using System.Reflection;

class Program

{

// 1. вывести на консоль все запущенные процессы

static void Main(string[] args)

{

TimerCallback timercallback = new TimerCallback(PrintTimer); //создается объект делегата TimerCallback

Timer timer = new Timer(timercallback, 0, 0, 2000);// создаем таймер

var allProc = Process.GetProcesses();//Получить информацию о обо всех процессах системы

foreach (Process process in allProc)

{

Console.WriteLine("-----------------");

Console.WriteLine("Proc ID: " + process.Id);

Console.WriteLine("Name: " + process.ProcessName);

Console.WriteLine("Priority: " + process.BasePriority);

Console.WriteLine("Threads amount: " + process.Threads.Count);

}

//2. исследовать текущий домен приложения

AppDomain domain = AppDomain.CurrentDomain;//домен приложения

Console.WriteLine(new string('\_',50));

Console.WriteLine($"domain - {domain.FriendlyName} \nsetupInfo - {domain.SetupInformation.ConfigurationFile} ");// имя детали конфигурации

Assembly[] assembly = domain.GetAssemblies();// возвр сборки которые были добавлены

foreach (var ass in assembly)

{

Console.WriteLine(ass.GetName().Name);

}

//создание и настройка домена

AppDomain newDomain = AppDomain.CreateDomain("NewDomain");//создаем новый домен

newDomain.Load(assembly[1].GetName().Name);//получаем имя сборки

AppDomain.Unload(newDomain);

Thread thread = new Thread(new ParameterizedThreadStart(PrintSimple));

thread.Priority = ThreadPriority.Lowest;//Свойство Priority хранит приоритет потока - значение перечисления ThreadPriority

thread.Start(20);//метод старт запускает поток

EvenCl ev = new EvenCl(20);

Thread.Sleep(10);

OddCl od = new OddCl(20);

ev.Thread.Join();// блокирует выполнение вызвавшего его потока до тех пор, пока не завершится поток, для которого был вызван данный метод

od.Thread.Join();

NewEven ne = new NewEven(20);

Thread.Sleep(10);//Sleep останавливает поток на определенное количество миллисекунд

NewOdd no = new NewOdd(20);

ne.thr.Join();

od.Thread.Join();

}

public static void PrintTimer(object x)

{

Console.WriteLine("ТАЙМЕР ЗАПУЩЕН");

}

public static void PrintSimple(object N)

{

Thread t = Thread.CurrentThread;//вовзвр выполняющий в данный момент поток

using (StreamWriter file = new StreamWriter("Simple.txt",false,System.Text.Encoding.Default))

{

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

{

if (IsSimple(i)) //выведите информацию о статусе потока приоритете

{

Console.WriteLine($"{i} - простое число {t.IsAlive} - жив? {t.Priority} - приоритет");

file.WriteLine(i + " simple ");

Thread.Sleep(300);}}}}//приост поток

public static bool IsSimple(int N)

{

for (int i = 2; i <= N / 2; i++)

{

if (N % i == 0)

return false;

}

return true;

}

}

class OddCl

{

int N;

public Thread Thread;

public OddCl(int \_N)

{

Thread = new Thread(this.OddFirst);

N = \_N;

Thread.Start();

}

void OddFirst() //позволяет организовать критическую секцию для нескольких процессов

{

Mutexx.mtx.WaitOne();//вход в критическую секцию,

// выводились сначала четные, потом нечетные числа

using (StreamWriter file = new StreamWriter("Numbers.txt", true, System.Text.Encoding.Default))

{

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

{

if (i % 2 != 0)

{

//Thread.Sleep(100);

Console.WriteLine($"{i} - нечет");

file.WriteLine(i + "нечет");

}

}

}

Mutexx.mtx.ReleaseMutex();//(выход из крит секции)

}

}

class EvenCl

{

int N;

public Thread Thread;

public EvenCl(int \_N)

{

Thread = new Thread(this.EvenLast);

N = \_N;

Thread.Start();

}

void EvenLast()

{

Mutexx.mtx.WaitOne();

using (StreamWriter file = new StreamWriter("Numbers.txt", true, System.Text.Encoding.Default))

{

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

{

if (i % 2 == 0)

{

Console.WriteLine($"{i} - чет");

file.WriteLine(i + "чет");}}}

Mutexx.mtx.ReleaseMutex();

}

}

static class Mutexx

{

public static Mutex mtx = new Mutex();//иниц нового экз класса

}

class NewOdd

{

int N;

public Thread thr;

public NewOdd(int \_N)

{

thr = new Thread(this.Run);

N = \_N;

thr.Start();

}

// последовательно выводились одно четное, другое нечетное.

void Run()

{

int a = 1;

while (a < N)

{

Mutexx.mtx.WaitOne();//блокир тек поток

using (StreamWriter file = new StreamWriter("EvenOdd.txt", true, System.Text.Encoding.Default))

{

if (a % 2 != 0)

{

file.WriteLine(a + " - нечет");}}

Mutexx.mtx.ReleaseMutex();//освоб объект

Thread.Sleep(200);

a++;}}}

class NewEven

{

int N;

public Thread thr;

public NewEven(int \_N)

{

thr = new Thread(this.Run);

N = \_N;

thr.Start();

}

// последовательно выводились одно четное, другое нечетное.

void Run()

{

int a = 1;

while (a < N)

{

Mutexx.mtx.WaitOne();

using (StreamWriter file = new StreamWriter("EvenOdd.txt", true, System.Text.Encoding.Default))

{

if (a % 2 == 0)

{

file.WriteLine(a + " - чет");}}

Mutexx.mtx.ReleaseMutex();

Thread.Sleep(200);

a++;}}}