

Параллельное программирование

Т. П. ГРЫЗЛОВА

ПРОЕКТИРОВАНИЕ ПРОГРАММ НА C#.

ЧИТАТЕЛИ - ПИСАТЕЛИ

РГАТА им. П. А.
Соловьева

```
static Semaphore rw1;  
    static Semaphore en;  
    static int rCount;
```

```
rw1 = new Semaphore(1, 1);  
    en = new Semaphore(1, 1);  
    rCount = 0;
```

```
static void ToWrite(object num)  
    {while (iteration < NI)  
        { rw1.WaitOne();  
          Thread.Sleep(500);  
          iteration++;  
          rw1.Release(1);  
        }//while  
    }//Writer
```

```

static void ToRead(object num)
{
    while (iteration < NI)
    {
        en.WaitOne(); rCount++;
        if (rCount == 1) { rw1.WaitOne(); }
        en.Release(1);
        ..... iteration++;
        en.WaitOne(); rCount--;
        if (rCount == 0) { rw1.Release(1); }
        en.Release(1);
    }
}
}

```

- 1) Первый читатель ждет на семафоре **rw1**, остальные ждут на семафоре **en**;
- 2) Если первый читатель захватил семафор **rw1**, то он освобождает семафор **en**, и тем самым дает возможность всем остальным процессам чтения войти по очереди в секцию, защищенную семафором **en**, и увеличить значение глобального счетчика
- 3) Все читатели могут читать ресурс, они все находятся в критической секции, защищенной семафором **rw1**
- 4) Перед выходом из режима чтения читатель снова захватывает семафор **en**, уменьшает значение счетчика и освобождает семафор **en**, давая возможность остальным процессам чтения корректно завершить работу с ресурсом
- 5) Последний читатель должен освободить семафор **rw1**