

TP 9 : Threads and semaphores

1 Review

1. What is the main difference between threads and processes ? Can we use them interchangeably ?
2. Download the program `mailbox.c` from the website. You have seen this example in class. What happens when the value of `MAX` is 10 ? 100 ? 1000 ? 1000000 ? Why does it not work as planned ? Can you fix it ?
Note : Observe that inside the `for` loop, we have only one line of code, i.e. `mails++`; Why still do we have a race condition ?
3. Now, write a program in C with two threads. The first thread displays even integers up to 100, the second the odd. Use semaphores to ensure the correct order of integers.

2 Producer-Consumer

A typical problem in concurrent programming is that of producer and consumer : one thread produces data that the other consumes. We consider the `procon.c` program. Here the producer gets characters from a file and sends them to a shared space with the thread 'consumer' who displays them on the screen. However, in the current version, no synchronization exists in between. Modify the program so that it displays the contents of a file correctly.

3 Binary Semaphores vs. Mutexes

Download the program `mutexvsem.c`. Run it. Do you see any errors ? Now, change the mutexes and have it use semaphores instead. Does it work ?

Can you justify your observation ?

4 Counting semaphores

Implement a logging queue using semaphores. You can use the template given in `login.c`. Your objective is to ensure that the resource (in our case the section enclosed in 'finite resource') is used by at most 12 users at a time. Create 15 threads and use a semaphore to manage the resource.

5 The function of Hénon

We will calculate the orbit of a dynamic system of dimension 2. The function of Hénon is described by the system

$$H_{a,b} = \begin{cases} x_{n+1} = a - by_n - x_n^2 \\ y_{n+1} = x_n. \end{cases}$$

We already have the function from last week. We will modify the program to use threads instead. We will use one thread to calculate the sequence $(x_n)_n$ and another thread for the sequence $(y_n)_n$. Propose a means of synchronization to ensure interleaving of the computations of x_n and y_n . Implement it.

We can plot the function with the command `gnuplot henon.p` after having downloaded the script "henon.p". The file `henon.dat` must be in the same folder as `henon.p`. (You will need `gnuplot` for this).

Observe the graph you get for values $a = 1.4$ et $b = -0.3$.