

CSE-344 SYSTEM PROGRAMMING MIDTERM REPORT

First I want to show the normal termination which does not show deadlock scenario.

```
@ubuntu: ~/storage/courses/6th_term/sys_prog/assignments/midterm/program (00.59)
Cook 0 placed dessert on the counter ----- counter items P:1, C:1, D:1 = 3
Student 3 sat at table 4 to eat (round 1) - empty tables:3
Cook 0 is going to the kitchen to wait for/get a plate --- kitchen items P:3, C:3, D:3 = 9
Student 3 left table 4 to eat again (round 1) - empty tables:4
Cook 2 is going to the kitchen to wait for/get a plate --- kitchen items P:3, C:3, D:3 = 9
Student 2 got food and is going to get a table (round 3) - # of empty tables: 4
Cook 2 is going to counter to deliver soup ----- counter items P:0, C:0, D:0 = 0
Student 2 sat at table 4 to eat (round 3) - empty tables:3
Cook 2 placed soup on the counter ----- counter items P:1, C:0, D:0 = 1
Student 2 left table 4 to eat again (round 3) - empty tables:4
Cook 2 is going to the kitchen to wait for/get a plate --- kitchen items P:2, C:3, D:3 = 8
Student 2 is done eating L=3 times - going home - GOODBYE!!!
Cook 2 is going to counter to deliver dessert ----- counter items P:1, C:1, D:0 = 2
Cook 2 placed dessert on the counter ----- counter items P:1, C:1, D:1 = 3
Cook 2 is going to the kitchen to wait for/get a plate --- kitchen items P:2, C:2, D:2 = 6
Cook 1 is going to counter to deliver main course ----- counter items P:1, C:0, D:0 = 1
Cook 1 placed main course on the counter ----- counter items P:1, C:1, D:0 = 2
Cook 1 is going to the kitchen to wait for/get a plate --- kitchen items P:2, C:2, D:2 = 6
Cook 0 is going to the kitchen to wait for/get a plate --- kitchen items P:2, C:2, D:2 = 6
Student 4 got food and is going to get a table (round 3) - # of empty tables: 4
Student 4 sat at table 4 to eat (round 3) - empty tables:3
Cook 2 is going to counter to deliver soup ----- counter items P:0, C:0, D:0 = 0
Student 4 left table 4 to eat again (round 3) - empty tables:4
Student 4 is done eating L=3 times - going home - GOODBYE!!!
Cook 2 placed soup on the counter ----- counter items P:1, C:0, D:0 = 1
Cook 2 is going to the kitchen to wait for/get a plate --- kitchen items P:1, C:1, D:2 = 4
Student 3 is going to the counter (round 2)
--> # of students at counter: 5 and counter items P:1, C:1, D:0 = 2
Cook 0 is going to counter to deliver main course ----- counter items P:1, C:0, D:0 = 1
Cook 0 placed main course on the counter ----- counter items P:1, C:1, D:0 = 2
Cook 1 is going to counter to deliver dessert ----- counter items P:1, C:1, D:0 = 2
Cook 0 is going to the kitchen to wait for/get a plate --- kitchen items P:1, C:1, D:1 = 3
Cook 1 placed dessert on the counter ----- counter items P:1, C:1, D:1 = 3
Cook 1 is going to the kitchen to wait for/get a plate --- kitchen items P:1, C:1, D:1 = 3
Cook 1 is going to counter to deliver soup ----- counter items P:0, C:0, D:0 = 0
Cook 1 placed soup on the counter ----- counter items P:1, C:0, D:0 = 1
Cook 1 is going to the kitchen to wait for/get a plate --- kitchen items P:0, C:1, D:1 = 2
Cook 1 is going to counter to deliver main course ----- counter items P:1, C:0, D:0 = 1
Cook 1 placed main course on the counter ----- counter items P:2, C:2, D:1 = 5
Cook 2 is going to the kitchen to wait for/get a plate --- kitchen items P:0, C:0, D:1 = 1
Cook 2 is going to counter to deliver dessert ----- counter items P:2, C:2, D:1 = 5
Cook 1 has finished serving ----- items at kitchen : 0 - going home - GOODBYE!!!
Cook 2 placed dessert on the counter ----- counter items P:2, C:2, D:2 = 6
Cook 2 has finished serving ----- items at kitchen : 0 - going home - GOODBYE!!!
Cook 0 has finished serving ----- items at kitchen : 0 - going home - GOODBYE!!!
Student 3 got food and is going to get a table (round 2) - # of empty tables: 4
Student 3 sat at table 4 to eat (round 2) - empty tables:3
Student 3 left table 4 to eat again (round 2) - empty tables:4
Student 3 is going to the counter (round 3)
--> # of students at counter: 5 and counter items P:1, C:1, D:1 = 1
Student 3 got food and is going to get a table (round 3) - # of empty tables: 4
Student 3 sat at table 4 to eat (round 3) - empty tables:3
Student 3 left table 4 to eat again (round 3) - empty tables:4
Student 3 is done eating L=3 times - going home - GOODBYE!!!
husnu@ubuntu:~/storage/courses/6th_term/sys_prog/assignments/midterm/program (00.59)$
```

Kitchen design

There are 3 int variable to keep count of different places.

Counter design

To be able to accomplish not to let students take plates individually and to be sure that the placement of plates on the counter efficiently to be taken by students, the 'tray' structure is used. In this way trays are filled first then put on counter.

General program flow

- Supplier gets the plates from truck and then according to the type of plate, increment corresponding int variables which are located in a shared memory.
- Cooks firstly fill the current tray and place it to counter. To fill the tray they are going for P, C, D to kitchen and take next necessary plate. In case of counter is full cooks wait for an availability with with the plate in their hands.
- Students wait for trays and take it open room on the counter then go to eat their meal in an empty table. This repeat L times.
- After supplier done delivery, cooks service and students eat they leave premises.
- After processes terminate, semaphores are destroyed, shared memory unlinked.
- In case of SIGINT termination, an information message also is shown.

Notes:

- First I was planning to to Bonus part too, but I could not handle it, so my command line input according to bonus part but the logic is like just one type of student is around.
- Also there are some issues that causes deadlock, but in some scenario there is no deadlock.
- The thing that I did not mentioned is applied according to the instruction text as far as I am able to do.