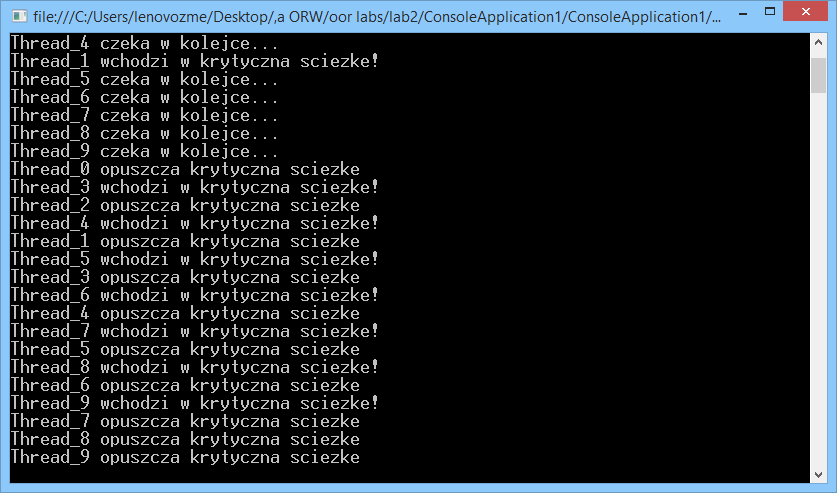
data 25.01.2016

Przedmiot : Programowanie równoległe i rozproszone

Laboratorium numer 2

Równoległość oparta na procesach

1. Celem laboratorium było tworzenie procesów i pokazanie współpracy między nimi
2. Wynik:



Jak widać na powyższym zdjęciu procesy były tworzone, czekały w kolejce i wchodziły kolejno na krytyczną ścieżkę i z niej wychodziły.

using System;

using System.Threading;

namespace CSharp\_Samples

{

public class SemaphoreSample

{

// tworzy objekt semaphore, tylko 3 miejsca dostepne

static Semaphore semaphoreObject = new Semaphore(3, 4);

private static void DoWork()

{

Console.WriteLine("{0} czeka w kolejce...", Thread.CurrentThread.Name);

// watek czeka na dostepne miejsce

semaphoreObject.WaitOne();

Console.WriteLine("{0} wchodzi w krytyczna sciezke!", Thread.CurrentThread.Name);

// symuluje prace przez czekanie 1 sek

Thread.Sleep(1000);

Console.WriteLine("{0} opuszcza krytyczna sciezke", Thread.CurrentThread.Name);

// wykonuje miejsce

semaphoreObject.Release();

}

public static void Main()

{

// tworzy 10 watkow, nazywa

// startuje prace na watkach

for (int i = 0; i < 10; i++)

{

Thread thread = new Thread(DoWork);

thread.Name = "Thread\_" + i;

thread.Start();

}

Console.ReadLine();

}

}

}

Repozytorium: