IN.5022 — Concurrent and Distributed Computing

Series 2

Due date: 5.10.2023, 12:00, on Moodle

Exercise 1

A coffee machine accepts only 10 cents and 1 franc. A customer has to pay 2.50 francs to buy a coffee from the machine. Specify the behaviour of the machine in pseudocode using guarded actions. Assume that the machine rejects any coin other than 10 cents and 1 franc, since it cannot recognize those coins. Also, assume that the machine will not issue a refund if a customer deposits excess money.

Exercise 2

Alice, Bob, and Carol are the members of a library that contains, among many other books, single copies of the books *A*, *B*. Each member can check out at most two books, and the members promise to return the books within a bounded period. Alice periodically checks out the book *A*, and Bob periodically checks out the book *B*. Carol wants to use both books at the same time. How will the librarian ensure that Carol receives her books? Write the program for Alice, Bob, Carol, and the librarian by completing the pseudocode below.

```
% Alice
do true → Request A; do ¬A → skip od; use A; release A od

% Bob

% Carol

% Librarian
```

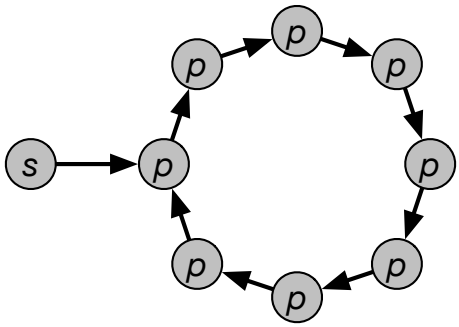
What kind of fairness is needed in your solution?

Exercise 3

Write a program in pseudocode that creates a ring of *N* processes and sends *M* times a token in the ring, as illustrated below.

The following constraints have to be fulfilled:

1. All ring processes have exactly the same code, except the starting process *s*: `start(N,M)` that launches the application.
2. Each time a process *p* receives a token, *p* sends the token to its successor and increments a local counter. It also prints the current value of the counter.
3. When there are no more tokens, each process terminates gracefully. The last value of the counter can be optionally written into a file.



Exercise 4 (optional)

Consult the "official" Erlang documentation available online at https://www.erlang.org/ and read module 2 (sequential programming).

Try to complete the first 3 exercises from https://www.erlang.org/course/exercises.html (excluding the last exercise on the Swedish date).

Exercise 5 (optional)

Install the official Erlang Docker image from https://hub.docker.com/_/erlang/ and launch it by following the instructions from the documentation. You can then experiment with the sample program on Moodle called `pingpong.erl` inside your Docker image. To do so, copy the file in the current directory and start an Erlang shell by typing "`erl`". Within the console, you will have to compile the program with "`c(pingpong).`", after which you can run it by writing "`pingpong:start().`" (do not forget the trailing dot after the commands). Make a screenshot of your console that showcases that you succeeded.