12.Apl 2023 Jeremiah Okpala /Misato Seki

PAIR PROGRAMMING ON THE MULTI-THREADING ASSIGNMENT

1. INTRODUCTION

In this report, we will write a code for digital watch using multi-threading.

2. WHAT IS THIS CODE DOING?

This code is a multi-threaded program written in C# that performs three tasks (displaying the current time, running a stopwatch, and setting and running an alarm) in parallel.

The program is managed by the TimeWatch class; the ShowCurrentTime method has an infinite loop that displays the current time every second, and the RunStopwatch method has an infinite loop that increments the stopwatch by one second.

The RunAlarm method allows the user to enter a time for the alarm and waits until that time before sounding the alarm. If the time entered is in a valid format, it calculates the time difference to the time and then pauses the thread until that time. If it is not in a valid format, it prints an error message.

The Main method creates an instance of the TimeWatch class, creates threads to perform the three tasks, and starts them. It waits for the threads to complete and finally prints the message "Done."

3. WHAT WE LEARNED

We realized that multi-threaded technology is used all around us. Because there are many devices, not only digital watches, but also smartphones, smart TVs, and other devices that perform multiple tasks simultaneously

.

Appendix

```
using System;
using System.Threading;

class Program
{
    static void Main(string[] args)
    {
        // Create a new time watch object
        TimeWatch watch = new TimeWatch();

        // Create three threads, one for each task
        Thread currentTimeThread = new Thread(new ThreadStart(watch.ShowCurrentTime));
        Thread stopwatchThread = new Thread(new ThreadStart(watch.RunStopwatch));
        Thread alarmThread = new Thread(new ThreadStart(watch.RunAlarm));

        // Start the threads
        currentTimeThread.Start();
        stopwatchThread.Start();
        alarmThread.Start();

        // Wait for the threads to complete
        currentTimeThread.Join();
        stopwatchThread.Join();
        alarmThread.Join();

        Console.WriteLine("Done.");
    }
}
```

```
class TimeWatch
{
    private bool _stopwatchRunning = false;
    private int _stopwatchSeconds = 0;

    public void ShowCurrentTime()
    {
        // This method continuously shows the current time
        while (true)
        {
            Console.WriteLine("Current time: {0:T}", DateTime.Now);
            Thread.Sleep(10000); // wait for 1 second
        }
    }

    public void RunStopwatch()
    {
        // This method runs a stopwatch
        while (true)
        {
            if (_stopwatchRunning)
            {
                Console.WriteLine("Stopwatch: {0}", _stopwatchSeconds++);
            }
            Thread.Sleep(1000); // wait for 1 second
        }
    }
```

```
58      public void RunAlarm()
59      {
60          // This method runs an alarm
61          while (true)
62          {
63              Console.WriteLine("Enter alarm time (format: HH):");
64              string input = Console.ReadLine();
65
66              DateTime alarmTime;
67
68              if (DateTime.TryParseExact(input, "HH:mm", null, System.Globalization.DateTimeStyles.None,
    out alarmTime))
69              {
70                  TimeSpan timeUntilAlarm = alarmTime - DateTime.Now;
71                  Console.WriteLine(timeUntilAlarm);
72
73                  if (timeUntilAlarm.TotalSeconds > 0)
74                  {
75                      Console.WriteLine("Alarm set for {0:T}.", alarmTime);
76                      Thread.Sleep((int)timeUntilAlarm.TotalMilliseconds); // wait for time until alarm
77                      Console.WriteLine("ALARM!");
78                  }
79                  else
80                  {
81                Console.WriteLine("Alarm set for {0:T}.", alarmTime);
82
83                  }
84              }
85              else
86              {
87                  Console.WriteLine("Invalid alarm time.");
```

Picture1. code for digital watch (C#).

```
Current time: 8:17:18AM
08:18Current time: 8:17:28AM

00:00:31.0969375
Alarm set for 8:18:00AM.
Current time: 8:17:38AM
Current time: 8:17:48AM
Current time: 8:17:58AM
ALARM!
Enter alarm time (format: HH):
exit status 143
```

Picture2. executed result.