

EPITA ING1 2011 S1 SYSTÈMES D'EXPLOITATION

Didier Verna

Documents et calculatrice interdits

1 Gestion des Processus (2 points)

Dans un système d'exploitation traditionnel, la transition vers le multithreading pose de nombreux problèmes de (re)conception. Dans Unix par exemple, l'existence de variables globales (comme `errno`) ou le principe des signaux sont typiques des problèmes en question.

1. Expliquez pourquoi `errno` est problématique dans un système multithreadé.
2. Quelle solution les Unix actuels emploient-ils ?

2 Ordonnancement (6 points)

Considérez l'ensemble des processus représentés ci-dessous. On suppose que ces processus sont arrivés dans l'ordre P1...P5, tous à l'instant $t = 0$. On négligera le temps de blocage sur des périphériques.

Processus	Durée de Cycle (ms)	Priorité
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

1. Dessinez quatre diagrammes de Gantt représentant l'exécution de ces processus pour des ordonnanceurs de type FCFS (premier arrivé premier servi), SJF (plus court d'abord), priorité sans réquisition (un niveau de priorité plus faible indique une priorité plus haute) et tourniquet avec une durée de quantum égale à 1ms.
2. Quel est le temps d'attente de chaque processus pour chacun des ordonnanceurs considérés ? Quel algorithme est le plus efficace ? Ce résultat vous semble-t-il cohérent ?

3 Synchronisation (6 points)

Pour aider à la synchronisation, le matériel peut fournir deux instructions atomiques : TaS (Test and Set) et Swap. On rappelle ici leur comportement :

- TaS prend un booléen, le met à vrai, et retourne son ancienne valeur.
- Swap échange simplement les valeurs de deux booléens.

Il est possible d'implémenter une solution d'exclusion mutuelle grâce aux sections d'entrée et de sortie suivantes :

Section d'entrée :

- (1) `lock ← False`
- (2) `while TaS (lock)`
- (3) `Wait`

Section de sortie :

- (1) `lock ← False`

1. Montrer que l'exclusion mutuelle est en effet garantie.
2. Expliquez pourquoi il y a cependant risque de famine.
3. Proposez une version équivalente de cet algorithme utilisant Swap plutôt que TaS.

4 **Pagination** (2 points)

1. Dans un système de gestion mémoire par pagination, expliquez pourquoi on a tout intérêt à choisir des cadres de page dont la taille soit une puissance de deux.
2. Soit une mémoire paginée dont l'espace est adressable sur 16 bits. On suppose que le pagineur définit 64 cadres de page dans cette mémoire. Quel est la taille de chaque cadre de page ?

5 **Fragmentation** (2 points)

Expliquer les notions de fragmentation interne et externe. Quel type de fragmentation se produit-il dans un système de pagination ?

6 **Politicware** (2 points)

Auprès de quel homme politique français va-t-on chercher conseil, quand on a du mal à choisir entre plusieurs algorithmes qui ont chacun leurs avantages et leurs inconvénients ?

Indication : « Mais, heuh ! »