1. POSIX (Portable Operating System Interface) - это набор стандартов, определенных IEEE, который обеспечивает совместимость между различными операционными системами UNIX и UNIX-подобными системами. POSIX определяет интерфейсы программирования приложений (API), системные вызовы и другие компоненты, которые позволяют приложениям быть переносимыми между различными операционными системами.
2. Аппаратное прерывание - это сигнал, отправляемый процессором или другим аппаратным устройством для прерывания текущей работы процесса и передачи управления обработчику прерывания. Аппаратные прерывания используются для обработки событий, таких как нажатие клавиши на клавиатуре, завершение операции ввода-вывода или сигнал от таймера.
3. Программное прерывание - это сигнал, отправляемый программным образом для прерывания текущей работы процесса и передачи управления обработчику прерывания. Программные прерывания используются для выполнения определенных задач, таких как обработка ошибок, синхронизация процессов или вызов системных функций.

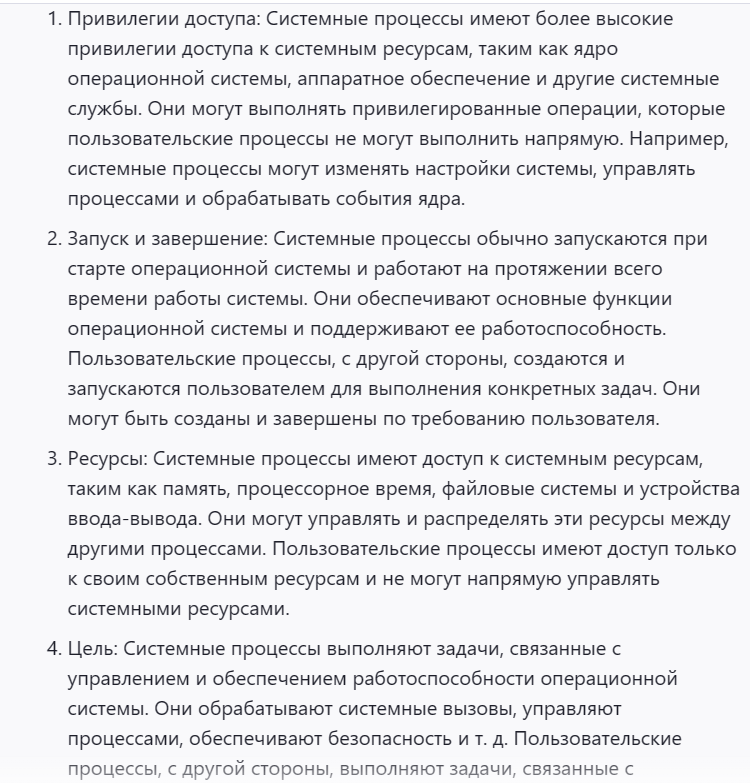
Системный вызов - это механизм, который позволяет приложениям взаимодействовать с операционной системой и использовать ее функциональность. Системный вызов является интерфейсом между приложением и ядром операционной системы. Когда приложение вызывает системный вызов, ядро операционной системы выполняет определенную операцию, например, создание файла или чтение данных из файла, и возвращает результат выполнения операции обратно в приложение

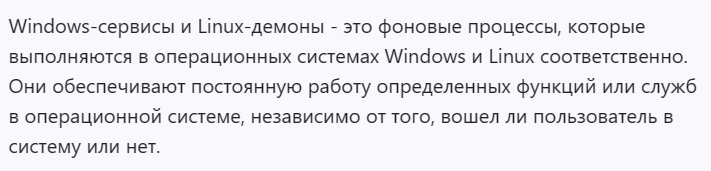
12.Процесс OS (операционной системы) - это экземпляр выполняющейся программы, которая имеет свою собственную память, регистры и другие ресурсы. Процесс является основной единицей планирования и управления в операционной системе, и каждый процесс выполняется в своем собственном изолированном адресном пространстве.

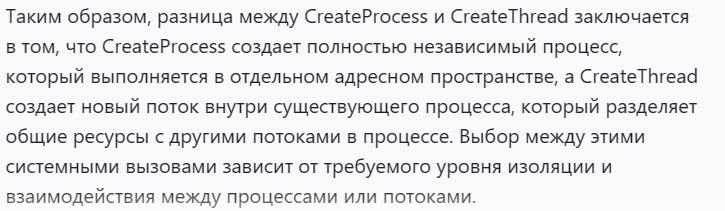
13.Контекст процесса OS - это состояние процесса, включающее значения регистров процессора, указатель стека, открытые файлы, информацию о памяти и другие ресурсы, необходимые для продолжения выполнения процесса после прерывания. Контекст процесса сохраняется при переключении между процессами и восстанавливается при возврате к выполнению процесса.

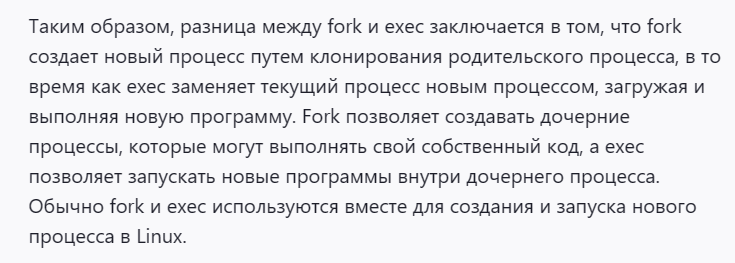
14.Процесс инициализации - это первый процесс, который запускается при загрузке операционной системы. Он имеет особое значение, поскольку его задача состоит в подготовке и настройке системы для работы и запуске других процессов.

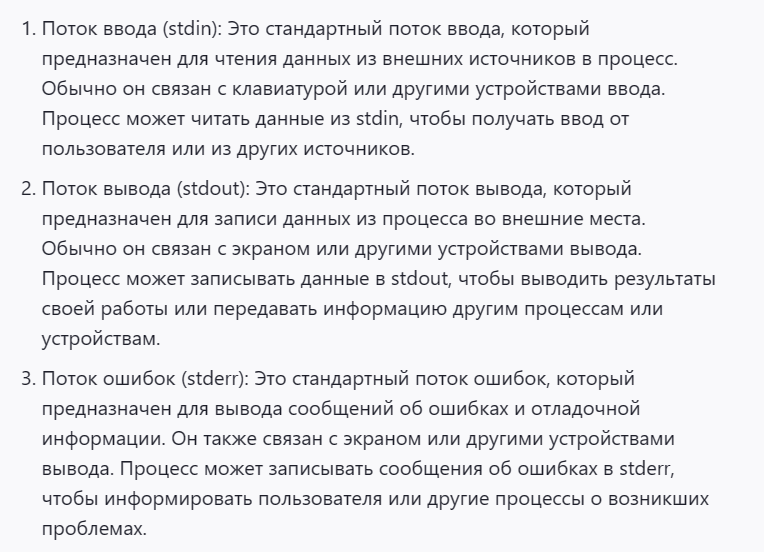
15.Области памяти адресного пространства процесса включают:

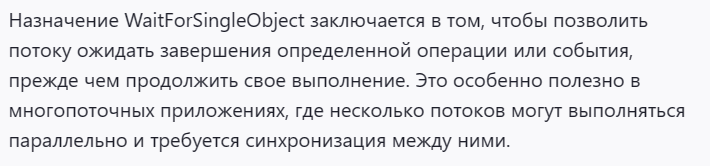
* Код программы: область памяти, содержащая исполняемый код программы.
* Данные: область памяти, используемая для хранения глобальных переменных и статических данных программы.
* Стек вызовов: область памяти, используемая для хранения локальных переменных и данных вызовов функций.
* Куча: область памяти, используемая для динамического выделения памяти во время выполнения программы.
* Остальная память: другие области памяти, такие как память ядра операционной системы или разделяемая память между процессами
* 

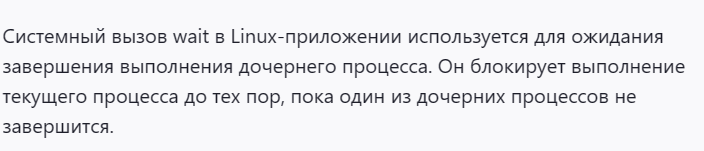












16.Стандартные потоки процесса - это три основных потока ввода-вывода, связанные с каждым процессом: стандартный ввод (stdin), стандартный вывод (stdout) и стандартный вывод ошибок (stderr). Стандартный ввод используется для чтения данных из процесса, стандартный вывод - для вывода данных, а стандартный вывод ошибок - для вывода сообщений об ошибках.

1. Некоторые системные вызовы Windows для создания процесса включают:

* CreateProcess: создает новый процесс и запускает исполняемый файл.
* CreateThread: создает новый поток внутри процесса.
* VirtualAlloc: выделяет виртуальную память для процесса.
* SetProcessAffinityMask: устанавливает маску аффинности процесса для определения процессоров, на которых он может выполняться.

1. Некоторые системные вызовы Linux для создания процесса включают:

* fork: создает новый процесс, являющийся точной копией родительского процесса.
* exec: заменяет текущий процесс новым процессом, загружая и выполняя новую программу.
* clone: создает новый процесс или поток с определенными параметрами.
* mmap: отображает файл или устройство в память процесса.

20.В Windows можно использовать такие утилиты, как Диспетчер задач (Task Manager), Командную строку (Command Prompt) с командой tasklist или PowerShell с командой Get-Process.

21.В Linux можно использовать утилиты, такие как ps, top или htop, чтобы увидеть перечень процессов.

1. Свойства процесса OS могут включать:

* Идентификатор процесса (PID): уникальный числовой идентификатор, присвоенный процессу операционной системой.
* Состояние процесса: указывает, выполняется ли процесс, приостановлен или завершен.
* Приоритет процесса: определяет относительную важность процесса для планировщика операционной системы.
* Родительский процесс: указывает на процесс, который создал текущий процесс.
* Используемые ресурсы: информация о памяти, процессорном времени и других ресурсах, используемых процессом.
* Открытые файлы: список файлов, открытых процессом для чтения или записи.
* Сегменты памяти: информация о различных областях памяти, используемых процессом, таких как код, данные и стек.