Interprocess Synchronization

Jacques Polart
ing4 – SI – groupe4

Concurrent Access To Shared Memory : Race Problems

1) syncError.c

2) durring the sleep function, another process could change the value of the shared variable i, but the current trhead already stored an old value of i and will based it's calculation on it. It won't take into account the changes of i, and therefore erased these changes.

Solving the Problem : Synchronizing access using semaphores

1) semaphore.c

a) two process or more, once one process hold the ressource, the others have to wait. There is nothing to do more to enforce mutual exclusion.

2) deadlock.c

3) launchApp.c

4) calculation.c
for this one, i used a semaphore to count how many results were stored into the shared array.
Each thread that store a result into the shared array, make a post on this semaphore then.
At fist i wanted to initialize this semaphore to -1, si it would take two post to pass through one wait, but it seem that a post unlock a wait in the queue independently on the counter value.
So i used two wait.

In the thread 4 the shared value is only readed so i didn't use the semaphore dedicated to protec this value.