

Empiler des block devices

Le but de ce TP est de pratiquer pour se familiariser avec la 3e couche d'abstraction du cours sur les fichiers, à savoir block devices. Aussi, l'étape de partitionnement est généralement celle qui crée le plus de confusions lors de l'installation d'un système sur le disque d'un ordinateur. Elle devrait ensuite vous paraître banale, de sorte que vous pourrez vous concentrer sur la stratégie de partitionnement.

Vous pouvez vous référer aux démos ainsi que le tableau du cours qui sont sur la page du cours pour vous inspirer.

Pour ne pas prendre de risques, nous ne toucherons pas directement les disques physiques (clefs USB, disque dur, carte SD, etc), mais plutôt des *périphériques de type boucle* (en: *loop devices*) facilement jetables.

Si un-e de vos camarades vous invite à remplacer `/dev/loop*` par `/dev/sd*`, ne l'écoutez-pas !

Si votre système GNU/Linux est émulé par WSL, il vous faudra passer à WSL2 (qui fait tourner un véritable noyau Linux avec `udev`), et tant pis pour l'IPv6.

Travailler en RAM

Avec les commandes `ls -l` et `findmnt`, vous pouvez constater que `/run/shm/` et `/dev/shm` sont :

- un lien symbolique vers un répertoire (savoir si c'est `/run/shm/` qui est un lien symbolique vers le répertoire `/dev/shm` ou `/dev/shm` qui est un lien symbolique vers le répertoire `/run/shm/` dépend de votre distribution)
- un répertoire sur lequel est monté un système de fichiers de type `tmpfs` (`tmp` pour "temporaire", `fs` pour "filesystem"). Il s'agit d'un système de fichiers en RAM.

Sauf si vous voulez conserver vos travaux, vous pouvez travailler dans ce répertoire pour ne pas utiliser d'espace sur le disque.

Jonglage

1. Basique

- Créez un fichier rempli de zéros de 100 MiB.
- Faites-en un périphérique de type boucle.
- Créez une table de partition de type `msdos`.
- Créez 2 partitions primaires de tailles 30 MiB et 70 MiB.
- Vérifiez que `lsblk` retourne quelque chose comme :

```
NAME          MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0          7:0    0   100M 0 loop
└─loop0p1      259:0    0    30M 0 part
└─loop0p2      259:1    0    70M 0 part
```

- Formatez la première partition en `vfat` et montez-la sur `/mnt/winme`.
- Formatez la seconde partition en `ext2` et montez-la sur `/mnt/posix`.
- Vérifiez que `findmnt` retourne quelque chose comme :

TARGET	SOURCE	FSTYPE	OPTIONS
...			
└─/mnt/winme	/dev/loop0p1	vfat	rw,relatime,fmask=002
└─/mnt/posix	/dev/loop0p2	ext2	rw,relatime

- Notez les nombreuses options de montage de la partition formatée en `vfat`. Ceci est lié au fait que ce système de fichiers n'est pas POSIX. Par exemple, comme il ne permet pas de

spécifier des permissions aux fichiers, `fmask=0022,dmask=0022` décrit (au moyen de masques) les permissions des fichiers réguliers et des répertoires.

- (j) Démontez les deux systèmes de fichiers et détachez les block devices.
2. Croisements

- (a) À partir de 2 fichiers réguliers de 100 MiB,
- (b) créez 4 partitions de 50 MiB
- (c) pour obtenir 2 block devices (de 100 MiB chacun) les mélangeant.
- (d) Vérifiez que `lsblk` retourne quelque chose comme :

```
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop3                7:0    0  100M  0 loop
├─loop3p1            259:3    0   49M  0 part
│ └─vg_p1-lvol10    253:0    0   96M  0 lvm
├─loop3p2            259:4    0   50M  0 part
│ └─vg_p2-lvol10    253:1    0   96M  0 lvm
loop4                7:1    0  100M  0 loop
├─loop4p1            259:5    0   49M  0 part
│ └─vg_p1-lvol10    253:0    0   96M  0 lvm
├─loop4p2            259:6    0   50M  0 part
│ └─vg_p2-lvol10    253:1    0   96M  0 lvm
```

- (e) Comme vous pouvez le constater, la représentation sous forme de forêt (plusieurs arborescences) n'est pas idéale car des blocs devices ont du être répétés pour éviter les croisements. Faites un dessin représentant plus fidèlement les 8 block devices et leurs relations.
3. Partiellement chiffré
- (a) À partir de fichiers réguliers `f1` de 100 MiB et `f2` de 50 MiB,
 - (b) obtenez un block device `b12` fait des 50 MiB finaux de `f1` et des 50 MiB de `f2`,
 - (c) et chiffrez `b12`.
 - (d) Puis, recomposez un block device global fait de `b12` après les 50 MiB initiaux de `f1`.
 - (e) Faites un dessin représentant les relations entre les 6 blocs devices créés.

Rafrâchissement

Il se peut lorsque vous détachez un fichier de son loop device et que vous rattachiez un autre fichier au même loop device que le noyau ne se rende pas compte que c'est un nouveau fichier de sorte que la commande `lsblk` va faire réapparaître les partitions de l'ancien loop device.

Si c'est le cas, pour dire au noyau de rejeter un oeil à la table de partitions du loop device, utilisez `partprobe`.

Headers msdos et GPT

Le but de cette partie est d'observer les en-têtes des tables de partitions au format `msdos` et `GPT`.

1. Créez deux fichiers remplis de zéros de 10 MiB chacun.
2. Promouvez ces deux fichiers en loop devices.
3. Avec la commande `hexdump`, vérifiez que les deux fichiers n'ont pas été touchés et sont encore plein de zéros.
4. Sur le premier loop device, créez une table de partitions `msdos` et sur le second, créez une table de partitions `GPT`.
5. Observez avec `hexdump` ce qui a été écrit dans chaque fichier. Observez-bien ce qui a été écrit pour le fichier avec la table `GPT`, et où ça a été écrit (la première colonne représente les adresses, les autres colonnes représentent les données).
6. Créez une première partition primaire de taille 2MiB dans chacun des périphériques, puis observez l'effet de votre opération sur les fichiers avec `hexdump`.
7. Créez une seconde partition primaire de taille 2MiB dans chacun des périphériques, puis observez l'effet de votre opération sur les fichiers avec `hexdump`.

8. Créez une troisième partition primaire de taille 2MiB dans chacun des périphériques, puis observez l'effet de votre opération sur les fichiers avec `hexdump`.
9. Créez une quatrième partition primaire de taille 2MiB dans chacun des périphériques, puis observez l'effet de votre opération sur les fichiers avec `hexdump`.
10. Créez une cinquième partition primaire de taille 2MiB dans chacun des périphériques, puis observez l'effet de votre opération sur les fichiers avec `hexdump`.
11. Quelle structure semble apparaître sur le périphérique avec une table de partition `msdos` ?
12. Quelle structure semble apparaître sur le périphérique avec une table `GPT` ?
13. Interprétez.
14. Documentez-vous sur les deux types de tables de partitions, par exemple sur wikipedia. Pour les tables `msdos`, cherchez à propos de "Master Boot Record", pour les tables `GPT`, cherchez à propos de "GUID Partition Table", et confirmez ou infirmez vos hypothèses.

RAID

Il existe d'autres façons de combiner des périphériques blocs pour en faire de nouveaux. Nous avons vu comment coller des périphériques blocs bout à bout avec LVM (de façon "linéaire"). Cela permet de simuler un grand disque avec plusieurs petits.

Mais d'autres combinaisons sont possibles.

Par exemple, au lieu d'aligner (et remplir) les disques les uns après les autres, on peut les "entrelacer" de sorte à tous les utiliser en parallèle, ce qui va augmenter la vitesse de lecture/écriture (RAID 0).

Par exemple, si on veut éviter de perdre des données suite à la casse d'un disque dur, au lieu d'acheter un disque dur soi-disant indestructible et donc très cher, on peut combiner des disques durs peu chers de façon redondante de sorte à ce que si l'un des disques se casse, les données ne sont pas perdues. Ce type de combinaison est appelé RAID pour "Redundant Arrays of Inexpensive Disks", il y en a de plusieurs types.

1. Documentez-vous sur les divers types de RAID et l'utilisation de LVM pour
 - [https://fr.wikipedia.org/wiki/RAID_\(informatique\)](https://fr.wikipedia.org/wiki/RAID_(informatique))
 - https://en.wikipedia.org/wiki/Standard_RAID_levels (plus clair mais en anglais)
 - <https://linuxfr.org/news/gestion-de-volumes-raid-avec-lvm>
 - `man 7 lvmraid`
2. Créez quelques périphériques boucles, combinez-les selon le RAID de votre choix, montez le dans l'arborescence, ajoutez des fichiers dans le répertoire où est monté votre RAID, détruisez l'un des périphériques boucles tiré au hasard (par exemple en le remplissant de données aléatoires avec un `dd` depuis `/dev/random`), vérifiez que vos données sont intactes (ou pas). Bref, expérimentez pour comprendre comment ça marche.

Bien sur, l'intérêt du RAID apparaît lorsqu'on combine des disques physiques différents. Avec des périphériques boucles dont les fichiers se trouvent sur un même disque physique, une panne du disque pourrait casser tous les `loop devices` d'un coup.

Redimensionnement à chaud, snapshot

LVM possède de nombreuses autres capacités, dont celle de redimensionner "à chaud" (ce qui signifie : "alors que les systèmes de fichiers sont montés dans l'arborescence"). Il offre aussi la possibilité de faire des "snapshots", c'est à dire de faire des photos de votre système de fichiers au temps `t`.

Si vous êtes motivé-e, vous pouvez-vous amuser avec ces fonctionnalités et rapporter vos expériences sur le `mattermost` et le `wiki`.