|  |  |
| --- | --- |
| Name: | Edward Eisenberger |
| ID# | 1066164 |
| Assignment 3 | |
| Date of Submission | June 7, 2019 |

Table of Contents

[Overview 3](#_Toc10816345)

[Part 1: ReaderWriterLock 3](#_Toc10816346)

[Summary 3](#_Toc10816347)

[Results 4](#_Toc10816348)

[Part 2: TPLTest 5](#_Toc10816349)

[Summary 5](#_Toc10816350)

[Results 5](#_Toc10816351)

# Overview

Assignment 2 consisted of implementing the examples in the Handout on “Multithreading in Csharp” from page 12 to page 23 (excluding process and thread monitoring).

# Part 1: ReaderWriterLock

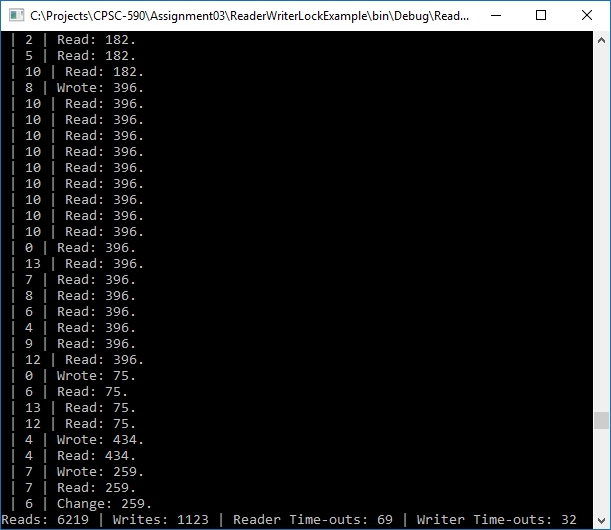
## Summary

ReaderWriterLock is a C# class in the System.Threading namespace as part of the .NET Framework. It is used to protect a shared resource by synchronizing access to a resource. The ReaderWriterLock allows either concurrent read access for multiple threads or write access for a single thread. This works best where most accesses are reads and writes are infrequent and short duration. This provides better throughput than a one-at-a-time lock, such as Monitor.

A coding example utilizing ReaderWriterLock can be found in the ReaderWriterLockExample project within the Assignment03 Visual Studio solution.

## Results

The following image is the output of the application.



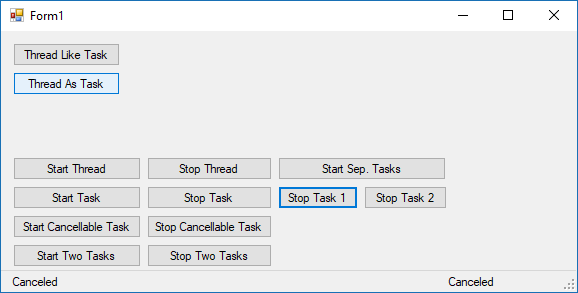
# Part 2: TPLTest

## Summary

Part 2 consisted of implementing the examples in the Task Parallel Library handout.

## Results

The following images are example outputs of the various examples in the Task Parallel Library handout.



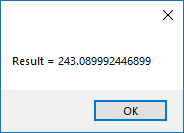


Figure 1 - Thread Like Task

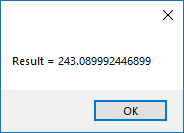


Figure 2 - Thread As Task

