Birzeit University - Faculty of Information Technology
Electrical & Computer Engineering Department - ENCS4330
Real-Time Applications & Embedded Systems - 1^{st} semester - 2021/22

Project #2
Interprocess communication techniques under Linux
Due: November 1, 2021

Instructor: Dr. Hanna Bullata

Palestine => Jordan crossing border management

We would like to create a multi-processing application that simulates the daily operation of the crossing border from Palestine to Jordan. The case can be described as follows:

- Passengers arrive randomly at the crossing border. Most of them are Palestinians and Jordanians. However, some of them might be foreigners. The percentage of Palestinian, Jordanian or foreign passengers should be user-defined.
- To cross the border, passengers need to have their papers checked first at crossing points by Palestinian officers. Most of these crossing points are for Palestinian and Jordanian passengers only while the remaining ones are only for foreigners.
- If passengers have their passports expired or do not have them (damn it!), they are denied crossing.
- The officers at the crossing points do not process passengers at the same rate. Passengers who wait a lot at these crossing points get impatient and might decide to return home if they are not handled within a time limit.
- Every time P passengers have crossed the crossing points, they are grouped in a hall waiting for buses to transfer them to the Jordanian side. Each bus can transfer a maximum number of passengers per trip (e.g. 50). The maximum number of passengers that buses can transfer in each trip should be user-defined.
- When the number of passengers in the hall has exceeded a certain threshold H_{max}, the crossing points stop handling passengers until the number of passengers in the hall drops below a threshold H_{min}. In the meantime, passengers continue to add to the queues of the crossing points.
- Assume that there are B buses that will handle passengers' transfer to the Jordanian side. Once a bus leaves the Palestinian side to the Jordanian side, it becomes unavailable for Tb_i seconds before it comes back to the Palestinian side.
- Assume that the passengers get out of our system once they transferred to the Jordanian side.
- The simulation should end if any of the following is true:
  - More than P_g passengers have been granted access at crossing points.
  - More than P_d passengers have been denied access at crossing points.
  - More than P_u passengers got too impatient and decide to cancel their trip to Jordan and return home.

Summary of symbols seen above

Symbol	Description
P	Number of passengers.
O	Number of Palestinian officers.
O_i	Officer i handling crossing point number i.
p_p, p_j, p_f	Percentage of Palestinian, Jordanian and foreign passengers respectively.
B	Number of buses.
Tb_i	Time required by bus i to transfer passengers from the Palestinian side to the Jordanian side and come back.
H_{min}, H_{max}	Max threshold and min threshold.
P_g	Number of passengers who have been granted access.
P_d	Number of passengers who have been denied access.
P_u	Number of passengers who get impatient and leave back home.
P_i	Patience level for passenger i above which he/she decides to return home.

What you should do

- Write the code for the above-described application using a multi-processing approach.
- Check that your program is bug-free. Use the `gdb` debugger in case you are having problems during writing the code (and most probably you will :-). In such a case, compile your code using the `-g` option of the `gcc`.
- In order to avoid hard-coding values in your programs, think of creating a text file that contains all the values that should be user-defined and give the file name as an argument to the main program. That will spare you from having to change your code permanently and re-compile. As a preliminary list, consider putting the fields described above (e.g. P, O, B, etc). Use and complete the following list:

  ```
  NUMBER_CROSSING_POINTS_P 5
  NUMBER_CROSSING_POINTS_J 3
  NUMBER_CROSSING_POINTS_F 1
  NUMBER_OFFICERS 9
  NUMBER_BUSES 4
  BUS_SLEEP_PERIOD 1 10
  ```

  The last line means that buses sleeping period will be in the interval of 1 - 10 seconds.
- If you are familiar with building graphical primitives under Linux, it will be great if you can do it to simulate the behavior of the border visually. The primitives can be as simple as boxes for passengers, triangles (for officers), circles (for buses) and the like. Otherwise, you need to use lots of `printf`!

Have fun and good luck.