UPMC
SORBONNE UNIVERSITÉS

NMV - NI436

Examen du 14 Novembre 2011

durée 2h - **Documents papier autorisés**
Téléphones portables, baladeurs et autres appareils électroniques doivent être éteints
Le barème n'est donné qu'à titre indicatif, pour vous permettre de juger de la difficulté des questions.
Attention l'énoncé est imprimé recto-verso sur 1 feuille

Marc Shapiro, Julien Sopena et Gael Thomas.

Vous devez rendre trois feuilles séparées contenant respectivement les exercices : 1 / 2 / 3 et 4.

Exercice 1 : Algorithmes concurrents à grain fin - 7 points

Question 1 – 1 point

Donner l'interface de la primitive CAS. Que fait-elle ? Comment s'en sert-on dans un programme ? Quels sont ses avantages et inconvénients ?

Question 2 – 1 point

Spécifiez une primitive "double CAS" faisant deux CAS de façon atomique (sous forme de pseudo-code ou d'un autre formalisme de votre choix, mais sans faire appel à CAS).

Question 3 – 2 points

Dans une liste chaînée à grain fin *HandOverHandList* il nécessaire de prendre deux verrous. Donnez un exemple d'exécution erronée de *add* et un exemple d'exécution erronée de *remove*, si le programme ne prenait qu'un verrou sur un seul des éléments. Quel invariant (ou post-condition) est violé dans chacun de vos deux exemples ?

Question 4 – 3 points

Supposons que vous ayez à votre disposition la primitive Double-CAS de la question 2. Est-il possible de modifier *HandOverHandList* afin de remplacer le verrouillage par l'utilisation de Double-CAS, sans utiliser de bit de marquage ? Si oui, (i) donnez le pseudo-code de *add* et *remove*, (ii) montrez que les deux exécutions erronées de la question 3 ne se produisent pas. Dans le cas contraire, expliquez pourquoi.

Exercice 2 : Noyau linux - 8 Points

Question 1 – 1 point

Pour quelle(s) raison(s) Linux conserve toujours un certain nombre de structures *file* inutilisées ?

Question 2 – 1 point

Que doit vérifier le noyau avant de supprimer une structure *inode* ?

Question 3 – 2 points

On considère un système ayant une partition racine / et une partition */home*. Sachant qu'un lien symbolique n'est qu'un fichier contenant un raccourci vers un autre fichier, dessinez le graphe des structures du *VFS*, au moment où la commande suivante se termine :

```
ln -s /etc/rc.conf /home/sopena/rc.conf
```

Question 4 – 2 points

On s'intéresse maintenant à la création de lien *physique*. En dessinant le graphe des structures du *VFS* tel qu'il aurait du être, montrez pourquoi il est impossible sous Linux d'exécuter la commande suivante :

```
ln /etc/rc.conf /home/sopena/rc.conf
ln: impossible de créer le lien direct « /home/sopena/rc.conf » => « /etc/rc.conf »
```

Question 5 – 2 points

Implémenter un module qui affiche, dans un fichier du *sysfs*, le nombre de modules chargés en mémoire. Vous pourrez, si besoin, utiliser la structure suivante.

```
struct list_head {
  struct list_head *next, *prev;
};
```

Exercice 3 : Moniteurs de Machines Virtuelles - 3 Points

Question 1 – 1 point

Expliquez en quelques lignes pourquoi un moniteur de machine virtuelle doit maintenir une table des pages ombre.

Question 2 – 2 points

On suppose un Moniteur de type II s'exécutant au dessus de Linux et exécutant un Linux. Les deux Linux sont configurés de la même façon et sont donc chargés à la même adresse 0xc700'000. Expliquez comment le moniteur peut éviter les collisions d'adresses.

Exercice 4 : Mémoire transactionnelle - 4 Points

Dans une mémoire transactionnelle, on suppose deux variables `var1` et `var2` initialisées à 0 et deux threads exécutant les codes suivants.

Thread 1 :

```
a. printf("1.a\n");
b. atomic {
c.   while(var1 == 0) retry;
d.   var2 = 1;
e. }
f. printf("1.f\n");
```

Thread 2 :

```
g. printf("2.g\n");
h. var1 = 1;
i. while(var2 == 0) ;
j. printf("2.j\n");
```

Question 1 – 2 points

Donnez, de façon exhaustive, les affichages possibles avec une mémoire transactionnelle retardée ? Vous expliquerez à chaque fois votre scénario.

Question 2 – 2 points

Donnez, de façon exhaustive, les affichages possibles avec une mémoire transactionnelle immédiate ? Vous expliquerez à chaque fois votre scénario.

