Operating System

Lab Assignment 3

Synchronization and Mutual Exclusion

Name : Mohamed Samir Shabaan

ID : 55

code organization :

- the code consist from three parts:
reaction.h : have the declaration for struct reaction and other function which we use in reaction.c file.

reaction-runner.c : to test the reaction.c .

reaction.c : which have three main fuctions
> reaction_init (struct reaction *reaction) : to initialize number of h atoms (make it equal to 0 in the begin of program) and initialize one mutex and two variable condition.

> reaction_h(struct reaction *reaction) : call by H-threads and using to increase the number of atom and send a signal to o-threads then wait to receive a signal then unlock the mutex.

> reaction_o(struct reaction *reaction): call by O-threads and wait until receive two h atoms (two signal) and then call make_water() function to create H2O then send two signal to two thread that sleeps to weak up each of them and the unlock the mutex.

2 – the various mutex and condition variables used:

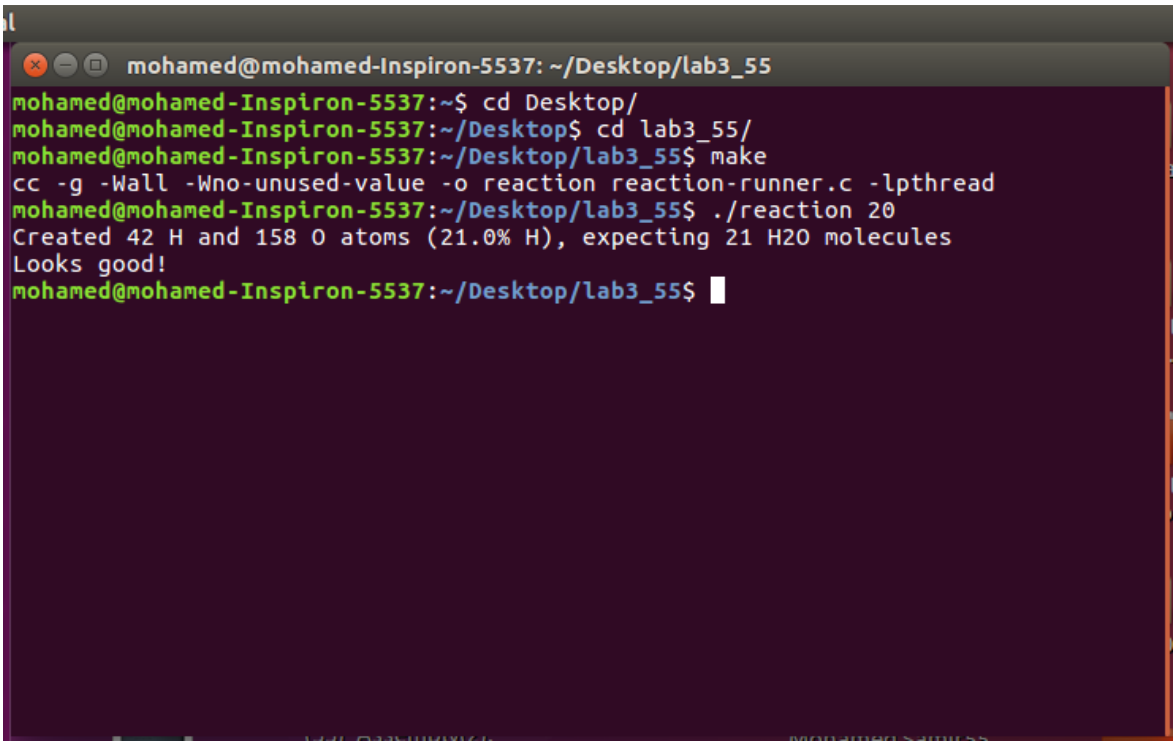
used one mutex two times in the code
first time to lock and unlock in the reaction_h function, the last time in reaction_o to lock and unlock.
Used two condition variables one to notify we received h atom and signal the sleep thread (waited thread on condition) that we have h atom and another one to notify that it done (we waited until have 2 h and receive one o) and signal the sleep thread (waited thread on condition) and signal the two thread that the make_water is done.

how to compile and run the code :

- Open the terminal and go to the project directory .
- Write make in terminal .
-test it by using ./reaction x where x percentage of hydrogen .
- can use the test in make file by write (make run) in terminal.

Sample runs :

- how to run and using ./reaction x



- run using write (make run) in terminal.