

Fichiers

Fichiers : plan

On va observer en 3 niveaux

- ▶ arborescence

```
ls -ilah, cd, cp -lar, rm -rf, mv, dd, tar -cxzvf, pwd -P,  
mkdir -p, rmdir, ln -s, readlink, file, du, find, cat, touch
```

- ▶ systèmes de fichiers

```
mkfs -t, fsck, mount, findmnt, stat, df, sync
```

- ▶ périphériques bloc (disque, partition, boucle, chiffrement, lvm, raid)

```
(g)parted, lsblk, cryptsetup, lvm, mdadm, losetup, dmsetup
```

Fichiers : arborescence

Arborescence de répertoires

Les fichiers sont organisés dans une arborescence de répertoires.

Filesystem Hierarchy Standard (FHS)

Elle définit le nom et le contenu des principaux répertoires.

`man 7 hier`

Notations et raccourcis

Notations:

- ▶ / désigne le répertoire racine
- ▶ . désigne le répertoire courant
- ▶ .. désigne le répertoire parent du répertoire courant (/.. = /)

Notations dépendant du shell courant :

- ▶ ~ désigne le répertoire maison de l'utilisateur courant. Souvent /home/<username>. Il est contrôlé par la variable d'environnement HOME et défini dans /etc/passwd.
- ▶ - désigne le répertoire précédemment visité (variable OLDPWD, uniquement pour la commande cd)

Chemin absolu, chemin relatif

Fichiers cachés

La commande `ls` et les gestionnaires de fichiers graphiques ne montrent pas les fichiers dont le nom commence par un point. Ça permet par exemple de rendre lisible le répertoire utilisateur, en cachant l'historique des commandes bash, des fichiers de cache ou de configuration, etc.

L'option `-a` permet à `ls` de ne pas ignorer ces fichiers.

```
man ls
```

Presque tout est fichier

Les fichiers sont une interface qui permet de modéliser de façon uniforme le fait de lire, écrire, dire qui a le droit d'accéder, ... à une ressource. Cette ressource n'est pas nécessairement une suite d'octets inscrits sur une mémoire de masse (disque dur, clef USB, etc).

(démonstration)

Types de fichiers

- ▶ - : fichiers réguliers
- ▶ d : répertoires (en: directories)
- ▶ l : liens symboliques (en: symlinks)
- ▶ p : tubes nommés (en: named pipes)
- ▶ s : socket
- ▶ b : périphériques en mode bloc (en: block device)
- ▶ c : périphériques en mode caractère (en: character device)

Fichiers réguliers

Les fichiers réguliers ne sont pas typés, le filesystem n'a nulle part où indiquer l'utilisation prévue d'un fichier : vidéo, image jpeg, image png, exécutable, document libreoffice, pdf, archive, fichier texte ASCII, fichier texte UTF-8, etc

Certaines applications utilisent l'extension du nom de fichier pour s'y retrouver rapidement, d'autres explorent le début du fichier pour deviner son utilisation possible ("magic number"):

```
man file
```

Les fichiers texte ne spécifient pas leur encodage, donc c'est au programme qui l'utilise de le déterminer comme il peut. Pour changer l'encodage d'un fichier texte:

```
man iconv
```

Liens symboliques

(en: symlink)

C'est l'analogue des raccourcis sous Windows.

C'est un fichier particulier qui contient une référence à un chemin, absolu ou relatif. Lorsqu'on y accède, l'OS accède au fichier ou répertoire représenté par le chemin ("déréférencement"), si celui-ci existe (lorsqu'il n'existe pas, on parle de lien cassé (en: broken link)).

L'option `--no-dereference` de certaines commandes permet d'appliquer la commande au lien symbolique lui-même et non pas à sa référence.

démo.

Tubes nommés et sockets

- ▶ Ils permettent aux processus de communiquer entre eux (dans les deux sens pour les sockets, dans un seul sens pour les tubes).
- ▶ On les trouve en général dans `/run`, mais comme chaque programme fait ce qu'il veut (contrairement aux périphériques qui sont gérés par `udev` dans un système de fichiers spécial monté dans `/dev`), on peut parfois en trouver ailleurs (`/tmp` voire dans le `HOME` de l'utilisateur qui a lancé le programme).
- ▶ Voir le cours de système et réseaux.

Périphériques

Les fichiers permettant d'interagir avec les périphériques se trouvent dans `/dev`.

Exception : les périphériques réseau n'ont pas de fichiers leurs correspondant (pas de `/dev/eth0` ou de `/dev/wlan0`).

Remarque: quand on exécute la commande `ls -l`, la colonne correspondant à la taille du fichier est remplacée par deux nombres :

- ▶ majeur indique quel pilote (en: driver) utilise le noyau pour manipuler (en: handle) le périphérique. La correspondance est écrite dans `/proc/devices`. Une liste étendue se trouve sur : <https://www.kernel.org/doc/html/latest/admin-guide/devices.html>
- ▶ mineur identifie le périphérique, par rapport aux autres du même majeur.

Périphériques en mode caractère

Un *périphérique en mode caractère* (en: *character device*) est un périphérique auquel on accède octet par octet.

Ces fichiers sont des interfaces avec des périphériques qui reçoivent ou émettent des flux de données (souris, camera, terminal, etc), mais aussi la mémoire vive.

Exemples (démonstration):

```
cat /dev/input/mouse0 | hexdump
```

```
echo Hello > /dev/tty1
```

```
echo Hello > /dev/pts/<num>
```

```
vlc v4l2:///dev/video0
```

Périphériques en mode caractère

Il y a aussi certains fichiers spéciaux au comportement étrange:

```
/dev/zero  
/dev/null  
/dev/full  
/dev/random  
/dev/urandom
```

Voir la section 4 du manuel, pour la liste:

```
man 4 [TAB] [TAB]
```

Périphériques en mode bloc

Les périphériques en mode bloc correspondent aux périphériques de stockage (et à leur dérivés), et seront traités spécifiquement.

Le tour du proprio

Demo.

Fichiers : système de fichier

Système de fichiers

Montage

L'arborescence est un patchwork de systèmes de fichiers montés les uns sur les autres.

À la base n'existe que le répertoire racine /. Les systèmes de fichiers se montent (en: mount) sur des répertoires existants, et offrent à leur tour des répertoires, sur lesquels on peut monter de nouveaux systèmes de fichiers de façon récursive.

Démo, avec une clef USB et des fs exotiques (sshfs et wikirofs).

```
findmnt  
mount  
umount  
/etc/fstab
```

Il n'est pas possible de démonter un système de fichier en cours d'utilisation, pour pas rester coincé-e :

```
man lsof  
man fuser
```

Système de fichiers sur périphériques en mode bloc

Désormais, nous nous intéressons aux systèmes de fichiers qui s'installent dans les périphériques en mode bloc.

Comment organiser une suite de blocs d'un périphérique de sorte à pouvoir y stocker et manipuler des répertoires, des fichiers, etc ?

`mkfs`

`fsck`

Inode

- ▶ Un fichier = un inode.
- ▶ Un inode contient les métadonnées (type de fichier, user propriétaire, permissions, etc) et des pointeurs vers son contenu.
- ▶ Le numéro d'inode est l'identifiant unique d'un fichier.
- ▶ Les fichiers n'ont pas de nom, les répertoires nomment les fichiers, un fichier peut être nommé plusieurs fois dans l'arborescence des répertoires.
- ▶ Le contenu d'un fichier est interprété différemment selon son type.

Pour voir les informations contenues dans un inode :

`stat`

À quoi ça ressemble un fs (exemple de ext2) ? Comment agissent les commandes `cp`, `cp -l = ln`, `ln -s`, `mv`, `rm` au niveau du ou des filesystems en jeu.

-> Tableau !

Inode

Attributs stockés dans un inode (penser à `ls -l`) :

- ▶ type du fichier (-, d, l, p, s, b, c)
- ▶ droits d'accès (12 bits)
- ▶ nombre de liens (physiques) sur cet inode (`Nlinks`)
- ▶ numéro de l'utilisateur propriétaire (`uid`)
- ▶ numéro de groupe propriétaire (`gid`)
- ▶ taille du fichier
- ▶ horodatages (en: timestamps)
 - ▶ date/time dernier accès en lecture (`atime` = access time)
 - ▶ date/time création du fichier (`btime` = birth time)
 - ▶ date/time dernière modification de l'inode (`ctime` = change time)
 - ▶ date/time dernière modification du contenu (`mtime` = modify time)
- ▶ des pointeurs vers les blocs de données (pour les gros fichiers, certains de ces blocs peuvent à leur tour contenir des pointeurs vers des blocs de données, etc)
- ▶ (éventuellement des attributs étendus (`xattr`) : ACL, selinux, ...)

Voir : <http://web.mit.edu/tytso/www/linux/ext2intro.html>

Inode et contenu des fichiers

Fichier régulier (fichier de type -)

- ▶ son contenu est le contenu du fichier (pas d'autre information)

Répertoire (fichier de type d) :

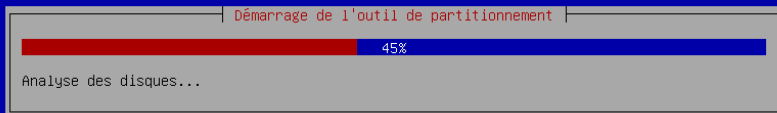
- ▶ son contenu est une liste de paires (nom de fichier, inode correspondant) Une telle paire est appelé lien (en: link) ou lien physique (en: hardlink)
- ▶ En particulier, le nom de fichier est une info du répertoire, et n'est pas contenu dans l'inode du fichier.
- ▶ Les fichiers n'ont pas de nom, ce sont les répertoires qui leur donnent un (ou plusieurs) nom(s).
- ▶ Un même fichier (identifié par son inode) peut apparaître dans plusieurs répertoires et sous plusieurs noms.

Lien symbolique (fichier de type l)

- ▶ son contenu est le chemin vers lequel il pointe

Fichiers : périphériques en mode bloc

Périphériques en mode bloc : motivation



Périphériques en mode bloc : motivation

[!!] Partitionner les disques

Le programme d'installation peut vous assister pour le partitionnement d'un disque (avec plusieurs choix d'organisation). Vous pouvez également effectuer ce partitionnement vous-même. Si vous choisissez le partitionnement assisté, vous aurez la possibilité de vérifier et personnaliser les choix effectués.

Si vous choisissez le partitionnement assisté pour un disque complet, vous devrez ensuite choisir le disque à partitionner.

Méthode de partitionnement :

Assisté - utiliser un disque entier

Assisté - utiliser tout un disque avec LVM

Assisté - utiliser tout un disque avec LVM chiffré

Manuel

<Revenir en arrière>

Périphériques en mode bloc : motivation

[!] Partitionner les disques

Disque partitionné :

SCSI2 (0,0,0) (sda) - ATA VBOX HARDDISK: 8.6 GB

Le disque peut être partitionné selon plusieurs schémas. Dans le doute, choisissez le premier.

Schéma de partitionnement :

Tout dans une seule partition (recommandé pour les débutants)

Partition /home séparée

<Revenir en arrière>

Périphériques en mode bloc : motivation

[!!] Partitionner les disques

Vous devez choisir une phrase secrète pour le chiffrement de SCSI2 (0,0,0), partition n° 5 (sda).

La robustesse du chiffrement dépend fortement de cette phrase secrète. Vous devez donc en choisir une qui ne doit pas être facile à deviner. Elle ne devrait pas correspondre à un mot ou une phrase provenant d'un dictionnaire, ou une phrase pouvant vous être facilement associée.

Une bonne phrase secrète doit être une combinaison de lettres, de chiffres et de signes de ponctuation. Elle devrait comporter au moins 20 caractères.

Phrase secrète de chiffrement :

[] Afficher le mot de passe en clair

<Revenir en arrière>

<Continuer>

Périphériques en mode bloc : motivation

[!!] Partitionner les disques

Voici la table des partitions et les points de montage actuellement configurés. Vous pouvez choisir une partition et modifier ses caractéristiques (système de fichiers, point de montage, etc.), un espace libre pour créer une nouvelle partition ou un périphérique pour créer sa table des partitions.

Partitionnement assisté

Configurer le RAID avec gestion logicielle

Configurer le gestionnaire de volumes logiques (LVM)

Configurer les volumes chiffrés

Configurer les volumes iSCSI

```
Groupe de volumes LVM aligator-vg, volume logique home - 4.0 GB Linux device-mapper (
n° 1      4.0 GB   f  ext4      /home
Groupe de volumes LVM aligator-vg, volume logique root - 3.0 GB Linux device-mapper (
n° 1      3.0 GB   f  ext4      /
Groupe de volumes LVM aligator-vg, volume logique swap_1 - 1.0 GB Linux device-mapper
n° 1      1.0 GB   f  swap      swap
Volume chiffré (sda5_crypt) - 8.1 GB Linux device-mapper (crypt)
n° 1      8.1 GB   K  lvm
SCSI2 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK
n° 1 primaire 510.7 MB   F  ext2      /boot
n° 5 logique  8.1 GB    K  chiffré    (sda5_crypt)
```

Annuler les modifications des partitions

Terminer le partitionnement et appliquer les changements

<Revenir en arrière>

<F1> aide; <Tab> déplacement; <Espace> sélection; <Entrée> activation boutons

Périphériques en mode bloc : motivation

Le parti pris de ce chapitre est de rendre banal ce genre de constructions en l'interprétant de la façon homogène suivante : les disques, les partitions, les volumes logiques LVM, les volumes chiffrés, le raid, les images disque iso, etc ne sont que des block devices obrenus les uns à partir des autres en formant un DAG.

Ainsi, vous serez en mesure de comprendre ce qu'il se passe et d'isoler l'origine du problème quand tout se casse.

Demo.

Périphériques en mode bloc

Un *périphérique en mode bloc* (en: *block device*) est un périphérique auquel on accède (lecture/écriture) par blocs d'octets (en général un bloc mesure entre 512 et 4096 octets).

C'est une abstraction de tout ce sur quoi on peut installer un système de fichiers.

Ils représentent les disques, clefs USB, cartes SD, partitions, cd-roms, et divers assemblages de ceux-ci.

On peut lister les périphériques en mode bloc et leurs relations avec la commande:

```
$ lsblk
```

On obtient une vue arborescente du DAG des relations entre block devices.

Partitionnement de périphériques bloc

On peut partitionner un périphérique bloc (en général un disque) en plusieurs sous périphériques bloc. Ça permet de :

- ▶ faire cohabiter plusieurs systèmes de fichiers
- ▶ d'isoler certaines parties du système
- ▶ faciliter certaines tâches (sauvegardes, mise en lecture seule)

Partitionner un périphérique bloc revient à écrire une table de partitions au début du périphérique indiquant diverses infos (signature, version, checksum, padding, etc) les adresses des partitions qui se trouveront sur le reste du périphérique.

Pour créer une table de partitions:

```
# parted
```

(voir aussi `sfdisk`, `fdisk`, `cfdisk`, `gparted`)

Partitionnement de périphériques bloc

Il y a différents types de tables de partitions avec leurs spécificités, par exemple :

- ▶ dans le cas de msdos, la table de partition tient sur le premier bloc de 512Mo (MBR), de sorte qu'au plus 4 partitions primaires sont possibles, et qu'elle ne peut adresser que des petits disques (2,199 To max).
- ▶ dans le cas de GPT, la table de partitions est répétée en fin de périphérique et le premier bloc est laissé vide.

Cryptsetup (Linux Unified Key Setup, LUKS)

On peut chiffrer un périphérique bloc <block-device> et on obtient un périphérique bloc qui apparaît comme /dev/mapper/<name> :

- ▶ chiffrer le périphérique bloc <block-device> :

```
# cryptsetup luksFormat <block-device>
```

- ▶ déchiffrer un périphérique bloc chiffré <block-device> et le rendre accessible comme un périphérique bloc nommé <name> :

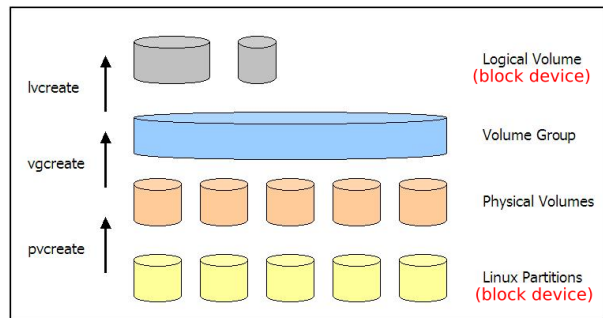
```
# cryptsetup open <block-device> <name>
```

- ▶ désactiver le périphérique bloc <name> :

```
# cryptsetup close <name>
```

Logical Volume Management (LVM2)

On peut aussi recoller des block devices et les redécouper pour en faire d'autres block devices, qui apparaissent dans `/dev/mapper/`.



LVM permet aussi de faire des snapshots (fr:instantanés), redimensionner à chaud, etc.

```
$ man 8 lvm
```

Autocompletion : `pv[TAB]`, `vg[TAB]`, `lv[TAB]`

Redundant Arrays of Inexpensive Disks (RAID)

Permet de combiner plusieurs block devices en un seul de sorte à augmenter les performances et/ou augmenter la tolérance aux pannes (redondance).

[https://fr.wikipedia.org/wiki/RAID_\(informatique\)](https://fr.wikipedia.org/wiki/RAID_(informatique))

https://en.wikipedia.org/wiki/Standard_RAID_levels

```
# apt install mdadm
```

```
# man mdadm
```

Péripheriques boucles (en: loop devices)

On peut promouvoir un fichier régulier en un block device:

```
# losetup -f <fichier>
```

```
# losetup -d /dev/loop<num>
```

par exemple une image ISO d'un CD ou DVD (cf TP virtualbox).

par exemple pour ajouter une partition d'échange (en: swap) rapidement.

par exemple pour récupérer les données d'un disque endommagé en travaillant sur une image pour ne pas abimer le disque encore plus.

par exemple pour se faire la main (cf TP).

Empilement de périphériques bloc

L'ordre dans lequel vous allez empiler les couches dépend de ce que vous voulez faire.

Typiquement, si vous voulez chiffrer votre disque, vous aurez une partition non chiffrée pour /boot mais le reste du disque pourra être chiffré avant d'être séparé en différentes partitions pour le reste de / et une partition pour le swap (partition d'échange qui peut être utilisée comme une extension de la RAM lorsque celle-ci est pleine), de sorte qu'une seule phrase de passe soit à taper. [Attention à ce que tel quel, le design donné dans cet exemple est sensible à l'*Evil maid attack*]

Certaines combinaisons ont peu d'intérêt: par exemple faire du RAID entre des périphériques bloc situés sur le même disque physique, n'augmentera ni la vitesse, ni ne protégera contre un crash du disque.

Périphériques en mode bloc : debriefing

Identifier dans l'écran de validation suivant le DAG des block devices avec leurs tailles, les filesystems et leurs points de montage dans l'arborescence.

Périphériques en mode bloc : debriefing

[!!] Partitionner les disques

Voici la table des partitions et les points de montage actuellement configurés. Vous pouvez choisir une partition et modifier ses caractéristiques (système de fichiers, point de montage, etc.), un espace libre pour créer une nouvelle partition ou un périphérique pour créer sa table des partitions.

Partitionnement assisté

Configurer le RAID avec gestion logicielle

Configurer le gestionnaire de volumes logiques (LVM)

Configurer les volumes chiffrés

Configurer les volumes iSCSI

Groupe de volumes LVM aligator-vg, volume logique home - 4.0 GB Linux device-mapper (

n° 1 4.0 GB f ext4 /home

Groupe de volumes LVM aligator-vg, volume logique root - 3.0 GB Linux device-mapper (

n° 1 3.0 GB f ext4 /

Groupe de volumes LVM aligator-vg, volume logique swap_1 - 1.0 GB Linux device-mapper

n° 1 1.0 GB f swap swap

Volume chiffré (sda5_crypt) - 8.1 GB Linux device-mapper (crypt)

n° 1 8.1 GB K lvm

SCSI2 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK

n° 1 primaire 510.7 MB F ext2 /boot

n° 5 logique 8.1 GB K chiffré (sda5_crypt)

Annuler les modifications des partitions

Terminer le partitionnement et appliquer les changements

<Revenir en arrière>

<F1> aide; <Tab> déplacement; <Espace> sélection; <Entrée> activation boutons

Périphériques en mode bloc : debriefing

Ouverture

- ▶ On a vu que les périphériques de stockage de masse sont accessibles en tant que fichiers de type b dans /dev/,
- ▶ on a vu que des systèmes de fichiers étaient installés dans certains de ces périphériques,
- ▶ on a vu que ces systèmes de fichiers étaient montés les uns sur les autres à partir de / pour constituer ensemble l'arborescence du système.
- ▶ Mais alors le système de fichier installé sur / qui donne naissance aux premiers répertoires /dev, /etc, /home, etc provient d'un filesystem installé dans un block device qui se trouve dans /dev.
- ▶ Y'a pas un serpent qui se mord la queue là ?

Dans un prochain épisode, nous verrons comment cela est possible, et pourquoi la partie "noyau" de la séquence de boot du système se décompose en deux parties.