

## Examen (1ère session) – 14 avril 2014

N. Sabouret

*L'épreuve dure 2h00. Tous les documents sont autorisés. Les exercices sont indépendants.*

### 1 Exercice 1 – Question de cours (4 points)

- (1 point) Dans un système comprenant uniquement une UC, une RAM, une horloge, un disque avec DMA, combien faut-il de lignes d'interruptions sur le bus? Expliquez.

*À strictement parler, il n'y a qu'une ligne d'interruption pour l'UC, mais il faut aussi une ligne pour l'horloge et une ligne pour le DMA.*

- Expliquez la différence entre la FAT et la méthode d'allocation indexée dans un système de fichiers (1 point)

*La FAT est une table en liste chaînée, alors que les blocs d'index sont ordonnés.*

- Donnez la structure en mémoire d'un processus multi-threadé (1 point)

*Communs: environnement, tas, code ; pour chaque thread: Pile.*

- Donnez les principaux éléments d'un File Control Block (1 point)

*Il faut au moins: dates, propriétaire, permissions/droits, taille, adresse du blocs de données*

### 2 Exercice 2 – Disques SSD (5 points)

On considère un disque SSD de 8 Mo répartis en blocs de 512 Ko numérotés de 0 à F (en hexadécimal). La table suivante donne les numéros de blocs physiques, les blocs logiques qu'ils contiennent (ou L lorsque le bloc est libre), la date de dernière modification (exprimée en unités de temps depuis le début de la vie du disque) et l'usure (en nombre d'utilisation depuis le début d'utilisation du disque) est donnée par la table suivante :

phys.:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
log. :	8	3	2	1	0	L	B	6	L	9	5	7	4	L	L	L
date:	121	112	123	107	110	100	130	181	120	151	132	128	117	050	0	0
usure:	7	8	7	9	6	5	8	6	7	9	6	8	8	3	0	0

Nous sommes à l'unité de temps 182 et on suppose que l'UC fait, dans l'ordre, des requêtes d'écriture sur les pages situées sur les blocs logiques suivants (1 requête par unité de temps) :

A 7 8 7 1 0 A B

- Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique qui ne fait aucun Wear Levelling. Expliquez vos choix (2 points)

Si on ne fait aucun WL, les blocs ne sont pas déplacés. Les blocs logiques 0, 1, 7, 8 et B sont donc modifiés dans les blocs physiques correspondants (4, 3, B, 0 et 6), dont l'usure augmente de 1 (2 pour le bloc B contenant le bloc logique 7) et l'ancienneté modifiée en fonction de l'usage (respectivement 187, 186, 185, 184 et 189). Enfin, pour le bloc logique A, il faut choisir un bloc libre. Il n'y a aucune consigne dans l'énoncé quant à la table d'association entre blocs physiques et blocs logiques, donc A peut être placé n'importe où (nous avons choisi le premier bloc libre, mais c'est discutable).

P:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
L:	8	3	2	1	0	A	B	6	L	9	5	7	4	L	L	L
D:	184	112	123	188	187	188	189	181	120	151	132	185	117	050	0	0
U:	8	8	7	10	7	7	9	6	7	9	6	10	8	3	0	0

2. Donnez, à l'aide d'une table similaire à celle fournie dans l'énoncé, le résultats de l'exécution en utilisant une politique de Wear Levelling **statique** avec un délai d'ancienneté fixé à 10 unités de temps. Expliquez clairement ce qu'il se passe à chaque pas de temps de l'exécution. (3 points)

**NB:** En cas de doute, toutes choses étant égales par ailleurs, vous choisirez le premier bloc dans l'ordre numérique.

182 Pour le bloc logique A, on cherche le bloc physique le moins utilisé (c'est E). Il est libre, on l'utilise.

183 Pour le bloc logique 7, même principe. On place sur donc en F et on marque B comme libre.

184 Pour le bloc 8, le moins utilisé est à nouveau E, mais il a été utilisé il y a moins de 10 pas de temps. Idem pour F. Donc on se place en D (qui est libre) et on marque 0 comme libre.

185 Pour le bloc 7 à nouveau, les plus anciens (D, E et F) sont non-libres et récents, donc on se place en 5 (qui est le suivant sur la liste, et qui est libre). On marque F comme libre.

186 Pour le bloc 1, on utilise F libre et on marque 3 comme libre.

187 Pour le bloc 0, blocs D, E et F sont peu utilisés mais récents. Les candidats sont donc 4, 5, 7 et A mais 5 et 7 sont récents aussi. Restent donc 4 et A qui sont tous les deux non-libres. On va donc prendre 4 qui, quelle chance, contient justement le bloc 0 (donc on n'a rien à déplacer).

188 Pour le bloc A, on est dans la même situation que précédemment (4, 5, 7, D, E et F peu utilisés mais trop récents), il ne reste que A, qui n'est pas libre. On marque donc E libre, on place A dans A, et on doit maintenant placer le bloc 5 ailleurs.

189 Pour le bloc 5, on a E qui est peu utilisé et libre.

190 Pour le bloc B, les candidats sont 0, 2 et 8 (les autres sont trop récents). Or 0 est libre, donc on l'utilise.

P:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
L:	B	3	2	L	0	7	B	6	L	9	A	L	4	8	5	1
D:	190	112	123	107	187	185	130	181	120	151	188	128	117	184	189	186
U:	8	8	7	9	7	6	8	6	7	9	7	8	8	4	2	2

### 3 Exercice 3 – Gestion de la mémoire (5 points)

On considère un système de gestion de mémoire de 8 Ko, gérée de manière segmentée et paginée avec un seul niveau de pagination. Un processus peut avoir au plus 8 segments. Chaque segment peut adresser au plus 4 ko de données ou de code. Enfin, la taille des cadres de page est fixée à 512 octets.

1. Quelle est la taille (en nombre de bits) de l'adresse logique? Expliquez. (1 point)

*Pour adresser 8 segments par processus, il faut 3 bits. Donc le sélecteur dans l'adresse logique est sur 3 bits. Si chaque segment peut adresser au plus 4 Ko de données, le décalage est sur 12 bits. L'adresse logique est donc sur 15 bits (chaque processus adresse au plus 32 Ko).*

2. Quelle est la taille (en nombre de bits) de l'adresse linéaire? Expliquez quelles sont les sous-parties qui composent cette adresse et leurs tailles respectives. (1 point)

*Si on a 8Ko de RAM, l'adresse physique est forcément sur 13 bits. Avec des cadres de pages de 512o, il faut 9 bits pour le décallage (et donc 4 bits pour le numéro de cadre). Donc l'adresse linéaire est sur 13 bits (avec le même découpage).*

3. Quelle est la taille de chaque table des pages et de la table des segments? Expliquez. (1 point)

*Chaque table des pages contient  $2^4$  lignes de 4 bits (numéro du cadre) + 1 bit de validité. Soit un total de  $16 \times 5 = 80$  bits (10 octets).*

*La table des segments contient **au plus** 8 lignes de 12 (limite) + 15 bits (base) chacune, soit en tout 216 bits (27 octets).*

4. On considère l'adresse logique 20 B7 (exprimée en hexadécimal, c'est-à-dire 0010 0000 1011 0111). Expliquez comment est calculée l'adresse physique et donnez sa valeur en indiquant clairement tous les calculs intermédiaires (2 points).

On suppose que le processus utilise 4 segments et que la table des segments contient les valeurs suivantes (toutes données en hexadécimal):

Segment:	Limite	Base
0	00 4A	0A 29
1	00 09	00 97
2	00 C8	0B 32
3	00 17	00 76

On suppose aussi que la table des pages concernée par la requête sera la suivante:

Page:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Cadre:	E	7	1	A	0	8	0	3	0	0	0	F	5	0	0	2
Valide:	1	1	1	1	0	1	0	1	1	0	0	1	1	0	0	1

*Adresse logique: 20 B7, se décompose en numéro de segment (2) et décalage (00 B7)*

*La table des segments indique que le segment 2 a une limite de 00 C8 qui est bien supérieure au décalage.*

*L'adresse linéaire est donc 0B 32 (base) + 00 B7 (décalage) = 0B E9 (0 1011 1101 0111 sur 13 bits), ce qui se décompose en 0101 (5) pour le numéro de page et 01 E9 pour le décalage.*

*D'après la table des pages, la page 5 est située sur le cadre 8 (1000).*

*L'adresse physique est donc 1 0001 1101 0111, c'est-à-dire 11 E9*

## 4 Exercice 4 – Système de fichiers (3 points)

On considère un système de fichier NFS dans lequel une machine S joue le rôle de serveur et une machine C joue le rôle de client. La machine serveur contient un répertoire à deux niveaux:

```
S/: toto titi f1.txt f2.txt
S/toto: f1.txt f3.txt
S/titi: f2.txt f3.txt
```

La machine client contient une structure de répertoire en graphe:

```
C/: a b
C/a: /a/b /a/c
C/b: <vide>
C/a/b: lien vers C/b
C/a/c: f1.txt
```

La machine cliente a monté le répertoire toto sur le point b.

1. Dessinez la structure des répertoires en mettant dans deux encadrés différents ce qui figure sur la machine S et ce qui figure sur la machine C. (1 point)

*Attention, il ne faut pas faire figurer f1 et f3 dans C:/b, et encore moins toto !*

S	C
+-- toto	+-- a
+-- f1.txt	+-- b <lien -> C:/b>
+-- f3.txt	+-- c
+-- titi	+-- f1.txt
+-- f2.txt	+-- b <lien -> S:/toto>
+-- f3.txt	
+-- f1.txt	
+-- f2.txt	

2. Décrivez précisément ce qu'il se passe au niveau du système d'exploitation et des systèmes de fichiers (sur les machines et sur le NFS) lorsque l'utilisateur de la machine C fait une commande de lecture du fichier a/b/f1.txt. Combien de requêtes sont échangées? (2 points)

*open: C regarde dans le répertoire pour trouver le FCB de a. Il regarde ensuite le répertoire de a pour trouver le FCB de b qui est un lien vers le b du répertoire principal. Il regarde donc dans le répertoire pour trouver le FCB de b qui est un FileDescriptor NFS. Il demande donc à charger le répertoire à l'aide de cette clef qui a été fournie au montage. Cela fait une première RPC vers S qui répond avec un nouveau FD correspondant au fichier ouvert.*

*read: C demande le contenu du fichier à l'aide du FD qui est stocké dans la table des fichiers ouverts. Cela fait une deuxième RPC.*

## 5 Exercice 5 – Ordonnancement de processus (3 points)

On considère un système de gestion de processus en temps partagé équitable, comme sous Linux, fonctionnant de la manière suivante:

- Chaque processus rentre dans la file avec une certaine priorité, qui correspond à son crédit initial;
- À chaque fin de quantum, le processus exécuté, s'il n'a pas terminé son calcul, perd un crédit;
- Lorsqu'aucun processus prêt n'a de crédit, tous les processus sont recredités de 50% de leur crédit plus leur priorité.

On suppose que le quantum de temps est fixé à 2 et on considère l'exécution suivante:

- À  $t=0$ , les processus suivants entrent dans la file des processus prêts:
  - P1 de priorité 3 dont l'exécution nécessite 4 pas de temps, puis une E/S qui prend 8 pas de temps, puis encore 2 pas de temps
  - P2 de priorité 1 dont l'exécution nécessite 6 pas de temps.
- À  $t=5$ , le processus P3 entre dans la file, il a une priorité de 2 et son exécution nécessite 6 pas de temps.

1. En négligeant le temps de commutation, donnez le diagramme de gantt résultant de l'exécution des processus (2 points).

*J'indique pour chaque date les processus en exécution, prêts et en attente, avec entre parenthèse leur temps restant et leur priorité)*

t	0	2	4	5	7	9	11
E	P1 (4, 3)	P1 (2, 2)	P2 (6, 1)	P3 (6, 2)	P3 (4, 1)	P2 (5, 1)	–
P	P2 (6, 1)	P2 (6, 1)	–	P2 (5, 1)	P2 (5, 1)	P3 (2, 0)	P3 (2, 0), P2 (3, 0)
A	–	–	P1 (8, 2)	P1 (7, 2)	P1 (5, 2)	P1 (3, 2)	P1 (1, 2)

recredit + sélection du plus prioritaire après recredit (P3)

t	11	12	14	16	17	18
E	P3 (2, 2)	P1 (2, 6)	P2 (3, 1)	P3 (1, 1)	P2 (1, 1)	–
P	P2 (3, 1)	P2 (3, 1), P3 (1, 1)	P3 (1, 1)	P2 (1, 0)	–	–
A	P1 (1, 6)	–	–	–	–	–

2. Donnez le temps d'attente moyen, le temps de rotation moyen et le débit sur cet exemple. Vous devrez indiquer le détail de vos calculs. (1 point)

*Temps d'attente:  $P1=0$ ,  $P2=4+4+3+1=12$ ,  $P3=2+4=6$ , moyenne = 6*

*Rotation:  $P1=14$ ,  $P2=18$ ,  $P3=17-5=12$ , moyenne=14.6*

*Débit:  $18/3=6$*