

03 - Filesystems et unités de stockage

Objectif

- Donner aux participants une méthode pour choisir, maintenir, réparer et organiser le stockage Linux.

1. Choisir un système de fichiers — Questions à poser

- Machine de bureau ou serveur ?
- Beaucoup de petits fichiers ou quelques très gros fichiers ?
- Besoin de snapshots ?
- Besoin de performances en écriture ?
- Besoin de robustesse maximale ?
- Stockage local, cluster ou haute disponibilité ?

2. Journalisation

- Sans journal, si une coupure survient pendant une écriture, le système peut ne pas savoir ce qui a été terminé ou non.
- Avec un journal, le système note son intention avant d'agir.
- Au redémarrage, il peut rejouer ou annuler proprement les opérations incomplètes.
- La plupart des systèmes modernes journalisent surtout les métadonnées : inode, emplacement, taille, état du fichier.
- Cela protège la structure du système de fichiers, sans garantir que chaque octet applicatif soit intact.

3. Comparatif synthétique — ext2

- Ancien, non journalisé.
- Peut encore être utile sur certains supports flash ou pour compatibilité spécifique, mais rarement recommandé aujourd'hui.

3. Comparatif synthétique — ext3

- Ext2 avec journalisation.
- Historiquement très utilisé, stable, mais largement remplacé par ext4.

3. Comparatif synthétique — ext4

- Choix généraliste solide.
- Stable, largement testé, bon compromis entre performance, maturité et simplicité.

3. Comparatif synthétique — XFS

- Très bon choix serveur, notamment pour les gros fichiers et les volumes importants.
- Peut être agrandi à chaud, mais pas réduit simplement.
- Moins adapté à certains workloads composés de nombreux très petits fichiers.

3. Comparatif synthétique — JFS

- Faible consommation CPU, bonnes performances, mais moins courant dans les déploiements Linux modernes que ext4 ou XFS.

3. Comparatif synthétique — Btrfs

- Système moderne avec fonctionnalités avancées : snapshots, compression transparente, checksums, pooling.
- Très intéressant, mais à présenter comme un choix qui demande de comprendre ses mécanismes.

4. Récupération de données : prudence d'abord

- Le cours insiste sur un point essentiel : toute manipulation directe du disque peut empirer la situation.
- Si les données sont critiques et irremplaçables, on éteint et on envisage une récupération spécialisée.
- Si l'on intervient, on travaille sur une image.

4. Récupération de données : prudence d'abord — ddrescue / dd_rescue

- Contrairement à `dd`, ces outils peuvent reprendre une récupération, gérer des erreurs et conserver un journal de progression.

```
ddrescue /dev/sdX image_disque.img image_disque.log
```

- Ensuite, on monte ou analyse l'image en lecture seule.

5. Outils de récupération — Foremost et Scalpel

- Outils de data carving.
- Ils recherchent des fichiers par signatures, en-têtes, pieds de fichiers et structures internes.
- Utile quand le système de fichiers ne suffit plus.

5. Outils de récupération — Extundelete

- Outil destiné aux partitions ext3/ext4 démontées.
- Il peut récupérer des fichiers supprimés récemment.

5. Outils de récupération — TestDisk et PhotoRec

- **TestDisk** : récupération de partitions perdues et réparation de disques non amorçables.
- **PhotoRec** : récupération de fichiers en ignorant le système de fichiers ; fonctionne même sur des systèmes très endommagés ou reformattés.

5. Outils de récupération — grep sur périphérique

- Pour un fichier texte, il est parfois possible de rechercher une chaîne directement sur la partition :

```
grep -a -C 200 -F 'Une ligne du fichier' /dev/sdXN > OutputFile
```


6. Vérifier l'utilisation disque — du

- `du` travaille au niveau des fichiers et répertoires.

```
du -sh /var/log  
du -Sh /var/log
```

6. Vérifier l'utilisation disque — df

- `df` travaille au niveau des systèmes de fichiers montés.

```
df -h  
df -i
```

- Faire remarquer la différence entre espace disque saturé et inodes saturés.

7. Maintenance ext2/ext3/ext4 — fsck et e2fsck

- `fsck` est un frontal qui appelle l'outil adapté au système de fichiers.

```
fsck /dev/sdb1  
fsck -t ext4 /dev/sdb1  
e2fsck -f /dev/sdb1
```

7. Maintenance ext2/ext3/ext4 — tune2fs

- Permet d'afficher et modifier certains paramètres ext.

```
tune2fs -l /dev/sdb1
```

- Ajouter un journal à un ext2 pour le convertir en ext3 :

```
tune2fs -j /dev/sdb1
```

7. Maintenance ext2/ext3/ext4 — Point d'attention

- On ne lance pas une réparation destructrice sur un système de fichiers monté en écriture, sauf cas très particulier et maîtrisé.

8. Maintenance XFS — xfs_repair

- Mode vérification sans modification :

```
xfs_repair -n /dev/sdb1
```

- Réparation :

```
xfs_repair /dev/sdb1
```

8. Maintenance XFS — xfs_db et xfs_fsr

- `xfs_db` : inspection et débogage avancé.
- `xfs_fsr` : réorganisation / défragmentation XFS.

9. Copie d'un disque système à chaud

- Le cours présente une méthode avancée : déplacer temporairement la racine dans un tmpfs grâce à `pivot_root`, arrêter les services, puis copier l'ancienne racine devenue inactive.
- Exemple de préparation :

```
mkdir /tmp/tmproot
mount none /tmp/tmproot -t tmpfs
mkdir /tmp/tmproot/{proc,sys,usr,var,oldroot}
cp -ax /{bin,etc,mnt,sbin,lib} /tmp/tmproot/
cp -ax /usr/{bin,sbin,lib} /tmp/tmproot/usr/
cp -a /dev /tmp/tmproot/dev
```

- Pivot :

```
pivot_root /tmp/tmproot/ /tmp/tmproot/oldroot
mount none /proc -t proc
mount none /sys -t sysfs
mount none /dev/pts -t devpts
```

9. Copie d'un disque système à chaud — Point d'attention

- Cette manipulation est avancée.
- En formation, elle est surtout utile pour expliquer le problème de cohérence d'une copie à chaud.

10. LVM : modes linéaire, stripe, miroir et snapshots — Volumes linéaires

- Concatènent plusieurs volumes physiques dans un volume logique.
- Simple et flexible, mais sans gain de performance ni redondance intrinsèque.

10. LVM : modes linéaire, stripe, miroir et snapshots — Volumes en stripe

- Répartissent les données sur plusieurs volumes physiques.
- Les E/S peuvent être parallélisées, ce qui améliore les performances séquentielles.
- En contrepartie, la panne d'un disque peut impacter tout le volume si aucune redondance n'existe en dessous.

10. LVM : modes linéaire, stripe, miroir et snapshots — Volumes en miroir

- Écrivent des copies identiques sur plusieurs périphériques.
- Protègent contre la panne d'un périphérique, au prix d'une consommation d'espace plus élevée.

10. LVM : modes linéaire, stripe, miroir et snapshots — Snapshots

- Un snapshot LVM capture l'état d'un volume à un instant donné.
- Quand une donnée du volume d'origine change, LVM conserve l'ancienne version dans l'espace du snapshot pour pouvoir reconstruire la vue figée.
- Usages :
 - sauvegarde cohérente sans arrêt long du service ;
 - test applicatif sur copie de données de production ;
 - vérification `fsck` sur snapshot.

10. LVM : modes linéaire, stripe, miroir et snapshots — Point d'attention

- Un snapshot n'est pas une sauvegarde.
- S'il est sur le même stockage et que le stockage est perdu, le snapshot est perdu aussi.