1. **Поясните понятие «мультизадачная OS с вытеснением».**

ОС, которая поддерживает выполнение нескольких задач одновременно и использует алгоритмы планирования, которые могут вытеснять текущую задачу с ЦП.

1. **Поясните понятие «циклическое планирование».**

Алгоритм планирования, при котором каждая задача получает фиксированный интервал времени для выполнения на процессоре. По истечении кванта времени процесс переходит в конец очереди планирования, и следующий процесс начинает выполнение.

1. **Поясните понятие «приоритетное планирование».**

Алгоритм планирования, который назначает приоритеты процессам.

1. **Поясните понятие «кооперативное планирование».**

Алгоритм планирования, при котором задача выполняется до тех пор, пока добровольно не освободят процессор.

1. **Поясните понятие «OS реального времени».**

Система, которая гарантирует фиксированное время для выполнения задачи.

1. **Поясните понятие «приоритет процесса».**

Числовое значение, которое определяет, насколько важен процесс для ОС.

1. **Поясните выражение «поток уступает процессор другому потоку».**

Поток добровольно переходит в состояние ожидания, чтобы другой поток мог получить доступ к процессору.

1. **Windows: как поток может уступить процессор?**

Sleep() – процесс становится в конец очереди.

1. **Windows: что такое базовый приоритет потока, как он вычисляется и диапазон его изменения?**

Сочетание класса приоритета процесса и приоритета потока, изменяется в пределах [1,31], по умолчанию – 8, приоритеты возрастающие;

1. **Windows: поясните назначение и принцип применения системного вызова ResumeThread.**

Для возобновления выполнения потока, который был приостановлен с помощью SuspendThread.

1. **Windows: поясните назначение и принцип применения системного вызова WaitForSingleObject.**

Для ожидания завершения выполнения объекта.

1. **Windows: поясните назначение и принцип применения системных вызовов.**

GetProcessPriorityBoost, GetThreadPriorityBoost позволяют получить текущие значения авто повышения приоритета для процесса и потока.

SetProcessPriorityBoost, SetThreadPriorityBoost задают это разрешение.

1. **Linux: поясните принцип идентификации процессов и потоков и поясните, почему он такой.**

Каждый процесс имеет уникальный PID, и каждый поток внутри процесса имеет уникальный TID, но PID и TID могут совпадать в разных процессах. Позволяет ОС однозначно идентифицировать процессы и потоки.

Номер процесса всегда равен номеру главного потока, потому что ядро не различает потоки и процессы.

1. **Linux: Поясните понятие «планировщик потоков».**

Алгоритм планирования, который определяет порядок выполнения потоков основываясь на приоритетах.

1. **Linux: поясните принцип использования значения nice – процесса, диапазон его изменения, для какого режима работы планировщика это значение применяется?**

nice – значение любезности, чем меньше, тем выше приоритет

nice – диапазон приоритетов [-20, 19], default = 0. Режим: SCHED\_OTHER – по умолчанию.

1. **Linux: перечислите политики планирования, какая действует по умолчанию?**

Linux, политики планирования процесса: 1) стандартная (OTHER); - по умолчанию 2) FIFO (приоритетное); 3) карусельная (RR round-robin) – циклическое; 4) пакетная политика(BATCH).

1. **Linux: с помощью какого системного вызова поток может уступить процессор.**

sched\_yield().