

## 4. Assignment

### Process Synchronisation

Issue: May 18—Due: May 25

#### Exercise 6: Using the POSIX semaphore interface

10 Points

There is a deep canyon somewhere in Kruger National Park, South Africa, and a single rope that spans the canyon. Baboons can cross the canyon by swinging hand-over-hand on the rope, but if two baboons going in opposite directions meet on the rope, they will fight and drop to their deaths. Furthermore, the rope is strong enough to hold an arbitrary number of baboons.

Assuming that we can teach the baboons to use semaphores, we would like to design a synchronisation scheme with the following properties:

- Once a baboon has begun to cross, it is guaranteed to get to the other side without running into a baboon going the other way.
- Neither deadlock nor starvation will occur, i.e., even a continuing stream of baboons crossing in one direction should not bar baboons going the other way indefinitely.

**1** Write a C program that implements one such baboon. A single command line argument, being either 'e' or 'w', determines whether the baboon is eastbound or westbound respectively.

Use the POSIX interfaces for semaphores and shared memory to implement your synchronisation scheme: `sem_open(3)`, `sem_close(3)`, `sem_wait(3)`, `sem_post(3)`, and `shm_open(3)`, `mmap(2)`.

Simulate the action of crossing the rope by printing 10 short messages with a delay of one second in between (Listing 1).

```
> ./baboon e
east 10
east 9
east 8
east 7
east 6
east 5
east 4
east 3
east 2
east 1
EAST
```

**2** Document the setup of semaphores and shared variables in your program, and reason why your code fulfils the requirements. Can you give an upper bound for the time a baboon has to wait?

Listing 1: Example of an eastbound baboon.

**Hints** Instead of starting from scratch, you may want to use the examples that we present in this week's tutorial<sup>1</sup>. Don't forget to pass the `-lrt` command line argument to `gcc`, as is described in the above manual pages.

You can test multiple baboons crossing the canyon by running multiple `baboon` processes simultaneously in different terminals.

**Filing** Send one single C source file containing your implementation and documentation attached to a plain text email to Alexander and Stefan<sup>2</sup>. Please have an eye on UTF-8 encoding and UNIX end-of-line conventions. Do not use whitespaces in filenames.

<sup>1</sup><http://www.inf.uni-konstanz.de/dbis/teaching/ss09/os/sync.tar.bz2>

<sup>2</sup>[alexander.holupirek@uni-konstanz.de](mailto:alexander.holupirek@uni-konstanz.de), [stefan.klinger@uni-konstanz.de](mailto:stefan.klinger@uni-konstanz.de)