Faculty of computers and Information
Cairo University
Fall 2019

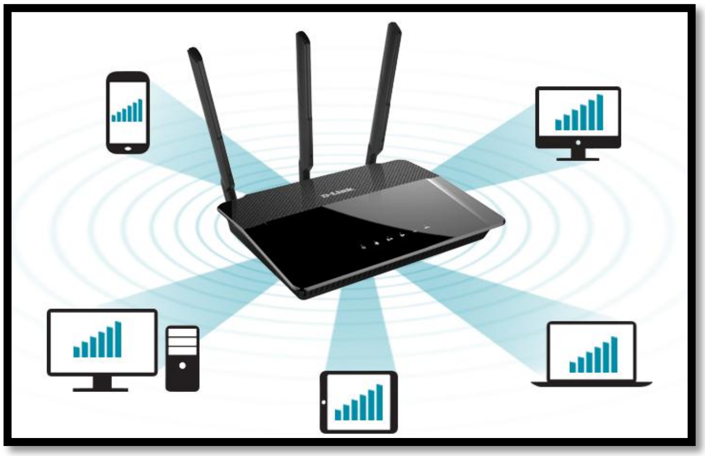
CS241: Operating System – 1
Assignment 2: Java Synchronization

Deadline & Submission:

1. The Assignment is group of 3 Maximum.
2. At least one team member should submit the compressed group solution as zip file containing the program under Acadox => tasks (name your assignment file "Assignment_2_ID1_ID2_G#_G#.zip").
   e.g. Assignment_2 _20168383_201638838_G1_G2.zip
3. The deadline for submitting the solution is 9 Nov. 2019 @ 11:59 PM.
4. Cheating could lead to serious consequences.
5. No submissions after deadline.

Problem description:

It is required to simulate a limited number of devices connected to a router's Wi-Fi using Java threading and semaphore. Routers can be designed to limit the number of open connections. For example, a Router may wish to have only *N* connections at any point in time. As soon as *N* connections are made, the Router will not accept other incoming connection until an existing connection is released. Explain how semaphores can be used by a Router to limit the number of concurrent connections.



Consider the following rules:

- The Wi-Fi number of connected devices is initially empty.
- If a client is logged in (print a message that a client has logged in) and if it can be served (means that it can reach the internet), then the client should perform the following activities:
  1. Connect
  2. Perform online activity
  3. Log out

  Note: these actions will be represented by printed messages, such that there is a random waiting time between the printed messages when a client connects, do some online activities and logged out.
- If a client arrives and all connections are occupied, it must wait until one of the currently available clients finishes his service and leave.
- After a client finishes his service, he leave and one of the waiting clients (if exist) will connect to the internet.

Problem Design

You program must contain the following classes:

1. **Router Class:** that contains a list of connection and methods to occupy a connection and release a connection.
2. **Semaphore Class:** as given the synchronization lab.
3. **Device Class:** represent different devices (threads) that can be connected to the router; each device has its own name (i.e. C1) and type (i.e. mobile, pc, tablet...) and it may perform three activities: connect, perform online activity and disconnect/logout.
4. **Network Class:** this class contains the main method in which the user is asked for two inputs:
   - ***N***: max number of connections a router can accept
   - ***TC***: total number of devices that wishes to connect).
   - ***TC lines that contain***: name of each device, and its type.

Program Output:

You will print the output logs in a file, which simulates the execution order of the devices threads and the printed messages of each device.

Example:

Sample Input:

```
What is number of WI-FI Connections?
2
What is number of devices Clients want to connect?
4
C1 mobile
C2 tablet
C3 pc
C4 pc
```

Sample output: (Note: output depends on the order of the executions of the threads)

```
- (C1)(mobile)arrived
- (C2)(tablet)arrived
- Connection 1: C1 Occupied
- Connection 2: C2 Occupied
- C4(pc) arrived and waiting
- C3(pc)arrived and waiting
- Connection 1: C1 performs online activity
- Connection 2: C2 performs online activity
- Connection 1: C1 Logged out
- Connection 1 : C4 Occupied
- Connection 1 : C4 performs online activity
- Connection 2: C2 Logged out
- Connection 2: C3 Occupied
```

Grading criteria:

Router Class	**15**
Semaphore Class	**5**
Device Class	**10**
Network Class	**10**
Output valid order (Code Run Properly)	**20**
GUI - Bonus (That shows the behavior of connections when occupied or released by a particular device)	**10**