## Prérequis

Les connaissances nécessaires à la certification LPIC-1 :

☒Notions de base sur les périphériques, partitions et systèmes de fichiers.

☒Édition de fichiers.

☒Commandes de gestion de répertoires et de fichiers.

## Objectifs

À la fin de ce chapitre, vous serez en mesure de :

☒Déterminer les caractéristiques des différents types de systèmes de fichiers Linux.

☒Créer et gérer les systèmes de fichiers Linux.

☒Configurer et gérer l'espace de swap.

☒Administrer le système de fichiers global Linux et superviser les périphériques SMART.

☒Configurer et administrer l'automontage de systèmes de fichiers de périphériques ou via le réseau.

☒Créer des systèmes de fichiers pour les CD-ROM et autres périphériques.

☒Connaître les caractéristiques des systèmes de fichiers chiffrés.

# A. Système de fichiers et périphériques

Ce sujet est divisé en trois parties de poids différents.

## 1. Administration du système de fichiers Linux

| | |
|---|---|
| **Poids** | 4 |
| **Objectifs** | Configurer et gérer le système de fichiers standard Linux, en combinant le montage de plusieurs systèmes de fichiers de types différents. |

### a. Compétences principales

- Configuration par le fichier `fstab`.
- Outils de gestion de l'espace de swap.
- Identification et montage des systèmes de fichiers par leur UUID.
- Compréhension des unités de montage de `systemd`.

### b. Éléments mis en œuvre

- `/etc/fstab`
- `/etc/mtab`
- `/proc/mounts`
- `mount`
- `umount`
- `blkid`
- `sync`
- `swapon`
- `swapoff`

## 2. Maintenance du système de fichiers Linux

| | |
|---|---|
| **Poids** | 3 |
| **Objectifs** | Administrer le système de fichiers Linux avec les outils système pour gérer les types de systèmes de fichiers standards et superviser les périphériques SMART. |

### a. Compétences principales

- Outils de gestion des systèmes de fichiers de type ext2, ext3 et ext4.
- Outils pour la gestion de base des systèmes de fichiers de type Btrfs, y compris les sous-volumes et les snapshots.
- Outils de gestion des systèmes de fichiers de type XFS.
- Connaissance des principes généraux des systèmes de fichiers de type ZFS.

### b. Éléments mis en œuvre

- mkfs (mkfs.*)
- mkswap
- fsck (fsck.*)
- tune2fs, dumpe2fs et debugfs
- btrfs, btrfs-convert
- xfs_info, xfs_check, xfs_repair, xfsdump et xfsrestore
- smartd, smartctl

## 3. Création et configuration de systèmes de fichiers optionnels

| | |
|---|---|
| **Poids** | 2 |
| **Objectifs** | Configuration de l'automontage de systèmes de fichiers de périphériques ou via le réseau, par autofs. Création de systèmes de fichiers pour les CD-ROM et autres périphériques. Notions de système de fichiers chiffré. |

### a. Compétences principales

- Fichiers de configuration d'autofs.
- Unités automount.
- Outils UDF et ISO 9660.
- Autres systèmes de fichiers pour CD-ROM (HFS).
- Extensions pour les systèmes de fichiers CD-ROM (Joliet, Rock Ridge, El Torito).
- Notions de base du chiffrement de données (dm-crypt, LUKS).

### b. Éléments mis en œuvre

- /etc/auto.master
- /etc/auto.[dir]
- mkisofs
- cryptsetup

## B. Administration du système de fichiers Linux

Pour stocker des fichiers et les mettre à disposition des applications, le système Linux utilise une arborescence globale, organisée en répertoires qui peuvent contenir d'autres répertoires et/ou des fichiers.

Les applications peuvent accéder à l'ensemble des fichiers du système de fichiers Linux, sous réserve du contrôle d'accès, sans avoir à connaître l'organisation physique du stockage de ces fichiers (sur un ou plusieurs disques durs, une ou plusieurs partitions de disques, des volumes logiques, sur la machine locale ou une machine distante, etc.).

L'administrateur configure l'organisation de l'arborescence globale du système de fichiers Linux, en utilisant un ou plusieurs systèmes de fichiers pour constituer cette arborescence.

Certains espaces de stockage sont en permanence accessibles à travers l'arborescence, d'autres sont amovibles et peuvent être rendus accessibles dynamiquement, par une commande du système ou de façon automatique (automontage).

Il est possible d'inclure dans l'arborescence globale du système de fichiers Linux des espaces de stockage se trouvant sur d'autres systèmes.

D'autre part, une partie de l'espace physique de stockage des informations doit être réservée pour les opérations de swap de la mémoire vive.

Le sujet 203.1 a pour objectif de s'assurer que le candidat a la maîtrise des opérations de configuration et de gestion du système de fichiers Linux, en combinant des systèmes de fichiers de types différents, permanents ou amovibles, éventuellement chiffrés, ainsi que de la gestion du swap.

### 1. Composants du système de fichiers Linux

L'arborescence du système de fichiers Linux a un répertoire de départ, le répertoire **racine** (*root*), noté /. Elle est constituée d'un ou plusieurs systèmes de fichiers autonomes, chacun monté sur un des répertoires de l'arborescence.

Un système de fichiers permet de structurer un espace de stockage, sous forme de fichiers et de répertoires. Cette organisation interne à l'espace de stockage est rendue accessible aux applications, sous forme de répertoires et de fichiers, par l'opération de **montage** sur un répertoire de l'arborescence globale.

Un système de fichiers non monté est vu comme un ensemble d'octets non structuré, accessible comme un tout, sans qu'il soit possible d'utiliser son organisation en répertoires et fichiers.

Les systèmes de fichiers peuvent être montés au démarrage du système sur l'arborescence globale du système de fichiers, ou pendant son fonctionnement, manuellement ou automatiquement.

Il existe différents types de systèmes de fichiers, qui peuvent être combinés entre eux au sein de l'arborescence globale du système de fichiers Linux.

### 2. Systèmes de fichiers physiques

Ces systèmes de fichiers sont associés à de l'espace de stockage physique. Ils sont organisés en répertoires et fichiers, qui sont stockés sur divers supports physiques : partitions de disques durs, volumes logiques, périphériques amovibles (CD-ROM, clé USB, etc.), espaces disques gérés par des systèmes distants (NFS, CIFS, SAN, NAS, etc.).

☞*Il existe de nombreux types de systèmes de fichiers physiques supportés par Linux, en particulier ext2, ext3, ext4, XFS, Btrfs et ZFS, qui seront décrits dans le cadre du sujet 203.2.*

## 3. Systèmes de fichiers virtuels

Certains types de systèmes de fichiers sont dits **virtuels**, car ils ne sont pas associés à un espace de stockage, mais sont directement gérés en mémoire vive. Ils doivent être montés pour donner accès à leurs répertoires et fichiers, mais leur durée de vie est limitée à leur période de montage : quand ils sont démontés (explicitement ou suite à l'arrêt du système), leur contenu est perdu.

Le type de système de fichiers virtuel tmpfs est conçu pour permettre de gérer en mémoire vive les fichiers temporaires. Il est utilisé par défaut par certaines distributions pour le contenu du répertoire `/tmp`. Ce mécanisme permet d'optimiser les temps d'accès pour ces fichiers qui n'ont pas vocation à être stockés de façon permanente.

Linux propose deux systèmes de fichiers virtuels standards, **proc** et **sys**, dont le but est de faciliter la communication avec le noyau. Ce dernier expose dynamiquement, sous forme de fichiers, des éléments de sa configuration et des objets qu'il gère (processus, fichiers, etc.). Ces deux systèmes de fichiers sont montés par défaut sur les répertoires `/proc` et `/sys` de l'arborescence globale du système de fichiers.

### a. Le système de fichiers virtuel proc

Ce système de fichiers virtuel, de type **proc**, est monté par défaut sous le répertoire `/proc`. Il est géré dynamiquement par le noyau pour permettre de suivre l'état des processus actifs. Chaque processus est associé à un répertoire dont le nom est l'identifiant du processus. Quand le processus se termine, le répertoire est supprimé. Ce mécanisme permet de suivre les différents attributs des processus du système.

Ce système de fichiers virtuel donne également accès à de nombreuses tables de contrôle du noyau : partitions, montages, utilisation de la mémoire, etc. Il fournit également des informations concernant les caractéristiques du matériel pris en charge par le noyau.

Exemples

*On lance l'éditeur Vim depuis une connexion SSH et on visualise les caractéristiques du processus ainsi créé depuis une autre connexion, dans le répertoire* `/proc` *:*

```
vi /etc/hosts
```

*Depuis un autre terminal :*

```
ps -ef | grep vim
pba       2348  2242  1 18:26 pts/0    00:00:00 vim /etc/hosts
```

*On cherche le répertoire correspondant au processus de PID 2348, dans* `/proc` *:*

```
ls -ld /proc/2348
dr-xr-xr-x. 9 pba pba 0 25 févr. 18:26 /proc/2348
```

*Contenu du répertoire :*

```
ls /proc/2348
```

```
attr              exe         mounts          projid_map    status
autogroup         fd          mountstats      root          syscall
auxv              fdinfo      net             sched         task
cgroup            gid_map     ns              schedstat     timers
clear_refs        io          numa_maps       sessionid     timerslack_ns
cmdline           limits      oom_adj         setgroups     uid_map
comm              loginuid    oom_score       smaps         wchan
coredump_filter   map_files   oom_score_adj   smaps_rollup
cpuset            maps        pagemap         stack
cwd               mem         patch_state     stat
environ           mountinfo   personality     statm
```

*Tous ces fichiers contiennent des informations sur le processus exécutant Vim. Par exemple, on peut lire la ligne de commande exécutée par le processus :*

```
cd /proc/2348
cat cmdline
```

```
vim/etc/hosts
```

*On peut connaître le répertoire courant du processus :*

```
ls -ld cwd
```

```
lrwxrwxrwx. 1 pba pba 0 25 févr. 18:29 cwd -> /home/pba
```

*cwd est un lien symbolique vers le répertoire courant du processus.*

*On sort du répertoire associé au processus et on quitte Vim dans l'autre terminal :*

```
cd ..
ls 2348
```

```
ls: impossible d'accéder à '2348': Aucun fichier ou dossier de ce type
```

*Dès que le processus est terminé, le noyau supprime le répertoire correspondant dans le système de fichiers virtuel proc.*

*Dans le système de fichiers virtuel proc, on dispose de fichiers et de répertoires concernant différents éléments matériels et logiciels gérés par le noyau :*

*Pour connaître la version du noyau :*

```
cat version
```

```
Linux version 4.18.0-147.el8.x86_64 (mockbuild@kbuilder.bsys.centos.org)
(gcc version 8.3.1 20190507 (Red Hat 8.3.1-4) (GCC)) #1 SMP Wed Dec 4 21:51:45
UTC 2019
```

*Pour les informations concernant le (ou les) processeur(s) :*

```
cat cpuinfo
processor : 0
vendor_id : GenuineIntel
cpu family : 6
model  : 142
model name : Intel(R) Core(TM) i3-7020U CPU @ 2.30 GHz
[...]
```

### b. Le système de fichiers virtuel sys

Ce système de fichiers virtuel, de type **sysfs**, est monté par défaut sous le répertoire `/sys`. Il fournit des informations concernant les caractéristiques des différents périphériques. Certains des pseudo-fichiers sont accessibles en écriture (sous contrainte des droits d'accès) et permettent de modifier dynamiquement des paramètres.

Exemple

*Sur une distribution Debian :*

```
ls /sys
block  class  FicSpecials   fs          kernel  power
bus    dev    firmware  hypervisor  module
```

*Ces répertoires contiennent différents répertoires et fichiers, permettant de connaître les différents éléments matériels et logiciels gérés par le noyau.*

```
ls /sys/dev/block
11:0  254:0  254:1  254:2  254:3  254:4  8:0  8:1  8:2  8:5
```

*Chaque répertoire de `/sys/dev/block` correspond à un périphérique associé à un fichier spécial en mode bloc. Le nom est composé du numéro de majeur et du numéro de mineur du périphérique :*

```
cd /sys/dev/block/11:0
cat device/model
CD-ROM
```

*Le fichier spécial de majeur `11` et de mineur `0` est associé au lecteur de CD-ROM de la machine.*

## 4. Identification des systèmes de fichiers

L'identification des systèmes de fichiers peut se faire de différentes manières : fichier spécial associé, chemin d'accès réseau, chemin d'accès de fichier, label, UUID.

### a. Fichier spécial en mode bloc

C'est la méthode traditionnelle Unix, reprise dans Linux. Elle permet d'identifier de façon unique le système de fichiers contenu dans l'espace de stockage associé au fichier spécial en mode bloc.

☞*Dans certains cas, la dénomination du fichier spécial correspondant au périphérique peut changer, par exemple si un disque physique est changé de connecteur, ou si l'ordre de détection des périphériques réseau est modifié.*