1. **Что такое фреймворк OS?**

Набор библиотек OS + API интерфейс.

1. **Что такое POSIX?**

POSIX — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов.

1. **Что такое аппаратное прерывание?**

**Прерывание** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *interrupt*) — сигнал от программного или аппаратного обеспечения, сообщающий [процессору](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) о наступлении какого-либо события, требующего немедленного внимания.

асинхронные, или внешние (аппаратные) — 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.

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;

1. **Что такое программное прерывание?**

программные (частный случай внутреннего прерывания) — инициируются исполнением 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 операционной системы, являются синхронными;

1. **Что такое системный вызов?**

Системный вызов - механизм вызова прикладной программой функции ядра OS.

1. **Что такое процесс OS?**

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

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

1. **Что такое контекст процесса OS?**

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

1. **Что такое адресное пространство процесса?**

адресное пространство (данные, программа, стек, куча);

Адресное пространство — это просто диапазон адресов, обозначающих определенное место в памяти.

С каждым процессом связано его

**адресное пространство** — список адресов ячеек памяти от нуля до некоторого максиму-

ма, откуда процесс может считывать данные и куда может записывать их. Адресное про-

странство содержит выполняемую программу, данные этой программы и ее стек. Кроме

этого, с каждым процессом связан набор ресурсов, который обычно включает регистры

(в том числе счетчик команд и указатель стека), список открытых файлов, необрабо-

танные предупреждения, список связанных процессов и всю остальную информацию,

необходимую в процессе работы программы. Таким образом, процесс — это контейнер,

в котором содержится вся информация, необходимая для работы программы.

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

процессу выделяется линейное адресное пространство (размер зависит от разрядности), сегменты: code, static, data, heap, stack;



сегмент кода – содержит машинные команды, Адресуется регистром CS;

сегмент данных – содержит данные, то есть константы и рабочие области, необходимые программе. Адресуется регистром DS;

сегмент стека – содержит адреса возврата в точку вызова подпрограмм. Адресуется регистром SS.

1. **Что такое стандартные потоки процесса?**

потоки имеющие зарезервированные номера - дескрипторы (номера), поток ввода (0), поток вывода (1), поток вывода ошибок (2).

1. **Перечислите системные вызовы Windows для создания процесса?**

CreateProcess, которая создает новый процесс с единственным потоком. При вызове этой функции требуется указать имя файла исполняемой программы

1. **Перечислите системные вызовы Linux для создания процесса?**

Вызов *fork* является единственным существующим в POSIX способом создания нового процесса. Он создает точную копию исходного процесса, включая все дескрипторы файлов, регистры и т. п. После

выполнения вызова *fork* исходный процесс и его копия (родительский и дочерний процессы) выполняются независимо друг от друга.

1. **С помощью каких утилит можно увидеть перечень процессов в Windows?**

tasklist

1. **С помощью каких утилит можно увидеть перечень процессов в Linux?**

Ps

1. **Перечислите свойства процесса OS.**

OS: основные свойства процесса:

* процессу соответствует исполняемый программный файл;
* у процесса есть PID;
* у процесса есть Parent PID;
* в Windows: HANDLE – идентификатор объекта OS;
* в OS есть процесс инициализации (родитель для всех);
* запуск и управление (создать, остановить, …) процессом осуществляется с помощью системных вызовов;
* процессы изолированы друг от друга;
* процессу выделяется линейное адресное пространство (размер зависит от разрядности), сегменты: code, static, data, heap, stack;
* контекст процесса – данные, которые сохраняются при переключении процессов и предназначенные для продолжения работы;
* процессу автоматически доступны три процесса: ввода, вывода, вывод ошибок.
* при запуске OS некоторые процессы (Windows-сервисы, Linux-демоны) загружаются и стартуют автоматически, как правило используются для внутреннего назначения;
* в составе ОS есть таблица, содержащая объекты ядра процессов (состояние, приоритет, указатели на другие объекты); есть средства OS позволяющие ее просматривать;
* процесс – единица работы OS.