**Задание 08.ответьте на следующие вопросы**

1. Что такое POSIX?

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 Что такое аппаратное прерывание  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, 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**

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). Переключение контекстов. В ядре специальный стек для переключения контекстов.

Что такое родительский и дочерний процесс?

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

1. Перечислите области памяти процесса и поясните их назначение.



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?

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.

**NtCreatProcess()** – 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.

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?
2. Поясните назначение системного вызова WaitForSingleObject в Windows-приложении.

Функция WaitForSingleObject() приостанавливает выполнение потока до тех пор, пока не произойдет одно из двух событий:

* истечет таймаут ожидания;
* ожидаемый объект перейдет в сигнальное (signaled) состояние.

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.
2. **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);