1. Что такое POSIX?

Институт инженеров по электротехнике и электронике (IEEE) разработал стандарт системы UNIX, названный POSIX, который в настоящее время поддерживается большинством версий UNIX. Стандарт POSIX определяет минимальный интерфейс системных вызовов, который должны поддерживать совместимые с ним системы UNIX. Фактически на данный момент POSIX-интерфейс поддерживается также рядом других операционных систем

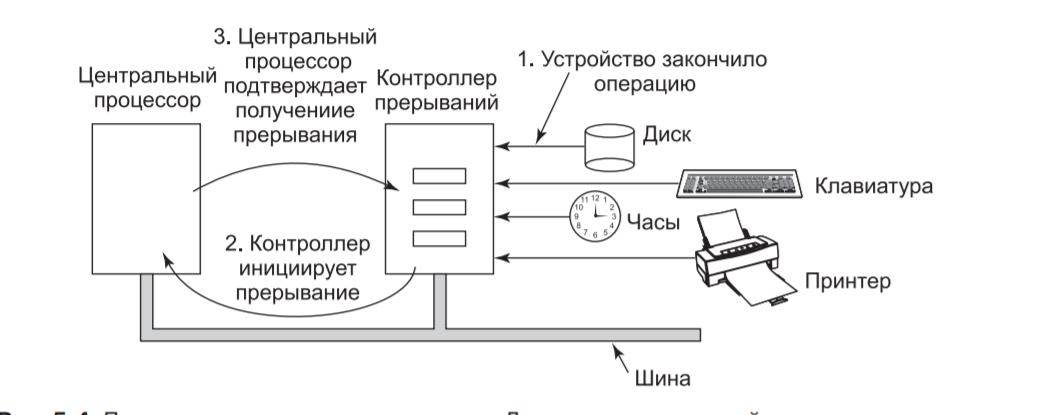
(набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой, библиотеку языка C и набор приложений и их интерфейсов.)

1. Что такое системный вызов?

механизм вызова прикладной программой функции ядра OS

1. Что такое аппаратное прерывание, программное прерывание?

реакция процессора 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, программное прерывание - вызвано либо исключительным состоянием в самом процессоре, либо специальной инструкцией в наборе команд, которая вызывает прерывание, когда инструкция выполняется , генерируемое в процессоре путем выполнения инструкции. Программные прерывания часто используются для реализации системных вызовов, поскольку они приводят к вызову подпрограммы с изменением уровня вызова ЦП



1. Что такое процесс?

единица работы OS - объект ядра

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

1. Что такое 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 – тот который имеет дочерний , дочерний процесс тот который имеет родителя ☺

Системный вызов fork создает точную копию исходного процесса, называемого родительским процессом (parent process). Новый процесс называется дочерним процессом (child process).

1. Что такое процесс инициализации OS?

сложный и многошаговый процесс запуска компьютера

При запуске операционной системы создаются, как правило, несколько процессов. Некоторые из них представляют собой высокоприоритетные процессы, то есть процессы, взаимодействующие с пользователями и выполняющие для них определенную работу. Остальные являются фоновыми процессами, не связанными с конкретными пользователями, но выполняющими ряд специфических функций.

1. Перечислите области памяти процесса и поясните их назначение.

В UNIX память каждого процесса делится на три сегмента: текстовый сегмент (то есть код программы), сегмент данных (переменные) и сегмент стека. Как показано на рис. 1.18, сегмент данных растет вверх, а стек растет вниз. Между ними существует часть неиспользованного адресного пространства. Стек заполняет пустое пространство автоматически по мере надобности.



1. Чем отличаются системные процессы от пользовательских?

Системные процессы являются частью ядра и всегда находятся в оперативной памяти. Такие процессы не имеют соответствующих им программ в виде исполняемых файлов и запускаются особым образом при инициализации ядра системы.

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

Фоновые процессы

приложение, автоматически исполняемое системой при запуске операционной системы Windows и выполняющиеся вне зависимости от статуса пользователя. -- ??

Демоны отличаются от обычных процессов только тем, что они работают в неинтерактивном режиме. Если с обычным процессом всегда ассоциирован какой-то терминал или псевдотерминал, через который осуществляется взаимодействие процесса с пользователем, то демон такого терминала не имеет. Демоны обычно используются для выполнения сервисных функций, обслуживания запросов от других процессов, причем не обязательно выполняющихся на данном компьютере. Пользователь не может непосредственно управлять демонами, он может влиять на их работу, только посылая им какие-то задания

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

CreatProcess() – данный вызов имеет множество параметров, многие из которых «по умолчанию»

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

Fork() и exec() и system()

После вызова Fork() создается почти такой же РСВ (блок управления процессом), только будет другой идентификатор у процесса, адресное пространство аналогичное.

Системный вызов fork возвращает нулевое значение для дочернего процесса и равное идентификатору дочернего процесса или PID — для родительского. Используя возвращенное значение PID, два процесса могут определить, какой из них родительский, а какой — дочерний.

Идентификатор берется из таблицы(она есть у каждой ОС).

Еxec() не создает новый процесс, а заменяет данные текущего процесса новыми данными.

1. Какие потоки данных доступны любому процессу автоматически?

автоматически создается **основной (main) поток** (выполняется функция ядра, создающая поток). В простейшем – один поток процесса

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

Вызов WaitForSingleObject используется для ожидания события. Ожидание может касаться множества возможных событий. Если в параметре указан процесс, то вызывающая программа дожидается окончания конкретного процесса.

Ожидает завершения 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:

- создается ядром OS по системному вызову;

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

- ресурсы: регистры, открытые файлы, родительский процесс, перечень связанных (дочерних) процессов, реальные страницы памяти, виртуальное адресное пространство, маркеры доступа (безопасность);

- процесс может создавать (с помощью системного вызова) дочерние процессы, в общем случае может образовываться дерево процессов;

- обычно соответствует работающей программе (например, exe-файлу в Windows);

- OS хранит список (или таблицу) объектов работающих процессов;

- при приостановке процесса в объекте процесса сохраняется вся информация (регистры, уведомления OS,… - контекст процесса), позволяющая возобновить работу процесса;

- процессы изолированы друг от друга;

- для обмена данными между процессами, применяется специальный механизм – механизм межпроцессного взаимодействия (IPC);