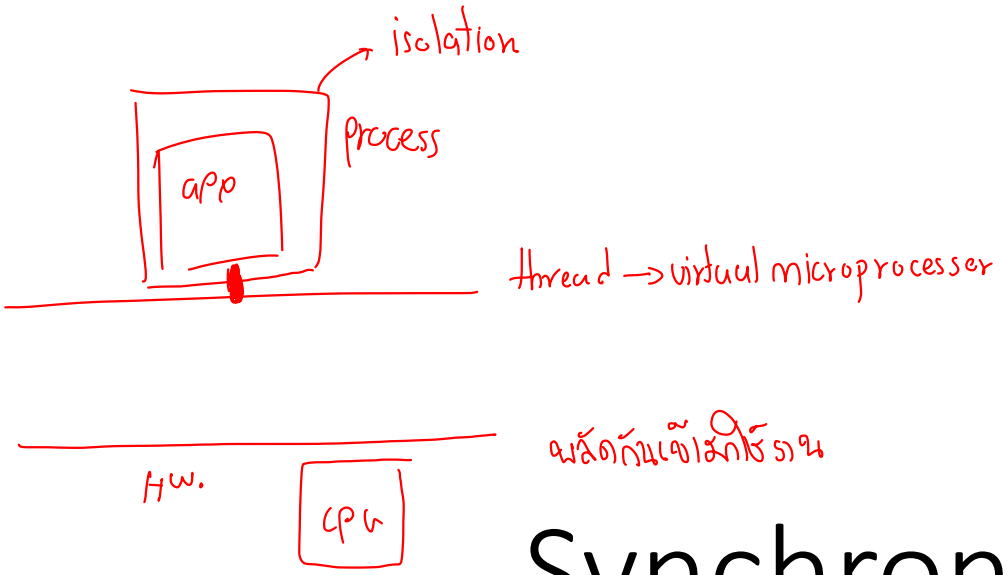


Synchronization

Experiments

1. No synchronization

```
using System.Threading;

namespace TestThreadNolock
{
    class Program
    {
        private static int x = 0;
        static object _lock = new object();

        static void FuncA()
        {
            int xx = 0;
            while (xx < 50)
            {
                Console.WriteLine("FuncA: round:{0} x={1}", xx, x);
                x++;
                xx++;
            }
        }

        static void FuncB()
        {
            int xx = 0;
            while (xx < 50)
            {
                Console.WriteLine("==FuncB: round:{0} x={1}", xx, x);
                x++;
                xx++;
            }
        }

        static void Main(string[] args)
        {
            Thread A = new Thread(new ThreadStart(FuncA));
            Thread B = new Thread(new ThreadStart(FuncB));
            A.Start();
            B.Start();
        }
    }
}
```

เขียนโปรแกรมเกี่ยวกับ thread → shared resoure → ใช้ทรัพยากรร่วม → เกิดปัญหา

→ race condition

→ how to protect → ไม่ให้เข้ามาใช้ได้ทีละคน

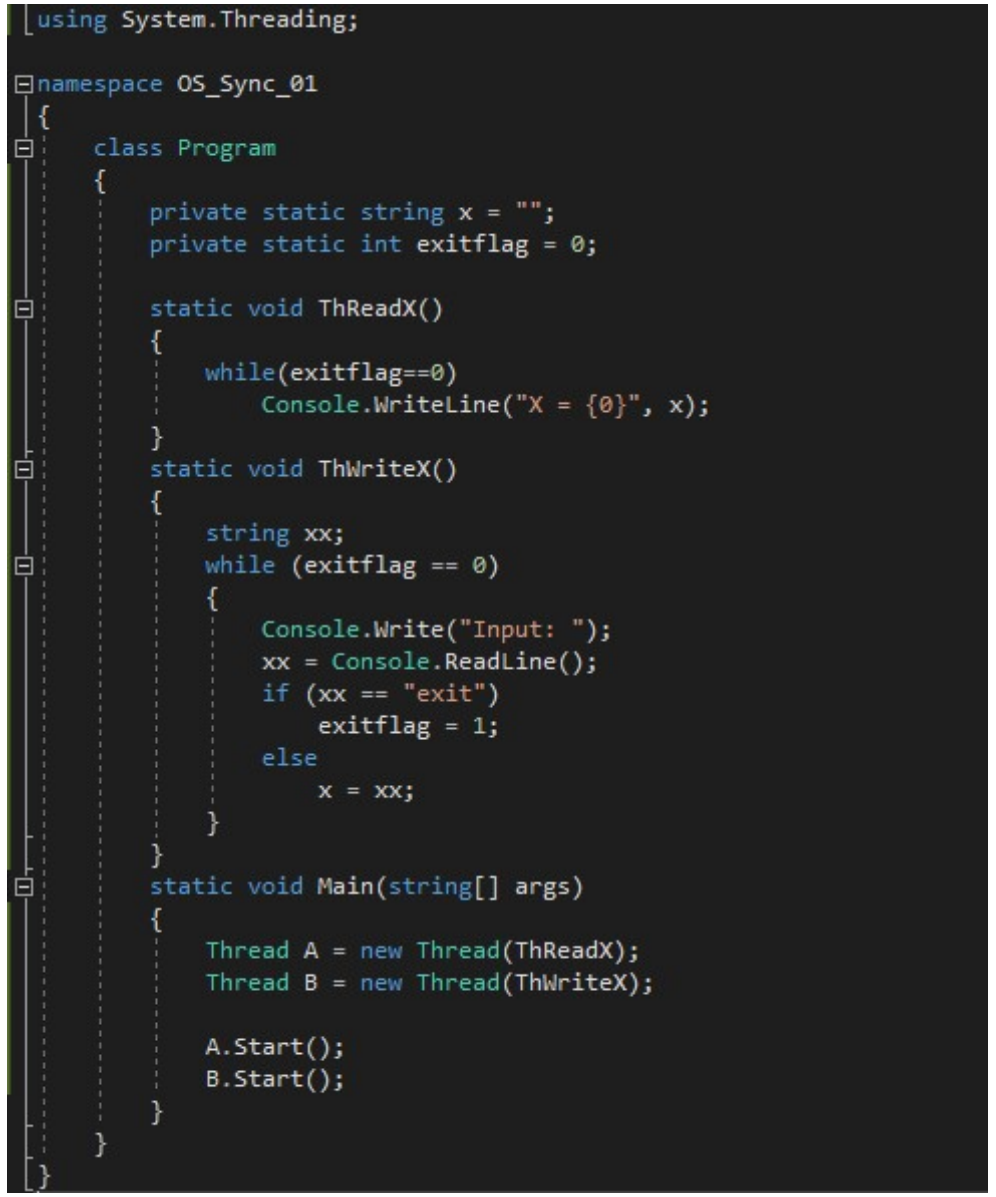
→ ดูแลแต่ละคนไม่ให้ผ่านประตู → thread เดียวที่เข้าไปใช้ only → เราเรียกว่า lock

2. Lock

```csharp
using System.Threading;

namespace TestThreadNolock
{
    class Program
    {
        private static int x = 0;
        static object _lock = new object();

        static void FuncA()
        {
            int xx = 0;
            while (xx < 50)
            {
                lock (_lock)
                {
                    Console.WriteLine("FuncA: round:{0} x={1}", xx, x);
                    x++;
                }
                xx++;
            }
        }

        static void FuncB()
        {
            int xx = 0;
            while (xx < 50)
            {
                lock (_lock)
                {
                    Console.WriteLine("==FuncB: round:{0} x={1}", xx, x);
                    x++;
                }
                xx++;
            }
        }

        static void Main(string[] args)
        {
            Thread A = new Thread(new ThreadStart(FuncA));
            Thread B = new Thread(new ThreadStart(FuncB));
            A.Start();
            B.Start();
        }
    }
}
```

- lock (_lock) → ขั้นตอนการ request key → if success โปรแกรมทำงานต่อได้ → not ← ทำงานไม่ได้
- x++; → จะไปได้ต้องมี key
- } → ทำการคืนกุญแจ
- overhead เล่นของ → ขอกุญแจ 50 รอบ → ทำไม่ได้ครบงานมาก → ขอครั้งเดียว
- ถ้าเอา lock ออกนอก while เอาไว้ → while ทำงานไม่ได้ → context switch → thread อื่นเข้าไม่ lock ไม่ได้ → เพราะพอเอา lock → set ready ให้อีก thread
- ต้องรอ thread อื่น → thread อื่นรอ หน้าที่ต้องทำก่อน ทำงานเสร็จ
- ทำงานเสร็จ → เสีย concept concern

3. No Synchronization



```
using System.Threading;

namespace OS_Sync_01
{
    class Program
    {
        private static string x = "";
        private static int exitflag = 0;

        static void ThReadX()
        {
            while(exitflag==0)
                Console.WriteLine("X = {0}", x);
        }
        static void ThWriteX()
        {
            string xx;
            while (exitflag == 0)
            {
                Console.Write("Input: ");
                xx = Console.ReadLine();
                if (xx == "exit")
                    exitflag = 1;
                else
                    x = xx;
            }
        }
        static void Main(string[] args)
        {
            Thread A = new Thread(ThReadX);
            Thread B = new Thread(ThWriteX);

            A.Start();
            B.Start();
        }
    }
}
```

4. Try #1

เราไม่รู้ thread ไหนทำงานเมื่อไหร่

```csharp
using System.Threading;

namespace OS_Sync_03
{
    class Program
    {
        private static string x = "";
        private static int exitflag = 0;
        private static int updateFlag = 0;
        private static object _Lock;

        static void ThReadX(Object i)
        {
            while (exitflag == 0)
            {
                while (updateFlag == 0) ;
                if (x!="exit")
                    Console.WriteLine("Thread {0} : X = {1}", i, x);
                updateFlag = 0;
            }
        }
        static void ThWriteX()
        {
            string xx;
            while (exitflag == 0)
            {
                Console.Write("Input: ");
                xx = Console.ReadLine();
                if (xx == "exit")
                    exitflag = 1;
                x = xx;
                updateFlag = 1;
            }
        }
        static void Main(string[] args)
        {
            Thread A = new Thread(ThReadX);
            Thread B = new Thread(ThWriteX);

            A.Start(1);
            B.Start();
        }
    }
}
```

ใช้ flag ช่วยแยก

5. Try #2

```csharp
using System.Threading;

namespace OS_Sync_03
{
    class Program
    {
        private static string x = "";
        private static int exitflag = 0;
        private static int updateFlag = 0;
        private static object _Lock;

        static void ThReadX(Object i)
        {
            while (exitflag == 0)
            {
                while (updateFlag == 0) ;
                if (x!="exit")
                    Console.WriteLine("Thread {0} : X = {1}", i, x);
                updateFlag = 0;
            }
        }
        static void ThWriteX()
        {
            string xx;
            while (exitflag == 0)
            {
                Console.Write("Input: ");
                xx = Console.ReadLine();
                if (xx == "exit")
                    exitflag = 1;
                x = xx;
                updateFlag = 1;
            }
        }
        static void Main(string[] args)
        {
            Thread A = new Thread(ThReadX);
            Thread B = new Thread(ThWriteX);
            Thread C = new Thread(ThReadX);
            Thread D = new Thread(ThReadX);

            A.Start(1);
            B.Start();
            C.Start(2);
            D.Start(3);
        }
    }
}
```

ถ้าเราต้องการให้ thread ใด thread หนึ่งอ่านค่าไปแล้ว

↓

ให้ thread ที่อ่านแสดงผล thread เดียว ต้องทำยังไร ?

6. Condition Variable

```
using System.Threading;

namespace OS_Sync_04
{
    class Program
    {
        private static string x = "";
        private static int exitflag = 0;
        private static int updateFlag = 0;
        private static object _Lock = new object();

        static void ThReadX(Object i)
        {

            while (exitflag == 0)
            {
                lock (_Lock)
                {
                    while (updateFlag == 0)
                        Monitor.Wait(_Lock);
                    if (x != "exit")
                        Console.WriteLine("Thread {0} : X = {1}", i, x);
                    updateFlag = 0;
                }
            }
            Console.WriteLine("Thread {0} exit", i);
        }
        static void ThWriteX()
        {
            string xx;
            while (exitflag == 0)
            {
                lock (_Lock)
                {
                    Console.Write("Input: ");
                    xx = Console.ReadLine();
                    if (xx == "exit")
                        exitflag = 1;
                    x = xx;
                    updateFlag = 1;
                    Monitor.Pulse(_Lock);
                    Thread.Sleep(100);
                }
            }
        }
        static void Main(string[] args)
        {
            Thread A = new Thread(ThReadX);
            Thread B = new Thread(ThWriteX);
            Thread C = new Thread(ThReadX);
            Thread D = new Thread(ThReadX);

            A.Start(1);
            B.Start();
            C.Start(2);
            D.Start(3);
        }
    }
}
```