Linux Serial HOWTO

di Greg Hankins versione 1.12, Luglio 1998

Questo documento descrive le funzionalità delle porte seriali non descritte dal Text-Terminal-HOWTO o dal Serial-Programming-HOWTO. Riguarda principalmente getty, schede seriali multiporta e i modem. Aggiornato da David S.Lawyer *bf347@lafn.org* . Traduzione italiana a cura di Christopher Gabriel, *cgabriel@pluto.linux.it*

Indice

1 Introduzione

Questo è il Linux Serial HOWTO. Descrive in modo dettagliato aspetti sulle porte seriali che non sono trattati in altri HOWTO. Include anche come impostare un modem in ambiente linux, qualche trucchetto e una sezione per la risoluzione dei problemi. Esistono altri HOWTO per i terminali (rif. 1.2 (HOWTO correlati)). Questo documento è indirizzato a Linux per macchine con hardware x86, ma potrebbe essere di aiuto per altre architetture.

1.1 Copyright

1.2 HOWTO correlati sulle porte seriali

- *Serial-Programming-HOWTO* vi può aiutare a scrivere programmi in linguaggio C (o parte di questi) che possono gestire una porta seriale, e altro ancora.
- *Text-Terminal-HOWTO* contiene tutto ciò che riguarda i terminali seriali (FIXME: dumb terminals?).

1.3 Note

Io, David Lawyer, sono il nuovo mantenitore (Greg Hawkins è l'autore del documento originale). Questa è una piccola revisione e non ho controllato se tutte le nozioni qui esposte sono aggiornate. Questo comporta tanto lavoro. Per cortesia informatemi se c'è qualcosa di sbagliato e/o mancante. In futuro dividerò questo HOWTO in tre parti:

- Informazioni sui terminali da aggiungere al Text-Terminal-HOWTO
- Informazioni sui modem in un nuovo Modem HOWTO
- Ulteriori informazioni sulle porte seriali aggiunte a questo documento.

Tutto questo avverrà (spero) nell'agosto o nel Settembre 1998.

1.4 Nuove versioni di questo documento

Nuove versioni del Linux Serial HOWTO saranno disponibili a *ftp://sunsite.unc.edu:/pub/Linux/docs/HOWTO/Seria* e nei siti mirror. Esistono anche altri formati, come il PostScript e DVI, reperibili nella cartella `other-formats` . Il Serial-HOWTO è anche disponibile a *http://sunsite.unc.edu/LDP/HOWTO/Serial-HOWTO.html* e sarà inviato al newsgroup *comp.os.linux.answers* mensilmente.

1.5 Datemi un aiuto!

Per favore inviatemi qualunque domanda, commento, suggerimento, o materiale addizionale. Sono sempre curioso di sapere cosa ne pensate di questo HOWTO. E sono spesso allora ricerca di miglioramenti! Chiedetemi esattamente cosa non capite, o cosa potrebbe essere descritto in maniera più chiara. Potete contattarmi via email a *bf347@lafn.org (David Lawyer)* .

2 Hardware Seriale Supportato

Linux può lavorare sicuramente con il seguente hardware seriale.

2.1 Schede Seriali Standard per PC

- Le schede seriali standard per PC (COM1 - COM4), alle quali possono essere collegate periferiche esterne (modem, mouse seriali, ecc.). Da quando queste non vengono più inserite nei PC, dato che sono integrate alla scheda madre, sono piuttosto difficili da trovare nei negozi specializzati.
- Modem interni standard per PC (COM1 - COM4)
- Quickpath Systems Port-Folio 550e (consentono di utilizzare gli IRQ 3, 4, 5, 9, 10, 11, 12, e 15)

Nota: a causa di un conflitto di indirizzo, non potete usare la porta COM4 e la scheda video IBM8514 simultaneamente. È dovuto ad un errore presente nella scheda IBM8514.

2.2 Modem Plug And Play (PnP)

La maggior parte dei modem attualmente sono PnP e molti di questi (con qualche accorgimento) possono operare con Linux. La miglior scelta è prendere un modem che abbia dei jumper per disattivare la funzionalità PnP. In molti casi, se impostate i jumper per un determinato numero di porta COM e IRQ, la funzionalità PnP verrà disattivata. Potete anche disattivare il PnP dal BIOS del vostro computer. Se non potete fare tutto questo, allora provate così:

- Avviate Win95/98, configurate il modem, e controllate che queste impostazioni siano mantenute anche quando avviate Linux. Se non è così, allora:
- Avviate Win95/98, configurate il modem, e quindi fate un *warm boot* di Linux in modo che le impostazioni restino inalterate. Potete eseguire questo con il boot loader `loadlin`.
- Prendetevi i PnP Tools da *http://www.redhat.com/linux-info/pnp* .

2.3 Schede Seriali Multiporta Non-Intelligenti (con UART 8250/16450/16550A)

Sono dette anche Adattatori seriali.
=> setserial mostra maggiori dettagli sulla configurazione.

- AST FourPort e cloni (4 porte) *
- Accent Async-4 (4 porte) *
- Arnet Multiport-8 (8 porte)
- Bell Technologies HUB6 (6 porte)
- Boca BB-1004 (4 porte), BB-1008 (8 porte), BB-2016 (16 porte) *
- Boca IOAT66 (6 porte)
- Boca 2by4 (4 porte seriali, 2 porte parallele)
- Byterunner (basso costo)
- Computone ValuePort V4-ISA (compatibile con AST FourPort) *
- Digi PC/8 (8 porte)
- GTEK BBS-550 (8 porte) Guardatevi anche il mini-howto su questa scheda.
- HUB-6 See Bell Technologies.
- Longshine LCS-8880, Longshine LCS-8880+ (compatibile AST FourPort)
- Moxa C104, Moxa C104+ (compatibile AST FourPort)
- PC-COMM (4 porte)
- *Sealevel Systems* `http://www.sealevel.com`
  COMM-2 (2 porte), COMM-4 (4 porte) e COMM-8 (8 porte)

- SIIG I/O Expander 2S IO1812 (4 porte)
- STB-4COM (4 porte)
- Twincom ACI/550
- Usenet Serial Board II (4 porte) *

In generale, Linux supporta qualsiasi scheda che utilizza un UART 8250, 16450, 16550, 16550A, 16650 (o compatibili), o un modem interno che emuli una delle suddette UART.

Nota: la BB-1004 e BB-1008 non supportano linee DCD e RI, e quindi non sono utilizzabili dai modem in entrata. Lavorano molto bene in altri ambiti.

2.4 Schede Seriali Multiporta Intelligenti

Assicuratevi che sia disponibile un driver per Linux funzionante. Questa lista è da aggiornare.

- Comtrol RocketPort (36MHz ASIC; 4, 8, 16 o 32 porte)
  Contatti: *info@comtrol.com* oppure *http://www.comtrol.com*
  stato del driver: assistenza tecnica di Comtrol
  locazione del driver: *ftp://tsx-11.mit.edu/pub/linux/packages/comtrol*
- Computone IntelliPort II (16MHz 80186; 4, 8, o 16 porte),
  IntelliPort II EXpandable (20MHz 80186; 16 - 64 ports)
  contatto: Michael H. Warfield, *mhw@wittsend.atl.ga.us*
  stato del driver: pre-ALPHA
- Cyclades Cyclom-Y (Cirrus Logic CD1400 UARTs; 8 - 32 porte),
  Cyclom-Z (25MHz MIPS R3000; 8 - 128 porte)
  contatto: *sales@cyclades.com* oppure *http://www.cyclades.com*
  stato del driver: assistenza tecnica di Cyclades
  locazione del driver: *ftp://ftp.cyclades.com/pub/cyclades* e incluso nel Kernel Linux sin dalla versione 1.1.75
- Decision PCCOM8 (8 porte)
  contatto: *pccom8@signum.se*
  locazione del driver: *ftp://ftp.signum.se/pub/pccom8*
- Digi PC/Xi (12.5MHz 80186; 4, 8, o 16 porte),
  PC/Xe (12.5/16MHz 80186; 2, 4, o 8 porte),
  PC/Xr (16MHz IDT3041; 4 o 8 porte),
  PC/Xem (20MHz IDT3051; 8 - 64 porte)
  contatto: *sales@dgii.com* oppure *http://www.dgii.com*
  stato del driver: ssistenza tecnica di Digi
  locazione del driver: *ftp://ftp.dgii.com/drivers/linux* e incluso nel Kernel Linux sin dalla versione 2.0
- Digi COM/Xi (10MHz 80188; 4 o 8 porte)
  contatto: Simon Park, *si@wimpol.demon.co.uk*
  stato del driver: ALPHA
  nota: Simon è spesso assente per mesi per motivi di lavoro e non può rispondere alle email. Mark Hatle, *fray@krypton.mankato.msus.edu* si è gentilmente reso disponibile per farvi avere i driver, se ne avete bisogno. Mark non mantiene il driver e non fa assistenza.

- Equinox SuperSerial Technology (30MHz ASIC; 2 - 128 porte)
  contatto: *sales@equinox.com* oppure *http://www.equinox.com*
  stato del driver: assistenza tecnica di Equinox
  locazione del driver: *ftp://ftp.equinox.com/library/sst*

- GTEK Cyclone (16C654 UARTs; 6, 16 e 32 porte),
  SmartCard (24MHz Dallas DS80C320; 8 porte),
  BlackBoard-8A (16C654 UARTs; 8 porte),
  PCSS (15/24MHz 8032; 8 porte)
  contatto: *spot@gtek.com* oppure *http://www.gtek.com*
  stato del driver: assistenza tecnica di GTEK
  locazione del driver: *ftp://ftp.gtek.com/pub*

- Hayes ESP (COM-bic; 1 - 8 porte)
  contatto: Andrew J. Robinson, *arobinso@nyx.net* oppure *http://www.nyx.net/~arobinso*
  stato del driver: assistenza dell'autore
  locazionde del driver: *http://www.nyx.net/~arobinso* e incluso nel Kernel Linux dalla versione 2.1.15

- Maxpeed SS (Toshiba; 4, 8 e 16 porte)
  contatto: *info@maxpeed.com* oppure *http://www.maxpeed.com*
  stato del driver: assistenza tecnica di Maxpeed
  locazione del driver: *ftp://maxpeed.com/pub/ss*

- Moxa C218 (12MHz 80286; 8 porte),
  Moxa C320 (40MHz TMS320; 8 - 32 porte)
  contatto: *info@moxa.com.tw* oppure *http://www.moxa.com.tw*
  stato del driver: assistenza di Moxa
  locazione del driver: *ftp://ftp.moxa.com.tw/drivers/c218-320/linux*

- SDL RISCom/8 (Cirrus Logic CD180; 8 porte)
  contatto: *sales@sdlcomm.com* oppure *http://www.sdlcomm.com*
  stato del driver: assistenza di SDL
  locazione del driver: *ftp://ftp.sdlcomm.com/pub/drivers*

- Specialix SIO (20MHz Zilog Z280; 4 - 32 porte),
  XIO (25MHz Zilog Z280; 4 - 32 porte)
  contatto: Simon Allen, *simonallen@cix.compulink.co.uk*
  stato del driver: BETA
  locazione del driver: *ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial*

- Stallion EasyIO-4 (4 ports), EasyIO-8 (8 porte), e
  EasyConnection (8 - 32 porte) - ognuna con Cirrus Logic CD1400 UARTs,
  Stallion (8MHz 80186 CPU; 8 o 16 porte),
  Brumby (10/12 MHz 80186 CPU; 4, 8 o 16 porte),
  ONboard (16MHz 80186 CPU; 4, 8, 12, 16 o 32 porte),
  EasyConnection 8/64 (25MHz 80186 CPU; 8 - 64 porte)
  contatto: *sales@stallion.com* oppure *http://www.stallion.com*
  stato del driver: assistenza di Stallion
  locazione del driver: *ftp://ftp.stallion.com/drivers/ata5/Linux* e incluso nel Kernel di Linux dalla versione 1.3.27

Una relazione sui prodotti di Comtrol, Cyclades, Digi e Stallion è stata inserita nel numero di Giugno 1995 del *Linux Journal*. L'articolo è disponibile a *http://www.ssc.com/lj/issue14* .

2.5 Hardware seriale non supportato

2.5.1 I Winmodem non funzionano

I modem che vengono definiti Winmodem, ad esempio il USR Sportster Winmodem e l'IBM Aptiva MWAVE, *non* sono supportati da Linux. Questi modem hanno un design proprietario che richiede speciali driver per Windows. Evitate questo tipo di modem.

2.5.2 I driver per Rockwell (RPI) possono funzionare.

I modem che richiedono i driver Rockwell RPI non sono completamente funzionanti dato che i driver software non funzionano con Linux. La RPI comprime la trasmissione e ne corregge gli errori usando software MS attraverso la CPU del vostro computer. Se desiderate utilizzare questi modem senza l'uso dell'RPI (quindi senza compressione né correzione di errore) allora potete facilmente disabilitare la RPI inviando al modem (attraverso una string di inizializzazione) il comando appropriato: +HO

3 Che cosa rappresentano i nomi delle porte seriali?

Una porta di I/O è un mezzo che permette l'ingresso e l'uscita di dati da un computer. Esistono svariati tipi di porte I/O come, ad esempio, le porte seriali, parallele, controller per disk drive, schede ethernet, ecc. Qua tratteremo le porte seriali, dato che i modem e i terminali sono periferiche seriali. Ogni porta seriale deve avere un indirizzo di I/O, e un interrupt (IRQ). Esistono quattro porte seriali corrispondenti a COM1 - COM4:

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

Se Linux non riconosce nessuna porta seriale quando si avvia, allora assicuratevi che il supporto seriali sia attivato e compilato nel kernel. In questo documento, mi riferisco a COM1 come `ttyS0`, COM2 come `ttyS1`, COM3 come `ttyS2`, e COM4 come `ttyS3`. Da notare il fatto che per default Linux utilizza gli stessi indirizzi IRQ per più porte. Non potete usare tutte le porte con questa configurazione, e dovete re-assegnare le porte con differenti indirizzi IRQ. Potete fare riferimento alla sezione 9 (Posso usare più di due porte seriali?) per impostare gli IRQ.

3.1 Periferiche: Modem, Mouse.

In qualche installazione vengono create due periferiche addizionali: `/dev/modem` per il vostro modem e `/dev/mouse` per il mouse. Tutte e due sono link simbolici alla periferica appropriata in `/dev` che avete specificato durante l'installazione.

Ci sono state discussioni sul merito di `/dev/mouse` e `/dev/modem`. Io scoraggio apertamente l'utilizzo di questi link. Ad esempio, se desiderate utilizzare il vostro modem per chiamare all'esterno potrete avere problemi poiché il lock file può non funziona correttamente con `/dev/modem`. Utilizzateli se li trovate comodi, ma *assicuratevi che puntino alla periferica corretta*. In ogni caso, se cambiate o rimuovete questi link, qualche applicazione (`minicom`per esempio) andranno configurate nuovamente.

3.2 La periferica cua

Ogni periferica tty ha una corrispondere periferica cua. Si è discusso a lungo circa la rimozione delle cua poiché forse è meglio utilizzare le ttyS. La principale differenza tra cua e ttyS è che per un comune comando open (senza argomenti) all'interno di una applicazione, la cua aprirà ugualmente la porta anche se i segnali di controllo del modem (come il DCD) non lo permettono (stty deve essere impostato per controllare i segnali del modem). In una situazione del genere una porta ttyS si rifiuterebbe di aprire la porta, ma può essere forzata a farlo con parametri specifici ai comandi.

3.3 Le porte seriali e la numerazione in /dev

```
/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
```

Da notare che tutte le distribuzioni devono avere queste periferiche già impostate correttamente (a meno che cua non venga abolita). Potete verificare questo con:

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

3.3.1 Creare periferiche in /dev

Se non trovate una periferica allora dovete crearla con il comando `mknod`. Se per esempio volete creare una periferica per `ttyS0`:

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

Potete usare anche lo script `MAKEDEV`, che si trova in `/dev`. Questo semplifica la creazione di periferiche. Un altro esempio, sempre per creare una `ttyS0`:

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

Questo script si occupa di creare le periferiche di ingresso e di uscita, e impostare correttamente i permessi.

3.4 Piccole note per Schede Multiporta Non Intelligenti

La periferica che viene utilizzata dalla vostra scheda multiporta dipende proprio dal tipo di scheda installata. Alcune di queste sono dettagliatamente descritte in `rc.serial` o in `0setserial` che si trova nel pacchetto `setserial`. Raccomando vivamente di prendere la più recente versione di `setserial` se state provando ad installare una scheda multiporta. Probabilmente avrete bisogno di creare queste periferiche. Potete utilizzare sia il comando `mknod` sia lo script `MAKEDEV`. Le periferiche per schede multiporta sono create aggiungendo "64 + numero porta". Quindi, se volete creare una periferica del tipo `ttyS17`, dovrete digitare:

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

Notate che "64 + 17 = 81". Usando invece lo script `MAKEDEV`, dovrete digitare:

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

Nota: il manuale SIIG per la IO1812 che descrive la COM5-COM8 è errato. Devono essere COM5=0x250, COM6=0x258, COM7=0x260, e COM8=0x268.

Nota: il registro di stato per l'interrupt della Digi PC/8 e'0x140.

Nota: per una AST Fourport dovrete specificare `skip_test` in `rc.serial`.

3.5 Note per Schede Multiporta Intelligenti

Leggete le informazioni che trovate assieme al driver. Queste schede utilizzano particolari periferiche, non quelle predefinite. Queste informazioni variano dipendentemente dall'hardware.

4 Programmi interessanti di cui dovete essere al corrente

4.1 Cosa è `getty`?

`getty` è un applicativo che gestisce i processi di login quando accedete ad un sistema Unix. Dovrete utilizzare `getty` se desiderate accedere al vostro Linux box con un modem. Al tempo stesso, non ne avete bisogno se chiamate all'esterno dallo stesso box. Esistono attualmente tre versioni che sono comunemente utilizzate con Linux: `getty_ps`, `mgetty` e `agetty`. La sintassi per questi programmi è differente, quindi controllate e assicuratevi di utilizzare la corretta sintassi per qualunque `getty` decidiate di utilizzare.

4.1.1 `getty_ps`

Molte distribuzioni hanno il pacchetto `getty_ps` installato (Debian usava agetty, ma adesso è passata a mgetty). Questo contiene due applicativi: `getty`, il quale è utilizzato per le console e i terminali, e `uugetty` per i modem. Io uso questa versione di `getty`, quindi è quella su cui porrò maggiore attenzione.

4.1.2 `mgetty`

`mgetty` è una versione di `getty` usata per lo più con i modem. Può essere utilizzata per terminali cablati, ma la documentazione è al 99% incentrata sui modem. Oltre a permettere le connessioni esterne, `mgetty` fornisce il supporto per i FAX e il riconoscimento automatico del PPP. La documentazione di `mgetty` (fornita in formato texinfo) è buona, e non necessita di aggiunte da parte mia. Fate riferimento a quella documentazione per le istruzioni di installazione. Potete trovare maggiori informazioni circa `mgetty` a *http://www.leo.org/~doering/mgetty/* .

4.1.3 `agetty`

`agetty` è la terza variante di `getty`. È una semplice, completamente funzionale, implementazione di `getty` creata appositamente per console virtuali o terminali piuttosto che modem .

4.2 Cosa è `setserial`?

`setserial` è un applicativo per vi permette di controllare e modificare i vari attributi di una periferica seriale, inclusi l'indirizzo della porta, l'interrupt e altre opzioni varie. Potete sapere quale versione avete, lanciando `setserial` senza argomenti. Se l'argomento fosse, per esempio, /dev/ttyS0, allora riceverete informazioni circa questa periferica.

Quando il vostro sistema Linux viene avviato, sono configurate solo le periferiche `ttyS{0-3}`, usando gli IRQ standard 4 e 3. Quindi, se avete altre periferiche seriali fornite da altre schede o se `ttyS{0-3}` hanno un IRQ non-standard, allora *dovete* usare `setserial` in modo da configurare queste porte seriali. Per la lista completa di opzioni consultate la pagina `man`.

5 Come posso effettuare una chiamata con il mio modem?

5.1 Requisiti Hardware

Potete utilizzare un modem sia esterno che interno. Quello interno è più economico e non occupa spazio sulla scrivania o sul tavolo. Quello esterno invece è più semplice da installare e ha delle luci che permettono di capire cosa sta succedendo. Devono essere spenti manualmente quando non vengono usati e continuano a consumare un piccolo quantitativo di energia elettrica anche una volta spenti.

5.1.1 Modem Esterni

Prima di tutto, assicuratevi di avere il cavo giusto. Il vostro modem richiede un cavo pin-to-pin senza i file intrecciati. Ogni rivenditore di computer dovrebbe averli. Assicuratevi inolte che abbia il giusto connettore. Se state usando una porta seriale DB25, allora sarà sempre un connettore maschio DB25. Non confondetevi con la porta parallela, che ha un connettore femmina DB25. Collegate a questo punto il vostro modem ad una delle porte seriali. Consultate il manuale del vostro modem su come farlo se avete bisogno di aiuto.

5.1.2 Modem Interni

Per un modem interno non avrete bisogno di un cavo. Un modem interno non necessita di una porta seriale, ne ha una inserita al suo interno. Tutto ciò di cui avete bisogno è configurare gli indirizzi della porta I/O. Consultate il manuale del vostro modem se non sapete cosa fare. Inoltre, potete leggere la sezione 9 (Posso usare più di due porte seriali?) se avete bisogno di scegliere un IRQ o un indirizzo.

Su alcune schede madri potete disattivare le porte seriali che il modem rimpiazza in modo tale da evitare conflitti. Questo può essere eseguito utilizzando i vari jumper presenti sulla scheda oppure dalle impostazioni del BIOS, dipende dal tipo di scheda madre. Consultate il manuale di questa.

Se siete in possesso di una scheda video IBM8514, allora dovete sapere che c'è un errore. Potrete avere problemi se tentate di utilizzare il vostro modem interno sul `ttyS3`. Se Linux non riconosce la vostra scheda modem interna su `ttyS3`, potete utilizzare `setserial` e il modem funzionerà correttamente. I modem interni su `ttyS{0-2}` non dovrebbero avere nessuno problema nell'essere riconosciuti. Linux non esegue nessuna auto-configurazione su `ttyS3` per via di questo problema con questo tipo di scheda video.

5.2 Parlare al vostro modem

Usate `kermit`, `minicom` o qualche altro applicativo di comunicazione per sperimentare la vostra configurazione, prima di passare all'utilizzo di applicazioni più complesse, tipo SLIP o PPP. Potete trovare le versioni più recenti di `kermit` a *http://www.columbia.edu/kermit/* . Per esempio, se il vostro modem è connesso a `ttyS3`, e la velocità è 115200 bps, dovrete fare così:

```
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                           ; digitate questo e premete INVIO
OK                                 ; il modem dovrebbe rispondere così
```

Se il vostro risponde ai comandi AT, allora sicuramente funzionerà con Linux. Adesso provate a chiamare un altro modem digitando:

```
ATDT123456
```

dove 123456 è il numero telefonico. Usate ATDP invece di ATDT se avete una linea a impulsi. Se la chiamata viene effettuata, allora il vostro modem funziona correttamente.

Per tornare al prompt di `kermit` premete CTRL+\ e poi C:

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

Questo è solo un piccolo test molto rude su come effettuare una chiamata manualmente. Il metodo normale è di utilizzare `kermit` in modo che effettui la chiamata automaticamente, utilizzando il suo database interno di numeri telefonici. Ad esempio, se utilizzate 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        ; Selezionare il tipo di modem
C-Kermit>set line /dev/ttyS3       ; Selezionare la porta di comunicazione
C-Kermit>set speed 115200          ; Impostare la velocità di trasmissione
C-Kermit>dial 7654321              ; Chiamare
 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:
```

fate riferimento alla sezione 11 (Programmi di Comunicazione) se avete bisogno di ulteriori informazioni o applicativi da utilizzare.

Quando chiamate con il vostro modem, impostate la velocità della porta seriale la più alta possibile, sempre che il vostro modem la supporti (non oltre i 155200 bps). Questa non è la stessa velocità del modem, come ad esempio 56.000 bps. Alcune versioni di Linux con una versione `libc` maggiore di 5.x hanno il supporto per 115.200 bps.

5.3 Configurare il modem per chiamare all'esterno

Se utilizzate il modem unicamente per chiamare, potete configurarlo come più vi piace. Se invece siete intenzionati anche a ricevere chiamate, allora *dovete* configurare il vostro modem alla stessa velocità a cui intendete impostare `getty`. Comunque sia, la maggior parte dei modem oggigiorno impostano automaticamente la velocità alla stessa della porta di comunicazione. Chi utilizza `getty` alla velocità di 115.200 bps ha automaticamente impostato il modem per quella velocità, quando `getty` invia a questo la stringa di inizializzazione alla porta a cui è connesso il modem. In generale, consultate il manuale del vostro modem per impostare una velocità ottimale.

5.4 Controllo di Flusso Hardware

Se il vostro modem supporta il controllo di flusso hardware (RTS/CTS), vi raccomando caldamente di utilizzarlo. È particolarmente importante per i modem che supportano la compressione dei dati. In primo luogo dovete attivare il controllo di flusso RTS/CTS sulla porta seriale stessa. Il miglior modo è di farlo al boot, ad esempio in `/etc/rc.d/rc.local` oppure `/etc/rc.d/rc.serial`. Assicuratevi che questi file siano eseguiti la file principale, `rc.sysinit`! Dovete seguire il seguente procedimento su ogni porta sulla quale desiderate installare il supporto per il controllo di flusso hardware:

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

Se non è impostato per default, dovete impostare il controllo di flusso RTS/CTS anche sul vostro modem. Il vostro programma di comunicazione può impostarlo per voi. Consultate il manuale del vostro modem per maggior informazioni.

6 Come posso effettuare chiamate e riceverle con il modem usando `getty_ps`?

Configurate il vostro modem in modo che possa effettuare chiamate correttamente. Se non avete letto la sezione 5 (Come posso effettuare una chiamata con il mio modem?) allora *leggetela adesso*! Contiene informazioni *molto* importanti sulla configurazione. Non importa che leggiate questa sezione se desiderate usare il modem esclusivamente per *effettuare* chiamate.

6.1 Configurare il modem per chiamare e ricevere

Per effettuare chiamate oppure riceverle dovete impostare il vostro modem in un modo particolare (ancora usando i comandi `AT` sul vostro modem) :

```
E1      command echo ON
Q0      result codes are reported
V1      verbose ON
```

```
S0=0    never answer (uugetty interpreta questo come una
        opzione WAITFOR )
```

Se non impostate questi parametri correttamente, la vostra stringa `INIT` potrebbe fallire, bloccando tutta la procedura. Ma ecco qualcosa di più circa i file di configurazione...

```
&C1     DCD è attivato solo dopo la connessione
&S0     DSR è sempre attivato
DTR on/off resetta il modem (dipende dalla casa produttrice - RTFM )
```

Questi parametri impostano il comportamente del modem quando una chiamata inizia e/o finisce.

Se il vostro modem non supporta l'inserimento di un profilo, potete impostare questi parametri attraverso la stringa `INIT` nel vostro file di configurazione. Leggete più sotto. Alcuni modem hanno alcuni DIP switch per effettuare queste modifiche. Assicuratevi che anche questi, se è il vostro caso, siano impostati correttamente.

Ho iniziato a collezionare configurazioni per svariati tipi di modem. Al momento però non sono molti.. quindi inviatemeli! aspetto qualcosa da voi! Potete trovare quelli giè presenti a *ftp://ftp.cc.gatech.edu/pub/people/gregh/modem-configs* .

Nota: per far resettare correttamente il mio modem USR Courier V.34 quando cade il DTR, ho dovuto impostare `&D2` e `S13=1` (questo assegna il bit 0 del registro S13). È stato confermato che questo funziona anche sul modem USR Sportster V.34 .

Nota: alcuni modem Supra gestisco il DCD in modo differente da altri modem. Se state utilizzando un modem Supra, provate a impostare `&C0` e *non* `&C1`. Dovete anche settare `&D2` per gestire correttamente DTR.

6.2 Installazione di getty_ps

Prendete la versione più recente da *sunsite.unc.edu:/pub/Linux/system/serial* . In particolare, se desiderate usare alte velocità (57600 e 115200 bps) dovete prendere la versione 2.0.7j o successive. Inoltre dovete anche avere la libc versione 5.x o successiva.

Per default, `getty_ps` è configurato per essere compatibile con il Linux FSSTND (File System Standard), il che significa che i file binari saranno in `/sbin/`, e i file di configurazione si chiameranno `/etc/conf.`{uu}`getty.ttyS`*N*. Questo non compare nella documentazione! Inoltre darà per scontato che i lock file andranno in `/var/lock`. Assicuratevi di avere la directory `/var/lock/`.

Se non desiderate la compatibilità con FSSTND, i file binari andranno in `/etc`, mentre quelli di configurazione in `/etc/default/`{uu}`getty.ttyS`*N*, e i lock file si andranno in `/usr/spool/uucp`. Vi raccomando di effettuare queste procedure come descritte se state usando UUCP, perché UUCP potrà avere problemi se muovete i lock file dalla posizione predefinita.

`getty_ps` può inoltre usare `syslogd` per registrare tutti i messaggi. Consultate le pagine man `syslogd(1)` e `syslogd.conf(5)` su come impostare `syslogd`, se ancora non lo avete installato. I messaggi sono registrati con priorità LOG_AUTH, gli errori usano LOG_ERR, e quelli di debugging LOG_DEBUG. Se non desiderate usare `syslogd` potete editare `tune.h` fra i file sorgenti di `getty_ps` per utilizzare un file per registrare i messaggi, che per default è `/var/adm/getty.log`.

Decidete se siete interessati ad utilizzare FSSTND e anche syslogd, potete scegliere di utilizzare una combinazione dei due. Modificate il `Makefile`, `tune.h` e `config.h` per confermare le vostre decisioni. Quindi compilate e installate seguendo le istruzioni incluse nel pacchetto dei sorgenti.

D'ora in avanti, tutti i riferimenti a `getty` saranno per `getty_ps`. Riferimenti invece a `uugetty` saranno per `uugetty` incluso nel pacchetto `getty_ps`. Queste istruzioni non funzionano con `mgetty` o `agetty`.

7 uugetty

7.1 Impostare uugetty

Assicuratevi di avere una periferica di entrata e una di uscita per la porta seriale alla quale è collegato il vostro modem. Se questo si trova su `ttyS3` dovete avere le periferiche `/dev/cua3` e `/dev/ttyS3`. Se non avete le periferiche corrette, leggete la sezione 3.3.1 (Creare periferiche in /dev) circa la creazione di queste periferiche, e createle. Se desiderate essere in grado di effettuare chiamate all'esterno mentre `uugetty` controlla la porta per eventuali chiamate in arrivo, usate la periferica `/dev/cua`N invece di `/dev/ttyS`N.

`uugetty` effettua importanti controlli sui lock file. Aggiornate `/etc/gettydefs` per aggiungere un voce per il vostro modem. Quando avete modificato `/etc/gettydefs`, potete verificare la correttezza della sintassi eseguendo:

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

7.1.1 Modem moderni

Se avete un modem a 9600 bps o più veloce con compressione dei dati, potete impostare la velocità della porta seriale. Per esempio:

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

Se volete che il vostro modem effettui il controllo di flusso hardware RTS/CTS, potete aggiungere `CRTSCTS` alla voce:

```
# 115200 velocità impostata con controllo di flusso hardware
F115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login: #F115200
```

7.1.2 Modem lenti e vecchiotti

Se siete in possesso di modem non veloci (inferiori a 9600 bps) allora, invece un una sola linea per una singola velocità, avrete bisogno di alcune linee per provare alcune impostazioni. Da notare il che queste linee sono collegate fra loro con l'ultima parola, ad esempio #38400. Sono necessarie delle linee vuote fra ciascuna voce del file di configurazione.

```
# Impostazioni del 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
```

7.1.3 Messaggio di benvenuto

Se lo desiderate, potete impostare `uugetty` in modo tale che visualizzi messaggi interessanti al momento del login. Nel mio esempio ho inserito il nome del sistema, la linea seriale, e la velocità utilizzata. Potete aggiungere molte altre cose:

```
@B    La velocità della linea in bps.
@D    La data corrente, espressa in MM/DD/YY.
@L    La linea seriale sulla quale si trova getty.
@S    Il nome del sistema.
@T    L'ora corrente, espressa in  HH:MM:SS (24 ore).
@U    Il numero di utenti attualmente collegati. E in pratica
      un conteggio effettuato in base al file /etc/utmp.
@V    Il valore di VERSION
Per visualizzare un carattere '@'singolo, utilizzate sia '/@'
che '/@@'.
```

7.2 Personalizzare `uugetty`

Esistono molti parametri che potete impostare per ciascuna porta seriale. Questi sono implementati in file di configurazione separati per ciascuna porta. Il file `/etc/conf.uugetty` sarà utilizzato da tutte le sessioni di `uugetty`, e `/etc/etc.uugetty.ttyS`*N* sarà utilizzato solo per quella specifica porta. Alcuni esempi di questi file di configurazione posso essere trovati fra i sorgenti di `getty_ps`, che si possono trovare praticamente in ogni distribuzione di Linux. Per motivi di spazio non sono stati trascritti qui. Nota: se state utilizzando versioni più vecchie di `getty` (precendenti alla 2.0.7e), oppure non state utilizzando FSSSTND, allora il file di configurazione sarà `/etc/default/uugetty.ttyS`*N*. Il `/etc/conf.uugetty.ttyS3` è così:

```
ALTLOCK=cua3
ALTLINE=cua3
# Linea da inizializzare
INITLINE=cua3
# timeout di disconnessione se inattiva
TIMEOUT=60
# stringa di inizializzazione del modem
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# questa riga imposta il tempo di ritardo prima di inviare
# il messaggio di benvenuto del login
DELAY=1
#DEBUG=010
```

Aggiugete le seguenti linee nel vostro `/etc/inittab`, in modo tale che `uugetty` sia in esecuzione sulla vostra porta seriale (effettuate le modifiche in modo da adattare questo al vostro specifico caso - file di configurazione, velocità e tipo di terminale):

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

Riavviate `init`:

```
linux# init q
```

Per le impostazioni sulla velocità nel vostro `/etc/inittab` vi consiglio di utilizzare la velocità più alta in bps supportata dal vostro hardware.

Adesso Linux controlla continuamente la porta seriale in attesa di connessioni. Chiamate da un'altra macchina e collegatevi al vostro sistema Linux.

`uugetty` ha molte altre opzioni, leggete il manuale di `getty(1m)` per una descrizione pì dettagliata. Fra le tante opzioni c'è anche quella di richiamata.

8 Come posso impostare un terminale connesso al mio PC?

Leggete il *Text-Terminal-HOWTO* per informazioni più dettagliate sui terminali. Le istruzioni in questa sezione descrivono come connettere un terminale, così come altri computer al vostro sistema Linux.

8.1 Requisiti Hardware

Assicuratevi di avere il cablaggio corretto. Un cavo null-modem comprato in un negozio di informatica può funzionare, ma quasi sicuramente non funzionerà il controllo di flusso hardware. Ma deve essere un cavo *null modem*! Molti negozi di computer lo chiamano cavo per stampanti seriali. Assicuratevi inoltre di stare usando la porta seriale, una DB25 o DB9 maschio, non la quella parallela (femmina, DB25 o Centronics).

Per il connettore DB25, avete bisogno al minino di questo:

```
PC maschio DB25                          Terminale 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
```

Questo cavo null modem funziona fra due computer che utilizzano il signale di handshake hardware RTS/CTS, ma molti terminali utilizzano l'handshake DTR che non può essere impostato con linux tramite stty. Una soluzione è di impostare stty crtscts e connettere il pin 20 del terminale (DTR) con il pin 5 del computer (CTS).

```
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
```

Se sulla propria porta seriale si ha un connettore DB9, si provi quanto segue:

```
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
```

In alternativa, un cavo null modem DB9-DB25 (come detto non può operare con un terminale con handshake hardware):

```
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)
```

(Sì, i pin 2 e 4 hanno *veramente* significati opposti tra i connettori DB9 e quelli DB25!)

Se non state utilizzando un cavo completamente null modem, forse potete utilizzare il seguente trucchetto: sul connettore dalla parte del vostro computer collegate il RTS e il CTS assieme, e collegate anche DSR, DCD e DTR assieme. In questo modo, quando uno dei due computer richiede un signal di handshake lo otterrà (da se stesso!).

Adesso che avete il cavo giusto, collegate il vostro terminale al vostro computer. Se potete, impostate il terminale in modo da ignorare i segnali di controllo del modem. Provate ad usare 9600 bps, 8 data bits, 1 stop bit, no parità nella configurazione del terminale (oppure attivate la parità se volete essere un po' più sofisticati).

8.2 Impostare `getty` (componente di `getty_ps`)

Consultate il vostro manuale di *getty* per altre applicazioni come ad esempio *agetty*. Aggiungete le impostazioni per il vostro terminale per `getty` se non sono già esistenti in `/etc/gettydefs`:

```
# Terminale Stupido a 38400 bps
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

# Terminale Stupido a 19200 bps
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# Terminale Stupido a 9600 bps
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

Se lo desiderate potete far stampare a video da `getty` informazioni interessanti sul login. Nel mio esempio ho inserito il nome del sistema e la porta seriale. Potete aggiungere molte altre cose:

```
@B    La velocità corrente in bps.
@D    La data corrente, nel formato MM/DD/YY.
@L    La porta seriale sulla quale è in esecuzione getty.
```

```
@S    Il nome del sistema.
@T    l'ora corrente, nel formato HH:MM:SS ( 24 ore ).
@U    IL numero degli utenti attualmente collegati.
@V    Il valore di VERSION.

Per visualizzare un singolo carattere '@', potete usare
sia '\@' sia '@@'.
```

Una volta modificato `/etc/gettydefs`, potete verificare che la sintassi sia corretta digitando:

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

Assicuratevi che non vi siano file di configurazione di `getty` oppure `uugetty` per la porta seriale alla quale il vostro terminale è collegato (`/etc/default/{uu}getty.ttyS`*N* oppure `/etc/conf.{uu}getty.ttyS`*N*), poiché può molto probabilmente interferire nell'esecuzione di `getty` su un terminale. Rimuovere questi file se esistono.

Editate il vostro file `/etc/inittab` per lanciare `getty` sulla porta seriale (sostituendo con le corrette informazioni sul vostro sistema - porta, velocità e tipologia del terminale):

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

Riavviate `init`:

```
linux# init q
```

A questo punto dovreste essere in grado di vedere un prompt di login sul terminale. Probabilmente dovrete premere *invio* per ottenere l'attenzione del terminale.

8.3 Suggerimenti per configurare un PC come terminale seriale

Molte persone configurano i loro PC come terminali connessi a un box linux. Per esempio, dei vecchi PC 8088 o 80286 sono perfetti per questo scopo. Tutto ciò di cui avete bisogno è un disco di boot DOS adatto al vostro tipo di terminale, e un programma di comunicazione da avviare. `kermit` è ideale a questo scopo. Potete trovare versioni precompilate di `kermit` per ogni sistema operativo esistente a *http://www.columbia.edu/kermit/* . Altri programmi DOS molto popolari come `procomm` funzionano molto bene. Assicuratevi di impostare corretamente le informazioni sulla porta seriale nelle impostazioni del programma di comunicazione del vostro terminale.

9 Posso usare più di due porte seriali?

Non avete bisogno di leggere questa sezione a meno che non intendiate utilizzare tre o piú periferiche seriali (assumendo il fatto che *non* avete una porta seriale).

Il numero delle porte seriali è limitato dal numero di interrupt (IRQ) e indirizzi di I/O che possiamo utilizzare. Queste non è una limitazione di Linux, ma del bus del PC. Ad ogni porta seriale *deve* essere assegnato un proprio indirizzo e dovrebbe avere il proprio interrupt. Se due periferiche condividono lo stesso interrupt, può funzionare correttamente se le due periferiche non operano contemporaneamente (nella maggior parte dei casi sono i due programmi che non devono essere in esecuzione assieme). Una periferica seriale può essere un porta seriale, un modem interno, oppure una scheda seriale multiporta.

Una scheda seriale multiporta sono appositamente progettate per avere porte seriali multiple che condividono lo stesso interrupt per tutte le porte sulla scheda. Linux riceve i dati da queste usando differenti indirizzi di I/O per ogni porta sulla scheda.

9.1 Scegliere gli interrupt per le periferiche seriali

Il vostro PC normalmente ha `ttyS0` e `ttyS2` associate all'IRQ 4, e `ttyS1` e `ttyS3` all'IRQ 3. Potete visualizzare gli IRQ utilizzati digitando *setserial /dev/ttyS2* ecc ecc. Il contenuto di `/proc/interrupts` mostra alcuni di questi. Per usare più di una periferica seriale dovrete riassegnare gli interrupt. Una buona scelta solitamente è quella di riutilizzare l'interrupt della porta parallela. Il vostro PC normalmente ha IRQ 5 e IRQ 7 per la porta parallela, ma poche persone usano due porte di questo genere. Potete riassegnare uno di questi interrupt ad una porta seriale e continuare ad usare felicemente la vostra porta parallela. Avete bisogno dell'applicazione `setserial` per eseguire questa operazione. In aggiunta, dovete fare alcuni tentativi con i jumper della vostra scheda, consultate il manuale di questa per maggiori informazioni. Impostate i jumper per l'IRQ che desiderate per ciascuna porta.

Dovete impostare le cose in modo che esista un solo interrupt per ciascuna periferica seriale. Ecco come Greg ha configurato il suo sistema in `/etc/rc.d/rc.local` - dovete eseguire le modifiche in un file che viene eseguito all'avvio del sistema:

```
/sbin/setserial /dev/ttyS0 irq 3        # il mio mouse serial
/sbin/setserial /dev/ttyS1 irq 4        # il mio terminale stupido Wyse
/sbin/setserial /dev/ttyS2 irq 5        # il mio modem Zoom
/sbin/setserial /dev/ttyS3 irq 9        # il mio modem USR
```

Impostazione standard degli IRQ:

```
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    non assegnato
IRQ 11    non assegnato
IRQ 12    non assegnato
IRQ 13    Math coprocessor
IRQ 14    Hard disk controller 1
IRQ 15    Hard disk controller 2
```

Non esiste una vera a propria regola per le scelte degli interrupt. Assicuratevi solamente che questo non venga utilizzato dalla scheda madre o da nessuna altra scheda presente nel Pc. Gli IRQ 2, 3, 4, 5 o 7 sono delle buone scelte. "non assegnato" significa che non gli è assegnata nessuna scheda o periferica standard. Da notare inoltre che gli IRQ 2 e IRQ 9 sono identici. Potete chiamarli 2 o 9, la periferica seriale fortunatamente funziona bene lo stesso. Se possedete una scheda seriale con un connettore per un bus a 16-bit, potete anche usare gli IRQ 10, 11, 12 oppure 15.

NON utilizzate gli IRQ 0, 1, 6, 8, 13 o 14! questi sono riservati alla scheda madre. Non sarà molto felice se gli usate i suoi IRQ prediletti! :) Quando avete impostato il tutto, controllate più volte `/proc/interrupts` per assicurarsi che non ci siano conflitti hardware.

9.2 Impostare gli indirizzi per le periferiche seriali

Dopo di che, dovete impostare l'indirizzo della porta. Controllate il manuale della vostra scheda per le impostazioni dei jumper. Come per gli interrupt, può esistere una sola periferica seriale per ciascun indirizzo. Le vostre porte sono usualmente configurate come segue:

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

Scegliete quale indirizzo desiderate per ogni periferica seriale e impostate i jumper di conseguenza. Io ho il mio modem sulla porta `ttyS3`, il mio mouse su `ttyS0` e terminale su `ttyS2`.

Quando riavviate il sistema, Linux dovrebbe riconoscere le porte seriali all'indirizzo al quale le avete configurate. Gli IRQ di Linux possono non corrispondere a quelli che avete impostato con i jumper. Non preoccupatevi. Linux non esegue nessun riconoscimento automatico degli IRQ, poiché questo può risultare fallimentare. Usate `setserial` per dare a Linux l'IRQ corretto per la vostra scheda, quello che avete impostato voi. Nel file `/proc/ioports` troverete quali indirizzi di porte di I/O sono in uso dopo l'avvio di Linux.

10 Come posso impostare le mie porte seriale ad una velocità maggiore? A quale velocità devo impostare il modem?

Questa sezione dovrebbe aiutarvi a scegliere la velocità da utilizzare per il vostro modem, quando lo usate con un programma di comunicazione, oppure con un programma `getty`. Normalmente viene impostata la velocità attraverso un parametro passato a `getty` oppure da un menu del vostro programma di comunicazione (oppure in un file di configurazione). Se per qualunque ragione il tutto non funziona per le alte velocità, esiste un modo per risolvere il problema utilizzando delle opzioni speciali con `setserial`: con le opzioni `spd_hi` oppure `spd_vhi` passate a setserial, la velocità 38400 attualmente imposta può essere modificata a 57600 (`spd_hi`) oppure a 115200 (`spd_vhi`).

Assicuratevi di possedere le porte UART 16550A per velocità alte. Ecco le impostazioni suggerite:

- Se avete un modem 28.8K (V.34), 33.6 (v.34), oppure 56K (V.90) utilizzate 115200 bps.
- Se avevete un modem 14400 bps (V.32bis), con la compressione dati V.42bis, utilizzate 57600 bps.
- Se avete un modem a 9600 bps (V.32), con la compressione dati V.42bis, utilizzate 38400 bps.
- Se siete in possesso di un modem più lento di 9600 bpx (V.32), impostate la velocità a quella più alta supportata dal vostro modem. Per esempio 300 bps (V.21/Bell 103), 1200 bps (V.22/Bell 212A), oppure 2400 bps (V.22bis).

Eseguite un test dalla riga di comando con `setserial` e quindi, se funziona, inserite il comando appropriato in `/etc/rc.d/rc.serial` oppure `/etc/rc.d/rc.local` o ancora `/etc/rc.boot/0setserial` in modo da essere avviato al boot del sistema. Assicuratevi di utilizzare un path valido per `setserial`, e un nome valido di periferica. Controllare le impostazioni delle porte seriali eseguendo:

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

11 Programmi e Utilità di comunicazione

Ecco una lista di alcuni programmi di comunicazione dalla quale potrete scegliere, disponibili via Ftp se non si trovano già assieme alla vostra distribuzione.

- `ecu` - comunicazione
- *C-Kermit* `http://www.columbia.edu/kermit/` - comunicazione portabili, supporto di script, seriale e TCP/IP che include il trasferimento di file e traduzione automatica del set di caratteri.
- `minicom` - programma di comunicazione simile a `telix`.
- `procomm` - programma comunicazione con supporto zmodem.
- `seyon` - basato su X-Windows.
- `xc` - pacchetto di comunicazione xcomm.
- Altri utili programmi sono `term` e `SLiRP`. Offrono funzionalità e TCP/IP attraverso shell account.
- `screen` è un altro programma per la multisessione. Funziona come le console virtuali.
- `callback` è un programma che permette al vostro modem di essere richiamato.
- `mgetty+fax` gestisce i FAX, e si offre come sostituto alternativo di `ps_getty`.
- `ZyXEL` è un programma di controllo per i modem ZyXEL U-1496. Gestisce le chiamate in entrata, in uscita, richiamo, FAX e funzione di casella vocale.
- Il software SLIP e PPP può essere reperito a: *ftp://sunsite.unc.edu/pub/Linux/system/network/serial* .
- Altre utilità: *ftp://sunsite.unc.edu/pub/Linux/system/serial* e *ftp://sunsite.unc.edu/pub/Linux/apps/serialcomm* o in uno dei tanti mirror. Queste sono le directory dove si trovano i programmi per comunicazioni seriali.

12 Porte Seriali: suggerimenti e miscellanea.

Ecco alcuni trucchetti che vi possono essere utili in svariate situazioni...

12.1 kermit e zmodem

Per utilizzare lo zmodem con `kermit` aggiungere le seguenti righe al vostro `.kermrc`:

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

Eventualmente sostituite la porta corretta sulla quale si trova il vostro modem. Quindi, per utilizzarlo, digitate `rz` or `sz <nome_file>` al prompt di `kermit`.

12.2 Impostare automaticamente il tipo di terminale

Per impostare automaticamente il tipo di terminale quando eseguite un login, basta aggiungere il tipo di terminale nel file `/etc/inittab`. Se avete un terminale VT100 sulla porta `ttyS1`, aggiungere "vt100" nel comando `gtty`:

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

Altri gettys hanno opzioni simili. Potete anche utilizzare `tset`, che può stabilire le caratteristiche del terminale al momento del login, e non dipende da nessuna impostazione standard.

12.3 Usare i colori con `ls` attraverso connessioni seriali.

Se il comando `ls` usato con i colori blocca il vostro terminale, disattivate questa opzione. `ls -color`, e `ls -colour` utilizzano le funzionalità colore di `ls`. Alcune installazioni hanno queste impostazioni già presenti. Controllate che nel file `/etc/profile` e `/etc/csh.cshrc` esista una riga per gli alias di `ls`. Se non desiderate modificare questa opzione a livello di sistema, create un alias di `ls` con `ls -no-color`.

12.4 Stampare su un stampare collegata ad un terminale seriale.

Esiste un programma, chiamato `vtprint`, disponibile a: *ftp://ftp.sdsu.edu/pub/vtprint* , e anche a *http://www.sdsu.edu/~garrett* .

Ne esiste anche un altro chiamato `xprt`. Può essere reperito a *ftp://sunsite.unc.edu/pub/Linux/system/printing* .

12.5 Può Linux configurare le periferiche seriali automaticamente?

Sì. Se il riconoscimento automatico non viene effettuato normalmente, attivatelo in modo che riconosca automaticamente le vostre periferiche seriali all'avvio del sistema. Aggiungete la linea seguente se necessario:

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

al vostro file `/etc/rc.d/rc.local` o `/etc/rc.d/rc.serial` oppure `/etc/rc.boot/0setserial`. Eseguite questa operazione per ogni porta seriale se desiderate la configurazione automatica. Assicuratevi ti fornigli una periferiche che esiste veramente sulla vostra macchina.

12.5.1 Note per schede multiporta

Per gli indirizzi sulla scheda e gli IRQ leggete il file `rc.serial` oppure `/etc/rc.boot/0setserial` fornito con programma `setserial`. Contiene molte informazioni sulle schede multiporta, inclusi indirizzi di I/O e i nomi delle periferiche.

12.6 Usare una console seriale.

Esiste un articolo nel numero 36 del *Linux Journal*, *http://www.ssc.com/lj/issue36/index.html* che spiega proprio come utilizzare una console seriale. Leggete il Text-Terminal-HOWTO.

13 Un piccolo passo avanti...

Non è necessario leggere questa sezione, ma può fornirvi ulteriori informazioni approfondite su Unix e il mondo delle telecomunicazioni.

13.1 Che cosa sono i lock file?

I lock file sono semplici file che indicano che una particolare periferica è in uso. Si trovano in `/usr/spool/uucp` oppure `/var/lock`. I file lock di linux utilizzano la seguente denominazione `LCK..`*name*, dove *name* è il nome della periferica, oppure il nome UUCP. Alcuni processi creano questi lock file per assicurarsi un accesso esclusivo alla periferica. Per esempio, se chiamate all'esterno con un modem, un file lock apparirà indicando che qualcuno sta già utilizzando quella periferica modem. I file lock spesso contengono il PID del processo che ha bloccato la periferica. Molti programmi cercano il lock, e controllano che sia ancora valido verificando la presenza dell'applicazione che lo detiene nella tabella dei processi. Se il lock è ancora valido, il programma esce (o per lo meno dovrebbe farlo). Se invece non è più valido, allora l'applicazione rimuove questo lock e utilizza la periferica, creando un nuovo lock file. Altri programmi invece terminano indicando solo che la periferica è in uso.

13.2 "baud" Vs. "bps"

"baud" e "bps" sono probabilmente i due termini usati più impropriamente nel mondo informatico e delle telecomunicazioni. Molte persone utilizzando questi termini indifferentemente, quando invece la storia non sta così.

baud

La misura baud indica quante volte al secondo un segnale, per esempio emesso da un modem (**mo**dulatore-**dem**odulatore), cambia. Per esempio, una baud rate di 1200 indica che un segnale cambia ogni 833 microsecondi. Baud rate comuni per i modem sono 50, 75, 110, 300, 600, 1200 e 2400. Molti modem ad alta velocità utilizzano hanno una baud rate a 2400. Per le limitazioni sulle linee telefoniche per voce una baud rate maggiore di 2400 è molto difficile da raggiungere, e forse funziona unicamente su linee con un segnale molto chiaro, senza interferenze. Molteplici bit possono essere gestiti per ogni baud, quindi il bit rate può superare notevolmente il baud rate. Il Baud prende il nome da Emile Baudot, l'inventore del telegrafo stampante asincrono.

bps

La misura in bps indica invece quanti bit per secondo sono trasmessi. Modem moderni trasmetto a 14.4K , 28.8K, 33.6K e 56K bps. Usando un modem con la compressione V.42 (compressione massima 4:1), è possibile raggiungere i 115.2K bps. Questo è ciò che molte persone confondono con i baud.

Quindi, se un modem ad alta velocità opera a 2400 baud, come può inviare dati a 14400 bps (o superiori)? I modem possono raggiungere elevato indice di velocità in bps (superiore a quello in baud) codificando un maggior numero di bit per ogni baud. Per questo il valore in bps sarà maggiore di quella in baud. Ad esempio se il vostro modem effettua un collegamento a 14400 bps significa che è collegato a 2400 baud ed inserisce in ogni cambiamento di fase 6 bit.

Come è iniziata questa confusione? Beh, quando un tempo i vecchi modem lenti erano considerati veloci, spesso la velocità in baud era equivalente a quella in bps. Un solo bit era inserito in ogni cambiamento di fase. I termini baud e bps erano utilizzabili indifferentemente, poiché corrispondevano allo stesso vlore. La confusione è iniziata quando i modem più moderni, che offrono prestazioni migliori in termini di velocità, hanno un indice in bps maggiore a quello in baud.

13.3 Che cosa sono gli UART? Come incidono nelle prestazioni?

Gli UART (**U**niversal **A**synchronous **R**eceiver **T**ransmitter) sono dei circuiti della vostra scheda seriale (se ne avete una) o sulla scheda madre del vostro PC. La funzione di questi UART possono essere eseguite anche da processori che effetuano anche altri tipi di operazioni. La finalità dei UART è quella di convertire i dati in bit, inviare questi alla linea seriale, e quindi ricostruire nuovamente i dati presso l'altra estremità della connessione seriale. Gli UART si scambiano dati in pacchetti dimensionati in byte, scelto per convenienza poiché corrispondono al numero di caratteri ASCII.

Immaginate di avere un terminale connesso al vostro PC. Quando digitate un carattere, il terminale passa quel carattere al trasmettitore (quindi anche un UART). Questo invia quel byte sulla linea seriale, un bit alla volta, ad un valore predefinito. La UART del PC al quale è connesso il terminale riceve questi dati in bit, e li riconverte in byte e li inserisce in un buffer.

Esistono due tipi diversi di UART. Probabilmente avrete sentito parlare dei UART 'stupidi' (i modelli 8250 e 16450) e di quelli FIFO (modello 16550A). Per capire queste differenze, per prima cosa esamineremo cosa succede quando un UART ha inviato oppure riceve dei dati.

L'UART di per sè non controlla in nessuno modo i dati, li invia e li riceve solamente. Per l'UART originale la CPU ottiene un interrupt dalla periferica seriale ogni volta che un byte viene inviato o ricevuto. La CPU quindi preleva il byte ricevuto dal buffer dell'UART e lo inserisce da qualche parte nella memoria, oppure assegna all'UART un altro byte da inviare. Gli UART 8250 e 16450 hanno un buffer di 1 solo byte. Questo significa che ogni volta che 1 byte viene inviato o ricevuto la CPU viene interrotta. A velocità basse può andare bene. Ma ad alta velocità la CPU diventa così occupata a dialogare con l'UART che non ha più tempo di servire gli altri task. In alcuni casi la CPU non riesce a gestire tutti gli interrupt generati dall'UART e il byte nel buffer viene sovrascritto, poiché i dati arrivano troppo velocemente.

14 Risoluzione dei Problemi

14.1 Ricevo sempre il messaggio "line *NNN* of inittab invalid"

Assicuratevi che state utilizzando la corretta sintassi per la vostra versione di `init`. Le differenti versioni di `init` che sono state rilasciate utilizzano sintassi differenti per il file `/etc/inittab`. Assicuratevi che state anche usando la corretta sintassi per la vostra versione di `getty`.

14.2 Quando provo ad effettuare una chiamata all'esterno, ricevo il messaggio "/dev/cua*N*: Device or resource busy"

Questo problema è dovuto al fatto che il DCD o il DTR non sono impostati correttamente. Il DCD deve essere impostato unicamente quando persiste una connessione (per esempio quando qualcuno si è collegato a voi chiamando dall'esterno), e non quando `getty` sta monitorando la porta. Controllate che il vostro modem sia configurato per impostare il DCD solo quando avviene una connessione. Il DTR deve essere invece usato quando qualunque altra cosa sta monitorando la porta, come ad esempio `getty`, `kermit`, o qualche altro programma.

Un'altra causa dell'errore "device busy" è che avete impostato la vostra porta seriale con un interrupt già in uso da qualche altra periferica hardware. Quando ogni periferica viene inizializzata questa richiede al kernel il permesso di usare il suo interrupt hardware. Il sistema operativo tiene traccia di quali interrupts sono assegnati e a chi, e se il vostro interrupt è già in uso, la vostra periferica non verrà inizializzata correttamente. In realtà essa non ha la minima idea di cosa stia succedendo, e l'unica cosa che può fare è rispondere con “device-busy” quando tentate di usarla. Controllate tutti gli interrupt di tutte le vostre periferiche (porte seriali, schede ethernet, controller SCSI, ecc.). Evitate conflitti di questo genere.

14.3 Continuo a ricevere il messaggio “Getty respawning too fast: disabled for 5 minutes”

Assicuratevi che il vostro modem sia configurato correttamente. Controllate i registri `E` e `Q`. Questo errore può capitare quando il modem cerca di dialogare con `getty`.

Assicuratevi anche che avete lanciato `getty` nel modo corretto dal vostro `/etc/inittab`. Usando una sintassi errata oppure una periferica sbagliata si possono verificare seri problemi.

Verificate che la sintassi del vostro `/etc/gettydefs` sia corretta usando questa procedura:

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

Questo errore si verifica anche quando fallisce l’inizializzazione di `uugetty`. Fate riferimento alla sezione 14.11 (getty o uugetty ancora non funzionano).

14.4 Le periferiche seriali sono lente, oppure funzionano in un sola direzione.

Molto probabilmente avere un conflitto di IRQ. Assicuratevi che non ci sia qualche IRQ condiviso. Verificate tutte le schede (porte seriali, schede ethernet, controller SCSI, ecc.). Controllate le impostazioni dei vari jumper e che i parametri per il programma `setserial` siano corretti per tutte le vostre periferiche. Inoltre controllate `/proc/ioports` e `/proc/interrupts` per eventuali conflitti.

14.5 Il mio modem è bloccato dopo che qualcuno effettua una chiamata

Questo può accadere quando il vostro modem non effettua il reset quando il DTR cade. Io vedo i miei LED RD e SD letteralmente impazzire quando questo accade a me. Avete bisogno di resettare il vostro modem. Molti modem Hayes-compatibili eseguono questa operazione con `&D3`, ma sul mio USR Courier devo impostare `&D2` e `S13=1`. Controllate sul manuale del vostro modem.

14.6 Ho un terminale connesso al mio PC, ma quando tento di effettuare un login tutto si blocca.

- Se state usando `getty`: Molto probabilmente non avete `CLOCAL` nel vostro `/etc/gettydefs` per il vostro terminale, e molto probabilmente non state usando un cavo pienamente null-modem. Avete bisogno di `CLOCAL`, che indica a Linux di ignorare i segnali di controllo del modem. Ecco come dovrebbe apparire il tutto:

```
# 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
```

Successivamente `terminate` il processo `getty` in modo che uno nuovo entrerà in esecuzione con le nuove impostazioni.

- Se state usando `agetty`: Aggiungete il flag `-L` alla linea di `agetty` nel vostro `/etc/inittab`. Questo farà ignorare i segnali di controllo del modem. Quindi riavviate `init` digitando `init q`. La configurazione dovrebbe essere più o meno così:

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

14.7 Ad alta velocità il mio modem perde alcuni dati

Se state tentando di utilizzare il vostro modem a una velocità superiore ai 38400 bps e non avete una UART 16550 allora dovete aggiornarla. Leggete la sezione 13.3 (Cosa sono le UART?).

14.8 All'avvio del sistema Linux non riporta le periferiche seriali come le ho in precedenza configurate

Questo è vero solo in parte. Linux non effettua nessuna verifica degli IRQ al momento del boot, effettua solo un rilevamento di periferiche seriali. Quindi, con una corretta impostazione, io ottengo questo all'avvio della macchina:

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

Dovete usare `setserial` per comunicare a Linux quale IRQ state usando.

14.9 `rz` e/o `sz` non funzionano quando chiamo la mia macchina Linux con un modem.

Se Linux cerca la periferica `/dev/modem` quando tentate di trasferire file, controllate i file `/etc profile` o `/etc csh.cshrc`. Probabilmente esiste una serie di alias definiti qui in alcune distribuzioni, come ad esempio Slackware. Questi alias confondono l'operato dei programmi zmodem. Eliminateli, oppure correggeteli.

14.10 Il mio schermo visualizza alcuni buffi caratteri

Questo accade su console virtuali su cui sono inviate informazioni di tipo binario direttamente allo schermo, oppure anche attraverso connessioni seriali. Il modo per risolvere questo problema è di digitare `echo ^v^[c`. Per la serie con tutti i caratteri, ecco qua:

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

14.11 `getty` o `uugetty` ancora non funzionano

Esiste una opzione chiamata `DEBUG` inclusa in `getty_ps`. Editate il vostro file di configurazione `/etc/conf.{uu}getty.ttyS`*N* e aggiungete `DEBUG=`*NNN*. Dove *NNN* è una delle seguenti combinazioni di numeri a seconda di cosa state cercando di debuggare:

```
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
```

Impostare `DEBUG=010` è un buon punto di partenza.

Se state eseguendo `syslogd`, informazioni di debug appariranno nei vostro file di log. Se non state eseguendo il `syslogd` queste informazioni appariranno in `/tmp/getty:ttyS`*N* per debuggare `getty` e `/tmp/uugetty:ttyS`*N* per `uugetty`, e in `/var/adm/getty.log`. Osservate attraverso queste informazioni di debug che cosa sta accadendo. Molto spesso avrete bisogno di perfezionare qualche impostazione o qualche parametro nella vostra configurazione, oppure di riconfigurare il vostro modem.

Potete anche provare `mgetty`. Alcune persone hanno più fortuna usando questo software.

15 Altre fonti di informazione

- Le pagine man di: `agetty(8)`, `getty(1m)`, `gettydefs(5)`, `init(1)`, `login(1)`, `mgetty(8)`, `setserial(8)`
- Il manuale del vostro modem o terminale
- NET-3 HOWTO: tutto sulle reti di comunicazione, inclusi SLIP, CSLIP e PPP
- PPP HOWTO: un aiuto con il PPP
- Printing HOWTO: Come impostare una stampante seriale
- Serial-Programming-HOWTO: per alcuni aspetti della programmazione seriale
- Term HOWTO: tutto quello che desiderate sapere sul programma `term`
- Text-Terminal-HOWTO: come funzionano e come installaree configurare un terminale testuale
- UPS HOWTO: configurare un UPS connesso alla vostra porta seriale
- UUCP HOWTO: per informazioni su come configurare UUCP
- Newsgroups di UseNet:

```
comp.os.linux.answers
FAQs, How-To's, READMEs, tutti riguardanti Linux.
comp.os.linux.hardware
Compatibilità hardware con il sistema operativo Linux.
comp.os.linux.networking
Reti e comunicazioni con Linux.
comp.os.linux.setup
l'installazione di Linux e l'amministrazione del sistema.
comp.terminals
```

- The Linux serial mailing list. Per iscriversi inviate una email a *majordomo@vger.rutgers.edu* , cpn "`subscribe linux-serial`" nel corpo del messaggio. Se inviate "`help`" nel corpo del messaggio riceverete un messaggio di aiuto. Il server inoltre gestisce altri mailing lists su Linux. Inviate il comando "`lists`" per una lista delle Mailing Lists presenti.

- FAQs sui Modem:

  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`

- Programmazione seriale: Serial-Programming-HOWTO

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

  `http://www.ora.com/catalog/posix/` name=POSIX Programmer's Guide>, di Donald Levine (ISBN 0-937175-73-0; O'Reilly)

16 Contributi

Gran parte dell'HOWTO originale è stata scritta da Greg Hankins *greg.hankins@cc.gatech.edu* . Ha anche riscritto un gran numero di contributi inviati da altri collaboratori per mantenere la fluidità e lo stile del documento. Ecco le sue parole: Ringrazio tutti coloro che hanno contribuito o commentato questo documento, la lista di tutte queste persone è troppo grande (spesso oltre la centinaia di persone). Ringraziamenti a Ted per aver risposto alle domanda sui driver per le schede seriali, a Kris Gleason che mantiene `getty_ps`, e Gert Döring che mantiene `mgetty`.

FINE DEL SERIAL HOWTO