

Модуль 1. Классификация, основные понятия и характеристики операционных систем

Лекция 01 (1.1)

Основные определения и принятые соглашения. Эволюция операционных систем. Мультипрограммирование. Режим разделения времени.

Литература

Основная:

Волосатова Т.М., Грошев С.В., Родионов С.В.
Основные концепции операционной
системы UNIX : учеб. пособие / Волосатова
Т. М., Грошев С. В., Родионов С. В. ; МГТУ им.
Н. Э. Баумана. - М. : Изд-во МГТУ им. Н. Э.
Баума-на, 2010.

Таненбаум Э., Бос Х. Современные
операционные системы : пер. с англ. /
Таненбаум Э., Бос Х. - 4-е изд. - СПб : Питер,
2018.

Дополнительная:

Иртегов Д. В. Введение в операционные системы : учеб. пособие
для вузов / 2-е изд., перераб. и доп. - СПб. : БХВ-Петербург, 2012.

Таллок М., Нортроп Т., Ханикэт Дж., Вилсон Э. Ресурсы Windows
7. – ВНВ Русская редакция, 2011.

Леонов В. КомандыLinux. – Эксмо, 2011 – 176с.

Петров Ю.К., Пугачев Е.К., Исследование процессов
инициализации и управления. Семейство UNIX. Электронное
учебное издание кафедры «Компьютерные системы и сети».
Методические указания по выполнению лабораторных работ -
М.: МГТУ имени Н.Э. Баумана, 2016.

Пугачев Е.К., Исследование среды управления системами
Windows. Методические указания по выполнению
лабораторной работы по дисциплине "Операционные
системы".– М.: МГТУ им. Н.Э. Баумана, 2014.

Жизненный цикл программы

Вспомним:

Этапы ЖЦ программ

Входные и выходные данные

Устройства ввода-вывода

Вычислительный процесс

Вопросы:

Когда мы имеем дело с операционной системой?

Дайте определение операционной системы?

Всегда ли нужна операционная система?

Этапы развития ЭВМ

С точки зрения организации вычислительного процесса

1 поколение

2 поколение

3 поколение

4 поколение

5 поколение (?)

Вопросы:

С каких еще точек зрения можно рассматривать эволюцию ЭВМ?

Какие бывают операционные системы?

Какие бывают ОС?

DOS

OS/2

Windows

Linux: AstraLinux, Ubuntu, ...

BSD: FreeBSD

QNX

RTOS

Вопросы:

Какие еще ОС вы знаете?

Основные принципы построения ОС

Частотный принцип

Принцип модульности

Принцип функциональной избирательности

Принцип генерируемости

Принцип функциональной избыточности

Принцип "по умалчиванию"

Принцип перемещаемости

Принцип защиты

Принцип независимости программ от внешних устройств

Принцип открытой и наращиваемой ОС

Вопросы:

Всем ли принципам построения должна отвечать ОС?

Какие принципы могут нарушаться для ОС различного назначения?

Концептуальные основы ОС

Базовые концепты:

процесс

ресурс

концепция виртуализации

взаимодействие с пользователем

дисциплины распределения ресурсов

концепция прерывания

Вопросы:

Может ли процесс рассматриваться как ресурс?

Обязательно ли ОС должна взаимодействовать с пользователем?

Система управления процессами

Система распределения ресурса

**Система распределения оперативной
памяти**

**Подсистема управления вводом-
выводом**

Система управления данными

Мультипрограммирование

Однопрограммные и
мультипрограммные системы
обработки данных

Вопросы:
Чем отличаются
мультипрограммирование и
многозадачность?

Задание

Область использования:

- банкомат
- вычислительный кластер
- телевизор
- мобильный телефон

Вопросы:

Какие требования предъявляются к ОС в каждой из областей?

Модуль 1. Классификация, основные понятия и характеристики операционных систем

Лекция 02 (1.2)

Многопользовательский режим работы. Режим работы и ОС реального времени.

Универсальные операционные системы и ОС специального назначения. Классификация операционных систем. Модульная структура построения ОС.

Повторение

Изученные термины

Вопросы:

Что такое мультипрограммирование?

Каковы принципы построения ОС?

Что такое процесс и что такое ресурс?

Многопользовательский режим работы ОС

Сущность "пользователь"

Что может скрываться под абстракцией
пользователя?

Одна программа - один пользователь

Одна программа - много пользователей

Много программ - один пользователь

Много программ - много пользователей

Вопросы:

Чем отличается
многопользовательский режим от
мультипрограммного?

Универсальные операционные системы и ОС специального назначения

Универсальные ОС - очевидно

ОС специального назначения:

- ОС реального времени
- сетевые ОС
- защищенные системы

Вопросы:

**Почему в гос.учреждениях
используется AstraLinux, а не Ubuntu?**

**Сертификация
(ФСТЭК)**

ОС реального времени

Где используется:

- автоматизация промышленных процессов
- транспорт
- ...

Реальное время:

- жесткое реальное время
- мягкое реальное время

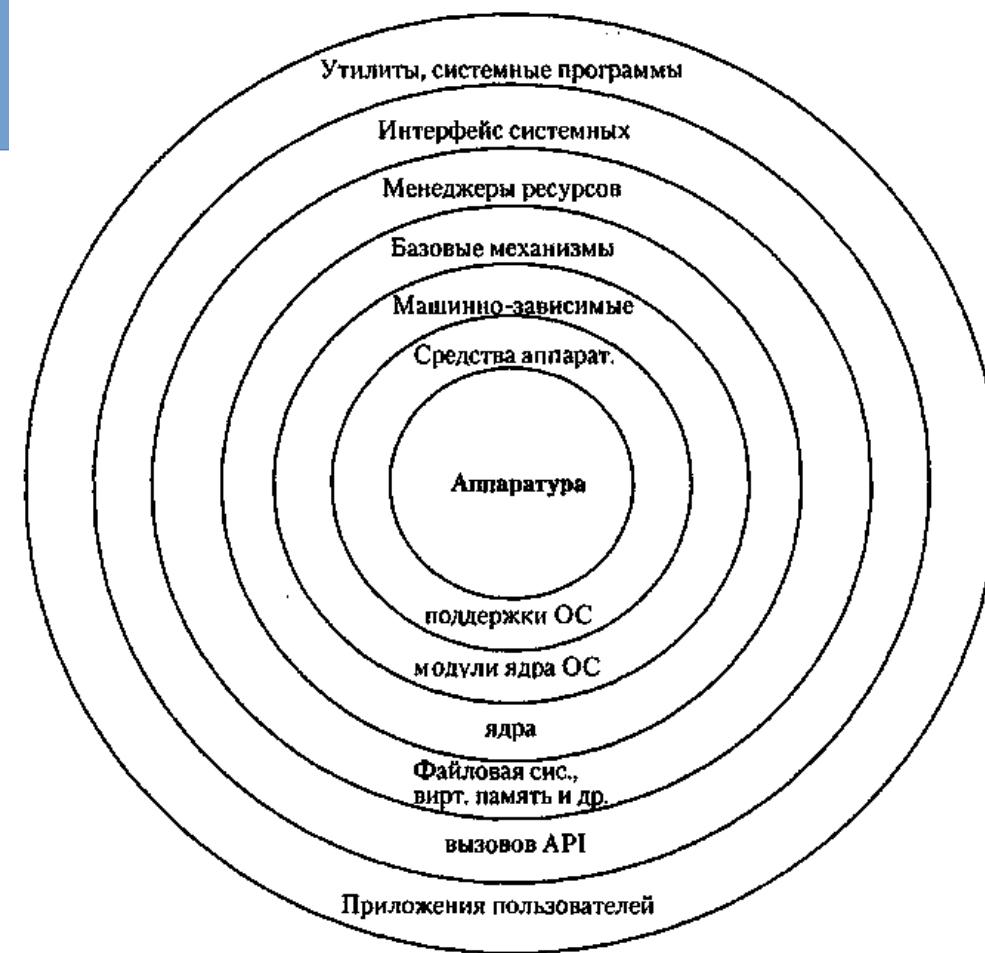
Вопросы:

Почему в мультипрограммных ОС сложно обеспечить жесткое реальное время?

Какого типа реальное время нужно использовать в ОС для беспилотного автомобиля?

Модульная структура ОС

Многослойная структура ОС



Модульная структура ОС

Типовые средства аппаратной поддержки ОС

- средства поддержки привилегированного режима
- средства трансляции адресов
- средства переключения процессов
- система прерываний
- системный таймер
- средства защиты областей памяти

Вопросы:

Как организована система прерываний персонального компьютера?

В чем отличие системного таймера и системными часами?

Модульная структура ОС

Функциональные компоненты ОС

- управление процессами
- управление памятью
- управление файлами и внешними устройствами
- защита данных и администрирование
- интерфейс прикладного программирования
- пользовательский интерфейс

Вопросы:

Какие типы пользовательских интерфейсов бывают?

Пользовательский интерфейс

Типы пользовательских ИФ:

- командная строка;
- графический (оконный);
- ...

Вопросы:

Какие новые ИФ могут появится в связи с появлением новых типов устройств?

Какие разновидности командной строки существуют?

Задание

Необходимо осуществить перенос ОС с персонального компьютера на мобильное устройство.

Вопросы:

Какие слои структуры ОС изменятся при переносе?

Модуль 1. Классификация, основные понятия и характеристики операционных систем

Лекция 03 (1.3)

Генерация операционных систем. Переносимость
операционных систем. Связь операционных систем с
архитектурой процессора

Генерация ОС

Инсталляция – лучший момент для настройки ОС под конкретную вычислительную машину

Загрузка – неплохой момент для настройки ОС под конкретную машину, но не лучший (почему?)

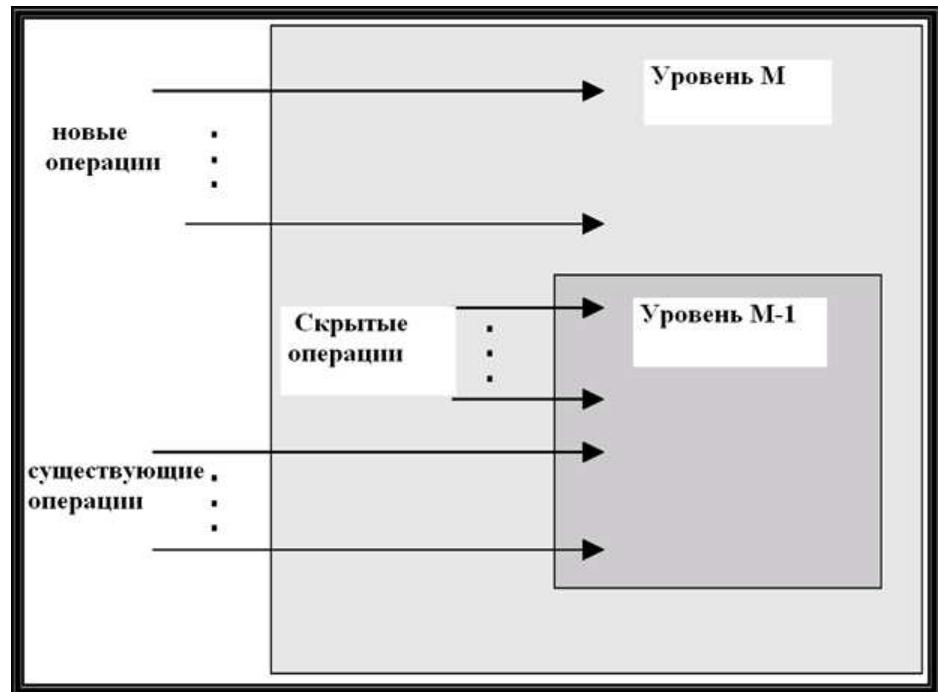
BIOS/EFI/UEFI - базовая конфигурация

Вопросы:

Какие элементы ОС могут зависеть от аппаратной платформы?

Что такое модульность (из курса программирования)?

Уровни абстракции ОС



Ядро операционной системы

Два принципа построения ядра ОС:

- монолитное ядро;
- микроядро.

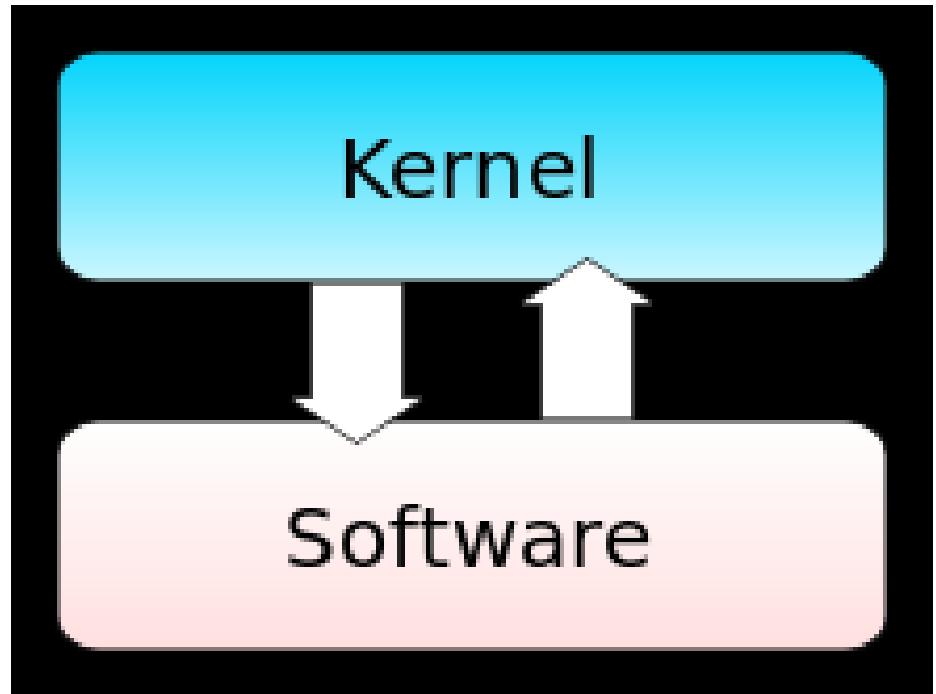
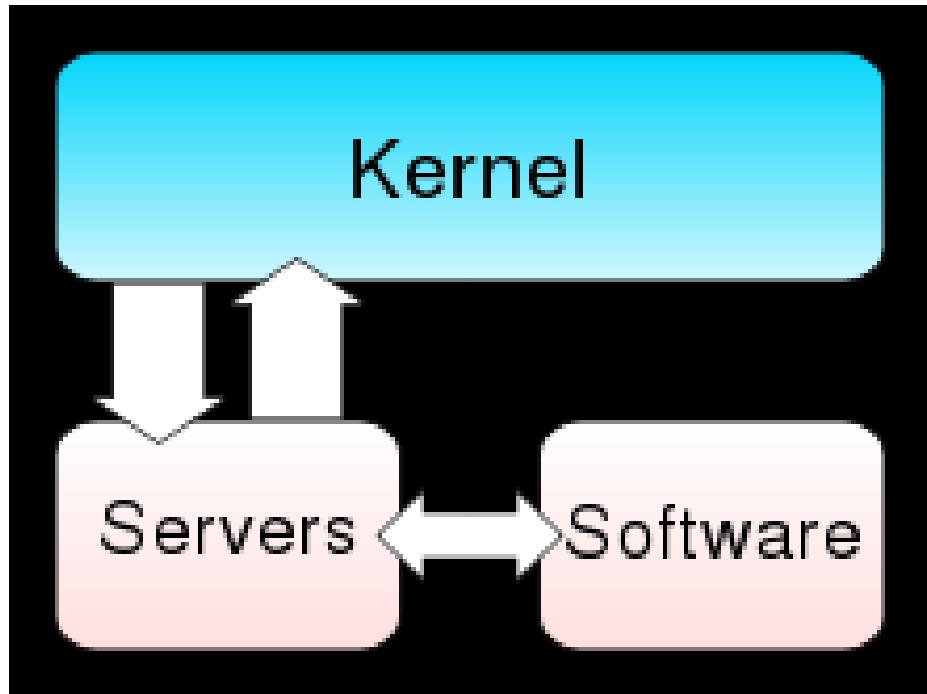
Другие подходы:

- гибридное ядро;
- экзоядров;
- наноядро;
- ваш вариант :)

Вопросы:

Какое ядро используется в ОС ваших телефонов?

Микроядро VS Монолитное ядро



Переносимость ОС

ОС зависит от архитектура процессора

ОС зависит от архитектуры ВС

Вопрос:

Как решается проблема
переносимости?

Первый пример переносимости – UNIX

Язык программирования

Минимизация специфического
ассемблерного кода

Создание API

Создание абстракций (всё есть файл)

Аппаратно-зависимые компоненты ОС

Набор средств аппаратной поддержки ОС

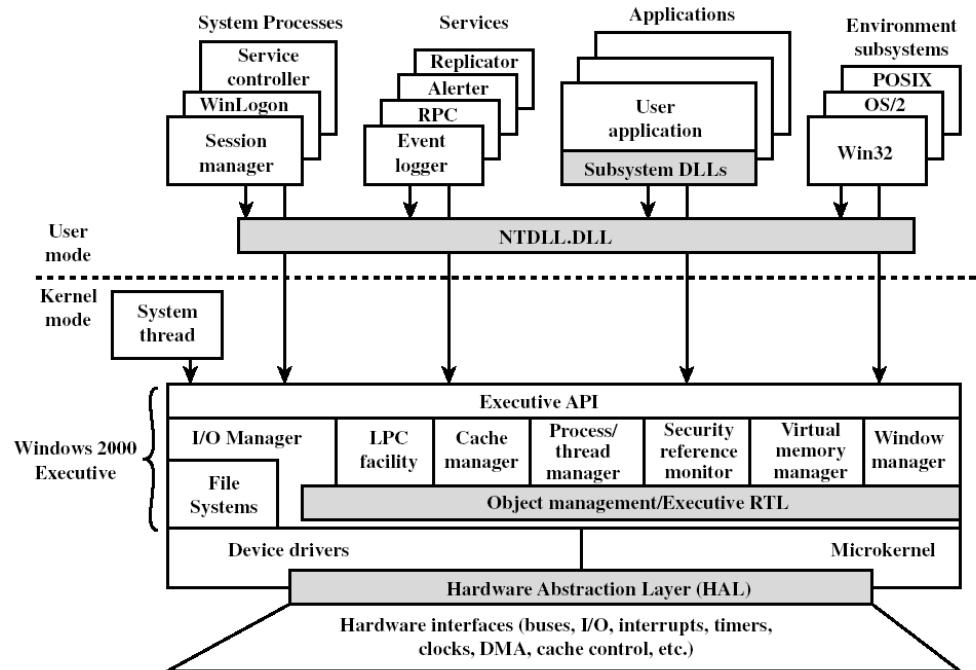
- средства поддержки привилегированного режима;
- средства трансляции адресов;
- средства переключения процессов;
- система прерываний;
- системный таймер;
- средства защиты областей памяти.

Вопросы:

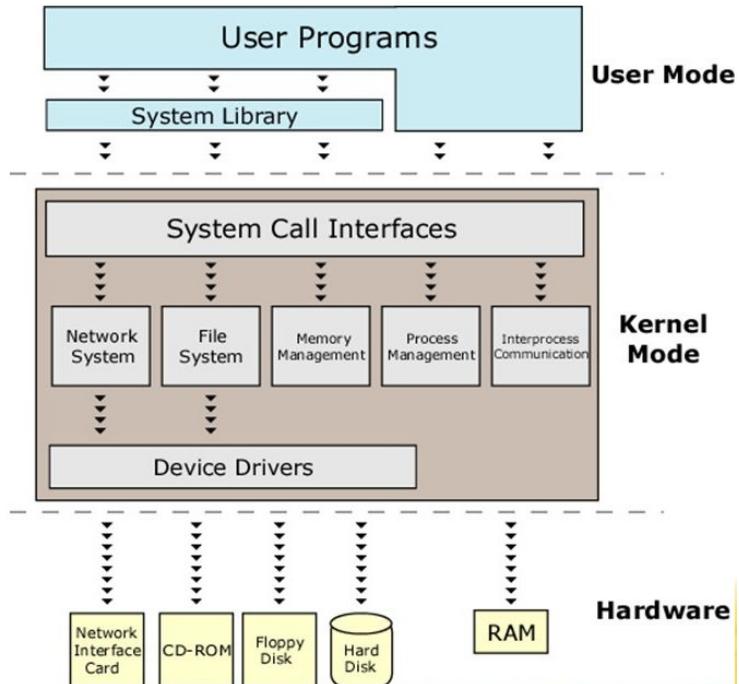
Какие архитектуры процессоров вам известны?

В чем отличие гарвардской архитектуры от стенфордской (фон неймановской)?

Переносимость Windows



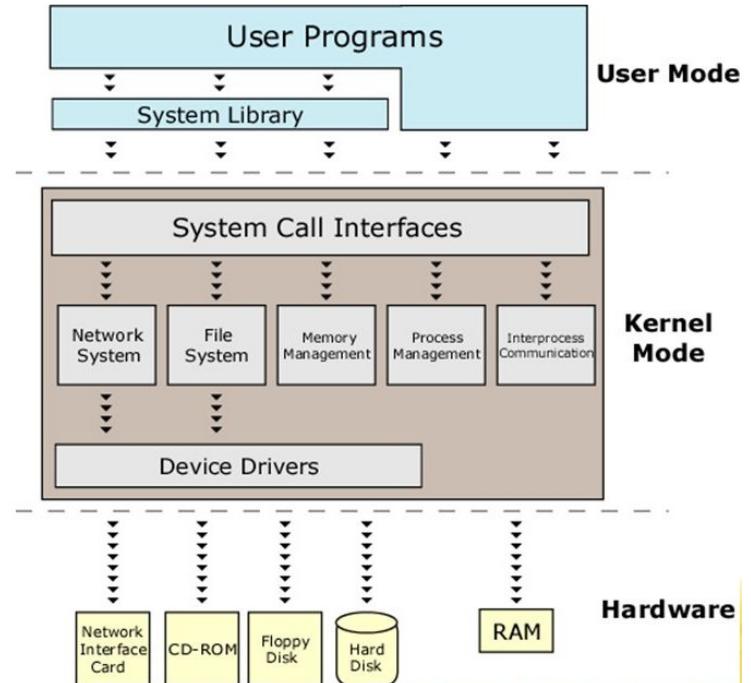
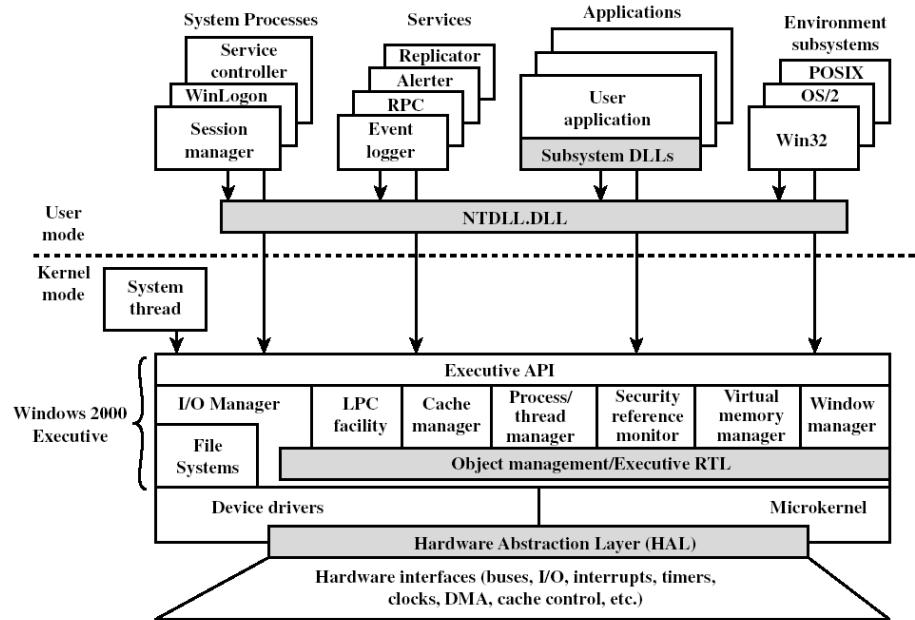
Переносимость Linux



Задание

Сравните между собой структуру каждой представленной ОС.
Какие достоинства и недостатки есть у каждой из них?
Какую из указанных ОС легче перенести на другую архитектуру ВС?

Для сравнения



Модуль 1. Классификация, основные понятия и характеристики операционных систем

Лекция 04 (1.4)

Понятие вычислительной системы. Виды вычислительных ресурсов. Потребности ОС в вычислительных ресурсах

Определение

Под вычислительной системой (ВС) будем понимать совокупность **взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения**, предназначенную для **сбора, хранения, обработки и распределения информации**.

Типы ВС

Различают вычислительные системы (ВС):

- однопрограммные и многопрограммные (в зависимости от количества программ, одновременно находящихся в оперативной памяти);
- индивидуального и коллективного пользования (в зависимости от числа пользователей, которые одновременно могут использовать ресурсы ВС);
- с пакетной обработкой и разделением времени (в зависимости от организации и обработки заданий);
- однопроцессорные, многопроцессорные и многомашинные (в зависимости от числа процессоров);
- сосредоточенные, распределенные (вычислительные сети) и ВС с теледоступом (в зависимости от территориального расположения и взаимодействия технических средств);
- работающие или не работающие в режиме реального времени (в зависимости от соотношения скоростей поступления задач в ВС и их решения);
- универсальные, специализированные и проблемно-ориентированные (в зависимости от назначения).

Режимы работы ВС

1. Мультипрограммирование
2. Режим реального времени
3. Однопрограммный режим работы вычислительной системы (ВС)
4. Мультипрограммный режим работы вычислительной системы (ВС)
5. Режим пакетной обработки
6. Режим коллективного доступа

Виды ресурсов ВС

Под ресурсом понимают какой-либо объект, который может распределяться внутри вычислительной системы (ВС) между конкурирующими за него процессами.

Ресурс выделяется процессу на определенный интервал времени.

Ресурсы запрашиваются, используются и освобождаются процессами.

Вопросы:

Какие ресурсы вам известны?

Классификация ресурсов ВС

По форме реализации:

- аппаратные ресурсы (Hard);
- программные ресурсы (Soft);
- информационные ресурсы.

По способу выделения ресурса различают:

- неделимые ресурсы – предоставляются процессу в полное распоряжение;
- делимые ресурсы – предоставляются процессу в соответствии с запросом на требуемое количество ресурса.

Классификация ресурсов ВС

По реальности существования различают:

- физический ресурс – реально существует и при распределении обладает всеми присущими ему свойствами;
- виртуальный ресурс – программно-аппаратная модель физического ресурса.

По месту размещения ресурса различают:

- локальные ресурсы – принадлежат автономному компьютеру;
- удаленные ресурсы – принадлежат рабочим станциям или серверам, входящим в состав сети.

Ресурсы и ОС

ОС выполняет две группы функций:

- предоставление пользователям и программистам вместо реальной аппаратуры компьютера расширенной виртуальной машины, с которой удобней работать и легче програмировать;
- повышение эффективности использования компьютера путем рационального управления его ресурсами.

Вопросы:

С какими ресурсами имеет дело ОС с точки зрения пользователя в соответствие с каждой группой функций?

Управление ресурсами в ОС

Управление ресурсами включает решение следующих задач:

- планирование ресурса, то есть определение, кому, когда, а для делимых ресурсов и в каком количестве, необходимо выделить данный ресурс;
- выделение ресурса;
- отслеживание состояния ресурса, то есть поддержание оперативной информации о том, занят или не занят ресурс, а для делимых ресурсов – какое количество ресурса уже распределено, а какое свободно;
- освобождение ресурса.

Вопросы:

Какие элементы должна иметь ОС для решения указанных задач?

Задание

Приведите примеры различных ресурсов ОС, которые вы знаете.

Проведите классификацию указанных ресурсов.

Какие есть особенности в каждом типе ресурсов?

Модуль 2. Основные принципы построения операционных систем

Лекция 05 (2.1)

Управление процессами и потоками. Алгоритмы планирования
вычислительных процессов

Процесс

Что такое процесс?

Типы систем с точки зрения одновременно выполняемых процессов:

- однозадачная;
- многозадачная.

!!! В каждый конкретный момент времени процессор выполняет только одной процесс

Атрибуты процесса

Счетчик команд

Регистры

Переменные

Вопросы:

Что такое счетчик команд?

Что такое регистры ЦП? Какие бывают регистры?

Иерархическая структура процесса

Иерархия процессов в *NIX

Родительские процессы

Дочерние процессы

Группы процессов

Процесс init

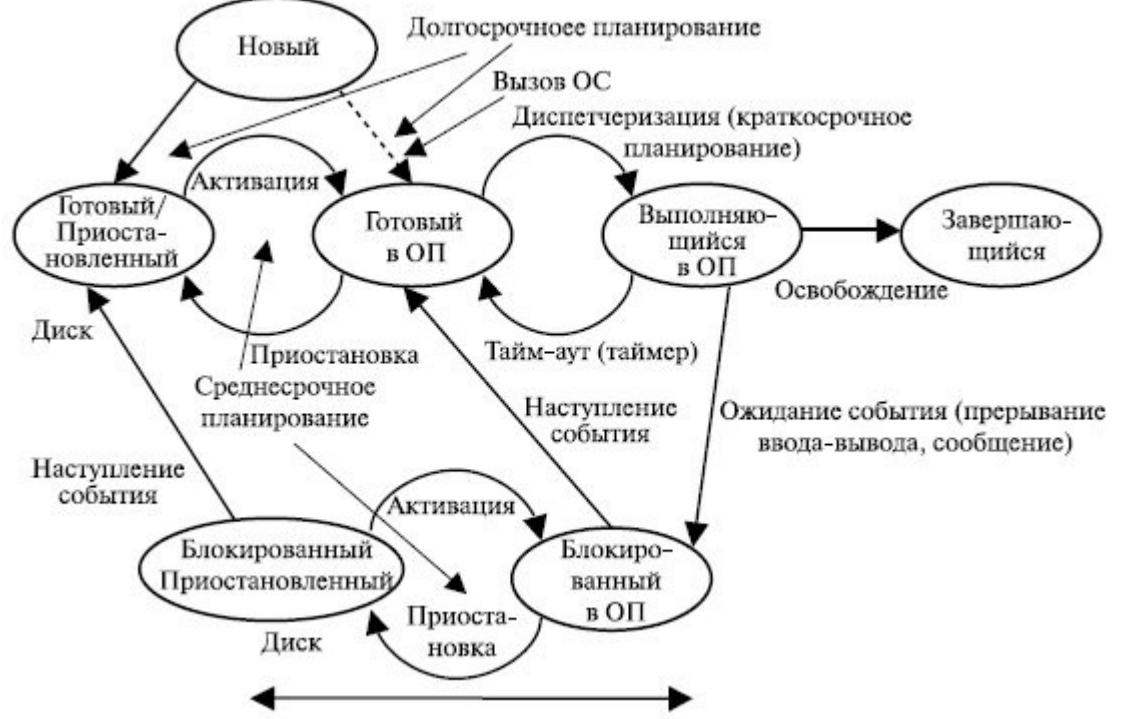
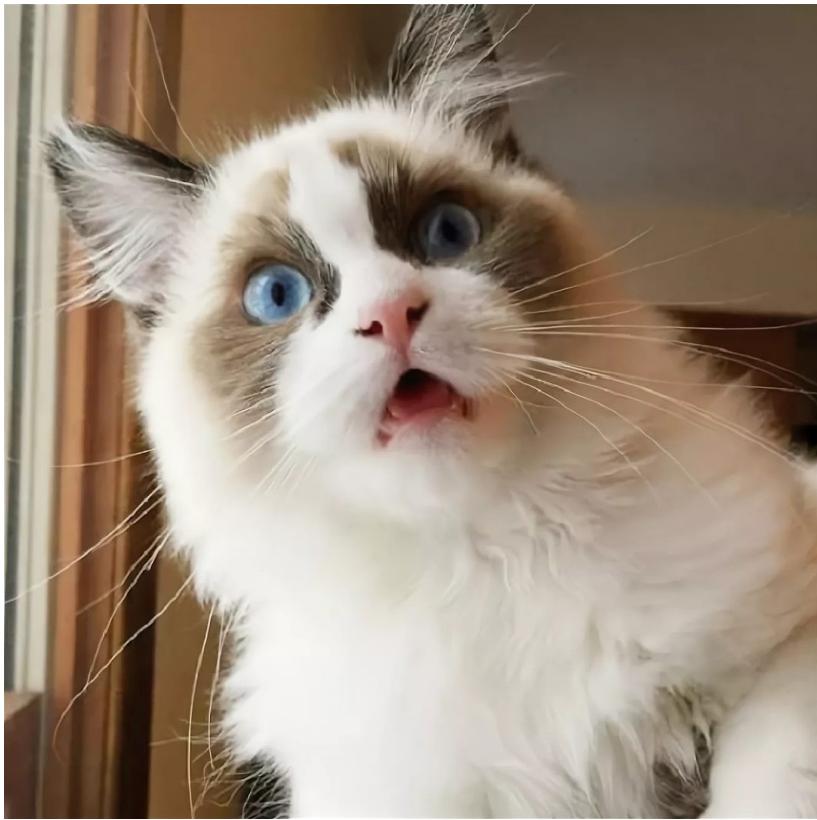
Иерархия процессов в Windows

Есть... но не используется

Граф состояний процесса



Граф состояний процесса



Граф состояний процесса

Состояния:

Выполнение (работа)

Готовность

Блокировка



Граф состояний процесса

Создание процесса

- ...

Завершение процесса

- ...



Планирование процессов

Планировщик процессов

Среды:

- 1) Системы пакетной обработки данных
- 2) Интерактивные системы
- 3) СРВ

Вопросы:

Приведите примеры работы нескольких процессов одновременно?
Когда это может быть нужно?

Алгоритмы планирования

1. Пакетные системы

- первым пришел – первым обслужен
- кратчайшая задача – первая
- наименьшее оставшееся время исполнения
- трехуровневое планирование

3. Системы реального времени

2. Интерактивные системы

- циклическое планирование
- приоритетное планирование
- самый короткий процесс – следующий
- гарантированное планирование
- лотерейное планирование
- справедливое планирование

Модуль 2. Основные принципы построения операционных систем

Лекция 06 (2.2)

Вытесняющие и невытесняющие алгоритмы планирования.
Средства синхронизации и взаимодействия процессов.

Типы многозадачности

Многозадачность:

- невытесняющая
- вытесняющая

Алгоритмы планирования процессов

1. Пакетные системы

- первым пришел – первым обслужен
- кратчайшая задача – первая
- наименьшее оставшееся время исполнения
- трехуровневое планирование

2. Интерактивные системы

- циклическое планирование
- приоритетное планирование
- самый короткий процесс – следующий
- гарантированное планирование
- лотерейное планирование
- справедливое планирование

Межпроцессное взаимодействие

Проблемы:

- передача информации от одного процессора к другому
- ситуация попадания двух процессов в критическую секцию
- согласование действий процессов

Вопросы:

- Что такое критическая секция?
- Что такое состояние состязания?
- Каковы условия избежания состязаний?

Способы обеспечения взаимного исключения

Способы избежания взаимного исключения:

Запрещение прерываний

Блокирующие переменные

Аппаратные решения

Передача сообщений

Барьеры

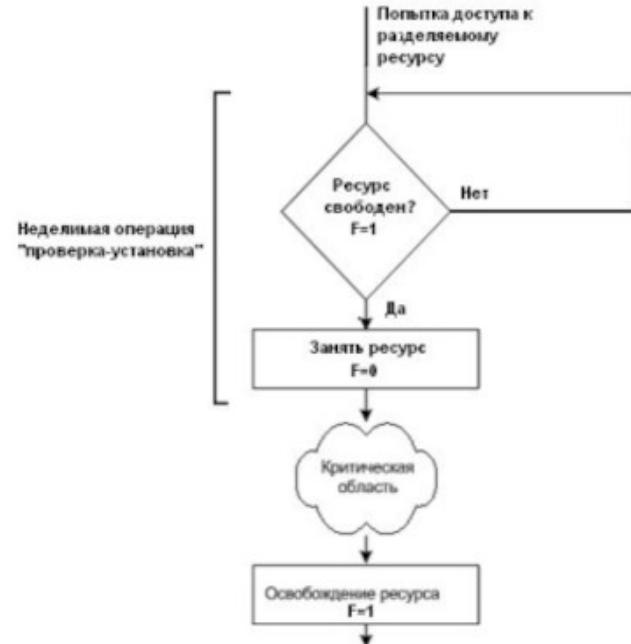
Блокирующие переменные

Семафоры (семафоры Дейкстры)

Мьютексы

Вопрос:

Почему операция “проверка – установка” должна быть неделимой?



Задачи

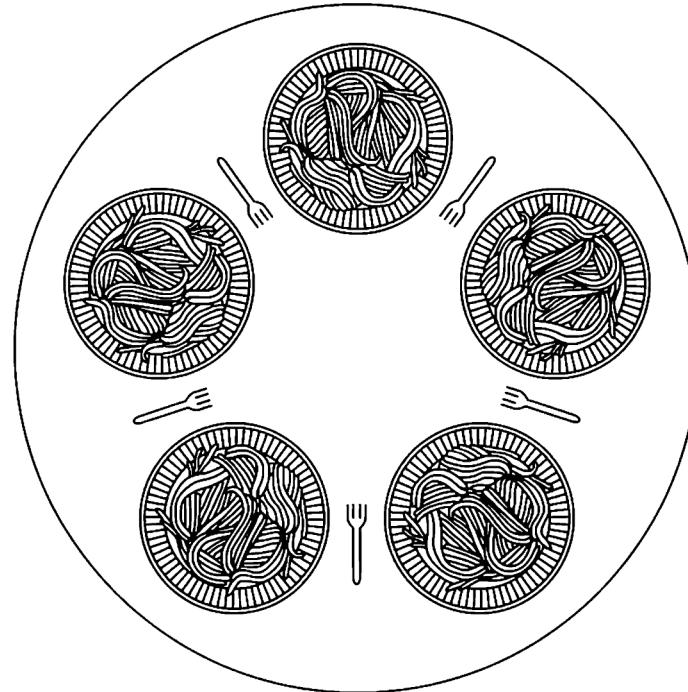
Классические проблемы:

Проблема обедающих философов

Проблема “Читателей и Писателей”

Проблема “Производителей и
Потребителей”

Проблема спящего брадобрея



Другие способы обеспечения взаимодействия

Мониторы

Аппаратные подходы

Пороги

Передача сообщений

Упражнения

Задача	Способ обеспечения взаимодействия
Проблема обедающих философов	
Проблема “Читателей и Писателей”	Мьютексы
Проблема “Производителей и Потребителей”	Семафоры
Проблема спящего брадобрея	

Потоки

Собственные ресурсы потока:

- счетчик команд
- регистры
- стек
- состояние

Разделяемые ресурсы потоков:

- адресное пространство
- глобальные переменные
- открытые файлы
- дочерние процессы
- сигналы и их обработчики
- информация об использовании ресурсов

Модуль 2. Основные принципы построения операционных систем

Лекция 07 (2.3)

Управление оперативной памятью. Методы распределения памяти

Управление памятью

Управление памятью в ранних
операционных системах:

Загрузка программы с внешнего
накопителя в память

Вопросы:

Какие внешние накопители
использовались в ранних ОС?

Функции управления памятью

С появлением мультипрограммирования появились новые функции:

- отслеживание свободной и занятой памяти
- выделение памяти процессам и освобождение памяти по их завершении
- вытеснение кодов и данных процессов из ОП на диск (и загрузка)
- настройка адресов программ
- дефрагментация памяти
- выделение и резервирование памяти для служебных нужд
- защита памяти

Вопросы:

Для каких служебных нужд может выделяться и резервироваться память?

Типы адресов

На разных этапах ЖЦ используются различные типы адресов:

- символические имена;
- виртуальные адреса (виртуальное адресное пространство);
- физические адреса.

Вопросы:

В каком случае транслятор генерирует машинный код сразу в физические адреса?

Модели адресных пространств

Существует несколько моделей адресных пространств:

- плоская, линейная
- сегментная
- другие (с более сложной организацией)

Вопросы:

Почему сейчас чаще всего используется именно плоская модель памяти? Какими факторами это обусловлено?

Преобразование виртуальных адресов в физические

Существующие подходы:

- перемещаемый загрузчик
- динамическое преобразование

Вопросы:

Какие есть достоинства и недостатки при использовании каждого подхода преобразования виртуальных адресов в физические?

Преобразование виртуальных адресов в физические

NB

Стоит различать максимально возможное виртуальное адресное пространство и назначенное (выделенное) процессу виртуальное адресное пространство. Максимальный размер виртуального адресного пространства ограничивается только разрядностью адреса, присущей данной архитектуре компьютера.

Вопросы:

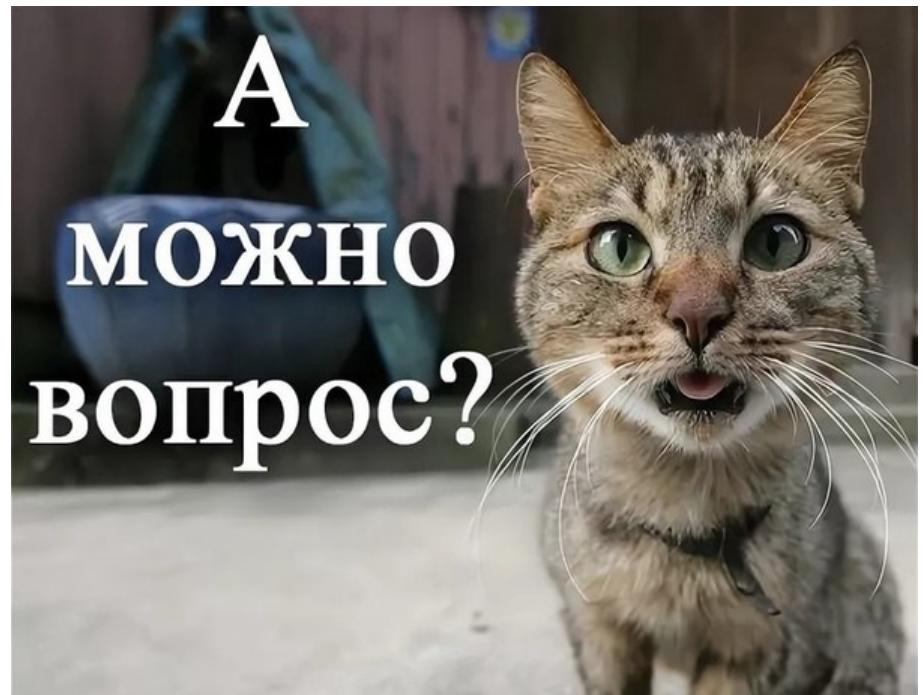
Какой объем максимального виртуального адресного пространства может быть у 32-разрядной системы? А у 64-разрядной?

Должен ли совпадать объем физической памяти в компьютере с максимальным размером виртуального адресного пространства?

Проблемы распределения памяти

При распределении памяти необходимо ответить на следующие вопросы:

- выделять ли непрерывную область памяти?
- можно ли менять расположение сегментов в ОП?
- что делать, если все сегменты программы не помещаются в имеющуюся память?



Классы алгоритмов распределения памяти

Существуют следующие классы алгоритмов:

- используется перемещение сегментов между ОП и диском
- внешняя память не привлекается

Методы распределения памяти

Без использования внешней памяти

а) с фиксированными разделами

б) с динамическими разделами

в) с перемещаемыми разделами

С использованием внешней памяти

г) страничное распределение

д) сегментное распределение

е) сегментно-страничное
распределение

Модуль 2. Основные принципы построения операционных систем

Лекция 08 (2.4)

Распределение памяти фиксированными разделами и
разделами переменной величины

Распределение памяти фиксированными разделами

Особенности:

Разделы фиксированной величины

Использовалось в основном в ЭВМ 70-х
годов прошлого века

Вопрос:

В какой момент может осуществляться
разбиение памяти?

Распределение памяти фиксированными разделами

Разбиение может быть выполнено:

вручную оператором во время старта системы

во время установки операционной системы

NB. После разбиения размеры разделов не меняются

Вопрос:

Чем определяется невозможность разбиения памяти во время работы ОС?

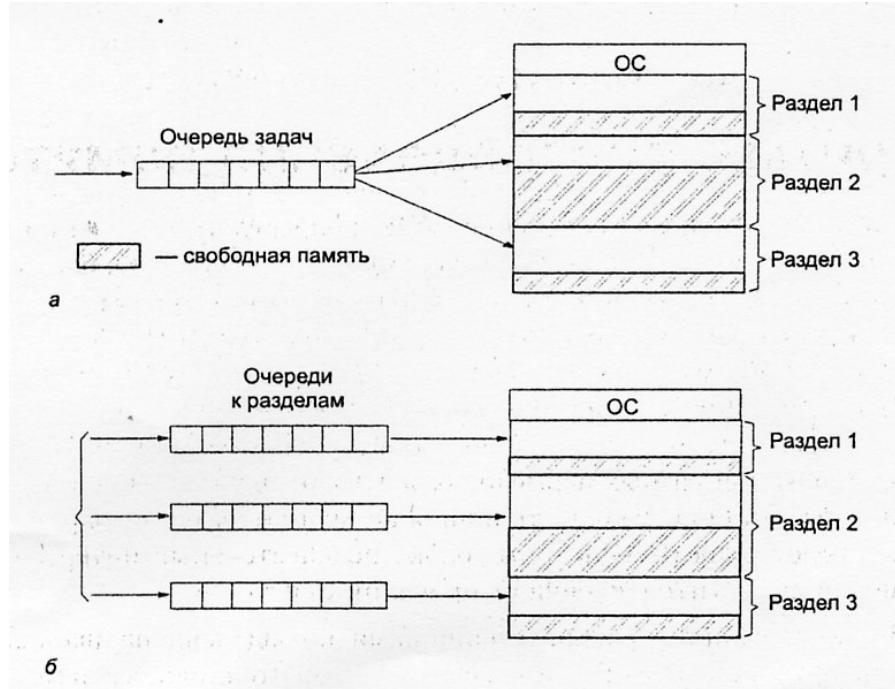
Распределение памяти фиксированными разделами

Подсистема управления памятью выполняет следующие задачи

Сравнивает объем памяти, требуемый для вновь поступившего процесса, с размерами свободных разделов и выбирает подходящий раздел.

Осуществляет загрузку программы в один из разделов и настройку адресов. Уже на этапе трансляции разработчик программы может задать раздел, в котором ее следует выполнять. Это позволяет сразу, без использования перемещающего загрузчика, получить машинный код, настроенный на конкретную область памяти.

Распределение памяти фиксированными разделами



С общей очередью

С отдельными очередями

Распределение памяти динамическими разделами

Использовалось в ОС IBM OS/360

Вопросы:

Каковы достоинства и недостатки распределения памяти фиксированными разделами?

В каких случаях и сегодня может использоваться распределение памяти фиксированными разделами?

Распределение памяти динамическими разделами

Функции ОС в данном методе управления памятью:

- а) ведение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти.
- б) при создании нового процесса – анализ требований к памяти, просмотр таблицы свободных областей и выбор раздела, размер которого достаточен для размещения кодов и данных нового процесса.
- в) загрузка программы в выделенный ей раздел и корректировка таблиц свободных и занятых областей. Данный способ предполагает, что программный код не перемещается во время выполнения, а значит, настройка адресов может быть проведена единовременно во время загрузки.
- г) после завершения процесса корректировка таблиц свободных областей.

Распределение памяти динамическими разделами

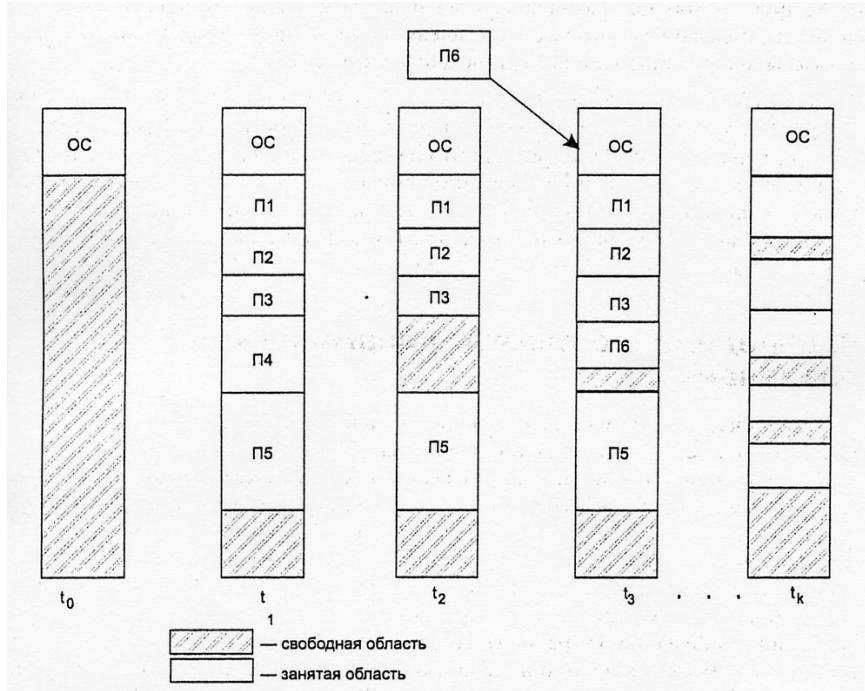
Правила выбора раздела:

«первый попавшийся раздел
достаточного размера»,

«раздел, имеющий наименьший
достаточный размер»

«раздел, имеющий наибольший
достаточный размер».

Распределение памяти динамическими разделами



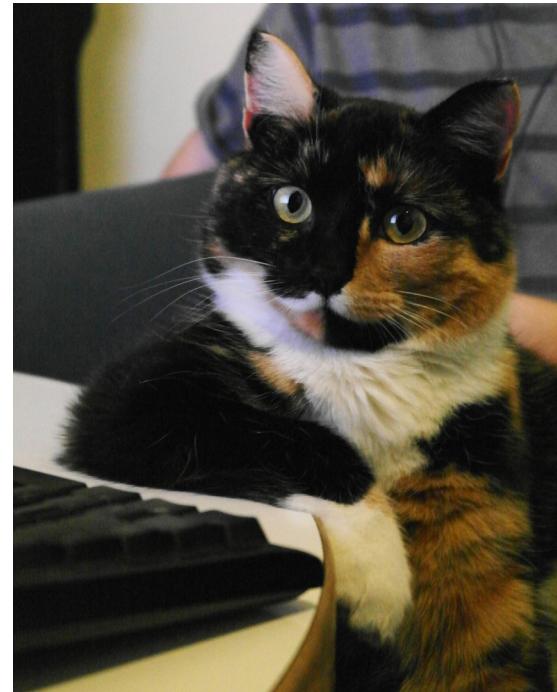
Вопросы:

Какие проблемы могут возникнуть при использовании данного подхода?

Вопросы

Приведите достоинства и недостатки каждого метода распределения памяти.

Как бы подходы вы могли предложить для решения возникающих проблем?



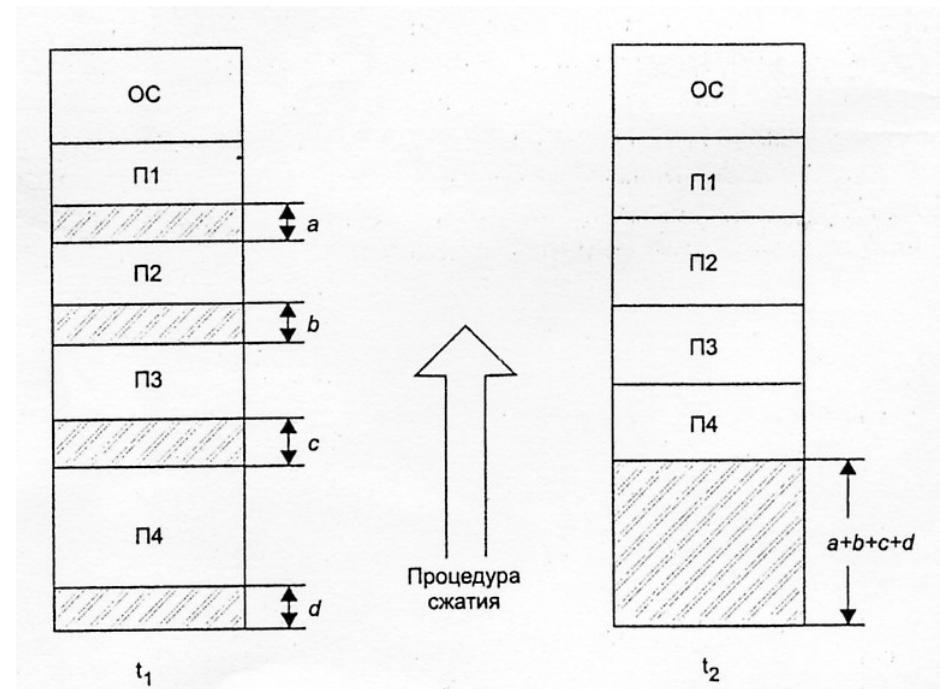
Вопросы

Варианты:

Перемещаемые разделы

Динамическое преобразование
адресов

Сжатие данных

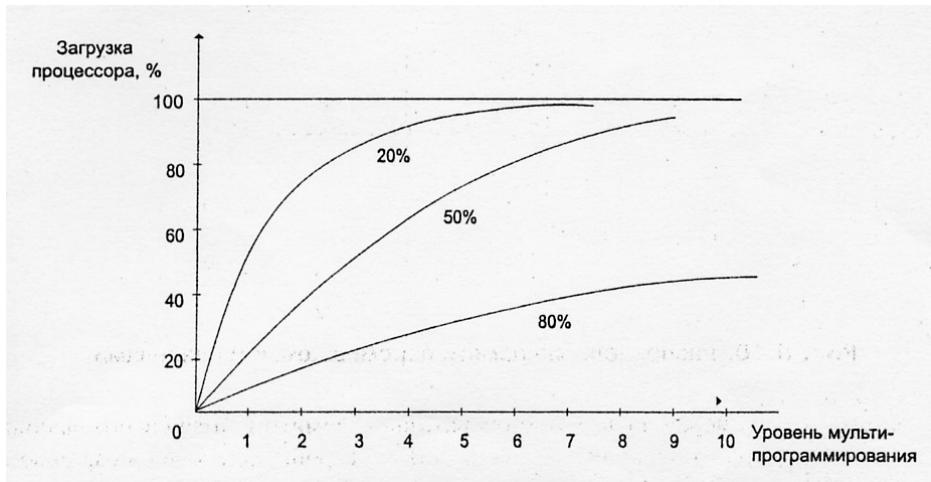


Модуль 2. Основные принципы построения операционных систем

Лекция 09 (2.5)

Методы распределения с использованием дискового
пространства. Понятие виртуальной памяти

Виртуальная память



Зависимость загрузки процессора от числа задач и интенсивности ввода-вывода

Виртуальная память

Задачи виртуальной памяти:

- Размещение данных в запоминающих устройствах различного типа, например часть кодов программы – в оперативной памяти, а часть – на диске;
- Выбор образов процессов или их частей для перемещения из оперативной памяти на диск и обратно;
- Перемещение по мере необходимости данных между памятью и диском;
- Преобразование виртуальных адресов в физические.

Вопросы:

Как осуществляется хранение кода/данных из ОП на жестком диске?

Что такое свопинг?

Что такое файл подкачки?

Классификация виртуальной памяти

- Страницчная виртуальная память организует перемещение данных между памятью и диском страницами частями виртуального адресного пространства, фиксированного и сравнительно небольшого размера.
- Сегментная виртуальная память предусматривает перемещение данных сегментами частями виртуального адресного пространства произвольного размера, полученными с учетом смыслового значения данных.
- Сегментно-страницчная виртуальная память используется двухуровневое деление: виртуальное адресное пространство делится на сегменты, а затем сегменты делятся на страницы. Единицей перемещения данных здесь является страница. Этот способ управления памятью объединяет себе элементы обоих предыдущих подходов.

Классификация виртуальной памяти

Виртуализация памяти может быть осуществлена на двух различных подходов:

- свопинг (swapping)-образы процессов выгружаются на диск и возвращаются в оперативную память целиком;
- виртуальная память(virtual memory)-между оперативной памятью и диском перемещаются части (сегменты, страницы и т.п.) образов процессов.

Структура swap-файла <linux/swap.h>

```
64 struct swap_info_struct {  
65     unsigned int flags;  
66     kdev_t swap_device;  
67     spinlock_t sdev_lock;  
68     struct dentry * swap_file;  
69     struct vfsmount *swap_vfsmnt;  
70     unsigned short * swap_map;  
71     unsigned int lowest_bit;  
72     unsigned int highest_bit;  
73     unsigned int cluster_next;  
74     unsigned int cluster_nr;  
75     int prio;  
76     int pages;  
77     unsigned long max;  
78     int next;  
79 };
```

<https://www.kernel.org/doc/gorman/html/understand/understand014.html>

flags This is a bit field with two possible values. SWP_USED is set if the swap area is currently active. SWP_WRITEOK is defined as 3, the two lowest significant bits, including the SWP_USED bit. The flags is set to SWP_WRITEOK when Linux is ready to write to the area as it must be active to be written to;

swap_device The device corresponding to the partition used for this swap area is stored here. If the swap area is a file, this is NULL;
sdev_lock As with many structs in Linux, this one has to be protected too. sdev_lock is a spinlock protecting the struct, principally the swap_map. It is locked and unlocked with swap_device_lock() and swap_device_unlock();

swap_file This is the dentry for the actual special file that is mounted as a swap area. This could be the dentry for a file in the /dev/ directory for example in the case a partition is mounted. This field is needed to identify the correct swap_info_struct when deactivating a swap area;

vfsmount This is the vfs_mount object corresponding to where the device or file for this swap area is stored;

swap_map This is a large array with one entry for every swap entry, or page sized slot in the area. An entry is a reference count of the number of users of this page slot. The swap cache counts as one user and every PTE that has been paged out to the slot counts as a user. If it is equal to SWAP_MAP_MAX, the slot is allocated permanently. If equal to SWAP_MAP_BAD, the slot will never be used;

lowest_bit This is the lowest possible free slot available in the swap area and is used to start from when linearly scanning to reduce the search space. It is known that there are definitely no free slots below this mark;

highest_bit This is the highest possible free slot available in this swap area. Similar to lowest_bit, there are definitely no free slots above this mark;
cluster_next This is the offset of the next cluster of blocks to use. The swap area tries to have pages allocated in cluster blocks to increase the chance related pages will be stored together;

cluster_nr This the number of pages left to allocate in this cluster;

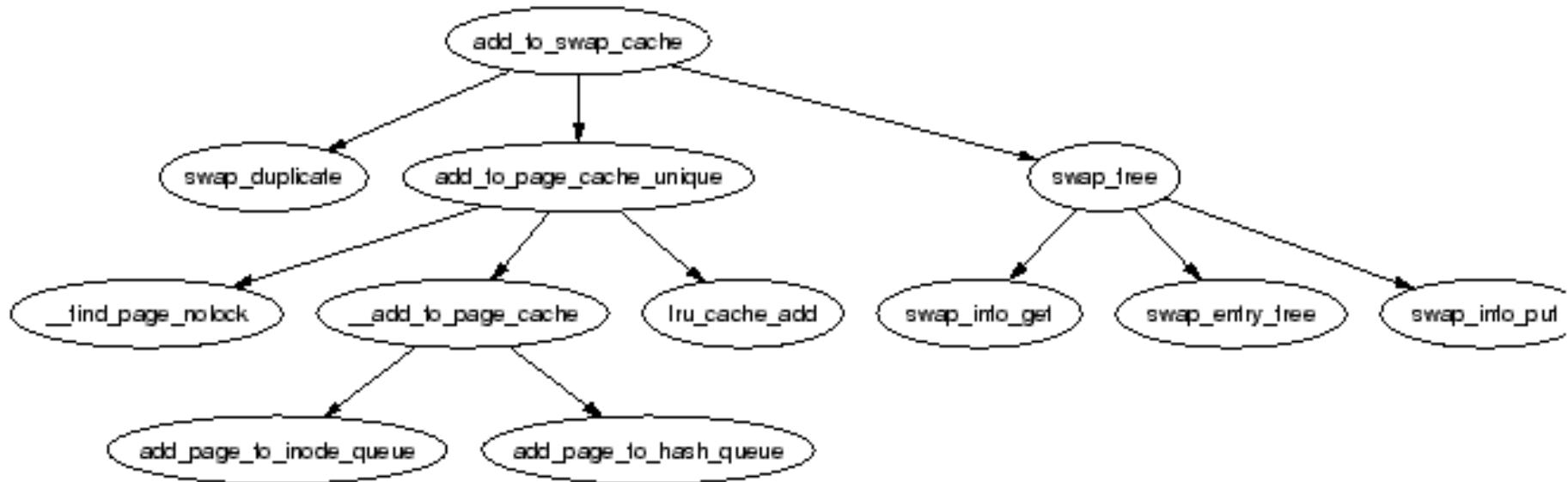
prio Each swap area has a priority which is stored in this field. Areas are arranged in order of priority and determine how likely the area is to be used. By default the priorities are arranged in order of activation but the system administrator may also specify it using the -p flag when using swapon;

pages As some slots on the swap file may be unusable, this field stores the number of usable pages in the swap area. This differs from max in that slots marked SWAP_MAP_BAD are not counted;

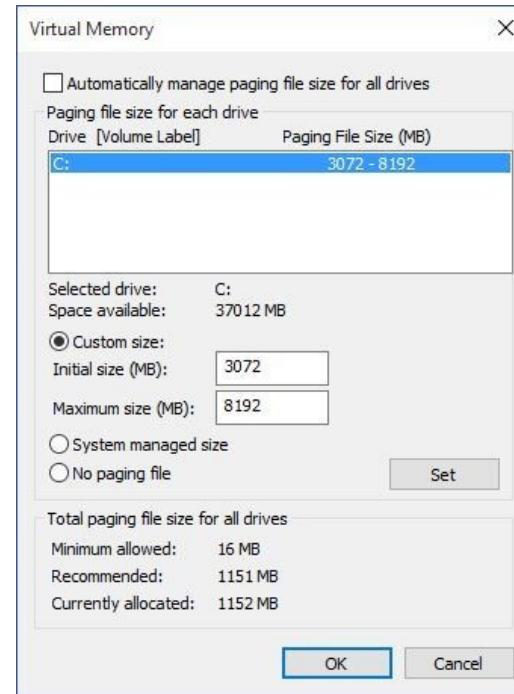
max This is the total number of slots in this swap area;

next This is the index in the swap_info array of the next swap area in the system.

Граф вызова swap-кэша



Файл подкачки Windows



Методы распределения с использованием дискового пространства

Страницное распределение памяти

Сегментное распределение памяти

Страницно сегментное распределение памяти

Вопросы

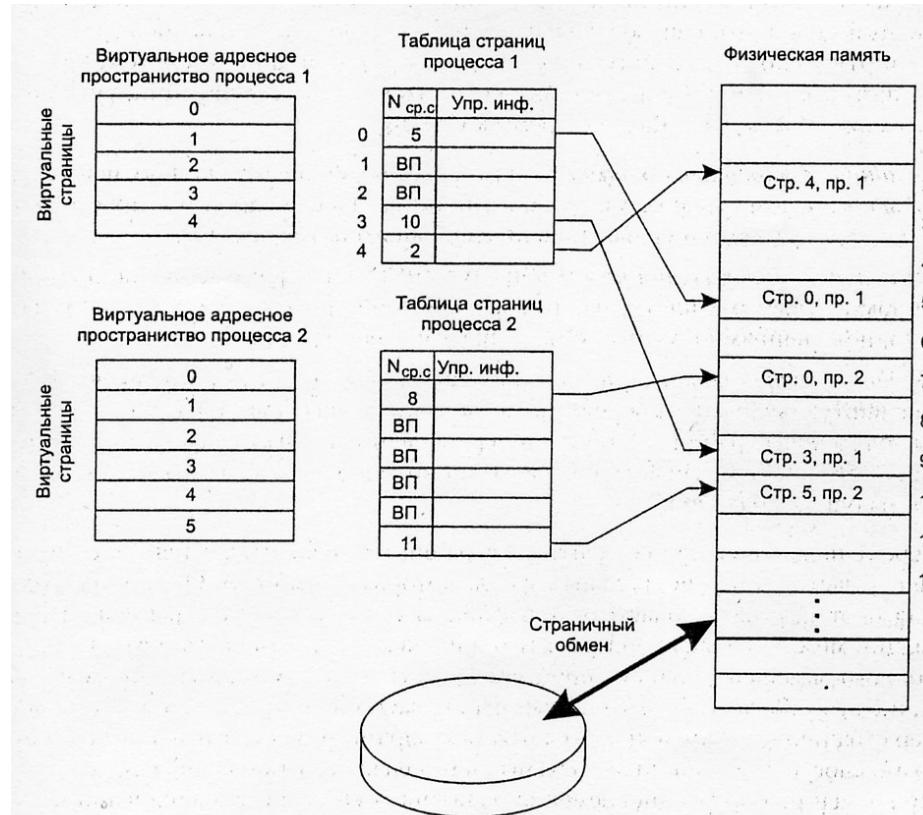
Как можно улучшить работу с фалом подкачки?

Модуль 2. Основные принципы построения операционных систем

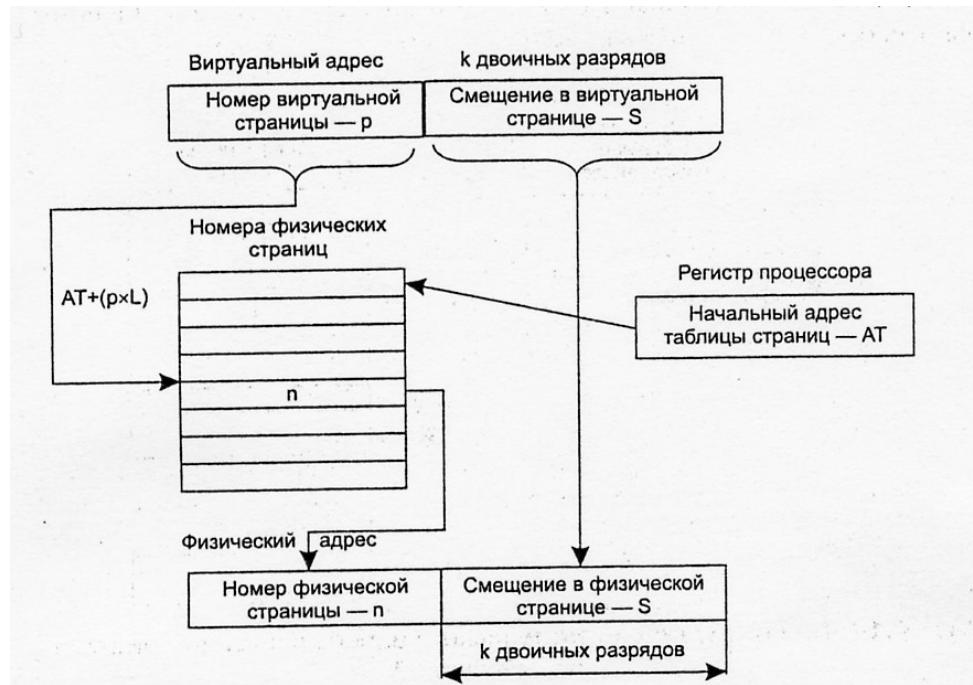
Лекция 10 (2.6)

Страницное, сегментное, страницно-сегментное
распределение. Свопинг. Средства вызова подпрограмм и задач

Страницное распределение



Преобразования виртуального адреса в физический при страничной организации памяти

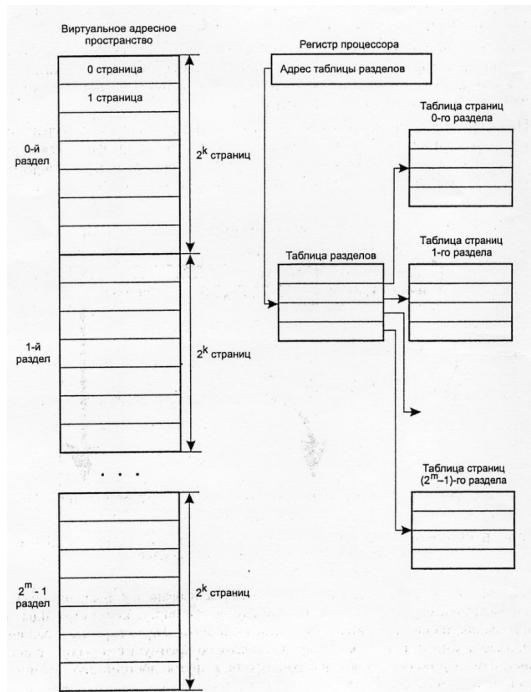


Критерии выбора страниц на выгрузку

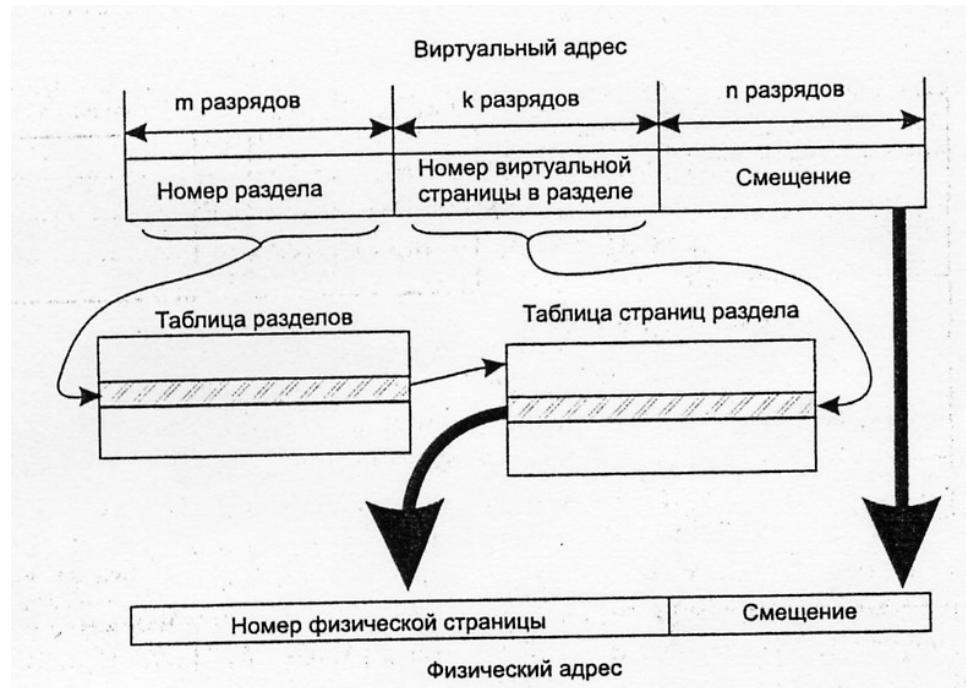
При выборе страницы на выгрузку могут быть использованы различные критерии:

- на диск выталкивается страница, к которой в будущем, начиная с данного момента, больше не будет обращений.
- случайный выбор страницы
- анализируется число обращений к странице за последний период времени

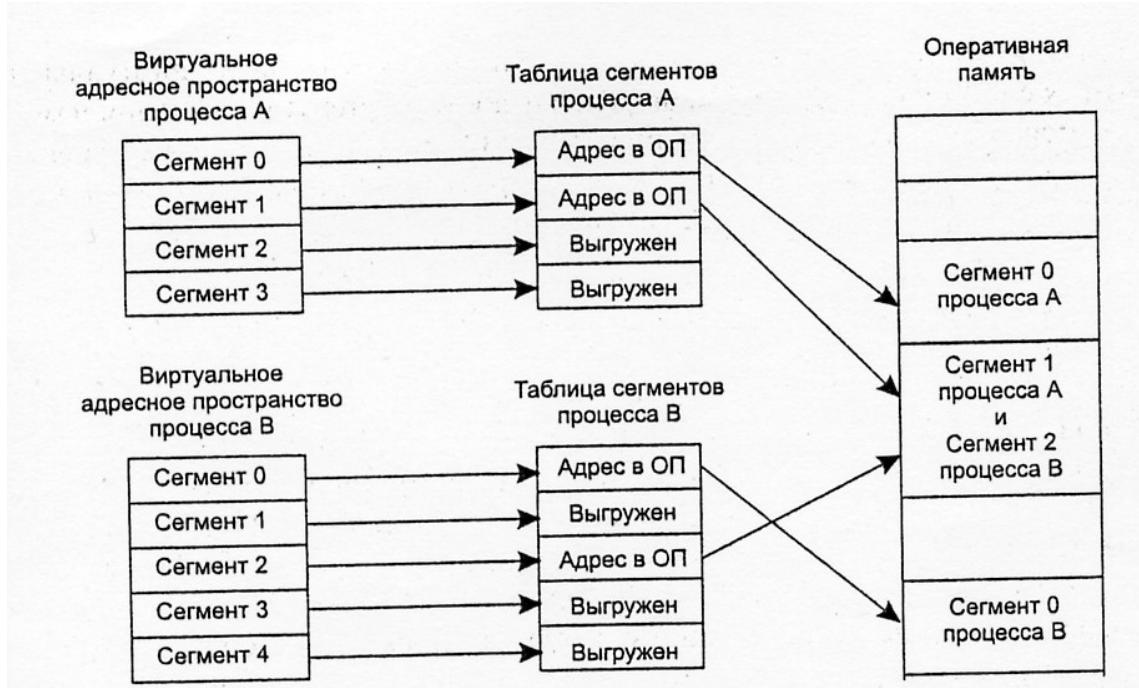
Сегменты+разделы



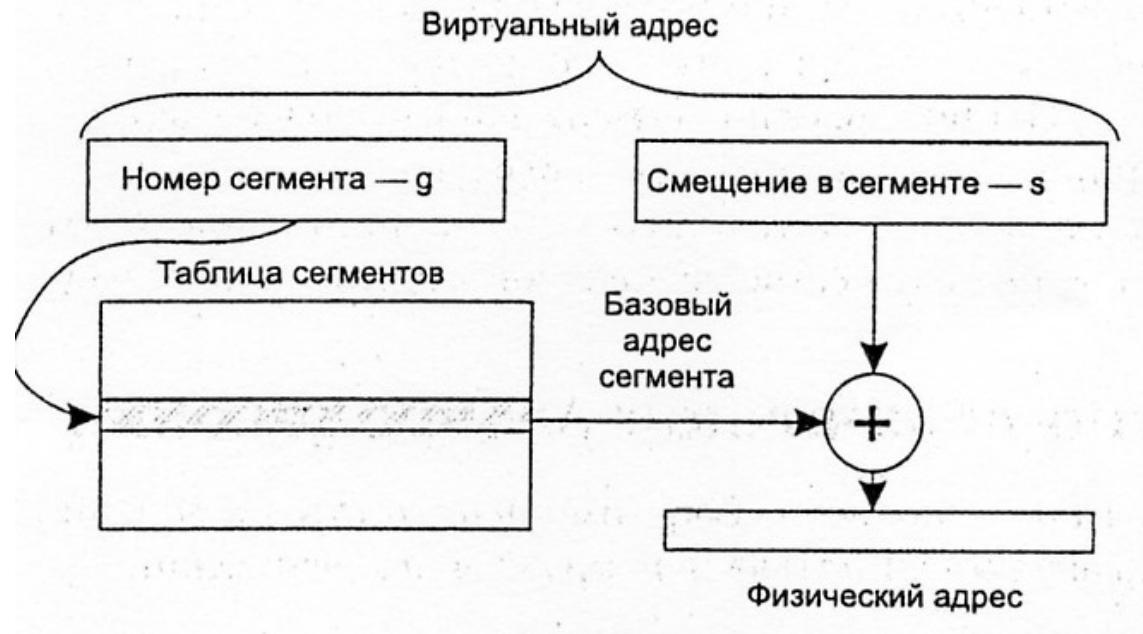
Сегменты+разделы (адресация)



Сегментное распределение



Преобразование виртуального адреса при сегментной организации



Вопросы

Назовите достоинства и недостатки каждого метода распределения памяти?

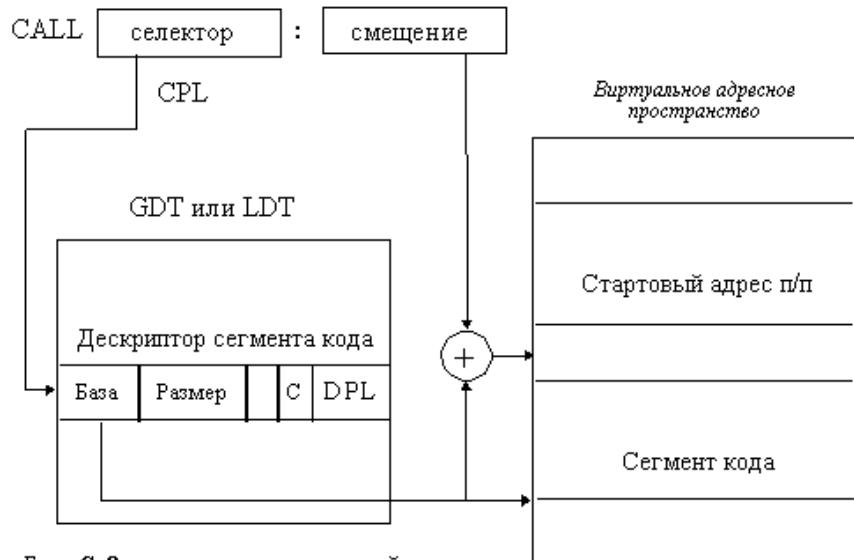
Как может быть организована сегментно-страничная организация памяти?

Как формируется адрес при сегментно-страничной организации?

Изобразите схему сегментно-страничной организации распределения памяти

Средства вызова подпрограмм и задач

Непосредственный вызов



Если С=0, то сегмент неподчиненный, доступ возможен при CPL ≤ DPL

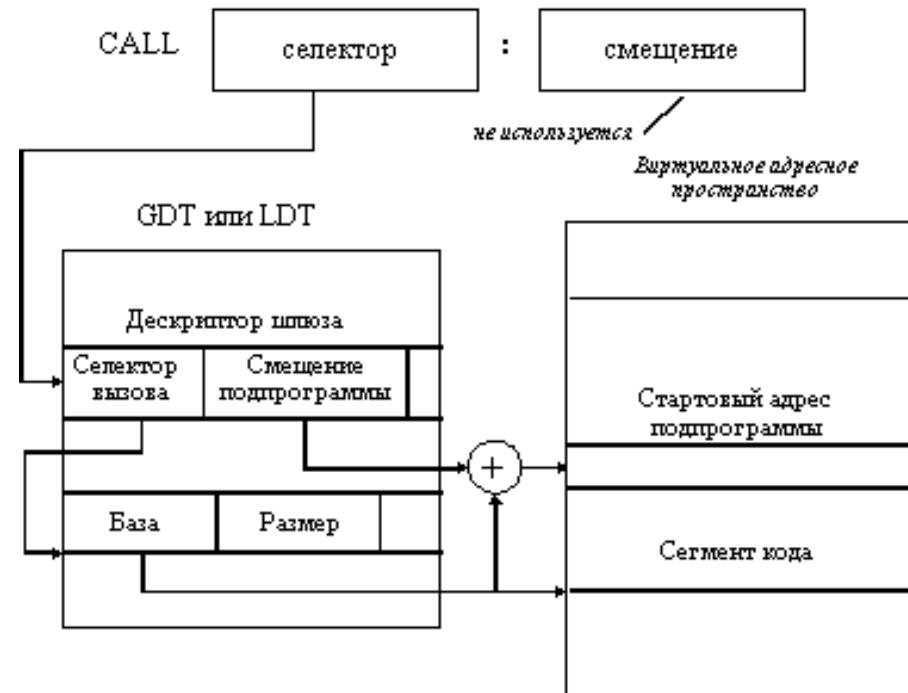
Если С=1, то сегмент подчиненный и доступ возможен всегда, но DPL заменяется на CPL при работе подпрограммы

Средства вызова подпрограмм и задач

Вызов подпрограммы через шлюз вызова

Формат дескриптора шлюза вызова подпрограммы

Селектор 0..15	Смещение 0..15					
Смещение 16..32	P	DPL	S=0	C (386) или 4 (286)	0..4 Счетчик слов (WC)	



Модуль 2. Основные принципы построения операционных систем

Лекция 11 (2.7)

Файловая система. Логическая организация файла. Физическая организация файлов. Права доступа к файлу

Подсистема ввода-вывода

Функция подсистемы ввода-вывода

Одна из главных задач ОС – обеспечение обмена данными между приложениями и ПУ компьютера

Подсистема ввода-вывода

Клиенты ПВВ:

- пользователи;
- приложения;
- некоторые компоненты самой ОС

Основные компоненты ПВВ:

- драйверы
- файловая система
- диспетчер прерываний

Решаемые ПВВ задачи

- 1) организация параллельной работы УВВ и ЦП
- 2) согласование скоростей обмена и кэширование данных
- 3) разделение устройств и данных между процессами
- 4) обеспечение удобного лог. ИФ между устройствами и остальными частями ОС
- 5) поддержка широкого спектра драйверов
- 6) поддержка нескольких ФС
- 7) динамическая загрузка и выгрузка драйверов
- 8) поддержка синхронных и асинхронных операций ВвВыв

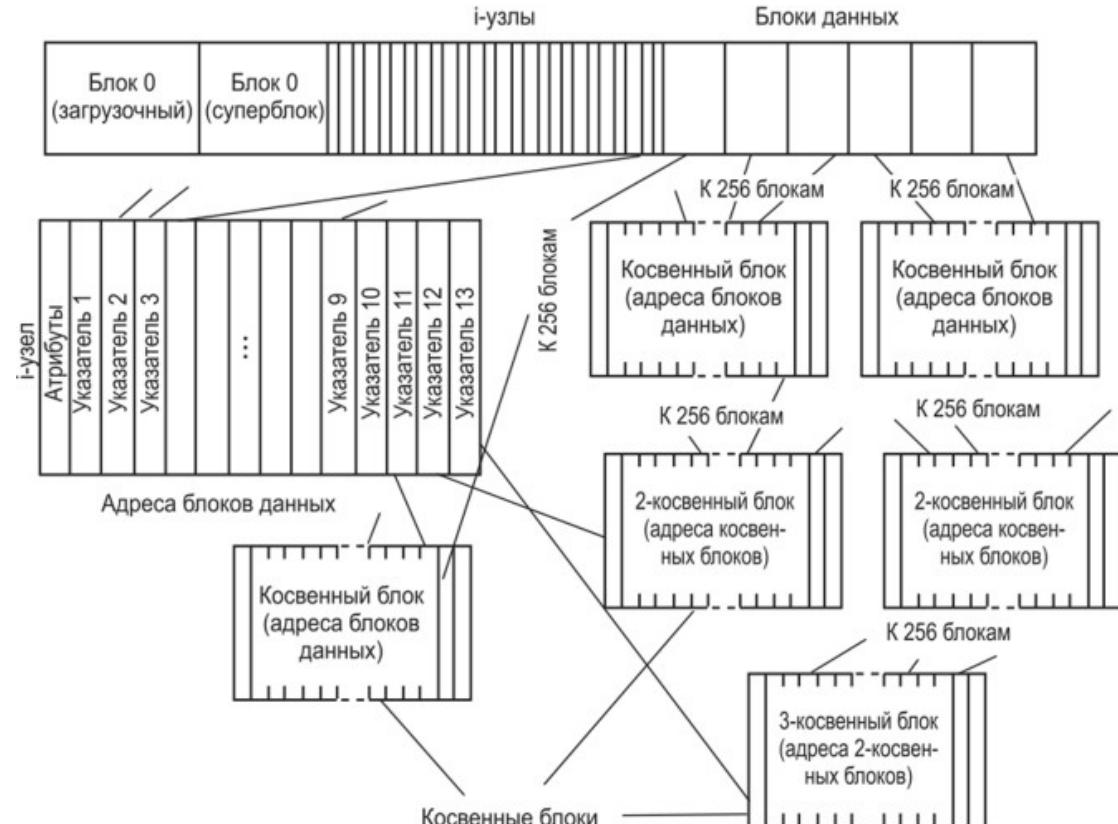
Организация файла

Иерархическая структура

Файл-устройство

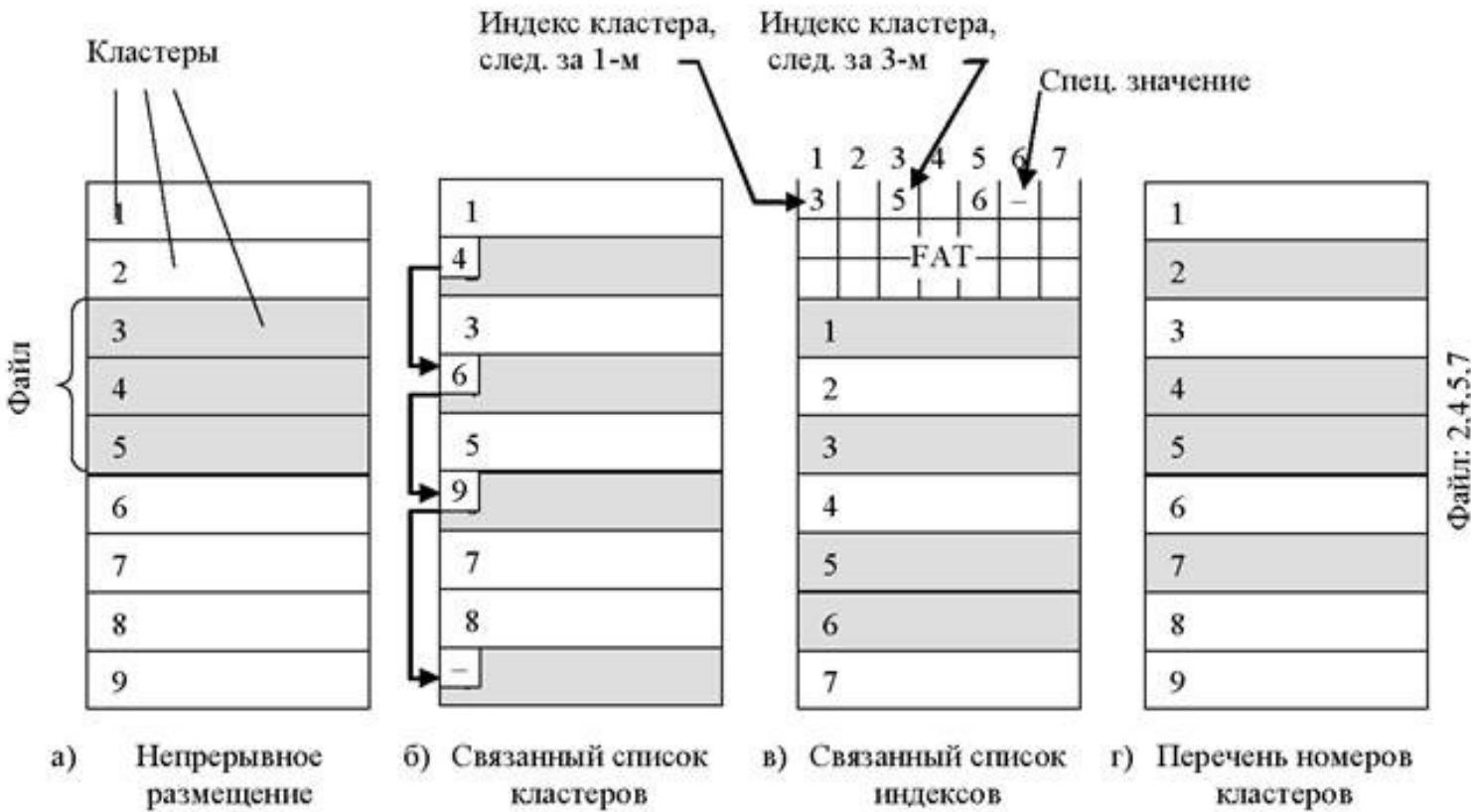
Концепция *NIX:

- всё есть файл
- имя – не атрибут файла
- атрибуты файла
- ссылки



Физическая структура файла

Что такое кластер



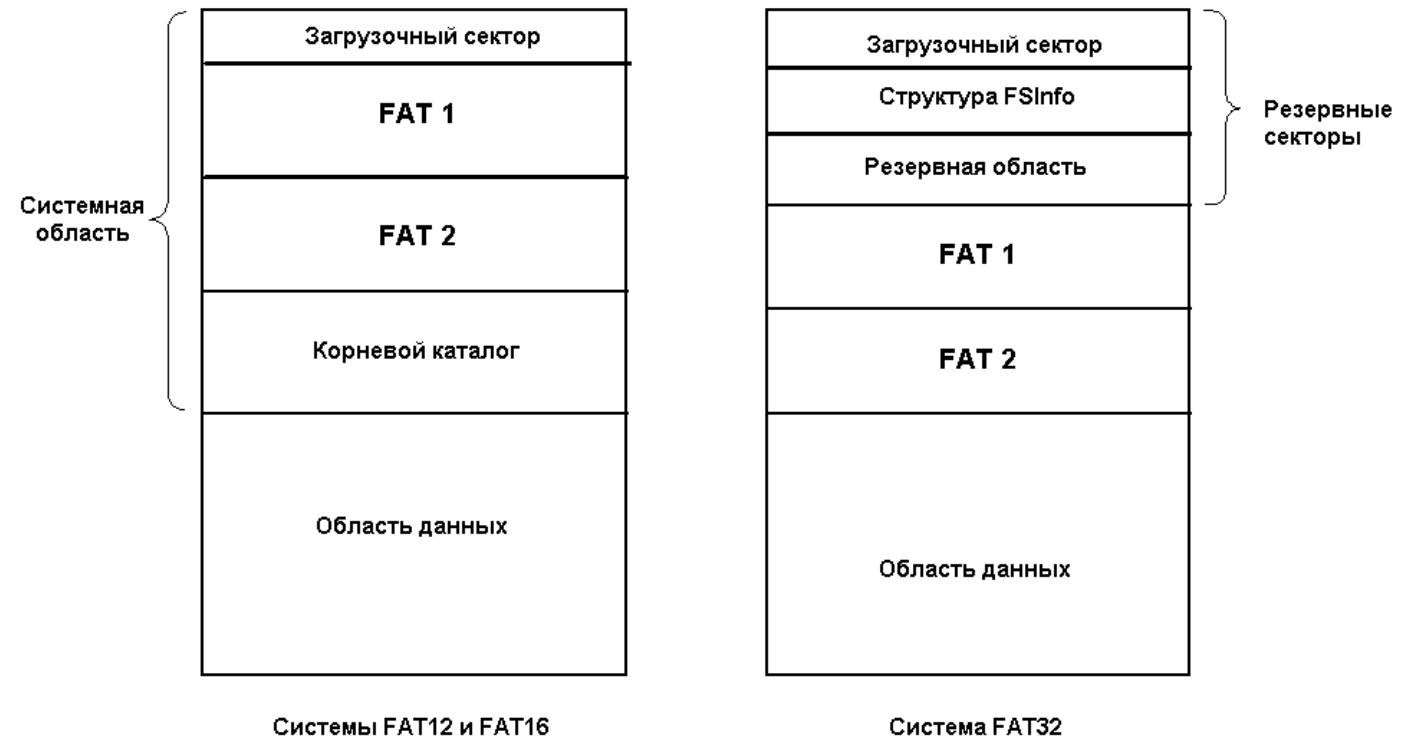
Файловая система FAT

Наследница ФС CP\М

Где используется:

DOS (MS DOS, DR-DOS)

Windows (Chicago)



Системы FAT12 и FAT16

Система FAT32

Файловая система NTFS

Windows NT

Надежность:

- MFT
- дублирование элементов
- шифрование

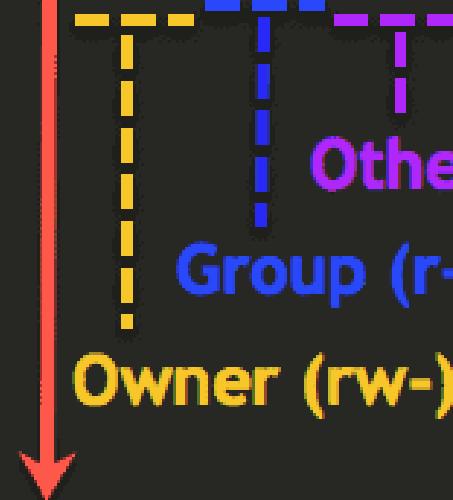


Права доступа к файлу

Mode	File Size			Last Modified	Filename
	Owner	Group			
drwxrwxrwx 2 sammy sammy	4096	Nov 10 12:15			everyone_directory
drwxrwx--- 2 root developers	4096	Nov 10 12:15			group_directory
-rw-rw---- 1 sammy sammy	15	Nov 10 17:07			group_modifiable
drwx----- 2 sammy sammy	4096	Nov 10 12:15			private_directory
-rw------- 1 sammy sammy	269	Nov 10 16:57			private_file
-rwxr-xr-x 1 sammy sammy	46357	Nov 10 17:07			public_executable
-rw-rw-rw- 1 sammy sammy	2697	Nov 10 17:06			public_file
drwxr-xr-x 2 sammy sammy	4096	Nov 10 16:49			publicly_accessible_directory
-rw-r--r-- 1 sammy sammy	7718	Nov 10 16:58			publicly_readable_file
drwx----- 2 root root	4096	Nov 10 17:05			root_private_directory

Схема прав доступа

```
# ls -l file  
-rw-r--r-- 1 root root 0 Nov 19 23:49 file
```



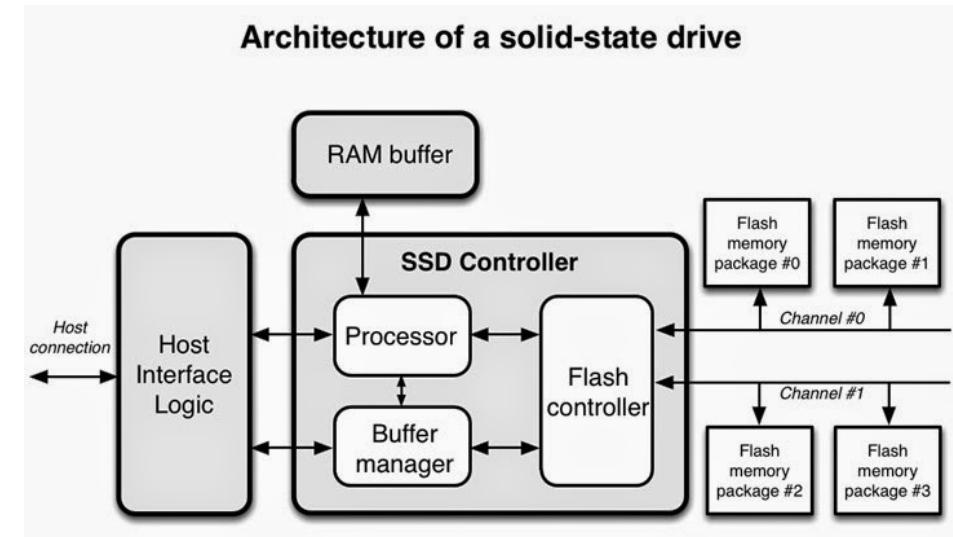
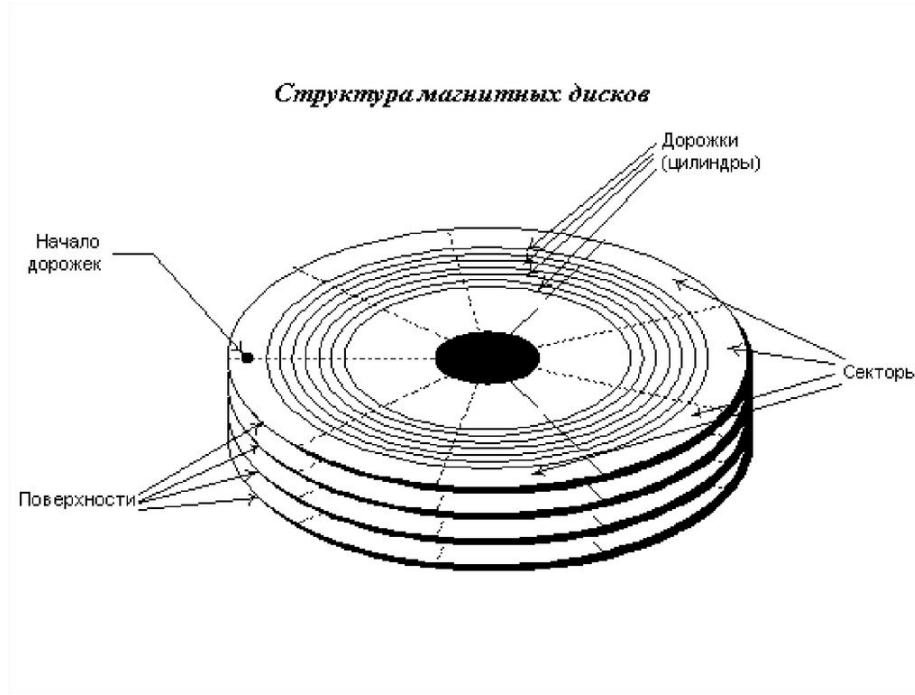
	r = Readable
	w = Writeable
	x = Executable
	- = Denied

Модуль 2. Основные принципы построения операционных систем

Лекция 12 (2.8)

Кеширование диска. Современные архитектуры файловых систем

Структура жесткого диска (HDD, SSD)



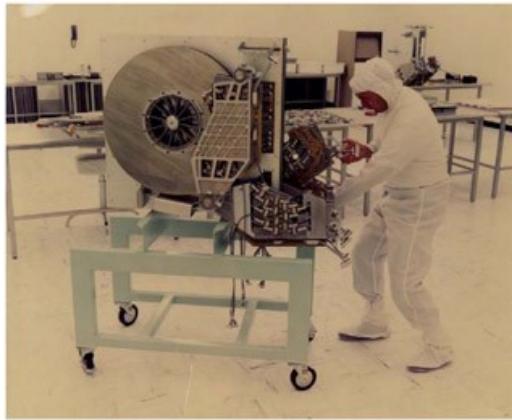
“Инопланетная технология”



Примеры жестких дисков



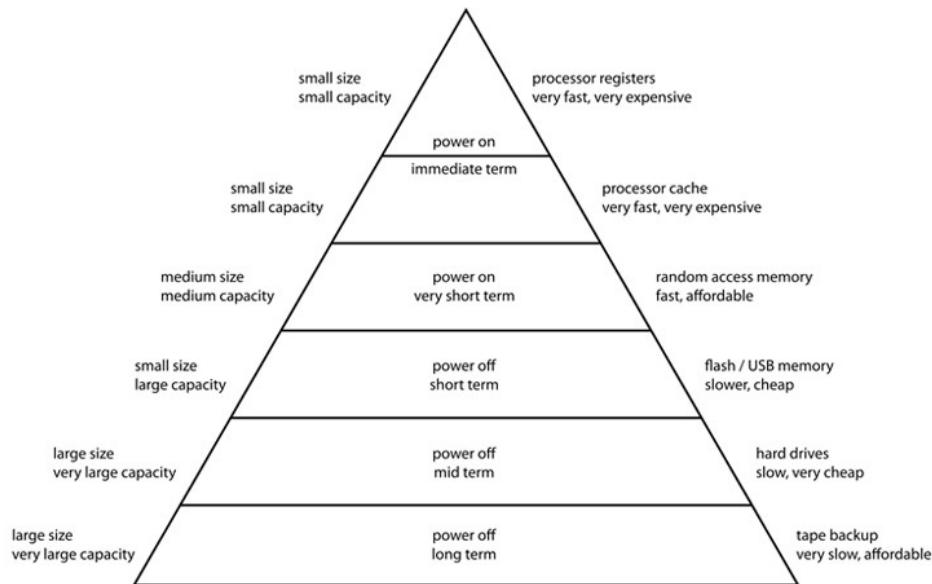
IBM Model 350 5MB Disk File. circa 1956



Circa 1975

Иерархия памяти

Computer Memory Hierarchy



Кэш-память позволяет
сбалансировать соотношение
цена/скорость/объем

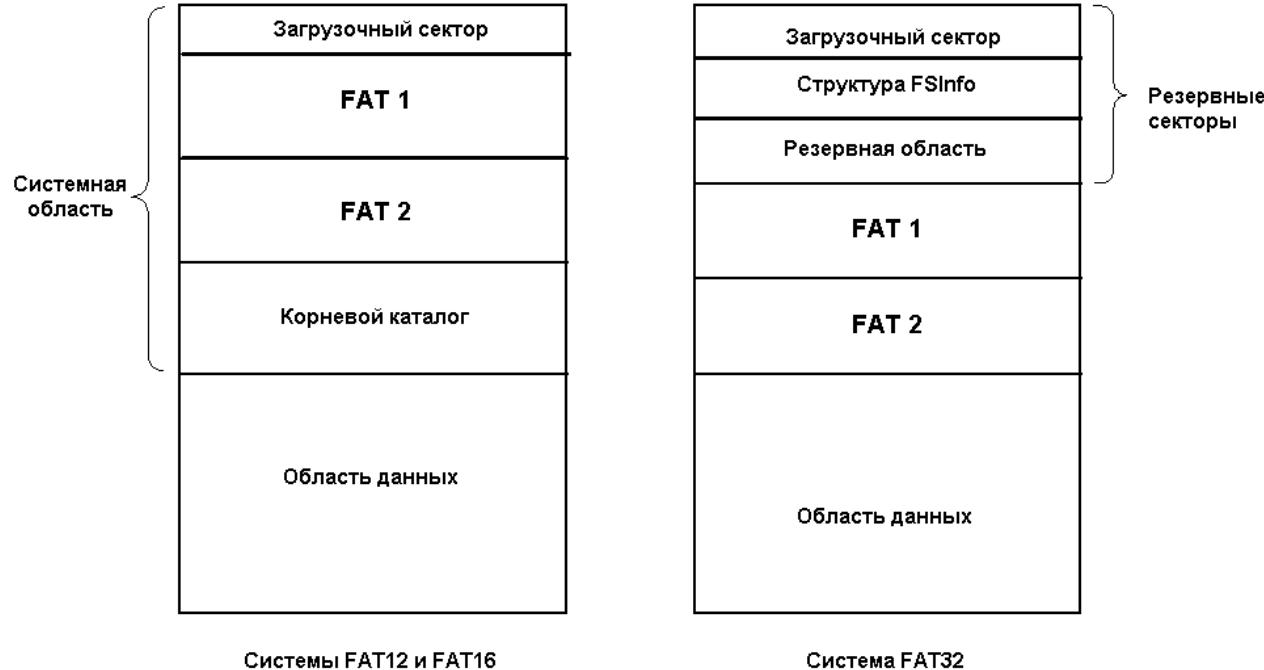
Разработка файловой системы

Задание:

Вы живете в 1958 году. Вам надо разработать простейшую файловую систему.

Какие последовательные инженерные решения вы будете выбирать для решения указанных задач?

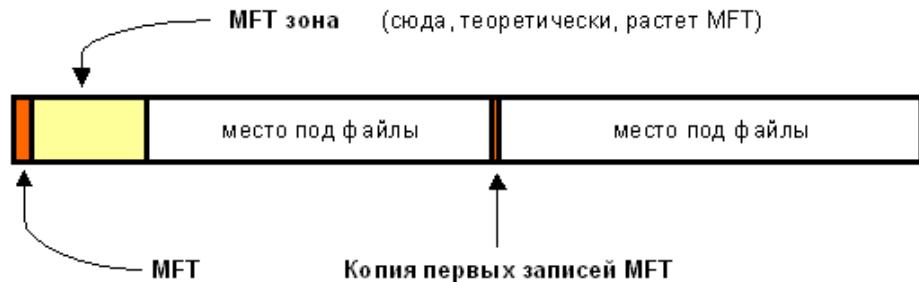
Файловая система DOS



Сравнение FAT16 и FAT32

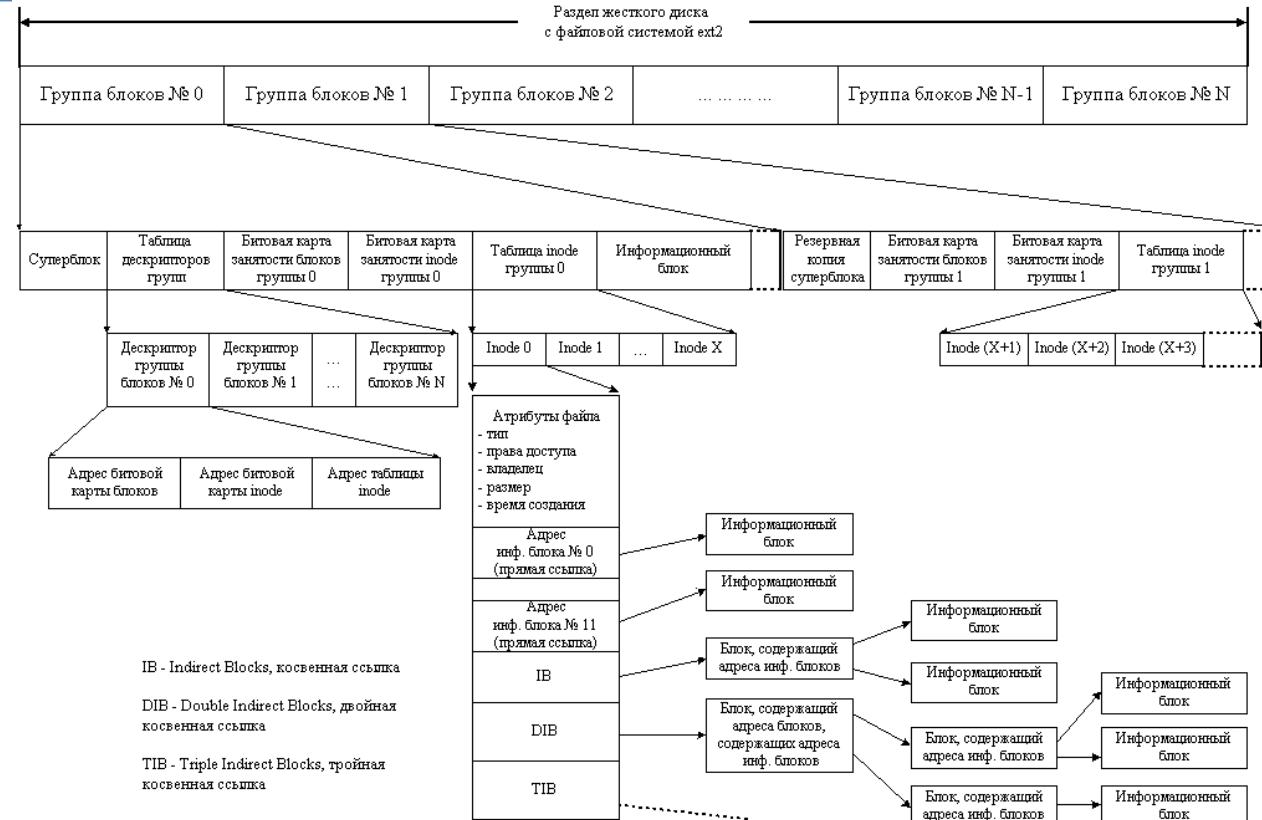
FAT 16		FAT 32	
	максимальный размер раздела – 2Гб (для Win2000 – 4Гб);		работает чуть медленнее, чем FAT16;
	при размере раздела > 512 Мб неэкономно расходует место на диске (из-за большого размера кластера);		разделы до 2Тб;
	распознается большинством ОС, используемыми на ПК;		большая эффективность использования места на диске;
	позволяет уплотнять диск программой сжатия данных Drivespace;		нельзя уплотнить с помощью программы сжатия данных;
	имеет корневой каталог фиксированного размера (512 записей).		старые версии DOS и многие другие ОС не "видят" разделы с форматом FAT 32;
			корневой каталог является обычным расширяемым каталогом.

Файловая система Windows NT

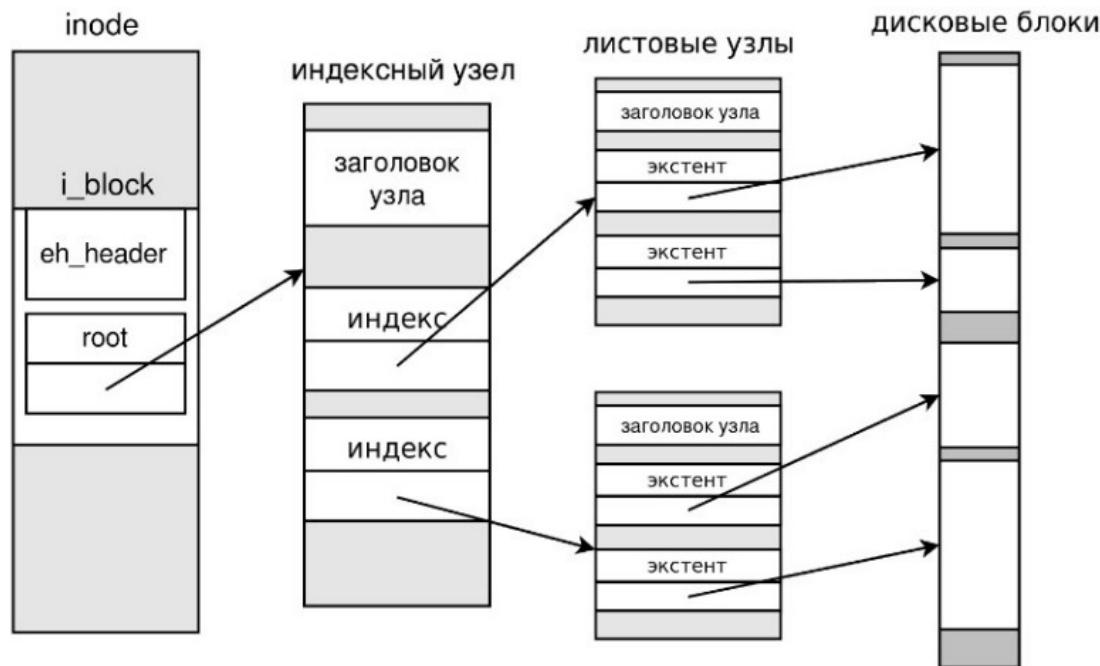


\$MFT	сам MFT
\$MFTmirr	копия первых 16 записей MFT, размещенная посередине диска
\$LogFile	файл поддержки журналирования (см. ниже)
\$Volume	служебная информация — метка тома, версия файловой системы, т. д.
\$AttrDef	список стандартных атрибутов файлов на томе
\$.	корневой каталог
\$Bitmap	карта свободного места тома
\$Boot	загрузочный сектор (если раздел загрузочный)
\$Quota	файл, в котором записаны права пользователей на использование дискового пространства (начал работать лишь в NT5)
\$Upcase	файл — таблица соответствия заглавных и прописных букв в именах файлов на текущем томе. Нужен в основном потому, что в NTFS имена файлов записываются в Unicode, что составляет 65 тысяч различных символов, искать большие и малые эквиваленты которых очень нетривиально.

Файловая система Linux - ext2



Журналируемая файловая система



Модуль 2. Основные принципы построения операционных систем

Лекция 13 (2.9)

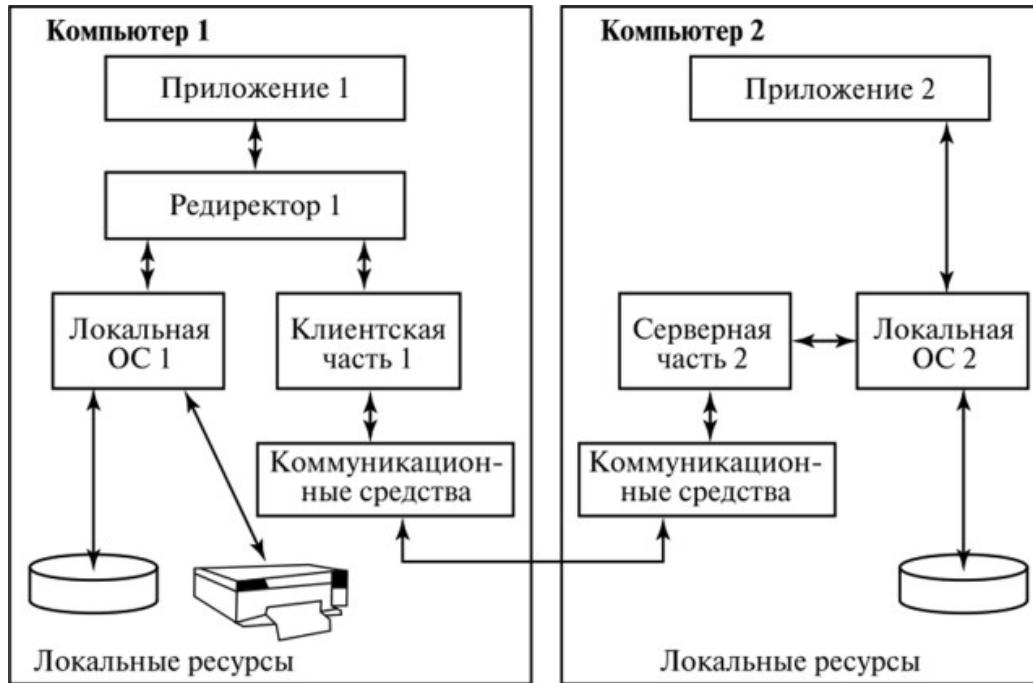
Управление распределенными ресурсами. Концепция
удаленного доступа

Проблемы взаимодействия операционных систем в гетерогенных сетях

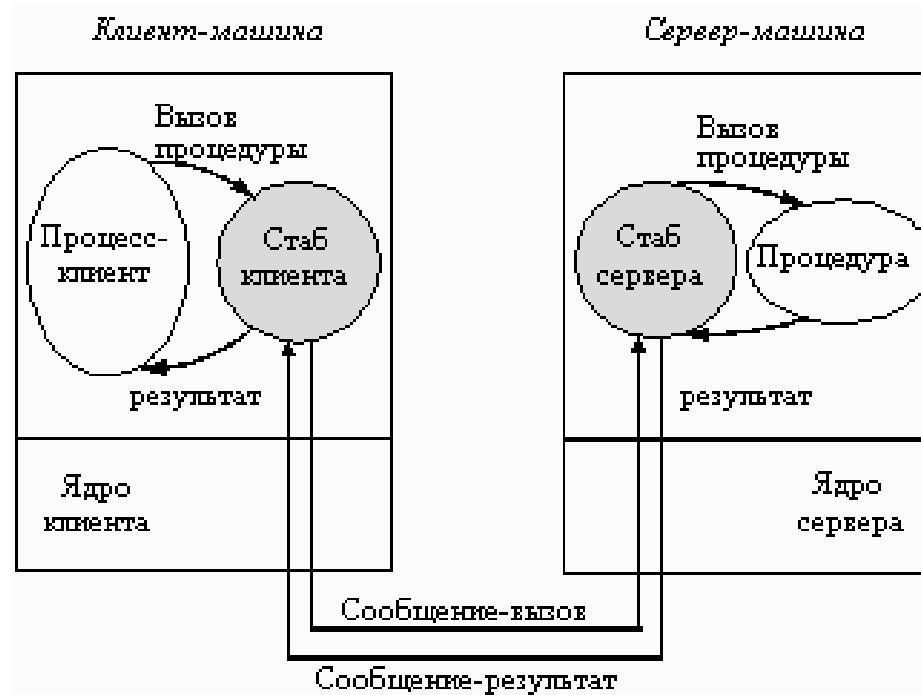
В сетевой ОС отдельной машины можно выделить:

- Средства управления локальными ресурсами компьютера**
- Средства предоставления собственных ресурсов и услуг в общее пользование**
- Средства запроса доступа к удаленным ресурсам и услугам и их использования**
- Коммуникационные средства ОС**

Взаимодействие компонентов ОС при взаимодействии компьютеров



Пример: удаленный вызов процедур



Этапы выполнения процедуры RPC



Синхронизация (алгоритм Лампорта)

0	1	2	
0	0	0	
6	A	8	10
12		16	20
18		24	B
24		32	C
30		40	50
36		48	60
42		56	70
48	D	64	80
54		72	90
60		80	100

(a)

0	1	2	
0	0	0	
6	A	8	10
12		16	20
18		24	B
24		32	C
30		40	50
36		48	60
42		61	70
48	D	69	80
70		77	90
76		85	100

(б)

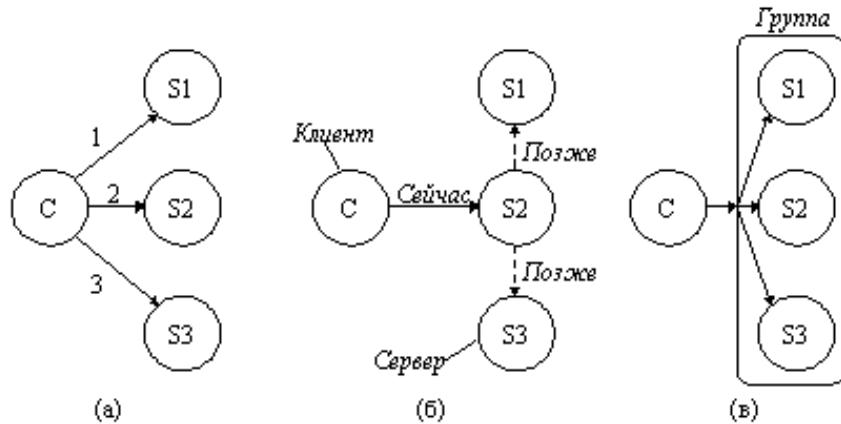
Реализация транзакций



Мультипротокольная транспортная сеть



Репликация



- а) Точная репликация файла;
- б) Ленивая репликация файла;
- в) Репликация файла, использующая группу

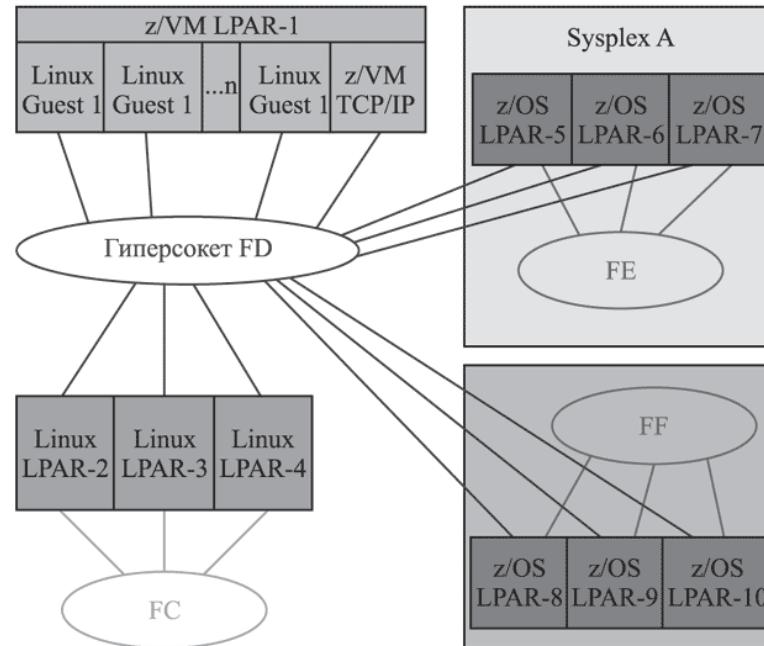
file.txt	1.14	2.16	3.19
prog.c	1.21	2.43	3.41

Символьное имя

Несколько двоичных
адресов (для S1, S2, S3)

Пример: Гиперсокеты в IBM zSeries

Сервер zSeries с несколькими LPAR и виртуальными серверами

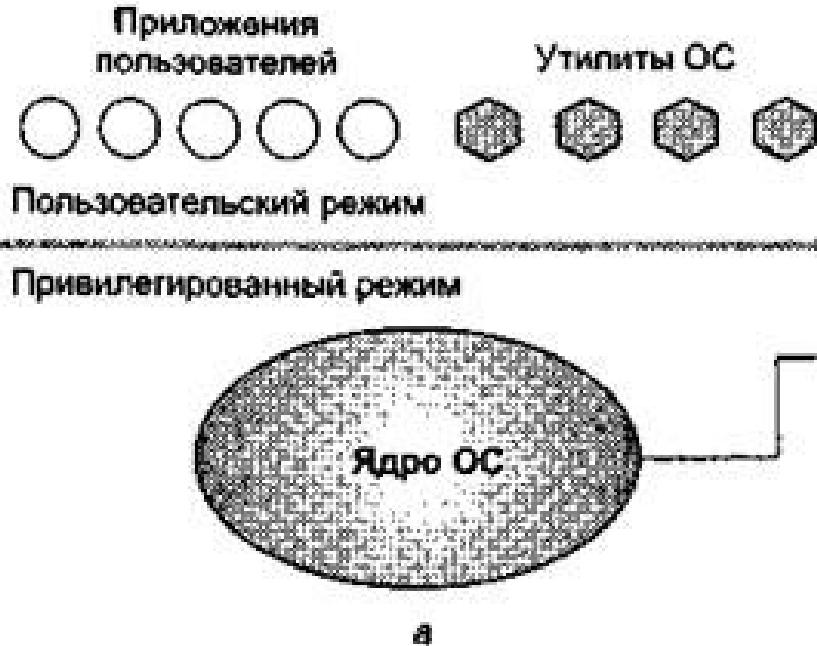


Модуль 3. Современные операционные системы различного назначения

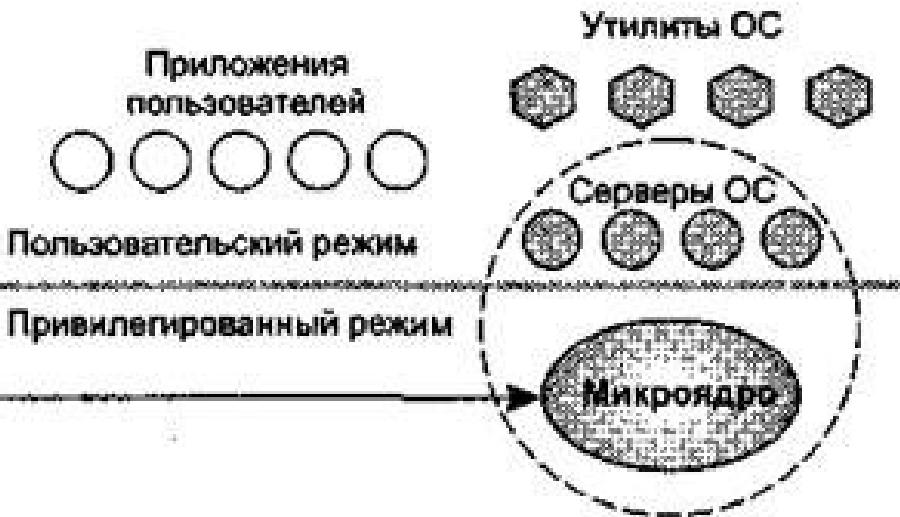
Лекция 14 (3.1)

Структура и особенности микроядерной ОС

Концепция микроядерной архитектуры



а



б

Преимущества и недостатки микроядерной архитектуры

Преимущества

Переносимость

Расширяемость

Конфигурируемость

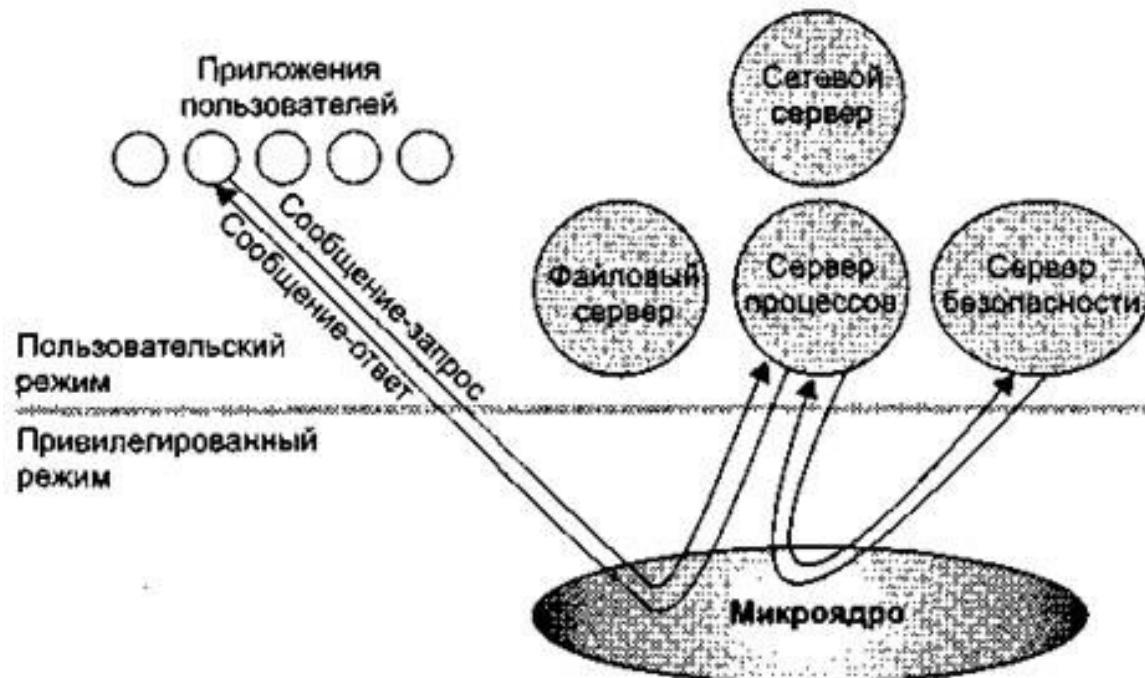
Надежность

Возможность поддержки
распределенных приложений

Недостатки

Снижение производительности

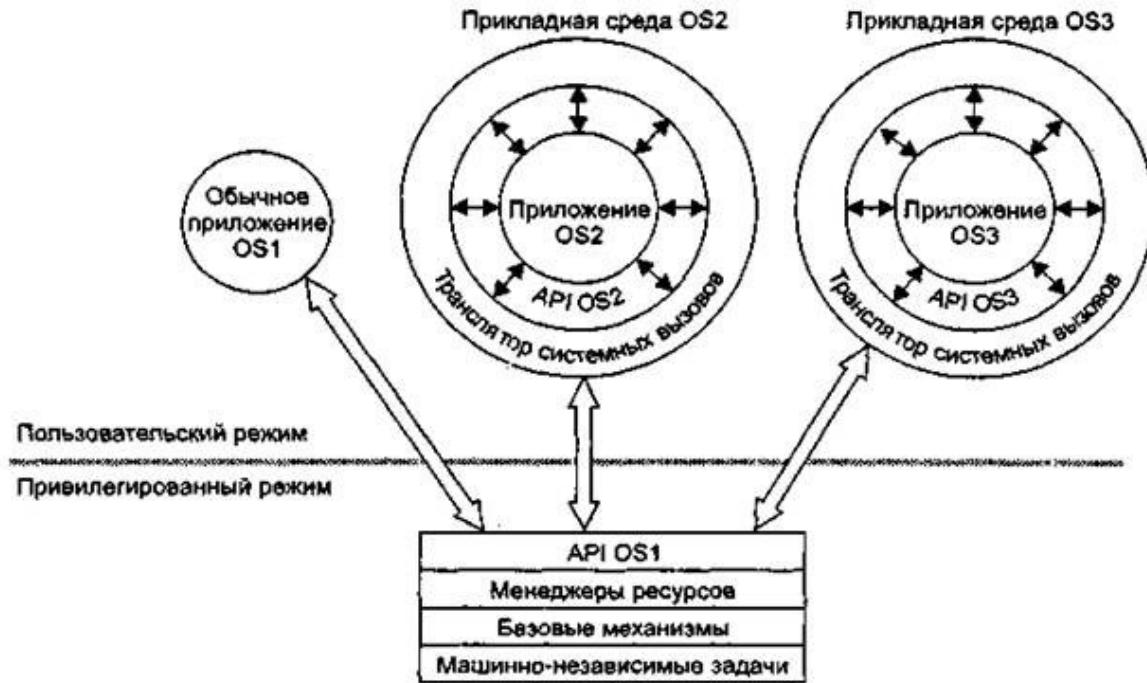
Реализация системного вызова в микроядерной архитектуре



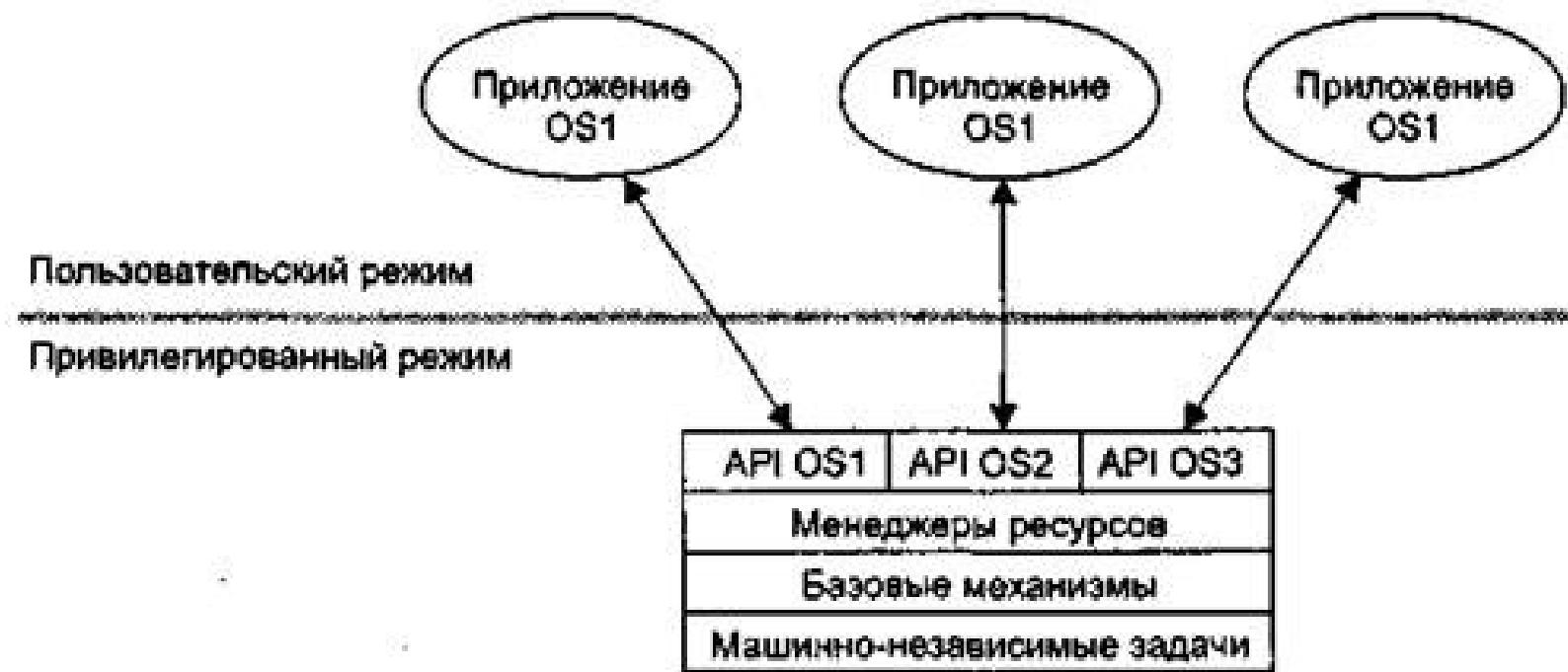
Выполнение системного вызова



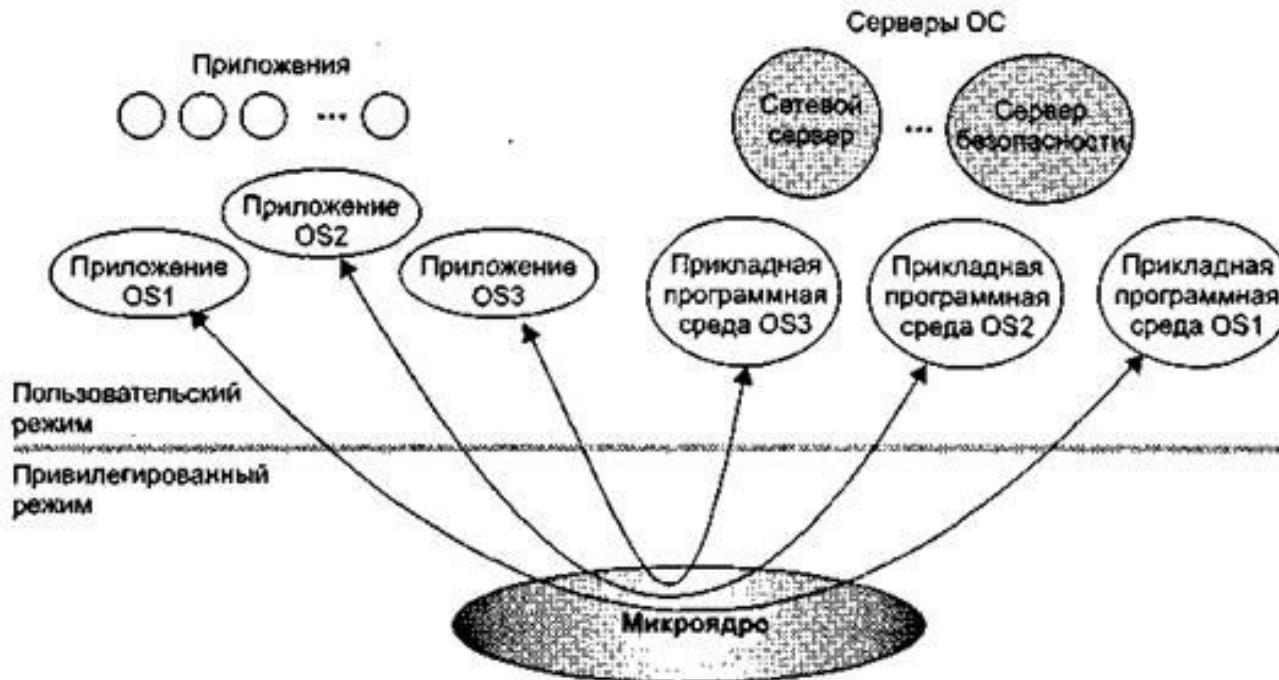
Прикладные программные среды, транслирующие системные вызовы



Реализация совместимости на основе нескольких равноправных API



Микроядерный подход к реализации множественных прикладных сред



Поколения микроядер

Первое поколение:

- Mach (Университет Карнеги-Меллона)
- ChorusOS (INRIA)

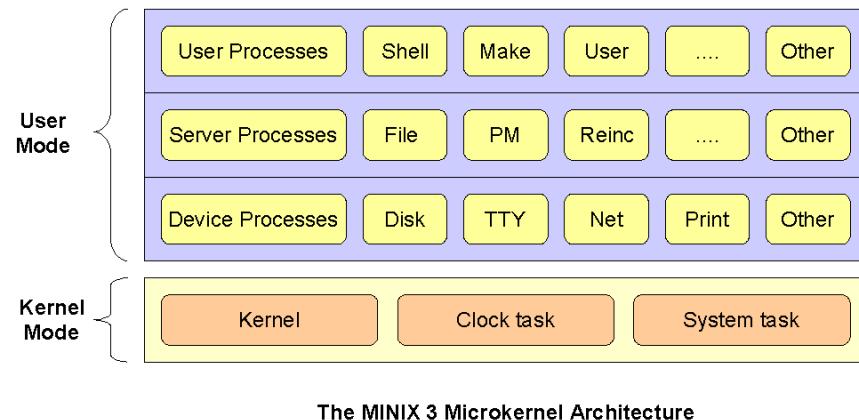
Второе поколение

- Minix
- L3 и L4

Третье поколение

- seL4
- Coyotos
- NOVA

Пример: MINIX 3



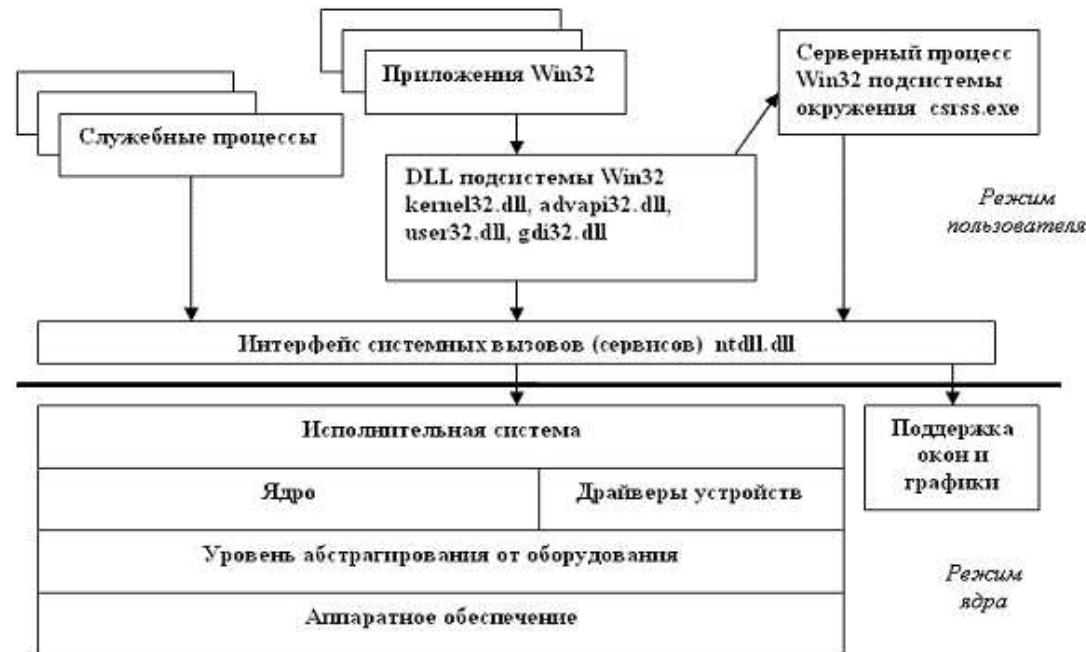
Разработка:
Эндрю Таненбаум

Модуль 3. Современные операционные системы различного назначения

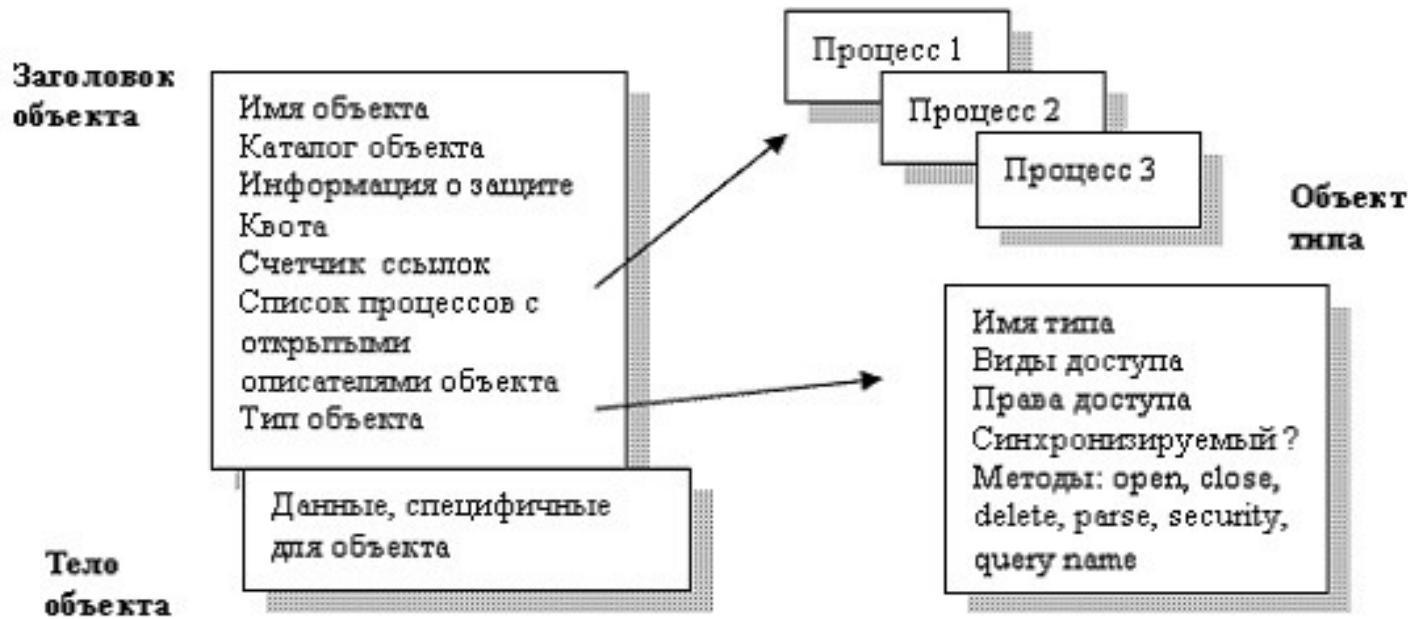
Лекция 15 (3.2)

Общая характеристика Windows7 и др. Структура Windows. Объекты. Диспетчеризация и контроль доступа. Типы объектов, структура объектов, управление объектами и защита объектов. Процессы и потоки. Планирование потоков. Обработка прерываний и исключений и др.

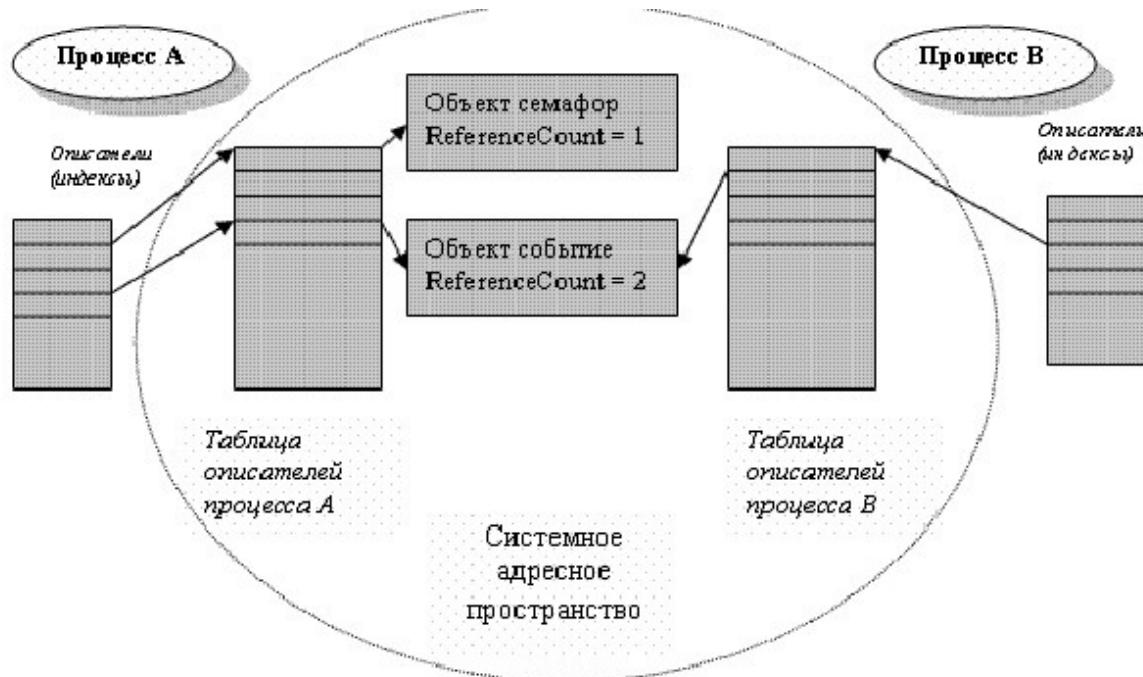
Структура ОС



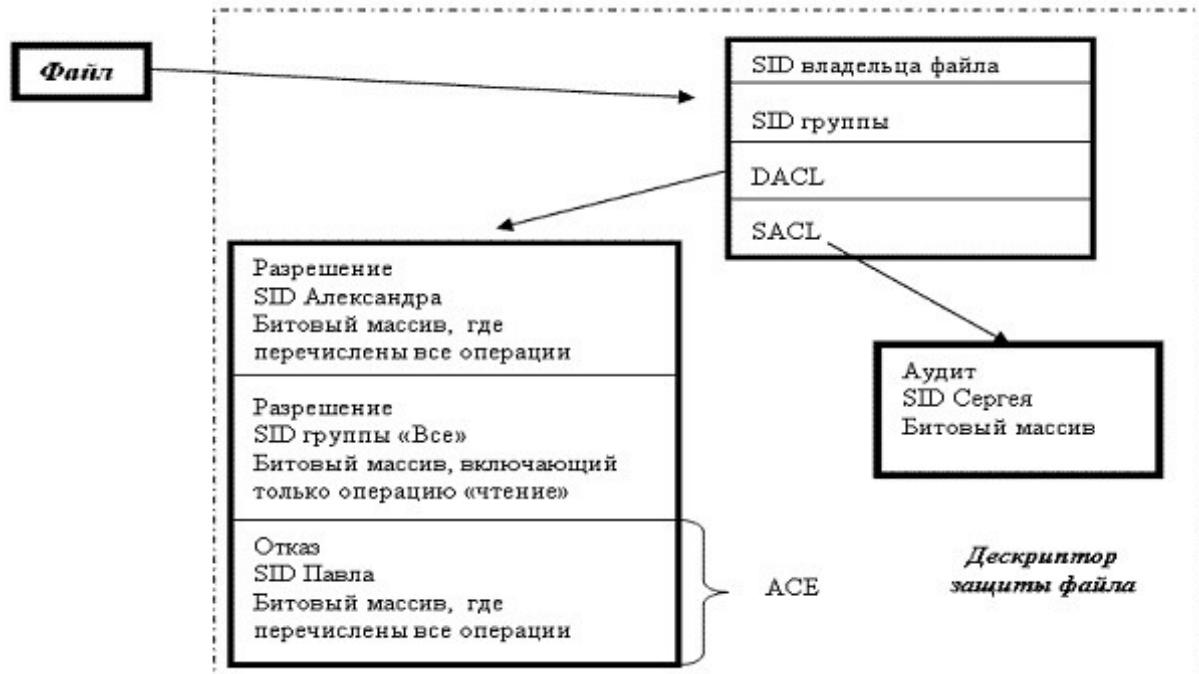
Объекты



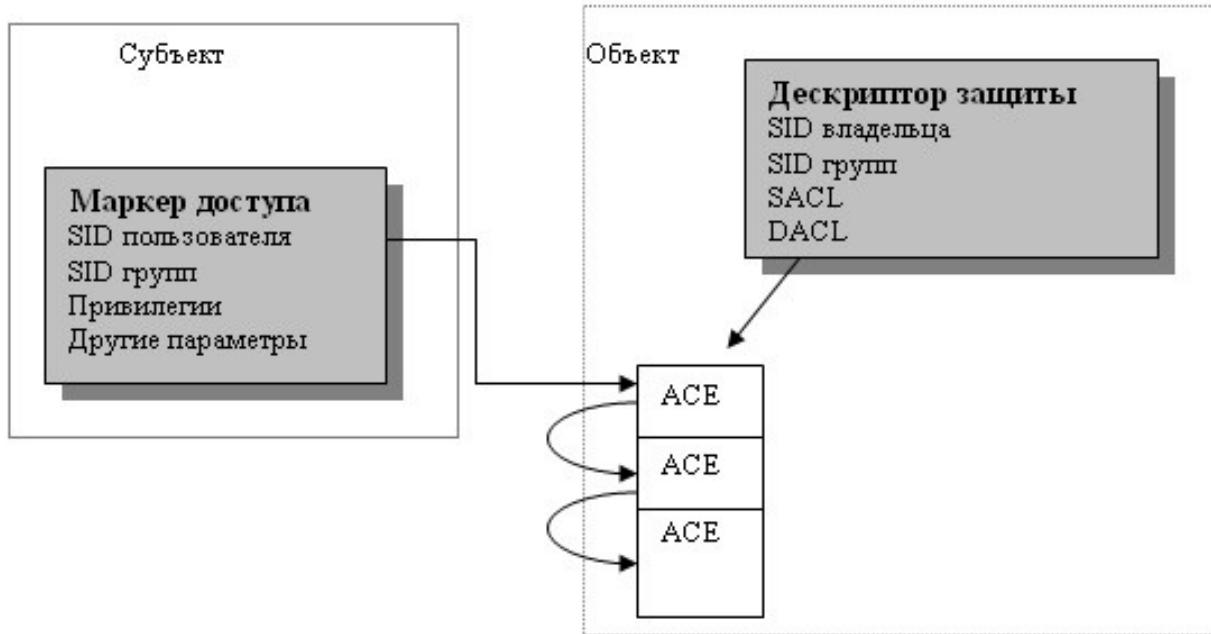
Объекты



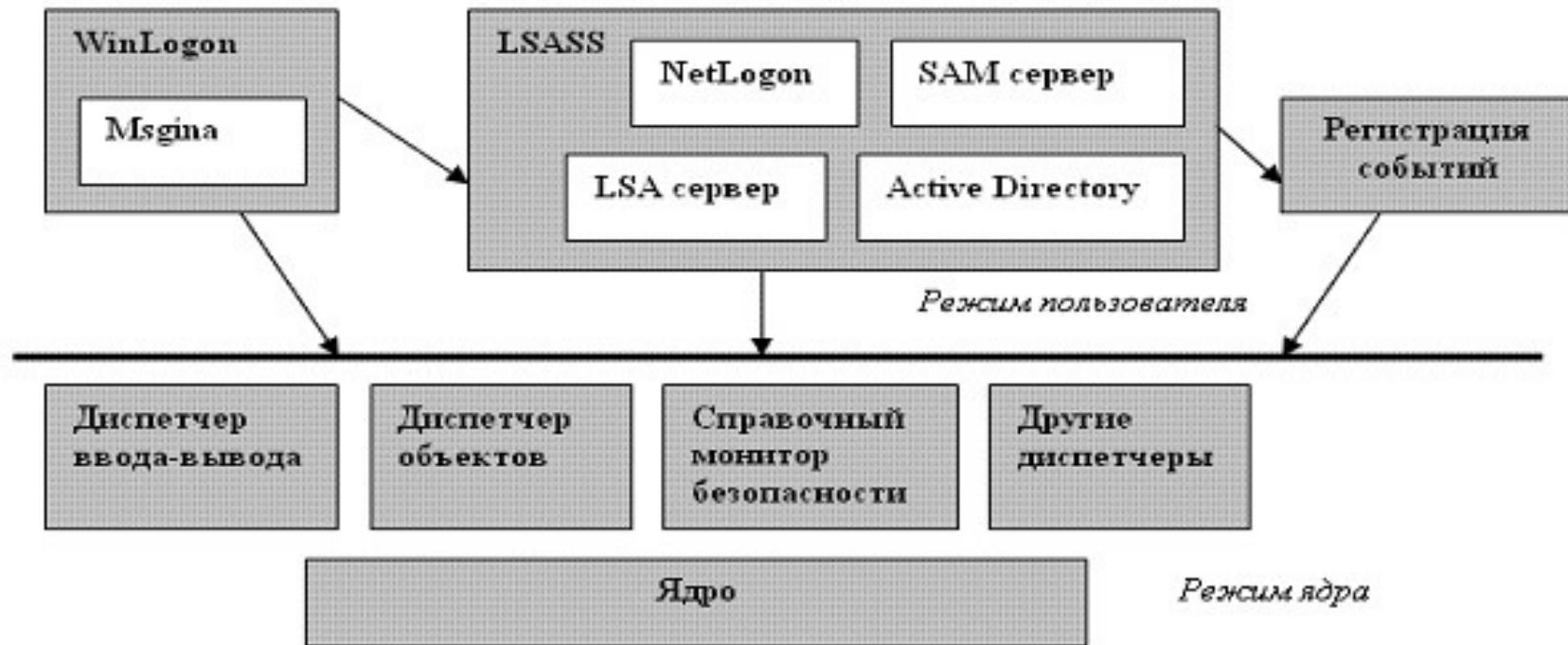
Диспетчеризация и контроль доступа



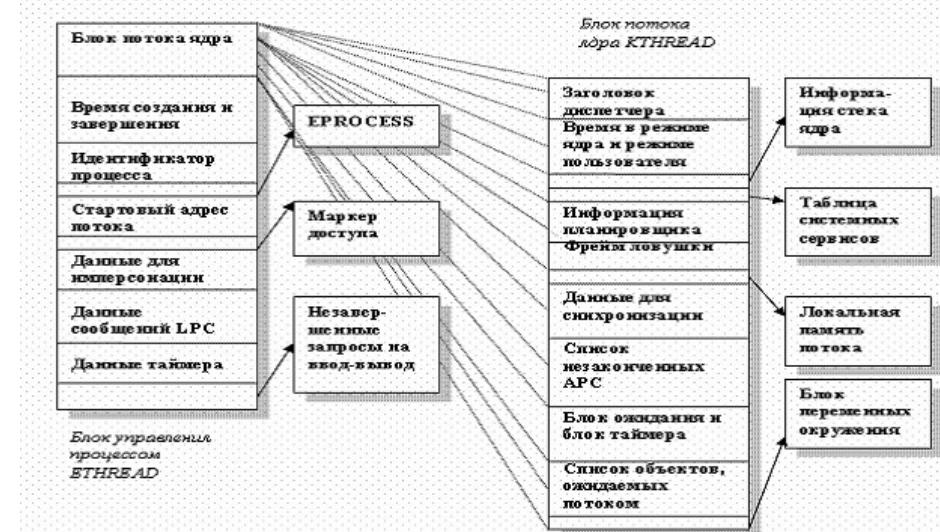
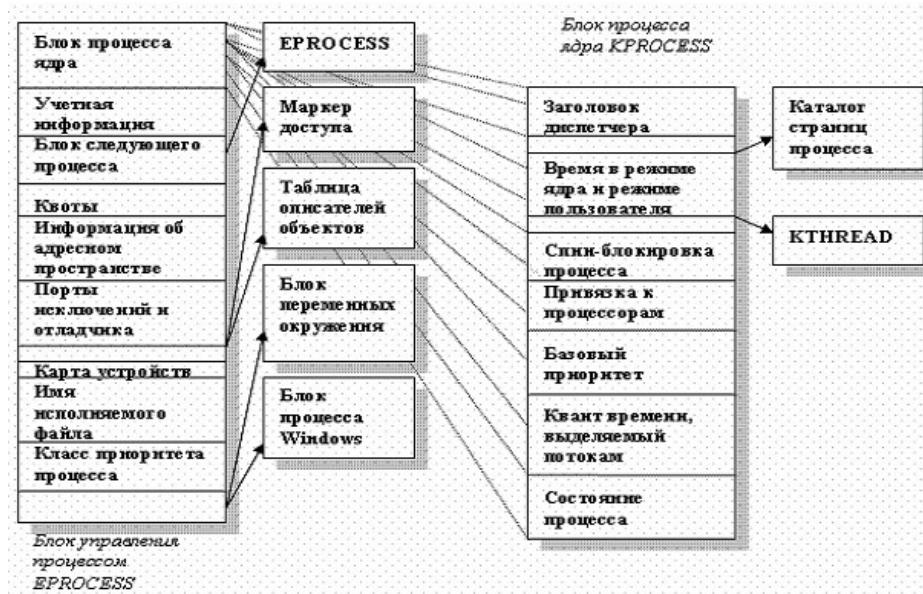
Диспетчеризация и контроль доступа



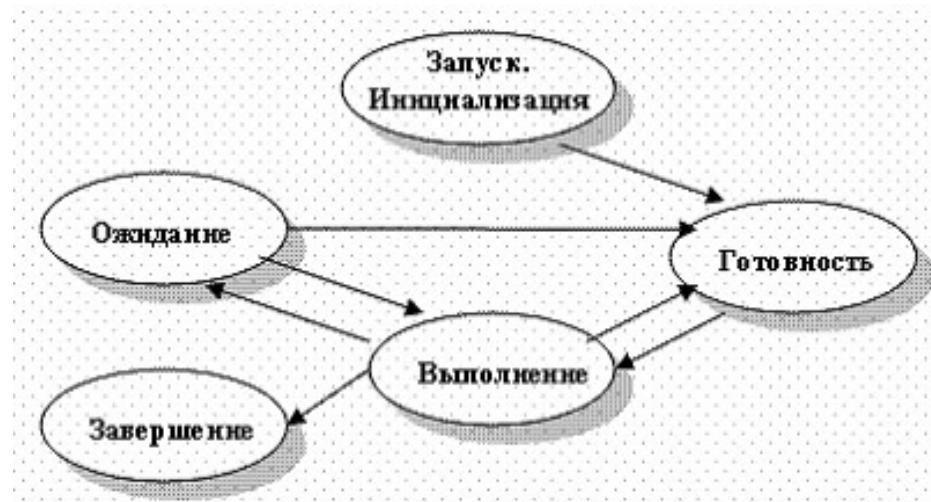
Структура системы защиты. Привилегии



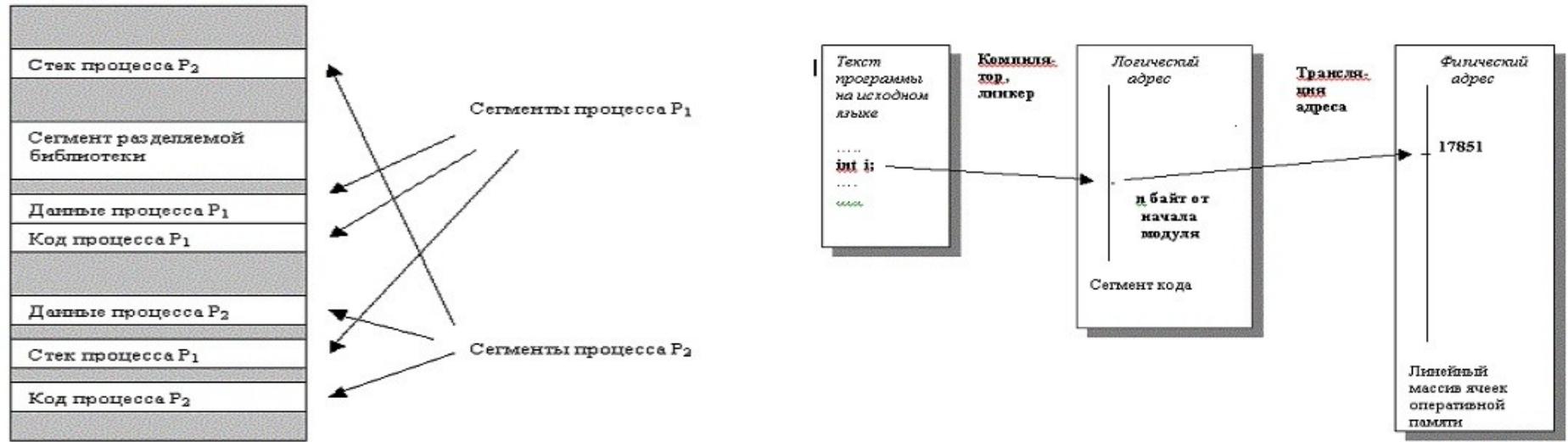
Процессы и потоки



Процессы и потоки



Виртуальное адресное пространство процесса

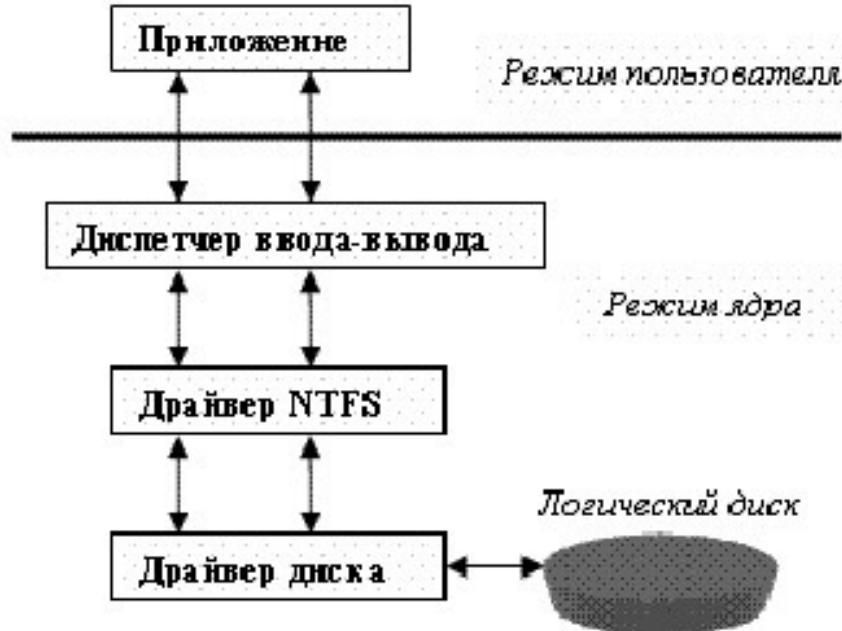


Модуль 3. Современные операционные системы различного назначения

Лекция 16 (3.3)

Диспетчер виртуальной памяти в Windows. Защита памяти.
Механизм реализации виртуальной памяти. Ядро Windows.
Система ввода/вывода и файловая система Windows. Об-
работка ввода/вывода. Драйверы однослойные и
многослойные. Сетевые возможности Windows. Сетевое API.

Файловая система



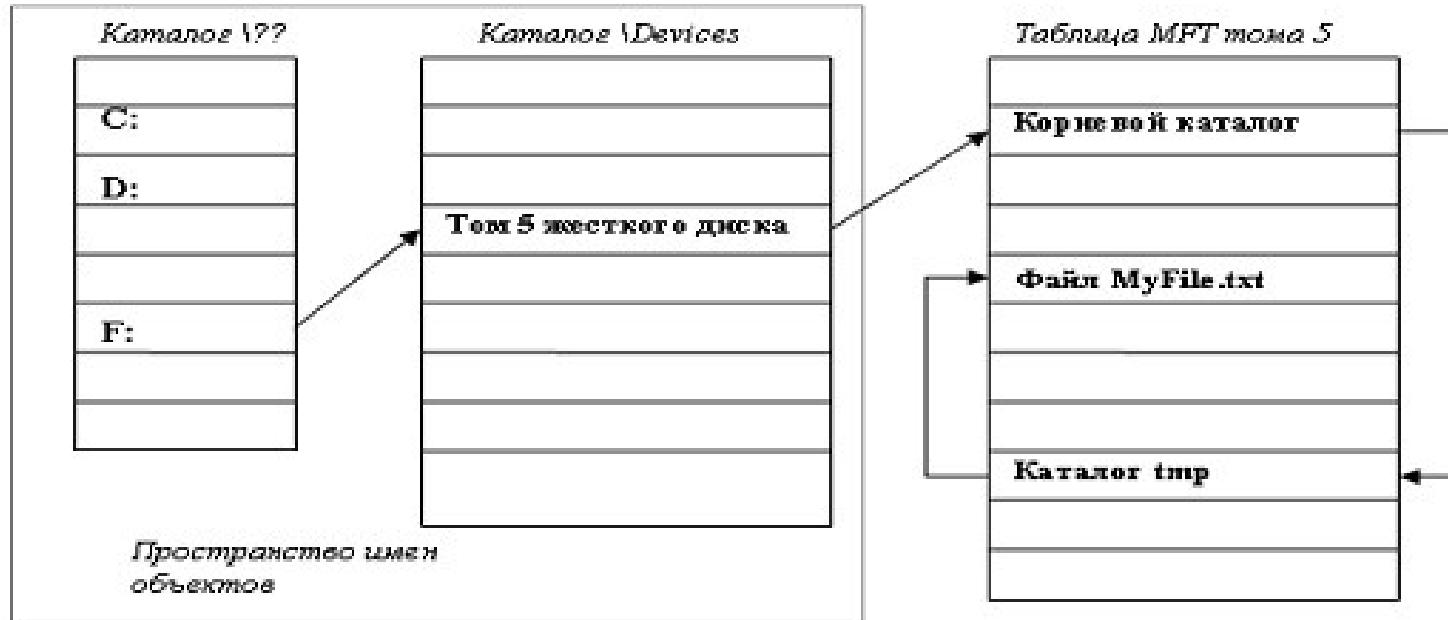
16
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

....
Второй файл пользователей
Первый файл пользователей

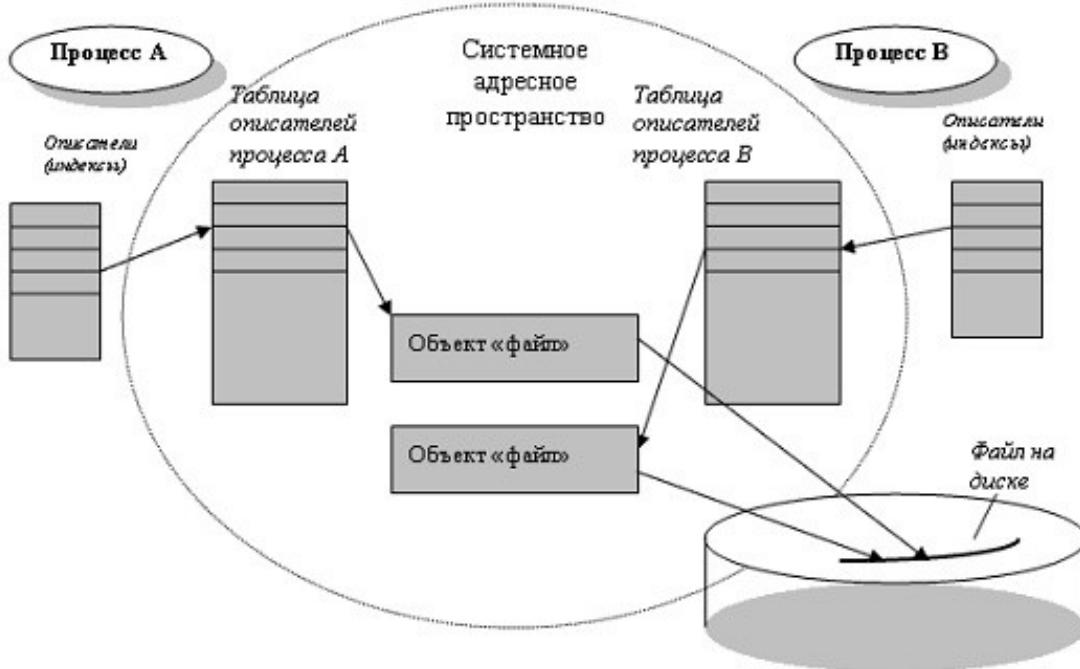
Зарезервированы

Расширения имен файлов, квоты
Преобразование регистра
Файл описателей защиты
Список плохих кластеров
Загрузочный сектор
Битовый массив учета занятых кластеров
Корневой каталог
Таблица определяемых атрибутов
Файл темы
Файл журнала для восстановления
Зеркальная копия MFT

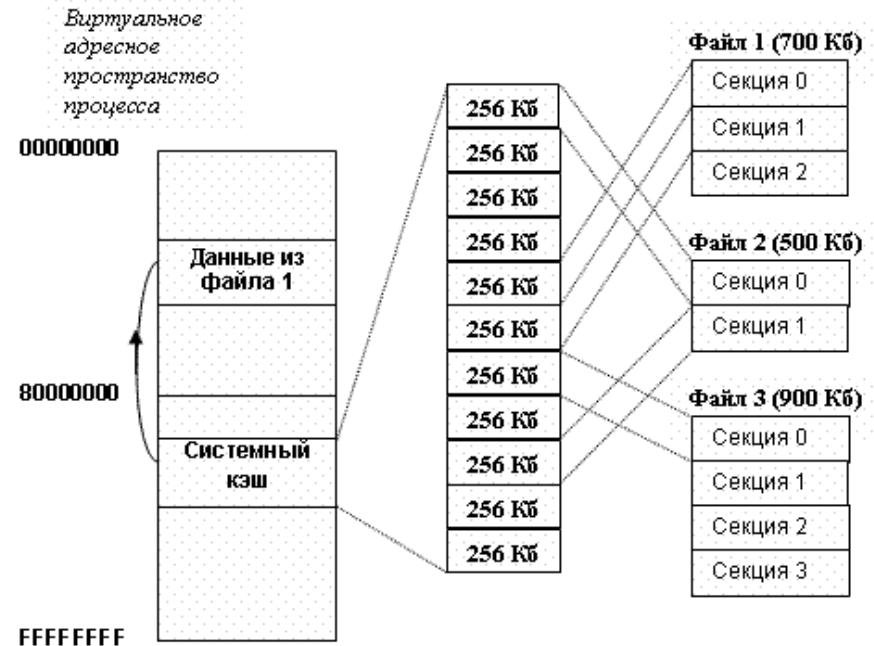
Файловая система (поиск)



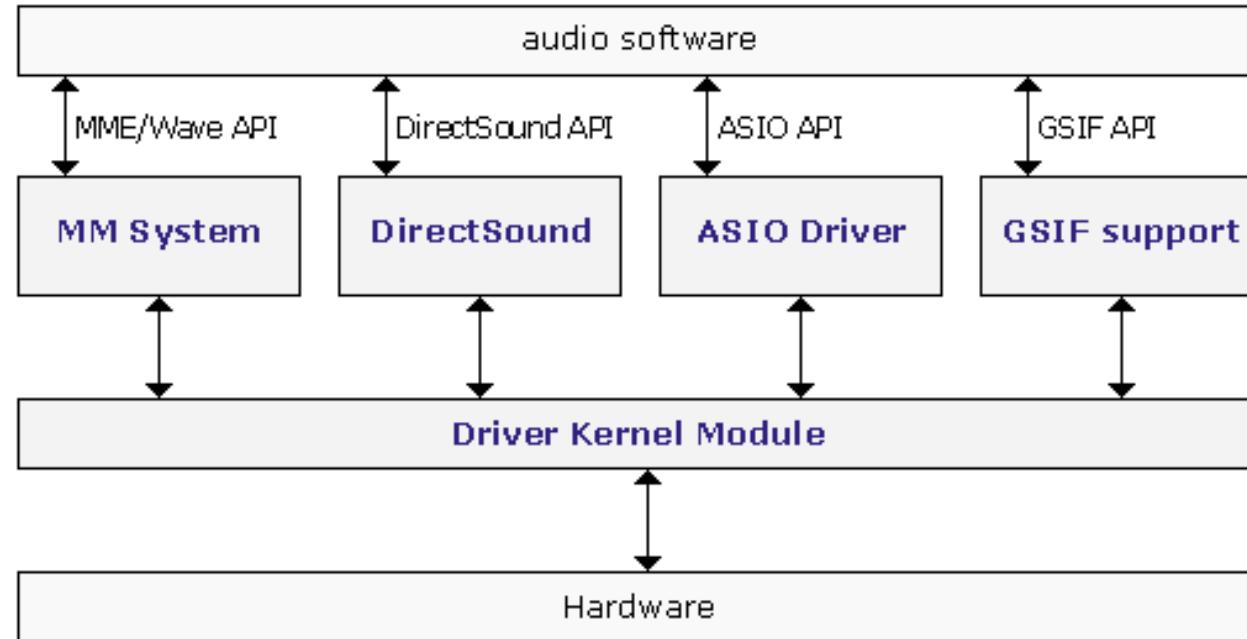
Файловая система (совместный доступ)



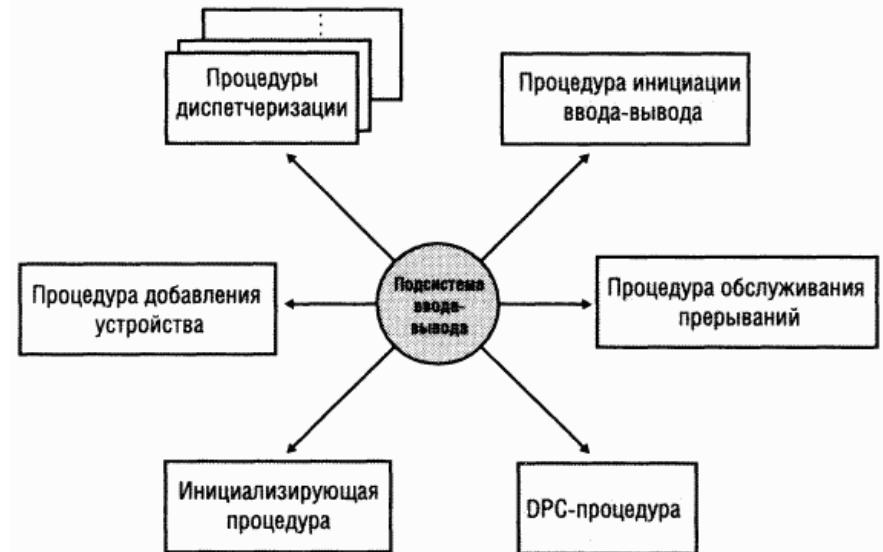
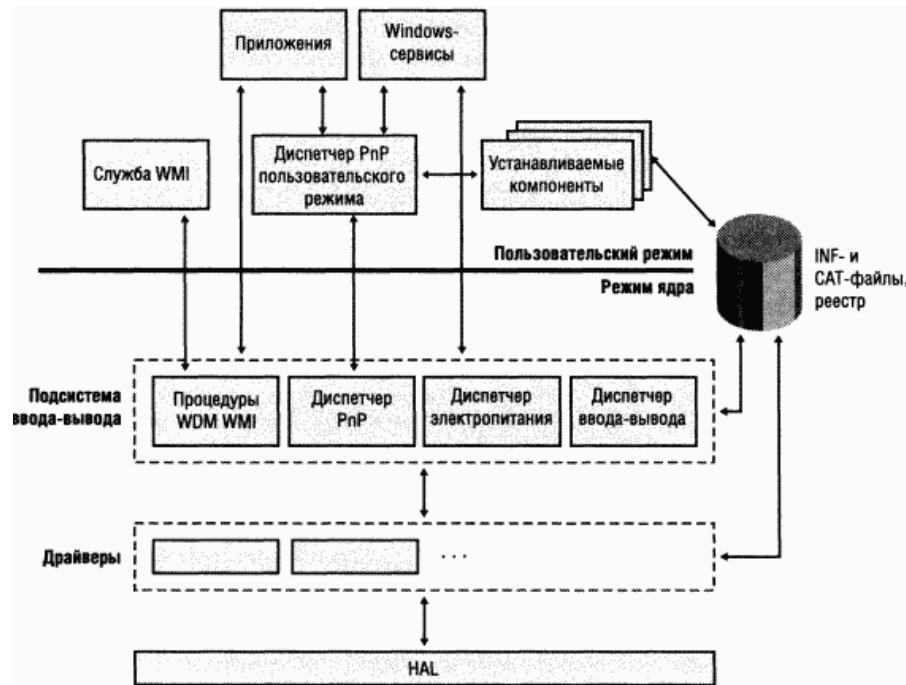
Файловая система (кэширование)



Драйверы



Подсистема Вв/Выв



Взаимодействие драйвера ФС и жесткого диска

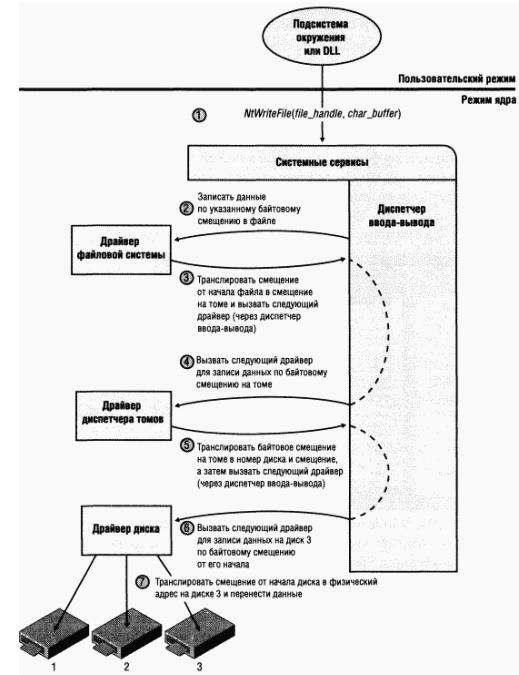
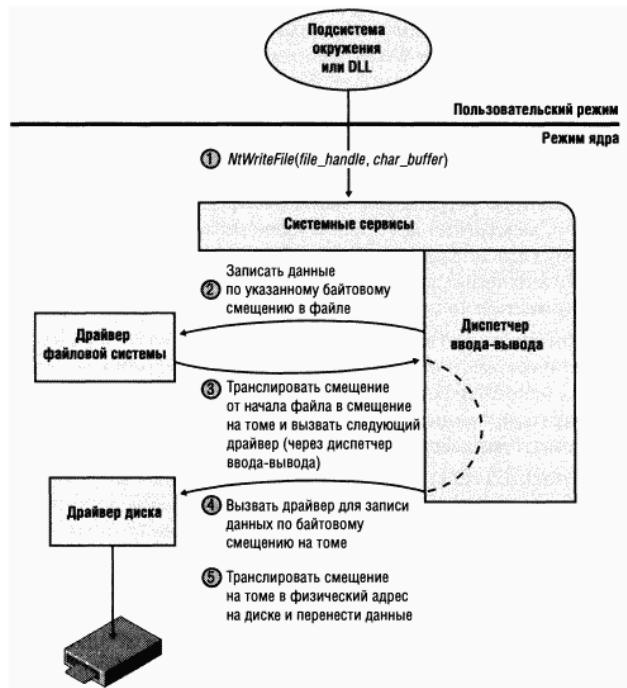
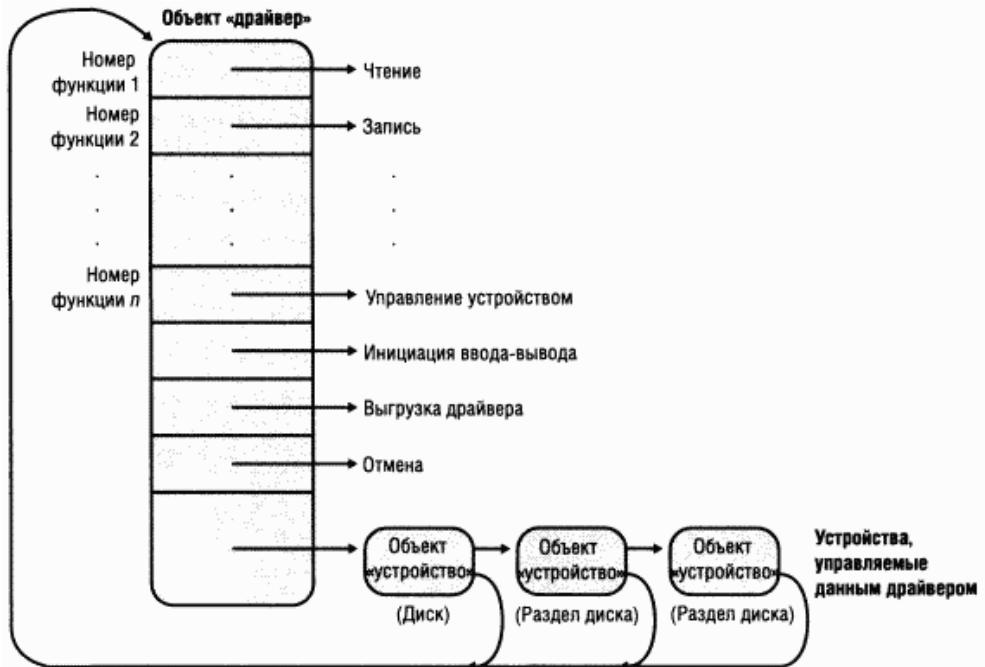
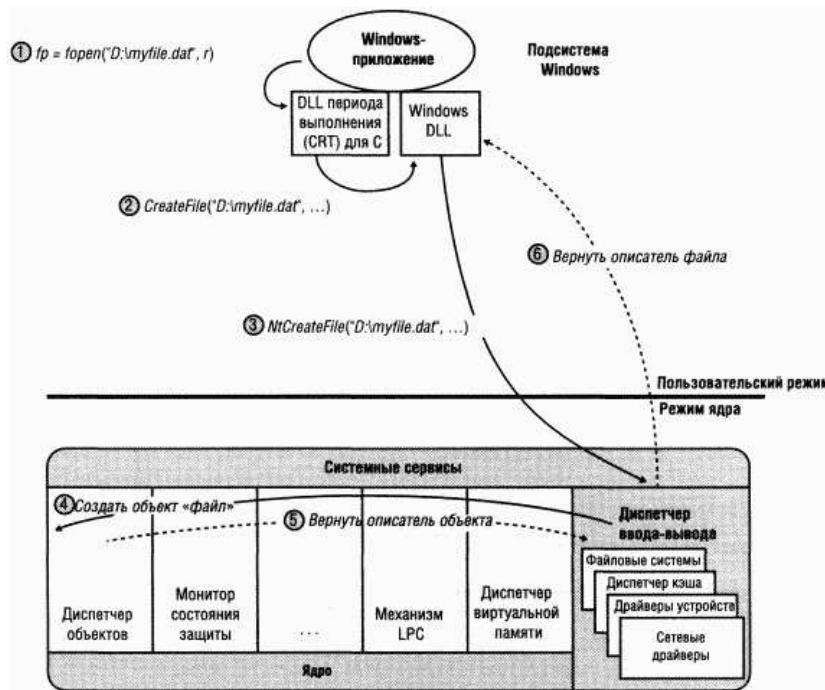


Рис. 9-4. Добавление промежуточного драйвера

Объект “драйвер”



Действия с объектом “файл”



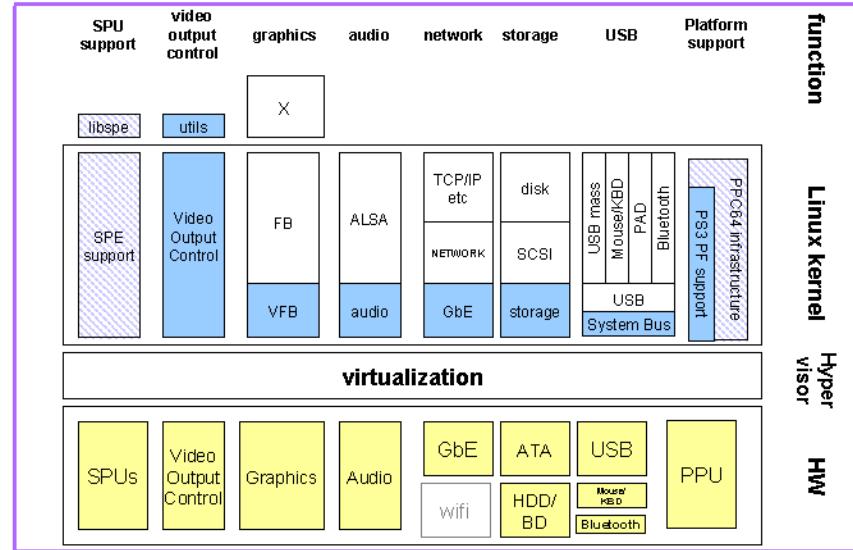
Модуль 3. Современные операционные системы различного назначения

Лекция 17 (3.4)

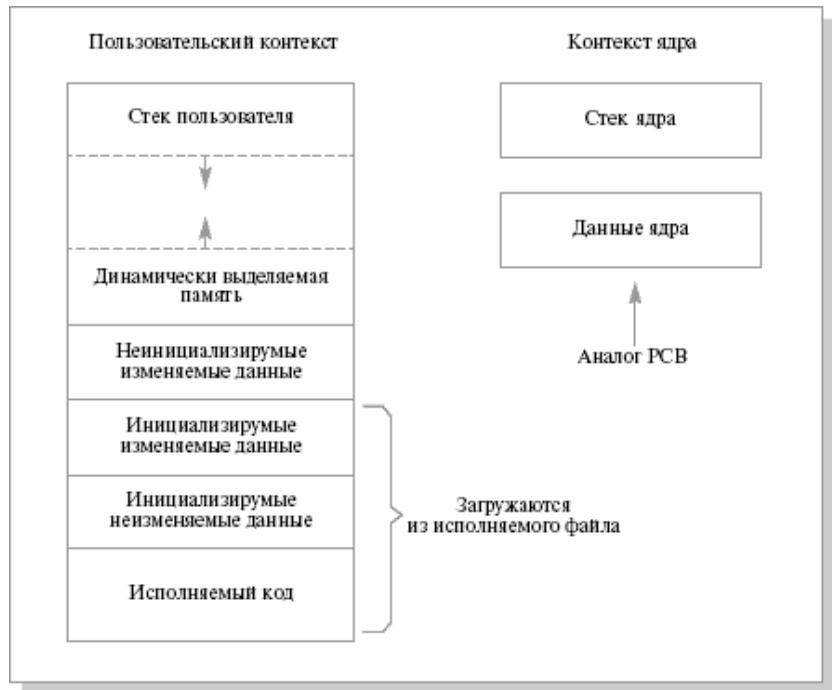
ОС UNIX

(Общая характеристика ОС UNIX и разновидности. Концептуальные особенности. Инструментальные средства ОС UNIX. Понятие процесса и ядра. Сегментация виртуального адресного пространства процесса. Структура контекста процесса. Идентификатор и дескриптор процесса. Структура и иерархия процессов. Программное управление процессами посредством системных вызовов ядра. Средства коммуникации процессов. Управление оперативной и внешней памятью и др.)

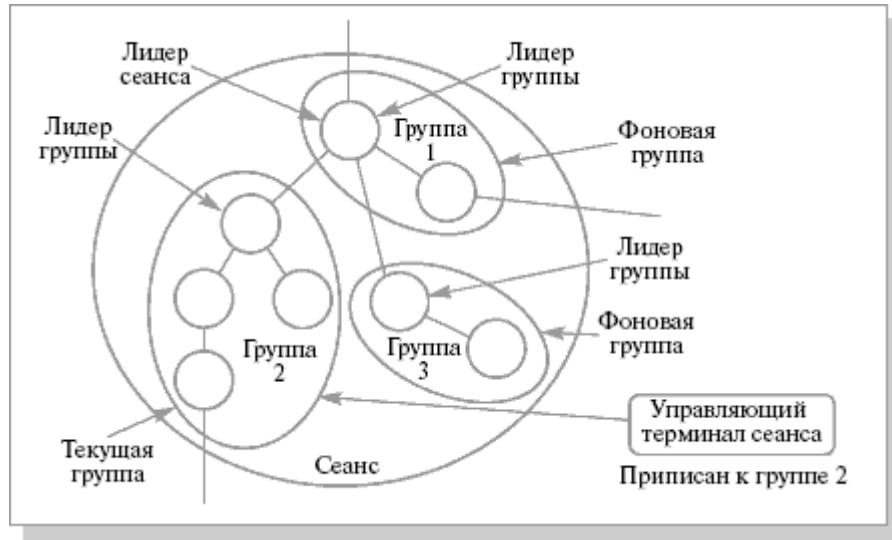
Архитектура



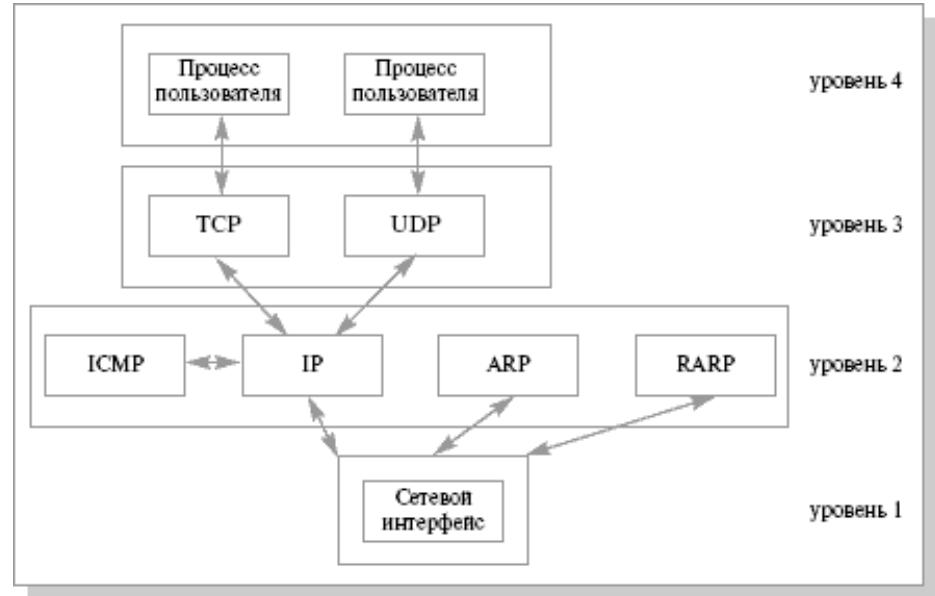
Контекст процесса и граф состояния процессса



“Группирование” процессов



Сети (референсные модели)



Клиент-серверное взаимодействие (UDP и TCP)

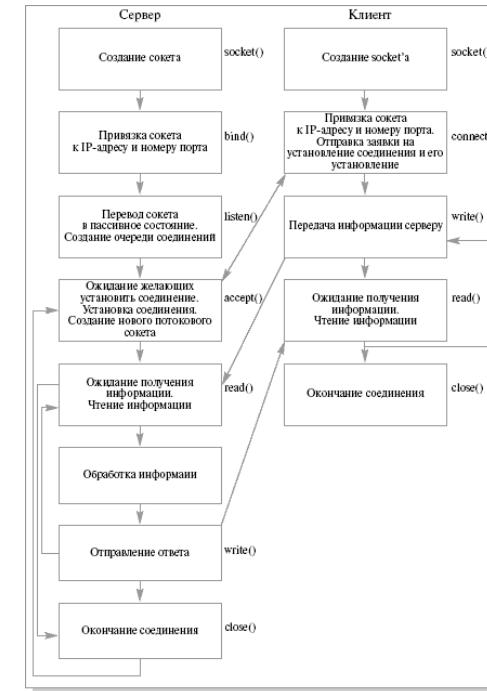


Схема работы TCP-сервера с параллельной обработкой запросов

