windows NT

Windows NT 3.1 (1993)
Windows NT 3.5 (1994)
Windows NT 3.51 (1995)
Windows NT 4.0 (1996)
Windows 2000 (2000)(5.0)
Windows XP (2001)(5.1)
Windows XP 64 (2003)(5.2)
Windows Server 2003 (2003)(5.2)
Windows Vista (2006)(6.0)
Windows Home Server (2007)(5.2)
Windows Server 2008 (2008)(6.0)
Windows Small Business Server (2008)(6.0)
Windows 7 (2009)(6.1)
Windows Server 2008 (2009)(6.1)
Windows Home Server 2011 (6.1)
Windows 8 (2012)(6.2)
Windows Server 2012(6.2)

Понятие WinAPI

Работа с аппаратным обеспечением DOS приложения

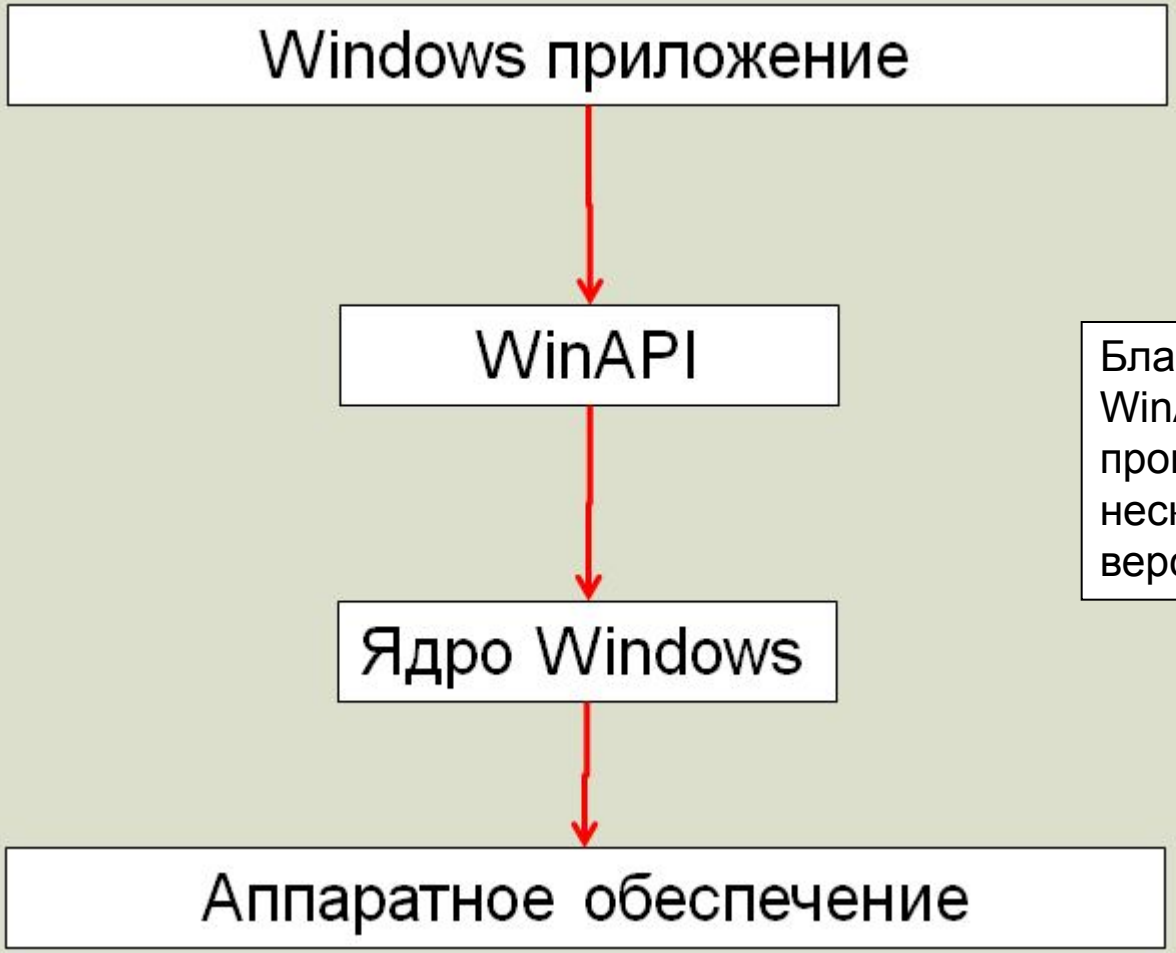


DOS приложение получает в своё распоряжение все ресурсы компьютера и может осуществлять ввод/вывод:

- через функции, предоставляемые операционной системой
- через функций базовой системы ввода/вывода (BIOS)
- работая с устройствами напрямую.

Понятие WinAPI

Общий принцип работы приложения в Windows



Значительная часть функций WinAPI поддерживается всеми версиями Windows

Благодаря использованию WinAPI большая часть программ может работать с несколькими последними версиями Windows

WinAPI предоставляет унифицированный интерфейс для работы с ядром

Понятие WinAPI

четыре типа пользовательских процессов

фиксированные **процессы поддержки системы** — например, процесс обработки входа в систему и диспетчер сеансов

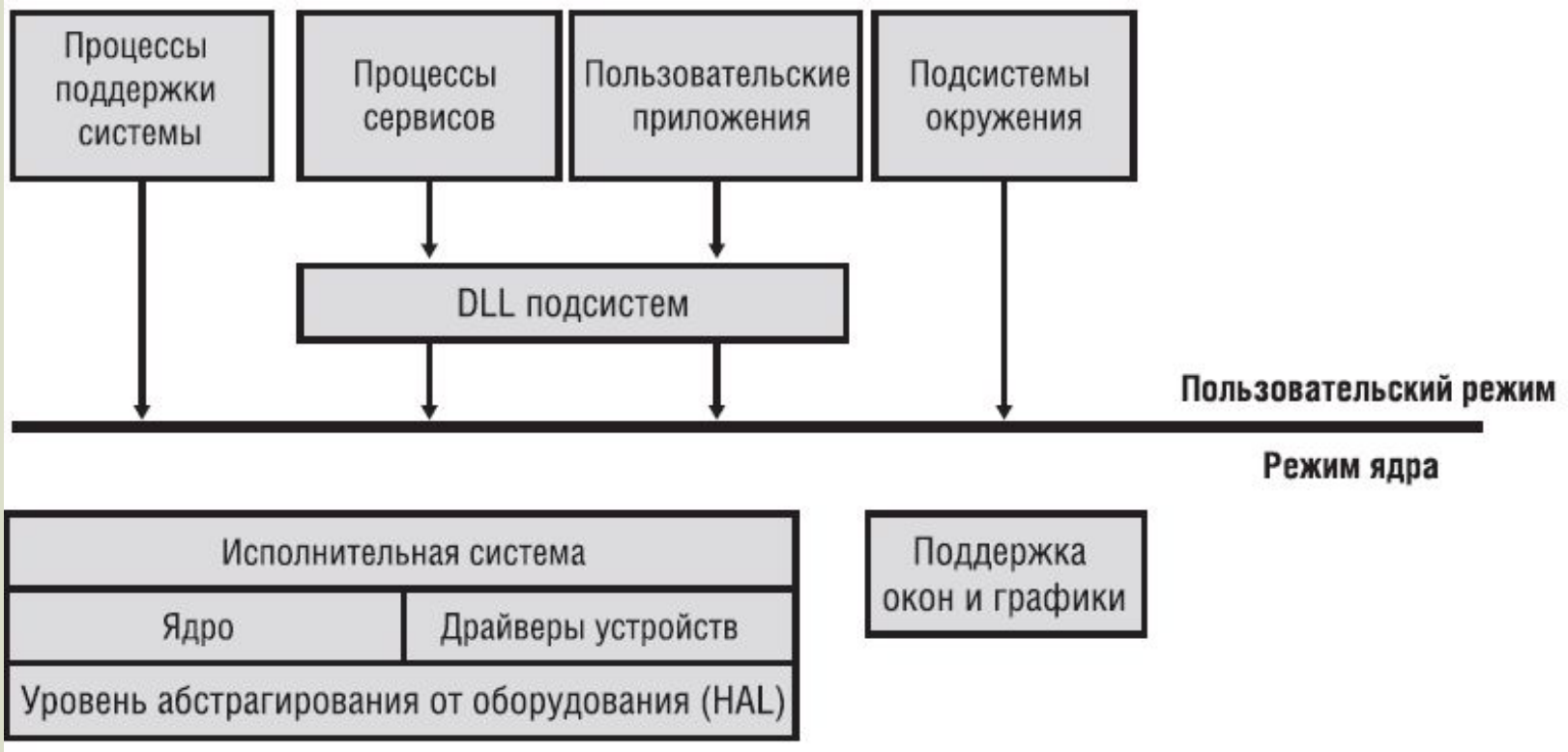
процессы сервисов (service processes) — носители Windows сервисов, такие как Task Scheduler и Spooler (диспетчер очереди печати).

пользовательские приложения

подсистемы окружения — реализованы как часть поддержки среды операционной системы, предоставляемой пользователям и программистам. Изначально Windows NT поставлялась с тремя подсистемами окружения: Windows, POSIX и OS/2. Последняя была изъята в Windows 2000. Windows XP исходно поставляется только с подсистемой Windows

Понятие WinAPI

Упрощенная схема архитектуры Windows



WinAPI реализуется с помощью dll, которые обеспечивают трансляцию документированных функций в соответствующие внутренние (и обычно недокументированные) вызовы системных сервисов Windows

Понятие WinAPI

UNICODE

WinAPI поддерживает два типа строк
- ANSI (один символ занимает один байт)
- UNICODE (один символ занимает два байта)

Большинство WinAPI функций имеет 2 реализации
- ANSI – к имени функции добавляется A
- UNICODE – к имени функции добавляется W

Например функция создания окна **CreateWindowEx** имеет две реализации
CreateWindowExA
CreateWindowExW

```
//Фрагмент файла winuser.h
#ifdef UNICODE
#define CreateWindowEx CreateWindowExW
#else
#define CreateWindowEx CreateWindowExA
#endif // !UNICODE
```

Понятие WinAPI

SDK и UNICODE

Как включить/выключить UNICODE
В самом начале программы объявить/отменить объявление символов **UNICODE** и **_UNICODE**

```
//выключаем UNICODE
#undef UNICODE
#undef _UNICODE
```

Как объявить символ UNICODE

```
wchar_t unicode_symbol;
```

Понятие WinAPI

SDK и UNICODE

Как определить строку UNICODE
Перед строкой добавить L

```
wchar_t unicode_str[100]=L"Превед медвед";
```

Макросы и определения для обеспечения совместимости
символ объявляется как **TCHAR**
указатель на строку как **LPTSTR**
строка определяется с помощью макроса **_TEXT**

```
TCHAR szMyString[100]=_TEXT("Превед медвед");
LPTSTR lpMyString=szMyString;
MessageBox(NULL,lpMyString,_TEXT("Сообщение"),MB_OK);
```

Понятие HANDLE

- Windows – многозадачная операционная система.
- Каждый процесс использует множество объектов.
- Иногда процессы совместно используют объект
- В общем случае процессу необходимо защитить свои объекты от действий других процессов

Примеры объектов:

- Файл
- Окно
- Мьютекс
- Динамически подключаемая библиотека

Процесс 1
объект1
объект2
объект3

Процесс 2
объект4
объект2
объект5

Процесс 3
объект6
объект2
объект7

Объект 2 совместно используется тремя процессами

Понятие HANDLE

- Созданием, удалением, учетом и фактической работой с объектами занимается ядро
- Каждый объект имеет уникальный идентификатор (или адрес)

Ядро
объект1
объект2
объект3
объект4
объект5
объект6
объект7

Объект	Идентификатор
объект1	0x25
объект2	0x3C
объект3	0x11
объект4	0xFA
объект5	0xE4
объект6	0x10
объект7	0x15

Понятие HANDLE

- Процессы обычно не работают с идентификаторами объектов
- Для работы с объектом процесс имеет его «описатель» (HANDLE)
- HANDLE имеет смысл только для конкретного процесса
- По факту HANDLE – целое число. Microsoft не документирует смысл его значения

Процесс 1
HANDLE1
HANDLE2
HANDLE3

HANDLE	Идентификатор
HANDLE1	0x25
HANDLE2	0x3C
HANDLE3	0x11

Пример реализации в некоторых версиях Windows

Понятие HANDLE

Ядро

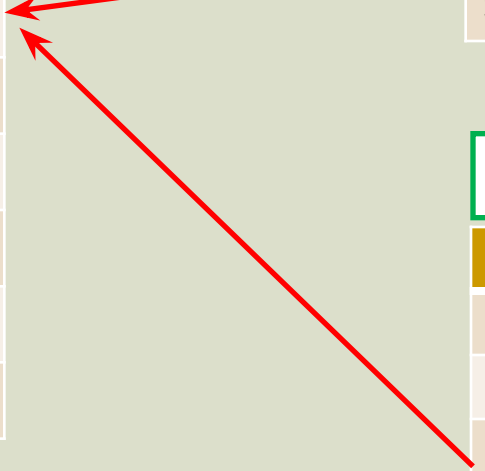
Объект	ID
объект1	0x25
объект2	0x3C
объект3	0x11
объект4	0xFA
объект5	0xE4
объект6	0x10
объект7	0x15

Процесс 1

HANDLE	ID
1	0x25
2	0x3C
3	0x11

Процесс 2

HANDLE	ID
1	0xFA
2	0xE4
3	0x3C



Процессы обращаются к объекту 2 через хэндл. Причем для Процесса 1 значение хэндла – «2», для Процесса 2 – «3».

Если Процесс 2 по ошибке будет манипулировать с неправильным хэндлом, для Процесса 1 изменения коснутся только совместных объектов