

Clef USB sous LINUX et FreeBSD

Jean-Marc LICHTLE

22 mai 2005

Table des matières

1	Introduction	2
2	Un peu de technique	2
2.1	La clef bizarre	2
2.2	Les outils que nous allons employer	3
3	Emploi sous KNOPPIX ou DAMN SMALL LINUX	3
4	Emploi avec FEDORA Core 1.B et DEBIAN Woody 3.0r2	5
5	Emploi avec MANDRAKE 9.2	6
5.1	Procédure de base	6
5.2	Améliorations	7
5.3	Reconnaissance automatique du matériel par Mandrake	8
5.4	MANDRAKE Move	9
6	FreeBSD	9
6.1	Utilisation directe par root	9
6.2	Montage de la clef par l'utilisateur	10
7	Partitionnement de la clef Iomega	11
8	Emploi d'une clef USB avec SUSE 9.0	12
9	Cas de la clef Lexar	13
10	Conclusion	14
11	L'auteur	15

1 Introduction

A priori l'emploi d'une clef USB sous LINUX ou FreeBSD est très simple. "Y'a qu'à" charger le module qui va bien, introduire la clef dans le port, après quoi "y'a plus qu'à" la "monter" dans l'arborescence. Vite dit. En pratique l'utilisateur débutant va se retrouver aux prises avec :

- Une clef USB pas tout à fait standard et qui dispose d'un partitionnement curieux.
- Une version de LINUX pleine de bonne volonté qui effectue la reconnaissance automatique de nouveaux périphériques, qui prend en charge à l'insu de votre plein gré (pas facile à placer celle-là !) une partie de la configuration ce qui constitue, à mon avis plus une gêne qu'une aide réelle, surtout pour l'utilisateur débutant.
- Au final, quand la clef est montée, des droits en écriture insuffisants pour l'utilisateur de base.

J'ai fait tous les essais décrits ici avec les systèmes dont je dispose, tout d'abord les distributions installées :

- Mandrake 9.2
- Fedora Core 1.B
- Debian Woody 3.0r2
- FreeBSD 5.2.1, qui comme son nom l'indique est un UNIX de Berkley plutôt qu'un LINUX.

ensuite les distributions "live" utilisables à la volée sans installation préalable :

- Mandrake Move
- Knoppix 3.3
- Damn Small Linux

et les clefs suivantes :

- PQI Intelligent Stick 2.0 128 Mo nommée dans la suite du texte I-Stick2
- IOMEGA 258 Mo désignée plus loin sous Iomega.
- LEXAR Jumpdrive2 dont le fonctionnement sera analysé en fin de document, en application de l'exposé.

2 Un peu de technique

2.1 La clef bizarre

Et oui, je suis évidemment, avec la chance qui me caractérise, tombé sur une clef d'un modèle rare (j'espère) qui n'est pas partitionné comme les autres. Ceux qui ont déjà utilisé sous LINUX des lecteurs de disquette ZIP savent que les disquettes de la marque IOMEGA sont partitionnées de façon assez curieuse puisque c'est la partition 4 qui est utilisée par défaut. Je pensais que cette particularité était spécifique aux disquettes, et bien non, IOMEGA réédite avec ses clefs USB, au moins celle que j'ai entre les mains ! Et bien entendu pas une ligne dans la notice (enfin je suppose, parce que j'avoue l'avoir lue en diagonale, elle ne parle que de systèmes d'exploitation que je n'utilise pas)

2.2 Les outils que nous allons employer

Les outils à notre disposition sont finalement peu nombreux ce qui va faciliter l'apprentissage :

- La lecture du journal du démarrage via la commande `dmesg` associée éventuellement à `less` ou `more`. Cette commande affiche en effet les messages qui ont défilé (rapidement) au boot et permet donc de les étudier à tête reposée. Mais `dmesg` ne sert pas seulement à lire ce qui s'est passé au démarrage, tous les événements importants qui se déroulent après le boot donnent lieu à des enregistrements qui viennent allonger la liste des messages, même si ceux-ci, une fois le boot terminé, ne s'affichent plus sur la console. On utilisera éventuellement `dmesg` combiné avec `less` ou `tail` pour sélectionner le passage intéressant.
- La lecture de la liste des modules chargés avec la commande `lsmod`, le chargement d'un module avec `modprobe` ou `insmod` ou sa suppression avec `rmmod`. Toutes ces commandes s'emploient, bien évidemment sous compte `root`. Il s'agit en effet d'intervenir au niveau du noyau du système, en ajoutant ou en supprimant ce que certains nommeraient un "driver". Notez que la différence entre `modprobe` et `insmod` tient essentiellement au fait que `insmod` tente strictement de charger le module que vous lui demandez, au risque de ne pas aboutir si des dépendances sont insatisfaites, alors que `modprobe` se charge de ses détails d'intendance.
- La consultation de la liste des périphériques montés via la commande `mount` employée sans arguments ou alors en affichant le contenu de `/etc/mstab` avec `cat /etc/mstab`. On pourra également consulter le fichier `/etc/fstab` (toujours avec `cat`) qui liste tous les périphériques qui sont susceptibles d'être montés et pour lesquels l'administrateur a prévu des réglages (point de montage, droits en écriture etc.).
- Avec certains systèmes, Mandrake en particulier, la lecture du contenu du sous répertoire `/dev` peut être pleine d'enseignements. Compte tenu du volume d'informations disponible un `ls | grep sd` peut, par exemple, limiter l'affichage aux entrées commençant par les lettres `sd`, c'est à dire `sda`, `sda1`, `sda2`, `sdb`, `sdb1` etc.. A la réflexion un `ls sd*` ferait d'ailleurs aussi bien l'affaire. Note pour les étourdi(e)s, lorsque j'écris `ls | grep le "|"` n'est pas un `"|"` comme dans `ludwigshaven` mais le symbole pipe obtenu par `[Alt Gr] - [6]`. Attention, si vous voyez s'afficher toute une liste, des dizaines et des dizaines de `sd*` cherchez alors les plus récents, ce sont ceux-là qui ont été modifiés lors de l'implantation de la clef.

3 Emploi sous KNOPPIX ou DAMN SMALL LINUX

Note : Tous les extraits de copie d'écran qui illustrent le chapitre sont issus d'essais réalisés avec DSL.

Les deux systèmes d'exploitation sont livrés sur live CD. Il suffit donc de booter sur le CD pour lancer KNOPPIX ou DSL directement sans installation ce qui est parfait pour faire des essais. J'ai donc commencé mes essais en suivant la procédure simple qui consiste à booter sans clef puis à mettre celle-ci en place une fois le système opérationnel. C'est en pratique le cas qui va se produire le plus souvent.

En fin de boot l'emploi de `lsmod` permet de lister les modules chargés dans le noyau. Là aussi

grep va limiter la longueur de la liste, par exemple (Damn Small Linux) :

```
# lsmod | grep usb
```

va afficher la liste des modules dont le nom contient la chaîne "usb" :

```
usb-uhci          21836    0  (unused)
usbcore           57088    1  [hid usb-uhci]
```

Il apparaît immédiatement que le module usb-storage, précisément celui qui doit prendre en charge la clef USB, n'est pas chargé. Notez d'ailleurs que usb-storage est également utile pour connecter d'autres périphériques, notamment les appareils photo.

Le moment est venu de mettre en place la clef et de charger le module au moyen de la commande :

```
# modprobe usb-storage
```

Avec Knoppix le module usb-storage se charge automatiquement dès l'introduction de la clef. Une nouvelle utilisation de lsmod donne maintenant une liste un peu plus complète et qui comporte notamment la ligne relative à usb-storage :

```
usb-storage       54460    0  (unused)
usb-uhci          21836    0  (unused)
usbcore           57088    1  [usb-storage hid usb-uhci]
```

Un petit coup de dmesg va dérouler la liste complète des messages de boot avec, à la fin, les messages résultant de l'installation de la clef :

```
hub.c: new USB device 00:11.2-2.2, assigned address 5
usb.c: USB device 5 (vend/prod 0x3538/0x31) is not claimed by any active driver.
```

et ceux qui sont provoqués par le chargement du module :

```
Initializing USB Mass Storage driver...
usb.c: registered new driver usb-storage
scsil : SCSI emulation for USB Mass Storage devices
  Vendor: I-Stick2  Model: IntelligentStick  Rev:
  Type:   Direct-Access                ANSI SCSI revision: 02
Attached scsi removable disk sda at scsil, channel 0, id 0, lun 0
SCSI device sda: 251904 512-byte hdwr sectors (129 MB)
sda: Write Protect is off
  sda: sda1
WARNING: USB Mass Storage data integrity not assured
USB Mass Storage device found at 5
USB Mass Storage support registered.
```

Vous noterez qu'après l'identification de la clef usb-storage donne le nom du "device" attaché à la clef et qui va permettre de le monter, ici sda1.

Il ne reste plus maintenant qu'à créer, si elle n'existe pas encore, l'entrée de sous répertoire qui va accueillir le contenu de la clef, par exemple usbdrive :

```
# mkdir /mnt/usbdrive
```

puis à "monter" la clef dans ce sous répertoire

```
# mount /dev/sda1 /mnt/usbdrive
```

Vous noterez que, dans un cas classique d'une clef ne comportant qu'une seule partition comme la I-Stick2, vous devez monter sda1. Dans le cas plus complexe de la clef Iomega dont la partition active est la sda4 vous devez remplacer sda1 par sda4 dans ce qui précède.

Nota : Les ennuis commencent ici avec la combinaison Knoppix / clef Iomega. Comme signalé plus haut le module usb-storage se charge automatiquement dès l'introduction de la clef. En fait Knoppix lance deux utilitaires, rebuildfstab et scanpartition qui semblent déroutés par l'architecture curieuse de la clef, si bien qu'une tentative de "montage" conduit à un plantage de la console. Pour ma part j'ai trouvé une parade peu glorieuse mais qui fonctionne, je commence par décharger le module usb-storage :

```
# rmmod usb-storage
```

puis je le recharge simplement :

```
# modprobe usb-storage
```

après quoi j'arrive à monter ma clef.

4 Emploi avec FEDORA Core 1.B et DEBIAN Woody 3.0r2

La distribution Fedora adopte finalement un comportement très sain, intermédiaire entre Knoppix et Mandrake objet du chapitre suivant, en ce sens que le module usb-storage se charge à l'implantation de la première clef mais que les montages restent manuels. Avantage par rapport à la Knoppix, le module usb-storage prend en compte correctement la clef Iomega. Par ailleurs, si votre écran affiche la console principale (n°1) alors vous voyez s'afficher "en live" les messages qui sont parallèlement mémorisés en vue d'une restitution à la demande par dmesg. Ce comportement est en fait très semblable à celui de FreeBSD comme nous le verrons un peu plus loin.

La Debian Woody calque son comportement sur celui de DSL, ni chargement automatique du module usb-storage, ni "montage" automatique, tout doit se faire à la main, l'utilisateur sera donc immédiatement prévenu en cas de dysfonctionnement. Cette analogie de comportement ne doit pas surprendre, DSL et Knoppix sont tous deux apparentés à Debian. Le fait que Knoppix fasse une partie du travail en chargeant le module n'est pas étonnant lorsqu'on imagine et apprécie le travail réalisé par les auteurs de cette distribution pour en faire une distribution reconnaissant automatiquement le maximum de matériels.

5 Emploi avec MANDRAKE 9.2

5.1 Procédure de base

La procédure est assez semblable à ce qui a été décrit plus haut, démarrage sans clef, vérification des modules chargés :

```
# lsmod | grep usb
```

pour afficher la liste des modules dont le nom contient la chaîne "usb" :

```
usb-uhci          25136    0 (unused)
usbcore           74988    1 [scanner printer usb-uhci]
```

Après installation de la clef une nouvelle utilisation de lsmod donne maintenant une liste un peu plus complète et qui comporte notamment la ligne

```
usb-storage      91616    0
```

Un petit coup de dmesg va dérouler la liste complète des messages de boot, avec à la fin les messages résultant de l'installation de la clef :

```
hub.c: new USB device 00:11.2-2.2, assigned address 5
usb.c: USB device 5 (vend/prod 0xa16/0x2299) is not claimed by any active driver.
```

et ceux qui sont provoqués par le chargement du module (ici exemple avec la clef Iomega) :

```
Initializing USB Mass Storage driver...
usb.c: registered new driver usb-storage
scsi1 : SCSI emulation for USB Mass Storage devices
  Vendor: IOMEGA    Model: Mini256MB*IOM2D5  Rev: 2.00
  Type:   Direct-Access                ANSI SCSI revision: 02
WARNING: USB Mass Storage data integrity not assured
USB Mass Storage device found at 5
USB Mass Storage support registered.
.....
sda: Write Protect is off
 /dev/scsi/host1/bus0/target0/lun0: p1 p2 p4
```

Vous noterez qu'après l'identification de la clef usb-storage donne une liste des partitions qui composent son système de fichiers, ici p1, p2 et p4.

Le moment est venu de vérifier que chacune de ces partitions correspond à un "device". Un petit

```
# ls /dev/sd*
```

liste immédiatement les devices sda, sda1, sda2 et sda4.

A la différence de ce qui se passait avec DSL ou Knoppix, cette fois Mandrake a été plus loin et a non seulement chargé automatiquement le module usb-storage (ce que faisait d'ailleurs aussi Knoppix) mais de plus, si vous utilisez une clef "normale" comme la I-Stick2, Mandrake a :

5.2 Améliorations

- Créé un sous répertoire /mnt/removable.
- Modifié le fichier /etc/fstab qui définit le montage des disques pour y ajouter une ligne relative à la clef.
- Monté la clef sur /mnt/removable.

Nota : Ici, encore une fois, la clef Iomega se distingue par un comportement particulier. En effet l'installation de la clef ne provoque ni modification du /etc/fstab, ni montage automatique. Nous sommes donc dans la situation de Knoppix, à ceci près qu'il est possible de monter la clef, en utilisant le bon device, sans décharger / recharger le module !

Avec la clef I-Stick2 l'intervention automatique au niveau du fichier /etc/fstab a consisté à y rajouter une ligne semblable à celle-ci :

```
none /mnt/removable supermount dev=/dev/sda1,fs=ext2:vfat,--,codepage=850,iocharset=iso8859-15,kudzu 0 0
```

Cette ligne va "monter" automatiquement (supermount) le périphérique /dev/sda1 sur /mnt/removable.

5.2 Améliorations

La ligne ajoutée à /etc/fstab est enregistrée. Au prochain démarrage, que la clef soit montée ou non, la ligne sera présente comme le confirmera la commande mount. Personnellement je pense que c'est un inconvénient, d'autant plus que la démarche décrite ci-dessus présente un défaut assez ennuyeux, seul l'administrateur peut écrire des données sur la clef USB. Ceci est facile à vérifier en faisant un `ls -alrt` dans le répertoire /mnt. Inutile de changer les droits d'accès de /mnt/removable avant le montage en espérant les conserver une fois la clef installée, cette voix est sans issue, du moins sous Linux. Nous verrons en effet plus loin que cette méthode fonctionne avec FreeBSD.

L'étude des pages de manuel, ici celle de mount, suggère que si vous voulez donner l'accès aux utilisateurs de base il convient de monter en précisant l'option `umask=0`. En montage manuel cela reviendrait donc à taper

```
# mount -o umask=0 /dev/sda4 /mnt/removable
```

L'option `umask=0` fixe les droits du sous répertoire une fois monté à 777, c'est à dire lecture, écriture et exécution pour tous. Cette syntaxe fonctionne bien entendu avec Mandrake, mais aussi Knoppix et DSL.

Pour résumer, Mandrake modifie /etc/fstab d'une façon qui ne nous arrange pas et de plus cette modification est enregistrée. On pourrait imaginer effacer la ligne à la main pour s'en débarrasser, ou alors la laisser et démonter / remonter la clef, cette fois avec les bons droits. Une autre solution consiste à modifier cette ligne pour l'adapter à nos besoins. J'ai utilisé un temps la syntaxe suivante :

```
none /mnt/istick supermount dev=/dev/sda1,fs=ext2:vfat,--,iocharset=iso8859-15,umask=0,codepage=850 0 0
```

largement inspirée de la ligne fstab relative au lecteur de disquette. Par rapport à la ligne d'origine créée automatiquement à l'implantation de la clef cette nouvelle version présente les avantages suivants :

5.3 Reconnaissance automatique du matériel par Mandrake

- Je personnalise le nom du sous répertoire en le désignant par le nom de la clef ce qui évite les confusions (j'ai une deuxième ligne semblable avec iomega et sda4).
- Les options font apparaître `umask=0`.

En résumé mon fichier `/etc/fstab` se termine donc par ces lignes :

```
none /mnt/istick supermount dev=/dev/sda1,fs=ext2:vfat,--,iocharset=
iso8859-15,umask=0,codepage=850 0 0
none /mnt/iomega supermount dev=/dev/sda4,fs=ext2:vfat,--,iocharset=
iso8859-15,umask=0,codepage=850 0 0
```

Il va sans dire que ce fichier ne présente réellement d'intérêt que pour la première clef installée. La deuxième se verra en effet attribuer un device `sdb` qui ne figure pas dans la liste. En fait, si la seconde clef est la Iomega il faudra la monter à la main, si c'est la I-Stick2, elle se montra directement sur `/mnt/removable`, avec création d'une ligne ad'hoc dans `/etc/fstab` et curieusement avec des droits `777` alors que la nouvelle ligne créée ne comporte pas l'option `umask=0`.

5.3 Reconnaissance automatique du matériel par Mandrake

La distribution Mandrake est connue et appréciée pour sa facilité d'installation et de configuration. Après des années d'essais divers, j'en reste à cette analyse simple et basique, Mandrake demeure, qu'on le veuille ou non, le ticket d'entrée incontournable pour faire tourner Linux sur un PC familial qui sera utilisé par plusieurs personnes. Dans ce genre d'application la plupart des utilisateurs ne souhaitent surtout pas se casser la tête et veulent simplement mettre en oeuvre des habitudes de travail rodées sur des systèmes d'exploitation commerciaux comme ceux qui sortent des chaînes à Redmond (USA). La philosophie de Mandrake, du moins dans sa version grand public, découle de ce constat. C'est ainsi qu'au boot la Mdk 9.2 charge un soft nommé `harddrake` et qui va réagir immédiatement au cas où il découvrirait un nouveau matériel. La couleur de fond de l'écran chargé par `harddrake`, un bleu bien connu, est très peu engageante. Quand à demander à la petite dernière de réagir correctement aux sollicitations qui lui sont faites voilà qui est certainement trop demander et probablement très dangereux. Vous aurez bien entendu droit ici à un cri du genre "papa c'est quoi cet écran bleu" si, par suite d'une différence de configuration, `harddrake` lance `diskdrake` au boot. Et cette différence peut simplement provenir du fait que, lors de la session précédente, vous avez utilisé la clef, périphérique que `harddrake` ne trouve plus au boot suivant.

Pour ma part j'aime les solutions expéditives. Dans un premier temps j'ai simplement effacé la ligne correspondante dans `/etc/rc5.d` (`S05harddrake@`), après quoi je me suis ravisé en me disant que ce genre de procédé n'était pas digne d'un administrateur système.

La bonne solution consiste donc à demander officiellement à Mdk de faire l'impasse sur la détection de matériel par la commande :

```
# chkconfig --level 123456 harddrake off
```

ce qui a pour effet de ne plus lancer automatiquement `harddrake` dans les niveaux d'exécution 1 à 6. A la réflexion il me semble que la suppression de `harddrake` ne relève pas simplement d'une

coquetterie d'administrateur parano mais d'un point élémentaire de sécurité. Si harddrake reconnaît un écart de configuration au niveau d'une clef USB il lance, tenez vous bien, diskdrake, un utilitaire qui permet entre autre, de partitionner, formater etc tous les disques présents sur le système et ce, sans demander le mot de passe administrateur ! En clair, le premier touriste qui démarre votre belle machine en y plantant une clef USB, a la possibilité de détruire complètement le système à la suite d'une fausse manoeuvre.

Par la suite, si dans votre activité d'administrateur, vous souhaitez malgré tout bénéficier des prestations de harddrake pour qu'il vous aide à détecter un matériel neuf, rien de plus simple, invoquez le "à la main" avec :

```
# /etc/rc.d/init.d/harddrake
```

5.4 MANDRAKE Move

J'ai testé ici l'édition diffusée par LINUX Planète, non la version vendue en coffret par MANDRAKE. Cette dernière diffère, paraît-il, par la prise en charge de la clef USB laquelle est "montée" comme répertoire utilisateur dans la version coffret.

Premier commentaire, Mandrake Move est sans contestation possible le champion toutes catégories de la reconnaissance de matériels puisqu'elle reconnaît et monte même un lecteur ZIP sur port parallèle qui s'était montré tout à fait discret pendant tous ces essais !

Il en découle que, suivant que la clef est installée avant ou après le boot, elle aura, ici c'est la I-Stick2, pour device sda1 ou sdb1, le lecteur ZIP prenant les adresses sdb4 ou sda4 selon le cas. Dans les deux cas le montage est d'ailleurs automatique mais avec une nuance :

- sur /mnt/windows si on boot avec la clef installée,
- et /mnt/removable si on installe la clef à posteriori.

Dans les deux cas le répertoire est accessible en lecture / écriture pour l'utilisateur de base.

Ayant réalisé ce test après avoir repartitionné la clef Iomega (voir plus loin) je ne suis pas en mesure d'analyser le fonctionnement de Mandrake Move avec cette clef particulière.

6 FreeBSD

6.1 Utilisation directe par root

En comparaison de ce qui précède la prise en charge des clefs USB par FreeBSD est presque simple. Le noyau, du moins dans la version qui équipe la version 5.2.1 de FreeBSD, est capable de reconnaître sans problème, aussi bien la clef I-Stick2 que la clef Iomega. Au montage de la clef la console 1 voit s'afficher les messages qui annoncent que le noyau vient de prendre en compte la clé ;

```
umass0: IOMEGA Mini256MB*IOM2D5, rev 2.00/0.71, addr 5
GEOM: create disk da0 dp=0xc2db6850
da0 at umass-sim0 bus 0 target 0 lun 0
da0: <IOMEGA Mini256MB*IOM2D5 2.00> Removable Direct Access SCSI-0 device
```

6.2 Montage de la clef par l'utilisateur

```
da0: 1.000MB/s transfers
da0: 246MB (504832 512 byte sectors: 64H 32S/T 246C)
```

pour la clef Iomega et :

```
umass1: I-Stick2 Flash Disk, rev 1.10/1.00, addr 6
GEOM: create disk da1 dp=0xc2dbd850
da1 at umass-sim1 bus 1 target 0 lun 0
da1: <I-Stick2 IntelligentStick > Removable Direct Access SCSI-2 device
da1: 1.000MB/s transfers
da1: 123MB (251904 512 byte sectors: 64H 32S/T 123C)
```

pour la clef I-Stick2. Cet affichage sur la console principale (n°1) est d'ailleurs aussi une caractéristique de Fedora. La commande `dmesg` permet, comme sous Linux, de relire ce texte après qu'il ait disparu de l'écran, ou d'en prendre connaissance depuis une autre console.

Je n'ai pas trouvé l'astuce qui permettait de voir, comme sous Linux, le numéro de la partition. Seul la consultation du contenu du répertoire `/dev/` permet de vérifier que dans le second cas, pour la clef I-Stick2, le device correspond à `da1s1` ("s" signifiant certainement ici "slice"=tranche), alors que dans le premier cas l'étude de `/dev/` ne donne aucun renseignement, `/dev` ne contenant que le device `/da0`. Le montage de la clef dans l'arborescence se fait très simplement, cette fois avec une commande `mount` différente de la commande courante sous linux, par exemple pour la clef I-Stick2

```
# mount_msdosfs /dev/da1s1 /mnt/usbdrive1
```

Pour la clef Iomega la syntaxe suivante est satisfaisante :

```
# mount_msdosfs /dev/da0 /mnt/usbdrive1
```

Vous aurez noté que les deux lignes de commande ne précisent pas d'option pour définir les droits d'accès. En fait, chose qui n'était pas possible sous Linux, le montage conserve les droits que possédait le répertoire avant le montage ! C'est simple et efficace.

6.2 Montage de la clef par l'utilisateur

Ce paragraphe a été ajouté en mai 2005. Il a été testé sur FreeBSD 5.4.

Dans le paragraphe précédent l'administrateur devait monter la clef pour l'utilisateur, situation sans problème si root et utilisateur sont la même personne mais génératrice d'inconfort si l'utilisateur n'est pas l'administrateur du système. L'idéal serait donc que l'utilisateur soit autorisé à monter lui-même sa clef.

Dans un système FreeBSD, du moins à partir de la branche 5, le répertoire `/dev` n'est pas enregistré sur le disque mais généré dynamiquement. C'est ainsi que l'entrée `/dev/da0s1` n'apparaîtra que quelques secondes après que la clef aura été mise en place. La gestion des "devices" est confiée au démon `devd` dont le fonctionnement est réglé par deux fichiers :

- `/etc/devfs.conf`
- `/etc/devd.conf`

Il suffit d'ajouter les lignes :

```
perm da0 0666
perm da0s1 0666
```

au fichier `devfs.conf` et les lignes :

```
attach 100 {
device-name "umass[0-9]*";
action "/bin/sleep 4; /etc/rc.d/devfs restart"
};
```

au fichier `devd.conf`.

La première modification définit des droits 0666 pour les "devices" qui sont susceptibles de recevoir la clef usb (adaptez éventuellement à votre configuration), la suivante va faire que, dès l'apparition d'un périphérique nommé `umass0` ou `umass1` etc (jusqu'à `umass9`), le démon `devd` va :

- faire une temporisation de 4 secondes laissant le temps au processus de reconnaissance de périphérique de terminer son travail,
- relancer `devfs` de telle sorte à tenir compte de la nouvelle configuration.

A partir de maintenant un utilisateur, même sans droits d'administration, pourra monter la clef USB sur un sous-répertoire dont il est propriétaire, par exemple un sous-répertoire de son répertoire utilisateur.

7 Partitionnement de la clef Iomega

Lassé de me bagarrer avec le partitionnement curieux de ce produit j'ai fini par modifier ce dernier pour rendre son format plus conventionnel. Cette petite manipulation m'a donné l'occasion d'utiliser pour la première fois `diskdrake`, ce qui m'a fait mettre le doigt sur le danger potentiel de laisser cet outil à la portée d'un utilisateur de base. Je recommande en fait de procéder à ce repartitionnement, même si, quand on est pas habitué à utiliser `diskdrake`, c'est un exercice qui laisse un peu les mains moites.

Cette façon de faire règle pas mal de problèmes. Dans la démarche précédente la première clef installée était reconnue et montée correctement. La suivante restait, malgré les précautions prises, le motif d'une bonne prise de tête. Dans la foulée du reformatage de la clef Iomega j'ai également modifié `/etc/fstab` de telle sorte à monter la première clé installée (`sda1`) sur `/mnt/usbdrive1`, la suivante (`sdb1`) sur `/mnt/usbdrive2`. Les montages sont donc simplement une question d'ordre d'installation. Mais, me direz vous, comment faire si les deux clefs sont implantées avant le boot pour savoir dans quel ordre elles ont été reconnues ? Simple, faites donc un petit `dmesg`. Autre possibilité : la commande :

```
# lsusb
```

que nous n'avons pas utilisé jusqu'à présent lancée sous compte administrateur donne la liste des périphériques USB reconnus. Voilà par exemple le résultat de cette commande sur ma machine :

```
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 002: ID 04b8:0103 Seiko Epson Corp. Perfection 610
Bus 001 Device 003: ID 05e3:0604 Genesys Logic, Inc.
Bus 001 Device 004: ID 03f0:0704 Hewlett-Packard
Bus 001 Device 005: ID 0a16:2299 Trek Technology (S) PTE, Ltd
Bus 001 Device 006: ID 3538:0031 Power Quotient International Co., Ltd
```

Les deux clefs sont les deux dernières lignes, l'ordre des montages correspond à l'ordre des lignes, Trek Technology est la clef Iomega, Power Quotient International la I-Stick2.

8 Emploi d'une clef USB avec SUSE 9.0

Je rajoute de paragraphes à mon exposé au fur et à mesure que je suis mis en présence d'une nouvelle distribution de LINUX. C'est le cas ici avec une SUSE 9.0 toute fraîche déballée de son film de protection qui associait le DVD avec la revue LINUX CD.

N'ayant plus la clef IStick (prêtée) je n'ai fait l'essai qu'avec la Iomega reformatée. La mise en place de la clef charge automatiquement le module usb-storage et ajoute une ligne au fichier fstab lequel prend dès lors la forme suivante :

```
/dev/hda1      /          reiserfs  defaults          1 1
/dev/hda5      /home      reiserfs  defaults          1 2
/dev/hda3      swap       swap      pri=42            0 0
devpts        /dev/pts   devpts    mode=0620,gid=5  0 0
proc          /proc      proc      defaults          0 0
usbdevfs      /proc/bus/usb usbdevfs  noauto           0 0
/dev/cdrecorder /media/cdrecorder auto      ro,noauto,user,exec 0 0
/dev/cdrom     /media/cdrom auto      ro,noauto,user,exec 0 0
/dev/dvd      /media/dvd auto      ro,noauto,user,exec 0 0
/dev/fd0      /media/floppy auto      noauto,user,sync  0 0
/dev/sda1 /media/sda1 auto sync,noauto,user,exec 0 0 #HOTPLUG B3Fu.lpiMmJPoCT2
```

On y retrouve en dernière ligne les traces de la modification à laquelle je viens de faire allusion. Le nouveau périphérique n'est pas monté automatiquement (noauto) mais l'utilisateur de base peut très bien demander son montage (user) avec la commande classique

```
$ mount /dev/sda1
```

commande qui a pour effet de donner l'accès au contenu de la clef via /media/sda1.

Vous noterez que le fait pour un utilisateur de monter la clef rend cet utilisateur "propriétaire" du périphérique, les autres utilisateurs ayant éventuellement un accès étant limités à un rôle de consultation sans droit de modification des fichiers ainsi qu'en témoigne la commande ls-lalrt lancée dans le sous répertoire /media :

drwxr-xr-x	2	root	root	48	2003-10-02	22:43	floppy
drwxr-xr-x	2	root	root	48	2003-10-02	22:43	cdrom
drwxr-xr-x	2	root	root	48	2004-05-04	18:06	dvd
drwxr-xr-x	2	root	root	48	2004-05-04	18:06	cdrecorder
drwxr-xr-x	7	root	root	176	2004-05-05	21:15	.
drwxr-xr-x	20	root	root	464	2004-05-05	21:46	..
drwxr-xr-x	8	jml	users	16384	2004-05-05	21:49	sda1

9 Cas de la clef Lexar

Pour la petite histoire j'ai étudié la question du montage des clefs en réponse à une demande de ma fille, étudiante 3ème cycle, qui était ennuyée par sa clef Lexar non reconnue par la Mandrake 9.2 qui équipe son portable. Le problème est d'autant plus chaud que cette clef devait être le moyen simple de réaliser rapidement une sauvegarde de sécurité de son travail. Par ailleurs sa configuration avait parfaitement fonctionné à l'époque de son installation lors des tests avec ma clef IStick.

Malgré mes conseils au téléphone, une collection de mails et l'envoi du brouillon de ce document, sans ce chapitre, rien n'y a fait. C'est finalement lorsque j'ai eu l'occasion de me trouver devant sa machine avec sa clef que j'ai pris la mesure du problème. Fort de toute ma science exposée plus haut ;-)) je n'arrivais malgré tout pas à monter cette !*#@§ ! de clef ! Pire encore, au bout d'une série de manipulations non documentées le contenu de la clef est apparu mais sans que je sois en mesure de reproduire le résultat.

Etant à mon domicile j'ai donc refait avec la clef Lexar la série d'essais décrite ci-dessus pour arriver au constat suivant :

- La clef Lexar monte sans le moindre problème sur DSL, Fedora, et FreeBSD !
- Par contre Mandrake, aussi bien dans la version installée 9.2 que dans la version "live" Mandrake Move ainsi que Knoppix posent un problème
- Je n'ai pas fait d'essais sur Debian, le disque ayant été reformaté entre temps.

Curieusement les distributions qui ne font pas de reconnaissance automatique du matériel semblent parfaitement à l'aise avec cette clef. Après un grand nombre d'essais sur les autres j'ai finalement mis au point l'analyse suivante : 192.168.1.1dd link#1 UHLW 2 0 r10

- Les deux distributions Mandrake n'ont aucun mal à reconnaître la clef (dmesg, lsusb etc..) mais c'est simplement le premier accès qui pose problème. En fait, sur un PC datant de fin 2002 avec un processeur à 1.6 GHz le premier accès met un peu plus d'une minute après quoi les opérations suivantes, lectures, écritures, démontages, remontages avec des droits différents etc se passent sans problème et à une vitesse normale.
- La Knoppix présente le même phénomène mais en deux temps, une fois au montage, une fois au premier accès.

Oh, bien sûr j'aurais été plus satisfait d'arriver à une solution brillante du style "c'est le thread du processus parent qui entre en conflit avec le module du mulot" mais finalement j'arrive à une procédure qui fonctionne de façon certaine :

-
- Surtout ne pas booter avec une clef qu'on ne connaît pas déjà en place dans le port USB. C'est la meilleure façon de se faire une grosse frayeur parce que le processus risque de faire une pose inhabituelle. Exactement le genre de truc qui donne des sueurs froides
 - Avec Mandrake, si vous avez une clef inconnue, ou une Lexar, l'installer en fin de boot, avant login, ouvrir immédiatement un terminal administrateur avec CTRL-ALT-F1 et faire un accès à cette clef avec, par exemple, cd ou ls. L'ouverture d'un terminal est possible dès que l'écran permettant le login est affiché, retour par CTRL-ALT-F7.
 - Bien entendu dans le cas de Knoppix il faudra au préalable en faire le montage.
 - Attendre une bonne minute pour réaliser un accès utilisateur depuis une application du genre navigateur ou éditeur. De toutes façons il faut déjà un peu de temps pour charger KDE, le temps pour Mandrake d'avancer dans le script de reconnaissance de clef, /etc/hotplug/usb.agent.

10 Conclusion

Il resterait certainement des tas de choses à dire sur le sujet. J'aurais pu, par exemple, élargir le sujet au montage d'autres périphériques comme les appareils photo. Je suppose que certaines informations développées ici pourraient constituer une aide dans certains cas. J'ai préféré toutefois limiter le propos en espérant que ces quelques pages pourront être utiles à quelqu'un. Je n'insisterais jamais assez sur la nécessité, face à un problème de clef USB, d'une démarche méthodique :

- Que s'est-il passé ? Voir l'affichage de la console principale (FEDORA, FreeBSD) ou les dernières lignes affichées par dmesg.
- Quels sont les "devices" créés ? Sous Linux un `ls -alrt /dev/sd*` peut être riche d'enseignements en fournissant une liste horodatée des entrées du répertoire /dev.
- Quels sont les périphériques USB reconnus par le système ? Ici c'est la commande `lsusb` qui peut vous venir en aide.
- Quels sont les périphériques montés ? Là c'est la commande `mount` qui est mise à contribution. Attention, la mention `supermount` ne signifie nullement que le périphérique soit effectivement déjà lu par le système. Il le sera au premier accès, si le périphérique est en place.

Avec cette petite trousse à outils vous devriez pouvoir vous sortir de toutes les situations. En cas de panique absolue et pour être certain de repartir sur des bases connues, vous pouvez décider, après avoir "démonté" tous les montages de clefs qui ne donnent pas satisfaction, de recharger le module `usb-storage` après l'avoir supprimé.

J'ai passé en revue un grand nombre de distributions Linux, ainsi qu'une version de BSD. L'objectif ici n'était pas de faire une étude académique de tous les cas possibles mais simplement de montrer que les comportements peuvent être très différents d'une distribution à l'autre. Le lecteur aura donc la charge de vérifier le comportement de sa distribution.

Mon sentiment est que le comportement le plus fiable est certainement celui de DSL ou de FreeBSD qui laisse tout le travail à la charge de l'administrateur, lequel devra successivement :

- Implanter la clef.

-
- Charger le module (pas avec FreeBSD).
 - Vérifier que la clef est bien prise en charge (dmesg ou console 1).
 - Vérifier éventuellement que le "device" attendu est bien créé.
 - Créer le sous répertoire de montage si nécessaire.
 - Monter la clef en fixant les droits des utilisateurs aux valeurs souhaitées.

11 L'auteur

Jean-Marc LICHTLE, ingénieur Arts et Métiers promotion CH73, auteur de différentes contributions que l'on peut trouver sur :

<http://www.lea-linux.org>

<http://jeanmarc.lichtle.free.fr>

J'ai pris l'habitude de mettre au propre mes notes pour que les difficultés surmontées produisent autant de documents mis à disposition de la communauté des utilisateurs des Logiciels Libres.

jml