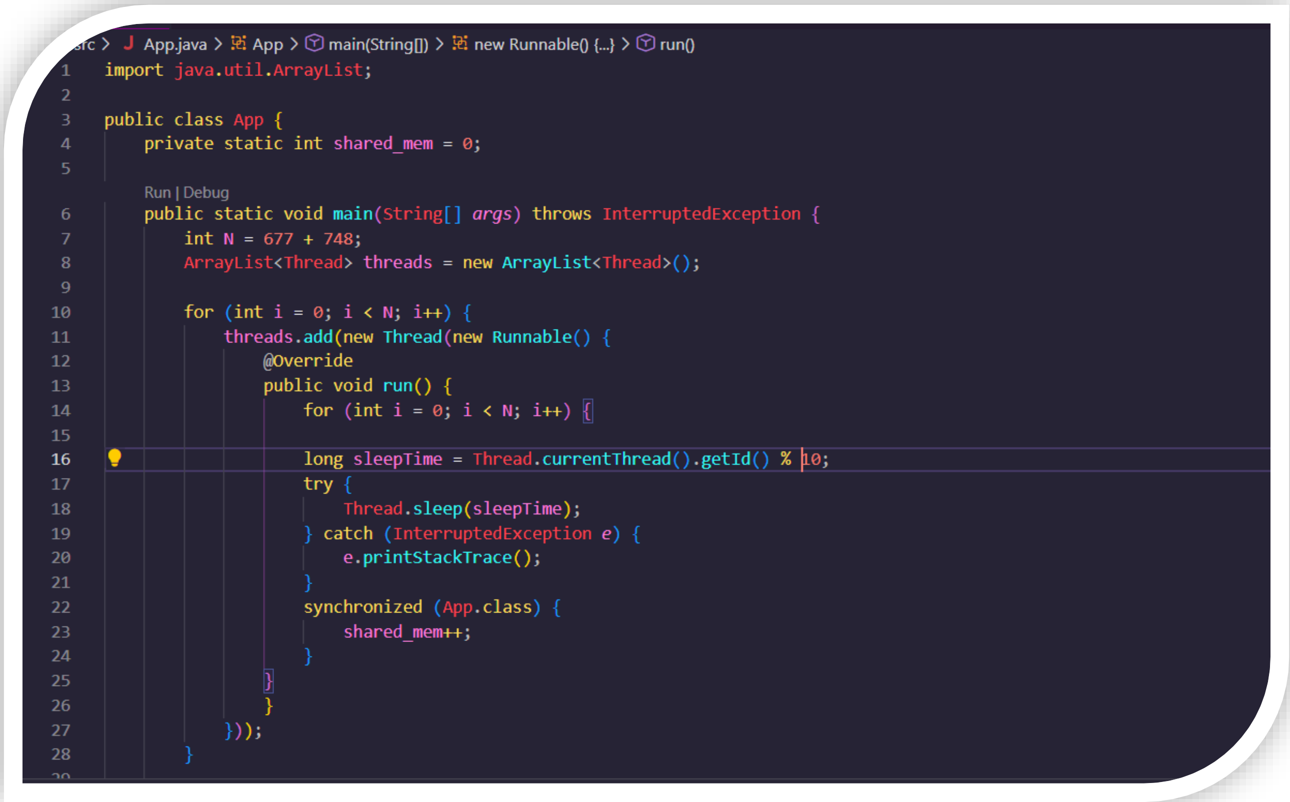
An-Najah National University

1918 AD 1337 Hijri

Batool Shilleh	**11923748**
Manar Mohamed	**11924677**
Sun-Tue-Thu	**11:00-12:00**

Code discussion:


rc > App.java > App > main(String[]) > new Runnable() {...} > run()
Run | Debug

```java
import java.util.ArrayList;

public class App {
    private static int shared_mem = 0;

    public static void main(String[] args) throws InterruptedException {
        int N = 677 + 748;
        ArrayList<Thread> threads = new ArrayList<Thread>();

        for (int i = 0; i < N; i++) {
            threads.add(new Thread(new Runnable() {
                @Override
                public void run() {
                    for (int i = 0; i < N; i++) {

                    long sleepTime = Thread.currentThread().getId() % 10;
                    try {
                        Thread.sleep(sleepTime);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                    synchronized (App.class) {
                        shared_mem++;
                    }
                }
                }
            }));
        }
```

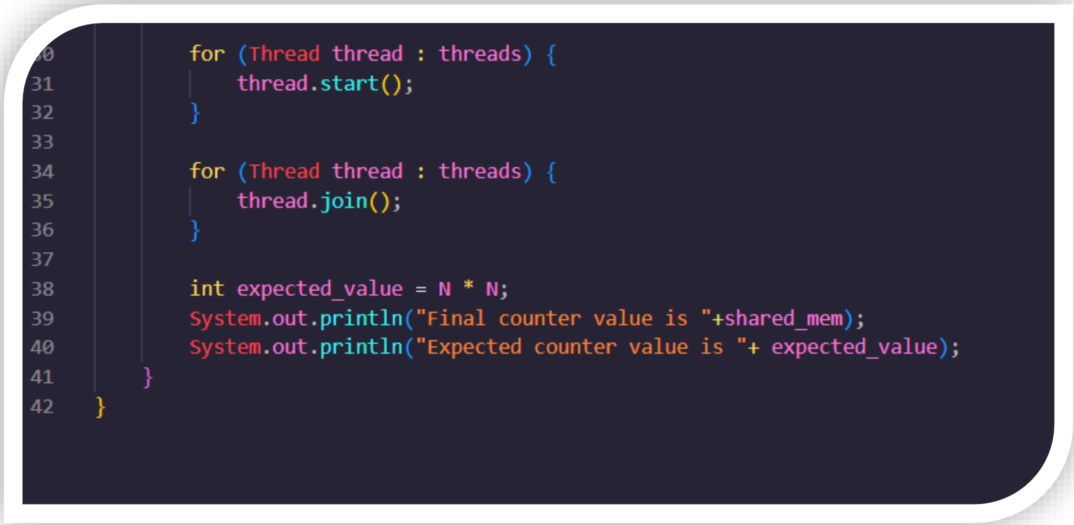
shared_mem is the thread shared variable whose value will change

N is the number of threads to be created

threads Arai are not porridge is stored in them

Inside the loop:

- We create a new thread and store it in threads
- We start by defining the run function that will run the operations inside the thread
- sleepTime We calculate the time during which Thread will sleep, which is 10% ID
- We start with the sleep process
- Then, before incrementing the shared_mem value, we call the synchronized function

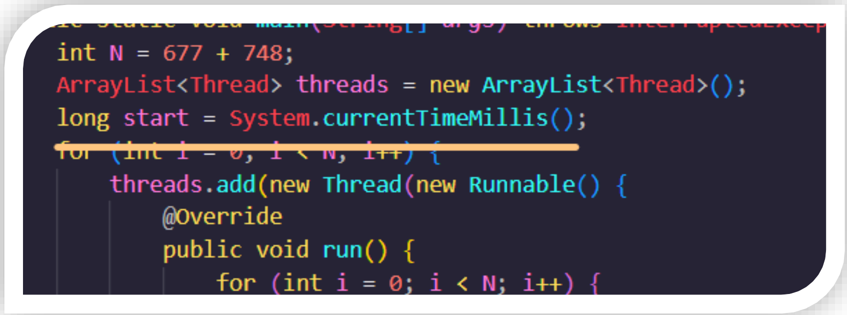


```
        for (Thread thread : threads) {
            thread.start();
        }

        for (Thread thread : threads) {
            thread.join();
        }

        int expected_value = N * N;
        System.out.println("Final counter value is "+shared_mem);
        System.out.println("Expected counter value is "+ expected_value);
    }
}
```

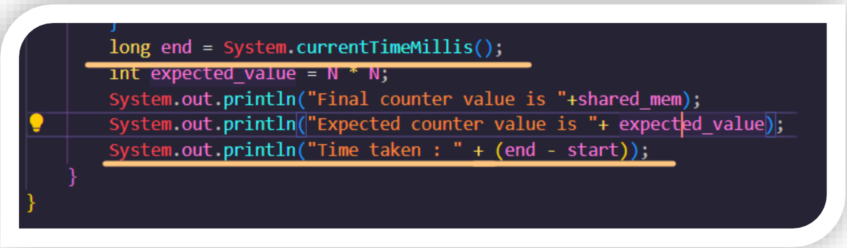
The first loop calls the start function to start its run task

The second loop forces the porridge to wait for the main porridge to finish its work

In the end, we calculated the expected value, printed it, and printed the resulting value to compare them



```
ic static void main(String[] args) throws InterruptedExcep
int N = 677 + 748;
ArrayList<Thread> threads = new ArrayList<Thread>();
long start = System.currentTimeMillis();
for (int i = 0; i < N; i++) {
    threads.add(new Thread(new Runnable() {
        @Override
        public void run() {
            for (int i = 0; i < N; i++) {
```



```
        }
        long end = System.currentTimeMillis();
        int expected_value = N * N;
        System.out.println("Final counter value is "+shared_mem);
        System.out.println("Expected counter value is "+ expected_value);
        System.out.println("Time taken : " + (end - start));
    }
}
```

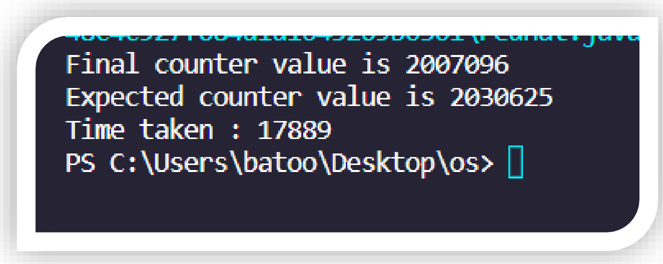
To calculate the time but this is not the way we want, we will use the command method

Linux command:
time java App

Windows command:
Measure-Command {java App}

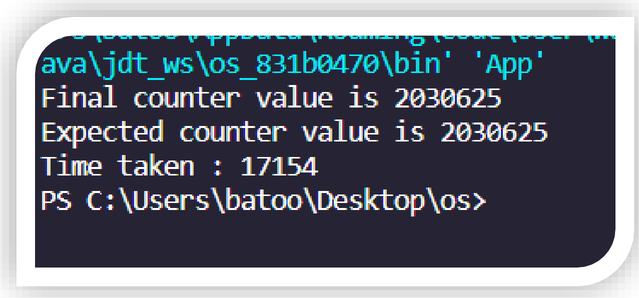
Output discussion:

- ## Windows:
  - ### Without synchronization:



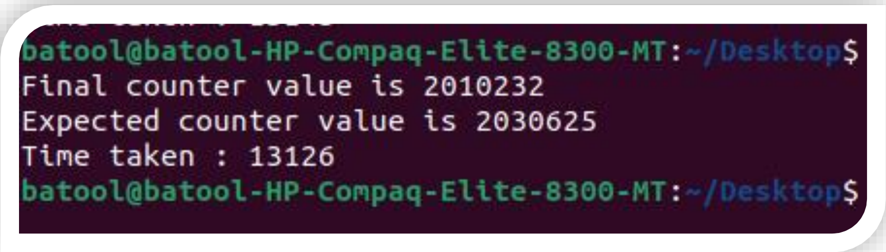
```
48c4e327f684a1a1e43203b6361\redhat.java
Final counter value is 2007096
Expected counter value is 2030625
Time taken : 17889
PS C:\Users\batoo\Desktop\os> 
```

  - ### With synchronization:



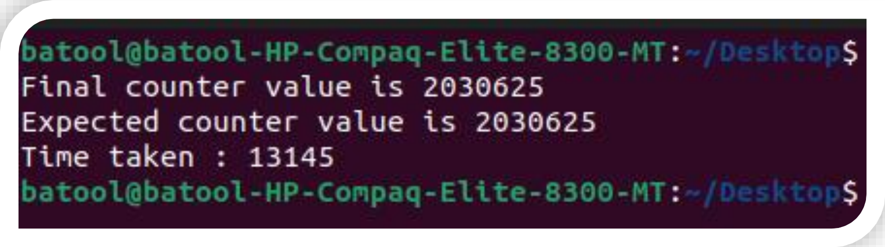
```
s\batoo\AppData\Roaming\Code\User\w
ava\jdt_ws\os_831b0470\bin' 'App'
Final counter value is 2030625
Expected counter value is 2030625
Time taken : 17154
PS C:\Users\batoo\Desktop\os>
```

- **Linux:**
  - **Without synchronization:**



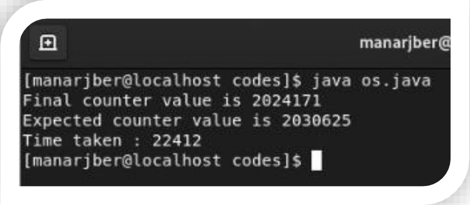
```
batool@batool-HP-Compaq-Elite-8300-MT:~/Desktop$
Final counter value is 2010232
Expected counter value is 2030625
Time taken : 13126
batool@batool-HP-Compaq-Elite-8300-MT:~/Desktop$
```

  - **With synchronization:**



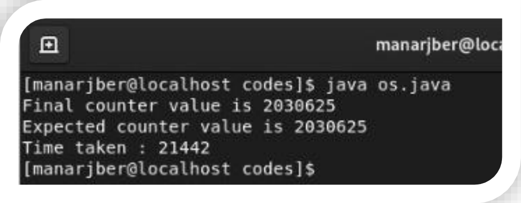
```
batool@batool-HP-Compaq-Elite-8300-MT:~/Desktop$
Final counter value is 2030625
Expected counter value is 2030625
Time taken : 13145
batool@batool-HP-Compaq-Elite-8300-MT:~/Desktop$
```

- **VM:**
  - **Without synchronization:**



```
[manarjber@localhost codes]$ java os.java
Final counter value is 2024171
Expected counter value is 2030625
Time taken : 22412
[manarjber@localhost codes]$ █
```

  - **With synchronization:**



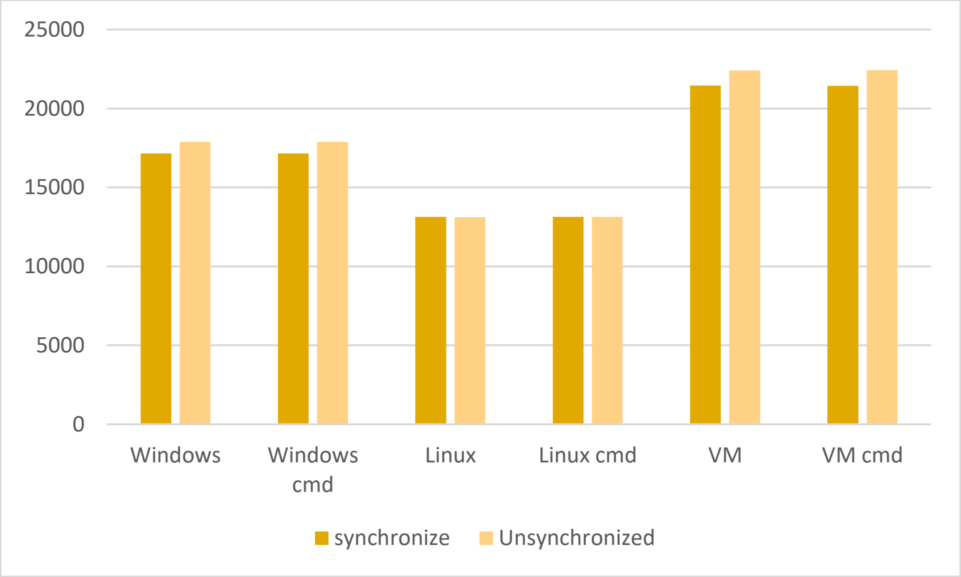
```
[manarjber@localhost codes]$ java os.java
Final counter value is 2030625
Expected counter value is 2030625
Time taken : 21442
[manarjber@localhost codes]$
```

What is happening?

We have noticed that when the process is stopped synchronization .The result was different from the expected output, and this is due to the absence of the synchronization process, which makes each porridge perform the required arithmetic operation, so after adding the synchronization part to the code, the results began to appear correctly

Data:

	Windows	Windows cmd	Linux	Linux cmd	VM	VM cmd
synchronize	17154	17153.22	13145	13143	21442	21440
Unsynchronized	17889	17887	13126	13128	22412	22415



Note:

Method = 11923748 + 11924677 = 23,848,425 % 3 = 0