

Examen Final

IFT 2245

24 avril 2019

Directives

- *Aucune* documentation autorisée.
- Pas de téléphone ou autre appareil électronique autorisé (sauf calculatrice).
- Répondre **dans le cahier**.
- Calculatrice est autorisée.
- Le pointage pour chaque question est entre parenthèses (total = 30).
- Les termes en anglais sont indiqués en *italics*.

Des choses qui pourraient vous aider

- 8b=1B, 2^{10} B=1KB, 2^{10} KB=1MB, 2^{10} MB=1GB

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

Écrivez votre nom et prénom et matricule en haut et sur la page couverture du cahier réponse.

Question 1. Accès mémoires (5 points au total)

(a) (2 points) Considérons un système avec:

- Le temps d'accéder à la mémoire centrale est **200ns**
- Le temps pour faire un cherche dans le *Translation Lookaside Buffer* (TLB) est **20ns**
- Le temps pour traiter une *page fault* est **8000ns**
- Le taux de réussite du TLB est **90%**
- Le taux de *page fault* est **1 “manque (miss)” par 100 accès**

Quel est le temps d'accès effectif (*effective access time* (EAT)) pour accéder à une page en mémoire?

Les détails suivants s'appliquent aux questions (b) - (d): Étant donné un système avec un espace d'adressage de 64 bits, une taille de page de 1 MB (2^{20} B) et avec la taille d'une entrée dans la table de pages de 8B (=64b):

(b) (1 point) Quelle est la taille (en *Bytes*) de la table de page à un niveau?

(c) (1 point) Considérant que nous voulons que chaque table de pages dans un système de pagination hiérarchique s'inscrive dans un *frame* de mémoire (c'est-à-dire ne soit pas plus grande qu'une page), combien de niveaux de pagination sont nécessaires et comment l'espace d'adressage est-il divisé?

$$\max = \frac{2^{20}}{2^3} = 2^{17}$$



3-levels

$$\begin{aligned} &\rightarrow \text{TLB hit } 90\% \\ &\rightarrow \text{TLB miss no. p.f.} \cdot (0.1 \times 0.99) \\ &\rightarrow \text{TLB miss} + \text{p.f.} \cdot (0.1 \times 0.01) \end{aligned}$$

$$(0.9)(220) + (0.1)(0.99)(420) + (0.01 \times 0.1)(8000) = 247$$

$$\begin{aligned} (a) (i) \text{ no page faults: } & \text{EAT} = (220)0.9 + (420)0.1 = 240 \text{ ns} \\ (ii) \text{ w/ p.f.} & = (240)0.99 + 8000(0.01) = 317.6 \text{ ns} \end{aligned}$$

1 1 → 1 4 4 → 4 4 4 → 4 5 5 5 5 5
 2 2 → 2 1 1 1 → 1 3 3 3 3 (10)
 3 3 → 3 5 5 5 → 5 6 6 6
 x x x x x x x x x x

Final - ift2245

Nom: _____ Matricule: _____

Question 4. Système de fichiers (5 points au total)

(a) (2 points) Un système de fichiers **ext2** utilise un adressage **64-bit** et une taille de bloc de **4KB**. Le structure de l'inode dans le système **ext2** est montré dans Table 1.

mode
propriétaires (<i>owners</i>)
les horodatages (<i>timestamps</i>)
nombre de blocs de taille (<i>size block count</i>)
12 blocs direct (<i>direct blocks</i>)
simple indirect
double indirect
triple indirect

$$(4KB)12 + (2^9)2^{12} + (2^9)(2^9)2^{12} + 2$$

TABLE 1 – structure inode

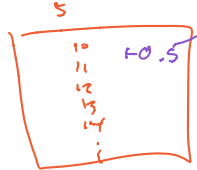
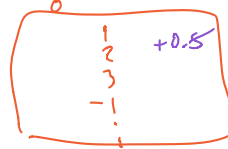
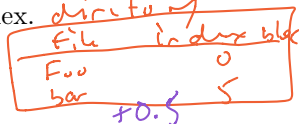
Quelle est la taille maximale qu'un fichier peut avoir pour qu'il n'a pas besoin d'utiliser le pointeur triple indirection dans l'inode ?

(b) (1 point) Quel est l'avantage d'une *file allocation table* (FAT) par rapport à un système de fichiers chaîné ?
don't have to load the blocks / better for random access

(c) (2 points) Dessine un système de fichiers (répertoire + disque de 20 blocs) pour un système de fichiers **indexé** contenant 2 fichiers:

- **foo.txt** besoin de 3 blocs
- **bar.txt** besoin de 5 blocs

Indiquer quels blocs sont libres et lesquels sont attribués à chaque fichier et montrer le contenu de chaque bloc d'index.



Question 5. Conteneurisation (2 points)

(a) (1 point) Quelle est la différence principale entre un **conteneur** et une **machine virtuelle** (dessinez une image si cela vous aide) ?
VM runs an OS inside, container manages communication and access w/ base OS.

(b) (1 point) Avec "Docker", quelle est la différence principale entre un conteneur et une image ?

image read only (static) container is writeable (dynamic)

Question 6. Questions aléatoires (4 points)

(a) (1 point) Qu'est-ce qu'une condition de course (*race condition*) ?

Order of execution determines outcome

(b) (1.5 points) Compte tenu de la définition et de l'utilisation ci-dessous de la fonction **compare_and_swap** utilisée pour la synchronisation des processus:

```

int compare_and_swap(int *value, int expected, int new_value) {
    int temp = *value;
    if (*value == expected)
        *value = new_value;
    return temp;
}

do {while (compare_and_swap(&lock, 0, 1) != 0)
    ; /* do nothing */
    /* critical section */
    lock = 0;
    /* remainder section */
} while (true);

```

Cette approche au problème de section critique garantit-elle (aucune justification nécessaire):

- i. Exclusion mutuelle? (oui/non) *+ .5*
- ii. Progrès? (oui/non) *+ .5*
- iii. Attente limitée? (oui/non) *+ .5*

(c) (1.5 points) Quel niveau d'ordonnanceur est associé aux tâches suivantes (court, moyen, ou long):

- i. Choisir parmi une liste de processus prêts à exécuter lequel doit être chargé en mémoire, *long +0.5*
- ii. Choisir parmi une liste de processus en mémoire lequel doit être exécuté par la CPU, *court +0.5*
- iii. Contrôler le degré de multiprogrammation par "swapping". *moyen +0.5*

Question 7. RAID (4 points)

(a) (1 point) Quels sont 2 avantages d'un système RAID d'avoir des disques en miroir (*mirroring*)? *throughput for reading +0.5, reliability +0.5*

(b) (3 points) Avec un système avec codage de Hamming à parité paire calculée de gauche à droite, vous recevez le code de Hamming 12 bits suivant:

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

i. Ce code a une erreur de 1 bit, quel bit?

ii. Quel est l'octet (*Byte*) correct?

$P1: 110111 = 71$
 $P2: 010111 = 70$
 $P4: 10010 = 70$
 $P8: 01110 = 71$

$P1 P2 1 P4 0 0 1 P8 \rightarrow 110$
 $\downarrow$
 9

