M2 Informatique Embarquée

Examen du 24 Mars 2010 16H00-18H00

Documents autorisés

Total sur 40 points. Questions indépendantes les unes des autres.

Regardez le barème avant de vous lancer.

1. Donnez une définition d'un système embarqué. En quoi un système embarqué diffère-t-il d'un système de type « poste de travail » ou d'un système de type « serveur pour centre de calcul »? Tentez d'être aussi exhaustif que possible dans la liste des critères différenciant ces types de système. D'après vous,ces distinctions vont-elles s'amplifier ou s'estomper dans l'avenir? Motivez votre réponse. Donnez 3 exemples de systèmes embarqués. *(4 points)*
2. Listez le plus exhaustivement possible les éléments qui interviennent dans le temps d'exécution d'un programme, en expliquant pour chacun d'eux quelle est leur influence. *(3 points)*
3. Est-il possible d'exécuter un programme binaire exécutable généré pour une machine / système Intel / Linux sur une machine système ARM / Linux? Pourquoi? Peut-on de **manière certaine** exécuter un programme binaire exécutable généré pour une machine / système Intel / Linux sur une autre machine / système Intel / Linux, par exemple passer un programme d'un système Red Hat à un système Debian? Pourquoi? Dans les cas où cette garantie est offerte qu'est-ce qui permet cette garantie? Dans les cas où cette garantie n'est pas offerte, a-t-on une solution de recours? *(3 points)*
4. Un processeur ARM7 ne dispose pas de MMU. Quels impacts cela a-t-il sur les environnements Linux tournant sur ce type de processeur? Comment se comportera l'extrait de programme ci-dessous dans cet environnement? Modifiez-le pour qu'il se comporte de la même manière que sur un processeur Intel x86. *(4 points)*

```
static int mypid;
...
     res = fork();
     mypid = getpid();
     if (res ==0) {
          printf("je suis le proc %d\n", mypid);
          execlp("toto",....);
     }
     printf("je suis le proc %d\n", mypid);
...
```

5. Quelle est la différence entre des bibliothèques (librairies) partagées et des bibliothèques (librairies) statiques? Expliquez quelle influence ces 2 types de bibliothèques ont sur la taille des binaires exécutables dans le système de fichiers et en mémoire. *(4 points)*
6. Quelles sont les différentes étapes intervenant dans la génération d'un binaire exécutable à partir des fichiers source toto.c et toto.h. Quelle est la différence entre un binaire relogeable et un binaire exécutable? Décrivez brièvement ce qu'est une chaîne de compilation croisée et en quoi elle diffère d'une chaîne de production native? *(3 points)*
7. Que se passe-t-il lors de l'exécution du code suivant sur un système n'ayant pas de mécanisme de détection de dépendance circulaire: *(4 points)*
   - 3 threads T1, T2 et T3 soumises à un ordonnancement de type FIFO avec la relation suivante

entre leurs priorités : Prio(T1) < Prio(T2) < Prio(T3)
- 2 mutex m1 et m2 correctement initialisés sans attribut particulier et pas encore utilisés

7.1. T1 exécute l'appel : `pthread_mutex_lock(&m1);`
7.2. T2 préempte T1
7.3. T2 exécute l'appel : `pthread_mutex_lock(&m2);`
7.4. T3 préempte T2
7.5. T3 invoque la fonction : `pthread_mutex_lock(&m2);`
7.6. T2 invoque la fonction : `pthread_mutex_lock(&m1);`
7.7. T1 invoque la fonction : `pthread_mutex_lock(&m2);`

Que se passe-t-il lors des étapes suivantes: 7.5., 7.6., 7.7. ? Pourquoi a-t-on ce comportement et cet ordre d'exécution ? Soyez clair et précis. Si nécessaire, faites une représentation graphique pour illustrer vos explications. Comment éviteriez-vous ce genre de situation ? Que se passerait-il si au lieu d'être soumises à un ordonnancement de type FIFO, ces threads étaient soumises à un ordonnancement de type temps-partagé?

8. Soit le cas de figure suivant : *(3 points)*
   - 3 threads T1, T2 et T3 soumises à un ordonnancement de type FIFO avec la relation suivante entre leurs priorités: Prio(T1) < Prio(T2) < Prio(T3)
   - 1 mutex M1

   a) T1 prend le mutex M1
   b) T3 préempte T1
   c) T3 est bloquée en prenant le mutex M1
   d) T2 préempte T1

   L'utilisation des mutex à héritage de priorité permettrait de résoudre le problème d'inversion de priorités ci-dessus en :

   8.1. attribuant à T1 la priorité de T3 à l'étape b)

   8.2. attribuant à T1 la priorité de T3 à l'étape c)

   8.3. attribuant à T2 la priorité de T3 à l'étape d)

   Décrivez (avec support graphique si nécessaire) le problème ainsi évité.

9. Quelle évolution la loi de Moore a-t-elle récemment suivi? Quel(s) problème(s) cela pose-t-il aux logiciels existants pour systèmes temps-réels? *(4 points)*

10. Quel est le rôle d'un ordonnanceur? Dessinez le diagramme d'état (et les transitions) d'une entité gérée par un ordonnanceur. Expliquez ce qui provoque les transitions d'un état à un autre. Décrivez brièvement ce que sont une politique d'ordonnancement de type FIFO et une politique de type EDF. Donnez les significations de ces sigles. *(4 points)*

11. Rappelez brièvement les grandes catégories de virtualisation de système, pour y classer les systèmes suivants. Parmi ces systèmes lesquels peut-on espérer utiliser dans des systèmes embarqués d'une part, et pour supporter des applications ayant des besoins temps-réel d'autre part. *(4 points)*
    - VmWare ESX, VmWare Workstation, KVM, Xen
    - VLX, QEMU, Adeos/Xenomai