

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); }
```

T: 42

Q: 50

A: 10

D: 3

R: 4

M: 42

P: 100

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.

Chaque exception et interruption fait entrer le contrôle dans le noyau. Il y aura 500 interruptions de minuterie pendant le quantum de P, et il y aura $500/10 = 50$ exceptions de traduction d'adresse, pour un total de 550 entrées de noyau.

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).

• *numéro de secteur*. Un disque peut contenir jusqu'à $2 \cdot 2^{30} / 4 \cdot 2^{10} = 2^{19}$ secteurs. Il faut donc au moins 19 bits pour représenter un numéro de secteur, soit un entier sur 32 bits, ou 4 octets.

• *longueur du fichier*. Un fichier ne peut dépasser la taille maximale d'un disque, donc $2 \cdot 2^{30} = 2^{31}$ Il faut donc des entiers sur 32 bits, soit 4 octets.

• *nombre total de zones et taille des zones*. En l'absence de contrainte, une zone physique d'un fichier doit être d'au moins 1 secteur, mais pouvoir atteindre presque le disque complet (à

l'exception des informations structurelles). Un fichier peut donc avoir 2^{19} zones de 1 secteur; il faut donc 32 bits ou 4 octets pour représenter le nombre total de zones du fichier. De même, un fichier peut avoir 1 zone de 2^{19} secteurs; il faut donc 32 bits ou 4 octets pour représenter la taille des zones du fichier. Notons qu'il ne serait pas très contraignant et sans doute assez logique de limiter chacune de ces quantités à 2^{16} , permettant ainsi d'utiliser des entiers sur 16 bits ou 2 octets, puisque ceci nous donne de toute façon 2^{32} secteurs, valeur nettement supérieure au nombre maximal de secteurs d'un disque.

- *adresse d'une extension.* L'adresse d'une partie extension d'un descripteur doit permettre de désigner d'une part le numéro de secteur disque où la partie extension est placée (19 bits), et d'autre part, sa position dans ce secteur. Celle-ci peut être définie simplement par le numéro de son premier octet (12 bits), donnant ainsi un total de 31 bits. Cette position peut aussi être le numéro d'ordre du descripteur dans le secteur, puisque tous les descripteurs sont de même taille. Par exemple, s'il y a au plus 32 descripteurs par secteur, 5 bits suffiront alors pour ce numéro, et une adresse de partie extension nécessitera 24 bits. Enfin, en suivant la politique de représentation des entiers en 16 ou 32 bits, on constate alors qu'il faut 32 bits ou 4 octets dans tous les cas.

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.

Pour déterminer le nombre de descripteurs de fichiers par secteur, il faut déterminer la taille d'un descripteur. Nous reprenons les données numériques de la question A.1, et les notons devant les informations du descripteur de fichier:

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

Il s'ensuit que le descripteur occupe $48 + 4 * q$ octets, où q est le nombre d'entrées de la table des premiers secteurs de blocs d'un descripteur. Les secteurs contenant des descripteurs étant chaînés entre eux, le lien de chaînage occupe 4 octets, laissant libres 4092 octets pour r descripteurs, et il s'ensuit que $r \leq 4092 / (48 + 4 * q)$. Pour une valeur de r compatible avec q , la perte est $4092 - r * (48 + 4 * q)$.

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.

Nous avons les valeurs suivantes:

q	r maximum	perte
31	23	136

32	23	44
33	22	132

Le choix de 32 est le plus optimal, puisqu'il correspond à la plus petite perte. Cependant, même s'il induisait une perte supérieure aux autres, il pourrait être conservé car il a l'avantage de permettre de représenter, dans chaque descripteur, qu'il soit partie principale ou extension, un nombre de zones qui est une puissance de 2.

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.

Le *nom* du fichier est nécessaire pour distinguer les fichiers entre eux, et permet à l'utilisateur de désigner ce fichier par une chaîne de caractères. La *longueur du fichier* N est la seule information de taille qui permette de savoir exactement quels sont les octets de l'espace alloué au fichier qui en font partie. La *taille des zones* T est nécessaire pour déterminer la taille des zones à allouer au fur et à mesure des besoins au fichier; elle permet également de déterminer à quelle zone appartient le $n^{\text{ième}}$ secteur du fichier. L'*adresse de l'extension suivante* est nécessaire pour parcourir les extensions du descripteur. La *table des numéros de premiers secteurs* permet de savoir où commence chaque zone sur disque, et donc détermine l'espace alloué au fichier.

Le *nombre total de zones* du fichier est une information redondante, en ce sens qu'elle peut être reconstruite à partir des autres. En effet, le nombre de secteurs S est égal à $(N-1)\%4096+1$, où $\%$ désigne la division entière. À partir de la taille des zones T , on peut déduire le nombre de zones $B = (S-1)\%T+1$. Elle évite de refaire ce calcul en particulier pour déterminer la taille de l'espace occupé par le fichier sur le disque. Enfin l'*adresse de la dernière extension* du fichier est également une information redondante, puisqu'elle peut être connue en parcourant les extensions successives. Elle est utile car l'allocation d'une nouvelle zone lors d'un allongement du fichier, entraîne la modification de cette extension.

Par ailleurs $(B-1)\%32$ nous donne le nombre d'extensions du fichier.