**Операционные системы**

**1. Монолитные операционные системы. Архитектура монолитной ОС, примеры таких систем. Микроядерные и наноядерные операционные системы.**

**Монолитные операционные системы** объединяют все компоненты ОС в единое ядро, работающее в привилегированном режиме процессора. Этот подход обеспечивает высокую скорость взаимодействия компонентов, но увеличивает уязвимость системы в целом.

**Архитектура монолитной ОС:** - Пользовательские приложения - Системные вызовы - Монолитное ядро (планировщик, файловая система, управление памятью, сетевые службы, драйверы) - Аппаратное обеспечение

**Характеристики**: высокая производительность за счет прямых вызовов функций, сложность отладки из-за тесных взаимосвязей, низкая отказоустойчивость, необходимость перекомпиляции ядра при изменениях.

**Примеры:** Linux, MS-DOS, ранние версии Windows и Unix.

**Микроядерные ОС** содержат минимальное ядро с базовыми функциями, остальные компоненты работают как процессы в пользовательском пространстве. Такая архитектура повышаеzт надежность, поскольку сбой в отдельном компоненте не приводит к краху всей системы.

**Характеристики:** повышенная надежность, модульность, снижение производительности из-за межпроцессного взаимодействия, улучшенная безопасность.

**Примеры:** MINIX, QNX, Windows NT (изначально), GNU Hurd.

**Наноядерные ОС** имеют еще более минимальное ядро, содержащее только механизмы виртуализации оборудования. Они предлагают максимальную модульность и надежность ценой дополнительных накладных расходов на координацию компонентов.

**Примеры:** KeyKOS, EROS, Coyotos.

**2. Архитектура UNIX. Файлы и устройства. Процессы. Драйверы файловой системы и их типы.**

**Основные компоненты UNIX:** - Пользовательские приложения и оболочка Shell - Системные утилиты и вызовы - Ядро (управление процессами/памятью, файловая система, драйверы) - Аппаратное обеспечение Философия UNIX предполагает, что каждая программа должна выполнять одну задачу, но делать это хорошо.

**Файлы и устройства:** действует принцип “всё есть файл”, предоставляющий единый интерфейс для работы с различными ресурсами. Это упрощает разработку приложений и обеспечивает гибкость системы.

**Типы файлов**: - Обычные файлы (данные) - Каталоги (метаданные о файлах) - Специальные файлы устройств (символьные и блочные) - Символические ссылки, именованные каналы, сокеты

**Процессы в UNIX:** - Имеют уникальный идентификатор (PID) - Создаются через fork() и exec() - Образуют иерархию родитель-потомок - Взаимодействуют через сигналы, каналы, сокеты, разделяемую память Состояние процесса меняется от создания до завершения и включает стадии выполнения, ожидания и блокировки. Планировщик ядра определяет, какой процесс получит время процессора.

**Драйверы файловой системы:** - Физические (ext4, XFS, JFS, UFS, ZFS, HFS+) - Виртуальные (procfs, sysfs, tmpfs) - Сетевые (NFS, SMB/CIFS, AFS)

**3. Основы информационной безопасности операционных систем. Концепции безопасности UNIX. Управление пользователями и правами доступа.**

**Концепции безопасности UNIX:** - Многопользовательская модель с разделением доступа - Модель доступа на основе прав - Принцип наименьших привилегий - Изоляция процессов через отдельные адресные пространства - Разделение пользовательского и привилегированного режимов Безопасность UNIX основана на сочетании механизмов контроля доступа, аутентификации и мониторинга.

**Управление пользователями:** - Пользователи имеют уникальный UID - Группы с уникальным GID - Суперпользователь (root, UID 0) - Данные хранятся в /etc/passwd, /etc/shadow, /etc/group

**Права доступа к файлам:** 9-битная система для владельца (u), группы (g) и остальных (o): - Чтение (r, 4) - Запись (w, 2) - Исполнение (x, 1) Для установки прав используется команда chmod с символьной или восьмеричной нотацией.

**Специальные биты:** - SUID (4000): выполнение с правами владельца - SGID (2000): выполнение с правами группы - Sticky bit (1000): удаление только владельцем(защита общих каталогов)

**Списки контроля доступа (ACL)** расширяют стандартную модель прав, позволяя задавать доступ для конкретных пользователей и групп.

**4. Понятие логической файловой системы. Монтирование и демонтирование. Физическая организация файловой системы.**

**Логическая файловая система** – интерфейс между программами и физической организацией данных. Определяет структуру каталогов, правила именования, метаданные, операции с файлами. Она абстрагирует приложения от деталей хранения данных на физических носителях.

**Монтирование** – подключение файловой системы к единому дереву: 1. Проверка целостности структуры 2. Считывание суперблока в память 3. Подключение к указанной точке монтирования

Монтирование может выполняться автоматически при загрузке системы (через /etc/fstab) или вручную (команда mount). Современные системы поддерживают автоматическое монтирование съемных устройств.

**Демонтирование** – отключение файловой системы: 1. Закрытие открытых файлов и сохранение результатов 2. Запись буферов на диск для сохранения целостности 3. Отсоединение от общего дерева файловой системы Принудительное демонтирование (umount -f) может привести к потере данных, если были незавершенные операции записи.

**Физическая организация:** - Суперблок (информация о файловой системе) - Таблица индексных дескрипторов (метаданные файлов) - Блоки данных (содержимое файлов) - Блоки каталогов (связи между именами и inode)

**5. Сеть в UNIX. Сетевая подсистема. Общие принципы работы. Понятие сокетов. Типы сокетов. Общие принципы взаимодействия ОС через сокеты.**

**Сетевая подсистема** включает драйверы сетевых устройств, реализации протоколов и API. Она отвечает за все аспекты сетевого взаимодействия, от управления физическими интерфейсами до поддержки протоколов прикладного уровня.

**Общие принципы:** многоуровневая модель (реализация стека протоколов), маршрутизация (определение пути передачи данных), буферизация (временное хранение для обработки), мультиплексирование (обработка множества соединений).

**Сокет** – интерфейс для межпроцессного взаимодействия через сеть, определяется IP-адресом, портом и протоколом. Абстракция сокета скрывает от приложения сложности сетевого взаимодействия и обеспечивает стандартный потоковый интерфейс.

**Типы сокетов:** - По области: UNIX-сокеты (локальные), Интернет-сокеты (AF\_INET/AF\_INET6) - По соединению: потоковые (TCP), датаграммные (UDP), сырые (низкоуровневые)

**Взаимодействие через сокеты** использует модель клиент-сервер: - Сервер: создание, привязка, прослушивание, принятие - Клиент: создание, подключение - Операции: socket(), bind(), listen(), accept(), connect(), send()/recv(), close()

Сокеты могут работать в блокирующем или неблокирующем режиме. В неблокирующем режиме операции не приостанавливают выполнение программы, что позволяет обрабатывать множество соединений в одном потоке.

**6. Управление службами операционной системы. Загрузка операционной системы. Системные службы.**

**Системные службы (демоны)** – фоновые программы, обеспечивающие функциональность ОС: сетевые службы, печать, журналирование, служба времени, планировщики. Они запускаются автоматически и не требуют взаимодействия с пользователем.

**Системы инициализации:** - SysV init: скрипты в /etc/rc\*.d, уровни выполнения, команды service и chkconfig - Systemd: параллельный запуск, зависимости, активация по запросу, systemctl - Upstart: ориентация на события, команды start/stop/restart

Systemd стал стандартом в большинстве современных дистрибутивов Linux и предлагает более быструю загрузку и продвинутые функции мониторинга. Он заменяет не только SysV init, но и множество других служб (cron, inetd, acpid).

**Загрузка ОС:**

1. BIOS/UEFI проводит самотестирование и инициализирует оборудование

2. Загрузчик (GRUB/LILO) загружается с диска и предоставляет меню выбора ОС

3. Загрузка и инициализация ядра, обнаружение оборудования 4. Монтирование корневой ФС и других необходимых файловых систем

5. Запуск init/systemd как первого процесса (PID 1)

6. Запуск системных служб в соответствии с конфигурацией

7. Запуск графической подсистемы (X Window System/Wayland) 8. Вход пользователя через менеджер входа или консоль