1. Что такое фреймворк OS?

Набор функций для взаимодействия с ос

2. 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.

3. Что такое аппаратное 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;

4. Что такое программное прерывание?

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

5. 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).

6. Что такое процесс OS?

Проце́сс — это в выполняемая в данный момент программа

7. Что такое контекст процесса OS?

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.

8. Что такое адресное пространство процесса?

А́дресное пространство (англ. address space) — совокупность всех допустимых адресов каких-либо объектов вычислительной системы — ячеек памяти, секторов диска, узлов сети и т. п., которые могут быть использованы для доступа к этим объектам при определенном режиме работы (состоянии системы).

9. Перечислите области памяти адресного пространства процесса и поясните их назначение.

Stack, Data, Heap, Static, Code

-----возрастание адресов---—

10. Что такое стандартные потоки процесса?

Стандартные потоки ввода-вывода в системах типа UNIX (и некоторых других) — потоки процесса, имеющие номер (дескриптор), зарезервированный для выполнения некоторых «стандартных» функций.

11. Перечислите системные вызовы Windows для создания процесса?

CreateProcess, CreateProcessAsUser, CreateProcessWithTokenW, CreateProcessWithLogonW

12. Перечислите системные вызовы Linux для создания процесса?

Начинается с Fork(), он создает точный клон вызывающего процесса, так называемый «дочерний» процесс

Менеджер исполнения exec() заменяет образ процесса этого клона новой программой, которая должна быть выполнена.

fork() создает новое адресное пространство, которое полностью идентично адресному пространству основного процесса. После выполнения этого системного вызова мы получаем два абсолютно одинаковых процесса - основной и порожденный. Функция fork() возвращает 0 в порожденном процессе и PID (Process ID - идентификатор порожденного процесса) - в основном. PID - это целое число.

Теперь, когда мы уже создали процесс, мы можем запустить программу с помощью вызова exec. Параметрами функции exec является имя выполняемого файла и, если нужно, параметры, которые будут переданы этой программе. В адресное пространство порожденного с помощью fork() процесса будет загружена новая программа и ее выполнение начнется с точки входа (адрес функции main).

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

Для управления процессами в командной строке есть две утилиты — tasklist и taskkill. Первая показывает список процессов на локальном или удаленном компьютере, вторая позволяет их завершить.

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

Ps ps показывает запущенные процессы

Top консольная команда, которая выводит список работающих в системе процессов и информацию о них

Lsof LSOF является аббревиатурой от List Of Opened Files, и предназначена она для отображения открытых файлов различными процессами и/или пользователями

15.

Перечислите свойства процесса OS.

Потоки выполнения отличаются от традиционных процессов многозадачной операционной системы тем, что:

• процессы, как правило, независимы, тогда как потоки выполнения существуют как составные элементы процессов

• процессы несут значительно больше информации о состоянии, тогда как несколько потоков выполнения внутри процесса совместно используют информацию о состоянии, а также память и другие вычислительные ресурсы

• процессы имеют отдельные адресные пространства, тогда как потоки выполнения совместно используют их адресное пространство

• процессы взаимодействуют только через предоставляемые системой механизмы связей между процессами

• переключение контекста между потоками выполнения в одном процессе, как правило, быстрее, чем переключение контекста между процессами.

основные свойства процесса:

- процессу соответствует исполняемый программный файл;

- у процесса есть PID;

- у процесса есть Parent PID;

- в Windows: HANDEL – 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.