

# Лекция 8

## Практическое задание по лекции

Используя учебный материал лекции, составьте терминологический словарь, состоящий по объему из 20 терминов и определений.

1. Взаимоисключение (mutual exclusion) - техника синхронизации, которая предотвращает одновременный доступ нескольких потоков к разделяемым ресурсам, обеспечивая эксклюзивный доступ к данным в определенный момент времени.
2. Критическая секция (critical section) - участок кода, который использует общие ресурсы и требует взаимного исключения для предотвращения конфликтов доступа к этим ресурсам.
3. Гонка (race condition) - ситуация, когда несколько потоков пытаются изменить общий ресурс одновременно, приводя к неопределенному или непредсказуемому результату.
4. Блокировка (locking) - механизм, который позволяет потоку получить эксклюзивный доступ к ресурсу, установив блокировку на нем, и освободить ресурс, сняв блокировку после завершения работы с ним.
5. Очередь (queue) - структура данных, которая используется для организации доступа к разделяемым ресурсам в многопоточной среде, где каждый поток ставится в очередь на ожидание доступа к ресурсу.
6. Атомарность - это свойство операции, которое означает, что она выполняется целиком или не выполняется вовсе. Если операция является атомарной, то другие потоки не могут перехватить ее выполнение посередине.
7. Мьютексы - объекты ядра, используемые для синхронизации доступа к единственному ресурсу. Инициализация мьютекса осуществляется функцией CreateMutex, для входа в критическую секцию используется функция WaitForSingleObject, а для выхода - ReleaseMutex. Поток, занявший мьютекс, получает права на владение им, и только этот поток может освободить мьютекс.
8. CreateSemaphore - функция, используемая для создания семафора. В число параметров функции входят начальное и максимальное значение переменной.
9. Синхронизация - это механизм обеспечения взаимодействия между потоками, который позволяет им совместно использовать общие ресурсы, избегая при этом конфликтов и ошибок.
10. Test and Set Lock (TSL) команды - инструкции вычислительных архитектур, позволяющие обеспечить атомарность последовательности операций при входе в критическую секцию. Эти команды выполняются быстро и используются для решения проблемы взаимного исключения при доступе к общим ресурсам.
11. Interlocked-функции - семейство функций, входящих в состав Win32 API, позволяющих выполнять операции атомарно и обеспечивать синхронизацию между потоками. Они реализованы на уровне операционной системы и зависят от аппаратной платформы.
12. InterlockedExchangeAdd - функция из семейства Interlocked-функций, позволяющая атомарно увеличивать значение переменной. Она применяется для обеспечения корректного приращения переменной в многопоточной среде.
13. InterlockedCompareExchange - функция из семейства Interlocked-функций, позволяющая атомарно проверять и модифицировать значение переменной в многопоточной среде. Она может быть использована в качестве альтернативы TSL-инструкции для решения проблемы взаимного исключения при доступе к общим ресурсам.
14. EnterCriticalSection - функция API ОС Windows для входа в критическую секцию потоков одного процесса в режиме пользователя. Реализована на основе Interlocked-функций и работает очень быстро.
15. LeaveCriticalSection - функция API ОС Windows для выхода из критической секции потоков одного процесса в режиме пользователя.

16. InitializeCriticalSectionAndSpinCount - функция API ОС Windows для инициализации критической секции с заданным числом циклов спин-блокировки перед входом в критическую секцию.

17. Спин-блокировка (spin lock) - поведение потока, при котором он продолжает опрос значения переменной-замка в пустом цикле перед входом в критическую секцию, связанное с его вращением в пустом цикле. Используется для ожидания освобождения критической секции на многопроцессорных машинах, где один поток крутится в цикле, а другой работает на другом процессоре и может изменить значение переменной-замка.

18. Объект ядра - структура данных, управляемая операционной системой, которая используется для синхронизации потоков различных процессов. Объекты ядра могут иметь имя, тайм-аут для времени ожидания и обладают рядом возможностей для реализации гибких сценариев синхронизации.

19. Сигнальное состояние (signaled state) - состояние объекта ядра, при котором он доступен для использования потоками, т.е. свободен.

20. Незанятое состояние (nonsignaled state) - состояние объекта ядра, при котором он не доступен для использования потоками, т.е. занят.