EXERCISE 0: DISTRIBUTED SYSTEMS - A NAÏVE VIEW

Learning Objectives
By the end of this exercise, you should be able to:

- Implement a naïve distributed system.
- Understand distributed locking challenges.

Exercise statement
We want to share an entire variable between N processes (henceforth servers) in accordance with the following restrictions:

1. Servers can only communicate with each other via sockets. Semaphores, shared memory, message queues and other Inter Process Communication mechanisms (IPCs) are expressly prohibited.
2. Through the socket, each server supports only two possible operations (you will have to make one frame for each operation) on the shared variable:
    - **int read(void)** - Returns the value of the shared integer variable.
    - **void update(int)** - Updates the value of the shared integer variable.
3. Each server must contain a *correct value* of the shared variable before using it.
4. Every server will have the following behaviour:

```
for (i=0; i<10; i++){
...
value = getCurrentValue();
...
updateCurrentValue(valor+1);
...
sleep(1000);
...
}
```

or

```
for (i=0; i<10; i++){
...
value = getCurrentValue();
...
sleep(1000);
...
}
```

The behavior of each server will be selected at the beginning of everything and will not be changed during runtime.

5. More frame types can be added as long as they do not directly affect the whole variable (e.g. you cannot design a frame that increments the current value + k units, or one that indicates the server is a read-only type).

6. The integer variable can be assumed to start initialized to 0.

7. All processes will start at the same time. It is allowed to be controlled manually (via keyboard input) that all servers are ready.

In addition to submitting the code, you must answer the following questions:

1. Explain how your algorithm works [Diagrams are welcome, you know how much I love those].

2. What limitations do you see in your algorithm?

3. How does the performance of your algorithm react when the number of processes is arbitrarily increased?

4. What would need to be changed, or how complicated would it be to allow new servers to be added to the system mid-run?