Nom :..... Groupe : .....

DS de Systèmes d'exploitation

durée : 1h50
aucun document autorisé

QCM (16 points)

Chaque question à choix multiples peut comporter **une ou plusieurs** bonne(s) réponse(s). 0,5 point par question, 0,5 pour la justification lorsqu'elle est demandée, pas de pénalisation des réponses fausses.

Système de fichier

On rappelle que les *i-nodes* sous BSD permettent d'adresser 12 blocs, 3 tables de niveau 1, 1 table de niveau 2 et 1 table de niveau 3. Chaque table a 1024 entrées, et occupe elle même 1 bloc.

---

1. Quelle est, en nombre de blocs, la quantité maximale des données que peut contenir un fichier ? (vous pouvez répondre par une formule plutôt que par un résultat numérique).

$12 + 3\times1024 + 1024^2 + 1024^3$

---

2. Quelle est le nombre total de blocs nécessaires pour stocker un fichier dont la taille est de 10 blocs ?

10

---

3. Quelle est le nombre total de blocs nécessaires pour stocker un fichier dont la taille est de 1025 blocs ?

$(12 + 1013) + 1$

---

4. Quelle est le nombre total de blocs nécessaires pour stocker un fichier dont la taille est de 4000 blocs ?

$(12 + 3\times1024 + 916) + 3 + 2$

---

5. Quel est le surcoût maximal du stockage avec ce système (sous la forme : nombre de blocs « accessoires » sur nombre de blocs « utiles ») ?

☐ 1/2 X 1/13 ☐ 1/1024 ☐ 1/1036

---

1. Un *i-node* contient un champs « type de fichier » ; quelle est la valeur que ce champs ne peut *pas* prendre ?

X lien physique ☐ lien symbolique ☐ périphérique ☐ répertoire

---

1. Justifiez

On appelle « lien physique » le fait d'avoir plusieurs noms pour le même i-node ; ceci est indépendant du type de fichier (qui peut, en théorie au moins, être n'importe lequel des types supportés : fichiers classique, répertoire, lien symbolique, périphérique…).

2. Par quelle(s) caractéristique(s) une clé USB se distingue-t-elle d'un disque dur ?

- ☐ accès séquentiel plus lent que l'accès aléatoire
- X nombre d'écritures limité
- X mode d'écriture différent pour des 0 et pour des 1
- ☐ nombre de lectures limité

Gestion de la mémoire

1. Lorsqu'un processus tente d'accéder à une page située dans l'espace d'échange, qui est responsable du chargement de cette page en mémoire ?

- ☐ le MMU
- X le système d'exploitation
- ☐ le processus
- ☐ le système de fichiers

On considère dans les questions 10 à 16 un processus dont la table des pages, à un instant donné, est représentée ci-contre. Dans la suite, on supposera que les remplacements de page se font uniquement entre pages du même processus, et que la mémoire est saturée, donc que le chargement d'une page en mémoire s'accompagne forcément du remplacement d'une autre page.

On indique que la séquence des accès mémoire jusqu'à maintenant a été 0,1,2,3,0,4,5,0. A l'instant considéré, le programme demande à écrire dans la page 1.

n° de page	n° de bloc	r	w	x	d	a
0	245	1	0	1	0	1
1	swap	1	1	0	0	0
2	swap	1	1	0	0	0
3	248	1	1	0	0	0
4	249	1	1	0	1	0
5	252	1	1	0	0	1

1. En appliquant la politique NRU (Not Recently Used), laquelle des pages doit être mise en espace d'échange ?

☐ 0 X 3 ☐ 4 ☐ 5

1. En appliquant la politique FIFO (*First In First Out*), laquelle des pages doit être mise en espace d'échange ?

X 0 ☐ 3 ☐ 4 ☐ 5

1. En appliquant la politique optimale, et en supposant que les prochains accès seront 0,5,2,3,4, laquelle des pages doit être mise en espace d'échange ?

☐ 0 ☐ 3 X 4 ☐ 5

1. Pour laquelle des page la mise en espace d'échange va-t-elle prendre le plus de temps ?

☐ 0 ☐ 3 X 4 ☐ 5

1. Justifiez.

Cette page a le bit dirty à 1, ce qui signifie qu'elle doit d'abord être ré-écrite dans l'espace d'échange avant d'être retirée de la mémoire.

2. Après le remplacement de page et l'écriture dans la page 1, les bits *d* et *a* de l'entrée 1 dans la table des pages auront pour valeur :

☐ d=0, a=0 ☐ d=0, a=1 ☐ d=1, a=0 X d=1, a=1

1. En supposant que les pages ont une taille de 100 octets, quelle est, pour le processus considéré, l'adresse physique correspondant à l'adresse logique 406 ?

☐ 255 ☐ 2406 ☐ 2496 X 24906

Processus et ordonnancement

1. Juste après un *fork*, quel(s) élément(s) suivant(s) peuvent différer entre le processus père et le processus fils ?

☐ les fichiers ouverts ☐ le programme exécuté X le PID X les variables locales

1. Juste après un appel à *exec*, quel(s) élément(s) suivant(s) peuvent changer pour le processus ?

☐ les fichiers ouverts X le programme exécuté ☐ le PID X les variables locales

1. Dans quel(s) état(s) peut se retrouver un processus juste après une préemption ?

☐ bloqué X éligible ☐ élu

1. Dans ordonnanceur *Multilevel Feedback*...

☐ le quantum des files moins prioritaires peut être plus court

X le quantum des files moins prioritaires peut être plus long

X le quantum de toutes les files peut être identique

☐ le quantum peut varier entre des processus d'une même file

1. Dans un ordonnanceur *Multilevel Feedback,* dans quelle condition un processus va-t-il descendre d'un niveau ?

☐ X il consomme la totalité de son quantum ☐ il y a trop de processus à son niveau

☐ il ne consomme pas la totalité de son quantum ☐ il y a trop peu de processus au niveau inférieur

1. Un système d'exploitation « temps réel »... *(une seule réponse)*

☐ exécute ses appels systèmes de manière asynchrone ☐ permet de connaître à l'avance le temps de réponse des appels système

X garanti le temps de réponse des appels système ☐ minimise le temps de réponse des appels système

1. Sous Linux, les « processus temps réel » sont... *(une seule réponse)*

☐ ceux ayant la priorité externe (*nice*) la plus élevée X considérés avant les autres par l'ordonnanceur

☐ gérés avec une politique d'ordonnancement « temps réel » ☐ uniquement présents dans la variante « temps réel » de Linux

Entrées-sorties et IPC

1. À propos des descripteurs de fichier standard :

X le descripteur 0 peut être ouvert en lecture

X le descripteur 0 peut être ouvert en écriture

X le descripteur 1 peut être ouvert en lecture

X le descripteur 1 peut être ouvert en écriture

1. Justifiez

Par convention, ils sont généralement pré-ouverts, respectivement en lecture (entrée standard) et en écriture (sortie standard), mais ce n'est qu'une convention. Ces fichiers peuvent être fermés et les descripteurs 0 et 1 réutilisés pour n'importe quel fichier.

2. Laquelle ou lesquelles de ces affirmations sont vraies pour un tube (*pipe*) ?

☐ Il permet une communication bidirectionnelle

☐ Il permet de faire communiquer des processus de machines différentes

X Il peut être matérialisé dans le système de fichiers

X Il peut être créé directement en ligne de commande entre deux processus

1. Laquelle ou lesquelles de ces affirmations sont vraies pour un *socket* ?

X Il permet une communication bidirectionnelle

X Il permet de faire communiquer des processus de machines différentes

X Il peut être matérialisé dans le système de fichiers

☐ Il peut être créé directement en ligne de commande entre deux processus

1. Laquelle de ces affirmation décrit le mieux la relation entre un signal et une interruption ?

☐ une interruption est une sorte de signal

☐ tous les signaux sont la manifestation logicielle d'une interruption matérielle

☐ un signal est une sorte d'interruption

☐ X certains signaux sont la manifestation logicielle d'une interruption matérielle

Synchronisation et parallélisme

1. Dans laquelle de ces situtations un processus peut-il se bloquer ?

☐ lorsqu'il crée un sémaphore

☐ lorsqu'il pose un jeton dans un sémaphore plein

☐ lorsqu'il pose un jeton dans un sémaphore sur lequel un autre processus est en attente

X lorsqu'il retire un jeton dans un sémaphore vide

1. Dans le problème des lecteurs-rédacteurs :

X deux lecteurs peuvent travailler en même temps

☐ deux rédacteurs peuvent travailler en même temps

☐ un lecteur et un rédacteur peuvent travailler en même temps

☐ un seul processus peut travailler à un moment donné

1. Dans le problème des producteur-consommateur :

   X deux consommateurs peuvent travailler en même temps

   X deux producteurs peuvent travailler en même temps

   X un producteur et un consommateur peuvent travailler en même temps

   ☐ un seul processus peut travailler à un moment donné

---

1. Dans le problème des philosophes :

   X deux voisins peuvent penser en même temps

   ☐ deux voisins peuvent manger en même temps

   X deux voisins peuvent avoir faim en même temps

   ☐ tous les philosophes peuvent avoir faim en même temps

Questions de cours (4 points)

Qu'est-ce qu'un inter-blocage ? Donnez un exemple.

C'est la situation dans laquelle plusieurs processus sont tous bloqués, mais ou l'événement qui pourrait débloquer chacun d'eux doit être produit par l'un des processus bloqués. Ainsi, cette situation ne peut pas progresser.

Exemple : dans une situation avec deux mutex A et B, lorsqu'un processus possède le mutex A et attend le mutex B, alors que l'autre possède le mutex B et attend le mutex A.

Autre exemple : dans une approche naïve du problème des philosophes, lorsque chaque philosophe prend la baguette de gauche, et attend la baguette de droite.

Qu'est-ce qu'une situation de famine ? Donnez un exemple.

C'est une situation dans laquelle un processus reste indéfiniment bloqué, généralement en attente d'un sémaphore. Cela peut se produire lorsqu'un autre processus a pris un jeton et ne le rend pas. Mais cela peut aussi se produire lorsque plusieurs processus se « passent le relais » (exemples dans le problème des lecteurs-rédacteurs ou des philosophes).