

# Тема 1. НАЗНАЧЕНИЕ ОПЕРАЦИОННЫХ СИСТЕМ

При рассмотрении истории развития ЭВМ принято выделять несколько этапов развития – поколений ЭВМ. Формальными признаками перехода от одного поколения ЭВМ к другому является смена используемой элементной базы, состава периферийных устройств и другие материальные показатели. Однако, с позиций сегодняшнего дня значительно более важным признаком перехода от одного поколения ЭВМ к другому является качественный показатель – способ взаимодействия человека с вычислительной машиной.

Идея компьютера была предложена английским математиком Чарльзом Бэбиджем (Charles Babage) в середине XIX века.

Вычислительные машины **первого поколения** появились в середине 40-х годов прошлого века, вскоре после окончания Второй мировой войны, в связи с созданием первых ламповых вычислительных устройств. Они представляли собой громоздкие ламповые машины с быстродействием 10-20 тыс. операций в секунду и имели классическую **фон-неймановскую архитектуру** (von Neumann machine).

Вычислительная машина является **машиной с архитектурой фон Неймана**, если:

1. **Используется двоичная система счисления.** Преимущество перед десятичной системой счисления заключается в том, что устройства можно делать достаточно простыми, арифметические и логические операции в двоичной системе счисления также выполняются достаточно просто.
2. **Применено программное управление ЭВМ.** Работа компьютера контролируется программой, состоящей из набора команд. Команды выполняются последовательно друг за другом. *Созданием машины с хранимой в памяти программой было положено начало тому, что мы сегодня называем программированием.*
3. **Память компьютера используется не только для хранения данных, но и программ.** Способ записи команд программ и данных одинаков. *Поэтому в определенных ситуациях над командами можно выполнять те же действия, что и над данными.*
4. **Ячейки памяти ЭВМ имеют адреса, которые последовательно пронумерованы.** В любой момент можно обратиться к любой ячейке памяти по ее адресу.
5. *Каждая программа выполняется последовательно, начиная с первой команды, если нет специальных указаний. Для изменения этой последовательности используются команды передачи управления .*

**Машина фон Неймана**, как и практически каждый современный ЭВМ общего назначения, **состоит из пяти основных компонентов**:

1. **Арифметико-логическое устройство** выполняет команды из определенного набора, называемого *системой команд*, над данными, хранящимися в памяти.
2. **Устройство управления**, организует последовательное выполнение алгоритмов, расшифровку команд, поступающих из запоминающего устройства, реагирует на аварийные ситуации и выполняет общие функции управления всеми узлами вычислительной машины. Обычно ОП и устройство управления объединяются в структуру, называемую **центральным процессором**. Следует обратить внимание, что требование именно последовательного, в порядке поступления из памяти (в порядке изменения адресов в счетчике команд) выполнения команд является принципиальной. Архитектуры, которые не соблюдают такого принципа, вообще не считаются фон-неймановской.
3. **Запоминающее устройство** - массив ячеек с уникальными идентификаторами (адресам), в которых хранятся команды и данные.
4. **Устройство ввода-вывода** обеспечивает связь ЭВМ с внешним миром, устройств, передающих информацию на переработку в ЭВМ и принимают результаты.

После загрузки программы в запоминающее устройство, машина фон-Неймана может работать автоматически, без вмешательства оператора. Каждая ячейка памяти машины имеет уникальный номер - адрес, а специальный механизм, чаще всего - **счетчик команд** - обеспечивает автоматическое выполнение необходимой последовательности команд, и определяет на каждом этапе адрес ячейки, из которой необходимо загрузить следующую команду.

Перед началом выполнения программы в счетчик записывается адрес ее первой команды. **Определение адреса** следующей команды происходит по одному из следующих сценариев:

- Если текущая команда не является командой передачи управления (то есть это просто арифметическая или логическая операция над данными), то к текущему значению счетчика добавляется число, равное длине текущей команды в минимально адресованных единицах информации.
- Если текущая команда - команда передачи управления (команда условного или безусловного перехода), которая изменяет последовательный ход выполнения программы, то в счетчик принудительно записывается адрес той команды, которая была заказана при выполнении перехода, где бы она ни находилась.

В большинстве машин **первого поколения** была реализована концепция хранимой программы, а для ввода-вывода использовалась перфокарта,

перфокарты, магнитная проволока, магнитная лента и печатающие устройства. В памяти содержатся команды, которые выполняются последовательно и под управлением которых обрабатываются данные. Память состоит из некоторых элементов, имеющих два устойчивых состояния; одному приписывается значение 0, другому – 1. Логически такой физический элемент независимо от его технической реализации рассматривается как двоичный разряд – бит (bit). Бит – это наименьшая «порция» памяти, а следовательно, и информации в любой двоичной системе, реализованной в ЭВМ. Элементы памяти объединяются технически в байты (byte) (1 байт=8 бит), а логически их можно объединять в более крупные блоки, например, слова [слово – вектор битов, рассматриваемый аппаратной частью ЭВМ как единое целое; число битов в слове, называемое длиной или размером слова, равно разрядности процессора (16, 32 или 64); память ЭВМ обычно разделяется на слова и, возможно, подразделяется ещё на байты].

Программирование для ЭВМ первого поколения было абсолютно «прозрачным» – между программистом с его программой и ЭВМ не было никаких посредников. Все действия по подготовке и загрузке программ, а также по обеспечению программы необходимыми данными выполнялись пользователем вручную. Программирование осуществлялось исключительно на машинном языке.

В машинах первого поколения процессор выполнял команды всех видов, включая операции ввода-вывода (I/O – input/output). Развитие системного программного обеспечения привело к появлению загрузчика – программы помещающей выполняемую программу в оперативную память и приводящей её в состояние готовности к исполнению.

Для удобства подготовки программ были созданы системы символического кодирования – ассемблеры, которые позволили числовые коды объектов программы заменить мнемоническими обозначениями.

Программа стала представляться в двух вариантах:

- **исходный модуль** (source code – текст программы на языке программирования), подлежащий преобразованию в эквивалентный объектный код компилятором (compiler – программа, предназначенная для трансляции высокоуровневого языка в абсолютный код или иногда на язык ассемблера).

- **абсолютный модуль.**

Использование ассемблера породило **следующую последовательность действий по обработке программ:**

- а) **загрузка транслятора ассемблера** с ленты, т.е. его размещение в ОЗУ для исполнения;

- б) **трансляция** (компиляция) исходного модуля и запись абсолютного модуля на ленту;

- в) **загрузка программы** (абсолютного модуля) с ленты;

- г) **выполнение программы.**

С середины 50-х годов XX века начался новый период (**второе поколение**) в развитии ВТ, связанный с появлением новой технической базы – **полупроводниковых элементов**. В машинах второго поколения использовались как электронные лампы, так и полупроводниковые приборы (дискретная транзисторная логика). Их оперативная память была построена на магнитных сердечниках. В это время стал расширяться диапазон применяемого оборудования ввода-вывода, появились высокопроизводительные устройства для работы с МЛ и первые виды оперативно-доступной памяти (магнитные барабаны и магнитные диски – МД).

В эти годы появились первые алгоритмические языки (Algol, Fortran, Cobol, BO, Commercial Translator, FACT, Mathmatic), и, таким образом, к библиотекам математических и служебных подпрограмм добавился новый тип системного ПО – **трансляторы**. Выполнение каждой программы стало включать большое количество вспомогательных работ: загрузка нужного транслятора, запуск транслятора и получение результирующей программы в машинных кодах, связывание программы с библиотечными подпрограммами, загрузка программы в ОП, запуск программы, вывод результатов на периферийное устройство.

Прообразом современных операционных систем явились **мониторные системы пакетной обработки** середины 50-х годов прошлого века, которые автоматизировали действия оператора по выполнению пакета заданий. Оператор составлял пакет заданий, которые (задания) в дальнейшем без его участия последовательно запускались на выполнение управляющей программой – монитором (monitor – супервизор или даже вся операционная система). Кроме того, монитор был способен самостоятельно обрабатывать наиболее часто встречающиеся при работе пользовательских программ аварийные ситуации, такие как отсутствие исходных данных, переполнение регистров, деление на ноль, обращение к несуществующей области памяти и т.д.

В 1965-1975 годах XX века переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхемам открыл путь к появлению ЭВМ третьего поколения (third generation), на которое пришлось два поколения ОС. В этот период были реализованы практически все основные концепции, присущие современным ОС: мультипрограммирование, мультипроцессирование, поддержка многотерминального многопользовательского режима, виртуальная память, файловые системы, разграничение доступа и сетевая работа. В эти годы начинается расцвет системного программирования.

Революционным событием данного этапа явилась промышленная реализация мультипрограммирования (multiprogramming system) — такой способ организации вычислительного процесса, при котором в памяти компьютера находится одновременно несколько программ, попеременно выполняющихся на одном процессоре.

Мультипрограммирование связано с отделением функции ввода-вывода от ЦП и передачи её специальным процессорам ввода-вывода – каналам.

Канал, или устройство обмена (channel), представляет собой специализированный процессор, выполняющий автономно (т.е. параллельно с обработкой информации в ЦП) операции ввода-вывода. Обычно обеспечивает форматирование и буферизацию и имеет все необходимые средства управления для обеспечения синхронной работы с устройством ввода-вывода. Таким образом, ЦП и каналы имеют доступ к общей информации в памяти, но скорость работы периферийных устройств (особенно содержащих механические части) и скорость работы ЦП несравнимы между собой (например, скорость выдачи строк печатающим устройством существенно ниже скорости их выдачи процессором). Отсюда следует, что для организации операционной системы необходимо накапливать вводимую и выводимую информацию в специальных областях – буферах ввода-вывода т.е. осуществлять буферизацию ввода-вывода.

К одному каналу могут подключаться несколько разных устройств ввода-вывода, при этом для ввода-вывода потоков данных на то или иное устройство используются специальные схемы управления, предусмотренные в составе канала.

Мультиплексный канал используется в случае, если устройство ввода-вывода имеет относительно низкое быстродействие. Операции ввода-вывода по такому каналу на отдельные устройства мультиплексированы, т.е. чередуются так, что одновременно могут работать несколько устройств.

Селекторный канал предназначен для устройств с высоким быстродействием. В этом случае за одну операцию ввода-вывода вводится или выводится целая запись, а затем производится переключение на следующее устройство. Как правило, переключение на другое устройство выполняется после ввода-вывода не одной, а нескольких записей. Пока селекторный канал обслуживает одно устройство, другие устройства не могут пересылать информацию.

Управляет работой подключённых устройств контроллер (controller). К контроллерам обычно подключаются периферийные устройства или каналы связи.

В мультипрограммных системах пакетной обработки пользователь по-прежнему был лишён возможности интерактивно взаимодействовать со своими программами. Для осуществления непосредственного взаимодействия с компьютером, был разработан вариант мультипрограммных систем – системы разделения времени, рассчитанные на многотерминальные системы. В таких системах каждому терминалу периодически выделяется доля процессорного времени. Диалоговый режим работы позволил в значительной степени повысить эффективность процесса разработки и отладки программ.

В компьютерах 60-х годов произошло существенное изменение в распределении функций между аппаратными и программными средствами компьютера.

Мультипроцессорная организация используется для распараллеливания вычислительных процессов с разделением данных и ресурсов и обеспечения высокой готовности системы. Мультипроцессорирование включает механизмы и процессы управления мультипроцессорной организацией путем обеспечения взаимодействия процессоров через общую память и средства межпроцессорных обменов.

Основу мультипроцессорирования составляют:

- общая разделяемая память;
- межпроцессорное взаимодействие;
- синхронизация часов.

Дополнительными средствами являются внешние прерывания. Прерывание - сигнал, который сообщает процессору, что необходимо временно отложить выполнение текущей задачи и переключиться на выполнение другой. Необходимость в прерываниях возникла в момент, когда стала задача о совместной работе процессора и медленных внешних устройств.

Третье поколение операционных систем датируется серединой 60-х – серединой 70-х годов XX века. Важной тенденцией этого периода является создание семейств программно-совместимых машин и операционных систем для них. Семейство ЭВМ (computer family) – группа вычислительных машин, которая образована несколькими поколениями некоторой вычислительной системы; машины одного и того же семейства имеют, как правило, аналогичную, но не всегда идентичную архитектуру.

Компьютеры третьего поколения были разработаны как машины общего назначения, предназначались для решения любых задач из любых областей приложений. Системы общего назначения рассчитаны на интерактивную работу одного или нескольких пользователей в режиме разделения времени. Большое внимание таких системах уделяется защите от ошибочных и злонамеренных программ и пользователей.

В начале 70-х годов XX века появились первые сетевые операционные системы, которые в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределённое хранение и обработку данных между несколькими компьютерами, связанными электрическими связями. Сетевые средства можно разделить на три функциональные части:

– серверную часть ОС – средства предоставления локальных ресурсов и услуг в общее пользование;

– клиентскую часть ОС – средства запроса доступа к удалённым ресурсам и услугам;

– транспортные средства ОС – средства, которые совместно с коммуникационной системой обеспечивают передачу сообщений между компьютерами сети).

**Четвёртое поколение ЭВМ** (fourth generation of computers) – обобщённое название ЭВМ, разработанных начиная с 80-х. Наиболее важный в концептуальном отношении критерий, по которому четвёртое поколение ЭВМ можно отделить от ЭВМ третьего поколения, состоит в том, что ЭВМ проектируются в расчёте на эффективное использование высокоуровневых языков и упрощения процесса программирования для конечного пользователя (в частности, это проявилось в развитии гибких средств отладки и диагностики). Основным режим работы больших ЭВМ (мэйнфреймов) – режим разделения времени, в котором в отличие от мультипрограммирования время использования процессора квантуется по тем или иным правилам.

Важной вехой в истории развития ОС явилось создание ОС UNIX, которая сравнительно легко переносилась на различные типы компьютеров. К середине 70-х годов программный код для UNIX был на 90% написан на языке C. Широкое распространение эффективных C-компиляторов сделало UNIX уникальной для того времени ОС, обладающей возможностью сравнительно лёгкого переноса на различные типы компьютеров.

К наиболее важным событиям развития ОС в 80-е годы XX века можно отнести разработку стека протоколов TCP/IP, становление Интернета, стандартизацию технологий локальных сетей, появление персональных компьютеров и операционных систем для них, а также попытки разработки ЭВМ пятого поколения.

Протокол TCP/IP (Transmission Control Protocol/Internet Protocol) представляет собой стек сетевых протоколов, повсеместно используемый для Интернета и других подобных сетей (например, данный протокол используется и в ЛВС). Название TCP/IP произошло от двух наиболее важных протоколов:

IP (интернет протокол) - отвечает за передачу пакета данных от узла к узлу. IP пересылает каждый пакет на основе четырехбайтного адреса назначения (IP-адрес).

TCP (протокол управления передачей) - отвечает за проверку корректной доставки данных от клиента к серверу. Данные могут быть потеряны в промежуточной сети. TCP добавлена возможность обнаружения ошибок или потерянных данных и, как следствие, возможность запросить повторную передачу, до тех пор, пока данные корректно и полностью не будут получены.

Начало 80-х годов XX века ознаменовалось появлением персональных компьютеров (ПК). Компьютеры стали широко использоваться неспециалистами, что потребовало разработки «дружественного» ПО, и предоставление этих «дружественных» функций стало прямой обязанностью ОС.

Первая версия наиболее популярной ОС раннего этапа развития персональных компьютеров – MS-DOS компании Microsoft была

однопрограммная однопользовательская ОС с интерфейсом командной строки, способная стартовать с дискеты.

Основными задачами для неё были управление файлами, расположенными на гибких и жёстких дисках в UNIX-подобной иерархической файловой системе, а также поочередной запуск программ.

Наибольшее влияние на развитие ПО для ПК оказала операционная среда Windows компании Microsoft, представлявшая собой надстройку над MS-DOS.

Сетевые функции также реализовывались в основном сетевыми оболочками, работавшими поверх ОС. При сетевой работе всегда необходимо поддерживать многопользовательский режим, при котором один пользователь – интерактивный, а остальные получают доступ к ресурсам компьютера по сети. В таком случае от ОС требуется хотя бы некоторый минимум функциональной поддержки многопользовательского режима.

**Операционная система** – это набор системных программ, контролирующих работу прикладных программ и системных приложений, управляющих всеми ресурсами вычислительной машины и исполняющих роль интерфейса между пользователями, программистами, прикладными программами, системными приложениями и аппаратным обеспечением компьютера.

*Назначение* операционных систем можно разделить на четыре основные составляющие:

## **1. Организация взаимодействия между программной и аппаратной частями компьютера.**

Типичные сервисы ОС:

- ***Инструменты для выполнения программ.*** ОС автоматически выполняет все действия необходимые для запуска исполняемого модуля на выполнение: загрузка в память программы и данных, подключение необходимых файлов устройств ввода-вывода, подготовка к работе других ресурсов, запуск программы.
- ***Обеспечение доступа к различным устройствам ввода-вывода.*** ОС предоставляет единый набор команд для всех устройств ввода-вывода, скрывая от пользователя особенности каждого конкретного блока компьютера.
- ***Обеспечение доступа к файлам.*** ОС предоставляет возможность работать с файлами, записанными на различные типы накопителей. Кроме этого могут обеспечиваться различные механизмы защиты при обращении к файлам.
- ***Инструменты для разработки программного обеспечения.*** Предоставляемые инструменты позволяют пользователю разрабатывать ПО не отвлекаясь на функционирование электронных устройств компьютера. Программист не должен



знать системы команд процессора, так как использует предлагаемые ОС функции высокого уровня.

- **Обеспечение доступа к системным ресурсам.** ОС управляет доступом к ресурсам компьютера, их защиту от несанкционированного использования, а так же разрешает конфликтные ситуации.
- **Обнаружение и обработка возникающих ошибок.** При возникновении сбоя в работе компьютера (ошибки, возникающие в аппаратной части или программном обеспечении) ОС выполняет действия, минимизирующие влияние ошибки на работу системы.
- **Учет использования ресурсов.** Для повышения производительности работы вычислительной системы ОС ведет учет эффективности использования ее ресурсов.

В результате реальная машина, способная выполнить только небольшой набор машинных команд, с помощью операционной системы превращается в **виртуальную машину**, выполняющую широкий набор гораздо более мощных функций.

**Виртуальная машина** (virtual machine) — программная система, эмулирующая аппаратное обеспечение реальной ЭВМ (процессор, оперативная память, накопители и другие устройства).

**2. Организация эффективного использования ресурсов компьютера.** ОС организует управление ресурсами компьютера в соответствии с заданными критериями эффективности, а так же распределяет ресурсы между выполняемыми программами (*процессами*).

Управление ресурсами включает решение следующих задач:

- **планирование ресурса** – определение процесса, которому будет передан ресурс или его часть;
- **удовлетворение запросов на ресурсы** – выделение ресурса процессу;
- **учет использования ресурса** – отслеживание информации о занятости ресурса или его части;
- **разрешение конфликтов между процессами**, претендующими на один и тот же ресурс.

Для решения этих общих задач управления ресурсами разные ОС используют различные алгоритмы, включая характеристики производительности, область применения и даже пользовательский интерфейс.

Пользователь ОС не имеет прямого доступа к большинству функций управления ресурсами.

**3. Обеспечение удобства аппаратных и программных средств вычислительной системы.** ОС содержит служебные программы, обеспечивающие резервное копирование, архивацию данных, проверку, очистку и дефрагментацию дисковых устройств и др. Так же имеются

диагностические программы и средства восстановления работоспособности системы и файлов.

**4. Обеспечение обратной совместимости.** ОС допускает разработку, тестирование и внедрение новых функций, не прерывая процесса нормального функционирования вычислительной системы. Необходимость обновления ОС вызвана постоянным развитием аппаратных средства, внедрением новых инструментов и сервисов, а так же необходимостью исправления ошибок.

*ОС состоит из ряда подсистем, наиболее важными из которых являются:*

**1. Подсистема управления процессами.** *Процесс* (задача) кратко определяется как программа в стадии выполнения. *Программа* – это статический объект, представляющий собой файл с кодами и данными.

Подсистема управления процессами предназначена для планирования очередности выполнения процессов, обеспечения процессов необходимыми ресурсами, обеспечения взаимодействия и синхронизации процессов.

Для идентификации определенного процесса используются дескриптор – структура, содержащая информацию о необходимых и фактически выделенных ресурсах (количестве оперативной памяти, процессорного времени, необходимых файлах, устройствах ввода-вывода), текущем состоянии (активное или заблокированное), приоритете, состоянии регистров и программного счетчика и др.

**2. Подсистема управления памятью.** Функции подсистемы:

- распределение физической памяти между процессами. Используются специальные алгоритмы определяющие время и место размещения очередного блока в памяти;
- организация размещения программ и данных в отведенных им областях памяти,
- настройка адресного пространства в выделенной процессу области памяти. Одним из наиболее популярных способов управления памятью в современных ОС является *виртуальная память*. Использование механизма виртуальной памяти позволяет использовать различные виды памяти таким образом, что программа считает, что в ее распоряжении имеется однородная оперативная память;
- защита областей памяти каждого процесса. Средства защиты пресекают попытки несанкционированного доступа путем аварийного завершения программы-нарушителя.

**3. Подсистема управления файлами.** Для реализации данной подсистемы используется *файловая система*, позволяет заменить низкоуровневый доступ с физической адресацией данных на высокоуровневый с логической адресацией, организует совместный доступ к файлам, защищает их от несанкционированного доступа.

**4. Подсистема управления внешними устройствами (подсистема ввода-вывода).** Подсистема является интерфейсом между ядром компьютера и всеми подключенными к нему устройствами.

*Так как количество этих устройств очень велико (принтеры, сканеры, мониторы, модемы и др.), и каждое устройство имеет свой набор команд, то возникает необходимость в написании драйверов.*

Драйвер – программа, управляющая конкретной моделью внешнего устройства и учитывающая все его особенности.

Наличие подходящих драйверов позволяет ОС правильно использовать внешние устройства ввода-вывода. Созданием драйверов занимаются как разработчики ОС, так и компании, выпускающие внешние устройства. Правильная работа драйверов обеспечивается с помощью заданного в данной версии ОС алгоритма взаимодействия.

**5. Подсистема защиты данных и администрирования.** Безопасность данных вычислительной системы обеспечивается:

- средствами защиты от сбоев и отказов аппаратуры и ошибок программного обеспечения. Важным средством защиты является аудит ОС, заключающийся в фиксации всех событий, от которых зависит безопасность системы.
- средствами защиты от несанкционированного доступа. Для этого организуется парольный вход в систему, а так же ограничение прав отдельных пользователей на использование ресурсов системы.

**6. Интерфейс прикладного программирования.** Возможности операционной системы доступны программисту в виде набора функций, который называется интерфейсом прикладного программирования (Application Programming Interface, API). Приложения обращаются к функциям API с помощью системных вызовов, похожих на вызов подпрограмм.

*Способ реализации системных вызовов зависит от структурной организации ОС, особенностей аппаратной платформы и языка программирования.*

**7. Пользовательский интерфейс.** ОС обеспечивает удобный интерфейс для пользователя компьютера.

Современные ОС поддерживают развитые функции пользовательского интерфейса для интерактивной работы за терминалами двух типов: алфавитно-цифрового и графического.

При использовании *алфавитно-цифрового терминала* пользователь использует набор команд, которые могут вводиться в интерактивном режиме или с помощью специального командного файла. Набор команд обычно позволяет запускать и останавливать приложения, работать с файловой системой, организовывать работу в сети, выполнять административные функции.

При использовании графического интерфейса команда выбирается с помощью меню или графических объектов.

Основные функции ОС:

- прием от пользователя заданий и команд в виде директив соответствующего языка или указаний с помощью устройств ввода и их обработка;
- прием и исполнение запросов на запуск, приостановку, остановку программ;
- загрузка в память исполняемых программ;
- передача управления программе (инициализация);
- идентификация всех программ и данных;
- обеспечение работы систем управления файлами и иных систем управления низкого уровня, например СУБД;
- обеспечение мультипрограммного режима;
- обеспечение функций по организации и управлению всеми операциями ввода-вывода;
- функционирование в режиме реального времени;
- распределение памяти и организация виртуальной памяти;
- планирование и диспетчеризация задач в соответствии со стратегией и дисциплинами обслуживания;
- организация механизмов обмена сообщениями и данными между программами;
- защита данных программы от воздействия других программ;
- наличие сервисных возможностей для восстановления в случае сбоя;
- обеспечение работ систем программирования.

В зависимости от специфики решаемых задач ОС делятся на:

- **системы пакетной обработки** – системы с высокой пропускной способностью, основным критерием оценки которых является количество задач, выполняемых вычислительной системой в единицу времени;
- **системы разделения времени** – системы позволяющие обеспечить комфортность работы пользователя;
- **системы реального времени** – системы позволяющие обеспечивать необходимые интервалы времени между запуском программы и получением результата.

*Каждый тип ОС имеет специфические внутренние механизмы и особые области применения. Некоторые операционные системы могут поддерживать одновременно несколько режимов, например, часть задач может выполняться в режиме пакетной обработки, а часть – в режиме реального времени или в режиме разделения времени.*

Современные ОС являются *мультипрограммными* (многозадачными) (multitasking) – на одном процессоре попеременно выполняются сразу несколько программ, которые совместно используют не только процессор, но и другие

ресурсы машины: оперативную и внешнюю память, устройства ввода-вывода, данные.

Системы пакетной обработки предназначены для решения вычислительных задач, не требующих быстрого получения результатов. Главной целью системы является решение максимального числа задач в единицу времени.

Схема работы:

- **формирование пакета заданий**, содержащего требование к системным ресурсам.
- **формирование множества выполняемых задач** из пакета заданий. Для обеспечения сбалансированности выбираются задачи, предъявляющие разные требования к ресурсам. Желательно одновременное присутствие вычислительных задач и задач с интенсивным вводом-выводом. Выбор нового задания из пакета заданий зависит от внутренней ситуации, складывающейся в системе, т.е. выбирается «выгодное» задание. Следовательно, в ВС, работающих под управлением пакетных ОС, невозможно гарантировать выполнение того или иного задания в течение определённого периода времени.
- **выполнение всего множества задач.** Совмещение по времени операций ввода-вывода и вычислений может достигаться разными способами:
  - **Использование каналов** — специализированных процессоров. ЦП передает указание каналу на выполнение программы ввода-вывода. После этого ЦП и канал работают параллельно.
  - **Использование собственных контроллеров устройств**, которые автономно обрабатывают команды, поступающие от ЦП. Контроллер и ЦП работают асинхронно. В промежутке между передачей команд контроллеру ЦП выполняет вычисления. Контроллер может сообщить ЦП о том, что он готов принять следующую команду, сигналом прерывания либо ЦП сам опрашивает состояние контроллера.

Общее время выполнения смеси задач часто оказывается меньше, чем их суммарное время последовательного выполнения. Однако выполнение отдельной задачи может занять больше времени, чем при монопольном выделении процессора этой задаче.

**В системах пакетной обработки переключение процессора с выполнения одной задачи на выполнение другой происходит по инициативе самой активной задачи**, например, когда она отказывается от процессора из-за необходимости выполнения операции ввода-вывода. Поэтому существует высокая вероятность того, что одна задача может надолго занять

процессор и выполнение интерактивных задач станет невозможным.

*Взаимодействие пользователя с ЭВМ, на которой установлена система пакетной обработки, сводится к тому, что он приносит задание, отдаёт его диспетчеру-оператору, а в конце дня после выполнения всего пакета заданий получает результат.*

Система пакетной обработки повышает эффективность функционирования аппаратуры, но снижает эффективность работы пользователя.

В системах **разделения времени** имеется возможность интерактивной работы сразу с несколькими приложениями. Для этого ОС принудительно периодически приостанавливает выполнение отдельного приложения, не дожидаясь добровольного освобождения процессора. Всем приложениям попеременно выделяется квант процессорного времени, и таким образом пользователи, запустившие программы на выполнение, получают возможность поддерживать с ними диалог.

**Системы реального времени** предназначены для управления различными техническими объектами или технологическими процессами. Для таких систем характерно наличие жестких временных ограничений (время реакции – response time; время наработки на отказ и др.). Важным является наличие достаточного запаса вычислительной мощности на случай пиковой нагрузки.

Различаются 2 типа систем реального времени:

- **Hard real-time** – системы, в которых при нарушении временных ограничений может возникнуть критическая ошибка (отказ) управляемого ею объекта. *Примеры: система управления двигателем самолета; система управления кардиостимулятором.* В таких системах данные хранятся в оперативной памяти или постоянном запоминающем устройстве (ПЗУ). Все необходимые для основного цикла работы системы данные быть должны предварительно быть загружены в память. Исключается любая виртуализация ресурсов.
- **Soft real-time** – системы, в которых нарушение временных ограничений не приводит к отказу управляемого ею объекта. *Примеры: система планирования рейсов на авиалиниях (сбой приводит к задержке, но не к катастрофе).*

В зависимости от особенностей использованного алгоритма управления процессором, операционные системы делят на следующие типы:

- многозадачные или однозадачные;
- однопользовательские или многопользовательские;
- с поддержкой многоуровневой обработки или без нее;
- однопроцессорные или многопроцессорные системы

По числу одновременно выполняемых задач операционные системы могут быть разделены на два класса:

- **однозадачные** системы поддерживают режим выполнения только одной задачи в отдельный момент времени (MS DOS);
- **многозадачные** системы поддерживают параллельное выполнение нескольких программ в один момент времени (UNIX, OS/2, Windows). ОС такого типа управляют разделением совместно используемых ресурсов, таких как процессор, оперативная память, файлы и внешние устройства, т.е. реализуют *мультипрограммный режим*.

По числу одновременно работающих пользователей ОС делятся на:

- **однопользовательские** (MS-DOS) в которых отсутствуют средства защиты для отдельных пользователей;
- **многопользовательские** (UNIX, Windows) которые имеют средства защиты информации отдельного пользователя от несанкционированного доступа других пользователей.

Многонитевая ОС позволяет разделить процессорное время не между задачами, а между их отдельными ветвями (нитеями).

Мультипроцессорная обработка – это способ организации вычислительного процесса в системах с несколькими процессорами, при котором несколько задач могут одновременно выполняться на разных процессорах системы.

В отличие от мультипрограммной обработки в которой в каждый момент времени выполняется только одна программа, в мультипроцессорных системах несколько задач выполняются действительно одновременно, так как имеется несколько обрабатывающих устройств – процессоров.

Мультипроцессорные системы могут иметь:

- **Симметричную архитектуру** в которой все процессоры однородны и единообразно подключены. Память распределена между всеми процессорами. Масштабируемость (возможность наращивания числа процессоров) ограничена. Такая архитектура обеспечивает высокую производительность, так как процессоры хорошо взаимодействуют между собой.
- **Ассиметричную архитектуру** в которой процессоры могут отличаться как своими характеристиками, так и функциональной ролью, которая поручается им в системе. Система может состоять из нескольких независимых устройств, каждое из которых содержит один или несколько процессоров. Каждое такое устройство называется кластером, а вся мультипроцессорная система – кластерной. Система с ассиметричной структурой хорошо масштабируется.

Вычислительный процесс может быть организован следующим образом:

- **симметричное мультипроцессирование** (symmetric multiprocessing (SMP)) предполагает равноправное использование всеми процессорами одной и той же копии ОС. Операционная

система может выполняться на любом процессоре. Любому свободному процессору может быть поручено любое задание. В решении одной задачи могут участвовать несколько процессоров. Все процессоры используют общую память и общие ресурсы. В случае отказа одного из процессоров симметричные системы сравнительно просто реконфигурируются. Может быть реализована только в системах с симметричной мультипроцессорной архитектурой. Большинство современных ОС поддерживают архитектуру SMP.

- **асимметричное мультипроцессирование (asymmetric multiprocessing)** предполагает специализацию отдельных процессоров. Один из процессоров, который является «ведущим» (master processor) распределяет ресурсы для «ведомых» (slave processors) процессоров. Ведомые процессоры работают только как обрабатывающие устройства и никаких действий по организации работы ВС не выполняют. Может быть реализована как для симметричной, так и для несимметричной мультипроцессорной архитектуры.

В зависимости от используемых аппаратных средств различают следующие типы ОС:

ОС мейнфреймов используются для больших компьютеров. ОС ориентированы на одновременную обработку большого количества заданий (как правило операций ввода-вывода). Обычно они предлагают три вида обслуживания:

- **пакетная обработка** позволяющая выполнять стандартные задания без присутствия пользователей (например, составление отчетов);
- **обработка транзакций** управляет очень большим количеством небольших запросов (например, контроль обработки запросов в банке);
- **системы разделения времени** позволяющие множеству удаленных пользователей одновременно выполнять задания на одной машине (например, работа с большой базой данных).

Имеют в своем составе средства передачи сообщений между компьютерами по линиям связи и средства разделение ресурсов компьютера между удаленными пользователями. Сетевые ОС содержат специальные программные компоненты, реализующие основные протоколы передачи данных.

Сетевые операционные системы подразделяются на следующие типы:

- **одноранговые** сетевые ОС устанавливаются на любой рабочей станции и могут использоваться самостоятельно в виде отдельных программных средств, либо входить в состав пакетов, другую половину которых



представляют программы, обслуживающие мощные компьютеры управления сетями – серверы;

- **серверные** ОС состоят из 2 частей и имеют клиент-серверную архитектуру.

Виды и функции серверов:

- **Файловый сервер** (file server) предоставляет доступ к подмножеству файловых систем, расположенных на дисках компьютера-сервера, другим компьютерам локальной сети;
- **Сервер приложений** (application server) предоставляет вычислительные ресурсы (память и процессор) и необходимое окружение для удаленного запуска определенных классов (как правило, больших) приложений с других компьютеров локальной сети.
- **Сервер баз данных** (database server) предоставляет доступ другим компьютерам сети к базам данных, расположенным на компьютере-сервере.
- **Веб-сервер** (Web server) предоставляет доступ клиентам к Web-страницам, расположенным на компьютере-сервере.
- **Прокси-сервер** (proxy-server) поддерживает эффективное обращение компьютеров локальной сети к Интернету, фильтрацию трафика, защиту от внешних атак. Прокси-сервер обычно встроен в операционную систему.
- **Сервер электронной почты** организует отправку и получение электронной почты для компьютеров локальной сети.

ОС персональных компьютеров предназначены для предоставления максимально удобного интерфейса для одного пользователя.

**ОС мини-компьютеров.** К классу мини-компьютеров относятся карманные персональные компьютеры и мобильные телефоны, планшеты. К особенностям ОС относится ограниченный объем постоянной памяти, маленьких размер экрана, особенности организации ввода-вывода данных;

**Встроенные ОС.** Простые операционные системы, устанавливаемые в принтерах, кассовых аппаратах и других внешних устройствах. Состоят из микроядра и функциональных блоков, обеспечивающих подключение в сеть внешнего устройства.

**ОС для облачных вычислений.** Облачные вычисления (cloud computing) – модель вычислений, основанная на динамически масштабируемых (scalable) и виртуализованных ресурсах (данных, приложениях, ОС и др.), которые доступны и используются как сервисы через Интернет и реализуются с помощью высокопроизводительных центров обработки данных (data centers).

**ОС для смарт-карт** предназначены для работы в смарт-картах. На них накладываются крайне жесткие ограничения по мощности процессора и

памяти. Часто такие ОС управляют только одной операцией (например электронным платежом).

К базовым концепциям структурной организации ОС относится способ построения ядра ОС.

**Ядро** (kernel) — центральная часть ОС, обеспечивающая приложениям доступ к ресурсам компьютера.

**Монолитное ядро**(monolithic kernel). При такой схеме операционной системы все ее компоненты являются составными частями одной программы, работающей в привилегированном режиме, используют общие структуры данных и взаимодействуют друг с другом путем непосредственного вызова процедур.

**Модульное ядро.** Модификация архитектуры монолитного ядра, которая позволяет вносить изменения в состав аппаратного обеспечения без перекомпиляции ядра.

**Микроядро**, работающее в привилегированном режиме, выполняет минимальный набор функций по управлению аппаратурой, остальные функции выполняют специализированные компоненты ОС — серверы, работающие в пользовательском режиме. Все компоненты системы взаимодействуют друг с другом путем передачи сообщений через микроядро.

**Гибридное ядро** является модификацией микроядра, позволяющие для ускорения работы запускать некоторые функции в пространстве ядра. Компоненты ядра располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений (как в микроядерных ОС), однако работают в одном адресном пространстве и используют общие структуры данных ( как в ОС с монолитным ядром).

**Экзоядро** ОС предоставляет только функции взаимодействия процессов, безопасное выделение и распределение ресурсов. *Доступ к устройствам на уровне контроллеров позволяет решать задачи, которые нехарактерны для универсальной ОС.*

**Наноядро** выполняет одну задачу — обработку аппаратных прерываний. После обработки прерывания наноядро отправляет данные о результатах обработки программному обеспечению более высокого уровня.