

Examen Final

IFT-2245

March 21, 2018

Directives

- Documentation autorisée: *une page manuscrite* recto.
- Pas de calculatrice, téléphone, ou autre appareil électronique autorisé.
- Répondre sur le questionnaire dans l'espace libre qui suit chaque question. Utiliser le verso des pages si nécessaire.
- Chaque question vaut 5 points pour un total maximum de 25 points.
- Les questions ne sont pas placées par ordre de difficulté.

0 Nom et prénom (1 point de bonus)

Écrire son nom et prénom et son code permanent en haut de chaque page.

1 Ordonnement

Soit les tâches suivantes:

Tâche	Burst time	Priorité
P1	3	3
P2	1	1
P3	9	2
P4	5	4
P5	6	5

Présumer qu'elles sont toutes arrivées au temps 0, dans l'ordre indiqué.

1. Illustrer dans des tableau de Gantt l'exécution qui en découle dans le cas de chaque ordonnancement:
 - (a) FCFS (First Come First Server)
 - (b) SJF (Shortest Job First)
 - (c) RR (Round Robin)
 - (d) Priorité non-préemptif
2. Pour chacun des cas, calculer le temps moyen d'attente (*waiting time*) et le temps moyen pour terminer l'exécution (*turnaround time*).

2 Mémoire virtuelle

Soit un programme qui roule sur une machine qui dispose de 4 cases (pages physiques ou *frames*). Il accède aux pages suivantes dans l'ordre indiqué:

2 1 9 2 1 4 5 1 9 3 8 3 1 2 1 3 2 9 3 9

En présumant que toutes les cases sont vides au départ, indiquer la séquence de *page faults* obtenues pour chacun des algorithmes de remplacement de page suivant:

1. FIFO (First In First Out)
2. LRU (Least Recently Used)
3. OPT (Optimal)

3 Accès mémoires

Soit un processeur avec des adresses logiques et physiques de 32bit, des pages de 8KBytes, et une pagination basée sur un table de pages hiérarchique à 2 niveaux.

1. Décrire comment les 32bit d'adresse logique sont divisés en 3 parties.
2. Écrire en pseudo code comment calculer l'adresse physique qui correspond à une adresse logique sur la base de chacune de ces 3 parties.
3. Décrire les différentes étapes que va suivre le processeur pour lire un mot à une adresse logique donnée (couvrir les différents cas possibles).

4 Disques durs

Soit un disque dur où chaque bloc contient 4KBytes et où les pistes extérieures contiennent 300 blocs chacune. Le disque tourne à une vitesse de 10000tpm et son temps moyen de recherche (*seek time*) est de 5ms.

1. Décrire la décomposition du temps moyen nécessaire pour accéder à un bloc d'un disque.
2. Quel est le taux de transfert maximum en megabytes par seconde (MB/s)?
3. Soit un processus P1 qui lit séquentiellement un fichier de 500MB placé de manière optimale sur le disque. Combien de temps faudra-t-il approximativement pour lire tout le fichier?
4. Soit un processus P2 qui lit un autre fichier de même taille dont les blocs, au lieu d'être bien contigus, sont répartis de manière complètement aléatoire sur l'ensemble du disque. Combien de temps faudra-t-il approximativement pour lire tout le fichier?
5. Quelle est la performance du disque mesurée en IO/s, c'est à dire en nombre de blocs qu'il peut servir chaque seconde dans le pire des cas?

5 Systèmes de fichiers

Soit un système de fichiers, avec indexation indirecte progressive, tel que le ext2 de Linux.

1. Lister tous les blocs qu'il faut modifier sur le disque lors de l'exécution de la fonction `unlink` qui efface un fichier de 3KB.
2. Spécifier l'ordre dans lequel ces opérations devraient être exécutées pour minimiser l'impact potentiel d'un crash à mi-course.
3. Sur la base de l'ordre précédent, indiquer après chaque opération quels problèmes apparaîtraient en cas de crash à ce moment.
4. Si le système de fichier est modifié pour garder un *journal*, indiquer quelles données devraient être écrites dans le journal pour l'opération précédente.
5. Estimer le coût en performance d'un tel journal pour cette opération; i.e. indiquer de combien de pourcents l'opération serait-elle ralentie dans le pire et dans le meilleur des cas.