

g

- Tout document autorisé
- Les réponses doivent être **claires et concises** : cela sera pris en compte lors de l'évaluation
- La durée de l'examen est 90 minutes

Exercice 1 (6 pts)

Pour le programme ci-dessous, quelle sortie serait imprimée lors de son exécution ? Supposons que tous les appels de fonction, de bibliothèque et de système réussissent.

```
int x = 42;
main()
{
    int rc, status;
    rc = fork();
    if (rc == 0) {
        func1(); exit(1);
    }
    else {
        rc = waitpid(rc, &status, 0);
        printf("R: %d", WEXITSTATUS(status));
        printf("M: %d\n", x);
        x = 100;
        printf("P: %d\n", x);
        _exit(2);
    }
    func2();
}
```

```
}
void func1()
{
    int rc, status;
    printf("T: %d\n", x);
    x = 10;
    rc = fork();
    if (rc == 0) {
        x = 50;
        printf("Q: %d\n", x);
        exit(3);
    }
    rc = waitpid(rc, &status, 0)
    printf("A: %d\n", x);
    printf("D: %d", WEXITSTATUS(status));
    exit(4);
}
void func2()
{ printf("C: %d\n", x); }
```

Exercice 2 (3 pts)

Supposons qu'un système d'exploitation utilise une stratégie à temps partagé avec priorité avec un quantum de 500 millisecondes. L'horloge matérielle génère une interruption de temporisateur chaque milliseconde. Supposons qu'un processus P est programmé pour s'exécuter et est distribué. Pendant son quantum, P ne fait aucun appel système. Cependant, les exceptions de traduction d'adresse se produisent une fois toutes les 10 millisecondes pendant que P est en cours d'exécution. Combien de fois au cours de son quantum le processus P entre-t-il dans le mode noyau ? Expliquez brièvement votre réponse.

Exercice 3 (11 pts)

On se propose d'étudier une structuration pour des disques dont la capacité est comprise entre 100 Mo et 2 Go., les secteurs étant de 4096 octets.

Les machines pour lesquelles ils sont destinés manipulent des octets, des entiers sur 16 bits ou des entiers sur 32 bits.

Les concepteurs du système de gestion de fichiers ont décidé de diviser chaque unité du disque de la façon suivante :

- *le descripteur du disque*, qui contient tous les renseignements nécessaires à son identification, les paramètres variables de la structure, et une table donnant accès aux fichiers.
- *la table d'allocation de l'espace*, représentant l'espace libre sur le disque.
- *la zone des fichiers*, qui contient les descripteurs de fichiers et l'espace de données de ces fichiers.

Un fichier est constitué de deux parties:

- *le descripteur de fichier* qui contient toutes les informations nécessaires à sa localisation, et qui seront détaillées ci-dessous.
- *l'ensemble des zones physiques*, qui contiennent les données du fichier. Pour un fichier, toutes les zones physiques du fichier ont la même taille, et sont constituées d'un nombre entier de secteurs contigus. Par contre les différentes zones d'un même fichier ne sont pas nécessairement contiguës.

Les descripteurs de fichiers doivent repérer l'ensemble des zones du fichier. Pour permettre à un fichier de contenir un nombre quelconque de zones, tout en ayant des descripteurs de taille fixe, les descripteurs sont décomposés en une partie principale (obligatoire) et des parties extensions (optionnelles). Les parties extensions éventuelles ont la même structure que les parties principales, les informations inutilisées étant simplement mises à 0 dans ce cas. Par la suite nous appellerons descripteur cette structure, qu'elle soit utilisée comme partie principale ou comme extension. Les descripteurs de fichiers sont regroupés dans des secteurs du disque. Ces secteurs sont chaînés entre eux, pour permettre la recherche des fichiers. Un descripteur de fichier a la structure suivante :

- le *nom* du fichier sur 16 octets,
- des informations diverses sur 12 octets,
- la *longueur du fichier*, c'est-à-dire le nombre total d'octets du fichier,
- le *nombre total de zones* du fichier,
- la *taille des zones* en nombre de secteurs,
- l'*"adresse" de l'extension suivante* (numéro de secteur et position dans le secteur),
- l'*"adresse" de la dernière extension* du fichier,
- la *table des numéros de premiers secteurs* des premières zones.

A.1- En considérant que tout entier sur disque a la même représentation qu'en mémoire centrale, c'est-à-dire 16 ou 32 bits, déterminer la taille en octets nécessaire à la représentation de chacune des informations suivantes:

- un numéro de secteur,
- la longueur du fichier, c'est-à-dire le nombre total d'octets du fichier,
- le nombre total de zones du fichier,
- la taille des zones en nombre de secteurs,
- l'*"adresse"* d'une extension (numéro de secteur et position dans le secteur).

A.2- Déterminer la taille d'un descripteur en fonction du nombre d'entrées q de la table des premiers secteurs de zones dans un descripteur. En déduire la valeur maximale du nombre r de descripteurs dans les secteurs du disque qui les contiennent, en fonction de q , et la perte qui résulte d'un choix donné de r et q compatibles.

A.3- Étudier en particulier les cas où ce nombre q prend les valeurs 31, 32 et 33. Calculer la perte d'espace dans chaque cas. Commenter le choix de 32.

A.4- Expliquer l'intérêt des différentes informations (autres que diverses) qui sont présentes dans la partie principale d'un descripteur. On distinguera celles qui sont nécessaires et celles qui sont redondantes, et on justifiera cette redondance. Montrer que l'on peut déduire de ces informations le nombre d'extensions du descripteur de fichier.