An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Computer Engineering Department

Course name - number: Operating Systems - 10636451

Report Grading Sheet

Instructor name: Dr. Sulaiman Abu Kharma	Assignment 2
Academic year: 2022/2023	Semester: Second

Students	
Amjad Kayed - 12028467	Riham Katout - 12029366

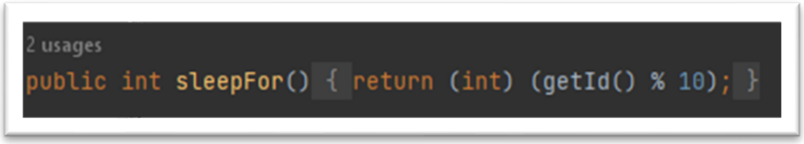
Contents

Part1 – Code: without synchronization

In the first part, we have two classes:

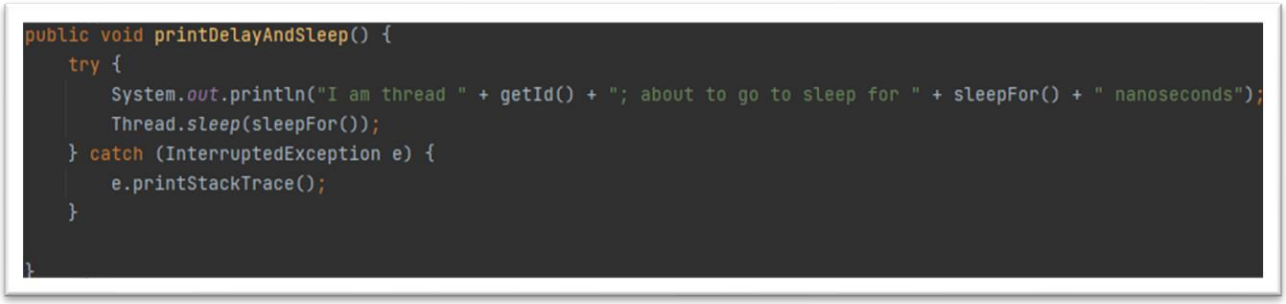
1. **“`Part1Thread`” class**, which represents the thread that will increase the value of the shared memory. It has these following methods:

- `sleepFor` – returns the time the thread will sleep



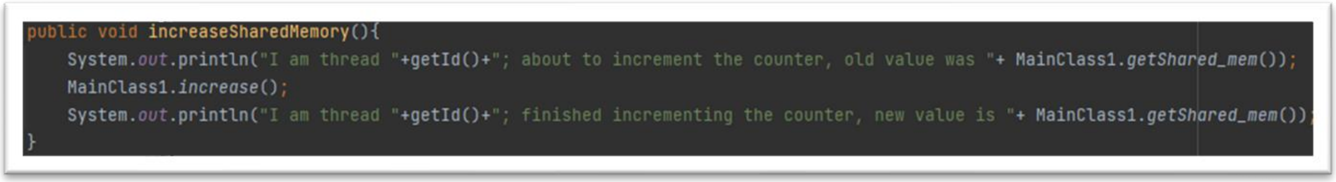
```
2 usages
public int sleepFor() { return (int) (getId() % 10); }
```

- `printDelayAndSleep` – prints the requested message then calls “`sleep`” method for the thread.



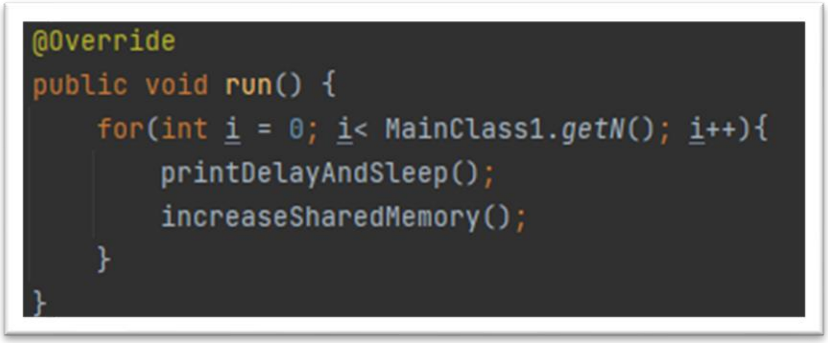
```
public void printDelayAndSleep() {
    try {
        System.out.println("I am thread " + getId() + "; about to go to sleep for " + sleepFor() + " nanoseconds");
        Thread.sleep(sleepFor());
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

}
```

- `increaseSharedMemory` – to call increase method from the main so we can increase the value of shared_mem



```
public void increaseSharedMemory(){
    System.out.println("I am thread "+getId()+"; about to increment the counter, old value was "+ MainClass1.getShared_mem());
    MainClass1.increase();
    System.out.println("I am thread "+getId()+"; finished incrementing the counter, new value is "+ MainClass1.getShared_mem());
}
```

- `run`, this method will automatically be called when the thread starts

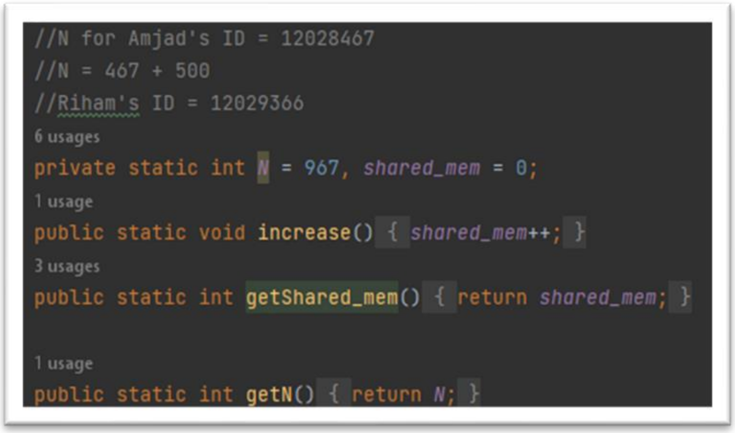


```
@Override
public void run() {
    for(int i = 0; i< MainClass1.getN(); i++){
        printDelayAndSleep();
        increaseSharedMemory();
    }
}
```

This is the main function in the class which will call “`printDelayAndSleep`” & “`increaseSharedMemory`” methods **N** times for each thread

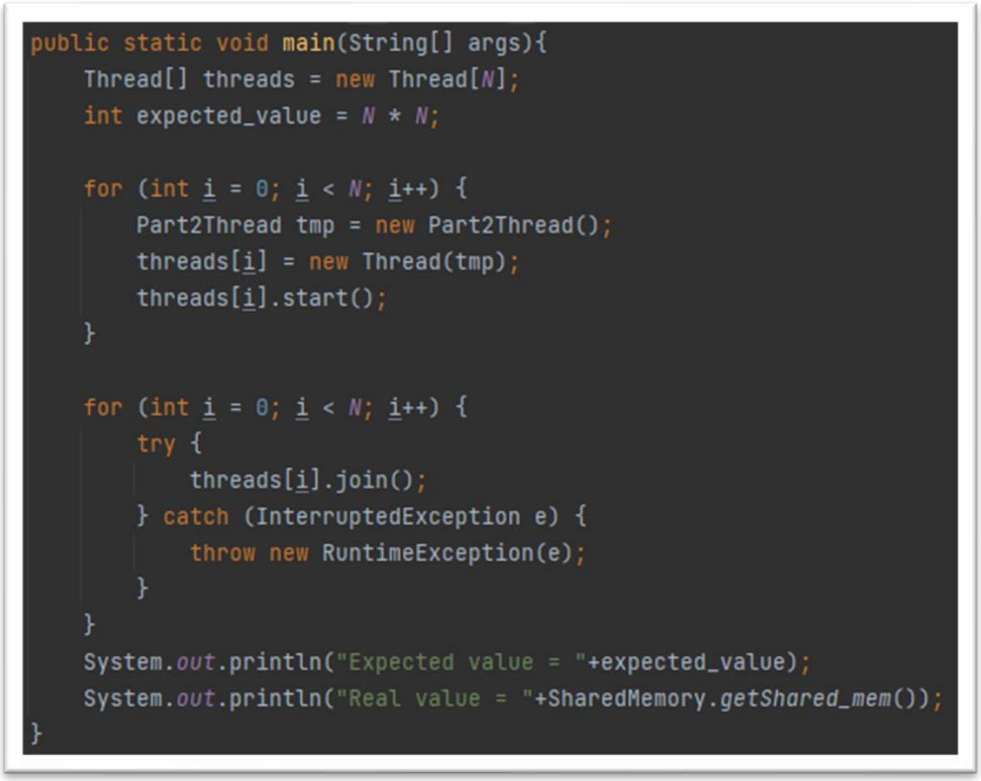
2. **“MainClass1”** class, which contains:

❖ shared-mem, its methods (getter & setter), the variable N and its getter



```
//N for Amjad's ID = 12028467
//N = 467 + 500
//Riham's ID = 12029366
6 usages
private static int N = 967, shared_mem = 0;
1 usage
public static void increase() { shared_mem++; }
3 usages
public static int getShared_mem() { return shared_mem; }

1 usage
public static int getN() { return N; }
```

❖ Main method



```
public static void main(String[] args){
    Thread[] threads = new Thread[N];
    int expected_value = N * N;

    for (int i = 0; i < N; i++) {
        Part2Thread tmp = new Part2Thread();
        threads[i] = new Thread(tmp);
        threads[i].start();
    }

    for (int i = 0; i < N; i++) {
        try {
            threads[i].join();
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    }
    System.out.println("Expected value = "+expected_value);
    System.out.println("Real value = "+SharedMemory.getShared_mem());
}
```

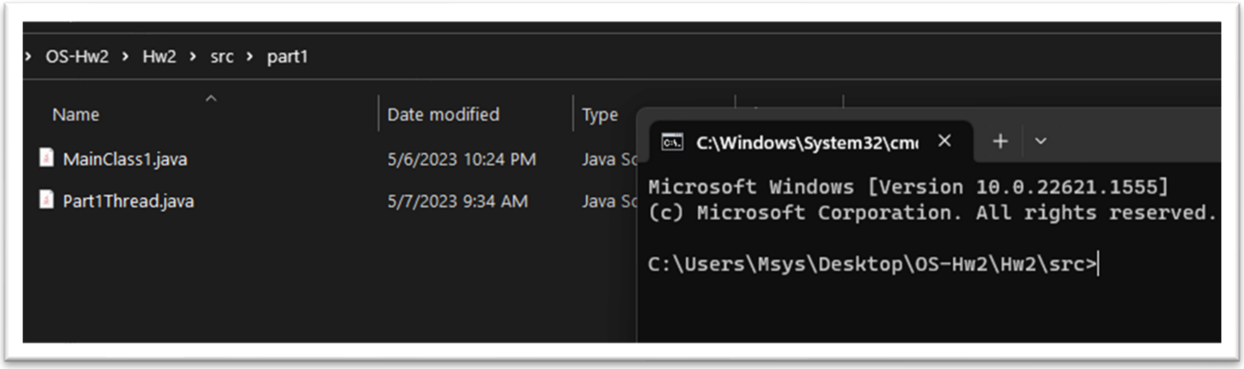
First, we created an array of threads with size **N**, then created **N** objects of our class Part1Thread, assigned each one to a pointer in our threads array and start it using “start” method.

The next step is iterating over threads array, and calling “join” method, so the parent thread won’t continue running until all the **N** threads finish their work.

Finally, we printed the expected value (= N^2; we have **N** threads each will call “increase” method **N** times), and the real value we have.

Part1 – Result

On windows11 OS

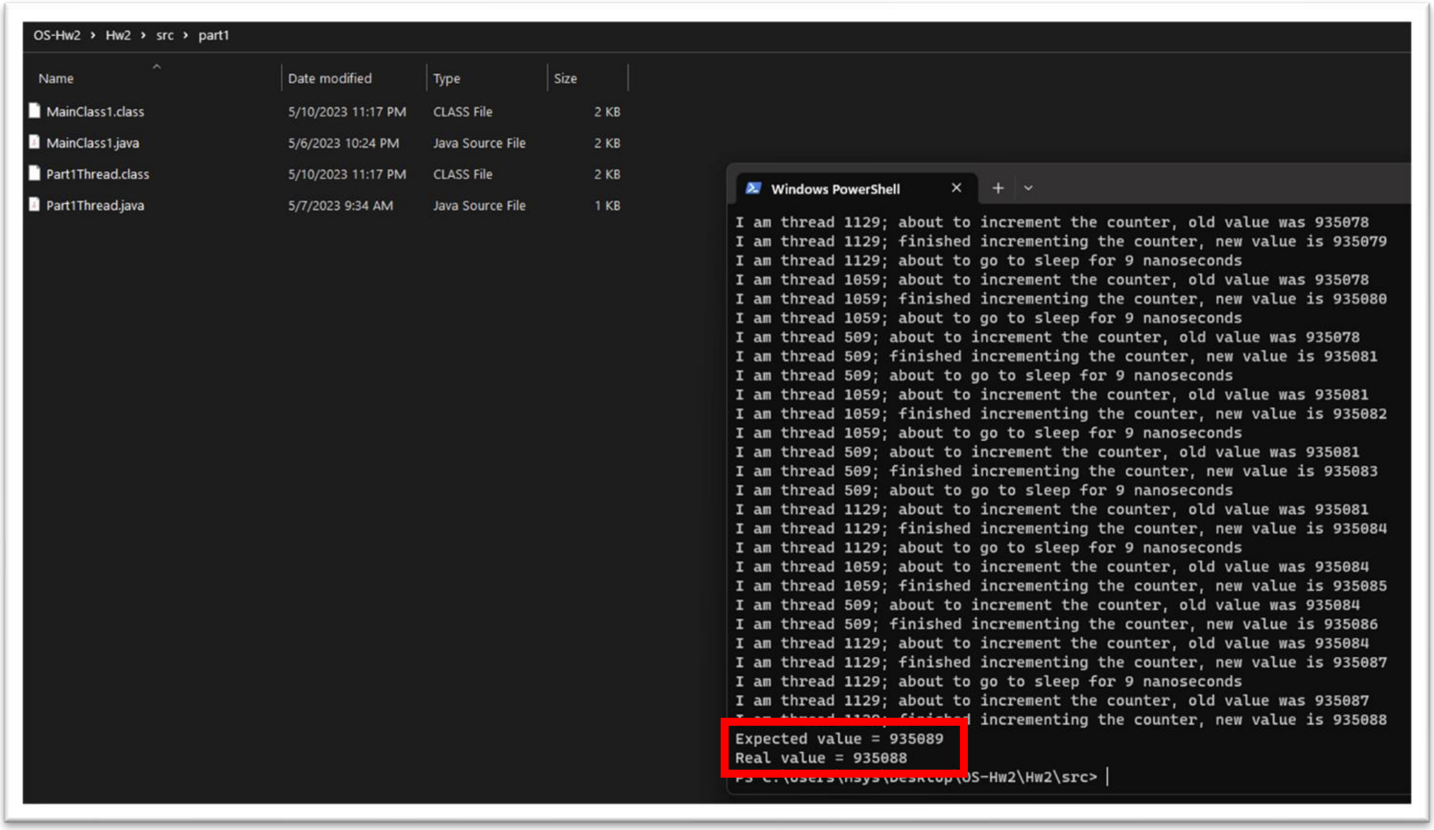


› OS-Hw2 › Hw2 › src › part1

Name	Date modified	Type
MainClass1.java	5/6/2023 10:24 PM	Java Sc
Part1Thread.java	5/7/2023 9:34 AM	Java Sc

```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Msys\Desktop\OS-Hw2\Hw2\src>
```

Figure 1: Part1 classes on windows before compiling.



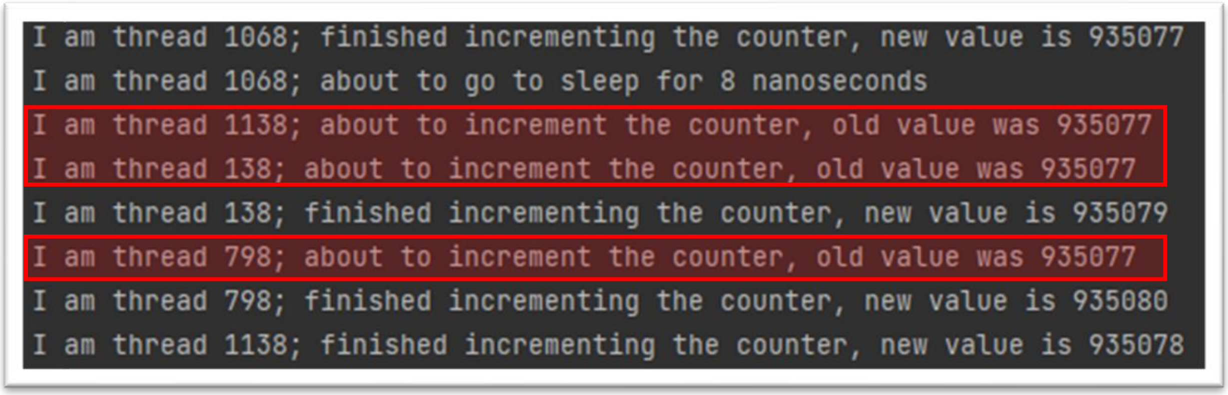
OS-Hw2 › Hw2 › src › part1

Name	Date modified	Type	Size
MainClass1.class	5/10/2023 11:17 PM	CLASS File	2 KB
MainClass1.java	5/6/2023 10:24 PM	Java Source File	2 KB
Part1Thread.class	5/10/2023 11:17 PM	CLASS File	2 KB
Part1Thread.java	5/7/2023 9:34 AM	Java Source File	1 KB

```
I am thread 1129; about to increment the counter, old value was 935078
I am thread 1129; finished incrementing the counter, new value is 935079
I am thread 1129; about to go to sleep for 9 nanoseconds
I am thread 1059; about to increment the counter, old value was 935078
I am thread 1059; finished incrementing the counter, new value is 935080
I am thread 1059; about to go to sleep for 9 nanoseconds
I am thread 509; about to increment the counter, old value was 935078
I am thread 509; finished incrementing the counter, new value is 935081
I am thread 509; about to go to sleep for 9 nanoseconds
I am thread 1059; about to increment the counter, old value was 935081
I am thread 1059; finished incrementing the counter, new value is 935082
I am thread 1059; about to go to sleep for 9 nanoseconds
I am thread 509; about to increment the counter, old value was 935081
I am thread 509; finished incrementing the counter, new value is 935083
I am thread 509; about to go to sleep for 9 nanoseconds
I am thread 1129; about to increment the counter, old value was 935081
I am thread 1129; finished incrementing the counter, new value is 935084
I am thread 1129; about to go to sleep for 9 nanoseconds
I am thread 1059; about to increment the counter, old value was 935084
I am thread 1059; finished incrementing the counter, new value is 935085
I am thread 509; about to increment the counter, old value was 935084
I am thread 509; finished incrementing the counter, new value is 935086
I am thread 1129; about to increment the counter, old value was 935084
I am thread 1129; finished incrementing the counter, new value is 935087
I am thread 1129; about to go to sleep for 9 nanoseconds
I am thread 1129; about to increment the counter, old value was 935087
I am thread 1129; finished incrementing the counter, new value is 935088
Expected value = 935089
Real value = 935088
PS C:\Users\Msys\Desktop\OS-Hw2\Hw2\src> |
```

Figure 2: Part1 compiling and run result.

- ➢ as we can see, the real result is different than the expected, that is caused by multithreads calling ”increase” method at the same time.

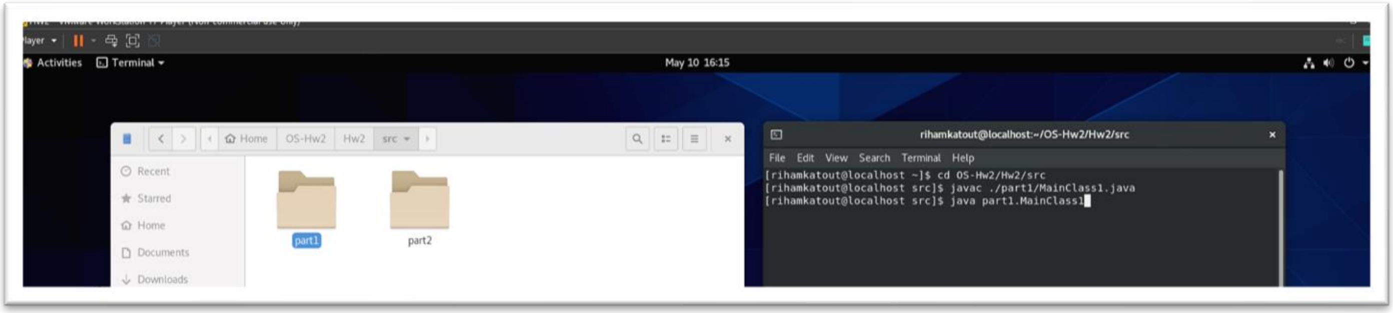


```
I am thread 1068; finished incrementing the counter, new value is 935077
I am thread 1068; about to go to sleep for 8 nanoseconds
I am thread 1138; about to increment the counter, old value was 935077
I am thread 138; about to increment the counter, old value was 935077
I am thread 138; finished incrementing the counter, new value is 935079
I am thread 798; about to increment the counter, old value was 935077
I am thread 798; finished incrementing the counter, new value is 935080
I am thread 1138; finished incrementing the counter, new value is 935078
```

Figure 3: Error example

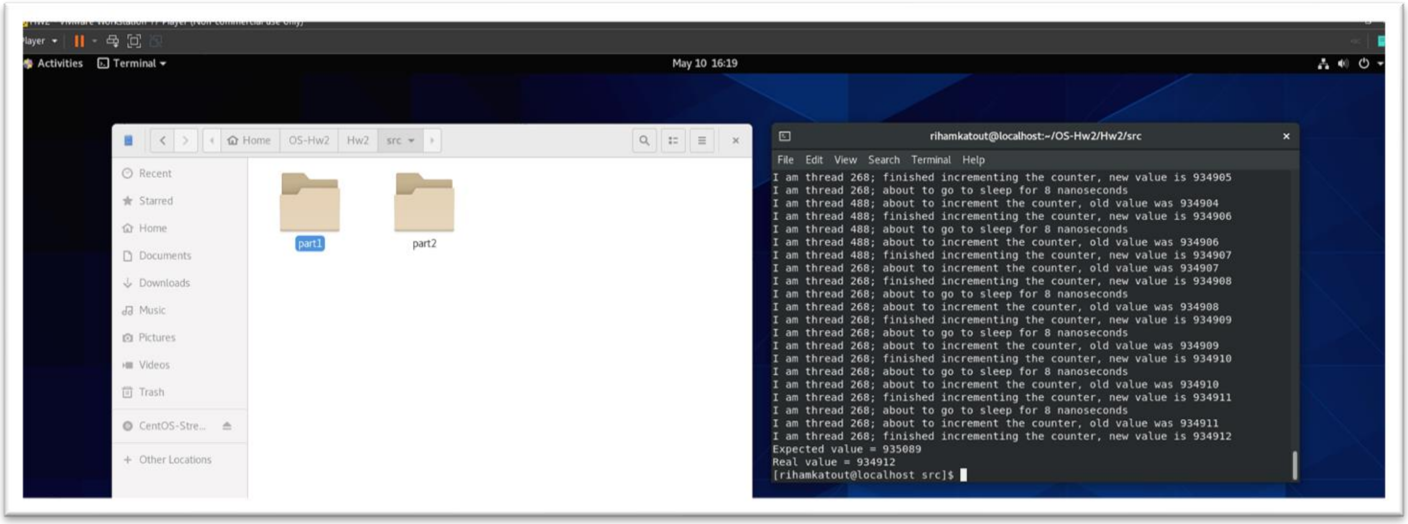
- ➢ we notice that there are 3 threads called “`increase`” method at the same time.

On centOS8



```
[rihamkatout@localhost ~]$ cd OS-Hw2/Hw2/src
[rihamkatout@localhost src]$ javac ./part1/MainClass1.java
[rihamkatout@localhost src]$ java part1.MainClass1
```

Figure 4: files of the project on CentOS



```
I am thread 268; finished incrementing the counter, new value is 934905
I am thread 268; about to go to sleep for 8 nanoseconds
I am thread 488; about to increment the counter, old value was 934904
I am thread 488; finished incrementing the counter, new value is 934906
I am thread 488; about to go to sleep for 8 nanoseconds
I am thread 488; about to increment the counter, old value was 934906
I am thread 488; finished incrementing the counter, new value is 934907
I am thread 268; about to increment the counter, old value was 934907
I am thread 268; finished incrementing the counter, new value is 934908
I am thread 268; about to go to sleep for 8 nanoseconds
I am thread 268; about to increment the counter, old value was 934908
I am thread 268; finished incrementing the counter, new value is 934909
I am thread 268; about to go to sleep for 8 nanoseconds
I am thread 268; about to increment the counter, old value was 934909
I am thread 268; finished incrementing the counter, new value is 934910
I am thread 268; about to go to sleep for 8 nanoseconds
I am thread 268; about to increment the counter, old value was 934910
I am thread 268; finished incrementing the counter, new value is 934911
I am thread 268; about to go to sleep for 8 nanoseconds
I am thread 268; about to increment the counter, old value was 934911
I am thread 268; finished incrementing the counter, new value is 934912
Expected value = 935089
Real value = 934912
[rihamkatout@localhost src]$
```

Figure 5: Compile and run result on CentOS

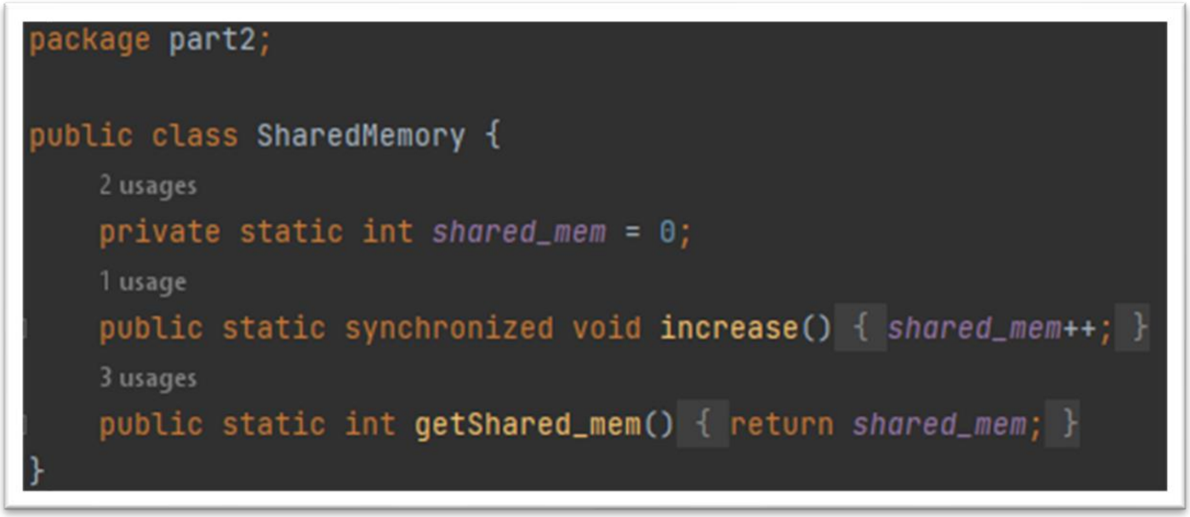
- ➢ Result is same to the one on windows

Part2 – Code: with synchronization

We used the third method (static synchronization) according to our IDs (12029366 + 12028467) % 3 = 2

In the second part, we have three classes:

1. **“SharedMemory” class**, contains shared_memory, its getter and setter.



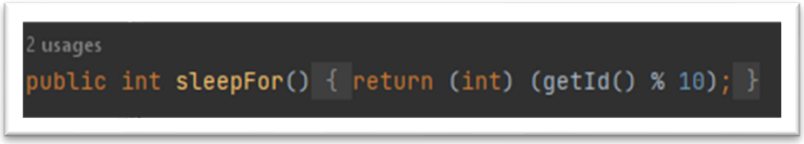
```
package part2;

public class SharedMemory {
    2 usages
    private static int shared_mem = 0;
    1 usage
    public static synchronized void increase() { shared_mem++; }
    3 usages
    public static int getShared_mem() { return shared_mem; }
}
```

Figure 6: ShaedMemory Class contents

- ➢ increase method is static & synchronized, that means that it is running only by one thread at the same time, the other threads will be waiting in a queue, so we can guarantee that we will get the correct value of it.

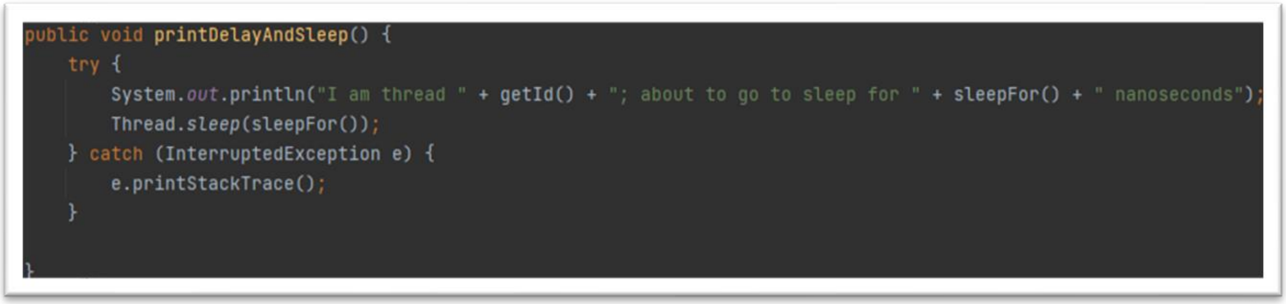
2. **“Part2Thread” class**, which represents the thread that will increase the value of the shared memory. It has these following methods:

- sleepFor – returns the time the thread will sleep



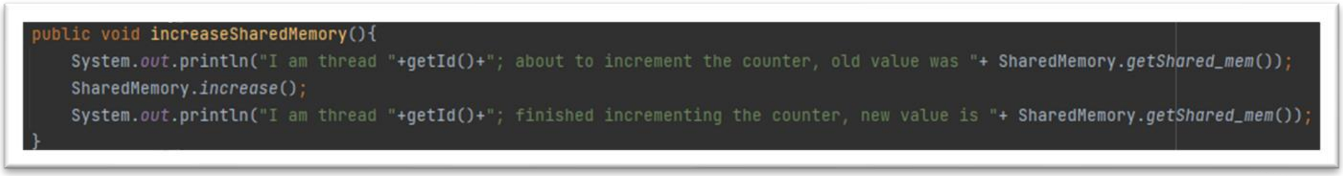
```
2 usages
public int sleepFor() { return (int) (getId() % 10); }
```

- printDelayAndSleep – prints the requested message then calls “sleep” method for the thread.



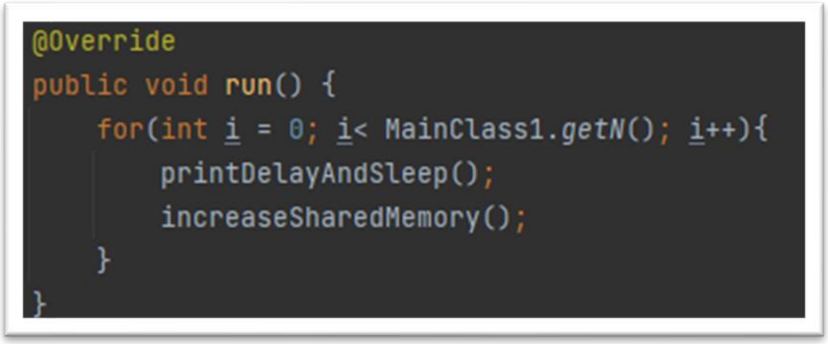
```
public void printDelayAndSleep() {
    try {
        System.out.println("I am thread " + getId() + "; about to go to sleep for " + sleepFor() + " nanoseconds");
        Thread.sleep(sleepFor());
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

}
```

- increaseSharedMemory – to call increase method from SharedMemory class



```
public void increaseSharedMemory(){
    System.out.println("I am thread "+getId()+"; about to increment the counter, old value was "+ SharedMemory.getShared_mem());
    SharedMemory.increase();
    System.out.println("I am thread "+getId()+"; finished incrementing the counter, new value is "+ SharedMemory.getShared_mem());
}
```

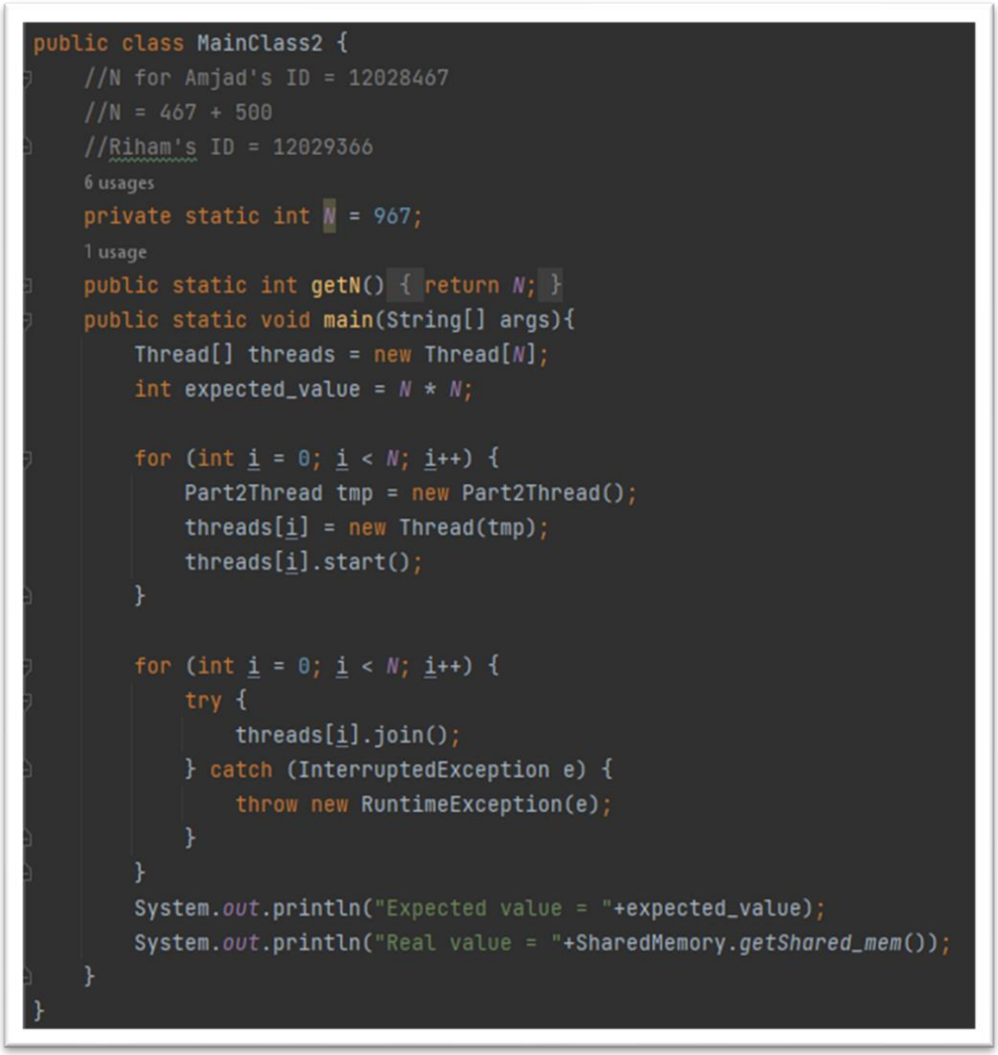
- run, this method will automatically be called when the thread starts



```
@Override
public void run() {
    for(int i = 0; i< MainClass1.getN(); i++){
        printDelayAndSleep();
        increaseSharedMemory();
    }
}
```

This is the main function in the class which will call “printDelayAndSleep” & “increaseSharedMemory” methods **N** times for each thread

3. **“MainClass2”** class, which contains **N** and the main method only.



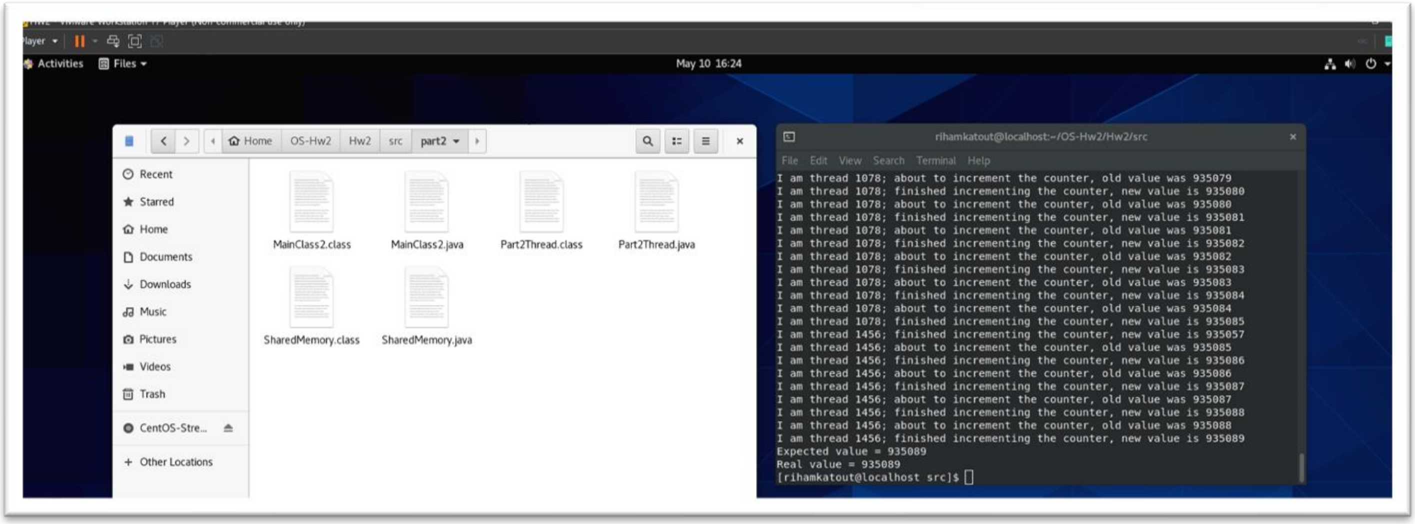
```java
public class MainClass2 {
    //N for Amjad's ID = 12028467
    //N = 467 + 500
    //Riham's ID = 12029366
    6 usages
    private static int N = 967;
    1 usage
    public static int getN() { return N; }
    public static void main(String[] args){
        Thread[] threads = new Thread[N];
        int expected_value = N * N;

        for (int i = 0; i < N; i++) {
            Part2Thread tmp = new Part2Thread();
            threads[i] = new Thread(tmp);
            threads[i].start();
        }

        for (int i = 0; i < N; i++) {
            try {
                threads[i].join();
            } catch (InterruptedException e) {
                throw new RuntimeException(e);
            }
        }
        System.out.println("Expected value = "+expected_value);
        System.out.println("Real value = "+SharedMemory.getShared_mem());
    }
}
```

Figure 7: MainClass2 - part2

The process is same to part1, but instead of calling the value of shared_mem from the main, we called it from `SharedMemory` class.

Part2 – Result

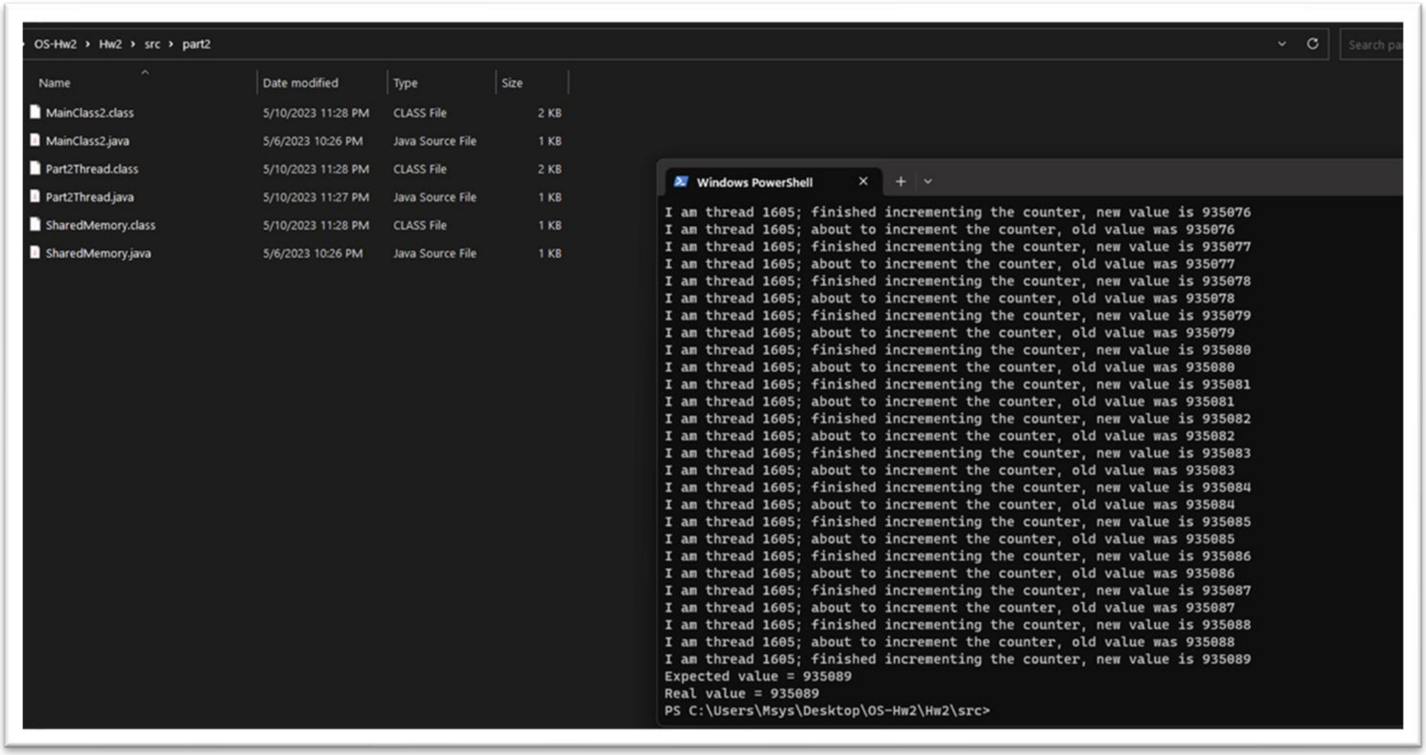
On centOS8



```
I am thread 1078; about to increment the counter, old value was 935079
I am thread 1078; finished incrementing the counter, new value is 935080
I am thread 1078; about to increment the counter, old value was 935080
I am thread 1078; finished incrementing the counter, new value is 935081
I am thread 1078; about to increment the counter, old value was 935081
I am thread 1078; finished incrementing the counter, new value is 935082
I am thread 1078; about to increment the counter, old value was 935082
I am thread 1078; finished incrementing the counter, new value is 935083
I am thread 1078; about to increment the counter, old value was 935083
I am thread 1078; finished incrementing the counter, new value is 935084
I am thread 1078; about to increment the counter, old value was 935084
I am thread 1078; finished incrementing the counter, new value is 935085
I am thread 1456; finished incrementing the counter, new value is 935057
I am thread 1456; about to increment the counter, old value was 935085
I am thread 1456; finished incrementing the counter, new value is 935086
I am thread 1456; about to increment the counter, old value was 935086
I am thread 1456; finished incrementing the counter, new value is 935087
I am thread 1456; about to increment the counter, old value was 935087
I am thread 1456; finished incrementing the counter, new value is 935088
I am thread 1456; about to increment the counter, old value was 935088
I am thread 1456; finished incrementing the counter, new value is 935089
Expected value = 935089
Real value = 935089
[rihamkatout@localhost src]$ 
```

Figure 8: running part2 on centOS

On windows11



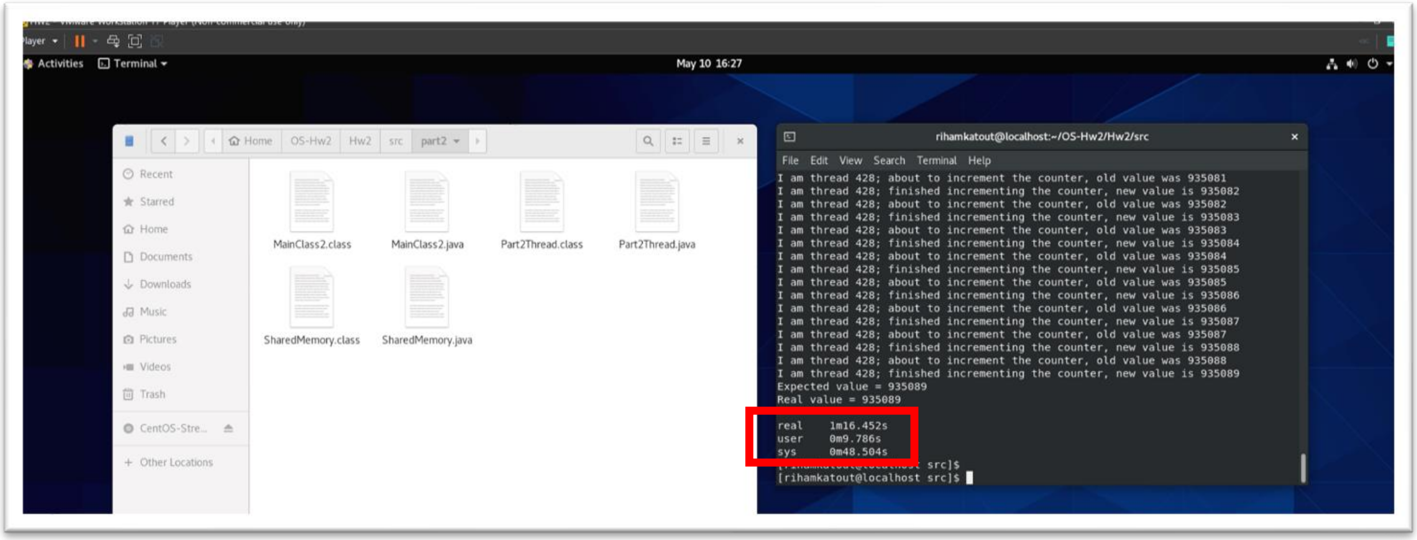
Name	Date modified	Type	Size
MainClass2.class	5/10/2023 11:28 PM	CLASS File	2 KB
MainClass2.java	5/6/2023 10:26 PM	Java Source File	1 KB
Part2Thread.class	5/10/2023 11:28 PM	CLASS File	2 KB
Part2Thread.java	5/10/2023 11:27 PM	Java Source File	1 KB
SharedMemory.class	5/10/2023 11:28 PM	CLASS File	1 KB
SharedMemory.java	5/6/2023 10:26 PM	Java Source File	1 KB

```
I am thread 1605; finished incrementing the counter, new value is 935076
I am thread 1605; about to increment the counter, old value was 935076
I am thread 1605; finished incrementing the counter, new value is 935077
I am thread 1605; about to increment the counter, old value was 935077
I am thread 1605; finished incrementing the counter, new value is 935078
I am thread 1605; about to increment the counter, old value was 935078
I am thread 1605; finished incrementing the counter, new value is 935079
I am thread 1605; about to increment the counter, old value was 935079
I am thread 1605; finished incrementing the counter, new value is 935080
I am thread 1605; about to increment the counter, old value was 935080
I am thread 1605; finished incrementing the counter, new value is 935081
I am thread 1605; about to increment the counter, old value was 935081
I am thread 1605; finished incrementing the counter, new value is 935082
I am thread 1605; about to increment the counter, old value was 935082
I am thread 1605; finished incrementing the counter, new value is 935083
I am thread 1605; about to increment the counter, old value was 935083
I am thread 1605; finished incrementing the counter, new value is 935084
I am thread 1605; about to increment the counter, old value was 935084
I am thread 1605; finished incrementing the counter, new value is 935085
I am thread 1605; about to increment the counter, old value was 935085
I am thread 1605; finished incrementing the counter, new value is 935086
I am thread 1605; about to increment the counter, old value was 935086
I am thread 1605; finished incrementing the counter, new value is 935087
I am thread 1605; about to increment the counter, old value was 935087
I am thread 1605; finished incrementing the counter, new value is 935088
I am thread 1605; about to increment the counter, old value was 935088
I am thread 1605; finished incrementing the counter, new value is 935089
Expected value = 935089
Real value = 935089
PS C:\Users\Msys\Desktop\OS-Hw2\Hw2\src>
```

Figure 9: running it on windows

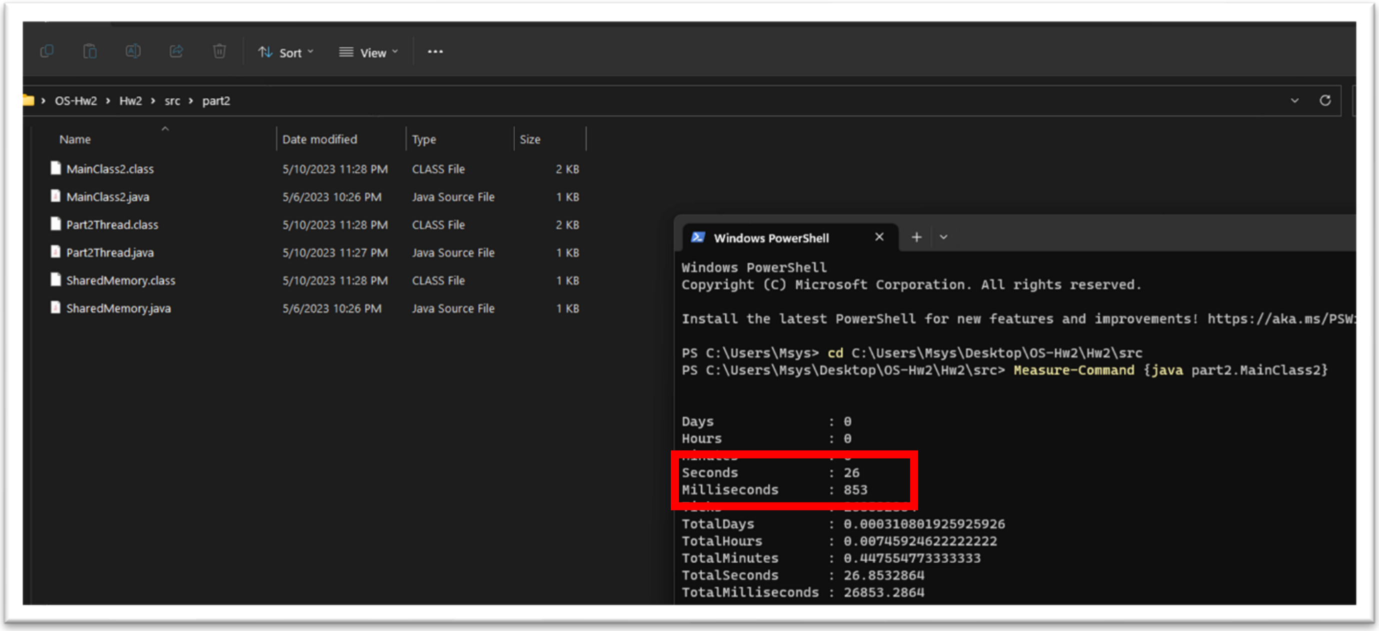
- We got the same value as we expected 😊

Compression between centOS & windows11



```
I am thread 428; about to increment the counter, old value was 935081
I am thread 428; finished incrementing the counter, new value is 935082
I am thread 428; about to increment the counter, old value was 935082
I am thread 428; finished incrementing the counter, new value is 935083
I am thread 428; about to increment the counter, old value was 935083
I am thread 428; finished incrementing the counter, new value is 935084
I am thread 428; about to increment the counter, old value was 935084
I am thread 428; finished incrementing the counter, new value is 935085
I am thread 428; about to increment the counter, old value was 935085
I am thread 428; finished incrementing the counter, new value is 935086
I am thread 428; about to increment the counter, old value was 935086
I am thread 428; finished incrementing the counter, new value is 935087
I am thread 428; about to increment the counter, old value was 935087
I am thread 428; finished incrementing the counter, new value is 935088
I am thread 428; about to increment the counter, old value was 935088
I am thread 428; finished incrementing the counter, new value is 935089
Expected value = 935089
Real value = 935089

real    1m16.452s
user    0m9.786s
sys     0m48.504s
[rihamkatout@localhost src]$
[rihamkatout@localhost src]$
```

Figure 10: CentOS using time command



Name	Date modified	Type	Size
MainClass2.class	5/10/2023 11:28 PM	CLASS File	2 KB
MainClass2.java	5/6/2023 10:26 PM	Java Source File	1 KB
Part2Thread.class	5/10/2023 11:28 PM	CLASS File	2 KB
Part2Thread.java	5/10/2023 11:27 PM	Java Source File	1 KB
SharedMemory.class	5/10/2023 11:28 PM	CLASS File	1 KB
SharedMemory.java	5/6/2023 10:26 PM	Java Source File	1 KB

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSW

PS C:\Users\Msys> cd C:\Users\Msys\Desktop\OS-Hw2\Hw2\src
PS C:\Users\Msys\Desktop\OS-Hw2\Hw2\src> Measure-Command {java part2.MainClass2}


Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 26
Milliseconds      : 853
Ticks             : 268532864
TotalDays         : 0.000310801925925926
TotalHours        : 0.00745924622222222
TotalMinutes      : 0.447554773333333
TotalSeconds      : 26.8532864
TotalMilliseconds : 26853.2864
```

Figure 9: Windows using measure-command command

➢ It’s clear that Windows is approximately 3times faster than CentOS VM, this is due many reasons.

- **Why Windows is faster?**

1. **Hardware configuration:** the host machine has better compatibility, such as specific drivers or hardware features that enhance Windows performance.

2. **Virtualization software:** different virtualization platforms may have varying levels of performance optimization for different operating systems.

3. **Resource allocation:** the amount of CPU, memory, and other resources allocated to each VM affect its performance

4. **Software configuration:** the specific software packages and configurations on the CentOS VM may have performance implications. Suboptimal settings or configurations on CentOS could lead to reduced performance compared to a well-optimized Windows setup.