Le Mini-HOWTO PLIP Linux

Andrea Controzzi, *controzzi@cli.di.unipi.it*
Traduction :Thierry Hamon *thierry.hamon@lipn.univ-paris13.fr*
v2.1, 12 mars 1998

Cet HOWTO vous permettra, je l'espère d'installer et d'utiliser PLIP (Parallel Line Interface Protocol).

Contents

NOTES :

Cette nouvelle version du mini-HOWTO comporte de nombreux changements mais aussi beaucoup d'améliorations. Bien que ce soit peu probable, mais néanmoins possible, des informations peuvent être fausses ou dépassées. S'IL VOUS PLAIT, informez moi des erreurs afin de fournir à chacun des informations correctes.

Les changements les plus importants sont :

- L'utilisation du format SGML pour rédiger l'HOWTO ;
- Une réponse générale à la question la plus souvent posée : "PLIP et Windows 95" (la réponse est négative, désolé) ;
- Une plus grosse et meilleure FAQ (merci aux compte-rendus des innombrables utilisateurs : nombreux sont ceux qui trouveront leur commentaires dans cette section, comme je l'avais promis) ;
- La suppression des fichiers d'Installation Rapide PLIP, qui étaient inutiles et dépassés ;
- La mise à jour de la section sur la connexion PLIP Dos-Linux

Pour toutes questions, corrections, commentaires et/ou suggestions, voici mon adresse électronique : *controzzi@cli.di.unipi.it* . N'hésitez pas à m'écrire si vous avez besoin d'aide. Il est possible que je ne réponde pas immédiatement, mais je répondrai ! Si après deux semaines vous ne recevez pas de rien de ma part, cela veut dire que notre système de courrier électronique a un problème : n'hésitez pas à m'écrire à nouveau, c'est mon devoir de vous aider. Je ne peux pas garantir que je résoudrai vos problèmes, mais j'essaierai. Si après plusieurs mails, vous ne recevez pas de réponse, vérifier votre adresse de retour. Plusieurs fois, mes réponses ne sont pas arrivées à cause de ce genre de problème, le plus souvent l'adresse de retour était `root@myhost`.

Avant d'envoyer un mail, lisez la FAQ. Ma réponse à une question déjà présente dans la FAQ ne sera pas meilleure (mais probablement pire ou incomplete) que la réponse que vous avez déjà lue.

Pour toutes questions sur PLIP avec DOS et Windows 95, s'il vous plait, envoyez un mail aux auteurs de ces sections. Je ne peux pas vous aider.

Tout d'abord, beaucoup d'informations techniques viennent du NET-2-HOWTO de Terry Dawson (Traduction française de Frederic Lepied et Jacques Chion – NdT). Ce Mini-HOWTO n'est pas supposé couvrir tous ses aspects et/ou le remplacer : mon but est de vous donner les moyens d'installer rapidement une connexion PLIP permanente, et SEULEMENT ça ! Toutes les autres informations viennent de mon expérience personnelle et de l'aide de nombreux utilisateurs qui m'ont envoyé des commentaires et des informations.

Lisez le NET-2-HOWTO et les autres documents pour les informations générale sur les réseaux et sur les fichiers de configuration que je propose de modifier.

1 Introduction : Qu'est-ce que PLIP et pourquoi l'utiliser ?

Il y a de nombreuses manières de créer une connexion entre plusieurs machines en local. PLIP, comme SLIP, permet d'établir une connexion locale entre deux machines mais utilise pour cela les ports parallèles.

Les ports parallèles transfèrent plus d'un bit à la fois. Il est ainsi possible d'obtenir des vitesses plus grandes qu'avec une interface série.

La vitesse atteinte dépend exclusivement de votre matériel (CPU et port paralléle) et de la charge du système, en général entre 5 Ko/s et 40 Ko/s.

L'interface PLIP est assez rapide pour permettre des fonctions TCP/IP décentes, comme NFS. Vous pouvez donc avoir un ordinateur avec tout le système Linux et un autre avec seulement le système minimal où vous pouvez monter tout le reste à partir de la machine principale.

Un inconvénient : la plupart des utilisateurs ont une machine ne possédant qu'un seul port parallèle. Ainsi, vous ne pourrez imprimer et utiliser une connexion PLIP en même temps. Même avec deux ports parallèles, ce n'est pas possible sans utiliser les modules du noyau.

Si vous avez deux ports parallèles ou plus, ce problème peut être contourné en utilisant un patch.

Enfin, je vous donnerai un bon moyen de configurer une liaison PLIP entre DOS et Linux.

Je n'insisterai pas trop : jusqu'à maintenant personne ne m'a rapporté avoir réussi une connexion entre Linux et Windows95.

2 Matériel nécessaire pour utiliser PLIP

Le matériel requis pour l'installation d'une interface PLIP est (évidemment) un port parallèle libre sur les deux machines et un câble. Si vous le pouvez, dans votre BIOS, configurez le port paralléle, au moins, comme un port "bi-directionel", et si possible dans le mode ECP ou EPP.

Pour le câble, voici un extrait du fichier `plip.c` des sources du kernel 2.0.33 :

```
Le câble utilisé est de facto un câble parallèle standard null
printer -- vendu comme un câble LapLink par divers magasins. Vous
aurez besoin d'un câble 12 brins pour en faire un vous même. Le
câblage est le suivant :
  SLCTIN      17 - 17
  GROUND      25 - 25
  D0->ERROR   2 - 15          15 - 2
  D1->SLCT    3 - 13          13 - 3
  D2->PAPOUT  4 - 12          12 - 4
  D3->ACK     5 - 10          10 - 5
  D4->BUSY    6 - 11          11 - 6

Ne pas connecter les autres broches, c'est-à-dire :

  D5, D6, D7 sont 7, 8, 9
  STROBE est 1, FEED est 14, INIT est 16
  Les masses supplémentaires sont 18, 19, 20, 21, 22, 23, 24
```

Mais je vous recommande fortement de lire le fichier `/usr/src/linux/drivers/net/README1.PLIP` et `README2.PLIP` pour plus d'informations au sujet du câble.

A mon avis, vous devez éviter de construire votre propre câble parallèle. Vous économiserez un peu d'argent en le fabriquant vous-même, mais vous pourrez aussi avoir pas mal de maux de tête. Si vous souhaitez vraiment construire votre câble parallèle, rappelez vous que vous le faites à vos risques et périls. J'ai reporté exactement ce qu'il est écrit dans le fichier `plip.c` mais je ne donne aucune garantie.

Un dernier mot au sujet de la longueur du câble : les câbles longs (c'est-à-dire plus de 3 mètres, 10 pieds) peuvent créer des problèmes à cause des interférences radio. Si vous avez besoin de câbles longs, vous devrez utiliser des câbles blindés de bonne qualité. Cependant, de très longs câbles ne sont pas recommandés : je pense que la taille maximale d'un câble doit être de 15 mètres (50 pieds).

Quoi qu'il en soit, quelqu'un m'a envoyé un mail en m'indiquant que son câble de 30 mètres fonctionne très bien. Si quelqu'un veut vraiment essayer une connexion PLIP entre son bureau et sa maison (200 mètres de long, par exemple), et qu'il a de l'argent à dépenser, il peut essayer, mais à ses risques et périls.

3 Reconfiguration du noyau

Je suppose que vous savez déjà comment configurer et compiler le noyau, sinon vous devez vous référer à d'autres documents : le Kernel-HowTo ou d'autres guides (Le Guide du Rootard, par exemple – NdT). Grâce au travail des développeurs du noyau, recompiler les derniers noyaux est vraiment devenu facile même pour le *commun des mortels*, alors *just do it* ;-) ! N'importe comment, pour être complet, voici un résumé rapide de ce que vous devez faire.

NB : Je suppose que vous utilisez un noyau de la série 2.0.xx. Maintenant, il n'y a plus aucune raison pour garder les noyau 1.2.xx . Il n'y a aucune instructions en rapport avec les noyaux 2.1.xx puisqu'ils sont destinés au développement.

Je suppose que vous utilisez `menuconfig` pour initialiser les options du noyau, mais les autres outils sont équivalents. Voici comment faire :

```
#make menuconfig
```

Je vous recommande fortement de sélectionner :

```
Loadable module support --->
```

et de valider l'option :

```
[*] Enable loadable module support
```

et, si c'est possible (c'est-à-dire si vous avez les modules 2.0.0) l'option :

```
[*] Kernel daemon support (e.g. autoload of modules)
```

Ensuite retournez en arrière et choisissez :

```
Networking options  --->
```

où vous pourrez au moins valider les options :

```
[*] Network firewalls
[*] TCP/IP networking
[*] IP: forwarding/gatewaying
```

Puis retournez en arrière et valider au moins :

```
[*] Network device support
<M> PLIP (parallel port) support
```

Si vous utilisez des modules, je vous conseille fortement de configurer PLIP comme module. Vous pouvez alors, si vous avez besoin d'utiliser une imprimante, aller à :

```
Character devices  --->
```

et configurer l'option comme module :

```
<M> Parallel printer support
```

Maintenant le support du noyau pour PLIP est disponible. Si c'est la première fois que vous compiler le noyau, regardez les autres options puis sauvez et sortez du menu.

Enfin compilez avec les commandes :

```
#make dep ; make clean
#make zlilo
```

Et, si vous utilisez des modules :

```
#make modules
#make modules_install
```

Maintenant relancer votre sytème.

4 Messages du noyau au démarrage

Après avoir reconfiguré et compilé le noyau en sélectionnant le support PLIP, quand vous démarrez le système si le noyau supporte directement PLIP, ou quand vous chargez ("plus tard", voir ci-dessous) le module PLIP au cas ou vous avez compilé PLIP comme un module, vous devriez obtenir quelque chose comme ça (les nombres peuvent différer) :

```
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip1: Parallel port at 0x378, using assigned IRQ 7.
```

Suivant la configuration de `klogd` et `syslogd`, le message pourra être écrit dans vos fichiers de messages système : PAS DE PANIQUE si vous ne voyez pas ces messages. Si vous avez compilé PLIP comme un module et que la commande `lsmod` montre que le module plip est chargé, c'est suffisant.

Prenez en note le nom de l'interface. Habituellement, il s'agit de plip1, mais ça peut être plip0 ou même plip2, plip3, etc. Il dépend de l'adresse d'E/S.

5 Configuration des fichiers

NB : Certaines distributions, comme la Debian, utilise des fichiers de configuration différents. Si vous avez une installation standard et que vous ne trouvez pas les fichiers `rc.inet*`, cherchez des fichiers de configuration dans le répertoire `/etc/init.d`.

Tout d'abord, n'oubliez pas de sauvegarder les fichiers que vous allez modifier :

```
#cd /etc/rc.d
#cp rc.inet1 rc.inet1.BACKUP
```

Ça peut servir.

Maintenant, si vous ne l'avez pas déjà fait, vous devez choisir les adresses IP des deux machines. Dans mes exemples, j'utiliserais deux IPs fictives à la place de celles que vous utiliserez, dans le format standard `xxx.xxx.xxx.xxx` .

Dans le fichier `/etc/rc.d/inet1.rc` des deux machines, ajoutez la ligne (il est préférable de l'écrire dans la dernière partie du fichier) :

```
/sbin/route add -net $(NETWORK) netmask $(NETMASK)
```

Où les variables NETWORK et NETMASK ont été initialisées au préalable. Si vous ne savez pas comment faire, reportez vous au NET-2-HOWTO.

Si, après avoir lancer cette commande, vous obtenez le message comme celui-là :

```
SIOCADDRT: network unreachable
```

Alors utilisez plutôt cette commande :

```
/sbin/route add -net ${NETWORK} netmask ${NETMASK} dev plip1
```

où, comme d'habitude, vous devrez utiliser le nom de l'interface donnée dans les messages du noyau.

Vous pouvez ignorer sans danger ces variables, uniquement pour le cas suivant :

Si vous voulez seulement connecter deux machines dans un réseau autonome, vous pouvez donner n'importe quelle adresse IP, c'est-à-dire 200.0.0.1 et 200.0.0.2 respectivement. Dans ce cas, vous pouvez mettre sans danger `NETWORK="200.0.0.0"` et `NETMASK="255.255.255.0"`. Ce sont les exemples d'IPs que j'utilise dans l'Installation Rapide de PLIP (voir ci-dessous).

NB : `200.0.0.1` et `200.0.0.2` ne sont que des exemples. Je vous conseille fortement de ne pas les utiliser : il peut s'agir des adresses réelles de machines sur Internet !

Je recommande de choisir votre adresse dans les plages d'*adresses privées* :

```
10.0.0.0      -   10.255.255.255
172.16.0.0    -   172.31.255.255
192.168.0.0   -   192.168.255.255
```

Dans le fichier `/etc/hosts` des deux machines, vous devez ajouter les entrées avec les IPs des machines que vous connectez via PLIP. Dans mon exemple, les entrées sont:

```
200.0.0.1       un      # c'est l'adresse IP de "un"
200.0.0.2       deux    # c'est l'adresse IP de "deux"
```

Où `un` et `deux` sont les noms que vous avez choisis pour les deux machines.

Si vous voulez activer le NFS, en plus de répondre `yes` pendant la configuration du noyau, vous devez ajouter dans `/etc/exports` les entrées qui décrivent les répertoires que vous souhaitez exporter. Dans mon exemple, pour être capable de monter le répertoire `/usr`, vous devez ajouter l'entrée :

```
/usr                    two  (ro)
```

Pour plus d'informations sur NFS, lisez la documentation spécifique. Ne me rapportez pas des problèmes au sujet de NFS, je ne pourrai pas vous aider.

Maintenant, relancer votre système.

6 Activation de la liaison PLIP

Enfin, voici les commandes qui active l'interface PLIP. Vous devez les exécuter en étant root (et bien sûr le câble doit être correctement branché).

NB : Si quelque chose d'inattendu survient, revérifiez le cable et les commandes. Si vous suivez les instructions correctement, mais qu'il y a toujours des erreurs, lisez la section concernant la FAQ. Beaucoup de reponses y sont déjà.

Tout d'abord, il faut vérifier qu'il n'y a pas de périphérique `lp` présent :

```
#cat /proc/devices
```

Vous ne devez pas voir de référence à `lp` comme ci-dessous :

```
6 lp
```

Si vous voyez cette référence, supprimez (temporairement) le périphérique lp avant de continuer. Si PLIP fonctionne, alors vous pourrez l'essayer avec lp plus tard. Pour supprimer le périphérique lp, vous devrez utiliser la commande `rmmod` si c'est un module. Sinon, il est inclu dans le noyau, vous devez alors recompiler le noyau avec lp comme un module (une idée beaucoup plus sage).

A nouveau, je vais utiliser les noms `un` et `deux` comme exemples. Sur `un`, vous devrez effectuer les étapes suivantes.

Si vous n'utilisez pas le démon pour l'automontage des modules et que vous compilez PLIP comme un module, vous devrez le charger :

```
# insmod plip
```

NB : Si votre port parallèle a une IRQ diférente de 7 et/ou une adresse d'E/S différente de 0x378, vous devrez l'indiquer à insmod. Trouvez l'IRQ et l'adresse E/S qui conviennent (la commande DOS `MSD` sera probablement utile, mais ne lui faites pas trop confiance) et tapez quelque chose comme ça :

```
# insmod plip io=0x278 irq=5
```

Habituellement, l'IRQ est 7 ou 5, et l'adresse E/S est 0x378, 0x278 ou 0x3bc. C'est important de vérifier que l'adresse et l'IRQ correspondent à la configuration du matériel (regarder les cavaliers sur les vielles cartes, ou le BIOS sur les cartes mères plus récentes).

Si vous êtes du genre paranoïaque, vous pouvez vérifier que les modules sont chargés avec :

```
#lsmod

Module:         #pages:  Used by:
plip                3          0
```

Notez que le nom de l'interface (plip0, plip1, etc ; pour plus de détails, lisez la section sur les messages du noyau ci-dessus), puis initialisez l'interface PLIP :

```
#ifconfig plip1 un pointopoint deux up
```

NB : Si votre port parallèle a une IRQ diférente de 7 et/ou une adresse d'E/S différente de 0x378, vous devrez l'indiquer à `ifconfig`. Utilisez l'IRQ et l'adresse d'E/S indiquées dans les messages du noyau et lancez quelque chose ça :

```
# ifconfig plip1 irq 7
# ifconfig plip1 io_addr 0x3bc
```

D'habitude, l'IRQ est 7 ou 5, alors que l'adresse d'E/S est 0x378, 0x278 ou 0x3bc.

Maintenant, vérifiez que ça marche :

```
#ifconfig

.....
.....
plip1     Link encap:10Mbps Ethernet  HWaddr FC:FC:C8:00:00:01
          inet addr:200.0.0.1  P-t-P:200.0.0.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          Interrupt:7 Base address:0x378
```

Ajoutez la route vers `deux`...

```
#route add deux
```

Et, si vous voulez aussi le système NFS pour `deux` :

```
#rpc.portmap
#rpc.mountd
#rpc.nfsd
```

Sur `deux`, les commandes sont les mêmes, mais vous devez écrire `un` à la place de `deux` et *vice-versa*.

Une de vos machines doit avoir probablement une seule connexion PLIP. Si c'est le cas, vous devez aussi taper la commande :

```
#route add default gw un
```

sur cette machine. Dans mon exemple, `deux` est un portable qui ne possède qu'une connexion PLIP avec `un`. J'ai donc taper la commande ci-dessus.

Enfin, on vérifie la connexion à partir de `un` :

```
#ping deux
```

et à partir de `deux` :

```
#ping un
```

Bien sûr, toutes ces commandes peuvent être exécutées automatiquement avec un script ou au démarrage. Vous devez seulement créer ce script et l'exécuter en tant que root quand vous avez besoin d'une connexion PLIP. Vous pouvez également ajouter une commande dans le fichier `/etc/rc.d/rc.inet2` pour qu'il soit exécuté au démarrage.

Pour ajuster la connexion PLIP, vous pouvez utliser la commande `plipconfig`. Regarder les pages du manuel pour plus d'informations.

Pour fermer la connexion PLIP, vous devez lancer la commande :

```
# ifconfig plip1 down
```

qui efface les entrées dans la table de routage. Si vous n'avez pas le démon pour l'automontage, alors enlever le module :

```
# rmmod plip
```

7 FAQ

Cette section résoudra, je l'espère, vos problèmes. Si vous avez d'autres questions, n'hésitez pas à m'envoyer un mail.

- J'ai ces messages au moment du démarrage (ou lorsque je charge le module plip) :

  ```
  SIOCSIFADDR: No such device
  SIOCADDRT: Network is unreachable
  ```

  et quand j'essaie d'installer la connexion comme expliquer ci-dessus, j'obtiens les messages d'erreurs :

  ```
  SIOCSIFADDR: No such device
  SIOCSIFDSTADDR: No such device
  SIOCADDRT: Network is unreachable
  mount clntudp_create: RPC: Port Mapper failure - RPC: Unable to send
  ```

  - Le support PLIP n'est pas disponible dans le noyau, pour différentes raisons :
    - Vous n'avez pas répondu `yes` à la question `PLIP support?` pendant la configuration du noyau ;
    - Vous avez répondu `yes` à la question `Printer support` pendant la configuration du noyau ;
    - Vous avez compilé PLIP comme un module, vous devez ensuite le charger ;
    - Le numéro du port n'est pas le bon. Vous avez, par exemple, écrit plip1 à la place de plip0.

- Est-ce qu'il y a un moyen pour que le noyau supporte à la fois PLIP et LP, en dehors des modules, peut-être avec deux ports parallèles ?
  - Oui, les deux moyens d'y arriver sont décrits dans la section *Patches et comment faire cohabiter PLIP et LP* ci-dessous :
    - Vous pouvez appliquer un patch afin que le noyau les supporte tous les deux ;
    - Vous pouvez appliquer un autre patch afin que le noyau utilise un port parallèle pour PLIP et un autre pour LP.

- J'ai écrit un script qui établit la connexion entre mes deux machines. Je configure automatiquement la connexion dans mon `rc.inet2` d'où j'appelle un script qui l'établit et lance NFS. Ma machine `deux` monte des répertoires de `un` ; j'ai ajouté les entrées correctes dans le fichier `/etc/fstab` de `deux`. Si je démarre `deux` alors que `un` est éteinte, `deux` s'arrête plusieurs minutes sur le message `mounting remote file systems...`.
  - Ce message est affiché car `deux` attend pour monter les systèmes de fichiers de `un`, mais si `un` est éteinte, vous devez attendre la fin du délai d'attente de `deux`. Pour éviter ça, vous pouvez :
    - Mettre en commentaire dans `rc.inet2`, la commande qui monte les systèmes de fichiers distants.
    - Enlever l'entrée dans le fichier `/etc/fstab` de `deux` et monter les systèmes de fichiers distants manuellement quand vous en avez besoin.
    - Une meilleur solution serait pour `deux` de détecter au démarrage si `un` fonctionne, et de monter le système de fichier si c'est le cas. Ce peut être réalisé en remplaçant la commande `mount` dans `rc.d`, ou autre part, par quelque chose comme ce qui suit :

```
if ping -c 5 one ; then
  mount one:/.....
fi
```

- Ma connexion fonctionne, mais la commande `ping` échoue. Je reçois le message suivant de la part du noyau :

```
plip1: timed out (1, 89)
```

  ou bien par des messages similaires.

  – Cela veut dire que votre *côté de la connexion* fonctionne. Votre machine envoie le signal, mais l'*autre côté* ne répond pas ou *votre côté* n'attend pas sur les bonnes IRQ/adresse d'E/S. C'est le problème le plus courant. Hélas, il y a beaucoup de raisons possible, habituellement un câble défectueux ou une mauvaise IRQ et/ou d'adresse d'E/S. Le mauvais IRQ est la source 60% des problèmes. Il est très probable qu'en le changeant ou en effectuant des modifications de la configuration, cela résolve le problème. Voici une liste détaillée des raisons possibles :

    ∗ Le câble n'est bien branché, ou bien il est coupé ou de mauvaise qualité. Vérifiez le, si possible, entre deux machines sous Linux où la connexion PLIP fonctionne déjà. Si ce n'est pas possible, alors vérifiez au moins le câble avec un testeur. Le fait que le câble fonctionne/ne fonctionne pas avec DOS/Windows95 est de bon/mauvais augure mais ce n'est pas une preuve.

    ∗ PLIP n'est pas installé sur la machine de l'*autre côté* ;

    ∗ Il s'agit d'une connexion avec un portable qui ne possède pas de port parallèle propre, voir la question suivante.

    ∗ Vous avez un port parallèle vraiment bon marché et qui n'est qu'un simple port d'imprimante. Il peut envoyer des donnéess mais pas en recevoir.

    ∗ Votre port parallèle n'est pas configuré (au moins) comme bi-directionel. Faites le dans le BIOS. Les configurations avancées du port parallèle comme EPP et ECP sont bien mieux.

    ∗ Les ports parallèles ont différentes IRQ. Vous devez donc charger le module plip (ou le module lp) avec une IRQ différente. Retournez à la section 6 (Activation de la liaison PLIP) et choisissez une autre IRQ.

    ∗ Certains périphériques peuvent déjà utiliser votre IRQ (qui est d'habitude l'IRQ 7). Ce peut être la carte son. Ne fait pas confiance aux programmes DOS comme MSD, essayez plutôt de charger le module plip avec une IRQ différente

- J'ai utilisé la bonne IRQ et la bonne adresse d'E/S, mais ça ne fonctionne toujours pas. J'ai récupéré les adresses à partir de la commande MSD.

  – On m'a envoyé message indiquant que MSD donnait des adresses erronées concernant le port. Essayez plutôt le programme : *http://www.cs.caltech.edu/˜huny/para13.zip* .

- Ma connexion est installé, `ping` fonctionne bien. Je reçois quelques fois le message suivant du kernel :

```
plip1: timed out (1, 89)
```

  ou des messages similaires.

  – Ces messages veulent dire que l'autre machine n'a pas répondu avant le delai d'expiration (time-out). Si tout fonctionne, vous pouvez les ignorer : habituellement, cela veut dire soit que l'autre machine est beaucoup plus lente que la votre, soit que c'est dû à du matériel plus ancien ou un charge plus lourde du système. Vous pouvez essayer d'ajuster la connexion PLIP avec la commande `plipconfig`.

- J'ai installé une connexion PLIP, mais si je lance la commande `ping`, j'obtiens 100% de données perdues. J'ai connecté ma machine à un portable.

- Certains ports parallèles de portables ne sont pas bon pour PLIP, car il s'agit de *ports imprimante* **uniquement**, c'est-à-dire qu'ils peuvent seulement transmettre mais pas recevoir de données. Je ne sais pas s'il existe un moyen d'y arriver.

  Le seul espoir est de :

  * Regarder dans le menu de configuration de votre portable. Il est peut-être possible de configurer le *port parallèle* comme un port parallèle à la place d'un port imprimante ;
  * Essayer `plip` en mode 0. Hélas je ne sais pas comment faire et si ça fonctionne, ou s'il est encore disponible dans les derniers noyaux.

- Quelles vitesses puis-je obtenir avec PLIP ?

  - C'est une question à laquelle il est difficile de répondre. Les performances dépendent fortement de différents facteurs :

    * La vitesse du CPU des deux côtés de la connexion ;
    * Le type de port parallèle et les configurations ;
    * La charge du système ;
    * Le type de connexion PLIP utilisée.

    Pour juste donner une idée, vous devriez obtenir environs 40Koctets/sec, beaucoup plus rapide que n'importe quel taux avec une connexion série et proche d'une carte Ethernet bas-niveau.

- Que se passe-t-il si j'ai besoin d'activer (`ifconfig up`) et désactiver (`ifconfig down`) plusieurs fois plip1 ?

  - Il semble que vous avez besoin d'ajouter l'option `-arp` à la commande `ifconfig` sauf la première fois après chaque démarrage. Je n'en est pas eu besoin, mais peut-être que quelqu'un le fait.

- J'ai lu que des numéros d'IP sont réservés pour les réseaux privés et que `200.0.0.1` et `200.0.0.2` ne sont pas dans ces intervalles. Doivent-ils être changés ?

  - Oui. Mais comme je l'ai souligné depuis le début, j'ai choisi ces adresses IPs uniquement pour leur simplicité. Vous êtes libre de les changer si vous le souhaitez;. Voici un extrait de NET-2-HOWTO(Il s'agit ici d'un extrait du document traduit par Frederic Lepied et Jacques Chion – NdT.) :

    > RFC1597 spécifie quelques adresses IP réservées pour les réseaux privés. Vous devez les utiliser pour éviter d'avoir des problèmes si vous vous retrouvez relié à l'Internet par accident. Ces adresses sont :
    >
    > ```
    > 10.0.0.0      -   10.255.255.255
    > 172.16.0.0    -   172.31.255.255
    > 192.168.0.0   -   192.168.255.255
    > ```

- Est-ce qu'il y a un moyen d'ajuster finement les paramètres de PLIP sans éditer le code source ?

  - Oui, il y en a un ! Essayez la commande `/sbin/plipconfig`. Voir la page du manuel pour plus d'informations.

- Mon système fonctionne avec Debian GNU/Linux, et sous Debian, les fichiers `/etc/rc.d.rc.inet1` et 2 n'existent pas. Où dois-je écrire les commandes de configuration de PLIP ?

  - Sur Debian GNU/Linux vous devez éditer le fichier `/etc/init.d/network`. Vous devez y mettre toutes les commandes que vous devriez écrire dans `rc.inet1` et 2.

- J'ai quelques problèmes à connecter deux machines avec PLIP. La première a le dernier noyau, la seconde utilise encore la version 1.0.x de PLIP :est-ce un problème ?
  - Oui, il vaut mieux avoir, dans la mesure du possible, la même version de PLIP sur les deux machines. Il est écrit dans le fichier `plip.c` que les versions actuelles de PLIP ne fonctionnent pas avec les versions 1.0.x de PLIP.
- Jusqu'à maintenant, PLIP fonctionnait sur 4 bits. Qu 'en est-il de la connexion PLIP sur 8 bits dont j'ai pu lire une description dans la documentation du noyau ? Je pense qu'il s'agit du Mode 1.
  - Ce Mini-HowTo est destiné à la configuration. Pour des informations techniques, s'il vous plait, lisez les fichiers `/usr/src/linux/net/README*.PLIP` ou contactez l'auteur. Je sais seulement ceci : le standard PLIP utilise le câble "null printer" et c'est le Mode 0 (ne le confondez pas avec plip0, qui est le nom de l'interface) qui utilise 4 bits. Le Mode 1 utilise 8 bits et devrait déja être disponible. Cependant, vous devrez faire vous-même le câble et la connexion ne marchera qu'entre 2 machines Linux. Je ne sais, une fois que vous avez le câble, comment configurer la connexion PLIP en Mode 1. Si quelqu'un sait, s'il vous plait, faites le moi savoir.

8 Patches pour utiliser PLIP et LP ensembles

Le meilleur moyen de faire cohabiter PLIP et LP est d'utiliser les modules du noyau : vous pouvez charger `plip.o` et le décharger quand vous avez besoin d'imprimer et *vice versa*. Si vous avez vraiment besoin d'utiliser PLIP et LP ensembles, essayez les patches décrits dans les sections suivantes.

8.1 PLIP et LP sur le même port

Si, pour une raison quelconque, vous souhaitez que PLIP et LP soient supportés directement par le noyau, vous pouvez essayer ces patches.

Vous devez modifier les parties de codes suivantes, MAIS sauvegarder d'abord les fichiers originaux :

```
******** modifications de linux/drivers/char/lp.c ***********************
struct lp_struct lp_table[] = {
        { 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
/*      { 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
        { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
*/
};
#define LP_NO 1

******** modifications de linux/drivers/net/Space.c ********************
#if defined(PLIP) || defined(CONFIG_PLIP)
    extern int plip_init(struct device *);
    static struct device plip2_dev = {
        "plip2", 0, 0, 0, 0, 0x278, 2, 0, 0, 0, NEXT_DEV, plip_init, };
    static struct device plip1_dev = {
        "plip1", 0, 0, 0, 0, 0x378, 7, 0, 0, 0, &plip2_dev, plip_init, };
```

```
/*    static struct device plip0_dev = {
        "plip0", 0, 0, 0, 0, 0x3BC, 5, 0, 0, 0, &plip1_dev, plip_init, };
*/
#   undef NEXT_DEV
#   define NEXT_DEV     (&plip1_dev)
#endif  /* PLIP */
```

Bien sûr, il y a l'avertissement standard : *j'ai reçu ces patches et je les ai mis ici tels quels, c'est-à-dire que* ***vous les essayez à vos risques et périls !*** Cela dit, votre plus gros désagrément ne devrait être que la restauration des fichiers et leur recompilation.

8.2 PLIP et LP sur deux ports différents

Si vous avez au moins deux ports parallèles, vous pouvez essayer ces patches. Ils devraient vous permettre d'utiliser PLIP sur un port et LP sur un autre.

1. Mettre en commentaire un ligne du fichier source du noyau `drivers/char/lp.c`

```
struct lp_struct lp_table[] = {
{ 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
{ 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
/* { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, }, */
};
    3 -> 2
```

2. Configuration du noyau

```
PLIP (parallel port) support (CONFIG_PLIP) [n] y

Parallel Printer support [y] y
```

3. Le message du noyau au démarrage est alors :

```
lp1 at 0x0378, using polling driver
.....
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip2: Parallel port at 0x278, using assigned IRQ 5.
```

Et à nouveau l'avertissement, voir la section 8.1 (PLIP et LP sur le même port).

9 Connexion PLIP entre Dos et Linux

Après la première version de ce Mini-HOWTO, plusieurs personnes m'ont écrites pour avoir des informations au sujet de la connexion entre une machine sous Linux et une autre sous Dos (ou Windows). Cet intérêt me conduit donc à ajouter cette section qui, je l'espère, viendra en aide à chacun.

Cette section provient d'un article que j'ai trouvé dans la *Linux Gazette* dont l'auteur est James McDuffie *mcduffie@scsn.net* . Il couvre l'installation de base d'une connexion PLIP entre deux machines, l'une sous Linux, l'autre sous Dos utilisant Windows et Trumpet Winsock. Il donne aussi l'adresse d'un programme intéressant qui permet d'ouvrir des fenêtres X-Window sous Windows.

La partie suivante est un ensemble de notes envoyées par James Vahn *jvahn@short.circuit.com*

où il décrit en profondeur comment installer cette connexion et comment résoudre plusieurs problèmes.

Pour toute question au sujet de cette section, s'il vous plait contactez James Vahn, pas moi.

9.1 Configuration d'une connexion entre Dos et Linux

Je suppose que vous avez déjà installer correctement le support PLIP du coté de la machine Linux et que vous avez le bon câble. Sinon retournez aux chapitres précédents.

Maintenant, du coté du Dos, vous avez, avant tout, besoin d'un pilote de paquets Ethernet. Vous pouvez en trouver un à cette adresse :

ftp://ftp.crynwr.com/drivers/plip.zip

Le programme s'exécute sous Dos et joue le rôle d'un pilote de paquets Ethernet. Si vous voulez utiliser PLIP sous Windows, vous avez également besoin de Trumpet Winsock. Il sert d'interface TCP/IP. Sinon, vous pouvez probablement trouver un logiciel TCP/IP pour Dos.

Maintenant retournez à l'ordinateur sous Linux et ajoutez l'adresse de l'ordinateur sous Dos dans le fichier `/etc/hosts`. Si votre machine Dos n'a pas de d'adresse IP enregistrée, vous pouvez choisir n'importe quelle adresse (Rappelez vous toutefois l'avertissement de la section 5 (Configuration des fichiers) au sujet des adresses IP).

Supposons que vous choisissez le nom `linux` pour la machine sous Linux et `dos` pour la machine sous Dos. Vous devez taper :

```
ifconfig plip1 linux pointopoint dos arp up
route add dos
```

Bien sûr si vous voulez que ce soit fait à chaque démarrage de Linux, vous devez ajouter ces lignes dans le fichier `/etc/rc.d/rc.inet` :

```
/sbin/ifconfig plip1 linux pointopoint dos arp up
/sbin/route add dos
```

Ces lignes configurent l'interface et lui ajoutent ensuite une route. Bien sûr, si vous utilisez le second port parallèle, vous devez écrire `plip2` à la place de `plip1`.

Retournez à la machine sous Dos/Windows et éditez le fichier `autoexec.bat`, vous devez ajouter les lignes suivantes :

```
c:\plip\plip.com 0x60
c:\tcpip\winsock\winpkt.com 0x60
```

Je suppose que vous avez mis `plip.com` (le pilote de paquets) dans le répertoire `c:\plip` et `winpkt.com` dans `c:\tcpip`, sinon vous devez mettre le bon chemin.

Ces lignes configurent le programme `plip.com` sur le vecteur de paquets `0x60` et chargent ensuite le programme `winpkt.com` qui vient sur le même vecteur que le programme Trumpet Winsock. Si le câble est sur un autre port que `LPT1`, vous devrez donner à `plip.com` le numéro d'IRQ et l'adresse d'E/S. Pour que le programme `winpkt.com` puisse tourner, Windows doit avoir accès au vecteur de paquets. A partir de maintenant, nous allons sous le programme de configuration de Trumpet Winsock. Vous n'avez qu'à désélectionner SLIP ou PPP et d'entrer `60` dans la boîte nommé `Packet vector`. Puis indiquez lui l'adresse IP que vous donnez à la machine sous Dos, celle de l'ordinateur sous Linux comme la passerelle par défaut et le Serveur de Nom comme l'IP de votre ordinateur, ou votre adresse d'ISP pour son serveur de nom (si vous allez vous connecter à l'Internet, voir ci-dessous). Fermez le programme de configuration, relancez Winsock, et ce devrait être bon ! Mettez Winsock dans le *groupe de démarrage* et il se lancera automatiquement à chaque démarrage de Windows.

Si vous voulez accéder à l'Internet à partir de la machine sous Windows, par l'intermédiaire de la machine sous Linux, vous avez besoin de configurer l'*IP Masquerading*. Pour plus d'informations dessus, lisez NET-2-HowTo. Votre ordinateur sous Windows utilise l'adresse IP de la machine sous Linux.

J'ai aussi trouvé un programme qui vous permet de lancer de programmes X-Window sous Windows ! Il se trouve à l'adresse *http://www.tucows.com/* .

Configurez-le en fonction des directions, puis vous n'avez plus qu'à faire un `telnet` depuis votre machine Windows et rediriger l'affichage sur votre machine (`DISPLAY=duncan:0.0` par exemple). Vous pouvez alors lancer le programme désiré.

Il n'y a rien de plus sympa que de faire tourner `xv` sous Windows ! J'espère que tout ceci vous a aidé.

9.2 Expérience pratique de connexion PLIP entre Dos et Linux

NB : J'ai reçu ce document de James Vahn *jvahn@short.circuit.com*

et je le mets ici sans le modifier. Ainsi, **pour les questions au sujet de cette section, il est beaucoup plus qualifié que moi. Il vaut donc mieux le contacter par courrier électronique.** Son expérience de connexion PLIP entre un ordinateur sous Dos n'ayant qu'un lecteur de disquettes et une machine sous Linux est le parfait exemple de la manière d'aborder les problèmes couramment rencontrés.

Dernière mise à jour : 11 juillet 1996

Ma machine Dos qui ne possède qu'un lecteur de disquette est en réseau par une connexion PLIP sur le deuxième port parallèle sur la machine Linux. Une imprimante est branchée sur le premier port parallèle. Les deux machines sont connectées en permanence et la machine sous Dos effectue un `telnet` sur Linux. Voici mes notes sur ma manière de l'obtenir.

Quand le noyau recherche les ports d'imprimantes, il les récupère tous, à moins que vous ayez spécifier de ne pas en rechercher un. Sinon PLIP ne pourrait pas trouver de port libre. Une méthode consiste à charger les pilotes comme des modules lorsque c'est nécessaire...

gniibe@mri.co.jp

écrit :

Je recommande d'utiliser PLIP/LP comme des modules du noyau, pour les raisons suivantes :

- les modules sont flexibles en cas changement de configuration ;
- la (re)compilation du noyau n'est pas facile pour les novices ;
- la co-existance de PLIP et LP n'est possible qu'avec les modules.

Avec PLIP/LP comme modules du noyau, vous pouvez spécifier sur quels ports seront PLIP et LP. Voici un exemple :

```
# insmod lp.o io=0x378
# insmod plip.o io=0x278 irq=2
```

Vous pouvez même utiliser deux ports parallèles :

```
# insmod plip.o io=0x278,0x3bc irq=2,5
```

Dans l'exemple ci-dessus,

`plip0` est placé à l'adresse 0x278 et son IRQ est 2 ;

`plip1` est placé à l'adresse 0x3bc et son IRQ est 5.

Les modules semblent apparemment être le meilleur moyen pour utiliser PLIP et LP ensembles. La méthode suivante montre comment installer modifier les sources du noyau afin de permettre à la fois l'imprimante et PLIP sur différents ports sans utiliser de modules. Si vous n'êtes pas familier avec les modules, vous pourrez trouver ça plus rapide à configurer.

Vous aurez besoin de modifier deux fichiers dans l'arborescence des sources du noyau. J'utilise le noyau 1.2.3 et effectuer quelques changements nécessaires dans le fichier `../linux/drivers/net/Space.c` pour qu'il soit conforme à mon système. Regardez aux alentours de la ligne 205 (Le numéro de la ligne dépend de la version du noyau. Pour un version plus récente (2.0.30), il faudra regarder autour de la ligne numéro 295 – NdT.) pour la définitions de PLIP afin d'être sûr que le port et l'IRQ conviennent, et noter quel pilote sera utilisé (plip0, plip1, plip2). Dans mon cas, le port `0x278` utilise l'IRQ 5 (la carte est ainsi configurée), alors que `Space.c` le définit avec l'IRQ 2. J'ai préféré faire cette modification plutôt que d'ouvrir la machine pour changer les jumpers. L'autre alternative consiste à spécifier l'IRQ dans la ligne de commande de `ifconfig`, mais le noyau démarrera avec le mauvais IRQ pour PLIP et ce pourrait être ennuyeux pour vous. Ici, ce n'est qu'un simple changement (un seul caractère).

L'étape suivante est la plus délicate :

Dans le fichier `../drivers/char/lp.c`, vous trouverez ce qui suit aux alentours de la ligne 35 (Même numéro de ligne pour la version 2.0.30 du noyau – NdT) :

```
struct lp_struct lp_table[] = {
    { 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
    { 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
/*  { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
 * 0x278 reserved for plip1
 *
 * };
 * #define LP_NO 3
 */
};
#define LP_NO 2
```

Notez que les changements à faire : une des définitions de port est mise en commentaire. Seulement deux ports sont donc définis. Le port 0x3bc ne fonctionnera probablement pas pour PLIP : la ligne d'IRQ n'est normalement pas conserver sur ces ports, comme on le trouve sur les vieux adaptateurs monochromes (MDA).

Vous avez effectué une sauvegarde de ces fichiers avant les changer, non ? Maintenant, configurez le nouveau noyau avec les supports `printer` (imprimante), `net` (réseau), `dummy` (support fictif) et `plip`.

Configuration du système. Voici mon fichier `/etc/rc.d/rc.inet1` :

```
#!/bin/bash
#
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0

/sbin/ifconfig dummy 200.0.0.1
/sbin/route add -net 200.0.0.0 netmask 255.255.255.0
/sbin/ifconfig plip1 arp 200.0.0.1 pointopoint 200.0.0.2 up
/sbin/route add 200.0.0.2
/sbin/ifconfig dummy down
```

Notez que `arp` est utilisé pour la connexion Dos-Linux et n'est apparemment pas nécessaire pour lors de connexions Linux-Linux.

Ensuite ajoutez dans le fichier `/etc/hosts` les lignes suivantes, elles assignent un nom à chaque machine :

```
200.0.0.1       console1
200.0.0.2       console2
```

La machine Dos porte le nom de `console2`. Rappelez vous l'avertissement d'Andrea au sujet des IPs : il est préférable d'utiliser les schémas officiels de numérotation.

Redémarrez la machine afin que les changements et le nouveau noyau prennent effet. Pendant la séquence de démarrage (ou en lançant `dmesg`), vous devez avoir les messages suivants si vous avez utiliser les patches, sinon quand les modules sont chargés :

```
lp0 at 0x03bc, using polling driver
lp1 at 0x0378, using polling driver
[....]
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip1: Parallel port at 0x278, using assigned IRQ 5.
```

La commande `route` affiche alors ceci :

```
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
console2        *               255.255.255.255 UH    1436   0      136 plip1
loopback        *               255.0.0.0       U     1936   0      109 lo
```

Et la commande `ifconfig plip1` affiche :

```
plip1     Link encap:10Mbps Ethernet  HWaddr FC:FC:C8:00:00:01
          inet addr:200.0.0.1  P-t-P:200.0.0.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:1500  Metric:1
          RX packets:132 errors:0 dropped:0 overruns:0
          TX packets:136 errors:0 dropped:0 overruns:0
          Interrupt:5 Base address:0x278
```

Regarder dans le fichier `/etc/inetd.conf` s'il est possible d'effectuer un `telnet`. Vous pouvez lire la page de manuel de `tcpd`, l'utilisation des fichiers `/etc/hosts.allow (ALL: LOCAL)` et `/etc/hosts.deny (ALL: ALL)`. Vous devriez être capable de faire un `telnet` sur une machine locale.

Pour Linux c'est fait, voyons maintenant du coté Dos. A nouveau, faites attention au port `0x3bc` s'il y en a un présent.

J'utilise le programme `telnet` de NCSA et le pilote PLIP de Crynwr trouvés sur les sites suivants :

```
ftp://ftp.ncsa.uiuc.edu/Telnet/DOS/ncsa/tel2308b.zip
ftp://ftp.crynwr.com/drivers/plip.zip
```

Soyez sûr que vous utilisez la version 2.3.08 du telnet de NCSA et la version 11.1 du pilote PLIP de Crynwr. S'il vous plaît, lisez le fichier `SUPPORT.DOC` de Crynwr qui doit se trouver quelque part.

Le fichier `CONFIG.TEL` : la grande partie du fichier est utilisable par défaut et pour économiser de la place, je n'ai mis ci-dessous que les informations dont vous avez besoins (je l'espère). Le deuxième port sur cette machine est configuré comme `0x278` sur l'IRQ 5.

```
myip=200.0.0.2
netmask=255.255.255.0       # masque de reseau
hardware=packet             # ensemble d'adaptation du reseau (interface du pilote de paquets)
interrupt=5                 # IRQ avec laquelle l'adaptateur est configure
ioaddr=60                   # vecteur d'interruption logiciel que le pilote utilise
#
#[...partie inchangee...]
#
# a la fin du fichier, mettez cette ligne :
name=console1 ; hostip=200.0.0.1 ; nameserver=1 ; gateway=1
```

(`console1` est le nom de la machine Linux, vous pouvez utiliser celui que vous désirez.)

J'ai fabriqué en câble de 3,5 mètre entre les deux machines, et (après m'être aperçu que je n'avais pas fait les bons cablâges) il n'y a eu aucun problème. Un câble standard d'imprimante 11-brins devrait aussi faire l'affaire. Le fichier `plip.c` des sources de Linux montre le câblage. Bien que mon câble permettent une connexion 17-17, je ne pense pas qu'il soit utilisé pour n'importe quoi et qu'il existe tel quel dans le commerce.

```
@echo off
plip.com 0x60 5 0x278
telbin -s console1
```

Les lignes précédentes vous connectent à la machine Linux sur `/dev/ttyp`. Le programme `telnet` de NCSA fournit 8 écrans virtuels et aussi agit comme un serveur ftp. L'interface PLIP fournit une assez bonne connexion, j'obtiens un taux de transfert de 6.5Koctets/s sur mes antiquités. J'espère que vous pourrez faire mieux.

10 Connexion PLIP entre Linux et Windows95

Cette section est vide. Je n'utilise pas Windows95 pour des choses sérieuses mais pour des jeux. Je n'ai donc pas essayé et une connexion PLIP avec Linux ne m'interresse pas. La question au sujet d'une telle connexion a gagné le concours de la question la plus posée. Je donne donc ici une réponse (pour l'instant) définitive.

Non, jusqu'à maintenant, personne ne m'a rapporté avoir réussi une connexion entre PLIP et Windows95. Si quelqu'un y arrive, dites le moi immédiatement : des milliers d'utilisateurs de PLIP sont dans l'attente de ces nouvelles.

11 Questions ? Commentaires ? Envoyez-moi vos réactions

Vous pouvez m'envoyer toutes les questions et commentaires par courrier électronique à l'adresse suivante : *controzz@cli.di.unipi.it*

Toute réaction est la bien venue, toute indication d'erreur est précieuse. La Foire Aux Questions devrait être plus importante dans la prochaine version, si vous m'envoyez des questions, et bien sûr les réponses si vous les trouver par vous même.

S'il vous plait, n'envoyer pas de questions qui se trouve déjà dans la FAQ.

Si vous devez me demander de l'aide, s'il vous plait soyez sûr de me faire connaître toutes les informations susceptibles de m'aider. Donnez moi au moins : la version du noyau, les commandes utilisées, les messages d'erreurs, le câble que vous utilisez et d'autres messages du système en rapport avec PLIP.

S'il vous plait, rappelez vous de ne pas m'envoyer de questions au sujet des connexions PLIP avec DOS/Windows 3.1/ Windows 95, je ne peux pas vous aider. Ces questions doivent être envoyées à James Vahn *jvahn@short.circuit.com* qui m'a envoyé la partie sur DOS. A nouveau : c'est inutile de poser à lui comme à moi des questions sur les connexsions PLIP avec Windows 95.

12 Où trouver les nouvelles versions de ce Mini-HowTo

Ce Mini-HowTo est maintenu par les coordinateurs des HowTos. Il est posté tous les mois sur le newsgroup *comp.os.linux.answers* . Il peut être trouvé dans le répertoire des HOWTOs de *sunsite.unc.edu* et de sites mirroirs de `sunsite`.

Un autre moyen de trouver cet Mini-HowTo (et de me contacter) d'aller sur ma page web personnelle :

http://www.cli.di.unipi.it/˜controzz/intro.html (en italien)

http://www.cli.di.unipi.it/˜controzz/intro_ e.html (en anglais)

13 Remerciements

Tous mes remerciements à :

- Rick Lim *ricklim@freenet.vancouver.bc.ca* pour les patches permettant l'utilisation de PLIP et LP ensembles ;
- Takeshi Okazaki *GBA03552@niftyserve.or.jp* pour les patches permettant l'utilisation de PLIP et LP sur deux ports parallèles différents ;
- Jim Van Zandt *jrv@vanzandt.mv.com* pour les conseils sur la partie *tutorielle* de cet HowTo ;
- Fernando Molina *fmolina@nexo.es* pour les comentaires utiles au sujet des IRQs et des adresses d'entrée-sortie.
- James Vahn *jvahn@short.circuit.com* pour la note sur la connexion PLIP entre Dos et Linux ;
- Ceux que ont postés des articles en rapport avec PLIP sur les newsgroupes linux et/ou qui m'ont envoyé des messages. La liste de tous ceux qui m'ont aidé avec des informations et des commentaires serait plus longue que le Mini-HowTo lui-même : Merci à tous !

14 Copyright