Travailler avec les systèmes de fichiers

Hiérarchie du système de fichiers

Slackware Linux conserve tous les fichiers et répertoire sous un unique répertoire `/`, communément dénommé *“root”* (racine). C'est la principale différence par rapport à Microsoft Windows. Pour toute partition de disque dur, CD-ROM, clé USB et même disquette, tout sera monté en tant que répertoire sous `/` et vous n'aurez aucune *“lettre de lecteur”*. Les périphériques peuvent être monté théoriquement n'importe où, mais Slackware défini des paramètres corrects par défaut pour vous. Par exemple, les disques CD-RW se retrouvent le plus souvent sous `/mnt/cd-rw`. Voici une courte liste des répertoires les plus courants présents dans n'importe quelle installation par défaut de Slackware Linux, et ce que vous pouvez y trouver.

Table 11.1. Organisation du système de fichiers

/	Répertoire racine, auquel tous les autres sont rattachés
/bin	Ensemble minimum de programmes binaires pour tous les utilisateurs
/boot	Le noyau, initrd et autres composants nécessaires pour démarrer Slackware
/etc	Fichiers de configuration du système
/dev	Ensemble de fichiers spéciaux permettant l'accès direct au matériel
/home	Répertoires des utilisateurs pour leurs données et configuration personnelles
/media	Répertoire pour le montage automatique de disques via DBUS/HAL
/mnt	Emplacement pour le montage temporaire de supports amovibles
/opt	Répertoire destiné à l'installation de logiciels tiers (le plus souvent propriétaire)
/proc	Système de fichier généré par le noyau pour les informations sur les processus
/root	Répertoire personnel de l'utilisateur root
/sbin	Ensemble minimum de programmes binaires pour le super-utilisateur
/srv	Données spécifiques accessibles sur ce système, tel que des pages web
/sys	Informations spécifiques à l'exécution du noyau
/tmp	Répertoire destiné aux fichiers temporaires, pour tous les utilisateurs
/usr	Tous les programmes, bibliothèques et fichiers partagés non essentiels
/var	Données volatiles, tels que les fichiers journaux

Systèmes de fichiers locaux

Le noyau Linux est capable de gérer un large ensemble de systèmes de fichiers, vous permettant de choisir parmi une longue liste de fonctionnalités celui le plus adapté à vos besoins. Heureusement, la plupart des systèmes de fichiers par défaut sont adaptés quelques soient vos besoins. Certains systèmes de fichiers ne sont conçus que pour des médias particuliers. Par exemple, le système de fichiers iso9660 n'est pratiquement utilisé que pour les CD et DVD.

ext2

ext2 est le plus ancien système de fichiers proposé par Slackware Linux pour conserver des données

sur des disques durs. Comparé aux autres systèmes de fichiers, ext2 est simpliste. Il est plus rapide que la plupart des autres pour la lecture et l'écriture de données mais il est dépourvu de fonction de journalisation. Cela signifie qu'après un crash le système de fichier doit être intégralement vérifié pour rechercher et (si tout va bien) corriger les erreurs rencontrées.

ext3

ext3 est le jeune cousin d'ext2. Il a été conçu pour remplacer ext2 dans la plupart des cas et partage beaucoup de code, mais en ajoutant le support de la journalisation. De fait, ext3 et ext2 sont si proches qu'il est possible d'effectuer une conversion de l'un à l'autre en fonctionnement sans perte de données. ext3 jouit d'une large popularité pour cela. De nombreux outils sont aussi disponibles pour la récupération de données depuis ce système de fichier en cas de problème matériel majeur. ext3 est un bon système de fichiers générique avec journalisation, mais ne supporte pas la comparaison face à d'autres dans les cas particuliers. Un inconvénient d'ext3 est que le système de fichiers doit être vérifié de manière exhaustive de temps à autre. Ce s'effectue lorsque le système de fichiers est monté, habituellement lorsque l'ordinateur démarre, et ajoute donc un délai supplémentaire.

ext4

ext4 est le dernier système de fichiers de la lignée des ext. Il a été conçu à partir d'ext3 avec de nouvelles idées sur ce qu'un système de fichiers doit apporter. Bien que Slackware supporte ext4, vous devez garder à l'esprit que ce système de fichier est encore jeune (particulièrement pour ce qui concerne les fichiers système) et qu'il est toujours en développement. Si vous privilégiez la stabilité aux performances vous préférerez sûrement utiliser un autre système de fichier tel qu'ext3. Cela dit, ext4 surpasse ext3 dans le domaine des performances, mais beaucoup de personnes ne lui font pas encore assez confiance pour ce qui est de la stabilité.

reiserfs

reiserfs est l'un des plus anciens systèmes de fichiers journalisés du noyau Linux et qui a été supporté par Slackware pendant de longues années. C'est un système de fichiers très rapide particulièrement adapté pour conserver, retrouver et écrire de nombreux fichiers de petite taille. Malheureusement, il y a peu d'outils pour récupérer des données lorsque vous subissez une panne de disque, et les partitions reiserfs sont plus fréquemment sujettes aux corruptions qu'ext3.

XFS

XFS est une contribution de SGI au noyau Linux et est l'un des meilleurs systèmes de fichiers pour travailler avec de larges volumes et des fichiers volumineux. XFS utilise plus de RAM que d'autres systèmes de fichiers, mais si vous avez besoin de travailler avec de gros fichiers ses performances compenseront sa faiblesse d'utilisation mémoire. XFS n'est pas très adapté pour une utilisation avec un ordinateur de bureau ou un portable, mais brille réellement sur un serveur qui gère des fichiers moyens ou gros tout au long de la journée. Comme ext3, XFS est un système de fichiers entièrement journalisé.

JFS

JFS est une contribution d'IBM au noyau Linux et est réputé pour sa réactivité, même dans des conditions extrêmes. Il peut s'étendre sur des volumes colossaux le rendant particulièrement adapté pour des NAS (*Network Attached Storage*). La longue histoire et les tests approfondis de JFS en font un des systèmes de fichiers journalisés les plus robustes disponibles sous Linux.

iso9660

iso9660 est un système de fichiers spécialement conçu pour les supports optiques tels que les CD et DVD. Comme les support optiques sont des médias en lecture seule, le noyau Linux n'inclue pas de support en écriture pour ce système de fichiers. Afin de créer des systèmes de fichiers iso9660 vous devez employer des outils en mode utilisateur (*user-land*) tels que ***mkisofs***(8) ou ***growisofs***(8).

vfat

Parfois vous pouvez avoir besoin de partager des données entre des ordinateurs Windows et Linux, mais sans pouvoir utiliser de réseau. À la place vous pouvez partager une partition de disque dur ou une clé USB. Le simple système de fichiers vfat est alors le meilleur choix car il est supporté par une large variété de systèmes d'exploitation. Malheureusement, étant un système de fichiers conçu par Microsoft, il ne conserve pas les permissions de manière similaire aux systèmes de fichiers habituels sous Linux. Cela signifie que des options spéciales doivent être utilisées pour autoriser des utilisateurs distincts à accéder aux données sur ce système de fichiers.

swap

Au contraire des autres systèmes de fichiers qui gèrent des fichiers et répertoires, les partitions d'échange (*swap*) gèrent de la mémoire virtuelle. Cela est très utile et évite au système de crasher si toute la RAM est utilisée. À la place, le noyau copie des pages de RAM dans le swap et les libère pour permettre leur utilisation par d'autres applications. Représentez-vous cela comme de la mémoire virtuelle supplémentaire pour votre ordinateur, de la mémoire virtuelle très lente. Une partition d'échange est souvent un garde-fou et vous ne devriez pas en dépendre pour un usage régulier. Ajoutez de la RAM à votre système si vous voyez que vous utilisez beaucoup de swap.

Utiliser mount

Maintenant que nous avons plus ou moins appris quels étaient les différents systèmes de fichiers disponibles sous Linux, il est temps de savoir comment les utiliser. Afin de pouvoir lire ou écrire des données sur un système de fichiers, celui doit être monté en premier lieu. Pour Cela nous utiliserons (naturellement) la commande mount(8). La première chose que nous devons décider est où nous voulons que le système de fichier soit intégré. Souvenez-vous qu'il n'y a pas d'équivalent aux lettres de lecteurs pour les systèmes de fichiers sous Linux. À la place, tous les systèmes de fichiers sont montés dans des répertoires. Le système de fichiers de base sur lequel vous avez installé Slackware est toujours / et les autres sont toujours dans des sous-répertoires de /. /mnt/hd est un endroit

habituel pour accueillir temporairement une partition, nous l'utiliserons donc dans notre premier exemple. Pour monter le contenu d'un système de fichiers nous devons indiquer quel système de fichiers est utilisé, où le monter et quelles options spécifiques utiliser.

```
darkstar:~# mount -t ext3 /dev/hda3 /mnt/hd -o ro
```

Disséquons cela. Mous avons un système de fichiers ext2 sur la troisième partition du premier disque IDE et nous avons décidé de monter sont contenu dans le répertoire /mnt/hd. De plus, nous l'avons monté en lecture seule, de cette manière personne ne peut en modifier le contenu. L'option `-t ext3` indique quel système de fichiers nous utilisons, dans ce cas ext3. Cela permet au noyau de savoir quel pilote utiliser. Le plus souvent mount peut déterminer cela de lui même, mais cela ne fait pas de mal de l'indiquer explicitement. Ensuite, nous disons à mount où trouver le contenu du système de fichiers. Ici nous avons choisi /mnt/hd. Enfin, nous devons décider d'options à utiliser si nécessaire. Celles-ci sont déclarée après l'option `-o`. Une courte liste des options les plus courantes est présentée ci-dessous.

Table 11.2. Options courantes de mount

ro	lecture seule (*read-only*)
rw	lecture et écriture (*read-write*) (utilisé par défaut)
uid	utilisateur propriétaire du contenu du système de fichiers
gid	groupe propriétaire du contenu du système de fichiers
noexec	empêche l'exécution de tout fichier du système de fichiers
defaults	options convenables pour la plupart des systèmes de fichiers

Si ceci est votre première installation Linux, les seules options dont vous aurez besoin sont *ro* et *rw*. L'exception à cette règle s'appliquera lorsque vous manipulerez des systèmes de fichiers qui ne gèrent pas les permissions Linux habituelles, tels que vfat ou NTFS. Dans ces cas, vous aurez besoin d'utiliser les options *uid* ou *gid* pour autoriser les utilisateurs non-root à accéder à ces systèmes de fichiers.

```
darkstar:~# mount -t vfat /dev/hda4 /mnt/hd -o uid=alan
```

Mais Alan, c'est épouvantable ! Je ne veux pas indiquer à mount quel système de fichiers ou quelles options utiliser à chaque fois que je monte un CD. Ce doit être plus simple que cela. Heureusement, oui on peut faire plus simple. Le fichier `/etc/fstab` contient toutes les informations pour les systèmes de fichiers définies pour vous par l'installateur et vous pouvez y ajouter vos propres entrées. `fstab` (5) ressemble à un simple tableau contenant le périphérique à monter ainsi que le type de son système de fichiers et arguments optionnels. Jetons-y un œil.

```
darkstar:~# cat /etc/fstab
/dev/hda1       /               reiserfs    defaults              1   1
/dev/hda2       /home           reiserfs    defaults              1   2
/dev/hda3       swap            swap        defaults              0   0
/dev/cdrom      /mnt/cdrom      auto        noauto,owner,ro,users 0   0
/dev/fd0        /mnt/floppy     auto        noauto,owner          0   0
devpts          /dev/pts        devpts      gid=5,mode=620        0   0
proc            /proc           proc        defaults              0   0
```

Si vous avez une entrée dans `fstab` pour votre système de fichiers, vous avez juste besoin d'indiquer

à mount le fichier de périphérique ou le point de montage.

```
darkstar:~# mount /dev/cdrom
darkstar:~# mount /home
```

Une dernière utilisation possible pour ***mount*** est d'afficher la liste des systèmes de fichiers actuellement monté et avec quelles options. Exécutez simplement ***mount*** sans aucun argument pour cela.

Systèmes de fichiers réseau

En plus des systèmes de fichiers locaux, Slackware supporte plusieurs systèmes de fichiers réseau autant en client que comme serveur. Cela vous permet de partager des données avec plusieurs ordinateurs de manière transparente. Nous présenterons les deux plus répandus : NFS et SMB.

NFS

NFS (*Network File System*) est le système de fichiers réseau sous Linux ainsi que pour plusieurs autres systèmes d'exploitation. Ses performances sont modestes mais il supporte toutes les permissions utilisées par Slackware. Afin de pouvoir utiliser NFS en tant que client ou serveur vous devez exécuter le daemon d'appel de procédures à distance (*Remote Procedure Call* - RPC). Cela se fait simplement en attribuant le droit d'exécution au fichier `/etc/rc.d/rc.rpc` et en le lançant. Une fois rendu exécutable il sera lancé à chaque fois que vous démarrerez Slackware.

```
darkstar:~# chmod +x /etc/rc.d/rc.rpc
darkstar:~# /etc/rc.d/rc.rpc start
```

Monter un partage NFS est un peu différent de monter un système de fichiers local. Au lieu d'indiquer un périphérique local, vous devez donner à mount le nom de domaine ou l'adresse IP du serveur NFS et le répertoire à monter séparés par le caractère deux points.

```
darkstar:~# mount -t nfs darkstar.example.com:/home /home
```

Activer un serveur NFS est un peu plus différent. En premier lieu, vous devez configurer chaque répertoire devant être exporté dans le fichier `/etc/exports`. `exports`(5) contient des informations à propos des répertoires à partager, qui pourra accéder au partage et quelles permissions spéciales à attribuer ou à révoquer.

```
# See exports(5) for a description.
# This file contains a list of all directories exported to other computers.
# It is used by rpc.nfsd and rpc.mountd.

/home/backup    192.168.1.0/24(sync,rw,no_root_squash)
```

La première colonne dans `exports` est la liste des fichiers à exporter par NFS. La deuxième colonne est la liste des systèmes qui peuvent accéder au partage ainsi que des permissions spéciales. Vous pouvez indiquer un hôte par son nom de domaine, adresse IP ou plage d'adresses (ce que j'ai dans le

cas présent). Les permissions spéciales sont toujours entre parenthèses. Pour une liste exhaustive vous aurez besoin de lire la page de manuel. Pour l'instant, la seule permission spéciale importante est *no_root_squash*. Habituellement l'utilisateur root ne peut pas accéder ou modifier les données d'un partage. À la place, l'utilisateur root est “écrasé” (*squashed*) et obligé d'agir en tant qu'utilisateur *nobody*. *no_root_squash* empêche cela.

Vous aurez aussi besoin d'exécuter le daemon NFS. La gestion du démarrage et de l'arrêt du serveur NFS se fait via le script `/etc/rc.d/rc.nfsd`. Rendez-le exécutable et lancez-le de la même manière que `rc.rpc` et vous êtes prêts.

SMB

SMB est le protocole de partage de fichier réseau de Windows. Se connecter à un partage SMB (souvent appelé partages Samba) est relativement aisé. Malheureusement, SMB n'est pas aussi bien supporté que NFS. Cependant il offre de grande performances et la connectivité avec les ordinateurs sous Windows. Pour ces raisons, SMB est le protocole de partage de fichier réseau le plus déployé sur les réseaux locaux. Exporter des partages SMB depuis Slackware se fait via le daemon Samba et configuré dans `smb.conf`(5). Malheureusement, configurer Samba en tant que service est au-delà du cadre de ce livre. Recherchez en ligne pour plus de documentation et référez-vous toujours à la page de manuel.

Heureusement, monter un partage SMB est facile et fonctionne quasiment comme pour monter un partage NFS. Vous devez indiquer à mount où trouver le serveur et quel partage vous souhaiter accéder exactement de la même façon. De plus vous devez indiquer un nom d'utilisateur et un mot de passe.

```
darkstar:~# mount -t cifs //darkstar/home /home -o
username=alan,password=secret
```

Vous devez vous demander pourquoi le type système de fichiers est cifs au lieu de smbfs. Avec d'anciennes versions du noyau Linux, on utilisait smbfs. Cela est tombé en désuétude en faveur d'un pilote cifs plus générique, plus performant et plus sécurisé.

Tous les partages SMB requièrent un nom d'utilisateur (*username*) et un mot de passe (*password*) comme options. Cela peut poser un problème de sécurité si vous souhaitez placer un partage Samba dans fstab. Vous pouvez éviter ce problème en utilisant l'option *credentials*. *credentials* indique un fichier qui contient les informations pour le nom d'utilisateur et le mot de passe. Aussi longtemps que ce fichier est soigneusement protégé et lisible uniquement par root, la vraisemblance que vos informations d'authentification soient compromises est faible.

```
darkstar:~# echo "username=alan" > /etc/creds-home
darkstar:~# echo "password=secret" >> /etc/creds-home
darkstar:~# mount -t cifs //darkstar/home -o credentials=/etc/creds-home
```

Navigation

Chapitre précédent : Permissions des systèmes de fichiers

Chapitre suivant : Vi

Sources

- Source originale : http://www.slackbook.org/beta

- Publication initiale d'Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traduction initiale de escaflown
- Traduction de Ellendhel

slackbook, filesystem, network filesystems, nfs, smb, mount