



รายงาน

Case study #1

สมาชิก

นายกิตติศักดิ์ ขาตัน 65015015
นายณรงค์ฤทธิ์ กลิ่นลอย 65015041
นายพนธกร พวงพยอม 65015101
นางสาวมูทิตา บัวขำ 65015123
นายวิฒชัย เตชะสือ 65015137
นายอภิเดช เทพสนิท 65015168

รายวิชา

Operating Systems รหัสวิชา 01076011
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Case Study #1

โดยในการแก้ไขใดๆ จะต้องมีการอธิบายว่าแก้ไขเพราะอะไร นอกจากนี้หากมีการแก้ไขหลาย version ต้องมีการอธิบายของแต่ละ version และ version ที่ใหม่กว่าแก้ไขอะไรของ version เดิม เป็นต้น

ให้สังเกตปัญหาที่เกิดขึ้นในระหว่างทำงาน และให้ตั้งข้อสังเกตตามความคิดของ นศ จากนั้นค้นคว้าหาสาเหตุของปัญหานั้นและวิธีการแก้ไข

ให้ นศ. ดัดแปลงแก้ไขโปรแกรมที่แนบมานี้ให้ทำงานได้เร็วยิ่งขึ้น (ให้ unrar ไฟล์ Problem01.rar ก่อนนำไปใช้งาน)

โดยในการแก้ไขใดๆ จะต้องมีการอธิบายว่าแก้ไขเพราะอะไร นอกจากนี้หากมีการแก้ไขหลาย version ต้องมีการอธิบายของแต่ละ version และ version ที่ใหม่กว่าแก้ไขอะไรของ version เดิม เป็นต้น

ให้สังเกตปัญหาที่เกิดขึ้นในระหว่างทำงาน และให้ตั้งข้อสังเกตตามความคิดของ นศ จากนั้นค้นคว้าหาสาเหตุของปัญหานั้นและวิธีการแก้ไข

ประมวลข้อมูลทั้งหมดเพื่อเขียนสรุปส่งและนำเสนอหน้าห้องต่อไป

สิ่งที่ต้องส่ง

1. source code ของ version ที่ นศ คิดว่าเป็น version ที่ดีที่สุด
2. สรุปสิ่งที่สังเกตพบ ปัญหา วิธีแก้ ลงในไฟล์ pdf และตั้งชื่อไฟล์โดยใช้รหัสของสมาชิกคันด้วยเครื่องหมาย _ เช่น 64010001_64010002_64010003.pdf **ให้เรียงลำดับรหัสจากน้อยไปหามากด้วย**
3. ไฟล์ presentation (ถ้ามี - ให้ตั้งชื่อเหมือนที่ระบุไว้ในข้อ 2.)
4. ไฟล์ Source code C# (.cs) ซึ่งเป็น version ที่ นศ คิดว่าดีที่สุด (ให้ตั้งชื่อเหมือนที่ระบุไว้ในข้อ 2.)
5. ตัวแทนกลุ่มเป็นผู้ส่งเพียงผู้เดียวสำหรับแต่ละกลุ่ม
6. ใส่ชื่อสมาชิกกลุ่มในแต่ละไฟล์ที่ส่งให้เรียบร้อย

เนื่องจากไฟล์ที่เป็นข้อมูลมีขนาดใหญ่การดาวน์โหลดจากระบบ GoEdu ค่อนข้างมีปัญหา ดังนั้นจึงให้ดาวน์โหลดจาก >>Google Drive<<

#Code Version Original

```
using System;
using System.Diagnostics;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Threading;

namespace Problem01
{
    class Program
    {
        static byte[] Data_Global = new byte[1000000000];
        static long Sum_Global = 0;
        static int G_index = 0;

        static int ReadData()
        {
            int returnData = 0;
            FileStream fs = new FileStream("Problem01.dat", FileMode.Open);
            BinaryFormatter bf = new BinaryFormatter();

            try
            {
                Data_Global = (byte[]) bf.Deserialize(fs);
            }
            catch (SerializationException se)
            {
                Console.WriteLine("Read Failed:" + se.Message);
                returnData = 1;
            }
            finally
            {
                fs.Close();
            }

            return returnData;
        }

        static void sum()
```

```

{
    if (Data_Global[G_index] % 2 == 0)
    {
        Sum_Global -= Data_Global[G_index];
    }
    else if (Data_Global[G_index] % 3 == 0)
    {
        Sum_Global += (Data_Global[G_index]*2);
    }
    else if (Data_Global[G_index] % 5 == 0)
    {
        Sum_Global += (Data_Global[G_index] / 2);
    }
    else if (Data_Global[G_index] % 7 == 0)
    {
        Sum_Global += (Data_Global[G_index] / 3);
    }
    Data_Global[G_index] = 0;
    G_index++;
}

static void Main(string[] args)
{
    Stopwatch sw = new Stopwatch();
    int i, y;

    /* Read data from file */
    Console.Write("Data read...");
    y = ReadData();
    if (y == 0)
    {
        Console.WriteLine("Complete.");
    }
    else
    {
        Console.WriteLine("Read Failed!");
    }

    /* Start */

```

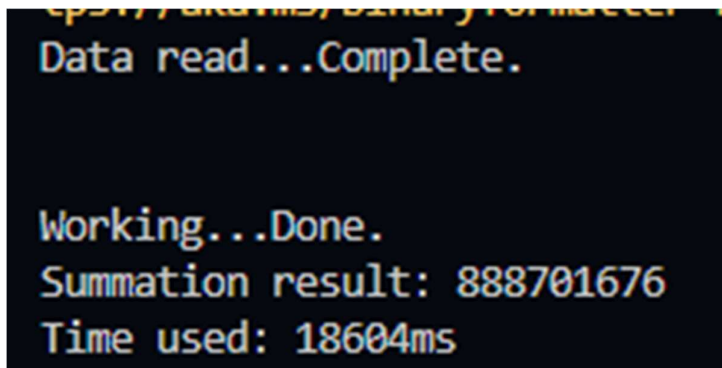
```

        Console.Write("\n\nWorking...");
        sw.Start();
        for (i = 0; i < 1000000000; i++)
            sum();
        sw.Stop();
        Console.WriteLine("Done.");

        /* Result */
        Console.WriteLine("Summation result: {0}", Sum_Global);
        Console.WriteLine("Time used: " + sw.ElapsedMilliseconds.ToString() + "ms");
    }
}
}

```

Output :



```

Data read...Complete.

Working...Done.
Summation result: 888701676
Time used: 18604ms

```

ปัญหา : โค้ดรันได้ช้าเนื่องจากมีการแชร์ทรัพยากรทำให้คอมพิวเตอร์แบ่งทรัพยากรมาใช้ในการรันโค้ดน้อย
และมีการทำงานแบบ $O(n)$

วิธีแก้ : เขียนโค้ดใหม่ให้สามารถรันได้ไวกว่าเดิมโดยการสร้าง Thread เพิ่ม

```

using System;
using System.Diagnostics;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Threading;

namespace Problem01
{

```

#Code Version 1

```
class Program
{
    static byte[] Data_Global = new byte[1000000000];
    static long Sum_Global = 0;

    static int ReadData()
    {
        int returnData = 0;
        FileStream fs = new FileStream("Problem01.dat", FileMode.Open);
        BinaryFormatter bf = new BinaryFormatter();

        try
        {
            Data_Global = (byte[])bf.Deserialize(fs);
        }
        catch (SerializationException se)
        {
            Console.WriteLine("Read Failed:" + se.Message);
            returnData = 1;
        }
        finally
        {
            fs.Close();
        }

        return returnData;
    }

    static void sum()
    {
        int G_index = 0;
        for (int i = 0; i < 1000000000; i++)
        {
            if (Data_Global[G_index] % 2 == 0)
            {
                Sum_Global -= Data_Global[G_index];
            }
            else if (Data_Global[G_index] % 3 == 0)
            {
                Sum_Global += (Data_Global[G_index] * 2);
            }
        }
    }
}
```

```

    }
    else if (Data_Global[G_index] % 5 == 0)
    {
        Sum_Global += (Data_Global[G_index] / 2);
    }
    else if (Data_Global[G_index] % 7 == 0)
    {
        Sum_Global += (Data_Global[G_index] / 3);
    }
    Data_Global[G_index] = 0;
    G_index += 1;
}
}

```

```

static void Main(string[] args)
{
    Stopwatch sw = new Stopwatch();
    Thread th1 = new Thread(sum);
    /* Read data from file */
    Console.Write("Data read...");
    int y = ReadData();
    if (y == 0)
    {
        Console.WriteLine("Complete.");
    }
    else
    {
        Console.WriteLine("Read Failed!");
    }

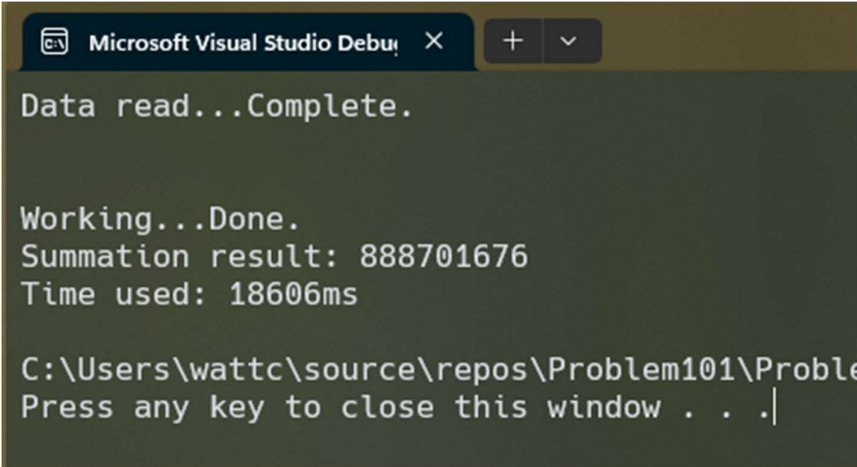
    /* Start */
    Console.Write("\n\nWorking...");
    sw.Start();
    th1.Start();
    th1.Join();
    sw.Stop();

    Console.WriteLine("Done.");
}

```

```
/* Result */
Console.WriteLine("Summation result: {0}", Sum_Global);
Console.WriteLine("Time used: " + sw.ElapsedMilliseconds.ToString() + "ms");
}
}
}
```

Output :



The screenshot shows a Visual Studio Debug Console window with a dark background. The output text is as follows:

```
Data read...Complete.

Working...Done.
Summation result: 888701676
Time used: 18606ms

C:\Users\wattc\source\repos\Problem101\Proble
Press any key to close this window . . .|
```

การแก้ปัญหา : จึงลองปรับใช้ 1 Thread ในการทำงาน แต่ประสิทธิภาพเวลาในการทำงานยังคงเท่าเดิมกับโค้ดดั้งเดิม

#Code Version 2

```
using System;
using System.Diagnostics;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Threading;

namespace Problem01
{
    class Program
    {
        static byte[] Data_Global = new byte[1000000000];
        static long Sum_Global = 0;

        static int ReadData()
        {
            int returnData = 0;
            FileStream fs = new FileStream("Problem01.dat", FileMode.Open);
            BinaryFormatter bf = new BinaryFormatter();

            try
            {
                Data_Global = (byte[])bf.Deserialize(fs);
            }
            catch (SerializationException se)
            {
                Console.WriteLine("Read Failed:" + se.Message);
                returnData = 1;
            }
            finally
            {
                fs.Close();
            }

            return returnData;
        }

        static void sum()
```

```

{
    int G_index = 0;
    for (int i = 0; i < 250000000; i++)
    {
        if (Data_Global[G_index] % 2 == 0)
        {
            Sum_Global -= Data_Global[G_index];
        }
        else if (Data_Global[G_index] % 3 == 0)
        {
            Sum_Global += (Data_Global[G_index] * 2);
        }
        else if (Data_Global[G_index] % 5 == 0)
        {
            Sum_Global += (Data_Global[G_index] / 2);
        }
        else if (Data_Global[G_index] % 7 == 0)
        {
            Sum_Global += (Data_Global[G_index] / 3);
        }
        Data_Global[G_index] = 0;
        G_index += 4;
    }
}

```

```

static void Main(string[] args)
{
    Stopwatch sw = new Stopwatch();
    Thread th1 = new Thread(sum);
    Thread th2 = new Thread(sum);
    Thread th3 = new Thread(sum);
    Thread th4 = new Thread(sum);
    /* Read data from file */
    Console.WriteLine("Data read...");
    int y = ReadData();
    if (y == 0)
    {
        Console.WriteLine("Complete.");
    }
}

```

```

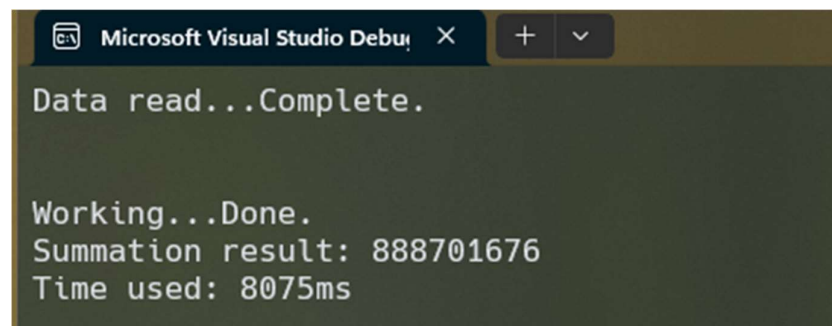
    }
    else
    {
        Console.WriteLine("Read Failed!");
    }

    /* Start */
    Console.WriteLine("\n\nWorking...");
    sw.Start();
    th1.Start();
    th2.Start();
    th3.Start();
    th4.Start();
    th1.Join();
    th2.Join();
    th3.Join();
    th4.Join();
    sw.Stop();
    Console.WriteLine("Done.");

    /* Result */
    Console.WriteLine("Summation result: {0}", Sum_Global);
    Console.WriteLine("Time used: " + sw.ElapsedMilliseconds.ToString() + "ms");
}
}
}

```

Output :



การแก้ปัญหา : ปรับใช้ 4 Thread ในการทำงาน โดยการทำงานทำให้แบ่งงานกันหลาย ๆ งานจึงทำให้การรันไวขึ้นแต่ผลลัพธ์ยังไม่น่าพึงพอใจ

#Code Version final

```
using System;
using System.Diagnostics;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Threading;

namespace Problem01
{
    class Program
    {
        static byte[] Data_Global = new byte[1000000000];
        static long Sum_Global = 0;

        static int ReadData()
        {
            int returnData = 0;
            FileStream fs = new FileStream("Problem01.dat", FileMode.Open);
            BinaryFormatter bf = new BinaryFormatter();

            try
            {
                Data_Global = (byte[])bf.Deserialize(fs);
            }
            catch (SerializationException se)
            {
                Console.WriteLine("Read Failed:" + se.Message);
                returnData = 1;
            }
            finally
            {
                fs.Close();
            }

            return returnData;
        }

        static void sum()
        {

```

```

int G_index = 0;
Process currentProcess = Process.GetCurrentProcess();
int threadCount = currentProcess.Threads.Count;
for (int i = 0; i < (1000000000/threadCount); i++)
{
    if (Data_Global[G_index] % 2 == 0)
    {
        Sum_Global -= Data_Global[G_index];
    }
    else if (Data_Global[G_index] % 3 == 0)
    {
        Sum_Global += (Data_Global[G_index] * 2);
    }
    else if (Data_Global[G_index] % 5 == 0)
    {
        Sum_Global += (Data_Global[G_index] / 2);
    }
    else if (Data_Global[G_index] % 7 == 0)
    {
        Sum_Global += (Data_Global[G_index] / 3);
    }
    Data_Global[G_index] = 0;
    G_index += 1;
}
}

static void Main(string[] args)
{
    Stopwatch sw = new Stopwatch();
    int totalThread = Environment.ProcessorCount;
    Thread[] th = new Thread[totalThread];
    for(int i = 0; i < totalThread; i++){
        th[i] = new Thread(sum);
    }
    /* Read data from file */
    Console.WriteLine("Data read...");
    int y = ReadData();
    if (y == 0)
    {

```

```

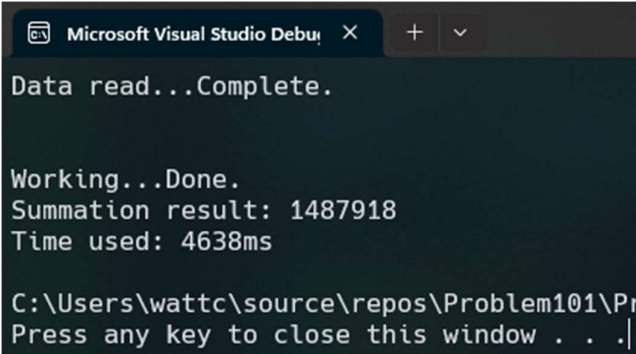
        Console.WriteLine("Complete.");
    }
    else
    {
        Console.WriteLine("Read Failed!");
    }

    /* Start */
    Console.WriteLine("\n\nWorking...");
    sw.Start();
    foreach(Thread thread in th){
        thread.Start();
    }
    foreach(Thread thread in th){
        thread.Join();
    }
    sw.Stop();
    Console.WriteLine("Done.");

    /* Result */
    Console.WriteLine("Summation result: {0}", Sum_Global);
    Console.WriteLine("Time used: " + sw.ElapsedMilliseconds.ToString() + "ms");
}
}
}

```

Result :



```

Microsoft Visual Studio Debug Console
Data read...Complete.

Working...Done.
Summation result: 1487918
Time used: 4638ms

C:\Users\wattc\source\repos\Problem101\Pr
Press any key to close this window . . .|

```

การแก้ปัญหา : การแบ่งข้อมูลใน Array Data_Global ออกเป็นส่วนๆให้แต่ละ Thread ทำงานกับส่วนข้อมูลที่แตกต่างกัน เพื่อหาจำนวน Thread ที่ CPU ใช้งานอยู่ และเพิ่มตัวแปรมารองรับจำนวน

Thread ทั้งหมดที่คอมพิวเตอร์ของเรามีและสร้างตัวแปร Array ที่จองขนาดเท่ากับ Thread ที่เรามีและสร้าง Thread โดยใช้ Loop ใน Array และทำการวน Loop ใน Thread join กัน