Synchronization practicle work

Daniel Hagimont

Daniel.Hagimont@irit.fr

USTH

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.

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.

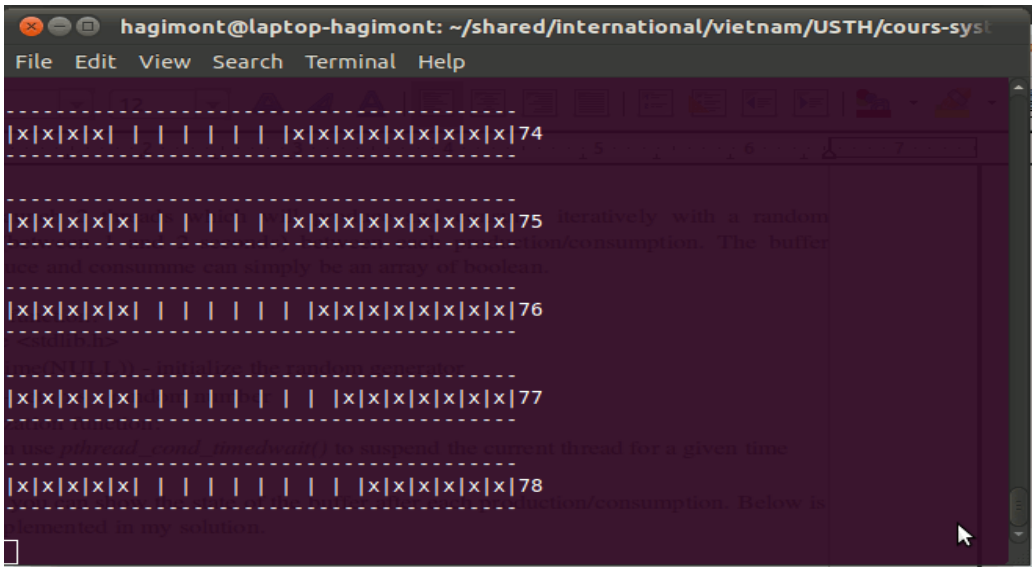
For the random function:

- 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.



Instructions

You are given a template (prodcons-template.c) which implements the following functions :
- init : initialize the application
- display : display the state (content) of the buffer
- sleep : suspend the current thread for a period of 1 to 2 seconds
- produce : the function to add an item in the buffer. **This function lacks synchronization.**
- consume : the function to remove an item from the buffer. **This function lacks synchronization.**
- producer : the function executed by a producer thread
- consumer : the function executed by a consumer thread
- main : the main function of the application. **You should here create a number of producer and consumer thread.**