

# EPITA ING1 2012 S1 SEXP

Didier Verna

Documents et calculatrice interdits  
Toute réponse non justifiée sera comptée comme nulle

## 1 Synchronisation (6 points)

En 1981, Peterson propose une solution hybride de synchronisation pour deux processus  $i$  et  $j$ , combinant une variable de tour et un tableau de demande. Les sections d'entrée et de sortie pour le processus  $i$  sont rappelées ci-dessous :

- |     |  |     |                                  |
|-----|--|-----|----------------------------------|
| (1) | demande $[P_i] \leftarrow$ vrai.           | (1) | demande $[P_i] \leftarrow$ faux. |
| (2) | tour $\leftarrow P_j$ .                    |     |                                  |
| (3) | tant que tour = $P_j$ et demande $[P_j]$ : |     |                                  |
| (4) | attendre.                                  |     |                                  |

Dans la solution de Peterson, on remarque (ligne 2 de la section d'entrée) que les processus se comportent poliment : chacun commence par laisser son tour à l'autre. On cherche à établir une variante plus « agressive » de cet algorithme, dans laquelle les processus commenceraient par *essayer* de prendre leur tour au lieu de le céder.

1. Modifier la ligne 2 de la section d'entrée afin de refléter cette idée.
2. Proposer un scénario particulier montrant que ce nouvel algorithme ne fonctionne pas correctement.
3. Proposez une deuxième modification de l'algorithme original
4. Prouvez que votre nouvelle version doublement modifiée fonctionne correctement cette fois-ci.

*Note : merci au glorieux inconnu qui m'a posé cette question cette année ;-)*

## 2 Mémoire Virtuelle (6 points)

Considérez la matrice suivante :

Integer matrix[100][100]

ainsi que les deux routines d'initialisation ci-dessous (implémentées par exemple en C) :

- |     |                             |     |                             |
|-----|-----------------------------|-----|-----------------------------|
| (1) | Integer i, j                | (1) | Integer i, j                |
| (2) | for i from 0 to 99          | (2) | for j from 0 to 99          |
| (3) | for j from 0 to 99          | (3) | for i from 0 to 99          |
| (4) | matrix[i][j] $\leftarrow$ 0 | (4) | matrix[i][j] $\leftarrow$ 0 |

Soit une mémoire paginée disposant de trois cadres de page (hors système). La taille des cadres est  $200 * \text{sizeof}(\text{Integer})$ . On imagine un processus P, chargé d'initialiser la matrice et figurant dans le premier cadre de page. On suppose que les deux autres cadres de page soient initialement vides et que le MMU dispose d'un algorithme de remplacement de pages de type LRU.

1. Rappeler le principe du LRU.
2. Quelle est la caractéristique de cet algorithme, notamment par rapport à l'algorithme optimal ?
3. Cet algorithme fait partie de ce qu'on appelle les « algorithmes de piles ». Quelle est la définition de cette famille d'algorithmes, et quelle est leur propriété fondamentale ?
4. De combien de défauts de page souffre chaque routine d'initialisation ?

### 3 Socioware (2 points)

Pourquoi, quand on va à Montgallet pour acheter une barrette (de RAM), faut-il toujours retirer du liquide avec sa carte bleue d'abord ? Justifiez votre réponse.

### 4 Ordonancement des disques (6 points)

Dans un système de gestion des disques par allocation contigüe, on peut éviter la fragmentation externe en faisant du compactage à chaque fois qu'un fichier est supprimé. À cause de (ou plutôt, grâce à) la contigüité, le temps de lecture (ou d'écriture) d'un fichier comprend le temps de positionnement des têtes, le temps de latence, puis un transfert à pleine vitesse.

1. Dans le cadre de l'ordonancement des disques, rappeler les définitions du temps de positionnement des têtes et du temps de latence.
2. Rappeler en quoi consiste la stratégie de compactage la plus simple.

Nous supposons un temps de positionnement des têtes de 5ms, un temps de latence de 4ms, et un taux de transfert à pleine vitesse de 8Mo/s (les débits et capacités sont exprimées en base 10).

1. Combien de temps prendrait le déplacement d'un fichier de 8Ko ?
2. Combien de temps prendrait le compactage de 8Go de disque ?
3. Pensez-vous que le compactage soit une stratégie réaliste ?