**Лекция №1**

 **Тема «История, назначение, функции и виды операционных систем»**

**Цель занятия:** Сформировать знания студентов об истории развития, назначения и видах операционных систем.

При включении компьютера загружается операционная система.
***Операционная система*** - набор программ, которые обеспечивают возможность использования аппаратуры компьютера. Ее задача заключается в том, чтобы сделать аппаратуру доступной и удобной для пользователя.

*Главное назначение ОС* - управление ресурсами компьютера.
Ресурсы компьютера: процессор, память, устройства ввода/вывода, данные.

***Функции ОС:***

* Определяет интерфейс пользователя.
* Обеспечивает разделение аппаратных ресурсов между пользователями.
* Дает возможность работать с общими данными в режиме коллективного пользования.
* Осуществляет восстановление информации в случае возникновения ошибок.

Изучение истории развития ОС показывает, что все существенные продвижения в области архитектуры ОС связаны с влиянием двух основных факторов:

* прогресс технологии, приводящий к быстрому возрастанию характеристик аппаратуры ЭВМ и к появлению принципиально новых типов аппаратуры;
* принципиально новые идеи, возникающие у проектировщиков.

Такие технологические прорывы, как изобретение магнитных дисков, микропроцессоров, создание высококачественных видеомониторов, настоятельно требовали радикальных изменений в технологии работы с компьютером, и вследствие этого обуславливали создание принципиально новых типов ОС или их отдельных подсистем. С другой стороны, некоторые идеи в области организации вычислительного процесса и интерфейса дали серьезный толчок совершенствованию архитектуры компьютеров.

**ПРЕДЫСТОРИЯ ОС**

Общая характеристика:

Задачи программиста

Недостатки

Прорыв в технологии

Вскоре после того, как в конце 40-х годов XX века были созданы первые электронные компьютеры, очень остро встала проблема повышения эффективности использования оборудования, и прежде всего центрального процессора.

Типичный компьютер первого – второго поколений представлял собой большую комнату, уставленную шкафами и увитую кабелями. Каждое из основных устройств – центральный процессор, оперативная память, накопители на магнитных лентах, устройства ввода с перфокарт, принтер – занимало один или несколько «шкафов» или «тумб», наполненных радиолампами и механическими частями.

Все это стоило больших денег, потребляло бешеное количество электроэнергии и регулярно ломалось.

В таких условиях машинное время стоило очень дорого. Тем не менее, обычная практика использования ЭВМ не способствовала экономии. Как правило, программист, разрабатывающий программу, заказывал ежедневно несколько часов машинного времени и в течение этого времени монопольно использовал машину. Выполнив очередной запуск отлаживаемой программы (которую надо было каждый раз вводить либо с клавиатуры, либо, в лучшем случае, с перфокарт), пользователь получал распечатку (чаще всего в виде массива цифр), анализировал результаты, вносил изменения в программу и снова запускал ее. Таким образом, в ходе сеанса отладки дорогостоящее оборудование простаивало 99% времени, пока программист осмысливал результаты и работал с устройствами ввода/вывода. Кроме того, сбой при вводе одной перфокарты мог потребовать начать сначала всю работу программы.

Возникла великая идея – использовать сам компьютер для повышения эффективности работы с ним же.

Одно из ответвлений этой идеи – создание языков и систем программирования – рассматривается в отдельных курсах. Другим важным шагом стало возложение на специальную компьютерную программу части тех функций, которые до этого выполнял оператор или сам программист.

Программы такого рода назывались обычно мониторами (не путать с монитором как устройством вывода, который в то время был редчайшей экзотикой!). Монитор принимал команды, состоящие, как правило, из 1-2 букв названия и 1-3 аргументов, заданных 8-ричными или 16-ричными числами. Типичными командами были, например:

·загрузка данных с перфокарт по указанному адресу памяти;

· просмотр и корректировка (с пишущей машинки) значений в указанном диапазоне адресов;

· пошаговое выполнение программы с выдачей результатов каждой команды на пишущую машинку;

· запуск программы с указанного адреса с заданием адресов контрольных точек остановки.

Несмотря на убогость, по нынешним меркам, подобных средств, они в свое время значительно повысили производительность работы программистов. Однако кардинального повышения загрузки процессора не произошло.

Временем широкого распространения мониторов в мире были 50-е годы прошлого века (в СССР – 60-е годы). В настоящее время нечто подобное можно встретить на самых примитивных микропроцессорных контроллерах.

### ВИДЫ ОПЕРАЦИОННЫХ СИСТЕМ

### ПАКЕТНЫЕ ОС

Историю собственно ОС можно начать с появления в конце 50-х годов первых систем, организующих работу по пакетному принципу.

Важнейшим организационным изменением, происшедшим на этом этапе развития, стало массовое изгнание программистов из машинных залов, как фактора, лишь вносящего сумятицу в работу.

Теперь от программиста требовалось собрать пакет перфокарт, содержащий его программу, данные к ней, а также управляющие перфокарты. Эти карты на специально разработанном **языке управления заданиями** (JCL, Job Control Language) объясняли операционной системе, чье это задание, что нужно сделать с программой (например, передать ее транслятору с Фортрана), что предпринять в случае успешной трансляции (вероятно, пустить на решение), что – при наличии ошибок (например, перейти к другой программе), откуда взять исходные данные (например, с такого-то цилиндра магнитного диска). Кроме того, там могли быть даже указания на то, сколько метров бумаги можно выделить на распечатку и какое максимальное время может занять работа программы.

Обойтись без столь подробных инструкций было нельзя, потому что программист не присутствовал при запуске задания и не мог вмешаться лично.

Подготовленный пакет передавался, вместе с другими подобными пакетами, оператору ЭВМ, перед которым стояли две основные задачи: чтобы в устройстве ввода не переводились пакеты заданий и чтобы в принтере не кончилась бумага. Когда процессор заканчивал обработку задания и печать его результатов, он вводил следующий пакет и приступал к его обработке. Так достигалась основная цель пакетного режима – исключить простои процессора из-за нерасторопности людей.

В скором времени разработчики ОС осознали, что вычерпаны далеко не все резервы повышения загрузки процессора. Операции ввода и печати требовали лишь очень небольшой доли от полной производительности процессора. Кроме того, в ходе работы программы случались обращения к периферийным устройствам (например, к магнитным лентам и, позднее, дискам), при выполнении которых процессор опять простаивал. Целесообразно было найти способ, чтобы в эти периоды ожидания загрузить процессор другой работой. Но для этого необходимо, чтобы в памяти процессора находились сразу несколько программ, тогда ОС смогла бы переключать процессор на выполнение той программы, которая в данный момент может работать.

Такая организация работы, когда в памяти находятся несколько программ и система в определенные моменты переключает выполнение с одной программы на другую, была названа **мультипрограммированием**. Эта важная идея в разных воплощениях пережила те пакетные системы, в которых она впервые была реализована, и является основой для функционирования практически всех современных ОС.

Среди наиболее развитых пакетных ОС с мультипрограммированием нельзя не назвать OS/360, основную ОС знаменитого в 60-70 гг. семейства ЭВМIBM 360/370.

### ОС С РАЗДЕЛЕНИЕМ ВРЕМЕНИ

На рубеже 60-70 гг. распространенным и не слишком дорогим периферийным устройством становятся мониторы (сначала монохромные и работающие только в текстовом режиме). При этом процессор и ОЗУ остаются самыми дорогими и громоздкими устройствами вычислительной системы. В этих условиях возникает и быстро приобретает популярность принципиально новый тип ОС – **системы с разделением времени**.

К одной ЭВМ подключается несколько десятков рабочих мест, оборудованных дисплеем (монитор + клавиатура) и совместно использующих вычислительные ресурсы ЭВМ. Процессорное время делится на кванты длительностью в несколько десятков миллисекунд и по истечении каждого кванта процессор может быть переключен на обслуживание другого процесса, другого дисплея. Поскольку теперь подготовку текстов программ выполняют сами программисты за дисплеями, а работа по редактированию текста требует очень малых затрат процессорного времени, процессор успевает обслужить все рабочие места практически без ощутимой задержки. Большая часть времени процессора уделяется небольшому числу рабочих мест, где в данный момент запущены на выполнение программы. При этом, разумеется, средняя скорость работы каждой программы уменьшается, по крайней мере во столько раз, сколько программ выполняется одновременно.

Режим разделения времени стал огромным облегчением для программистов, которые вновь смогли в некоторой степени почувствовать себя «хозяевами» ЭВМ и получили возможность запускать программы на трансляцию и отладку хоть каждые 5 минут. Это позволило сократить сроки разработки и отладки программ.

Для трудоемких вычислительных заданий, предусматривающих счет по ранее отлаженным программам, режим разделения времени менее эффективен, чем пакетный, поскольку частое переключение процессора между выполняемыми программами требует дополнительных затрат времени.

Системы разделения времени используются в режиме диалога с пользователем, поэтому вместо громоздких, детализированных операторов JCL в них используются более простые команды, выполняющие элементарные действия – запуск программы, выдача на экран файла или каталога, копирование или удаление файла и т.п. Пользователю не нужно предвидеть заранее все возможные исходы выполнения команды, гораздо проще увидеть результат выполнения на экране и после этого принять решение, какую команду выполнять следующей. В то же время, некоторые часто повторяющиеся последовательности команд удобно описать один раз в виде «пакетного задания» и затем использовать при необходимости. В этом плане системы разделения времени сохраняют те удобные возможности, которые предоставляли пакетные системы.

Первоначально в качестве аппаратной основы систем разделения времени должны были использоваться «большие» ЭВМ, которые позднее стало принято называть «мейнфреймами» (mainframes). Позднее, по мере прогресса вычислительной техники, это стало по плечу даже миниЭВМ (так назывался в те годы класс компьютеров, занимавших всего лишь один-два небольших шкафчика). Следует особо упомянуть серию миниЭВМ PDP-11, имевшую широчайшее распространение во всем мире в течение полутора десятков лет.

Этот период (70-е годы в мире, 80-е в СССР) характерен глубоким развитием теории и практики создания мощных ОС, содержащих развитые средства управления процессами и памятью, реализующих многопользовательский режим работы. Из большого числа подобных систем особого упоминания заслуживает UNIX – единственная система, благополучно дожившая до нашего времени.

### ОДНОЗАДАЧНЫЕ ОС ДЛЯ ПЭВМ

В середине 70-х годов был изобретен микропроцессор, а к началу 80-х микропроцессоры стали догонять по функциональным характеристикам ранее использовавшиеся «большие» процессоры. Эта ситуация сделала почти бесполезным режим разделения времени: зачем делить один процессор между многими задачами и многими пользователями, если проще и дешевле дать отдельный микропроцессор каждому пользователю? Разделение времени осталось целесообразным разве что в отношении суперкомпьютеров.

Появление и бурное распространение персональных компьютеров (ПК) вызвало к жизни новое поколение ОС, которые оказались во много раз проще своих предшественниц. Ненужной оказалась многопользовательская защита. На первых порах показалась ненужной и многозадачность. Все это можно было расценить как явный регресс в развитии ОС.

Наиболее популярной ОС для ранних восьмиразрядных ПК была система CP/M известной тогда фирмы Digital Research, однако с появлением в начале 80-х знаменитой машины IBM PC лидерство было прочно перехвачено системой MS-DOS фирмы Microsoft.

### МНОГОЗАДАЧНЫЕ ОС ДЛЯ ПК С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ

Быстрое развитие технологии привело к тому, что к концу 80-х годов ПК оказались в состоянии решать значительно более сложные и трудоемкие задачи, чем раньше. При этом многие из достижений прежних этапов развития ОС оказались вновь востребованными, но теперь уже в новых условиях, среди которых надо назвать резкое повышение мощности процессоров и объема памяти, появление высококачественных графических мониторов и развитие сетевых технологий.

Стала реальной такая вещь, как многозадачная ОС для ПК. Надо сказать, что первоначально идея системы, в которой один пользователь запускает одновременно несколько приложений, большинству специалистов казалась пустым пижонством и вызывала насмешки: «Почему бы не выполнить несколько программ по очереди?». Сейчас с таким взглядом смешно даже спорить.

А все же, как бы вы обосновали пользу многозадачности для современных ОС типа Windows?

На смену ОС, которые выполняли текстовые команды, вводимые пользователем с клавиатуры, пришли системы, в которых взаимодействие с пользователем основано на использовании GUI (Graphical User Interface, графический интерфейс пользователя).

Значительная часть ПК работает в составе локальных вычислительных сетей. Это привело к тому, что вопросы защиты данных пользователя вновь приобрели первостепенное значение.

**СОВРЕМЕННЫЕ ТЕНДЕНЦИИ В РАЗВИТИИ ОС**

На основе опыта использования многих современных ОС, можно выделить следующие основные тенденции в их развитии.

1. **Графические оболочки**. Любая современная ОС имеет графический пользовательский интерфейс, причем (по вполне понятным причинам острой конкуренции между фирмами-разработчиками) графические оболочки для всех ОС примерно одинаковы по возможностям. Подчас пользователю трудно сориентироваться, в какой именно ОС он работает, хотя для конечных пользователей (непрограммистов), по-видимому, такая унификация удобна.
2. **Поддержка новых сетевых технологий и Web-технологий**. Сети и Интернет активно развиваются. Появляются новые стандарты и протоколы – IPv6, HTML 5 (для облачных вычислений) и т.д. Современные ОС развиваются в направлении поддержки всех новых сетевых технологий.
3. **Усиленное внимание к механизмам безопасности и защиты**. Во многом благодаря инициативе Trustworthy Computing, начатой фирмой Microsoft в 2002 г., а также ввиду все усиливающейся киберпреступности, все современные ОС уделяют повышенное внимание безопасности: при просмотре веб-страниц браузеры выполняют их проверку на отсутствие phishing; загрузки и инсталляции программ из сети выполняются только с явного согласия пользователя и т.д.
4. **Поддержка многопоточности и многоядерных процессоров**. Ввиду широкого распространения многоядерных процессоров, все современные ОС имеют библиотеки программ, поддерживающие эту возможность аппаратуры. Именно благодаря многоядерной архитектуре, становится реально возможным параллельное выполнение потоков (threads).
5. **Поддержка распределенных и параллельных вычислений**. Современные ОС имеют в своем составе высокоуровневые библиотеки, позволяющие разрабатывать параллельные алгоритмы решения задач – например, поддерживающие стандарты параллелизма OpenMP и MPI.
6. **Виртуализация ресурсов и аппаратуры**. Современные ОС имеют в своем составе средства виртуализации, позволяющие выполнять приложения для других платформ в изолированных виртуальных машинах, в которые могут быть инсталлированы другие операционные системы.
7. **Развитие файловых систем** с целью защиты информации и значительного увеличения размера файлов (для мультимедиа). Современные требования обработки мультимедийной информации приводят к тому, что старые файловые системы (например, FAT) оказываются недостаточными для хранения мультимедийных файлов. Например, максимальный размер файла в системе FAT – 4 гигабайта – легко может быть превышен при переписи на компьютер цифровой видеопленки длительностью 10-15 минут. Поэтому разрабатываются новые файловые системы, допускающие хранение очень больших файлов, например, система ZFS в ОС Solaris. Другим требованием является обеспечение конфиденциальности информации, которое приводит к необходимости реализации в файловых системах возможности криптования (которая реализована, например, в файловой системе ZFS).
8. **Поддержка облачных вычислений** – совсем новая тенденция в развитии ОС, начало которой положила "облачная" ОС Windows Azure фирмы Microsoft.

**Контрольные вопросы:**

1. *Охарактеризуйте основные этапы в развитии ОС.*
2. *Расскажите о видах операционных систем.*
3. *Охарактеризуйте основные тенденции в развитии ОС.*
4. *Чем отличаются однозадачные и многозадачные ОС?*