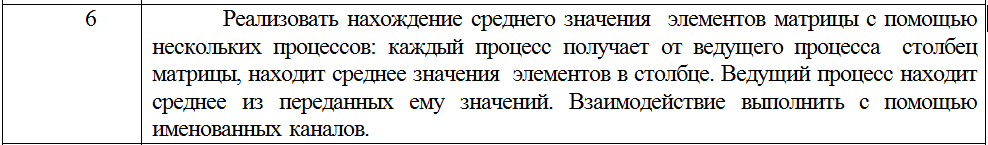
**Лабораторная работа № 7**

**Межпроцессное взаимодействие на примере именованных каналов**

**Цель работы:** Изучение механизмов межпроцессного взаимодействия InterProcessCommunication (IPC) в ОС Windows.



Client

using System;  
using System.Collections.Generic;  
using System.IO;  
using System.IO.Pipes;  
  
namespace Lab7\_Client  
{  
 public class Program  
 {  
  
 public static int[,] matrix =  
 {  
 {1, 2, 3},  
 {2, 3, 4},  
 };  
  
 static void Main(string[] args)  
 {  
 using (NamedPipeClientStream pipeClient = new NamedPipeClientStream(".", "testpipe", PipeDirection.**In**))  
 {  
 using (NamedPipeClientStream pipeClientOut = new NamedPipeClientStream(".", "testpipe2", PipeDirection.**Out**))  
 {  
 *// Подключение к каналу или ожидание пока канал не будет доступен.* Console.Write("Attempting to connect to pipe...");  
 pipeClient.Connect();  
 pipeClientOut.Connect();  
  
 Console.WriteLine("Connected to pipe.");  
 Console.WriteLine("There are currently {0} pipe server instances open.", pipeClient.NumberOfServerInstances);  
  
 List<int> list = new List<int>();  
  
 using (StreamReader sr = new StreamReader(pipeClient))  
 {  
 string temp;  
  
 while ((temp = sr.ReadLine()) != null)  
 {  
 list.Add(int.Parse(temp));  
 }  
 }  
  
 using (StreamWriter sw = new StreamWriter(pipeClientOut))  
 {  
 foreach (var i in list)  
 {  
 sw.WriteLine(FindSum(i));  
 }  
 }  
 }  
 }  
  
 Console.Write("Press Enter to continue...");  
 Console.ReadLine();  
 }  
  
 public static double FindSum(int row)  
 {  
 int sum = 0;  
  
 for (var i = 0; i < 3; i++)  
 {  
 sum += matrix[row, i];  
 }  
  
 return sum / 3.0;  
 }  
 }  
}

Server

using System;  
using System.IO;  
using System.IO.Pipes;  
  
namespace Lab7\_Server  
{  
 public class Program  
 {  
 static void Main(string[] args)  
 {  
 using (NamedPipeServerStream pipeServer = new NamedPipeServerStream("testpipe", PipeDirection.**InOut**))  
 {  
 using (NamedPipeServerStream pipeServerIn = new NamedPipeServerStream("testpipe2", PipeDirection.**In**))  
 {  
 pipeServer.WaitForConnection();  
 pipeServerIn.WaitForConnection();  
  
 using (StreamWriter sw = new StreamWriter(pipeServer))  
 {  
 sw.AutoFlush = true;  
  
 for (var i = 0; i < 2; i++)  
 {  
 sw.WriteLine(i);  
 }  
 }  
  
 using (StreamReader sr = new StreamReader(pipeServerIn))  
 {  
 int sum = 0;  
  
 string temp;  
  
 while ((temp = sr.ReadLine()) != null)  
 {  
 sum += int.Parse(temp);  
 }  
  
 Console.WriteLine(sum / 2.0);  
 }  
 }  
  
 }  
 }  
 }  
}

