

# Examen Final

IFT-2245

March 21, 2018

## Directives

- Documentation autorisée: une page recto de notes *manuscrites*.
- Répondre sur le questionnaire dans l'espace libre qui suit chaque question. Utiliser le verso des pages si nécessaire.
- Chaque question vaut 5 points avec un total maximum de 25 points (les 5 meilleures réponses).
- 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 matricule en haut de chaque page.

## 1 Ordonnancement et étreinte mortelle

Soit un multiprocesseur avec 8 cœurs et 18 disques (vides et interchangeable) attachés. Soit une séquence infinie de *jobs* soumis au système, où chacun requiert un accès exclusif à un maximum de 4 disques, quoiqu'il n'en utilise que 3 pendant tout le début de son exécution et n'en utilise 4 que peu de temps vers la fin.

1. Présume un ordonnanceur qui, pour éviter les étreintes mortelles (*deadlocks*), assigne à chaque programme 4 disques dès le départ qui ne sont libérés que quand le programme se termine. Quel est le nombre maximum de *jobs* qui peuvent être actifs à un même moment? Quel est le nombre maximum, minimum et moyen de disques qui sont inactifs?
2. Suggère un système d'ordonnancement différent pour augmenter l'utilisation des disques tout en évitant encore les étreintes mortelles.
3. Quel est le nombre maximum, minimum et moyen de disques qui sont inactifs avec ce nouvel algorithme?

## 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:

1 2 7 5 2 4 1 5 7 1 3 6 3 1 2 1 3 2 7 3

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. CLOCK (algorithme de la seconde chance)

### 3 Accès mémoires

Soit un processeur 64bit avec des adresses logiques effectivement limitées à 43bit (les 21bit du haut sont ignorés), des pages de 8KBytes, et une pagination basée sur une table de pages hiérarchique.

1. Combien de niveaux vont être nécessaires?
2. Décrire comment les 64bit d'adresses logiques seront divisés (quelles parties pour quels indexes de la table des pages, etc...).
3. Écrire en pseudo code comment calculer l'adresse physique qui correspond à une adresse logique sur la base de chacune de ces parties.
4. Décrire les différentes étapes que va suivre le processeur pour lire un mot à une adresse logique donnée (couvrir les différents cas possibles).
5. Pour un processus dont l'espace d'adressage est constitué de 3 pages (1 page de code, 1 page de pile, et 1 page de données globales), quelle est la quantité de mémoire minimale et maximale occupée par la table des pages?

## 4 Disques RAID

Soit un système avec 4 disques identiques qui ont chacun les caractéristiques suivantes: 1TB de donnée, 100MB/s de bande passante, et un temps d'accès moyen de 10ms. Indiquer les performances d'un système RAID utilisant ces 4 disques selon les différentes configurations ci-dessous:

|  | RAID-0 | RAID-1 | RAID-5 | RAID-10 |
|--|--------|--------|--------|---------|
| Taille utile<br>(en TB)                    |        |        |        |         |
| Bande passante en lecture<br>(en MB/s)     |        |        |        |         |
| Bande passante en écriture<br>(en MB/s)    |        |        |        |         |
| Temps d'accès moyen en lecture<br>(en ms)  |        |        |        |         |
| Temps d'accès moyen en écriture<br>(en ms) |        |        |        |         |

*Notes:* Au cas où vous ne vous souvenez plus quel numéro correspond à quoi:

- RAID-0: disques agrégés par bandes, aussi connu sous le nom de “*striping*”.
- RAID-1: disques en miroir, aussi connu sous le nom de “*mirroring*”.
- RAID-5: disques agrégé par bandes à parité répartie. Sur les 4 disques, un quart des données seront des données redondantes de parité.
- RAID-10: disques en miroir agrégés ensuite par bande (un “*stripe of mirrors*”).

## 5 Systèmes de fichiers

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

1. Lister tous les blocs qu'il faut modifier sur le disque lors de l'exécution de la fonction `open` qui crée un fichier (de taille zéro).
2. Spécifier l'ordre dans lequel ces opérations devraient être exécutées pour minimiser l'impact potentiel d'un crash à mi-course. Justifier brièvement.
3. Estimer le coût en performance pour cette opération si le système de fichiers est modifié pour garder un *journal*; i.e. indiquer de combien de pourcents l'opération serait-elle ralentie dans le pire et dans le meilleur des cas.