Le Linux HOWTO pour les ports série

Greg Hankins, *greg.hankins@cc.gatech.edu*
traduit par Eric Festinger, *eric@midix.frmug.org*
v1.11, 15 Novembre 1997

Ce document décrit la configuration sous Linux des ports série, des programmes associés, et des périphériques qui y sont connectés.

Contents

1 Introduction

Ce document est la traduction française du Linux Serial HOWTO. Il tente de décrire la configuration des modems et terminaux sous Linux, de donner quelques conseils, et enfin de résoudre les problèmes liés aux ports série. Il est destiné plus particulièrement aux plateformes Intel x86, même s'il doit être applicable à d'autres architectures.

1.1 Copyright

1.2 Nouvelles versions de ce document

La version originale la plus récente de ce document est disponible à *ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/* ainsi que sur les sites miroirs. D'autres formats (PostScript, DVI...) existent dans le répertoire *other-formats* . Ce HOWTO est également consultable à *http://sunsite.unc.edu/LDP/HOWTO/Serial-HOWTO.html* et est posté mensuellement dans le forum de discussions *comp.os.linux.answers* .

La version française est disponible sur *http://www.freenix.fr/linux/HOWTO/Serial-HOWTO.html* , et est postée tous les mois dans *fr.comp.os.linux.annonces* .

1.3 Retour d'informations

Merci de me faire parvenir par email à *greg.hankins@cc.gatech.edu* toutes questions, commentaires et suggestions. Je suis extrêmement intéressé par ce que vous pensez à propos de ce HOWTO, et par les éventuelles améliorations. Faites-moi part de ce que vous ne comprenez pas ou de ce qui nécessite des éclaircissements. Je répondrais à tous les emails, même si ça peut me prendre une semaine ou plus. Je reçois beaucoup de messages de `root` ou de sites mal configurés. Assurez-vous de m'envoyer un message électronique en tant que personne réelle, ainsi que du bon fonctionnement de votre système de messagerie. Si vous n'avez toujours pas de réponse de ma part au bout de deux semaines, merci de m'envoyer à nouveau votre message.

Je peux également être joint à :
Greg Hankins
College of Computing
801 Atlantic Drive
Atlanta, GA 30332-0280
par courrier postal, et à *http://www.cc.gatech.edu/staff/h/Greg.Hankins/* .

Merci d'inclure le numéro de version de ce HOWTO dans votre message. Ceci est la version 1.11.

1.4 Avertissement

Votre système peut varier par rapport aux exemples, et les solutions présentées dans ce HOWTO peuvent ne pas fonctionner.

2 Matériel supporté

Les matériels suivants fonctionnent correctement sous Linux.

2.1 Cartes série standards pour PC

- cartes série standards pour PC (COM1 - COM4), auxquelles peuvent être connectés des périphériques série externes (modems, souris série, etc.)
- modems internes standards pour PC (COM1 - COM4)
- Quickpath Systems Port-Folio 550e (autorisant les IRQ 3, 4, 5, 9, 10, 11, 12, ou 15)

Remarque : en raison de conflits d'adresses, vous ne pouvez pas utilisez simultanément COM4 et une carte vidéo IBM8514. Ceci est dû à un bogue de la carte IBM8514.

2.2 Modems Plug And Play (PnP)

Même si les modems PnP fonctionnent sous Linux, je ne les recommande pas. Néanmoins, voici quelques conseils :

- Essayez de désactiver les fonctionnalités PnP dans le BIOS de l'ordinateur et du modem.
- Démarrez sous Win95, configurez le modem, et vérifiez que les paramètres sont sauvegardés lorsque vous redémarrez sous Linux. Si ce n'est pas le cas :
- Démarrez sous Win95, configurez le modem, et redémarrez à chaud (*warm boot*) sous Linux, de manière à ce que les paramètres soient préservés. Vous pouvez effectuer cette manipulation avec `loadlin`.
- Récupérez les outils PnP à *http://www.redhat.com/linux-info/pnp* .
- Et, bien sûr, utilisez un *vrai* modem :-)

2.3 Cartes série multiports simples (équipées d'UART 8250/16450/16550A)

- AST FourPort et compatibles (4 ports)
- Accent Async-4 (4 ports)
- Arnet Multiport-8 (8 ports)
- Bell Technologies HUB6 (6 ports)
- Boca BB-1004 (4 ports), BB-1008 (8 ports), BB-2016 (16 ports)
- Boca IOAT66 (6 ports)
- Boca 2by4 (4 ports série, 2 ports parallèles)
- Computone ValuePort V4-ISA (compatible AST FourPort)
- Digi PC/8 (8 ports)
- GTEK BBS-550 (8 ports)
- Longshine LCS-8880, Longshine LCS-8880+ (compatible AST FourPort)
- Moxa C104, Moxa C104+ (compatible AST FourPort)
- PC-COMM (4 ports)
- *Sealevel Systems* `http://www.sealevel.com` COMM-2 (2 ports), COMM-4 (4 ports) et COMM-8 (8 ports)
- SIIG I/O Expander 2S IO1812 (4 ports)
- STB-4COM (4 ports)
- Twincom ACI/550
- Usenet Serial Board II (4 ports)

De manière générale, Linux supporte n'importe quelle carte équipée d'un UART 8250, 16450, 16550 ou 16550A (ou compatible), et n'importe quel modem émulant un de ces composants.

Remarque : les BB-1004 et BB-1008 ne supportent pas les signaux DCD et RI, et ne sont donc pas utilisables pour des modems répondeurs. Ils fonctionnent néanmoins très bien dans les autres cas.

2.4 Cartes série multiports intelligentes

- Comtrol RocketPort (ASIC 36MHz ; 4, 8, 16 ou 32 ports)
  contact : *info@comtrol.com* ou *http://www.comtrol.com*
  état du gestionnaire de périphérique : supporté par Comtrol
  emplacement : *ftp://tsx-11.mit.edu/pub/linux/packages/comtrol*
- Computone IntelliPort II (80186 16MHz ; 4, 8, ou 16 ports),
  IntelliPort II EXpandable (80186 20MHz ; 16 - 64 ports)
  contact : Michael H. Warfield, *mhw@wittsend.atl.ga.us*
  état du gestionnaire de périphérique : pré-ALPHA

- Cyclades Cyclom-Y (UART Cirrus Logic CD1400 ; 8 - 32 ports),
  Cyclom-Z (MIPS R3000 25MHz ; 8 - 128 ports)
  contact : *sales@cyclades.com* ou *http://www.cyclades.com*
  état du gestionnaire de périphérique : supporté par Cyclades
  emplacement : *ftp://ftp.cyclades.com/pub/cyclades* et inclus dans le noyau Linux depuis la version 1.1.75

- Decision PCCOM8 (8 ports)
  contact : *pccom8@signum.se*
  emplacement : *ftp://ftp.signum.se/pub/pccom8*

- Digi PC/Xi (80186 12.5MHz ; 4, 8, ou 16 ports),
  PC/Xe (80186 12.5/16MHz ; 2, 4, ou 8 ports),
  PC/Xr (IDT3041 16MHz ; 4 ou 8 ports),
  PC/Xem (IDT3051 20MHz ; 8 - 64 ports)
  contact : *sales@dgii.com* ou *http://www.dgii.com*
  état du gestionnaire de périphérique : supporté par Digi
  emplacement : *ftp://ftp.dgii.com/drivers/linux* et inclus dans le noyau Linux depuis la version 2.0

- Digi COM/Xi (80188 10MHz ; 4 ou 8 ports)
  contact : Simon Park, *si@wimpol.demon.co.uk*
  état du gestionnaire de périphérique : ALPHA
  Remarque : régulièrement, Simon ne peut pas lire son mail pendant plusieurs mois à cause de son travail. Mark Hatle, *fray@krypton.mankato.msus.edu* s'est gracieusement porté volontaire pour rendre le gestionnaire de périphérique disponible si vous en avez besoin. Néanmoins, Mark ne s'occupe pas de la maintenance ni du support de ce gestionnaire.

- Equinox SuperSerial Technology (ASIC 30MHz ; 2 - 128 ports)
  contact : *sales@equinox.com* ou *http://www.equinox.com*
  état du gestionnaire de périphérique : supporté par Equinox
  emplacement : *ftp://ftp.equinox.com/library/sst*

- GTEK Cyclone (UART 16C654 ; 6, 16 et 32 ports),
  SmartCard (Dallas DS80C320 24MHz ; 8 ports),
  BlackBoard-8A (UART 16C654 ; 8 ports),
  PCSS (8032 15/24MHz ; 8 ports)
  contact : *spot@gtek.com* ou *http://www.gtek.com*
  état du gestionnaire de périphérique : supporté par GTEK
  emplacement : *ftp://ftp.gtek.com/pub*

- Hayes ESP (COM-bic ; 1 - 8 ports)
  contact : Andrew J. Robinson, *arobinso@nyx.net* ou *http://www.nyx.net/~arobinso*
  état du gestionnaire de périphérique : supporté par l'auteur
  emplacement : *http://www.nyx.net/~arobinso* et inclus dans le noyau Linux depuis la version 2.1.15

- Maxpeed SS (Toshiba ; 4, 8 et 16 ports)
  contact : *info@maxpeed.com* ou *http://www.maxpeed.com*
  état du gestionnaire de périphérique : supporté par Maxpeed
  emplacement : *ftp://maxpeed.com/pub/ss*

- Moxa C218 (80286 12MHz ; 8 ports),
  Moxa C320 (TMS320 40MHz ; 8 - 32 ports)

contact : *info@moxa.com.tw* ou *http://www.moxa.com.tw*
état du gestionnaire de périphérique : supporté par Moxa
emplacement : *ftp://ftp.moxa.com.tw/drivers/c218-320/linux*

- SDL RISCom/8 (Cirrus Logic CD180 ; 8 ports)
  contact : *sales@sdlcomm.com* ou *http://www.sdlcomm.com*
  état du gestionnaire de périphérique : supporté par SDL
  emplacement : *ftp://ftp.sdlcomm.com/pub/drivers*
- Specialix SIO (Zilog Z280 20MHz ; 4 - 32 ports),
  XIO (25MHz Zilog Z280 ; 4 - 32 ports)
  contact : Simon Allen, *simonallen@cix.compulink.co.uk*
  état du gestionnaire de périphérique : BETA
  emplacement : *ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial*
- Stallion EasyIO-4 (4 ports), EasyIO-8 (8 ports), et
  EasyConnection (8 - 32 ports) - toutes avec des UART,
  Stallion (CPU 80186 8MHz ; 8 ou 16 ports),
  Brumby (CPU 80186 10/12 MHz ; 4, 8 ou 16 ports),
  ONboard (CPU 80186 16MHz ; 4, 8, 12, 16 ou 32 ports),
  EasyConnection 8/64 (CPU 80186 25MHz ; 8 - 64 ports)
  contact : *sales@stallion.com* ou *http://www.stallion.com*
  état du gestionnaire de périphérique : supporté par Stallion
  emplacement : *ftp://ftp.stallion.com/drivers/ata5/Linux* et inclus dans le noyau Linux depuis la version 1.3.27

Un descriptif des produits Comtrol, Cyclades, Digi, et Stallion est paru dans le numéro 14 de juin 1995 du *Linux Journal*. Il est disponible à *http://www.ssc.com/lj/issue14* .

2.5 Matériel non supporté

Les modems dits “Winmodems”, par exemple l'USR Sportster Winmodem et l'IBM Aptiva MWAVE, ne sont *pas* supportés sous Linux. Ils utilisent des technologies propriétaires nécessitant des gestionnaires de périphériques spéciaux pour Windows. Les modems nécessitant des gestionnaires Rockwell RPI sont à éviter pour la même raison.

Les cartes série intelligentes nécessitant des gestionnaires non disponibles sous Linux, ne pourront pas fonctionner.

3 Comment se nomment les ports série ?

Un port d'E/S permet de recevoir des données depuis, et d'en envoyer à partir d'un ordinateur. Il existe plusieurs types de ports d'E/S, comme les ports série, les ports parallèles, les contrôleurs de disques durs, les cartes Ethernet, etc. Les modems et les terminaux dont nous allons traiter n'utilisent que des ports série. Chacun de ces ports possède une adresse d'E/S, et un vecteur d'interruption (IRQ). Les quatre ports série suivants correspondent à COM1 - COM4 :

```
ttyS0 (COM1) adresse 0x3f8 IRQ 4
ttyS1 (COM2) adresse 0x2f8 IRQ 3
ttyS2 (COM3) adresse 0x3e8 IRQ 4
ttyS3 (COM4) adresse 0x2e8 IRQ 3
```

Si aucun port série n'est détecté lors de l'amorçage de Linux, assurez-vous que leur support a été sélectionné et compilé dans le noyau. Dans ce document, je me réfère à COM1 pour `ttyS0`, COM2 pour `ttyS1`, COM3 pour `ttyS2`, et à COM4 pour `ttyS3`. Par défaut, ces ports ont des IRQ communes, et vous ne pouvez pas les utiliser simultanément dans cette configuration. Certaines de ces IRQ doivent être réassignées. Reportez-vous à la section 8 (Puis-je utiliser plus de deux ports série ?).

Sur certaines installations, deux périphériques supplémentaires vont être créés : `/dev/modem` pour le modem et `/dev/mouse` pour la souris. Ce sont des liens symboliques sur le périphérique approprié (dans le répertoire `/dev`) que vous avez spécifié lors de l'installation. Si vous possédez une souris de type bus PS/2, `/dev/mouse` pointera correctement sur le périphérique de la souris bus.

Plusieurs discussions ont porté sur les avantages et inconvénients de ces pseudo-périphériques `/dev/mouse` et `/dev/modem`. Personnellement, je décourage *fortement* l'utilisation de ces liens. En particulier, si vous prévoyez d'employer plusieurs modems pour les connexions entrantes sur `/dev/modem`, des problèmes peuvent apparaître à cause des fichiers de verrouillage (*lock files*). Vous pouvez néanmoins les utiliser si vous le désirez, mais *assurez-vous qu'ils pointent bien sur les bons périphériques*. Cependant, si vous supprimez ces liens, certaines applications (comme `minicom`) peuvent ne plus fonctionner sans reconfiguration.

3.1 Périphériques série et numéros dans /dev

```
/dev/ttyS0 majeur 4, mineur 64         /dev/cua0 majeur 5, mineur 64
/dev/ttyS1 majeur 4, mineur 65         /dev/cua1 majeur 5, mineur 65
/dev/ttyS2 majeur 4, mineur 66         /dev/cua2 majeur 5, mineur 66
/dev/ttyS3 majeur 4, mineur 67         /dev/cua3 majeur 5, mineur 67
```

Remarque : toutes les distributions doivent être livrées avec ces périphériques déjà correctement configurés, ce que vous pouvez vérifier en tapant les commandes :

```
linux% ls -l /dev/cua*
linux% ls -l /dev/ttyS*
```

3.1.1 Créer les périphériques dans /dev

Si un périphérique est absent, vous devez le créer avec la commande `mknod`. Par exemple, pour `ttyS0` :

```
linux# mknod -m 666 /dev/cua0 c 5 64
linux# mknod -m 666 /dev/ttyS0 c 4 64
```

Le script `MAKEDEV`, situé dans le répertoire `/dev`, peut être utilisé pour simplifier leur création. Par exemple, pour `ttyS0` :

```
linux# cd /dev
linux# ./MAKEDEV ttyS0
```

Cette commande crée les périphériques d'entrée et de sortie correspondants, et positionne également les bonnes permissions.

3.2 Remarques pour les cartes multiports simples

Les périphériques utilisés par les cartes multiports sont dépendants du type de carte que vous possédez. Ils sont listés en détail dans le fichier `rc.serial` fourni avec le programme `setserial`. Je recommande vivement de récupérer la version la plus récente de `setserial` si vous devez utiliser une telle carte. Vous devrez alors sûrement créer ces périphériques, soit avec la commande `mknod`, soit avec le script `MAKEDEV`. Ils doivent être déclarés en ajoutant 64 au numéro du port. Par exemple, pour `ttyS17` :

```
linux# mknod -m 666 /dev/cua17 c 5 81
linux# mknod -m 666 /dev/ttyS17 c 4 81
```

Car “64 + 17 = 81”. En utilisant le script `MAKEDEV`, il suffit d’entrer :

```
linux# cd /dev
linux# ./MAKEDEV ttyS17
```

Remarque : le listing des ports COM5-COM8 pour les IO1812 est erroné dans le manuel SIIG. Les valeurs doivent être remplacées par COM5=0x250, COM6=0x258, COM7=0x260, et COM8=0x268.

Remarque : le registre d’état d’interruption (*Interrupt Status Register*) de la carte Digi PC/8 se trouve en 0x140.

Remarque : pour la carte AST Fourport, le fichier `rc.serial` doit spécifier le paramètre `skip_test`.

3.3 Remarques pour les cartes multiports intelligentes

Lisez les informations fournies avec le gestionnaire. Ces cartes utilisent des périphériques non-standards. Ces informations varient suivant le matériel.

4 Quelques programmes intéressants à connaître

4.1 Qu’est-ce que `getty` ?

`getty` est un programme qui gère la phase d’entrée en session (*login*) sur un système Unix. Il est indispensable si vous souhaitez vous loguer sur votre machine Linux avec un modem. Il n’est par contre pas utile pour des communications sortantes. Il existe trois versions couramment utilisées avec Linux : `getty_ps`, `mgetty` et `agetty`. Chacune possède sa propre syntaxe. Aussi, assurez-vous d’utiliser celle correspondant à votre version de `getty`.

4.1.1 À propos de `getty_ps`

La plupart des distributions installent directement le paquetage `getty_ps`. Il contient deux programmes : `getty` pour la console et les terminaux, et `uugetty` pour les modems. Je parlerai essentiellement de cette version, car c’est celle que j’utilise.

4.1.2 À propos de `mgetty`

`mgetty` est un `getty` spécialisé pour l’utilisation avec des modems. Par rapport aux autres versions, `mgetty` gère les fax, et détecte automatiquement les connexions PPP. La documentation est excellente et suffisante. Veuillez vous y reporter pour l’installation. Les informations les plus récentes sur `mgetty` peuvent être trouvées à *http://www.leo.org/~doering/mgetty/* .

4.1.3 À propos de `agetty`

`agetty` est la troisième variante de `getty`. C’est une version simple complètement fonctionnelle, et plus particulièrement destinée à l’utilisation des consoles virtuelles et des terminaux plutôt que des modems.

4.2 Qu'est-ce que `setserial` ?

`setserial` est un programme autorisant la consultation et la modification des différents attributs (adresse de port, interruption, etc.) d'un périphérique série. Vous pouvez déterminer le numéro de version installée en lançant `setserial` sans argument.

Lorsque votre système Linux démarre, seuls `ttyS{0-3}` sont configurés avec les IRQ positionnées par défault à 4 et 3. Si vous possédez d'autres ports série sur d'autres cartes, ou si `ttyS{0-3}` utilisent des IRQ non standards, vous *devez* lancer ce programme pour configurer ces ports. Consultez le manuel pour une description complète des options.

5 Comment puis-je appeler avec mon modem ?

5.1 Matériel nécessaire

Tout d'abord, assurez-vous d'avoir le bon câble. Votre modem requiert un câble droit, sans inversion de fils. Vous devriez pouvoir vous en procurer dans n'importe quel magasin d'informatique. Vérifiez également le genre des connecteurs. Pour un port à 25 broches, ce sera toujours un modèle mâle DB25, à ne pas confondre avec les ports parallèles à embase femelle DB25. Connectez votre modem sur un de vos ports série. Consultez si nécessaire le manuel de votre modem.

5.1.1 Remarques pour les modems internes

Pour un modem interne, aucun câble n'est nécessaire. De même, aucun port série n'est requis, car la carte modem en possède un interne. Vous devez seulement configurer votre modem pour qu'il utilise une interruption (IRQ) et une adresse d'E/S libres. Reportez-vous au manuel de votre modem, ainsi qu'à la section 8 (Puis-je utiliser plus de deux périphériques série ?) si vous avez besoin d'aide pour choisir les interruptions et les adresses.

Sur certaines cartes mères, il peut être nécessaire de désactiver les ports série afin d'éviter les conflits avec le modem. Cette manipulation peut être effectuée avec des cavaliers ou les paramètres du BIOS. Consultez le manuel de votre carte mère.

Suite à un bogue des cartes vidéo IBM8514, les modems internes ne peuvent pas fonctionner sur `ttyS3`. Si Linux ne détecte pas le modem sur `ttyS3`, il suffit alors d'utiliser `setserial` pour que le modem fonctionne correctement. Un modem interne `ttyS{0-2}` ne devrait pas poser de problème de détection. Linux n'effectue pas d'autoconfiguration du `ttyS3` à cause de ce bogue lié à la carte vidéo.

5.2 "Parlez" à votre modem

Avant de vous lancez dans le paramétrage assez complexe de SLIP ou PPP, utilisez `kermit` ou n'importe quel autre programme de communication pour tester votre configuration. La version la plus récente de `kermit` peut être téléchargée depuis *http://www.columbia.edu/kermit/* . L'exemple suivant suppose que votre modem est sur `ttyS3`, et que sa vitesse est fixée à 115200 bps :

```
linux# kermit
C-Kermit 6.0.192, 6 Sep 96, for Linux
 Copyright (C) 1985, 1996,
  Trustees of Columbia University in the City of New York.
Default file-transfer mode is BINARY
Type ? or HELP for help.
```

```
C-Kermit>set line /dev/ttyS3
C-Kermit>set carrier-watch off
C-Kermit>set speed 115200
/dev/ttyS3, 115200 bps
C-Kermit>c
Connecting to /dev/ttyS3, speed 115200.
The escape character is Ctrl-\ (ASCII 28, FS)
Type the escape character followed by C to get back,
or followed by ? to see other options.
ATE1Q0V1                  ; vous tapez cette chaine puis la touche Entree
OK                        ; le modem doit repondre comme ca
```

Si votre modem répond aux commandes AT, vous pouvez supposer qu'il fonctionne correctement du côté Linux. Il ne reste plus qu'à essayer d'appeler un autre modem :

```
ATDT7654321
```

où 7654321 est un numéro de téléphone. Utilisez ATDP à la place de ATDT si vous n'êtes pas connecté à un central numérique. Votre modem fonctionne si l'appel aboutit.

Pour revenir à l'invite `kermit`, appuyez simultanément sur les touches Ctrl et backslash, puis sur C :

```
Ctrl-\-C
(Retour a linux)
C-Kermit>quit
linux#
```

Il s'agissait juste d'un test de numérotation "à la main". La méthode normale est de laisser `kermit` numéroter à votre place à l'aide de sa base de données interne de modems. Par exemple, pour un modem US Robotics (USR) :

```
linux# kermit
C-Kermit 6.0.192, 6 Sep 1997, for Linux
 Copyright (C) 1985, 1996,
  Trustees of Columbia University in the City of New York.
Default file-transfer mode is BINARY
Type ? or HELP for help
C-Kermit>set modem type usr        ; Selection du type de modem
C-Kermit>set line /dev/ttyS3       ; Choix du peripherique
C-Kermit>set speed 115200          ; Vitesse de communication
C-Kermit>dial 7654321              ; Numerotation
 Number: 7654321
 Device=/dev/ttyS3, modem=usr, speed=115200
 Call completed.<BEEP>
Connecting to /dev/ttyS3, speed 115200
The escape character is Ctrl-\ (ASCII 28, FS).
Type the escape character followed by C to get back,
or followed by ? to see other options.

Welcome to ...

login:
```

Reportez-vous à la section 10 (Programmes de communication) pour plus de liens.

Lorsque vous appelez avec votre modem, paramétrez la vitesse au débit maximal supporté par votre modem. Les systèmes Linux avec une version de la `libc` supérieure à 5.x supportent des vitesses jusqu'à 115200 bps. `libc` est habituellement située dans `/lib`. Vérifiez dans ce répertoire la version que vous avez. Si Linux ne reconnaît pas des vitesses de 57600 ou 115200 bps, vous devez alors utiliser le program `setserial` pour passer votre port série à une vitesse supérieure. Reportez-vous à la section 9 (Comment configurer mes ports série à des vitesses supérieures ?). Sélectionnez ensuite une vitesse de 38400 bps dans votre programme de communications.

5.3 Configuration du modem en mode appelant

Pour appeler, vous pouvez configurer votre modem comme vous le voulez. Si vous souhaitez l'utiliser pour des communications entrantes, vous *devez* paramétrer votre modem à la même vitesse que pour `getty`. Ainsi, si vous voulez exécuter `getty` à 38400 bps, fixez également la vitesse à 38400 bps lors du paramétrage du modem, pour éviter une incompatibilité de vitesses avec l'ordinateur. En général, les réglages d'usine avec correction d'erreurs et contrôle de flux matériel correspondent au réglage optimal en mode appelant. Consultez le manuel du modem pour obtenir ces réglages.

5.4 Contrôle de flux matériel

Je vous recommande vivement d'utiliser le contrôle de flux matériel (RTS/CTS) si votre modem le supporte. Cette fonctionalité est particulièrement importante lorsque la compression de données est active. Premièrement, vous devez autoriser le contrôle de flux RTS/CTS sur le port série. Le mieux est de le faire au démarrage, par exemple dans `/etc/rc.d/rc.local` ou `/etc/rc.d/rc.serial`. Assurez-vous que ces fichiers sont bien exécutés par le fichier principal `rc.sysinit`. Ensuite, pour chacun des ports série pour lesquels vous désirez activer le contrôle de flux matériel, vous devez entrer la commande suivante :

```
stty crtscts < /dev/ttyS3
```

Vous devez également activer le contrôle de flux RTS/CTS sur le modem. Consultez le manuel du modem, car le paramétrage varie suivant les fabricants. N'oubliez pas de sauvegarder cette configuration, si votre modem le permet.

6 Comment puis-je appeler et être appelé avec mon modem et getty_ps ?

Faites déjà fonctionner correctement votre modem en mode appelant. Si vous n'avez pas encore lu la section 5 (Comment puis-je appeler avec mon modem ?), *lisez-la maintenant* ! Elle contient des informations importantes sur la configuration. Il *n'est pas nécessaire* de lire cette section si vous souhaitez juste *appeler* avec votre modem.

6.1 Configuration en modes appelant et appelé

Encore une fois, le modem *doit* être spécialement configuré (avec les commandes `AT`) afin de pouvoir être utilisé aussi bien en appelant qu'en appelé :

```
E1      echo local active en mode commande
Q0      envoie les codes de retour
V1      envoie les reponses sous la forme de donnees textuelles
S0=0    decrochage automatique desactive (uugetty s'en occupe avec l'option WAITFOR)
```

Si ces options ne sont pas sélectionnées, il se peut que la chaîne d'initialisation définie dans le fichier de configuration, ne fonctionne pas.

```
&C1     le signal DCD est haut seulement apres la connexion
&S0     DSR est toujours haut
        un changement d'etat du DTR reinitialise le modem (dependant du fabricant - RTFM)
```

Ces réglages déterminent le comportement du modem au début et à la fin d'un appel.

Si votre modem ne permet pas la sauvegarde d'un profil, ces paramètres peuvent être indiqués dans la chaîne d'initialisation du fichier de configuration. Certains modems sont pourvus de commutateurs DIP qui affectent les valeurs des registres. Assurez-vous également qu'ils sont correctement configurés.

J'ai débuté une compilation de configurations pour différents modèles de modems. Pour l'instant, j'en ai seulement quelques uns. Ne vous privez donc pas de m'envoyer vos paramétrages opérationnels. Cette collection est récupérable à *ftp://ftp.cc.gatech.edu/pub/people/gregh/modem-configs* .

Remarque : pour que mon modem USR Courier V.34 se réinitialise correctement à l'ouverture du signal DTR, je suis obligé de forcer `&D2` et `S13=1` (le bit 0 du registre S13 est mis à 1). On m'a confirmé que ça fonctionnait aussi sur les modems USR Sportster V.34.

Remarque : certains modems Supra traitent le signal DCD différement. Dans ce cas, essayez avec `&C0` et *non* `&C1`. `&D2` doit également être positionné afin de gérer correctement le DTR.

6.2 Installer getty_ps

Récupérez la dernière version à partir de *sunsite.unc.edu:/pub/Linux/system/serial* . Les versions 2.0.7j et ultérieures sont indispensables pour les hauts débits (57600 and 115200 bps). Vous devez aussi avoir la libc 5.x ou supérieure.

Par défaut, `getty_ps` est conforme au FSSTND Linux (*File System Standard* - norme de système de fichiers), ce qui implique que les binaires résident dans `/sbin`, et les fichiers verrous (*lock files*) dans `/var/lock`. Assurez-vous donc que le répertoire `/var/lock` existe bien.

Si vous ne souhaitez pas la conformité FSSTND, les binaires seront stockés dans `/etc`, les fichiers de configuration dans `/etc/default/{uu}getty.ttyS`*N*, et les fichiers verrous dans `/usr/spool/uucp`. Je recommande cette méthode en cas d'utilisation conjointe avec UUCP afin que les fichiers verrous soient bien partagés entre `getty_ps` et UUCP.

`getty_ps` peut aussi utiliser `syslogd` pour enregistrer les messages. Reportez-vous aux pages de manuel `syslogd(1)` et `syslog.conf(5)` pour configurer `syslogd`, s'il ne tourne pas déjà. Les messages sont enregistrés avec la priorité LOG_AUTH, les erreurs avec LOG_ERR, et les informations de débogage avec LOG_DEBUG. Si vous ne souhaitez pas utiliser `syslogd`, éditez `tune.h` dans le répertoire des fichiers source de `getty_ps` pour enregistrer les messages dans un fichier, `/var/adm/getty.log` par défaut.

Vous devez donc décider si vous voulez la conformité FSSTND, les possibilités offertes par syslog, ou une combinaison des deux. Modifiez en conséquence les fichiers `Makefile`, `tune.h` et `config.h`. Il ne reste plus qu'à compiler et installer le paquetage, en suivant les instructions fournies.

À partir de maintenant, toutes les références à `getty` concerneront `getty_ps`, et celles à `uugetty` le programme `uugetty` fourni avec le paquetage `getty_ps`. Ces instructions ne sont pas applicables à `mgetty` ou `agetty`.

6.3 Configurer uugetty

`uugetty`, qui assure d'importantes vérifications sur les fichiers verrous, sera utilisé pour se connecter sur votre modem. Mettez à jour `/etc/gettydefs` pour inclure les différentes entrées pour les modems, si elles ne sont pas déjà présentes (remarquez que ces entrées sont rebouclées sur elles-mêmes et autorisent donc différentes vitesses - une ligne vide est nécessaire entre deux entrées) :

```
# Entrees modem
115200# B115200 CS8 # B115200 SANE -ISTRIP HUPCL #@S @L @B login: #57600

57600# B57600 CS8 # B57600 SANE -ISTRIP HUPCL #@S @L @B login: #38400

38400# B38400 CS8 # B38400 SANE -ISTRIP HUPCL #@S @L @B login: #19200

19200# B19200 CS8 # B19200 SANE -ISTRIP HUPCL #@S @L @B login: #9600

9600# B9600 CS8 # B9600 SANE -ISTRIP HUPCL #@S @L @B login: #2400

2400# B2400 CS8 # B2400 SANE -ISTRIP HUPCL #@S @L @B login: #115200
```

Si votre modem permet des vitesses de 9600 bps ou plus avec compression de données, vous pouvez fixer une vitesse constante pour le port série et laisser le modem choisir la vitesse de connexion. Le fichier `/etc/gettydefs` se simplifie et ne nécessite plus qu'une seule ligne pour le modem, par exemple :

```
# vitesse fixe 115200
F115200# B115200 CS8 # B115200 SANE -ISTRIP HUPCL #@S @L @B login: #F115200
```

Si le contrôle de flux RTS/CTS est activé sur votre modem, le drapeau `CRTSCTS` peut être ajouté dans les entrées :

```
# vitesse fixe 115200 avec controle de flux materiel
F115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login: #F115200
```

Si vous le désirez, `uugetty` peut imprimer quelques informations sur le système dans la bannière de login. Dans mes exemples, j'ai fait figurer le nom du système, le port série, ainsi que la vitesse courante en bps. D'autres champs sont disponibles :

```
@B    La vitesse courante en bps (evaluee lorsque le @B est affiche).
@D    La date courante (MM/DD/YY).
@L    Le port serie auquel getty est attache.
@S    Le nom du systeme.
@T    L'heure courante (HH:MM:SS / 24-heures).
@U    Le nombre d'utilisateurs actuellement connectes.  Il s'agit du
      nombre d'entrees dans le fichier /etc/utmp qui possedent un
      champ ut_name non nul.
@V    La valeur de VERSION, telle qu'elle est definie dans le fichier
      des valeurs par defaut.
Pour afficher un simple '@', utilisez soit '\@', soit '@@'.
```

Ensuite, assurez-vous de connaître un périphérique d'entrée et de sortie pour le port série dédié au modem. Si votre modem est attaché à `ttyS3`, aucun périphérique correct n'existe et vous devez donc le créer (voir la section 3.1.1 (Créer les périphériques dans <TT>/dev</TT>)). Si vous souhaitez pouvoir appeler avec votre modem alors que `uugetty` est en attente de connexion sur le même port, utilisez le périphérique `/dev/cua`*N* à la place de `/dev/ttyS`*N*.

Une fois que vous avez terminé l'édition de `/etc/gettydefs`, vous pouvez en vérifier la syntaxe par :

```
linux# getty -c /etc/gettydefs
```

6.4 Personnaliser `uugetty`

De nombreuses options peuvent être configurées dans des fichiers séparés pour chacun des ports présents. Le fichier `/etc/conf.uugetty` sert à *toutes* les instances de `uugetty`, alors que `/etc/conf.uugetty.ttyS`N est spécifique à un seul port. Des exemples de fichiers de configuration sont disponibles parmi les fichiers sources de `getty_ps` livrés avec la plupart des distributions Linux. Ces exemples ne sont pas repris dans ce document pour des problèmes d'espace. Notez que pour les versions anciennes de `getty` (avant la 2.0.7e), ou si vous n'utilisez pas la conformité FSSTND, le fichier par défaut réside dans `/etc/default/uugetty.ttyS`N. Mon fichier `/etc/conf.uugetty.ttyS3` ressemble à :

```
# exemple de fichier de configuration uugetty pour un modem compatible
# Hayes en mode appele
#
# fichier verrou alternatif... s'il existe, uugetty est relance afin de
# reinitialiser le modem
ALTLOCK=cua3
ALTLINE=cua3
# port serie a initialiser
INITLINE=cua3
# delai de liberation de la ligne en cas d'inactivite...
TIMEOUT=60
# chaine d'initialisation du modem...
# format : <expect> <send> ... (protocole de connexion)
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# delai d'attente avant l'envoi de la banniere
DELAY=1
#DEBUG=010
```

Ajoutez la ligne suivante dans `/etc/inittab`, afin que `uugetty` soit automatiquement lancé sur le port série (en adaptant pour votre configuration la localisation du fichier de configuration, le port, la vitesse, et le type de terminal par défaut) :

```
S3:456:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS3 ttyS3 F115200 vt100
```

Redémarrez `init` :

```
linux# init q
```

La vitesse maximale supportée par le modem a été sélectionnée dans `/etc/inittab`.

À partir de maintenant, Linux est en attente de connexions sur le port série. Appelez depuis une autre machine et loguez-vous sur votre système Linux.

`uugetty` possède de nombreuses autres options, décrites en détail dans la page de manuel `getty(1m)`. Entre autres choses, on peut y trouver des fonctionnalités de programmation horaire (*scheduling*), et de rappel (*ringback*).

7 Comment configurer un terminal raccordé à mon PC ?

Ces instructions permettent aussi bien de connecter des terminaux que d'autres ordinateurs sur le port série de votre machine Linux.

7.1 Besoins matériels

Assurez-vous de posséder le bon type de câble. Un câble *null modem* acheté dans n'importe quel magasin informatique fera l'affaire, mais il doit s'agir effectivement d'un câble *null modem* ! De nombreux magasins d'informatique les vendent comme des câbles pour imprimante série. Vérifiez que vous utilisez bien le port série (embase mâle DB25 ou DB9), et non le port parallèle (embase femelle DB25 ou Centronics).

Pour un connecteur DB25, il faut au minimum :

```
DB25 male du PC                         DB25 du terminal
TxD   Donnees emises        2 --> 3     RxD   Donnees recues
RxD   Donnees recues        3 <-- 2     TxD   Donnees emises
SG    Masse du signal       7 --- 7     SG    Masse du signal
```

Si vous voulez profiter du contrôle de flux matériel, la connexion null modem doit être intégralement câblée :

```
DB25 male du PC                         DB25 du terminal
TxD   Donnees emises        2 --> 3     RxD   Donnees recues
RxD   Donnees recues        3 <-- 2     TxD   Donnees emises
RTS   Demande pour emettre  4 --> 5     CTS   Pret a emettre
CTS   Pret a emettre        5 <-- 4     RTS   Demande pour emettre
DSR   Poste de donnees pret 6
                            |
DCD   Detection de porteuse 8 <-- 20    DTR   Terminal de donnees pret
SG    Masse du signal       7 --- 7     SG    Masse du signal
                                  6     DSR   Poste de donnees pret
                                  |
DTR   Terminal pret        20 --> 8     DCD   Detection de porteuse
```

Si vous avez un connecteur DB9, vous pouvez essayer :

```
DB9 du PC                               DB25 du terminal
RxD   Donnees recues        2 <-- 2     TxD   Donnees emises
TxD   Donnees emises        3 --> 3     RxD   Donnees recues
SG    Masse du signal       5 --- 7     SG    Masse du signal
```

Pour un câble complet DB9-DB25, on obtient :

```
DB9 du PC                               DB25 du terminal
RxD   Donnees recues        2 <-- 2     TxD   Donnees emises
TxD   Donnees emises        3 --> 3     RxD   Donnees recues
                                  6     DSR   Poste de donnees pret
                                  |
DTR   Terminal pret         4 --> 8     DCD   Detection de porteuse
SG    Masse du signal       5 --- 7     SG    Masse du signal
DCD   Detection de porteuse 1
                            |
DSR   Poste de donnees pret 6 <-- 20    DTR   Terminal de donnees pret
RTS   Demande pour emettre  7 --> 5     CTS   Pret a emettre
CTS   Pret a emettre        8 <-- 4     RTS   Demande pour emettre
(RI   Indicateur sonnerie   9 pas necessaire)
```

(Eh oui, les broches 2 et 3 ont *effectivement* la signification opposée entre les connecteurs DB9 et DB25 !)

Ceux qui ne possèdent pas de câble null modem complet devront user d'une astuce afin que le PC reçoive (de lui-même) les acquittements qu'il demande avec le contrôle de flux matériel : sur le connecteur côté PC, rebouclez ensemble RTS et CTS, ainsi que DSR, DCD et DTR.

Maintenant que avez le bon type de câble, il vous reste à connecter votre terminal sur votre ordinateur. Si possible, demandez au terminal d'ignorer les signaux de contrôle pour modems, et essayez de le paramétrer avec une vitesse de 9600 bps, 8 bits de données, 1 bit de stop, et pas de parité.

7.2 Configurer getty

Si elles ne sont pas déjà présentes, ajoutez dans `/etc/gettydefs` les entrées destinées à `getty` pour votre terminal,

```
# Entree de terminal simple a 38400 bps
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

# Entree de terminal simple a 19200 bps
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# Entree de terminal simple a 9600 bps
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

Si vous le désirez, `getty` peut imprimer quelques informations sur le système dans la bannière de login. Dans mes exemples, j'ai fait figurer le nom du système ainsi que le port série. D'autres variables sont disponibles :

```
@B    La vitesse courante en bps (evaluee lorsque le @B est affiche).
@D    La date courante (MM/DD/YY).
@L    Le port serie auquel getty est attache.
@S    Le nom du systeme.
@T    L'heure courante (HH:MM:SS / 24-heures).
@U    Le nombre d'utilisateurs actuellement connectes.  Il s'agit du
      nombre d'entrees dans le fichier /etc/utmp qui possedent un
      champ ut_name non nul.
@V    La valeur de VERSION, telle qu'elle est definie dans le fichier
      des valeurs par defaut.
Pour afficher un simple '@', utilisez soit '\@', soit '@@'.
```

Une fois l'édition de `/etc/gettydefs` terminée, vous pouvez en vérifier la syntaxe par :

```
linux# getty -c /etc/gettydefs
```

Assurez-vous qu'il n'existe pas déjà de fichier de configuration `getty` ou `uugetty` pour le port série auquel vous avez attaché votre terminal (`/etc/default/`{uu}`getty.ttyS`*N* ou `/etc/conf.`{uu}`getty.ttyS`*N*), car il risque probablement d'interférer avec l'instance de `getty` du terminal. Supprimez le fichier s'il existe.

Éditez la ligne suivante dans `/etc/inittab`, afin que `getty` soit lancé sur le port série (en adaptant pour votre configuration le port, la vitesse, et le type de terminal par défaut) :

```
S1:456:respawn:/sbin/getty ttyS1 DT9600 vt100
```

Redémarrez `init` :

```
linux# init q
```

Vous devez maintenant voir apparaître une invite de session sur le terminal. Il peut néanmoins être nécessaire de taper sur la touche `Entrée` pour avertir le terminal.

7.3 Remarques sur la configuration d'un PC en tant que terminal

Nombreux sont ceux qui se servent d'autres PC comme des terminaux rattachés à des systèmes Linux. Par exemple, des vieux PC 8088 ou 80286 sont parfaits pour cet usage. Vous avez juste besoin d'une disquette de démarrage DOS et d'un programme de communications pour faire tourner votre terminal-PC. `kermit` fonctionne très bien dans cette configuration. Des versions compilées de `kermit` pour n'importe quel système d'exploitation sont disponibles sur *http://www.columbia.edu/kermit/* . D'autres programmes populaires de communications sous DOS, comme `telix` ou `procomm` remplissent également très bien cette fonction. Soyez juste sûr de fournir des informations correctes lors du paramétrage des communications de votre terminal-PC.

8 Puis-je utiliser plus de deux ports série ?

Il n'est pas nécessaire que vous lisiez cette section, à moins que vous utilisiez au moins trois ports série... (mais pas sur une carte multiports).

N'importe quel port série libre est utilisable. La seule limitation réside dans le nombre d'interruptions (IRQ) et d'adresses de port d'entrée/sortie à utiliser. Cette limitation n'est pas spécifique à Linux, mais est due au bus du PC. Chaque périphérique série (port série, modem interne, carte série) doit posséder ses propres interruption et adresse.

Les cartes série multiports sont spécialement conçues pour partager la même interruption sur tous les ports qu'elles possèdent. Linux communique avec elles en sélectionnant une adresse d'entrée/sortie différente pour chaque port de ces cartes.

8.1 Choisir les interruptions des périphériques série

Par défaut sur un PC, l'IRQ 4 est partagée entre `ttyS0` et `ttyS2`, et l'IRQ 3 entre `ttyS1` et `ttyS3`. Les interruptions en cours d'utilisation sont regroupées dans `/proc/interrupts`. Pour se servir de plus de deux ports série, leur interruption doit être réassignée, par exemple en choisissant celle d'un port parallèle. En effet, les IRQ 5 et 7 sont sur un PC normalement attribuées aux ports parallèles, mais peu de personnes en possèdent deux. Vous pouvez donc réassigner une de ces interruptions à un port série, et continuer à utiliser joyeusement votre port parallèle ! Cette manipulation requiert le programme `setserial`. De plus, il vous faudra ajuster certains cavaliers sur vos cartes (à vérifier sur les manuels des cartes) pour fixer les IRQ.

Vous devez vous arranger pour avoir une IRQ unique pour chacun des ports série. Voici comment je configure les miens dans `/etc/rc.d/rc.local` (cette initialisation doit avoir lieu lors du démarrage) :

```
/sbin/setserial /dev/ttyS0 irq 3        # ma souris
/sbin/setserial /dev/ttyS1 irq 4        # mon terminal Wyse
/sbin/setserial /dev/ttyS2 irq 5        # mon modem Zoom
/sbin/setserial /dev/ttyS3 irq 9        # mon modem USR
```

Assignations standards des IRQ :

```
IRQ  0    Base de temps 0
IRQ  1    Clavier
IRQ  2    Cascade pour le deuxieme controleur d'IRQ
```

```
IRQ  3    Port serie 2
IRQ  4    Port serie 1
IRQ  5    Port parallele 2
IRQ  6    Disquette
IRQ  7    Port parallele 1
IRQ  8    Horloge temps-reel (RTC)
IRQ  9    Redirige vers l'IRQ2
IRQ 10    Non assignee
IRQ 11    Non assignee
IRQ 12    Non assignee
IRQ 13    Coprocesseur mathematique
IRQ 14    Controleur de disques durs 1
IRQ 15    Controleur de disques durs 2
```

Il n'existe pas vraiment de Bonne Technique pour choisir les interruptions. Il faut juste s'assurer qu'elles ne sont pas déjà utilisées par la carte mère, ou une autre carte. 2, 3, 4, 5, ou 7 est généralement un bon choix. “Non assignée” signifie que rien de standard n'utilise cette IRQ. Notez également que l'IRQ2 et l'IRQ9 sont identiques : vous pouvez donc spécifier aussi bien 2 que 9, le gestionnaire de périphériques est très compréhensif. Si vous possédez une carte série avec un connecteur de bus 16 bits, vous pouvez également utiliser l'IRQ 10, 11, 12 ou 15.

N'utilisez pas les IRQ 0, 1, 6, 8, 13 et 14 ! Elles sont réservées par la carte mère. Une fois le paramétrage effectué, vérifiez l'absence de conflit avec `/proc/interrupts`.

8.2 Choisir les adresses des périphériques série

Vous devez ensuite fixer les adresses des ports. Vérifiez le manuel pour la position des cavaliers. Comme pour les interruptions, chaque périphérique série doit posséder sa propre adresse. Par défaut, les ports sont paramétrés de la manière suivante :

```
ttyS0 adresse 0x3f8
ttyS1 adresse 0x2f8
ttyS2 adresse 0x3e8
ttyS3 adresse 0x2e8
```

Choisissez les adresses à affecter à chaque port série, et positionnez les cavaliers sur la carte en conséquence. Ainsi, j'ai mon modem sur `ttyS3`, ma souris sur `ttyS0`, et mon terminal sur `ttyS2`.

Au prochain démarrage, Linux devrait voir vos ports séries aux adresses que vous avez sélectionnées. Par contre, les IRQ vues par Linux peuvent ne pas correspondre avec celles que vous avez configurées. Ne vous inquiétez pas : Linux ne sait pas détecter les IRQ au démarrage, car c'est assez aléatoire et le résultat peut être faux. Utilisez `setserial` pour spécifier à Linux quelles interruptions sont associées aux ports, puis vérifiez après redémarrage de Linux les adresses des ports d'entrée/sortie avec `/proc/ioports`.

9 Comment configurer des vitesses plus importantes pour mes ports série ? À quelle vitesse dois-je fixer mon modem ?

Cette section a pour but de vous aider à déterminer la vitesse du modem à utiliser avec un programme de communications ou `getty`.

- Si votre modem ne supporte que des vitesses inférieures à 9600 bps (V.32), choisissez la vitesse la plus importante, par exemple 300 bps (V.21/Bell 103), 1200 bps (V.22/Bell 212A), ou 2400 bps (V.22bis).
- Si vous possédez un modem 9600 bps (V.32), intégrant la compression de données V.42bis, choisissez 38400 bps. Le taux *théorique* de compression est de 4:1, soit “4 * 9600 = 38400”.
- Si vous possédez un modem 14400 bps (V.32bis), intégrant la compression de données V.42bis, configurer le port à 57600 bps (4 * 14400 = 57600) avec l'option `spd_hi` de `setserial`. Pour les modems 28800 ou 33600 bps (V.FC ou V.34), utiliser l'option `spd_vhi` (4 * 28800 = 115200).

  Enfin, fixez la vitesse à 38400 bps dans votre programme de communications ou `/etc/inittab`. Vous êtes maintenant en grande vitesse ! Assurez-vous de posséder des UART 16550A.

  Il existe des vitesses nommées 57600 et 115200 dans les versions de `libc` supérieures ou égales à 5.x (à vérifier dans le répertoire `/lib`). Dans ce cas, vous pouvez les sélectionner directement (sans passer par `setserial`) si vos applications ont été compilées de manière à en tirer avantage. Il existe tellement de distributions Linux, que le mieux à faire est d'essayer directement ces vitesses si votre distribution est assez récente.

Avant de fixer la configuration de `setserial` dans `/etc/rc.d/rc.serial` ou `/etc/rc.d/rc.local` afin qu'elle soit prise en compte au démarrage, il est préférable de l'essayer manuellement. Par exemple, pour fixer `ttyS3` à 115200 bps, mon fichier `/etc/rc.d/rc.local` contient :

```
/sbin/setserial /dev/ttyS3 spd_vhi
```

Assurez-vous que le chemin d'accès à `setserial` est valide, de même que le nom du périphérique. Le paramétrage d'un port série peut être vérifié par :

```
setserial -a /dev/ttyS3
```

10 Programmes et utilitaires de communications

Voici une liste de logiciels de communications que vous pouvez télécharger par FTP, s'ils ne sont pas fournis dans votre distribution :

- `ecu` - un programme de communications
- *C-Kermit* `http://www.columbia.edu/kermit/` - programme de communications portable et scriptable, par liaison série ou TCP/IP, incluant le transfert de fichiers et la traduction des jeux de caractères
- `minicom` - un programme de communications du genre de `telix`
- `pcomm` - un programme de communications du genre de `procomm` avec protocole zmodem
- `seyon` - un programme de communications sous X11
- `xc` - paquetage de communications xcomm
- D'autres programmes utiles sont `term` et `SLiRP` qui offrent des fonctionnalités TCP/IP à partir d'un compte shell.
- `screen` est un autre programme multi-sessions qui se comporte comme les consoles virtuelles.
- `callback` est un programme permettant à votre modem de se faire rappeler.
- `mgetty+fax` gère le mode FAX, et offre un `getty` alternatif.

- `ZyXEL` est un programme de contrôle pour les modems ZyXEL U-1496. Il gère les appels entrants et sortants, la sécurité pour le rappel automatique, le mode FAX, et possède des fonctions de messagerie vocale.
- SLIP et PPP peuvent être téléchargés depuis *ftp://sunsite.unc.edu/pub/Linux/system/network/serial* .
- On trouve également d'autres programmes pour les liaisons série sur *ftp://sunsite.unc.edu/pub/Linux/system/serial* et *ftp://sunsite.unc.edu/pub/Linux/apps/serialcomm* ou sur l'un des nombreux sites miroirs.

11 Trucs et astuces sur les liaisons série

Je vous présente ici quelques astuces que vous pourrez trouver utiles...

11.1 `kermit` et zmodem

Pour utiliser le protocole zmodem avec `kermit`, ajoutez les lignes suivantes dans votre `.kermrc`:

```
define rz !rz < /dev/ttyS3 > /dev/ttyS3
define sz !sz \%0 > /dev/ttyS3 < /dev/ttyS3
```

Soyez sûr de positionner le port série auquel votre modem est rattaché. Il suffit alors de taper `rz` ou `sz` <`filename`> à l'invite de commandes de `kermit`.

11.2 Paramétrer les types de terminaux automagiquement

Pour sélectionner automagiquement le type de terminal au début d'une session, ajoutez-le dans l'entrée correspondante de `/etc/inittab`. Par exemple, avec un terminal VT100 sur `ttyS1`, j'ajouterais "vt100" en paramètre de la commande `getty` :

```
S1:456:respawn:/sbin/getty ttyS1 DT9600 vt100
```

Vous pouvez également utiliser `tset` pour positionner les caractéristiques du terminal à l'entrée en session, indépendamment des valeurs par défaut.

11.3 `ls` en couleurs sur les connexions série

Si l'émulation de votre terminal ne supporte pas correctement l'option couleur de `ls`, désactivez-la. Certaines installations l'utilisent par défaut (`ls -color` ou `ls -colour`). Vérifiez les alias sur `ls` dans `/etc/profile` et `/etc/csh.cshrc`. Vous pouvez également créer un nouvel alias de `ls` sur `ls -no-color`, si vous ne souhaitez pas modifier les valeurs par défaut.

11.4 Imprimer sur une imprimante attachée à un terminal

Il existe pour cela un programme appelé `vtprint`, disponible sur *ftp://ftp.sdsu.edu/pub/vtprint* , et sur *http://www.sdsu.edu/~garrett* .

Un deuxième programme est `xprt`. Il peut être téléchargé depuis *ftp://sunsite.unc.edu/pub/Linux/system/printing* .

11.5 Linux peut-il configurer les périphériques série automagiquement ?

Effectivement, Linux peut détecter et paramétrer les périphériques série automatiquement au démarrage en ajoutant par exemple la ligne :

```
/sbin/setserial /dev/ttyS3 auto_irq skip_test autoconfig
```

dans votre fichier `/etc/rc.d/rc.local` ou `/etc/rc.d/rc.serial`. Répétez la manipulation pour chacun des ports série qui doit être configuré automatiquement. Soyez juste sûr de fournir un nom de périphérique existant sur votre machine.

11.5.1 Remarques pour les cartes multiports

Référez-vous au fichier `rc.serial` fourni avec `setserial` pour les adresses et les IRQ de ces cartes. Beaucoup de paramètres y sont détaillés concernant les cartes multiports, y compris les adresses et les noms des périphériques à utiliser.

11.6 Utiliser une console série

Un article dans le numéro 36 du *Linux Journal* (*http://www.ssc.com/lj/issue36/index.html*) explique comment utiliser un terminal comme une console. Malheureusement, l'adresse électronique de l'auteur ne semble pas correcte car les messages reviennent. J'espère que cet article figurera bientôt sur le site WEB du *Linux Journal* à l'URL mentionnée ci-dessus. *(NdT : cet article est maintenant disponible en ligne à* `http://www.ssc.com/lj/issue36/console.html` *)*

11.7 Obtenir des débits plus importants

Si vous jugez le débit de vos ports série lent, ou si des erreurs d'écrasement de caractères à la réception (*overrun errors*) se produisent sur un système muni de disques durs (E)IDE, utilisez `hdparm` pour modifier certains des paramètres de ces disques, notamment le démasquage des autres IRQ pendant les interruptions disque. Cette manipulation augmentera le taux de réponse et aidera à supprimer les écrasements de caractères. Lisez bien les pages de manuel, car certaines combinaisons sont plus ou moins bien supportées suivant les disques et les contrôleurs utilisés, et peuvent provoquer une corruption du système de fichiers.

Regardez également l'utilitaire `irqtune` qui permet de modifier la priorité de l'IRQ d'un périphérique, par exemple celle du port série lié au modem. La FAQ d'`irqtune` est disponible sur *http://www.best.com/~cae/irqtune* .

12 Un pas de plus...

Cette section n'est absolument pas nécessaire à la compréhension des ports série sous Linux, mais peut néanmoins vous apporter quelques éclaircissements sur Unix et le monde des télécommunications.

12.1 Qu'est-ce que les fichiers verrous ?

Un fichier verrou est simplement un fichier qui indique qu'un périphérique particulier est en cours d'utilisation. Ils sont regroupés dans `/usr/spool/uucp` ou `/var/lock`. Sous Linux, ils se nomment `LCK..`*nom* où *nom* est soit un nom de périphérique, soit un nom de site UUCP. Certains processus créent de tels fichiers pour s'assurer l'accès exclusif aux périphériques. Par exemple, si vous appelez avec votre modem, un fichier verrou va être créé pour avertir les autres processus que le modem est déjà en cours d'utilisation. Les verrous contiennent principalement le PID du processus qui s'est attribué le périphérique. La plupart des programmes regardent dans un premier temps si un fichier verrou existe, puis s'il est toujours valide en recherchant dans la table des processus le programme qui a verrouillé le périphérique. Si le fichier verrou est toujours valide, alors le nouveau programme doit s'arrêter. Dans le cas contraire, certains programmes suppriment le fichier verrou périmé, et utilisent le périphérique en créant leur propre verrou à l'intérieur du processus. D'autres programmes s'arrêtent tout simplement et préviennent l'utilisateur que le périphérique est déjà utilisé.

12.2 “baud” et “bps”

“baud” et “bps” font certainement partie des termes les plus mal utilisés dans le monde de l'informatique et des télécommunications. Ils sont souvent considérés à tort comme équivalents, alors qu'ils ne le sont pas !

baud

Le baud représente la fréquence de (dé)modulation d'un signal, par exemple celui envoyé ou reçu par un modem (**mo**dulateur-**dém**odulateur), c'est-à-dire le nombre de fois où il change par seconde. Par exemple, 1200 bauds implique que le signal change d'état toutes les 833 microsecondes. Les fréquences de (dé)modulation les plus courantes pour un modem sont 50, 75, 110, 300, 600, 1200, et 2400 bauds. La plupart des modems hautes vitesses fonctionnent à 2400 bauds. Les fréquences plus élevées sont plus difficiles à atteindre. Cette limitation est principalement due aux basses performances des lignes téléphoniques dédiées essentiellement au transport de la voix. Plusieurs bits sont alors encodés par baud, ce qui permet d'obtenir des taux de transfert en bits plus élevés qu'en bauds. L'étymologie du mot “baud” vient de l'inventeur de l'imprimante télégraphique asynchrone, Émile Baudot.

bps

Le taux de transfert en bps représente le nombre de bits transmis en une seconde. Les vitesses en bps les plus courantes sont 50, 75, 110, 300, 1200, 2400, 9600, ... 115200. Les modems avec compression V.42bis (rapport maximal de 4 à 1) activée sont capables *thérориquement* d'atteindre des vitesses de 115200 bps. C'est l'usage mauvais le plus courant du terme “baud”.

Donc, si les modems hautes vitesses fonctionnent à 2400 bauds, comment peuvent-ils émettre à 14400 bps ? En fait, les modems atteignent des taux de transfert en bps supérieurs à la fréquence de modulation, en codant plusieurs bits dans un seul changement d'amplitude ou de phase du signal. Ainsi, quand 2 bits ou plus sont codés par baud, la vitesse en bps dépasse la fréquence en bauds. Quand votre modem se connecte à 14400 bps, il envoie effectivement 6 bits à chaque changement de phase, à 2400 bauds.

Cette confusion entre bauds et bps date du temps où les taux de transfert étaient faibles, et où la vitesse en bps égalait la fréquence de modulation en bauds. Par exemple, un modem à 300 bps fonctionne à une fréquence de 300 bauds. Les deux valeurs ont commencé à diverger avec l'apparition des modems hautes vitesses.

12.3 Qu'est-ce que les UART ? Comment influent-ils sur les performances ?

Les UART (***U****niversal* ***A****syncronous* ***R****eceiver* ***T****ransmitter* - Émetteur Récepteur Universel Asynchrone) sont des circuits électroniques équipant les cartes série des PC. Ils servent à transformer les données en bits, à les envoyer sur la ligne série, puis à reconstruire ces données à l'autre bout. Ils travaillent sur des octets, qui correspondent justement à la taille des caractères ASCII.

Supposons qu'un terminal est connecté à votre PC. Quand vous tapez un caractère, le terminal l'envoie à un émetteur (qui est aussi un UART). L'émetteur envoie alors cet octet sur la ligne série, bit par bit, et à une vitesse spécifique. Du côté PC, l'UART réceptionne ces bits, reconstruit l'octet, et enfin le place dans une zone mémoire.

Il existe essentiellement deux types d'UART : les simples (8250 et 16450), et ceux dotés de FIFO (16550A). Pour comprendre les différences qui existent entre ces circuits, il faut examiner ce qui se passe quand un UART envoie ou reçoit un octet.

L'UART lui-même ne fait rien avec les données ; il s'occupe juste de les émettre ou les réceptionner. Le CPU reçoit une interruption du périphérique série chaque fois qu'un octet part ou arrive. Le CPU déplace alors l'octet reçu du périphérique série vers une zone mémoire, ou alors il redonne à l'UART un nouvel octet à transmettre. Les UART 8250 et 16450 ne possèdent qu'une zone tampon d'un octet, ce qui signifie que le CPU est interrompu à chaque caractère. Ça fonctionne aux vitesses faibles, mais pour des vitesses élevées, le CPU est tellement occupé par l'UART, qu'il n'a plus assez de temps pour les autres tâches. Dans certains cas, le CPU ne peut pas répondre à l'interruption assez vite, et les caractères sont alors écrasés (*overrun error*).

Les UART 16550 sont alors utiles car ils sont équipés de tampons FIFO de 16 octets, c'est-à-dire qu'ils peuvent émettre ou recevoir jusqu'à 16 octets avant d'interrompre le CPU. Bien que ce seuil d'interruption soit rarement fixé à 16, ça représente quand même un avantage significatif par rapport aux UART dotés d'un tampon d'un seul octet. Le CPU est alors interrompu moins souvent et peut consacrer plus de temps aux autres tâches. Les données ne sont donc plus perdues. (Il existe également un UART 16550, considéré comme un 16450 car il est défectueux.)

En général, les 8250 et 16450 peuvent suffire pour des vitesses n'excédant pas 38400 bps. À des vitesses supérieures, vous pouvez voir apparaître des pertes de caractères. D'autres systèmes d'exploitation pour PC, comme DOS (définition approximative dans ce cas), qui ne sont pas multitâches, peuvent à la rigueur mieux se débrouiller avec des 8250 ou des 16450. C'est pour cela que certaines personnes ne constatent aucune perte de données, jusqu'au moment où elles passent à Linux.

Les cartes multiports intelligentes utilisent des DSP (*Digital Signal Processors* - processeurs de signaux digitaux) pour assurer des fonctions supplémentaires de contrôle et de mises en mémoire tampon, ce qui permet de décharger encore plus le CPU. Par exemple, les cartes Cyclades Cyclom et Stallion EasyIO possèdent un UART RISC Cirrus Logic CD1400. La plupart des autres cartes sont dotées de CPU 80186, voire de CPU RISC spécifiques, pour assurer les entrées/sorties série.

Gardez à l'esprit que ces UART ne sont pas mauvais, mais qu'ils ne suffisent pas pour des vitesses élevées. Vous ne devriez pas rencontrer de problème en connectant un terminal ou une souris sur de tels UART. Par contre, les 16550A sont définitivement un *must* pour les modems hautes vitesses.

Les cartes à base d'UART 16550A sont à peine plus chères que celles à base de 16450. De même, il est possible de remplacer les UART 16450 existants sur une carte par des 16550A. Ces circuits sont compatibles broche à broche. Certaines cartes sont même équipées à cet effet d'UART montés sur support. Sinon, il faudra souder. Remarquez que vous vous éviterez sûrement pas mal de soucis en achetant directement une nouvelle carte.

13 Résoudre les problèmes

13.1 Le message "line *NNN* of inittab invalid" s'affiche en permanence

Assurez-vous d'utiliser une syntaxe correcte pour votre version de `init`. En effet, chaque programme `init` ou presque a une syntaxe spécifique pour son fichier de configuration `/etc/inittab`. De même, vérifiez les paramètres que vous passez à votre `getty`.

13.2 Le message “/dev/cua$N$: Device or resource busy” apparaît quand j’essaie d’appeler

Ce problème peut survenir quand les signaux DCD ou DTR ne sont pas positionnés correctement. DCD doit être levé seulement lors d’une connexion (*ie* quelqu’un est déjà connecté), et non pas quand `getty` scrute le port. Assurez-vous donc que le modem est configuré pour lever le signal DCD seulement en connexion. DTR doit être positionné dès qu’un processus utilise ou scrute la ligne, par exemple `getty`, `kermit`, ou n’importe quel autre programme de communications.

Une autre cause possible est que l’IRQ assignée au port série est déjà prise par un autre périphérique. En effet, lors de son initialisation, chaque périphérique demande l’autorisation à Linux d’utiliser l’IRQ sélectionnée. Linux garde une trace de l’affectation des interruptions, et si une IRQ est en cours d’utilisation, votre périphérique ne pourra pas s’initialiser. Celui-ci n’a aucun moyen de vous prévenir, excepté par le message “device-busy” lorsque vous tentez de l’utilisez. Vérifiez alors les interruptions de toutes vos cartes (série, ethernet, SCSI, etc.) et les conflits éventuels.

13.3 Le message “Id SN respawning too fast: disabled for 5 minutes” s’affiche périodiquement

Assurez-vous que votre modem est correctement configuré. Examinez particulièrement les registres `E` et `Q`. Ce problème peut apparaître lorque `getty` discute avec le modem.

Vérifiez également les paramètres que vous passez à `getty` dans `/etc/inittab`. Une syntaxe ou un nom de périphérique erroné peut causer de sérieux problèmes.

La syntaxe de `/etc/gettydefs` peut être vérifiée par la commande suivante :

```
linux# getty -c /etc/gettydefs
```

Ce problème arrive quelquefois lors de l’échec de l’initialisation de `uugetty`. Reportez-vous à la section 13.11 (getty ou uugetty ne fonctionne toujours pas).

13.4 Mes périphériques série sont lents ou unidirectionnels

La cause la plus probable est un conflit d’IRQ. Assurez-vous qu’aucune IRQ n’est partagée. Vérifiez les cavaliers sur les différentes cartes (série, ethernet, SCSI, etc.) ainsi que les paramètres passés à `setserial` pour tous les périphériques série. Les conflits peuvent être localisés avec `/proc/ioports` et `/proc/interrupts`.

13.5 Mon modem ne répond plus après le raccrochage, ou `uugetty` ne se relance plus automatiquement

Cela peut se produire si le modem n’est pas réinitialisé lorque le signal DTR retombe. J’ai vu les LED RD et SD devenir folles quand ça m’est arrivé. Il faut alors réinitialiser le modem. La plupart des modems compatibles Hayes ont besoin de la commande `&D3`, mais sur mon USR Courier, je dois positionner `&D2` et `S13=1`. Vérifiez dans la documentation de votre modem.

13.6 Un terminal est connecté à mon PC, mais il se bloque dès que j’ai entré mon nom d’utilisateur

- Si vous utilisez `getty`, vous devez faire figurer `CLOCAL` dans l’entrée correspondante de `/etc/gettydefs`, et utiliser un câble null-modem complet. L’option `CLOCAL` indique à Linux d’ignorer les signaux de contrôle spécifiques aux modems :

```
# Entree de terminal simple a 38400 bps
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

# Entree de terminal simple a 19200 bps
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# Entree de terminal simple a 9600 bps
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

Ensuite, tuez (`kill`) le processus `getty` afin qu'une nouvelle instance soit lancée avec les entrées mises à jour.

- Si vous utilisez `agetty`, ajoutez l'option `-L` à la ligne correspondante de `/etc/inittab`, pour indiquer que vous désirez ignorer les signaux de contrôle spécifiques aux modems. Redémarrez `init` en tapant `init q`. L'entrée doit ressembler à :

```
s1:345:respawn:/sbin/agetty -L 9600 ttyS1 vt100
```

13.7 Mon modem perd des données à haute vitesse

Si vous essayez de faire fonctionner votre modem à plus de 38400 bps, vous devez obligatoirement passer en UART 16550A. Reportez-vous à la section 12.3 (Qu'est-ce que les UART ?).

13.8 Au démarrage, Linux n'affiche pas les ports série tels que je les ai paramétrés

Effectivement, Linux ne cherche pas à détecter les IRQ au démarrage, mais seulement les ports série. Il suppose que vous utilisez les interruptions par défaut, car leur détection est hasardeuse et peut se révéler inexacte.

Ainsi, même si j'ai forcé `ttyS2` à l'IRQ 5, je vois toujours :

```
Jan 23 22:25:28 misfits vmunix: tty02 at 0x03e8 (irq = 4) is a 16550A
```

quand Linux se lance. Il faut alors utiliser `setserial` pour indiquer les IRQ à Linux. Après le démarrage, vous pouvez vérifier le paramétrage effectif dans le fichier `/proc/interrupts`.

13.9 `rz` et/ou `sz` ne fonctionne pas quand j'appelle mon système Linux avec un modem

Si Linux recherche le périphérique `/dev/modem` quand vous tentez de transférer des fichiers, regardez les alias définis dans `/etc/profile` et `/etc/csh.cshrc`. Ils peuvent être nombreux suivant les distributions (notamment Slackware) et redéfinir les programmes zmodem. Enlevez ces alias, ou corrigez-les.

13.10 Mon écran imprime des caractères bizarres

Ce phénomène se produit sur les consoles virtuelles, et parfois sur les lignes série, quand elles reçoivent des données binaires. Il faut alors taper `echo ^v^[c`, c'est-à-dire :

```
linux% echo <ctrl>v<esc>c
```

13.11 getty ou uugetty ne fonctionne toujours pas

`getty_ps` fournit une option `DEBUG` que l'on peut spécifier dans le fichier de configuration `/etc/conf.{uu}getty.ttyS`*N*. Éditez-le pour ajouter la ligne `DEBUG=`*NNN* où *NNN* est une combinaison de valeurs octales définissant les informations que vous voulez obtenir :

```
D_OPT   001          configuration des options
D_DEF   002          traitement du fichier des valeurs par defaut
D_UTMP  004          traitement de utmp/wtmp
D_INIT  010          initialisation de la ligne (INIT)
D_GTAB  020          traitement du fichier gettytab
D_RUN   040          autres diagnostics lors de l'execution
D_RB    100          traitement du mode de rappel (ringback)
D_LOCK  200          traitement des fichiers verrou pour uugetty
D_SCH   400          traitement de la programmation horaire (schedule)
D_ALL   777          tout
```

Positionner `DEBUG=010` est un bon point de départ.

Si `syslogd` tourne, ces informations apparaîtront dans les fichiers log. Dans le cas contraire, elles seront enregistrées dans `/tmp/getty:ttyS`*N* pour `getty`, dans `/tmp/uugetty:ttyS`*N* pour `uugetty`, et dans `/var/adm/getty.log`. Consultez ces fichiers pour déterminer ce qui se passe. Vous devrez très probablement ajuster certains paramètres dans le fichier de configuration, et reconfigurer votre modem.

Vous pouvez également essayer `mgetty` : certaines personnes ont plus de chance avec...

14 Autres sources d'information

- les pages de manuel pour : `agetty(8)`, `getty(1m)`, `gettydefs(5)`, `init(1)`, `login(1)`, `mgetty(8)`, `setserial(8)`
- la documentation de votre modem
- NET-3 HOWTO : tout sur les réseaux, dont SLIP, CSLIP, et PPP
- PPP HOWTO : aide sur PPP
- Printing HOWTO : pour installer une imprimante série
- Term HOWTO : tout ce que vous voulez savoir sur le programme `term`
- UPS HOWTO : pour la connexion d'onduleurs sur un port série
- UUCP HOWTO : pour des informations sur le paramétrage de UUCP
- forums de discussion Usenet :

```
comp.os.linux.answers
        FAQ, How-To, README, etc. a propos de Linux.
comp.os.linux.hardware
        compatibilite materielle avec le systeme d'exploitation Linux.
comp.os.linux.networking
        reseaux et communications sous Linux.
comp.os.linux.setup
        installation et administration systeme sous Linux.
fr.comp.os.linux
        forum de discussion sur Linux en francais
```

- la liste de diffusion consacrée aux ports série sous Linux. Pour vous abonnez, envoyez un message électronique à *majordomo@vger.rutgers.edu* , avec “`subscribe linux-serial`” dans le corps du message. La commande “`help`” dans le corps du message vous permet de recevoir une message d’aide sur les commandes supportées. Ce serveur gère également d’autres listes consacrées à Linux ; pour les connaître, envoyez la commande “`lists`”.
- un document traitant des communications série et des cartes multiports est disponible sur : *http://www.cyclades.com* .
- FAQ sur les modems :
  Navas 28800 Modem FAQ `http://web.aimnet.com/~jnavas/modem/faq.html`
  Curt’s High Speed Modem Page `http://www.teleport.com/~curt/modems.html`
- programmation série :

  Advanced Programming in the UNIX Environment `http://heg-school.aw.com/cseng/authors/stevens/advanced/advanced.nclk` , par W. Richard Stevens (ISBN 0-201-56317-7; Addison-Wesley)

  POSIX Programmer’s Guide `http://www.ora.com/catalog/posix/` , by Donald Lewine (ISBN 0-937175-73-0; O’Reilly)

15 Contributions

Il ne m’aurait pas été possible de rédiger ce HOWTO tout seul. Même si j’en ai écrit la plus grande partie, j’ai dû reprendre pas mal d’articles existants. Merci à tous ceux qui ont contribué à ce document, ou qui l’ont commenté. La liste de ces personnes est maintenant trop longue pour figurer ici (plus d’une centaine). J’adresse plus particulièrement mes remerciements à Ted T’so pour toutes ses réponses sur les périphériques série, à Kris Gleason pour la maintenance de `getty_ps`, et à Gert Döring pour `mgetty`.

Note du traducteur : un grand merci à Olivier Tharan

et Florian Salamin *pour la relecture, ainsi qu’à* Eric Dumas *pour la gestion des traductions françaises et à* Sébastien Blondeel *pour la mise en place d’un système centralisé des relectures.*

FIN DU LINUX HOWTO POUR LES PORTS SÉRIE