OS lab 3

Synchronization and Mutual Exclusion

++++++++++++++++

Made By:

Mohamed Ramadan Mohamed

57

Outline:

- Problem Definition.
- Code Organization.
- Main Problems.
- Mutex and condition variables used.
- Other variables.
- How to run the program.

Problem Definition:

- Complete three function using pthread mutex to simulate a train station boarding and loading.

Code Organization:

- Main Functions:

  1- *station_load_train():* this function is called whenever a train enters the station. An assumption is made that one train enters the station at once. When a train enters the station the waiting passengers enters it until it's full or the station is empty. If a passenger arrived and the train is still loading it also enters the train.

  2- *station_wait_for_train():* whenever a passenger enters the station it calls this function. If an available train exists, the passenger checks if it has available seats and then enters it. Else the passenger waits.

3- *station_on_board():* this is called when the passenger is seated into the train.

Main Problems:

there are multiple synchronization problems here:

1: the load_train and wait for train checking for and modifying the current passengers counter simultaneously.

2: the passengers comes when the train is still leaving. Which makes the program enters a deadlock.

3: the train leaving before the passenger finishes *station_on_board()*.

Mutex and condition variables used:

-*global_mutex:* the global mutex for the three function. One mutex is used because all the three functions uses *passengers_count* and *ava_seat* and *boarding_count* so only one program can check or modify them.

-*passenger_cond:* a condition variable used when the train is waiting passengers to ride the train. When this is signaled from the train any waiting passenger will be notified and the adequate amount will take the train.

-*train_cond:* whenever a passenger boards onto the train it signals this condition so the train checks if it's okay to leave the station.

Other variables:

-*passenger_count:* this represents the amount of passengers that already on the station or boarding the train.

-*ava_seat:* empty seats in the current train.

-*boarding_count:* passengers boarding the train but aren't on the train yet.

How to run the program:

this program runs only in unix-like systems because of system calls. So in it's assumed that it will run on a unix-like system.

1- change directory to the program folder “OSLAB3” using 'CD'.
2- type 'make' this will compile the code and make an executable file.
3 – type `caltrain` and the program will run.