SYNC - Assignment 5

By Arthur ADAM and Alexandre FLION
Please note that every file mentioned in this PDF is provided in the Canvas Submission.

Part 1 - Baboon Crossing

The baboon crossing problem is a synchronisation problem where baboon must cross a canyon by using a rope. The rope can only contains X baboon at the same time and if 2 baboons are going in 2 different directions, they will fight and die.

A. Variables

Our implementation contains 5 variables:

- A semaphore setted up to X, in order to cap the number of baboon to 5 on the rope
- A mutex to handle the state of the current moving baboon
- A string defining the state of the transition
- 2 Semaphores in order to handle the north baboons and the south baboons

B. Loop

When the thread is launched, the mutex is locked and the thread checks for the state. If the state isn't `EMPTY` or if the state is different than the direction or even if the capacity of the rope is equal to 5, the mutex is signaled and the loop is refreshed using a `continue`.
After checking this, the thread uses the capacity semaphore and makes it wait. Right after, the current state is set to the destination (either `NORTH` or `SOUTH`) and the mutex is signaled.
After the mutex has been signaled, the critical section is called and the capacity semaphore is signaled in order to make the loop continue.

C. Reflection

Our implementation is not a starvation-free implementation. When we tested the program, we saw that one side cross during a long period of time before the crossing side changes.
That gives us a hint that our implementation isn't starvation-free.

Part 2 - Modus Hall With Error

A. What goes wrong? When?

There is a big starvation problem in the given code.

Indeed to change the state of the road you need extreme luck as you first need to have no one left from the other side on the road, and you also need to have one of the new side currently locked on condition variable's wait line (`me.cv.wait()`) to be unlocked by the line releasing the mutex at the end of the thread implementation.

All of this is so unlikely that it is pretty rare to see the road switch state from one side to another, which in the end causes starvation.

PS: While looking around on this problem we found out about a bug in the simulator. It has now been reported here.