**Лабораторна робота №1. Робота з потоками за допомогою засобів мови програмування C#**

**Мета роботи**: вивчити основні оператори, класи та їх методи доступні у фреймворку Microsoft .Net для роботи з потоками.

**Програма**:

using System;

using System.Threading;

namespace lab1

{

class ThreadTest

{

static bool doneStatic;

bool done;

static void Main(string[] args)

{

int p = 1;

Console.Write("Enter #: ");

bool result = Int32.TryParse(Console.ReadLine(), out p);

if (!result)

return;

Console.WriteLine("Running program #{0}", p);

if (p == 1) {

Thread thr = new Thread(WriteY);

thr.Start();

while(true)

Console.Write("X");

}

else if (p == 2) {

new Thread(Go).Start();

Go();

}

else if (p == 3) {

ThreadTest tt = new ThreadTest();

new Thread(tt.GoInstance).Start();

tt.GoInstance();

}

else if (p == 4) {

new Thread(GoWithDone).Start();

GoWithDone();

}

else if (p == 5) {

new Thread(GoWithDoneAfter).Start();

GoWithDoneAfter();

}

else if (p == 6) {

new Thread(ThreadSafe.Go).Start();

ThreadSafe.Go();

}

}

void GoInstance() {

if (!done) { done = true; Console.WriteLine("done"); }

}

static void Go() {

for (int cycles = 0; cycles < 5; cycles++) {

Console.Write('P');

}

}

static void GoWithDone() {

if (!doneStatic) {

doneStatic = true;

Console.WriteLine("Done!");

}

}

static void GoWithDoneAfter() {

if (!doneStatic) {

Console.WriteLine("Done!");

doneStatic = true;

}

}

static void WriteY()

{

while(true)

Console.Write("Y");

}

}

class ThreadSafe {

static bool done;

static object locker = new object();

public static void Go() {

lock (locker) {

if (!done) {

Console.WriteLine("Done");

done = true;

}

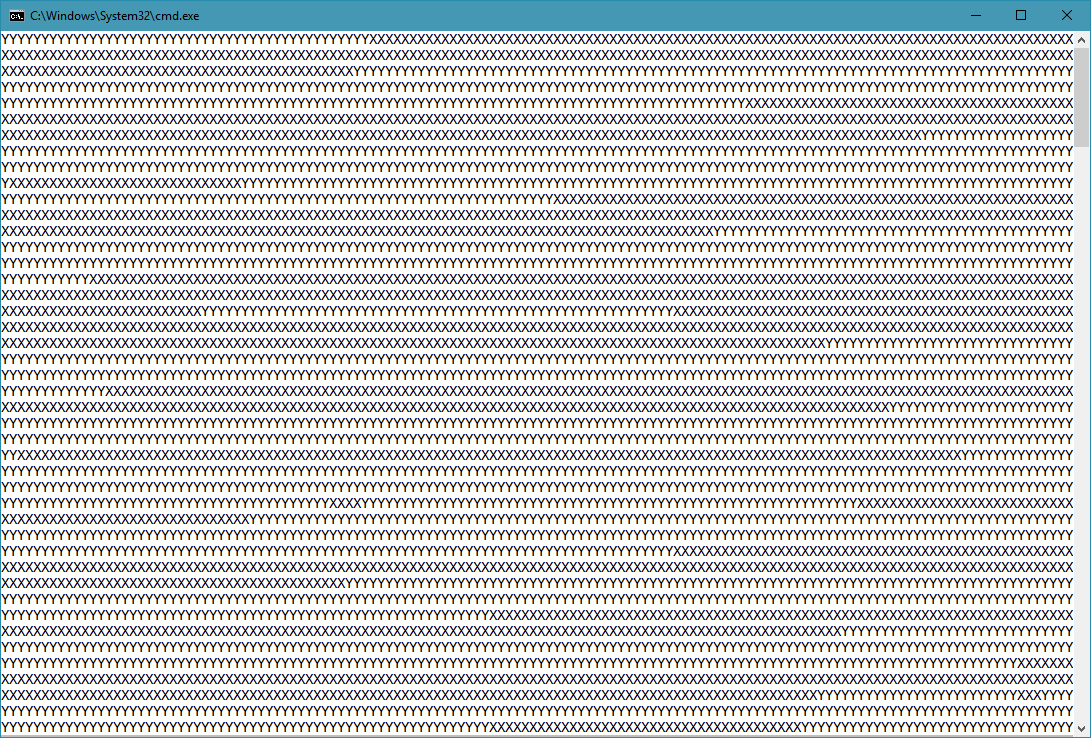
}

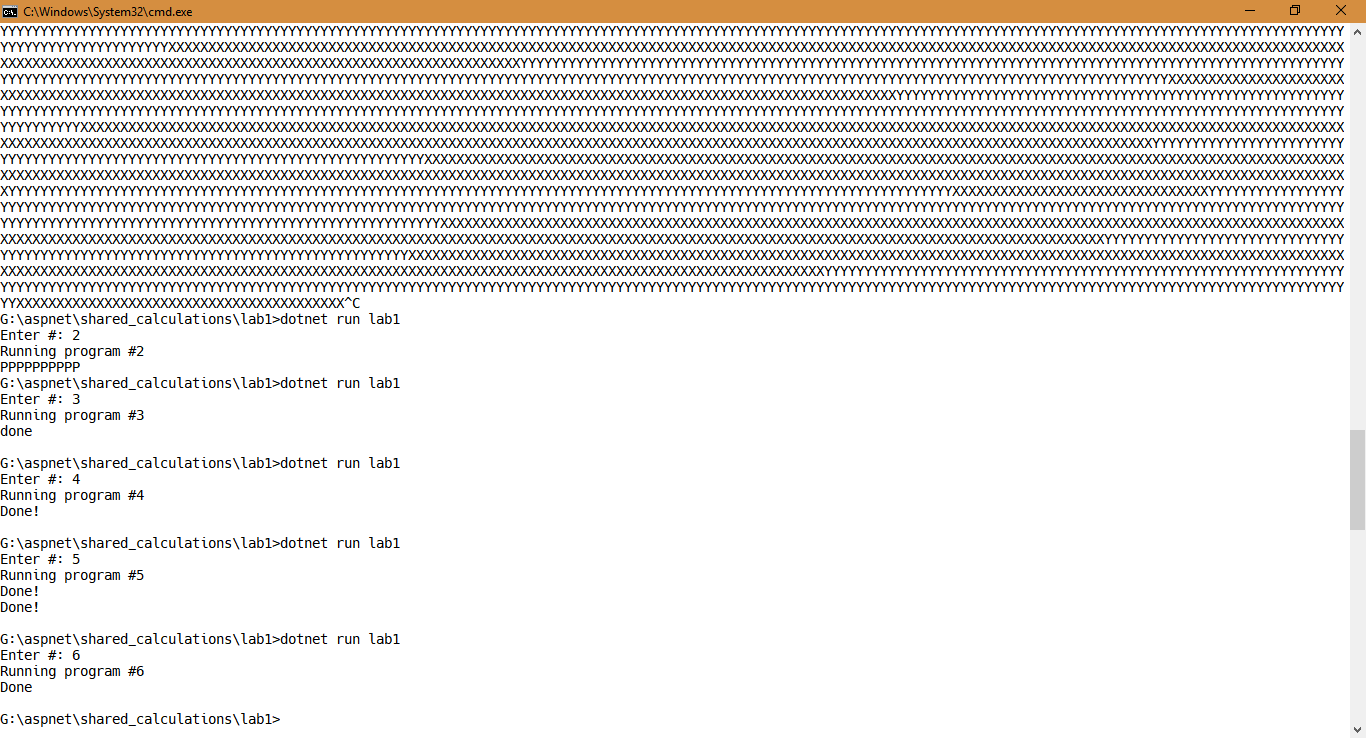
}

}

}

**Вивід**:





**Висновки**:

Як бачимо CLR почергово надає керування потокам, таким чином виводи чергуються, якщо придивитись то можна помітити що послідовності з однакових символів доволі великі, це показує що операція передачі контролю іншому потоку коштує ресурсів, тому система обмежує цю саму частоту передачі контролю.