1. **Что такое POSIX?**

POSIX (англ. Portable Operating System Interface — переносимый интерфейс операционных систем) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем.

1. **Что такое системный вызов?**

Системный вызов - механизм вызова прикладной программой функции ядра OS. Системный вызов осуществляется с помощью программного прерывания (RISC/x86 – int) или новый механизм Intel x86\_64 - SYSENTER/SYSEXIT, AMD - SYSCALL/SYSRET).

1. **Что такое аппаратное прерывание, программное прерывание?**

Аппаратные прерывания – события, которые исходят от внешних аппаратных устройств (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши.

Программные прерывания – инициируются исполнением специальной инструкции в коде программы. Программные прерывания, как правило, используются для обращения к функциям встроенного программного обеспечения (firmware), драйверов и операционной системы

1. **Что такое процесс?**

В этом вопросе, как я понимаю, требуют кратко пояснить ответ, ибо дальше будет похожий вопрос, где надо ответить более подробно.

Процесс OS – единица работы OS – объект ядра OS + адресное пространство

1. **Что такое контекст процесса?**

Контекст процесса – данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы.

Контекст процесса: адресное пространство, содержимое регистров (общего назначения, счетчик команд, состояния процессора, вершина стека, …), объекты ядра OS (объекты процессов, потоков, безопасности, файлов и пр.), стек ядра (для этого процесса).

1. **Что такое родительский и дочерний процесс?**

Родительский процесс – процесс, который имеет дочерние процессы. В UNIX подобных системах такие процессы вызвают fork(), что создаёт точную копию (дочерний процесс) данного процесса.

Дочерний процесс – процесс, который был создан другим процессом. Дочерний процесс, как я понимаю, обязан иметь родителя, в ином случае такой процесс называется процесс-сирота.

1. **Что такое процесс инициализации OS?**

Процесс инициализации ОС - это набор действий, которые выполняются операционной системой при запуске компьютера (условно можно поделить на аппаратную загрузку и загрузку ОС).

Аппаратная загрузка:

* Загрузка BIOS или UEFI: BIOS или UEFI - это микропрограмма, которая хранится на материнской плате и отвечает за начальную загрузку компьютера. Она проверяет аппаратное обеспечение системы и загружает загрузчик операционной системы
* Загрузка загрузчика: Загрузчик - это небольшая программа, которая загружает операционную систему в оперативную память

Загрузка OS:

* Загрузка ядра операционной системы в оперативную память
* Инициализация аппаратного обеспечения системы
* Загрузка драйверов устройств
* Создание системных процессов и служб
* Загрузка пользовательского интерфейса (если такой имеется)

1. **Перечислите области памяти процесса и поясните их назначение**



CODE – это область памяти, содержащая машинный код программы

STATIC – это данные, которые выделяются в памяти при запуске программы и остаются в памяти до завершения программы

HEAP – это область памяти, используемая для хранения динамических данных. Динамические данные - это данные, которые выделяются в памяти во время выполнения программы. Куча используется для хранения массивов, объектов и других данных, размер которых не известен заранее

DATA – это область памяти, используемая для хранения глобальных переменных и статических данных. Глобальные переменные доступны из всех функций программы. Статические данные хранятся в памяти до завершения программы

STACK – это LIFO структура данных. Стек используется для хранения локальных переменных и параметров функций. Когда функция вызывается, ее параметры помещаются на стек. Когда функция возвращает управление, ее локальные переменные удаляются со стека

1. **Чем отличаются системные процессы от пользовательских?**

Системные процессы – процессы запускаемые автоматически при запуске OS (СМЕЛОВ)

Пользовательские процессы – соответственно те, которые запускаются по ходу работы пользователя.

1. **Что такое Windows-сервисы, Linux-демоны?**

Windows-сервис – это приложение, которое выполняется в фоновом режиме и не требует взаимодействия пользователя. Службы Windows могут быть запущены автоматически при загрузке компьютера или вручную пользователем. Они могут быть остановлены или приостановлены пользователем или системой.

Выделяют два типа Windows служб:

* **Системные службы** - это службы, которые необходимы для работы операционной системы.
* **Пользовательские службы** - это службы, которые предоставляют дополнительные функции для пользователей.

Linux-демон – да по-факту всё тоже самое, что и Windows-службы, только для Linux. Демоны обычно запускаются автоматически при загрузке системы и остаются запущенными до тех пор, пока система не будет выключена. Они могут быть остановлены или перезапущены пользователем или системой.

1. **С помощью каких системных вызовов можно создать дочерний процесс в Windows? Поясните разницу**

В Windows для создания дочернего процесса можно использовать следующие системные вызовы:

* CreateProcessA() - создает дочерний процесс, используя ASCII-строку для имени образа процесса.
* CreateProcessW() - создает дочерний процесс, используя Unicode-строку для имени образа процесса.
* CreateProcessAsUserA() - создает дочерний процесс, используя учетные данные другого пользователя.
* CreateProcessAsUserW() - создает дочерний процесс, используя учетные данные другого пользователя.

1. **С помощью каких системных вызовов можно создать дочерний процесс в Linux? Поясните разницу**

В Linux для создания дочернего процесса можно использовать следующие системные вызовы:

* fork() – создает точную копию родительского процесса.
* clone() – создает дочерний процесс с возможностью изменения его атрибутов
* vfork() – создает дочерний процесс, который разделяет адресное пространство с родительским процессом (устаревший)
* exec family – основное различие от **fork():** функции семейства exec заменяют текущий процесс новым процессом.
* System – солянка из fork, exec и wait.

1. **Какие потоки данных доступны любому процессу автоматически?**

Любому процессу автоматически доступны три потока: ввода (0), вывода (1), вывод ошибок(2).

1. **Поясните назначение системного вызова WaitForSingleObject в Windows-приложении**

Системный вызов WaitForSingleObject() в Windows-приложении используется для ожидания сигнала от объекта синхронизации. Объект синхронизации - это объект, который используется для координации доступа к ресурсам несколькими потоками или процессами.

WaitForSingleObject() принимает два параметра:

* hHandle - это дескриптор объекта синхронизации.
* dwMilliseconds - это интервал времени в миллисекундах, в течение которого вызывающий поток ожидает сигнала. Если значение dwMilliseconds равно INFINITE, вызывающий поток будет ждать сигнала бесконечно долго

WaitForSingleObject() возвращает следующие значения:

* WAIT\_OBJECT\_0 - объект синхронизации был сигнализирован.
* WAIT\_TIMEOUT - время ожидания истекло.
* WAIT\_FAILED - произошла ошибка.

1. **Поясните назначение системного вызова wait в Linux-приложении**

Системный вызов wait() в Linux-приложении используется для ожидания завершения дочернего процесса.

wait() принимает один параметр:

* status - это указатель на структуру, которая будет содержать статус завершения дочернего процесса.

wait() возвращает следующие значения:

* -1 - произошла ошибка.
* PID - PID дочернего процесса, который завершился.

1. **Дайте развернутое определение процесса OS**