Sebastian Grzelak

EOPSY LAB5

SLEEPING BARBER PROBLEM

Solution description:

Generally, I have splitted the program for 3 steps. The step description is bellow:

1) The algorithm of sleeping barber is not compliceted. I have created three types of barbers dedicated by constant number at the beginning of program. Additionally I have added another constat which define the number of client generation. All that processes work in infinite while loop and serve the clients (barbers processes) and create the clients (clients generator processes).

First step was to write algorithm of sleeping barber as a multiprocess program without any semaphores. To observe what will happened and what are the reasons of such a situation. Firstly I could observe that the memory is not shared between processes, so that was for me a huge problem which was solved in second step.

2) To solve the problem of memory sharing I decided to added a semaphore. I have created a semaphore which allows me to share the structure of data between processes. Thanks this the variable of clients number was modified by all working processes. Then appear another problem, the clients variable in some case was a negative value. It means that many processes have access to this variable in the same time. That problem was solved in third step.

3) In this step I have created the semaphore(mutex) which allowed me lock and unlock the critical part of code. The cirtical part of code was the place where clients number is modified, so I have to give the access to this code only for one proccess in the same time. The creation of mutex, it is a binary semaphore 0 or 1, which helped me to lock/unlock part of code. Then appeared last problem, mainly to stop proccessess which do not have work in specific time and start them when are necessary.

4) The creation of semaphore for every barber process helped me to stop/start them when it is necessary. The every barber has a unique index and thanks this I could know which barber process I will stop or start.

The algorithm of sleeping barber is not necessary to explain because this is my own invention and it can be seen in my code.