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.

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** - возникают как реакция микропроцессора на физический сигнал от некоторого устройства (клавиатура, системные часы, клавиатура, жесткий диск и т.д.), по времени возникновения эти прерывания асинхронны, т.е. происходят в случайные моменты времени;

программные - вызываются искусственно с помощью соответствующей команды из программы (int), предназначены для выполнения некоторых действий операционной системы, являются синхронными;

1. ***Что такое процесс***?

**Проце́сс** — это в выполняемая в данный момент программа

**процесс OS** – единица работы OS - **объект ядра OS+адресное пространство**

1. ***Что такое контекст процесса?***

Каждый **процесс** UNIX имеет **контекст**, под которым понимается вся информация, требуемая для описания **процесса**. Эта информация сохраняется, когда выполнение **процесса** приостанавливается, и восстанавливается, когда планировщик предоставляет процессу вычислительные ресурсы.

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

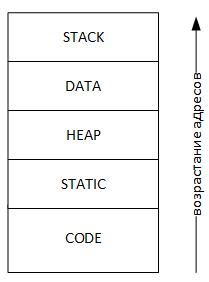
***Родительский процесс*** — это процесс, который создал («породил») один или несколько дочерних процессов («процессов потомков»). За счёт чего, процесс может стать дочерним или родительским и наоборот.( Процесс, который вызвал fork, является родительским процессом, а вновь созданный процесс-дочерним процессом.)

**Дочерний процесс** - это процесс, созданный другим процессом (родительским процессом). Этот метод относится к многозадачным операционным системам и иногда называется подпроцессом или традиционно подзадачей.

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

в OS есть **процесс инициализации** (родитель для всех);

Процесс инициализации запускает все процессы, которые должны быть запущены и является для них родительским процессом

1. ***Перечислите области памяти процесса и поясните их назначение.***
2. 
3. Stack - для временного хранения данных программы, результатов промежуточных вычислений
4. Data – для постоянного хранения инициализированных данных программы
5. Heap – для хранения динамических данных программы
6. Static – для хранения статических данных программы
7. Code – для хранения скомпилированного кода программы

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

**системные** **процессы:** процессы запускаемые автоматически при запуске OS;

**Системные процессы** - процессы, которые запускаются без вашего ведома, т. к. их работа необходима для работы системы, в общем их лучше не трогать, если не знаешь за что отвечает определенный процесс.   
**Пользовательские процессы** - приложения, которые запустил пользователь, т. е процесс запущен от имени пользователя.

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

windows-сервисы; Linux-демоны-процессы запускаемые автоматически при запуске OS, работающая в фоновом режиме без прямого взаимодействия с пользователем.

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

CreateProcess, CreateProcessAsUser, CreateProcessWithTokenW, CreateProcessWithLogonW

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

System

Fork

Execv

system - Функция system() определена в стандартной библиотеке языка С и позволяет вызывать из программы системную команду, как если бы она была набрана в командной строке.

execv - execv вызывает уже скомпилированное приложение, а не создает новое

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

процессу автоматически доступны три потока: ввода, вывода, вывод ошибок.

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

В случае многопоточности иногда нам нужно дождаться завершения потока, прежде чем продолжить выполнение других действий.Для этого вы можете использовать функцию Windows API WaitForSingleObject или WaitForMultipleObjects. Обе эти функции ждут, пока объект не будет помечен как сигнализированный, прежде чем вернуться.

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

Она блокирует вызывающий процесс до тех пор, пока один из его дочерних процессов не завершится (или не произойдет ошибка).

1. ***Дайте развернутое определение процесса OS.***

Совокупность взаимосвязанных системных ресурсов, расположенных в адресном пространстве, в контексте которой организуется выполнение потоков. Если компьютерная программа сама по себе — лишь пассивная последовательность инструкций, то процесс — непосредственное выполнение этих инструкций.