**Вопросы к зачету по курсу параллельная обработка данных**

1. Процессы и потоки. Сценарии использования.
2. Критические секции
3. Ошибки соревнования (race errors)
4. Примитивы взаимоисключения, почему недостаточно обычных переменных
5. Мертвые блокировки (deadlocks) и способы их избежать
6. Ошибка потерянного пробуждения (Lost wakeup bug)
7. Семафоры
8. Транзакции
9. Реактивное программирование: Основная идея
10. Реактивное программирование: Паттерн наблюдатель (observer) и его реализация в Rx.
11. Реактивное программирование: Push и Pull модели
12. Реактивное программирование: Преимущества и недостатки
13. Реактивное программирование: Ориентированность на события
14. Реактивное программирование: Эластичность
15. Реактивное программирование: Отказоустойчивость
16. Реактивное программирование: Отзывчивость
17. CQRS: Основные идеи
18. Event Sourcing: Основные идеи
19. Создание потоков в .net. Основные возможности класса Thread.
20. Атомарные операции в .net, класс Interlocked
21. Модификатор volatile
22. Блокировка в .net c помощью lock и Monitor
23. Раздельная блокировка чтения и записи с помощью ReaderWriterLock
24. Пул потоков .net
25. Асинхронные операции в .net. Три способа получения результата.
26. Блокировки средствами ядра
27. Task parallel library, основные возможности класса Task
28. Создание асинхронных методов с помощью async и await
29. Использование SemaphoreSlim для асинхронных блокировок
30. Неблокирующие коллекции (из пространства имен ConcurrentCollections)