

§3 Принципы построения и состав ОС

3.1 Принципы построения и требования к ОС.

3.2 Компоненты ОС.

3.2.1 Ядро. Программы ядра. Функции ядра.

3.3 Типы архитектуры ядра ОС.

3.1 Основные принципы построения ОС

Частотный принцип.

Основан на выделении оперативной памяти по частоте использования.

Действия и данные, которые часто используются, располагаются в оперативной памяти, для обеспечения наиболее быстрого доступа.

Основное средство - организация многоуровневого планирования.

- долгосрочное: редкие и длинные операции управления деятельностью системы.

- краткосрочное: часто используемые и короткие операции.

Система инициирует или прерывает исполнение программ, предоставляет или забирает динамически требуемые ресурсы, и прежде всего центральный процессор и память.

Принцип модульности. Основан на разделении ОС на отдельные функциональные модули.

Модуль - это функционально законченный элемент системы, выполненный в соответствии с принятыми межмодульными интерфейсами (предполагает возможность замены его на любой другой при наличии соответствующих интерфейсов).

Привилегированные модули функционируют в привилегированном режиме, при котором отключается система прерываний, и никакие внешние события не могут нарушить последовательность вычислений.

Реентерабельные модули допускают повторное многократное прерывание исполнения и повторный запуск из других задач.

Повторно входимые модули допускают многократное параллельное использование, однако не допускают прерываний.

Максимальный эффект от использования достигается, если принцип распространяется и на ОС, и на прикладные программы, и на аппаратуру.

Принцип функциональной избирательности.

Подразумевает выделение некоторых модулей, которые должны постоянно находиться в оперативной памяти для повышения производительности вычислений. **Эту часть ОС называют ядром.**

Принцип функциональной избыточности.

Учитывает возможность проведения одной и той же операции различными средствами.

Позволяет быстро и достаточно адекватно адаптировать ОС к определенной конфигурации ВС, обеспечить максимально эффективную загрузку технических средств при решении конкретного класса задач и получить при этом максимальную производительность.

Принцип генерируемости ОС.

- определяет такой способ организации архитектуры ядра ОС, который позволял бы настраивать его, исходя из конкретной конфигурации вычислительного комплекса и круга решаемых задач.

Эта процедура выполняется редко, перед достаточно протяженным периодом эксплуатации ОС.

Процесс генерации осуществляется с помощью специальной программы-генератора и соответствующего входного языка.

Наиболее ярко этот принцип используется в ОС Linux, которая позволяет не только генерировать ядро ОС, но указывать состав подгружаемых, т.н. транзитных модулей.

В остальных ОС конфигурирование выполняется в процессе инсталляции.

Принцип умолчания.

Применяется для облегчения организации связи с системами, как на стадии генерации, так и при работе с системой.

Принцип основан на хранении в системе:

- базовых описаний,
- структур процесса,
- модулей,
- конфигураций оборудования и данных, определяющих:
прогнозируемые объемы требуемой памяти, времени счета программы,
потребности во внешних устройствах.

В целом применение этого принципа позволяет сократить число параметров устанавливаемых пользователем, когда он работает с системой.

Принцип перемещаемости.

Предусматривает построение модулей, исполнение которых не зависит от места расположения в оперативной памяти.

Принцип независимости ПО от внешних устройств.

Заключается в том, что связь программы с конкретными устройствами производится не на уровне трансляции программы, а в период планирования ее использования.

При работе программы с новым устройством, перекомпиляция не требуется.

Принцип виртуализации.

Принцип позволяет представить структуру системы в виде определенного набора планировщиков процессов и распределителей ресурсов (мониторов), используя единую централизованную схему.

Принцип совместимости - определяет возможность выполнения ПО, написанного для другой ОС или для более ранних версий данной ОС.

- на уровне исполняемых файлов (готовую программу можно запустить на другой ОС, используются специально разработанные эмуляторы)
- на уровне исходных текстов программ (требует наличия соответствующего транслятора и также совместимости на уровне системных вызовов и библиотек).

Принцип открытости (возможность доступа для анализа как системным специалистам, так и пользователям) и **наращиваемости** (возможность введения в состав ОС новых модулей и модификации существующих).
клиент-серверная архитектура = широкие возможности по наращиваемости.

Принцип мобильности (переносимости). Подразумевает возможность перенесения ОС с аппаратной платформы одного типа на платформу другого типа.

Правила:

- большая часть ОС пишется на языке, который имеет трансляторы на всех платформах, предназначенных для использования.
- минимизируют или исключают те фрагменты кода, которые непосредственно взаимодействуют с аппаратными ресурсами.
- аппаратно-зависимый код изолируется в нескольких хорошо локализуемых модулях.

Принцип безопасности - защита ресурсов одного пользователя от другого, а также предотвращения захвата всех системных ресурсов одним пользователем, включая и защиту от несанкционированного доступа.

Стандарт NCSC (National Computer Security Center) 1985 года, т.н.

Оранжевой книге, системы подразделяются на 7 категорий:

D, C1, C2, B1, B2, B3, A1, где A является классом с максимальной защитой.

Большинство современных ОС отвечают требованиям уровня C2:

- средства секретного входа, позволяющие идентифицировать пользователя путем ввода уникального имени и пароля при входе в систему;
- избирательный контроль доступа, позволяющий владельцу ресурса определить, кто имеет доступ к ресурсу и его права;
- средства учета и наблюдения (аудита), обеспечивающие возможность обнаружения и фиксации событий, связанных с безопасностью системы и доступом к системным ресурсам;
- защита памяти, подразумевающая инициализацию перед повторным использованием.

Требования к современным ОС

Главное требование - выполнение основных функций эффективного управления ресурсами и обеспечение удобного интерфейса для пользователя и прикладных программ.

Современная ОС должна поддерживать мультипрограммную обработку, виртуальную память, многооконный графический интерфейс пользователя, а также выполнять многие другие необходимые функции и услуги.

Эксплуатационные требования:

- Расширяемость – означает, что код ОС должен быть написан таким образом, что дополнения и изменения могут вноситься без нарушения целостности системы.
- Переносимость - означает обеспечение возможности переноса операционной системы с одной аппаратной платформы на другую.
- Совместимость - означает, что операционная система должна включать средства для выполнения приложений (программ), подготовленных для других операционных систем

- Надежность и отказоустойчивость – означает, что 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.
- Безопасность - означает, что операционная система должна содержать средства защиты ресурсов одних пользователей от других, должна обеспечивать удобство внесения последующих изменений и дополнений.
- Производительность. ОС должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа.

3.2 Компоненты ОС.

Современные ОС представляют собой хорошо структурированные модули системы.

Все модули ОС разделяются на две группы:

- основные – модули ядра, выполняющие основные функции ОС (ядро ОС);
- вспомогательные - модули, выполняющие вспомогательные функции ОС (обслуживающие программы и т.д.).

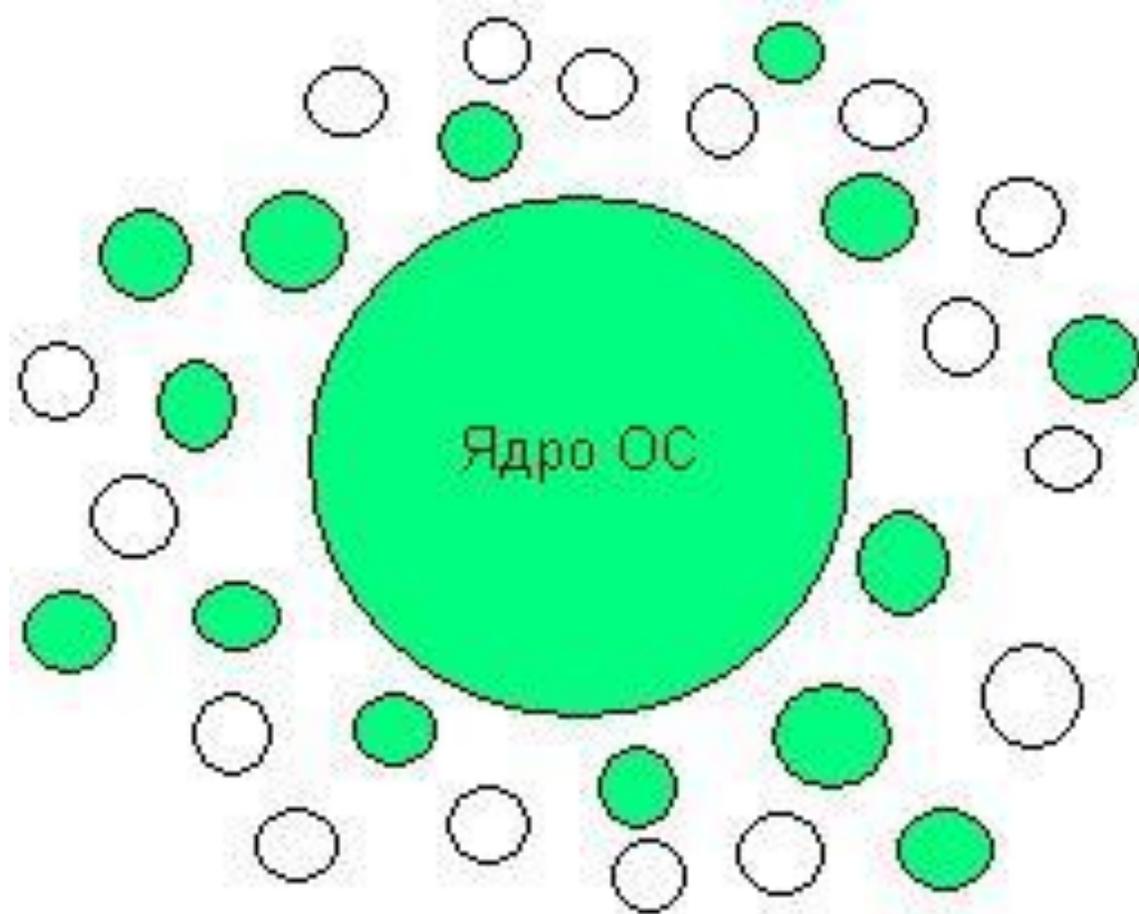
Вспомогательные модули ОС обычно загружаются в оперативную память вычислительной системы только на время выполнения своих функций.

Вспомогательные модули ОС делятся на следующие основные группы:

- 1) *утилиты* – программы, решающие отдельные задачи управления и сопровождения вычислительной системы, например программы архивации данных, сжатия дисков, тестирования дисков и т.д.;
- 2) *системные обрабатывающие программы* – текстовые или графические редакторы, компиляторы, компоновщики, отладчики и другие программы, входящие в комплект поставки данной ОС;

3) *программы предоставления пользователю дополнительных услуг* – нестандартный вариант пользовательского интерфейса, включающий вспомогательные и даже игровые программы;

4) *библиотеки процедур и функций различного назначения, облегчающие разработку пользовательских приложений, например библиотека математических функций, функций работы с устройствами ввода-вывода и т.д.*



● — Вспомогательные модули ОС

○ — Пользовательские приложения

3.2.1 Ядро ОС

Ядро операционной системы (Kernel) – основная часть кода операционной системы, постоянно находящаяся в оперативной памяти, управляющая всей операционной системой.



Слои ядра:

- Средства аппаратной поддержки ОС, то есть те, которые прямо участвуют в организации вычислительных процессов: средства поддержки привилегированного режима, систему прерываний, средства переключения контекстов процессов, средства защиты областей памяти и т. п.

- Машинно-зависимые компоненты ОС. Этот слой образуют программные модули, в которых отражается специфика аппаратной платформы компьютера. Это позволяет разрабатывать вышележащие слои на основе машинно-независимых модулей, существующих в единственном экземпляре для всех типов аппаратных платформ, поддерживаемых данной ОС.

- Базовые механизмы ядра.

Модули данного слоя не принимают решений о распределении ресурсов — они только отрабатывают принятые «наверху» решения, что и дает повод называть их исполнительными механизмами для модулей верхних слоев. Например, решение о том, что в данный момент нужно прервать выполнение текущего процесса А и начать выполнение процесса В, принимается менеджером процессов на вышележащем слое, а слою базовых механизмов передается только директива о том, что нужно выполнить переключение с контекста текущего процесса на контекст процесса В.

- Менеджеры ресурсов. Этот слой состоит из мощных функциональных модулей, реализующих стратегические задачи по управлению основными ресурсами вычислительной системы. Обычно на данном слое работают менеджеры (называемые также диспетчерами) процессов, ввода-вывода, файловой системы и оперативной памяти.

- Каждый из менеджеров ведет учет свободных и используемых ресурсов определенного типа и планирует их распределение в соответствии с запросами приложений. Для исполнения принятых решений менеджер обращается к нижележащему слою базовых механизмов.
- Внутри слоя менеджеров существуют тесные взаимные связи, т. к. для выполнения процессу нужен доступ одновременно к нескольким ресурсам — процессору, области памяти, возможно, к определенному файлу или устройству ввода-вывода.

- Интерфейс системных вызовов. Этот слой является самым верхним слоем ядра и взаимодействует непосредственно с приложениями и системными утилитами, образуя прикладной программный интерфейс операционной системы. Функции API (Application Programming Interface), обслуживающие системные вызовы, предоставляют доступ к ресурсам системы в удобной и компактной форме, без указания деталей их физического расположения. Для осуществления таких комплексных действий системные вызовы обычно обращаются за помощью к функциям слоя менеджеров ресурсов, причем для выполнения одного системного вызова может понадобиться несколько таких обращений.

Все модули ядра или большая их часть постоянно находятся в оперативной памяти для обеспечения высокой скорости работы ОС.

Размер ядра играет важную роль в производительности системы (т.к. оперативная память имеет ограничения).

Ядро ОС содержит: управляющие программы; систему управления файлами; командный процессор; программы, обеспечивающие интерфейс пользователя; драйверы устройств; менеджеры ресурсов.

Современные ОС поддерживают как минимум два режима:

- Пользовательский режим (работа с приложениями, памятью, сетями, безопасностью и т.д.)
- Привилегированный режим - режим ядра (kernel mode) – режим супервизора (supervisor mode) (планирование потоков, обработка прерываний, межпроцессное взаимодействие и т.д.).

Управляющие программы.

Управляющие программы управляют работой ВС, обеспечивая в первую очередь автоматическую смену заданий для поддержания непрерывной работы ЭВМ при переходе от одной программы к другой без вмешательства оператора.

Управляющая программа определяет порядок выполнения обрабатываемых программ и обеспечивает необходимым набором услуг для их выполнения.

Основные *функции управляющих программ:*

- Последовательное или приоритетное выполнение каждой работы.
- Хранение, поиск и обслуживание данных независимо от их организации и способа хранения.

Управляющие программы *делятся* на: программы управления задачами, программы управления данными, программы управления восстановлением.

Программы управления задачами:

- считывают входные потоки задач,
- обрабатывают их в зависимости от приоритета,
- инициируют одновременное выполнение нескольких заданий,
- вызывают процедуры,
- ведут системный журнал.

Программы управления данными обеспечивают способы организации, идентификации, хранения, каталогизации и выборки обрабатываемых данных. Эти программы управляют вводом-выводом данных с различной организацией, объединением записей в блоки и разделением блоков на записи, обработкой меток томов и наборов данных.

Программы управления восстановлением после сбоя обрабатывают прерывания от систем контроля, регистрируют сбои в процессоре и внешних устройствах, формируют записи о сбое в журнале, анализируют возможность завершения затронутой сбоем задачи и переводят систему в состояние ожидания, если завершение задачи невозможно.

Функции ядра:

1. Создание/уничтожение процесса.
2. Переключение процесса из состояния в состояние.
3. Диспетчеризация (регулировка).
4. Приостановка и активизация процессов.
5. Синхронизация процессов.
6. Организация взаимодействия между процессами.
7. Манипулирование блоком управления процессом (БУП).
8. Обработка прерываний.
9. Поддержка операций ввода/вывода.
0. Поддержка распределения и перераспределения памяти.
1. Поддержка работы файловой системы.

3.3 Типы архитектуры ядра ОС.

Архитектура ядра операционной системы – это структурная и функциональная организация ОС на основе некоторой совокупности программных модулей.

Различают:

- Монолитное ядро.
- Модульное ядро.
- Микроядро.
- Экзоядро.
- Наноядро.
- Гибридное ядро.

Д.3.!!!

Дать краткую характеристику каждому типу архитектуры