

Windows Live™

Spaces Web

Giriş Hotmail Spaces OneCare

Oturum aç

Spaces giriş sayfası ▼

.Net Zone Profil Arkadaşlar Blog Diğer ▼

 Araçlar ▼  

## Blog

### Tarih

Girdiler

Özet

Şubat 2008

Ocak 2008

Aralık 2007

Kasım 2007

Ekim 2007

Eylül 2007

Ağustos 2007

Temmuz 2007

Haziran 2007

Mayıs 2007

Nisan 2007

Mart 2007

Şubat 2007

Ocak 2007

Aralık 2006

Kasım 2006

Ekim 2006

Eylül 2006

Ağustos 2006

Temmuz 2006

Haziran 2006

Mayıs 2006

Nisan 2006

Mart 2006

Şubat 2006

Ocak 2006

Aralık 2005

Kasım 2005

Ekim 2005

Eylül 2005

Ağustos 2005

Temmuz 2005

Haziran 2005

Mayıs 2005

Nisan 2005

Mart 2005

Şubat 2005

Ocak 2005

Aralık 2004

Yazdır

02 Eylül

### Fast Reader/Writer lock in C#

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
namespace Locks
{
    /// <summary>
    /// Condition Variable (CV) class.
    /// </summary>
    public class CV
    {
        private readonly object syncLock = new object(); // Internal lock.
        private readonly object m; // The lock associated with this CV.
        public CV(object m)
        {
            lock ( syncLock )
            {
                this.m = m;
            }
        }
        public void Wait()
        {
            bool enter = false;
            try
            {
                lock ( syncLock )
                {
                    Monitor.Exit(m);
                    enter = true;
                    Monitor.Wait(syncLock);
                }
            }
            finally
            {
                if ( enter )
                    Monitor.Enter(m);
            }
        }
        public void Pulse()
        {
            lock ( syncLock )
            {
                Monitor.Pulse(syncLock);
            }
        }
        public void PulseAll()
        {
            lock ( syncLock )
            {
                Monitor.PulseAll(syncLock);
            }
        }
    }
}

public sealed class RW
{
    private readonly object syncRoot = new object(); // Internal lock.
    private int i = 0; // 0 or greater means readers can pass; -1 is active
    writer.
    private int readWaiters = 0; // Readers waiting for writer to exit.
    private int writeWaiters = 0; // Writers waiting for writer lock.
    private CV wQ; // Condition variable.
    public RW()
    {
        wQ = new CV(syncRoot);
    }
    /// <summary>
    /// Get a lock that is held
```

