

**TD n°1**  
**Systèmes de fichiers, premières commandes**

**Exercice 1**

On dispose d'un disque dur de  $C$  secteurs de 512 octets chacun, que l'on voudrait exploiter sous Linux avec une seule partition de type Ext2. Les blocs utilisés pour découper les fichiers sont longs de  $B$  secteurs chacun. Chaque entrée dans la table des i-nodes (également appelée table d'index) comporte  $T$  pointeurs de blocs sur 32 bits, dont  $T-2$  directs, 1 indirect simple, et 1 indirect double. On suppose que  $C$  est suffisamment grand pour négliger l'espace alloué pour l'exploitation du disque (superbloc, répertoire racine, table des i-nodes, ...)

1. Dans le système de fichiers Ext2, on utilise une bitmap dont chacun des bits représente l'état « bloc alloué » (valeur à 1) ou « bloc libre » (valeur à 0) pour représenter le fait qu'un bloc est alloué ou non. Quelle est la taille, en secteurs, de cette bitmap d'allocation pour le problème considéré ?
2. En supposant que le plus grand fichier sur disque que l'on doit adresser soit long de  $L$  secteurs, à quelle valeur  $T_0$  la quantité  $T$  doit-elle nécessairement être supérieure ?
3. Quel sera alors, en secteurs, la taille de la table des i-nodes si l'on suppose que l'ensemble des blocs est alloué à un seul fichier ?
4. Un client vient de réceptionner un disque dur de 1TO, et vous demande de le configurer en système Ext2 pour le monter sur un serveur de fichiers Linux. Ses besoins sont les suivantes :
  - 500,000 fichiers à stocker au plus ;
  - Au moins 99% des fichiers auront une taille inférieure à 1 MO, et seront très fréquemment consultés ;
  - La taille d'un fichier ne peut excéder 4GO, mais des fichiers de cette taille ne sont pas à exclure.

On voudrait que la table des i-nodes tienne le moins de place possible afin de résider éventuellement en mémoire centrale (le système ne recopie alors sur disque que les parties modifiées de la table lors de chaque fermeture de fichier).

Quelle configuration (taille des blocs, de la table des i-nodes) proposez-vous ? Quelle serait la taille de la table des i-nodes en mémoire centrale ?

Que pourriez-vous proposer pour améliorer votre solution (quitte à modifier le système Ext 2) ?

## Exercice 2

Note : `find`, `xterm`, `test`, `sort`, et `[` sont des commandes externes. `[ x ]` est équivalent à `test x`, et `[ ! x ]` est la négation de `[ x ]`.

1. Les lignes suivantes sont-elles syntaxiquement correctes ? Si oui, que font-elles ?
  - a. `COL=25 xterm & xterm ;`
  - b. `ls \{" * \"} \ ;`
  - c. `if test -d /etc then echo /etc existe ; fi`
  - d. `${var:-ls -ls} | wc`
  - e. `Function Machin () { ls "$1" ; }`
2. Que font les lignes de commandes ou fonctions suivantes ?
  - a. `test -d /etc && ls /etc | wc -l | echo message`
  - b. `function mafonc () {  
test -d "$1" && { ls "$1" | wc -l || echo message ; } }`
  - c. `echo "/usr/local/bin/*"`
  - d. `$ for f in /etc/*; do  
> if [ ! "${f%~}" = "$f" ]  
> then echo "fichier $f"  
> fi  
> done | sort -r`

## Exercice 3

1. Dire ce que font ou ce qu'affichent les lignes de commandes suivantes, en considérant que chaque ligne est entrée dans un nouveau shell (indépendance des lignes les unes vis-à-vis des autres). Si vous pensez qu'une ligne provoque une erreur ou si la réponse est indéterminée, indiquez-le clairement (inutile de justifier).

1. `x=12 echo $x+1`
2. `x=12 ; echo $((x+1))`
3. `x=y ; echo $((x*2))`
4. `[ -d /etc ] && ls -l /etc | wc -l`
5. `ssh user@chose 'ls -l /???'`
6. `nom=/usr/lib/X11R6/libFS.a ; echo ${nom%/*} ; echo ${nom##/*}`

2. Dire ce que fait la fonction suivante :

```
function mystere () {  
local mot=$1  
[ "${mot#${mot%?}}" = "${mot%${mot#?}}" ] || { echo "Pas bon"; return 1; }
```

```
mot=${mot%?}
mot=${mot#?}
```

```
[ "$mot" ] || { echo "Bon"; return 0; }
mystere "$mot"
}
```