

Systeme d'exploitation Systeme de gestion de fichiers

Florent DEVIN

27 novembre 2009

1 Préambule

Dans ce TP, si les commandes sont écrites en bleu, c'est qu'il faut les taper en tant qu'utilisateur lambda. Si les commandes apparaissent en noir, c'est qu'il faut que vous tapiez les commandes en tant que super utilisateur.

2 Montage d'une clef USB

Tapez la commande :

```
dmesg
```

Introduisez la clef USB, attendez un peu, puis tapez de nouveau la commande :

```
dmesg
```

Vous devriez obtenir quelque chose qui ressemble à :

```
[ 2148.425539] usbcore: registered new interface driver libusual
[ 2148.438148] Initializing USB Mass Storage driver...
[ 2148.438251] scsi6 : SCSI emulation for USB Mass Storage devices
[ 2148.438345] usbcore: registered new interface driver usb-storage
[ 2148.438348] USB Mass Storage support registered.
[ 2148.438475] usb-storage: device found at 4
[ 2148.438477] usb-storage: waiting for device to settle before scanning
[ 2153.437387] usb-storage: device scan complete
[ 2153.438408] scsi 6:0:0:0: Direct-Access      hp          DriveKey II      1.01 PQ: C
[ 2153.444359] sd 6:0:0:0: [sdb] 503808 512-byte hardware sectors (258 MB)
[ 2153.445486] sd 6:0:0:0: [sdb] Write Protect is off
[ 2153.445497] sd 6:0:0:0: [sdb] Mode Sense: 23 00 00 00
[ 2153.445503] sd 6:0:0:0: [sdb] Assuming drive cache: write through
```

```
[ 2153.447108] sd 6:0:0:0: [sdb] 503808 512-byte hardware sectors (258 MB)
[ 2153.458909] sd 6:0:0:0: [sdb] Write Protect is off
[ 2153.458923] sd 6:0:0:0: [sdb] Mode Sense: 23 00 00 00
[ 2153.458929] sd 6:0:0:0: [sdb] Assuming drive cache: write through
[ 2153.461117] sdb: sdb1
[ 2153.469566] sd 6:0:0:0: [sdb] Attached SCSI disk
[ 2153.477217] sd 6:0:0:0: Attached scsi generic sg2 type 0
```

L'antépénultième ligne vous indique le périphérique utilisable pour monter votre clef USB. Il faut donc si vous souhaitez monter le clef USB, rattacher ce périphérique au système de fichier. Pour ce faire, il suffit d'utiliser la commande `mount`.

Il y a plusieurs façons d'utiliser la commande `mount`. Soit vous l'utilisez en ligne de commande, soit vous l'utilisez via un fichier de configuration.

2.1 Ligne de commande

Pour utiliser la commande `mount`, il vous faut être `root`. Tapez donc la commande :

```
su -
```

Cette commande (**Switch User**) vous permet d'utiliser l'identité d'un autre utilisateur¹. Sans argument, elle équivaut à demander l'utilisation du compte `root`. Le `-` après la commande indique que vous souhaitez utiliser l'environnement de l'utilisateur². Après avoir entré la commande `su`, vous devez entrer le mot de passe de `root`. Une fois ceci fait, vous êtes `root`. Vous pouvez donc utiliser la commande `mount`. En supposant que le périphérique associé à votre clef USB est le périphérique `sdb`, tapez la commande :

```
mount -t auto /dev/sdb1 /tmp/schmumbrle
```

Cette commande doit provoquer une erreur. Ceci est normal, car la commande `mount` permet de rattacher un système de fichier à l'arborescence de Linux. L'option `-t auto` indique que vous souhaitez que linux détermine automatiquement le système de fichiers présent. `/dev/sdb1` indique que vous voulez monter la première partition du disque `sdb`. Enfin, le dernier argument de la commande indique là où vous souhaitez rattacher votre périphérique. Créez un répertoire `/media/usb`, puis montez votre clef USB dans ce répertoire.

```
mkdir -p /media/usb
mount -t auto /dev/sdb1 /media/usb
```

¹Pour peu que vous connaissiez le mot de passe.

²Ce qui signifie la plupart du temps charger son fichier `.bashrc`

Une fois ceci fait, vous avez accès à votre clef USB. Pour la suite du TP, je vous recommande de sauvegarder toutes les données qui se trouvent sur la clé dans un répertoire. Nous allons, dans ce TP, modifier le système de fichiers, et les partitions. Ce qui fait que les données sur cette clef seront effacées. Une fois ceci fait démontez la clef USB :

```
umount /media/usb
```

2.2 Fichier de configuration

Nous pouvons automatiser le montage des périphériques via un fichier de configuration. Ce fichier n'est modifiable que par *root*. Éditez le fichier `/etc/fstab`. Le listing 1 représente un fichier `fstab`. Dans ce fichier, les informations apparaissent clairement. Le réglage initial est relativement simple. Toute la problématique est d'utiliser correctement les options disponibles.

Listing 1 – Extrait d'un fichier `fstab`

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
# /dev/sda1
/dev/sda1 / ext3 relatime,errors=remount-ro 0 1
# /dev/sda9
/dev/sda9 /data ext3 relatime 0 2
# /dev/sda8
/dev/sda8 /home ext3 relatime 0 2
# /dev/sda7
/dev/sda7 /tmp ext3 relatime 0 2
# /dev/sda6
/dev/sda6 /usr ext3 relatime 0 2
# /dev/sda5
/dev/sda5 /var ext3 relatime 0 2
# /dev/sda2
/dev/sda2 none swap sw 0 0
/dev/scd0 /media/cdrom0 udf,iso9660 user,noauto,exec,utf8 0 0
//odysseus/Public /home/fd/Public smbfs credentials=/etc/.credential
,users,noauto 0 0
```

Rajouter à la fin du fichier `/etc/fstab`, la ligne suivante :

```
/dev/sdb1 /media/usb auto noauto 0 0
```

Cette ligne permet la prise en compte d'un nouveau périphérique (`/dev/sdb1`³) et de rattacher ce périphérique au système de fichiers, dans le répertoire `/media/usb`. Pour que cette prise en compte puisse fonctionner, il faut bien évidemment que le répertoire `/media/usb` existe. Le paramètre suivant est le type de système de fichier. Le paramètre `noauto` indique au système qu'il ne faut pas automatiquement monter le périphérique au démarrage.

En tant qu'utilisateur `lambda`, ouvrez un terminal. Dans ce terminal, tapez la commande :

³C'est à dire votre clef USB

```
mount /media/usb
```

Le système vous répond :
mount : only root can mount /dev/sdb1 on /media/usb.

Si vous tapez la commande précédente dans le terminal où vous êtes `root`, vous vous apercevez que vous pouvez effectivement monter le périphérique. Nous allons modifier le fichier `/etc/fstab` pour que tous les utilisateurs puissent monter le périphérique. Dans un premier temps, démontez le périphérique :

```
umount /media/usb
```

Maintenant changez la ligne que nous avons rajoutée, par la ligne suivante :

```
/dev/sdb1 /media/usb auto noauto , users , exec 0 0
```

En modifiant cette ligne, nous venons juste de permettre qu'un utilisateur puisse monter le périphérique (`users`). Nous avons aussi permis l'exécution de programme depuis ce système de fichiers (`exec`).

3 Création de partitions

Dans cette partie, nous allons nous intéresser à la création, et le montage de différents types de système de fichiers. Pour créer des partitions, il existe plusieurs façons. Nous allons utiliser la ligne de commande pour ce faire. Il existe des logiciels en mode graphique, mais souvent quand on a besoin de modifier les partitions, nous n'avons pas accès à ce type de logiciel. Nous allons considérer dans la suite de ce TP, que la clef USB se trouve sur `/dev/sdb`. Si ce n'est pas le cas pour vous, il vous appartient d'adapter le contenu du TP. *Dans tous les cas, ne modifiez jamais la table des partitions de `/dev/sda`.* Bien évidemment la manipulation des tables de partitions n'est accessible qu'en tant que super utilisateur. Lancez la commande :

```
fdisk /dev/sdb
```

Appuyer sur la lettre `p`, puis validez. Ceci affiche la table des partitions de la clef USB. Nous allons effacer cette partition, et en créer de nouvelles. Effacez la partition courante. Ceci se fait en entrant : `d` Vérifiez que vous n'avez plus de partition en affichant la table de partition grâce à la commande `p`. Cela devrait vous afficher :

```
Command (m for help): p
```

```
Disk /dev/sdb: 1999 MB, 1999044096 bytes  
32 heads, 63 sectors/track, 1936 cylinders
```

```
Units = cylinders of 2016 * 512 = 1032192 bytes
Disk identifier: 0x4491216e
```

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

```
Command (m for help):
```

Nous allons créer plusieurs partitions. Dans un premier temps, nous allons créer une partition primaire de 500 méga. Tapez la séquence suivante :

```
n
p
1
1
500
```

Vérifiez que vous avez ceci :

```
Command (m for help): p
```

```
Disk /dev/sdb: 1999 MB, 1999044096 bytes
32 heads, 63 sectors/track, 1936 cylinders
Units = cylinders of 2016 * 512 = 1032192 bytes
Disk identifier: 0x4491216e
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	500	503968+	83	Linux

```
Command (m for help):
```

Créons trois autres partitions. En fait sur un disque vous ne pouvez créer que trois partitions primaires, les autres doivent être des partitions logiques. Dans un premier temps, nous allons indiquer que nous souhaitons créer des partitions logiques. Pour créer de telles partitions, il faut avant créer une partition étendue. Cela se fait grâce à la séquence :

```
n
e
2
<entrée>
<entrée>
```

Ensuite nous allons créer trois partitions logiques. Pour ce faire, faites les séquences suivantes :

```
n
1
```

```
<entrée>  
1000
```

```
n  
l  
<entrée>  
1500
```

```
n  
l  
<entrée>  
<entrée>
```

Une fois toutes les séquences réalisées, vous devriez avoir :

```
Command (m for help): p  
  
Disk /dev/sdb: 1999 MB, 1999044096 bytes  
32 heads, 63 sectors/track, 1936 cylinders  
Units = cylinders of 2016 * 512 = 1032192 bytes  
Disk identifier: 0x4491216e  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sdb1             1           500     503968+   83  Linux  
/dev/sdb2             501        1936    1447488    5  Extended  
/dev/sdb5             501        1000     503968+   83  Linux  
/dev/sdb6            1001        1500     503968+   83  Linux  
/dev/sdb7            1501        1936    439456+   83  Linux  
  
Command (m for help):
```

Nous allons changer le type des partitions pour en avoir une de type FAT 32, et une de type NFS. Pour ce faire entrez les séquences suivantes :

```
t  
5  
b  
t  
6  
7
```

Vous obtenez ceci :

```
Command (m for help): p  
  
Disk /dev/sdb: 1999 MB, 1999044096 bytes  
32 heads, 63 sectors/track, 1936 cylinders  
Units = cylinders of 2016 * 512 = 1032192 bytes
```

```
Disk identifier : 0x4491216e
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	500	503968+	83	Linux
/dev/sdb2		501	1936	1447488	5	Extended
/dev/sdb5		501	1000	503968+	b	W95 FAT32
/dev/sdb6		1001	1500	503968+	7	HPFS/NTFS
/dev/sdb7		1501	1936	439456+	83	Linux

```
Command (m for help) :
```

Enfin quittons le logiciel `fdisk`, en validant `w`

La création de partitions n'entraîne pas le formatage de celles-ci. Formater une nouvelle partition est un étape essentielle, puisque c'est le formatage qui permet d'installer le système de fichiers sur la partition. Nous allons donc formater les différentes partitions. Nous ne pouvons pas formater la partition NTFS, car le système NTFS, est un système de fichier propriétaire, et dont les spécifications n'ont pas été rendues publiques. Le formatage de cette partition sera à faire via *Windows*.

Nous allons dans un premier temps formater les partitions Linux, ainsi que la partition en FAT 32. Entrez les commandes suivantes. Attention ces commandes prennent du temps :

```
mkfs.ext3 /dev/sdb1
mkfs.ext3 /dev/sdb7
mkfs.vfat /dev/sdb5
```

Vous pouvez maintenant accéder à ces trois partitions. Vérifiez le en montant ces partitions dans trois répertoires différents, par exemple sous `/media/usb1`, `/media/usb2`, pour les deux partitions Linux, et `/media/usb3` pour la partition en FAT 32.

Les partitions en FAT 32 sont des partitions pouvant être lues/écrites depuis un windows, et un Linux. Les partitions Linux peuvent être lues sur windows, avec un programme particulier. Les partitions NTFS peuvent être lues depuis Linux. Je vous déconseille fortement l'écriture sur ce type de partition depuis un Linux.

Démontez toutes les partitions de votre clef USB, enlevez la clef, puis ré-introduisez la, attendez un peu, et tapez la commande `dmesg`. Vous pouvez voir apparaître l'ensemble des partitions que vous avez créé.

3.1 Cryptage de partition

Dans cette section, nous allons apprendre à crypter une partition linux. Ceci permet de crypter des données essentielles, et donc de rajouter une sur-couche pour la protection des données. Pour ce faire, nous allons d'abord installer les programmes permettant la gestion des partitions cryptées. Tapez la commande :

```
apt-get install cryptsetup cryptmount
```

Une fois ceci fait, tapez la commande :

```
cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb7
```

Ceci permet de faire un lien entre la partition cryptée et le *mapper*. Pour simplifier, la partition cryptée se trouve dans `/dev/sdb7`, et la partition non cryptée dans `/dev/mapper/sdb7`. Tapez la commande :

```
cryptsetup luksOpen /dev/sdb7 sdb7
```

Ceci permet “d’activer” la partition cryptée. Une fois ceci fait, nous allons formater la partition. Pour rappel les données non cryptées se trouvent dans `/dev/mapper/sdb7`. Il faut donc faire :

```
mkfs.ext3 /dev/mapper/sdb7
```

Enfin, on crée le point de montage, puis on monte le système de fichiers :

```
mkdir -p /media/crypt1/  
mount /dev/mapper/sdb7 /media/crypt1/
```

Vous avez accès à une partition cryptée. Pour démonter la partition, il faut dans l’ordre démonter la partition et fermer le luks :

```
umount /media/crypt1/  
cryptsetup luksClose sdb7
```

Vous pouvez vérifier que vous ne pouvez pas monter la partition, si vous n’ouvrez pas le luks.

3.2 Montage automatique

Il y a la possibilité de monter automatiquement au démarrage une partition cryptée. Pour ce faire, il faut modifier le fichier `/etc/crypttab`, et le fichier `/etc/fstab`. Dans le fichier `/etc/crypttab`, il faut ajouter la ligne suivante :

```
sdb7 /dev/sdb7 none luks
```

et dans le fichier `/etc/fstab`

```
/dev/mapper/sdb7 /media/crypt1 auto noauto ,users ,exec 0  
0
```

Si vous redémarrez votre ordinateur avec votre clef USB branchée, votre système demande le mot de passe pour permettre l’utilisation de la partition cryptée.

Si vous souhaitez que cette clef soit montée automatiquement, il suffit simplement de changer le `noauto` par `auto`. Dans ces conditions, la clef sera montée, mais le propriétaire sera `root`. Vous pouvez indiquer à qui appartient la clef par défaut. Il vous suffit de récupérer votre *uid*. Celui-ci se trouve dans le fichier `/etc/passwd`. Normalement, le votre devrait être 1000. Il ne vous reste plus qu'à modifier la ligne du fichier `/etc/fstab` :

```
/dev/mapper/sdb7          /media/crypt1    auto    auto ,uid=1000,exec 0  
0
```

4 Montage d'une image ISO

Linux permet aussi de monter une image iso, sans avoir besoin de la graver sur un CD-ROM. Dans un premier temps, récupérez l'image ISO, associé au TP. Ensuite, tapez les lignes de commande suivantes :

```
mkdir /tmp/iso  
mount -o loop -t auto img.iso /tmp/iso
```

Vous avez accès dans le répertoire `/tmp/iso` à l'ensemble des fichiers contenu dans l'image.