1. Дайте определение понятию «синхронизация потоков».

Понятие "синхронизация потоков" относится к координации выполнения нескольких потоков в многопоточной программе. Оно описывает методы и механизмы, используемые для упорядочивания доступа к общим ресурсам или обмена информацией между потоками. Синхронизация потоков позволяет избежать гонок данных (race conditions) и других проблем, возникающих при параллельном выполнении кода.

1. Объясните понятие «взаимная блокировка».

"Взаимная блокировка" (deadlock) возникает, когда два или более потока находятся в состоянии ожидания ресурсов, которые контролируют другие потоки. Каждый из этих потоков блокирует ресурс, который требуется другому потоку для продолжения выполнения, при этом сам ожидает ресурс, контролируемый другим потоком. В результате ни один из потоков не может продолжить свое выполнение, поскольку он ожидает освобождения ресурса, которое не произойдет. Взаимная блокировка является нежелательным состоянием и может привести к замедлению или даже зависанию программы.

1. Перечислите механизмы авторизации OS.

Механизмы авторизации операционной системы (OS) включают в себя следующие методы:

- Парольная авторизация: Пользователь должен предоставить правильный пароль для доступа к системе или определенным ресурсам.

- Авторизация по ключу: Используется криптографический ключ для проверки подлинности и авторизации пользователя.

- Авторизация на основе сертификатов: Пользователь предоставляет цифровой сертификат, который содержит информацию о его подлинности, выпущенный доверенным удостоверяющим центром.

- Биометрическая авторизация: Используются физиологические или поведенческие характеристики пользователя, такие как отпечатки пальцев, сканирование сетчатки глаза или распознавание голоса.

- Авторизация на основе ролей: Пользователь получает доступ к определенным ресурсам в зависимости от своей роли или привилегий в системе.

1. Поясните в чем разница между механизмом **mutex** и **semaphore**.

Разница между механизмом mutex и semaphore заключается в их функциональности и способе использования:

- Mutex (мьютекс) применяется для обеспечения взаимного исключения, то есть только один поток может захватить мьютекс и получить доступ к защищаемому ресурсу. Остальные потоки будут ожидать освобождения мьютекса. Mutex может быть разблокирован только тем потоком, который его захватил.

- Semaphore (семафор) используется для контроля доступа к определенному количеству ресурсов. Семафор содержит счетчик, который отслеживает количество доступных ресурсов. Потоки могут захватывать и освобождать семафоры. Если счетчик семафора равен нулю, поток будет ожидать освобождения ресурса.

Основная разница между mutex и semaphore заключается в их использовании. Mutex применяется для обеспечения взаимного исключения в отношении конкретного ресурса, в то время как semaphore используется для контроля доступа к определенному количеству ресурсов.

1. Почему **mutex,** **semaphore, event** создают объект ядра OS, а **critical section** нет.

Объекты ядра операционной системы (OS) создаются для обеспечения низкоуровневых механизмов синхронизации и взаимодействия между потоками и процессами. Mutex, semaphore и event являются примитивами синхронизации, которые требуют поддержки ядра для своего функционирования. Они являются общими для различных приложений и могут использоваться в разных контекстах.

Критическая секция (critical section) - это участок кода, который должен быть выполнен только одним потоком в определенный момент времени. Критическая секция не требует создания объекта ядра OS. Вместо этого она контролируется самим программным кодом и обеспечивает взаимное исключение путем использования механизмов синхронизации, таких как мьютексы или семафоры.

Поскольку критическая секция контролируется непосредственно программой, а не ядром операционной системы, ее использование более быстрое и эффективное в сравнении с примитивами синхронизации, требующими взаимодействия с ядром. Однако, это также означает, что программисту необходимо самостоятельно обеспечить правильное использование критической секции и избежать возможных проблем, таких как гонки данных или взаимная блокировка.