




ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

Включил камеру - спас занятие!



Правила вебинара



Активно участвуем и включаем камеры (+83%)



Задаем вопросы голосом или в чат



Off-topic обсуждаем в Telegram



Вопросы в чате вижу, могу ответить не сразу

Взаимодействие потоков



Ягур Алексей
Руководитель курса

Маршрут вебинара

Примитивы взаимодействия



Дополнительные инструменты



Практика



Тестирование



Не забыть включить запись!



Взаимодействие потоков



Ягур Алексей
Руководитель курса

Маршрут вебинара

Примитивы взаимодействия



Дополнительные инструменты



Практика

Взаимодействие потоков

- `Monitor`
- `CountdownEvent`
- `Barrier`
- `AutoResetEvent`
- `ManualResetEvent`
- `ManualResetEventSlim`

Взаимодействие потоков

- **Monitor**
- **CountdownEvent**
- **Barrier**
- **AutoResetEvent**
- **ManualResetEvent**
- **ManualResetEventSlim**

Monitor

```
object _lock = new object();
Thread t1 = new Thread(T1);
t1.Start();

void T1()
{
    Console.WriteLine("T1");
    lock (_lock)
    {
        Console.WriteLine("T1 entered");
        Thread.Sleep(1000);
        Console.WriteLine("T1 wait");
        Monitor.Wait(_lock);
        Console.WriteLine("T1 after wait");
    }
    Console.WriteLine("T1 left");
}
```

```
Thread t2 = new Thread(T2);
t2.Start();

void T2()
{
    Console.WriteLine("T2");
    lock (_lock)
    {
        Console.WriteLine("T2 entered");
        Thread.Sleep(1000);
        Console.WriteLine("T2 pulse");
        Monitor.Pulse(_lock);
        Console.WriteLine("T2 after pulse");
    }
    Console.WriteLine("T2 left");
}
```




Демо

Взаимодействие потоков

- **Monitor**
- **CountdownEvent**
- **Barrier**
- **AutoResetEvent**
- **ManualResetEvent**
- **ManualResetEventSlim**

CountdownEvent

```
CountdownEvent countObject = new CountdownEvent(0);
var elementsCount = 10;
int[] input = Enumerable.Range(0, elementsCount).ToArray();
int[] result = new int[elementsCount];

for (int i = 0; i < elementsCount; ++i)
{
    countObject.AddCount();
    int j = i;
    Task.Factory.StartNew(() =>
    {
        Thread.Sleep(1000); // Имитируем вычисления
        result[j] = j * 59 - 16 + 42;
        countObject.Signal();
    });
}

countObject.Wait();
result.Dump();
```

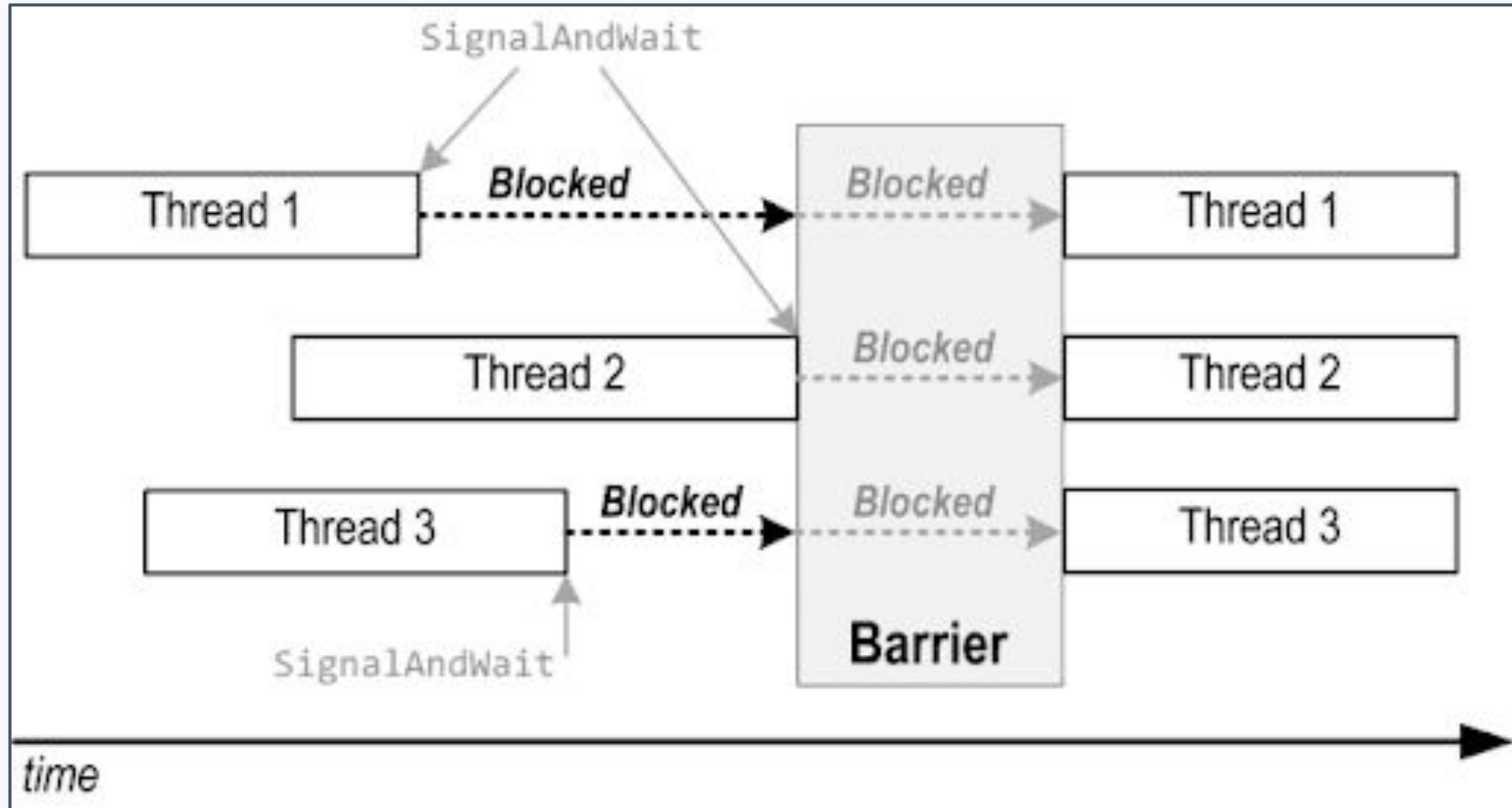



Демо

Взаимодействие потоков

- **Monitor**
- **CountdownEvent**
- **Barrier**
- **AutoResetEvent**
- **ManualResetEvent**
- **ManualResetEventSlim**

Barrier



Barrier

```
Barrier bar;
int numberOfWorkers = 10;

void Main()
{
    bar = new Barrier(numberOfWorkers, AnnounceEndOfPhase);
    Parallel.For(0, numberOfWorkers, DoWork);
}

void DoWork(int id)
{
    Thread.Sleep(new Random().Next(500, 1000));
    Console.WriteLine($"[Фаза {bar.CurrentPhaseNumber}] Часть данных №{id} успешно скачана");
    bar.SignalAndWait();
    Thread.Sleep(new Random().Next(500, 1000));
    Console.WriteLine($"[Фаза {bar.CurrentPhaseNumber}] Обучение на данных №{id} завершено");
    bar.SignalAndWait();
    Thread.Sleep(new Random().Next(500, 1000));
    Console.WriteLine($"[Фаза {bar.CurrentPhaseNumber}] Серия тестов №{id} прошла успешно");
    bar.SignalAndWait();
}
```




Демо

Взаимодействие потоков

- `Monitor`
- `CountdownEvent`
- `Barrier`
- `AutoResetEvent`
- `ManualResetEvent`
- `ManualResetEventSlim`

AutoResetEvent

— примитив синхронизации, автоматически блокирующий движение следующих потоков после пропуска одного, находившегося в состоянии ожидания.

© Алексей Ягур



AutoResetEvent

```
AutoResetEvent autoResetEvent = new AutoResetEvent(true);

Parallel.For(0, 10, DoWork);

// Программа для запоминания 10 слов
void DoWork(int id)
{
    Console.WriteLine($"Поток {id} генерируем звучание слова");
    Thread.Sleep(new Random().Next(500, 1000)); // Генерируем звучание
    autoResetEvent.WaitOne();
    Console.WriteLine($"[{id}] Воспроизводим звук");
    Thread.Sleep(new Random().Next(1500, 3000)); // Воспроизводим звук
    autoResetEvent.Set();
}
```




Демо

Взаимодействие потоков

- **Monitor**
- **CountdownEvent**
- **Barrier**
- **AutoResetEvent**
- **ManualResetEvent**
- **ManualResetEventSlim**

ManualResetEvent

— примитив синхронизации, разрешающий движение всех ожидающих потоков после его “открытия”.

Движение запрещается только после закрытия вручную.

© Алексей Ягур



ManualResetEvent

```
ManualResetEvent manualResetEvent = new ManualResetEvent(true);

Parallel.For(0, 10, DoWork);

void DoWork(int id)
{
    Console.WriteLine($"Поток {id} генерируем звучание слова");
    Thread.Sleep(new Random().Next(500, 1000)); // Генерируем звучание
    manualResetEvent.WaitOne();
    Console.WriteLine($"[{id}] Воспроизводим звук");
    Thread.Sleep(new Random().Next(1500, 3000)); // Воспроизводим звук
    manualResetEvent.Set();
    manualResetEvent.Reset();
}
```


The background of the slide is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The entire image is tinted with a blue color scheme. A horizontal band across the middle of the image features a gradient from teal on the left to dark blue on the right, overlaid with a faint, white geometric network pattern of lines and dots.

Демо

Маршрут вебинара

Примитивы взаимодействия



Дополнительные инструменты



Практика

Дополнительные инструменты

→ **Interlocked**

→ **Volatile**

Interlocked example

```
int x = 0;

Parallel.For(0, 10000, i => x++);
Parallel.For(0, 10000, i => Interlocked.Increment(ref x));
Parallel.For(0, 10000, i => Interlocked.Decrement(ref x));
Interlocked.Add(ref x, 50);

x.Dump();
```


Volatile example

```
static bool stop;

void Main()
{
    var t = Task.Run(Function);

    Thread.Sleep(1000);

    stop = true;
    Console.WriteLine("Ждём поток...");

    t.Wait();
    Console.WriteLine("Дождались поток! =)");
}

void Function()
{
    int x = 0;

    while (stop == false)
    {
        x++;
    }
    Console.WriteLine(x);
}
```

Маршрут вебинара

Примитивы взаимодействия



Дополнительные инструменты



Практика




Практика



Тестирование



The background of the slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white geometric network pattern of lines and dots. A horizontal band of solid blue color runs across the middle of the slide, serving as a background for the text.

**Заполните, пожалуйста,
опрос о занятии по ссылке в
чате**

Ответы на вопросы



Ягур Алексей
Руководитель курса