

Examen Final

IFT 2245

26 avril 2022

Directives

- Vous avez droit à une page de notes (recto-verso) écrite à la main.
- Calculatrice est autorisée.
- Répondez dans le cahier d'examen fourni à l'exception de Question 1 que vous pouvez répondre sur cette feuille.
- Le pointage pour chaque question est entre parenthèses (total = 20).
- Les traductions en anglais sont en *italics*.
- Vous pouvez répondre aux questions en anglais ou en français.
- Notez clairement toutes les suppositions que vous faites.

Question 0. *Nom et prénom (1 point de bonus)*

Écrivez votre nom et votre matricule en haut de ce papier et sur votre cahier d'examen.

Question 1. *Questions à choix multiple (4 points)*

(i) (0.5 points) Dans un système de stockage RAID, quel est l'avantage de la (*mirroring*) de disque (cochez tout ce qui s'applique)

- Il augmente le débit (*throughput*) de lecture
- Il augmente le débit (*throughput*) d'écriture
- Il économise de l'espace disque
- Il augmente la fiabilité

(ii) (0.5 points) Mettez les étapes suivantes nécessaires pour lire un fichier avec l'instruction `read(index)` dans le bon ordre :

- 2 Trouver la bonne entrée dans la table des fichiers ouverts du système (*system-wide open-file table*)
- 3 Charger le bloc de contrôle de fichier (*file-control block*)
- 1 Trouver la bonne entrée dans la table des fichiers ouverts par processus (*per-process open-file table*)
- 4 Lire les blocs de données

(iii) (1 point) Vous recevez le code de Hamming (à parité paire calculée de gauche à droite) suivant sans erreur :

1 0 1 0 0 0 1 0 1 0 0 1

Quel est le vrai byte qui a été transmis ?

- A. 1 0 0 0 1 0 0 0
- B. 0 0 0 1 0 0 0 1
- C. 1 0 0 1 1 0 0 1
- D. 1 0 1 1 0 1 1 1
- E. aucune de ces réponses

(iv) (1 point) Vous recevez le code de Hamming (à parité paire calculée de gauche à droite) suivant :

0 0 1 0 0 1 0 0 1 1 0 1

Il y a une erreur dans l'un des bits. Lequel ? (Quelle est sa position dans la chaîne de bits ?)

- A. 10
- B. 6
- C. 4
- D. 5
- E. aucune de ces réponses

(v) (0.5 points) De quelles manières pouvons-nous rendre les données disponibles à partir d'un conteneur Docker en cours d'exécution (cochez tout ce qui s'applique) ?

- Nous pouvons utiliser les *Linux CGroups*
- Nous pouvons les monter (*mount*) lorsque nous exécutons le conteneur
- Nous pouvons les copier dans l'image Docker lorsque nous créons l'image
- On peut y accéder avec un lien symbolique

(vi) (0.5 points) De quelles manières pouvons-nous créer une nouvelle image Docker ?

- Nous pouvons le télécharger à partir du registre (avec *docker pull*)
- Nous pouvons commettre un conteneur en cours d'exécution (avec *docker commit*)
- Nous pouvons construire l'image à partir de Dockerfile (avec *docker build*)
- Nous pouvons exécuter l'image (avec *docker run*)

Question 2. Systèmes de fichiers (3 points)

Pour les trois questions suivantes, considérez que la taille de bloc est de **4KB** et que l'espace d'adressage est **64 bits**.

(i) (1 point) Quelle est la taille de fichier maximale accessible par un système de fichiers avec une **table FAT** avec 50 lignes ?

50 * 4 KB

$$4KB \times 2^9$$

(ii) (1 point) Quelle est la taille de fichier maximale accessible par un système de fichiers **indexé** en supposant un seul bloc d'index ?

(iii) (1 point) Quelle est la taille de fichier maximale accessible par un **inode** dans un système de fichiers **ext2** ?

$$\left(12 + 2^9 + 2^9 \cdot 2^9 + 2^9 \cdot 2^9 \cdot 2^9 \right) 4KB$$

Question 3. Accès à la mémoire de masse (2 points)

Compte tenu les informations suivantes :

- Taille d'un bloc est 8KB
- Le disque tourne a une vitesse de 7200 révolutions par minute
- Le taux de transfert est 1Gb/s
- Le seek time moyen est 4ms

$$8KB \times \frac{8b}{1B} \cdot \frac{1s}{14b} \cdot \frac{1KB}{2^{20}KB} = \frac{64}{2^{20}} = 0.06ms$$

(i) (1 point) Quel est le temps de transfert, T_t , pour un bloc (en ms) ?

(ii) (1 point) Quel est le temps moyen d'entrée/sortie, T_{io} , pour un bloc (en ms) ?

$$T_e = 0.5$$

$$0.06 + 4 + 4 \cdot 17 = 8.23$$

Question 4. Ordonnancement de disque (2 points)

Un disque a 140 pistes. À l'instant présent, la tête de lecture est sur la piste 30 en se déplaçant dans le sens de l'augmentation des numéros de piste. La séquence des requêtes de lecture de piste à venir est :

[80, 10, 25, 100]

$$(i) 30, 80, 100, 10, 25$$

(i) (1 point) Quel est la distance totale parcourue par la tête de lecture avec l'algorithme C-LOOK ?

$$100 - 30 +$$

(ii) (1 point) Quel est la distance totale parcourue par la tête de lecture avec l'algorithme SCAN ?

$$100 - 10 +$$

$$(ii) 30, 80, 100, 140, 25, 10$$

$$25 - 10$$

$$140 - 30 + 140 - 10 = 175$$

$$= 240$$

Question 5. Mémoire virtuelle (3 points)

Un processus qui s'exécute sur une machine a été alloué **3 frames** en mémoire. Il accède aux pages suivantes dans l'ordre indiqué :

$$taille \Delta = 5$$

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

En présumant que toutes les *frames* sont vides au départ (pagination à la demande pur), indique le contenu des *frames* et le nombre total de *page faults* pour chacun des algorithmes de remplacement de page suivants :

(i) (1 point) Deuxième Chance (*Second Chance* ou "horloge")

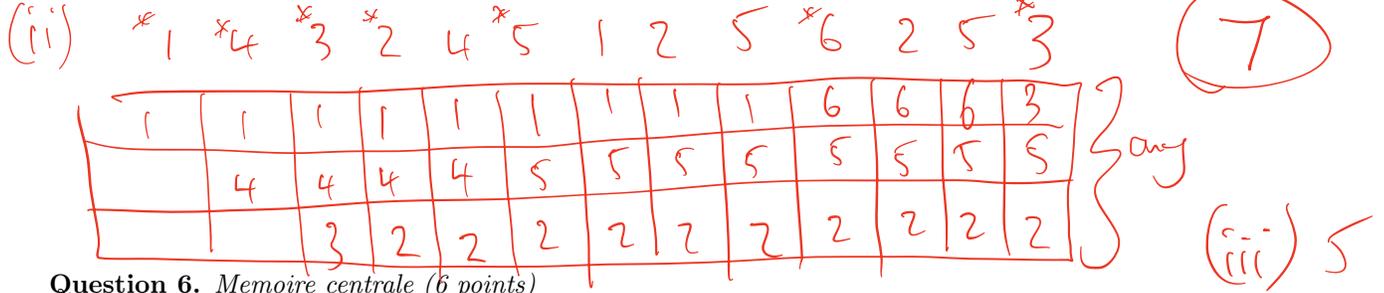
(ii) (1 point) Optimal

(iii) (1 point) Supposons la même séquence d'accès aux pages comme indiqué ci-dessus. Si la valeur de $\Delta = 5$, quelle est la taille **maximale** de la *working set* pour la durée de la séquence.

$$(i) \quad *1 \quad *4 \quad *3 \quad *2 \quad 4 \quad *5 \quad *1 \quad 2 \quad 5 \quad *6 \quad 2 \quad *5 \quad *3$$

9

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



Question 6. Mémoire centrale (6 points)

(i) (2 points) En supposant un espace d'adressage **32-bits**, quelle est la taille du frame (page) pour que la table de page dans un système de pagination à 1 niveau tienne dans un frame dans la mémoire principale (Réponse en KB.)? **128 KB**

(ii) (1 point) Considérez un système de pagination avec un "translational look-aside buffer" (TLB) où la table de pages est stockée dans la mémoire principale. Étant donné qu'un temps de recherche dans le TLB est $\epsilon = 40$, et que le temps d'accès à la mémoire principale est $m = 400$, trouvez le taux de réussite du TLB (*TLB hit ratio*) α ($0 \leq \alpha \leq 1$) pour que le temps d'accès effectif (EAT) pour accéder à la mémoire soit réduit à 80% par rapport au cas où il n'y a pas de TLB.

En considérant le Fig.1 :

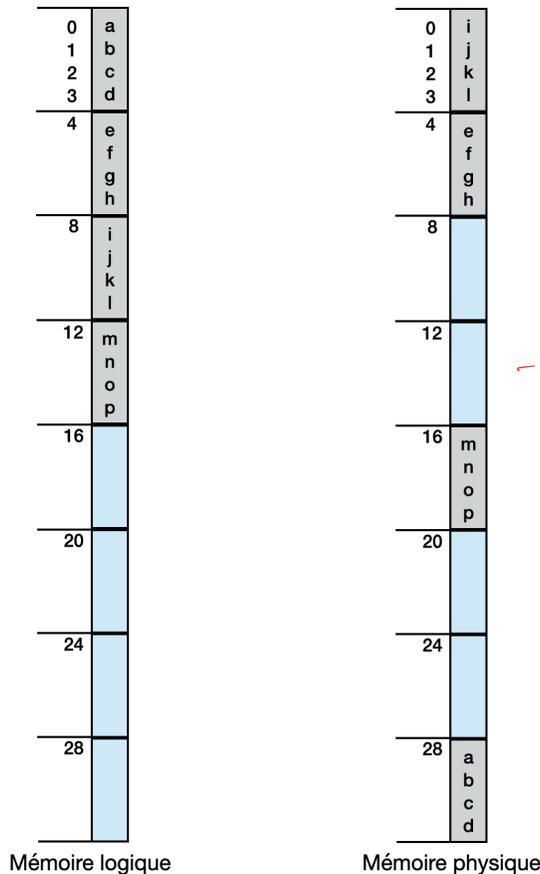
(iii) (1 point) Comment l'espace d'adressage logique est-il divisé (combien de bits pour le numéro de page et combien de bits pour le décalage) ?

(iv) (2 points) Dessinez la table des pages pour ce processus et incluez les bits valides/non valides dans chaque ligne.



(iv)

0	7	✓
1	1	✓
2	0	✓
3	4	✓
4		i
5		i
6		i
7		i



(ii) sans TLB
 access = 800.
 wait EAT = (800)0.8
 = 640
 $EAT = (40 + 400)\alpha$
 $+ (40 + 800)(1-\alpha)$
 $640 = -400\alpha$
 $-840 = -400\alpha$
 $\alpha = \frac{200}{400} = 0.5$

FIGURE 1 – Question 6 (iii) et (iv)