Noyaux et Systèmes d'Exploitation

Durée: 3 heures
Documents et calculatrices interdits
Les résultats des calculs sous forme non-développée sont admis.
Les questions bonus donnent des points sommés à la note finale du partiel.
Une copie bien présentée sera toujours mieux notée qu'une autre.

I - Architecture (8 points)

Face à l'euphorie de la communauté devant linux, A. Tanenbaum écrit en 1992 que "linux est obsolète". L. Torvalds intervient alors et marque le début d'un long débat à propos de l'architecture des OS: monolithique pour linux (Torvalds), micro-noyau pour minix (Tanenbaum). Tanenbaum affirme que les "micro-noyaux ont gagné" face aux vieux noyaux monolithiques, or linux réintroduit une architecture monolithique.

1) (4 points) Quels sont les arguments de A. Tanenbaum en faveur des architectures micro-noyau?
2) (4 points) Quels sont les arguments de L. Torvalds en faveur des architectures monolithiques?

II - Boot (14 points)

1) Firmware
   (a) (1 point) Quel est le rôle d'un firmware?
   (b) (1 point) Pourquoi les firmwares sont-ils en ROM (Read-Only Memory)?
   (c) (1 point) Quelle est généralement la premère tâche réalisée par un firmware en ROM?
   (d) Le BIOS fait office de firmware sur les ordinateurs IBM-PC.
      α) (1 point) Comment s'interface un bootloader avec le BIOS?
      β) (4 points) Donner deux assertions (contrat d'interface) nécessaires "en sortie" du BIOS pour qu'un bootloader tel que GRUB puisse correctement s'exécuter?
2) Bootloader
   (a) (1 point) Quel est le rôle d'un bootloader?
   (b) (1 point) Pourquoi les bootloaders historiques des ordinateurs IBM-PC se décomposent-ils en plusieurs binaires?
   (c) (4 points) Proposer une interface (control-flow ou data-flow ou contrat) bootloader/OS sur laquelle pourrait se reposer votre OS.
3) (1 point **bonus**) Soit un système pouvant démarrer sous différents modes, maintenance et opérationnel, en fonction de la position d'un interrupteur. Proposer et justifier les interfaces firmware/bootloader et bootloader/OS permettant de réaliser ce système. Chaque mode à son propre binaire en ROM. Le mode maintenance permet de télécharger un nouveau binaire pour le mode opérationnel.

III - Interruptions (7 points)

1) (1 point) Donner une solution apportée par les interruptions.
2) (a) (2 points) Pourquoi cherche-t-on à limiter le nombre d'interruptions et leurs durées de traitement?
   (b) (2 points) En quoi cela améliore-t-il les performances de manière significative, notamment en cas de charge importante? Pour ce faire, vous pouvez détailler le traitement d'arrivée d'un paquet réseau.
3) (2 points) Pourquoi le top d'horloge (tick) d'un système général est particulièrement adapté au mode "edge-triggered"?
4) (1 point **bonus**) Comment sont gérées les interruptions dans un contexte multi-coeur?

IV - Mémoire (16 points)

1) (2 points) Donner deux différences entre la ségmentation et la pagination.
2) (1 point) Quel est l'inconvénient de la technique de mirroring?
3) (4 points) Proposer une alternative au mirroring sans en perdre l'avantage majeur.

4) L'un des avantages des processus légers (threads) est qu'ils évitent de vider la TLB lors d'un changement de contexte entre des processus légers d'un même processus.
   (a) (1 point) Comment est indexée une telle TLB (se vidant lors d'un changement de contexte de processus) ?
   (b) (2 points) Proposez une modification simple à cette indexation afin d'éviter la vidange de la TLB en cas de changement de contexte entre processus.
5) Sur une architecture ayant des adresses virtuelles de 32 bits, **dessiner** les mappings (association pages virtuelles/pages physique) permettant:
   (a) (2 points) au kernel de s'exécuter dans un mode d'adressage réel 16 bits.
   (b) (2 points) à deux processus de partager un buffer.
   (c) (2 points) d'utiliser la technique de "copy-on-write" après un appel système 'fork'.
6) (1 point **bonus**) Comment permettre le partage de code (bibliothèque partagée) avec la ségmentation?

V - Multi-tâche & Ordonnancement (10 points)

1) (a) (2 points) Comment sont catégorisées les tâches que l'on retrouve dans les systèmes généraux? Quelles sont leur caractéristique?
   (b) (2 points) Quels sont leur objectif?
   (c) (2 points) Proposer une heuristique simple permettant de déterminer à quelle catégorie appartient un processus.
2) Les ordonnanceurs circulaires mémorisent la liste des processus prêts et chaque processus n'apparaît qu'une seule fois dans cette liste.
   (a) (1 point) Que se passerait-il si un processus y apparaissait deux fois?
   (b) (1 point) Pourquoi permettrait-on l'insertion multiple d'un processus dans cette liste?
3) (2 points) La plupart des ordonnanceurs circulaires utilisent un quantum fixe. Donner un argument en faveur d'un petit quantum et un autre en faveur d'un grand quantum, ainsi que la catégorie de processus favorisé.

VI - Concurrence (15 points)

1) (a) (1 point) Qu'est-ce qu'un changement de contexte matériel (hardware context switch)?
   (b) (1 point) Qu'est-ce qu'un changement de contexte logiciel (software context switch)?
   (c) (2 points) Quels sont leurs avantages et inconvénients?
2) (a) (1 point) En quoi consiste l'implémentation d'une bibliothèque de processus légers en mode utilisateur (user-mode threads)?
   (b) (1 point) En quoi consiste l'implémentation d'une bibliothèque de processus légers en mode noyau (kernel-mode threads)?
   (c) (4 points) Quels sont leurs avantages et inconvénients?
   (d) (1 point) Que sont les activations d'ordonnanceur (scheduler activations)?
3) (a) (2 points) Quelles sont les particularités des deux principales architectures logicielles multi-coeur: SMP et NUMA?
   (b) (2 points) Citer quelques éléments à prendre en compte lors de l'implémentation d'un ordonnanceur sur ces deux architectures.
4) (1 point **bonus**) Dans un processeur moderne, il existe plusieurs niveaux de caches. Comment sont-ils gérés dans un contexte multi-coeur?

VII - Virtualisation (9 points)

1) (1 point) En quoi consiste la translation dynamique de code?
2) La virtualisation a pour philosophie d'intercepter toute exception et instruction privilégiée en forçant une transition vers l'hyperviseur.
   (a) Dans le cas **sans** support matériel de virtualisation:
      α) (1 point) Donner un exemple d'instruction privilégiée à intercepter.
      β) (2 points) Dessiner et expliquer brièvement le flot de contrôle (control-flow) d'un appel système fait par une application s'exécutant sur l'OS guest.
      γ) (1 point) Quel est l'inconvénient majeur de ce mode d'hypervision?
   (b) Dans le cas **avec** support matériel de virtualisation:

α) (1 point) Quel est l'ajout majeur au matériel permettant de supporter la virtualisation?

β) (2 points) Dessiner et expliquer brièvement le flot de contrôle (control-flow) d'un appel système d'une application s'exécutant sur l'OS guest.

γ) (1 point) Quel est l'avantage majeur de ce mode d'hypervision?

3) (1 point **bonus**) Les hyperviseurs communéments utilisés aujourd'hui (vmware, xen...) sont dits hybrides. Expliquer en quoi le sont-ils avec des exemples concrets de ce mode hybride.