Fanti Andrea 235808

Lab 10 exercise 02

Since we assume arguments cannot be passed to `pthread_create()`, each thread retrieves its ID from a global variable called `cur_id`.
As more than one thread may be trying to get its ID at the same time, the access to `cur_id` is controlled by means of a semaphore.
In particular, after a thread is created it is possible that its scheduled execution begins while another thread is creating threads: this thread may then "steal" the ID of another thread.

To prevent this from happening, the parent before executing `pthread_create` locks a semaphore that is unlocked by the child when it retrieved its ID.

This wasn't done using a mutex because the behaviour of unlocking a mutex from a thread different than the one that locked it is undefined.

The `gentree` vector indicates who is the parent of a certain thread (for example, at position 3 there is stored the ID of the parent thread of the thread with ID 3).

Notice that the first position of the array is never used since IDs start from 1. This approach was adopted to have a more understandable code.