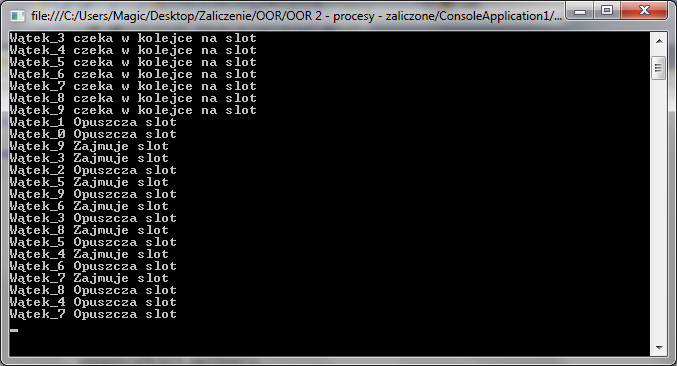
Akademia Marynarki Wojennej

|  |  |  |  |
| --- | --- | --- | --- |
| Imię i nazwisko | | Grupa | Data ćwiczenia |
| Maciej Maliszewski | | 125NCI\_A | 06.12.2015 r. |
| Tytuł ćwiczenia  **2** | **Równoległość oparta na procesach** | | |
| ***Programowanie równoległe i rozproszone***  ***dr inż.A.Zacniewski*** | | | |

Celem laboratorium było napisanie programu który tworzy wiele procesów oraz organizuję przebieg ich wykonywania korzystając z kolejki.

Semaphore – Ogranicza liczbę wątków które mają dostęp do zasobów

Kod programu

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 na slot", Thread.CurrentThread.Name);

// watek czeka na dostepne miejsce

semaphoreObject.WaitOne();

Console.WriteLine("{0} Zajmuje slot", Thread.CurrentThread.Name);

// symuluje prace przez czekanie 1 sek

Thread.Sleep(1000);

Console.WriteLine("{0} Opuszcza slot", 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 = "Wątek\_" + i;

thread.Start();

}

Console.ReadLine();

}

}

}