

TP Programmation répartie

Synchronisation - Barrière

Exercice 1 : Ordonnement de deux processus

Deux threads doivent afficher un message chacun (le message sera différent). Le thread 1 doit afficher son message avant celui du thread 2. Proposez et codez une solution à ce problème. La fonction lancer par les threads sera différente. Le résultat de votre programme doit ressembler à cela :

```
Message 1 du thread 1  
Message 2 du thread 2
```

Exercice 3 : Barrière avec des sémaphores

Proposez une solution au problème de barrière avec un mutex et un sémaphore. Il y aura N threads qui devront utiliser la même fonction thread. Chaque thread affiche un message au début, attend un temps aléatoire (`sleep(rand()%10)`), affiche un message à la barrière et juste avant de se terminer.

Exercice 4 : Barrière avec une variable de condition

Proposez une solution au problème de barrière avec un mutex et une variable de condition. Il y aura N threads qui devront utiliser la même fonction thread. Chaque thread affiche un message au début, attend un temps aléatoire (`sleep(rand()%10)`), affiche un message à la barrière et juste avant de se terminer.