

Ядро ОС.

Компоненты ОС.

Принципы

построения ОС.

- **Ядро́** — центральная часть операционной системы (ОС), обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, память, внешнее аппаратное обеспечение, внешнее устройство ввода и вывода информации.

Модули ядра ОС

- Модули ядра ОС выполняют следующие базовые функции ОС:
 - управление процессами
 - управление памятью
 - управление устройствами ввода-вывода
- Ядро обеспечивает решение задачи **организации вычислительного процесса**: загрузка/выгрузка страниц, обработка прерываний и т.п.
- Другая задача – поддержка приложений, создание для них **прикладной программной среды**. Приложения обращаются к ядру с запросами (**системными вызовами**) для выполнения базовых операций (открытие и чтение файла, вывод информации на дисплей и т.п.)
- Функции выполняемые ядром ОС требуют высокой скорости выполнения и для этого размещаются постоянно в оперативной памяти.

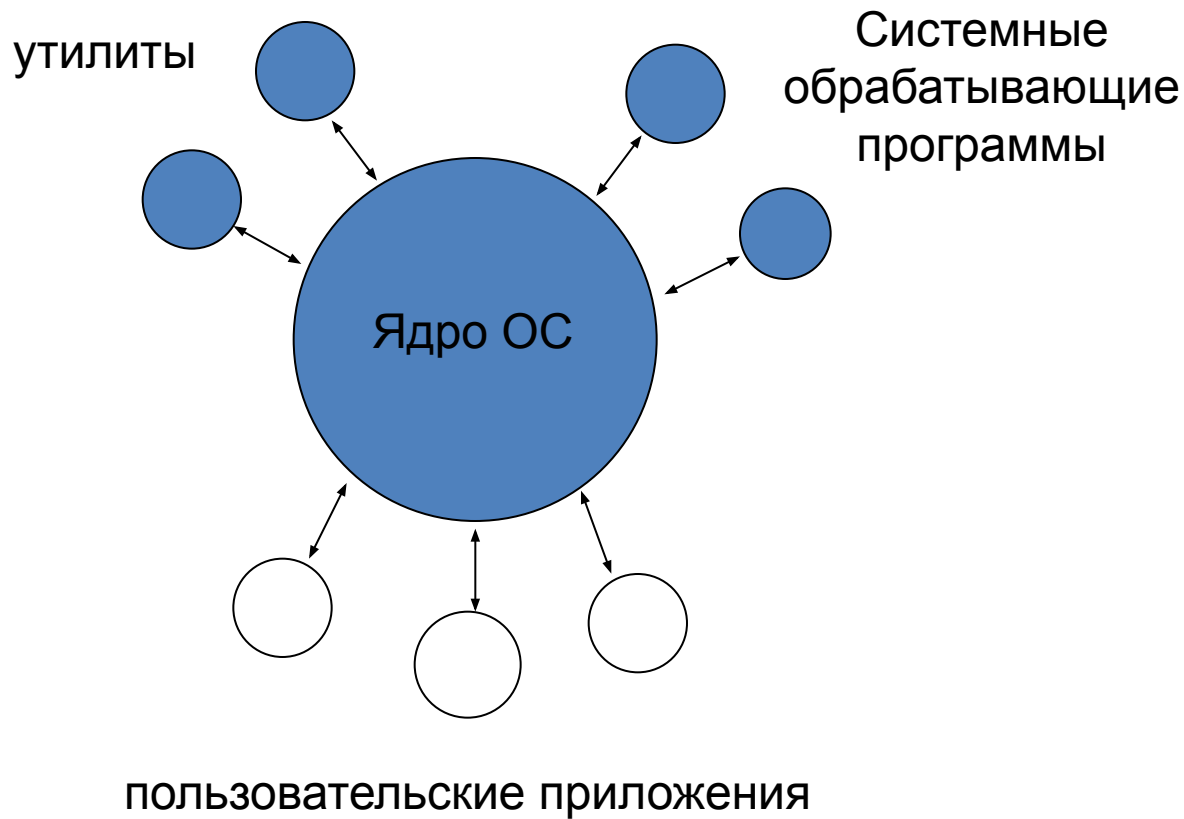
Вспомогательные модули операционной системы

- Вспомогательные модули выполняют полезные, но менее обязательные функции. Например:
 - архивирование информации;
 - дефрагментация данных на диске;
 - поиск необходимого файла и т.п.
- Вспомогательные модули часто оформляются как обычные приложения и провести границу между ними и обычными приложениями сложно.
- Деление на основные и вспомогательные модули ОС условно. Некоторые программы переходят из разряда вспомогательных модулей в основные и наоборот.

Вспомогательные модули операционной системы

- Вспомогательные модули ОС условно разделяются на следующие группы:
 - *Утилиты* – приложения, решающие отдельные задачи управления и сопровождения ОС
 - *Системные обрабатывающие программы* – текстовые и графические редакторы, компиляторы, компоновщики и т.п.
 - *Программы предоставления пользователю дополнительных услуг* – специальный вариант пользовательского интерфейса, калькулятор, игры и т.п.
 - *Библиотеки процедур* – модули различного назначения, упрощающие разработку приложений.
- Вспомогательные модули обращаются к функциям ядра ОС посредством системных вызовов.

Ядро и вспомогательные модули операционной системы



Привилегированный режим процессора

- Для надежного управления работой приложений ядро ОС должно обладать некоторыми привилегиями по отношению к остальным приложениям.
- Обеспечивается привилегированный режим специальными средствами аппаратной поддержкой. Процессор компьютера поддерживает как минимум два режима работы – **пользовательский** (user mode) и **привилегированный** (kernel mode).
- Приложения в пользовательском режиме не могут выполнять некоторые критичные команды (переключение процессора с задачи на задачу, доступ к механизму выделения и защиты областей памяти и т.п.).

Детализация структуры ядра

- Ядро, являясь структурным элементом ОС, может быть логически разложено на ряд слоев:
 - Средства аппаратной поддержки ОС
 - Машинно-зависимые компоненты ОС (включает модули, отражающие специфику аппаратной платформы компьютера)
 - Базовые механизмы ядра (включает наиболее примитивные операции ядра – переключение контекстов процессов, диспетчеризация прерываний), модули выполняют решения принятые на более высоких уровнях
 - Менеджеры ресурсов (реализует задачи стратегического управления), включает менеджеры – диспетчеры процессов, ввода-вывода и т.п.
 - Интерфейсы системных вызовов (включает модули взаимодействия с приложениями и системными утилитами)

Переносимость операционной системы

- Под переносимостью операционной системы понимается способность использования ОС на различных аппаратных платформах с минимальными изменениями в ее структуре. Для уменьшения числа машинно-зависимых модулей разработчики ОС ограничивают универсальность машинно-независимых модулей. Например, Windows разработана для нескольких типов процессоров и для многопроцессорных систем используются собственные модули.
- Для обеспечения переносимости следуют следующим правилам:
 - Большая часть кода написана на языке, трансляторы которого существуют для всех планируемых платформ;
 - Объем машино-зависимых частей кода должен быть минимизирован;
 - Аппаратно-зависимый код должен быть изолирован в нескольких модулях
- В идеале машино-зависимые модули ядра полностью экранируют остальную часть ОС от конкретных деталей аппаратной платформы (кэши, контроллеры прерываний и т.п.).

Микроядерная архитектура

- Концепция микроядерной архитектуры заключается в выделении в качестве работающего в привилегированном режиме части ОС, ответственном за небольшой набор системных функций (управление процессами, обработка прерываний, управление виртуальной памятью, пересылка сообщений). Данная часть ОС называется **микроядром**.
- Все остальные высокоуровневые функции ядра разрабатываются в виде приложений, работающих в пользовательском режиме – **серверы ОС**.
- Взаимодействие между обычными приложениями и серверами ОС осуществляется через механизм обращений. **Клиентское приложение** отправляет запрос к серверу ОС через микроядро ОС. Такой механизм обеспечивает защиту работы приложений.

Принципы построения ОС

- **Принцип генерируемости.** Позволяет настроить ядро и остальные компоненты ОС исходя из конкретной конфигурации ЭВМ и круга решаемых задач. Процедура настройки называется инсталляцией.
- **Принцип функциональной избыточности.** Обеспечивает возможность выполнения одной и той же операции различными способами и средствами, что определяет универсальность и гибкость ОС.

- **Принцип независимости программ от внешних устройств.** Позволяет осуществлять обмен данными и управление внешними устройствами независимо от их характеристик. Это достигается за счет того, что связь программ с конкретными устройствами производится не на уровне трансляции программы, а в период ее исполнения. Например, программе, выполняющей обработку последовательного набора данных, безразлично, какой носитель будет использоваться для их хранения. Непосредственное управление обменом данными между системой и внешними устройствами выполняют специальные программы, называемые драйверами.
- **Принцип совместимости.** ОС должна иметь средства для выполнения прикладных программ, написанных для других ОС. Следует различать совместимость на уровне двоичных кодов и на уровне исходных текстов. Понятие совместимости включает также поддержку пользовательских интерфейсов других ОС.

- **Принцип расширяемости** (открытой и наращиваемой ОС). Аппаратные средства компьютера устаревают за несколько лет, а ОС может использоваться десятилетиями (например, ОС UNIX). Поэтому необходимо чтобы в ОС можно было легко внести изменения и дополнения, не нарушая ее целостности. Изменения ОС обычно заключаются в приобретении ею новых свойств, например поддержке новых типов внешних устройств или новых сетевых технологий. Расширяемость достигается за счет модульной структуры ОС. Взаимодействие модулей осуществляется только через функциональный интерфейс.
- **Принцип переносимости (мобильности)**. Код ОС должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы одного типа на аппаратную платформу другого типа. Аппаратные платформы различаются не только типом процессора, но и архитектурой всего компьютера. переносимые ОС имеют несколько вариантов реализации для разных платформ, т. е. являются многоплатформенными.

- **Принцип надежности и отказоустойчивости.** Система должна быть защищена как от внутренних, так и от внешних ошибок, сбоев и отказов. Ее действия должны быть всегда предсказуемыми, а приложения не должны иметь возможности наносить вред ОС. Важно, включает ли ОС программную поддержку аппаратных средств обеспечения отказоустойчивости, таких как дисковые массивы (RAID) или источники бесперебойного питания.
- **Принцип максимальной производительности.** ОС должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа. На производительность ОС влияет архитектура ОС, многообразие функций, качество программирования кода, аппаратная платформа, на которой работает ОС.
- **Принцип обеспечения безопасности вычислений.** Операционная система должна защищать данные от несанкционированного доступа, обладать средствами защиты ресурсов одних пользователей от других пользователей.

- **Принцип модульности.** Предусматривает построение ОС из функционально законченных модулей. Выполнение модулей ОС не должно зависеть от их расположения в памяти. Перед размещением модуля в памяти производится его настройка под фактические адреса. Существенную роль при этом играют способы адресации процессора и алгоритм распределения памяти, реализованный в ОС.
- **Принцип функциональной избирательности.** В ОС выделяются наиболее важные и часто используемые модули, которые являются основой системы. Эту часть называют ядром ОС. Модули ядра выполняют такие базовые функции ОС, как управление процессами, памятью, устройствами ввода-вывода, системой прерываний. Модули ядра постоянно находятся в оперативной памяти и называются резидентными. Остальные системные модули хранятся на жестком диске и называются транзитными.