Synchronization exercise

Daniel Hagimont

Daniel.Hagimont@enseeiht.fr

USTH

March 2018

Objectives

The objective of this exercise is to use POSIX thread management and synchronization primitives in order to implement the producer/consummer example described in the associated lecture.

Instructions

You have to launch 2 threads which will produce and consume iteratively with a random temporization (between 1 and 2 seconds) between each production/consumption. The buffer where you produce and consumme can simply be an array of boolean.

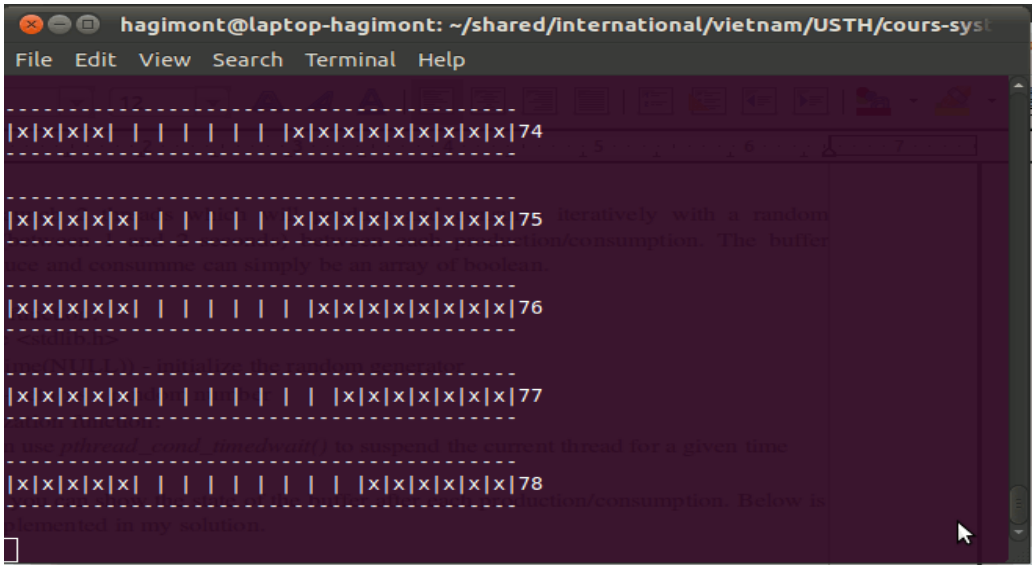
For the random function:

- include <stdlib.h>
- srand(time(NULL)) - initialize the random generator
- rand() - returns a random number

For the temporization function:

- you can use *pthread_cond_timedwait()* to suspend the current thread for a given time

For the display, you can show the state of the buffer after each production/consumption. Below is the display I implemented in my solution.


hagimont@laptop-hagimont: ~/shared/international/vietnam/USTH/cours-syst
File Edit View Search Terminal Help
|x|x|x|x| | | | | | |x|x|x|x|x|x|x|x|x|74
|x|x|x|x|x| | | | | | |x|x|x|x|x|x|x|x|x|75
|x|x|x|x|x| | | | | | |x|x|x|x|x|x|x|x|x|76
|x|x|x|x|x| | | | | | |x|x|x|x|x|x|x|x|x|77
|x|x|x|x|x| | | | | | | |x|x|x|x|x|x|78