

Aucun document autorisé

Durée : 90 minutes

1 Pages mémoire et blocs disque

Rappelons qu'un disque lit ou écrit des données par blocs complets, un à la fois.

On suppose que la mémoire virtuelle (paginée à la demande) d'une machine utilise des pages (et des cadres de pages) de 4Ko, avec un disque de *swap* qui a des temps d'accès de 10 ms.

Dans le cours, on a supposé que la taille des pages mémoire correspondait à la taille des blocs du disque. On remet ici en cause cette hypothèse, et on s'intéresse aux conséquences en termes de performances du *swapping*.

- Que se passe-t-il si on utilise un disque de swap qui a

1 des blocs de 2 Ko (une page mémoire utilisera 2 blocs)

2 des blocs 8 Ko (en mettant deux blocs par page)

- Dans les deux cas, on peut retrouver des performances “optimales” en consentant un léger sacrifice de matériel. Proposez une solution, évaluez le coût financier (on suppose une mémoire virtuelle de 64 Go, 500 Euros pour un disque de 1024 Go octet de qualité pro).

3

2 Systèmes RAID

On dispose de 3 disques de 1 T octet montés en RAID 5.

Quelle est la capacité du système de disques ainsi constitué ? comment sont répartis les blocs 0,1,2,3,4... ?

4 (Faire un schéma)

Quelles opérations d'entrées-sorties faut-il faire sur les différents disques pour écrire dans le bloc 1 ? Précisez quelles opérations peuvent être menées simultanément.

5

3 Pagination à la demande

On suppose une mémoire paginée avec de 3 cadres de pages. Les pages, numérotées 0,1, ... sont appelées selon la séquence : 0, 1, 2, 3, 4, 1, 2, 4, 0, 1

On étudie l'effet de différents algorithmes de remplacement de page.

3.1 Algorithme FIFO

6 Utilisez le tableau suivant pour montrer le déroulement de cet algorithme

Page	0	1	2	3	4	1	2	4	0	1
Cadre 1										
Cadre 2										
Cadre 3										
Défauts										

Nombre de défauts de page =

3.2 Algorithme LRU

7 Détaillez le sigle "LRU"

8 Expliquez son fonctionnement en une phrase

9 Montrez le déroulement sur le tableau

Page	0	1	2	3	4	1	2	4	0	1
Cadre 1										
Cadre 2										
Cadre 3										
Défauts										

3.3 Comparaison stratégies LRU et FIFO

Pourquoi la stratégie LRU est elle *en pratique* meilleure que FIFO ?

10

3.4 Un scénario pathologique

Construire un scénario pathologique pour lequel LRU fait **plus de défauts de page** que FIFO (on trouve un contre-exemple simple avec 2 cadres, 3 pages, et une séquence de 5 références)

11 Donnez la séquence de références de pages, montrez le déroulement dans les deux cas, et concluez

4 Ordonnancement des processus

Les baladeurs/téléphones audio-video récents sont de véritables ordinateurs qui font tourner de multiples applications simultanément avec un système multi-tâche préemptif avec priorités. Certaines tâches ont des contraintes "temps-réel" sans être très prioritaires. Voici le pseudo-code d'une application qui joue un morceau de musique.

```
tantque le fichier mp3 n'est pas fini
  lire quelques secondes d'enregistrement
  décoder dans un tampon
  tant que le circuit son n'est pas prêt
    { attendre 10 ms }
  transmettre le tampon au circuit son
  attendre 1s
fin tant que
```

Dans les systèmes classiques, il y a une fonction `sleep(durée)` qui met le processus à l'état bloqué, le ramène à l'état prêt au bout du délai demandé, ce qui ne garantit pas qu'il redevienne actif de suite. Le risque dans cette application serait que la durée d'attente effective (retour à l'état prêt) soit trop longue, et que le circuit son ne soit pas suffisamment alimenté en données, d'où un son haché.

L'idéal serait d'indiquer une fourchette "`attendre de 10 à 15 ms`".

12 Quel serait l'effet de cette instruction, au niveau de l'ordonnancement ?

5 Mémoire partitionnée

Soient des partitions “libres” de mémoire de tailles 100K, 500K, 200K et 600K (dans cet ordre).

Des processus arrivent, avec des tailles 212K, 417K, 112K et 426K (dans cet ordre d’arrivée). Comment sont ils placés

13 Avec la stratégie First-fit (première zone libre) ?

14 Avec Best-fit ?

15 Avec Worst-fit ?

16 Quel algorithme fournit ici le meilleur usage de la mémoire ?

6 Gestion du bras du disque

On considère un disque dur présentant des pistes numérotées de 0 à 127.

Initialement, la tête de lecture/écriture est à la verticale de la piste 15. Des requêtes arrivent pour des accès aux pistes suivantes : 100, 30, 27, 55, 16, 122, 44, 63, 56 (dans cet ordre).

Quel serait le déplacement total de la tête de lecture/écriture (en nombre de pistes) après que toutes les demandes aient été satisfaites,

17 avec l'algorithme “premier arrivé premier servi” (détaillez votre réponse)

18 avec “le plus petit déplacement d’abord ?”

7 Adressage

Considérons un système de mémoire virtuelle ayant les caractéristiques suivantes :

- taille d’une page et d’un cadre de page (ou bloc ou case) = 1Ko (un kilo-octet),
- taille de la mémoire physique (ou vive ou principale) = 32 Mo (32 méga-octets),
- taille de la mémoire virtuelle = 512 Mo,
- utilisation combinée des techniques de pagination et segmentation : l’espace d’adressage virtuel d’un processus est composé de segments contigus. Chaque segment peut contenir entre une et 128 pages. La numérotation des pages d’un segment est relative au segment.

Donnez le format d’une adresse virtuelle (nombre de bits des numéros de segment, de page, et du déplacement)

19 en justifiant votre réponse

Donnez le format d’une adresse réelle (nombre de bits du numéro de bloc et du déplacement

20 en justifiant votre réponse