Оглавление

[Лекция 1 3](#_Toc64280653)

[Структура компьютерной системы 3](#_Toc64280654)

[Классификация операционных систем 4](#_Toc64280655)

[Лекция 2 6](#_Toc64280656)

[Архитектура операционных систем 6](#_Toc64280657)

[API: 6](#_Toc64280658)

[Монолитное ядро 6](#_Toc64280659)

[Модульное ядро 6](#_Toc64280660)

[Микроядро 7](#_Toc64280661)

[Гибридная архитектура 7](#_Toc64280662)

[Виртуальные машины 7](#_Toc64280663)

[Лекция 3 8](#_Toc64280664)

[Процессы. Управление процессами. 8](#_Toc64280665)

[Функции ОС 8](#_Toc64280666)

[Состояния процесса 8](#_Toc64280667)

[Части РСВ 9](#_Toc64280668)

[Порождение программ 9](#_Toc64280669)

[Многоразовые операции 9](#_Toc64280670)

[Операции над процессом 9](#_Toc64280671)

[Лекция 4 10](#_Toc64280672)

[Планирование процессов. Основные алгоритмы 10](#_Toc64280673)

[Планирование 10](#_Toc64280674)

[Уровни планирования 10](#_Toc64280675)

[Цели планирования 10](#_Toc64280676)

[Параметры планирования 10](#_Toc64280677)

[Гарантированное планирование 11](#_Toc64280678)

[Многоуровневые очереди 11](#_Toc64280679)

[Лекция 5 12](#_Toc64280680)

[Управление памятью 12](#_Toc64280681)

[Принципы управления памятью 12](#_Toc64280682)

[Функции ОС по управ. Памятью 12](#_Toc64280683)

[Связывание процессов 12](#_Toc64280684)

[Схема управ. памятью 12](#_Toc64280685)

[Своппинг 12](#_Toc64280686)

[Борьба с фрагментацией 12](#_Toc64280687)

[Модели виртуальной памяти 13](#_Toc64280688)

[Лекция 6 13](#_Toc64280689)

[Хранение данных. Файловых системы 13](#_Toc64280690)

[Хранение данных 13](#_Toc64280691)

[История 13](#_Toc64280692)

[Основные идеи построения 13](#_Toc64280693)

[Иерархия каталогов 13](#_Toc64280694)

[Имена файлов 13](#_Toc64280695)

[Операции над файлами 13](#_Toc64280696)

[Операции над директориями 14](#_Toc64280697)

[Схема работы ФС 14](#_Toc64280698)

[Способы выделения пр-ва для файлов 15](#_Toc64280699)

[Индексация узла 15](#_Toc64280700)

[Контроль свободного пространства 15](#_Toc64280701)

[Монтирование ФС 15](#_Toc64280702)

# Лекция 1

## Структура компьютерной системы

Аппаратное обеспечение ->

Операционная система ->1)2)

1)Прикладное программное обеспечение

-СУБД

-…

-Текст редактор

2)Системное программное обеспечение

- Компилятор

-Ассемблер

-Дефрагментатор

Операционная система – базовое системное программное обеспечение, управляющее работой компьютера и являющееся посредником между аппаратурой, прикладным программным обеспечением и пользователем компьютера.

**Цели работы операционной системы:**

1)обеспечение выполнение прикладных программм

2)обеспечение удобства, эффективности, надежности, безопасности использования компьютерной системы.

3)обеспечение использования внешних устройств.

4)обеспечение безопасности, надежности и защиты данных.

**Функции операционной системы:**

1) обеспечение выполнения программ

2)управление оперативной памятью

3)управление внешней памятью

4)управление вводом-выводом

5)предоставление пользовательского интерфейса

6)обеспечение безопасности

7)организация сетевого взаимодействия

Структура ОС->1)2)

1)Привилегированный режим

-ядро-низкоуровневая основа ОС

-подсистема управления ресурсами

--драйверы – прогр. модули управ. устройст.

--управ. оператив. и внешней памятью, процессорам и др.

2)Пользовательский режим

- Системные библиотеки

--прикладной программный интерфейс (API)

-Пользоват. Оболочки (shell)

--предостав. пользователю интерфейс

--text интерфейс (CLI)

--Графический интерфейс(GUI)

## Классификация операционных систем

1.По типу ядра

-Система с монолитным ядром

-Система с микроядром

-Система с гибридным ядром

-Объекты – микроядра

2.По кол-ву одновременно выполняемых задач

- Однозадачные : MS-DOS, PC-DOS, DOS CP/M, ISIS

- Многозадачные : Multics, Linux, Unix, AmegaOS.

3.По способу организации вычислений

-Системы пакетной обработки:

-Системы разделения времени

-Системы реального времени:QNX,CHORUS,LynxOS,VRTX

4.По кол-ву одновременно работ. Пользователей

-однопользовательские

-многопользовательские

5.По кол-ву поддерживаемых процессоров

-однопроцессорные

-многопроцессорные

6.По поддержке сети

-локальные

-сетевые

7.По роли в сетевом взаимодействии

-серверные

-клиентские

8.По типу лицензии

-Открытые

-проприетарные

# Лекция 2

## Архитектура операционных систем

### API:

Application programming interface

Интерфейс прикладного программирования

-это набор готовых классов, процедур, функции, структур и констант, предоставляемых для использования во внешних программных продуктах

-он может быть у любого программного обеспечения

-ОС, библиотеки, приложения, веб-приложения

-многообразие API

1) У каждой ОС свой API

2) Прослойки – промежуточные API Gtk ,Qt ,Wine ,Cyqwin ,стандарт С …

-строгое описание сигнатур функций

Ядро(kernel) – это центральная часть ОС, обеспечивающая приложениям координируемый и контролируемый доступ к ресурсам компьютера.

**Архитектура ядер**

-Монолитное

-Модульное

-Микроядро

-Гибридное

### Монолитное ядро

-все части работают в адресном пр-ве как одна программа

-каждая процедура может вызывать каждую

-все процедуры ядра работают в привилегированном режиме

-пользовательские программы взаимодействуют с ядром через системные вызовы и переключение уровня привилегий.

### Модульное ядро

-модули

-большинство современных монолитных ядер позволяют во время работы загружать модули

-модули не самостоятельных

-статически связаны с ядром

Многоуровневые системы

-реализуются в системах с несколькими уровнями привилегий

-процедура уровня N может вызывать только процедуры уровня N-1

-все или почти все уровни работают в привилегированном режиме

### Микроядро

-выполняет строгий минимум ф-ий в привилегированном режиме

-компоненты системы взаимодействуют друг с другом путем передачи сообщений через микроядро

### Гибридная архитектура

-все подходы обладают недостатками и достоинствами

-современные ОС общего назначения объединяют основные подходы для уменьшения недостатков

-например, многоуровневый подход применяется практически везде

Linux=монолитное ядро + модульная архит.

Windows=микроядро + элементы монолитной архит.

### Виртуальные машины

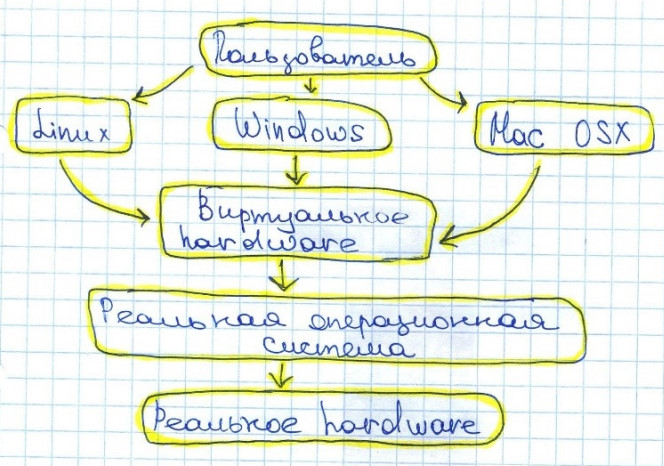
-это виртуальная исполнительная среда с определенным набором команд

\*скриптовые языки

\*языки компилируемые в промежуточный код

\*ОС предоставляющие виртуальное оборудование

\*Гипервизоры используют возможность оборудования по виртуализации аппаратного обеспечения



# Лекция 3

## Процессы. Управление процессами.

### Функции ОС

1. Управление памятью
2. Управление процессами
3. Управление оборудованием
4. Интерфейс

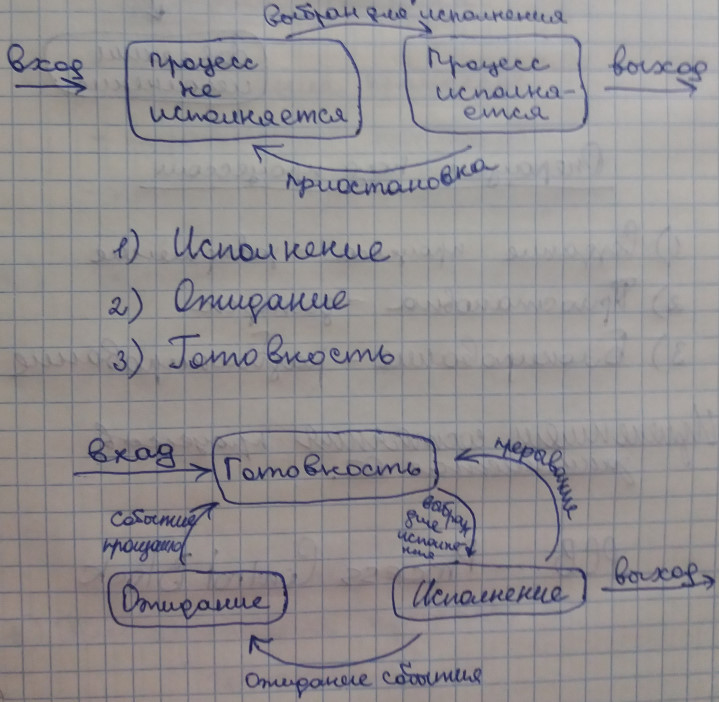
Программа – это статическая последовательность команд

Процесс – совокупность набора исполняющих команд ассоциированных с ними ресурсов и данных состояния исполнения находящаяся под управлением ОС.

-Если кол-во задач <= числа процессоров в системе, то результат очевиден

-В случаях одного процессора время приходится делить

### Состояния процесса



### Части РСВ

1. Регистровый контекст
2. Системный контекст
3. Пользовательский контекст

### Порождение программ

1) Дублирование процессов

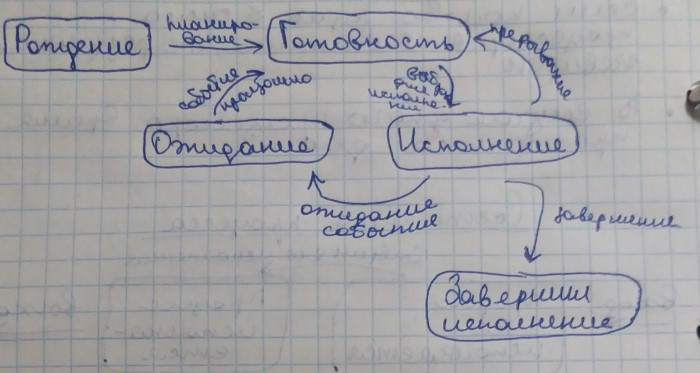
-параллельные программы

-несколько процессов для одной программы

2) Замена пользовательского контекста

-В рамках одного процесса могут последовательно исполнится несколько различных программ

### Многоразовые операции

1. Приостановка процесса –> «готовность»
2. Блокирование процесса –> «ожидание»
3. Разблокирование процесса –> «готовность»

### Операции над процессом

1. Создание процесса – завершение
2. Приостановка – запуск
3. Блокирование – разблокирование

Изменением состояния процессов занимается ОС.

PCB – Process Control Block

# Лекция 4

## Планирование процессов. Основные алгоритмы

### Планирование

\*Ресурсы ограничены

-вычислительные в том числе

-вычислительное у-во 1

-процессов несколько

-задача распределить все процессы во времени

Планировщик – это модуль ОС, отвечающий за разделение времени имеющихся процессоров между выполняющимися процессами.

->может принимать решения о выборе для исполнения нового процесса из числа находящихся в состоянии готовность

### Уровни планирования

-Планирование заданий

\* в системах пакетного н

\* выбор какое задание их пакета сейчас будет исполняться

-Планирование использования процессора

\*появляется только в мультипроцессных системах

-> краткосрочное планирование

-> миллисекунда

-> своппинг

### Цели планирования

1. Выбор алгоритма планирования определяется целями, которые необходимо
2. Справедливость – гарант каждому процессу опр. Часть времени
3. Эффективность
4. Сокращение полного времени выполнения
5. Сокращение времени ожидания
6. Сокращение времени отклика

### Параметры планирования

-> параметры

--> Системы

--> процессов

-> Статические

-> Динамические

CPU burst - промежуток времени непрерывного использования про-ра

I/O burst - промежуток времени непрерывного ожидания ввода/вывода

### Гарантированное планирование

N пользователей, должны получить 1/N времени

Ti – длительность сеанса пользователя i

Ji– выделенное процессорное время

Ji ~ Ti/N

Коэффициент справедливости

JiN/Ti

### Многоуровневые очереди

Процессы ядра (FCFS) -> Интерактивные пользовательские (RRL0) -> Фоновые системные (FCFS) -> Фоновые пользовательские (RR50)

# Лекция 5

## Управление памятью

Память – относится к основным ресурсам вычислительной системы

Менеджер памяти – модуль ОС, который управляет памятью

Сегментация памяти:

* Изначально участки памяти, хранящие информацию которую система отображает в памяти нескольких процессов
* Двумерный адрес
* Разные сегменты
* Поддержка оборудованием

### Принципы управления памятью

1. Разделение памяти на физическую и логическую

* Виртуальные адреса

1. Иерархия памяти

### Функции ОС по управ. Памятью

1. Отображение логических адресов на физическую память
2. Распределение памяти между конкурирующими процессами
3. … адресных пространств процессов
4. Учет свободной и занятой памяти

### Связывание процессов

1. Этап компиляции
2. Этап загрузки
3. Этап выполнения

### Схема управ. памятью

1. Один процесс на всю память

* Простейщий своппенг

1. Фиксированные разделы

* Одна очередей но всем разделами
* Несколько очередей к каждому разделу

### Своппинг

* В системах с разделением времени

- перенос памяти процесса полностью на жесткий диск

- восстанов может быть в ту же область памяти или в другую

- хватит времени >> время загрузки свопа

- выгружается записанная память, неработающих процессов

-выгрузка в область на жест. диске в обхож файле. сист.

### Борьба с фрагментацией

* Не требовать непрерывности адресного пространства процесса(Статичная память)
* Дефрагментация

- перемещения занятых(свободных) участков для … их в одну область

- приводит и перебазированию процессов

### Модели виртуальной памяти

* Страничная
* Сегментная
* Сегментно-страничная

# Лекция 6

## Хранение данных. Файловых системы

### Хранение данных

* Оперативная память не подходит для долговременного хранения больших объемов данных
* У-ва внешней(вточичной) памяти, большой ёмкости

### История

Магнитные ленты-> Программисты сами выбирали способ хранения данных на носителе-> Сложно поддерживать данные и архивы-> Централизованные системы управления файлами!

### Основные идеи построения

1)Внешняя память ОС разделена на блоки фиксированного размера(4096 байт)

2) Файл набор блоков

3) Адреса блоков данных файла хранятся в отдельном блоке внешней памяти

\* в индекснам узле

\* по номеру байта вычисляют блок -> его адрес

### Иерархия каталогов

* ФС представляет собой иерархич. Структуру каталогов
* Путь – уникальное имя объектов в ФС
* Сами каталоги играют роль индексов в ссылками на свои подкаталоги

### Имена файлов

Файл – абстрактный именованный объект

Именование( длина имени 256 символов)

Доступ к данным в файле

1. Последовательный
2. Случайный
3. Метода основ. на индексации

### Операции над файлами

-create

-delete

-open

-close

-seek

-read

-write

-get attributes

-set attributes

-rename

### Операции над директориями

-create

-delete

-open dir

-close dir

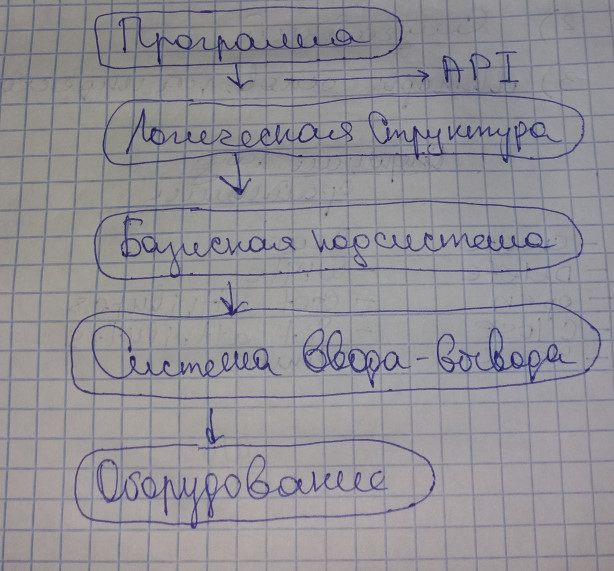
-read dir

-rename

-link

-unlink

### Схема работы ФС



### Способы выделения пр-ва для файлов

* Непрерывная последовательность блоков

- диазительнаа начало+размер

«+» простота в реализации

Предводительность

«-» фрагментация проблема поиска места

-невозможность оценить размер файла зарание

### Индексация узла

* С каждым файлом связан индесный узел

- I - node

-небольшая таблица с атрибутами файла и его блоками на диске

-запись файла в директории содержит в указатель на его i-node

### Контроль свободного пространства

1. Битовый вектор

* Каждый блок – бит
* 0 или 1 – занят или свободен
* 001111000111100011000001…

1. Связанный список

* Список всех свободных блоков
* Хранение указателей на первый свободный
* Необходим обход всего списка

### Монтирование ФС

1. ФС необходимо … чтобы она была доступна для процессов
2. Mount «ус-во» <тогда подкл>

