



# **ОПЕРАЦИОННЫЕ СИСТЕМЫ**

**ст. преп. Овчинников Александр Львович**



## Структура курса

«Операционные системы»:

Лекции - 28 часов;

Лабораторные занятия – 28 часов;

Самостоятельная работа – 72 часа;

**Экзамен**

# Структура курса лекционного материала:

<b>АРХИТЕКТУРА ОПЕРАЦИОННЫХ СИСТЕМ</b>	
Л1	Определение операционной системы, этапы развития ОС, типовая структура ОС. Классификация ОС
Л2	Архитектура и функции ОС. Подсистемы управления процессами, памятью, ввода-вывода, организации взаимодействия с пользователем, администрирования.
<b>УПРАВЛЕНИЕ ПРОЦЕССАМИ</b>	
Л3	Понятие процесса. Порождающий и дочерний процесс. Контекст и дескриптор процесса. Алгоритмы планирования процессов. Вытесняющие и не вытесняющие алгоритмы планирования.
Л4	Взаимодействие параллельных процессов. Понятие взаимного исключения. Организация взаимных исключений языковыми средствами. Программная реализация примитивов взаимного исключения. Алгоритм Деккера. Алгоритм Петерсона.
Л5	Взаимодействие нескольких параллельных процессов. Понятие семафоров. Тупики. Алгоритм предотвращения тупиков – алгоритм банкира. Алгоритм Медника. Обнаружение и восстановление после тупика.

# Структура курса лекционного материала:

	<b>УПРАВЛЕНИЕ ПАМЯТЬЮ</b>
Л6	Типы адресов. Распределение памяти фиксированными разделами. Распределение памяти разделами переменной величины. Перемещаемые разделы. Стратегии размещения информации в оперативной памяти.
Л7	Методы распределения памяти с использованием дискового пространства. Понятие виртуальной памяти. Способы организации виртуальной памяти. Механизмы отображения адресов.
Л8	Виртуальная память при страничной организации. Виртуальная память при сегментной организации. Сегментно-страничная организация виртуальной памяти. Алгоритмы замещения страниц.
Л9	Иерархия запоминающих устройств. Принцип кэширования данных. Объективные свойства данных. Временная и пространственная локальность.
	<b>УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ</b>
Л10	Понятие периферийных устройств. Понятие прерывания. Виды прерываний. Механизм обработки прерываний. Физическая организация устройств ввода-вывода. Организация программного обеспечения ввода-вывода. Обработка прерываний. Драйверы устройств. Независимый от устройств слой операционной системы. Пользовательский слой программного обеспечения.

# Структура курса лекционного материала:

	<b>ФАЙЛОВАЯ СИСТЕМА</b>
Л11	Понятие файловой системы. Общая модель файловой системы. Отображаемые в память файлы. Современные архитектуры файловых систем. Конкретные реализации файловых систем.
Л12	Файловая система NTFS. Имена файлов. Типы файлов. Логическая организация файла. Физическая организация и адрес файла. Права доступа к файлу.
	<b>СЕТЕВЫЕ, МНОГОПРОЦЕССОРНЫЕ ОС. ЗАЩИТА ИНФОРМАЦИИ В ОС.</b>
Л13	Сетевые и распределенные операционные системы. Взаимодействие удаленных процессов. Организации передачи информации между удаленными процессами. Особенности алгоритмов управления многопроцессорных ОС. Основные понятия информационной безопасности ОС.

# Структура курса лекционного материала:

## ОРГАНИЗАЦИЯ СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМ

Л14

Обзор современных ОС. Организация и структура ОС Windows. Объекты ядра. Управление памятью. Интерфейс прикладного программирования.

Общая характеристика семейства операционных систем UNIX. Управление процессами в UNIX. Демоны. Управление пользователями в UNIX. Файловые системы Unix. Защита информации в UNIX.

# Структура лабораторного практикума курса:

<b>Лр1</b>	<b>Изучение возможностей командной оболочки ОС Windows.</b> Цель работы: Ознакомится с командной оболочки ОС Windows. Изучить основные команды командной оболочки ОС Windows. Приобрести практические навыки написания сценариев оболочки (bat-файлов).
<b>Лр2</b>	<b>Использование программного интерфейса Win API. Процессы и потоки в ОС Windows.</b> Цель работы: Изучение возможности использования программного интерфейса приложений (API) операционных систем Windows. Приобретение практических навыков создания и управления процессами и потоками, используя Win API в средах программирования Borland Delphi и C++ Builder, Visual Studio.
<b>Лр3</b>	<b>Использование программного интерфейса Win32 API. Синхронизация процессов и потоков в ОС Windows.</b> Цель работы: Изучение программного интерфейса приложений(API) операционных систем Windows. Приобретение практических навыков синхронизации процессов и потоков, используя Win API в средах программирования Borland Delphi и C++ Builder, Visual Studio.
<b>Лр4</b>	<b>Синхронизация процессов и потоков в ОС Windows. Семафоры.</b> Цель работы: Изучение программного интерфейса приложений(API) операционных систем Windows. Приобретение практических навыков синхронизации потоков, с использованием семафоров в средах программирования Borland Delphi и C++ Builder, Visual Studio.

# Структура лабораторного практикума курса:

<b>Лр5</b>	<b>Управление памятью и обмен данными между процессами в ОС Windows.</b> Цель работы: Изучение программного интерфейса приложений(API) операционных систем Windows. Приобретение практических навыков управления памятью, изучение механизмов обмена данными между процессами в средах программирования Borland Delphi и C++ Builder, Visual Studio.
<b>Лр6</b>	<b>Изучение ОС семейства UNIX.</b> Цель работы: Ознакомится с ОС семейства UNIX. Изучить основные команды оболочки bash ОС семейства UNIX. Приобрести практические навыки написания сценариев оболочки ОС семейства UNIX.
<b>Лр7</b>	<b>Процессы в ОС семейства UNIX.</b> Цель работы: Изучить возможности создания процессов в ОС семейства UNIX. Ознакомится с функциями управления процессами ОС семейства UNIX.



## Литература:

- 1 Дейтел, Х. М.** Операционные системы [Текст] : пер. с англ. / Х. М. Дейтел, П. Дж. Дейтел, Д. Р. Чофнес ; ред. **С. М. Малявко.** - 3-е изд. - М. : БИНОМ, 2011. - ISBN 0-13-182827-4. Т. 1 : Основы и принципы. - 2011. - 1024 с.
- 2 Таненбаум, Э.** Операционные системы: разработка и реализация [Текст] : пер. с англ. / Э. Таненбаум. - СПб. и др. : Питер, 2006. - 576 с.
- 3 Столингс В.** Операционные системы, 4-е издание / В. Столингс – К.: Вильямс, 2002. - 848 с. : ил.
- 4 Таненбаум Э.** Современные операционные системы 2-е издание.: / Э. Таненбаум. – СПб: Питер, 2005. – 1038 с. ил.

# Литература:

5

**Олифер В.Г.** Сетевые операционные системы / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2001. – 544 с.

6

**681.3.066 В 467**

**Вильямс А.**

Системное программирование в Windows 2000 для профессионалов/ А. Вильямс; [Пер с англ. П Анджан]. -СПб.;М.;Харьков: Питер, 2001.-621,[3] с. :ил + 1 CD-ROM. -(Для профессионалов). -Алф. указ.: с. 608-621 . -ISBN 5-272-00152-4. -1-57610-280-7 (англ.): 43.0043.00

7

**681.3.066(075) Р 581**

**Робачевский А.М.**

Операционная система UNIX: Учеб. пособие для студ. вузов/ А.М. Робачевский. - СПб.: ВHV-Санкт-Петербург, 2001.-514,[4] с. :ил. -Предм. указ.: с. 509-514. -ISBN 5-8206-0030-4: 18.42грн.

8

**681.3.066(083) Р 358**

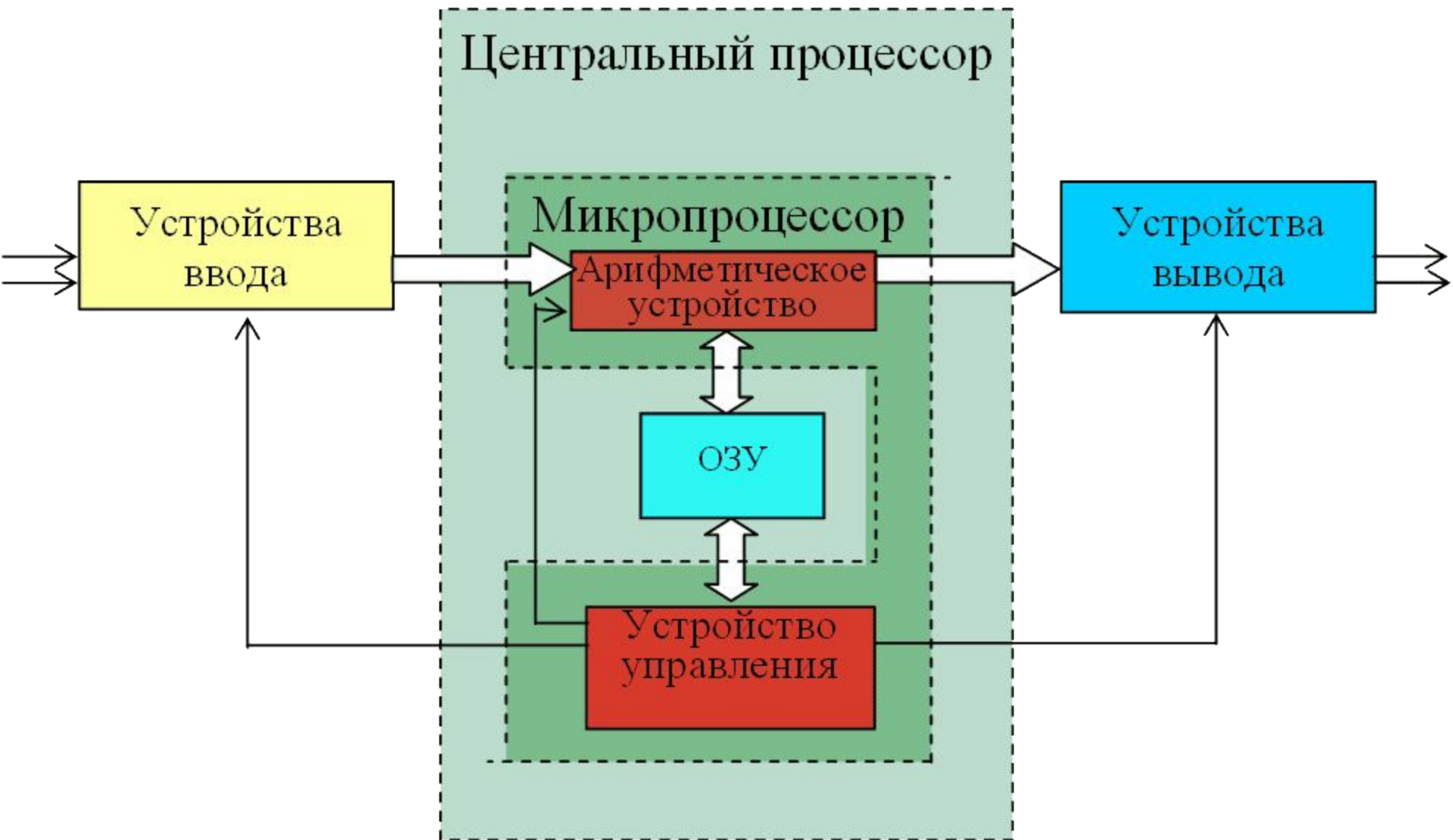
**Рейчард К.**

Unix : Справочник/ К. Рейчард, Э. Фостер-Джонсон. -2-е изд. -СПб.: Питер, 2000 374,[10] :с. -(Справочник). -Алф. указ.: с. 363-374. -ISBN 5-314-00190-X. -1-55828-549-0 (англ.): 13.30грн

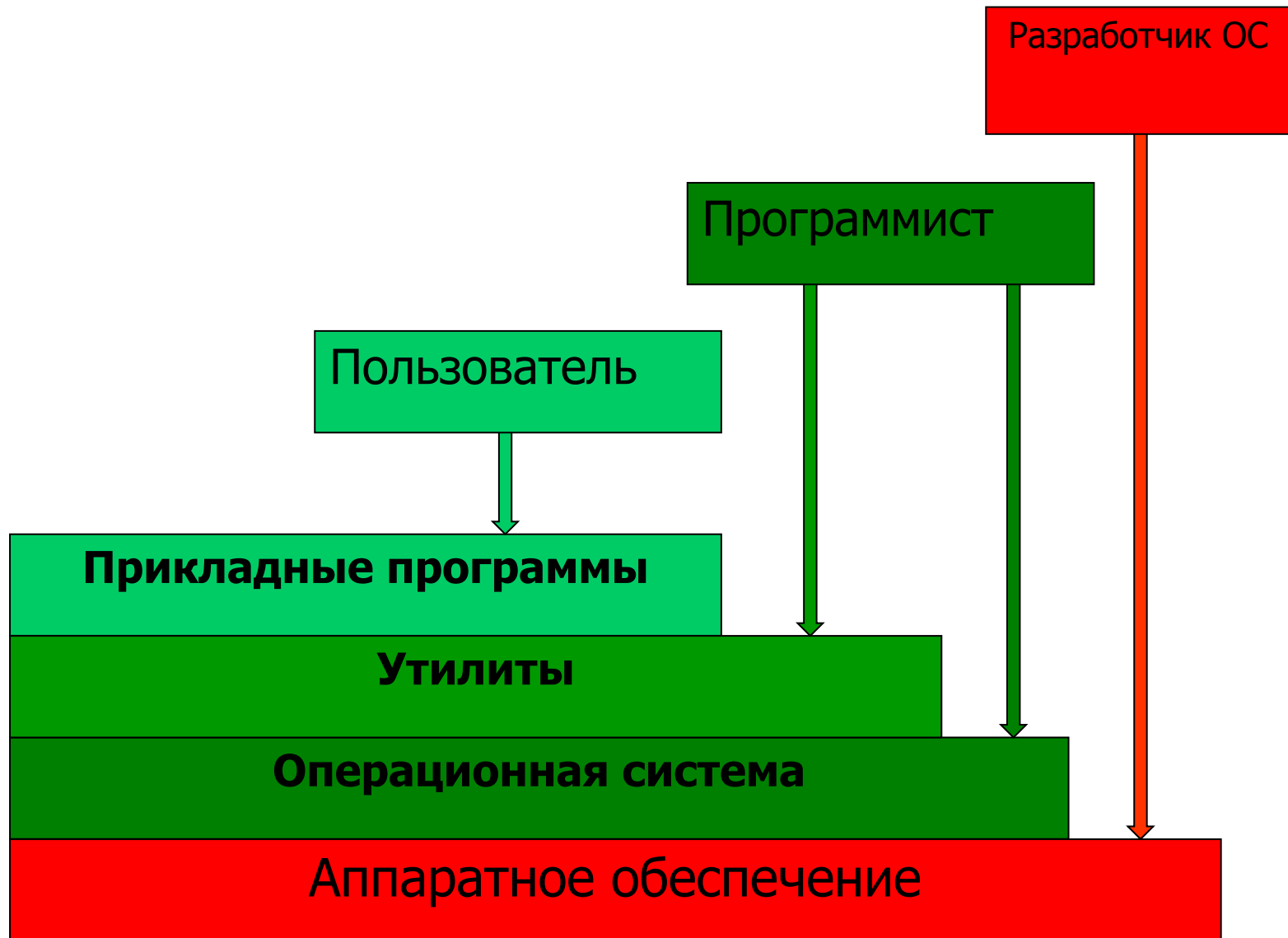
# Литература:

9	<p><b>681.3.066(083) P 358</b></p> <p><b>Рейчард К.</b></p> <p>Linux : Справочник: Пер.с англ./ К.Рейчард, П.Фолькердинг. -М.;СПб.;Харьков: Питер, 2001.-473,[7] с. :ил. -Алфавитный указатель. -<b>ISBN 5-314-00186-1</b>: 10.10грн.</p>
10	<p><b>681.3.066 М 17</b></p> <p><b>Максвелл С.</b></p> <p>Ядро Linux в комментариях : Пер. с англ./ С. Максвелл. -К.: Издательство "ДиаСофт", 2000.-484,[4] с. :ил + 1 CD-ROM. -<b>ISBN 966-7393-46-1</b> (рус.). -1-57610-469-9 (англ.): 49.1049.10</p>

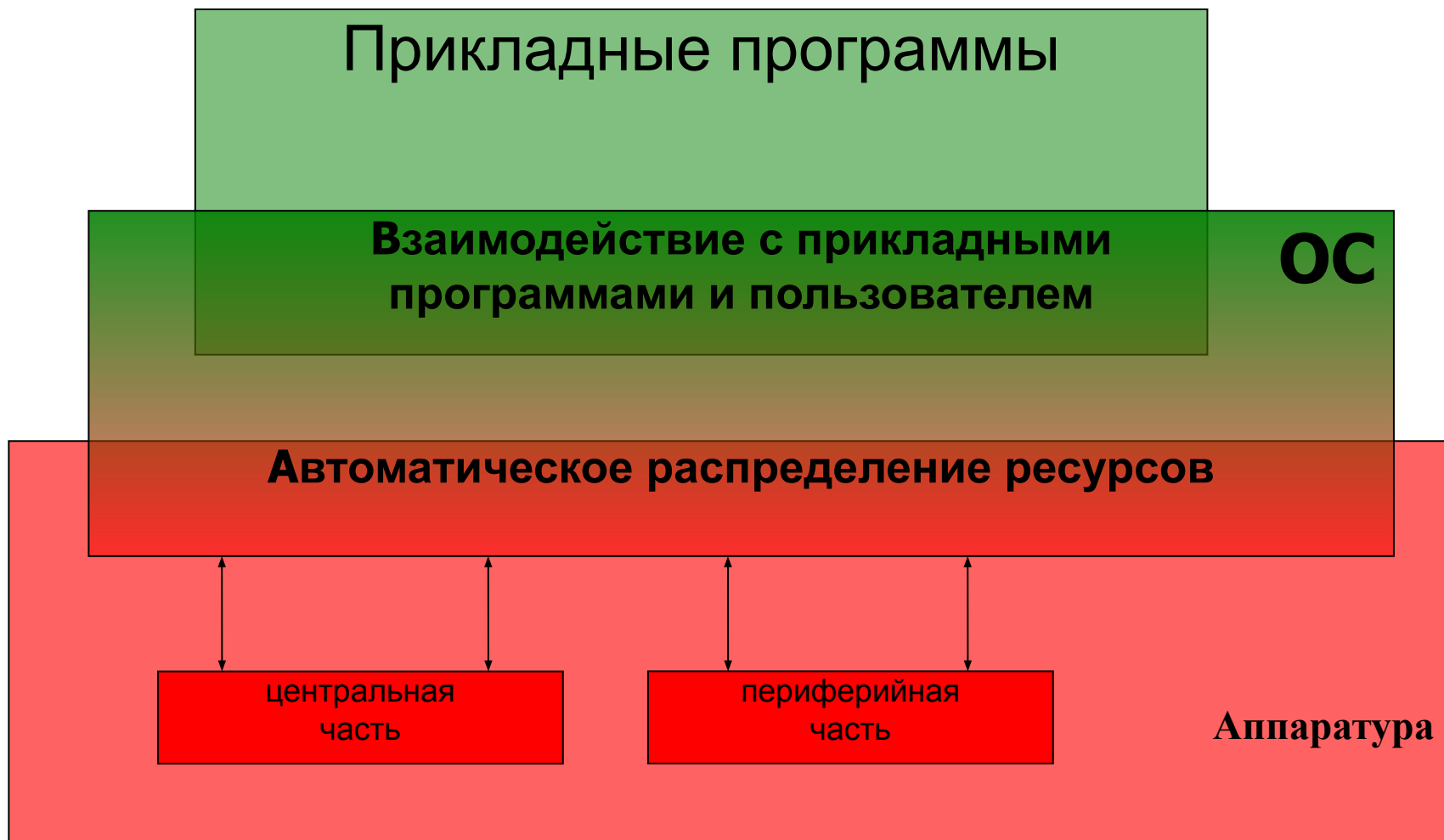
# Лекция 1: Введение в ОС



# Лекция 1: Введение в ОС



# Лекция 1: Введение в ОС



# Определение ОС

ОС – это набор программных средств, выполняющих функции *обеспечения интерфейса пользователя* с оборудованием (Hardware) и *повышения эффективности функционирования* вычислительной системы путем рационального управления ее ресурсами.

*Операционная система как виртуальная машина*

- Обеспечивает интерфейс между пользователями и компьютером
- Скрывает особенности оборудования (скрываются детали обработки прерываний, управления процессором, управления памятью (например, создаётся иллюзия неограниченного размера оперативной памяти) и т.п.

*Операционная система как менеджер ресурсов обеспечивает*

- Упорядоченное и контролируемое распределение процессоров, памяти и других ресурсов между различными программами с минимизацией простоя оборудования
- Буферизацию(кеширование) данных на диске, для организации более эффективного функционирования подсистемы памяти
- Управление ресурсами и их защиту для многопользовательских компьютеров

*Операционная система как защитник пользователей и программ*

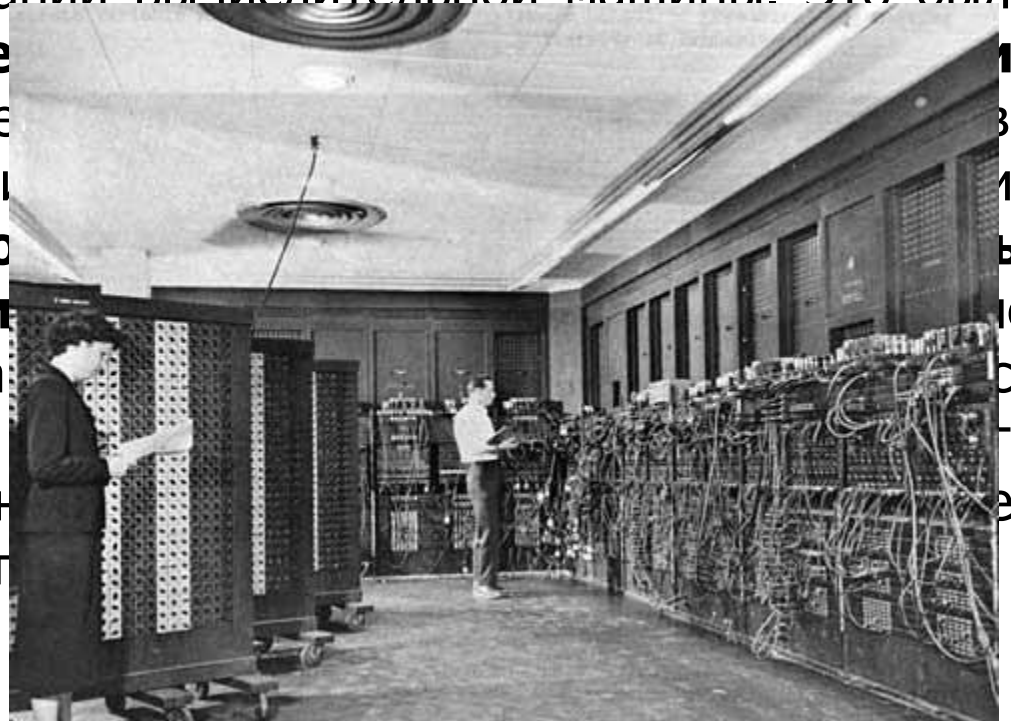
- Проблема обеспечения безопасной совместной работы нескольких пользователей
- Никто не должен удалять или повреждать чужие файлы
- Программы одних пользователей не должны произвольно вмешиваться в работу программ других пользователей
- Должны пресекаться попытки несанкционированного использования вычислительной системы
- Операционная система организует безопасную работу пользователей и их программ



## Эволюция ОС

### Первый период (1945 -1955)

Некоторый **прогресс в создании цифровых вычислительных машин** произошел после второй мировой войны. В **середине 40-х** были созданы **первые ламповые вычислительные устройства**. В то время одна и та же группа людей участвовала и в проектировании, и в эксплуатации, и в программировании вычислительной машины. Это была скорее **научно-исследовательская вычислительная техника**, а не инструмент решения каких-либо прикладных областей. **Происходило исключительно на машинном уровне** было и речи, все задачи организационно выполнялись вручную каждым программистом без другого системного программиста математических и служебных подпрограмм.



## Эволюция ОС

### Второй период (1955 - 1965)

С **середины 50-х годов** начался новый период в развитии вычислительной техники, связанный с появлением новой технической базы - **полупроводниковых элементов**. Компьютеры второго поколения стали более надежными, теперь они смогли непрерывно работать настолько долго, чтобы на них можно было возложить **выполнение действительно практически важных задач**. Именно в этот период произошло **разделение персонала на программистов и операторов, эксплуатационщиков и разработчиков вычислительных машин**.

В эти годы **появились первые алгоритмические языки**, а, следовательно, и **первые системные программы - компиляторы**. Стоимость процессорного времени возросла, что потребовало уменьшения непроизводительных затрат времени между запусками программ.

**Появились первые системы пакетной обработки**, которые просто автоматизировали запуск одной программы за другой и тем самым увеличивали коэффициент загрузки процессора. **Системы пакетной обработки явились прообразом современных операционных систем**, они стали первыми системными программами, предназначенными для управления вычислительным процессом. В ходе реализации систем пакетной обработки был разработан формализованный язык управления заданиями, с помощью которого программист сообщал системе и оператору, какую работу он хочет выполнить на вычислительной машине.

## Эволюция ОС

### Третий период (1965 - 1980)

Следующий важный период развития вычислительных машин относится к 1965-1980 годам. В это время в технической базе произошел **переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхемам**, что дало гораздо больше возможностей для построения компьютеров.

Для этого периода характерны **совместимых машин**. Первые компьютеры, построенные на интегральных микросхемах, были построены в начале 60-х годов. Построенное в начале 60-х годов компьютерное оборудование второго поколения по критерию совместимых машин стала общепринятой (например, серия ЕС ЭВМ)

**Программная совместимость систем.** Такие операционные системы использовались в малых вычислительных системах, периферии, в коммерческой **Операционные системы, построенные с учетом противоречивым требованиям**,

состояли из многих миллионов ассемблерных строк, написанных тысячами программистов, и содержали тысячи ошибок, вызывающих нескончаемый поток исправлений. В каждой новой версии операционной системы исправлялись одни ошибки и вносились другие.



# Эволюция ОС

Однако, несмотря на необозримые размеры и множество проблем, **OS/360 и другие ей подобные операционные системы машин третьего поколения действительно удовлетворяли большинству требований потребителей.** Важнейшим достижением ОС данного поколения явилась реализация **мультипрограммирования**. *Мультипрограммирование* - это способ организации вычислительного процесса, при котором на одном процессоре попеременно выполняются несколько программ. Пока одна программа выполняет операцию ввода-вывода, процессор не простаивает, как это происходило при последовательном выполнении программ (однопрограммный режим), а выполняет другую программу (многопрограммный режим). При этом каждая программа загружается в свой участок оперативной памяти, называемый разделом.

Другое нововведение - **спулинг (spooling - Simultaneous Peripheral Operation On Line** – одновременная работа с периферийными устройствами в интерактивном режиме). Спулинг (реализация ввода-вывода без участия центрального процессора) - способ организации вычислительного процесса, в соответствии с которым задания считывались с перфокарт на диск в том темпе, в котором они появлялись в помещении вычислительного центра, а затем, когда очередное задание завершилось, новое задание с диска загружалось в освободившийся раздел.

Наряду с мультипрограммной реализацией систем пакетной обработки появился новый тип ОС - **системы разделения времени**. Вариант мультипрограммирования, применяемый в системах разделения времени, нацелен на создание для каждого отдельного пользователя иллюзии единоличного использования вычислительной машины.

## Эволюция ОС

### Четвертый период (1980 - настоящее время)

Следующий период в эволюции операционных систем связан с **появлением больших интегральных схем (БИС)**. В эти годы произошло резкое возрастание степени интеграции и удешевление микросхем. Компьютер стал доступен отдельному человеку, и наступила эра персональных компьютеров. С точки зрения архитектуры персональные компьютеры ничем не отличались от класса миникомпьютеров типа PDP-11, но вот цена у них существенно отличалась. Если миникомпьютер дал возможность иметь собственную вычислительную машину отделу предприятия или университету, то персональный компьютер сделал это возможным для отдельного человека.

**Компьютеры стали широко использоваться неспециалистами**, что потребовало разработки "дружественного" программного обеспечения, это и положило конец кастовости программистов.

На рынке операционных систем доминировали две системы: MS-DOS и UNIX. Однопрограммная однопользовательская ОС MS-DOS широко использовалась для компьютеров, построенных на базе микропроцессоров Intel 8088, а затем 80286, 80386 и 80486. Мультипрограммная многопользовательская ОС UNIX доминировала в среде "не-интеловских" компьютеров, особенно построенных на базе высокопроизводительных RISC-процессоров.

# Эволюция ОС

В середине 80-х стали бурно развиваться **сети персональных компьютеров**, работающие под управлением сетевых или распределенных ОС.

В сетевых ОС пользователи должны быть осведомлены о наличии других компьютеров и должны делать логический вход в другой компьютер, чтобы воспользоваться его ресурсами, преимущественно файлами.

Каждая машина в сети выполняет свою **собственную** локальную операционную систему, отличающуюся от ОС автономного компьютера наличием дополнительных средств, позволяющих компьютеру работать в сети.

Сетевая ОС не имеет фундаментальных отличий от ОС однопроцессорного компьютера. Она обязательно **содержит программную поддержку для сетевых интерфейсных устройств** (драйвер сетевого адаптера), а также средства для удаленного входа в другие компьютеры сети и средства доступа к удаленным файлам, однако эти дополнения существенно не меняют структуру самой операционной системы.

# Эволюция ОС

Одним из не так давно появившихся, но бурно развивающихся направлений в мире ОС можно считать *мобильные операционные системы* - системы для смартфонов, планшетов, КПК или других мобильных устройств.

Мобильные операционные системы сочетают в себе функционал ОС для ПК с функциями для мобильных и карманных устройств (сенсорный экран, сотовая связь, Bluetooth, Wi-Fi, GPS-навигация, и т.п.).

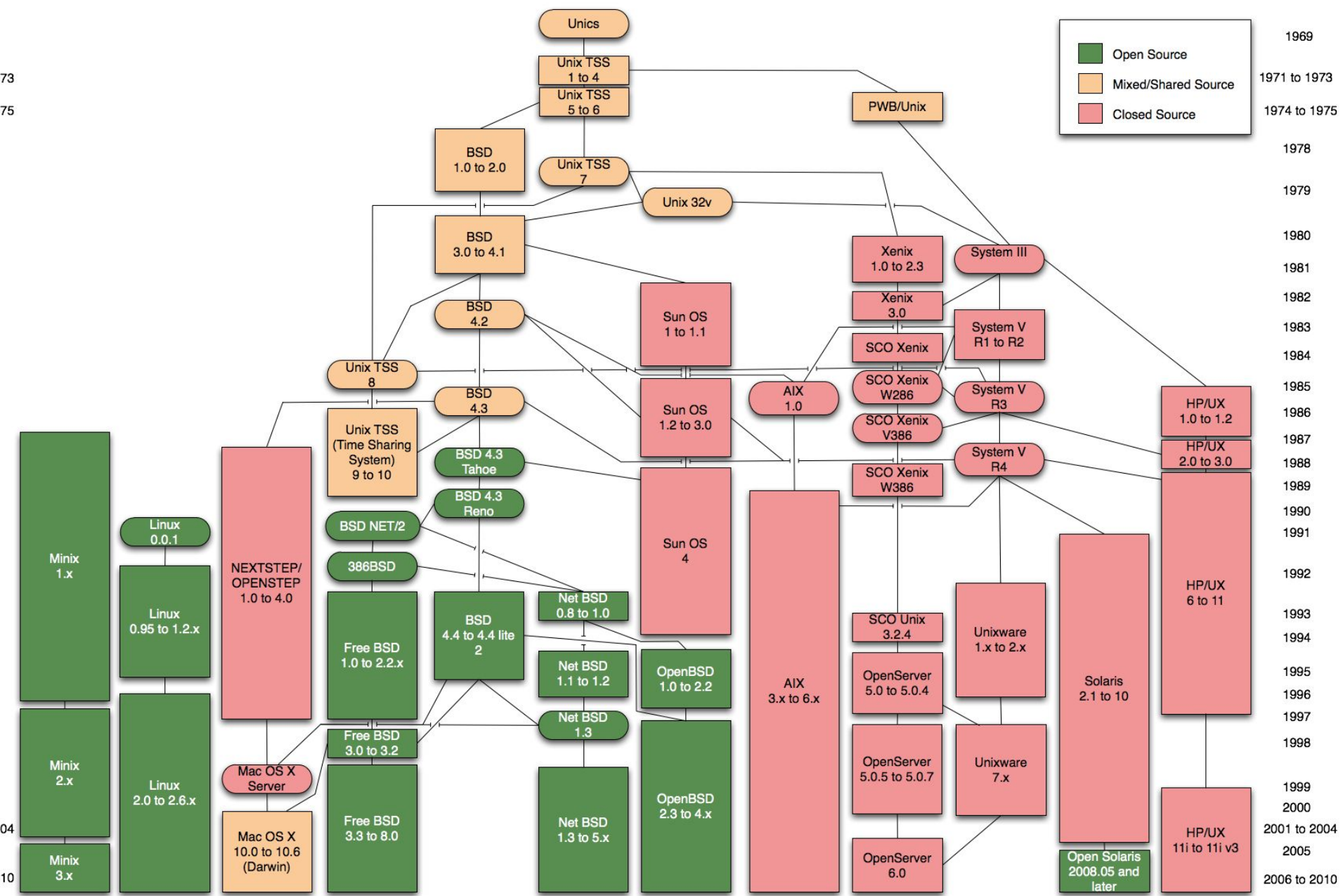
Портативные устройства мобильной связи (например, смартфоны) обычно содержат две операционные системы. Основную программную платформу взаимодействия с пользователем дополняет вторая, низкоуровневая проприетарная операционная система реального времени, обслуживающая радиооборудование.

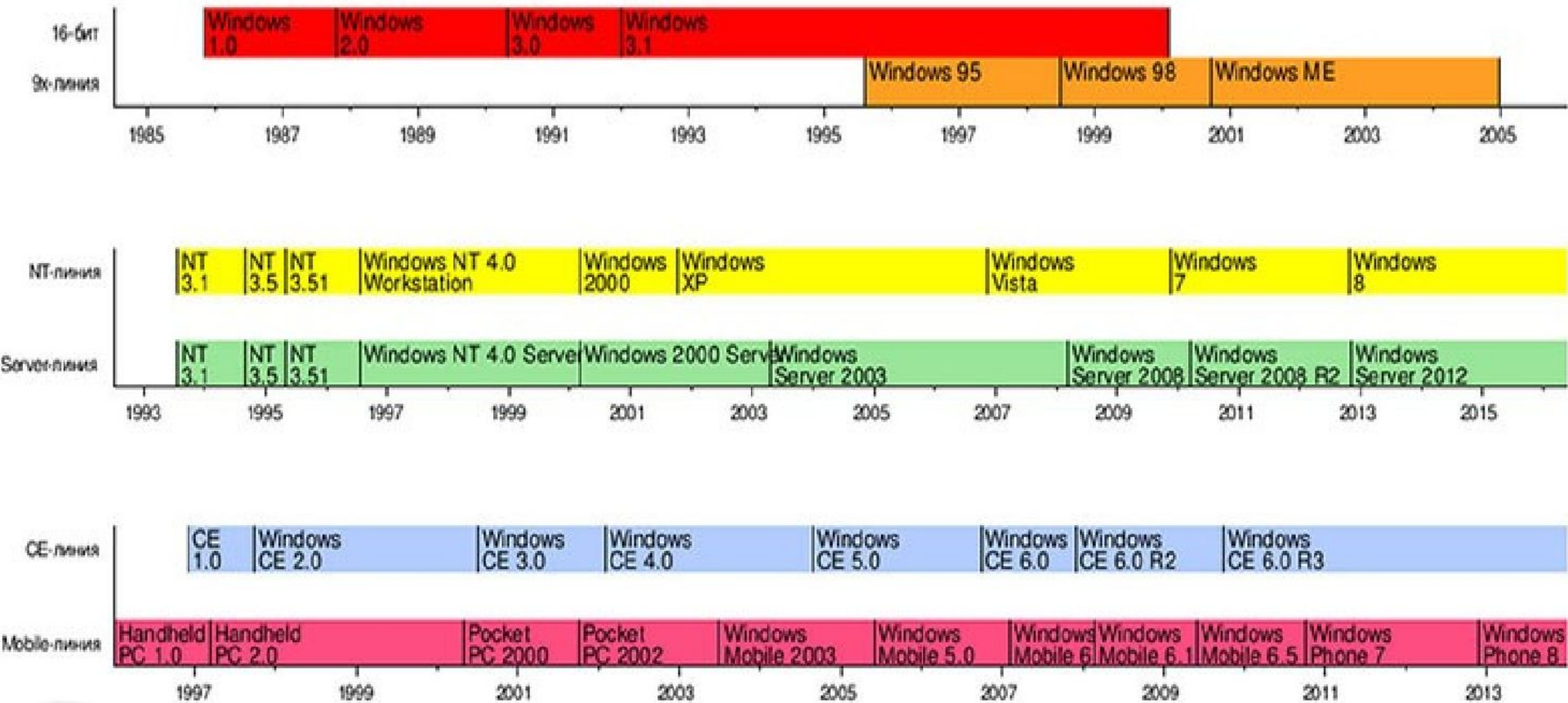
# Лекция 1: Введение в ОС

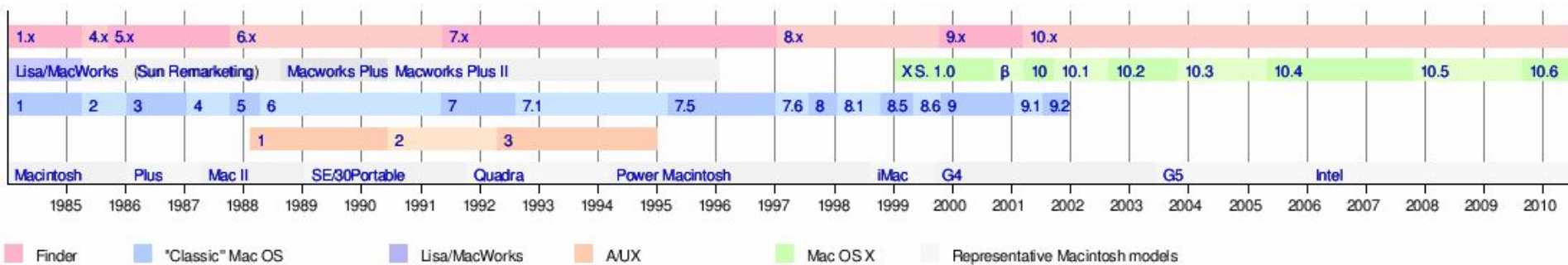
## Краткая историческая справка

1965-1969	Разработка ОС MULTICS (Bell Labs, General Electric). Кен Томпсон и Деннис Ритчи.
1969	Создание ARPANET
1971	Выпуск первой редакции UNIX - UNICS (Uniplexed Information and Computing System - Брайн Керниган).
1972	Выпуск второй редакции UNIX (Язык B на базе BCPL - автор Кен Томпсон).
1973	Добавление в язык B системы типов. Создание языка C (Деннис Ритчи)
1974	System Network Architecture (SNA)
1974	В июле Томпсон и Ритчи опубликовали в журнале Communications of the ACM историческую статью "UNIX Timesharing Operating System"
1975	Билл Гейтс и Пол Аллен создали фирму Microsoft
1976	Создание библиотеки STDIO. Установка UNIX V6 в университете Беркли (дипломник Билл Джой). Создание BSD (Berkeley Software Distribution). Добавление компилятора Pascal и редактора ex (vi).
1976	Официальная дата рождения компьютерного пиратства. В печати публикуется открытое письмо Билла Гейтса, который жалуется на незаконное использование программного обеспечения, выпускаемого Microsoft, обладателями первых микрокомпьютеров.
1977	Ричард Миллер, Джюрис Рейндфельдс (кафедра ВТ в университете Воллонгонга - Австралия) создают первую версию UNIX (версия 7) для PDP-7.









Finder "Classic" Mac OS Lisa/MacWorks AUX Mac OS X Representative Macintosh models

## Классификация ОС

Операционные системы могут различаться:

- особенностями реализации внутренних алгоритмов управления основными ресурсами (процессорами, памятью, устройствами ввода-вывода);
- типами аппаратных платформ;
- областями использования ОС;
- особенностями использованных методов проектирования.

# Особенности

## алгоритмов управления ресурсами

В зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят на:

- многозадачные и однозадачные;
- поддерживающие многопоточную (многопоточковую) обработку и не поддерживающие ее;
- многопроцессорные и однопроцессорные.
- многопользовательские и однопользовательские;

# Классификация ОС

## Особенности использованного алгоритма управления процессором

По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

- **однозадачные** (например, MS-DOS, MSX) ;
- **многозадачные** (ОС ЕС, OS/2, UNIX, Windows 95-XP);

Способ распределения процессорного времени между несколькими одновременно существующими в системе процессами (или нитями) во многом определяет специфику ОС. Среди множества существующих вариантов реализации многозадачности можно выделить две группы алгоритмов:

- **вытесняющая** многозадачность (Windows NT, OS/2, UNIX).
- **невывтесняющая** многозадачность (Windows 3.x);

### Поддержка многонитевости.

Важным свойством операционных систем является возможность распараллеливания вычислений в рамках одной задачи.

Многонитевая ОС разделяет процессорное время не между задачами, а между их отдельными ветвями (нитьями, потоками).

## Особенности использованного алгоритма управления процессором

### Многопроцессорная обработка

Важным свойством ОС является отсутствие или наличие в ней средств поддержки многопроцессорной обработки - **мультипроцессирования**.

Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами.

В наши дни становится общепринятым введение в ОС функций поддержки многопроцессорной обработки данных.

Такие функции имеются в операционных системах

- o Solaris 2.x фирмы Sun;
- o Open Server 3.x компании Santa Crus Operations;
- o OS/2 фирмы IBM;
- o Windows NT фирмы Microsoft;
- o NetWare 4.1 фирмы Novell.

**Многопроцессорные ОС** могут классифицироваться по способу организации вычислительного процесса в системе с многопроцессорной архитектурой: асимметричные ОС и симметричные ОС.

## Особенности использованного алгоритма управления процессором

### Поддержка многопользовательского режима

По числу одновременно работающих пользователей ОС делятся на:

- **однопользовательские** (MS-DOS, Windows 3.x, ранние версии OS/2);
- **многопользовательские** (UNIX, Windows NT).

Главным **отличием многопользовательских систем от однопользовательских** является **наличие средств защиты информации** каждого пользователя от несанкционированного доступа других пользователей.

Следует заметить, что не всякая **многозадачная** система является **многопользовательской**, и не всякая **однопользовательская** ОС является **однозадачной**.

Были рассмотрены признаки классификации ОС, связанные с управлением только одним типом ресурсов - процессором.

Важное влияние на облик операционной системы в целом, на возможности ее использования в той или иной области оказывают особенности и других подсистем управления локальными ресурсами - подсистем управления памятью, файлами, устройствами ввода-вывода.



## Особенности аппаратных платформ

По типу аппаратуры различают операционные системы

- персональных компьютеров;
- мини-компьютеров;
- мейнфреймов(OS/390 <- OS/360);
- кластеров;
- серверов и сетей ЭВМ.

Среди перечисленных типов компьютеров могут встречаться как однопроцессорные варианты, так и многопроцессорные. В любом случае специфика аппаратных средств, как правило, отражается на специфике операционных систем.

## Особенности областей использования

Многозадачные ОС подразделяются на три типа в соответствии с использованными при их разработке критериями эффективности:

- **системы пакетной обработки (например, ОС ЕС);**
- **системы разделения времени (UNIX, VMS);**
- **системы реального времени (QNX, RT/11).**

### Системы пакетной обработки

*Системы пакетной обработки* предназначались **для решения задач в основном вычислительного характера**, не требующих быстрого получения результатов. Главной целью и **критерием эффективности систем пакетной обработки является максимальная пропускная способность**, то есть решение максимального числа задач в единицу времени.

### Системы разделения времени

*Системы разделения времени* призваны исправить основной недостаток систем пакетной обработки - изоляцию пользователя-программиста от процесса выполнения его задач. Каждому пользователю системы разделения времени предоставляется терминал, с которого он может вести диалог со своей программой. **Критерием эффективности систем разделения времени является не максимальная пропускная способность, а удобство и эффективность работы пользователя.**

# Особенности областей использования

## Системы реального времени

Системы реального времени применяются для управления различными техническими объектами, такими, например, как станок, спутник, научная экспериментальная установка или технологическими процессами.

Во всех этих случаях **существует предельно допустимое время, в течение которого должна быть выполнена та или иная программа**, управляющая объектом, в противном случае может произойти авария.

Таким образом, **критерием эффективности для систем реального времени является их способность выдерживать заранее заданные интервалы времени между запуском программы и получением результата (управляющего воздействия)**. Это время называется **временем реакции системы**, а соответствующее свойство системы - **реактивностью**.

Для этих систем мультипрограммная смесь представляет собой фиксированный набор заранее разработанных программ, а выбор программы на выполнение осуществляется исходя из текущего состояния объекта или в соответствии с расписанием плановых работ.

Некоторые операционные системы могут совмещать в себе свойства систем разных типов, например, часть задач может выполняться в режиме **пакетной обработки**, а часть - в режиме **реального времени** или в **режиме разделения времени**.

## Особенности методов построения

Способы построения ядра системы - **монолитное** ядро или **микроядерный** подход;

**Объектно-ориентированного подхода** дает возможность использовать все его достоинства:

- аккумуляцию удачных решений в форме стандартных объектов,
- возможность создания новых объектов на базе имеющихся с помощью механизма наследования;
- защиту данных за счет их инкапсуляции во внутренние структуры объекта;
- структурированность системы, состоящей из набора хорошо определенных объектов.

Наличие нескольких **прикладных сред** дает возможность в рамках одной ОС одновременно выполнять приложения, разработанные для других ОС .

**Распределенная организация** ОС позволяет упростить работу пользователей и программистов в сетевых средах.

Характерными признаками распределенной организации ОС являются:

- наличие единой справочной службы разделяемых ресурсов;
- единой службы времени;
- использование механизма вызова удаленных процедур (RPC) для прозрачного распределения программных процедур по машинам;
- многонитевой обработки, позволяющей распараллеливать вычисления в рамках одной задачи и выполнять эту задачу сразу на нескольких компьютерах сети

# Архитектура и Архитектура и функции ОС современной ПЭВМ

Функции операционной системы автономного компьютера обычно группируются либо в соответствии с типами локальных ресурсов, которыми управляет ОС, либо в соответствии со специфическими задачами, применимыми ко всем ресурсам. Иногда такие группы функций называют подсистемами. Наиболее важными подсистемами ОС являются:

- подсистема управления процессами;
- подсистема управления памятью;
- подсистема управления файлами и внешними устройствами.

**Подсистемы общие для всех ресурсов:**

**подсистема защиты данных;**

**подсистема администрирования.**

**подсистема интерфейса прикладного программирования;**

**подсистема пользовательского интерфейса;**

## Подсистема управления процессами

Важнейшей частью операционной системы, непосредственно влияющей на функционирование вычислительной машины, является подсистема управления процессами.

Для **каждого** вновь создаваемого процесса **ОС генерирует системные информационные структуры**, которые содержат данные о потребностях процесса в ресурсах вычислительной системы, а также о фактически выделенных ему ресурсах. Таким образом, **процесс можно** также **определить как** некоторую **заявку на потребление системных ресурсов**.

Чтобы процесс мог быть выполнен, операционная система должна назначить ему область оперативной памяти, в которой будут размещены коды и данные процесса, а также предоставить ему необходимое количество процессорного времени. Кроме того, процессу может понадобиться доступ к таким ресурсам, как файлы и устройства ввода-вывода.

В **информационные структуры процесса часто включаются** вспомогательные данные, характеризующие **историю пребывания процесса в системе** (например, какую долю времени процесс потратил на операции ввода-вывода, а какую на вычисления), его **текущее состояние** (активное или заблокированное), **степень привилегированности** процесса (значение приоритета). Данные такого рода могут учитываться операционной системой при принятии решения о предоставлении ресурсов процессу.

## Подсистема управления процессами

В **мультипрограммной операционной системе** одновременно может существовать несколько процессов. Часть процессов порождается по инициативе пользователей и их приложений, такие процессы обычно называют **пользовательскими**. Другие процессы, называемые **системными**, инициализируются самой операционной системой для выполнения своих функций.

На протяжении периода существования процесса его выполнение может быть многократно прервано и продолжено. Для того, чтобы возобновить выполнение процесса, необходимо восстановить состояние его операционной среды. **Состояние операционной среды** идентифицируется:

- **состоянием регистров и программного счетчика;**
- **режимом работы процессора;**
- **указателями на открытые файлы;**
- **информацией о незавершенных операциях ввода-вывода**
- кодами ошибок выполняемых процессом системных вызовов и т. д.

Эта информация называется **контекстом процесса**.

Говорят, что при смене процесса происходит переключение контекстов.

## Подсистема управления процессами

В операционной системе **нет однозначного соответствия между процессами и программами**. Один и тот же программный файл может породить несколько параллельно выполняемых процессов, а процесс может в ходе своего выполнения сменить программный файл и начать выполнять другую программу. Для реализации сложных программных комплексов полезно бывает организовать их работу в виде нескольких параллельных процессов, которые периодически взаимодействуют друг с другом и обмениваются некоторыми данными. Так как операционная система защищает ресурсы процессов и не позволяет одному процессу писать или читать из памяти другого процесса, то для оперативного взаимодействия процессов ОС должна предоставлять особые средства, которые называют средствами **межпроцессного взаимодействия**.

Таким образом, подсистема управления процессами:

1. **Планирует выполнение процессов**, то есть распределяет процессорное время между несколькими одновременно существующими в системе процессами.
2. **Занимается созданием и уничтожением процессов**.
3. **Обеспечивает процессы необходимыми системными ресурсами**.
4. **Поддерживает синхронизацию процессов**.
5. **Обеспечивает взаимодействие между процессами**.



## Подсистема управления памятью

Управление памятью включает:

- **распределение имеющейся физической памяти между всеми существующими в системе в данный момент процессами;**
- **загрузку кодов и данных процессов в отведенные им области памяти;**
- **настройку адресно-зависимых частей кодов процесса на физические адреса выделенной области;**
- **защиту областей памяти каждого процесса.**

Существует большое разнообразие алгоритмов распределения памяти. Они могут отличаться, например, количеством выделяемых процессу областей памяти (в одних случаях память выделяется процессу в виде одной непрерывной области, а в других — в виде нескольких несмежных областей), степенью свободы границы областей. В некоторых системах распределение памяти выполняется страницами фиксированного размера, а в других — сегментами переменной длины.

Одним из наиболее популярных способов управления памятью в современных операционных системах является так называемая **виртуальная память**. Наличие в ОС механизма виртуальной памяти позволяет программисту писать программу так, как будто в его распоряжении имеется однородная оперативная память большого объема, часто существенно превышающего объем имеющейся физической памяти. На самом деле все данные, используемые программой, хранятся на диске и при необходимости частями (сегментами или страницами) отображаются в физическую память.

## Подсистема управления памятью

Защита памяти — это избирательная способность предохранять выполняемую задачу от записи или чтения памяти, назначенной другой задаче. Правильно написанные программы не пытаются обращаться к памяти, назначенной другим. Однако реальные программы часто содержат ошибки, в результате которых такие попытки иногда предпринимаются. **Средства защиты памяти, реализованные в операционной системе, должны пресекать несанкционированный доступ процессов к чужим областям памяти.**

Таким образом, функциями ОС по **управлению памятью** являются:

- отслеживание свободной и занятой памяти;
- выделение памяти процессам и освобождение памяти при завершении процессов;
- защита памяти;
- вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти недостаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место,
- настройка адресов программы на конкретную область физической памяти.

## Подсистема управления

### файлами и внешними устройствами

Способность ОС к «экранированию» сложностей реальной аппаратуры очень ярко проявляется в одной из основных подсистем ОС — **файловой системе**. Операционная система **виртуализирует** отдельный набор данных, хранящихся на внешнем накопителе, в виде **файла** — простой неструктурированной последовательности байтов, имеющей символическое имя.

**Файловая система** ОС выполняет преобразование символических имен файлов, с которыми работает пользователь или прикладной программист, в физические адреса данных на диске, организует совместный доступ к файлам, защищает их от несанкционированного доступа.

Подсистема **управления внешними устройствами**, называемая также **подсистемой ввода-вывода**, исполняет роль интерфейса ко всем устройствам, подключенным к компьютеру.

Программа, управляющая конкретной моделью внешнего устройства и учитывающая все его особенности, обычно называется **драйвером** этого устройства.

Поддержание **высокоуровневого унифицированного интерфейса прикладного программирования** к разнородным устройствам ввода-вывода является одной из наиболее важных задач ОС.

## Защита данных и администрирование

Безопасность данных вычислительной системы обеспечивается средствами **отказоустойчивости** ОС, направленными на **защиту от сбоев и отказов аппаратуры** и **ошибок программного обеспечения**, а также средствами **защиты от несанкционированного доступа**. В последнем случае ОС защищает данные от ошибочного или злонамеренного поведения пользователей системы

Функции защиты ОС тесно связаны с функциями **администрирования**, так как именно администратор определяет права пользователей при их обращении к разным ресурсам системы — файлам, каталогам, принтерам, сканерам и т. п. Кроме того, администратор ограничивает возможности пользователей в выполнении тех или иных системных действий.

Важным средством защиты данных являются **функции аудита** ОС, заключающиеся в фиксации всех событий, от которых зависит безопасность системы. Например, попытки удачного и неудачного логического входа в систему, операции доступа к некоторым каталогам и файлам, использование принтеров и т. п. Список событий, которые необходимо отслеживать, зачастую определяет администратор ОС.

## Защита данных и администрирование

Поддержка **отказоустойчивости** реализуется операционной системой, как правило, **на основе резервирования**. Чаще всего в функции ОС входит поддержание нескольких копий данных на разных дисках или разных дисковых накопителях. Резервируются также и другие устройства ввода-вывода.

При отказе одного из избыточных устройств операционная система должна быстро и прозрачным для пользователя образом произвести реконфигурацию системы и продолжить работу с резервным устройством.

Поддержка отказоустойчивости также входит в обязанности системного администратора. В состав ОС обычно входят утилиты, позволяющие администратору выполнять регулярные операции резервного копирования для обеспечения быстрого восстановления важных данных.

## Интерфейс прикладного программирования

Прикладные программисты используют в своих приложениях обращения к ОС, когда для выполнения тех или иных действий им требуется особый статус, которым обладает только операционная система.

Возможности операционной системы доступны прикладному программисту в виде набора функций, называющегося **интерфейсом прикладного программирования (Application Programming Interface, API)**.

Для разработчиков приложений все особенности конкретной операционной системы представлены особенностями ее API. Поэтому операционные системы с различной внутренней организацией, но с одинаковым набором функций API кажутся им одной и той же ОС, что упрощает стандартизацию операционных систем и обеспечивает переносимость приложений между внутренне различными ОС, соответствующими определенному стандарту на API.

Например, следование общим стандартам API UNIX, одним из которых является **стандарт Posix**, позволяет говорить о некоторой обобщенной операционной системе UNIX, хотя многочисленные версии этой ОС от разных производителей иногда существенно отличаются внутренней организацией.

## Интерфейс прикладного программирования

Приложения выполняют обращения к функциям API с помощью **системных вызовов**. Способ, которым приложение получает услуга операционной системы, очень похож на вызов подпрограмм. Информация, нужная ОС и состоящая обычно из идентификатора команды и данных, помещается в определенное место памяти, в регистры и/или стек. Затем управление передается операционной системе, которая выполняет требуемую функцию и возвращает результаты через память, регистры или стеки. Если операция проведена неуспешно, то результат включает индикацию ошибки.

Способ реализации системных вызовов зависит от структурной организации ОС, которая, в свою очередь, тесно связана с особенностями аппаратной платформы. Кроме того, он зависит от языка программирования. При использовании ассемблера программист устанавливает значения регистров и/или областей памяти, а затем выполняет специальную инструкцию вызова сервиса или программного прерывания для обращения к некоторой функции ОС. При использовании языков высокого уровня функции ОС вызываются тем же способом, что и написанные пользователем подпрограммы, требуя задания определенных аргументов в определенном порядке.

## Пользовательский интерфейс

Операционная система должна обеспечивать удобный интерфейс не только для прикладных программ, но и для человека, работающего за терминалом. Этот человек может быть конечным пользователем, администратором ОС или программистом.

Современные ОС поддерживают развитые функции пользовательского интерфейса для интерактивной работы за терминалами двух типов: **алфавитно-цифровыми и графическими.**

Программный модуль ОС, ответственный за чтение отдельных команд или же последовательности команд из командного файла, иногда называют **командным интерпретатором.**

Ввод команды может быть упрощен, если операционная система поддерживает **графический пользовательский интерфейс.** В этом случае пользователь для выполнения нужного действия с помощью мыши выбирает на экране нужный пункт меню или графический символ.