

IF-3-SYS : Systèmes d'exploitation. 5 juin 2018

NOM Prénom :

Consignes

- L'examen dure 1h30 et ce sujet comporte 8 pages.
- Répondez directement sur le sujet. Commencez par écrire votre nom ci-dessus.
- Écrivez lisiblement et surtout sans ratures. Utilisez un brouillon (vraiment).
- Le barème donné entre parenthèses a uniquement une valeur indicative.
- Tous documents papier autorisés. Pas de téléphone ni appareil quelconque.
- Pour les calculs en binaire, aidez-vous du tableau page 8.

1 Questions de cours

Dans les questions à choix multiples, entourez V pour dire «vrai», ou F pour dire «faux». Si vous ne savez pas répondre, il est plus rentable de n'entourer ni V ni F plutôt que de choisir au hasard.

Question 1 (1pt) Pour chaque affirmation ci-dessous, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F Les appels système disponibles dépendent du langage de programmation utilisé.
- ☐ V ☐ F Les appels système disponibles dépendent du shell utilisé.
- ☐ V ☐ F Les appels système disponibles sont chacun identifié par un numéro unique.
- ☐ V ☐ F Les appels système disponibles sont les mêmes sous Ubuntu, Fedora, et Android.

Question 2 (1pt) Pour chaque action ci-dessous, indiquez s'il s'agit d'une action autorisée pour un programme userland (entourez V) ou bien s'il s'agit d'une action privilégiée, c.à.d. réservée au noyau (entourez F).

- ☐ V ☐ F modifier arbitrairement le registre PC (Program Counter)
- ☐ V ☐ F modifier arbitrairement le registre SP (Stack Pointer)
- ☐ V ☐ F modifier arbitrairement le contenu de la pile
- ☐ V ☐ F modifier arbitrairement le contenu de la table des pages

Question 3 (1pt) Pour chaque affirmation ci-dessous, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F Dans un système multitâche, le noyau change de processus actif avec une période de quelques millisecondes.
- ☐ V ☐ F Pendant que le noyau traite un appel système dit «non bloquant», le processus correspondant continue de s'exécuter.
- ☐ V ☐ F Pour implémenter la préemption, il faut un composant matériel qui nous envoie des interruptions.
- ☐ V ☐ F Vis-à-vis de l'ordonnanceur, un processus attendant la fin d'une lecture sur disque et un processus attendant la libération d'un mutex, sont dans le même «état».

Question 4 (1pt) Pour chaque affirmation ci-dessous, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F Différents threads du même processus ont la même pile d'exécution.
- ☐ V ☐ F Différents threads du même processus ont la même table de pages.
- ☐ V ☐ F Différents threads du même processus ont le même CPU.
- ☐ V ☐ F Différents threads du même processus ont les mêmes sections critiques.

Question 5 (1pt) Pour chaque affirmation ci-dessous, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F En général, l'espace d'adressage virtuel est plus petit que la mémoire physique.
- ☐ V ☐ F L'adresse physique d'une page virtuelle est toujours la même durant l'exécution du programme.
- ☐ V ☐ F Grâce à la mémoire virtuelle, les accès à la mémoire principale sont, en moyenne, plus rapides.
- ☐ V ☐ F La taille des pages virtuelles est toujours la même que la taille des pages physiques.

Question 6 (1pt) On considère une machine avec des adresses virtuelles sur 32 bits et des pages de mémoire virtuelle de 1024 octets. Sur combien de bits est codée la partie *offset* d'une adresse virtuelle ? Répondez en décimal.

Question 7 (1pt) On veut copier dans la variable «b» la quatrième valeur du tableau «a». Complétez la dernière ligne du programme ci-dessous. Vous n'avez pas le droit d'utiliser l'opérateur d'indexation «[]».

```
int a [10] = { ... };
int b;
int *ptr = &a;

b =
```

Question 8 (1pt) Les affirmations ci-dessous se rapportent à la stratégie d'allocation *first-fit*. Pour chacune d'entre elles, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F Cette stratégie n'a typiquement pas besoin de considérer l'intégralité de la freelist.
- ☐ V ☐ F Cette stratégie n'est jamais victime du problème de fragmentation du tas.
- ☐ V ☐ F Cette stratégie nécessite d'initialiser le tas avec des blocs libres de tailles bien choisies.
- ☐ V ☐ F Cette stratégie cherche à exploiter en priorité les blocs libres les plus petits.

Question 9 (1pt) Dans le programme ci-dessous, la fonction `ajouter()` peut être exécutée de façon concurrente par de nombreux threads, ce qui pose un problème de synchronisation des accès. Corrigez ce problème : il s'agit d'une part de décider la valeur initiale du sémaphore S (cadre de gauche) et d'autre part d'invoquer judicieusement P(S) et/ou V(S) dans le code (cadre de droite).

Conditions initiales

```
int value = 0 ;

semaphore S =
```

```
void ajouter(int delta)
{

    value = value + delta;

}
```

Question 10 (1pt) Les affirmations ci-dessous se rapportent à la fonction `sem_post()`. Pour chacune d'entre elles, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F Cette fonction incrémente la valeur du sémaphore.
☐ V ☐ F Cette fonction est typiquement bloquante pour le thread appelant.
☐ V ☐ F Cette fonction doit être appelée uniquement dans une section critique.
☐ V ☐ F Cette fonction est typiquement implémentée par un appel système.

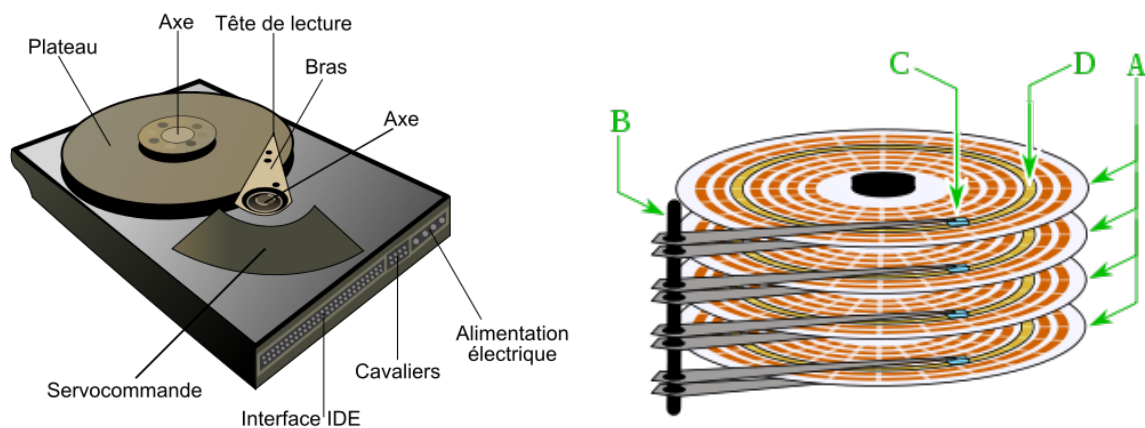
Question 11 (1pt) Quel est le nom de la commande shell permettant de créer un nouveau dossier ?

Question 12 (1pt) Pour chacune des affirmations ci-dessous, indiquez si elle est vraie ou fausse.

- ☐ V ☐ F L'appel système `copy()` permet de copier un fichier d'un dossier à un autre.
☐ V ☐ F L'approche FAT permet d'accéder rapidement à n'importe quel endroit à l'intérieur d'un fichier.
☐ V ☐ F Le format des données stockées dans un fichier est indiqué dans l'i-node correspondant.
☐ V ☐ F Sous Unix, le dossier racine d'un volume s'appelle toujours « / ».

2 Problème : ordonnancement des requêtes sur le disque dur

Dans cette partie, on s'intéresse au fonctionnement d'un disque dur magnétique. Les éléments principaux sont illustrés ci-dessous (images tirées de wikipedia). Le disque dur est un ensemble de *plateaux* métalliques (repère A) tournant à vitesse constante autour d'un axe fixe. Chaque plateau est divisé en *pistes* circulaires (repère D), elles-mêmes divisées en secteurs. Les entrées-sorties se font par l'intermédiaire d'une *tête de lecture* magnétique (repère C) située à l'extrémité d'un *bras mécanique* (repère B) : pour lire/écrire un secteur, on doit positionner le bras sur la bonne piste puis attendre que le secteur voulu passe sous la tête de lecture.



Pour simplifier, on se limitera au cas d'un unique plateau composé de N pistes numérotées de 0 à $N-1$. Le déplacement de la tête se fait à la vitesse d'une piste par une unité de temps. Par ailleurs, on négligera la durée du transfert de données : une fois la tête arrivée sur la piste souhaitée, on considère que le bras peut repartir immédiatement.

Question 14 (2pts) La stratégie précédente n'est clairement pas efficace : il serait plus malin de visiter en premier les pistes les plus proches. Appelons cette nouvelle stratégie PPDA pour «Plus Proche D'Abord». Autrement dit : à chaque fois que le bras est libre, il choisit parmi les requêtes en attente celle qui lui demande le moins de trajet. Décrivez ci-dessous les déplacements de la tête dans ces conditions.

instant	position de la tête	requêtes en attente
0	0	73, 105, 21

Question 15 (1pt) Sur notre exemple, cette nouvelle stratégie PPDA obtient de bien meilleures performances que la stratégie naïve PAPS. Pourtant dans la vraie vie, aucun contrôleur de disque n'implémente PPDA car cette stratégie souffre d'un défaut majeur. De quel problème s'agit-il ? Quelles en sont les causes et les conséquences ? Lisez la question suivante *avant* de répondre à celle-ci.

Question 16 (1pt) Proposez ci-dessous un scénario illustrant le problème identifié à la question précédente.

Position de la tête à $t=0$:

--

Requêtes reçues :

[illegible]

Question 17 (2pts) Pour échapper au problème associé à la stratégie PPDA, on propose une nouvelle stratégie baptisée «Va-et-Vient» ou V&V. L'idée est de choisir une direction (par exemple, vers les n^{os} de piste croissants) et d'avancer uniquement dans cette direction. Chaque fois qu'on passe par une piste pour laquelle il y a une requête en attente, on traite cette requête. Ce balayage dure aussi longtemps qu'il reste des requêtes en attente «devant nous». Dans le cas contraire, on inverse la direction du mouvement pour reprendre le balayage en sens opposé.

Dans le tableau ci-dessous, décrivez les déplacements de la tête selon la stratégie «Va-et-Vient», toujours pour notre même scénario d'étude.

[illegible]

Question 18 (2pts) Dans notre scénario d'étude, la stratégie V&V donne de meilleures performances que PPDA. Mais ce n'est pas toujours vrai dans le cas général. Dans le tableau ci-dessous, donnez un scénario où PPDA va plus vite que V&V.

Position de la tête à $t=0$:

Requêtes reçues :

instant d'arrivée	numéros de piste demandés
0	

Question 19 (2pts) Le réel avantage de la stratégie V&V sur PPDA n'est pas la performance mais l'existence de garanties sur la qualité de service. On appelle *temps de réponse* la durée qui s'écoule entre la réception d'une requête et l'instant où la tête de lecture arrive sur la piste demandée. Pour un disque appliquant la stratégie V&V, quel est le pire temps de réponse que peut subir une requête quelconque ? Justifiez votre réponse.

Annexe : quelques puissances de 2

$2^0 = 1$	$2^{16} = 65\,536$	$2^{32} = 4\,294\,967\,296$	$2^{48} = 281\,474\,976\,710\,656$
$2^1 = 2$	$2^{17} = 131\,072$	$2^{33} = 8\,589\,934\,592$	$2^{49} = 562\,949\,953\,421\,312$
$2^2 = 4$	$2^{18} = 262\,144$	$2^{34} = 17\,179\,869\,184$	$2^{50} = 1\,125\,899\,906\,842\,624$
$2^3 = 8$	$2^{19} = 524\,288$	$2^{35} = 34\,359\,738\,368$	$2^{51} = 2\,251\,799\,813\,685\,248$
$2^4 = 16$	$2^{20} = 1\,048\,576$	$2^{36} = 68\,719\,476\,736$	$2^{52} = 4\,503\,599\,627\,370\,496$
$2^5 = 32$	$2^{21} = 2\,097\,152$	$2^{37} = 137\,438\,953\,472$	$2^{53} = 9\,007\,199\,254\,740\,992$
$2^6 = 64$	$2^{22} = 4\,194\,304$	$2^{38} = 274\,877\,906\,944$	$2^{54} = 18\,014\,398\,509\,481\,984$
$2^7 = 128$	$2^{23} = 8\,388\,608$	$2^{39} = 549\,755\,813\,888$	$2^{55} = 36\,028\,797\,018\,963\,968$
$2^8 = 256$	$2^{24} = 16\,777\,216$	$2^{40} = 1\,099\,511\,627\,776$	$2^{56} = 72\,057\,594\,037\,927\,936$
$2^9 = 512$	$2^{25} = 33\,554\,432$	$2^{41} = 2\,199\,023\,255\,552$	$2^{57} = 144\,115\,188\,075\,855\,488$
$2^{10} = 1\,024$	$2^{26} = 67\,108\,864$	$2^{42} = 4\,398\,046\,511\,104$	$2^{58} = 288\,230\,376\,151\,711\,744$
$2^{11} = 2\,048$	$2^{27} = 134\,217\,728$	$2^{43} = 8\,796\,093\,022\,208$	$2^{59} = 576\,460\,752\,303\,423\,488$
$2^{12} = 4\,096$	$2^{28} = 268\,435\,456$	$2^{44} = 17\,592\,186\,044\,416$	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
$2^{13} = 8\,192$	$2^{29} = 536\,870\,912$	$2^{45} = 35\,184\,372\,088\,832$	$2^{61} = 2\,305\,843\,009\,213\,693\,952$
$2^{14} = 16\,384$	$2^{30} = 1\,073\,741\,824$	$2^{46} = 70\,368\,744\,177\,664$	$2^{62} = 4\,611\,686\,018\,427\,387\,904$
$2^{15} = 32\,768$	$2^{31} = 2\,147\,483\,648$	$2^{47} = 140\,737\,488\,355\,328$	$2^{63} = 9\,223\,372\,036\,854\,775\,808$
			$2^{64} = 18\,446\,744\,073\,709\,551\,616$

Attention, en fonction du contexte, les lettres k, M, G ont souvent une signification différente. Par exemple, 1MB «un méga-octet» représente typiquement 2^{20} octets, alors que 1MHz «un mégahertz» représente 10^6 hertz. Dans le doute, demandez des précisions.