

<p style="text-align: center;">info524 : Systèmes d'exploitation TD 6 : systèmes de fichiers, exemples</p>
--

Pierre Hyvernats et Florian Hatat
Pierre.Hyvernats@univ-savoie.fr
Florian.Hatat@univ-savoie.fr

Question 1. Quelles différences y a-t-il entre renommer un fichier (fonction `rename`) et copier (en changeant le nom) et effacer l'ancienne version du fichier ?

Exercice 1 : examen 2010/2011

Question 1. Sur un système Linux habituel (avec système de fichiers à base d'inodes), la copie d'un fichier :

```
$ cp projet.tar projet-backup.tar
```

prend un temps proportionnel à la taille du fichier.

Par ailleurs, le déplacement d'un fichier :

```
$ mv projet-tmp.tar projet-final.tar
```

est *en général* instantané.

Expliquez la différence en décrivant ce qui se passe au niveau du système de fichiers.

Question 2. Sur un système Linux habituel (avec système de fichiers à base d'inodes), le déplacement d'un fichier peut *parfois* prendre autant de temps qu'une copie :

```
$ mv /tmp/projet.tar $HOME/projet.tar
```

Pourquoi ?

Question 3. La fonction POSIX pour supprimer un fichier en C ne s'appelle pas "`remove`" comme attendu, mais "`unlink`".

En vous appuyant sur le système de fichier EXT2/3 vu en cours et en TD, expliquez ce que fait vraiment cette fonction et justifiez son nom.

Question 4. Les liens physiques peuvent poser des problèmes sur un système multi-utilisateurs avec quotas de disques. (C'est à dire que les utilisateurs ont une limite d'utilisation de disque.)

Un utilisateur malveillant Eve peut créer des liens physiques des fichiers de Alice dans son propre répertoire :

```
eve$ ln /home/alice/projet.tgz /home/eve/tmp/
```

Quel problème cela peut-il poser pour Alice ?

Exercice 2 : Le système FAT

Le système de fichiers de MS-DOS n'utilise pas d'inodes, mais une "FAT" ("File Allocation Table"). Les fichiers sont découpés en blocs de taille fixe, et la FAT est un tableau qui contient, pour chaque adresse de bloc utilisé, l'adresse du bloc suivant (ou 0 pour le dernier bloc). Les blocs libres contiennent un numéro spécial (-1 par exemple)

Question 1. On suppose que la FAT d'un tels système est égale à

0	1	2	3	4	5	7	8	9	10	11	12	13	14	15
-1	-1	0	-1	12	-1	8	7	-1	0	10	13	2	-1	-1

En supposant que les blocs font 512 octets, et que le bloc 4 est le début d'un fichier,

- quels sont les blocs utilisés par le fichier ?
- Quelle est la taille occupée par les données du fichier sur le disque ?

Que pensez-vous du fichier commençant au bloc 7 ?

Question 2. La FAT est stockée au début de la partition. La première version du système, FAT12 stockait les adresses des blocs sur 12 bits, et chaque bloc faisait 512 octets.

Par contre, comme il n'est pas commode de manipuler les bits par paquets de 12, chaque adresse était en fait stockée sur 2 octets, c'est-à-dire 16 bits.

Quelle était la taille maximale d'une partition pour ce système de fichiers ?

Quelle était la taille de la FAT sur la partition ?

Question 3. Pour autoriser des disques plus gros, Microsoft a permis l'utilisation de blocs de 512 octets, 1Ko, 2Ko, 4Ko, 8Ko, 16Ko et 32Ko et a étendu les adresses sur 16 bits : FAT16.

Quelle était la taille maximale d'une partition pour ce système de fichiers ? En sachant qu'un disque pouvait avoir au plus 4 partitions, quelle était la taille maximale d'un disque pour ce système ?

Quelle était la taille de la FAT sur la partition ?

Question 4. Pour le système FAT32, les adresses utilisent 28 bits, stockées sur 4 octets. Répondez aux questions précédentes pour ce système.

Question 5. La taille d'une partition est stockée sur 32 bits pour des secteurs de 512 octets. Quelle limite cela impose-t-il sur la taille des partitions ?

Question 6. Pour une partition de 2 Go, quels avantages a-t-on à utiliser FAT32 par rapport à FAT16 ?

Question 7. Un répertoire MS-DOS est représenté sur 32 octets de la manière suivante :

- octets 0 à 7 : nom du fichier,
- octets 8 à 10 : extension,
- octet 11 : attributs (lecture seule, fichier caché, fichier système, ...),
- octets 12 à 21 : octets "réservés",
- octets 22 et 23 : heure,
- octets 24 et 25 : date,
- octets 26 et 27 : numéro du premier bloc,
- octets 28 à 31 : taille du fichier.

La date est stockée comme suit :

- 5 bits pour le jour,
- 4 bits pour le mois,
- 7 bits pour l'année, en comptant à partir de 1980.

L'heure est stockée comme suit :

- 5 bits pour les secondes,
- 6 bits pour les minutes,
- 5 bits pour les heures.

Expliquez pourquoi la précision de la date de modification des fichiers sur ce système est de 2 secondes seulement. Est-il possible de faire mieux sans rajouter de bits pour les champs date et heure ?

Question 8. Le "bug de l'an 2000" n'a pas de sens pour le système de fichiers MS-DOS. Calculez la date du "bug de l'an MS-DOS".

Peux-t-on repousser cette fin du monde sans rajouter des bits aux champs date et heure ?