Assignment 3

Operating System

Synchronization
semaphores

By:

MHD Maher Azkoul
438017578
s438017578@st.uqu.edu.sa

Anas Nawawi
438008655
s438008655@st.uqu.edu.sa

course: operating systems, class: 3, no: 24

Professor:

pr. AbdulBaset Gaddah

Umm Al-Qura University
College of Computers and Information Systems
Operating Systems Course – 14012203 - 1441

Dec 15, 2019

Github Link

Umm Al-Qura University
Operating Systems – 1441
Assignment 3: synchronization
Date: Sun - 15 Dec 2019
Pr. AbdulBaset Gaddah

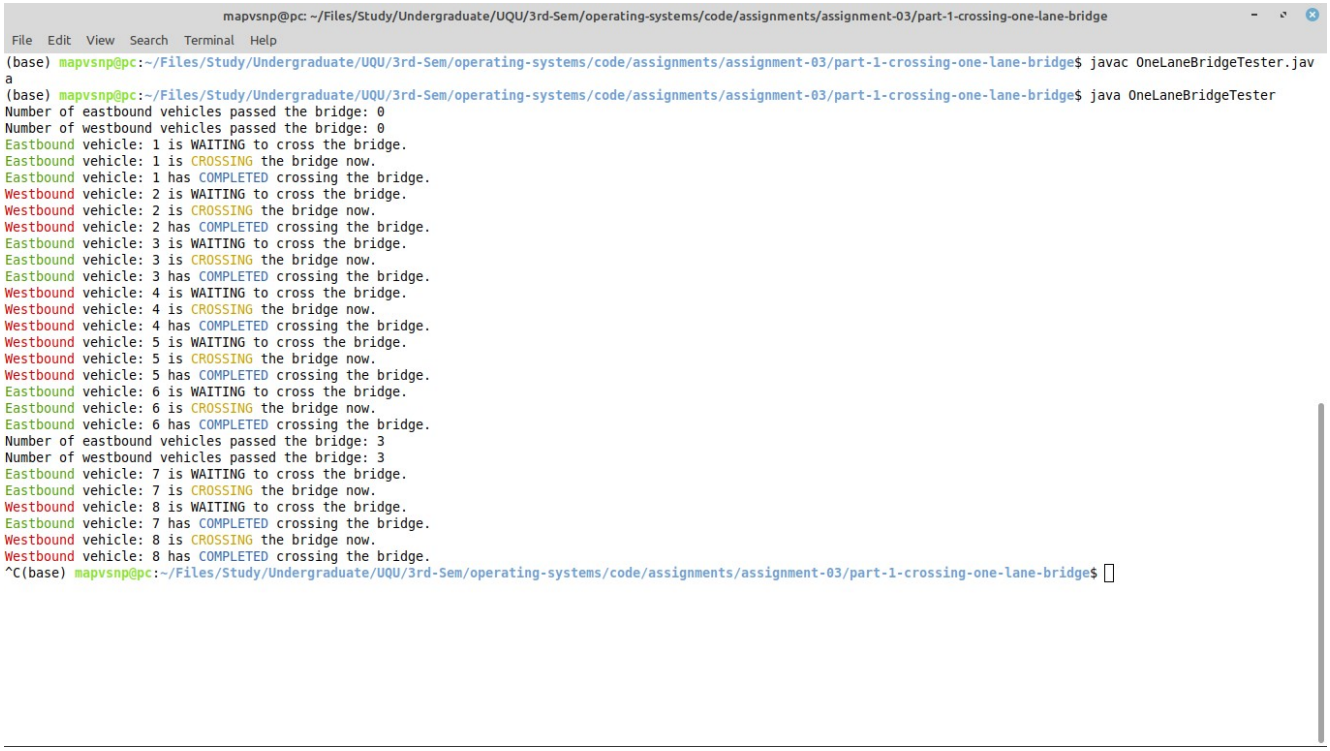
Student: MHD Maher Azkuol
ID: 438017578
Student: Anas Nawawi
ID: 438008655
class: 3

Part I - One-Lane Bridge v1:

Use Java Semaphores to simulate one-lane-bridge:

This program simulates a traffic scenario where we need a semaphores to control the traffic. The program simulates two bounds connected with a bridge that have only one lane

Output Screenshot:



Umm Al-Qura University
Operating Systems – 1441
Assignment 3: synchronization
Date: Sun - 15 Dec 2019
Pr. AbdulBaset Gaddah

Student: MHD Maher Azkuol
ID: 438017578
Student: Anas Nawawi
ID: 438008655
class: 3

Explanation:

The whole program is divided to parts:

- **one-lane-bridge-tester:** run each part of the simulation and connect them together.
- **bridge:** simulates the bridge that connect the two bounds and only one car can cross at the same time.
- **bounds(east and west):** simulates the bounds that vehicles come a cross in order to arrive to the bridge, simulate vehicles by creating vehicles threads.
- **vehicles**: simulates vehicles that cross the bridge.

First of all, the one-lane-bridge-tester class creates the bridge and the two bounds. After that, each bound starts to create vehicles to simulates that vehicles are coming. Each vehicle simulates crossing the bound by waiting for a while, and then arrive to the bridge. The bridge has a semaphore to manage the vehicles to prevent deadlock and to let vehicles cross the bridge with the mutual exclusion technique.

Umm Al-Qura University
Operating Systems – 1441
Assignment 3: synchronization
Date: Sun - 15 Dec 2019
Pr. AbdulBaset Gaddah

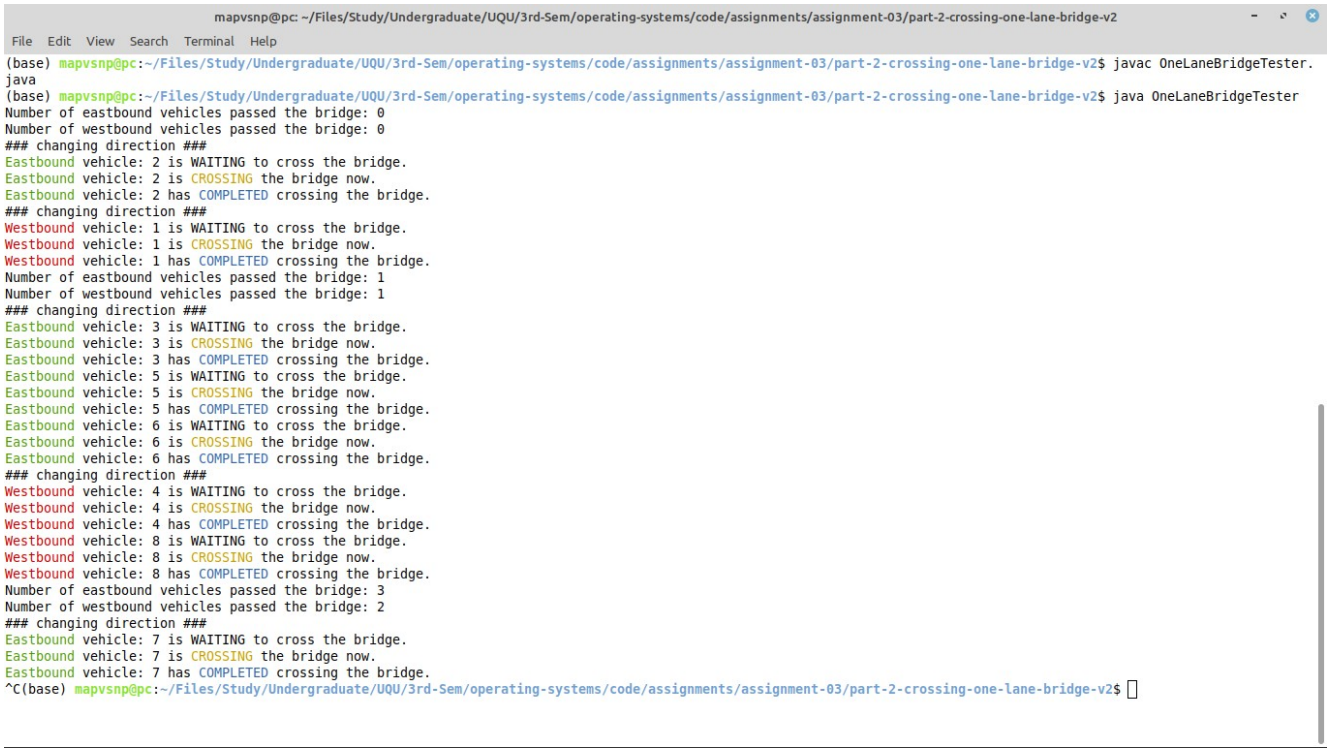
Student: MHD Maher Azkuol
ID: 438017578
Student: Anas Nawawi
ID: 438008655
class: 3

Part II - One-Lane Bridge v2:

Modify one-lane-bridge to simulate two traffic lights next to each bound.

This program is a modified version of the first part of the assignment (one lane bridge) but simulates two traffic lights instead of using one semaphore

Output Screenshot:



Explanation:

The whole program is divided to parts:

- one-lane-bridge-tester: run each part of the simulation and connect them together.
- bridge: simulates the bridge that connect the two bounds and only one car can cross at the same time. The bridge has a traffic light at each side of it.
- bounds(east and west): simulates the bounds that vehicles come a cross in order to arrive to the bridge, simulate vehicles by creating vehicles threads.
- vehicles: simulates vehicles that cross the bridge.

First of all, the one-lane-bridge-tester class creates the bridge and the two bounds. After that, each bound starts to create vehicles to simulates that vehicles are coming. Each vehicle simulates crossing the bound by waiting for a while, and then arrive to the bridge. The bridge has a two semaphorea to manage the vehicles to prevent deadlock and to let vehicles cross the bridge with the mutual exclusion technique. The addition to this program is that the bridge let vehicles of a direction to cross the bridge for some amount of time, and then change to the other direction (just like the traffic lights).

Umm Al-Qura University
Operating Systems – 1441
Assignment 3: synchronization
Date: Sun - 15 Dec 2019
Pr. AbdulBaset Gaddah

Student: MHD Maher Azkuol
ID: 438017578
Student: Anas Nawawi
ID: 438008655
class: 3

Part III - Cigarette Smokers:

Write your solution for the classical synchronization problem (cigarette smokers).

The smokers require three ingredients in order to smoke a cigarette: tobacco, paper, and matches. Each smoker has one of the three ingredients and waits for the other two, smokes the cigar once he obtains all ingredients, and repeats this forever. The agent has an infinite supply of all three ingredients. The agent repeatedly places two of the ingredients on the table at a time.

Output Screenshot:

back to content ^

Explanation:

The whole program is divided to parts:

- cigarette smokers test: run each part of the simulation and connect them together.
- table: simulates the table that the agent will put on it and smokers will pick up things from it, so it should be synchronized.
- smokers: simulates the smokers that wait the agent to supply the by complimentary ingredients so they can smoke.
- agent: provide ingredients for the smokers and put it on the table.

First of all, the cigarette smokers test class will create the agent and the table and the smokers, each smoker has an ingredient, and then the agent starts by putting randomly selected ingredients on the table and let the complimentary smoker smokes by releasing its semaphore, then the smoker starts to smoke and the agent stop while the smoker is smoking because he acquire the agent's semaphore. After the smoker finishes smoking, he release the agent and the agent starts its work again. This process will repeat for defined number of times. After that, each smoker print out the number of cigarettes he had smoked.