# Lab 7 Report - Operative Systems

Github @ steinhs

## Exercise 1

a) No.

b) Achieving exactly *Tconcurrent=Tserial/p* probably won't be possible in most programs, if any. Executing a concurrent program will produce overhead when having to create and manage the threads. That includes initializing, finalizing, fetching from libraries and communication between threads. In short, we are creating more resources that the program has to manage.
There is an increase in performance, but we won't see that much of a difference in smaller programs compared to bigger computations.

## Exercise 2

a) Outputs of mutex.c

```
global variable count is: 1000
global variable count is: 1000
global variable count is: -1000
global variable count is: 1000
global variable count is: -1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: -1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: 1000
global variable count is: -1000
global variable count is: -1000
global variable count is: 1000
```

The program consists of two functions which either increases or decreases a number by 1000. These functions are run on two separate threads, using the same integer that is set to zero beforehand. Since the two threads are not communicating with each other in any ways and are using the same integer, they collide. This results in the output of the program being either 1000 or -1000.

b) Outputs AFTER lock/unlock added:

```
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
global variable count is: 0
```

Also remembering to add *mutex_destroy* before exiting program.