Linux Serial HOWTO 中譯版

作者: Greg Hankins *greg.hankins@cc.gatech.edu*
譯者: *Asd L. Chen* & *C.W.Huang*
v1.11, 15 November 1997 翻譯日期: 18 November 1997

這份文件敘述如何在 Linux 機器上設定串列通訊設備.

目錄

1 簡介

這是 Linux Serial HOWTO.包含關於如何在 Linux 下設定數據機以及終端機的所有細節,還有一些使用串列的訣竅,以及故障排除的資料. 這份文件著重在 Intel x86 硬體上執行的 Linux,雖然在其它平台上也許也能運作.

1.1 版權聲明

1.2 這份文件的最新版本

新版的 Serial HOWTO 會放到
ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/Serial-HOWTO 以及映射節點．有一些其它像是 Postscript 以及 DVI 格式的版本放在 `other-formats` 目錄裡． 這一份 Serial HOWTO 也可在 *http://sunsite.unc.edu/LDP/HOWTO/Serial-HOWTO.html* 取得．它也會每月定期的貼到 *comp.os.linux.answers*．

1.3 回饋

請將任何的問題，評論，建議或補充內容寄送給我．我永遠渴望聽到你對這份 HOWTO 的想法． 我也總是小心地注意改進! 確實地告訴我你有什麼不懂的地方，或者是有什麼地方可以寫得更清楚． 你可以經由這個電子郵件位址 *greg.hankins@cc.gatech.edu* 與我連繫． 所有的郵件我都會回覆，不過可能要花一個星期或更長的時間，看我多忙而定． 我收到一大堆來自 `root` 及錯誤設定的站台． 請以眞實身份寄信給我，並且確定你的電子郵件系統是正常運作使得我能夠回信給你． 如果兩個星期之內你還沒收到我的回音，請嘗試再寄一次．

也可以藉由下面的地址以遲緩的傳統信件聯絡我:
Greg Hankins
College of Computing
801 Atlantic Drive
Atlanta, GA 30332-0280
以及經由全球資訊網到我的網頁
http://www.cc.gatech.edu/staff/h/Greg.Hankins/.

撰寫時請將 Serial HOWTO 文件的版本編號含入，這份是 1.11 版．

1.4 免責聲明

你的狀況可能很不一樣．這裡的回答可能無法在所有的系統及各種設定的組合下運作．

2 已經支援的串列硬體

以知 Linux 可以與下列的串列硬體配合運作．

2.1 標準 PC 串列埠

- 標準 PC 串列埠 (COM1 - COM4)，能夠讓外接的串列裝置(數據機，串列滑鼠等)連接的
- 標準 PC 內接式數據機 (COM1 - COM4)
- Quickpath Systems Port-Folio 550e (可用的 IRQ 爲 3, 4, 5, 9, 10, 11, 12, 及 15)

注意: 由於位址衝突，你不能同時使用 COM4 及 IBM8514 視訊卡．這是 IBM8514 視訊卡的一個 bug．

2.2 即插即用型(PnP)數據機

即插即用型數據機能夠在 Linux 上運作．然而，我不建議使用它們，因爲有困難． 你有好多事情要試:

- 試著關掉你電腦 BIOS 及數據機上的 PnP 功能．
- 啓動 Windows 95，設定數據機，然後看看在你啓動 Linux 後這些設定能否保留下來． 如果不行的話，再來:
- 啓動 Windows 95，設定數據機，然後使用暖開機啓動 Linux 以便讓設定能保留下來． 你可以使用 `loadlin` 載入器來啓動．
- 從 *http://www.redhat.com/linux-info/pnp* 取回 PnP 工具．
- 然後你就有個可用的數據機．

2.3 一般多埠串列卡(附 8250/16450/16550A UART 晶片)

- AST FourPort and clones (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 serial ports, 2 parallel ports)
- Computone ValuePort V4-ISA (AST FourPort compatible)
- Digi PC/8 (8 ports)
- GTEK BBS-550 (8 ports)
- Longshine LCS-8880, Longshine LCS-8880+ (AST FourPort compatible)
- Moxa C104, Moxa C104+ (AST FourPort compatible)
- PC-COMM (4 ports)
- *Sealevel Systems* COMM-2 (2 ports), COMM-4 (4 ports) and 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)

一般來說， Linux 能夠支援所有使用 8250, 16450, 16550, 16550A（或與其相容）UART 晶片的串列卡， 或者是能夠摸擬上列 UARTs 之一的內接式數據機．

注意: BB-1004 以及 BB-1008，它們並不支援 DCD 及 RI 線路，因而無法讓接受撥接的數據機使用．它們在其它用途上則表現正常．

2.4 智慧型多埠串列卡

- Comtrol RocketPort (36MHz ASIC; 4, 8, 16 or 32 ports)
  contact: *info@comtrol.com* or *http://www.comtrol.com*
  driver status: supported by Comtrol
  driver location: *ftp://tsx-11.mit.edu/pub/linux/packages/comtrol*

- Computone IntelliPort II (16MHz 80186; 4, 8, or 16 ports),
  IntelliPort II EXpandable (20MHz 80186; 16 - 64 ports)
  contact: Michael H. Warfield, *mhw@wittsend.atl.ga.us*
  driver status: pre-ALPHA

- Cyclades Cyclom-Y (Cirrus Logic CD1400 UARTs; 8 - 32 ports),
  Cyclom-Z (25MHz MIPS R3000; 8 - 128 ports)
  contact: *sales@cyclades.com* or *http://www.cyclades.com*
  driver status: supported by Cyclades
  driver location: *ftp://ftp.cyclades.com/pub/cyclades* and included in Linux kernel since version 1.1.75

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

- Digi PC/Xi (12.5MHz 80186; 4, 8, or 16 ports),
  PC/Xe (12.5/16MHz 80186; 2, 4, or 8 ports),
  PC/Xr (16MHz IDT3041; 4 or 8 ports),
  PC/Xem (20MHz IDT3051; 8 - 64 ports)
  contact: *mailto:sales@dgii.com* or *http://www.dgii.com*
  driver status: supported by Digi
  driver location: *ftp://ftp.dgii.com/drivers/linux* and included in Linux kernel since version 2.0

- Digi COM/Xi (10MHz 80188; 4 or 8 ports)
  contact: Simon Park, *si@wimpol.demon.co.uk*
  driver status: ALPHA
  note: Simon is often away from email for months at a time due to his job. Mark Hatle, *fray@krypton.mankato.msus.edu* has graciously volunteered to make the driver available if you need it. Mark is not maintaining or supporting the driver.

- Equinox SuperSerial Technology (30MHz ASIC; 2 - 128 ports)
  contact: *sales@equinox.com* or *http://www.equinox.com*
  driver status: supported by Equinox
  driver location: *ftp://ftp.equinox.com/library/sst*

- GTEK Cyclone (16C654 UARTs; 6, 16 and 32 ports),
  SmartCard (24MHz Dallas DS80C320; 8 ports),
  BlackBoard-8A (16C654 UARTs; 8 ports),
  PCSS (15/24MHz 8032; 8 ports)
  contact: *spot@gtek.com* or *http://www.gtek.com*
  driver status: supported by GTEK
  driver location: *ftp://ftp.gtek.com/pub*

- Hayes ESP (COM-bic; 1 - 8 ports)
  contact: Andrew J. Robinson, *arobinso@nyx.net* or *http://www.nyx.net/~arobinso*
  driver status: supported by author
  driver location: *http://www.nyx.net/~arobinso* and included in Linux kernel since version 2.1.15
- Maxpeed SS (Toshiba; 4, 8 and 16 ports)
  contact: *info@maxpeed.com* or *http://www.maxpeed.com*
  driver status: supported by Maxpeed
  driver location: *ftp://maxpeed.com/pub/ss*
- Moxa C218 (12MHz 80286; 8 ports),
  Moxa C320 (40MHz TMS320; 8 - 32 ports)
  contact: *info@moxa.com.tw* or *http://www.moxa.com.tw*
  driver status: supported by Moxa
  driver location: *ftp://ftp.moxa.com.tw/drivers/c218-320/linux*
- SDL RISCom/8 (Cirrus Logic CD180; 8 ports)
  contact: *sales@sdlcomm.com* or *http://www.sdlcomm.com*
  driver status: supported by SDL
  driver location: *ftp://ftp.sdlcomm.com/pub/drivers*
- Specialix SIO (20MHz Zilog Z280; 4 - 32 ports),
  XIO (25MHz Zilog Z280; 4 - 32 ports)
  contact: Simon Allen, *simonallen@cix.compulink.co.uk*
  driver status: BETA
  driver location: *ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial*
- Stallion EasyIO-4 (4 ports), EasyIO-8 (8 ports), and
  EasyConnection (8 - 32 ports) - each with Cirrus Logic CD1400 UARTs,
  Stallion (8MHz 80186 CPU; 8 or 16 ports),
  Brumby (10/12 MHz 80186 CPU; 4, 8 or 16 ports),
  ONboard (16MHz 80186 CPU; 4, 8, 12, 16 or 32 ports),
  EasyConnection 8/64 (25MHz 80186 CPU; 8 - 64 ports)
  contact: *sales@stallion.com* or *http://www.stallion.com*
  driver status: supported by Stallion
  driver location: *ftp://ftp.stallion.com/drivers/ata5/Linux* and included in linux kernel since 1.3.27

其中 Comtrol, Cyclades, Digi, 及 Stallion 的產品在 1995 年六月份的 *Linux Journal* 上有篇回顧文章．這篇文章可在 *http://www.ssc.com/lj/issue14* 找到．

2.5 不支援的串列硬體

宣稱"Winmodems"的數據機，例如 USR Sportster Winmodem 及 IBM Aptiva MWAVE， 無法在 Linux 上使用．這些數據機具有專門的設計需要特別的 Windows 驅動程式． 避免使用這樣的數據機．除此之外，基於相同理由也別使用需要 Rockwell RPI 驅動程式的數據機．

沒有 Linux 所需驅動程式的智慧型串列埠也不能使用．

3 串列埠的名稱是什麼?

一個輸出/入(I/O)埠是用來將資料送入電腦及從電腦取出的通路· 有許多種類的輸出/入埠,例如串列埠,平行埠,磁碟控制器,乙太網路卡等等· 我們要處理的是串列埠,因爲數據機及終端機是串列裝置· 每一個串列埠必須有一個輸出/入位址,以及一個插斷號碼(IRQ)· 有四個串列設備對應到 COM1 - COM4:

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

如果 Linux 啓動時沒有偵測到任何串列裝置,請確定在編譯核心時有加入串列支援· 在這份文件中,我是以 ttyS0 參照 COM1,ttyS1 參照 COM2,以 ttyS2 參照 COM3,而以 ttyS3 參照 COM4· 注意這些裝置預設的 IRQ 是重疊的·你無法由這些預設值使用所有的串列埠,你必須重新指定不同的 IRQ· 參見 8 (我能使用超過兩個的串列裝置嗎?) 一節說明如何設定 IRQ·

某些安裝方式會建立兩個額外的設備,給你數據機使用的 /dev/modem 以及給你滑鼠使用的 /dev/mouse·它們都是鏈結到你在安裝時所指定適當 /dev 裡的設備的符號鏈結(除非你用的是匯流排滑鼠(bus mouse),那麼 /dev/mouse 將會指到匯流排滑鼠這個設備)·

已經有些關於使用 /dev/mouse 以及 /dev/modem 優缺點的討論· 我強烈地反對使用這些鏈結,如果你打算用你的數據機接受撥接的話,要是你使用了 /dev/modem 那麼你將會因爲鎖定檔案(lock files)無法正確運作而遭遇到困難· 如果你喜歡它們的話就用吧,但是要確定它們指向正確的設備· 然而,如果你改變或除去這個連結,某些應用程式(例如 minicom)可能要重新設定·

3.1 在 /dev 裡的串列設備及主要和次要號碼

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

要注意到所有的發行套件都應該已經正確地建立了這些設備· 你可以用下法檢查:

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

3.1.1 在 /dev 中建立設備

如果你沒有其中的一個設備,你必須自己用 mknod 指令建立它· 例如,假定你需要建立 ttyS0 設備:

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

你也可以使用 MAKEDEV 這個指令稿,在 /dev 下應該會有· 這簡化了設備的建立·例如,如果你需要建立 ttyS0 設備則應該輸入:

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

這將會處理關於連線進來以及連線出去的設備之建立，並設定正確的權限．

3.2 一般多埠板的注意事項

你的多埠板所使用的設備視你使用那一種多埠板而定． 這些細節詳細列在隨附於 setserial 程式的 rc.serial 裡面． 如果你嘗試使用多埠板的話我強烈建議你取得最新版本的 setserial． 你可能會需要建立這些設備．無論是用 mknod 指令或是 MAKEDEV 指令稿都可以． 多埠板所用設備是以加上 “64 + 埠號” 來建立的．所以，如果你想爲 ttyS17 建立設備，你應該輸入：

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

要注意到 ”64 + 17 = 81”．使用 MAKEDEV 指令稿的話，你應該輸入：

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

注意: SIIG 的 IO1812 手冊上列出的 COM5-COM8 是錯的．它們應該是 COM5=0x250, COM6=0x258, COM7=0x260, 以及 COM8=0x268．

注意: Digi PC/8 Interrupt Status Register 是 0x140．

注意: 對於 AST Fourport，你必須在 rc.serial 中指定 skip_test．

3.3 智慧型多埠板的注意事項

閱讀隨附驅動程式而來的資訊． 這些板使用了特殊的裝置，而非標準的．這些資訊視你的硬體而定．

4 你應知道的有趣程式

4.1 什麼是 getty?

getty 是在你簽入 Unix 機器的時候處理簽入程序的一支程式． 如果你希望你的 Linux 機器能夠使用數據機撥入的話你會需要 getty． 如果你只需要用數據機撥出的話你不需要用 getty． 有三種經常配合 Linux 使用的版本: getty_ps, mgetty 以及 agetty． 這些程式的語法是不同的，所以不論你用什麼 getty 請確定你使用了正確的語法．

4.1.1 關於 getty_ps

大部分 Linux 安裝套件都安裝了 getty_ps 套件．它包含了兩個程式: getty 是由主控台，以及終端機設備所使用，而 uugetty 則是由數據機所使用的． 因爲我本身是使用這個版本的 getty，所以在這份文件中我將會著重於此套件．

4.1.2 關於 mgetty

mgetty 是一個在數據機上使用的 getty 版本. 除了能夠允許簽入之外, mgetty 也提供傳真(FAX)支援及自動 PPP 偵測. 這套 mgetty 的文件寫得相當地好，因而不需要再補充什麼.請參考它的安裝說明. 你也可以在 *http://www.leo.org/~doering/mgetty/* 找到 mgetty 的最新資訊.

4.1.3 關於 agetty

agetty 是第三種 getty 變型. 它是 getty 一個簡單的，完整的功能實作，適合用在虛擬主控台或終端機而非數據機上.

4.2 什麼是 setserial?

setserial 是一支可以讓你查看並修改串列設備各種屬性的程式，包括它們所使用的埠位址，中斷，以及其它串列埠選項. 你可以藉由執行不加參數的 setserial 來獲取版本訊息.

在你的 Linux 系統啟動時，只有 ttyS{0-3} 會以預設的 IRQ 4 以及 3 來進行配置. 所以，如果你有任何由其它界面卡提供的串列埠或是 ttyS{0-3} 使用非標準的硬體中斷(IRQ)的話, 你必須用這支程式來配置這些串列埠.要知道所有的選項請參考線上手冊.

5 我怎麼用數據機撥號出去?

5.1 硬體需求

首先，要確定你擁有正確的連接線.你的數據機需要一條直接的 (straight through) 連接線，沒有任何跳線 (no pins crossed). 任何一家電腦商店都有.要確定接頭的形式是對的.如果你使用的是 DB25 串列埠，它應該是公的 DB25 接頭. 把你的數據機接到其中一個串列埠上.如果你需要關於這個動作的輔助說明請參考你的數據機手冊.

5.1.1 內接式數據機注意事項

對於內接式數據機而言，你不需要連接線.內接式數據機不需要串列埠，它自己有內建一個. 你只要配置一個沒有被使用的硬體中斷給它，並且配置該埠的輸出入位址.如果你卡在某個問題上請參考你的數據機手冊. 同樣，如果你需要關於選擇中斷以及位址上的幫助，參閱 8 (我能使用超過兩個的串列裝置嗎?) 一節的說明.

在某些主機板上你必須關掉被數據機取代的串列埠以避免衝突. 這可經由跳線或 BIOS 的設定來辦到，視你的主機板而定.查閱你的主機板手冊.

由於 IBM8514 視訊卡的一個錯誤，如果你想要將你的內接式數據機設在 ttyS3 的話可能會遭遇一些問題. 如果 Linux 沒有偵測到你設在 ttyS3 上的內接式數據機，你可以使用 setserial 程式來設定然後數據機將會正常運作. 設在 ttyS{0-2} 上的內接式數據機在偵測上應該不會有任何問題. 由於此視訊卡的錯誤 Linux 在 ttyS3 沒有任何自動偵測的動作.

5.2 與你的數據機交談

在你使用複雜的通訊程式如 SLIP 或 PPP 之前先以 kermit 或其它簡單的通訊程式來測試這些設定·你可以從 *http://www.columbia.edu/kermit/* 以及映射節點上找到最新的 kermit 版本· 例如，要是你的數據機設在 ttyS3 上，而且可以接受 115200 bps 的話，你可以這樣做:

```
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                            ; 這是你鍵入的並按下 Enter 鍵
OK                                  ; 數據機應該回應這個
```

如果你的數據機對 AT 指令群有回應，那麼你可以假定你的數據機在 Linux 這一端能正確運作· 現在試著撥給另一台數據機:

```
ATDT7654321
```

其中 7654321 是一個電話號碼·如果你的是脈衝線路(pulse line)的話用 ATDP 而不是 ATDT· 如果可以撥得出去，你的數據機就可正常運作·

回到 kermit 的提示符號下，按住 Ctrl 鍵，按下 \ (backslash) 鍵，然後放開 Ctrl 鍵，再按下 C 鍵:

```
Ctrl-\-C
(Back at linux)
C-Kermit>quit
linux#
```

這個不過是用"手動"方式的撥號測試·正常的方法是讓 kermit 用它內建的數據機資料庫自動爲你撥號，例如，使用 US Robotics (USR) modem:

```
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         ; 選擇數據機型式
C-Kermit>set line /dev/ttyS3        ; 選擇通訊裝置
C-Kermit>set speed 115200           ; 設定通訊速度
```

```
C-Kermit>dial 7654321              ; 撥號
 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:
```

如果你需要一些指引，查閱有關 10 (通訊軟體) 一節說明．

當你用你的數據機撥號出去時，將你的數據機設到它所能支援的最高速度． Linux 的 libc 5.x 版以上已經支援到 115200 bps． libc 通常放在 /lib，因此到那裡看一下你用什麼版本． 如果 Linux 無法認得 57600 或 115200 bps，那麼你必須用 setserial 程式來將你的串列埠設到更高的速度． 關於要怎麼設定參閱 9 (如何將我的串列埠設定爲更高速度?) 該節說明． 然後在通訊程式裡將速度設到 38400 bps．

5.3 撥出用的數據機配置

只要撥號出去的話，你可以任意以你想要的方式配置你的數據機． 如果你計畫用你的數據機來接受撥接，那麼你必須將你的數據機配置成與你想要執行的 getty 在相同速度設定下． 所以，如果你想要在 38400 bps 的速度下執行 getty 的話，在配置你數據機的時候將速度設成 38400 bps． 這樣做是爲了避免你的電腦與數據機之間的速度無法配合． 一般來說，開啓錯誤校正及硬體流量控制的出廠設定值是最適合撥出用的數據機，這些設定值可查閱你的數據機手冊．

5.4 硬體流量控制

如果你的數據機支援硬體流量控制(RTS/CTS)的話， 我強烈地建議你去使用它． 這對於支援資料壓縮的數據機將格外重要．首先，你必須在串列埠本身上面啓動 RTS/CTS 流量控制功能． 這最好是在啓動系統的時候做，像是放在 /etc/rc.d/rc.local 或是 /etc/rc.d/rc.serial 裡面． 要確定這些檔案會被主要的 rc.sysinit 檔案執行！你需要對每一個你想要啓動硬體流量控制的串列埠執行下列操作：

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

你也必須在你的數據機上啓動 RTC/CTS 流量控制．要怎麼做請參考你的數據機手冊，因爲這在不同的數據機製造商之間是不同的． 如果你的數據機支援設定儲存的話則要確定將配置存入你的數據機．

6 我要怎麼樣用數據機以 getty_ps 撥入及撥出?

讓你的數據機能夠正確地撥號出去．如果你還沒有讀過 5 (我怎麼用數據機撥號出去) 一節的話現在去讀！ 它包含有非常重要的設定資訊．如果你只想用數據機來撥出的話你不需要讀此節．

6.1 撥入及撥出的數據機配置

要用來接受撥接以及撥號出去的話，那麼你必須確實地設定你的數據機（再一次對你的數據機使用 AT 指令群）：

E1	本地回應指令
Q0	回報結果回應
V1	開啓冗長模式
S0=0	不要接聽 (uugetty 以 WAITFOR 選項來處理)

如果你沒有正確地設定，放在你配置檔裡的起始(INIT)字串可能會失敗，因而停住整個程序．但是，下面有更多配置檔的選項...

&C1	DCD 只有在連線以後才開啓
&S0	DSR 永遠爲開啓
	DTR 開啓/關閉 重置數據機（是數據機製造商而定 - RTFM)

這些會影響你數據機有關開始以及結束撥號的動作．

如果你的數據機不支援儲存設定，你可以透過在你配置檔裡的起始字串來設定這些東西．參閱下面所述．有一些數據機是以 DIP 開關來影響暫存器的設定，也要確定這些設定是正確的．

我已經開始著手收集不同種類數據機的設定． 到目前爲止，我還只有它們的一小部分，如果你想要寄送給我你目前所用配置，非常歡迎！ 你可以從這些取得設些設定: *ftp://ftp.cc.gatech.edu/pub/people/gregh/modem-configs*．

注意: 爲了讓我的 USR Courier V.34 數據機在 DTR 訊號掉下去後正常地重置，我必須設定 &D2 及 S13=1(這是設定暫存器 S13 的 bit 0)．這確定能在 USR Sportster V.34 上運作正常．

注意: 某些 Supra 數據機處理 DCD 訊號的方式異於其它的數據機． 如果你在用 Supra 數據機，試著設定爲 &C0 而不是 &C1． 你也必須設定 &D2 使之正常地處理 DTR 訊號．

6.2 安裝 getty_ps

從 *ftp://sunsite.unc.edu/pub/Linux/system/serial* 最新的版本． 特別地，如果你想使用高速通訊 (57600 或 115200 bps)，你必須取得 2.0.7j 或以後的版本． 你也必須有 libc 5.x 或之後的版本．

依照預設值，getty_ps 將會照 Linux 的檔案系統標準(FSSTND, File System Standard)來配置，這個意思是二進位碼將會放在 /sbin 下，而配置檔將會以 /etc/conf.{uu}getty.ttyS*N* 來命名． 這點並不能從文件敘述中找到！它也會假定 lock files 將得要放到 /var/lock 目錄裡．要確定你有 /var/lock 這個目錄．

如果你不想依照 FSSTND 的話，二進位碼將會放到 /etc 裡面，而配置檔將會放到 /etc/default/{uu}getty.ttyS*N* 裡面，而 lock files 則將會放到 /usr/spool/uucp 裡面． 如果你有使用 UUCP 的話那麼我建議這樣做，因爲如果你把 lock files 搬到 UUCP 無法找到的地方去那麼它將會出問題的．

getty_ps 也可使用 syslogd 來記錄訊息．如果你目前沒有執行它的話，參閱 syslogd(1) 以及 syslog.conf(5) 的線上手冊來設定 syslogd． 訊息會優先以 LOG_AUTH 記錄，錯誤使用 LOG_ERR 記錄，而除錯使用 LOG_DEBUG． 如果你不想使用 syslogd 的話那麼你可以編輯在 getty_ps 原始檔案裡的 tune.h 用一個記錄檔來取代，預設檔名爲 /var/adm/getty.log．

決定好是否要依照 FSSTND 以及使用 syslog · 你也可以選擇兩者的組合 · 編輯 Makefile, tune.h 以及 config.h 來反應你所作的決定 · 然後, 根據套件裡的說明編譯及安裝 ·

現在起, 所有談到 getty 的地方都是指 getty_ps 而言 · 談到 uugetty 則是指 getty_ps 套件裡的 uugetty 而言 · 這些指令在 mgetty 或 agetty 上將不能運作 ·

6.3 設定 uugetty

爲了要能從你的數據機撥入及撥出, 我們想要用 uugetty · uugetty 執行很重要的 lock file 檢查 · 如果沒有的話更新 /etc/gettydefs 以便爲數據機增加項目 (注意到這裡的項目互相參考, 它們並非固定速度 - 每個項目之間的空白行是必要的):

```
# Modem entries
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
```

如果你有 9600 bps 或更快且具資料壓縮功能的數據機, 你可以鎖定你的串列埠速度而讓數據機去處理傳輸的其它速度 · 然後, 不像上面在 /etc/gettydefs 裡的一連串列表, 只需爲數據機加上一行, 例如:

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

如果你將你的數據機設成 RTS/CTS 執行硬體流量控制的話, 那麼你可以將 CRTSCTS 加到這些項目裡:

```
# 115200 fixed speed with hardware flow control
F115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login: #F115200
```

如果你要的話, 你可以讓 uugetty 在簽入時的標題上印出些有趣的東西 · 在我的例子裡, 我印出系統名稱, 串列線路, 以及目前的每秒位元傳輸率 · 你可以加入其它的東西:

```
@B    目前的每秒位元傳輸率(bps, 在看到 @B 時估計的) ·
@D    現在日期, 格式 MM/DD/YY ·
@L    getty 所連接的串列線路 ·
@S    系統名稱 ·
@T    現在時間, 格式 HH:MM:SS (24 小時制) ·
@U    目前簽入的使用者 · 這是在 /etc/utmp 檔中非空 ut_name 欄位項目的計數值 ·
@V    版本值, 在預設檔中給定的 ·
如果要顯示 '@' 符號的話, 用 '\@' 或 '@@' ·
```

接下來，確定你數據機所接的串列埠有輸出及輸入的設備· 如果你的數據機設在 ttyS3 的話你將會需要 /dev/cua3，以及 /dev/ttyS3 這兩個設備· 如果你沒有正確的設備，參閱 3.1.1 (在 /dev 中建立設備) 該節說明如何建立設備，並且建立這些設備· 如果你希望在你的 uugetty 正在監視這個串列埠是否有人簽入時還能夠用數據機撥出的話，用 /dev/cua$N$ 設備而不要用 /dev/ttySN·

你正在編輯 /etc/gettydefs 時，你可以這樣來檢查它的語法是否正確:

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

6.4 自訂 uugetty

你有許多參數可以用來改變你所擁有的每一個埠·這些是以分別對每一個埠使用不同配置檔的方式來實作的· 這個 /etc/conf.uugetty 檔案會被所有的 uugetty 行程所使用，而 /etc/conf.uugetty.ttySN 只由那個埠使用· 預設的配置檔範例可以在 getty_ps 的原始檔案裡找到，這在絕大部分的 Linux 發行套件裡面都會有· 因爲篇幅的考量，沒有把它們列在這裡· 要注意的是如果你用的是比較舊的 getty 版本(比 2.0.7e 版還舊)，或者是你沒有依照 FSSTND 的話，那麼預設的檔案將會是 /etc/default/uugetty.ttySN· 我的 /etc/conf.uugetty.ttyS3 看起來像這樣:

```
# sample uugetty configuration file for a Hayes compatible modem to allow
# incoming modem connections
#
# alternate lock file to check... if this lock file exists, then uugetty is
# restarted so that the modem is re-initialized
ALTLOCK=cua3
ALTLINE=cua3
# line to initialize
INITLINE=cua3
# timeout to disconnect if idle...
TIMEOUT=60
# modem initialization string...
# format: <expect> <send> ... (chat sequence)
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# this line sets the time to delay before sending the login banner
DELAY=1
#DEBUG=010
```

將下面那一行加到你的 /etc/inittab 裡去，那麼 uugetty 就會在你的串列埠上執行(用你環境的正確資訊來取代 _ 配置檔的位置，串列埠名，速度，及預設終端機形式):

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

重新啓動 init:

```
linux# init q
```

對於在你 /etc/inittab 裡面的速度參數，你會希望使用你數據機所能支援的最高每秒位元傳輸率·

現在 Linux 將會隨時監視你的串列埠是否有連線·從別的地方撥號並簽入你的 Linux 機器·

uugetty 還有更多的選項，參閱 getty (lm) 線上手冊獲得更完整的敘述． 別的不提它還有排程(scheduling)的功能，以及回撥的能力．

7 我怎麼設定終端機連線到我的 PC 上?

本節的指示能讓終端機及其它電腦連上你的 Linux 機器的串列埠．

7.1 硬體需求

要確定你有一條正確的連接線．在電腦商店買的 null modem 連接線就可以勝任． 但是它一定得要是一條 *null modem* 線！許多的電腦站將這類的連接線說成是串列印表接線． 確定你是使用你的串列埠，公的 DB25 或是 DB9 接頭，而不是你的並列埠(母的 DB25 或 centronics)．

對 DB-25 接頭而言，至少你得要有:

```
PC male DB25                               Terminal DB25
TxD   Transmit Data          2 --> 3      RxD   Receive Data
RxD   Receive Data           3 <-- 2      TxD   Transmit Data
SG    Signal Ground          7 --- 7      SG    Signal Ground
```

如果你想要擁有流量控制以及硬體連線協商(handshaking)的話，你必須要有一條完整的 null modem 接線：

```
PC male DB25                               Terminal DB25
TxD   Transmit Data          2 --> 3      RxD   Receive Data
RxD   Receive Data           3 <-- 2      TxD   Transmit Data
RTS   Request To Send        4 --> 5      CTS   Clear To Send
CTS   Clear To Send          5 <-- 4      RTS   Request To Send
DSR   Data Set Ready         6
                             |
DCD   Carrier Detect         8 <-- 20     DTR   Data Terminal Ready
SG    Signal Ground          7 --- 7      SG    Signal Ground
                                   6      DSR   Data Set Ready
                                   |
DTR   Data Terminal Ready   20 --> 8      DCD   Carrier Detect
```

如果你的串列埠上有一個 DB9 的接頭，試試看下面這個:

```
PC DB9                                     Terminal DB25
RxD   Receive Data           2 <-- 2      TxD   Transmit Data
TxD   Transmit Data          3 --> 3      RxD   Receive Data
SG    Signal Ground          5 --- 7      SG    Signal Ground
```

或者，一條完整的 DB9 對 DB25 null modem 連接線：

```
PC DB9                                      Terminal DB25
RxD   Receive Data          2 <-- 2         TxD   Transmit Data
TxD   Transmit Data         3 --> 3         RxD   Receive Data
                                  6         DSR   Data Set Ready
                                  |
DTR   Data Terminal Ready   4 --> 8         DCD   Carrier Detect
GND   Signal Ground         5 --- 7         GND   Signal Ground
DCD   Carrier Detect        1
                            |
DSR   Data Set Ready        6 <-- 20        DTR   Data Terminal Ready
RTS   Request To Send       7 --> 5         CTS   Clear To Send
CTS   Clear To Send         8 <-- 4         RTS   Request To Send
(RI   Ring Indicator        9 not needed)
```

(沒錯! 第二支及第三支腳在 DB9 連接頭上確實和 DB25 接頭意義相反!)

如果使用的並不是完整的 null modem 連接線的話，你可能得要做下面這個欺騙的技巧: 在你電腦這邊的接頭上，把 RTS 跟 CTS 接在一起，而且也把 DSR, DCD 以及 DTR 接在一起． 用這個方法，當電腦要求一個正確的硬體連線協商信號時，它就可以得到(從它自己發出的)．

現在你已經有正確的連接線，把你的終端機連接到你的電腦上． 如果你可以的話，告訴你的終端機忽略數據機控制信號． 試試看用 9600 bps, 8 data bits, 1 stop bit, no parity bits 來設定終端機．

7.2 設定 getty

在 /etc/gettydefs 裡面增加一個給 getty 用來連接你終端機項目，如果還沒有這項的話:

```
# 38400 bps Dumb Terminal entry
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

# 19200 bps Dumb Terminal entry
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# 9600 bps Dumb Terminal entry
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

如果你要的話，你可以讓 getty 在簽入的標題上印出一些有趣的東西． 在我的例子裡，我印出系統名稱以及串列線路．你可以加入其它的東西:

```
@B    目前的每秒位元傳輸率(bps, 在看到 @B 時估計的)．
@D    現在日期, 格式 MM/DD/YY．
@L    getty 所連接的串列線路．
@S    系統名稱．
@T    現在時間，格式 HH:MM:SS (24 小時制)．
@U    目前簽入的使用者．這是在 /etc/utmp 檔中非空 ut_name 欄位項目的計數值．
@V    版本值，在預設檔中給定的．
如果要顯示 '@' 符號的話，用 '\@' 或 '@@'．
```

當你正在編輯 /etc/gettydefs 時，你可以這樣來檢查它的語法是否正確:

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

確定沒有 getty 或 uugetty 的設定檔 (/etc/default/{uu}getty.ttySN 或 /etc/conf.{uu}getty.ttySN)在你的終端機連接的串列埠上， 因爲這也許會干擾到在終端機上執行的 getty．如果有的話請移除它．

編輯你的 /etc/inittab 讓 getty 在你的串列埠上執行(用你環境的正確資訊來取代 - 配置檔的位置，串列埠名，速度，及預設終端機形式):

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

重新起始 init:

```
linux# init q
```

現在你應該會在你的終端機上看到簽入提示．也許你得按返回鍵(return)幾次讓終端機注意到．

7.3 將 PC 設定成終端機的注意事項

許多人將其它 PC 設定成終端機連接到 Linux 機器上．例如，舊的 8088 或是 286 PC 非常適合這個用途． 你需要的只是一張包含適合讓你的終端機-個人電腦 (Terminal-PC) 使用的 DOS 磁片，以及讓你的終端機-個人電腦執行的通訊程式． kermit 在這個用途上表現得非常好． 你可以在這台機器 *http://www.columbia.edu/kermit/* 上找到幾乎可以讓任何現存的作業系統使用的已編譯的 kermit 版本． 其它流行的 DOS 通訊程式像是 telix 以及 procomm 也將會有一樣好的表現． 要確定輸入正確的串列埠資訊到你終端機-個人電腦的通訊設定裡去．

8 我能使用超過兩個的串列裝置嗎?

你不需要閱讀這一節，除非你想要使用三個或更多的串列設備...(假定你沒有多埠卡的話)．

假如你有另一個多餘的串列埠，是的，你可以． 你可以使用的串列埠數目受限於我們必須使用的硬體中斷要求以及埠輸出入位址的數目． 這並不是 Linux 的限制，而是 PC 匯流排的限制．必須指定中斷跟位址給每一個串列設備． 串列設備可以是串列埠，內接式數據機或是多埠卡．

多埠卡是爲了擁有許多個串列埠來分享卡上所有串列埠所使用的相同 IRQ 而特別設計的． Linux 對卡上的每一個埠使用不同的輸出入位址以便從它們獲取資料．

8.1 選擇串列設備的中斷位址

你的 PC 一般會將 ttyS0 以及 ttyS2 設在 IRQ 4 而將 ttyS1 以及 ttyS3 設在 IRQ 3． 經由檢視 /proc/interrupts 你可以知道那些 IRQs 正被使用． 要使用兩個以上的串列設備的話，你必須重新指定一個中斷． 一個好選擇是從你的並列埠那邊重新指定一個中斷．你的 PC 一般會設定 IRQ5 或是 IRQ7 當你的並列埠，但是很少人會使用兩個並列埠． 你可以重新指定其中一個中斷給串列設備，並而仍然能順利地使用一個並列埠． 你將會需要用到 setserial 程式來做這件事．此外，你必須調整你板子上的跳接設定(Jumpers)，查看你主機板的說明文件． 跳接你想要給每個埠使用的 IRQ．

你將得要做些設定讓每個串列設備都有一個，並且是唯一的一個中斷． 這裡是我在 /etc/rc.d/rc.local 裡所用的設定 - 你應該在啓動系統時的某個地方做這件事．

```
/sbin/setserial /dev/ttyS0 irq 3        # my serial mouse
/sbin/setserial /dev/ttyS1 irq 4        # my Wyse dumb terminal
/sbin/setserial /dev/ttyS2 irq 5        # my Zoom modem
/sbin/setserial /dev/ttyS3 irq 9        # my USR modem
```

標準的 IRQ 指定:

```
IRQ  0    Timer channel 0
IRQ  1    Keyboard
IRQ  2    Cascade for controller 2
IRQ  3    Serial port 2
IRQ  4    Serial port 1
IRQ  5    Parallel port 2
IRQ  6    Floppy diskette
IRQ  7    Parallel port 1
IRQ  8    Real-time clock
IRQ  9    Redirected to IRQ2
IRQ 10    not assigned
IRQ 11    not assigned
IRQ 12    not assigned
IRQ 13    Math coprocessor
IRQ 14    Hard disk controller 1
IRQ 15    Hard disk controller 2
```

在選擇中斷時眞的沒有什麼正確的．只要確定它不是已經被主機板，或是其它界面卡所使用． 2, 3, 4, 5 或是 7 是好選擇．"not assigned"的意思是目前並沒有什麼東西在標準情況下會固定使用這些硬體中斷要求． 也要注意到 IRQ 2 跟 IRQ 9 是一樣的．你可以稱它爲 2 或 9，串列驅動程式非常了解這一點． 如果你有一塊 16-bit 匯流排連接頭的串列卡，你也可以使用 IRQ 10, 11, 12 或 15．

只要確定你使用的並非是 IRQ 0, 1, 6, 8, 13 或是 14! 它們是由你的主機板所使用的． 如果你拿走她的 IRQs 的話，會使得她很不高興． 完成之後，再次檢查 /proc/interrupts 以確定沒有衝突發生．

8.2 設定串列設備的位址

接下來，你必須設埠位址．查看你主機板手冊的跳接設定．就像中斷，每個位址只能有一個串列設備． 你的埠通常會配置成像下面這樣：

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

選擇你想要給每一個串列設備使用的位址並且依此來設定跳接． 我將我的數據機設在 ttyS3，我的滑鼠在 ttyS0，而我的終端機在 ttyS2．

當你重新啓動時， Linux 應該會在你所設定的位址看到你的串列埠． Linux 所看到的 IRQ 可能不會跟你跳接設定的相符． 不必擔心這一點． Linux 在它啓動的時候並沒有做任何的 IRQ 偵測， 因爲 IRQ 偵測是危險並且有可能被欺騙的．使用 setserial 來告訴 Linux 那個埠使用那個 IRQ． 你也可以檢查 /proc/ioports 來看看在 Linux 啓動後那些輸出輸入埠位址被使用．

9 我怎樣將我的串列埠設到較高的速度? 我的數據機又應該使用什麼樣的速度?

這一節應該可以幫你找出當你透過通訊程式,或是 getty 程式使用你的數據機時應該使用什麼速度·

- 如果你所有的是比 9600 bps (V.32) 數據機還要慢的東西,將你的速度設到你的數據機所能支援的最高速度· 例如 300 (V.21/Bell 103), 1200 (V.22/Bell 212A), 或 2400 bps (V.22bis)·
- 如果你有 9600 bps (V.32) 的數據機,並且支援 V.42bis 資料壓縮,使用 38400 來當作你的速度· V.42bis 壓縮具有 4:1 的理論值,也就是"4 * 9600 = 38400"·
- 如果你有 14400 bps (V.32bis) 的數據機,並且支援 V.42bis 資料壓縮功能的話,使用 setserial 程式,加上 spd_hi 旗標將你的串列埠配置成使用 57600 bps (4 * 14400 = 57600) 的速度· 如果你有 28800 或 33600 bps (V.FC 或是 V.34) 的數據機則使用 spd_vhi 旗標 (4 * 28800 = 115200)·

  然後,在你的通訊程式,或是 /etc/inittab 裡使用 38400 當作速度設定· 這是現在你能設的最高速度·要確定你有 16550A 通用非同步接收轉換器(UART)·

  如果你的 libc 版本至少是 5.x,那麼會有叫做 57600 及 115200 的速度· libc 放在 /lib,所以在那裡看一下你用的版本是什麼· 如果你的應用程式已經編譯以取用此優點,你可以直接使用它們(不必用 setserial), 許多的套件都已經有了,因此如果你有最近版本的 Linux 套件,最好的方法就是嘗試使用這些更高的速度·

先在命令列下用 setserial 指令測試,然後在你確定它們可以運作後,將你所作的修改放進 /etc/rc.d/rc.serial 或 /etc/rc.d/rc.local 以便在系統啓動執行它們· 在我的 /etc/rc.d/rc.local 裡,我是這樣設定我的 ttyS3 到 115200 bps 的:

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

確定你以合法的路徑使用 setserial,以及合法的裝置名稱· 你可以下法檢查一個串列埠的設定:

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

10 通訊程式及工具

這裡有一個通訊程式的列表,如果你所使用的發行套件裡頭並沒有的話,這些都可以在一般的 FTP 站找到·

- ecu - 一套通訊程式
- *C-Kermit* <http://www.columbia.edu/kermit/> - 可攜的,可程序化的串列及 TCP/IP 通訊程式 ,包含傳檔及字完轉換功能
- minicom - 類似 telix 的通訊程式
- procomm - 包含 zmodem,類似 procomm 的通訊程式
- seyon - 以 X 爲基礎環境的通訊程式
- xc - xcomm 通訊套件

- 其它有用的程式是 term 及 SLiRP．它們提供使用 shell 帳號的 TCP/IP 功能．
- screen 是另一個多區段作業(multi-session)的程式．這個程式就像的動作就像虛擬控制台．
- callback 這支程式會使你的數據機回撥給你．
- mgetty+fax 處理傳真事務，並且提供一個不同的 getty．
- ZyXEL 是 ZyXEL U-1496 數據機的控制程式．它處理接受撥接，撥號，回撥保密，傳真，以及語音信箱功能．
- SLIP 及 PPP 的軟體可在 *ftp://sunsite.unc.edu/pub/Linux/system/network/serial* 找到．
- 其它東西可以在 *ftp://sunsite.unc.edu/pub/Linux/system/serial* 以及 *ftp://sunsite.unc.edu/pub/Linux/apps/serialcomm* 或許多的映射節點之一找到．這些是保留所有串列程式的目錄．

11 串列使用秘訣

這裡有一些你可能發現很有用的串列祕訣...

11.1 kermit 以及 zmodem

要在 kermit 下使用 zmodem 的話，將下列內容加到你的 .kermrc 裡去:

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

要確定正確地放入你數據所在的串列埠．然後，要使用的話，只要在 kermit 的提示符號下鍵入 rz 或 sz <檔名> 即可．

11.2 自動設定終端機型式

要在當你簽入的時候自動設定你的終端機形式的話，把終端機的形式加到 /etc/inittab 的項目裡去． 如果我有一台 vt100 終端機在 ttyS1 上，我要對 getty 指令加上 “vt100”:

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

你也可以使用 tset，可以在你簽入的時候建立終端機的特性，而且不需要依賴任何的預設值．

11.3 在串列連線下的彩色 ls

如果 ls 因爲彩色特性而擾亂了你的終端機模擬，把它關掉． ls -color 以及 ls -colour 都使用彩色特性．有些安裝方式使得 ls 預設爲使用彩色． 檢查 /etc/profile 以及 /etc/csh.cshrc 裡的 ls 別名． 如果你不想修改系統預設值的話，你也可以設給 ls 別名 ls -no-color．

11.4 從終端機所連接的印表機進行列印

有一個叫做 vtprint 的程式可以做得到，在 *ftp://ftp.sdsu.edu/pub/vtprint* 及 *http://www.sdsu.edu/~garrett* 可取得．

外一支叫 xprt 的程式也能做到．它可在 *ftp://sunsite.unc.edu/pub/Linux/system/printing* 找到．

11.5 Linux 能不能夠自動地配置串列設備?

可以．要讓 Linux 在啓動的時候自動地偵測並且設定串列設備的話，把這一行:

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

加到你的 /etc/rc.d/rc.local 或是 /etc/rc.d/rc.serial 檔案裡面． 對每一個你希望自動配置的串列埠都要做．要確定所給的是一個眞正存在於你電腦上的設備名稱．

11.5.1 多埠卡的注意事項

對於卡的位址，以及 IRQs 查閱 setserial 程式隨附的 rc.serial． 它有很多關於多埠卡的細節，包括輸出入位址以及設備名稱．

11.6 使用串列控制台

在第三十六期的 *Linux Journal* 在 *http://www.ssc.com/lj/issue36/index.html* 有篇文章解釋如何使用串列控制台(serial console)． 不幸的是，寄電子郵件給作者都被退回．我希望這篇文章會很快地上述 URL 的 *Linux Journal* 網站中．

11.7 更高的串列流量

如果在一部使用 (E)IDE 磁碟機的系統上你發現流量很慢而且串列埠常溢滿，你可以取得 hdparm． 這是一個可以修改 (E)IDE 參數，包含讓一磁碟 IRQ 動作期間其它 IRQs 仍可工作． 這會改善反應速度並幫助消除溢滿．確定小心地讀過了線上手冊，因爲某些磁碟/控制卡組合不喜歡它並且可能損毀檔案系統．

也可以看一下一個叫 irqtune 的工具，可以改變一個裝置 IRQ 的優先順序，例如你數據機所在的串列埠．這也可能改善你系統的串列流量． irqtune 的常見問題解答(FAQ)可在 *http://www.best.com/~cae/irqtune* 找到．

12 更進一步...

這一節並不需要閱讀，但是可能帶給你一些對於 Unix 以及電子通訊世界更進一步的了解．

12.1 什麼是鎖定檔案(lock files)?

Lock files 只是單純的一個檔案用來表示某個特定的設備正在使用中． 它們保存在 /usr/spool/uucp 或是 /var/lock 裡面． Linux 的 lock files 名字都是 LCK..*name*，其中的 *name* 不是設備名稱就是一個

UUCP 節點名稱．有些程序建立這些 locks 以便讓它們自己可以獨佔該設備的存取權．例如如果你從你的數據機撥號出去，將會出現一個 lock 告訴其它程序現在已經有人正在使用數據機．Lock files 裡面主要是包含鎖住該設備的程序的行程號碼(PID)．大部份的程式會查閱 lock，然後檢查行程表(process table)中鎖定該設備的行程來試著決定這個 lock 是否仍然有效．如果發現這個 lock 是有效的話，這個程式(應該)要結束．如果不是，某些程式會刪除過時的 lock，然後使用該設備，並在行程執行時建立他們自己的 lock．其它的程式就只好結束並告訴你該設備正在使用中．

12.2 "baud" 對 "bps"

"baud" 跟 "bps" 可能是電腦/電子通訊這個領域裡最常被錯用的術語之一．當它們事實上並不相同的時候，很多人仍交替使用這些術語．

飽率**(baud)**

飽率是每秒鐘數據機(調變解調器，**mo**dulator-**dem**odulator)送出多少次訊號改變的度量．例如，飽率爲 1200 的含意是信號每隔 833 微秒(microsecond)即改變一次．普遍的飽率有 50, 75, 110, 300, 600, 1200 以及 2400．大部份的高速數據機所使用的飽率爲 2400．因爲語音級(voice-grade)電話線路的頻寬限制，超過 2400 的飽率很難達的到，而只能在非常純淨的電話線品質下才能運作．"baud"一詞在 Emile Baudot 之後才有，他是非同步電報印表機(asynchronous telegraph printer)的發明者．

每秒傳輸位元**(bps)**

每秒位元傳輸率是每秒鐘傳輸多少位元的度量．普遍每秒位元傳輸率有 50, 75, 110, 300, 1200, 2400, 4800, 9600, ... 115200．在使用 V.42bis 壓縮(壓縮最大率 4:1)的數據機下，理論上每秒位元傳輸率可以高達 115200．這是許多人在錯用 "baud" 這個字時所指的意思．

所以，如果高速數據機以 2400 baud 運作，它們如何能夠傳送 14400 bps? 數據機對每個 baud 編碼來表示某一數量的位元而使得 bps 能大於 baud．這樣一來，當二個或更多位元編碼成一個 baud 的時候，bps 就會超越 baud．如果你的數據機是以 14400 bps 連線的話，它將會以 2400 baud 的速度每個 baud 傳送六個位元．

這個混淆是怎麼樣開始的? 嗯，回到今天的低速數據機還是昨天的高速數據機那個時代，每秒位元傳輸率跟信號改變率實際上是完全相等的．每個 bit 編碼成一個 baud．人們交替使用 bps 以及 baud 是因爲它們的數字相同．例如一台 300 bps 的數據機其信號改變率也是 300．當高速數據機普及後一切都改變了，而且每秒位元傳輸率超過了信號改變率．

12.3 什麼是 UARTs? 它們如何影響效能?

UARTs (通用非同步接收轉換器，**U**niversal **A**syncronous **R**eceiver **T**ransmitter)是你 PC 串列卡上的晶片．它們的功用是把資料轉成位元，將這些位元送上串列線路，然後在另一端重新建立資料．UARTs 是以位元組爲單位處理資料，很方便地也是美國標準交換碼(ASCII)的字元大小．

假定是你有一台連接到你 PC 的終端機．當你鍵入一個字元時，終端機把它交給它的轉換器(也是 UART 的一種)．該轉換器以指定的速率把這個位元組送到串列線路上，每次一個位元．在 PC 這一端，接收位元的 UART 拿走所有的位元，然後重新組成位元組並且放到緩衝區裡去．

有兩種不同的 UARTs 類型．你可能有聽過愚蠢的(dumb)UARTs - 8250 還有 16450，以及先進先出式 UARTs - 16550A．要了解它們的不同，首先讓我們來檢驗當 UART 送出或接收一個位元組時所發生的事．

這 UART 本身並不能對資料做什麼處理，它只是送出以及接收它． 每次送出或接收一個位元組時，CPU 都會從串列設備接到一個中斷． 然後 CPU 就把接收到的位元組從 UARTs 的緩衝區搬到記憶體的某處，或是提供另一個位元組給 UART 傳送． 8250 以及 16450 UARTs 只有一個位元組的緩衝區．這個意思是，每送出或接收一個位元組就會中斷 CPU 一次． 在低速的情況下這樣沒有問題．但是在高速的傳輸率下，CPU 會忙於處理 UART 而沒有時間照料其它工作． 在某些情況下，CPU 沒有足夠的餘裕及時處理中斷服務，該位元組將因資料如此快速地傳入而被覆寫．

這就是 16550A UARTs 有用的地方．這些晶片具有 16 個位元組的先進先出佇列． 這個意思是它在必須中斷 CPU 之前可以接收或轉換達 16 個位元組． 它不僅是能等，而且 CPU 現在也可以同時傳送全部 16 個位元組的資料． 雖然中斷界限(threshold)很少有設到 16 的，這對於其它只有一個位元組緩衝區的 UARTs 而言仍然是一個重要的優點． CPU 收到較少的中斷，因此可以自由地做其它事情．資料不會漏失，每個人都很高興． (也有 16550 這個 UART，但是它跟 16450 一樣因爲它有問題．)

一般來說， 8250 以及 16450 UARTs 應該能夠勝任高達 38400 bps 的工作速率． 在高於 38400 bps 的速率下，你可能開始見到資料漏失． 其它的 PC 作業系統(這裡的定義很寬鬆)，像 DOS 並非是多工的，所以它們有可能可以用 8250 或 16450s 卻應付的更好． 這就是爲什麼有些人並沒有看到資料漏失，直到他們改用 Linux 才發生．

無-UART(Non-UART)以及智慧型多埠卡使用數位處理機(DSP)晶片來處理額外的緩衝以及控制，如此更能減輕 CPU 的負擔． 例如，Cyclades Cyclom 及 Stallion EasyIO 這些卡使用 Cirrus Logic CD-1400 RISC 晶片，還有許多卡使用 80186 CPU 或更特別的 RISC CPU 來處理串列輸出入．

心裡要記得，這些愚蠢型的 UART 並非是壞的，只是它們不適合做高速傳輸． 當你將終端機或滑鼠連接到這些 UARTs 時應該不會遭遇困難． 但是對高速數據機而言，16550A 可以說是必須的．

你可以花多一點點的錢購買有 16550A UARTs 的串列卡，只要詢問電腦商該卡所使用的是什麼型號的 UARTs 即可． 或者你想升級你現在的卡，你可以只要簡單地購買 16550A 晶片並且置換你現在的 16450 UARTs 晶片． 它們的腳位相容(pin-to-pin compatible)．有些卡有 UARTs 插槽來作這個用途使用． 如果不是的話你可以焊接．注意，如果你有錢，買一張新卡將可能會省掉你許多麻煩，它們在 $50 元美金以下．

13 故障排除

13.1 我不斷地收到 **“line *NNN* of inittab invalid”**

要確定你對你的 `init` 版本使用正確的文法．這裡所沒有的一些 `init` 會在 `/etc/inittab` 檔案裡使用不同的文法． 要確定你對你的 `getty` 版本使用正確的文法．

13.2 當我試著撥號時，它說 **“/dev/cua*N*: Device or resource busy”**

這個問題可能發生在 DCD 或 DTR 沒有設定正確的時候． DCD 應該只在眞的有連線時(像是有人撥接進來)才被設定，而不是在 `getty` 持續監看該埠的時候． 檢查並確定你的數據機正確地配置成只在眞正連線時才設定 DCD． DTR 應該在任何東西使用或監看線路時設定，像是 `getty`, `kermit`, 或其它通訊程式．

另一個引起 “device busy” 錯誤的普遍原因是你設定你的串列埠使用一個已經被某種其它東西使用的中斷． 在每一個設備初始化的時後，它會要求 Linux 允許它使用它的硬體中斷． Linux 持續追蹤那個中斷被指定給誰，而如果你的中斷已經被佔用，你的設備將不能適當地初始化． 該設備眞的並沒有什麼

辦法告訴你發生的這件事，除了當你嘗試去使用它的時候，它會回應 “device-busy” 錯誤訊息． 檢查你所有的卡(串列，乙太網路，SCSI 界面等等)．找尋硬體中斷衝突的地方．

13.3 我持續接到 “Id S*N* respawning too fast: disabled for 5 minutes”

確定你的數據機有正確的配置．查看暫存器 `E` 和 `Q`． 這可能發生在你的數據機跟 `getty` 溝通的時候．確定你正確地從 `/etc/inittab` 呼叫 `getty`． 使用錯誤的文法或設備名稱將會引起嚴重的問題．以下法檢查你的 `/etc/gettydefs` 文法是否正確:

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

這也可能發生在 `uugetty` 初使化失敗時．參閱 13.11 (getty 或 uugetty 仍然無法運作) 一節．

13.4 串列設備很慢或是串列設備只能單向傳送

你的硬體中斷可能有衝突．確定沒有硬體中斷是被分享的． 檢查你所有的卡(串列卡, 乙太網路卡, SCSI 等等)確定你串列設備配的跳接設定以及 `setserial` 參數是正確的． 同時檢查 `/proc/ioports` 及 `/proc/interrups` 以確定是否有衝突發生．

13.5 我的數據機在某人斷線後癱瘓或是 `uugetty` 並沒有重新執行

這在 DTR 訊號掉下而你的數據機沒有重置時會發生． 這個問題在我身上發生的時候我看到我的 RD 跟 SD LEDs 瘋狂地閃爍．你需要讓你的數據機重置． 大部份的 Hayes 相容數據機使用 `&D3` 來做這件事，但是在我的 USR Courier 上，我得要設 `&D2` 以及 `S13=1`．查閱你的數據機手冊．

13.6 我將我的終端機連到我的 PC 上，但是在我輸入簽入名稱之後，它就鎖住不動

- 如果你使用 `getty`: 在你的 `/etc/gettydefs` 項目裡可能沒有設 `CLOCAL` 給終端機，而且可能你用的並不是完整的 null modem 連接線． 你需要 `CLOCAL` 來告訴 Linux 忽略數據機控制信號．它看起來像這裡這樣:

  ```
  # 38400 bps Dumb Terminal entry
  DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

  # 19200 bps Dumb Terminal entry
  DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

  # 9600 bps Dumb Terminal entry
  DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
  ```

  接下來，用 `kill` 砍掉該 `getty` 行程這樣新行程會以新的項目產生．

- 如果你使用 `agetty`: 加上 `-L` 旗標到你的 `/etc/initab` 中的 `agetty` 那行． 這會使得它忽略數據機控制信號．然後鍵入 `init q` 以重新執行 `init`． 這個項目看起來像這樣:

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

13.7 在高速下，我的數據機漏失資料

如果你嘗試在大於 38400 bps 的速率下使用你的數據機，而你並沒有 16550A UARTs 的話，你應該要將它們升級． 有關 UARTs 的說明參閱 12.3 (什麼是 UARTs?) 一節．

13.8 在系統啓動時，Linux 沒有依照我的配置回報串列設備．

這是事實．Linux 在系統啓動時並沒有做任何的 IRQ 偵測，它只做串列設備偵測． 所以，不要理會它所顯示有關硬體中斷的部份，因爲它只是假定使用標準的硬體中斷． 這是因爲硬體中斷偵測不可靠，而且可能被瞞騙而這樣做的．

所以即使我的 ttyS2 設在 IRQ5，我仍然看到

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

在 Linux 啓動時，你必須使用 setserial 來告訴 Linux 你所使用的硬體中斷． Linux 啓動後，你可以查看 /proc/interrupts 檔以了解眞正被配置的 IRQ 是什麼．

13.9 當我使用我的數據機叫用 Linux 機器時 rz 以及/或是 sz 不動

如果 Linux 在你嘗試傳送檔案時會尋找 /dev/modem 的話，查看 /etc/profile 以及 /etc/csh.cshrc．某些發行套件會在這些地方定義一些別名，最著名的是 Slackware． 這些別名擾亂了 zmodem 程式．把它們拿掉或是更正它們．

13.10 我的螢幕印出看起來很好玩的字元

這在當你把二進位資料送往螢幕的時候會發生在虛擬控制台上，或者有時候會發生在串列連線上． 修復這個問題的方法是輸入 echo ^v^[c．因爲控制字元之故，它是:

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

13.11 getty 或 uugetty 仍然無法運作

getty_ps 上有個 DEBUG 選項．編輯你的 /etc/conf.{uu}getty.ttyS*N* 配置檔並加上 DEBUG=*NNN*． 其中 *NNN* 是下列的數字組合之一，根據你想要偵測什麼錯誤而定:

```
D_OPT   001         option settings
D_DEF   002         defaults file processing
D_UTMP  004         utmp/wtmp processing
D_INIT  010         line initialization (INIT)
D_GTAB  020         gettytab file processing
D_RUN   040         other runtime diagnostics
D_RB    100         ringback debugging
D_LOCK  200         uugetty lockfile processing
D_SCH   400         schedule processing
D_ALL   777         everything
```

設定 DEBUG=010 是個開始的好地方·

如果你正在執行 syslogd 的話，偵錯資訊將會出現在你的記錄檔裡· 如果你沒有執行 syslogd 那麼 getty 的資訊將會出現在 /tmp/getty:tty*SN* 裡而 uugetty 的資訊則會出現在 /tmp/uugetty:tty*SN* 裡，而且它們也會被放到 /var/adm/getty.log 裡去· 查看偵錯資訊看看發生了什麼·最可能的是，你將得要調整一些你配置檔裡的參數，並且重新配置你的數據機·

你應該也去試試 mgetty·有些人在使用它時更爲幸運·

14 其它的資訊來源

- 下列指令的線上手冊(man pages): agetty(8), getty(1m), gettydefs(5), init(1), login(1), mgetty(8), setserial(8)
- 你的數據機手冊
- NET-3 HOWTO: 所有關於網路的東西，包括了 SLIP, CSLIP, 以及 PPP
- *PPP HOWTO*: 有助於使用 PPP
- Printing HOWTO: 如何設定串列印表機
- Term HOWTO: 所有你想知道關於 term 程式的事情
- UPS HOWTO: 設定連上你的串列埠的 UPS
- UUCP HOWTO: 建立 UUCP 的資訊
- Usenet 新聞群組:

  comp.os.linux.answers
  　　關於 Linux 的 FAQs, How-To's, READMEs 等等·
  comp.os.linux.hardware
  　　Linux 作業系統的硬體相容性·
  comp.os.linux.networking
  　　Linux 的網路及通訊·
  comp.os.linux.setup
  　　Linux 的安裝及系統管理·

- Linux 串列電子郵件列表(mailing list)·要加入的話，寄電子郵件到 *majordomo@vger.rutgers.edu*，文章內容寫 "subscribe linux-serial"· 如果你在內容上寫 "help"，你會得到一份輔助說明· 這個伺服器還提供許多其它的 Linux 列表·送出 "lists" 指令可以得到所有電子郵件列表的列表·
- 討論串列通訊及多埠串列卡的白皮書可自 Cyclades 的 *http://www.cyclades.com* 取得·
- Modem FAQs:
  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>
- 串列程式寫作: *Advanced Programming in the UNIX Environment* <http://heg-school.aw.com/cseng/authors/stevens/advanced/advanced.nclk>, by 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 銘謝

獨立撰寫這份 HOWTO 是不可能的事． 雖然這份 HOWTO 的內容大部份都是我寫的，我經常重寫這些感謝來維持許多在寫作方式以及內容上有貢獻的人． 感謝每個曾經貢獻經驗或建議的人，這些人的列表已經長得無法列出(超過一百人)． 特別感謝 Ted T'so 對於有關串列驅動程式問題的回答，還有負責維護 `getty_ps` 的 Kris Gleason，以及負責維護 `mgetty` 的 Gert Döring．

SERIAL HOWTO 結束