



TP Systèmes d'Exploitation

L'ensemble des commandes tapées en mode CLI et leurs résultats sont requis. Un document au format texte (réalisé par exemple sous gedit) reprenant les numéros des questions, les commandes et leurs résultats ainsi qu'un document papier pour les représentations graphiques et vos commentaires constitueront les éléments à remettre à l'issue du TP.

*Note : Il sera tenu compte de la présentation
JUSTIFIER VOS RÉPONSES*

L'objectif de ce TP de s'initier au système de fichiers

0/ Prérequis : Machine sous **Linux Ubuntu**, shell bash

1/ Droits des fichiers

Soient les utilisateurs créés précédemment, einstein et bohr, et le groupe physiciens auquel einstein et bohr appartiennent.

- 1.1 Se connecter en tant que einstein.
- 1.2 Créer un répertoire *personnel* sous le homedir
- 1.3 Créer un fichier texte secret.txt contenant 'Ceci est privé' dans le répertoire personnel
- 1.4 Dans un autre fenêtre terminal, se connecter en tant que bohr. et lire le fichier secret.txt
- 1.5 Retreindre les droits du fichier secret.txt de façon que seul einstein puisse lire/modifier ce fichier.
- 1.6 Depuis bohr, tentez de nouveau de lire le fichier secret.txt
- 1.7 Proposer une solution afin que tous les fichiers du repertoire personnel ne soit accessible qu'a son propriétaire.
- 1.8 En tant que einstein, créer un répertoire *partage* accessible en lecture/écriture à tous les membres du groupe physiciens.
- 1.9 Vérifier que bohr peut créer, modifier les fichiers du répertoire partage.
- 1.10 Vérifier que einstein peut modifier les fichiers créés par bohr
- 1.11 Proposer une solution afin que membres du groupe physiciens puissent lire/modifier les fichiers mais ne puissent effacer que les fichiers dont ils sont les propriétaires.

2 / Identification

- 2.1 Via l'utilisation de la commande *lshw*, identifier les disque et volumes de votre système
- 2.2 Observer les différents paramètres du filesystem obtenus par la commande *tune2fs*
- 2.3 Installer le paquet *smartmontools*
- 2.4 Observer les paramètres fournis par la commande *smartmonctl*

3/ Inode

- 3.1 Dans votre répertoire, créer un fichier texte non vide
- 3.2 Afficher les informations de base sur ce fichier
- 3.3 Afficher les informations contenues dans l'inode associé
- 3.4 A quoi correspondent les différents temps (atime,ctime,mtime,crtime)
- 3.5 A l'aide de l'outil *debugfs*, identifier le(s) bloc(s) de données de ce fichier.
- 3.6 Via la commande *dd* suivante, lire le bloc de donnée
`dd if=<device> bs=4096 count=1 |skip=<n° bloc> | hexdump -C`
- 3.7 Choisir un fichier de grande taille en utilisant la commande suivante : `find / -type f -size +1G`
- 3.8 L'allocation des blocs est elle contigue ?

- 3.9 Si l'allocation des extents n'est pas contigue, essayer de défragmenter le fichier
- 3.10 Créer un (ou plusieurs) lien physique (hard link) sur un fichier via la commande `ln`
- 3.11 Observer via la commande `ls -il`
- 3.12 Déplacer le lien dans un autre répertoire ?
- 3.13 Quel impact ? Pourquoi ?
- 3.14 Création d'un (ou plusieurs) lien symbolique (soft link) sur un fichier via la commande `ln`
- 3.15 Observer via la commande `ls -il`
- 3.16 Déplacer le lien - Qu'observer vous ?
- 3.17 Déplacer le fichier - Qu'observer vous ? (`ls -L`)
- 3.18 Repérer l'inode du lien symbolique
- 3.19 Afficher son extent
- 3.20 Que constatez vous ?
- 3.21 Quel est l'avantage de Fast Link ?
- 3.22 Créer un fichier texte non vide et identifier son i-node
- 3.23 Identifier son bloc de données
- 3.24 Lire le bloc de données sur le disque (via `dd`)
- 3.25 Effacer le fichier par l'utilisation de la commande `rm`
- 3.26 Relire le bloc de données précédent
- 3.27 Conclure sur le fonctionnement de `rm`
- 3.28 Etudier la commande `shred`
- 3.29 Redémarrer au 3.22 mais en 3.25 effacer le fichier par l'utilisation de `shred`
- 3.30 Relire le bloc de données précédent

Remarque : Le FS utilise des caches mémoire, la commande suivante permet de forcer le système à écrire les données en cache sur disque : `echo 3 > /proc/sys/vm/drop_caches`

4/ Montage et fsck

Pour la suite de l'exercice, un disque fictif sera utilisé

- 4.1 Créer un disque fictif de taille 1GB par la suite de commandes suivante

```
dd if=/dev/zero of=disque1.dsk bs=1M count=1000
```

```
mkfs.ext4 disque1.dsk
```

```
mkdir /mnt/d1
```

```
mount -t auto -o loop disque1.dsk /mnt/d1
```

- 4.2 Valider le bon fonctionnement du montage en créant un fichier non vide dans cette partition
- 4.3 Editer le fichier avec `vi` dans une autre fenêtre terminal
- 4.3 Démonter la partition
- 4.4 Quel est le message affiché ? Quelle est sa signification ?
- 4.5 Identifier le problème via les commandes `lsof` et `fuser`

- 4.6 Sur la partition nouvellement créée, ajouter un répertoire, puis ajouter 2 à 3 fichiers (non vides) à ce répertoire

- 4.7 Repérer l'inode du répertoire

4.8 Via la commande '`clri <inode répertoire> | debugfs -w <partition>`', les données de l'inode du répertoire sont mises à zero, le répertoire n'est plus accessible (ne pointe plus vers son bloc de données) mais les fichiers existent toujours

- 4.8 Démonter la partition

- 4.9 Utiliser la commande `e2fsck` pour tenter une réparation du file system

- 4.10 Peut-on retrouver les fichiers du répertoire ?