Ahmad Hakam Alhafi

139623

Section 2

Using Semaphore:

**I made 2 classes, Computation which implements Runnable interface, and the Main class which includes the main method.**

**In the Computation, I created the run() function and wrote my code in it, and put s.acquire(); in the start of my code, and end it with s.release();**

**In the Main, I created 3 threads and set their names to (thread: 1, thread: 2 and thread: 3) and start them. And I made a new semaphore, and its name is s, and I gave it the value 1, which means that just one thread can access the resources at a time, and when it finishes its work, another thread starts its own work, and so on.**

A screenshot of a computer

Description automatically generated with medium confidence

**The new 3 lines after running the code.**

Text

Description automatically generated

**The output :), You can try another cases like making 2 or three keys.**

Using Mutex lock:

**I made 2 classes, Computation which implements Runnable interface, and the Main class which includes the main method.**

**In the Computation, I created the run() function and wrote my code in it, and put m.acquire(); in the start of my code, and end it with m.release();**

**In the Main, I created 3 threads and set their names to (thread: 1, thread: 2 and thread: 3) and start them. And I made a new mutex, and its name is m, mutex lock give the access to the resources for one thread at a time, and when it finishes its work, another thread starts its own work, and so on.**

A screenshot of a computer

Description automatically generated with medium confidence

**The new 3 lines after running the code.**

Text

Description automatically generated

**The output :)**