

0.1 USB sous Linux 2.6

Depuis plusieurs années, les périphériques USB de toutes sortes (clés, imprimantes, DD...) sont correctement supportés sous Linux. Mieux, de nombreuses distributions viennent avec des modules prêts à l'emploi pour l'USB. Une fois qu'elles ont détectées quel type de chipset votre carte mère utilise, elle charge d'elle-même le bon module. Ce document décrit une méthode manuelle et générique pour bénéficier à chaque démarrage de l'USB. Notez que la grande majorité des distributions disposent d'outils capables de réaliser tout cela automatiquement. Il servira simplement de référence à tous les tutoriaux traitant de périphériques USB. Notez enfin qu'il ne traite que des noyaux 2.6.x.

0.1.1 1. Chipset de la carte mère

Il faut savoir que sous Linux, il y'a 3 drivers (ou modules) pour l'USB. Deux pour l'USB 1, chacun en fonction du chipset de votre carte mère et un générique pour l'USB 2. Avant d'aller plus loin, il est important de déterminer quel chipset vous utilisez, pour cela tapez :

```
$ lspci -v | grep HCI
```

Chez moi j'obtiens :

```
00 :11.2 USB Controller : VIA Technologies, Inc. VT6202 [USB 2.0 controller]
    (rev 18) (prog-if 00 [UHCI])
00 :11.3 USB Controller : VIA Technologies, Inc. VT6202 [USB 2.0 controller]
    (rev 18) (prog-if 00 [UHCI])
00 :11.4 USB Controller : VIA Technologies, Inc. VT6202 [USB 2.0 controller]
    (rev 18) (prog-if 00 [UHCI])
```

Ce qui signifie que j'ai un chipset VIA supportant la norme UHCI (Universal Host Controller Interface). UHCI est la norme USB 1 développée par Intel et utilisée avec des chipsets Intel ou Via. Il existe une autre norme USB 1, c'est l'OHCI (Open Host Controller Interface), développée par Compaq. Elle est utilisée par des chipset Compaq, SIS ou ALI. Enfin il existe, heureusement, une seule et unique norme USB 2, la EHCI (Enhanced Host Controller Interface). Notez qu'il se peut que votre carte mère soit compatible à la fois USB 1 et 2. Dans ce cas vous aurez un couple de xHCI. Il peut s'agir d'UHCI et EHCI ou d'OHCI et d'EHCI.

0.1.2 2. Compilation du noyau

Dans cette partie, je ne vais évidemment pas détailler la compilation du noyau, d'abord parce que, vous n'en avez probablement pas besoin. En effet, les distributions Mandriva 10 et plus, Fedora Core 2 et plus, Debian avec noyau 2.6, Slackware avec noyau 2.6, sont déjà prêtes pour l'USB avec noyau 2.6. Les utilisateurs de ces distributions peuvent passer au 3. De plus, il y'a un didacticiel sur la compilation du noyau ICI¹. Je vais néanmoins indiquer les options de compilation pour ceux que ça intéresse :

- Device Drivers —>
- USB support —>
 - <M> Support for Host-side USB
 - USB verbose debug messages

¹<http://www.trustonme.net/didactels/285.html>

- — Miscellaneous USB options
- * USB device filesystem
- * Enforce USB bandwidth allocation (EXPERIMENTAL)
- * Dynamic USB minor allocation (EXPERIMENTAL)
- — USB Host Controller Drivers
- <M> EHCI HCD (USB 2.0) support
- <M> OHCI HCD support
- <M> UHCI HCD (most Intel and VIA) support

0.1.3 3. Chargement des modules

Maintenant que vous êtes sûr que votre noyau a bien le strict nécessaire pour utiliser l'USB, il vous suffit de charger les bons modules. Pour commencer tapez :

```
/bin/dmesg | grep usb
```

Si vous obtenez quelque chose, c'est que l'USB est reconnu et que le bon driver est chargé, vous pouvez arrêter de lire ce tutorial. Si vous n'obtenez rien c'est que l'USB n'est pas chargé automatiquement, la suite vous concerne donc. Je rappelle que ce qui suit ne concerne que ceux dont la précédente commande est restée sans réponse. Tapez :

```
ls /lib/modules/$(uname -r)/kernel/drivers/usb/host/
```

Si vous obtenez :

```
ehci-hcd.ko ohci-hcd.ko uhci-hcd.ko
```

Vous pouvez passer au 4, sinon, recompilez votre noyau, comme indiqué dans le 2.

0.1.4 4. Déterminer le bon module

Si vous lisez ceci, c'est que votre noyau est compilé, avec le nécessaire pour l'USB en module, mais que le bon module n'est pas chargé automatiquement. Il vous reste donc à déterminer quel module charger puis à automatiser son chargement. Comme dit plus haut, le bon module dépend de votre carte mère. Si vous avez une carte à base de chipset Via ou Intel, tapez :

```
modprobe uhci-hcd
```

Si tout se passe bien, notez le nom du module, ici "uhci-hcd". Si vous avez un message d'erreur, notez le message d'erreur et demandez de l'aide sur le forum, en décrivant toutes les étapes que vous avez suivies, le plus précisément. Si vous avez une carte mère à base de Compaq, SiS ou ALi, tapez :

```
modprobe ohci-hcd
```

Si tout se passe bien, notez le nom du module, ici "ohci-hcd". Si vous avez un message d'erreur, notez le message d'erreur et demandez de l'aide sur le forum, en décrivant toutes les étapes que vous avez suivies, le plus précisément. Si votre carte supporte l'USB2, quel que soit son chipset, tapez :

```
modprobe ehci-hcd
```

Si tout se passe bien, notez le nom du module, ici "ehci-hcd" et passez au 5. Si vous avez un message d'erreur, notez le message d'erreur et demandez de l'aide sur le forum, en décrivant toutes les étapes que vous avez suivies, le plus précisément.

0.1.5 5. Chargement automatique des modules

A ce stade vous savez quel(s) module(s) charger et vous souhaitez automatiser ce chargement. Pour fixer les idées, je vais supposer que vous ayez une carte mère à base de Via et supportant l'USB 2. Vous avez donc probablement noté comme module : uhci-hcd et ehci-hcd. Si ce n'est pas votre cas, adaptez ! Pour le chargement automatique à proprement parler, tout dépend de votre distribution.

- Pour les utilisateurs de Debian, assurez-vous d'avoir les lignes suivantes décommentées (pas de # devant) dans votre /etc/modules :

```
uhci-hcd
ehci-hcd
```

- Pour les utilisateurs de Slackware, assurez-vous d'avoir les lignes suivantes décommentées (pas de # devant) dans votre /etc/rc.d/rc.modules :

```
/sbin/modprobe uhci-hcd
/sbin/modprobe ehci-hcd
```

- Pour les utilisateurs de Mandriva, assurez-vous d'avoir les lignes suivantes décommentées (pas de # devant) dans votre /etc/modprobe.conf :

```
install usb-interface /sbin/modprobe uhci-hcd; /sbin/modprobe
ehci-hcd; /bin/true
```

- Pour les utilisateurs de Fedora, assurez-vous d'avoir les lignes suivantes décommentées (pas de # devant) dans votre /etc/modprobe.conf :

```
alias usb-controller ehci-hcd
alias usb-controller1 uhci-hcd
```

- Pour les autres, rajoutez les lignes suivantes dans /etc/rc.d/rc.local :

```
/sbin/modprobe uhci-hcd
/sbin/modprobe ehci-hcd
```

0.1.6 6. Configuration du système de fichiers

Pour permettre à certains outils d'accéder aux périphériques USB connectés à votre ordinateur, vous devez monter le système de fichiers virtuel USB, dans le répertoire /proc/bus/usb/. Avant de faire quoi que ce soit, vérifiez d'abord que votre distribution ne s'en est pas chargée à votre place, pour cela tapez :

```
$ grep hcd /proc/bus/usb/devices
```

Chez moi j'obtiens :

```
S : Manufacturer=Linux 2.6.7 uhci_hcd
S : Manufacturer=Linux 2.6.7 uhci_hcd
S : Manufacturer=Linux 2.6.7 uhci_hcd
```

Si vous obtenez quelque chose de similaire, vous pouvez arrêter de lire ce tutorial. Dans tous les autres cas, tapez la commande suivante en root :

```
# mount -t usbfs none /proc/bus/usb
```

Si vous avez un message d'erreur, notez le message d'erreur et demandez de l'aide sur le forum, en décrivant toutes les étapes que vous avez suivies, le plus précisément. Si tout se passe bien, ajoutez la ligne suivante à la fin de votre /etc/fstab :

```
none /proc/bus/usb usbdevfs defaults 0 0
```