PLP Assignment Report

The code & comments

File **FirstScenario.cs**:

```
using System;
using System.Threading;
using System.Diagnostics;

namespace PLP_Assignment
{
    public class FirstScenario
    {
        private static SemaphoreSlim single = new SemaphoreSlim(1); // Since we want one
            student each time

        //  This method will be called as a thread
        private static void ProcessStudent(int student_number)
        {
            single.Wait();  //  A student has came, so we decrease semaphore value
            //  Let's ask the student 5 questions
            for (int i = 1; i <= 5; i++)
            {
                Console.WriteLine("Asking student {0} question {1}", student_number, i);
                Thread.Sleep(1000);
            }
            single.Release();   //  The student has left, so we increase the semaphore value
        }

        public static void Execute()
        {
            //  Fireup 16 new threads
            for (int i = 1; i <= 16; i++)
            {
                new Thread(() => {
                    ProcessStudent(i);
                }).Start();
            }
        }
    }
}
```

File **SecondScenario.cs**:

```
1  using System;
2  using System.Threading;
3
4  namespace PLP_Assignment
5  {
6      public class SecondScenario
7      {
8          private static Barrier barrier          = new Barrier(2);
9          private static SemaphoreSlim semaphore  = new SemaphoreSlim(2); // Since we want two
               students each time
10         private static void ProcessStudent(object student_number)
11         {
12             semaphore.Wait();
13             for (int i = 1; i <= 5; i++)
14             {
15                 Thread.Sleep(1000);
16                 Console.WriteLine("Asking student {0} question {1}", student_number, i);

17             }
18             semaphore.Release();
19         }
20
21         public static void Execute()
22         {
23             for (int i = 1; i <= 16; i++)
24             {
25                 new Thread(() => {
26                     ProcessStudent(i);
27                 }).Start();
28                 if (barrier.ParticipantCount < 2) // If number of current participants in
                       barrier is less than 2, then add the current thread
29                 {
30                     barrier.SignalAndWait();
31                 }
32             }
33         }
```

File **ThirdScenario.cs**:

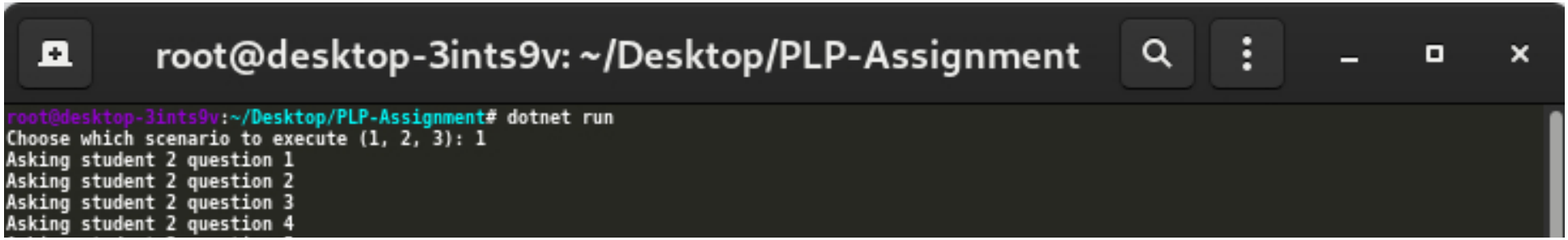
```
1  // Same as second scenaio, but with 4 students
2  using System;
3  using System.Threading;
4
5  namespace PLP_Assignment
6  {
7      public class ThirdScenario
8      {
9          private static Barrier barrier          = new Barrier(4);
10         private static SemaphoreSlim semaphore  = new SemaphoreSlim(4);
11         private static void ProcessStudent(object student_number)
12         {
13             semaphore.Wait();
14             for (int i = 1; i <= 5; i++)
15             {
16                 Thread.Sleep(1000);
17                 Console.WriteLine("Asking student {0} question {1}", student_number, i);

18             }
19             semaphore.Release();
20         }
21
22         public static void Execute()
23         {
24             for (int i = 1; i <= 16; i++)
25             {
26                 new Thread(() => {
27                     ProcessStudent(i);
28                 }).Start();
29                 if (barrier.ParticipantCount < 4)
30                 {
31                     barrier.SignalAndWait();
32                 }
33             }
34         }
35     }
```

File **Program.cs**:

```
// Entry point
using System;
using System.Threading.Tasks;
using System.Diagnostics;

namespace PLP_Assignment
{
    public class Program
    {
        public static int Main(string[] args)
        {
            Console.Write("Choose which scenario to execute (1, 2, 3): ");
            int choice = Convert.ToInt32(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    FirstScenario.Execute();
                    break;

                case 2:
                    SecondScenario.Execute();
                    break;

                case 3:
                    ThirdScenario.Execute();
                    break;
            }
            return 0;
        }
    }
}
```

Console Screen Output

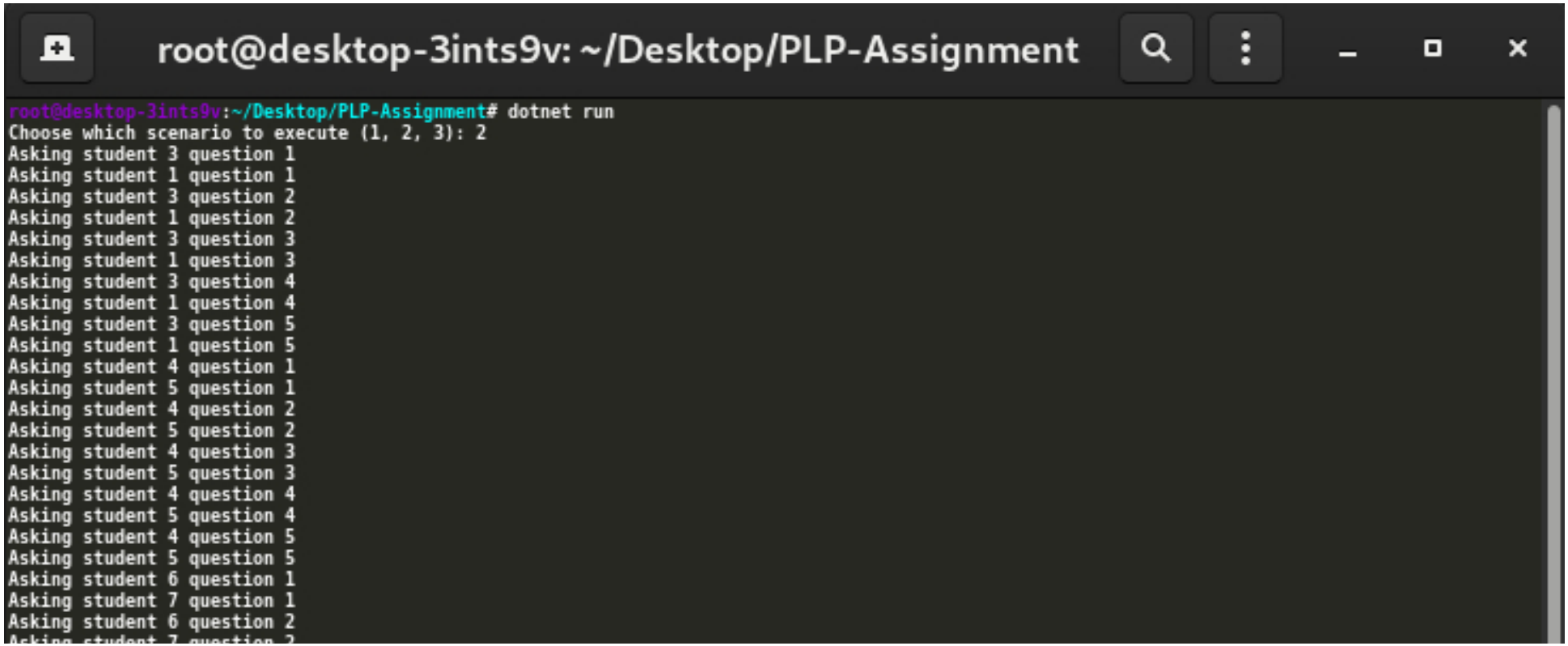
First scenario:



```
Asking student 2 question 5
Asking student 3 question 1
Asking student 3 question 2
Asking student 3 question 3
Asking student 3 question 4
Asking student 3 question 5
Asking student 5 question 1
Asking student 5 question 2
Asking student 5 question 3
Asking student 5 question 4
Asking student 5 question 5
Asking student 9 question 1
Asking student 9 question 2
Asking student 9 question 3
Asking student 9 question 4
Asking student 9 question 5
Asking student 7 question 1
Asking student 7 question 2
Asking student 7 question 3
Asking student 7 question 4
Asking student 7 question 5
Asking student 4 question 1
Asking student 4 question 2
Asking student 4 question 3
Asking student 4 question 4
Asking student 4 question 5
Asking student 10 question 1
Asking student 10 question 2
Asking student 10 question 3
Asking student 10 question 4
Asking student 10 question 5
Asking student 8 question 1
Asking student 8 question 2
Asking student 8 question 3
Asking student 8 question 4
Asking student 8 question 5
Asking student 6 question 1
Asking student 6 question 2
Asking student 6 question 3
Asking student 6 question 4
Asking student 6 question 5
Asking student 11 question 1
Asking student 11 question 2
Asking student 11 question 3
Asking student 11 question 4
Asking student 11 question 5
Asking student 12 question 1
Asking student 12 question 2
Asking student 12 question 3
Asking student 12 question 4
Asking student 12 question 5
Asking student 13 question 1
Asking student 13 question 2
Asking student 13 question 3
Asking student 13 question 4
Asking student 13 question 5
```

```
Asking student 14 question 1
Asking student 14 question 2
Asking student 14 question 3
Asking student 14 question 4
Asking student 14 question 5
Asking student 15 question 1
Asking student 15 question 2
Asking student 15 question 3
Asking student 15 question 4
Asking student 15 question 5
Asking student 16 question 1
Asking student 16 question 2
Asking student 16 question 3
Asking student 16 question 4
Asking student 16 question 5
Asking student 17 question 1
Asking student 17 question 2
Asking student 17 question 3
Asking student 17 question 4
Asking student 17 question 5
root@desktop-3ints9v:~/Desktop/PLP-Assignment#
```

Second scenario:



```
Asking student 7 question 2
Asking student 6 question 3
Asking student 7 question 3
Asking student 6 question 4
Asking student 7 question 4
Asking student 6 question 5
Asking student 7 question 5
Asking student 8 question 1
Asking student 9 question 1
Asking student 8 question 2
Asking student 9 question 2
Asking student 8 question 3
Asking student 9 question 3
Asking student 8 question 4
Asking student 9 question 4
Asking student 8 question 5
Asking student 9 question 5
Asking student 10 question 1
Asking student 11 question 1
Asking student 10 question 2
Asking student 11 question 2
Asking student 11 question 3
Asking student 10 question 3
Asking student 10 question 4
Asking student 11 question 4
Asking student 10 question 5
Asking student 11 question 5
Asking student 12 question 1
Asking student 13 question 1
Asking student 12 question 2
Asking student 13 question 2
Asking student 12 question 3
Asking student 13 question 3
Asking student 12 question 4
Asking student 13 question 4
Asking student 13 question 5
Asking student 12 question 5
Asking student 14 question 1
Asking student 15 question 1
Asking student 14 question 2
Asking student 15 question 2
Asking student 14 question 3
Asking student 15 question 3
Asking student 14 question 4
Asking student 15 question 4
Asking student 14 question 5
Asking student 15 question 5
Asking student 17 question 1
Asking student 16 question 1
Asking student 17 question 2
Asking student 16 question 2
Asking student 17 question 3
Asking student 16 question 3
Asking student 17 question 4
Asking student 16 question 4
Asking student 17 question 5
Asking student 16 question 5
```

```
root@desktop-3ints9v:~/Desktop/PLP-Assignment#
```

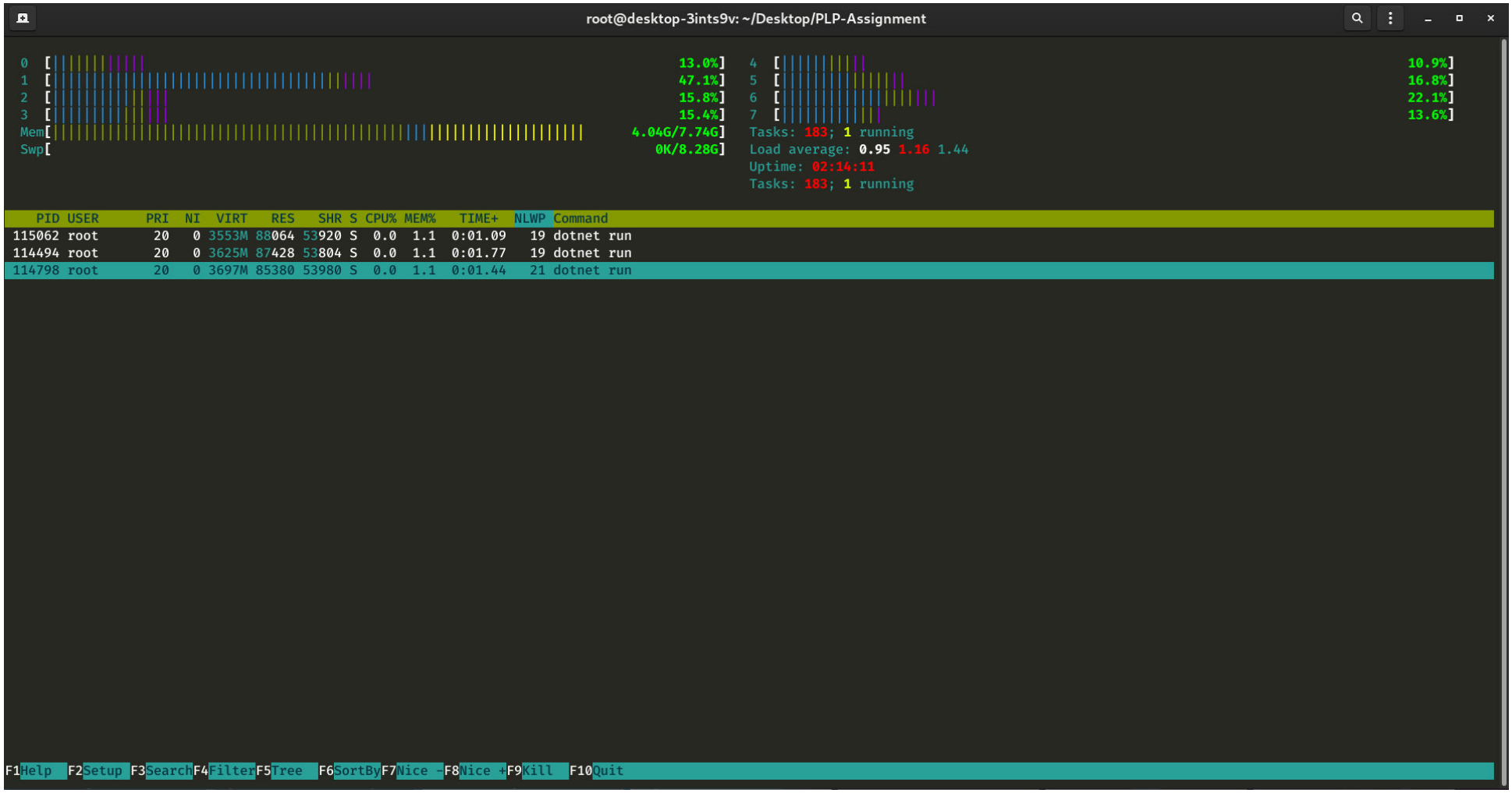
Third scenario:

```
root@desktop-3ints9v: ~/Desktop/PLP-Assignment
root@desktop-3ints9v:~/Desktop/PLP-Assignment# dotnet run
Choose which scenario to execute (1, 2, 3): 3
Asking student 4 question 1
Asking student 1 question 1
Asking student 3 question 1
Asking student 5 question 1
Asking student 4 question 2
Asking student 5 question 2
Asking student 3 question 2
Asking student 1 question 2
Asking student 4 question 3
Asking student 5 question 3
Asking student 3 question 3
Asking student 1 question 3
Asking student 4 question 4
Asking student 5 question 4
Asking student 3 question 4
Asking student 1 question 4
Asking student 4 question 5
Asking student 5 question 5
Asking student 3 question 5
Asking student 1 question 5
Asking student 6 question 1
Asking student 7 question 1
Asking student 8 question 1
Asking student 9 question 1
Asking student 6 question 2
Asking student 7 question 2
Asking student 8 question 2
Asking student 9 question 2
Asking student 6 question 3
Asking student 7 question 3
Asking student 8 question 3
Asking student 9 question 3
Asking student 6 question 4
Asking student 7 question 4
Asking student 8 question 4
Asking student 9 question 4
Asking student 6 question 5
Asking student 8 question 5
Asking student 7 question 5
Asking student 9 question 5
Asking student 10 question 1
Asking student 11 question 1
Asking student 12 question 1
```

```
Asking student 13 question 1
Asking student 10 question 2
Asking student 11 question 2
Asking student 13 question 2
Asking student 12 question 2
Asking student 10 question 3
Asking student 11 question 3
Asking student 13 question 3
Asking student 12 question 3
Asking student 11 question 4
Asking student 10 question 4
Asking student 13 question 4
Asking student 12 question 4
Asking student 11 question 5
Asking student 13 question 5
Asking student 10 question 5
Asking student 12 question 5
Asking student 14 question 1
Asking student 15 question 1
Asking student 17 question 1
Asking student 16 question 1
Asking student 14 question 2
Asking student 15 question 2
Asking student 16 question 2
Asking student 17 question 2
Asking student 14 question 3
Asking student 17 question 3
Asking student 15 question 3
Asking student 16 question 3
Asking student 17 question 4
Asking student 14 question 4
Asking student 15 question 4
Asking student 16 question 4
Asking student 17 question 5
Asking student 15 question 5
Asking student 14 question 5
Asking student 16 question 5
root@desktop-3ints9v:~/Desktop/PLP-Assignment#
```

Threads Window Output

I'm working on Linux, so i used a tool called htop to monitor threads count.



Execution times

- **For the first scenario:** one student is asked in parallel, so basically it reverts to sequential.
  Each student is asked 5 questions, and each question takes 1 second, and there exist 16 students, so the **minimum** required time is:
  5 * 1 * 16 = 80 seconds.
- **For the second scenario:** two students are asked in parallel, therefore it takes half the previous time: 80 / 2 = 40 seconds.
- **For the third scenario:** four students are asked in parallel, therefore it takes quarter the time of the first scenario: 80 / 4 = 20 seconds.

Speed Up Results

- The second scenario has an execution time faster by 100% than the first
- The third scenario has an execution time faster by 200% than the first

Those results are logical, because at first we were processing one student in parallel (sequential), while in the second we processed two students in parallel, but in the third we processed four students in parallel.