

***Тема 1.**

Операционные системы

* Основные понятия

Появления и развития системного обеспечения началась с осознания, что любая программа требует операций ввода-вывода данных. Это произошло в 50-е годы прошлого столетия.

Операционная система (ОС) - это совокупность программных средств, обеспечивающих управление аппаратными ресурсами компьютера, поддержка выполнения программ, взаимодействие программ с аппаратной частью, другими программами и пользователем.

*Классификация ПО

Программы, работающие на компьютере, можно разделить на **3 категории**:

- **прикладные программы**, непосредственно обеспечивающие выполнение необходимых пользователям работ: редактирование текстов, рисование картинок, обработка информационных массивов и т.д.;
- **системные программы**, выполняющие различные вспомогательные функции, например создание копий используемой информации, выдачу справочной информации о компьютере, и т.д.;
- **инструментальные системы** (системы программирования) обеспечивающие создание новых программ для компьютера.

* Системное ПО

Системные программы

Системное ПО представляет собой комплекс управляющих и обрабатывающих программ, описаний и инструкций, обеспечивающих функционирование вычислительной системы, а также разработку и использование программ пользователей.

Состав системного программного обеспечения почти не зависит от характера решаемых задач пользователей.

Программы системного ПО различаются по функциональному назначению и характеру исполнения.

Они делятся на **испытательные** программы, **системы программирования** (СП) и операционные системы (ОС).

* Системное ПО.

* Операционная система

Операционная система.

Среди всех системных программ операционная система занимает особое место.

ОС - программа, которая **загружается** при включении компьютера.

Она **осуществляет** диалог с пользователем, управление компьютером, его ресурсами (оперативной памятью, местом на диске и т.д.), запускает другие (прикладные) программы на выполнение.

Операционная система **обеспечивает** пользователю и прикладным программам удобный способ общения (интерфейс) с устройствами компьютера.

*Функции ОС

* (кратко)

Операционные системы предназначены для выполнения следующих (тесно взаимосвязанных) функций:

- управление данными;
- управление задачами (заданиями, процессами);
- связь с пользователем.

*Функции ОС

Основные функции операционных систем:

- 1) **прием** от пользователя (или от оператора системы) **заданий**, или команд, сформированных на соответствующем языке, и их обработка.
- 2) **загрузка** в оперативную память подлежащих исполнению **программ**;
- 3) **распределение памяти**, а в большинстве современных систем и организация виртуальной памяти;
- 4) **запуск программы**;

*Функции ОС

- 5) **идентификация** всех программ и данных;
- 6) **прием и исполнение** различных **запросов** от выполняющихся приложений;
- 7) обслуживание всех **операций ввода-вывода**;
- 8) **обеспечение работы** систем управлений файлами (**СУФ**) и/или систем управления базами данных (**СУБД**);
- 9) **обеспечение** режима **мультипрограммирования**, то есть организация параллельного выполнения двух или более программ на одном процессоре, создающая видимость их одновременного исполнения;
- 10) **планирование** и диспетчеризация **задач**;
- 11) **организация** механизмов **обмена** сообщениями и данными между выполняющимися программами;

*Функции ОС

12) **обеспечение взаимодействия** связанных между собой компьютеров (для сетевых ОС);

13) **защита** одной программы от влияния другой, обеспечение сохранности данных, защита самой операционной системы от исполняющихся на компьютере приложений;

14) **аутентификация и авторизация пользователей.**

Аутентификация – процедура проверки имени пользователя и его пароля на соответствие тем значениям, которые хранятся в его учетной записи.

Авторизация – в соответствии с учетной записью пользователя, который прошел аутентификацию, ему назначаются определенные права.

*Функции ОС

15) **удовлетворение жестким ограничениям** на время ответа в режиме реального времени (характерно для операционных систем реального времени);

16) **обеспечение работы систем программирования**, с помощью которых пользователи готовят свои программы;

17) **предоставление услуг** на случай частичного сбоя системы.

* Основные задачи ОС

Основные задачи ОС следующие:

- **увеличение пропускной способности** ПЭВМ (за счет организации непрерывной обработки потока задач с автоматическим переходом от одной задачи к другой и эффективного распределения ресурсов ПЭВМ по нескольким задачам);
- **уменьшение времени реакции** системы на запросы пользователей;
- **упрощение работы** разработчиков программных средств и пользователей, за счет предоставления им значительного количества языков программирования и разнообразных сервисных программ.

*Классификация ОС

ОС могут классифицироваться по следующим показателям:

- **количество пользователей:** однопользовательские ОС (MS DOS, Windows) и многопользовательские (VM, Unix, Linux);
- **доступ:** пакетные (OS 360), интерактивные (Windows, Unix , Linux) и системы реального времени (QNX, Neutrino,RSX);
- **количество решаемых задач:** однозадачные ОС (MS DOS) и многозадачные ОС (Windows, Unix , Linux).

*Понятие операционных сред и оболочек

Операционная система выполняет функции управления вычислениями в компьютере, распределяет ресурсы вычислительной системы между различными процессами, и образует ту программную среду, в которой выполняются прикладные программы пользователей. Такая среда называется операционной.

Термин «**операционная среда**» означает, прежде всего, соответствующие интерфейсы, необходимые программам и пользователям для обращения к управляющей (супервизорной) части операционной системы с целью получить определенные сервисы.

*Понятие операционных сред и оболочек

Операционная среда может включать несколько интерфейсов (оболочек).

Программы-оболочки относятся к классу системных программ. Они обеспечивают более удобный и наглядный способ общения с компьютером, чем штатные средства ОС.

Некоторые программы не заменяют «штатную оболочку», а **дополняют ее или добавляют** в нее новые функции.

* Обработка прерываний в ОС

Прерывание — это принудительная передача управления от выполняемой программы к системе (а через нее — к соответствующей программе обработки прерывания), происходящая при возникновении определенного события.

Идея прерывания была предложена в середине 50-х годов.

Основная цель введения **прерываний** — реализация асинхронного режима функционирования и распараллеливание работы отдельных устройств вычислительного комплекса.

*Обработка прерываний в ОС

Механизм прерываний реализуется аппаратно-программными средствами. Прерывание влечет изменение порядка выполнения команд процессором .

Механизм обработки прерываний подразумевает

выполнение шагов:

- 1) установление факта прерывания;
- 2) запоминание состояния прерванного процесса вычислений.
- 3) управление аппаратно передается на подпрограмму обработки прерывания;

*Обработка прерываний в ОС

- 4) сохранение информации о прерванной программе, которую не удалось спасти помощью аппаратуры;
- 5) собственно выполнение программы, связанной с обработкой прерывания;
- 6) восстановление информации, относящейся к прерванному процессу;
- 7) возврат на прерванную программу.

Шаги 1-3 реализуются **аппаратно**, шаги 4-7 — **программно**.

*Обработка прерываний в ОС

Прерывания, возникающие при работе вычислительной системы, можно разделить на два основных класса: **внешние** (их иногда называют асинхронными) и **внутренние** (синхронные).

Внешние прерывания вызываются асинхронными событиями, которые происходят вне прерываемого процесса.

Внутренние прерывания вызываются событиями, которые связаны с работой процессора, и являются синхронными с его операциями.

Существуют **программные прерывания**.

* Обработка прерываний в ОС

Процессор может обладать средствами защиты от прерываний: отключение системы прерываний, маскирование (запрет) отдельных сигналов прерывания.

Программное управление специальными регистрами маски (маскирование сигнала прерывания) позволяет реализовать различные дисциплины обслуживания:

- с относительными приоритетами, то есть обслуживание не прерывается даже при наличии запросов с более высокими приоритетами.
- с абсолютными приоритетами, то есть всегда обслуживается прерывание с наивысшим приоритетом.
- по принципу стека (последним пришел, первым обслужен).

* Организация вычислительного процесса

Операционная система — это набор программ, обеспечивающий организацию **вычислительного процесса** на ЭВМ.

Режим работы — это способ (правила) распределения наиболее важных ресурсов системы.

Вычислительный процесс в системе представляется в виде последовательности, как правило, ветвящейся, простых процессов — одноразовых работ, выполняемых ресурсами ВС.

Ресурсы ВС — это средства, необходимые для вычислений.

Событие - это изменение состояния ресурса, изменение его характеристик.

* Вычислительный процесс и его состояния, дескриптор процесса

Как понятие **процесс** является определенным видом абстракции.

Последовательный процесс, иногда называемый задачей (task), — это отдельная программа с ее данными, выполняющаяся на последовательном процессоре. Под последовательным понимается такой процессор, в котором текущая команда выполняется после завершения предыдущей.

Концепция **процесса** предполагает два аспекта: во-первых, он **является носителем данных** и, во-вторых, он собственно и **выполняет операции**, связанные с обработкой этих данных.

* Вычислительный процесс и его состояния, дескриптор процесса

Необходимо отличать **системные управляющие вычислительные процессы**, представляющие работу супервизора операционной системы и занимающиеся распределением и управлением ресурсов, от всех других процессов: задач пользователей и системных обрабатывающих процесс.

Для **системных управляющих процессов**, в отличие от обрабатывающих, в большинстве операционных систем **ресурсы распределяются изначально и однозначно**. Эти вычислительные процессы сами управляют ресурсами системы, в борьбе за которые **конкурируют все остальные процессы**.

* Вычислительный процесс и его состояния, дескриптор процесса

Процесс может находиться в активном и пассивном состоянии.

В активном состоянии процесс может конкурировать за ресурсы вычислительной системы, а в пассивном состоянии он известен системе, но за ресурсы не конкурирует.

* Вычислительный процесс и его состояния, дескриптор процесса

В свою очередь, **активный** процесс может быть в одном из следующих **состояний**:

- **выполнения** — все затребованные процессом ресурсы выделены;
- **готовности к выполнению** — ресурсы могут быть предоставлены, тогда процесс перейдет в состояние выполнения;
- **блокирования, или ожидания**, — затребованные ресурсы не могут быть предоставлены, или не завершена операция ввода-вывода.

* Вычислительный процесс и его состояния, дескриптор процесса

ОС организует для нового процесса информационную структуру – так называемый дескриптор процесса (описатель задачи, блок управления задачами), и процесс (задача) начинает выполнение.

В общем случае дескриптор процесса содержит следующую информацию:

- идентификатор процесса;
- тип (класс) процесса;
- приоритет процесса,
- переменную состояния процесса;

* Вычислительный процесс и его состояния, дескриптор процесса

- **контекст задачи**, то есть защищенную область памяти (или адрес такой области), в которой хранятся текущие значения регистров процессора, когда он прерывается, не закончив работы;
- **информацию о ресурсах**, которыми процесс владеет и/или имеет право пользоваться;
- **место** (или его адрес) **для организации общения** с другими процессами;

*** Вычислительный процесс и его состояния, дескриптор процесса.**

- **параметры времени запуска** (момент времени, когда процесс должен активизироваться, и периодичность этой процедуры);
- **адрес задачи на диске в ее исходном состоянии** и адрес на диске, куда она выгружается из оперативной памяти, если ее вытесняет другая задача.

* Виды ресурсов и возможности их разделения

Термин **ресурс** обычно применяется по отношению к **многократно используемым**, относительно **стабильным** и часто недостающим объектам, которые запрашиваются, задействуются и освобождаются в период их активности.

* Виды ресурсов и возможности их разделения

Основные виды ресурсов вычислительной системы:

1. **Процессор**, точнее — процессорное время.
Процессорное время делится попеременно (параллельно).
2. Память. Оперативная память может делиться и одновременно, и попеременно. Внешняя память может разделяться и одновременно, а доступ к ней всегда разделяется попеременно.
3. **Внешние устройства**, как правило, могут разделяемыми и выделенными.

* Виды ресурсов и возможности их разделения

4. **Программные модули** (прежде всего, системные).
Могут быть **однократно и многократно используемыми**.
Однократно используемые могут быть правильно
выполнены только один раз, являются неделимым
ресурсом.
Повторно используемые программные модули могут
быть непривилегированными, привилегированными и
реентерабельными.

* Виды ресурсов и возможности их разделения

Привилегированные программные модули работают при отключенной системе прерываний.

Непривилегированные программные модули могут быть прерваны во время своей работы (нельзя считать разделяемыми). В реентерабельных модулях обеспечено сохранение промежуточных результатов для прерываемых вычислений.

Существуют еще **повторно входимые программные модули**. Они состоят привилегированных секций.

* Виды ресурсов и возможности их разделения

5. Информационные ресурсы, в качестве ресурсов могут выступать данные как в виде переменных, находящихся в оперативной памяти, так и в виде файлов.

Информационные ресурсы можно разделять, если они используются только для чтения.

* Классификация операционных систем

Операционные системы принято классифицировать: по назначению, по режиму обработки задач, по способу взаимодействия с системой и, наконец, по способам построения (архитектурным особенностям системы).

Прежде всего, традиционно различают *ОС общего и специального назначения*. ОС специального назначения, в свою очередь, подразделяются на ОС **для носимых микрокомпьютеров и различных встроенных систем**, организации и ведения баз данных, решения задач реального времени и т. п.

* Классификация операционных систем

По режиму обработки задач различают ОС, обеспечивающие **однопрограммный** и **мультипрограммный** (мультизадачный) режимы. Современные ОС для персональных компьютеров реализуют и мультипрограммный, и мультизадачный режимы.

* Классификация операционных систем

По способу взаимодействия с компьютером ОС разделяют на диалоговые системы и системы **пакетной обработки**.

При организации работы с вычислительной системой в диалоговом режиме существуют **однопользовательские** (однотерминальные) и **мультитерминальные** ОС. В мультитерминальной ОС с одной вычислительной системой одновременно могут работать несколько пользователей, каждый со своего терминала.

* Классификация операционных систем

Основной особенностью **операционных систем реального времени (ОСРВ)** является обеспечение обработки поступающих заданий **в течение заданных интервалов времени**, которые нельзя превышать. Одной из наиболее известных ОСРВ для персональных компьютеров является ОС QNX.

* Классификация операционных систем

По архитектурному принципу операционные системы разделяются на **микроядерные и макроядерные** (монолитные). В некоторой степени это разделение условно но, однако можно в качестве яркого примера микроядерной ОС привести ОСРВ QNX, тогда как в качестве монолитной можно назвать Windows 95/98 или Linux.

* Управление задачами

Операционная система выполняет следующие основные функции, связанные с управлением процессами и задачами:

- создание и удаление задач;
- планирование процессов и диспетчеризация задач;
- синхронизация задач, обеспечение их средствами коммуникации.

* Управление задачами

Решение вопросов, связанных с тем, **какой задаче** следует **предоставить процессор в данный момент**, возлагается на специальный модуль операционной системы, чаще всего называемый **диспетчером задач**.

Вопросы же подбор вычислительных процессов, которые не только можно, но и **целесообразно решать** возлагаются на **планировщик процессов**.

Иногда используют термин **стратегия обслуживания**.

* Управление задачами

Среди **стратегий**, прежде всего, можно выбрать следующие:

- по возможности **заканчивать вычисления** (вычислительные процессы) в том же самом **порядке**, в котором они были **начаты**;
- отдавать предпочтение более **коротким вычислительным задачам**;
- предоставлять всем пользователям (процессам пользователей) **одинаковые услуги**, в том числе и одинаковое время ожидания.

* Управление задачами

Дисциплины диспетчеризации (дисциплины обслуживания) – правила формирования очереди готовых к выполнению задач.

Различают **два больших класса** дисциплин обслуживания: **бесприоритетные и приоритетные**.

При бесприоритетном обслуживании выбор задачи производится в некотором заранее установленном порядке без учета их относительной важности и времени обслуживания.

При реализации **приоритетных дисциплин** обслуживания отдельным задачам предоставляется преимущественное право попасть в состояние исполнения.

* Управление задачами

В концепции **приоритетов** имеем следующие **варианты**:

- **приоритет**, присвоенный задаче, является величиной **постоянной**;
- **приоритет изменяется** в течение времени решения задачи (динамический приоритет).

Диспетчеризация с динамическими приоритетами требует дополнительных расходов на вычисление значений приоритетов исполняющихся задач, поэтому во многих операционных системах реального времени используются методы диспетчеризации на основе абсолютных приоритетов.

* Управление задачами в ОС. Диспетчеризация

Наиболее часто используемые **дисциплины диспетчеризации**.

1. **FCFS** (First Come First Served – первым пришел, первым обслужен) – задачи обслуживаются в порядке их появления.
2. **SJN** (Shortest Job Next — следующим выполняется самое короткое задание) требует, чтобы для каждого задания была известна оценка в потребностях машинного времени.

* Управление задачами в ОС. Диспетчеризация

3. **SRT** (Shortest Remaining Time) — следующим будет выполняться задание, которому осталось меньше всего выполняться на процессоре.

4. **Карусельная дисциплина** обслуживания RR (Round Robin) — каждая задача получает процессорное время квантами времени (time slice) q .

* Управление задачами в ОС.

Диспетчеризация

Одна из проблем, которая возникает при выборе подходящей дисциплины обслуживания — это **гарантия обслуживания**. Гарантировать обслуживание можно, например, следующими тремя способами:

Выделять минимальную долю процессорного времени **некоторому классу** процессов, если по крайней мере один из них готов к исполнению.

Выделять минимальную долю процессорного времени некоторому **конкретному** процессу, если он готов к выполнению.

Выделять столько процессорного времени некоторому **процессу**, чтобы он мог выполнить свои вычисления к сроку.

* Управление памятью и виртуальным адресным пространством

Имена переменных и входных точек программных модулей составляют **пространство символьных имен**. Иногда это адресное пространство называют **логическим**.

Физическая память представляет собой упорядоченное множество ячеек реально существующей оперативной памяти, и все они пронумерованы, то есть к каждой из них можно обратиться, указав ее порядковый номер (адрес).

* Управление памятью и виртуальным адресным пространством

Системное программное обеспечение должно **связать** каждое указанное пользователем **символьное имя с физической ячейкой памяти**, то есть осуществить **отображение** пространства имен на физическую память компьютера.

* Управление памятью и виртуальным адресным пространством

В общем случае это отображение осуществляется в **два этапа**: сначала системой программирования, а затем операционной системой. Между этими этапами обращения к памяти имеют форму **виртуального адреса**. Множество всех допустимых значений виртуального адреса для некоторой программы определяет ее виртуальное адресное пространство или **виртуальную память**.

* Управление памятью и виртуальным адресным пространством

Частные случаи отображения пространства символьных имен на физическую память:

- **полная тождественность** виртуального адресного пространства физической памяти (абсолютная двоичная программа);
- тождественность **виртуального адресного пространства** исходному логическому пространству имен;
- **промежуточные** варианты.

* Управление памятью и виртуальным адресным пространством

Возможны **три ситуаций**:

- объем виртуального адресного пространства программы **меньше** объема физической памяти (сейчас почти не встречается);
- объем виртуального адресного пространства программы **равен** объему физической памяти (встречается очень часто, особенно характерна была для недорогих вычислительных комплексов);
- объем виртуального адресного пространства программы **больше** объема физической памяти (теперь это самая обычная ситуация).

* Управление памятью и виртуальным адресным пространством

Простое непрерывное распределение — это самая простая схема, согласно которой вся память условно может быть разделена на **три области**:

- область, **занимаемая операционной системой**;
- область, в которой размещается **исполняемая задача**;
- **незанятая ничем** (свободная) область памяти.

Операционная система строится таким образом, чтобы постоянно в оперативной памяти полагалась только **самая нужная ее часть**. Эту часть операционной системы стали называть **ядром**.

* Управление памятью и виртуальным адресным пространством

Остальные модули операционной системы, не относящиеся к ее ядру, могут быть **обычными диск-резидентными (или транзитными)**, то есть загружаться в оперативную память только по необходимости и после своего выполнения вновь освобождать память.

* Управление памятью и виртуальным адресным пространством

Если нужно создать программу, логическое адресное пространство которой должно быть **больше, чем свободная область памяти**, или даже чем весь возможный объем оперативной памяти, то используется **распределение с перекрытием** — так называемые **оверлейные структуры** (от overlay — перекрытие, расположение поверх чего-то). Этот метод распределения предполагает, что вся программа может быть разбита на части — **сегменты**. Пока в оперативной памяти располагаются выполняющиеся сегменты, остальные находятся во внешней памяти.

* Распределение памяти статическими и динамическими разделами

Память задаче может выделяться одним сплошным участком (в этом случае говорят о методах **неразрывного распределения памяти**) или несколькими порциями, которые могут быть размещены в разных областях памяти (тогда говорят о **методах разрывного распределения**).

Память, не занятая ядром операционной системы, может быть разбита на несколько частей — **разделов**.

* Разделы с фиксированными границами

Разбиение всего объема оперативной памяти на несколько разделов может осуществляться **единовременно** или **по мере необходимости** оператором системы.

Увеличить число параллельно выполняемых приложений можно за счет **свопинга** (swapping) –задача может быть выгружена на магнитный диск (перемещена во внешнюю память), а на ее место загружается либо более привилегированная, либо просто готовая к выполнению другая задача, находившаяся на диске в приостановленном состоянии.

* Разделы с подвижными границами

Одной из первых операционных систем, в которой был реализован такой способ распределения памяти, была OS MVT.

Специальный планировщик (диспетчер памяти) ведёт **список адресов свободной** оперативной памяти. При появлении новой задачи он выделяет для нее раздел, объем которой равен необходимому, либо чуть больше, модифицирует список свободных областей памяти.

Способы выделения памяти под новый раздел:

- **первый** подходящий участок;
- **самый подходящий** участок;
- **самый неподходящий** участок.

* Сегментная, страничная и сегментно-страничная организация памяти

Первым среди разрывных методов распределения памяти был **сегментный**. Программа разбивается на логические элементы — сегменты, **каждому сегменту выделяется физическая память**.

Вся информация о текущем размещении сегментов задачи в памяти обычно сводится в таблицу сегментов, чаще такую таблицу называют **таблицей дескрипторов сегментов задачи**.

Виртуальный адрес для этого способа будет состоять из двух полей — **номера сегмента и смещения относительно начала сегмента**.

* Дисциплины замещения сегмента

- **правило FIFO** (First In First Out — первый пришедший первым и выбывает);
- **правило LRU** (Least Recently Used — дольше других неиспользуемый);
- **правило LFU** (Least Frequently Used — реже других используемый);
- **случайный** (random) выбор сегмента.

* Страничная организация памяти

Оперативная память **разбивается на физические страницы**, а программа — **на виртуальные страницы** (размер их совпадает).

Часть виртуальных страниц задачи размещается в оперативной памяти, а часть — во внешней.

Обычно место во внешней памяти, в качестве которой в абсолютном большинстве случаев выступают накопители на магнитных дисках называют **файлом подкачки**, или страничным файлом (paging file).

Иногда этот файл называют swar-файлом

* Страничная организация памяти

Для каждой задачи имеется **таблица страниц** для трансляции адресных пространств.

Для **описания каждой страницы** диспетчер памяти операционной системы заводит соответствующий **дескриптор**.

По номеру виртуальной страницы в таблице дескрипторов страниц текущей задачи находится соответствующий элемент (дескриптор).

* Страничная организация памяти

При обращении к виртуальной странице, **не оказавшейся** в данный момент в оперативной памяти, возникает **прерывание**, и управление передается диспетчеру памяти, который должен найти свободное место.

Обычно предоставляется первая же свободная страница. Если свободной физической страницы нет, то диспетчер памяти по одной из вышеупомянутых **дисциплин замещения** (LRU, LFU, FIFO, случайный доступ) определит страницу, подлежащую расформированию или сохранению во внешней памяти.

* Страничная организация памяти

Пробуксовка — это ситуация, при которой загрузка нужной страницы вызывает перемещение во внешнюю память той страницы, с которой мы работаем.

* Сегментно-страничный способ

Виртуальное пространство задачи делится на сегменты, а каждый сегмент — на виртуальные страницы, которые нумеруются в пределах сегмента. Оперативная память делится на физические страницы.

Загрузка задачи выполняется операционной системой странично, при этом часть страниц размещается в оперативной памяти, а часть — на диске.

Для каждого сегмента создается своя таблица страниц.

Для каждой задачи создается таблица сегментов.

Виртуальный адрес состоит из трех компонентов: сегмента, страницы и индекса.

* Основные концепции организации ввода-вывода в ОС

Главный принцип: любые операции по управлению вводом-выводом объявляются **привилегированными** и могут **выполняться** только кодом самой **операционной системы**.

Для обеспечения этого принципа в большинстве процессоров вводятся **режимы пользователя и супервизора** (привилегированный режим, или режим ядра).

Как правило, в режиме супервизора выполнение команд ввода-вывода **разрешено**, а в пользовательском режиме — **запрещено**.

* Основные концепции организации ввода-вывода в ОС

Существуют устройства:

- **разделяемые** — допускают разделение посредством механизма доступа. Примеры — накопитель на магнитных дисках, устройства чтения компакт-дисков — устройства с прямым доступом;
- **неразделяемые** — устройства с последовательным доступом. Примеры — принтер, накопитель на магнитных лентах.

* Основные концепции организации ввода-вывода в ОС

Управление вводом-выводом осуществляется компонентом операционной системы, который часто называют **супервизором ввода-вывода**.

Основные задачи супервизора:

1) супервизор задач (модуль супервизора ОС) **получает запросы** от прикладных задач на выполнение тех или иных операций, в том числе на ввод-вывод. Эти запросы проверяются на корректность и, если они соответствуют спецификациям и не содержат ошибок, то обрабатываются дальше;

* Основные концепции организации ввода-вывода в ОС

2) супервизор ввода-вывода получает **запросы на ввод-вывод от супервизора задач** или от программных модулей самой операционной системы.

3) супервизор ввода-вывода **вызывает соответствующие распределители** каналов и контроллеров, планирует ввод-вывод. Запрос на ввод-вывод либо тут же выполняется, либо ставится в очередь на выполнение.

* Основные концепции организации ввода-вывода в ОС

4) супервизор ввода-вывода **инициирует операции** ввода-вывода и (при использовании прерываний) предоставляет процессор диспетчеру задач, чтобы передать его первой задаче, стоящей в очереди на выполнение;

5) при получении сигналов прерываний от устройств ввода-вывода супервизор **идентифицирует** эти сигналы и передает управление соответствующим программам обработки прерываний;

.

* Основные концепции организации ввода-вывода в ОС

6) супервизор ввода-вывода **осуществляет передачу** сообщений об ошибках, если таковые происходят в процессе управления операциями ввода-вывода.

7) супервизор ввода-вывода **посылает сообщения** о завершении операции ввода-вывода запросившей эту операцию задаче и снимает ее с состояния ожидания ввода-вывода, если задача ожидала завершения операции

* Основные концепции организации ввода-вывода в ОС

Имеется **два** основных режима ввода-вывода:

- 1) **режим обмена с опросом** готовности устройства ввода-вывода;
- 2) **режим обмена с прерываниями.**

Если управление вводом-выводом осуществляет центральный процессор, то это программный канал обмена данными между **внешними устройством и оперативной памятью** (в отличие от канала прямого доступа к памяти, при котором управление вводом-выводом осуществляет **специальное дополнительное оборудование**).

* Основные концепции организации ввода-вывода в ОС

Основные системные таблицы ввода-вывода

Для управления всеми операциями ввода-вывода и отслеживания состояния всех ресурсов, занятых в обмене данными, операционная система должна иметь соответствующие информационные структуры, которые отображают информацию:

- **состав устройств ввода-вывода** и способы их подключения;
- **аппаратные ресурсы**, закрепленные за имеющимися в системе устройствами ввода-вывода;

* Основные концепции организации ввода-вывода в ОС

- **логические** (символьные) имена устройств ввода-вывода, используя которые вычислительные процессы **могут запрашивать** те или иные операции ввода-вывода;
- **адреса размещения** драйверов устройств ввода-вывода и области памяти для хранения текущих значений переменных, определяющих работу с этими устройствами;

* Основные концепции организации ввода-вывода в ОС

- **области памяти** для хранения информации о текущем состоянии устройства ввода-вывода и параметрах, определяющих режимы работы устройства; данные о текущем процессе, который работает с данным устройством;
- **адреса тех областей памяти**, которые содержат **данные**, собственно и участвующие в операциях ввода-вывода (получаемые при операциях ввода данных и выводимые на устройство при операциях вывода данных).

* Основные концепции организации ввода-вывода в ОС

Создаются **три системных таблицы**.

Первая таблица содержит информацию обо всех устройствах ввода-вывода, подключенных к вычислительной системе — **таблица оборудования**, а каждый элемент этой таблицы называется UCB (Unit Control Block — блок управления устройством ввода-вывода).

* Основные концепции организации ввода-вывода в ОС

Каждый элемент USB содержит информацию:

- **тип устройства**, его конкретная модель, символическое - имя и характеристики устройства;
- **способ подключения** устройства;
- **номер и адрес канала** (и подканала), если такие используются для управления устройством;
- **информация о драйвере**, который должен управлять этим устройством;

* Основные концепции организации ввода-вывода в ОС

- информация о том, используется или нет буферизация при обмене данными с устройством, **«имя» буфера**;
- **установка тайм-аута** и ячейки для счетчика тайм-аута; состояние устройства;
- **поле указателя для связи задач**, ожидающих устройство;
- возможно, множество других сведений.

* Основные концепции организации ввода-вывода в ОС

Вторая таблица предназначена для реализации принципа независимости от устройства.

Вторая системная таблица называется **таблицей виртуальных логических устройств** (Device Reference Table, DRT).

Назначение ее — **установление связи** между виртуальными (логическими) устройствами и реальными устройствами, описанными посредством первой таблицы (таблицы оборудования).

* Основные концепции организации ввода-вывода в ОС

Третья таблица — таблица прерываний — необходима для организации обратной связи между центральной частью и устройствами ввода-вывода.

Указывает для каждого сигнала запроса на прерывание тот элемент УСВ, который сопоставлен данному устройству.

Каждое устройство либо имеет свою линию запроса на прерывание, либо разделяет линию запроса на прерывание с другими устройствами, но при этом имеется механизм второго уровня адресации устройств ввода-вывода

***Задание 1**

Составить таблицу сравнительных характеристик ОС (сравнить не менее 5-ти ОС)

***Задание 2**

Выполнить доклад-презентацию (не менее 10-ти слайдов) по одной из ОС (согласовать с одногруппниками, чтобы не было повторений).