Text of exercise #1

Matteo Corain S256654

Laboratory #3 – System and device programming – A.Y. 2018-19

Write a C program using Pthreads that tests the sem_trywait system call on a producer and consumer problem using two buffers. In particular, the main thread must create a producer and a consumer thread and must wait for their termination. The producer thread:

- Loops 10000 times sleeping a random number of milliseconds (1-10);
- Fills a variable ms with the current time in milliseconds, using the function:

```
long long current_timestamp() {
    struct timeval te;
    gettimeofday(&te, NULL); // get current time
    long long milliseconds = te.tv_sec*1000LL + te.tv_usec/1000;
    return milliseconds;
}
```

- Selects randomly a buffer, urgent or normal, in which it will put the value of ms; it must select the normal buffer 80% of the times;
- Prints the message: "putting <ms> in buffer <buffer>", where <ms> is the value, and <buffer> is either urgent or normal;
- Puts the value of ms in the selected buffer;
- Signals on a semaphore that something has been produced.

The consumer thread:

- Sleeps 10 milliseconds;
- Waiting that something has been produced;
- Tries to get an ms from the urgent buffer, but if it is empty it can proceed to get it from the normal buffer;
- Prints the value of ms and the buffer identity (urgent or normal) from which it has got this value.