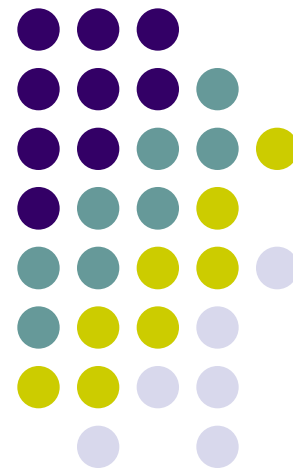
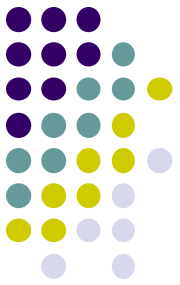


Введение

Операционные системы

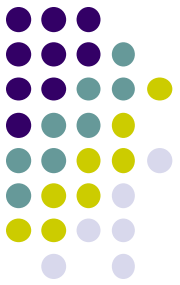
Лекция 1





План

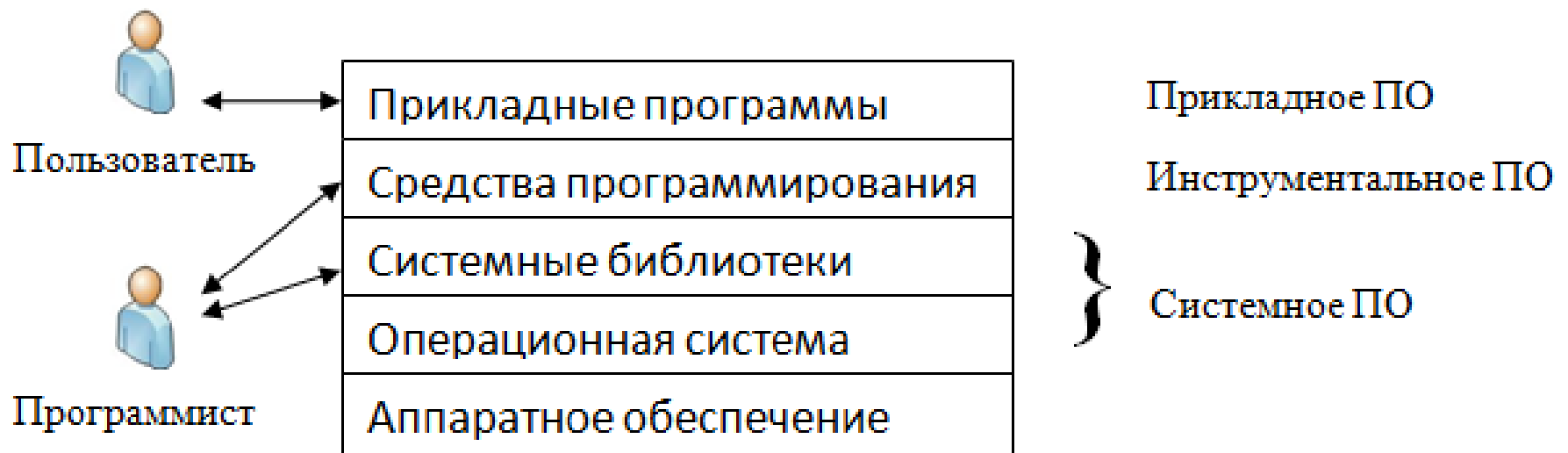
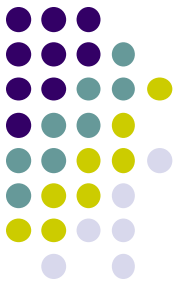
- Основные понятия
- Функции операционной системы
- История операционных систем
- Классификация операционных систем



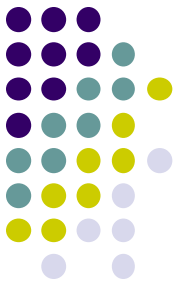
Основные понятия

- *Операционная система* (сокращенно ОС) – это базовый комплекс взаимосвязанных компьютерных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

Основные понятия

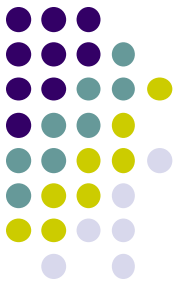


Компоненты вычислительной системы



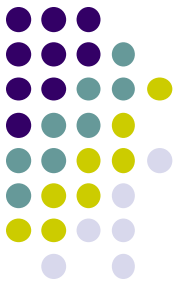
Основные понятия

- Взаимодействие всех программ с операционной системой осуществляется при помощи *системных вызовов* (system calls)
- Набор системных вызовов образует *интерфейс прикладного программирования* (Application Programming Interface, API)



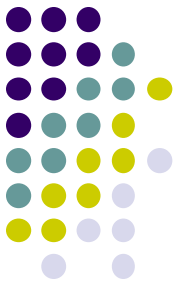
Основные понятия

- *Операционная оболочка (operation shell)* – комплекс программ, ориентированных на определенную ОС и предназначенный для облегчения диалога между пользователем и компьютером при выполнении определенных видов деятельности на компьютере.



Функции ОС

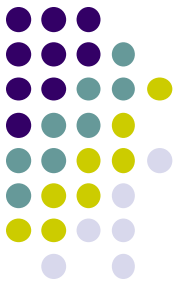
- загрузка приложений в оперативную память и их выполнение
- управление оперативной памятью
- обеспечение управления доступом к данным на энергонезависимых носителях с помощью файловой системы
- обеспечение стандартизованного доступа к периферийным устройствам



Функции ОС

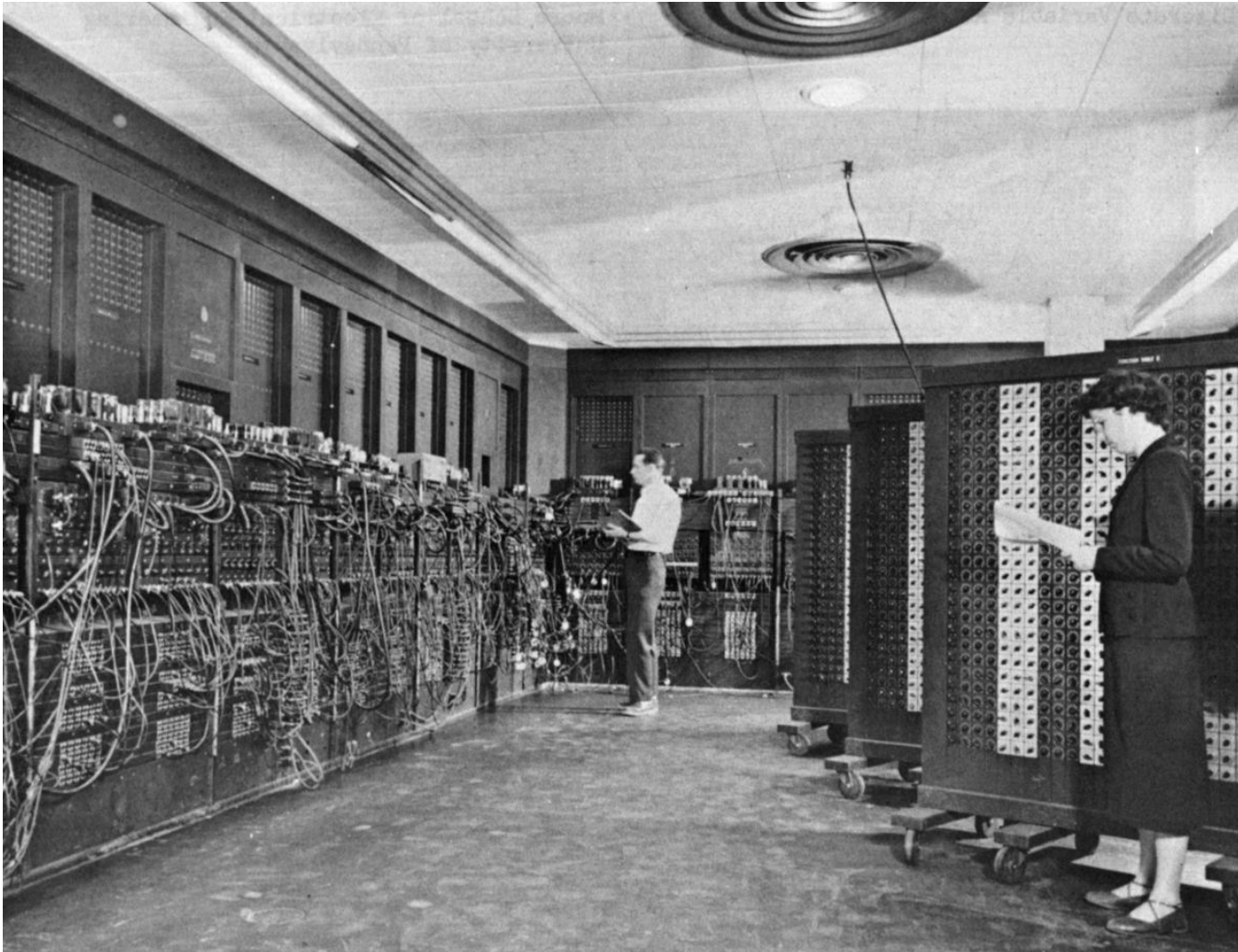
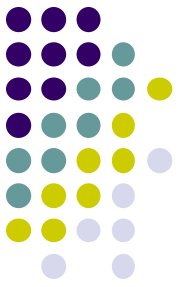
- обеспечение пользовательского интерфейса
- обеспечение сетевого взаимодействия
- защита системных ресурсов, данных и программ пользователя, процессов и самой операционной системы от несанкционированного доступа

История операционных систем



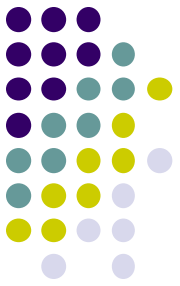
- ***Первый период (1945 -1955)***
 - Первые ламповые вычислительные устройства (1946-48гг.).
 - одна и та же группа людей участвовала и в проектировании, и в эксплуатации, и в программировании вычислительной машины
 - Программирование осуществлялось исключительно на машинном языке.

История операционных систем

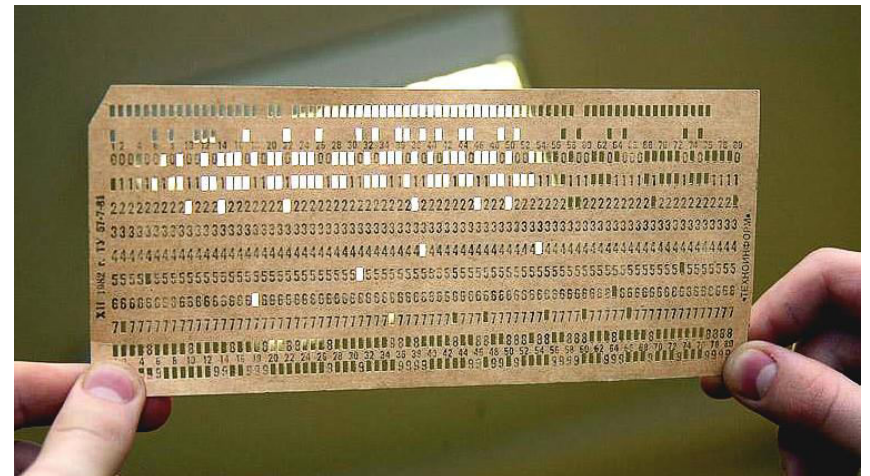
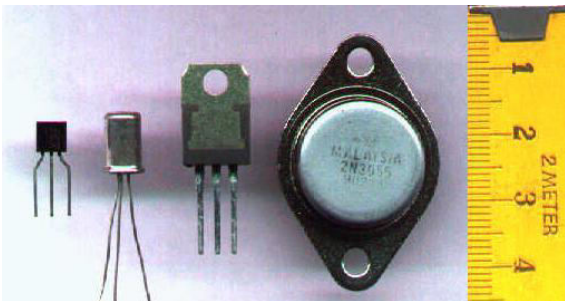


ЭНИАК

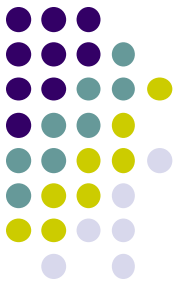
История операционных систем



- ***Второй период (1955 – 1965)***
 - Появление полупроводниковых элементов
 - разделение персонала на программистов и операторов
 - первые алгоритмические языки
 - первые системы пакетной обработки
 - язык управления заданиями

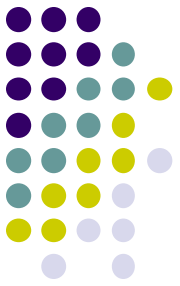


История операционных систем

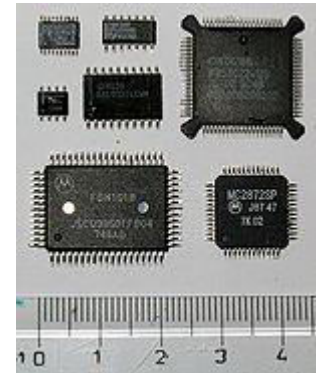


- ***Третий период (1965 – 1980)***
 - переход к интегральным микросхемам
 - создание семейств программно-совместимых машин (IBM/360)
 - ОС состояли из многих миллионов ассемблерных строк и содержали тысячи ошибок (OS/360 и другие)
 - реализация мультипрограммирования
 - спулинг (spooling)
 - появился новый тип ОС – системы разделения времени

История операционных систем

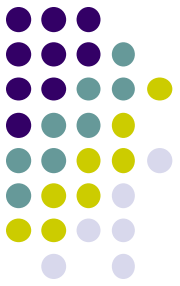


IBM System/360



Интегральные схемы
содержат сотни
миллионов транзисторов

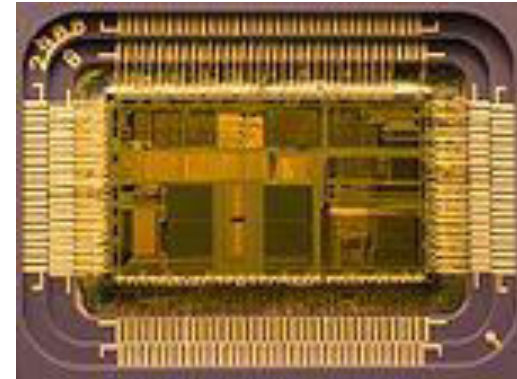
История операционных систем



- **Четвертый период (1980 – настоящее время)**
 - появление больших интегральных схем (БИС)
 - эра персональных компьютеров
 - доминировали две ОС: MS-DOS и UNIX
 - сетевые или распределенные ОС.

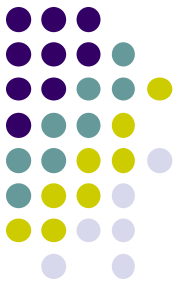


Altair-8800



Микропроцессор заменил множество интегральных схем

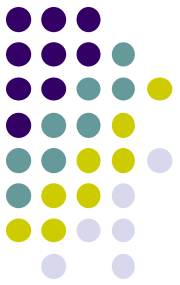
Классификация операционных систем



по числу одновременно выполняемых задач

- однозадачные (CP/M, MS-DOS, PC-DOS)
- многозадачные
 - Системы пакетной обработки (ОС ЕС)
 - Системы с разделением времени (UNIX, VMS)
 - Системы реального времени (QNX, Windows CE, SafeRTOS)

Классификация операционных систем



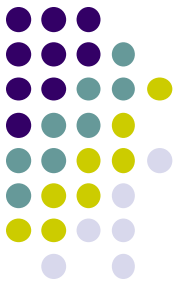
**по числу одновременно работающих
пользователей на ЭВМ ОС
разделяются на**

- **однопользовательские**
- **многопользовательские**

по типу лицензии:

- **проприетарная**
- **открытая**

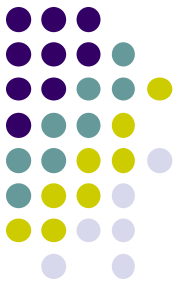
Классификация операционных систем



по архитектуре:

- Микроядерные (QNX, Minix)
- Монолитные (Linux, FreeBSD)
- Гибридные (Windows)

Классификация операционных систем



по использованию процессора:

- однопроцессорные
- многопроцессорные системы

по возможности сетевого взаимодействия:

- локальные (MS-DOS)
- сетевые (Microsoft Windows, UNIX, Novell NetWare)

Классификация операционных систем



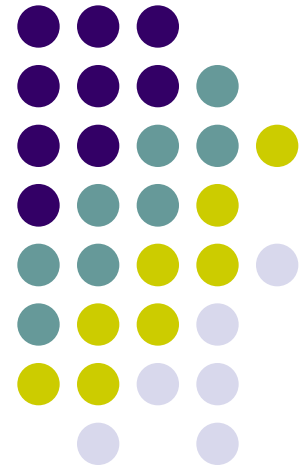
по применению:

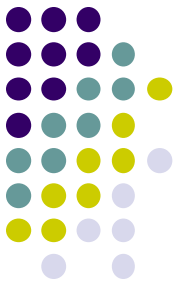
- рабочих станций (Linux, FreeBSD, Windows 7, Windows 8, OS X)
- Серверов (Solaris, FreeBSD, Linux и Windows Server)
- для мобильных устройств (Android, iOS)
- для сетевых маршрутизаторов
- ОС реального времени
- встроенные ОС (Embedded Linux, VxWorks)
- специализированные (управление производством, обучение, и т. п.)

Структура ОС

Операционные системы

Лекция 2





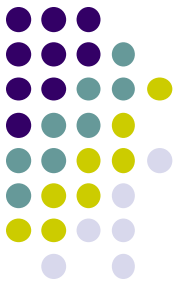
План

- Компоненты ОС
- Архитектура ядра ОС
- Требования к современным ОС

Компоненты ОС

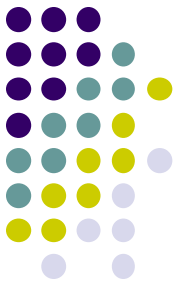


- Группы компонентов:
 - ядро
 - системные библиотеки
 - пользовательская оболочка



Компоненты ОС

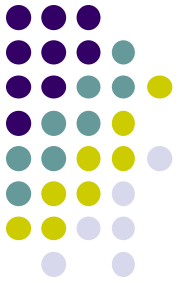
- Ядро обеспечивает:
 - управление процессами, планировка задач
 - обработку прерываний
 - операции ввода/вывода
 - базовое управление памятью



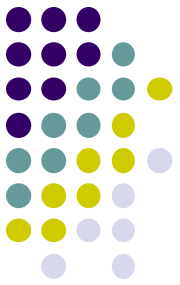
Компоненты ОС

- Режимы работы процессора:
 - привилегированный
 - непривилегированный (пользовательский)
- Привилегии обеспечиваются за счет запрета выполнения в пользовательском режиме команд, связанных с операциями:
 - переключением процессора с задачи на задачу
 - управлением устройствами ввода-вывода
 - доступом к механизмам распределения и защиты памяти

Компоненты ОС

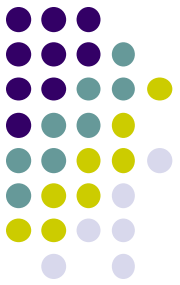


- Модули:
 - резидентные
 - транзитные



Архитектура ядра ОС

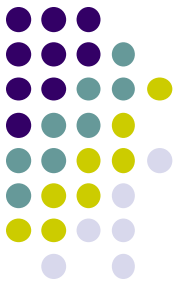
- Основные типы архитектур ядра:
 - монолитное (макроядро – в ОС для встраиваемых систем, Linux)
 - микроядро (QNX и Symbian OS, Minix3, GNU Hurd)
 - гибридное (MacOS X, Windows NT, DragonFly BSD)



Архитектура ядра ОС

- Монолитное ядро (макроядро) компонуется как одна программа, работающая в привилегированном режиме.
- Такое ядро делится на основные компоненты и модули, реализующие дополнительную функциональность
- Переход из пользовательского режима в режим ядра осуществляется через системные вызовы — интерфейс ядра операционной системы.

Архитектура ядра ОС



Монолитная архитектура

Системные библиотеки

Режим пользователя



Режим ядра

Системные вызовы

Диспетчер
процессов

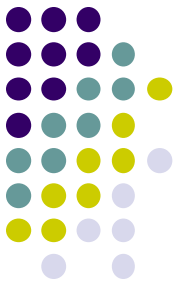
Диспетчер
памяти

Обработчик
прерываний

Управление
доступом

Модули

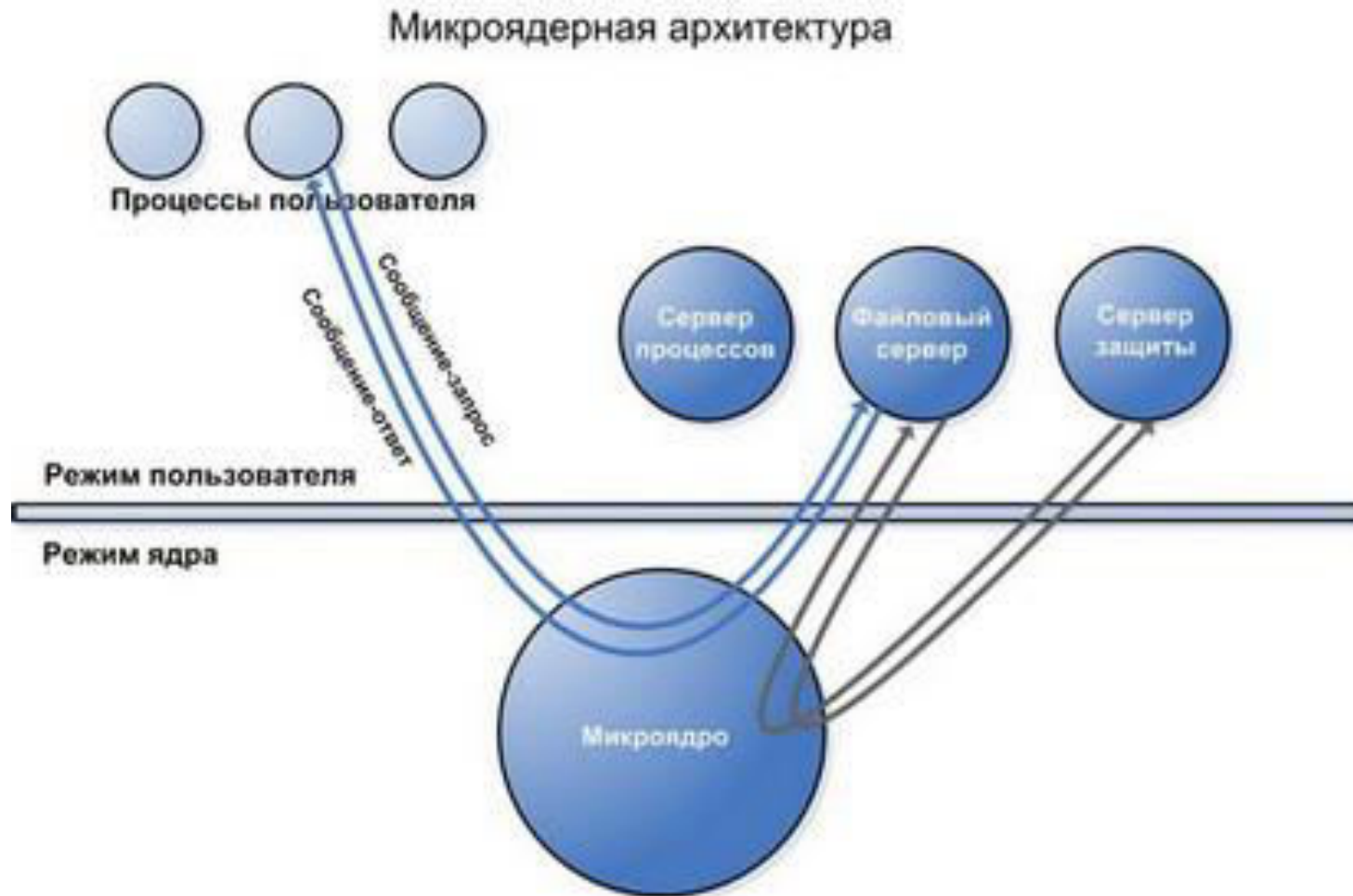
Аппаратная часть

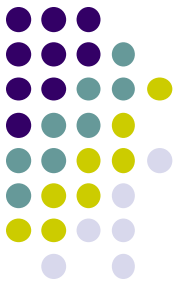


Архитектура ядра ОС

- Микроядро работает в привилегированном режиме и выполняет только минимум функций по управлению аппаратурой.
- Более высокоуровневые функции операционной системы выполняются специализированными компонентами — серверами, работающими в пользовательском режиме.
- Управление и обмен данными при этом осуществляется через передачу сообщений, доставка которых является одной из основных функций микроядра

Архитектура ядра ОС

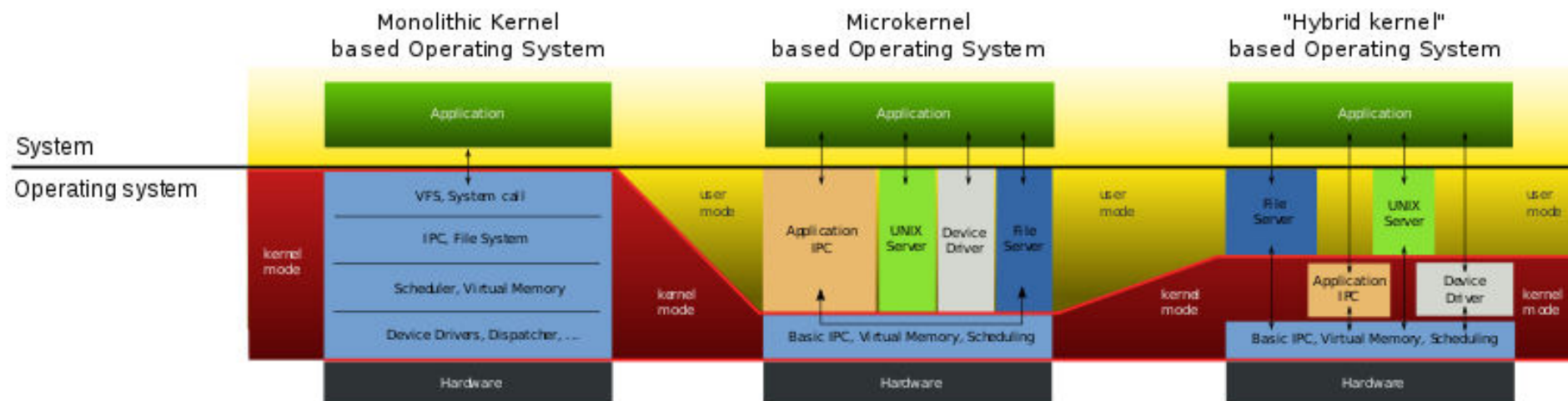




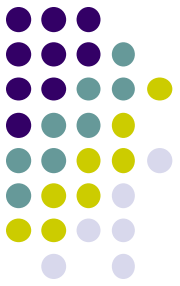
Архитектура ядра ОС

- **Гибридное ядро** - модифицированное микроядро, позволяющие для ускорения работы запускать «несущественные» части в пространстве ядра.
- Меньше по размеру, чем монолитное ядро
- Более гибкое, чем монолитное ядро
- Может работать медленнее

Архитектура ядра ОС



Требования к современным ОС

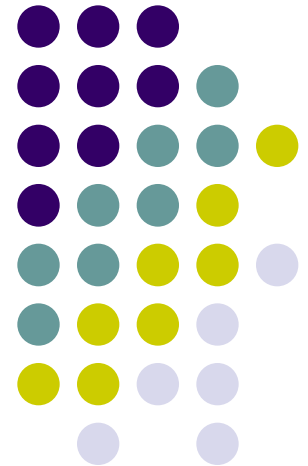


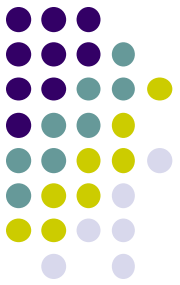
- расширяемость
- переносимость
- совместимость
- надежность и отказоустойчивость
- безопасность
- производительность

Процессы и потоки

Операционные системы

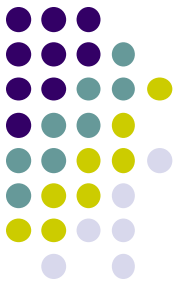
Лекция 3





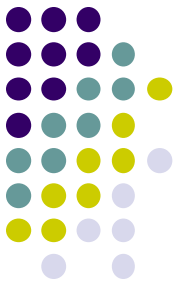
План

- Понятие процесса
- Управление процессами
- Иерархия процессов
- Состояния потоков
- Понятие потока



Понятия процесса

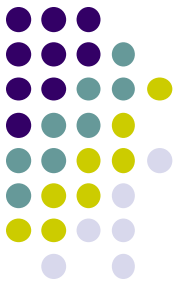
- *Процесс* (задача) –
- При управлении процессами ОС использует два основных типа информационных структур:
 - дескриптор процесса
 - контекст процесса



Понятия процесса

Дескриптор содержит:

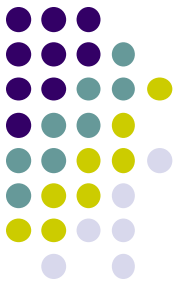
- PID (process identifier)
- информацию о состоянии процесса
- степень привилегированности процесса
- местоположение кодового сегмента
- данные о родственных процессах, в том числе PPID (parent process identifier)
- данные о событиях, которые ожидает процесс и др.



Понятия процесса

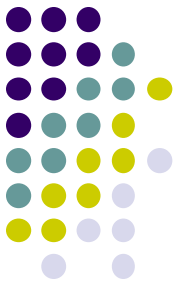
Контекст процесса содержит информацию:

- состояние аппаратуры компьютера
- значение счетчика команд
- содержимое регистров общего назначения
- режим работы процессора
- флаги
- маски прерываний
- указатели на открытые файлы
- информация о незавершенных операциях ввода-вывода
- коды ошибок выполняемых процессом системных вызовов и др.



Понятия процесса

- *Виртуальное адресное пространство процесса –*
- Образ процесса
- Средства межпроцессной связи

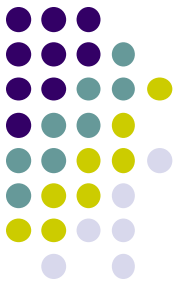


Понятия процесса и потока

Поток обладает следующими характеристиками:

- состояние выполнения потока
- сохраненный контекст не выполняющегося потока
- стек выполнения
- статическая память для локальных переменных
- доступ к памяти и ресурсам процесса, которому этот поток принадлежит

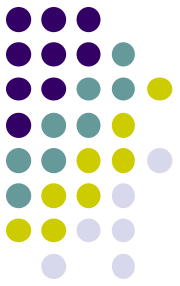
Управление процессами



Управление процессами состоит в

- создании
- уничтожении
- приостановлении
- возобновлении
- изменении приоритета
- переключении состояний

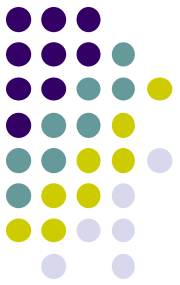
Управление процессами



Основные события, приводящие к созданию процессов:

- загрузка системы
- работающий процесс подает системный вызов на создание процесса
- запрос пользователя на создание процесса

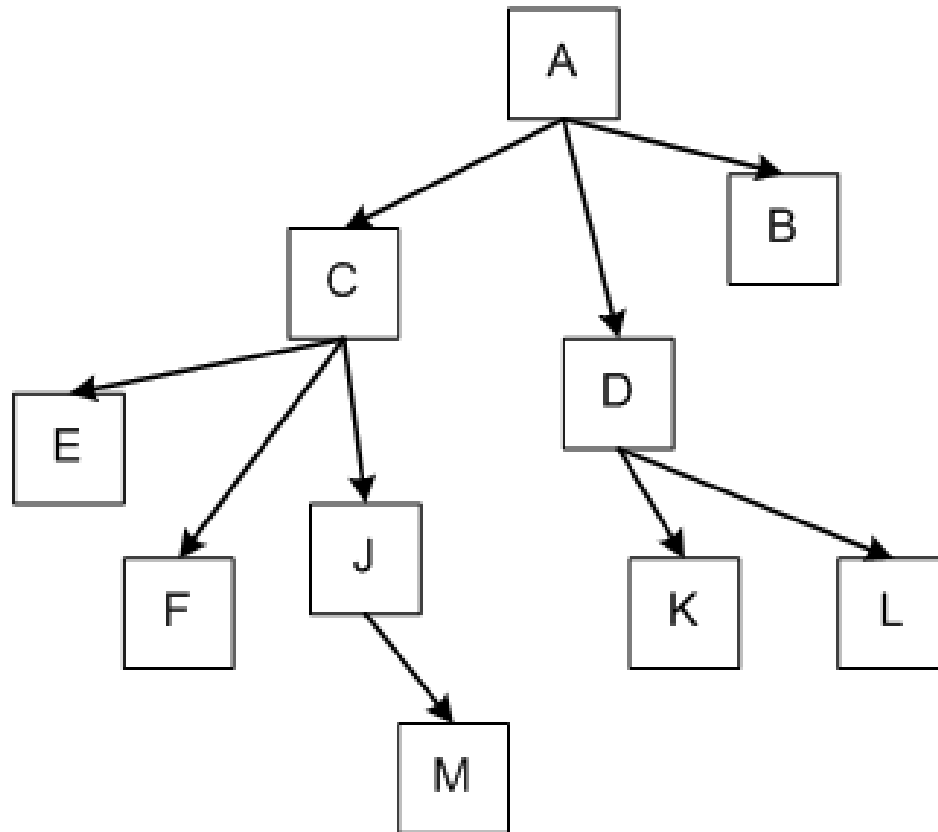
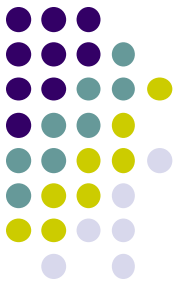
Управление процессами



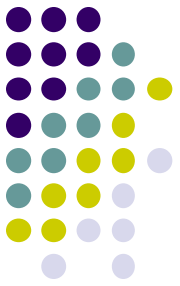
События, приводящие к завершению процесса:

- плановое завершение
- плановый выход по известной ошибке
- выход по неисправимой ошибке
- уничтожение другим процессом

Иерархия процессов

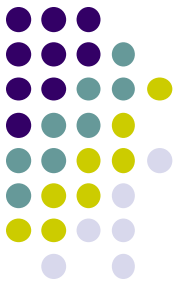


Иерархия процессов



В UNIX-системах:

- процесс, созданный системным вызовом *fork()*, является дочерним к предыдущему процессу.
- Прародитель всех процессов – процесс *init* (PID=1)
- Конфигурационный файл */etc/inittab*
- PPID процесса *init* равен 0

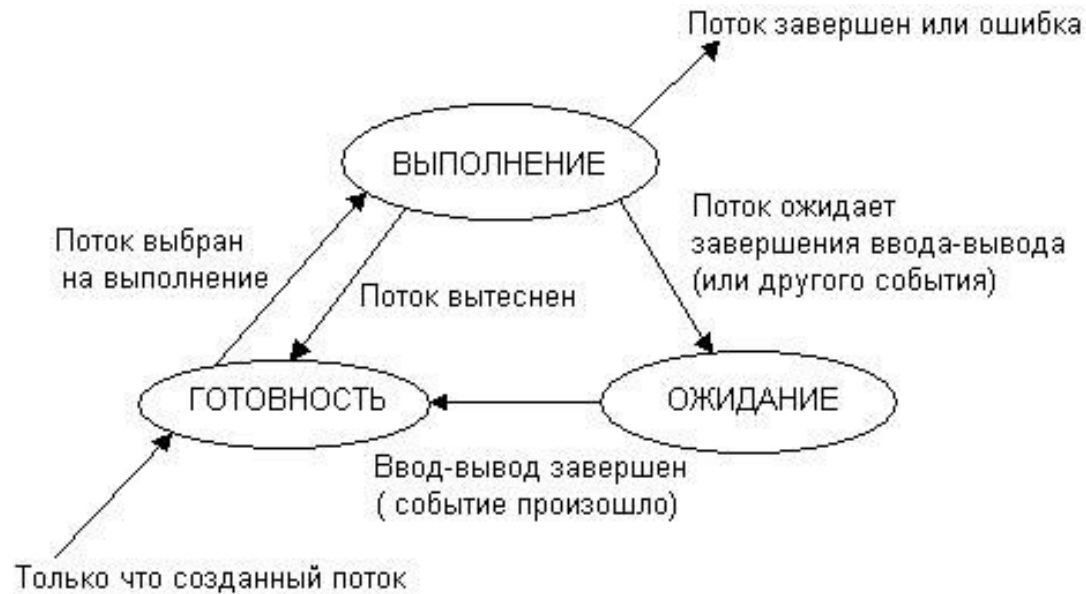
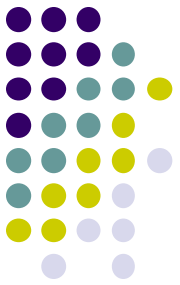


Состояния потоков

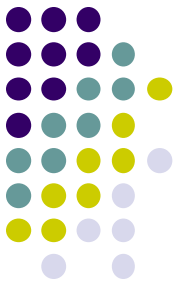
Поток может находиться в нескольких состояниях:

- Выполнение
- Ожидание
- Готовность

Состояния потоков



Состояния потоков

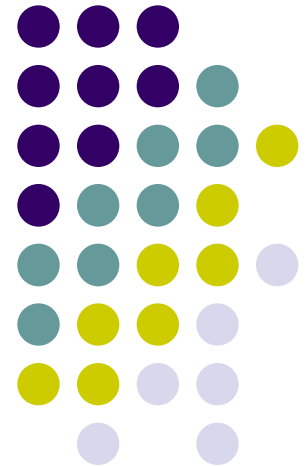


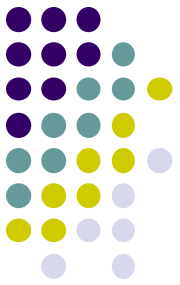
Размещение потоков в очереди

Планирование ПОТОКОВ

Операционные системы

Лекция 4

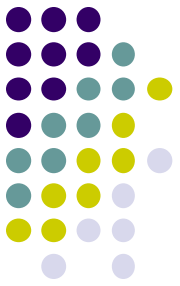




План

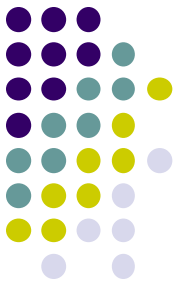
- Планирование и диспетчеризация
- Вытесняющие и невытесняющие алгоритмы планирования
- Алгоритмы, основанные на квантовании
- Алгоритмы, основанные на приоритетах
- Смешанные алгоритмы планирования

Планирование и диспетчеризация



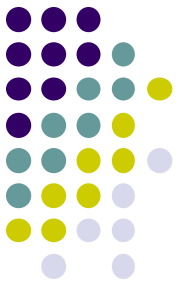
- Планирование
 - динамическое (on-line)
 - статическое (off-line)
- Диспетчеризация

Вытесняющие и невытесняющие алгоритмы планирования



- *Невытесняющая многозадачность* (non-preemptive multitasking)
- *Вытесняющая многозадачность* (preemptive multitasking)

Алгоритмы, основанные на квантовании



Граф состояний потока в системе с квантованием

Алгоритмы, основанные на квантовании

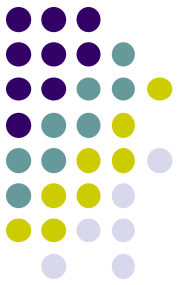
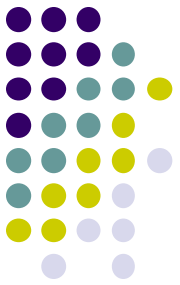


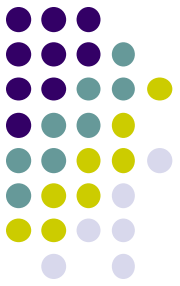
Иллюстрация расчета времени ожидания в очереди

Алгоритмы, основанные на квантовании



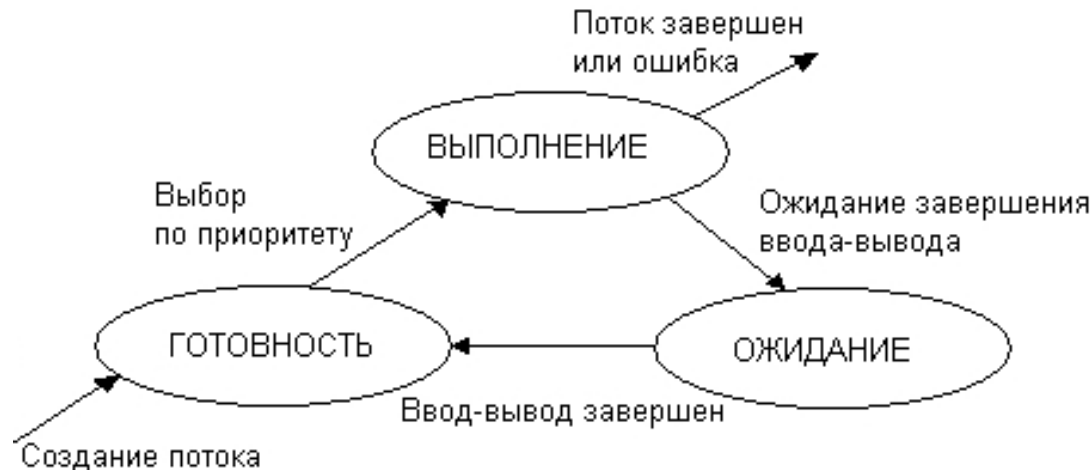
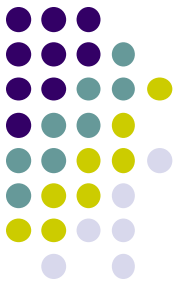
Квантование с предпочтением потоков, интенсивно обращающихся к вводу-выводу

Алгоритмы, основанные на приоритетах



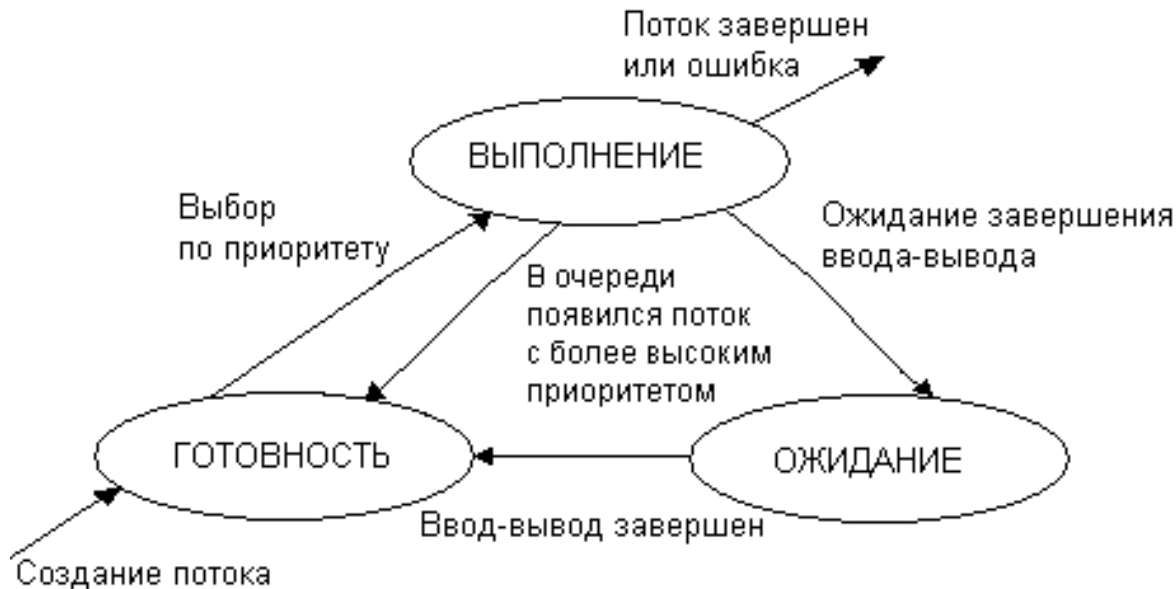
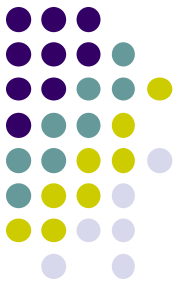
- Приоритеты:
 - динамические
 - фиксированные
- Алгоритмы приоритетного планирования:
 - обслуживание с относительными приоритетами
 - обслуживание с абсолютными приоритетами

Алгоритмы, основанные на приоритетах



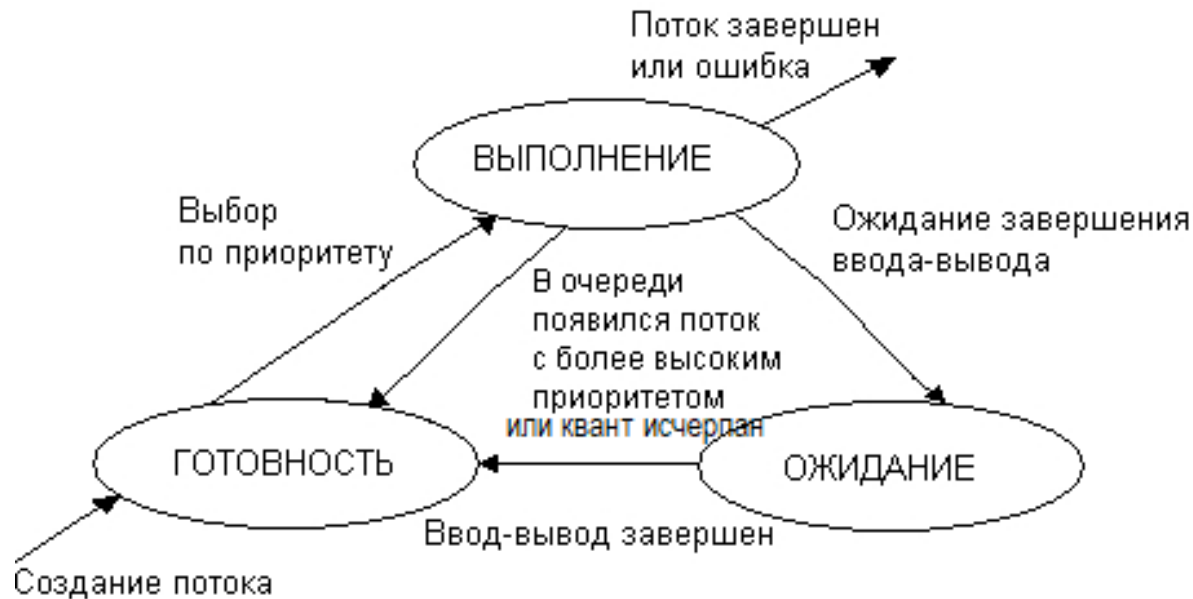
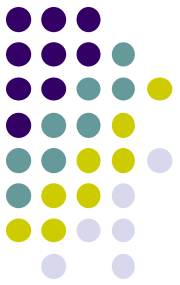
Граф состояний потоков системе с относительными приоритетами

Алгоритмы, основанные на приоритетах



Граф состояний потоков в системе с абсолютными приоритетами

Смешанные алгоритмы планирования

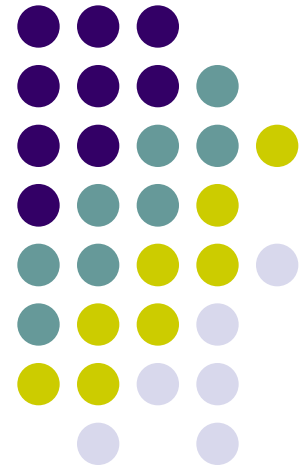


Граф состояний в системах с планированием на основе приоритетов и квантования

Синхронизация ПОТОКОВ

Операционные системы

Лекция 5

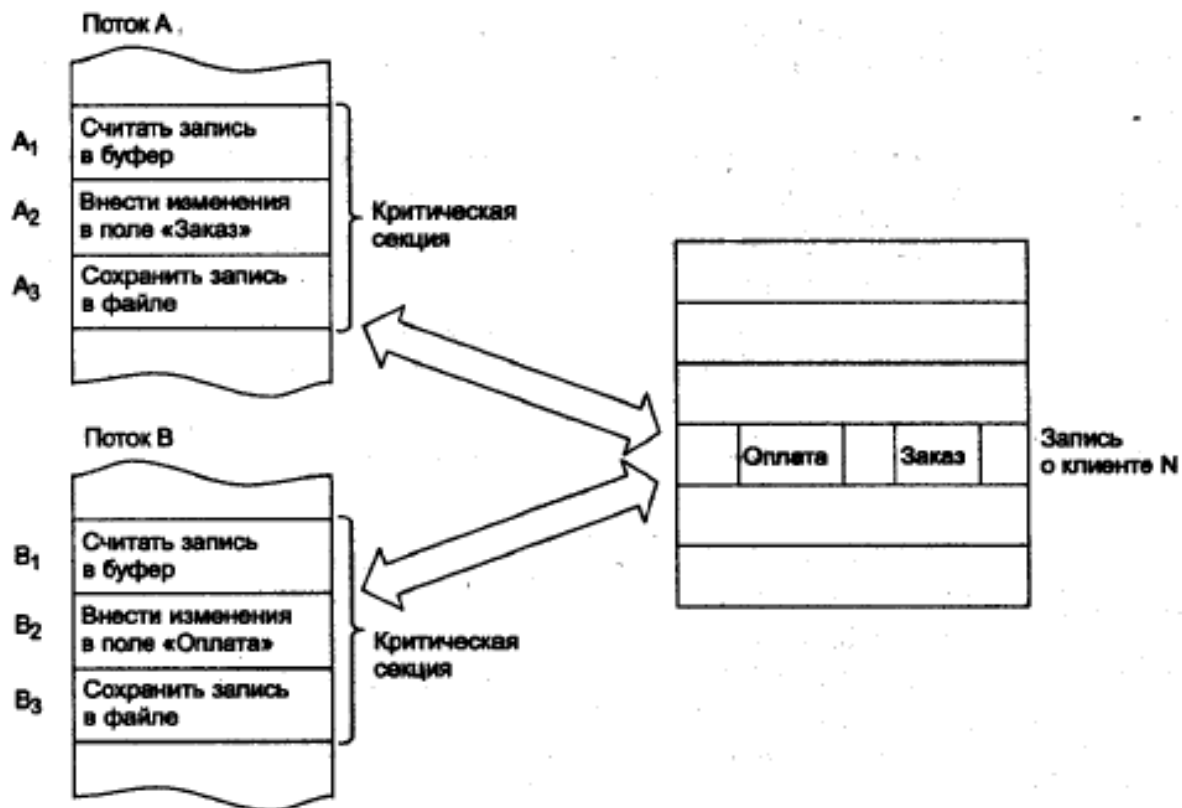


План

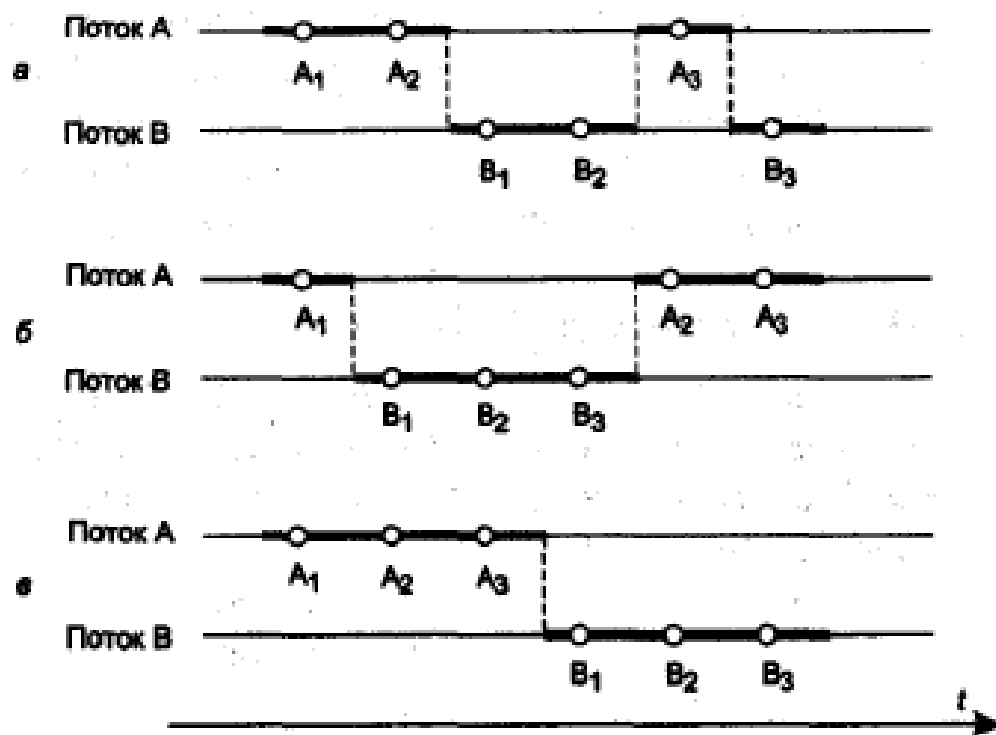


- Гонки
- Тупики
- Сравнение объектов синхронизации

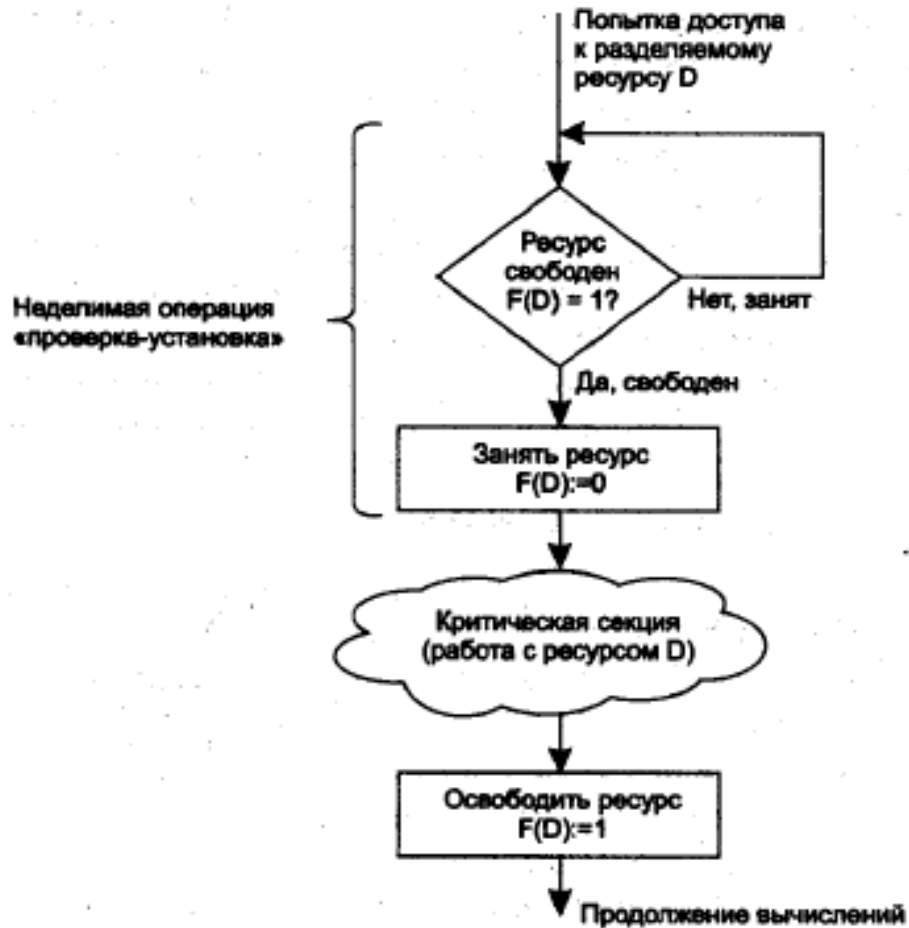
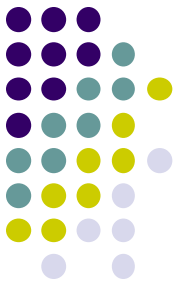
Гонки



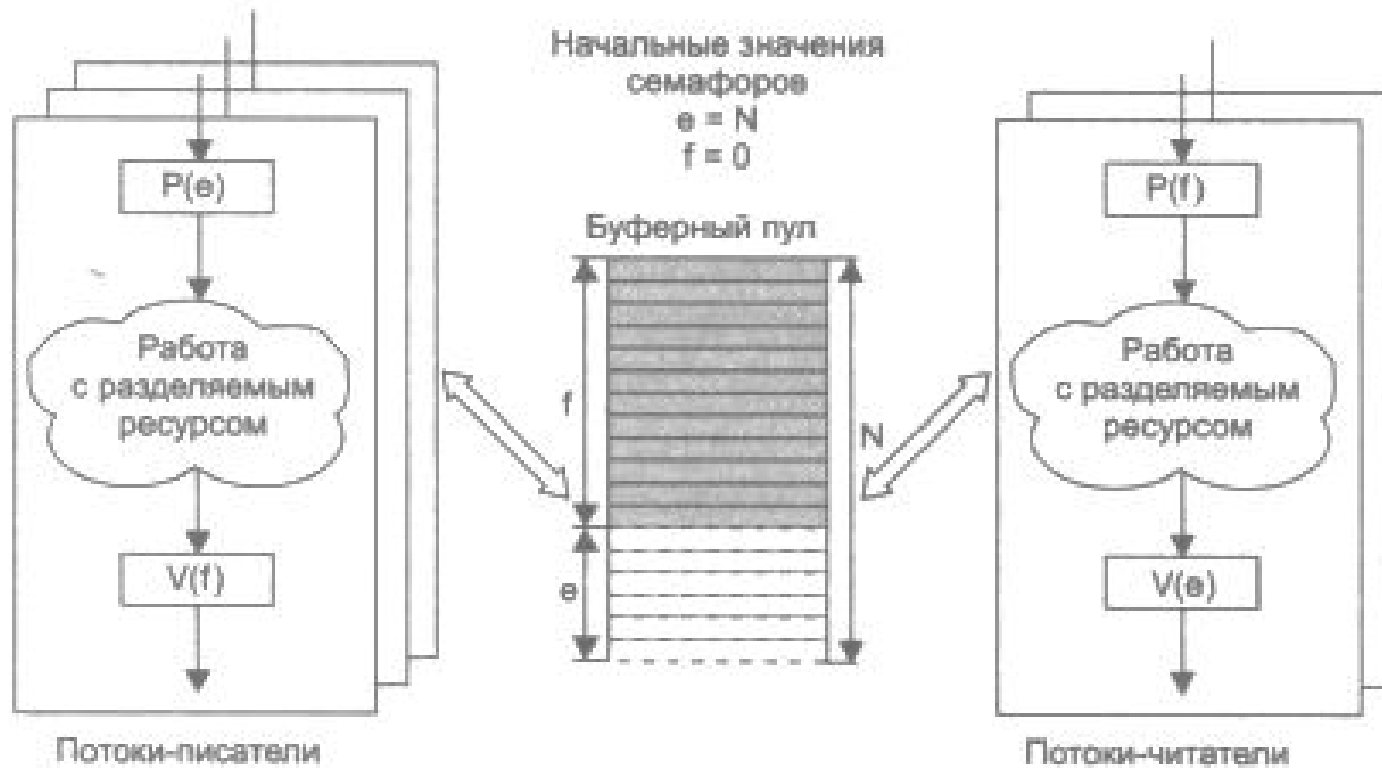
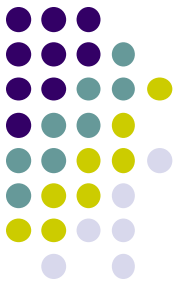
Гонки



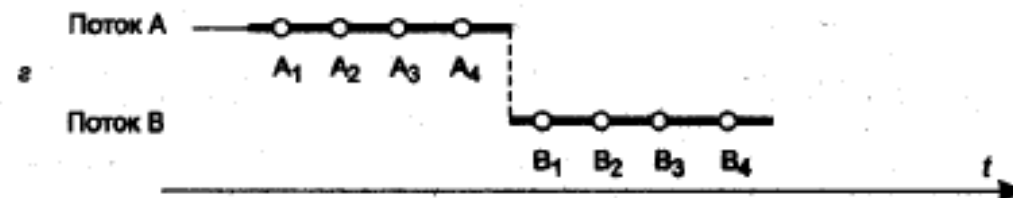
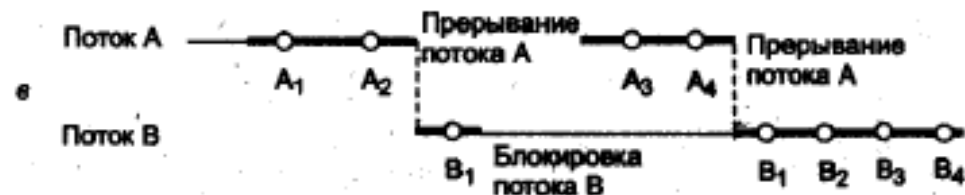
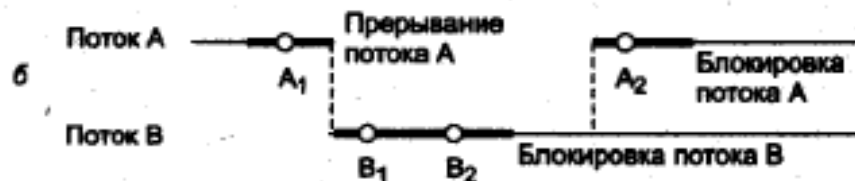
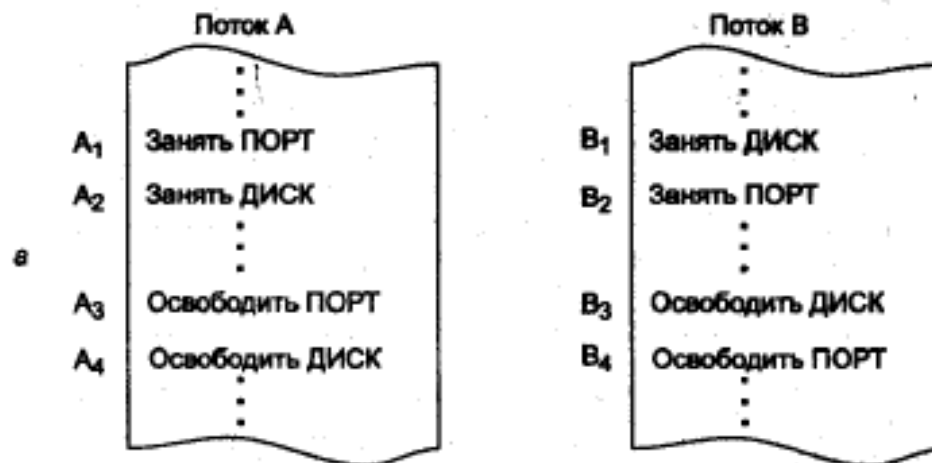
Критические секции



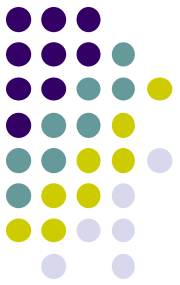
Семафоры



Тупики



Сравнение объектов синхронизации

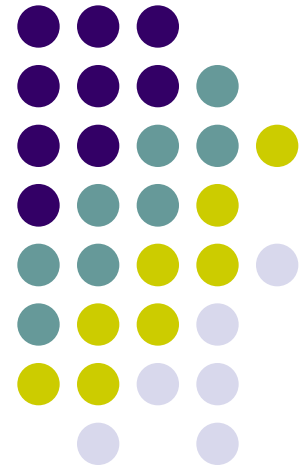


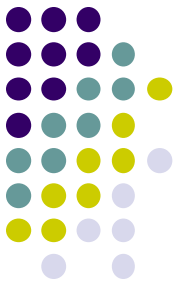
Объект	Относительная скорость	Доступ нескольких потоков	Подсчет числа обращений к ресурсу
Критическая секция	быстро	нет	нет (эксклюзивный доступ)
Мьютекс	медленно	да	нет (эксклюзивный доступ)
Семафор	медленно	да	автоматически
Событие	медленно	да	да

Управление памятью

Операционные системы

Лекция 6

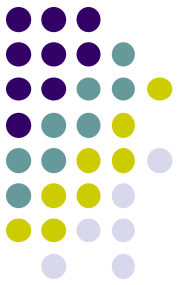




План

- Иерархия памяти
- Типы адресов
- Функции операционной системы по управлению памятью
- Подходы к преобразованию виртуальных адресов в физические
- Виртуализация памяти

Иерархия памяти



Обычное время доступа

1 нс

2 нс

10 нс

10 мс

100 с



Регистры

Кэш

Основная память

Магнитный диск

Магнитная лента

Обычный объем

<1 Кбайт

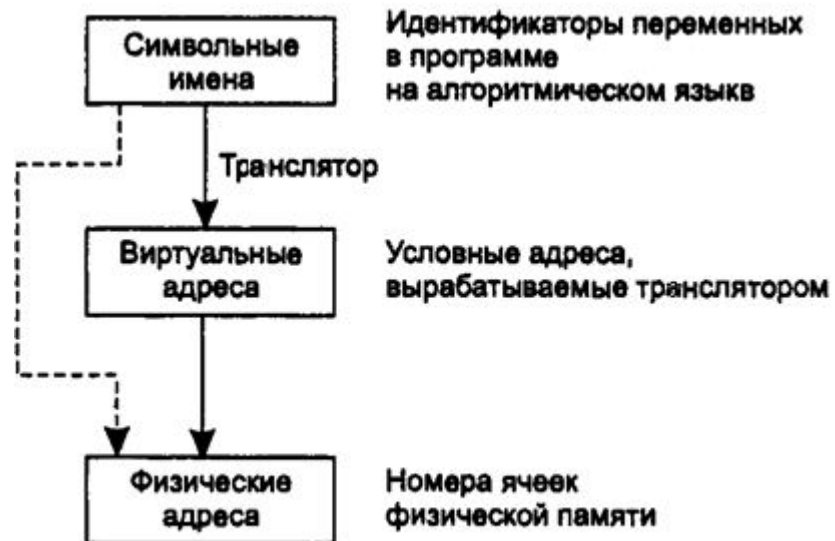
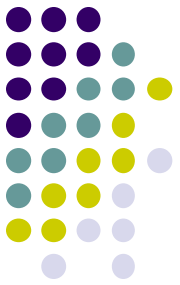
4 Мбайт

512–2048 Мбайт

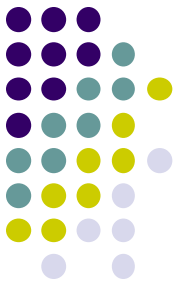
200–1000 Гбайт

400–800 Гбайт

Типы адресов

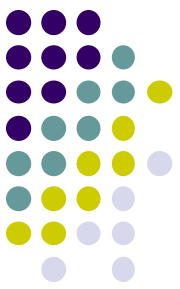


Функции операционной системы по управлению памятью

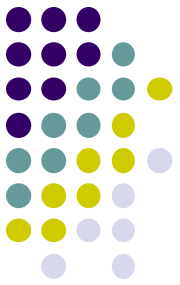


- отслеживание свободной и занятой памяти
- выделение памяти процессам и ее освобождение
- вытеснение процессов из оперативной памяти на диск и возвращение в оперативную память
- настройка адресов программы на конкретную область физической памяти
- динамическое выделение памяти процессам
- дефрагментация освобожденной динамической памяти
- выделение памяти для создания служебных структур операционной системы
- защита памяти

Подходы к преобразованию виртуальных адресов в физические

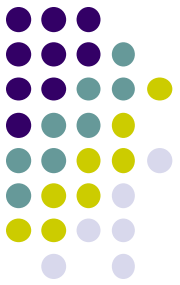


- Переход от виртуальных адресов к физическим может осуществляться двумя способами:
 - Загрузка совместно с заменой виртуальных адресов физическими
 - Динамическое преобразование виртуальных адресов



Виртуализация памяти

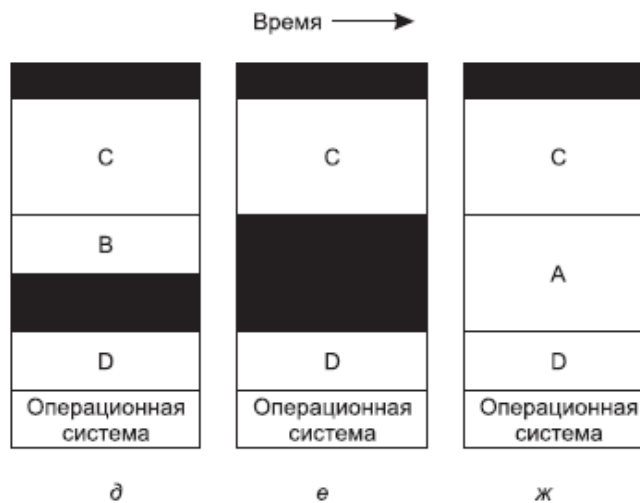
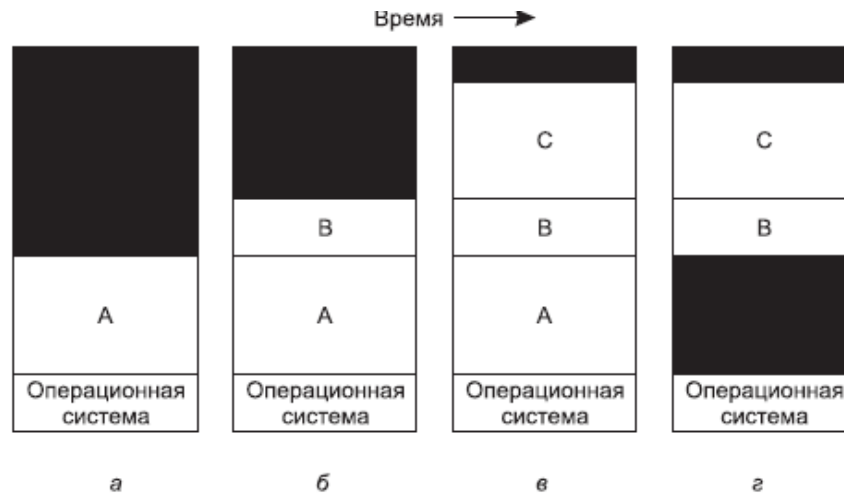
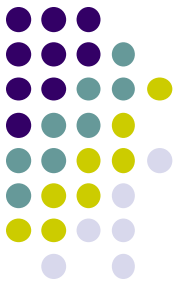
- Виртуализация включает решение следующих задач:
 - размещение данных в запоминающих устройствах разного типа
 - выбор образов процессов или их частей для перемещения
 - перемещение данных между памятью и диском
 - преобразование виртуальных адресов в физические



Виртуализация памяти

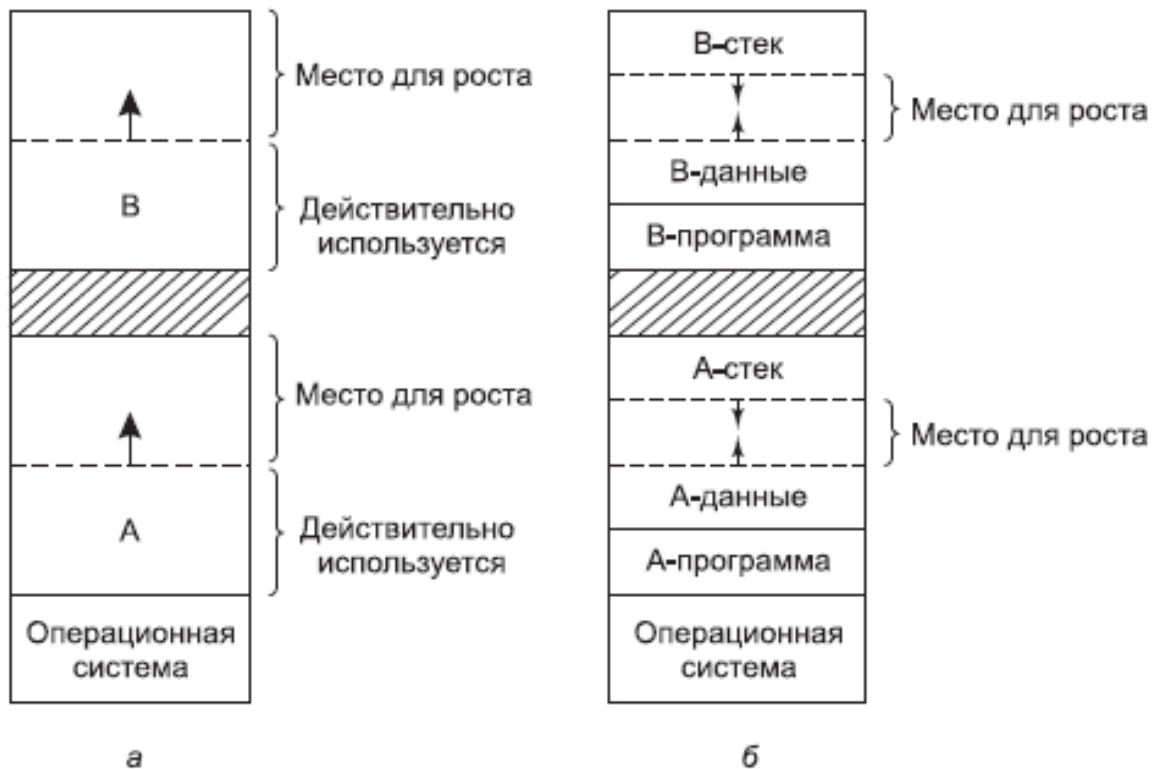
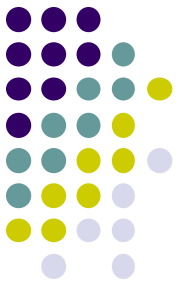
- Виртуализация памяти может быть осуществлена на основе двух подходов:
 - Свопинга (swapping)
 - Механизма виртуальной памяти (virtual memory)

СВОПИНГ



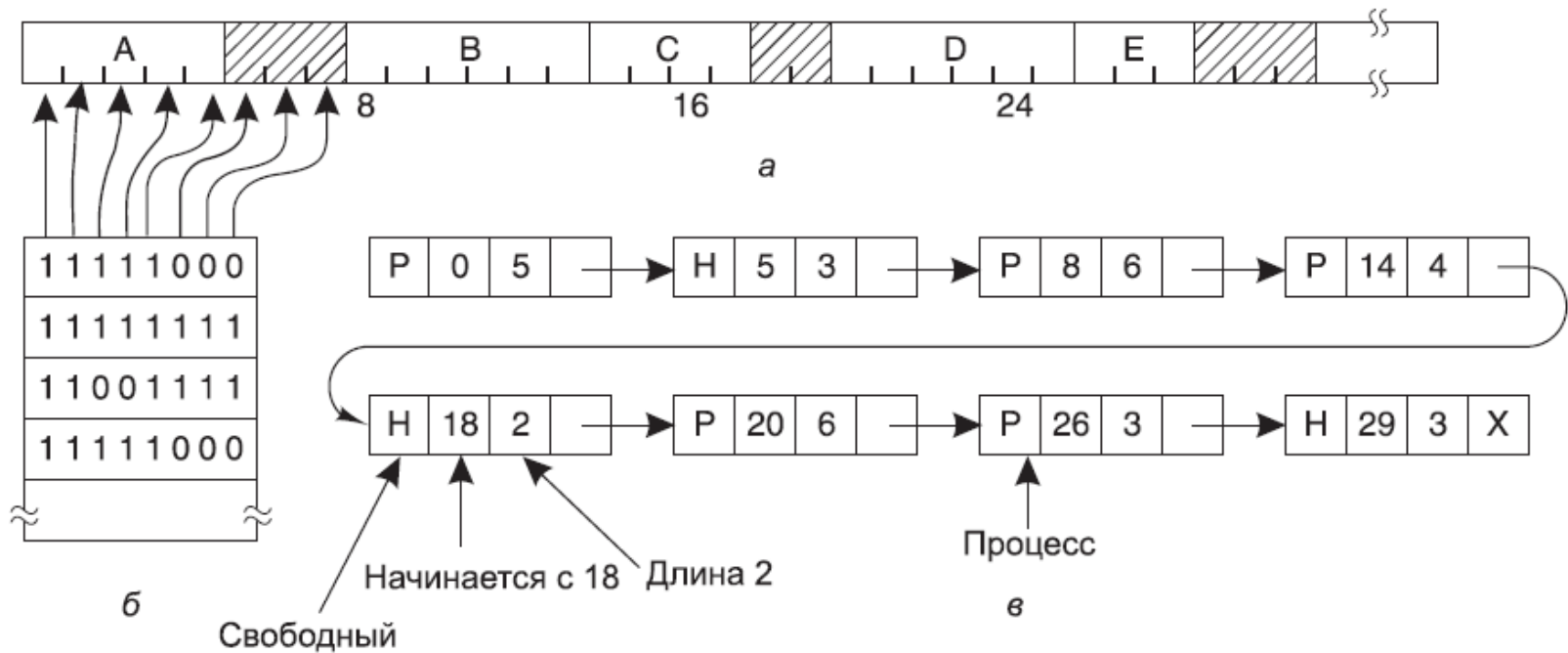
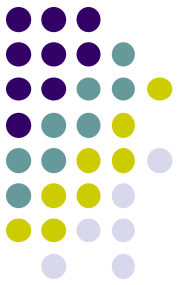
Изменения в выделении
памяти по мере появления
процессов в памяти и
выгрузки их из нее

СВОПИНГ



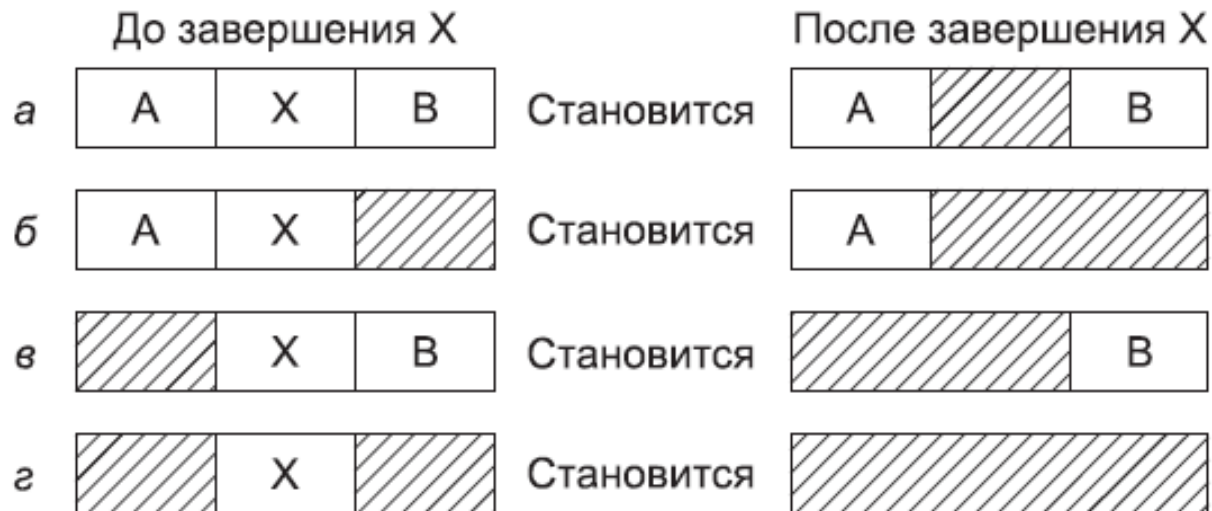
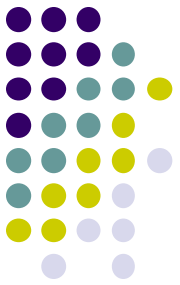
Выделение памяти: а – под разрастающийся сегмент данных;
б – под разрастающийся стек и сегмент данных

Управление свободной памятью

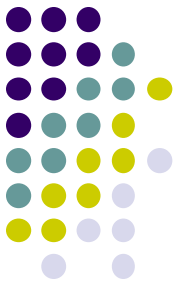


Часть памяти с пятью процессами и тремя свободными пространствами, единичные блоки памяти разделены вертикальными штрихами: а – заштрихованные области являются свободными; б – соответствующая битовая матрица; в – та же информация в виде списка

Управление свободной памятью



Четыре комбинации соседей для завершающегося процесса X



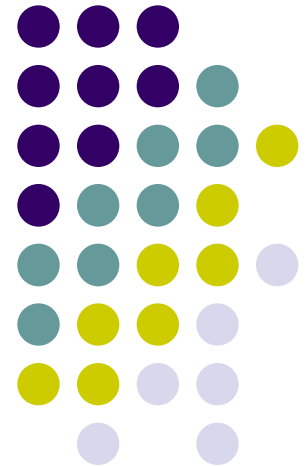
Виртуальная память

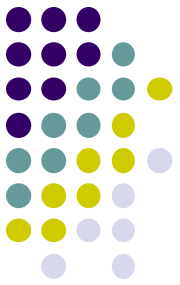
- У каждой программы имеется собственное адресное пространство, разбитое на части
- Между ОП и диском перемещаются части адресного пространства.
- Для временного хранения вытесненных участков используется *страничный файл* (page file, paging file) или *файл свопинга*

Виртуальная память

Операционные системы

Лекция 7

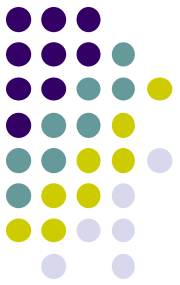




План

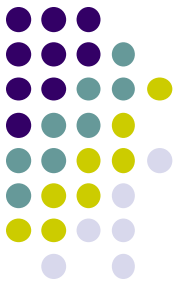
- Страничная организация памяти
 - Таблицы страниц
 - Ускорение работы страничной организации памяти
 - Таблицы страниц для большого объема памяти
- Сегментация
 - Реализация чистой сегментации
 - Сегментация со страничной организацией памяти: система Intel x86

Страничная организация памяти



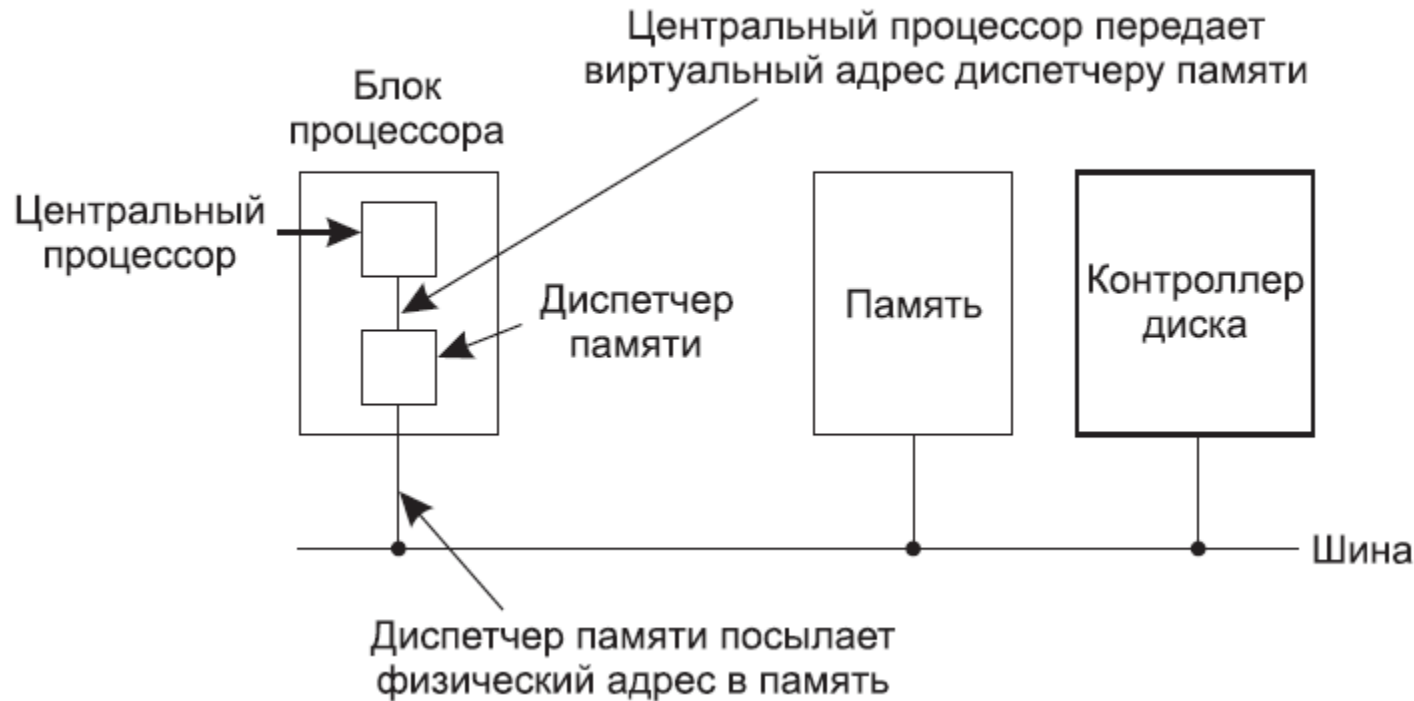
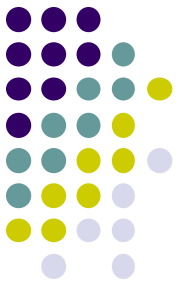
- Большинство систем виртуальной памяти используют технологию под названием **страничная организация памяти** (paging)
- При использовании виртуальной памяти виртуальные адреса поступают в **диспетчер памяти** (Memory Management Unit (**MMU**)), который отображает виртуальные адреса на адреса физической памяти

Страничная организация памяти



- Виртуальное адресное пространство состоит из блоков фиксированного размера, называемых **страницами**.
- Соответствующие блоки в физической памяти называются **страничными блоками**.
- Страницы и страничные блоки имеют одинаковые размеры.

Страничная организация памяти

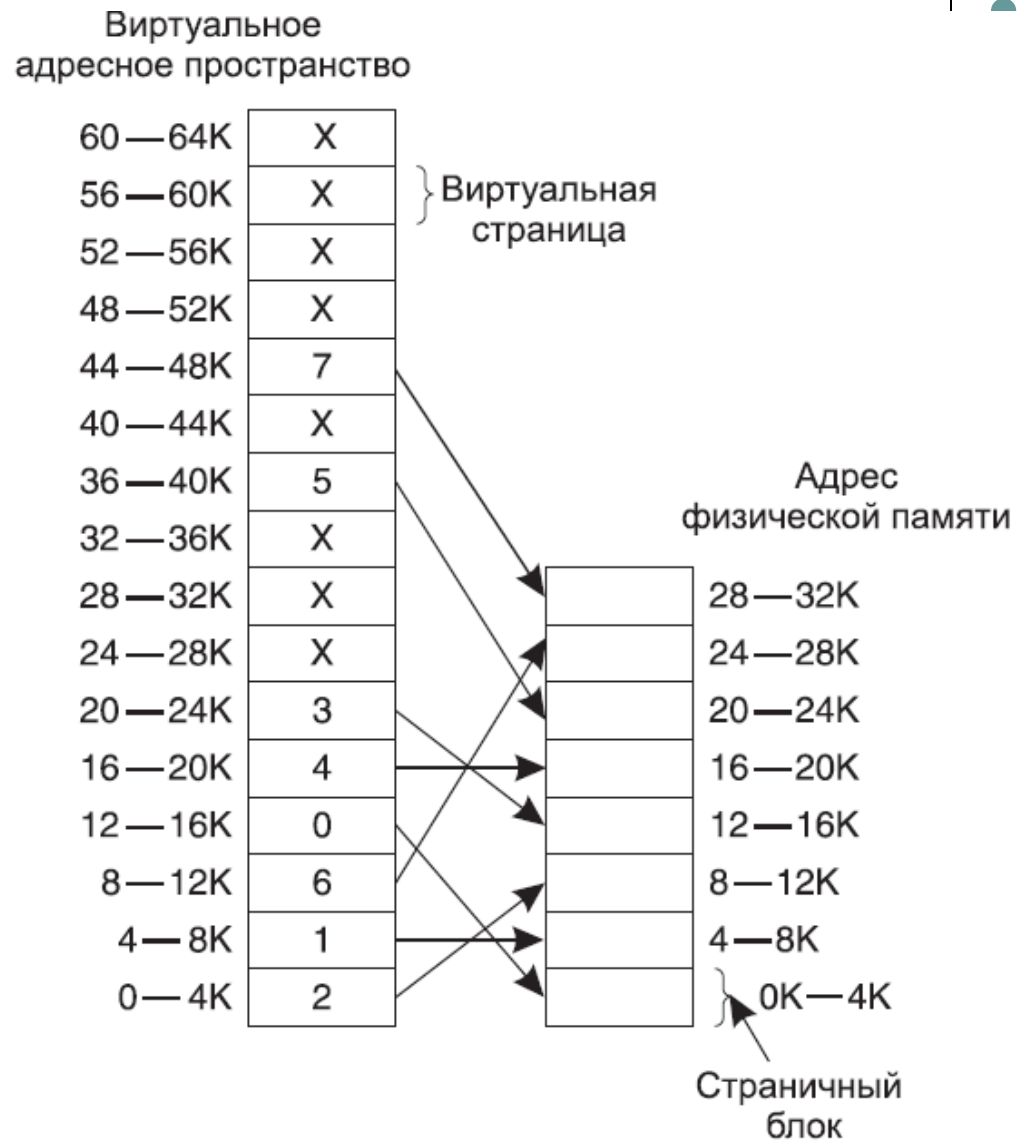


- Расположение и предназначение диспетчера памяти

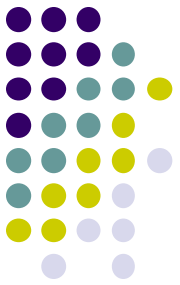
Страничная организация памяти



- Связь между виртуальными адресами и адресами физической памяти, получаемая с помощью таблицы страниц



Страничная организация памяти



- Реальное оборудование отслеживает присутствие конкретных страниц в физической памяти за счет **бита присутствия-отсутствия**.
- Если программа ссылается на неотображаемые адрес, возникает системное прерывание - **ошибка отсутствия страницы** (page fault).

Страничная организация памяти

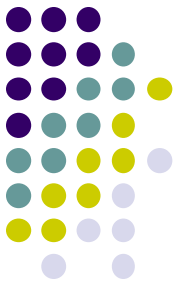


- Преобразование диспетчером памяти виртуального адреса в физический для 16 страниц по 4 Кбайт



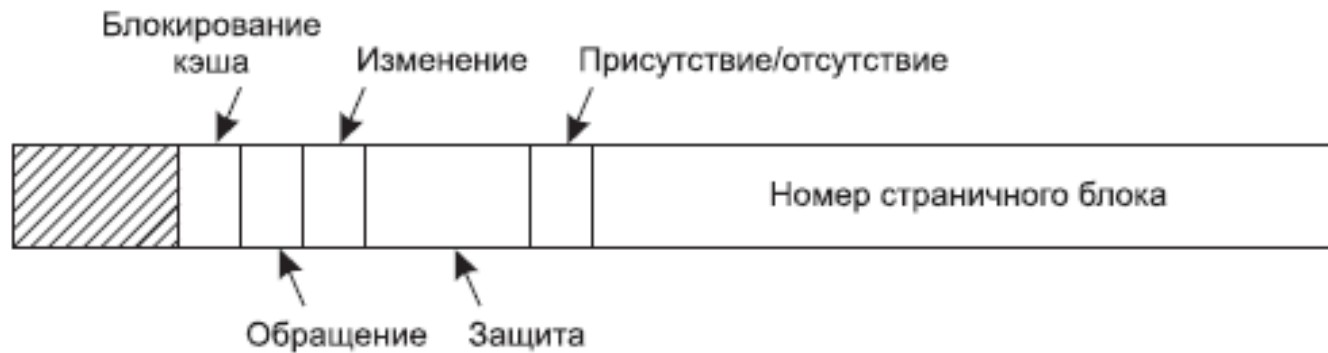
Физический адрес на выходе (24580)

Виртуальный адрес на входе (8196)



Таблицы страниц

- Типичная запись таблицы страниц



Ускорение работы страничной организации памяти



1. Отображение виртуального адреса на физический должно быть быстрым.
2. Если пространство виртуальных адресов слишком обширное, таблица страниц будет иметь весьма солидный размер.

Интенсивному чтению подвергается лишь небольшая часть записей таблицы страниц, а остальная часть практически не используется.

Ускорение работы страничной организации памяти



- Найденное решение состояло в оснащении компьютеров небольшим устройством для отображения виртуальных адресов на физические без просмотра таблицы страниц.
- **буфер быстрого преобразования адреса** (Translation Lookaside Buffer (TLB)), которое иногда еще называют **ассоциативной памятью**

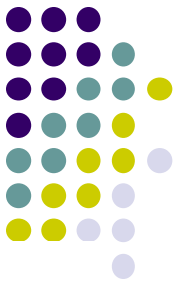
Ускорение работы страничной организации памяти



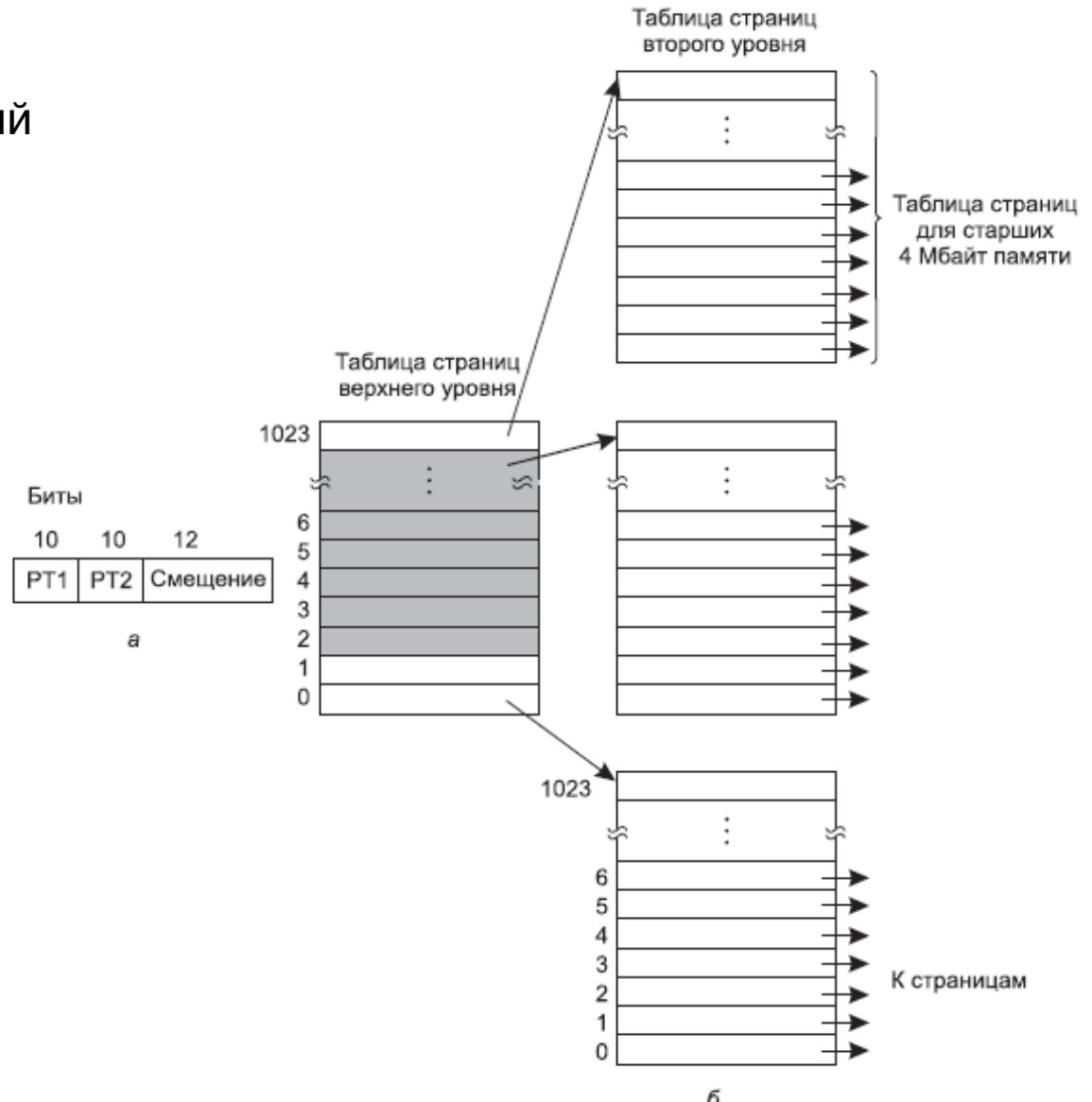
- Буфер быстрого преобразования адреса, используемый для ускорения страничного доступа к памяти

Задействована	Виртуальная страница	Изменена	Защищена	Страничный блок
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

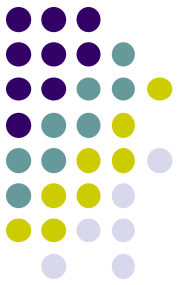
Таблицы страниц для большого объема памяти



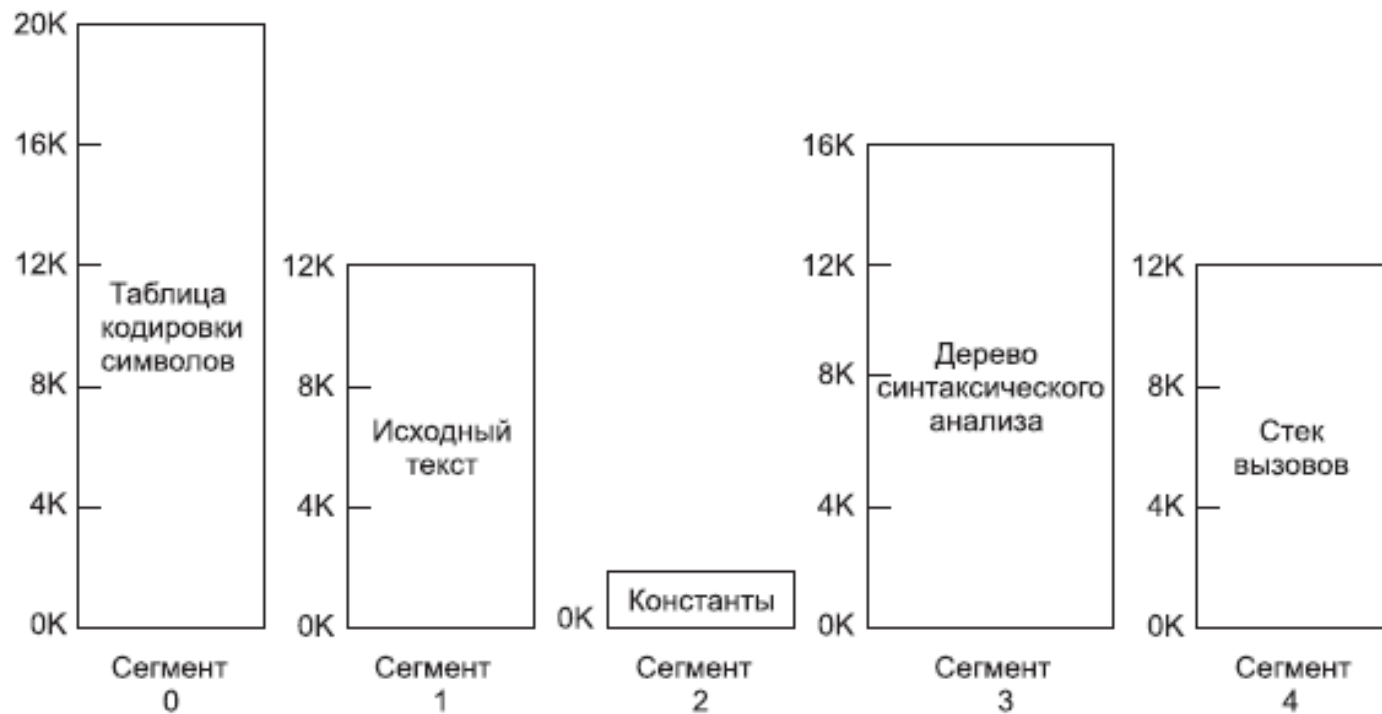
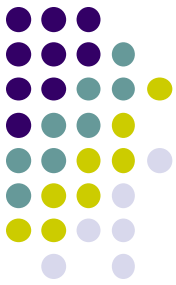
Многоуровневая таблица
страниц: a — 32-разрядный
адрес с двумя полями
таблиц страниц; b —
двухуровневая таблица
страниц



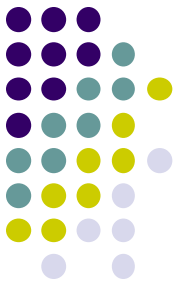
Сегментация



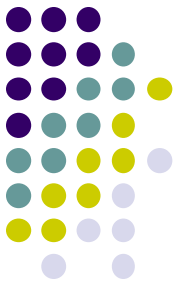
Сегментация



Сегментация



Преобразование логического адреса при сегментной организации памяти

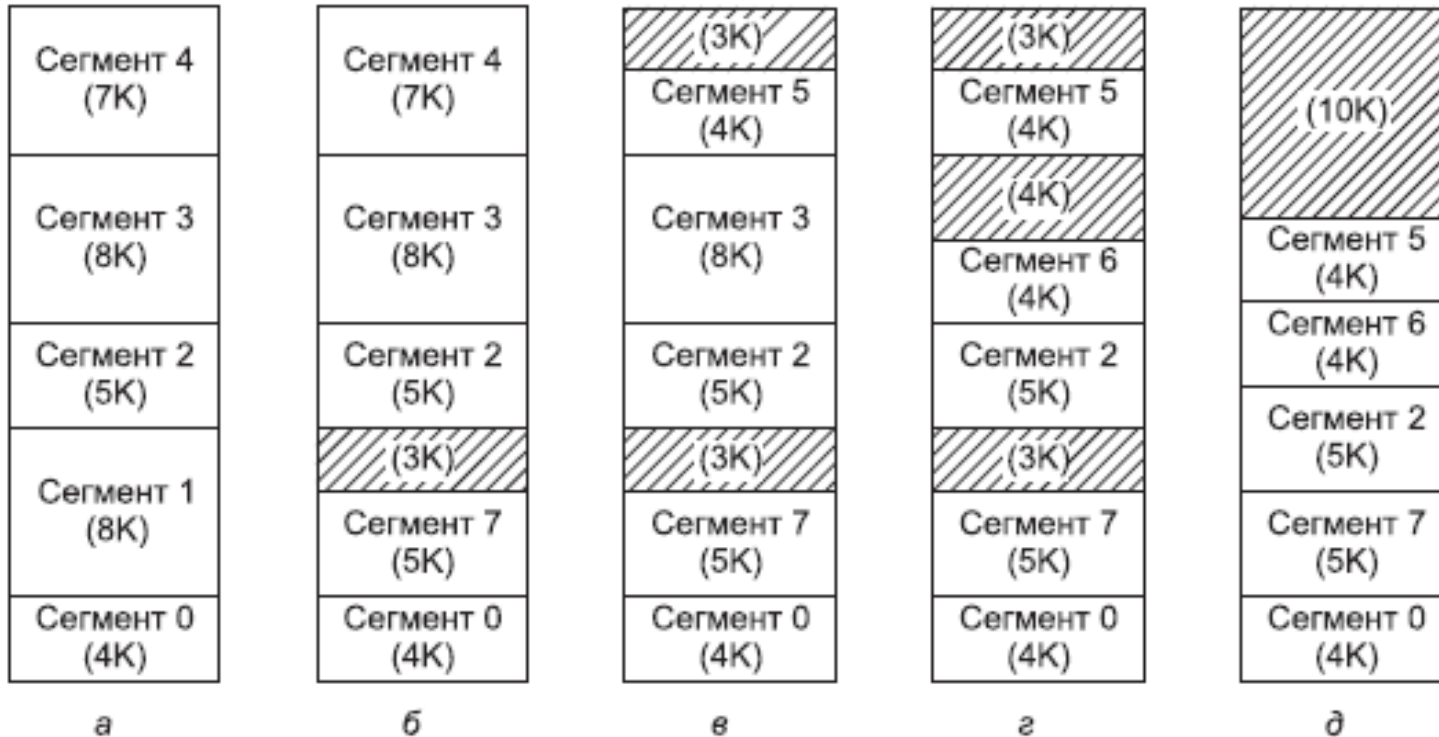
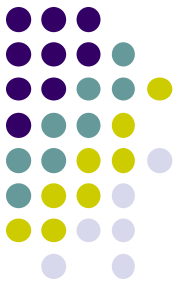


Сегментация

- Сравнение страничной организации памяти и сегментации

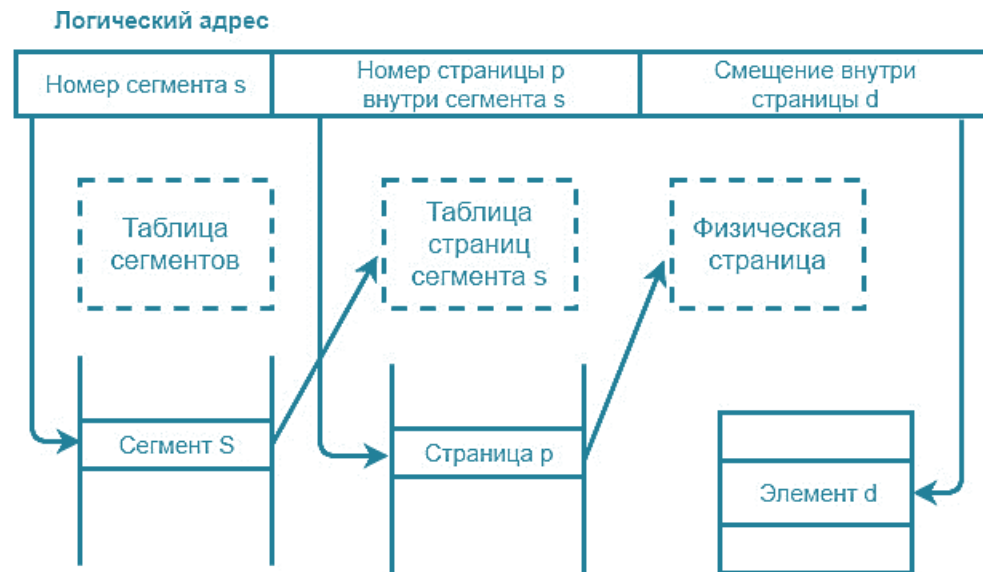
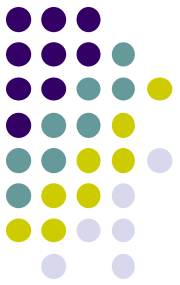
Вопрос	Страничная организация	Сегментация
Нужно ли программисту знать, что используется именно эта технология?	Нет	Да
Сколько имеется линейных адресных пространств?	1	Много
Может ли все адресное пространство превысить размер физической памяти?	Да	Да
Могут ли различаться и быть отдельно защищены процедуры и данные?	Нет	Да
Можно ли без особого труда предоставить пространство таблицам, изменяющим свой размер?	Нет	Да
Облегчается ли для пользователей совместный доступ к процедурам?	Нет	Да

Реализация чистой сегментации



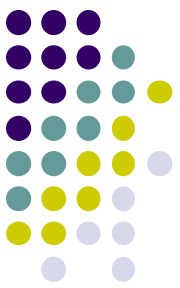
- Физическая память: а — г — нарастание внешней фрагментации; д — избавление от внешней фрагментации за счет уплотнения

Сегментация со страничной организацией памяти

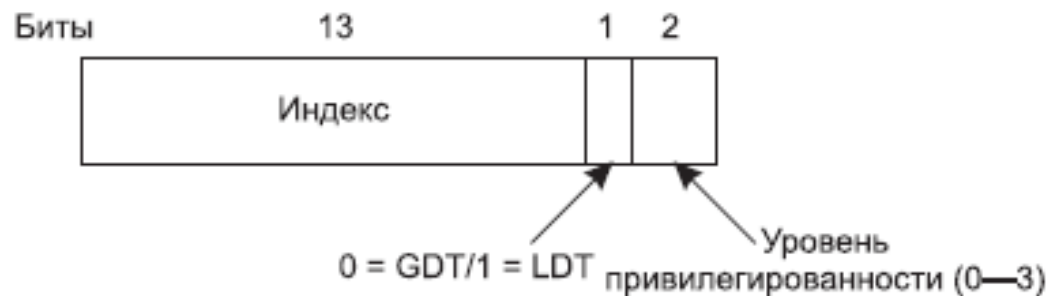


Упрощенная схема формирования физического адреса при сегментно-страничной организации памяти

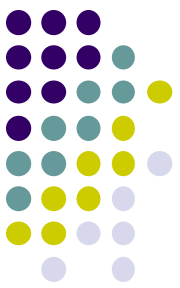
Сегментация со страничной организацией памяти: система Intel x86



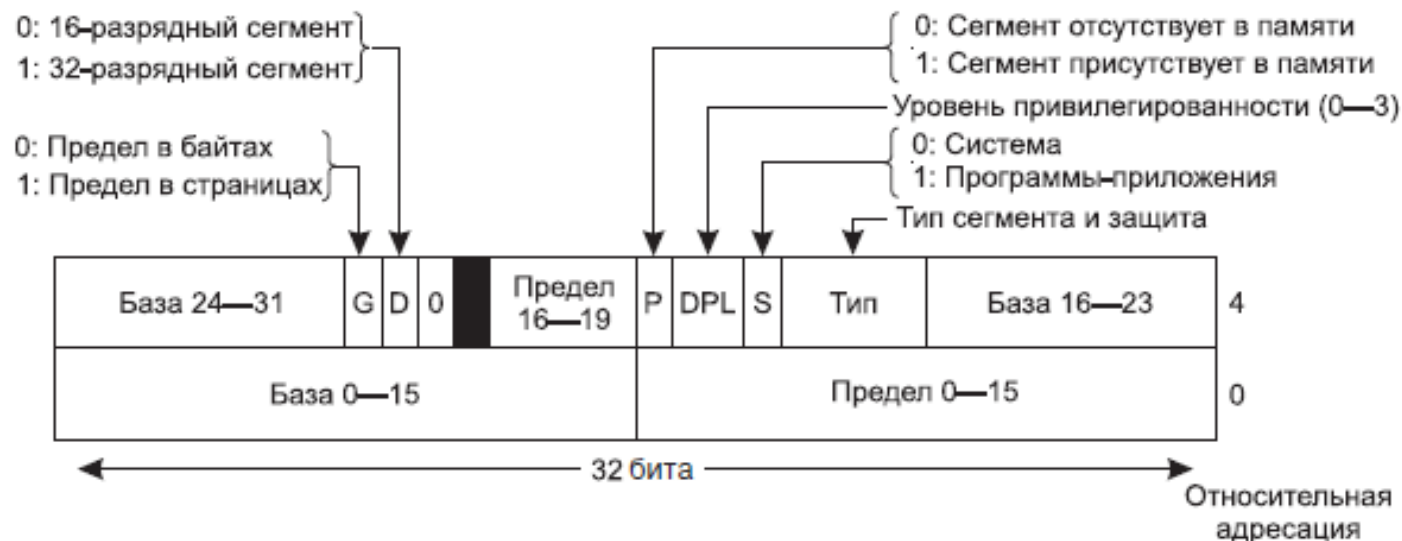
- Селектор системы Pentium



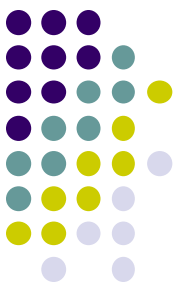
Сегментация со страничной организацией памяти: система Intel x86



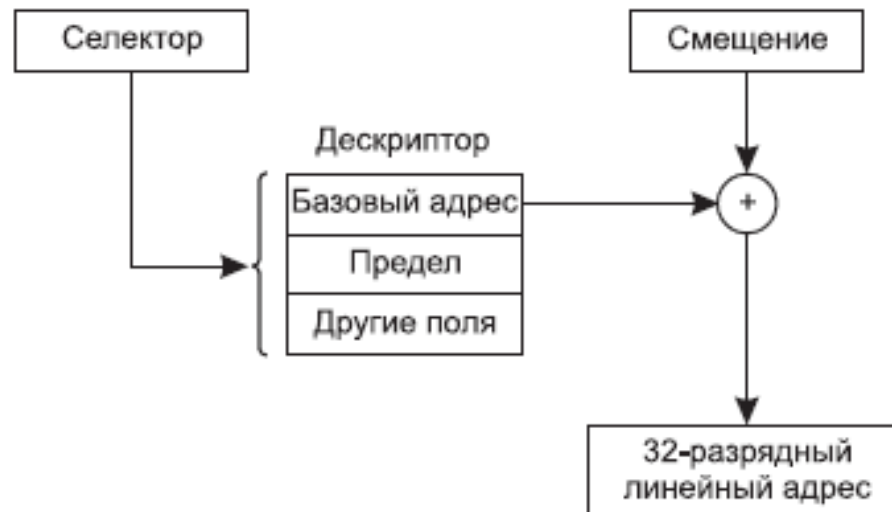
- Дескриптор сегмента кода в системе Pentium. Сегменты данных имеют незначительные отличия



Сегментация со страничной организацией памяти: система Intel x86



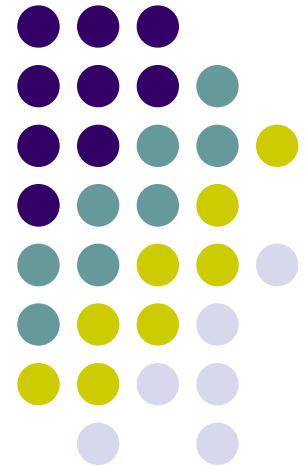
- Преобразование пары «селектор — смещение» в линейный адрес



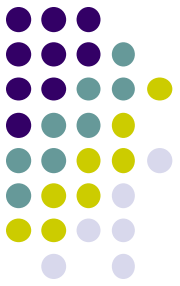
Ввод и вывод информации

Операционные системы

Лекция 8

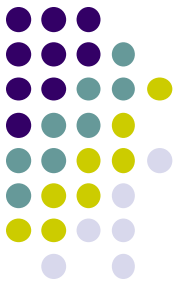


План



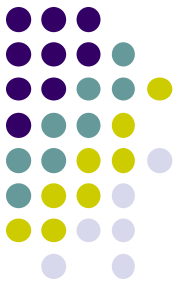
- Основные задачи управления вводом-выводом
- Физическая организация устройств ввода-вывода
- Обобщенная структура подсистемы ввода-вывода

Основные задачи управления ВВОДОМ-ВЫВОДОМ



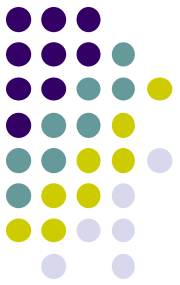
- организация параллельной работы процессора и устройств ввода-вывода
- согласование скоростей работы процессора, оперативной памяти и устройств ввода-вывода
- разделение устройств ввода-вывода между процессами
- обеспечение удобного логического интерфейса к устройствам ввода-вывода

Физическая организация устройств ввода-вывода



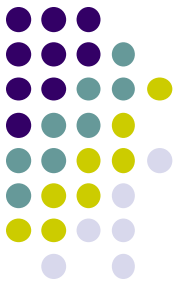
- Типы устройств ввода-вывода:
 - блок-ориентированные устройства
 - байт-ориентированные устройства
- Внешнее устройство обычно состоит из механического и электронного компонента.

Обобщенная структура подсистемы ввода-вывода



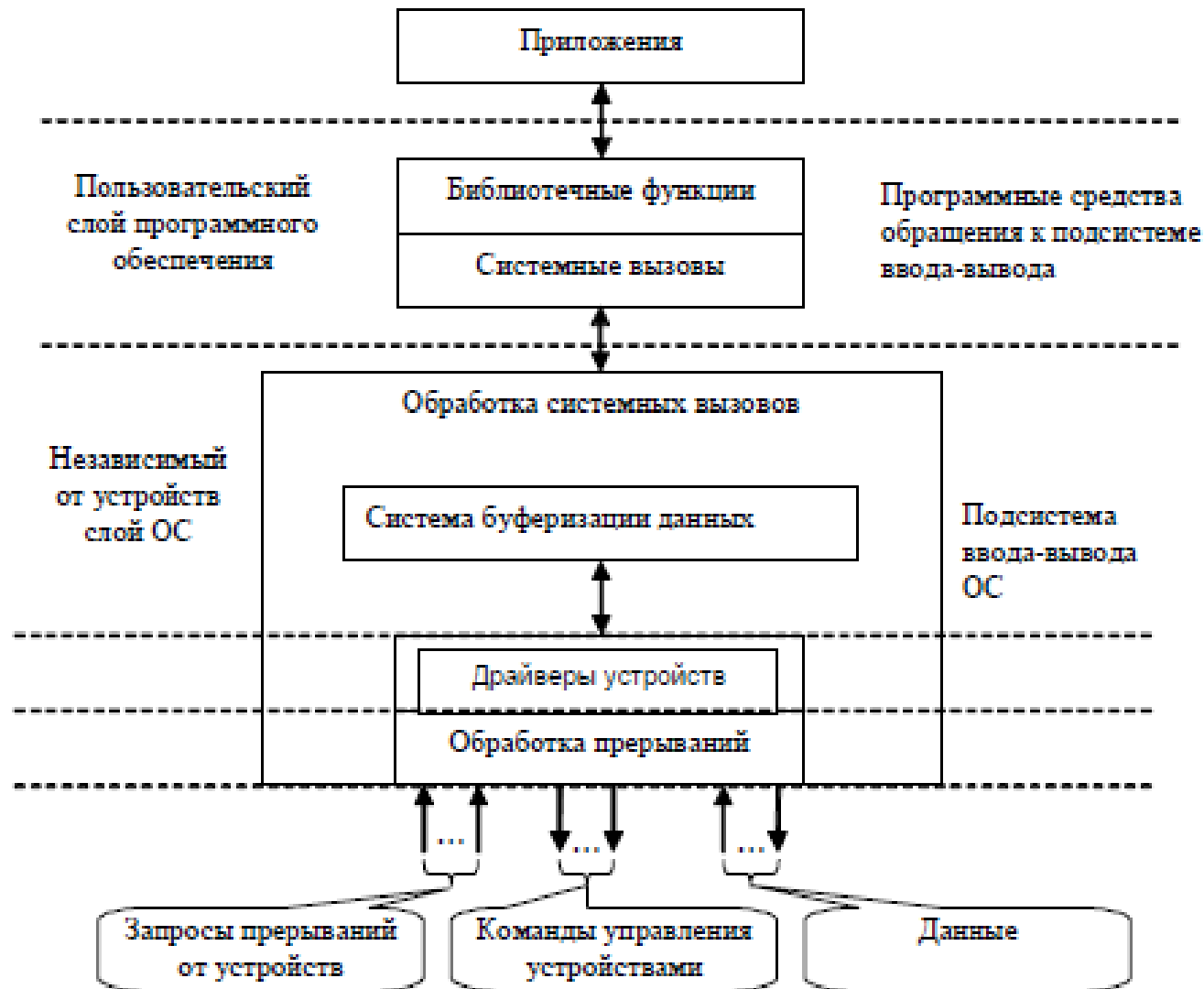
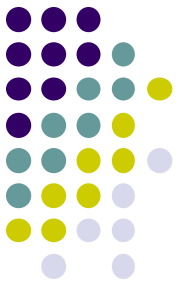
- Основная идея организации программного обеспечения ввода-вывода состоит в разбиении его на несколько уровней

Обобщенная структура подсистемы ввода-вывода



- Программное обеспечение ввода-вывода разделяется на четыре слоя:
 - обработка прерываний
 - драйверы устройств
 - независимый от устройств слой операционной системы
 - пользовательский слой программного обеспечения

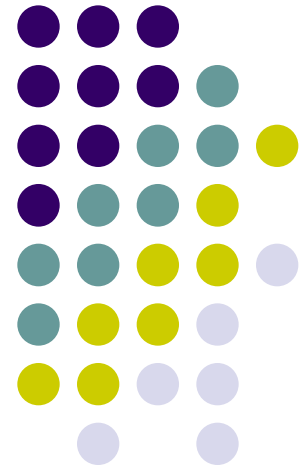
Обобщенная структура подсистемы ввода-вывода

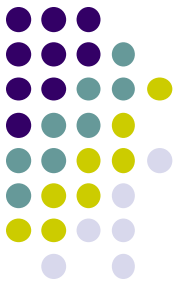


Файловые системы

Операционные системы

Лекция 9

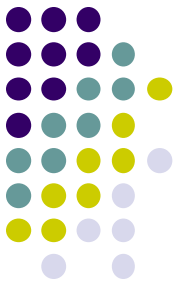




План

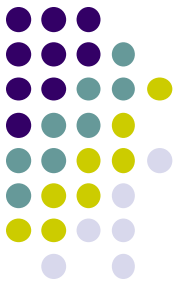
- Функции файловой системы
- Типы и атрибуты файлов
- Имена файлов
- Монтирование
- Подходы к логической организации файлов
- Физическая организация файловой системы

Функции файловой системы



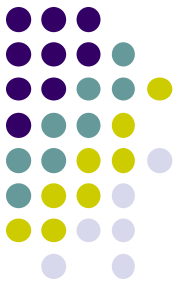
- *Файл* –
- *Файловая система* –
- Основные функции файловой системы:
 - предоставление пользователю логической модели для работы с файлами
 - отображение этой модели на физическую организацию внешнего устройства

Типы и атрибуты файлов

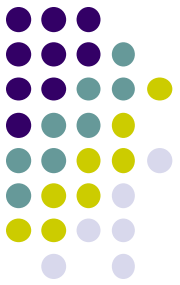


- Типы файлов:
 - обычные файлы
 - файлы-каталоги
 - специальные файлы
 - СИМВОЛЬНЫЕ СВЯЗИ
 - именованные конвейеры
 - файлы, отображаемые на память

Типы и атрибуты файлов

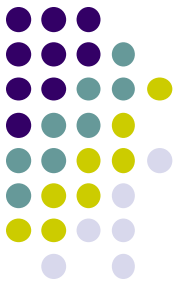


- возможные атрибуты:
 - тип файла
 - владелец файла (NTFS, Unix)
 - создатель файла
 - пароль для доступа к файлу (NTFS, Unix)
 - информация о возможности доступа, т.е. права доступа (NTFS, Unix)
 - времена создания; последнего доступа и последнего изменения
 - текущий размер файла



Типы и атрибуты файлов

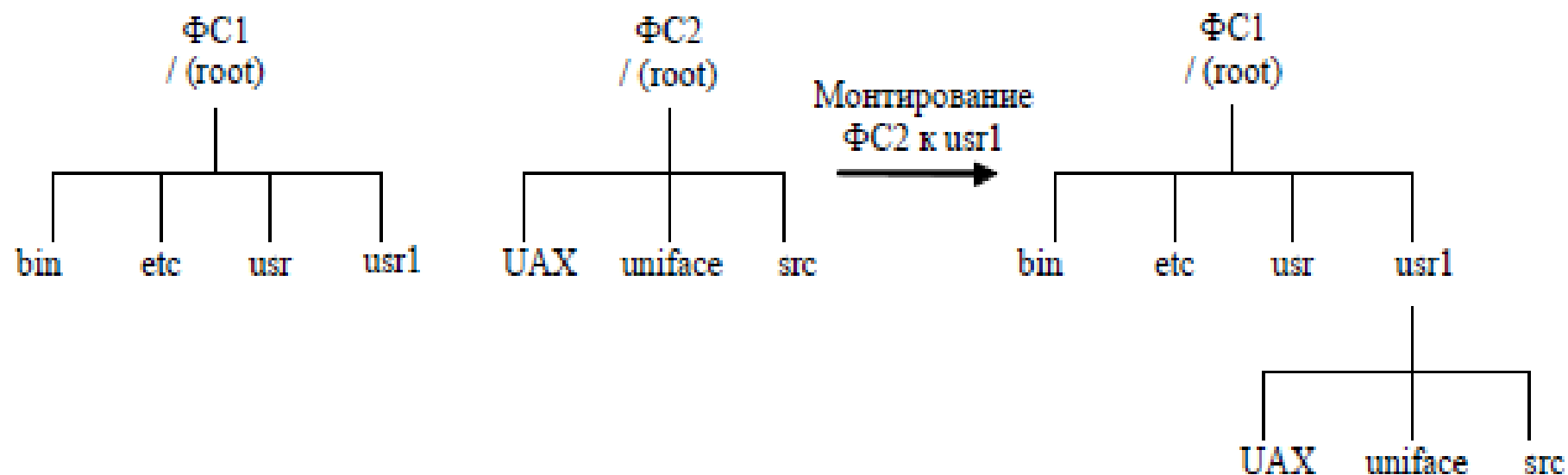
- максимальный размер файла;
- признак «только для чтения»
- признак «скрытый файл»(FAT, NTFS)
- признак «системный файл»
- признак «архивный файл»
- признак «двоичный/символьный» (Unix)
- признак «временный» – удалить после завершения процесса (Unix)
- признак блокировки (NTFS, Unix)
- длина записи
- указатель на ключевое поле в записи
- длина ключа



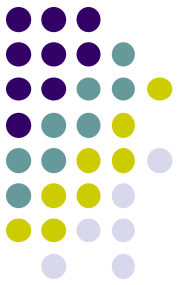
Имена файлов

- Все типы файлов имеют символьные имена.
- В иерархически организованных файловых системах обычно используются следующие типы имен файлов:
 - простое (короткое) символьное имя
 - полное (составное) символьное имя
 - относительное символьное имя
 - уникальное имя (числовой идентификатор)

Монтирование

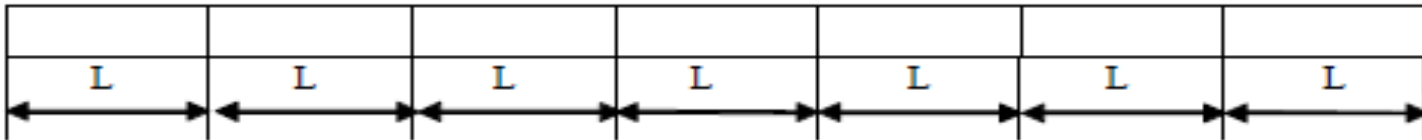
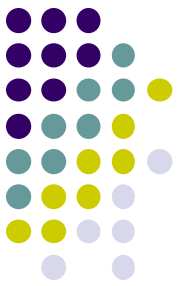


Подходы к логической организации файлов

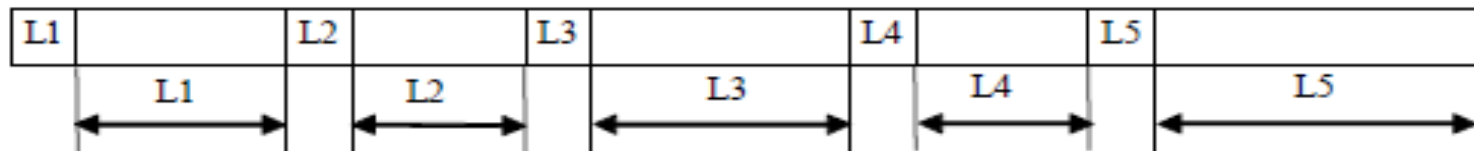


- Способы доступа к записям файла:
 - последовательный
 - прямой

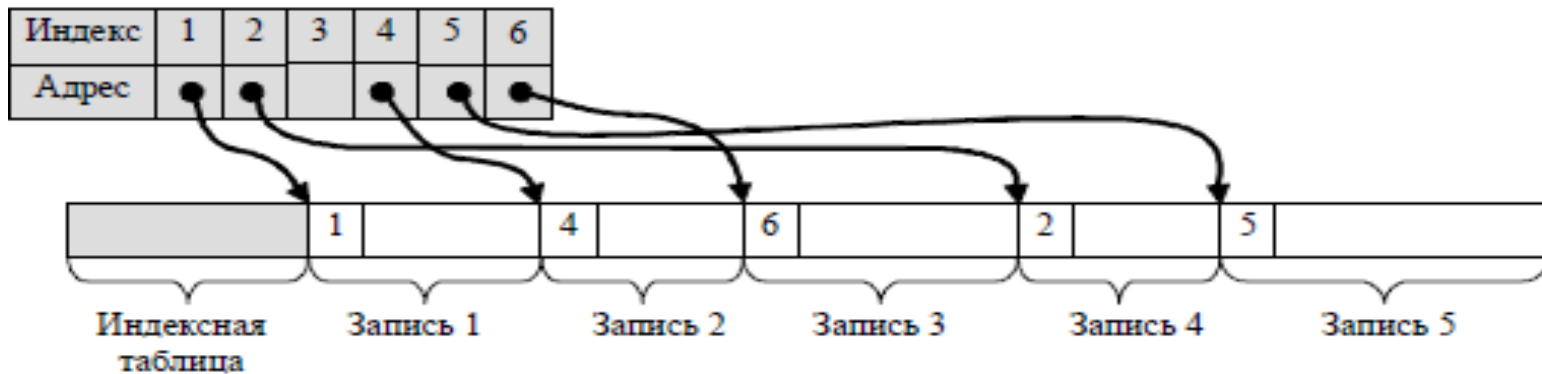
Подходы к логической организации файлов



Последовательная организация с записями фиксированной длины

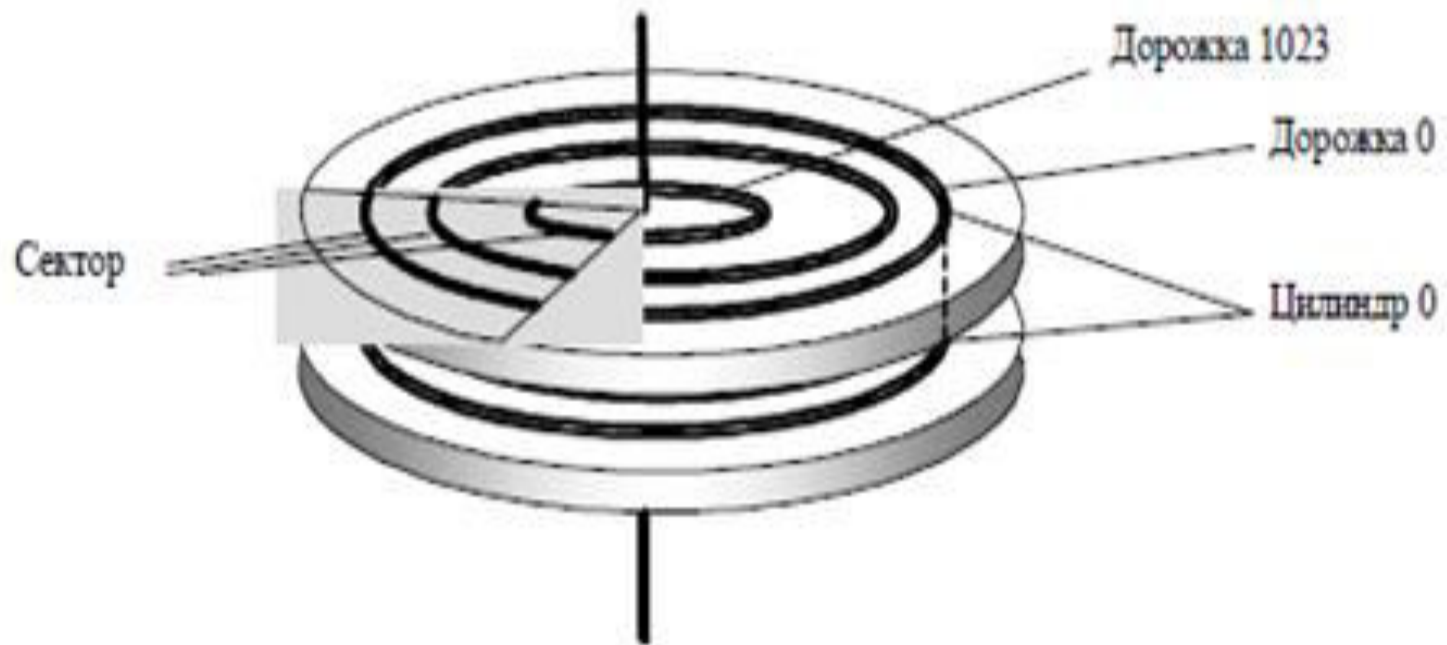
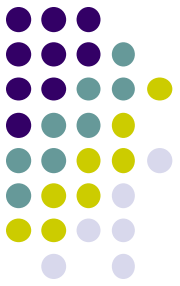


Последовательная организация с записями переменной длины

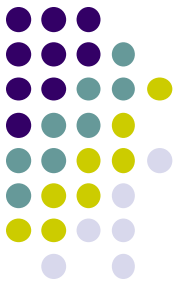


Прямая организация. Индексированный файл

Физическая организация файловой системы

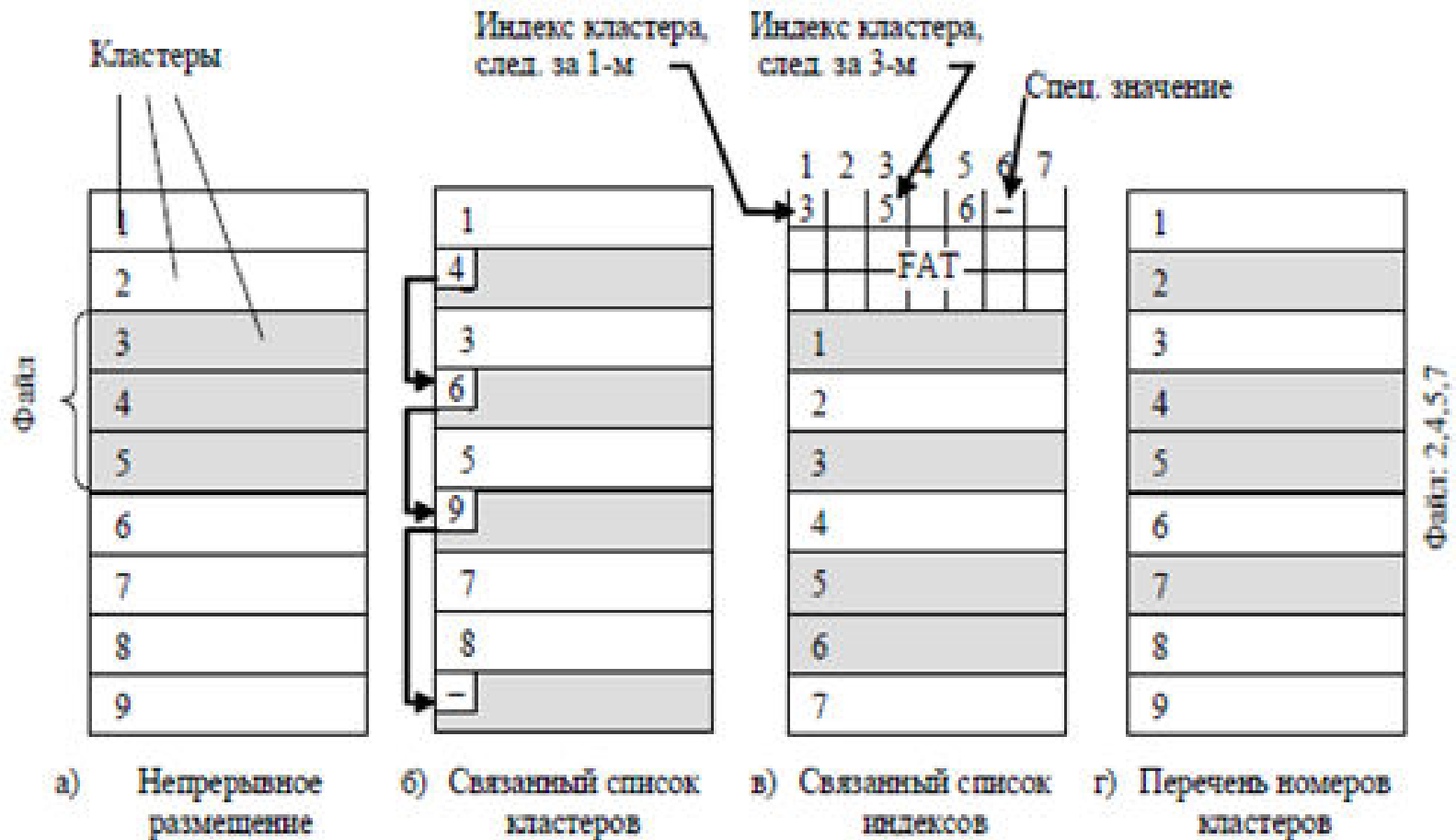
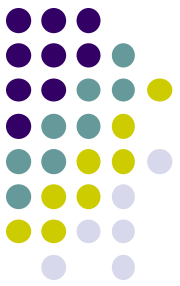


Физическая организация файловой системы



- Основные критерии эффективности физической организации файла:
 - скорость доступа к данным
 - объем адресной информации файла
 - степень фрагментированности дискового пространства
 - возможность увеличения размера файла

Физическая организация файловой системы



Физическая организация файловой системы



Критерий/организация	Непрерывное размещение	Связанный список кластеров	Связанный список индексов	Перечисление номеров кластеров
Скорость доступа	Высокая: нет затрат на поиск и считывание кластеров файла (+)	Невысокая: доступ к кластерам последовательный (–)	Высокая: доступ близок к прямому (+)	Высокая: прямой доступ к кластеру (+)
Объем адресной информации	Минимален: номер первого кластера и объем файла (+)	Минимален: номер первого кластера (+)	Минимален: номер первого кластера и FAT (+)	Длина адреса зависит от размера файла (–)
Степень фрагментации диска	Высокая (–)	На уровне кластеров фрагментация отсутствует (+)	На уровне кластеров фрагментация отсутствует (+)	На уровне кластеров фрагментация отсутствует (+)
Возможность увеличения размера файла	Проблемы: необходимость выделения сплошного участка (–)	Число кластеров легко наращивается (+)	Число кластеров легко наращивается (+)	Увеличение файла увеличивает длину кластера (–)