

## Lab 1 - Basic Concurrency in Java

- Group X
- Lastname, Firstname and Lastname, Firstname

### Task 1: Creating and joining threads

Source files:

- `task1/Main.java` (main file)

To compile and execute:

```
javac Main.java  
java Main
```

### Task 2: Simple Synchronization

#### Task 2a: Race conditions

Source files:

- `task2/MainA.java` (main file)

To compile and execute:

```
javac MainA.java  
java Main
```

#### Task 2b: Synchronized keyword

Source files:

- `task2/MainB.java` (main file)

To compile and execute:

```
javac MainB.java  
java Main
```

#### Task 2c: Synchronization performance

Source files:

- `task2/MainC.java` (main file)

To compile and execute:

```
javac MainC.java  
java Main <N>
```

Where N is number of threads to execute with.

In figure 1, we see how the execution time scaled with the number of threads ...

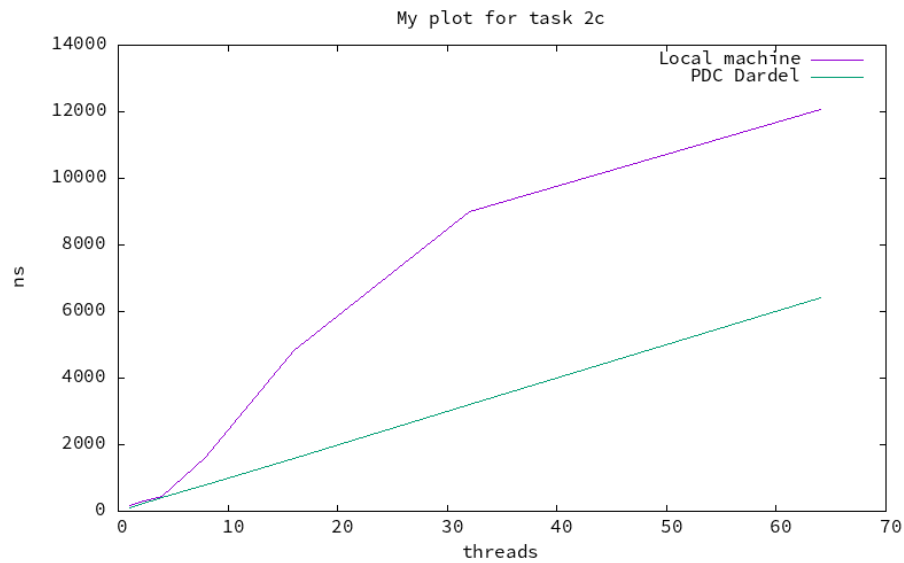


Figure 1: My plot for task 2c

**Task 3: Guarded blocks using wait()/notify()**

**Task 4: Producer-Consumer Buffer using Condition Variables**

Source files:

- task4/Main.java: main file
- task4/Buffer.java: my producer-consumer buffer

**Task 5: Counting Semaphore**

**Task 6: Dining Philosophers**

**Task 7: Deadlock Debugging**