Assignment 1: Promela

Exercise 1.1: Verifying the log program

The commented code is attached.

Exercise 1.2: Analysis of algorithm on slide 35

What does the code do?

After passing the initial assert, the two processes will take turns running the critical section. However if one of the processes is preempted between setting `flag[_pid] = 0` and setting `flag[_pid] = 1`, then the other will be able to continuously run the critical section.

How does it work?

The programs uses three components:

- **`flag`**: Each process has a flag. The process sets the flag when ready to run the critical section.
- **`turn`:** Is a variable. The turn variable allows the first process hitting the guard to pass, when the other process requests a turn by setting the variable to its own `_pid`. Without this variable, the system would be susceptible to a deadlock, if both processes sets their flags before passing the guard.
- **Guard:** The guard allows only one process to enter the critical section at a time. When hitting the guard a process will wait for the other process to complete the critical section, or the other to request a turn.

How do you verify that?

To verify that the processes takes turns, a variable, `oldturn`, is introduced, holding the `_pid` of the last process to run the critical section. During the critical process, before setting `oldturn`, it is asserted, that `oldturn` holds the `_pid` of the other process. This solution has some special cases:

- The `oldturn` variable will not have been set on the first run of the critical section.
- If one process has not passed the "`again`" label, the other process will be able to continuously run the critical section.
- If one of the processes is preempted between setting `flag[_pid] = 0` and setting `flag[_pid] = 1`, then the other will be able to continuously run the critical section.

Lasse Brøsted Pedersen 10769
Christian Vraa 201200715