	<p>IUT - Université Bordeaux 1 Département Informatique</p> <p>Devoir surveillé ASR3-système</p>	<p>15 novembre 2013</p> <p>Groupe</p> <p>Nom</p> <p>Prénom</p>
---	---	--

Sans documents – Durée 1h30 min
Justifiez les réponses

1 Algorithmes de remplacement de page

1 Que signifient les acronymes FIFO et LRU (traduction + explication), dans le contexte des stratégies de remplacement de page ?

Solution : Il s'agit de déterminer le cadre de page qui va être libéré
 FIFO first-in first-out : les cadres de pages sont sélectionnés à tour de rôle.
 LRU least recetnly used : on sélectionne la page dont la dernière utilisation est la plus ancienne.

2 Pour tester les performances d'algorithmes de remplacement de pages, on peut écrire des programmes de simulation pour évaluer le nombre de chargements causés par des suites de référence à des pages.

Que pensez vous de l'idée de tirer ces suites complètement au hasard ?

Solution : Mauvaise idée, parce que les suites aléatoires ne rendent pas compte des phénomènes de localité : il y a une forte probabilité que les mêmes pages soient référencées dans un intervalle de temps court. Les résultats ne seront pas réalistes.

3 Déterminez le nombre de chargements de page engendrés par l'algorithme FIFO sur la séquence de références

7 0 1 2 3 0 4 2 3 0 3 2 1 2

avec 4 cadres de page :

	7	0	1	2	3	0	4	2	3	0	3	2	1	2	
FIFO															Nombre de chargements =

Solution : Calculé par programme :

```
./fifo 4 7 0 1 2 3 0 4 2 3 0 3 2 1 2
Algo FIFO 4 cadres, 14 references
0) 7 => | 7.| | | | | <=
1) 0 => | 7 | 0.| | | | <=
2) 1 => | 7 | 0 | 1.| | | <=
3) 2 => | 7 | 0 | 1 | 2.| <=
4) 3 => | 3.| 0 | 1 | 2 | <=
5) 0 => | 3 | 0.| 1 | 2 |
6) 4 => | 3 | 4.| 1 | 2 | <=
7) 2 => | 3 | 4 | 1 | 2.|
8) 3 => | 3.| 4 | 1 | 2 |
9) 0 => | 3 | 4 | 0.| 2 | <=
10) 3 => | 3.| 4 | 0 | 2 |
11) 2 => | 3 | 4 | 0 | 2.|
12) 1 => | 3 | 4 | 0 | 1.| <=
13) 2 => | 2.| 4 | 0 | 1 | <=
```

9 chargements

4 même question avec l'algorithme LRU

	7	0	1	2	3	0	4	2	3	0	3	2	1	2	
LRU															Nombre de chargements =

Solution : Calculé par programme :

```

./lru 4 7 0 1 2 3 0 4 2 3 0 3 2 1 2
Algo LRU 4 cadres, 14 references
0) 7 => | 7.| | | | <=
1) 0 => | 7 | 0.| | | <=
2) 1 => | 7 | 0 | 1.| | <=
3) 2 => | 7 | 0 | 1 | 2.| <=
4) 3 => | 3.| 0 | 1 | 2 | <=
5) 0 => | 3 | 0.| 1 | 2 |
6) 4 => | 3 | 0 | 4.| 2 | <=
7) 2 => | 3 | 0 | 4 | 2.|
8) 3 => | 3.| 0 | 4 | 2 |
9) 0 => | 3 | 0.| 4 | 2 |
10) 3 => | 3.| 0 | 4 | 2 |
11) 2 => | 3 | 0 | 4 | 2.|
12) 1 => | 3 | 0 | 1.| 2 | <=
13) 2 => | 3 | 0 | 1 | 2.|

```

7 chargements

2 Processus

Dans un centre de calcul des années 60, on exécute des travaux scientifiques qui consistent essentiellement à ... calculer. On néglige les entrées-sorties.

Trois travaux A, B, C arrivent pratiquement en même temps. Leurs temps de calcul respectifs sont estimés à 2, 5 et 3 minutes.

2.1 Système mono-tâche

On suppose que le système d'exploitation est mono-tâche. Déterminez le temps d'attente moyen pour chacun des cas suivants. Un graphique permettra de justifier vos affirmations :

- 5 Les travaux sont lancés l'un après l'autre, dans l'ordre d'arrivée A, B, C.

Solution :

1

1234567890

| | |

AABBBBBCCC

A se finit au bout de 2 minutes, B en $2+5=7$, C en $7+3= 10$. Moyenne = $19/2 = 6.33$ min.

6 Lancement du plus court d'abord

Solution :

```
          1
1234567890
|   |   |
AACCCBBBBB
```

Moyenne = $(2+5+10)/3 = 17/3 = 5,66$ min.

2.2 Système multi-tâches

On suppose maintenant que le système est multi-tâches avec ordonnancement préemptif. Le quantum de temps petit par rapport aux durées d'exécution. On néglige le temps passé à la commutation de contexte et à la gestion du temps partagé. Pas de miracle à attendre : la durée totale d'exécution sera toujours de 10 minutes.

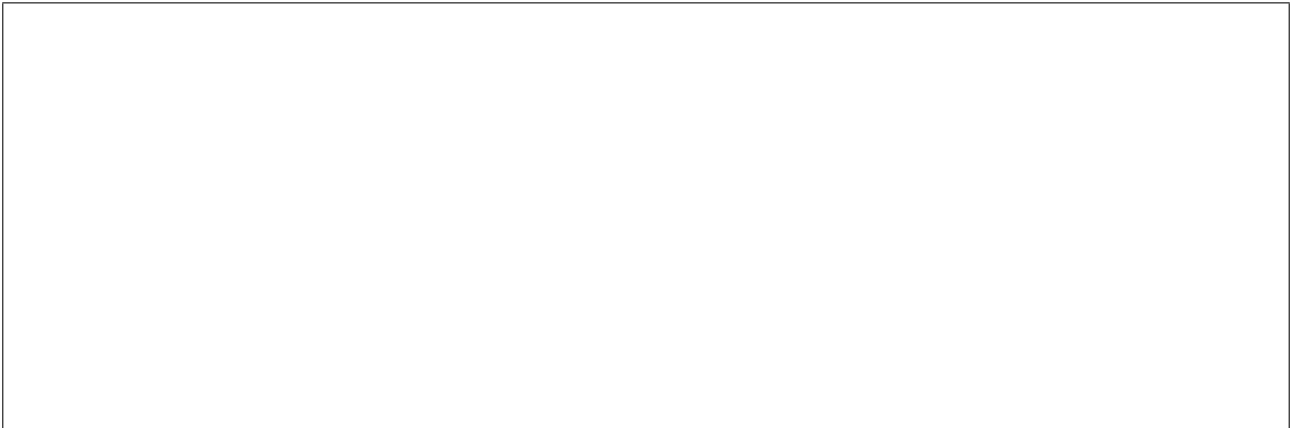
7 Question de cours : Expliquez ce qu'est le *quantum de temps* dans un ordonnancement préemptif. De quel ordre de grandeur est-il ?

Solution : C'est la durée au bout de laquelle le système d'exploitation fait passer un processus actif, qui ne demande pas d'entrées sorties, à l'état prêt.
Il est de l'ordre du 50 au 1000-ième de seconde.

8 Exécution en parallèle¹ avec ordonnancement par l'algorithme du tourniquet.

1. penser que si N travaux se partagent le processeur, ils vont tous N fois moins vite que si ils étaient seuls sur la machine....

Indication : Au début, il y a 3 processus en parallèle, il faut donc 3 fois 2 = 6 minutes (en temps réel) pour que A (le plus court) se termine. Il reste donc 5-2=3 minutes de calcul (en temps relatif) à faire pour B, et 3-2=1 minute pour C.



Solution :

Au début, il y a 3 processus en parallèle, il faut donc 3 fois 2 = 6 minutes pour que A se termine.

AaaAaa
 BbbBbb
 CccCcc

A, B, et C ont alors avancé de 2 minutes chacun en temps relatif. Il reste donc à exécuter 3 minutes de temps relatif pour B et 1 pour C. Il faut donc 2 minutes de temps réel pour que C se termine.

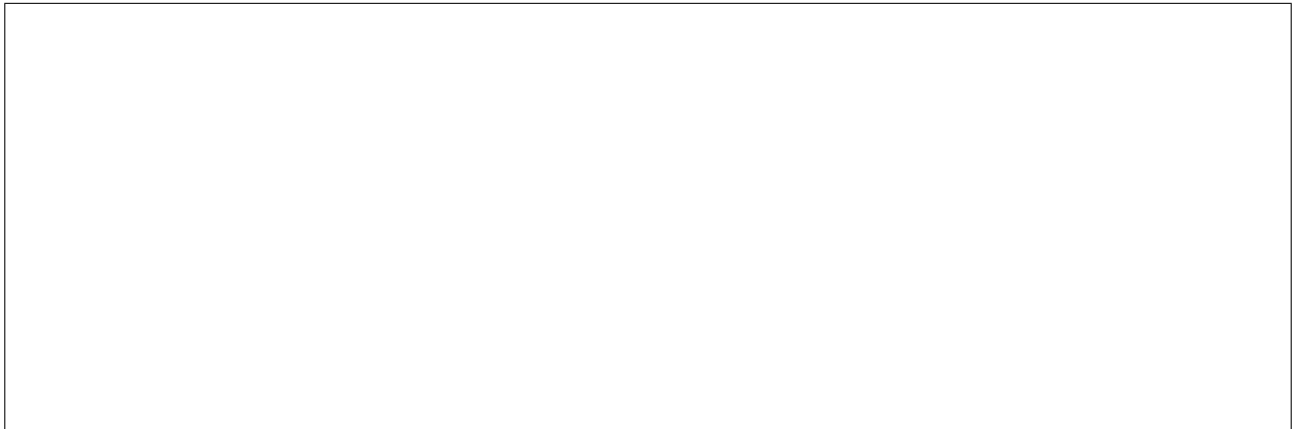
AaaAaa
 BbbBbbBb
 CccCccCc

puis C finit tout seul ses deux minutes

 1
 1234567890
 | | |
 AaaAaa
 BbbBbbBb
 CccCccCcCC

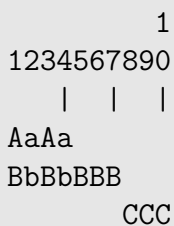
Moyenne = $(6+8+10)/3 = 22/3 = 7,33$

9 Exécution selon les priorités, avec tourniquet entre processus de même niveau : A et B ont la même priorité, qui est supérieure à celle de C.



Solution :

Au début, A et B tournent en parallèle. il faut donc 2 fois 2 = 4 minutes pour que A se termine. Il reste alors 5-2=3 minutes pour B, qui s'exécute tout seul parce qu'il est prioritaire sur C. Et ensuite C tourne tout seul



Moyenne = $(4+7+10)/3 = 21/3 = 7$ minutes

3 Mémoire

L'ordinateur PDP-10, commercialisé par Digital Equipment Corporation dans les années 60-70, a une grande importance historique : très répandu dans les universités et les laboratoires de recherche américains (environ 700 exemplaires vendus), c'est la machine qui a popularisé le temps partagé.²

Les PDP-10 avaient des mots de 36 bits, avec des adresses sur 18 bits. La documentation utilisait la notation octale pour simplifier : un chiffre octal représente 3 bits, dont une adresse (18 bits) s'écrit sur 6 chiffres octal. On met des points pour faciliter la lecture :

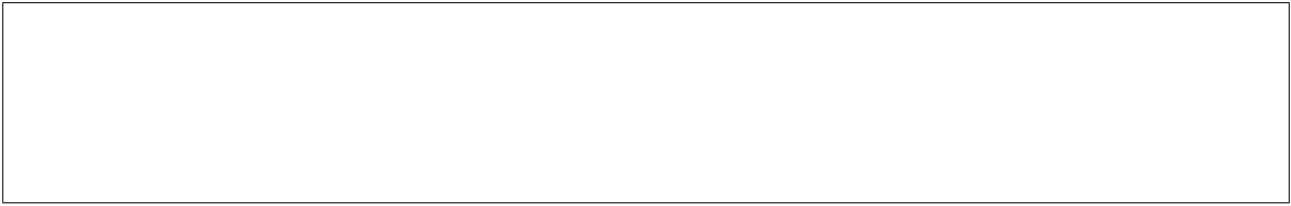
Exemple 063.217 (oct) = 000.110.011.010.001.111 (binaire).

10 Le premier modèle de PDP-10, le KA10 (1968), n'avait pas de pagination, et ses capacités de mémoire réelle et virtuelle étaient égales. Calculez cette capacité³.

Solution : Les adresses sont codées sur 18 bits, la capacité est de $2^{18} = 256K$ mots.

2. Bill Gates et Paul Allen ont développé leur premier Basic pour ALTAIR 8800 sur un simulateur qui tournait sur le PDP-10 de leur école.

3. Rappel : $2^8 = 256$, $2^{10} = 1K$, $2^{20} = 1M$, $2^{30} = 1G$



11 L'espace mémoire d'un processus comporte deux segments, appelés partie haute et partie basse, identifiés par le bit de poids fort de l'adresse virtuelle (bit à 1 pour la partie "haute").

17	16 15 ... 1 0
partie	déplacement

Indiquez si les adresses suivantes

- 752.676
- 012.543
- 454.210

sont en partie haute, ou en partie basse.

Comment reconnaître en, un coup d'oeil, dans une adresse écrite en octal, si elle appartient à la partie haute ou à la partie basse ?

Solution : Si le bit de poids fort, en binaire, est à 1, le chiffre de gauche, en octal, est 4, 5, 6, ou 7 (1xx en binaire). premier bit est à 1

12 Comment trouver rapidement le *déplacement* d'une adresse qui est en partie haute ? Donnez un exemple.

Solution : On soustrait 4 du chiffre gauche de l'adresse exprimée en octal. Exemple 765.432 donne 365.432.

13 Pour chaque partie, un registre de base + un registre limite servent à la *génération d'adresse*. On suppose que les registres contiennent les valeurs ci-dessous, données en octal :

	base	limite
partie basse	040.000	015.000
partie haute	300.000	200.000

Expliquez le rôle de ces registres, en prenant l'exemple de l'adresse logique 002.000.

Solution :

- L'adresse logique 002.000 commence par 0, elle est dans la partie basse de l'espace du processus
- elle est inférieure à 015.000, la valeur du registre limite, elle est donc correcte.
- l'adresse physique est obtenue en lui ajoutant le contenu du registre de base = $040.000 + 002.000 = 042.000$.



14 Calculez, quand c'est possible, les adresses physiques correspondant aux adresses logiques indiquées. Sinon, dites pourquoi. (Faites les calculs en octal directement).

adresse logique	haute / basse ?	déplacement	physique
012.345			
543.210			
654.321			

Solution :

- 012.345 est en partie basse (commence par 0), est inférieure au registre limite, et correspond à l'adresse physique $012.345 + 040.000 = 052.345$.
- 543.210 est en position 143.210 de la partie haute, adresse physique $143.210 + 030.000 = 173.210$.
- 654.321 est en position 254.321 de la partie haute, ce qui dépasse le contenu du registre limite : adresse invalide.

15 Le système d'exploitation TENEX (Bobrow, 1972) a été développé sur un PDP-10 spécialement modifié pour y ajouter un mécanisme de pagination. Quel est l'intérêt de la pagination ?

Solution :

- La gestion de la mémoire du PDP-10 "de base" passe par l'allocation contigüe, qui conduit à des problèmes de fragmentation.
- la pagination apporte une meilleure gestion de l'espace, et permet le "swapping" sur disque

16 La machine avait des pages de 512 mots, et la capacité de la mémoire physique était étendue à 1 méga mots (en tores de ferrite). Montrez où se trouve le numéro de page dans une adresse logique, en prenant l'exemple de 123.456.

Solution : Les pages comporte 512 mots, la position dans une page est codée sur 9 bits ($512 = 2^9$), et occupe donc les 3 derniers chiffres en octal.
Le numéro de page est donc codé sur les 3 premiers chiffres : 123 sur l'exemple.

17 Précisez la taille de la table des pages (nombre de cases, nombre de bits par case).

Solution :

- Le numéro de page est sur 9 bits : il y a 512 pages dans l'espace logique d'un processus.
- la mémoire est de 1 Mega mots, soit 2048 cadres de page de 512 mots. Un numéro de cadre de page est codé sur 11 bits.
- taille de la table : 512×11 bits.

4 RAID

On dispose d'un boîtier de 5 disques de 2 Go, on peut le configurer de diverses façons.

18 Qu'est-ce que le RAID 1? Quelle est la *capacité utile* du boîtier disque si on adopte cette configuration ?

Solution : Le boîtier forme un disque virtuel dont les blocs sont répartis sur les 5 disques. La capacité est de $5 \times 2 = 10Go$.

19 Même question avec une configuration en RAID 5.

Solution : Le boîtier forme un disque virtuel. Un bloc de parité pour chaque "rangée" permet d'assurer la reconstitution des données en cas de panne d'un disque. Par rapport au RAID 4, ces blocs de parité sont alternés pour répartir la charge. La capacité utile est donc de $(5 - 1) \times 2 = 8Go$.

20 Que se passe-t-il, dans chacun des cas, si un des disques tombe en panne ?

Solution : RAID 1 : On perd les données.

RAID 5 :

- on peut continuer à travailler : les données du disque défaillant peuvent être reconstituées à partir des informations redondantes.
- le fonctionnement est alors en mode dégradé : la fiabilité n'est plus garantie, puisque une autre panne entraîne la perte totale des données.
- on a donc intérêt à remplacer le disque défaillant au plus tôt et à la reconstruction du système de disques.