703013-5 PS Operating Systems
Exam – June 27th 2018, 12:15-14:00

General information: Talking, chatting, texting or any other form of live human-human communication is not allowed and may lead to a negative grade. However, you are allowed to use all other offline and online resources such as previous homework solutions, manpages, Stackoverflow, etc.

Note: Please make sure that

- the first line of your source code explains how to compile and run your application (alternatively you can also create a Makefile or shell script),
- your code compiles properly without any errors or warnings,
- your code is well-commented (explaining why e.g. a particular function is called),
- your program performs proper cleanups of any resources it acquires,
- you submit your solution before 14:00 in OLAT (submission will be automatically closed!)

The task below has a total of 10 Points, deviating from the requirements above may reduce the number of points you get. A minimum of 5 points is required for a positive exam grade.

Task

Your task is to implement a logging service with PThreads and mutual exclusion using a single spinlock. Consider the following use case:

- Your program should start 3 service threads that frequently log messages to a common location (e.g. a global variable), and a logger thread that reads from this location and prints the contents on the terminal.
- Each service thread should log exactly 5 messages and then exit.
- The logger thread should print exactly 15 messages and then exit.
- In order to be allowed to log a message, the service threads must first get a "ticket" from the logger for each message. This ticket can be represented by an integer that is increased for each message, giving each message a unique number.
- The 3 service threads should log messages that hold both the number of their ticket and their identity (`web_server`, `database`, and `weather_station`). Note that both the ticket number and the identity must be part of the message sent to the logger. Hence, the output of the logger thread could look as follows:

```
1 web_server
2 database
3 weather_station
...
```

or

```
1 database
2 database
3 web_server
...
```

- Make sure that the program waits for all service threads and the logger thread to finish before exiting.