

Курс операционные системы

Тема 1.

Основные понятия

Основные темы курса

- 1. Программно-аппаратный комплекс и место операционной системы в нем.**
- 2. Интерфейсы операционных систем.**
- 3. Процессы и потоки. Управление, планирование и синхронизация.**
- 4. Управление памятью. Методы, алгоритмы и средства.**
- 5. Подсистема ввода-вывода. Файловые системы.**
- 6. Диагностика и мониторинг устройств компьютера.**
- 7. Администрирование современных операционных систем.**

Основная литература

Назаров С.В., Широков А.И. Многопользовательские операционные системы. – М.: Изд. дом МИСиС, 2010. 194 стр. (в разделе Монографии электр. библиотеки)

Широков А.И., Калашникова О.Н., Крапухина Н.В. и др. Многопользовательские операционные системы. Лаб. практикум. 2-е изд. Изд. дом МИСиС,, 2010, 132 стр. (№ 592)

Назаров С.В., Широков А.И. Технологии многопользовательских операционных систем. – М.: Изд. дом МИСиС, 2012. 296 стр. (в разделе Монографии электр. библиотеки)

Дополнительная литература

1. Таненбаум Э. Современные операционные системы. Изд-е 4. СПб.: Питер, 2010.
2. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2005.
3. Назаров С.В., Гудыно Л.П., Кириченко А.А. Операционные системы. Практикум. Учеб. Пособие. – М.: КУДИЦ-ПРЕСС, 2008.

4. Гордеева А.В, Мочалова А.Ю.
Системное программное обеспечение.
СПб.: Питер, 2003, - 736 с.: ил.

5. Бурк Робин, Хорват Б. Девид и др.
UNIX для системных администраторов.
Энциклопедия пользователя. К.:
Издательство «Диа-Софт». 1998-864 с..

6. Робачевский А., Немнюгин С., Стесин
О. Операционная система UNIX. 2-е изд.,
СПб: БХВ-Петербург, 2010, 656 с.

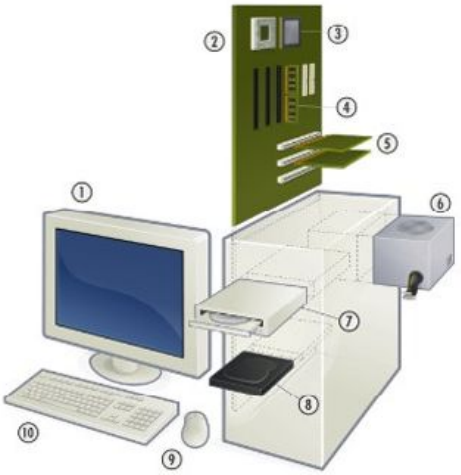
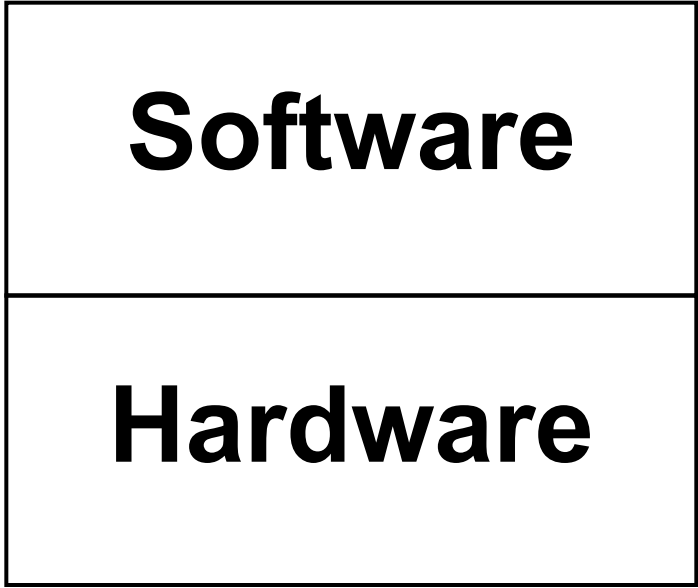
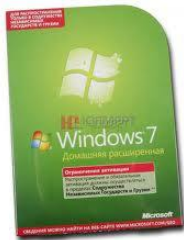
7. Таккет. Дж. Использование Linux. (Специальное издание). К.; М.; СПб: Издательский дом “Вильямс”, 1998. – 576 с.: ил.

8. Карпов В.Е., Коньков К.А.. Основы операционных систем. (Под редакцией В.П. Иванникова) /Курс лекций. Учебное пособие/ Интернет-университет Информационных технологий. Москва, 2004, 632 с. Адрес в Интернет для курса:
<http://www.intuit.ru/department/os/baseoperatesys/>

Структура компьютера (программно-аппаратного комплекса)

Компьютер - это устройство для переработки информации.

Он делится, прежде всего, на две части: аппаратуру и программное обеспечение.



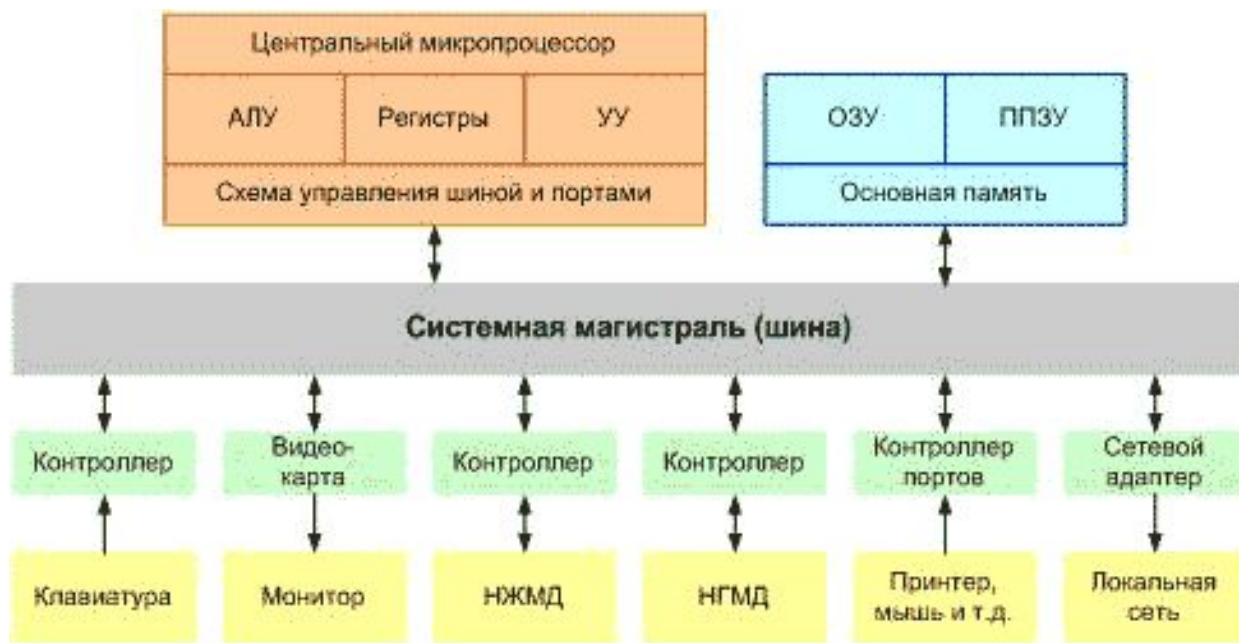
http://romaxus.ru/parser_pictures/resource_images/321166.jpg

http://edengard.com/uploads/posts/2012-02/1328674433_tvtfk3amwzkwgij.jpeg

<http://im7-tub-ru.yandex.net/i?id=478288413-37-72>

<http://im0-tub-ru.yandex.net/i?id=371100191-08-72>

Аппаратура персонального компьютера представляет собой отдельные устройства объединенные на основе шинной структуры.



Со страницы «Программное обеспечение» Википедии

В компьютерном сленге часто используется слово **софт** от английского слова **software**, которое в этом смысле впервые применил в статье в American Mathematical Monthly математик из Принстонского университета Джон Тьюки (англ. John W. Tukey) в 1958 году.

Software также можно делить на две части:
прикладные программы пользователей и
системное программное обеспечение.

Software

**1. Прикладные
программы**

**2. Системное ПО
(операционная система)**

В составе второй выделяют, прежде
всего, операционную систему (ОС).

В Википедии статья, посвященная программному обеспечению, выделяет

а) три типа по назначению:

1. *системное;*
2. *прикладное;*
3. *инструментальное.*

б) Три типа по способу распространения и использования:

1. *несвободное/закрытое;*
2. *открытое;*
3. *свободное.*

Можно предложить классификацию программных продуктов в которой выделяются такие четыре их типа :

1. Системное.
2. Прикладное.
3. Средства разработки.
4. Утилиты.

Иногда выделяется пять составляющих системного программного обеспечения:

1. Операционные системы.
2. Системы управления файлами.
3. Интерфейсные оболочки для взаимодействия пользователя с ОС и программные среды.
4. Системы программирования.
5. Утилиты.

см. отличия в «Википедии»

По мере развития функциональности иногда выделяют четыре уровня программного обеспечения :

1. базовый уровень;
2. системный уровень;
3. служебный уровень;
4. прикладной уровень.

<http://rudocs.exdat.com/docs/index-51800.html> и

Информатика для юристов и экономистов/ под ред. Симоновича В.А.

http://bibliostreet.ru/i/trytr/informatika_dlja_juristov_i_ekonomistov_-_simonovich1.html

Измененная схема классификации Программного Обеспечения по уровню функциональности

4. Прикладной уровень.
3. Служебный уровень.
2. Системный уровень.
1. Базовый уровень.

=====

0. Аппаратура

Расширим представление о структуре программно-аппаратного комплекса

Электронная таблица	Электронная почта
Игры	Текстовый процессор

ПРИЛОЖЕНИЯ



Управление приложениями
Управление аппаратурой

ОПЕРАЦИОННАЯ СИСТЕМА



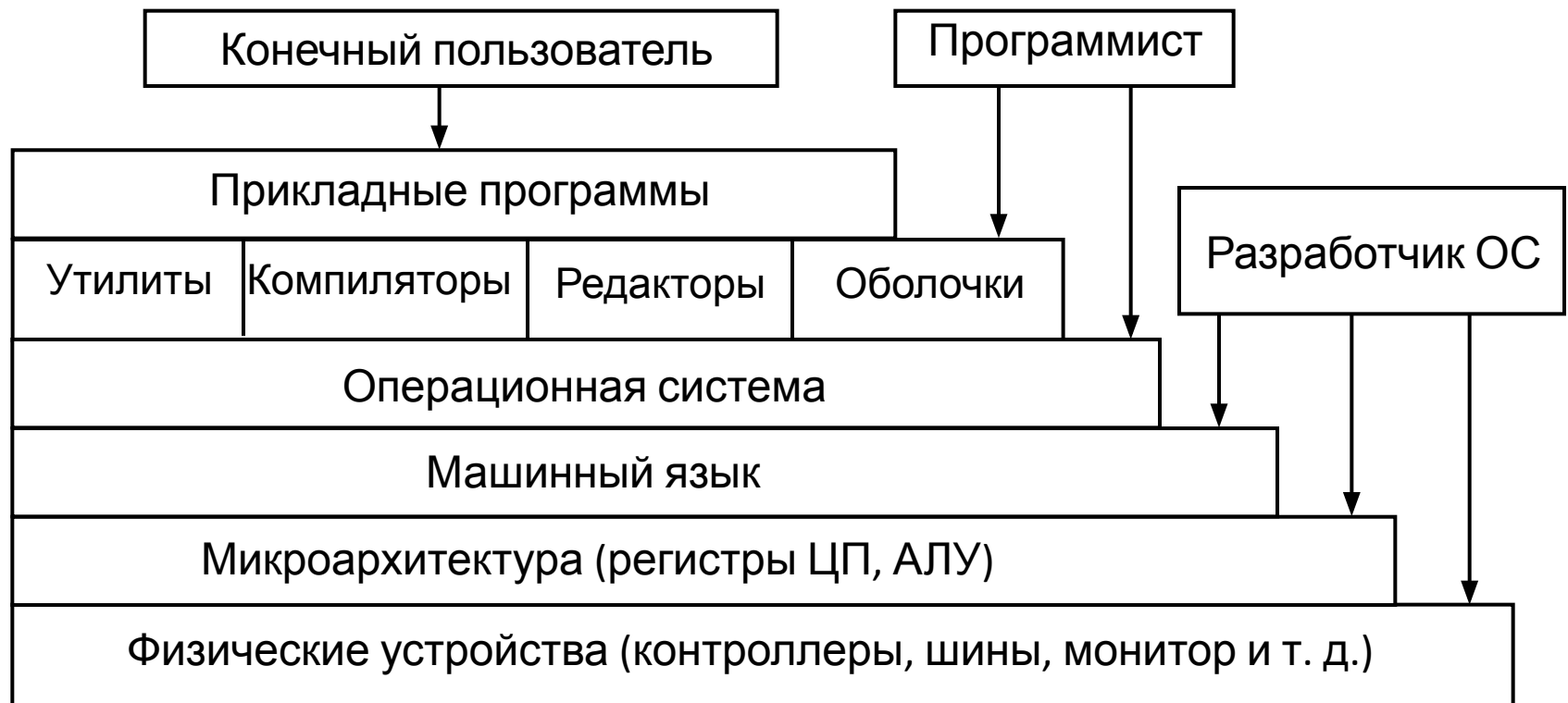
Центральный процессор (CPU)	Диск	Монитор
CD ROM	Принтер	Клавиатура

АППАРАТУРА

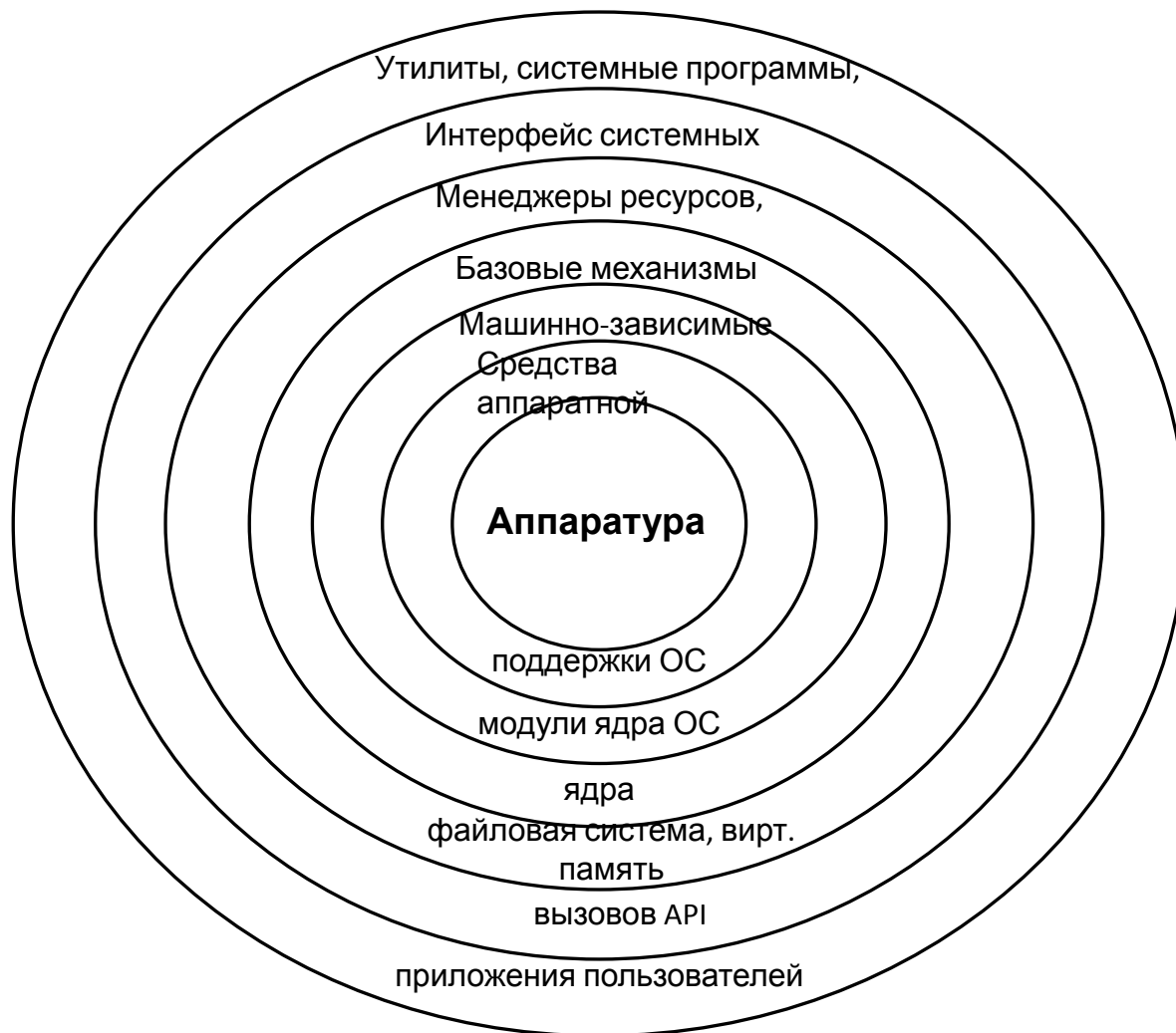
Расширим представление об уровнях компьютерной системы выделением «слоев» в его аппаратной части

Банковская система	Заказ билетов	Веб-браузер	Программы-приложения
Компиляторы	Редакторы	Интерпретаторы команд	
Операционная система			Системные программы
Машинный язык			Оборудование, аппаратура
Микроархитектура			
Физические устройства			

Расширим представление о многослойной структуре компьютера, добавив Пользователей



Еще одно представление о многослойной структуре компьютера



Опишем уровни подробнее

0. Этот уровень составляет *АППАРАТУРА*.

1. *Средства аппаратной поддержки ОС.*

Значительная часть функций ОС может выполняться аппаратными средствами. Чисто программных ОС сейчас нет. Как правило, в современных системах всегда есть средства аппаратной поддержки ОС, которые прямо участвуют в организации вычислительных процессов. К ним относятся: система прерываний, средства поддержки привилегированного режима, средства поддержки виртуальной памяти, системный таймер, средства переключения контекстов процессов (информация о состоянии процесса в момент его приостановки), средства защиты памяти и др.

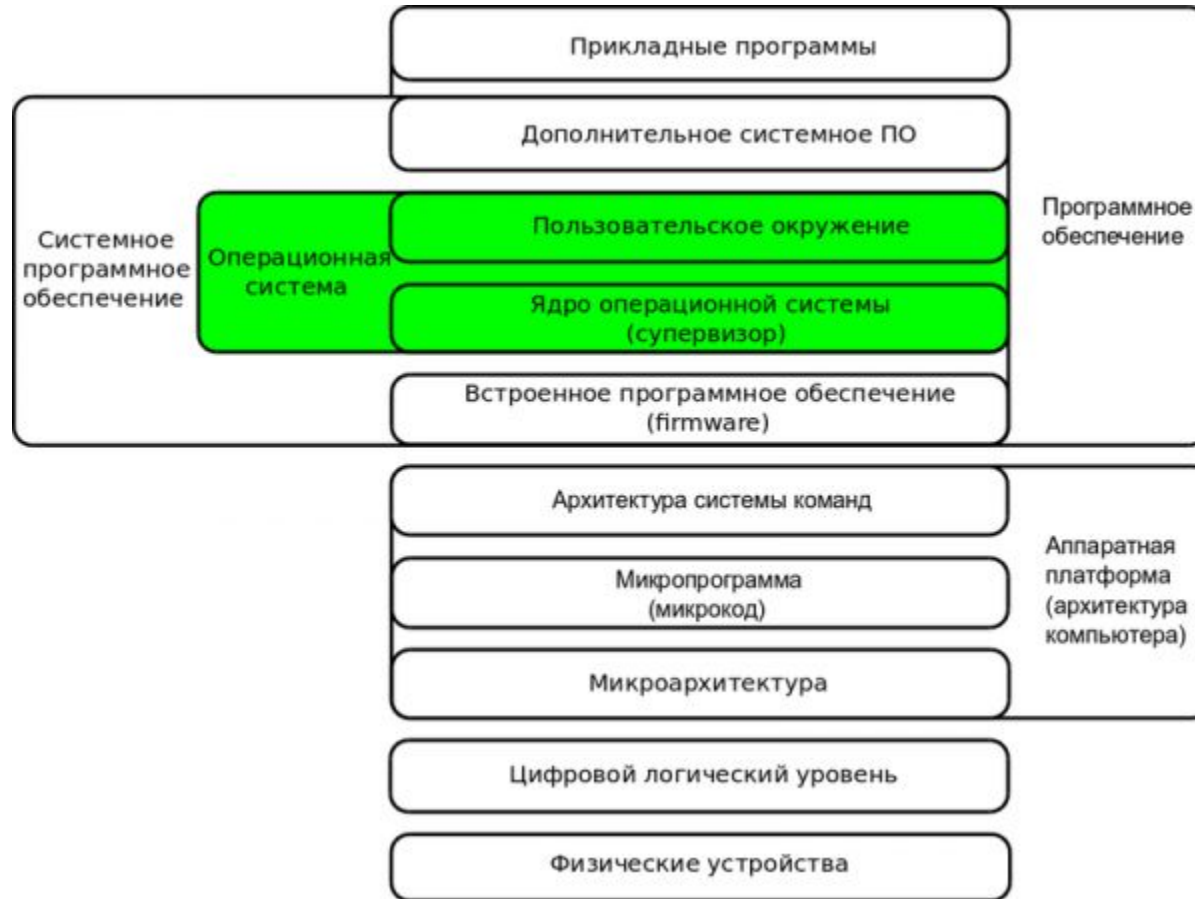
2. *Машинно-зависимые модули ОС.* Этот слой образует модули, в которых отражается специфика аппаратной платформы компьютера. Назначение этого слоя – «экранирование» вышележащих слоев ОС от особенностей аппаратуры, например, HAL (Hardware Abstraction Layer), уровень аппаратных абстракций).

3. *Базовые механизмы ядра.* Этот слой модулей выполняет наиболее примитивные операции ядра: программное переключение контекстов процессов, диспетчерскую прерываний, перемещение страниц между основной памятью и диском и т.п. Модули этого слоя не принимают решений о распределении ресурсов, а только обрабатывают решения, принятые модулями вышележащих уровней. Поэтому их часто называют исполнительными механизмами для модулей верхних слоев ОС.

4. Менеджеры ресурсов. Модули этого слоя выполняют стратегические задачи по управлению ресурсами вычислительной системы. Это менеджеры (диспетчеры) процессов ввода-вывода, оперативной памяти и файловой системы. Каждый менеджер ведет учет свободных и используемых ресурсов и планирует их распределение в соответствии запросами приложений.

5. Интерфейс системных вызовов. Это верхний слой ядра ОС, взаимодействующий с приложениями и системными утилитами, он образует прикладной программный интерфейс ОС. Функции API, обслуживающие системные вызовы, предоставляют доступ к ресурсам системы в удобной компактной форме, без указания деталей их физического расположения.

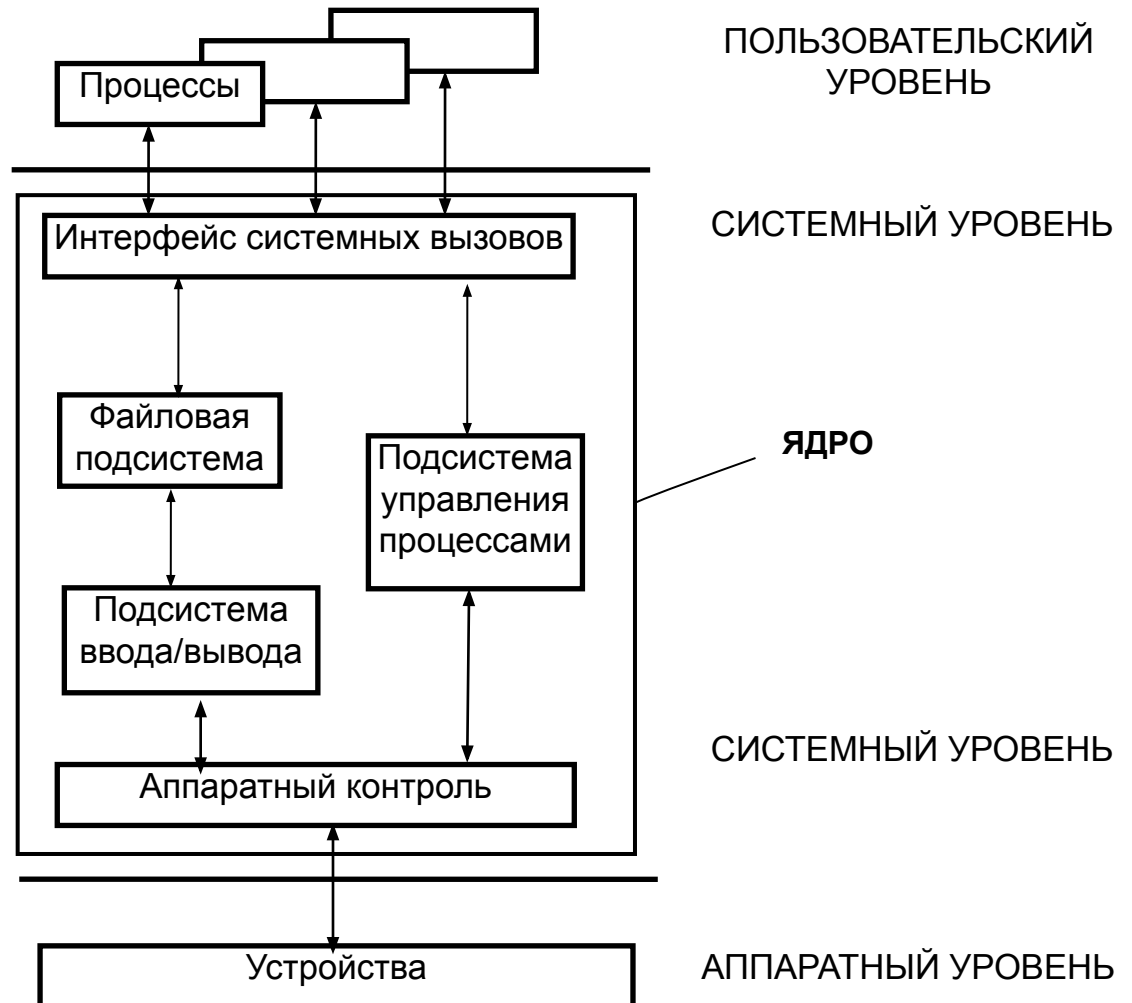
Еще одна схема, иллюстрирующая место **операционной системы** в многоуровневой структуре **компьютера**



Следующая схема программно-аппаратного комплекса выделяет отдельные подсистемы ядра ОС UNIX

Осуществляют управление:

- процессами;
- файлами;
- ВВОДОМ-ВЫВОДОМ.



К структуре программных средств

3	Приложения (процессы) пользователей или ПОЛЬЗОВАТЕЛЬСКИЕ ПРОЦЕССЫ
2	ИНТЕРФЕЙС СИСТЕМНЫХ ВЫЗОВОВ
1	ЯДРО
0	Аппаратное обеспечение или АППАРАТУРА

К структуре программных средств 2

4	Приложения (процессы) пользователей или ПОЛЬЗОВАТЕЛЬСКИЕ ПРОЦЕССЫ
3	БИБЛИОТЕЧНЫЕ ФУНКЦИИ
2	ИНТЕРФЕЙС СИСТЕМНЫХ ВЫЗОВОВ
1	ЯДРО
0	Аппаратное обеспечение или АППАРАТУРА

К структуре программных средств 3

4	Приложения (процессы) пользователей		
3	Библиотеки		Стандартные приложения
2	динамические	статические	
1	Системные вызовы		
0	Аппаратное обеспечение		

Библиотечные функции

Библиотека представляет собой набор подпрограмм, который позволяет коду этих подпрограмм использоваться в виде отдельных модулей. Исполняемые файлы и библиотеки создают взаимные ссылки в процессе т.н. "линковки" или связывания, который осуществляется редактором связей (linker). В первом приближении, библиотеки можно разделить на статические и динамические. Статические библиотеки являются набором объектных файлов и традиционно имеют расширение **".a"** в UNIX-подобных ОС и **".lib"** в Windows. Когда программа связывается со статической библиотекой, машинный код из объектных файлов для каждой используемой программой функции библиотеки копируется из библиотеки в конечный исполняемый файл.

Динамические библиотечные функции

DLL ([англ. *dynamic-link library*](#) — «библиотека динамической компоновки», «динамически подключаемая библиотека») — понятие операционных систем [Microsoft Windows](#) и [IBM OS/2](#), динамическая [библиотека](#), позволяющая многократное использование различными программными приложениями. К *DLL* относятся также элементы управления [ActiveX](#) и [драйверы](#). В мире [UNIX](#) аналогичные функции выполняют так называемые [общие объекты](#) аналогичные функции выполняют так называемые общие объекты ([англ. *shared objects*](#)).

Формат файлов *DLL* придерживается тех же соглашений, что и [формат исполняемых файлов](#), сочетая код, таблицы и ресурсы, отличаясь лишь интерпретацией некоторых полей.

Википедия - Динамически подключаемая библиотека

Динамические библиотеки хранятся обычно в определенном месте и имеют стандартное [расширение](#). Например, файлы `.library` в логическом томе *Libs*: в [AmigaOS](#) в AmigaOS; в [Microsoft Windows](#) в AmigaOS; в Microsoft Windows и [OS/2](#) файлы библиотек общего пользования имеют расширение [.dll](#); в [UNIX](#)-подобных ОС — обычно [.so](#); в [Mac OS](#) — [.dylib](#).

Википедия - Библиотека (программирование)

Библиотечные файлы

Компоновкой (linking) называется процесс подключения внешних функций к приложению. Существуют два типа компоновки:

статическая и *динамическая*.

Статическая компоновка выполняется на стадии компиляции, а динамическая -- во время работы программы.

При динамической компоновке программные модули, в которых находятся функции, также предварительно компилируются в объектные (.OBJ) файлы. Вместо группировки в библиотечных файлах они объединяются в исполняемых файлах Windows особого типа, называемых *библиотеками динамической компоновки (DLL)*.

Создавая библиотеку DLL, программист указывает, какие из входящих в нее функций должны быть доступны для других работающих приложений.

Этапы работы программно-аппаратного комплекса

ЭТАП I. Управление аппаратурой (часть 1).

После включения компьютера выполняются действия, предшествующих загрузке операционной системы,- инициализации оборудования и запуск стартовых системных программ. После этого управление получает операционная система.

ЭТАП II. Управление процессами (часть 1).

Операционная система готова выполнить прикладные задания. В системе они представлены процессами.

ЭТАП III. Управление процессами (часть 2).

Каждый порожденный в системе процесс, в определенные моменты времени, может потребовать доступ к системным ресурсам.

Этапы работы 2

ЭТАП IV. Управление аппаратурой (часть 2).

На этом этапе операционная система дает возможность прикладным программам работать с разнообразным оборудованием, не отвлекаясь на детали.

ЭТАП V. Управление аппаратурой (часть 3).

После выполнения команды выключения компьютера операционная система должна завершить все выполняющиеся процессы как прикладные, так и системные.

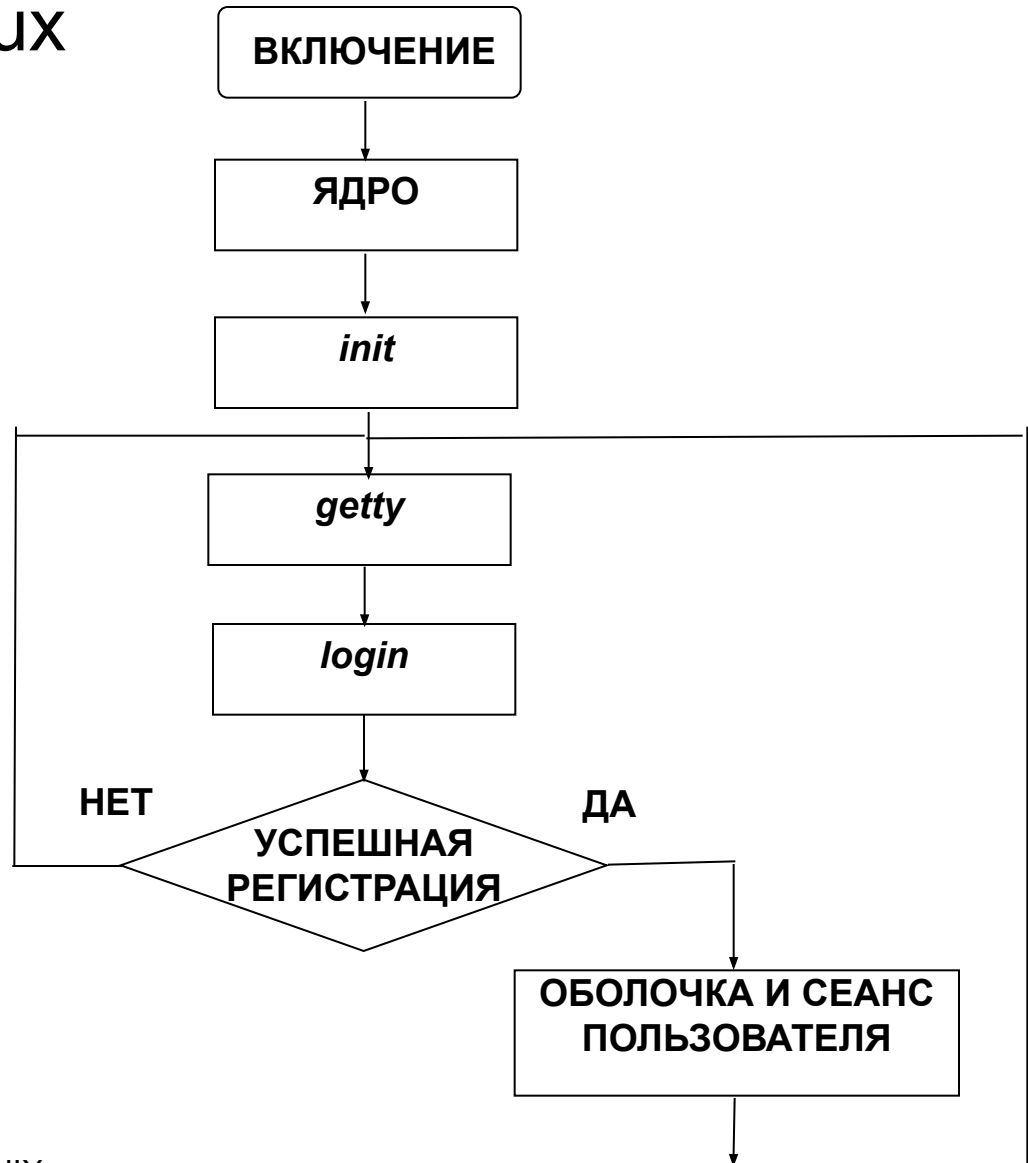
Бурк Робин, Хорват Б. Девид и др. UNIX для системных администраторов. Энциклопедия пользователя.

ЗАМЕЧАНИЕ.

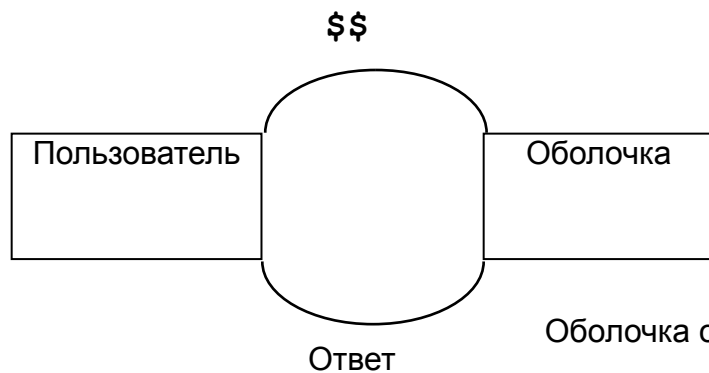
Современное оборудование вносит коррективы в эту схему.

Например, устройства подключаемые к USB, добавляются и удаляются из системы после её загрузки и до выключения.

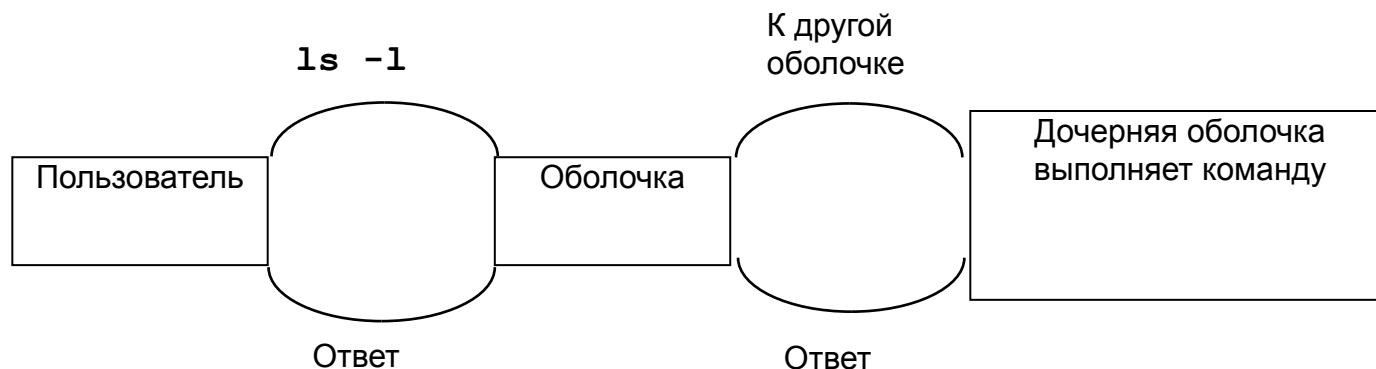
Порядок исполнения системных процессов при загрузке UNIX/Linux



Схемы выполнения команд UNIX

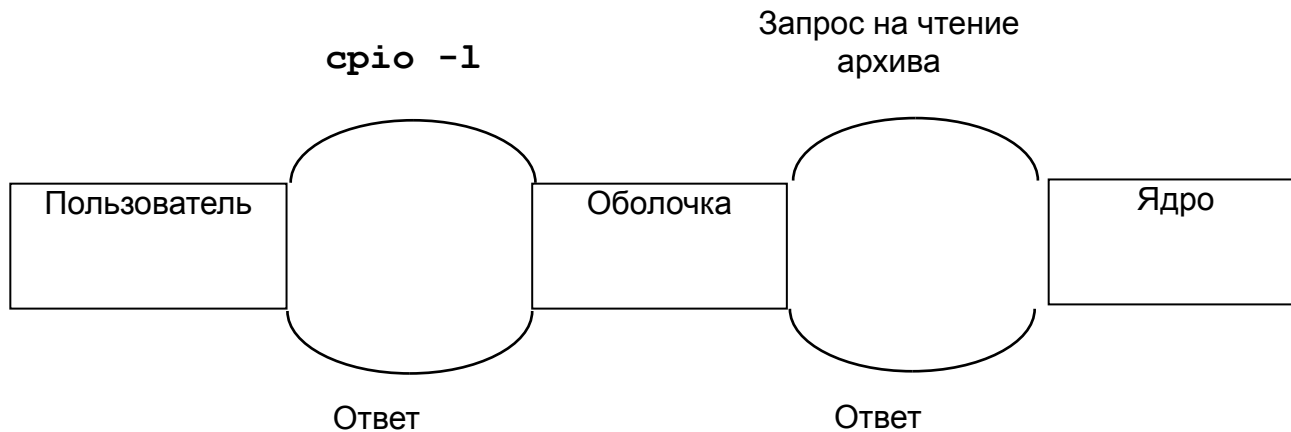


Оболочка отвечает на запрос пользователя без вызова других программ



Оболочка вызывает дочернюю оболочку для выполнения запроса, когда она не может обработать запрос непосредственно

Схемы выполнения команд UNIX 2



Командный процессор должен вызвать ядро, когда делаются запросы к техническим средствам компьютера

СИСТЕМНЫЕ ВЫЗОВЫ

Приведем пример:

[Робачевский А.М. Операционная система UNIX.]

"Системные вызовы осуществляются так же, как и вызовы обычных подпрограмм и функций С. Например, можно считать данные из файла, используя библиотечную подпрограмму fread

`nread=fread(inputbuf, OBJSIZE, numberobjs, fileptr)`

или при помощи низкоуровневого системного вызова read

`nread=read(filedes, inputbuf, BUFSIZE)`

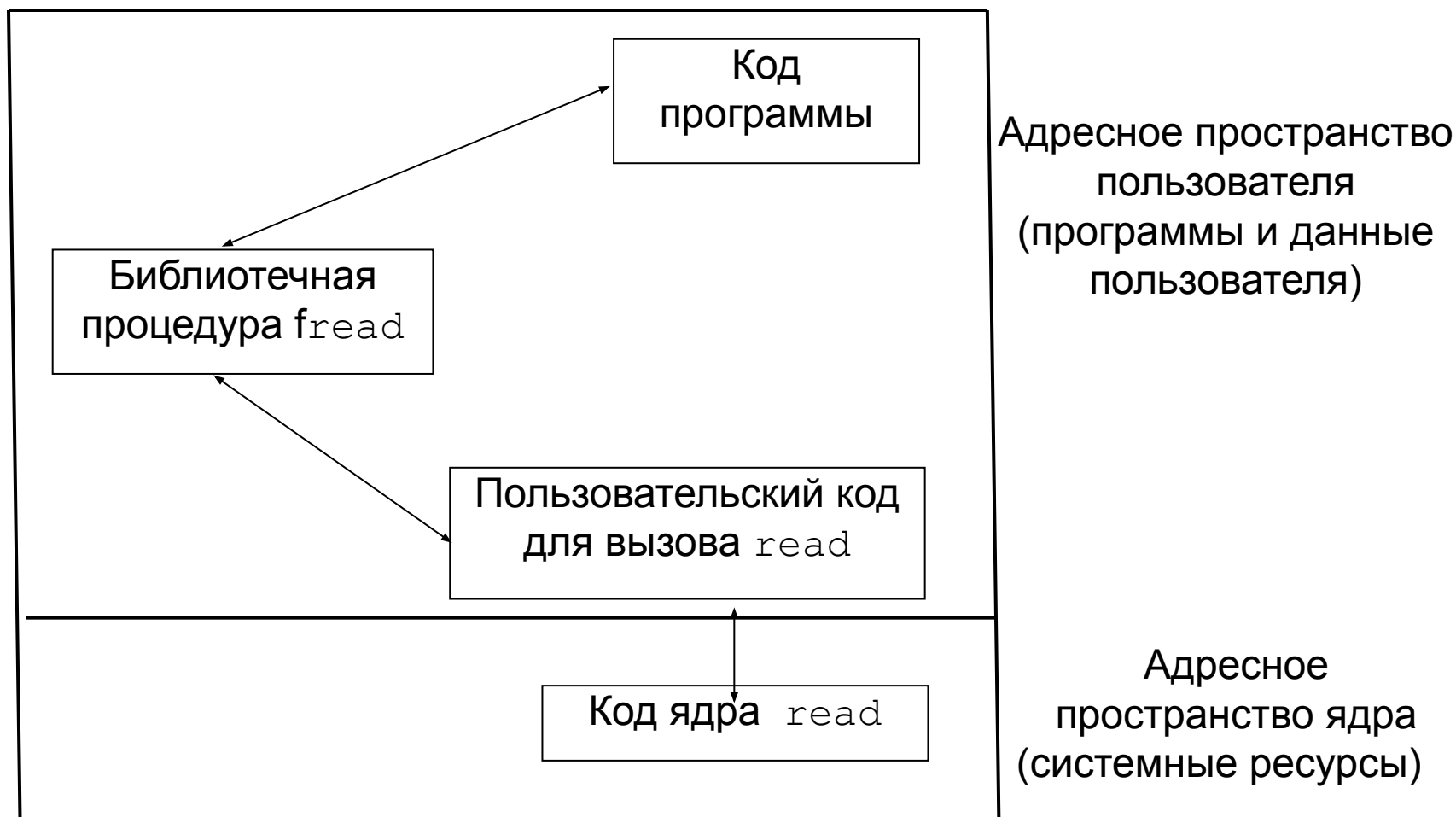
Фрагмент программы, ПОЯСНЯЮЩИЙ СИСТЕМНЫЕ ВЫЗОВЫ

Робачевский А.М. Операционная система UNIX.

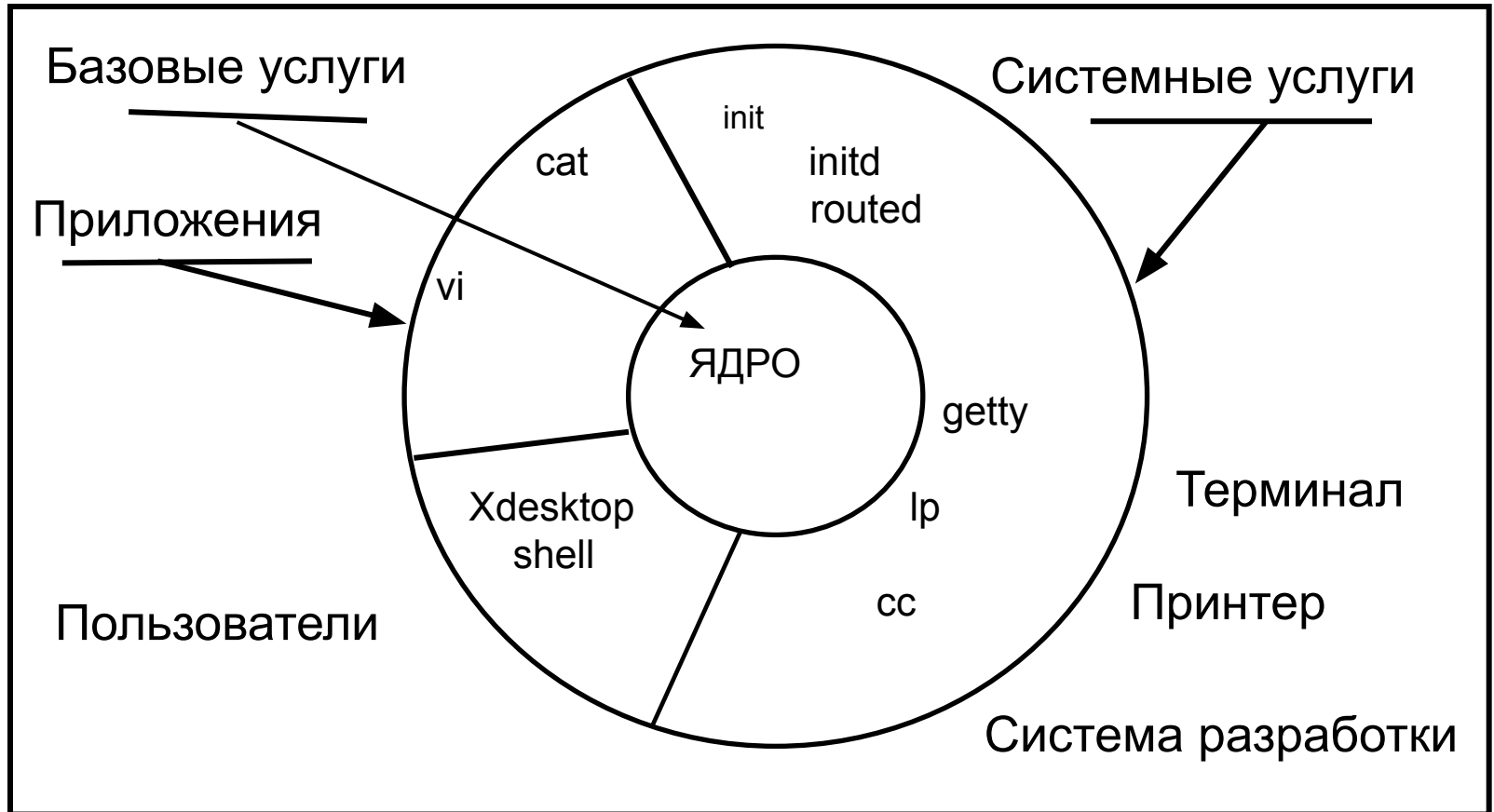
```
main()  
{ int fd;  
  char buf[80];  
/*Откроем файл - получим ссылку (файловый дескриптор) fd*/  
  fd = open("file1", O_RDONLY);  
  read(fd, buf, sizeof(buf)); /*Считаем в буфер buf 80 символов*/  
  close(fd); /*Закроем файл*/  
}
```

При этом операции открытия (`open`), чтения (`read`) и закрытия (`close`) файла выполняются ядром по запросу задачи, а функции `open(2)`, `read(2)` и `close(2)` являются системными вызовами

Связь между кодом программы, библиотечной программой и системным ВЫЗОВОМ

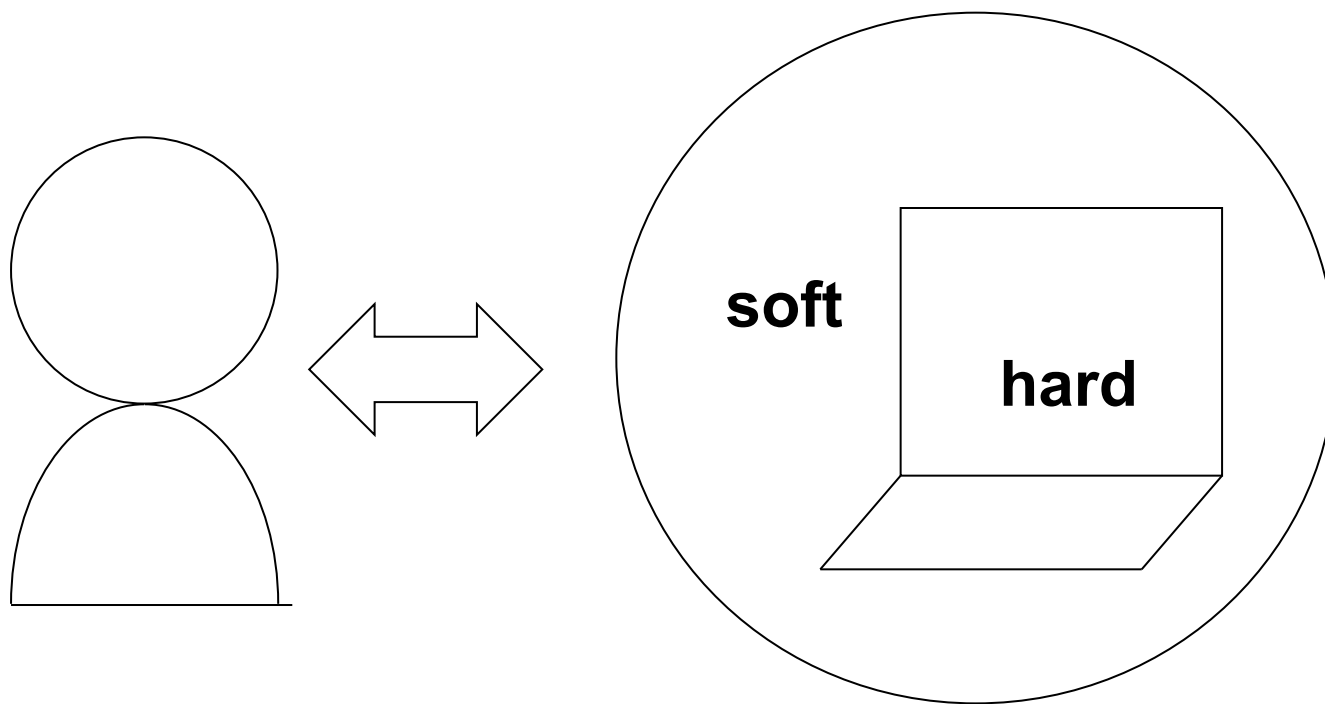


Модель системы UNIX

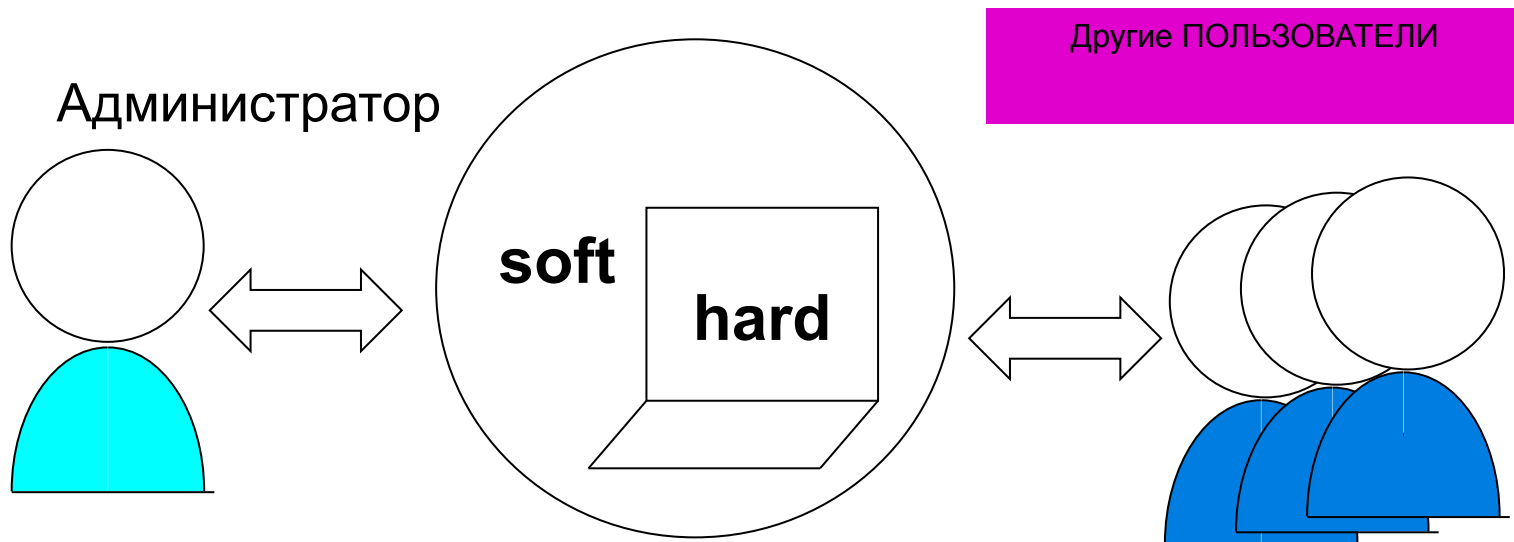


Пользователь и компьютер (программно-аппаратный комплекс)

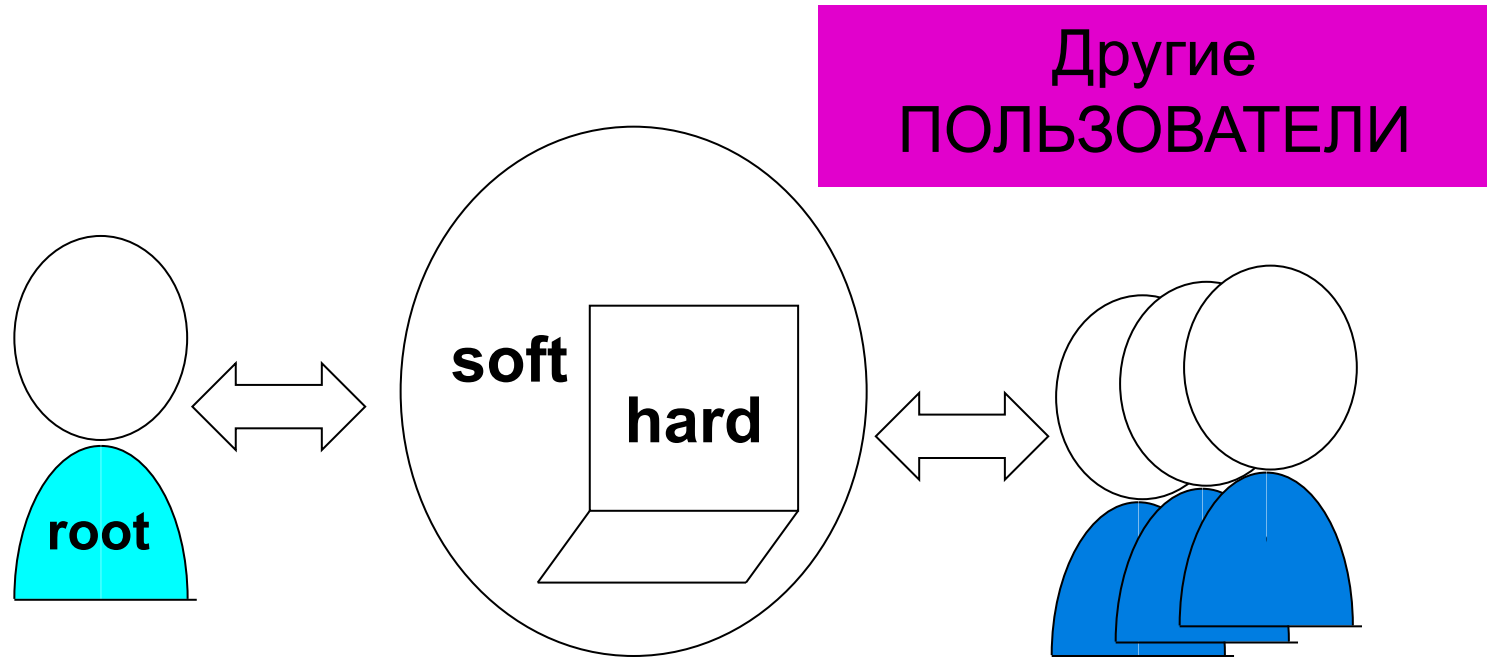
Пользователь и персональный компьютер (простой вариант – Win9X)



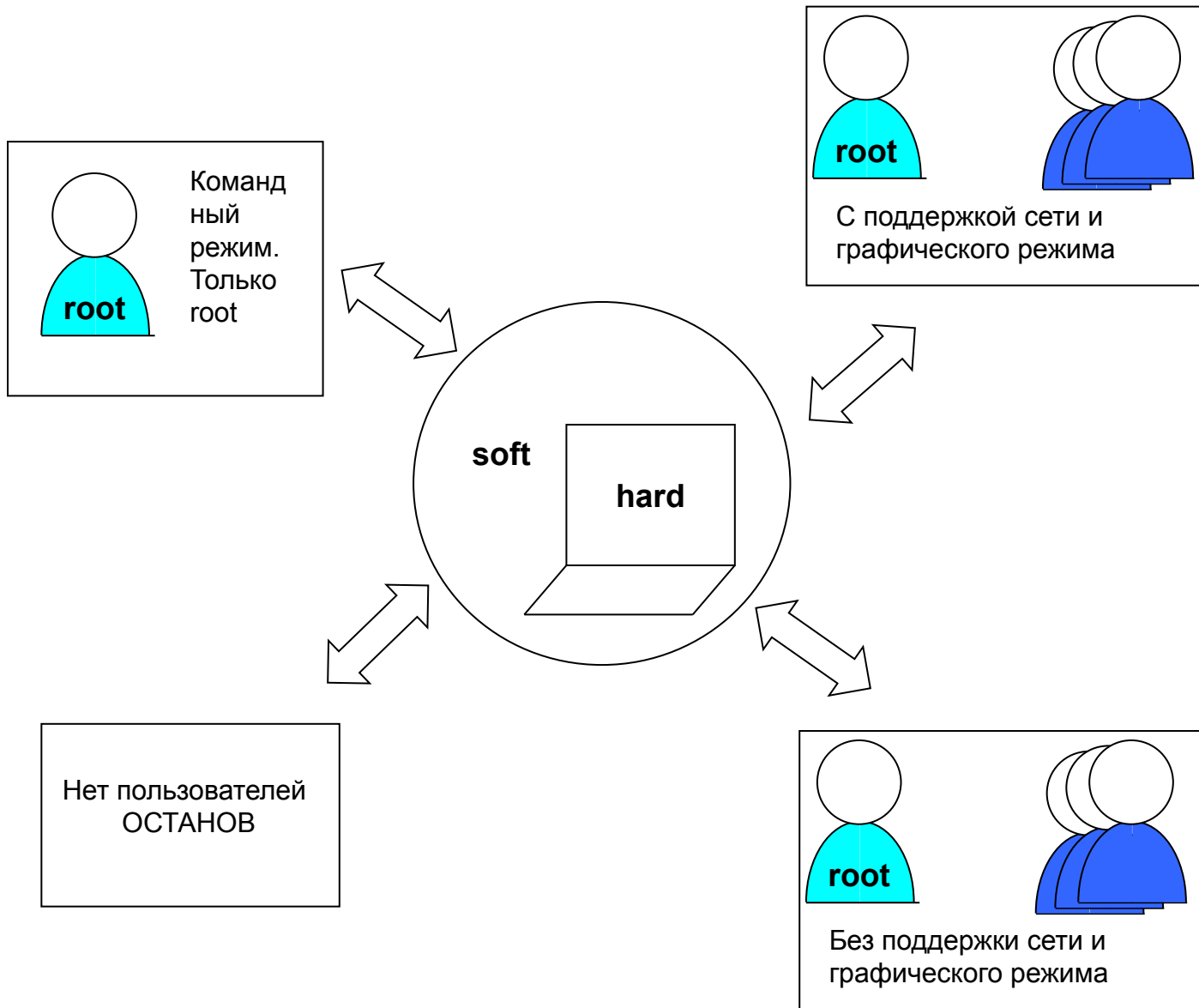
Пользователь и компьютер современных версий Windows



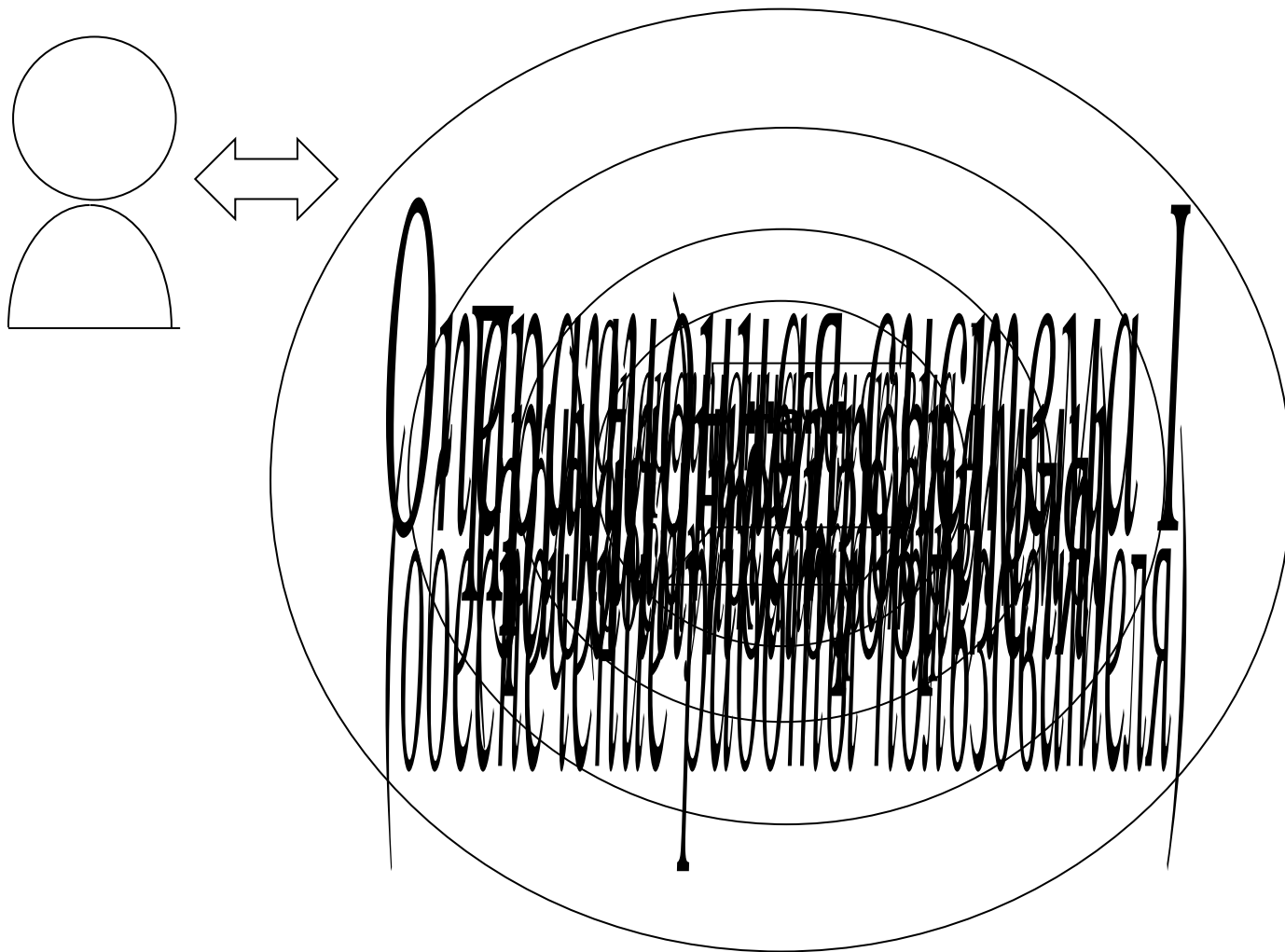
Группы пользователей семейства UNIX/Linux



Виртуальные пользователи (bin, adm, sys, nobody и др.



Пользователи при различных уровнях работы операционных систем UNIX.

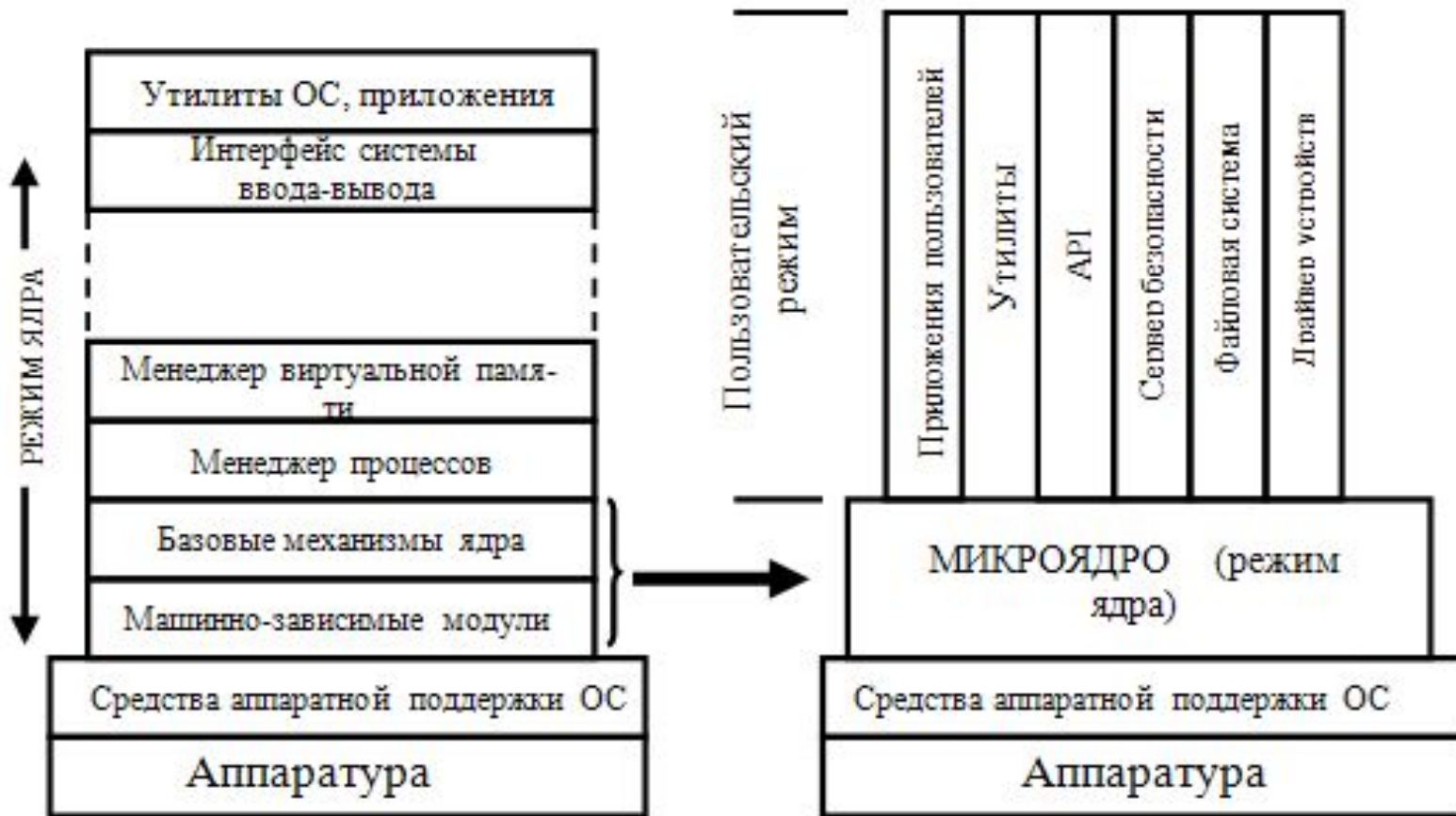


Объединение аппарата + компьютер должно подчиняться воле ПОЛЬЗОВАТЕЛЯ.

Микроядерная архитектура ОС

Переход к микроядерной архитектуре

Классическая многоуровневая архитектура ОС не лишена проблем.



В современных операционных системах различают следующие виды ядер

Наноядро (НЯ). Крайне упрощённое и минимальное ядро, выполняет лишь одну задачу – обработку аппаратных прерываний, генерируемых устройствами компьютера.

Микроядро (МЯ) предоставляет только элементарные функции управления процессами и минимальный набор абстракций для работы с оборудованием.

Экзоядро (ЭЯ) предоставляет лишь набор сервисов для взаимодействия между приложениями, а также необходимый минимум функций, связанных с защитой: выделение и высвобождение ресурсов, контроль прав доступа и т.д.

Монолитное ядро (МНЯ) предоставляет широкий набор абстракций оборудования. Все части ядра работают в одном адресном пространстве.

Модульное ядро (Мод. Я) – современная, усовершенствованная модификация архитектуры МЯ. В отличие от «классических» МНЯ, модульные ядра не требуют полной перекомпиляции ядра при изменении состава аппаратного обеспечения компьютера.

Гибридное ядро (ГЯ) – модифицированные микроядра, позволяющие для ускорения работы запускать «несущественные» части в пространстве ядра. Имеют «гибридные» достоинства и недостатки.

О реализации ядра Windows NT

Наиболее тесно элементы микроядерной архитектуры и элементы монолитного ядра переплетены в ядре Windows NT. Хотя Windows NT часто называют микроядерной операционной системой, это не совсем так.

Микроядро NT слишком велико (более 1 Мбайт), чтобы носить приставку «микро». Компоненты ядра Windows NT располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений, как и положено в микроядерных операционных системах. В то же время все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных, что свойственно операционным системам с монолитным ядром.

Версии Windows NT

Википедия Windows NT.

Название (<i>кодовое название</i>), варианты	номер версии
Windows NT 3.1	3.1.528
Windows NT 3.5 (<i>Daytona</i>)	3.5.807
Windows NT 3.51 (<i>Tukwila</i>)	3.51.1057
Windows NT 4.0 (<i>Indy</i>)	4.0.1381
Windows 2000 (<i>Cairo</i>)	5.0.2195
Windows XP (<i>Whistler</i>)	5.1.2600
Windows Server 2003 (<i>Whistler Server, Windows .NET Server</i>)	5.2.3790
Windows Vista (<i>Longhorn</i>)	6.0.6000
Windows Server 2008 (<i>Longhorn Server</i>)	6.0.6001
Windows 7 (<i>Blackcomb, Vienna</i>)	6.1.7600
Windows Server 2008 R2	6.1.7600

Определение
понятия
Операционная Система

К определению термина ОС

<p>Карпов В.А. и Коньков К.Е. (ИНТЕРНЕТ УНИВЕРСИТЕТ)</p>	<p>Танненбаум А.</p>	<p>Олиферы В.Г. и Н.А. (автономный компьютер)</p>
<ul style="list-style-type: none">◆ виртуальная машина (!);◆ менеджер ресурсов;◆ защита пользователей и программ;◆ постоянно функционирующее ядро;	<ul style="list-style-type: none">◆ расширенная машина;◆ менеджер ресурсов;	<ul style="list-style-type: none">◆ ОС как виртуальная машина;◆ ОС как система управления ресурсами;

К определению ОС (функции)

ИНТЕРНЕТ УНИВЕРСИТЕТ Карпов В.А. Коньков К.Е.	Танненбаум А.	Олиферы В.Г. и Н.А. (автономный компьютер)
<ul style="list-style-type: none">• обеспечение программ средствами синхронизации и коммуникации;• управление памятью;• управление ФС;• управление вводом/выводом;• обеспечение безопасности;	<ul style="list-style-type: none">• процессы;• взаимоблокировка;• ввод/вывод;• файлы;• безопасность;• оболочка; <p style="text-align: center;"><u>(не является частью ОС)</u></p>	<ul style="list-style-type: none">• управление процессами• управление памятью• управление файлами и внешними устройствами• защита данных• интерфейс прикладных программ• пользовательский интерфейс

Приведем определение ОС из Википедии

http://ru.wikipedia.org/wiki/%CE%EF%E5%F0%E0%F6%E8%EE%ED%ED%E0%FF_%F1%E8%F1%F2%E5%EC%E0

Операцио́нная систе́ма, сокр. **ОС** ([англ.](#) *operating system*, *OS*) — комплекс управляющих и обрабатывающих [программ](#)) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как [интерфейс](#)) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между [устройствами](#)) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами [вычислительной системы](#)) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и [прикладными программами](#)) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и прикладными программами, а с другой стороны — предназначены для управления устройствами, управления [вычислительными процессами](#)) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной

Основные функции (простейшие ОС):

- загрузка приложений в оперативную память и их выполнение;
- стандартизованный доступ к периферийным устройствам (устройства ввода-вывода);
- управление оперативной памятью (распределение между процессами, виртуальная память);
- управление доступом к данным на энергонезависимых носителях (таких как жёсткий диск, компакт-диск и т. д.), как правило, с помощью файловой системы;
- пользовательский интерфейс;
- сетевые операции, поддержка стека протоколов

Дополнительные функции:

- параллельное или псевдопараллельное выполнение задач (многозадачность);
- взаимодействие между процессами: обмен данными, взаимная синхронизация;
- защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений;
- разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

Истоки появления «виртуальных машин»

Возможность операционной системы выполнять приложения, написанные для других операционных систем называется **совместимостью**.

Совместимость приложений может быть

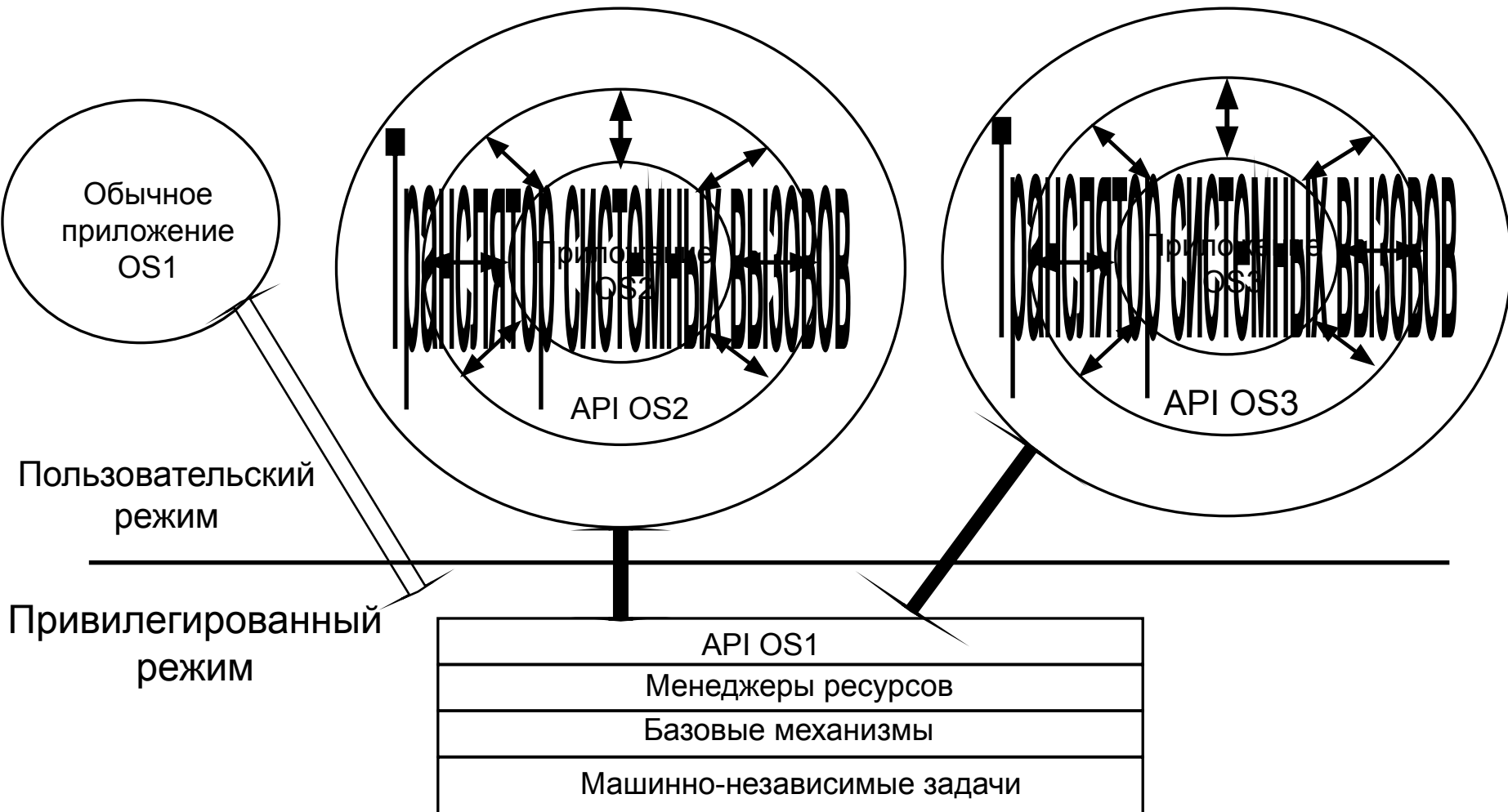
- на двоичном уровне;
- на уровне исходных текстов.

Если процессоры имеют разную архитектуру, необходимо организовать **эмуляцию** двоичного кода (аппаратно или программно).

Организация множественных прикладных сред

Прикладная среда OS2

Прикладная среда OS3



Реализация виртуальной машины

Программа пользователя	Программа пользователя	Программа пользователя
MS-DOS	Linux	Windows
Вирт. hard	Вирт. hard	Вирт. hard
Реальная операционная система		
Реальное hardware		

Виртуальные машины – современный подход реализации множественных прикладных сред

Понятие «монитор виртуальных машин» (**МВМ**) возникло в конце 60-х годов как программный уровень абстракции, разделявший аппаратную платформу на несколько виртуальных машин.

В 90-х годах исследователи из Стэнфордского университета начали изучать возможность применения ВМ для преодоления ограничений оборудования и операционных систем у компьютеров с массовой параллельной обработкой (Massively Parallel Processing, MPP). Они организовали VMware (www.vmware.com).

Современные программные средства ВМ – VMWare Workstation (VMware, Inc.), Virtual PC (Microsoft), VirtualBox (Oracle) и другие.

Облачные вычисления

В настоящее время появились идеи организации «облачных» вычислений.

Концепция. Особая клиент-серверная технология: использование клиентом ресурсов группы серверов в сети, взаимодействующих таким образом, что:

- 1) для клиента вся группа выглядит как единый виртуальный сервер;
- 2) клиент может прозрачно и с высокой гибкостью менять объемы потребляемых ресурсов.

Таким технологиями активно занимаются многие фирмы, в том числе Oracle/Sun, IBM, Google.

Классификация ОС

Классификация ОС по признакам (1-5)

1. По назначению ОС делятся на **универсальные и специализированные**. Специализированные ОС, как правило, работают с фиксированным набором программ (функциональных задач).

2. По способу загрузки можно выделить **загружаемые ОС и системы, постоянно находящиеся в памяти** вычислительной системы. Последние, как правило, специализированные и используются для управления работой специализированных устройств.

3. По особенностям алгоритмов управления ресурсами (процессор)

3.1. Поддержка многозадачности (многопрограммности). По числу одновременно выполняемых задач ОС делятся на 2 класса: однопрограммные (**однозадачные**) – например, MS-DOS, MSX, и многопрограммные (**многозадачные**) – например, ОС ЕС ЭВМ, OS/360, OS/2, UNIX, Windows разных версий.

3.2. Поддержка многопользовательского режима. По числу одновременно работающих пользователей ОС делятся: на **однопользовательские** (MS-DOS, Windows 3x, ранние версии OS/2) и **многопользовательские** (UNIX, Windows NT/2000/2003/XP/Vista/7).

3.3. Виды многопрограммной работы. Специфику ОС во многом определяет способ распределения времени между несколькими одновременно существующими в системе процессами (или потоками). По этому признаку можно выделить 2 группы алгоритмов: ***не вытесняющая многопрограммность*** (Windows3.x, NetWare) и ***вытесняющая многопрограммность*** (Windows 2000/2003/XP, OS/2, Unix).

3.4. Многопроцессорная обработка. Важное свойство ОС – отсутствие или наличие средств поддержки многопроцессорной обработки. По этому признаку можно выделить ОС ***без поддержки мультипроцессирования*** (Windows 3.x, Windows 95) и ***с поддержкой мультипроцессирования*** (Solaris, OS/2, UNIX, Windows NT/2000/2003/XP/Vista/7).

4. По области использования и форме эксплуатации. Обычно здесь выделяют три типа в соответствии с использованными при их разработке критериями эффективности:

- ***системы пакетной обработки*** (OS/360, ОС ЕС);
- ***системы разделения времени*** (UNIX, VMS);
- ***системы реального времени*** (QNX, RT/11).

Первые (пакетной обработки) предназначались для решения задач в основном вычислительного характера, не требующих быстрого получения результатов. Критерий создания таких ОС – максимальная пропускная способность при хорошей загрузке всех ресурсов компьютера. В таких системах пользователь отстранен от компьютера.

Системы разделения времени обеспечивают удобство и эффективность работы пользователя, который имеет терминал и может вести диалог со своей программой.

Системы реального времени предназначены для управления техническими объектами (станок, спутник, технологический процесс, например доменный и т.п.), где существует предельное время на выполнение программ, управляющих объектом.

5. По аппаратной платформе (типу вычислительной техники), для которой они предназначаются

5.1. **Операционные системы для смарт-карт.**

Некоторые из них могут управлять только одной операцией, например, электронным платежом. Некоторые смарт-карты являются JAVA-ориентированным и содержат интерпретатор виртуальной машины JAVA. Апплеты JAVA загружаются на карту и выполняются JVM-интерпретатором. Некоторые из таких карт могут одновременно управлять несколькими апплетами JAVA, что приводит к многозадачности и необходимости планирования.

5.2. **Встроенные операционные системы.**

Управляют карманными компьютерами (Palm OS, Windows CE – Consumer Electronics – бытовая техника), мобильными телефонами, телевизорами, микроволновыми печами и т.п.

5.3. **Операционные системы для персональных компьютеров.**

Например, Windows 9.x, Windows XP, Linux, Mac OS X и др.

5.4. **Операционные системы мини-ЭВМ.**

Например, RT-11 для PDP-11 – ОС реального времени, RSX-11 M для PDP-11 – ОС разделения времени, UNIX для PDP-7.

5.5. *Операционные системы мэйнфреймов* (больших машин). Например, OS/390, происходящая от OS/360 (IBM). Обычно ОС мэйнфреймов предполагает одновременно три вида обслуживания: пакетную обработку, обработку транзакций (например, работа с БД, бронирование авиабилетов, процесс работы в банках) и разделение времени.

5.6. *Серверные операционные системы.*

Например, UNIX, Windows 2000, Linux. Область применения – ЛВС, региональные сети, Intranet, Internet.

5.7. *Кластерные операционные системы.*

Кластер – слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений и представляющихся пользователю единой системой, например, Windows 2000 Cluster Server, Windows 2008 Server, Sun Cluster (базовая ОС – Solaris).

Классификация ОС по признаками (А-Д)

А. Реализация многозадачности

1. Многозадачные (Unix, OS/2, Windows)
2. Однозадачные (MS DOS)

Б. Поддержка многопользовательского режима.

По числу одновременно работающих пользователей выделяют

1. Однопользовательские (MS DOS, Windows 3.1)
2. Многопользовательские (UNIX, WINDOWS NT)

В. Многопроцессорная обработка.

Повышение производительности вычислительных систем сегодня реализуется не только конструирование скоростного процессора, но и разделения исполняемых команд многими процессорами.

Многопроцессорная обработка реализована в Linux, Solaris, Windows NT. Различают симметричные (все действия децентрализованы на отдельном процессоре) и асимметричные (выделяется главный процессор)

Г. Системы реального времени.

Среди многозадачных ОС можно выделить кроме пакетных систем и систем разделения времени выделяют системы реального времени. Они применяются для управления техническими объектами или технологическими процессами. Это накладывает специфические требования на их функции (пример - QNT).

Д. Архитектура ядра.

Имеется два класса: с монолитным ядром (Linux) и микроядерной архитектурой (Mac OS использует технологию Mach).

Танненбаум «Зоопарк операционных систем»

- 1) *Операционные системы для мейнфреймов (OS/390, OS/360).*
- 2) *Серверные операционные системы (Solaris, FreeBSD, Linux и Windows Server 200x).*
- 3) *Многопроцессорные операционные системы (Windows и Linux)*
- 4) *Операционные системы для персональных компьютеров (Linux, FreeBSD, Windows Vista и Macintosh).*
- 5) *Операционные системы карманных персональных компьютеров (Symbian OS и Palm OS1).*

Прим. Ред. Сейчас у нас сложно назвать популярной Palm OS. Зато явно стоит упомянуть WindowsMobile, встречающуюся в разных версиях и различные специализированные версии ОС Linux, включая недавно появившуюся систему Android.

- 6) *Встроенные операционные системы (QNX и VxWorks).*
- 7) *Операционные системы сенсорных узлов (TinyOS).*
- 8) *Операционные системы реального времени.*
- 9) *Операционные системы для смарт-карт.*

Некоторые смарт-карты рассчитаны на применение языка Java. Это значит, что ПЗУ смарт-карты содержит интерпретатор Java Virtual Machine (JVM) — виртуальной машины Java.

Список операционных систем в Википедии содержит более 350 наименований, разделенных на такие группы:

1. Проприетарные

2. Свободные

3. Клоны MS-DOS

4. Авторские/внутренние, не-UNIX

5. «Домашние» OS

6. Операционные системы для БК

7. Операционные системы для Spectrum

8. Прикладные системы

а) Наладонный компьютер (PDA); б) Смартфоны;

в) Нетбуки; г) Маршрутизаторы; д) МиниАТС; е) ОС

для микроконтроллеров, встраиваемые ОС, ОС реального времени

9. Вымышленные операционные системы

Требования, предъявляемые к Операционным Системам

по книге Назаров С.В., Широков А.И.
Многопользовательские операционные системы.

1. **Эффективность.** Под эффективностью вообще любой технической (да и не только технической) системы понимается степень соответствия системы своему назначению, которая оценивается некоторым множеством показателей эффективности.

2. **Надежность и отказоустойчивость.**

Операционная система должна быть, по меньшей мере, так же надежна, как компьютер, на котором она работает. Система должна быть защищена как от внутренних, так и от внешних сбоев и отказов. В случае ошибки в программе или аппаратуре система должна обнаружить ошибку и попытаться исправить положение или, по крайней мере, постараться свести к минимуму ущерб, нанесенный этой ошибкой пользователям.

3. Безопасность (защищенность). Ни один пользователь не хочет, чтобы другие пользователи ему мешали. ОС должна защищать пользователей и от воздействия чужих ошибок, и от попыток злонамеренного вмешательства (несанкционированного доступа). С этой целью в ОС как минимум должны быть средства аутентификации – определения легальности пользователей, авторизации – предоставления легальным пользователям установленных им прав доступа к ресурсам, и аудита – фиксации всех потенциально опасных для системы событий.

4. Предсказуемость. Требования, которые пользователь может предъявить к системе, в большинстве случаев непредсказуемы. В то же время пользователь предпочитает, чтобы обслуживание не очень сильно менялось в течение предположительного времени.

5. *Расширяемость.* В отличие от аппаратных средств компьютера полезная жизнь операционных систем измеряется десятками лет. Примером может служить ОС UNIX, да и MS-DOS. Операционные системы изменяются со временем, как правило, за счет приобретения новых свойств, например, поддержки новых типов внешних устройств или новых сетевых технологий. Если программный код модулей ОС написан таким образом, что дополнения и изменения могут вноситься без нарушения целостности системы, то такую ОС называют расширяемой. Операционная система может быть расширяемой, если при ее создании руководствовались принципами модульности, функциональной избыточности, функциональной избирательности и параметрической универсальности.

6. *Переносимость.* В идеальном случае код ОС должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы (которые различаются не только типом процессора, но и способом организации всей аппаратуры компьютера) одного типа на аппаратную платформу другого типа. Переносимые ОС имеют несколько вариантов реализации для разных платформ, такое свойство ОС называется также многоплатформенностью. Достигается это свойство за счет того, что основная часть ОС пишется на языке высокого уровня (например С, С++ и др.) и может быть легко перенесена на другой компьютер (машинно-независимая часть), а некоторая меньшая часть ОС (программы ядра) является машинно-зависимой и разрабатывается на машинном языке другого компьютера

7. Совместимость. Существует несколько «долгоживущих» популярных ОС (разновидности UNIX, MS-DOS, Windows3.x, Windows NT, OS/2), для которых наработана широкая номенклатура приложений. Для пользователя, переходящего с одной ОС на другую, очень привлекательна возможность – выполнить свои приложения в новой операционной системе. Если ОС имеет средства для выполнения прикладных программ, написанных для других операционных систем, то она совместима с этими системами. Следует различать совместимость на уровне двоичных кодов и совместимость на уровне исходных текстов. Кроме того, понятие совместимости включает также поддержку пользовательских интерфейсов других ОС.

8. Удобство. Средства ОС должны быть простыми и гибкими, а логика ее работы ясна пользователю. Современные ОС ориентированы на обеспечение пользователю максимально возможного удобства при работе с ними. Необходимым условием этого стало наличие у ОС графического пользовательского интерфейса и всевозможных мастеров – программ, автоматизирующих активизацию функций ОС, подключение периферийных устройств, установку, настройку и эксплуатацию самой ОС.

9. Масштабируемость. Если ОС позволяет управлять компьютером с различным числом процессов, обеспечивая линейное (или почти такое) возрастание производительности при увеличении числа процессоров, то такая ОС является масштабируемой

К свойству масштабируемость.

Производители могут предлагать свои ОС в различных, различающихся ценой и производительностью, конфигурациях.

Например, Microsoft продает [Назаров С.В. Операционные среды, системы и оболочки. Основы структурной и функциональной организации: Учеб. Пособие. – М.: - КУДИЦ-ПРЕСС, 2007.]:

- Windows 2003 Server (до 4-х процессоров) – для малого и среднего бизнеса;
- Windows 2003 Advanced Server (до 8 процессоров, 2-узловой кластер) – для средних и крупных предприятий;
- Windows 2003 DataCenter Server (16-32 процессора, 4-узловой кластер) – для особо крупных предприятий.

Эволюция операционных систем

Предшественником ОС следует считать служебные программы (загрузчики и мониторы), а также библиотеки часто используемых подпрограмм, начавшие разрабатываться с появлением универсальных компьютеров 1-го поколения (конец 1940-х годов).

Служебные программы минимизировали физические манипуляции оператора с оборудованием, а библиотеки позволяли избежать многократного программирования одних и тех же действий (осуществления операций ввода-вывода, вычисления математических функций и т.п.).

В 1950-60-х годах сформировались и были реализованы основные идеи, определяющие функциональность ОС: пакетный режим, разделение времени и многозадачность, разделение полномочий, реальный масштаб времени, файловые структуры и файловые системы.

Пакетный режим

Необходимость оптимального использования дорогостоящих вычислительных ресурсов привела к появлению концепции «пакетного режима» исполнения программ. Пакетный режим предполагает наличие очереди программ на исполнение, причём ОС может обеспечивать загрузку программы с внешних носителей данных в оперативную память, не дожидаясь завершения исполнения предыдущей программы, что позволяет избежать простоя процессора.

Разделение времени и многозадачность

Уже пакетный режим в своём развитом варианте требует разделения процессорного времени между выполнением нескольких программ. Необходимость в разделении времени (многозадачности, мультипрограммировании) проявилась ещё сильнее при распространении в качестве устройств ввода-вывода телетайпов (а позднее, терминалов с электронно-лучевыми дисплеями). Поскольку скорость клавиатурного ввода (и даже чтения с экрана) данных оператором много ниже, чем скорость обработки этих данных компьютером, использование компьютера в «монопольном» режиме (с одним оператором) могло привести к простою дорогостоящих вычислительных ресурсов.

Разделение времени и многозадачность 2

Разделение времени позволило создать «многопользовательские» системы, в которых один (как правило) центральный процессор и блок оперативной памяти соединялся с многочисленными терминалами. При этом часть задач (таких, как ввод или редактирование данных оператором) могла исполняться в режиме диалога, а другие задачи (такие, как массивные вычисления) в пакетном режиме.

Разделение полномочий

Распространение многопользовательских систем потребовало решения задачи разделения полномочий, позволяющей избежать возможности модификации исполняемой программы или данных одной программы в памяти компьютера другой (содержащей ошибку или злонамеренно подготовленной) программы, а также модификации самой ОС прикладной программой.

Реализация разделения полномочий в ОС была поддержана разработчиками процессоров, предложивших архитектуры с двумя режимами работы процессора ? «реальным» (программе доступно всё адресное пространство компьютера) и «защищённым» (в котором доступность адресного пространства ограничена диапазоном, выделенном при запуске программы на исполнение).

Реальный масштаб времени

Применение универсальных компьютеров для управления производственными процессами потребовало реализации «реального масштаба времени» («реального времени»), синхронизации исполнения программ с внешними физическими процессами.

Включение функции реального масштаба времени в ОС позволило создавать системы, одновременно обслуживающие производственные процессы и решающие другие задачи (в пакетном режиме и (или) в режиме разделения времени).

Если операция должна быть проведена точно в срок (или в определенный период времени), то мы имеем дело с системой **жесткого реального времени**.

Другой разновидностью подобных систем является система мягкого реального времени, в которой хотя и не желательно, но вполне допустимо несоблюдение срока какого-нибудь действия.

Файловые системы и структуры

Постепенная замена носителей с последовательным доступом (перфолент, перфокарт и магнитных лент) накопителями произвольного доступа (на магнитных дисках) привели к необходимости создавать новые методы и средства системной работы с информацией. Выделяется два уровня работы с данными: логический (для доступа к ним) и физический (внутренне устройство данных на носителе).

Этапы эволюции ОС 1

Одной из первых ОС, использующих эти новейшие решения, была операционная система MCP (главная управляющая программа), созданная фирмой Burroughs для своих компьютеров B5000 в 1963 году.

В этой ОС были реализованы многие концепции и идеи, ставшие впоследствии стандартными для многих операционных систем:

- мультипрограммирование;
- мультипроцессорная обработка;
- виртуальная память;
- возможность отладки программ на исходном языке;
- написание операционной системы на языке высокого уровня.

Этапы эволюции ОС 2

Известной системой разделения времени того периода стала система CTSS (Compatible Time Sharing System) – совместимая система разделения времени, разработанная в Массачусетском технологическом институте (1963 год) для компьютера IBM-7094 [37].

Эта система была использована для разработки в этом же институте совместно с Bell Labs и General Electric системы разделения времени следующего поколения MULTICS (Multiplexed Information And Computing Service). Примечательно, что эта ОС была написана в основном на языке высокого уровня EPL (первая версия языка PL/1 фирма IBM).

Этапы эволюции ОС 3

Одним из важнейших событий в истории операционных систем считается появление в 1964 году семейства компьютеров под названием System/360 фирмы IBM, а позже – System/370 [11]. Это было первой в мире реализацией концепции семейства программно и информационно совместимых компьютеров, ставшей впоследствии стандартной для всех фирм компьютерной отрасли.

Нужно отметить, что основной формой использования ЭВМ, как в системах разделения времени, так и в системах пакетной обработки, стал многотерминальный режим.

Этапы эволюции ОС 4

В начале 70-х годов появились первые сетевые ОС, которые позволили не только рассредоточить пользователей, как в системах телеобработки данных, но и организовать распределенное хранение и обработку данных между компьютерами, соединенных электрическими связями.

Известен проект ARPANET МО США. В 1974 году IBM объявила о создании собственной сетевой архитектуры SNA для своих мэйнфреймов, обеспечивающей взаимодействие типа «терминал-терминал», «терминал-компьютер», «компьютер-компьютер».

В Европе активно разрабатывалась технология построения сетей с коммутацией пакетов на основе протоколов X.25.

Этапы эволюции ОС 5

К середине 70-х годов наряду с мэйнфреймами широкое распространение получили мини-компьютеры (PDP-11, Nova, HP). Архитектура мини-компьютеров была значительно проще, многие функции мультипрограммных ОС мэйнфреймов были усечены. Операционные системы мини-ЭВМ стали делать специализированными (RSX-11M – разделение времени, RT-11 – ОС реального времени) и не всегда многопользовательскими.

Важной вехой в истории мини-компьютеров и вообще в истории операционных систем явилось создание ОС UNIX. Написал эту систему Кен Томпсон (Ken Thompson), один из специалистов по компьютерам в BELL Labs, работавший над проектом MULTICS. Собственно, его UNIX – это усеченная однопользовательская версия системы MULTICS.

Этапы эволюции ОС 6

Все восьмидесятые годы характерны появлением все более совершенных версий UNIX: Sun OS, HP-UX, Irix, AIX и др. Для решения проблемы их совместимости были приняты стандарты POSIX и XPG, определяющие интерфейсы этих систем для приложений.

Еще одним знаменательным событием для истории операционных систем было появление в начале 80-х годов персональных компьютеров. Они послужили мощным толчком для распределения локальных сетей, в результате поддержка сетевых функций стала для ОС ПК необходимым условием. Однако и дружелюбный интерфейс, и сетевые функции появились у ОС ПК не сразу.

Этапы эволюции ОС 7

В 1987 году появилась первая многозадачная ОС для ПК – OS/2, разработанная Microsoft совместно с IBM. Эта была хорошо продуманная система с виртуальной памятью, графическим интерфейсом и возможностью выполнять DOS-приложения. Для нее были созданы и получили распространение сетевые оболочки LAN Manager (Microsoft) и LAN Server (IBM).

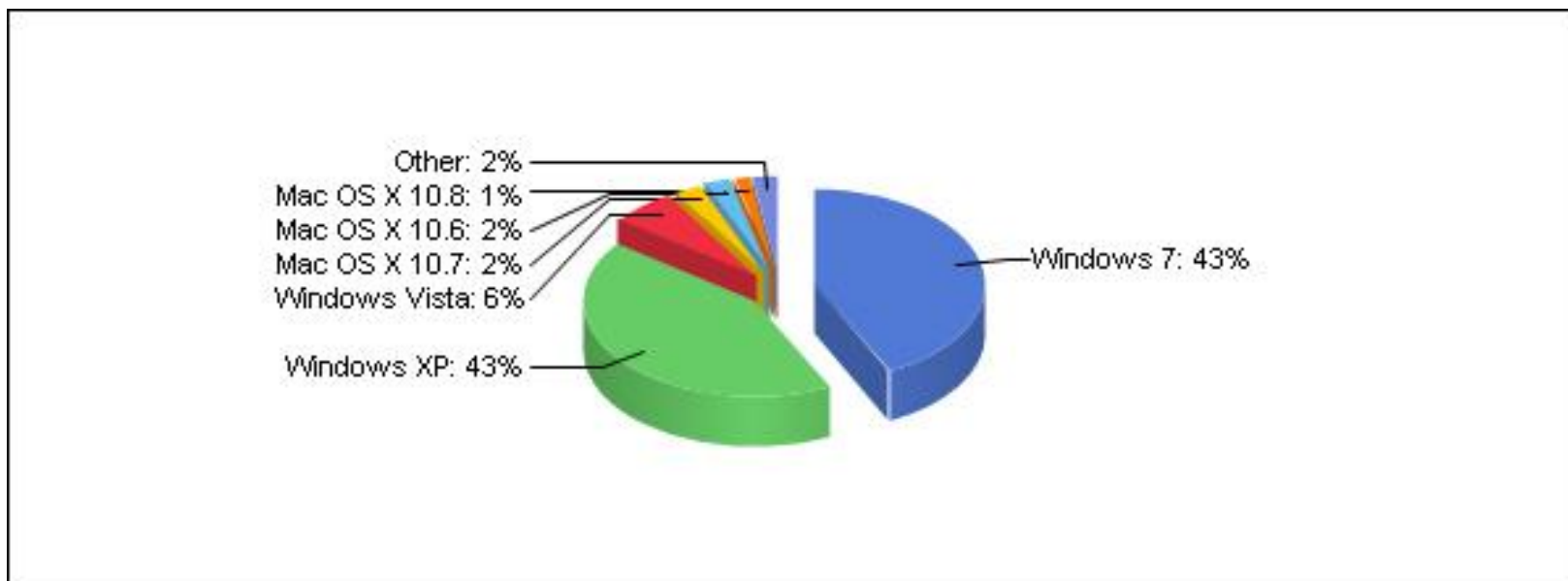
В 80-е годы были приняты основные стандарты на коммуникационные технологии для локальных сетей: в 1980 г. – Ethernet, в 1985 г. – Token Ring, в конце 80-х – FDDI (Fiber Distributed Data Interface).

Этапы эволюции ОС 8

В 90-е годы практически все операционные системы, занимающие заметное место на рынке, стали сетевыми. Сетевые функции встраиваются в ядро ОС, являясь ее неотъемлемой частью. В ОС используются средства мультиплексирования нескольких стеков протоколов, за счет которого компьютеры могут поддерживать одновременную работу с разнородными серверами и клиентами. Появились специализированные ОС, например, сетевая ОС IOS компании Cisco System.

Основные операционные системы

Доли операционных систем для ПК в августе 2012 года (статистика Net Applications).



Типы аппаратных платформ

Мобильные устройства

Персональный компьютер (включая мобильные модели типа notebook)

Серверы

Мейнфреймы (от английского mainframe)

Суперкомпьютеры

ЕЩЕ.

1-9 Зоопарк операционных систем. Таненбаум.

1-7 пункт 5 классификации ОС в книге Назаров С.В., Широков А.И.

Многопользовательские операционные системы.

Классификация компьютеров по областям применения

<http://www.mkgmt.ru/files/material-static/138/contents.htm>

- ПК и рабочие станции
- Серверы
- Мейнфреймы
- Кластерные архитектуры

К терминологии программного обеспечения

- *Программа*
- *Программный продукт*
- *Программное средство*
- *Программный комплекс*

Программа - законченный продукт, пригодный для запуска своим автором на системе, на которой он был разработан.

Программный продукт - программа, которую любой человек может запускать, тестировать, исправлять и развивать. Такая программа должна быть написана в обобщенном стиле, тщательно оттестирована и сопровождается подробной документацией.

Такие определения расположены, например, по адресу
http://mf.grsu.by/Kafedry/kaf001/academic_process/048/28

КОНСПЕКТ ОБЗОРНОЙ ЛЕКЦИИ. Для студентов специальности
Т1002 «Программное обеспечение информационных технологий» (А.М.Кадан, к.т.н., доцент)

Программное средство - это программа или логически связанная совокупность программ на носителях данных, снабженная программной документацией.

Программный комплекс - набор взаимодействующих программ, согласованных по функциям и форматам, точно определенным интерфейсам, и вкуче составляющих полное средство для решения больших задач.

ГОСТ 19781-90

**ОБЕСПЕЧЕНИЕ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ
ПРОГРАММНОЕ**

Термины и определения

**Software of data processing systems. Terms and
definitions**

**Программа (Program) - данные,
предназначенные для управления конкретными
компонентами системы обработки информации
в целях реализации определенного алгоритма**

ГОСТ 28806-90
КАЧЕСТВО ПРОГРАММНЫХ СРЕДСТВ
Термины и определения
Software quality. Terms and definitions

Программа (*en program*): - см. ГОСТ 19781.

Программный продукт (*en software product*):

Программное средство, предназначенное для поставки, передачи, продажи пользователю.

ГОСТ 28806-90

КАЧЕСТВО ПРОГРАММНЫХ СРЕДСТВ

Термины и определения

Software quality. Terms and definitions

Программное средство; ПС (*en software*):

Объект, состоящий из программ, процедур, правил, а также, если предусмотрено, сопутствующих им документации и данных, относящихся к функционированию системы обработки информации.

Примечание. Программное средство представляет собой конкретную информацию, объективно существующую как совокупность всех значимых с точки зрения ее представления свойств каждого из материальных объектов, содержащих в фиксированном виде эту информацию.

Глоссарий (glossary.ru)

Программное изделие - программа или логически связанная совокупность программ:

- записанная на носителях данных;
- являющаяся продуктом промышленного производства;
- снабженная программной документацией;
- предназначенная для широкого распространения посредством продажи или методами freeware, shareware или OEM.

Программный комплекс - набор взаимодействующих программ:

- согласованных по функциям и форматам;
- имеющих единообразные, точно определенные интерфейсы;
- составляющих полное средство для решения больших задач;