

**Project #3**  
**POSIX threads under Unix/Linux**  
**Due: May 28, 2024**

**Instructor:** Dr. Hanna Bullata

## Pharmaceutical medicine inspection & packaging Simulation

We would like to build a combined multi-processing and multi-threading application that simulates the behavior of pharmaceutical medicine inspection & packaging employees. As their name indicates, inspection & packaging employees are responsible to first make sure the medicine has been produced against the proper standard and then the same employees are responsible to place the medicine in packages and then place the drug prescription in the same package. The system can be described as follows:

- Assume the pharmaceutical factory produces a user-defined number of different medicines using a user-defined number of production lines. Some of the production lines produce liquid medicine while other lines produce pill-based medicine.

Assume the production operation on these lines is mostly automated so only the inspection & packaging part is done manually.

- If the produced medicine is liquid, part of the inspection involves checking visually on the following:
  - Liquid level inside bottles is within the normal range,
  - Liquid color matches expected color,
  - Medicine bottles are properly sealed,
  - Correct label is properly placed on the bottle,
  - Medicine expiry date is clearly printed on the bottle and is correct.
- If the produced medicine is made up of pills, part of the inspection involves checking visually on the following:
  - No plastic container is missing any pill,
  - Pills in the plastic containers have the correct color and size,
  - Medicine expiry date is clearly printed on the plastic container back.

Assume all the above failures are random but must belong to user-defined ranges.

- If the production line relates to bottled medicine, the inspection & packaging employees must place the medicine in the bottle containers and add the folded prescriptions to the packages. Otherwise, if the production line relates to pill-based medicine, the employees must make sure to put the right number of plastic containers in the packages in addition to the folded prescriptions (single prescription per package).
- The medicine production lines do not all function at the same speed. When the need arises, some employees are moved between production lines to help boost inspection & packaging operations.
- The simulation ends if any of the following is true:
  - The pharmaceutical factory has produced a medicine type above a user-defined threshold.

- Too many bottled medicine are out of spec.
- Too many pill-based medicine are out of spec.
- The simulation has been running for more than a user-defined amount of time (in minutes).

## What you should do

- Implement the above problem on your Linux machines using a combined multi-processing and multi-threading approach. The production lines are processes and the inspection & packaging employees are threads that belong to a particular production line.
- Think clearly how you would have inspection & packaging employees switch from a production line to another and then return to their original production line.
- Compile and test your program.
- 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.
- Use graphics elements from `opengl` library in order to best illustrate the application. Nothing fancy, just simple and elegant elements are enough.
- Be realistic in the choices that you make!
- Send the zipped folder that contains your source code and your executable before the deadline. If the deadline is reached and you are still having problems with your code, just send it as is!