Excercise 2

You can run the following command to execute the program.

```
cd ex2

docker build -t lab2_ex2 .

docker run --rm -ti -v $(pwd)/:/root/lab lab2_ex2 /bin/bash -c " cd
/root/lab; gcc -o ex2 pid.c -lpthread; ./ex2"
```

Regarding the two experimental conditions of ex2, because 4700 pids are too many, I used 10 pids from 0 to 9, and the screenshots are as follows:

```
Sleeping for a while ...
1th thread with PID = 4
Releasing PID!
Sleeping for a while ...
2th thread with PID = 1
Releasing PID!
Sleeping for a while ...
3th thread with PID = 8
Releasing PID!
Sleeping for a while ...
4th thread with PID = 5
Releasing PID!
Sleeping for a while ...
5th thread with PID = 2
Releasing PID!
Sleeping for a while ...
6th thread with PID = 9
Releasing PID!
Sleeping for a while ...
7th thread with PID = 6
Releasing PID!
Sleeping for a while ...
8th thread with PID = 3
Releasing PID!
Sleeping for a while ...
9th thread with PID = 0
Releasing PID!
Sleeping for a while ...
10th thread with PID = 7
Releasing PID!
Sleeping for a while ...
11th thread with PID = 4
```

Here I loop 20 times, because the picture is too long to be intercepted, from the graph can be learned to loop to the 10th time when the pids have been allocated, and from the 11th time to start PID = 4 began to be reallocated.