

TP noté

Épreuve du 7 mai 2020
Durée : 2h
Documents autorisés

On souhaite rédiger un programme admettant la syntaxe suivante :

```
prodcons taille nconso prod ... prod
```

Le programme doit implémenter le mécanisme du producteur-consommateur sur un tampon borné de taille *taille* (> 0) avec *nconso* consommateurs (> 0). Les producteurs sont spécifiés par les arguments suivants, sous la forme d'une chaîne de caractères comprenant un nombre n ($n \geq 0$) et un caractère c^1 : le producteur concerné doit écrire n fois le caractère c dans le tampon, puis se terminer. Les consommateurs affichent sur la sortie standard les caractères lus au fur et à mesure. Lorsque tous les producteurs ont terminé et que les consommateurs ont consommé toutes les données en attente, le programme doit naturellement s'arrêter.

Dans l'exemple ci-dessous, on utilise une taille de tampon de 3 caractères, avec 5 consommateurs et 2 producteurs, le premier émettant 10 fois le caractère « x » et le deuxième émettant 15 fois le caractère « = ».

```
> ./prodcons 3 5 10x 15=  
xxxx=x=====x==x=xxx==
```

Votre programme doit prendre en compte les spécifications suivantes :

- le programme doit comporter un thread par consommateur et un thread par producteur
- les variables globales ne sont pas autorisées
- les seuls mécanismes de synchronisation autorisés sont les conditions et les verrous d'exclusion mutuelle POSIX
- un thread ne doit être réveillé que si c'est nécessaire, il ne doit pas y avoir de réveil inutile
- il ne doit bien sûr pas y avoir d'attente active, ni de temporisation
- votre programme doit s'arrêter lorsque tous les producteurs se sont terminés et que toutes les données produites ont été consommées

Vous trouverez sur moodle un squelette de programme pour démarrer.

Bonus

Si vous avez terminé, implémentez une vérification dans le thread principal : chaque thread producteur doit avoir produit le bon nombre de caractères, et le nombre total de caractères produits par les producteurs est identique au nombre total de caractères consommés par les consommateurs. Pour cela, chaque thread doit renvoyer le nombre de caractères traités.

1. On notera que le caractère c ne doit pas être un chiffre.