1. **Что такое POSIX?**

Набор стандартов, описывающее интерфейс взаимодействия прикладной программы и ОС.

1. **Что такое системный вызов?**

Системный вызов - механизм вызова прикладной программой функции ядра OS.

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

Прерывание — сигнал от ПО или АО, сообщающий процессору о наступлении какого-либо события, требующего немедленного внимания.

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

Программные (частный случай внутреннего прерывания) — реакция на исполнение специальной [инструкции](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8) в коде [программы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), предназначены для выполнения некоторых действий ОС, являются синхронными;

1. **Что такое процесс?**

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

Процессом, по существу, является программа во время ее выполнения.

1. **Что такое контекст процесса?**

Контекст процесса – данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы; (адресное пространство, содержимое регистров, объекты ядра OS, стек ядра (для этого процесса))

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

Процессы делятся на родительские и дочерние. Родительский процесс – процесс, создавший один или более дочерних процессов, дочерний – созданный родительски. У дочернего процесса может быть только один родительский. Если у дочернего процесса нет родительского, то он был создан непосредственно ядром.

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

Процесс инициализации OS – первый процесс, выполняемый ядром ОС при запуске устройства.

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

Сегменты: code, static, data, heap, stack;

* Stack – стек адресов возврата в точку вызова подпрограммы, регистр SS.
* Data - содержит данные, то есть константы и рабочие области, необходимые программе, адресуется регистром DS;
* Heap – сегмент памяти, где динамически выделяется память в ходе выполнения программы
* Static
* Code - содержит машинные команды, адресуется регистром CS

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

Системные процессы являются частью ядра и всегда находятся в оперативной памяти. Процесс ОС управляет каким-либо ресурсом в интересах прикладного процесса.

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

процессы запускаемые автоматически при запуске OS; Windows: windows-сервисы; Linux-демоны.

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

Функция **CreateProcess**создает новый процесс и его главный поток. Новый процесс выполняет указанный исполняемый файл. Функция CreateProcess предназначена для запуска новой программы. Функции WinExec и LoadModule также работают, но не предоставляют таких возможностей, как CreateProcess.

Функция **LoadModule** загружает и исполняет приложение или создает новый экземпляр существующего приложения.

Самый простой способ запустить какую-то программу из своего приложе­ния — использовать функцию **WinExec.**

1. **С помощью каких системных вызовов можно создать дочерний процесс в Linux? Поясните разницу.**
2. С помощью функции System(). Этот способ сравнительно прост, однако неэффективен и создает определенные риски с точки зрения безопасности.
3. С помощью функций fork() и exec() - более продвинутая техника с точки зрения гибкости, скорости и безопасности.

Команда exec заменяет текущий интерпретатор shell указанной командой. Обычно она используется для того, чтобы закрыть текущий интерпретатор и запустить другой.

Fork – создает копию родительского процесса.

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

Потоки, имеющие зарезервированные дескрипторы: поток ввода (0), поток вывода (1), поток вывода ошибок (2).

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

Вызов *WaitForSingleObject* используется для ожидания события. Если в параметре указан процесс, то вызывающая программа дожидается окончания конкретного процесса.

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.**

Процесс OS – единица работы OS - объект ядра OS + адресное пространство. Процессом, по существу, является программа во время ее выполнения.

С каждым процессом связан набор ресурсов (регистры, список открытых файлов, необработанные предупреждения, список связанных процессов и др. информацию), необходимых в процессе работы программы. Таким образом, процесс — это контейнер, в котором содержится вся информация, необходимая для работы программы.