
SYSTÈMES D'EXPLOITATION

EXAMEN

3 heures
avec documents de cours

N.B. : - Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Question 1

(6 points)

On s'intéresse à un disque dur constitué d'un unique disque magnétique de p pistes réparties en s pistes contenant des informations de structure sur les fichiers stockés (que l'on peut se représenter par exemple comme des pistes d'i-nœuds) et d pistes de données proprement dites (on a donc $p = s + d$). La vitesse de rotation du disque est de θ tours par seconde, et la vitesse de déplacement du bras du disque est de β pistes par seconde.

Les seules opérations réalisées sur ce disque sont des transactions atomiques consistant à rajouter une petite quantité d'informations (une ligne de texte) au bloc de données courant d'un fichier journal ouvert au *boot* du système : il faut lire le bloc de structure contenant les informations sur le fichier à modifier, réécrire le bloc de structure modifié (changement de dates et de taille du nœud du fichier, mais sans intérêt ici), et opérer de même pour les données du bloc de données. On suppose qu'il n'y a pas de cache disque.

(1.1) (1,5 points)

En supposant que toutes les pistes de structure sont regroupées à une des extrémités du disque, exprimez, en définissant des hypothèses raisonnables, le temps moyen t_1 de réalisation d'une transaction en fonction des paramètres précédents.

Calculez t_1 pour $s = 63$, $d = 756$, $\theta = 6000$ tours/min, et $\beta = 40$ pistes/ms.

(1.2) (1,5 points)

Au moyen des mêmes hypothèses, exprimez le temps moyen t_2 de réalisation de la même transaction si l'ensemble des pistes de structure est situé au milieu du disque. Calculez la valeur de t_2 avec les valeurs ci-dessus.

(1.3) (1,5 points)

Proposez un agencement des pistes permettant une réduction plus importante du temps de transaction. Exprimez ce temps t_3 au moyen des mêmes paramètres (supposez pour simplifier que d est un multiple de s). Calculez la valeur de t_3 avec les valeurs ci-dessus.

(1.4) (1,5 points)

Jusifiez et exprimez le temps de transaction t_4 obtenu avec un disque disposant de deux bras de lecture indépendants au lieu d'un seul, en faisant l'hypothèse que le deuxième bras n'induit pas de surcoût au niveau du contrôleur. Calculez la valeur de t_4 avec les valeurs ci-dessus.

Question 2

(5,5 points)

On s'intéresse à la gestion des signaux par les différentes versions d'Unix (en particulier BSD et System V). Toutes les versions d'Unix disposent de l'appel système `signal()`, qui permet à un processus de définir la fonction à appeler lorsqu'il recevra un signal donné, et de l'appel système `kill()`, qui permet d'envoyer un signal donné à un processus donné.

- (2.1) (1,5 points)
 Dans la version System V, l'adresse de la routine de traitement d'un type de signal donné, contenue dans la table des signaux du processus, est remise à sa valeur par défaut (en général, l'adresse de `exit()`) juste avant de traiter le signal reçu avec la routine dont l'adresse était contenue dans la table.
 Quel problème cela peut-il poser ? Ecrivez un rapide source C permettant d'exhiber ce phénomène.
- (2.2) (2,5 points)
 Le but initial de cette remise à la valeur par défaut était d'éviter un débordement de la pile dans le cas où un processus recevait des signaux plus vite qu'il ne pouvait les traiter.
 Proposez une implémentation permettant d'apporter une solution à la fois à ce cas pathologique ainsi qu'à celui de la question précédente. Quels inconvénients votre implémentation a-t-elle ? Est-ce grave ?
- (2.3) (1,5 points)
 La table des signaux d'un processus est habituellement située dans sa structure `u (j user i)`. Quel problème de performance cela pose-t-il lorsque le processus est censé ignorer le type de signal qui lui est envoyé ? Comment peut-on pallier ce problème ?

Question 3

(8,5 points)

On s'intéresse à une implémentation sur un système Unix du partage transparent de fichiers à travers un réseau local. L'architecture envisagée est basée sur une modification de la bibliothèque standard, sans toucher au noyau lui-même ; elle est donc différente des approches intervenant directement dans le noyau au niveau des accès aux i-nœuds, comme le font la plupart des implémentations récentes d'Unix.

- (3.1) (0,5 points)
 Pour nommer un fichier distant, on peut envisager soit de rajouter un caractère spécial aux chemins pour séparer la partie `j nom de machine i` de la partie `j chemin i`, comme par exemple `j serveur !/projetA/src/brol.c i`, soit en utilisant un préfixe spécial, du type `j /./serveur/projetA/src/brol.c i`. Quel est l'intérêt de la deuxième approche par rapport à la première ?
- (3.2) (3 points)
 Faites un schéma explicatif d'une implémentation client-serveur permettant à un processus local d'accéder de façon transparente à des fichiers distants. En particulier, faites apparaître les rôles respectifs du processus utilisateur local, de la bibliothèque standard (`libc`), du noyau local, du réseau, etc. Quelles sont les modifications à apporter à la bibliothèque standard ?
- (3.3) (1,5 points)
 Pour des raisons évidentes de sécurité, les requêtes exécutées sur la machine distante doivent avoir les mêmes droits que si l'utilisateur les exécutait directement sur cette machine. Quels problèmes cela pose-t-il, et comment les résoudre ?
- (3.4) (1,5 points)
 Que faut-il faire lorsqu'un processus exécute un `fork()` ?
- (3.5) (1 point)
 Que se passe-t-il si deux processus accèdent en modification au même fichier distant ? Le bit `O_EXCL` de l'appel système `open()` est-il efficace dans ce cas pour créer un fichier verrou ? Si cela était possible, où faudrait-il créer le fichier verrou associé à un fichier partagé donné ?
- (3.6) (1 point)
 Un programme binaire obtenu d'une personne tierce va-t-il pouvoir utiliser les fonctionnalités de l'architecture que vous avez définie ? Qu'en concluez-vous ?