The Linux Modem-HOWTO

David S.Lawyer mailto:dave@lafn.org v0.10, Maggio 2000

Aiuto per selezionare, connettere, configurare, risolvere problemi e comprendere i modem in un PC. Vedere il Serial-HOWTO per dispositivi seriali multiporta.

Indice

1 Introduzione

1.1 Modem DSL, cable e ISDN in altri HOWTO

Questo documento riguarda i modem convenzionali per PC, principalmente modem per il bus ISA. In ogni caso, ad ogni nuova versione sono aggiunte nuove informazioni sui modem per il bus PCI.

- modem DSL: vedere il mini-howto ADSL
- Cable-Modems-HOWTO (un tempo un mini-Howto di LDP) `http://www.cs.unm.edu/~vuksan/linux/Cable-Modem.html`
- ISDN Howto (non è un HOWTO di LDP) `http://sdb.suse.de/sdb/en/html/isdn.html` : driver per Modem ISDN. La maggior parte delle informazioni correlate sono in tedesco. Per un tutorial su ISDN vedere `http://public.swbell.net/ISDN/overview.html` .

Vedere anche 23 (Appendice D: Altri tipi di Modem)

1.2 Non compresi: modem PCMCIA, PPP

Per i modem sul bus PCMCIA vedere il PCMCIA-HOWTO: dispositivi PCMCIA seriali e modem. Questo HOWTO non tratta PPP (usato per connettersi ad Internet via modem) o programmi di comunicazione. Mostra però come usare programmi di comunicazione per verificare che il vostro modem funzioni bene e possa eseguire delle chiamate telefoniche. Se volete usare un modem per connettervi ad Internet allora dovete impostare PPP. C'è parecchia documentazione per PPP (incluso un PPP-HOWTO) ma potrebbe essere obsoleta o non applicabile alla vostra situazione. Parte di questa potrebbe essere trovata in /usr/doc/ppp o simile.

1.3 Copyright, liberatoria, marchi registrati e crediti

1.3.1 Copyright (in lingua originale)

1.3.2 Liberatoria

Sebbene non ho l'intenzione di mettervi fuori strada, probabilmente ci sono parecchi errori in questo documento. Vi invito a farmeli notare. poiché questa è documentazione libera, dovrebbe essere ovvio che io non posso essere ritenuto legalmente responsabile per qualsiasi errore.

1.3.3 Marchi registrati

Qualsiasi nome di marca (che inizia con lettera maiuscola) dovrebbe essere considerato come marchio registrato. Detti marchi registrati appartengono ai rispettivi proprietari.

Hayes è un marchio registrato dalla Microcomputer Products Inc. Uso winmodem per indicare qualsiasi modem che richiede MS-Windows e non nel senso di un marchio.

1.3.4 Crediti

Quanto segue è solamente una rozza approssimazione di come è stata creato (fino al 2000) questo documento: circa 1/4 del materiale qui presente è stato preso tale quale dal Serial HOWTO versione 1.11 di Greg Hankins `mailto:gregh@cc.gatech.edu` (con il suo permesso). Circa un altro quarto è stato preso sempre dal Serial HOWTO e revisionato. La restante metà è stata creata ex-novo dall'autore David S. Lawyer `mailto:dave@lafn.org` .

1.4 Contattare l'autore

Per favore non mandatemi e-mail chiedendomi quale modem acquistare o se un certo modem funzionerà sotto Linux. Cercate nella vasta lista in 2.4 (Evitare: Software (interni) Modem). Inoltre, per favore non chiedetemi come configurare un modem a meno che abbiate già scorso questo HOWTO e non ci siate ancora riusciti.

Per favore fatemi sapere circa qualsiasi errore in fatti, opinioni, logica, grammatica, chiarezza, link, ecc. Ma per prima cosa, se la data è più vecchia di un mese, controllate di avere l'ultima versione. Per cortesia inviatemi qualsiasi altra informazione che pensate possa appartenere a questo documento.

1.5 Nuove versioni di questo HOWTO

Nuove versioni del Modem-HOWTO escono ogni mese o quasi visto che la situazione dei modem va rapidamente cambiando (e visto che sto ancora imparando). I vostri problemi potrebbero essere risolti nell'ultima versione. Sarà disponibile da consultare e/o scaricare nei siti mirror di LDP. Per una lista di tali siti vedere

`http://sunsite.unc.edu/LDP/mirrors.html` . Se volete solo controllare velocemente la data dell'ultima versione andate a

`http://metalab.unc.edu/LDP/HOWTO/Modem-HOWTO.html` e confrontatelo con la versione che state attualmente leggendo: v0.10, Maggio 2000.

1.6 Novità in questa versione

Modem-Sharing (condivisione di modem) mini-howto, modem digitali, modem Newcom, in più no response to AT (nessuna risposta ad AT) e can't find modem (non posso trovare il modem).

1.7 Cos'è un modem ?

Un modem è un dispositivo che consente di inviare segnali digitali attraverso una normale linea telefonica non predisposta per segnali digitali. Se le linee telefoniche fossero tutte digitali un modem non sarebbe necessario. Esso consente al proprio computer di connettersi e comunicare con il resto del mondo. Quando si usa un modem, in genere si usa un programma di comunicazione, un browser web (che include un programma di comunicazione) per utilizzare il modem ed immettersi nella linea telefonica. Utenti esperti possono fare in modo che altri utenti siano in grado di connettersi tramite linea telefonica al loro computer ed usarlo. Questa procedura si chiama dial-in.

Ci sono due tipi principali di modem per un PC: esterni ed interni.

Quelli esterni si collocano all'esterno del PC mentre quelli interni sono inseriti all'interno e non si vedono. I modem esterni si collegano al PC tramite una porta seriale situata nel retro del PC. Il modem interno è una scheda che si inserisce dentro al computer ed ha una invisibile porta seriale incorporata. Per un confronto più dettagliato vedere 2.1 (Esterno contro Interno). Quindi quando si dispone di un modem interno, si dispone anche di una porta seriale dedicata (detta porta può quindi essere usata solo con il modem e non con qualsiasi altro dispositivo tipo un altro modem od una stampante). In Linux, le porte seriali sono chiamate ttyS0, ttyS1, etc. (e corrispondono in genere rispettivamente a COM1, COM2, etc. in Dos/Windows).

Non si deve confondere la porta seriale con l'Universal Serial Bus (USB) che usa uno speciale connettore modulare e potrebbe essere usato con i modem in futuro. Vedere 4 (Modem e Porte seriali: Nozioni di base) per ulteriori dettagli sui modem e le porte seriali.

I modem spesso includono la capacità di inviare fax (Fax Modem). Vedere 14.2.3 (Fax) per una lista di software per i fax. I Voice modem possono funzionare come una segreteria telefonica e gestire voicemail. Vedere 11.6 (Voicemail).

1.8 Installazione rapida

1.8.1 Installazione di modem esterni

Con un cavo passante o modem, connettere il modem ad una porta seriale del PC. Assicurarsi di conoscere il nome della porta seriale: nella maggior parte dei casi COM1 è ttyS0, COM2 è ttyS1 ecc. Potrebbe essere necessario consultare il menù delle impostazioni del BIOS per determinarlo. Collegare il cavo di alimentazione per dare corrente al modem. Vedere 1.8.3 (Per tutti i modem) per successive istruzioni.

1.8.2 Modem interni (su bus ISA)

(Per il bus PCI vedere 6.1 (Supporto al bus PCI in fase di completamento) e 2.5 (Modem PCI).) Se il modem dice che funzionerà solo sotto MS Windows, siete sfortunati. Se avete già due porte seriali, rendete il modem quale terza porta seriale (ttyS2 = COM3). Cercare un numero di IRQ libero da usare. Nel passato IRQ5 era spesso non usato ma oggi IRQ 5 è usato anche dalle schede audio. Poi impostare i ponticelli (jumper) (o simili) del modem interno all'IRQ libero e ad un indirizzo di IO tipo 3E8 (ttyS2).

O simili (nella frase precedente) può essere un poco più difficile. Se il modem ha un Plug and Play (PNP) per il bus ISA, possiamo ovviare usando il programma isapnp che è incluso negli isapnptools. Vedere man isapnp o la FAQ per esso. Vedere anche Plug-and-play HOWTO. Con un BIOS PnP, dovreste essere in grado di dire al menù di impostazione del CMOS che non avete un sistema operativo PNP e successivamente il BIOS potrebbe impostare un corretto IRQ e indirizzo di I/O nella scheda del modem. Se volete forzare il BIOS in modo che imposti un determinato IRQ e/o indirizzo IO, allora dovreste essere in grado di farlo usando Windows9x sullo stesso PC. Potrebbe impostare gli indirizzi nella memoria flash del BIOS PnP dove potranno essere usati per la configurazione con Linux così come con Windows. Consultate il Plug-and-play HOWTO e cercate forced (forzato), che ricorre in diversi punti. Per Windows3.x potreste fare la stessa cosa usando ICU sotto Windows 3.x. Potrebbe anche esserci il modo di disabilitare il PnP usando il software (sotto Windows) fornito con il modem.

Infine si deve anche cercare il file dove viene lanciato setserial ed aggiungere una riga che potrebbe essere qualcosa come: setserial /dev/ttyS2 irq5. Eccetto che per setserial v2.15 e successivi potreste (se la vostra distribuzione ve lo consente) semplicemente lanciare setserial da riga di comando ed il risultato sarà salvato in un file di configurazione. Vedere 9.1 (Cos'è setserial) per maggiori informazioni. Vedere la successiva sottosezione 1.8.3 (Per tutti i Modem) per ulteriori istruzioni per una veloce installazione

1.8.3 Per tutti i modem

Collegare il modem alla linea telefonica. Poi lanciare un programma di comunicazione tipo minicom ed aprire il menu di configurazione per la porta seriale. Assegnare una velocità di trasmissione (baud rate) alcune volte maggiore del bit rate del proprio modem. Vedere 13.4 (Tabella delle velocità) per la velocità ideale da usare. Fornire il nome completo della porta seriale tipo /dev/ttyS1. Impostare il controllo di flusso hardware (RTS/CTS). Ora occorre salvare le impostazioni ed uscire da minicom. Poi rilanciare minicom, digitare AT per vedere se il vostro modem è lì e risponde con OK. Poi portarsi nell'elenco dei numeri da selezionare (dial directory) (o menu) e comporre un numero.

2 Modem per un PC con Linux

2.1 Esterni contro Interni

Un modem per un PC può essere sia interno che esterno. Quello interno è installato all'interno del vostro PC (dovete rimuovere viti etc, per installarlo) e quello esterno si collega semplicemente ad un connettore di porta seriale sul PC. I modem interni sono meno costosi, è meno probabile che subiscano perdite di dati a causa del sovraccarico del buffer, in genere consumano meno elettricità, e non occupano spazio sulla vostra scrivania.

I modem esterni sono molto più semplici da installare e richiedono minore configurazione. Hanno delle luci che possono fornire un indizio circa quello che sta accadendo, ed aiutare a risolvere i problemi. Anche il fatto che la porta seriale ed il modem possano essere fisicamente separati aiuta nella risoluzione di problemi. I modem esterni sono facilmente spostabili su un altro computer.

Sfortunatamente la maggior parte dei modem esterni non hanno degli interruttori per spegnere l'alimentazione di corrente e probabilmente consumano un poco di corrente anche quando sono spenti (a meno di staccare la spina dal muro). Ogni watt che usano vi costa circa $1 per anno. Un altro possibile svantaggio di un modem esterno è che sarete forzati ad usare una porta seriale esistente che potrebbe anche non supportare una velocità di oltre 115.200k (sebbene fino al tardo 1998 anche la maggior parte dei modem interni non la supportavano -ma alcuni sì). Se un nuovo modem interno avesse una UART 16550 potrebbe mettere meno carico sulla CPU (ma quasi nessuno lo fa al tardo 1998).

I modem interni presentano un particolare problema per Linux, ma funzioneranno bene come i modem esterni a patto che voi evitiate quell'alta percentuale di essi che funzionano solo sotto MS Windows, ed anche a patto che spendiate un poco di tempo (a volte molto tempo) per configurarli correttamente. Alcuni dei modem che funzionano solo sotto MS Windows sono, sfortunatamente, non espressamente descritti come tali. Se ne comprate uno nuovo, assicuratevi di poterlo rendere e che possiate essere rimborsati se non funzionerà sotto Linux.

Visto che la maggior parte dei nuovi modem sono plug-and-play avete diversi modi per gestirli:

- Usare il programma isapnp
- Lasciare che la configurazione venga fatta dal BIOS PnP
- Modificare il kernel per creare un Linux PnP (attualmente non disponibile)

Ciascuno dei metodi sopracitati ha degli svantaggi. La documentazione di isapnp è difficile da comprendere sebbene leggere il Plug-and-Play HOWTO (allo stato attuale incompleto) aiuti a comprendere. Se volete che sia il BIOS PnP ad eseguire la configurazione, tutto quello che dovete fare è assicurarvi che esso sappia che non avete un sistema operativo PnP. Ma potrebbe non eseguirla correttamente. Per scoprire quello che è stato fatto vedere 6.5.2 (Cos'è impostato nell'hardware della mia porta seriale?). Modificare il kernel ha funzionato in passato, ma nessuna modifica sembra attualmente disponibile. Controllate il sito web per questo.

Ci sono molti utenti di Linux che dicono che è molto più semplice prendere un modem esterno e collegarlo. Ma visto che le nuove periferiche oggi sono per la maggior parte PnP, potreste aver bisogno di gestirle, così perché rimandare l'inevitabile? Comunque la soluzione più vantaggiosa (e più costosa) è quella di un modem esterno (se avete una porta seriale libera).

2.2 Modem esterni

2.2.1 Modem esterni PnP

Molti modem esterni sono etichettati Plug and Play (PnP), ma dovrebbero funzionare bene anche come modem non PnP. Visto che in genere si collega il modem ad una porta seriale che ha i suoi propri numeri di IRQ ed indirizzi di IO, il modem non necessita di capacità PnP per impostarli. Comunque, la porta seriale stessa potrebbe necessitare di una configurazione (numero IRQ e indirizzo IO) a meno che la configurazione di default vada bene.

Come può un modem esterno essere chiamato PnP visto che non può essere configurato tramite PnP? Bè, esso ha una speciale identificazione PnP costruita al suo interno che può essere letta (attraverso la porta seriale) da un sistema operativo PnP. Sistemi operativi di questo tipo dovrebbero quindi sapere che voi avete un modem in una certa porta e dovrebbero anche conoscerne il numero di modello. Quindi potreste non avere bisogno di configurare i programmi applicativi dicendogli su quale porta sia il modem (tipo /dev/ttyS2 o COM3). Ma visto che non avete un sistema operativo PnP dovrete configurare il vostro programma applicativo manualmente fornendogli l'identificativo del /dev (tipo /dev/ttyS2).

2.2.2 Connessione ed installazione

Connettere un modem esterno è facile, confrontato con la connessione della maggior parte degli altri disposivi ad una porta seriale che richiedono diversi tipi di cavi null modem. I modem usano cavi diretti, senza pin incrociati. La maggior parte dei negozi di computer dovrebbero averne. Assicuratevi di avere il corretto genere. Se state il vostro computer usa una porta seriale B09 o B25, sarà sempre maschio, ovvero il connettore sul cavo dovrà essere femmina. Attaccate il vostro modem ad una delle vostre porte seriali. Se vi va bene accettare l'IRQ e l'indirizzo IO di default della porta alla quale lo state connettendo, dovreste essere pronti per lanciare il vostro programma di configurazione e configurare il modem stesso.

2.2.3 Cosa significano le luci (LED)

- TM Test del Modem
- AA (Auto Answer) Autorisposta (se attivata, il vostro modem risponderà ad una chiamata in arrivo)
- RD (Receive Data) linea di Ricezione Dati = RxD
- SD (Send Data) linea di invio dati = TxB
- TR (Terminal Ready) terminale pronto = DTR (impostato dal vostro PC)
- RI (Ring indicator) indicatore di squillo (se attivo, qualcuno sta chiamando il vostro modem)
- OH (Off Hook) (se disattivo, il vostro modem ha appeso la comunicazione)
- MR (Modem Ready) modem pronto = DSR ??
- EC (Error Correction) correzione di errore
- DC (Data Compression) compressione dati
- HS (High Speed) alta velocità (per questo modem)

2.3 Modem interni

Un modem interno è installato in un PC rimuovendo il coperchio ed inserendo la scheda modem in un alloggiamento libero della scheda madre. Ci sono modem per alloggiamenti ISA ed altri per quelli PCI. Mentre i modem esterni si connettono alla porta seriale (tramite un corto cavo), i modem interni hanno la porta seriale costruita all'interno del modem. In altre parole, la scheda modem è sia una porta seriale che un modem.

L'impostazione degli indirizzi IO ed IRQ per una porta seriale era un tempo effettuata dai jumper sulla scheda. Questi piccoli cubetti neri rettangolari di circa 5x4x2 mm si spingevano sopra i pin della scheda. I modem Plug-and-Play (in realtà la parte porta seriale del modem) non usano i jumper per impostare gli indirizzi, ma invece sono configurati inviando comandi di configurazione ad essi (tramite lo spazio di indirizzo IO sul bus ISA all'interno del computer). Detti comandi di configurazione possono essere inviati da un BIOS PnP, dal programma isapnp (solo per il bus ISA) o da un sistema operativo PnP. La loro configurazione è parte del sistema operativo Window 95/98. Sotto Linux si ha una scelta di modi (nessuno dei quali è sempre facile) per configurare gli io-irq.

1. Usare isapnp che potrebbe essere lanciato automaticamente ad ogni avvio
2. Usare un BIOS PnP a sè stante (che viene eseguito ad ogni avvio)
3. Modificare Linux per renderlo un sistema operativo PnP

2.4 Software Modem (interni - la maggior parte winmodem)

I software modem demandano la maggior parte (o quasi tutto) il lavoro del modem al chip del processore principale (CPU) del vostro computer (tipo un chip Pentium). Complessi programmi software proprietari (driver) svolgono questo compito sulla CPU. Vasta parte dei modem interni costruiti dopo la seconda metà del 1998 *non* funzionano con Linux visto che sono software modem che funzionano solo sotto Windows e sono spesso chiamati winmodem. Sebbene alcuni volontari fossero disponibili per cercare di scrivere driver Linux per questi modem, le specifiche non sono ancora state rese disponibili quindi questo non può essere fatto. Prima del 2000 circa, nessun software modem poteva essere usato con Linux a causa della mancanza di driver per essi sotto Linux.

Poi, finalmente, verso la fine del 1999, sembrava che due software modem potessero funzionare sotto Linux, la Lucent Technologies forniva in via non ufficiale un codice binario Linux per supportare i suoi software modem PCI, ma sono stati riportati dei bug nelle prime versioni. PC-TEL introduceva un nuovo software modem per Linux. Ci saranno altre ditte che seguiranno questa strada per andare a creare dei linmodem? Per un elenco di modem che funzionano/non funzionano sotto Linux vedere

elenco modem `http://www.o2.net/~gromitkc/winmodem.html` . Un progetto per far funzionare i winmodem sotto Linux si trova in `http://linmodems.org`

C'è anche a disposizione una mailing list. Ci sono altri tentativi in corso di reverse-engineering con almeno un'evidenza di un winmodem che sia stato fatto funzionare sotto Linux (ma non a piena funzionalità). Così mentre state leggendo questo documento potrebbero esserci ulteriori linmodem.

Se viene reso disponibile del codice per far lavorare un winmodem sotto Linux, allora si potrebbe chiamare linumodem. È ancora un winmodem?. Forse sì visto che funziona anche sotto MS Windows. Il termine Winmodem è il marchio di fabbrica di un certo tipo di winmodem.

Ecco un'ulteriore più precisa terminologia riguardante i software modem. HSP (Host Signal Processor) significa che il processore ospitante (il vostro chip CPU) crea il codice necessario per produrre il segnale elettrico sulla linea telefonica. Il modem in sè crea semplicemente una qualsivoglia onda elettrica che la CPU gli dice di creare. Di contro, un modem controllerless può creare le onde di sua iniziativa (ma non può controllare il modem). Non contiene l'attrezzatura per gestire i byte che sono inviati o ricevuti Non può comprimere stringhe di byte; non può verificare errori; non può comporre i pacchetti. In altre parole non può controllare il modem ma invece deve essere la CPU a svolgere tutto questo lavoro usando il software. I Rockwell HCF (Host Controlled Family) fanno questo. Se il software che fa tutto questo potesse essere portato sotto Linux, dopo non ci sarebbe questo problema. A parte quanto sopra, un modem che non simula una porta seriale non funzionerà sotto Linux.

Come determinare se un modem interno funzionerà sotto Linux? Per prima cosa vedete se il nome o la descrizione di esso indica che trattasi di software modem: HSP, HCF, HSF, controllerless, host-controlled, host-based, e soft-... modem. Se è un software modem funzionerà solo nei rari (fino ad ora) casi in cui sia disponibile un driver Linux. Se non conoscete il modello del modem ad avete anche Windows sul vostro Linux PC, cliccate sull'icona Modem del Pannello di Controllo. Poi controllate l'elenco dei modem all'indirizzo web citato 4 paragrafi sopra. Se anche questo non funziona (o non è possibile fare), potete controllare la confezione (o il manuale) e cercare la sezione che dice qualcosa tipo Minimum System Requirements (Sistema minimo richiesto) o semplicemente System Requirements (Requisiti di sistema). Potrebbe essere stampato molto in piccolo. Leggete attentamente. Se viene elencato Windows oppure una CPU Pentium come uno dei requisiti, allora probabilmente non funzionerà sotto Linux.

Altrimenti potrebbe funzionare sotto Linux se non viene esplicitamente indicato che dovete avere Windows. La scritta designed for Windows (progettato per Windows) potrebbe semplicemente voler dire che supporta interamente il plug-and-play Microsoft, il che va bene visto che Linux usa le stesse specifiche plug-and-play (ma è più difficile configurarlo sotto Linux). L'essere progettato per Windows quindi non dà indizi sul fatto che funzioni o meno sotto Linux. Potreste verificare il sito Web del produttore oppure chiedere informazioni tramite posta elettronica. Una volta ho visto una pagina web che dichiarava specificamente che un modello funzionava sotto Linux implicitamente dicendo che un altro modello non funzionava.

A parte il problema di ottenere un driver, quali sono i pro e i contro di un software modem? Visto che il software modem usa la CPU per svolgere la maggior parte del suo lavoro, esso richiede meno parti elettroniche sulla scheda e quindi costa meno. Allo stesso tempo, la CPU è pesantemente impegnata dal modem, quindi il funzionamento generale potrebbe rallentare. Questo è particolamente vero se altri processi che impegnano la CPU intensamente sono in esecuzione allo stesso tempo. Naturalmente se non state usando il software modem non c'è alcun degrado di prestazioni. Vale la pena risparmiare sul prezzo per questo inconveniente? In alcuni casi sì, specialmente se il modem viene usato raramente o non si hanno in esecuzione altri processi impegnativi per la CPU mentre si usa il modem. Questi sono quindi i casi in cui l'uso del software modem è economicamente giustificato. Quanto risparmiato sul costo del modem potrebbere essere usato per una migliore CPU che sarà in grado di velocizzare un poco tutto l'insieme. Però i circuiti elettronici sulla scheda del modem possono svolgere il lavoro con più efficacia rispetto ad una CPU. Così se usate molto il modem è probabilmente meglio evitare i software modem (e quindi potreste usare una CPU meno potente :-).

2.5 Modem PCI

Una scheda modem PCI è una di quelle che si inseriscono in un alloggiamento del bus PCI sulla scheda madre di un PC. Sfortunatamente, sembra che la maggior parte dei modem PCI non funzioni sotto Linux ma sono in atto dei tentativi di supportare alcuni di essi. Vedere 6.1 (Supporto del bus PCI in fase di completamento)

2.6 Quali modem interni potrebbero non funzionare sotto Linux

- 2.4 (Software Modem (interni)) funzionano solo nei rari casi dove sia disponibile un driver Linux.
- 2.5 (Modem PCI) la maggior parte non funziona sotto Linux
- 2.6.1 (Modem MWave e DSP) potrebbero funzionare, ma solo se prima fate partire Windows/Dos ogniqualvolta accendete il PC.

- I modem con driver 2.6.2 (RPI (Rockwell)) funzionano ma con prestazioni ridotte.

2.6.1 Modem MWave e DSP

Questi modem usano i DSP (Digital Signal Processors) che sono programmati tramite algoritmi che devono essere scaricati dall'hard disk verso la memoria DSP appena prima di usare il modem. Sfortunatamente lo scaricamento viene spesso fatto da programmi Dos/Windows così la cosa non si può fare da Linux. I modem comuni che funzionano sotto Linux hanno spesso anche un DSP (e la cosa potrebbe essere menzionata nella confezione), ma il programma che lo lancia è situato all'interno del modem. Questo non è un modem DSP nel senso inteso da questa sezione e dovrebbe funzionare bene sotto Linux. Un esempio di modem DSP è l'Aptiva MWave dell'IBM.

Se un modem DSP simula una porta seriale, allora è usabile da Linux che comunica con i modem attraverso la porta seriale. Se avete anche Dos/Windows sullo stesso PC potreste essere in grado di usare il modem: prima installate i driver sotto DOS (usando i driver DOS e non quelli Windows). Poi lanciate Dos/Windows e caricate il driver per il modem in modo da programmare il DSP. Poi senza spegnere il computer, passate a Linux.

Potreste scrivere un file batch (a dire il vero uno script) per fare questo. Ecco un esempio, ma dovete modificarlo per adattarlo alla vostra situazione.

```
rem mwave è un file batch fornito dal costruttore del modem
call c:\mww\dll\mwave start
rem loadlin.exe è un programma DOS che carica Linux da DOS (Vedere Config-HOWTO).
c:\linux\loadlin f:\vmlinuz root=/dev/hda3 ro
```

Si potrebbe creare un icona sul desktop di Windows che punti a questo file batch, quindi impostare la proprietà dell'icona su Esegui in modalità MSDOS. Poi cliccando su questa icona si imposta il modem, quindi si passa a Linux. Un altro modo possibile per caricare Linux dal DOS è premere CTL-ALT-CANC dicendo al sistema operativo di fare un reboot (a patto che abbiate impostato le cose in modo che possiate caricare direttamente Linux alla ripartenza del PC). Il modem rimane sulla stessa porta com (stesso indirizzo IO) che è stato usato sotto DOS.

Il modem Newcom ifx necessita di una piccola modifica al kernel per farlo funzionare correttamente visto che la sua simulazione della porta seriale non è standard. La modifica ed altre informazioni per usare questo modem con Linux si trova in `http://maalox.pharmacy.ohio-state.edu/~ejolson/linux/newcom.html` .

2.6.2 Driver Rockwell (RPI)

I modem che richiedono i driver Rockwell RPI (Protocollo di Interfaccia Rockwell) possono essere usati con linux anche il software driver funziona solo sotto Windows. Questo accade perché il software Windows di cui non si dispone, esegue solo compressione e correzione di errori. Se vi va bene usare un modem senza compressione dati e correzione errori, allora lo potete usare con Linux. Per fare questo dovete disabilitare il RPI inviando al modem (tramite stringa di inizializzazione) un comando disabilita RPI ogni volta che accendete il modem. Sul mio modem il comando è +HO. Non avere a disposizione la compressione dati potrebbe non essere poi quel gran svantaggio visto che la maggior parte dei grossi file che scaricate da Internet sono già compressi ed ulteriori tentativi di compressione potrebbero tramutarsi in un piccolo rallentamento.

3 Modem Pool (Gruppi di modem), Modem digitali

Un modem pool è un gruppo di modem sulla stessa scheda (come le schede modem multiporta) oppure molti modem su di un chassis esterno (qualcosa come un modem esterno). I modem possono essere analogici simili a quelli usati dai PC di casa o ufficio (non possono inviare a 56k anche se sono Modem a 56k). Potrebbero anche essere modem digitali che possono inviare dati a 56k. Per 56k in realtà intendo tutte le velocità sopra i 33,6k, visto che i modem a 56k non possono avere una vera velocità di 56k. I modem digitali richiedono una connessione digitale alla linea telefonica e non usano alcuna porta seriale. Tutti questi modem pools esigono l'installazione di driver speciali per loro.

3.1 Modem pool analogici, schede modem multiporta

Essi sono parecchi modem analogici (quelli comuni da casa/ufficio) forniti sia su di una scheda plug-in che su di uno chassis esterno. Ogni come ha una porta seriale al suo interno. C'è in genere un sistema di condivisione di interrupt o di gestione di interrupt nella loro circuteria, quindi parecchio carico viene levato alla CPU. Notate che questi modem non sono modem digitali e quindi non saranno capaci di usare 56k quando vengono chiamati.

Ecco un elenco di alcune ditte che costruiscono schede modem multiporta. In genere ci sono 8 modem per scheda. Le schede sottoelencate funzionano sotto Linux ed i siti web dovrebbero guidarvi ad un driver per esse.

Schede modem multiporta:

- MultiModemISI di Multi-Tech Systems. 56K o 33.6K, PCI o ISA, 4 od 8 porte. ibridi ISDN/56K.

  `http://www.multitech.com/products/`

- RAStel di Moreton Bay Products. 56K PCI o ISA, 4 o 8 porte. Anche 2 modem + 2 porte seriali libere (?).

  `http://www.moretonbay.com.au/MBWEB/product/rastel/rastel.htm`

- RocketModem di Comtrol. ISA 33.6K, 4 od 8 porte.

  `http://www.comtrol.com/SALES/SPECS/Rmodem.htm`

- AccelePort (RAS Family) di Digi.

  `http:/www.dgii.com/digi.cfm?p=940564.pi.prd.00000046`

3.2 Modem digitali

I modem digitali sono molti diversi da quelli analogici che la maggior parte delle persone usano sui propri PC. Esse richiedono una connessione digitale alla linea telefonica e non usano porte seriali per interfacciarsi al computer. Invece, si interfacciano direttamente al PC tramite una scheda speciale (che potrebbe anche contenere il modem digitale). Essi sono in grado di inviare a 56k, cosa che i modem analogici non sono in grado di fare. Essi rappresentano spesso una componente dei remote access servers (server di accesso remoto) o digital modem pools (gruppi di modem digitali).

I cavi che portano i segnali digitali dalla compagnia telefonica sono stati concepiti per alte larghezze di banda così che gli stessi cavi possono portare diverse chiamate telefoniche. Questo viene fatto tramite il time-division multiplexing. Così il primo compito da eseguire è quello di separare le chiamate telefoniche ed inviare ciascuna di esse al proprio modem digitale. C'è anche il compito da svolgere nella direzione opposta combinando tutte le singole chiamate in un'unica linea. Questi compiti sono svolti da qualcosa che è talvolta chiamata .. concentrator

Il modem digitale trasporta il segnale digitale dalla compagnia telefonica e, dopo averlo elaborato, lo mette sul bus del PC (presumibilmente inviandolo in un buffer in memoria). Allo stesso modo, gestisce l'invio di segnali digitali nella direzione opposta verso una linea telefonica digitale. Quindi esegue solomente una conversione digitale a digitale e non tratta dati analogici in alcun modo. Quindi non è veramente un modem visto che non modula nessuna portante analogica. Quindi il nome modem digitale è improprio ma esso svolge un compito che in precedenza spettava ai modem. Così alcuni modem seriali definiscono se stessi come digital signal processors (processori di segnale digitale), remote access server (server di accesso remoto) ecc. e potrebbero anche non menzionare mai la parola modem. Questa è una terminologia tecnicamente corretta.

Detti sistemi possono esser un server proprietario a se stante, uno chassis che contiene modem digitali che si connettono ad un PC tramite una speciale scheda di interfaccia, o semplicemente essere una scheda essi stessi. Digi definisce detta scheda un remote access server concentrator adapter (adattatore per concentratore di server di accesso remoto). Una descrizione non completa di quello che serve per diventare un ISP (Internet Service Provider) è: Vedere *Cosa mi serve per essere un ISP?* . an ISP?>. Cyclades promuove i propri prodotti quindi è consigliabile fare qualche confronto sui prezzi prima di comprare qualsiasi cosa.

Prima di leggere questo

4 Modem e porte seriali Nozioni di base

Non occorre capire le fondamenta per usare ed installare un modem. Ma esserne a conoscenza può aiutare a capire cosa c'è che non va quando sorgono dei problemi. Dopo avere letto questa sezione, se si vuole approfondire, conviene consultare la sezione 19 (Come funzionano i modem) in questo documento (ancora incompleta). Maggiori dettagli sulla porta seriale (inclusa la maggior parte di questa sezione) potrà essere trovata in Serial-HOWTO.

4.1 Il modem converte da digitale ad analogico (e viceversa)

La maggior parte delle principali linee telefoniche sono già digitali ma le linee che portano presso la vostra abitazione (o posto di lavoro) sono generalmente analogiche, il che vuol dire che sono state predisposte per trasmettere un'onda elettrica che è l'esatta replica dell'onda sonora generata dalla vostra voce. Un'onda elettrica di questo tipo è chiamata analogica. Se vista in un oscilloscopio sembra una curva sinusoidale di varia frequenza ed ampiezza. Un segnale digitale assume invece una forma squadrata. Ad esempio 3 v (volt) potrebbe essere un bit con valore 1, e 0 v potrebbe corrispondere ad un bit di valore 0. Per la maggior parte delle porte seriali (usate dai modem esterni) +12 v equivale ad un bit 0, e -12 v ad un bit 1 (alcune porte hanno valori di + o - 5 v).

Per inviare dati dal proprio computer attraverso la linea telefonica, il modem acquisisce il segnale digitale dal computer e lo converte in analogico. Lo fa prima creando un'onda sinusoidale analogica, quindi MODulandola. Visto che il risultato rappresenta comunque un dato digitale, potrebbe anche chiamarsi segnale digitale invece che analogico. Ma assomiglia ad un segnale analogico e quasi tutti lo chiamano analogico. Dall'altro capo della linea telefonica un altro modem DEModula questo segnale recuperando il segnale puro digitale originale. Mettete insieme l'inizio delle due parole sopracitate: mod e dem ed ecco l'origine della parola modem (ovviamente dovete togliere una delle due d). Un modem è quindi un MODulatore-DEModulatore. Per sapere cos'è la modulazione occorre consultare la sezione 19.1 (La modulazione in dettaglio).

4.2 Cos'è una porta seriale?

4.2.1 Introduzione alla seriale

La porta seriale è un dispositivo di I/O (Input/Output).

Visto che i modem hanno una porta seriale frapposta tra loro ed il computer, è necessario conoscere la porta seriale così come il modem.

La porta seriale è un dispositivo di IO (Inupt/Output). La maggior parte dei PC hanno due porte seriali. Ciascuna ha un connettore a 9 pin (talvolta a 25 pin) sul retro del computer. I programmi per computer possono inviare dati (byte) al pin di trasmissione (output) e ricevere dati dal pin di ricezione (input). Gli altri pin servono per controlli e per la messa a terra.

La porta seriale è molto di più che un semplice connettore. Essa converte i dati da paralleli a seriali e cambia la rappresentazione elettrica dei dati. All'interno del computer, i bit di dati scorrono parallelamente (usando diversi cavi allo stesso tempo). Il flusso seriale è uno scorrere di dati attraverso un solo cavo (così come sul pin di trasmissione e ricezione del connettore seriale). Perché la porta seriale possa creare un flusso di questo tipo, deve convertire i dati da paralleli (all'interno del PC) a seriali (e viceversa).

La maggior parte dell'elettronica della porta seriale si trova in un chip del computer (o in una sezione di un chip) conosciuto come UART. Per maggiori dettagli sugli UART consultare la sezione 15 (Cosa sono gli UART? In che modo influenzano le prestazioni?) Ma potreste volere prima finire questa sezione, così da poter meglio capire come l'UART si pone all'interno dello schema globale delle cose.

4.2.2 Pin e cavi

I vecchi PC usavano connettori a 25 pin ma in realtà se ne usano circa 9, per cui la maggior parte dei connettori attuali sono di soli 9 pin. Ognuno dei quali è generalmente connesso ad un cavo. Oltre ai due cavi usati per ricevere e trasmettere i dati, un altro pin (cavo) è la messa a terra. Il voltaggio di ogni cavo è misurato in relazione al cavo di terra. Quindi il numero minimo di cavi da usare per una trasmissione bilaterale di dati è 3. È possibile anche fare a meno del segnale di terra ma con degradate prestazioni e talvolta con errori.

Ci sono altri cavi che servono per effettuare solo dei controlli (invio di segnali) e non per inviare byte. Tutti questi segnali potrebbero essere condivisi da un unico cavo, ma, al contrario, esiste un cavo separato dedicato ad ogni tipo di segnale. Alcuni (o tutti) questi cavi di controllo sono chiamati linee di controllo del modem. Questi cavi di controllo sono impostati (on) a +12 volt oppure nello stato negativo (off) a -12 v. Uno di questi cavi segnala al computer di interrompere l'invio di dati attraverso la porta seriale. Al contrario, un altro cavo segnala al dispositivo connesso alla porta seriale di interrompere l'invio di dati al computer. Se il dispositivo connesso è un modem, altri cavi possono segnalare al modem di appendere la comunicazione o dire al computer che la connessione alla linea telefonica è stata effettuata o che il telefono sta squillando (cioè qualcuno sta tentando di connettersi). Vedere il Serial-HOWTO: Pinout and Signals per ulteriori dettagli

4.2.3 Il modem interno contiene una porta seriale

Per un modem interno non si sono connettori a 9 pin, ma il comportamento è quasi esattamente come se i cavi summenzionati esistessero. Invece di un segnale a 12 volt attraverso un cavo che porta lo stato di una linea di controllo del modem, il modem interno usa semplicemente un bit di stato nella propria memoria (un registro) per determinare lo stato di questo cavo virtuale. La porta seriale dei modem interno è vista dal computer proprio come una porta seriale reale. Ivi inclusi anche i limiti di velocità che si possono impostare nelle porte seriali ordinarie come ad esempio 115200 bit per secondo. Sfortunatamente per Linux molti modem interni attuali non funzionano esattamente in questo modo ma invece usano un software (eseguito dalla CPU) per svolgere la maggior parte del lavoro del modem. Sfortunatamente, questo software è disponibile solo per i sistemi operativi MS Windows (non sono stati portati su Linux). Quindi non potete usare la maggior parte di questi modem con Linux. Vedere 2.4 (Software Modem (interni)).

4.3 Indirizzi IO & IRQ

Poiché il computer deve comunicare con ciascuna porta seriale, il sistema operativo deve sapere che ciascuna porta seriale esiste e dove essa si trovi (il suo indirizzo di I/O). Esso deve anche conoscere quale cavo (numero di IRQ) deve usare la porta seriale per richiedere i servizi della CPU del PC. Quindi ogni dispositivo di porta seriale deve immagazzinare nella propria memoria non volatile sia l'indirizzo di I/O che il suo Interrupt reQuest Number: IRQ. Vedere 4.5 (Interrupt). Il bus PCI non funziona esattamente in questo modo visto che il bus PCI ha il suo proprio sistema di interrupt. Ma, visto che il BIOS che riconosce il PCI imposta i chip per mappare questi interrupt PCI come IRQ, praticamente si comporta proprio come descritto qui sopra ad eccezione del fatto che la condivisione degli interrupt è concessa (2 o più dispositivi possono usare lo stesso numero di IRQ).

Gli indirizzi I/O non sono uguali agli indirizzi di memoria. Quando un indirizzo I/O viene immesso nel bus indirizzi (address bus) del computer, un altro cavo viene elettrificato. Questo dice sia alla memoria principale di ignorare l'indirizzo, che a tutti i dispositivi che hanno indirizzi I/O (come le porte seriali) di controllare quell'indirizzo per vedere se combacia con quello del dispositivo. Se c'è corrispondenza, allora il dispositivo di I/O legge il dato sul bus dati.

4.4 Nome: ttyS0, ttyS1, ecc.

Le porte seriali sono etichettate come ttyS0, ttyS1, ecc. (generalmente corrispondenti a COM1, COM2, ecc. in DOS/Windows). La directory /dev ha un file speciale per ogni porta. Digitate ls /dev/ttyS* per vederli. Il fatto che possa esistere (ad esempio) un file ttyS3, non significa necessariamente che esista anche una corrispondente porta seriale fisica. Quale di questi nomi (ttyS0, ttyS1, ecc.) si riferisca a quale porta seriale viene determinato come segue. Il driver seriale (software) mantiene una tabella che mostra quale indirizzi I/O corrisponde a quale ttyS. Questa mappatura di nomi (tipo ttyS1) riferita a indirizzi I/O (e IRQ) può essere sia impostata che verificata dal comando setserial. Vedere 9.1 (Cos'è Setserial). Questo `non` imposta gli indirizzi di IO e di IRQ sull'hardware stesso (che è impostato invece dai ponticelli o tramite software plug-and-play). Quindi quale porta fisica corrisponda ad esempio a ttyS1 dipende sia da quello che il driver della seriale pensa (tramite setserial) che da quello che è impostato nell'hardware. Se viene fatto un errore, la porta fisica potrebbe non corrispondere ad alcun nome (tipo ttyS2) e quindi con può essere usata. Vedere 8 (Dispositivi di porta seriale /dev/ttyS2, ecc.) per ulteriori dettagli.

4.5 Interrupt

I byte entrano dalla linea telefonica al modem, sono converiti da analogico a digitale dal modem, quindi passati attraverso la porta seriale verso la loro destinazione all'interno del vostro computer. Quando la porta seriale riceve un certo numero di byte (potrebbe essere impostata a 1, 4, 8 o 14) nel suo buffer FIFO, segnala alla CPU di recuperarli inviando un segnale elettrico noto come interrupt su di certo cavo generalmente usato solo da quella porta. Quindi il FIFO attende un certo numero di byte quindi genera un interrupt.

Comunque questo interrupt potrebbe anche essere inviato se c'è un inaspettato ritardo mentre si attende l'arrivo del prossimo byte (conosciuto come timeout). Quindi se i byte sono ricevuti lentamente (come ad esempio se qualcuno digita su una tastiera di terminale) potrebbe essere generato un interrupt per ogni byte ricevuto. Per alcuni chip UART la regola è questa: se si possono ricevere 4 byte di seguito, ma nessuno di questi 4 arriva, allora la porta smette di aspettare altri byte ed invia un interrupt per recuperare i byte attualmente nel FIFO. Naturalmente, se il FIFO è vuoto, non verrà inviato nessun interrupt.

Ogni interrupt (all'interno del computer) è contraddistinto da un numero (IRQ) e la porta seriale deve sapere quale conduttore usare per inviare il segnale. Ad esempio, ttyS0 normalmente usa l'IRQ n. 4 conosciuto come IRQ4 (o IRQ 4). Una lista di questi interrupt ed altro può essere trovata in man setserial (cercare Configuring Serial Ports). Gli interrupt sono inviati ogniqualvolta la porta seriale necessiti di ricevere l'attenzione della CPU. È importante fare questo periodicamente poiché il buffer all'interno della porta seriale può trattenere solo 16 (1 nelle vecchie porte seriali) byte in arrivo. Se la CPU non riesce a rimuovere prontamente i byte ricevuti, poi non esiste più spazio libero per gli altri che stanno arrivando ed il piccolo buffer potrebbe sovraccaricarsi generando una perdita di byte di dati.

Per un modem esterno non c'è nessun modo (tipo controllo di flusso) per interrompere il flusso abbastanza rapidamente da prevenire questo. Per un modem interno il buffer FIFO a 16 byte si trova sulla stessa scheda ed un buon modem non ci andrà a scrivere se questo è pieno. Quindi un buon modem interno non sovraccaricherà il buffer da 16 byte ma potrebbe avere bisogno di usare 4.9.4 (Controllo di flusso da modem a modem) per evitare che il modem stesso vada in sovraccarico. Questo rappresenta un vantaggio del modem interno rispetto ad un esterno.

Gli interrupt sono inviati anche quando la porta seriale ha appena mandato 16 dei suoi byte dal piccolo buffer di trasmissione verso il cavo esterno. In questo modo si fa spazio per 16 successivi byte da inviare all'esterno. L'interrupt notifica alla CPU il fatto così che si possano immettere ulteriori byte nel piccolo buffer di trasmissione per essere inviati. Ancora, quando una linea di controllo del modem cambia il proprio stato, viene inviato un interrupt. I buffer sopra menzionati sono tutti buffer hardware. La porta seriale ha anche degli ampi buffer nella memoria principale. Questo verrà spiegato più tardi.

Gli interrupt veicolano molte informazioni ma solo indirettamente. L'interrupt propriamente detto semplicemente dice ad un chip chiamato interrupt controller che una certa porta seriale necessita attenzione. L'interrupt controller poi invia il segnale alla CPU. La CPU attiva uno speciale programma per servire la porta seriale. Il programma viene chiamato routine di servizio di interrupt (interrupt service routine) che è parte del software del dispositivo seriale. Esso cerca di scoprire cosa è successo alla porta seriale, quindi svolge il compito come ad esempio il trasferimento di byte da (per) il buffer hardware della porta. Questo programma può facilmente scoprire cosa è accaduto poiché la porta seriale ha dei registri che puntano indirizzi di I/O conosciuti dal software del driver seriale. Questi registri contengono informazioni sullo stato della porta seriale. Il software legge questi registri e ispezionandone il contenuto, scopre cosa è accaduto, quindi esegue l'azione appropriata.

4.6 Compressione di dati (da parte del Modem)

Prima di continuare con le nozioni di base sulla porta seriale, occorre capire una certa cosa fatta dal modem: la compressione dei dati. In alcuni casi questo compito è in realtà svolto dal software gestito dalla CPU del computer ma, sfortunatamente, al momento attuale questo software funzione solamente in ambiente MS Windows. Ci occuperemo quindi del caso in cui il modem stesso esegue la compressione poiché questo è quello che accade così che il modem possa funzionare in ambiente Linux.

Per inviare dati più velocemente attraverso le linee telefoniche uno potrebbero comprimere (codificare) i dati usando uno schema di codifica personalizzato, che esso stesso dipende dai dati. Il dato codificato è più piccolo dell'originale (meno byte) e può essere inviato attraverso Internet in minore tempo. Questo processo è chiamato compressione di dati.

Se scaricate file da Internet, essi sono probabilmente già compressi e non è possibile per il modem tentare di comprimerli ulteriormente. Il vostro modem può riconoscere che quelli che stanno transitando sono dati già compressi ed astenersi dal tentare di comprimerli ancora. Se state ricevendo dati che sono stati già compressi dall'altro modem, il vostro modem li decomprimerà e creerà molti più byte di quelli che sono stati spediti attraverso la linea telefonica. Quindi il flusso di dati dal vostro modem all'interno del vostro computer sarà maggiore di quello dalla linea telefonica verso di voi. Il rapporto di questo flusso viene chiamato rapporto di compressione. Rapporti di compressione superiori a 4 sono possibili, ma non molto probabili.

4.7 Correzione d'errore

Analogamente alla compressione dati, i modem potrebbero essere impostati per eseguire una correzione d'errore. Sebbene questo comporti un abbassamento del flusso di byte/secondo, il fatto che questa correzione d'errore tolga i bit di inizio e fine in realtà accresce il flusso di bit/secondo.

Per l'interfaccia della porta seriale con il mondo esterno, ogni byte composto da 8 bit ha 2 ulteriori bit aggiunti ad esso: un bit di inizio e un bit di stop. Senza la correzione di errore, questi bit di inizio e stop extra generalmente passano attraverso il modem verso la linea telefonica. Ma quando la correzione d'errore è attivata, questi bit extra sono eliminati e i byte di 8 bit sono composti in pacchetti. Questo è più efficiente e genera un flusso di bit/secondo più alto a dispetto del fatto che c'è qualche ulteriore byte aggiunto per l'intestazione dei pacchetti e per la correzione degli errori

4.8 Flusso di dati (velocità)

I dati (byte che rappresentano caratteri, immagini, ecc.) passano dal vostro computer al vostro modem, quindi all'esterno verso la linea telefonica (e viceversa). I rapporti di flusso (come ad esempio 56k (56000) bit/secondo) sono chiamati (non correttamente) velocità. Ma quasi tutti dicono velocità al posto di rapporto di flusso. Se non c'è compressione di dati il rapporto di flusso dal computer al modem dovrebbe essere circa lo stesso di quello che passa attraverso la linea telefonica.

In realtà ci sono due differenti velocità da considerare al vostro capo della linea telefonica.

- La velocità della linea telefonica stessa (velocità DCE) da modem-a-modem
- La velocità dalla porta seriale del vostro computer al vostro modem (velocità DTE)

Quando si compone un numero per connettersi ad un altro modem all'altro capo della linea telefonica, il vostro modem spesso visualizza un messaggio tipo CONNECT 28800 oppure CONNECT 115200. Cosa significa?. Bè, potrebbe essere sia la velocità DCE che quella DTE. Se essa è maggiore di quella delle specifiche del modem, allora deve trattarsi della velocità da-modem-a-computer (DTE). Questo è l'esempio di CONNECT 115200 mostrato in precedenza. 28800 deve trattarsi invece della velocità da-modem-a-modem (DCE) visto che la porta seriale non ha questa velocità. Si potrebbe configurare il modem perché possa visualizzare entrambe le velocità. Alcuni modem visualizzano entrambe le velocità e visualizzano la velocità da modem-a-modem come (ad esempio): CARRIER 28800

Se avete un modem interno non vi aspettereste che ci sia un limite di velocità DTE dal vostro modem al vostro computer visto che il modem risiede all'interno del computer ed è praticamente una parte di esso. Ma questo limite c'è visto che il modem contiene al suo interno una porta seriale dedicata.

È importante capire che la velocità media è spesso minore di quella specificata, specialmente nella corta linea DTE (dal computer a modem). Attese (o tempi morti) generano una minore velocità media. Queste attese possono essere lunghe attese di forse un secondo a causa del 4.9 (Controllo di flusso). Di contro le attese possono anche essere molto brevi (tempi morti) corrispondenti a diversi micro-secondi che separano la fine di un byte e l'inizio dell'altro. In più, i modem passano a velocità inferiori se le condizioni della linea telefonica sono meno che perfette. Per una discussione riguardo quale sia la migliore velocità DTE vedere la sezione 13 (Quale velocità dovrei usare).

4.9 Controllo di flusso

Il controllo di flusso è la capacità di fermare il flusso dei byte in un cavo. Deve anche provvedere a fare ripartire il flusso senza perdere byte. Il controllo di flusso è necessario ai modem per consentire un salto nei rapporti di velocità.

4.9.1 Esempio di controllo di flusso

Ad esempio, consideriamo il caso in cui voi connettiate il vostro modem esterno a 36.6k tramite un corto cavo alla vostra porta seriale. Il modem invia e riceve byte attraverso la linea telefononica e 36.6k bit per secondo (bps). Non esegue nessuna compressione dati o correzione di errore. Voi avete impostato la velocità della porta seriale a 115,200 bit/secondo (bps) e state inviando dati dal vostro computer alla linea telefonica. Quindi il flusso dal vostro computer al vostro modem attraverso il corto cavo è di 115.2k bps. In ogni caso il flusso da modem verso la linea telefonica è solo di 33.6k. Visto che il flusso di dati (115.2k) sta entrando nel modem più velocemente del flusso di dati in uscita, il modem deve conservare il flusso in eccesso (115.2k -33.6k = 81.6k) in uno dei suoi buffer. Questo buffer andrebbe fatalmente in sovraccarico (esaurirebbe lo spazio a disposizione) a meno che il flusso a 115.2k venga interrotto.

Ecco che il controllo di flusso viene in soccorso. Quando il buffer del modem è quasi pieno, il modem invia un segnale di stop alla porta seriale. La porta seriale passa il segnale di stop al device driver ed il flusso a 115.2k bps viene fermato. Intanto il modem continua ad inviare dati a 33.6k bps recuperando i dati precedentemente accumulati nei suoi buffer. Visto che ora non sta arrivando niente nel buffer, il livello di byte inizia a diminuire. Quando non sono rimasti che pochi byte nel buffer, il modem invia un segnale di partenza alla porta seriale ed il flusso a 115.2k dal computer al modem riprende. In effetti, il controllo di flusso crea un rapporto di flusso medio (in questo caso 33.6k) che è significativamente inferiore a quello in entrata di 115.2k bps. Questo è il controllo di flusso start-stop.

Quello di cui sopra è un semplice esempio di flusso di controllo per il flusso da computer al modem, ma esiste anche il controllo di flusso usato nella direzione opposta: dal modem (od altro dispositivo) al computer. Ogni direzione di flusso coinvolge 3 buffer: 1. quello nel modem 2. quello nel chip UART (detto FIFO) 3. nella memoria principale, gestito dal driver seriale. Il controllo di flusso protegge alcuni buffer dal rischio di sovraccarico. I piccoli buffer UART FIFO non sono protetti in questo modo ma dipendono invece da una veloce risposta agli interrupt che essi generano. FIFO significa First In First Out (il primo che entra è il primo che esce) e rappresenta quindi il modo in cui vengono gestiti i byte. Tutti e 3 i buffer usano la regola FIFO ma solo uno di essi si identifica anche con questo nome. Questa è l'essenza del controllo di flusso ma ci sono ancora ulteriori dettagli.

Non dovreste averne bisogno spesso del controllo di flusso nella direzione modem - PC. Per un complesso esempio di un caso dove è richiesto vedere Complex Flow Control Example nel Serial-HOWTO. Ma se non avete una velocità impostata tra il modem ed il computer (velocità della porta seriale) sufficientemente alta, allora dovrete rallentare il flusso dal modem al PC. Per fare questo dovete fermare l'incombente flusso di byte dalla linea telefonica. Il vostro modem deve dire all'altro modem di interrompere l'invio. Vedere 4.9.4 (Controllo di flusso da-modem-a-modem)

4.9.2 Controllo di flusso hardware contro il controllo di flusso software

Se possibile, è meglio usare il flusso di controllo hardware che usa due linee dedicate di controllo del modem per inviare i segnali di stop e start. I modem moderni usano quasi sempre il controllo di flusso hardware tra il modem e la porta seriale

Il controllo di flusso software usa le linee principali di ricezione e trasmissione per inviare i segnali di start e stop. Usa i caratteri di controllo ASCII DC1 (start) e DC3 (stop) a questo scopo. Essi sono semplicemente inseriti nel regolare flusso di dati. Il controllo di flusso software è non solo più lento nel reagire ma anche non consente l'invio di dati binari a meno di prendere speciali preacauzioni. Visto che è probabile che dati binari possano contenere DC1 e DC3, particolari accorgimenti devono essere presi per distinguere tra un DC3 che significa uno stop del controllo di flusso ed un DC3 che è parte del codice binario. La stessa cosa per DC1. Per far funzionare il controllo di flusso software con i dati binari occorre il supporto sia del modem (hardware) che del software.

4.9.3 Sintomi della mancanza di un controllo di flusso

Conoscere la teoria del controllo di flusso può essere di uso pratico. Per esempio usavo il mio modem per accedere ad Internet e tutto sembrava funzionare bene. Ma dopo alcuni mesi ho provato ad inviare un grosso file dal mio PC al ISP ottenenendo un grande numero di errori e ritrasmissioni (ma alla fine con Kermit sono riuscito a spedire un grosso file dopo parecchi tentativi). La ricezione nell'altra direzione (dal ISP a me) funzionava bene. Il problema risultò essere nella disabilitazione del controllo di flusso. Il buffer del mio modem si sovraccaricava durante l'invio di grossi file visto che nessun segnale di stop era mai inviato al computer per interrompere l'invio di dati al modem. Non c'erano problemi nella direzione dal modem al mio computer visto che la capacità (diciamo 115.2k) era sempre superiore del flusso attraverso la linea telefonica. La risoluzione consistette nell'abilitare il controllo di flusso inserendo un comando di attivazione del controllo di flusso nella stringa di inizializzazione del modem (avrebbe dovuto essere abilitato per default ma qualcosa era andato storto).

4.9.4 Controllo di flusso da-modem-a-modem

Questo è il controllo di flusso dei dati inviati attraverso le linee telefoniche tra due modem. In pratica, esso esiste solo quando è attivata la correzione di errori. In verità, anche senza correzione di errore è possibile attivare il controllo di flusso software tra modem, ma esso potrebbe interferire con l'invio di dati binari così non viene usato spesso.

4.10 Il percorso del flusso di dati; Buffer

Sebbene sia stato detto molto in proposito, tra cui il controllo di flusso, un paio di buffer FIFO a 16 byte nelle porte seriali (nell'hardware) ed un paio di buffer più ampi all'interno del modem, esistono ancora un altro paio di buffer. Essi sono degli ampi buffer (forse di 8k) nella memoria principale conosciuti anche come buffer di porta seriale. Quando un programma applicativo invia byte alla porta seriale (e al modem), essi vengono posti prima nel buffer di trasmissione della porta seriale nella memoria principale. La coppia consiste in questo buffer di trasmissione ed in quello di ricezione per il flusso di byte dalla parte opposta.

Il device driver seriale estrae diciamo 16 byte dal buffer di trasmissione, un byte alla volta e li mette nel buffer di trasmissione da 16 byte della porta seriale per la trasmissione. Una volta in questo buffer, non c'è modo di impedire che essi vengano trasmessi. Essi sono poi inviati al modem che dispone anch'esso di un buffer di dimensioni adeguate (diciamo 1k). Quando il device driver (che riceve ordini dal controllo di flusso) interrompe il flusso dei byte in uscita dal computer, interrompe in realtà il flusso di byte in uscita dall'ampio buffer di trasmissione della memoria principale. Anche dopo che questo è accaduto ed il flusso verso il modem è stato fermato, un programma applicativo può continuare a spedire i byte presenti nel buffer da 8k fino a che esso non si riempie.

Quando è pieno, il programma applicativo non può inviargli ulteriori byte (un istruzione di scrittura write in un blocco di programma in C) e l'applicazione si interrompe temporaneamente ed attende fino a che si libera un poco di spazio nel buffer. Quindi uno stop esercitato dal controllo di flusso è in definitiva capace di fermare il programma che sta inviando i byte. Anche se questo programma si interrompe il computer non smette necessariamente di elaborare. Potrebbe passare ad eseguire altri processi mentre sta aspettando causa lo stop del controllo di flusso. Quello suindicato era un esempio ultrasemplificato visto che un'altra alternativa è quella di fare sì che il programma applicativo stesso faccia qualcosa d'altro mentre sta attendendo di scrivere

4.11 I comandi del modem

I comandi al modem sono inviati ad esso dal programma di comunicazione attraverso lo stesso conduttore usato per inviare dati. I comandi sono delle brevi stringhe ASCII. Esempi sono AT&K3 per abilitare il controllo di flusso hardware (RTS/CTS) tra il computer ed il modem; e ATDT5393401 serve per comporre il numero 5393401. Notate che tutti i comandi sono prefissati da AT. Alcuni comandi come l'attivazione del controllo di flusso aiutano a configurare il modem. Altri comandi come il comporre un numero fanno veramente qualcosa. Ci sono circa un centinaio di differenti possibili comandi. Quando il vostro software di comunicazione parte, lancia una stringa di inizializzazione init string composta da comandi al modem per configurarlo. Tutti i comandi sono inviati sulla linea ordinaria dei dati prima che il modem componga un numero (o riceva una chiamata).

Una volta che il modem è connesso ad un altro modem (modo on-line), tutto quello che il vostro computer manda al vostro modem va direttamente verso l'altro modem e non è interpretato dal modem come un comando. C'è un modo per fuggire da questo modo operativo e tornare al modo comandi dove tutto quello che viene inviato al modem viene interpretato come un comando. Il computer invia semplicemente +++ con un determinato periodo di tempo prima e dopo questo +++. Se lo spazio temporale è corretto, il modem si pone in modo comandi. Un altro modo di fare questo è tramite un segnale ad una certa linea di controllo del modem.

Ci sono svariate liste dei comandi modem su Internet. La sezione 18.5 (Siti Web) ha dei collegamenti ad un paio di questi siti. Diversi modelli e marche di modem non usano esattemente lo stesso gruppo di comandi. Così quello che va bene per un modem potrebbe non andare bene per un altro. Alcuni comandi comuni (non si garantisce che funzionino su tutti i modem) sono elencati in questo HOWTO nella sezione 7.4 (Altri comandi modem)

4.12 Software seriale: il modulo del device driver

Il device driver per la porta seriale è il software che fa funzionare la porta seriale. Viene ora fornito come modulo seriale. Questo modulo viene generalmente caricato automaticamente se necessario. Il kernel 2.2+ farà questo. Nei kernel precedenti, dovete avere `kerneld in esecuzione per far sì che i moduli si autocaricano su richiesta. Altrimenti il modulo seriale necessita di essere esplicitamente elencato in` etc/modules. Prima che i moduli divenissero popolari con Linux, il driver seriale era generalmente costruito all'interno del kernel. Se esso è ancora incorporato nel kernel (potreste avere selezionato questa opzione quando avete compilato il kernel) non lasciate che il modulo seriale venga caricato. Se lo fate finirete con avere due driver seriali, viene rilevato che non potete usare le porte seriali ed otterrete un errore I/O error se tentate di aprirle.

Quando il modulo seriale è caricato, visualizza un messaggio sullo schemo circa l'esistenza di porte seriali (spesso mostrando un IRQ errato). Ma una volta che il modulo è usato da `setserial` per dire al device driver qual'è l'IRQ corretto, allora dovreste vedere una seconda schermata simile alla prima ma con il corretto IRQ, ecc. Vedere 9.1 (Cos'è Setserial) per ulteriori informazioni su `setserial`.

Si potrebbe modificare il driver modificando il codice sorgente del kernel. La maggior parte del driver seriale si trova nel file serial.c. Per dettagli inerenti la scrittura di programmi per la porta seriale vedere Serial-Programming-HOWTO (attualmente in fase di revisione da parte di Vern Hoxie).

5 Configurazione: introduzione

Se volete usare il modem solo in ambiente MS Windows/Dos, allora potete installare praticamente qualsiasi modem e tutto andrà bene. In un ambiente Linux non è in genere così facile a meno di usare un modem esterno. Tutti i modem esterni dovrebbero funzionare bene (anche se sono etichettati come Plug and Play), Ma anche la maggior parte dei nuovi modem interni sono Plug-and-Play (PnP) ed hanno porte seriali PnP. Se è un modem ISA potreste avere bisogno di usare il programma Linux isapnp per configurarle. Vedere il Plug-and-Play HOWTO per ulteriori informazioni.

Visto che ogni modem ha una porta seriale associata la configurazione del modem si svolge in due parti:

- Configurazione del modem stesso: eseguita dal programma di comunicazione
- Configurazione della porta seriale del modem: eseguita solo parzialmente dal programma di comunicazione

La maggior parte delle configurazioni di cui sopra (ma non necessariamente la maggior parte dello sforzo) sono svolte dal programma di cumunicazione che si usa con il modem come ad esempio `minicom` o `seyon`, `wvdial` (per PPP). Se usate il modem per rendere disponibile il vostro computer dall'esterno, allora il programma `getty` che usate per presentare a chi si collega il prompt di login, sarà di aiuto nella configurazione. Vale a dire che per configurare il modem (e la maggior parte della porta seriale) dovete configurare il programma di comunicazione (tipo il PPP dialer o `getty`).

Sfortunatamente la configurazione di cui sopra non esegue la configurazione a basso livello della porta seriale: l'impostazione dei suoi indirizzi IO ed IRQ sia nell'harware che nel driver. Se siete fortunati, la cosa potrebbe accadere automaticamente quando caricate Linux. L'impostazione nell'hardware veniva in precedenza svolta dai jumper ma oggi è fatta dal software Plug-and-Play.

Ma c'è un serio problema: Linux (almeno fino al tardo 1999) non è un sistema operativo plug-and-play ma dispone di strumenti Plug-and-Play che potreste usare per impostare la configuzione sebbene essi non siamo molto facili da usare. Questo può essere un difficile problema per voi. La prossima sezione elaborerà più approfonditamente questo aspetto.

6 Configurare la porta seriale

6.1 Supporto per il bus PCI in fase di completamento

Il driver seriale del kernel 2.2 non contiene un supporto per il bus PCI. Ma i kernel 2.3 e 2.4 finalmente supporteranno alcune schede seriali PCI (e schede modem). La maggior parte delle schede PCI necessitano di uno speciale supporto nel driver. Il driver legge il numero identificativo memorizzato nella scheda per determinare come (o se) supportare la scheda. Se avete una scheda PCI che siete convinti non sia un winmodem ma non funziona comunque, allora potreste essere d'aiuto per tentare di creare un driver per essa. Per fare questo dovrete contattare il curatore del serial driver, Theodeore (Ted) Y. Ts'o. Ma per prima cosa controllate l'elenco dei modem al sito `http://www.o2.net/~gromitkc/winmodem.html`

per le ultime informazioni sui modem PCI e relativi argomenti.

Inviategli tramite posta elettronica una copia dell'output di lspci -vv con complete informazioni circa il modello ed il costruttore del modem PCI (o della porta seriale). Egli cercherà di approntarvi un driver di prova che potrebbe fare al caso vostro. Dovrete recuperarlo, compilarlo e possibilmente ricompilare il vostro kernel. Poi dovrete testare il driver per vedere se funziona bene e fare una relazione dei risultati a Ted Ts'o. Se siete disposti a fare tutto quanto sopradescritto (e questa è l'ultima versione di questo HOW-TO) allora inviategli quanto richiesto a: `mailto:tytso@mit.edu` .

I modem PCI sono ben standardizzati. Alcuni usano la memoria principale per comunicare con il PC. Se vedete indirizzi esadecimali a 8 cifre probabilmente non funzioneranno sotto Linux. Alcuni richiedono particolari abilitazioni dell'IRQ. L'output di lspci può aiutare a determinare se può essere supportato. Se vedete una porta IO a 4 cifre e nessun indirizzo di memoria lungo, il modem potrebbe funzionare semplicemente dicendo a setserial l'IO della porta e l'IRQ. Alcune persono hanno fatto funzionare un modem PCI 3COM 3CP5610 in questo modo

6.2 Introduzione alla configurazione

Nella maggior parte dei casi, la configurazione viene eseguita automaticamente e voi non dovete fare nulla. Ma talvolta dovete configurare (o semplicemente volete controllare la configurazione). Se è questo il caso, dovete per prima cosa sapere qualcosa circa le due parti necessarie per configurare la porta seriale sotto Linux.

La prima parte (configurazione a basso livello) è assegnare un indirizzo IO, un IRQ ed un nome (tipo ttyS2). Questa coppia IO-IRQ deve essere impostata nell'hardware e deve anche essere passata al driver seriale. Potremo chiamare questa parte in breve come configurazione di io-irq. `setserial` viene usato per informare il driver. I metodi PnP, i jumper, ecc, sono usati per impostare l'hardware. Dettagli saranno forniti successivamente. Se dovete configurare ma non comprendete alcuni dettagli è poi facile avere dei guai.

La seconda parte (configurazione ad alto livello) consiste nell'assegnare una velocità (tipo 38.4K bit/secondo), selezionare il controllo di flusso, ecc. Questo viene spesso fatto dai programmi di comunicazione come PPP, minicom, o da getty (che potreste lanciare sulla porta così che altri possano collegarsi attraverso di essa). In ogni caso dovrete dire a questi programmi quale velocità volete, ecc., usando un menu od un file di configurazione. Questa configurazione di alto livello può essere fatta anche con il programma `stty`. `stty` è anche utile per vedere lo stato corrente se avete dei problemi. Vedere anche la sezione Stty del Serial-HOWTO.

Quando Linux parte, viene compiuto un tentativo per rilevare e configurare (a basso livello) alcune porte seriali. Quello che accade esattamente dipende dal vostro BIOS, hardware, distribuzione di Linux, ecc. Se le porte seriali funzionano bene, potrebbe non esserci bisogno di effettuare ulteriori configurazioni. I programmi applicativi tendono spesso ad eseguire una configurazione di alto livello, ma potrebbere richiedere informazioni che voi gli dovreste fornire. Con le porte seriali Plug-and-Play (spesso inserite in un modem interno), la situazione è diventata più complessa. Ecco i casi in cui occorre eseguire una configurazione a basso livello (impostare gli indirizzi IRQ e IO):

- Volete usare più di due porte seriali
- Installazione di una nuova porta seriale (come nel caso di un modem interno)
- Avete problemi con la porta seriale(i)

Per i kernel 2.2+ dovreste essere capaci di usare più di due porte seriali senza configurare a basso livello, condividendo gi interrupt. La cosa funziona solo se l'hardware seriale lo supporta e potrebbe essere altrettanto difficile che configurare a basso livello. Vedere 6.6.2 (Condivisione di interrupt e i Kernel 2.2+)

La configurazione a basso livello (impostare gli indirizzi IRQ e IO) sembra causare più problemi (rispetto a quella ad alto livello), sebbene per la maggioranza sia completamente automatica e nessuna configurazione si debba effettuare. Quindi la quasi totalità di questa sezione verte su questo argomento. A meno che il driver seriale sappia il corretto indirizzo IRQ e IO la porta non funzionerà per niente. Probabilmente non sarà neanche individuata da Linux. Anche se essa fosse trovata, potrebbe lavorare in modo estremamente lento se l'IRQ è sbagliato. Vedere 16.10 (Estremamente lento: il testo appare sullo schermo lentamente e dopo lunghi ritardi).

Nel mondo Wintel, l'indirizzo IO ed IRQ sono chiamati risorse e quindi siamo configurando certe risorse. Ma ci sono molti altri tipi di risorse così che il termine potrebbe avere molti altri significati. Ricapitolando, la configurazione a basso livello consiste nell'impostare due valori (un numero di IRQ e un indirizzo IO) in due posti:

1. nei registri di memoria dell'hardware della porta seriale stessa
2. nel device driver (spesso lanciando `setserial` in fase di boot)

Potreste dare un occhiata ai messaggi di avvio (fase di boot). Essi sono in genere corretti. Ma se state avendo problemi, c'è una buona probabilità che alcuni di questi messaggi non mostrino la corretta configurazione dell'hardware (e d'altronde non sono deputati a farlo). Vedere 6.4 (Indirizzi I/O e IRQ: Messaggi in fase di boot).

6.3 Errori comuni commessi nel riconfigurare a basso livello

Ecco alcuni degli errori più comuni che possono compiere:

- comando setserial: Viene lanciato (senza l'opzione autoconfig) e si pensa che abbia controllato l'hardware (non l'ha fatto).
- messaggi di setserial: Essi vengono visualizzati sullo schermo in fase di boot ed erroneamente si crede che il risultato mostri come l'hardware sia veramente configurato.
- /proc/interrupts: Quando il loro dispositivo seriale non è in uso essi non vedono lì i suoi interrupt ed erroneamente concludono che la porta seriale non può essere trovata (o che non abbia un interrupt impostato).
- /proc/ioports: Si pensa che questa mostri la configurazione dell'hardware mentre mostra solo gli stessi dati (che possono anche essere errati) di setserial.

6.4 Indirizzi I/O e IRQ: messaggi in fase di boot

In molti casi le vostre porte verranno automaticamente configurate a basso livello in fase di boot (ma non sempre correttamente). Per vedere cosa sta succedendo, guardate i messaggi di avvio sullo schermo. Non trascurate di controllare i messaggi del BIOS prima che Linux venga caricato (nessun esempio mostrato qui). Questi messaggi BIOS possono essere arrestati premendo il tasto Pause. Usate Shift-PagSu per scorrere i messaggi dopo che sono passati sullo schermo. Shift-PagGiù li scorrerà nel senso opposto. Il comando `dmesg` potrebbe esser usato ogniqualvolta si voglia vedere alcuni messaggi ma spesso ne mancano di importanti. Ecco un esempio di messaggi in fase di boot (così come nel tardo 1999). Notate che ttyS00 è lo stesso di /dev/ttyS0.

```
Per prima cosa vedete quello che è stato rilevato (ma l'irq è solo un
ipotesi):

Serial driver version 4.27 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
ttyS02 at 0x03e8 (irq = 4) is a 16550A

Più tardi potete vedere quello che era stato salvato, ma anche questo
non è necessariamente corretto:
```

```
Loading the saved-state of the serial devices...
/dev/ttyS0 at 0x03f8 (irq = 4) is a 16550A
/dev/ttyS1 at 0x02f8 (irq = 3) is a 16550A
/dev/ttyS2 at 0x03e8 (irq = 5) is a 16550A
```

Notate che qui vi è un leggero disallineamento: il primo messaggio mostra ttyS2 a irq=4 mentre il secondo lo mostra a irq=5. Potreste anche avere solo il primo messaggio. In molti casi l'ultimo messaggio è quello corretto. Ma se state avendo problemi il messaggio potrebbe fuorviarvi. Prima di leggere la spiegazione di tutta questa complessità nel resto di questa sezione, potreste semplicemente provare ad usare la vostra porta seriale e vedere se tutto va bene. Se è il caso, potrebbe non essere essenziale leggere oltre.

Il secondo messaggio deriva dal programma `setserial` che viene lanciato in fase di boot. Mostra quella che il device driver pensa sia la corretta configurazione. Ma questo potrebbe essere sbagliato. Ad esempio l'irq potrabbe essere in realtà impostato a irq=8 nell'hardware (entrambi i messaggi sono sbagliati). irq=5 potrebbere esistere perché qualcuno ha incorrettamente impostato questo valore nel file di configurazione (o simile). Il fatto che Linux talvolta prenda degli IRQ sbagliati dipende dal fatto che non verifica gli IRQ. Semplicemente assume quelli standard (primo messaggio) o accetta quello che gli si dice quando viene configurato (secondo messaggio). Nessuno di questi è necessariamente corretto. Se il driver seriale ha l'IRQ sbagliato, la porta seriale è molto lenta o non funziona per niente.

Il primo messaggio è il risultato di Linux che verifica le porte seriali ma non verifica gli IRQ. Se una porta viene mostrata in questa fase essa esiste ma il suo irq potrebbe essere sbagliato. Linux non controlla gli IRQ perché il farlo non è a prova di errore. Esso assume che gli IRQ sono come mostrato perché questi sono i valori standard. Potreste controllare manualmente con `setserial` usando le opzioni `autoconfig` e `auto_irq` ma non si garantisce che sia esatto.

I dati mostrati nei messaggi del BIOS (che vedete per primi) è quello che è impostato nell'hardware. Se la porta seriale è Plug-and-Play PnP allora è possibile che `isapnp` venga lanciato e modifichi queste impostazioni. Cercate dei messaggi in questo senso dopo che Linux è partito. L'ultimo messaggio relativo alla porta seriale mostrato nell'esempio di cui sopra dovrebbe coincidere con i messaggi del BIOS (che è possibile siano stati modificati da isapnp). Se sono diversi allora dovreste aver bisogno di cambiare le impostazioni nell'hardware della porta od usare setserial per dire al driver quello che è attualmente impostato nell'hardware.

Inoltre, se avete porte seriali Plug-and-Play (PnP), Linux non le troverà a meno che l'IRQ e l'IO siano stati impostati all'interno dell'hardware dal software PnP. Questa è una comune ragione per la quale i messaggi di avvio non mostrano una porta seriale che fisicamente esiste. L'harware del PC (un BIOS PnP) potrebbe automaticamente configurare a basso livello questo. La configurazione PnP sarà spiegata più avanti.

6.5 Quali sono l'indirizzo IO e l'IRQ correnti della mia porta seriale?

La sezione precedente indica come tentare di fare questo guardando i messaggi di avvio. Se essi vi forniscono sufficienti informazioni, allra potreste anche saltare la lettura di questa sezione. Se no allora ci sono alcuni altri modi per scoprirli.

Ci sono davvero due risposte alla domanda Quali sono i miei IRQ e IO? 1. Quello che il device driver pensa che sia impostato (questo è quello che setserial in genere imposta e mostra). 2. Quello che in realtà è impostato nell'hardware. Potrebbe essere gli stessi. Se non losono avrete problemi visto che il driver ha informazioni sbagliate circa la porta seriale fisica. Se il driver ha l'IO sbagliato tenterà di inviare dati ad una porta seriale inesistente o, ancora peggio, ad un dispositivo esistente che non è una porta seriale. Se ha l'IRQ sbagliato il driver non riceverà le richieste di interrupt dalla porta seriale, causando una risposta molto lenta o mancante. Vedere 16.10 (Estremamente lento: il testo appare sullo schermo lentamente e dopo lunghi ritardi). Se ha il tipo sbagliato di UART potrebbero esserci problemi. Per determinare se entrambe le coppie di IRQ-IO sono identiche dovreste scoprire come sono impostati sia nel driver che nell'hardware.

6.5.1 Cosa pensa il device driver?

Questo è facile da scoprire. Basta guardare ai messaggi di avvio o digitare setserial -g /dev/ttyS*. Se tutto è a posto, allora quello che vedrete sarà impostato anche nell'hardware. Ci sono alcuni ulteriori modi di trovare queste informazioni guardando dei file nella directory /proc. Una ragione importante per comprendere questi ulteriori modi è per avvisarvi che essi mostrano solo quello che il device driver pensa che sia. Alcuni vedono certi file nella directory /proc ed erroneamente pensano che quello che vedono sia quello che è impostato nell'hardware ma non è necessariamente così.

/proc/ioports mostrerà gli indirizzi di IO che i driver stanno usando. /proc/interrupts mostra gli IRQ che sono stati usati dai driver dei processi attualmente in esecuzione (che hanno dispositivi aperti). Notate che in entrambi i casi di cui sopra potete solo vedere quello il driver pensa che sia e non necessariamente quello che è veramente impostato nell'hardware. /proc/interrupts mostra anche quanti interrupt sono stati invocati (spesso migliaia) per ogni dispositivo). Potete ricavare un indizio da questo perché se vedete un gran numero di interrupt invocati significa che c'è un qualche hardware da qualche parte che sta usando quell'interrupt. Talvolta il vedere solo pochi interrupt non significa che quell'interrupt sia stato fisicamente generato da una qualche porta seriale. Quindi se non vedete quasi interrupt per una porta che state cercando di usare, quell'interrupt potrebbe non essere stato impostato dall'hardware ad questo implica che il driver sta usando l'interrupt sbagliato. Per vedere /proc/interrupt per controllare su di un programma che state attualmente eseguendo (tipo minicom) dovete mantenere il programma in esecuzione mentre controllate. Per fare questo cercate di saltare in una shell senza uscire dal programma.

6.5.2 Cos'è impostato nell'hardware della mia porta seriale?

Come scoprire quali indirizzi IO e IRQ sono realmente impostati nel dispositivo hardware? Forse i messaggi del BIOS vi danno alcune informazioni prima che Linux inizi il caricamento. Usate il tasto shift-PagSu per risalire attraverso i messaggi di avvio e cercate i primissimi che provengono dal BIOS. Questa era la situazione prima che Linux partisse. Setserial non può modificarla ma isapnp o pciutils sì.

Un metodo brutale è cercare la rilevazione con setserial usando l'opzione autoconfig. Avrete bisogno di indovinare gli indirizzi per poterli poi verificare. Vedere 9.1 (Cos'è setserial). Per una porta seriale PCI, usate il comando lspci (per i kernel <2.2 guardate in /proc/pci). Se la vostra porta seriale è Plug-and-Play leggete le successive due sottosezioni.

Per una porta impostata tramite jumper, ecco come essi sono impostati. Se la porta non è Plug-and-Play (PnP) ma è stata impostata usando un programma DOS, allora è impostata secondo quanto deciso da chi ha lanciato quel programma.

6.5.3 Cosa è impostato nell'hardware della mia porta seriale?

Le porte PnP non mantengono la loro configurazione nell'hardware quando viene spento il PC. Questo è in contrasto con i jumper (non-PnP) che restano immutati anche quando si spegne la corrente. Se avete una porta ISA PnP, potrebbe raggiungere lo stato nella quale non abbia alcun indirizzo IRQ e IO e sia effettivamente disabilitata. Dovrebbe essere ancora possibile trovare la porta usando il programma `pnpdump`.

Per il Plug-and-Play (PnP) sul bus ISA si potrebbe tentare con il programma `pnpdump` (che è parte di `isapnptools`). Se usate l'opzione –dumpregs dovrebbe informarvi circa gli effettivi indirizzi IO e IRQ impostati nella porta.

Riguardo alle porte PnP controllarne la configurazione sotto DOS/Windows potrebbe non essere di molto aiuto. Windows mantiene le sue informazioni di configurazione nel suo Registro che non viene usato da Linux. Potrebbe fornire la memoria non volatile del BIOS di alcune informazioni ma potrebbe non essere in sicrono con quelle della configurazione corrente di Windows nel Registro ??. Se lasciate che un BIOS PnP faccia una configurazione automatica quando lanciate Linux (ed avete detto al BIOS che non avete un sistema operativo PnP quando fate partire Linux), allora Linux dovrebbe usare una qualsivoglia configurazione si trovi nella memoria non volatile del BIOS.

6.6 Scegliere gli IRQ seriali

Se avete un vero Plug-and-Play impostato dove sia il sistema operativo che il BIOS PnP configurano tutti i vostri dispositivi, non sceglierete i vostri IRQ. PnP determina quello che pensa sia meglio e li assegna. Ma se usate gli strumenti di Linux per il Plug-and-Play (isapnp e pcitools) allora dovete essere voi a sceglierli. Se già conoscete quale IRQ volete usate potete saltare questa sezione a meno che non vogliate sapere che l'IRQ O ha un uso speciale (vedere il paragrafo seguente).

6.6.1 L'IRQ 0 non è un IRQ

Sebbene IRQ 0 sia in realtà il timer (nell'hardware), esso ha uno speciale significato nell'impostare una porta seriale con setserial. Esso dice al driver che non c'è un interrupt per quella porta ed il driver allora userà metodi di polling. È piuttosto inefficiente ma può essere tentato se c'è un conflitto di interrupt o degli interrupt sono male impostati. Il vantaggio di assegnarlo è che non avete bisogno di sapere quale interrupt è impostato nell'hardware. Dovrebbe essere usato solo come espediente temporaneo fino a che non siate in grado di trovare un vero interrupt da usare.

6.6.2 Condivisione di interrupt e i Kernel 2.2+

La regola generale è che ogni dispositivo dovrebbe usare un IRQ unico e non condividerlo. Ma ci sono situazioni dove la condivisione è permessa come nella maggior parte delle schede multi-porta. Anche quando è permesso, potrebbe essere non molto efficiente visto che ogni volta che viene invocato un interrupt condiviso, occorre effettuare un controllo per determinare da dove proviene. Sebbene quindi sia possibile, è meglio attribuire ad ogni dispositivo il proprio interrupt.

Per i kernel precedenti il 2.2, gli IRQ seriali potevano essere condivisi tra di loro solo per la maggioranza delle schede multiporta. A partire dal kernel 2.2 gli IRQ seriali possono talvolta essere condivisi tra tutte le porte seriali. Per far sì che la condivisione funzioni nel kernel 2.2 esso deve essere compilato con CONFIG_SERIAL_SHARE_IRQ e l'hardware della porta seriale deve supportare la condivisione (così che se due seriali mettono due diversi voltaggi nello stesso cavo di interrupt, solo il voltaggio che significa questo è un interrupt prevarrà). Quindi anche se avete il 2.2, è meglio evitare la condivisione.

6.6.3 Quale IRQ scegliere?

L'hardware seriale spesso ha solamente un numero limitato di IRQ che possono essere impostati. Inoltre voi non volete dei conflitti di IRQ. Così non è che ci sia molta scelta. Il vostro PC di norma dovrebbe avere impostato `ttyS0` e `ttyS2` all'IRQ 4 e `ttyS1` e `ttyS3` all'IRQ 3. `/proc/interrupts` mostrerà quali IRQ sono usati da programmi attualmente in esecuzione. È meglio non usare uno di questi. Prima che l'IRQ 5 venisse usato per le schede audio, era spesso usato per una porta seriale.

Ecco come Greg (l'autore originale di Serial-HOWTO) ha impostato i suoi in /etc/rc.d/rc.serial. rc.serial è un file (uno script di shell) che viene lanciato in avvio (potrebbe avere un percorso diverso). Per versioni di setserial superiori a 2.15 non è più fatto in questo modo, ma questo esempio mostra la scelta di IRQ.

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

Assegnazioni di IRQ standard:

```
IRQ  0    Timer channel 0 (Potrebbe significare "no interrupt".
          Vedi sotto)
IRQ  1    Tastiera
IRQ  2    Cascade per il controller 2
IRQ  3    porta seriale 2
IRQ  4    porta seriale 1
IRQ  5    porta parallela 2, scheda audio
IRQ  6    Floppy
IRQ  7    porta parallela 1
IRQ  8    Real-time clock
IRQ  9    Rediretto a IRQ2
IRQ 10    non assegnato
IRQ 11    non assegnato
IRQ 12    non assegnato
IRQ 13    coprocessore matematico
IRQ 14    controller 1 di dischi fissi
IRQ 15    controller 2 di dischi fissi
```

Non esiste la Cosa Giusta da fare quando si scelgono gli interrupt. Semplicemente assicuratevi che esso non sia usato dalla scheda madre, o da qualsiasi altra scheda. 2, 3, 4, 5, 7, 10, 11, 12 o 15 sono scelte possibili. Notate che IRQ 2 è la stessa cosa di IRQ 9. Potete invocare sia 2 che 9, il driver seriale è molto comprensivo. Se avete una scheda seriale molto vecchia potrebbe essere incapace di usare gli IRQ 8 e superiori.

Accertatevi di non usare gli IRQ 1, 6, 8, 13 o 14!. Questi sono usati dalla vostra scheda madre. La farete molto scontenta se gli rubate i suoi IRQ. Quando avete finito, ricontrollate `/proc/interrupts` mentre i programmi che usano gli interrupt sono in esecuzione ed assicuratevi che non vi siano conflitti.

6.7 Scegliere gli indirizzi – Conflitti della scheda video con ttyS3

L'indirizzo IO delle schede video IBM 8514 (ed altre simili) è 0x?2e8 dove ? è 2, 4, 8 o 9. Questo può causare conflitto (ma non dovrebbe se la porta seriale è ben concepita) con l'indirizzo IO di `ttyS3` in 0x02e8 se la porta seriale ignora lo 0 esadecimale iniziale (molte lo fanno). Queste sono cattive notizie se tentate di usare `ttys3` a quell'indirizzo di IO.

Nella maggioranza dei casi dovreste usare l'indirizzo predefinito se possibile. Gli indirizzi mostrati rappresentano il primo indirizzo in un intervallo di 8 byte. Ad esempio 3f8 comprende in realtà 3f8-3ff. Ogni dispositivo seriale (così come altri tipi di dispositivi che usano indirizzi IO) abbisogna del proprio univoco intervallo di indirizzi. Non dovrebbero esserci confilitti. Ecco gli indirizzi predefiniti per le porte seriali:

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

6.8 Impostare gli indirizzi IO e IRQ nell'hardware (per lo più per PnP)

Dopo che è impostato nell'hardware non dimenticate di assicurarvi che sia anche impostato nel driver usando `setserial`. Per porte seriali non-PnP essi sono impostati sia nell'hardware da jumper o facendo girare un programma DOS (senza jumper) per impostarli (questo potrebbe disabilitare PnP). Il resto di questa sottosezione riguarda solo le porte seriali PnP. Ecco una lista dei possibili metodi per configurare una porta seriale:

- Usando un menù di impostazioni CMOS per un BIOS PnP (in genere solo per i modem esterni su ttyS0 (Com1) e ttyS1 (Com2))
- Lasciando che un BIOS PnP configuri automaticamente una porta seriale PnP Vedere 6.8.1 (Usare un BIOS PnP per configurare I0-IRQ)
- Non facendo nulla se avete sia una porta seriale PnP ed un sistema operativo Linux PnP (vedere Plug-and-Play-HOWTO).
- Usando `isapnp` per una porta seriale PnP non-PCI
- Usando pciutils (pcitools) per il bus PCI

Gli indirizzi di IO e IRQ devono essere impostati (da PnP) nei propri registri ogni volta che il sistema viene acceso visto che l'hardware PnP non tiene memoria di cosa era stato impostato prima che venisse spento il PC. Un semplice modo di fare questo è lasciare che il BIOS PnP sappia che voi non avete un sistema operativo PnP ed il BIOS automaticamente lo farà ogni volta che si fa partire. Questo potrebbe causare problemi sotto Windows (che è un sistema operativo PnP) se voi lanciate Windows mentre il BIOS pensa che Windows non sia un sistema operativo PnP. Vedere il Plug-and-Play HOWTO.

Il Plug-and-Play era concepito per automatizzare la configurazione io-irq, ma per Linux, allo stato attuale, ha reso la vita più complicata, I kernel standard per Linux non supportano il plug-and-play molto bene. Se usate una patch al kernel di Linux per convertirlo a sistema operativo plug-and-play, allora tutto quanto di cui sopra dovrebbe essere gestito dal sistema operativo automaticamente. Ma quando volete usare questo per automatizzare la configurazione di dispositivi diversi dalla porta seriale, potreste scoprire che dovete comunque configurare i driver manualmente visto che molti driver Linux non sono scritti per supportare un sistema operativo Linux PnP. Se usate `isapnptools` od il BIOS per configurare plug-and-play questi metteranno semplicemente i due valori nei registri della sezione della porta seriale della scheda del modem e probabilmente dovrete comunque impostare setserial. Nulla di tutto questo è facile o è molto ben documentato all'inizio del '99. Vedere il Plug-and-Play-HOWTO e la FAQ di isapnptools.

6.8.1 Usare un BIOS PnP per configurare IO e IRQ

Mentre la spiegazione su come usare un sistema operativo PnP o isapnp per configurare l'io-irq dovrebbe essere di corredo al relativo software, questo non è il caso se volete lasciare al BIOS PnP l'esecuzione di questa configurazione. Non tutti i BIOS PnP possono farlo. Il BIOS ha in genere un menù CMOS per impostare le prime due porte seriali. Questo menù potrebbe essere difficile da trovare e per un BIOS Award si trova sotto chipset feautures setup. C'è spesso ben poco tra cui scegliere. A meno di indicazioni diverse nei menù, queste prime due porte vengono impostate agli indirizzi IO e IRQ standard. Vedere 8.1 (Nomi e numeri dei dispositivi di porte seriali).

Che vi piaccia o no, quando accendete un PC, un BIOS PnP inizia ad eseguire la configurazione PnP (io-irq) dei dispositivi hardware. Potrebbe eseguire il lavoro parzialmente e lasciare il resto al sistema operativo PnP (che voi probabilmente non avete) o, se pensa che voi non abbiate il sistema operativo PnP, potrebbe configurare tutti i dispositivi PnP ma non configurare i device driver. Questo è quello che volete ma non è sempre facile scoprire cosa ha fatto esattamente il BIOS PnP.

Se dite al BIOS che non avete un sistema operativo PnP, allora il BIOS PnP dovrebbe configurare tutte le porte seriali PnP, non solo le prime due. Un modo indiretto per controllare quello che fa il BIOS (se avete Windows 9x sullo stesso PC) è forzare una configurazione sotto Windows. Vedere il Plug-and-Play-HOWTO e cercare forced. È più facile usare il menù CMOS BIOS che potrebbe ignorare quello che avete forzato sotto Windows. Potrebbe esserci un'opzione nel BIOS che può impostare o disabilitare questa capacità di ignorare.

Se aggiungete un nuovo dispositivo PnP, il BIOS dovrebbe cambiare la sua configurazione PnP per accoglierlo. Potrebbe anche cambiare gli io-irq di dispositivi esistenti, se necessario, per evitare qualsiasi conflitto. A questo scopo, tiene una lista dei dispositivi non PnP a patto che abbiate detto al BIOS che questo dispositivi non PnP sono configurati con io-irq. Un modo di dire questo al BIOS consiste nel lanciare un programma sotto DOS/Windows chiamato ICU.

Ma come scoprire cosa ha fatto il BIOS così che possiate impostare i device driver con queste informazioni? Il BIOS stesso può fornire alcune informazioni, sia nei suoi menù di setup o tramite messaggi sullo schermo quando accendete il computer. Vedere 6.5.2 (Cos'è impostato nell'hardware della mia porta seriale ?)

6.9 Passare gli indirizzi IRQ e IO a setserial

Una volta che avete impostato gli indirizzi IO e IRQ nell'hardware (o fatto in modo che questo venga fatto dal PnP) avete bisogno anche di assicurarvi che il comando setserial venga lanciato ogni volta che viene lanciato Linux. Vedere la sottosezione 9.1.4 (Configurazione in fase di avvio)

6.10 Altre configurazioni

6.10.1 Configurare il flusso di controllo hardware (RTS/CTS)

Vedere 4.9 (Controllo di flusso) per una spiegazione. Si dovrebbe sempre usare il controllo di flusso hardware (ad eccezione di modem obsoleti che non l'hanno). Il vostro programma di comunicazione o `getty` dovrebbe avere una opzione per impostarlo (e se siete fortunati potrebbe già essere stato abilitato per default). Occorre che sia impostato sia all'interno del modem (tramite la stringa di inizializzazione o per default) che nel device driver. Il vostro programma di comunicazione dovrebbe mettere a posto entrambi (se lo configurate correttamente).

Se nessuna delle manovre sopradescritte consente l'attivazione del controllo di flusso hardware, dovete provvedere voi. Per il modem assicuratevi che esso sia impostato tramite stringa di inizializzazione o per default. Se dovete dire al device driver di farlo è meglio agire alla partenza mettendo un un file che viene lanciato in fase di avvio. Vedere la sottosezione 9.1.4 (Configurazione in fase di avvio). Dovete anche aggiungere quanto segue a tale file per ogni porta seriale (l'esempio è ttyS2) per la quale volete abilitare il flusso di controllo hardware:

```
stty crtscts < /dev/ttyS2
```

Se volete vedere se il controllo di flusso è abilitato eseguite quanto segue: in minicom (o simile) digitate AT&V per vedere come è configurato il modem e cercate &K3 che vuol dire controllo di flusso hardware. Poi controllate se il device driver lo rileva digitando stty -a < /dev/ttyS2. Cercate crtscts (senza il segno meno che lo disabilita).

7 Configurazione del modem (esclusa la porta seriale)

7.1 Trovare il vostro modem

Prima di usare molto tempo per configurare il vostro modem, dovete assicurarvi che esso possa essere trovato e che i comandi AT o simili possano essere ad esso inviati. Quindi suggerisco che voi prima eseguiate una semplice configurazione usando il programma di comunicazione che userete sulla porta per vedere se funziona. In caso affermativo, il modem è stato trovato. Altrimenti vedere 16.1 (Il modem è fisicamente presente ma non può essere rilevato). Un winmodem potrebbe essere difficile da rilevare e non funzionerà sotto Linux

7.2 Comandi AT

Così come per la porta seriale nella quale risiede un modem, anche il modem stesso richiede di essere configurato. Il modem si configura inviandogli dei comandi AT (o simili) sulla stessa linea seriale usata per inviare dati. Essi sono brevi e criptici comandi ASCII, tutte le stringhe di comando sono prefissate dalle lettere AT. Ad esempio: ATZ&K3. Qui ci sono due comandi: Z e &K3. Sfortunatamente ci sono molte diverse variazioni nel gruppo di comandi AT, così che quello che funziona per un modem potrebbe non funzionare per un altro. Quindi non vi è garanzia che i comandi AT dati in questa sezione funzioneranno per tutti i modem. Altro punto è che per far sì che il modem reagisca alla stringa di comando AT, deve essere inviato un carattere di ritorno a capo alla fine della stringa.

Dette stringhe di comando vengono automaticamente inviate al modem dai programmi di comunicazione o sono inviate direttamente da voi. La maggior parte dei programmi di comunicazione forniscono una schermata nella quale potete digitare i comandi direttamente al vostro modem. Questo è comodo per impostare il modem com'è prima di spegnerlo senza dover ricordare ogni volta come lo si era impostato.

Se avete un manuale per il vostro modem potrete probabilmente dare una scorsa al gruppo di comandi AT ivi indicati. Altrimenti potrete cercate di trovarli su Internet. Potreste usare un motore di ricerca ed includere alcuni reali comandi nella stringa di ricerca per evitare di cercare siti che parlano semplicemente di questi comandi ma dei quali non offrono una lista. Potreste anche provare alcuni dei siti riportati nella sottosezione 18.5 (Siti Web)

7.3 Stringhe di inizializzazione: salvarle e richiamarle

Gli esempi dati in questa sottosezione sono tratti dal gruppo di comandi AT Hayes. Tutte le stringhe di comando devono essere prefissate dalle due lettere AT (ad esempio: AT&C1&D3). Quando un modem viene acceso, si autoconfigura automaticamente con una delle configurazioni che ha salvate nella sua memoria non volatile. Se la configurazione è soddisfacente, non c'è altro da fare.

Se non risulta soddisfacente, si potrebbe sia alterare la configurazione memorizzata o configurare il modem ogniqualvolta venga usato inviandogli un stringa di comandi nota come init string (stringa di inizializzazione). Generalmente un programma di comunicazione fa questo. Quello che invia dipende da come avete configurato il programma di comunicazione o quale script avete scritto per esso se usate Kermit. Potete in genere modificare la init string che il vostro programma di comunicazione usa e cambiarla come vi pare. Talvolta il programma di comunicazione vi consente di selezionare il modello del vostro modem, quindi userà una init string che pensa sia la più adatta per quel modem.

La configurazione che il modem usa quando viene acceso per la prima volta potrebbe essere rappresentata da una init string. Potreste pensare a questa come una stringa di default (chiamata profilo). Se il vostro programma di comunicazione invia al modem un'altra stringa (l'init string), allora questa stringa modificherà la configurazione predefinita. Ad esempio se la init string contiene solo due comandi, allora solo queste due voci verranno cambiate. Comunque, alcuni comandi richiamano un profilo memorizzato all'interno del modem così che un singolo comando di questo tipo nella init string può di conseguenza cambiare tutta quanta la configurazione.

I modem moderni hanno alcuni profili diversi memorizzati tra i quali scegliere che si trovano nella memoria non-volatile (rimangono lì anche quando spegnete il modem). Nel mio modem ci sono due profili impostati dalla ditta costruttrice (0 ed 1, nessuno dei quali può essere cambiato) e due profili definiti dall'utente (0 ed 1) che l'utente può impostare e memorizzare. Il vostro modem potrebbe averne di più. Quale di questi profili definiti dall'utente venga usato all'accensione depende da un altro valore memorizzato nel profilo. Se viene impartito il comando &Y0 allora verrà usato il profilo 0 alla prossima accensione. Se invece troviamo un 1 invece che uno 0 allora il profilo 1 sarà usato all'accensione.

Ci sono anche comandi per richiamare (riusare) ciascuno dei 4 profili memorizzati. Uno potrebbe mettere un comando di questo tipo nella init string. Naturalmente se verrà richiamato lo stesso profilo così come è stato automaticamente caricato all'accensione, non cambia nulla a meno che il profilo attivo sia stato modificato dopo l'accensione. Visto che potrebbe essere stato modificato, è una buona idea usare una specie di init string anche se non fa null'altro che richiamare un profilo memorizzato.

Per richiamare un profilo salvato (usate 1 invece che 0 per il profilo 1):
Z0 recupera il profilo utente 0 e reimposta il modem (riappende, ecc.)
&F0 recupera il profilo impostato dalla ditta costruttrice 0

Una volta inviati i comandi al modem per configurarlo nel modo che volete (incluso richiamare il profilo della casa costruttrice per poi modificarlo un poco), potreste volere salvare questo come profilo definito dall'utente:
&W0 salva la configurazione corrente nel profilo utente

La maggior parte della gente non si preoccupa di salvare una buona configurazione nei propri modem ma, invece, inviano al modem una stringa più lunga ogni volta che il modem viene usato. Un altro metodo consiste nel riprisitinare la configurazione di default della casa costruttrice all'inizio della stringa di inizializzazione, quindi modificarla leggermente aggiungendo qualche altro comando alla fine della init string. Agendo in questo modo, non c'è pericolo di modificare il profilo definito dall'utente che viene caricato all'accensione

Si può anche scegliere una stringa di inizializzazione fornita da qualcun altro che sostenga che sia buona per il vostro modem ecc. Alcuni programmi di comunicazione hanno una libreria di stringhe di inizializzazione dalla quale scegliere. Il metodo più difficile (e quello che vi insegnerà di più riguardo ai modem) è studiare il manuale del modem e scrivere una stringa da soli. Potrete poi salvare questa configurazione all'interno del modem così che una stringa di inizializzazione non sarà necessaria. Una terza alternativa è iniziare con una stringa di inizializzazione che qualcun altro ha scritto, poi modificarla per adattarla alle vostre esigenze.

Ora se guardate le init string usate dai programmi di comunicazione potreste vedere simboli che non sono comandi modem validi. Questi simboli sono comandi al programma di comunicazione stesso (tipo ˜ che significa effettuare una breve pausa) e non sono inviati al modem

7.4 Altri comandi modem

Una prossima edizione di questo HOWTO potrebbe contenere anche qualcosa di più su questo ma il resto di questa sezione è per la maggior parte quello che si trova nel vecchio Serial-HOWTO. Tutte le stringhe devono iniziare con AT. Ecco alcuni codici AT Heyes che dovrebbero essere nella stringa (se non sono stati impostati usando le impostazioni predefinite del costruttore o da una configurazione salvata)

```
E1      eco comandi ON
Q0      riporta i codici di risposta
V1      verbose ON
S0=0    non rispondere mai (uugetty fa questo con l'opzione WAITFOR)
```

Ecco alcuni altri codici riguardanti il controllo delle linee DCD e DSR del modem:

```
&C1 DCD attivato solo dopo la connessione
&S0 DSR sempre attivato
```

Questi riguardano quello che fa il vostro modem quando inizia o finisce una comunicazione. Si potrebbe impostare anche quallo che fa DTR ma è più complicato.

Se il vostro modem non supporta un profilo salvato, lo potete impostare attraverso una stringa `INIT` in un file di configurazione (o simile). Alcuni vecchi modem hanno degli interruttori DIP che variano le impostazioni dei registri. Assicuratevi che siano impostati anch'essi correttamente.

Greg Hankins ha una libreria di impostazioni modem per diversi tipi. Se volete inviargli la vostra configurazione di lavoro fatelo a: `mailto:greg@cc.gatech.edu` . Potete recuperare queste impostazioni a *ftp://ftp.cc.gatech.edu/pub/people/gregh/modem-configs* .

Note: perché il suo USR Courier V.34 si reinizializzi correttamente dopo che cade DTR, Greg Hankins ha dovuto impostare `&D2` and `S13=1` (questo imposta il bit 0 del registro S13). È confermato che la cosa vale anche per gli USR Sportster V.34.

Nota: alcuni Supra trattano DCD in modo diverso rispetto agli altri modem. Se state usando un Supra, provate ad impostare `&C0` e *not* `&C1`. Dovrete anche impostare `&D2` per gestire DTR correttamente

8 Dispositivi di porta seriale /dev/ttyS2, ecc.

Per creare dispositivi nella directory dei dispositivi vedere il Serial-HOWTO: Creating Devices In the /dev directory.

8.1 Nomi e numeri dei dispositivi di porta seriali

I dispositivi in Linux hanno numeri primari e secondari. Ogni porta seriale può avere due possibili nomi, nella directory /dev: ttyS e cua. I loro driver si comportano in modo leggermente differente. Il dispositivo cua è disapprovato e potrebbe non essere più usato in futuro. Vedere 8.3 (Il device cua).

Dos/Windows usano il nome COM mentre il programma `setserial` usa tty00, tty01 ecc. Non confondete questi con dev/tty0, /dev/tty1, ecc. che sono usati per le console (il monitor del vostro PC) ma non sono porte seriali. La tavola seguente è per il caso standard (ma il vostro potrebbe essere diverso

```
                                                indirizzo
dos             prim. sec.          prim. sec.     IO
COM1  /dev/ttyS0  4,  64;  /dev/cua0  5,  64      3F8
COM2  /dev/ttyS1  4,  65;  /dev/cua1  5,  65      2F8
COM3  /dev/ttyS2  4,  66;  /dev/cua2  5,  66      3E8
COM4  /dev/ttyS3  4,  67;  /dev/cua3  5,  67      2E8
```

Notate che tutte le distribuzioni dovrebbero avere dei dispositivi ttyS (e molte distribuzioni hanno anche l'obsoleto cua). Potreste verificarlo digitando (non preoccupatevi se non trovate alcun obsoleto dispositivo cua):

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

8.2 Collegare con link ttySN a /dev/modem ?

In alcune installazioni, saranno creati due dispositivi extra, `/dev/modem` per il vostro modem e `/dev/mouse` per il vostro mouse. Entrambi sono dei link simbolici agli appropriati dispositivi in `/dev` che avete specificato durante l'installazione (a meno che non abbiate un bus mouse, allora `/dev/mouse` punterà al dispositivo del bus mouse).

Ci sono state alcune discussioni riguardo a `/dev/mouse` e `/dev/modem`. L'uso di questi link è sconsigliato. In particolare, se state pensando di usare il vostro modem per ricevere chiamate potreste avere problemi perché i file di lock potrebbero non funzionare correttamente se usate `/dev/modem`. Comunque, se cambiate o eliminate questi link, alcune applicazioni potrebbero necessitare di una riconfigurazione.

8.3 Il dispositivo cua

Ad ogni dispositivo ttyS corrisponde un dispositivo cua. Ma il dispositivo cua è disapprovato, così è meglio usare ttyS (a meno che cua sia richiesto). C'è differenza tra cua e ttyS ma un programmatore previdente può fare sì che una porta ttyS si comporti esattamente come una porta cua, così non c'è più realmente bisogno di cua. A meno che alcuni vecchi programmi non richiedano l'uso di cua.

Qual'è la differenza? La differenza principale tra cua e ttyS si riferisce a quanto succede in un programma C quando un normale comando open cerca di aprire la porta. Se una porta cua è stata impostata per controllare i segnali di controllo del modem, la porta potrebbe essere aperta anche se il segnale di controllo DCD del modem dice che non è vero. Una astuta programmazione (aggiungendo lineee addizionali al programma) può forzare una porta ttyS a comportarsi anch'essa in questo modo. Ma una porta cua può essere ancora più facilmente programmata per aprirsi per comporre una chiamata in uscita anche quando il modem non riesce ad identificare DCD (visto che nessuno ci ha chiamato e non c'è portante). Ecco perché cua era una volta usata per chiamate in uscita e ttyS era usata per chiamate in entrata.

A partire dal kernel 2.2, un messaggio di avvertimento verrà immesso nel log del kernel quando si usa cua. Questo è il presagio che cua fra un poco sparirà.

9 Degli interessanti programmi di cui dovreste essere a conoscenza

9.1 Cos'è setserial ?

Questa parte si trova in 3 HOWTO: Modem, Serial e Text-Terminal. Ci sono lievi differenze, in relatizione a quale HOWTO appaiono.

9.1.1 Introduzione

Non usate mai `setserial` con i Laptop (PCMCIA). `setserial` è un programma che vi permette di comunicare al software del device driver l'indirizzio IO della porta seriale, quale interrupt (IRQ) è impostato nell'hardware della porta quale tipo di UART avete, ecc. Può anche mostrare come il driver sia attualmente impostato. In più può rilevare l'hardware per cercare di determinare il tipo di UART e l'IRQ. Ma ci sono seri limiti. Vedere 9.1.2 (Probing). Notate che non si possono impostare l'IRQ, ecc. nell'hardware delle porte seriali PnP.

Se avete solo una o due porte seriali, generalmente esse si imposteranno correttamente senza usare setserial. Altrimenti (o se ci sono problemi con la porta seriale) dovrete probabilmente avere a che fare con setserial. A parte la pagina di manuale di `setserial`, leggete anche le informazioni in `/usr/doc/setserial.../` o `/usr/share/doc/setserial`. Dovrebbero dirvi come setserial sia gestito nella vostra distribuzione di Linux.

`Setserial` è spesso lanciato automaticamente in fase di avvio da uno script di shell allo scopo di assegnare IRQ, ecc. al driver. Setserial funziona solo se il modulo seriale è caricato (o se l'equivalente era compilato nel vostro kernel). Se doveste (per qualche ragione) scaricare il modulo seriale più tardi, i cambiamenti precedentemente effettuati da `setserial` saranno persi dal kernel (ma non da /etc/serial.conf). Quindi `setserial` deve essere lanciato di nuovo per reimpostarli. Oltre ad essere lanciato da uno script di avvio, qualcosa di simile a `setserial` viene lanciato quando il modulo seriale viene caricato. Quindi quando osservate i messaggi di avvio sullo schermo potrebbe sembrare che sia stato lanciato due volte ed in effetti si è verificato questo.

Setserial può impostare il tempo nel quale la porta continua ad operare dopo essere stata chiusa (per fare uscire qualsiasi carattere rimasto ancora nel suo buffer nella RAM principale). Questo è necessario a basse velocità di baud (1200 o inferiori). È anche necessario a velocità più alte se ci sono molte attese causate dal controllo di flusso. Vedere closing wait nella pagina di manuale.

Setserial non imposta né l'IRQ né gli indirizzi IO nella porta seriale da solo. Questo compito viene svolto da jumper o dal plug-and-play. Dovete passare a setserial gli identici valori che sono stati impostati nell'hardware. Non inventatevi dei valori che pensate possano andare bene da passare a setserial. Comunque, se sapete l'indirizzo I/O ma non conoscete l'IRQ potere ordinare a setserial di tentare di determinarlo.

Potete vedere una lista di possibili comandi da usare digitando `setserial` senza parametri. Non verranno mostrati le opzioni a una lettera tipo -v per verboso (?), che dovreste in genere usare quando dovete risolvere dei problemi. Notate che setserial chiama un indirizzo IO porta. Se digitate

```
setserial -g /dev/ttyS*
```

vedrete alcune informazioni circa il modo in cui il device driver è configurato per le vostre porte. Aggiungete una -v all'opzione -g per saperne di più sebbene poche persone avranno a che fare (o comprendere) queste informazioni aggiuntive visto che i valori predefiniti in genere funzionano bene. In casi normali l'hardware è impostato nello stesso modo in cui setserial lo riporta, ma se avete problemi allore esiste una buona probabilità che setserial si stia sbagliando. Infatti, potete lanciare setserial assegnando un indirizzo IO puramente fittizio, un qualsiasi IRQ e qualsiasi tipo di UART vi pare. Poi la prossima volta che digitate setserial ... verranno visualizzati questi falsi valori senza nessuna protesta. Naturalmente il driver della porta seriale non funzionerà correttamente (o non funzionerà del tutto) con questi valori inventati.

Mentre gli assegnamenti fatti da setserial sono persi quando il PC viene spento, un file di configurazione potrebbe reimpostarli (oppure una configurazione precedente) quando il PC viene acceso di nuovo. Nelle versioni più recenti, quello che voi cambiate tramite setserial viene salvato automaticamente in un file di configurazione. Nelle versioni più vecchie, il file di configurazione viene modificato solo se lo editate manualmente, quindi la configurazione rimane la stessa ad ogni avvio. Vedere 9.1.5 (Script/file di configurazione).

9.1.2 Rilevazione (probing)

Con le opzioni appropriate, `setserial` può rilevare (ad un dato indirizzo I/O) una porta seriale, ma dovete indovinare l'indirizzo di I/O. Se gli chiedete di rilevare /dev/ttyS2 per esempio, tenterà di rilevarlo solo all'indirizzo nel quale pensa sia ttyS2 (2F8). Se dite a setserial che ttyS2 si trova ad un indirizzo diverso, allora dovrete rilevare a quell'indirizzo, etc. Vedere 9.1.2 (Rilevazione).

Lo scopo di questo è vedere se lì c'è una uart, e se sì, quale IRQ abbia. Usate setserial come ultima risorsa visto che ci sono metodi più veloci per farlo tipo wvdialconf per rilevare i modem, guardare i messaggi che compaiono per primi all'avvio, o usando `pnpdump -dumpregs`. Per cercare di identificare l'hardware fisico usate i parametri -v (verbose) e il comando `autoconfig` per `setserial`. Se il messaggio che ne risulta mostra un tipo di uart come 16550A, allora tutto è a posto. Se invece mostra `unknown` (sconosciuto) per il tipo di uart, allora potrebbe non esserci una porta seriale a quell'indirizzo I/O. Alcune porte seriali a buon mercato non si identificano correttamente cosi se vedete `unknown` potreste comunque avere una porta seriale a quell'indirizzo.

A parte l'auto-rilevazione per il tipo di uart, setserial può anche auto-rilevare gli IRQ ma questo non sempre funziona correttamente. In versioni di setserial 2.15 e superiori, i risultati della vostra ultima prova di rilevazione possono essere salvati nel file di configurazione `/etc/serial.conf` che verrà usato al prossimo avvio di Linux. In fase di avvio, quando il modulo di serial si carica, una rilevazione per le UART viene fatta automaticamente ed i risultati visualizzati sullo schermo. Ma l'IRQ che mostra potrebbe essere errato. La seconda visualizzazione dello stesso è il risultato di uno script che in genere non esegue alcuna rilevazione e quindi non dà informazioni attendibili riguardo a come l'harware sia effettivamente impostato. Mostra sono dati di configurazione che qualcuno ha scritto all'interno dello script o dati che sono stati salvati in /etc/serial.conf.

Potrebbe essere che due porte seriali abbiano entrambe lo stesso indirizzo di IO impostato nell'hardware. Naturalmente ciò non è consentito, ma talvolta accade comunque. La rilevazione identifica una porta seriale quando in realtà ce ne sono due. Comunque, se hanno diversi IRQ, alla la rilevazione dell'IRQ potrebbe mostrare IRQ = 0. Per me accade questo solo se prima uso `setserial` per attribuire all'IRQ un valore fittizio.

9.1.3 Può Linux configurare i dispositivi seriali automagicamente?

Sì, ma ... La vostra distribuzione potrebbe già avere fatto questo in fase di avvio. Ma potreste volerli personalizzare. È facile da fare per setserial precedenti al 2.15. Semplicemente aggiungete alcune righe al file che lancia setserial all'avvio. Vedere 9.1.6 (Vecchi metodi di configurazione: modificare uno script) Ad esempio, per ttyS3 dovreste aggiungere:

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

al file che lancia setserial all'avvio. Fate questo per ogni porta seriale che volete autoconfigurare. Assicuratevi di passare un nome di dispositivo che effettivamente esiste sulla vostra macchina. In alcuni casi, questo non funzionerà bene a causa dell'hardware, così potreste volere assegnarli un irq e/o un tipo di uart ad esempio

9.1.4 Configurazione in fase di avvio

Quando il kernel carica il modulo seriale (o se l'equivalente modulo è incorporato al kernel) allora solo `ttyS{0-3}` sono auto-rilevate ed il driver viene impostato ad IRQ 4 e 3 (a prescindere dalle reali impostazioni dell'hardware). Potete vedere questo nei messaggi di avvio proprio come se `setserial` fosse stato lanciato. Se usate 3 o più porte, questo potrebbe generare un conflitto di IRQ.

Per risolvere detti conflitti passando a setserial i veri IRQ (o per altre ragioni) potrebbe esserci un file da qualche parte che lancia `setserial` di nuovo. Questo accade all'inizio della fase di avvio prima che qualsiasi processo usi la porta seriale. In effetti, la vostra distribuzione potrebbe avere impostato le cose in modo che il programma setserial venga lanciato automaticamente da uno script di avvio in fase di boot. Ulteriori informazioni su come gestire questa situazione dovrebbero trovarsi in file chiamati setserial.. o simile nella directory /usr/doc/ o /usr/share/doc/.

9.1.5 Script/file di configurazione

Il vostro obbiettivo è modificare (o creare) uno script nel ramo /etc che lanci setserial in fase di avvio. La maggior parte delle distribuzioni forniscono un file di questo tipo (ma inizialmente potrebbe non risiedere nel ramo /etc). In più, setserial 2.15 e superiore spesso hanno un file /etc/serial.conf che viene usato dallo script di cui sopra così che non dovrete modificare direttamente lo script che lancia setserial. Inoltre semplicemente usando setserial da riga comandi (2.15 e superiore) potrebbe alterare questo file di configurazione.

Quindi prima della versione 2.15 di setserial tutto quello che dovevate fare era modificare uno script. Dopo la 2.15 potreste avere bisogno di fare una delle seguenti cose: 1. modificare uno script. 2. modificare `/etc/serial.conf.` o 3. lanciare setserial da riga comandi il cui risultato sarà l'immediata modifica di `/etc/setserial.conf`. Quali di queste cose dobbiate fare dipende sia dalla vostra specifica distribuzione, e da come l'avete impostata.

Per di più neppure serial.conf viene mai modificato. Invece si usa semplicemente `setserial` da riga comandi.

9.1.6 Modificare uno script (dopo la versione 2.15: forse no)

Prima di setserial 2.15 (1999) non c'era un file /etc/serial.conf per configurare setserial. Quindi dovrete scoprire il file che lancia setserial in fase di avvio e modificarlo. Se non esiste dovrete crearne uno (o mettere i comandi in un file che viene lanciato nelle prime fasi di avvio). Se detto file viene correntemente usato, probabilmente si trova da qualche parte nel ramo /etc. Ma Redhat <6.0 lo ha fornito in /usr/doc/setserial ma dovrete spostarlo nel ramo /etc prima di usarlo. Potreste usare locate per cercare di trovare un file di questo tipo. Ad esempio potreste digitare: locate serial*.

Lo script `/etc/rc.d/rc.serial` era comunemente usato in passato.

Se un file di questo tipo viene fornito, dovrebbe contenere una serie di esempi commentati. Rendendo attivi e/o modificando alcuni di questi esempi, dovreste essere in grado di impostare tutto quanto correttamente. Assicuratevi di usare un percorso valido per `setserial` ed un valido nome di dispositivo. Potreste eseguire un test mandando in esecuzione questo file manualmente (basta digitare il suo nome da superuser) per vedere se funziona bene. Testare in questo modo è molto più veloce che eseguire ripetuti riavvii. Naturalmente potete anche testare un singolo comando di `setserial` semplicemente digitandolo sulla riga comandi.

Se volete che setserial determini automaticamente la uart e l'IRQ per ttyS3 dovreste aggiungere qualcosa tipo:

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

Fate ciò per ogni porta seriale che volete auto configurare. Assicuratevi di fornire un nome di device che esiste veramente sulla vostra macchina. In alcuni casi questo non funzionerà correttamente a causa dell'hardware così se sapete quali siano la uart e l'irq, potreste volerli assegnare esplicitamente con setserial. Ad esempio:

```
/sbin/setserial /dev/ttyS3 irq 5 uart 16550A  skip_test
```

Per versioni 2.15 e superiori (a patto che la vostra distribuzione implementi la modifica, Redhat non l'ha fatto) potrebbe essere più complicato da fare visto che il file che lancia setserial all'avvio, /etc/init.d/setserial o simile non era previsto che fosse modificato dall'utente. Vedere 9.1.7 (Nuovi metodi di configurazione usando /etc/serial.conf).

9.1.7 Nuovi metodi di configurazione usando /etc/serial.conf

Prima della versione 2.15 di setserial, il modo di configurare setserial era di modificare manualmente lo script di shell che lanciava setserial in fase di avvio, Vedere 9.1.6 (Modificare uno script (dopo la versione 2.15: forse no)). A partire dalla versione 2.15 (1999) di `setserial` quest script di shell non viene modificato ma piuttosto prende i suoi dati da un file di configurazione: `/etc/serial.conf`. In più non dovrete più avere bisogno di modificare serial.conf poiché usando il comando setserial da riga comandi potrebbe automaticamente fare sì che serial.conf sia modificato in modo appropriato.

Si intendeva agire in questo modo così che voi non avete bisogno di modificare alcun file per impostare (o cambiare) setserial, così che possa fare la cosa giusta ogni volta che Linux viene avviato. Ma ci sono delle trappole pericolose perché non è veramente setserial che modifica serial.conf. La confusione è moltiplicata poiché diverse distribuzioni gestiscono la cosa in modo differente. In più, voi potreste modficarlo così da farlo lavorare in modo diverso.

Quello che spesso accade è questo: Quando spegnete il vostro PC lo script che lancia setserial in fase di avvio viene di nuovo lanciato, ma questa volta esegue solo la parte che la situazione di stop dice di fare: Esso usa setserial per trovare qual è lo stato attuale di setserial e mette questo informazioni nel file `serial.conf`. Così quando lanciate setserial per cambiare il file serial.conf, il cambiamento non avviene immediatamente, ma solo quando eseguite un normale spegnimento del PC.

Ora potreste forse indovinare quale problema potrebbe sopraggiungere. Supponiamo che voi non spegniate normalmente (usando l'interruttore, ecc.) e che i cambiamenti non vengano salvati. Supponiamo che state sperimentando con setserial e dimentichiate di lanciarlo un'ultima volta per ripristinare lo stato originale (o fare in modo che gli errori vengano riportati al loro stato originale). In questo caso saranno salvate le vostre impostazioni sperimentali.

Se modificate manualmente serial.conf, allora la vostra modifica viene cancellata quando spegnete visto che viene ripristinato lo stato di setserial allo spegnimento. C'è un modo di disabilitare il cambiamento di setserial in fase di spegnimento ed è quello di rimuovere ###AUTOSAVE### o simile dalla prima riga di serial.conf. In almeno una distribuzione, la rimozione di ###AUTOSAVE### dalla prima riga viene fatto automaticamente dopo la prima volta che si spegne il PC appena dopo l'installazione. Il file serial.conf dovrebbe contenere degli esempi commentati per aiutarvi.

Il file più comunemente usato per lanciare setserial all'avvio (in conformità con il file di configurazione) è ora /etc/init.d/setserial (Debian) o /etc/init.d/serial (Redhta), o ecc., ma normalmente non dovrebbe essere modificato. Per la 2.15 Redhat 6.0 ha semplicemente un file /usr/doc/setserial-2.15/rc.serial che dovete spostare in /etc/init.d/ se volete che setserial venga lanciato in fase di avvio.

Per disabilitare una porta, usate `setserial` per impostarla uart none. Il formato di /etc/serial.conf sembra che sia quello dei parametri digitati dopo setserial da riga comandi con una riga per ogni porta. Se non usate autosave, potreste modificare /set/serial.conf manualmente.

BUG: Fino a luglio 1999 c'era un bug/problema visto che con ###AUTOSAVE### solo i parametri di setserial visualizzati da setserial -Gg /dev/ttyS* venivano salvati, ma gli altri parametri no. Usate il flag -a con setserial per vedere tutti i parametri. Questo bug affliggerà solo una piccola minoranza di utenti visto che i parametri non salvati sono in genere OK per la maggior parte delle situazioni. È stato riportato come bug e potrebbe essere già stato corretto. Per fare sì che le impostazioni correnti vengano salvate da setserial nel file di configurazione (serial.conf) senza spegnere il PC, fate quello che fareste normalmente quando spegnete: Lanciate lo script di shell `/etc/init.d/{set}serial stop`. Il comando stop salverà la configurazione corrente ma la porta seriale continuerà a funzionare bene.

In alcuni casi potreste avere sia il nuovo che il vecchio metodo di configurazione installati ma si spera che solo uno di essi venga lanciato in fase di avvio. Debian classificava come obsoleti i file con ...pre-2.15.

9.1.8 IRQ

Per default, sia ttyS0 che ttyS2 condividono l'IRQ 4, mentre ttys1 e ttyS3 condividono l'IRQ 3. Ma non è permessa la condivisione degli interrupt seriali, a meno di: 1. avere un kernel 2.2 o superiore, 2. si è compilato il supporto per questa cosa, e 3. il vostro hardware seriale lo supporta. Vedere 6.6.2 (Condivisione di interrupt e i Kernel 2.2+) Se avete solo due porte seriali ttyS0 e ttyS1, siete comunque a posto, visto che non esistono conflitti di condivisione di IRQ per dispositivi che non esistono. Se aggiungete un modem interno e conservate ttyS0 e ttyS1, allora dovreste cercare di trovare un IRQ inutilizzato per assegnarlo al vostro device driver. Se IRQ 5 non è usato da una scheda audio, potrebbe essere adatto da usare per un modem.

Per impostare l'IRQ nell'hardware potreste avere bisogno di usare isapnp, un BIOS PnP o modificare Linux per renderlo PnP. Per aiutarvi a determinare quale IRQ di ricambio potreste avere, digitate man setserial e cercate diciamo IRQ 11.

9.2 Cos'è isapnp ?

`isapnp` è un programma per configurare i dispositivi Plug-and-Play (PnP) sul bus ISA inclusi i modem interni. È incluso in un pacchetto chiamato isapnptools ed include un altro programma, pnpdump che trova tutti i vostri dispositivi ISA PnP e vi mostra le opzioni per configurarli in un formato che potrebbe essere aggiunto al file di configurazione di PnP: /etc/isapnp.conf. Potrebbe anche essere usato con l'opzione –dumpregs per mostrare l'indirizzo IO e l'IRQ della porta seriale del modem correnti. Il comando isapnp potrebbe essere incluso in un file di avvio così che esso sia lanciato ogni volta che si accende il computer e quindi configuri i dispositivi ISA PnP. Si può fare questa anche se il vostro BIOS non supporta il PnP. Vedere il Plug-and-Play-HOWTO.

9.3 Cos'è wvdialconf?

`wvdialconf` cercherà di trovare quale porta seriale (ttyS?) ha un modem su di essa. Crea anche un programma di configurazione per il programma wvdial. `wvdial` È usato per semplificare le chiamate in uscita verso un ISP (Internet provider) usando il protocollo PPP. Ma non avete bisogno di installare PPP per potere usare `wvdialconf`. Esso rileverà solo modem che non sono in uso. Concepirà anche automaticamente una stringa di inizializzazione adatta ma talvolta la creerà sbagliata. Visto che il comando non ha opzioni, è semplice da usare ma dovete passargli il nome di un file nel quale mettere la stringa di inizializzazione (ed altri dati). Ad esempio digitate: wvdialconf nome_del_mio_file_di_configurazione.

9.4 Cos'è stty?

`stty` è come setserial ma imposta il baud rate ed altri parametri della porta seriale. Digitando stty -a < /dev/ttyS2 dovreste visualizzare come è configurata ttyS2. La maggior parte delle impostazioni sono per cose che non dovreste mai avere bisogno di usare coi modem (tipo cose usate solo per terminali degli anni '70). Il vostro pacchetto di comunicazione dovrebbe automaticamente impostare tutta la corretta configurazione per i modem. Ma stty è talvolta utile per risolvere dei problemi.

Due valori impostati da stty sono: 1. Flusso di controllo hardware tramite crscts e 2. Ignorare il segnale DCD da modem: clocal. Se il modem non sta inviando segnali DCD e clocal è disabilitato (stty mostra -clocal), allora un programma potrebbe non essere in grado di aprire la porta seriale. Se la porta non si può aprire il programma potrebbe bloccarsi, attendendo (spesso vanamente) per un segnale DCD dal modem.

Minicom imposta clocal automaticamente quando viene lanciato quindi non ci sono problemi. Ma la versione 6.0.192 di Kermit si pianta quando imposto -clocal e provo set line.... Se -clocal è impostato e non c'è segnale DCD allora anche il comando stty si pianterà e non c'è modo di impostare clocal (a meno di lanciare minicom). Ma minicom reimposterà -clocal laddove esista. Un modo di uscire da questo è usare minicom per inviare AT&C al modem (per ottenere il segnale DCD), quindi uscire da minicom senza resettare così che il segnale DCD rimanga attivo. Poi si può riusare stty.X

10 Provare il modem (effettuare una chiamata)

10.1 Siete pronti per effettuare una chiamata?

Una volta che avete collegato il vostro modem e sapete quale porta seriale è attiva, siete pronti ad usarlo. Prima di provare ad accedere ad Internet o consentire che altri si colleghino al vostro computer, provate prima qualcosa di più semplice come chiamare alcuni numeri per vedere se il vostro modem funziona bene. Se non sapete che numero chiamare, chiedete nei negozi di computer dei numeri di banche dati, ecc. oppure vedete se una libreria locale ha un numero di telefono per il suo catalogo in linea.

Poi assicuratevi di essere pronti a telefonare. Sapete su quale porta seriale (tipo ttyS2) si trova il vostro modem? Avreste dovuto scoprirlo quando avete configurato l'io-irq della vostra porta seriale. Avete deciso quale velocità userete per questa porta? Vedere 13.4 (Tabella delle velocità) per una scelta rapida oppure 13 (Che velocità dovrei usare con il mio modem?) per ulteriori dettagli. Se non avete idea di quale velocità impostare, impostatela un po' superiore a quella certificata del vostro modem. Ricordate anche che se vedete un menù dove un opzione è hardware flow control (controllo di flusso hardware) e/o RTS/CTS o simili, selezionatela. È un cavo telefonico attivo quello collegato al vostro modem? Potreste connettere il cavo ad un vero telefono per controllare che produca il segnale di linea.

Ora dovete selezionare un programma di comunicazione da usare per chiamare il numero. Fra questi programmi includiamo minicom, seyon (X-windows) e kermit. Vedere la sezione 14 (Programmi di comunicazione) per notizie circa alcuni programmi di comunicazione. Un paio di esempi sono presentati più avanti: 10.2 (Chiamare con Minicom) e 10.3 (Chiamare con Kermit).

10.2 Chiamare con minicom

Minicom viene incluso nella maggior parte delle distribuzioni di Linux. Per configurarlo dovete esser collegati come root. Digitate minicom -s per configurarlo. Verranno direttamente visualizzati i menù di configurazione. Oppure potete anche digitare minicom, poi digitare ˆA per vedere la riga di stato in fondo. Essa invita a digitare ˆA Z per un aiuto (se avete già digitato ˆA digitate solo z). Dal menù di aiuto (help menù) andate al menù di Configurazione (Configuration menu).

Non occorre impostare la maggior parte delle opzioni se si vuole semplicemente effettuare una chiamata. Per la configurazione dovete fornire due semplici voci: il nome della porta seriale connessa al modem (tipo /dev/ttyS2) e la velocità (tipo 115200). Questi valori si impostano nel menù serial port. Apritelo ed impostate i valori. Poi (se possibile) impostate il controllo di flusso hardware (RTS/CTS). Poi salvate. Quando digitate la velocità, dovreste anche vedere qualcosa tipo 8N1 che dovreste lasciare stare. Vuole dire: Byte da 8 bit, Nessuna parità, 1 bit di stop aggiunto ad ogni byte. Se non trovate la velocità che desiderate, una velocità minore funzionerà comunque per una prova. Uscite (digitando return) quando avete finito e salvate la configurazione come default (dfl) usando il menù. Potreste uscire da minicom e poi rilanciarlo per controllare se adesso trova la porta seriale ed inizializza il modem, oppure potete tornare all'help e dire a minicom di inizializzare il modem.

Ora siete pronti per chiamare. Ma prima, dalla videata principale che ottenete dopo avere digitato minicom, assicuratevi che ci sia un modem presente digitando AT poi premendo il tasto <enter>. Dovreste vedere un OK. Se questo non accade c'è qualcosa di sbagliato e sarà impossibile tentare una chiamata.

Se ricevete OK tornate all'help e selezionate l'elenco telefonico. Potete modificarlo e digitare un numero di telefono, ecc. nell'elenco, poi scegliete dial (componi) per chiamare. In alternativa, potete digitare il numero manualmente (selezionando manual poi digitando il numero sulla tastiera). Se non funziona annotate accuratamente ogni messaggio di errore visualizzato e cercate di scoprire cosa non va. Per vedere se minicom ha trovato il modem, semplicemente lanciatelo e digitate direttamente qualcosa. Tutti i comandi al modem devono essere prefissati da AT. Scrivendo quindi semplicemente AT dovreste ricevere in risposta un OK dal modem.

10.3 Effettuare una chiamata con kermit

Trovate l'ultima versione di `kermit` in *http://www.columbia.edu/kermit/* . Ad esempio, diciamo che il vostro modem si trova in `ttyS3` e la velocità è 115200 bps. Dovreste fare questo:

```
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 poi Enter
OK                          ; il modem dovrebbe rispondere così
```

Se il modem risponde correttamente ai comandi `AT` allora si deve supporre che il vostro modem funziona correttamente per quanto riguarda Linux. Ora provate a chiamare un altro modem digitando:

```
ATDT7654321
```

dove 7654321 è un numero di telefono. Usate ATDP in luogo di ATDT se avete una linea ad impulsi. Se la chiamata esce, il modem sta funzionando.

Per tornare al prompt di `kermit`, tenete premuto il tasto Ctrl, premete la barra rovesciata, quindi rilasciate il tasto Ctrl, quindi premete il tasto C:

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

Questo è stata una semplice prova usando un primitivo metodo di chiamata a mano. Il metodo abituale è lasciare che `kermit` componga il numero per voi con il suo database di modem interno e le sue capacità di composizione automatica, ad esempio usando 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        ; Seleziona il tipo di modem
C-Kermit>set line /dev/ttyS3       ; Seleziona il dispositivo di comunicazione
```

```
C-Kermit>set speed 115200          ; Imposta la velocit... di connessione
C-Kermit>dial 7654321              ; Compone
 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:
```

11 Dial-in - ricevere chiamate

11.1 Introduzione

Il dial-in è quando impostate il vostro PC così che altri possano chiamare il vostro numero di telefono ed usare il vostro PC. Il punto di vista è il vostro PC. Quando componete un numero esterno dal vostro PC state nel contempo facendo un dial-in in un altro computer (ma non potete fare il dial-in nel vostro proprio computer).

Il dial-in funziona così. Qualcuno con un modem compone il vostro numero telefonico. Il vostro modem risponde alla chiamata e si connette. Una volta che il chiamante è connesso, il vostro PC (tramite il programma `getty`) invia il processo di login per il chiamante. Il metodo originale era quello di inviare un prompt di login allo schermo del chiamante (login manuale). Ma un metodo più moderno (se usate `mgetty`) è lanciare PPP (pppd) e lasciar fare al PPP il login al chiamante in modo automatico (non c'è bisogno di digitare manualmente un nome o una password). Vedere il PPP-HOWTO (una nuova revisione è attesa tra poco) e la documentazione di `mgetty` per maggiori dettagli.

Dopo che il chiamante si è collegato, egli usa il vostro PC. Usare il vostro PC significa che il chiamante ha un account di shell e può usare il vostro PC proprio come se si fosse collegato tramite console (o terminale di testo). Potrebbe anche significare che egli si può connettere ad Internet attraverso il vostro PC (tramite PPP). Il programma che usate nel vostro PC per gestire il dial-in è chiamato `getty` o `mgetty`. Vedere 11.2.1 (A proposito di mgetty)

Se vi aspettate che qualcuno sia in grado di collegarsi a 56k, non è possibile a meno che:

1. Abbiate una connessione digitale alla compagnia telefonica tipo una linea trunkside-T1 o ISDN.
2. Usiate modem digitali speciali (vedere 3.2 (Modem digitali))
3. Abbiate un ...concentrator (concentratore) o simile per interfacciare i vostri modem digitali alle linee digitali della compagnia telefonica.

Un ... concentrator potrebbe essere chiamato anche modem concentrator oppure remote access concentrator (concentratore di accesso remoto) o potrebbe essere incluso in un remote access server (server di accesso remoto) che includa i modem digitali, ecc. Questo tipo di impostazioni vengono usate dagli Internet Service Provider (ISP).

11.2 Getty

`getty` è il programma che lanciate per il dial-in. Non ne avete bisogno per chiamate verso l'esterno (dialout).

Oltre a presentare un prompt di login, risponde anche al telefono. Originariamente getty veniva usato per il login verso un computer da un terminale stupido. È attualmente usato per il login verso una console Linux. Ci sono alcuni programmi getty differenti con nomi leggermente diversi. Solo alcuni funzionano con i modem per il dialin. Questo programma `getty` viene in genere lanciato in fase di boot. Deve essere chiamato dal file /etc/inittab. Potreste trovare un esempio in questo file di una chiamata a getty nel quale potreste avere bisogno di fare qualche piccola modifica. Se usate un programma getty diverso da quello mostato nell'esempio, allora dovreste apportare delle modifiche importanti visto che le opzioni hanno diversi formati.

Ci sono quattro diversi programmi getty tra cui scegliere che possono essere usati coi modem per il dialin: `mgetty`, `uugetty`, `getty_em` e `agetty`. Alcuni dettagli vengono forniti nelle sottosezioni seguenti. `getty` è il più semplice (e più debole) dei quattro ed alcuni lo considerano principalmente da usarsi con terminali di testo collegati direttamente. `mgetty` supporta fax e voice mail, invece `uugetty` no. `mgetty` si afferma manchi di alcune delle caratteristiche di `ugetty`. `getty_em` è una versione semplificata di `ugetty`. Quindi `mgetty` è probabilmente la vostra scelta migliore a meno che abbiate già familiarità con `ugetty` (o troviate difficile reperire `mgetty`). La sintassi per questi programmi getty differisce, così assicuratevi di stare usando la sintassi corretta in `/etc/inittab` per qualsiasi getty usiate.

11.2.1 Mgetty

`mgetty` fu scritto come rimpiazzo di `uugetty` che esisteva molto prima di `mgetty`. Entrambi si possono usare con i modem. Sebbene `mgetty` possa anche essere usato per terminali direttamente connessi, la documentazione per questa cosa è difficile da trovare e mgetty (a metà 1999) non supporta il controllo di flusso software (usato da molti terminali) senza ricompilarlo. Questo difetto viene classificato come un bug. Oltre a consentire login in dialup, `mgetty` fornisce supporto FAX ed autoriconoscimento del PPP. C'è un programma addizionale chiamato `vgetty` che gestisce le caselle vocali per alcuni modem. La documentazione per `mgetty` è buona (eccetto per le caselle vocali), e non necessita di supplementi. Per cortesia riferitevi ad essa per le istruzioni di installazione. Potete trovare le ultime informazioni su `mgetty` a *http://www.leo.org/˜doering/mgetty/* e `http://alpha.greenie.net/mgetty`

11.2.2 uugetty

`getty_ps` contiene due programmi: `getty` viene usato per le console ed i terminali, e `ugetty` per i modem. Greg Hankins (già autore del Serial-HOWTO) usava `ugetty` così le sue considerazioni circa questo sono là incluse. Vedere 12 (Uugetty). Gli altri getty sono ben trattati dalla documentazione che li accompagna.

11.2.3 getty_em

Questa è una versione semplificata di uugetty. Fu scritta da Vern Hoxie dopo che fu completamente confuso dai complessi file di supporto che occorrono a getty_ps e uugetty.

Fa parte della raccolta di utilità ed informazioni di Vern Hoxie reperibili tramite ftp da `scicom.alphacdc.com/pub/linux` . Il nome della raccolta è serial_suite.tgz. Quando eseguite il login in scicom come anonimi, dovete usare il vostro indirizzo e-mail completo come password. Ad esempio: greg.hankins@cc.gatech.edu

11.2.4 agetty e mingetty

`agetty` è una semplice e completamente funzionale implementazione di `getty` che meglio si adatta alle console virtuali od ai terminali piuttosto che per i modem. Ma, date determinate condizioni favorevoli, funziona bene anche con i modem (a meno che voi non eseguiate una chiamata quando agetty è in esecuzione in fase di attesa di chiamate). `agetty` nella distribuzione Debian viene semplicemente chiamata `getty`.

`mingetty` è un piccolo getty che funzionerà solo per le console (monitor), quindi non si può usare con i modem per il dialin.

11.3 Cosa succede quando qualcuno ci chiama? (dial-in)

Il chiamante lancia un certo programma di configurazione che compone il vostro numero di telefono ed il vostro telefono squilla. Ci sono due differenti modi nei quali il vostro PC può rispondere al telefono. Un modo è che il modem risponda automaticamente alla chiamata. L'altro modo è che getty rilevi lo squillo ed invii un comando al modem che gli imponga di rispondere alla chiamata. Una volta che viene risposto alla chiamata, il vostro modem invia dei toni all'altro modem (e viceversa). I due modem negoziano il modo con il quale essi comunicheranno e quando finiscono il vostro modem invia un messaggio di CONNECT (connessione) o simile a `getty`. Quando `getty` riceve questo messaggio invia un prompt di login attraverso la porta seriale. Qualche volta `getty` invoca semplicemente un programma chiamato `login` per gestire il login. `getty` in genere viene lanciato in fase di boot ma deve aspettare fino a quando viene effettuata una connessione prima di fare partire il prompt di login.

Ora ulteriori dettagli sui due modi di rispondere ad una chiamata. Impostando il registro S0 del modem a 3, il modem automaticamente risponde al terzo squillo. Se è impostato a 0 allora il modem risponderà alla chiamata solo se getty gli invia il comando A (Risposta) mentre il telefono sta squillando. In realtà il comando è ATA visto che tutti i comandi modem devono essere prefissati da AT. Potreste pensare che sia meglio utilizzare la capacità del modem di rispondere automaticamente alla chiamata, ma in verità è meglio fare sì che sia `getty` a rispondere. Se il modem non risponde automaticamente, si parla di risposta manuale (anche se `getty` la gestisce in modo automatico).

Per il caso di risposta manuale, `getty` apre la porta in fase di boot e resta in ascolto. Quando il telefono squilla, un messaggio RING viene inviato a `getty` che sta ascoltando. Poi se `getty` vuole rispondere allo squillo, invia al modem il comando ATA. Il modem poi esegue la connessione ed invia il messaggio CONNECT ... a `getty` che invia un prompt di login al chiamante.

Nel caso della risposta automatica, viene usata la linea CD Carrier detect (Portante rilevata) che va dal modem alla porta seriale per rilevare quando viene fatta una connessione. Funziona così. In fase di boot, `getty` cerca di aprire la porta seriale, ma fallisce visto che in genere non c'è segnale CD dal modem. Allora il programma `getty` si ferma all'istruzione di apertura (open) nel programma fino a che il segnale CD compare. Quando un segnale CD arriva (forse ore dopo), allora la porta viene aperta e `getty` invia il prompt di login. Mentre `getty` sta aspettando all'altezza dell'istruzione di apertura, altri processi possono essere lanciati visto che Linux è un sistema operativo multiprocessing. Quello che fa svegliare `getty` è un interrupt che viene generato quando la linea CD dal modem pone il suo stato ad on.

Potreste chiedervi come getty sia capace di aprire la porta seriale nel caso della risposta manuale visto che non vi è segnale CD. Bene, si può scrivere un programma per forzare l'apertura della porta anche se non è presente un segnale CD.

11.4 Perché la risposta manuale è meglio

La differenza tra i due modi di risposta si palesa quando il computer è spento ma il modem sta ancora lavorando. Nel caso manuale, il messaggio RING viene inviato a getty ma visto che il computer è spento, getty non è pronta, quindi non ci sarà mai risposta al telefono. Non ci sono addebiti telefonici quando non si risponde alla chiamata. Nel caso della risposta automatica, c'è risposta alla telefonata ma non verrà mai inviato un prompt di login visto che il computer è spento. La bolletta cresce mentre l'attesa continua. Se la chiamata è gratuita, non fa molta differenza, sebbene possa essere frustrante attendere per un prompt di login che mai arriverà. `mgetty` usa la risposta manuale. `uugetty` può fare questo tramite uno script di configurazione.

11.5 Callback

Si definisce callback quando qualcuno per primo chiama il vostro modem. Poi voi ottenete un po' di informazioni dal chiamante e lo richiamate. Perché si vuole fare questo? Una ragione è risparmiare sulla bolletta se voi potete telefonare al chiamante con tariffe più convenienti rispetto alle sue. Un altro è assicurarvi che il chiamante sia davvero colui che sostiene di essere. Se un chiamante vi contatta e dice di chiamare dal suo abituale numero telefonico, un modo per verificarlo è di effettuare una nuova chiamata a quel numero.

C'è un programma per Linux chiamato callback che funziona con mgetty. Si trova in `ftp://ftp.icce.rug.nl/pub/unix/` . Istruzioni passo-passo su come installarlo (e PPP) si trovano a `http://www.stokely.com/unix.serial.port.resources/callback.html`

11.6 Casella vocale (Voice Mail)

La casella vocale è come una segreteria telefonica eseguita da un computer. Per fare questo dovete avere un modem con il supporto voice ed il relativo software. Invece di incidere i messaggi su nastro, essi vengono salvati in forma digitale sul disco. Quando qualcuno vi chiama, ascolterà un messaggio di benvenuto e poi può lasciare un messaggio per voi. Sistemi più avanzati possono avere caselle postali selezionabili dal chiamante e messaggi da fare ascoltare selezionabili dall'utente. Software gratuito è disponibile in Linux per la semplice risposta, ma non sembra essere ancora a disposizione per capacità più avanzate.

So di due diversi pacchetti di voicemail per Linux. Uno è minimale (vedere 14.2.4 (Voicemail Software)). L'altro, più avanzato, ma attualmente scarsamente documentato, è `vgetty`. È una aggiunta opzionale al ben documentato e largamente distribuito programma `mgetty`. Supporta i comandi modem vocali tipo ZyXEL. Nella distribuzione Debian potete ottenere il pacchetto mgetty-voice in aggiunta al pacchetto mgetty e mgetty-doc. La documentazione obsoleta è stata rimossa da mgetty, ma quella messa al suo posto è latitante (a meno che voi usiate l'opzione -h (aiuto) quando lanciate certi programmi, ecc). Ma si potrebbero consultare i messaggi circa l'uso che vengono inviati nel newsgroup di mgetty. Vedere 11.2.1 (A proposito di mgetty) e >. Sembra che `vgetty` sia attualmente non molto stabile ma che venga usato con successo e che il suo sviluppo continui. Se questa è l'ultima versione di questo HOWTO qualcuno che è familiare con vgetty mi faccia cortesemente sapere lo stato attuale delle cose.

12 Uugetty per dial-in (dal vecchio Serial-HOWTO)

Sappiate che potreste usare mgetty come (migliore?) alternativa a `uugetty`. `mgetty` è più nuovo e più famoso di uugetty. Vedere 11.2 (Cos'è getty?) per un breve confronto tra questi 2 getty.

12.1 Installare getty_ps

Visto che uugetty è parte di getty_ps dovete prima installare getty_ps. Se non lo avete procuratevi l'ultima versione da *`metalab.unc.edu:/pub/Linux/system/serial`* . In particolare, se volete usare alte velocità (57600 and 115200 bps), dovete procurarvi la versione 2.0.7j o superiore. Dovrete anche avere libc 5.x o superiore.

Per default, `getty_ps` sarà configurato come Linux FSSTND (File System Standard) compatibile, il che significa che si troverà in `/sbin`, ed i file di configurazione saranno chiamati `/etc/conf.{uu}getty.ttyS`*N*. Questo non è ben chiaro dalla documentazione! Si aspetta che i file di lock vadano in `/var/lock`. Assicuratevi di avere una directory `/var/lock`.

Se non volete la compatibilità FSSTND, i file binari andranno in `/etc`, quelli di configurazione in `/etc/default/{uu}getty.ttyS`*N* ed i file di lock in `/usr/spool/uucp`. Vi raccomando di agire in questo modo se state usando UUCP, visto che UUCP avrà problemi se spostate i file di lock in posti dove non sa di doverli cercare.

getty_ps può anche usare syslogd per registrare messaggi. Vedere le pagine di manuale per syslogd(1) e syslog.conf(5) per impostare syslogd, se già non è in esecuzione. I messaggi sono registrati con priorità LOG_AUTH, gli errori usano LOG_ERR e per il debugging si usa LOG_DEBUG. Se non volete usare syslogd potete modificare tune.h nei file sorgente di getty_ps per usare invece un file di registrazione di messaggi, diciamo /var/adm/getty.log per default.

Decidete se volete la compatibilità FSSTND e la capacità di syslog. Potrete anche scegliere una combinazione dei due. Modificate Makefile, tune.h e config.h per adeguarli alle vostre decisioni. Poi compilate ed installate in base alle istruzioni incluse nel pacchetto.

Da questo punto in poi, tutti i riferimenti a getty faranno capo a getty_ps. I riferimenti a uugetty si riferiranno a uugetty che viene incluso nel pacchetto getty_ps. Queste istruzioni non funzioneranno per mgetty o agetty.

12.2 Impostare uugetty

Con uugetty potreste chiamare l'esterno con il vostro modem mentre uugetty sta controllando la porta per eventuali login. uugetty esegue degli importanti controlli di file lock. Aggiornate /etc/gettydefs per includere una voce di riferimento al vostro modem. Per un aiuto sul significato delle voci che mettete in /etc/gettydef, vedere la serial_suite raccolta da Vern Hoxie. Come recuperarla è spiegato nella sezione 11.2.3 (A proposito di getty_em). Quando avete finito di modificare /etc/gettydefs, potrete verificare l'esattezza della sintassi facendo:

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

12.2.1 I modem moderni

Se avete dei modem con velocità di 9600 bps e superiori con compressione dati potete bloccare la vostra porta seriale con una sola velocità. Ad esempio:

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

Se avete impostato il modem per un controllo di flusso hardware RTS/CTS dovete aggiungere a CRTSCTS alle voci:

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

12.2.2 Vecchi e lenti modem

Se avete un modem lento (sotto i 9600 bps), allora invece di una sola riga per una singola velocità, avrete bisogno di parecchie righe per tentare velocità diverse. Notate che queste righe sono legate tra loro dalla ultima parola della riga come ad esempio #38400. Le righe vuote sono richieste dopo ogni voce

```
# Voci 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
```

12.2.3 Messaggio di login

Se volete, potete far stampare a `uugetty` delle cosette interessanti nel messaggio di login. Negli esempi di Greg, egli ha il nome del sistema, la linea seriale, la velocità in bps corrente. Potete aggiungere altre cose:

```
@B    La velocità in bps corrente (determinato quando viene visto @B)
@D    La data corrente nel formato MM/GG/AA.
@L    La linea seriale alla quale uugetty è attaccato
@S    Il nome del sistema
@T    L'ora corrente nel formato HH:MM:SS (24 ore).
@U    Il numero degli utenti attualmente collegati. Si tratta del
      conteggio del numero di voci nel file /etc/utmp file che
      hanno un campo ut_name non vuoto
@V    Il valore di VERSION, così come risulta nei file di default
Per visualizzare un singolo carattere '@' usate '\@' o '@@'.
```

12.3 Personalizzare uugetty

Ci sono molti parametri da affinare per ogni porta che avete. Essi sono implementati in file di configurazione separati per ogni porta. Il file `/etc/conf.uugetty` verrà usato da *tutte* per tutti i riferimenti a `uugetty` e `/etc/conf.uugetty.ttyS`*N* sarà usato solo da quella porta. Esempi di file di configurazione possono essere trovati con i file sorgente di `getty_ps`, che sono inclusi in diverse distribuzioni di Linux. Per motivi di spazio non sono elencati qui. Notate che se state usando vecchie versioni di `uugetty` (inferiori a 2.0.7e) o non state usando FSSTND, allora il file di default sarà `/etc/default/uugetty.ttyS`*N*. Il `/etc/conf.uugetty.ttyS3` di Greg si presenta come segue:

```
# configurazione di esempio di uugetty per un modem Hayes compatibile
# per consentire chiamate modem dall'esterno
#
# riga da inizializzare
INITLINE=ttyS3
# timeout per disconnettere se inattivo ...
TIMEOUT=60
# stringa di inizializzazione del modem ...
# formato: <expect> <send> ... (chat sequence)
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# questa riga imposta il tempo da far trascorrere prima di inviare il
# messaggio di login
DELAY=1
#DEBUG=010
```

Aggiungete la seguente riga al vostro `/etc/inittab`, così che `uugetty` sia eseguito sulla vostra porta seriale (sostituendo le informazioni corrette per il vostro ambiente - run-level (2345 o 345 ecc.) posizione del file di configurazione, porta, velocità e tipo di terminale predefinito)

```
S3:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS3 ttyS3 F115200 vt100
```

Rilanciate `init`:

```
linux# init q
```

Per i parametri di velocità in `/etc/inittab`, usate la più alta velocità in bps che il vostro modem può supportare.

Ora Linux controllerà la vostra porta seriale per individuare delle connessioni. Componete il numero da un'altra macchina ed entrate nel vostro sistema Linux.

`uugetty` ha molte altre opzioni, controllate la pagina di manuale per `uugetty` (spesso chiamato semplicemente `getty`) per una completa descrizione. Tra le altre, c'è la capacità di scheduling ed anche di ringback automatica.

13 Che velocità dovrei usare con il mio modem?

Per velocità si intende in verità il rapporto di flusso dei dati (data flow rate) ma quasi tutti lo chiamano velocità. Per tutti i modem moderni non si ha la possibilità di scegliere la velocità che il modem usa sulla linea telefonica visto che viene scelta automaticamente come la più alta possibile date le circostanze. Ma voi avete la possibilità di scegliere che velocità sarà usata nelle comunicazioni tra il modem e il vostro computer. Questa viene chiamata velocità DTE che è l'acronimo di Data Terminal Equipment (il vostro computer è un DTE). Dovete impostare questa velocità ad un valore abbastanza alto in modo che questa parte del tragitto compiuta dal segnale non costituisca un collo di bottiglia. L'impostazione per la velocità DTE è la velocità massima. La maggior parte delle volte probabilmente opererà ad una velocità inferiore.

Per un modem esterno, la velocità DTE è la velocita (in bit per secondo) del flusso che scorre nel cavo tra il modem ed il PC. Per un modem interno, è concettualmente lo stesso, visto che il modem emula anche una porta seriale. Può sembrare ridicolo avere un limite di velocità nella comunicazione tra un computer ed una scheda modem che è direttamente collegata all'interno del computer ad bus con una velocità notevolmente superiore. Ma sarà così fino a quando i modem interni includeranno una porta seriale dedicata che ha limiti di velocità (e velocità impostabili).

13.1 Velocità e compressione dati

Che velocità scegliere? Se non fosse per la compressione dati si potrebbe scegliere una velocità DTE esattamente uguale a quella del modem. La compressione dati prende i byte inviati dal computer al modem e li codifica in un numero minore di byte. Ad esempio, se il flusso (velocità) dal PC al modem era di 20,000 byte/secondo (bps) ed il rapporto di compressione è di 2 a 1, allora solo 10,000 byte/secondo usciranno verso la linea telefonica. Quindi per un rapporto di compressione di 2:1 occorre impostare una velocità doppia rispetto alla velocità massima del modem sulla linea telefonica. Se il rapporto di compressione è di 3 a 1, occorre impostarla tre volte più veloce.

13.2 Dove imposto la velocità?

La velocità DTE è normalmente impostata da un menù nel vostro programma di comunicazione o da un'opzione data al comando getty se qualcuno vi chiama. Non potete impostare la velocità DCE modem-a-modem.

13.3 Non posso impostare una velocità sufficientemente elevata

Dovete scoprire la velocità più alta supportata dal vostro hardware. Alla fine del 1998 la maggior parte dell'hardware supportava velocità fino a 115.2 bps. Pochi modem interni a 56K supportano i 230.4K bps. Recenti kernel di Linux supportano le alte velocità (superiori a 115.2K) ma potreste avere difficoltà nell'usarle per una o entrambe delle seguenti ragioni:

1. Il programma applicativo (o stty) non accetta l'alta velocità.
2. Setserial ha una velocità di default di 115,200 (ma questo default si può cambiare facilmente)

13.3.1 Com'è impostata la velocità nell'hardware: il divisore e il baud_base

Ecco un'elenco dei divisori più comunemente usati e delle loro velocità corrispondenti (assumendo una velocità massima di 115.200): 1 (115.2K), 2 (57.6K), 3 (38.4K), 6 (19.2K), 12 (9.6K), 24 (4.8K), 48 (2.4K), 96 (1.2K), ecc. Il driver seriale imposta la velocità nell'hardware inviando al medesimo solamente un divisore (un numero intero positivo). Questo divisore divide la velocità massima dell'hardware, la velocità quindi risulta più lenta (eccetto che per divisore 1 che ovviamente dice all'hardware di lavorare a velocità massima).

In genere, se specificate una velocità di 115.2K (nel vostro programma di comunicazione o tramite stty) allora il driver seriale imposta l'hardware della porta a divisore 1 che ovviamente imposta la velocità massima. Se disponete di hardware con velocità massima di diciamo 230.4K, allora specificando 115.2K risulterà un divisore 1, quindi in realtà avrete la velocità di 230.4K. Che è la velocità doppia di quanto avete impostato. In effetti, per qualsiasi velocità che impostate, la velocità reale sarà raddoppiata. Se avete hardware che potrebbe andare a 460.8K, allora la velocità reale sarebbe il quadruplo di quella impostata.

13.3.2 Trucchetti per impostare la velocità

Per correggere questi valori (ma non sempre il problema verrà risolto) potreste usare setserial per modificare il baud_base alla vera velocità massima della vostra porta tipo 230.4K. Quindi se impostate la velocità (tramite la vostra applicazione o da stty) a 230.4K, verrà usato un divisore 1 ed otterrete la stessa velocità di quella da voi impostata. PROBLEMA: stty e molti programmi di comunicazione (alla metà del 1999) presentano ancora 115.2K quale velocità massima impostabile e non vi lasceranno impostarla a 230.4K, ecc.. Quindi in questi casi uno soluzione potrebbe essere non cambiare nulla con `setserial`, ma tenersi in mente che la velocità reale è sempre il doppio di quella che avete impostata.

C'è un altro trucco che non è molto meglio. Per usarlo impostate il baud_base (con setseria) alla velocità massima del vostro hardware. Questo corregge il conteggio così che se impostate 115.2K avrete effettivamente quella velocità. Ora dovete però ancora scoprire come impostare una velocità più alta se il vostro programma di comunicazione (o simile) non ve lo consente. Fortunatamente, setserial ha un modo per farlo: usate il parametro spd_cust con divisor 1. Poi quando impostate la velocità a 38400 nel programma di comunicazione, il divisore verrà impostato ad 1 nella porta ed opererà alla massima velocità. Ad esempio:
setserial /dev/ttyS2 spd_cust baud_base 230400 divisor 1
Non cercate di usare divisor per altri scopi diversi dallo speciale uso illustrato qui sopra (con spd_cust).

Se ci sono due o più alte velocità che volete usare che il vostro programma di comunicazione non riesce ad impostare, allora non è così facile come sopra descritto. Ma si applicano gli stessi principi. Potreste mantenere la baud_base di default e tenere presente che quando impostate una velocità in realtà impostate solo il divisore. Così la vostra velocità reale sarà sempre la vostra velocità massima divisa da qualunque divisore sia impostato nel driver seriale. Vedere 13.3.1 (Com'è impostata la velocità nell'hardware: il divisore ed il baud_base)

13.3.3 La frequenza del cristallo non è il baud_base

Notate che l'impostazione del baud_base è in genere molto inferiore rispetto a quella dell'oscillatore di cristallo nel hardware viste la frequenza del cristallo spesso si ottiene dividendo per 16 nell'hardware per ottenere la vera velocità massima. La ragione per la quale la frequenza del cristallo deve essere più altà è che può essere usata per ottenere diversi campioni di ogni bit per determinare se è un 1 o uno 0.

13.4 Tabella delle velocità

Conviene avere almeno una UART 16650 per modem a 56K. ma pochi modem la supportano. Un'alternativa è avere una 16550 che è stata truccata per dare 230400 bps. Ecco alcune velocità consigliate per impostare la vostra linea seriale se la velocità del vostro modem è:

- 56K (V.90) usate 115200 bps o 230400 bps (percentualmente di poco più veloce?)
- 28.8K (V.34), 33.6K (V.34) usate 115200 bps
- 14400 bps (V.32bis), con compressione dati V.42bis usate 57600 bps
- 9600 bps (V.32), con compressione dati V.42bis usate 38400 bps
- modem minori di 9600 bps (V.32), impostate la velocità più alta che il modem supporta.

14 Programmi di comunicazione ed utilità

PPP è di gran lunga il più usato. È usato per accedere ad Internet. Per chiamare librerie pubbliche, BBS, ecc. `minicom` è il più popolare seguito da `Seyon` (solo per X-Windows) e `Kermit`.

14.1 Minicom contro kermit

Minicom è solo un programma di comunicazione mentre Kermit è sia un programma di comunicazione che un protocollo di trasferimento file. Qualcuno potrebbe usare il protocollo Kermit in Minicom (a patto che si abbia installato Kermit sul PC). Minicom è basato sui menù mentre Kermit è basato sulla riga comando (interattiva nello speciale prompt di Kermit). Mentre Kermit è un programma libero, la documentazione non è tutta libera. Non c'è un manuale dettagliato fornito e viene suggerito di acquistare un libro come manuale. Comunque Kermit ha un aiuto in linea interattivo che dice tutto, ma manca di spiegazioni che guidino il principiante. I comandi possono essere messi in un file script così che non si debbano ripetere ogni volta. Kermit (inteso come programma di comunicazione) è più potente di Minicom.

Sebbene tutta la documentazione di Minicom sia libera, non è così esaustiva come quella di Kermit. Visto che occorre un permesso per includere Kermit in una distribuzione commerciale e visto che la documentazione non è totalmente libera, alcune distribuzioni non includono Kermit. La mia opinione è che è più facile impostare Minicom e c'è meno da imparare.

14.2 Lista di software di comunicazione

Ecco una lista di alcuni software di comunicazione dalla quale potrete scegliere. Se non sono presenti nella vostra distribuzione, dovrebbero essere disponibili via FTP. Gradirei commenti che paragonino i programmi di comunicazione. Quelli meno popolari sono obsoleti?

14.2.1 I meno popolari

- `ecu` - un programma di comunicazione
- `pcomm` - programma di comunicazione tipo `procomm` con zmodem
- `xc` - pacchetto di comunicazione xcomm

14.2.2 I più popolari

- programmi per collegarsi ad internet (ppp dialer) `chat`, `wvdial`
- `minicom` - programma di comunicazione tipo `telix`. Supporta gli script, zmodem, kermit
- *C-Kermit* `http://www.columbia.edu/kermit/`

  - portabile, supporta gli script, comunicazioni seriali e TCP/IP incluso il trasferiemnto di file, la trascodifica del set di caratteri e supporta zmodem
- `seyon` - programma di comunicazione in ambiente X
- `xc` - programma di comunicazione in ambiente X

14.2.3 Fax

- `efax` un piccolo programma fax
- `hylafax` un programma fax basato sul modello client-server
- `mgetty+fax` gestisce quanto concerne il fax e il login per chiamate in entrata

14.2.4 Software per Voicemail

- *mvm* `http://www-internal.alphanet.ch/~schaefer/mvm/` è un VoiceMail minimale for Linux
- vgetty è un'estensione di mgetty che gestisce voicemail per alcuni modem. Dovrebbe essere compreso in recenti release di mgetty.

14.2.5 Chiamate in entrata (Dial-in) (usa getty)

- `mgetty+fax` è per i modem ed è ben documentato (eccetto che per voicemail alla metà del 1999). Gestisce anche i fax e risulta un'alternativa a `uugetty`. Sta includendo capacità di voicemail (usando vgetty). Vedere 11.2.1 (mgetty)
- `uugetty` è anch'esso per i modem. È parte del pacchetto `ps_getty`. Vedere 12 (getty_ps)

14.2.6 Altri

- `callback` voi chiamate un modem remoto, poi quel modem riappende e vi richiama (per risparmiare sulla bolletta).
- `SLiRP` e `term` forniscono un servizio tipo PPP che potete lanciare nello spazio utente su di un computer remoto con uno shell account Vedere 14.3 (term e SLiRP) per ulteriori dettagli.
- `ZyXEL` è un programma di controllo per i modem ZyXEL U-1496. Gestisce chiamate, connessioni, richiamate di sicurezza, FAX e funzioni vocali di mailbox
- Software SLIP e PPP può essere trovato in: *ftp://metalab.unc.edu/pub/Linux/system/network/serial* .
- Altre cose si possono trovare in: *ftp://metalab.unc.edu/pub/Linux/system/serial* e *ftp://metalab.unc.edu/pub/Linux/apps/serialcomm* o in uno dei molti mirror. Queste sono le directory dove sono archiviati i programmi.

14.3 SLiRP e term

`SLiRP` e `term` sono programmi che servono solo se avete un account shell in dial-up su di una macchina tipo Unix e volete avere l'equivalente di un account PPP (o simile) senza essere autorizzati ad averlo (forse perché non volete pagare per questo, ecc.). `SLiRP` è più popolare di `term` che è praticamente obsoleto.

Per usare `SLiRP` installatelo nel vostro account nel computer remoto. Poi chiamate l'account e lanciate SLiRP sul remoto e PPP sul vostro PC locale. Ora avete una connessione PPP attraverso la quale potete lanciare un browser web sul vostro PC tipo Netscape, ecc. Potrebbero esserci alcuni problemi visto che SLiRP non è così valido come un vero account PPP. Alcuni account potrebbero fornire SLiRP visto che esso fa risparmiare sugli indirizzi IP (non avete indirizzo IP mentre usate SLiRP).

`term` è qualcosa di simile a SLiRP solo che dovete eseguire `term` sia sul computer locale che quello remoto. Non c'è PPP sulla linea telefonica visto che `term` usa il suo proprio protocollo. Per usare `term` dal vostro PC vi occorrerà una versione term-aware di ftp per eseguire ftp, ecc. Quindi è più facile usare SLiRP visto che la versione normale di ftp funziona bene con SLiRP. C'è un Term HOWTO non aggiornato.

15 Cosa sono le UART? In che modo influenzano le prestazioni?

15.1 Introduzione alle UART

(Questa sezione è anche nel Serial-HOWTO)

UART (**U**niversal **A**synchronous **R**eceiver **T**ransmitter) - ricevitore/trasmettitore asincrono universale - sono chip seriali sulla scheda madre del vostro PC (o su una scheda modem interna). La funzione della UART può anche essere svolta da un chip che fa anche altre cose. Sui computer più vecchi come la maggior parte dei 486, i chip erano sulla scheda del controller I/O. Alcuni computer più vecchi hanno schede seriali dedicate.

Scopo della UART è convertire i byte dal bus parallelo del PC in un flusso seriale di bit. Il cavo che esce dalla porta seriale è seriale ed ha solo un cavo per ogni direzione di flusso. La porta seriale invia un flusso di bit, un bit alla volta. Al contrario, il flusso di bit che entra dalla porta seriale via cavo esterno viene convertito in byte paralleli che il computer può comprendere. Le UART trattano dati divisi in pezzi della dimensione di un byte, che per convenienza è anche la dimensione dei caratteri ASCII.

Diciamo che avete un terminale collegato al vostro PC. Quando digitate un carattere, il terminale consegna quel carattere al proprio trasmettitore (che è anche una UART). Il trasmettitore invia quel byte alla linea seriale, un bit alla volta ad una velocità specifica. Dalla parte del PC, l'UART ricevente prende tutti i bit e ricostruisce (parallelizza) il byte e lo mette in un buffer.

Oltre alla conversione tra seriale e parallelo, la UART esegue alcune altre cose come derivazione (effetto collaterale) dei suoi compiti primari. Il voltaggio usato per rappresentare i bit viene anch'esso convertito (cambiato). I bit extra (chiamati bit di start e stop) sono aggiunti ad ogni byte prima che venga trasmesso. Vedere la sezione Voltage Waveshapes del Serial HOWTO per dettagli. Inoltre, mentre il rapporto di flusso (in byte per secondo) sul bus parallelo all'interno del computer è molto alto, il rapporto di flusso in uscita dalla UART dalla parte della porta seriale è molto più basso. L'UART ha un elenco fisso di rapporti (velocità) che può usare come interfaccia per la sua porta seriale.

15.2 Due tipi di UART

Ci sono due tipi principali di UART: UART stupide (dumb) e UART FIFO. Le UART stupide sono le 8250, 16450, le prime 16550 e le prime 16650. Esse sono obsolete ma se imparate come lavorano, è poi facile comprendere come funzionano le moderne UART FIFO (le ultime 16550, 16550A, 16c552, le ultime 16650 e 16750 e 16C950).

Esiste un poco di confusione per quanto riguarda le 16550. I primi modelli avevano un bug e lavoravano correttamente solo come 16450 (non FIFO). I modelli più recenti con questo bug corretto, vennero chiamate 16550A, ma la maggior parte dei costruttori non fecero proprio il cambiamento di nome e continuarono a chiamarle 16550.

La maggior parte delle 16550 in uso oggi sono come le 16550A. Linux la rileverà come 16550A anche se il manuale dell'hardware (o l'etichetta) dicono che si tratta di una 16550. Una situazione simile esiste per le 16650 (solo che è peggio visto che il costruttore sembra che non ammetta che c'è qualcosa di sbagliato). Linux rileverà un'ultima 16650 come una 16650V2. Se esso la rileva come una 16650 ci sono cattive notizio e verrà usata come se avesse un buffer da un byte.

15.3 FIFO

Per comprendere la differenza tra stupide (dumb) e FIFO (First In, First Out - Il primo che entra è il primo che esce) esaminiamo prima cosa succede quando una UART ha inviato o trasmesso un bute. La UART stessa non può fare nulla con i dati che passano attraverso di essa, semplicemente li riceve e li spedisce. Nella UART stupide originali, la CPU riceveva un interrupt dal driver seriale ogniqualvolta un byte veniva ricevuto o trasmesso. La CPU poi spostava il byte ricevuto dal buffer della UART in una qualche parte della memoria, oppure dava alla UART un altro byte da spedire. Le UART 8250 e 16450 hanno un buffer di 1 byte. Ciò significa che ogni volta che 1 byte viene inviato o ricevuto, la CPU riceveva un interrupt. A basse velocità di trasferimento, questo andava bene. Ma ad alte velocità di trasferimento la CPU diventava così impegnata a gestire la UART, che non aveva il tempo per soddisfare adeguatamete gli altri compiti. In alcuni casi, la CPU non era pronta a servire le richieste di interrupt in tempo ed il byte era sovrascritto perché arrivava troppo velocemente. Questo viene chiamato overrun o overflow (sovraccarico).

Ecco dove le UART FIFO sono utili. I chip delle 16550A (o 16550) hanno un buffer FIFO di 16 byte. Questo significa che può ricevere fino a 14 byte (od inviarne 16) prima che debba inviare un interrupt alla CPU. Non solo essa può attendere ulteriori byte, ma la CPU può poi trasferirli tutti e 14 (o più) in un colpo solo. Questo è un vantaggio significativo rispetto alle altre UART, cha hanno buffer di 1 solo byte. La CPU riceve meno interrupt ed è libera di fare altre cose. I dati non sono sono persi e tutti sono contenti. Notate che la soglia di interrupt dei buffer FIFO (livello di attivazione) può essere impostata a meno di 14. 1, 4 e 8 sono altro possibili scelte.

Mentre la maggior parte dei PC hanno delle 16550 con 16 byte di buffer, le UART migliori hanno buffer ancora più grandi. Notate che l'interrupt viene inviato leggermente prima che il buffer si riempia del tutto (diciamo ad un trigger level di 14 byte per un buffer da 16 byte). Questo consente di fare spazio a qualche alto byte da ricevere durante il tempo che trascorre mentre la richiesta di interrupt viene esaudita. Il trigger level può essere impostato a diversi valori consentiti dal software kernel. Il trigger level di 1 sarà quasi come una UART stupida (a parte il fatto che comunque ha ancora spazio per altri 15 byte prima che venga inviato l'interrupt).

Se digitate qualcosa mentre state visitando una BBS, i caratteri che digitate escono attraverso la porta seriale. I vostri caratteri digitati che si vedono sullo schermo sono quelli che sono stati riecheggiati dalla linea telefonica attraverso il vostro modem, quindi attraverso la vostra porta seriale verso lo schermo. Se si ha un buffer a 16 byte sulla porta seriale che trattiene i caratteri fino a che ne ha 14, dovrete digitare parecchi caratteri prima di poter vedere quello che avete digitato in precedenza. (prima che essi appaiono sullo schermo). Questo può creare confusione ma esiste un timeout per prevenire ciò. Quindi normalmente si vede un carattere sullo schermo non appena lo si digita.

Il timeout lavora in questo modo per il buffer della UART che riceve: se i caratteri arrivano uno di seguito all'altro, allora una richiesta di interrupt viene inviata solo quando il 14° carattere raggiunge il buffer. Ma se un carattere arriva ed il successivo non arriva subito dopo, viene richiesto un interrupt. Questo succede anche se non ci sono 14 caratteri nel buffer (potrebbe anche essercene solo uno). Quindi quando quello che si digita passa attraverso questo buffer, esso si comporta come un buffer da 1 byte anche se in realtà è un buffer da 16 byte (a meno che la velocità di battitura non sia centinaia di volte superiore al normale). Vi è un timeout anche per il buffer di trasmissione.

15.4 Numeri di modello di UART

Ecco un elenco di UART. *TL* è il *T*rigger *L*evel

- 8250, 16450, prime 16550: Obsolete con buffer da 1 byte
- 16550, 16550A, 16c552: buffer da 16 byte TL=1,4,8,14
- 16650: buffer da 32-byte. Velocità fino a 460.8 Kbps
- 16750: buffer da 64-byte per trasmissione, 56-byte per ricezione. Velocità fino a 921.6 Kbps
- Hayes ESP: buffer da 1K

Quelle obsolete vanno bene solo per modem non superiori a 14.4.k (DTE con velocità fino a 38400 bps). Per i modem moderni occorre almeno una 16550 (e non una delle prime 16550). I modem V.90 56k potrebbero essere percentualmente di molto piu' veloci con una 16650 (specialmente se si scaricano file non compressi). Il vantaggio principale della 16650 è la maggiore dimensione del suo buffer visto che una velocità extra non è necessaria a meno che il rapporto di compressione del modem sia alto. Alcuni modem interni a 56k potrebbero essere provvisti di una 16550 ??

Schede multiporta intelligenti e senza UART usano chip DSP per il buffering e il controllo addizionale, liberando la CPU ancora di più. Ad esempio, le schede Cyclades Cyclom e Stallion EasyIO usano una Cirrus Logic CD1400 RISC UART, e molte schede usano CPU 80186 od anche CPU RISC speciali per gestire l'IO seriale.

La maggioranza dei PC più nuovi (486, Pentium o superiori) hanno una 16550A (generalmente chiamata solo 16550). Se avete un qualcosa di veramente vecchio il chip può essere scollegato in modo che ci si possa aggiornare acquistando un chip 16550A rimpiazzando la UART 16450 esistente. Se le funzionalità sono state immesse in un altro tipo di chip, allora siete sfortunati. Se la UART è agganciata ad incastro, allora l'aggiornamento è facile (se siete in grado di trovare un rimpiazzo). Le nuove e le vecchie hanno i pin compatibili. Potrebbe essere più praticabile acquistare semplicemente una nuova scheda seriale su Internet (alcuni negozi al dettaglio ne hanno a disposizione oggi).

16 Risoluzione di problemi

16.1 Il mio modem è fisicamente al suo posto ma non può essere trovato

I messaggi di errore potrebbero essere qualcosa del tipo No modem detected (nessun modem rilevato) , Modem non responding (il modem non risponde), o (strano) You are already online (Sei già collegato) (da minicom). Se avete installato un modem interno (con la porta seriale inclusa nel modem) o ne state usando uno esterno e non sapete a che porta seriale sia connesso il problema è cercare la porta seriale. Vedere 16.9 (La mia porta seriale è fisicamente presente ma non può essere trovata). Questa sezione riguarda come scoprire quale porta seriale ha un modem ad essa connesso.

C'è un programma che cerca i modem sulle porte seriali comunemente usate chiamato wvdialconf. Digitate semplicemente wvdialconf <un-nuovo-nome-file>. Verrà creato il nuovo file come file di configurazione ma non avrete bisogno di esso a meno che non andiate ad usare wvdial per telefonare. Vedere 9.3 (Cos'è wvdialconf ?). Sfortunatamente, se il vostro modem è in modalità online data, wvdialconf visualizzerà No modem detected. Vedere 16.1.1 (Nessuna risposta ad AT).

Il vostro problema potrebbe essere dovuto ad un winmodem (o simile) che non può essere usato con Linux. Vedere 2.4 (Evitare la maggior parte dei software modem). Il programma setserial potrebbe essere usato per identificare le porte seriali ma non rileverà se ci sono modem collegati ad esse. Quindi è meglio provare ad usare prima wvdialconf.

Un altro modo di vedere se c'è un modem su una porta è lanciare minicom sulla porta (dopo avere precedentemente impostato minicom sulla corretta porta seriale – dovrete salvare le impostazioni, quindi uscire da minicom e farlo ripartire). Poi digitate AT e dovreste vedere OK (o 0 se è impostato per restituire codici numerici di risultato (digit result codes)). I risultati potrebbero essere:

- Nessuna risposta. Vedere 16.1.1 (Nessuna risposta ad AT)
- Se ci vogliono parecchi secondi per ottenere una risposta (incluso anche solo il cursore che si sposta di una riga) allora vedere 16.10 (Estremamente lento: il testo appare sullo schermo lentamente e dopo lunghi ritardi)
- Alcuni strani caratteri appaiono, ma non in risposta ad AT. Questo con ogni probabilità significa che il vostro modem è ancora connesso a qualcosa dall'altra parte della line telefonica che sta inviando alcuni criptici pacchetti o simile.

16.1.1 Nessuna risposta ad AT

Il modem dovrebbe inviarvi un OK in risposta al vostro AT, che digitate al modem (usando minicom o simile). Se non vedete OK (e in parecchi casi non vedete neppure AT che avete digitato) il modem non sta rispondendo (dando per scontato che ci sia davvero un modem verso la porta sulla quale state digitando).

Una ragione per la quale un vero modem non risponde è che si trova in modo online data dove non può accettare alcun comando AT. Potrebbe essere in uso con un altro processo. Se detto processo è in esecuzione sulla porta, potreste vederlo digitando ps -t ttyS2 o simile. Comunque il processo che sta usando la porta seriale (dove si trova il modem) potrebbe essere in esecuzione su di un terminale tipo /dev/ttyS1 e non sarà individuato usando il comando sopracitato.

Potreste stare usando il modem ed improvvisamente siete stati bruscamente disconnessi (come ad esempio uccidendo il processo con segnale 9). In quel caso il vostro modem non viene reimpostato a modalità comandi dove può interagire con i comandi AT. Ecco quindi il messaggio di minicom You are already online, hangup first (siete già in linea, interrompete la comunicazione prima). Bene, sembra che siate in linea ma potreste non essere collegati con nessuno attraverso la linea telefonica, wvdial visualizzerà modem not responding (il modem non risponde) nella medesima situazione.

Per risolvere questo problema come soluzione estrema spegnete e riavviate il computer. Un altro modo è inviare +++ al modem per dirgli di ritornare a modalità comandi dalla modalità in linea. Da entrambe le parti della sequenza +++ deve esserci circa 1 secondo di ritardo (nulla inviato durante il guard time). Questo potrebbe non funzionare se un altro processo sta usando il modem visto che la sequenza +++ potrebbe essere attaccata ad altri caratteri inseriti prima o dopo il +++ (durante il guard time). Ironicamente, anche se la linea del modem fosse inattiva, inviandogli un inaspettato +++ probabilmente si darà il via ad uno scambio di pacchetti (o simile) che violeranno il richiesto guard time così che +++ non fa quello che voi avreste voluto. +++ in genere si trova nella stringa che viene chiamata hangup string (stringa di interruzione) così se dite a minicom (o simile di interrompere la comunicazione) potrebbe funzionare. Un altro modo di fare questo è semplicemente quello di uscire da minicom e poi rilanciarlo.

16.2 Non posso avvicinarmi ai 56K dal mio modem a 56K

Deve esserci un livello di disturbo molto basso sulla linea perché il modem possa solo avvicinarsi ai 56K. Alcune linee telefoniche sono così cattive che le velocità ottenibili sono molto più lente di 56K (tipo 28.8 od anche meno). Alcune volte telefoni supplementari connessi alla stessa linea possono causare problemi. Per controllare questo potreste cercare di connettere il vostro modem direttamente nel punto dove la linea telefonica si immette nell'edificio, disconnettendo qualsiasi altra cosa che quella linea alimenti (se nessuno ha qualcosa in contrario).

16.3 L'invio o lo scarico di file è lento o interrotto.

Il controllo di flusso (per il PC e/o da modem a modem) potrebbe non essere abilitato. Per il caso dell'invio di file: Se avete impostato un'altra velocità DTE (tipo 115.2K) allora il flusso dal vostro modem al vostro PC potrebbe funzionare bene ma molto del flusso di invio nell'altra direzione non passerà completamente a causa del collo di bottiglia della linea telefonica. Questo risulterà in molti errori e nel reinvio dei pacchetti. Potrebbe anche volerci un tempo oltremodo lungo per spedire un file. In alcuni casi, i file non arrivano neppure.

Per il caso dello scarico di file: Se state scaricando dei grandi file non compressi o delle pagine web, (ed il vostro modem usa la compressione dei dati) oppure se avete impostata una bassa velocità DTE, allore lo scarico potrebbe essere interrotto a causa della mancanza del controllo di flusso.

16.4 Ricevendo una chiamata continuo a ricevere line NNN of inittab invalid

Ovvero: `riga NNN di inittab non valida`

Assicuratevi di usare la sintassi corretta per la vostra versione di `init`. I diversi `init` esistenti usano sintassi differenti nel file `/etc/inittab`. Assicuratevi di usare la sintassi corretta per la vostra versione di `getty`.

16.5 Continuo a ricevere "Id S3 respawning too fast: disabled for 5 minutes"

Id S3 è solo un esempio. In questo caso cercate la riga che inizia con S3 in `/etc/inittab`. Questa è la causa del problema. Assicuratevi che la sintassi per questa riga sia corretta, che il dispositivo (ttyS3) esista e che possa essere trovato.

Assicuratevi che il vostro modem sia configurato correttamente. Guardate i registri `E` e `Q`. Questo può capitare quando il vostro modem sta chiaccherando con `getty`.

Per uugetty, verificate che la sintassi del vostro `/etc/gettydefs` sia corretta eseguendo il seguente comando:

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

Questo può anche capitare quando l'inizializzazione di `uugetty` fallisce. Vedere la sezione 16.7 (uugetty non funziona ancora).

16.6 Il mio modem è bloccato dopo che qualcuno ha agganciato, oppure uugetty non riparte

Questo può capitare quando il vostro modem non effettua il reset dopo che è caduto il DTR. Greg Hankins ha visto i suoi led RD e SD impazzire quando questo è successo. Dovete fare resettare il vostro modem. Molti modem Hayes compatibili fanno questo con `&D3`, ma sul suo USR Courier, ha dovuto impostare `&D2` e `S13=1`. Controllate il manuale del modem (se ne avete uno).

16.7 uugetty non funziona ancora

Esiste un'opzione `DEBUG` all'interno di `getty_ps`. Modificate 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 quello che state cercando di debuggare:

```
D_OPT   001          impostazione opzioni
D_DEF   002          elaborazione file di default
D_UTMP  004          elaborazione di utmp/wtmp
D_INIT  010          inizializzazione della linea (INIT)
D_GTAB  020          elaborazione del file gettytab
D_RUN   040          altre diagnostiche di runtime
D_RB    100          ringback debugging
D_LOCK  200          elaborazione del lockfile di uugetty
D_SCH   400          elaborazione schedule
D_ALL   777          tutto
```

Impostare `DEBUG=010` è un buon punto da cui partire.

Se avete in esecuzione `syslogd`, le informazioni di debugging appariranno nei vostri file di log. Se non avete in esecuzione `syslogd` le informazioni appariranno in `/tmp/getty:ttyS`*N* per il debugging `getty` e `/tmp/uugetty:ttyS`*N* per `uugetty` ed in `/var/adm/getty.log`. Controllate la informazioni di debugging e vedete cosa sta succedendo. Molto probabilmente, dovrete regolare alcuni parametri nel vostro file di configurazione e riconfigurare il modem.

Potete anche provare `mgetty`. Alcune persone hanno avuto miglior fortuna con quest'ultimo.

Change log: Apr. 00: 2 porte allo stesso indirizzo

16.8 Le seguenti sottosezioni sono sia nel Serial che nel Modem HOWTO:

16.9 La mia porta seriale è fisicamente lì ma non può essere trovata

Se un dispositivo (tipo un modem) pare funzionare allora la porta seriale è stata trovaata. Se non funziona per niente, allora occorre assicurarsi che la vostra porta seriale possa essere trovata.

Controllate i menù ed i messaggi del BIOS. Per i bus PCI usate lspci Se è una porta seriale PnP su bus ISA provate pnpdump –dumpregs e/o consultate il Plug-and-Play HOWTO. Usando scanport otterrete una esplorazione di tutte le porte sul bus ISA e potreste scopire una porta sconosciuta che potrebbe essere una porta seriale (ma la porta non viene verificata). Il PC potrebbe bloccarsi. Potreste provare a rilevarla con setserial. Vedere 9.1.2 (Probing). Se nulla sembra passare attraverso la porta, essa potrebbe esserci ma avere un interrupt sbagliato. Vedere 16.10 (Estremamente lento: il testo appare sullo schermo lentamente e dopo lunghi ritardi)

Se due porta hanno lo stesso indirizzo IO allora la verifica indicherà in modo errato solo una porta. La rilevazione Plug-and-play troverà entrambe le porte così questo potrebbe essere un problema solo se almeno una delle porte non è Plug-and-play. Errori di tutti i tipi potrebbero verificarsi per i dispositivi che stanno condividendo una porta ma il fatto che ci siano due dispositivi sulla stessa porta non sembra essere stato rilevato (eccetto, si spera, da voi). Se gli IRQ sono diversi la rilevazione dell'IRQ con setserial potrebbe scopire questa situazione non riuscendo a rilevare un IRQ. Vedere 9.1.2 (Rilevamento).

16.10 Estremamente lento: il testo appare sullo schermo lentamente e dopo lunghi ritardi

È probabilmente un conflitto od una errata impostazione di interrupt. Ecco alcuni sintomi di quello che potrebbe succedere la prima volta che cercate di usare un modem, terminale o stampante. In alcuni casi digitate qualcosa ma sullo schermo non appare nulla se non dopo parecchi secondi. Solo l'ultimo carattere digitato potrebbe apparire. Potrebbe anche essere solo un invisibile carattere di <return> così tutto quel che noterete sarà che il cursore scende di una riga) In altri casi dove dovrebbero comparire molti dati sullo schemo, sono un gruppo di circa 16 caratteri compare. Poi c'è un'attesa di parecchi secondi per il prossimo gruppo di caratteri. Potreste anche ricevere messaggi di errore di input overrun (o trovarli nei file di log)

Per ulteriori dettaglio sui sintomi e perché questo accade vedere il Serial-HOWTO sezione Interrupt Problems Details.

Se la cosa coinvolge anche dispositivi Plug-and-Play, vedere anche il Plug-and-Play HOWTO.

Come veloce controllo per vedere se veramente si tratta di un problema di interrupt, impostate l'IRQ a zero con setserial. Questo dice al driver di non usare gli interrupt ma il polling. Se sembra che questo risolva i problemi di lentezza, allora si tratta di un problema di interrupt. Dovreste comunque cercare di risolverlo visto che il polling usa esagerate risorse del computer.

Cercare di trovare il conflitto di interrupt potrebbe non essere semplice visto che si suppone che Linux non consenta alcun conflitto di interrupt e di conseguenza vi invii un errore di 16.17 (/dev/ttys?: Device or resource busy (Dispositivo o risorsa impegnata)) se pensa che stiate tentando di creare un conflitto. Ma un vero conflitto può essere creato se setserial dispone di informazioni errate. Quindi usare setserial non farà rilevare alcun conflitto (e neppure guardando in /proc/interrupts che basa le suo informazioni su setserial). Dovete quindi sapere quello che setserial pensa così che possiate evidenziare quello che è sbagliato e cambiarlo quando avrete determinato quello che è veramente impostato nell'hardware.

Quello che dovete fare è verificare come è impostato l'hardware controllando i jumper od usando software PnP per controllare le vere impostazioni dell'hardware. Per il PnP potete lanciare pnpdump –dumpregs (se è un bus ISA) o lspci (bus PCI). Confrontate i risultati con quello che Linux pensa sia impostato nell'hardware.

16.11 Abbastanza lento: Mi aspettavo che fosse un poco più veloce

Una ragione potrebbe essere che qualunque cosa sia sulla porta seriale (tipo un modem, un terminale, una stampante) non funziona così veloce come pensate che dovrebbe. Un modem a 56k raramente funziona a 56k ed Internet spesso è congestionata ed ha colli di bottiglia che rallentano le cose. Se il modem dall'altra parte non ha una connessione digitale alla linea telefonica (ed usa uno speciale modem digitale non in vendita nella maggior parte dei negozi di computer), allora le velocità oltre i 33,6k non sono possibili.

Un'altra possibile ragione è che il driver seriale pensa che abbiate una porta seriale obsoleta (UART 8250, 16450, o le prime 16550). Vedere 15 (Cosa sono le UART?). Usate setserial -g /dev/ttyS*. Se mostra qualsiasi cosa meno di 16550A, allora è probabilmente proprio questo il problema. Se invece setserial lo identifica così sbagliando, modificatela. Vedere 9.1 (Cos'è Serial?) per maggiori informazioni. Naturalmente se avete veramente una porta obsoleta, mentire a setserial su questo non farà altro che peggiorare le cose.

16.12 La schermata di avvio mostra IRQ sbagliati per le porte seriali.

Linux non esegue nessuna identificazione di IRQ in avvio. Quando il modulo di setserial si carica esegue semplicemente una rilevazione del dispositivo seriale. Quindi non fate caso a quello che dice circa l'IRQ, perché sta semplicemente assumendo gli IRQ standard. Così è fatto, perché l'identificazione degli IRQ è inaffidabile e potrebbe essere ingannevole. Ma se e quando setserial viene lanciata da uno script di avvio, cambia gli IRQ e visualizza il nuovo (ed auspicabilmente corretto) stato nella schermata di partenza. Se l'IRQ sbagliato non viene corretto da una seguente visualizzazione sullo schermo, allora avete un problema.

Quindi, anche se ho la mia `ttyS2` impostata ad IRQ 5, continuo a vedere

```
ttyS02 at 0x03e8 (irq = 4) is a 16550A
```

all'inizio quando parte Linux (i vecchi kernel potrebbero mostrare ttyS02 come tty02) Dovete usare `setserial` per dire a Linux che IRQ state usando.

16.13 Cannot open /dev/ttyS?: Permission denied

Ovvero: `Impossibile aprire /dev/ttyS?:  Permesso negato`

Controllare i permessi su questa porta con ls -l /dev/ttyS?. Se siete i proprietari di questa ttyS? allora dovete avere i permessi di lettura e scrittura: crw con il c (Character device) in colonna 1. Se non siete i proprietari dovreste invece vedere rw- nelle colonne 8 e 9 che significa che tutti hanno diritti di lettura e scrittura su di esso. Usate chmod per cambiare i permessi. Ci sono modi più complicati per ottenere accessi tipo appartenere ad un gruppo che ha permessi di gruppo.

16.14 Operation not supported by device per ttySx

Ovvero: `Operazione non supportata dal dispositivo`

Questo vuol dire che un operazione richiesta da setserial, stty, ecc. non può essere effettuata perché il kernel non la supporta. In precedenza questo era spesso causato dal fatto che il modulo seriale non era stato caricato. Ma con l'avvento del PnP, potrebbe più probabilmente dire che non c'è un modem (od altro dispositivo seriale) all'indirizzo dove il driver (e setserial) pensa che sia. Se non c'è un modem lì, i comandi inviati a quell'indirizzo ovviamente non vengono eseguiti. Vedere 6.5.2 (Cos'è impostato nell'hardware della mia porta seriale?).

Se il modulo seriale non era caricato ma lsmod mostra che ora è caricato potrebbe essere il caso che sia stato caricato adesso ma non lo era quando si è ricevuto il mesasggio di errore. In molti casi il modulo verrà automaticamente caricato quando necessario (se si riesce a trovare). Per forzare il caricamento del modulo seriale si potrebbe elencarlo nel file /etc/modules.conf o /etc/modules. Il vero modulo dovrebbe risiedere in /lib/modules/.../misc/serial.o.

16.15 Cannot create lockfile. Sorry (Spiacente, non posso creare il file di lock)

Ovvero: `Spiacente, impossibile creare il file di lock.`

Quando viene aperta una porta da un programma viene creato un file di lock in /var/lock. Errati permessi per la directory lock non consentiranno la creazione di un file di lock. Usare ls -ld /var/lock per vedere se i permessi sono a posto: in genere rwx per tutti (ripetuto 3 volte). Se è sbagliato, usate chmod per mettere a posto lo cose. Naturalmente, se non esiste una directory di lock nessun file di lock potrà esservi creato. Vedere la sottosezione del Serial-HOWTO: What Are Lock Files.

16.16 Device /dev/ttyS? is locked. (Il dispositivo /dev/ttyS? è bloccato)

Ovvero: `Il dispositivo` dev/ttyS? è bloccato./

Questo vuol dire che qualcun altro (o qualche altro processo) sta presumibilmente usando la porta seriale. Ci sono diversi modi per cercare di scoprire quale processo la sta usando. Un modo è dare un occhiata al contenuto del file di lock (/var/lock/LCK...). Dovrebbe essere l'identificativo del processo. Se l'identificativo è diciamo 261 digitate ps 261 per scoprire cosa sia. Poi, se il processo non è più necessario, potrebbe essere gentilmente eliminato da kill 261. Se si rifiuta di essere eliminato usate kill -9 261 per forzare l'eliminazione, ma in questo caso il file di lock non sarà rimosso, e dovrete provvedere manualmente. Naturalmente se non c'è un processo 261 allora basta che rimuoviate il file di lock, ma nella maggior parte dei casi il file di lock dovrebbe esser stato automaticamente rimosso se contiene un identificativo di processo chiuso (come 261).

16.17 /dev/ttyS?: Device or resource busy (Dispositivo o risorsa occupati)

Significa che il dispositivo al quale state tentando di accedere (o usare) è presumibilmente occupato (in uso) o che una risorsa di cui necessita (tipo un IRQ) è presumibilmente in uso da parte di un altro dispositivo. Talvolta è davvero occupato ma in altri casi sembra erroneamente in uso.

resource busy (risorsa impegnata) spesso vuol dire (ad esempio per `ttyS2`) Non puoi usare `ttyS2` visto che un altro dispositivo sta usando l'interrupt di ttyS2. Il potenziale conflitto di interrupt è determinato da quello che pensa setserial. Un messaggio di errore più accurato dovrebbe essere Non posso usare `ttyS2` visto che i dati di setserial (e quelli del kernel) indicano che un altro dispositivo sta usando l'interrupt di `ttyS2`. Se due dispositivi usano lo stesso IRQ e voi fate partire uno solo dei due dispositivi, tutto è a posto perché non c'è ancora conflitto. Ma quando in seguito cercate di lanciare il secondo dispositivo (senza chiudere il primo) otterrete il messaggio di errore ... resource busy. Questo perché il kernel tiene traccia solamente di quale IRQ è realmente in suo e i conflitti non capitano a meno che i dispositivi siano in uso (aperti)

Ci sono due casi: Potrebbe esserci un vero conflitto di interrupt che si sta evitando. Ma se setserial ha sbagliato, allora non dovrebbe esserci alcuna ragione perché `ttyS2` non possa essere usata, eccetto che setserial ha erroneamente previsto un conflitto. Quello che occorre fare è scoprire quale interrupt setserial pensa che stia usando `ttyS2`. Più facile a dirsi che a farsi visto che non potete usare il comando setserial per `ttyS2` visto che l'IRQ per ttyS2 è presumibilmente occupato ed otterreste lo stesso messaggio di errore ... busy. Per risolvere o eseguite un riavvio oppure: uscite od eliminate tutti i processi che potrebbero cauasre il conflitto. Se riavviate: 1. osservate i messaggi in fase di avvio relativi alle porte seriali. 2. Sperate che il file che lancia setserial all'avvia non sia esso stesso a creare ancora lo stesso conflitto.

Se pensate di sapere quale IRQ stia usando `ttyS2` allora potreste dare uno sguardo a /proc/interrupts per scoprire chi altro sta attualmente usando questo IRQ. Potreste anche voler fare un doppio controllo perché qualsiasi IRQ mostrato qui (e da setserial) sia corretto (lo stesso di quello impostato nell'hardware). Un modo per provare se c'è o meno un conflitto di interrupt è impostare l'IR1 a 0 (polling) usando setserial. Se il messaggio di risorsa occupata (busy) scompare, probabilmente c'è un potenziale conflitto di interrupt. Non è una buona idea lasciarlo permanentemente impostato a 0 visto verranno usate più risorse della CPU.

Questo paragrafo è principalmente per il caso in cui un modem è usato sia per chiamare che per ricevere chiamate. Se il segnale DSD è inviato alla porta, quello porta penserà che sia occupata. Questo problema può sorgere quando state cercando di chiamare con un modem quando DTC o DTR non sono implememntati correttamente. DCD dovrebbe essere attivo quando c'è una effettiva connessione (ad esempio qualcuno ci ha chiamati), non quando `getty` sta guardando la porta. Assicuratevi che il vostro modem sia configurato per attivare DCD solo quando c'è una connessione. DTR dovrebbe essere attivo ogniqualvolta qualcuno sta usando o guardando la linea, tipo `getty`, `kermit`, o qualche altro programma di comunicazione.

16.18 Strumenti per la risoluzione dei problemi

Questi sono alcuni dei programmi che potreste aver bisogno di usare per la risoluzione dei problemi:

- lsof /dev/ttySx farà un elenco delle porte seriali che sono aperte.
- setserial mostra ed imposta la configurazione hardware a basso livello di una porta (quella che il driver pensa che sia). Vedere 9.1 (Cos'è setserial).
- stty mostra ed imposta la configurazione di una porta (ad eccezione di quanto gestito da setserial). Vedere la sezione del Serial-HOWTO: Stty,
- modemstat o statserial che mostrano lo stato corrente dei vari segnali di linea (tipo DTR, CTS, ecc.) di segnale (tipo DTR, CTS, ecc)
- irqtune darà agli interrupt della porta seriale una priorità maggiore per migliorare le prestazioni.
- hdparm per la messa a punto dell'hard-disk potrebbe fornire un ulteriore aiuto
- lspci mostra gli effettivi IRQ, ecc, dell'hardware del bus PCI.
- pnpdump –dumpregs mostra gli effettivi IRQ, ecc., dell'hardware per i dispositivi PnP sul bus ISA.
- Alcuni file nella directory /proc (tipo ioports e interupts).

17 Aggiornamenti delle memorie Flash

Molti modem possono essere aggiornati riprogrammando la loro memoria flash tramite un programma di aggiornamento che si ottiene da Internet. Inviando questo programma dal PC tramite la porta seriale al modem, il modem immagazzinerà il programma nella sua memoria non volatile (quella che rimane anche quando il modem viene spento). Le istruzioni di installazione vertono generalmente su quanto occorre fare sotto Windows così dovrete scoprire come fare l'equivalente sotto Linux (a meno che si voglia installare l'aggiornamento sotto Windows). L'invio di un programma al modem viene spesso chiamato download.

Se l'ultima versione di questo HOWTO contiene ancora questa richiesta (vedere 1.5 (Nuove versioni di questo HOWTO)) per favore illustratemi la vostra esperienza installando questi aggiornamenti, che potrebbe essere utile ad altri.

Ecco l'idea di base per eseguire un aggiornamento (upgrade). Per prima cosa, potrebbe esserci un comando che dovete inviare al vostro modem per dirgli che quello che segue è un aggiornamento della flash ROM. In un caso questo comando era AT**. Per fare questo lanciano un programma di comunicazione (tipo minicom) e digitare. Prima digitate AT <enter> per vedere se vostro modem è lì e risponde OK.

Poi, dovete inviare un file (talvolta due file) direttamente al modem. I programmi di comunicazione (come minicom) spesso usano zmodem o kermit per inviare file al modem (e oltre) ma essi mettono il file dentro dei pacchetti ai quali aggiungono delle intestazioni, mentre voi volete che sia inviato il file esatto, non uno modificato. Ma il programma kermit ha un comando transmit che invierà il file direttamente (senza usare i pacchetti kermit), così questo è un modo di inviare un file direttamente. Minicom al 1998 non ha questa capacità.

Un altro modo di inviare il(i) file potrebbe essere uscire dal programma di comunicazione aprendo una shell (in minicom si usa ˆAJ) poi: `cat nome_file_aggiornamento > /dev/ttyS2` (se la vostra porta seriale è ttyS2). Poi tornare al programma di comunicazione (digitando fg al prompt della riga comandi in minicom) per vedere cosa è successo.

Ecco una sessione di esempio per un determinato modem Rockwell (C-a è ˆA):

```
- Lanciare minicom
- Digitare AT** : vedere "Download initiated .."
- C-a J
- cat FLASH.S37 > /dev/modem
- fg : vedere "Download flash code .."
- C-a J
- cat 283P1722.S37 > /dev/modem
- fg : vedere "Device successfully programmed"
```

18 Altre fonti di informazione

18.1 Varie

- pagine di manuale per: `agetty(8)`, `getty(1m)`, `gettydefs(5)`, `init(1)`, `isapnp(8)`, `login(1)`, `mgetty(8)`, `setserial(8)`
- Il vostro manuale del modem (se esiste). Alcuni modem vengono rilasciati senza manuale.
- *Serial Suite* `ftp://scicom.alphacdc.com/pub/linux` di Vern Hoxie è una collezione di trucchi sulla cura e l'alimentazione delle porte seriali in Linux più alcuni semplici programmi.
- La Linux serial mailing list. Per essere inclusi, inviate una email a *majordomo@vger.rutgers.edu* , con "`subscribe linux-serial`" nel corpo del messaggio. Se inviate '"`help`" nel corpo del messaggio, riceverete un messaggio di aiuto. Il server gestisce anche molte altre liste di Linux. Inviate il comando "`lists`" per un elenco delle mailing list.

18.2 Libri

Non stato capace di trovare un buon libro aggiornato sui modem.

- The Complete Modem Reference by Gilbert Held, 1997. Contiene troppe informazioni che riguardano argomenti superati. Maggiori informazioni aggiornate possono essere trovate su Internet.
- Modems For Dummies by Tina Rathbone, 1996. (Non l'ho mai visto.)
- Ultimate Modem Handbook di Cass R. Lewart, 1998.

18.3 HOWTO

- Cable-Modem mini-howto

- ISDN Howto (non è un Howto di LDP) `http://www.suse.de/Support/sdb_e/isdn.html` : drivers per ISDN Modems. La maggior parte delle informazioni è in tedesco
- Linux-Modem-Sharing mini-howto. Computer in rete condividono un singolo modem che chiamare l'esterno (tipo la condivisione di una stampante).
- Modems-HOWTO: In Francese (non usato nel creare questo Modem-HOWTO)
- NET-3-4-HOWTO: tutto sul networking, incluso SLIP, CSLIP, e PPP
- PPP-HOWTO: aiuto sul PPP incluso il set-up del modem.
- Serial-HOWTO ha informazioni sulle schede seriali multiporta usate sia per terminali che per i banchi di modem. Tratta la porta seriale più dettagliatamente di questo HOWTO.
- Serial-Programming-HOWTO: per certi aspetti della programmazione delle porte seriali
- Text-Terminal-HOWTO: (incluso la connessione con modem)
- UUCP-HOWTO: per informazioni sull'impostazione di UUCP

18.4 newsgroup Usenet

- comp.os.linux.answers: FAQ, How-To, README, ecc. su Linux.
- comp.os.linux.hardware: Compatibilità hardware con il sistema operativo Linux.
- comp.os.linux.setup: Installazione di Linux e amministrazione del sistema.
- comp.dcom.modems: Modem per tutti i sistemi operativi.

18.5 Siti Web

- Elenco di modem che funzionano/non funzionano sotto Linux
  `http://www.o2.net/~gromitkc/winmodem.html`
- Comandi AT per modem Hayes:
  Technical Reference for Hayes (tm) Modem Users `http://www.hayes.com/TechSupport/techref/`
- *Rockwell-based modem commands* `http://www.rss.rockwell.com/techinfo/`
- Modem FAQ:

  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`
- Molte informazioni sui modem a 56k *56k Modem = v.Unreliable* `http://808hi.com/56k/`
- *Links to modem manufacturers* `http://www.56k.com/links/Modem_Manufacturers/`
- *Identifying modems by FCC ID* `http://www.sbsdirect.com/fccenter.html`

19 Appendice A: Come funzionano i Modem analogici (tecnica) (non finita)

19.1 La modulazione nei dettagli

19.1.1 Introduzione alla modulazione

Questa parte descrive i metodi di modulazione usati per i modem convenzionali. Non tratta i metodi di alta velocità (modulus conversion), talvolta usati dai 19.1.6 (I modem a 56k (v.90)). Ma anche i modem a 56k usano i metodi di modulazione qui descritti.

La modulazione è la conversione di un segnale digitale rappresentato da valori binary (0 o 1) in un segnale analogico che ricorda un'onda sinusoidale. Il segnale modulato consiste in un segnale di un onda sinusoidale pura portante che è modificata per veicolare informazioni. Un'onda portante sinusoidale pura, non cambiando in frequenza e voltaggio, non genera un flusso di informazioni (ad eccezione del fatto che sia presente una portante). Per fare convogliare le informazioni modifichiamo (o moduliamo) questa portante. Ci sono 3 tipi principali di modulazione: frequenza, ampiezza e fase. Verranno di seguito spiegate.

19.1.2 Modulazione di frequenza

Il più semplice metodo di modulazione è la modulazione di frequenza. La frequenza è misurata in cicli per secondo (di un'onda sinusoidale). È il conto del numero di volte in cui la forma dell'onda sinusoidale ripete se stessa in un secondo. È la stessa cosa del numero di volte che raggiunge il valore più alto in un secondo. La parola Hertz (abbreviato Hz) viene usato per intendere cicli per secondo.

Un semplice esempio di modulazione di frequenza è dove una frequenza significa uno 0 binario ed un'altra significa 1. Ad esempio, per alcuni modem obsoleti da 300 baud, 1070 Hz significa uno 0 binario mentre 1270 Hz significa un 1 binario. Questo viene chiamato Frequency shift keying. Invece di solo due possibili frequenze, se ne potrebbero usare di più per consentire la trasmissione di più informazioni. Se abbiamo 4 differenti frequenze (che chiameremo A, B, C, e D), allora ognuna di queste potrebbe supportare un paio di bit. Ad esempio per inviare 00 si dovrebbe usare la frequenza A. Per inviare un 01, usiamo la frequenza B, per 10 usiamo C, per 11 usiamo D. In questo modo, usando 8 frequenze differenti potremo inviare 3 bit per ogni cambiamento di frequenza. Ogni volta che raddoppiamo il numero delle frequenze possibili, incrementiamo il numero di bit che possiamo rappresentare di 1.

19.1.3 Modulazione di ampiezza

Una volta capito l'esempio della modulazione di frequenza sopra indicato, incluso le possibilità di rappresentare alcuni bit tramite un singolo spostamento nella frequenza, è facile capire sia la modulazione di ampiezza e la modulazione di fase. Per modulazione di ampiezza basta cambiare l'altezza (voltaggio) dell'onda sinusoidale equivalente per cambiare la frequenza dell'onda sinusoidale. Per un caso semplice ci potrebbero essere solo 2 livelli di ampiezza consentiti, una rappresentato da un bit 0 e l'altro rappresentato da un bit 1. Come spiegato per il caso della modulazione di frequenza, l'avere diverse possibili ampiezze risulterà in maggiori informazioni che si trasmettono per cambiamento in ampiezza.

19.1.4 Modulazione di fase

Per cambiare la modulazione di fase di un onda sinusoidale in un certo perido di tempo, dobbiamo smettere di inviare questa vecchia onda sinusoidale ed immediatamente iniziare ad inviare una nuova onda sinusoidale della stessa frequenza ed ampiezza. Se iniziamo ad inviare la nuova onda sinusoidale allo stesso livello di voltaggio (e con la stessa pendenza) presente quando abbiamo interrotto l'invio della vecchia onda sinusoidale, non ci sarà nessun cambio di fase (e nessun cambio identificabile in assoluto). Ma supponiamo di avere iniziato la nuova onda sinusoidale ad un differente punto della curva dell'onda sinusoidale. Si dovrebbe probabilmente verificare un improvviso sbalzo di voltaggio al punto temporale dove la vecchia onda sinusoidale si è fermata ed è iniziata la nuova. Questo è uno spostamento di fase ed è misurato in gradi (deg.) Uno spostamento di fase di deg 0 (o di deg 360) significa nessun cambiamento in assoluto, mentre uno spostamento di fase di 180 inverte il voltaggio (e pendenza) dell'onda sinusoidale. Rendendolo in altri termini, uno spostamento di fase di 180 deg semplicemente salta in avanti di un mezzo periodo (180 deg) al punto di transizione. Naturalmente si potrebbe saltare di, diciamo, 90 deg o 135 deg. Così come nell'esempio della modulazione di frequenza, maggiori sono i possibili spostamenti di fase, maggiori bit possono essere rappresentati da un singolo spostamento di fase.

19.1.5 Modulazione combinata

Invece di selezionare semplicemente sia la frequenza, l'ampiezza o la modulazione di fase, possiamo scegliere di combinare i metodi di modulazione. Supponiamo di avere 256 frequenze possibili, quindi possiamo trasmettere un byte (8 bit) per ogni spostamento di frequenza (visto che 2 elevato alla 8 equivale a 256). Supponiamo anche di avere altre 256 differenti ampiezze così che ogni spostamento in ampiezza rappresenti un byte. Supponiamo anche che ci siano 256 spostamenti di fase possibili. Poi in un certo momento vogliamo fare uno spostamento in tutte e 3: frequenza, ampiezza e fase. Questo significherebbe spedire 3 byte per ognuna di queste transazioni.

Nessun metodo di modulazione in uso oggi fa veramente questo. Non è pratico a causa del tempo relativamente lungo che occorrerebbe per rilevare tutti i 3 tipi di cambiamento. Il problema principale è che frequenti cambi di fase possono far sembrare che sia accaduto un cambio in frequenza laddove in realtà non è successo.

Per evitare questi problemi si potrebbe cambiare simultaneamente solo la fase e l'ampiezza (senza nessun cambio di frequenza). Questa viene chiamata modulazione di fase-ampiezza (qualche volta chiamata anche quadrature amplitude modulation = QAM). Questo metodo è usato per le comuni velocità dei modem di 14,4k, 28.8k, e 33.6k. Il solo caso significativo dove questo metodo di modulazione non viene oggi usato è per i modem a 56k. Ma anche i modem a 56k usano esclusivamente QAM (modulazione di fase-ampiezza) nella direzione dal vostro PC in uscita verso la linea telefonica. Qualche volta anche nell'altra direzione si ritorna alla modulazione QAM quando le condizioni della linea non sono sufficientemente buone. Quindi QAM (modulazione di fase-ampiezza) rimane ancora il metodo più largamente usato nelle ordinarie linee telefoniche.

19.1.6 I Modem a 56k (v.90)

Il metodo di modulazione usato sopra i 33.6k è completamente diverso dalla comune modulazione fase-ampiezza. Visto che le chiamate telefoniche ordinarie sono convertite in segnali digitali nelle centraline locali della compagnia telefonica, la velocità più elevata con la quale si possono spedire dati digitali tramite una ordinaria chiamata telefonica è la stessa di quella che la compagnia telefonica usa lungo la sua porzione digitale della trasmissione della chiamata telefonica. Qual è questa velocità? Beh, è vicina ai 64Kbps. Dovrebbe essere 64k ma talvolta alcuni bit sono rubati per scopi di segnalazione. Ma se la compagnia telefonica sa che il collegamento non è per la voce, i bit potrebbero non essere rubati. Verrà presentato il caso dei 64k, quindi verrà spiegato perché la velocità reale è più bassa (56k o meno – in genere significativamente meno).

Quindi 64k è la maggiore velocità possibile per una chiamata telefonica ordinaria usando la porzione digitale del circuito che era stata concepita per inviare le codifiche digitali della voce umana. Per potere usare 64k, il modem deve sapere esattamente come la compagnia telefonica faccia la sua codifica digitale del segnale analogico. Questo compito è troppo complicato se entrambi gli estremi di una chiamata telefonica hanno un'interfaccia analogica alla compagnia telefonica. Ma se da una parte si ha una interfaccia digitale, allora è possibile (almeno in una direzione). Quindi se il vostro ISP ha una interfaccia digitale con la compagnia telefonica, l'ISP può inviare un certo segnale digitale attraverso la linea telefonica verso il vostro PC. Il segnale digitale dall'ISP viene convertito in analogico alla centralina telefonica vicina alla locazione fisica del vostro PC (forse vicino a casa vostra). Poi è compito del vostro modem cercare di capire esattamente che cos'era quel segnale digitale. Se può fare questo, allora la trasmissione a 64k (la velocità del segnale digitale della compagnia telefonica) è possibile in questa direzione.

Che metodo usa la compagnia telefonica per decodificare in digitale i segnali analogici? Usa il metodo di campionare l'ampiezza del segnale analogico alla velocità di 8000 campioni per secondo. Ogni ampiezza campione è codificata come un byte a 8 bit (tipo ASCII). (Notare: 8 x 8000 = 56k). Questa viene chiamata Pulse Code Modulation = PCM. Questi byte sono poi inviati digitalmente sui circuiti digitali della compagnia telefonica dove diverse chiamate condividono un singolo circuito, usando uno schema di time-sharing chiamato time division multiplexing. Poi finalmente nella locale centralina telefonica vicina a casa vostra, il segnale digitale viene demultiplexato risultando nello stesso segnale digitale così come originariamente creato da PCM. Questo segnale viene poi riconvertito in analogico ed inviato a casa vostra. Ogni byte da 8-bit crea una certa ampiezza del segnale analogico. Il vostro modem deve determinare cosa era quel byte PCM a 8 bit basandosi sulla ampiezza analogica che rileva.

Questa è (in un certo senso) una demodulazione di ampiezza ma non realmente. Non si tratta di demodulazione di ampiezza perché non vi è portante. In verità, viene chiamata conversione di modulo (modulus conversion) che è l'inverso di PCM. Per determinare il segnale digitale che la compagnia telefonica ha usato per creare il segnale analogico, il modem deve campionare questo segnale di ampiezza analogica esattamente agli stessi punti temporali che la compagnia telefonica ha usato quando ha creato il segnale analogico. Per fare questo viene un generato un segnale temporizzato dal segnale residuo di 4k Hz sulla linea telefonica analogica. La creazione dei campioni di ampiezza che escono dalla vostra casa/ufficio ad 8k campioni/sec circa creano un segnale di 4k. Supponete che ogni altra ampiezza fosse di polarità opposta. Allora dovrebbe essere stata creata un'onda simile alla sinusoidale di 4k Hz. Ogni ampiezza è in un certo senso un simbolo ad 8 bit e quando si campionano le ampiezze è conosciuto come symbol timing.

Ora la codifica di queste ampiezze in PCM non è lineare. A basse ampiezze un incremento di 1 nel byte PCM rappresenta un incremento molto più piccolo nell'ampiezza del segnale analogico rispetto a quella che sarebbe se l'ampiezza che viene campionata fosse più alta. Quindi per basse ampiezze è difficile distinguere tra valori di byte adiacenti. Per facilitare le cose, alcuni codici PCM rappresentanti ampiezze molto basse non sono usati. Questo dà un delta più ampio tra le possibili ampiezze e fa sì che il modem le riconosca correttamente con più facilità. Quindi la metà dei livelli di ampiezza non sono usati dal v.90. Questo è equivalente ad ogni simbolo (livello di ampiezza consentito) che rappresenta 7 bit invece di 8. Ecco da dove proviene il 56k: 7 bit/simbolo x 8k simboli/secondo = 56k bps. Naturalmente ogni simbolo è in realtà generato da 8 bit ma solo 128 byte dei possibili 256 sono effettivamente usati. C'è una tavola di codici che mappa questo 128 byte a 8 bit con quelli a 7 bit.

Ma è un poco più complicato di questo. Se le condizioni della linea non rasentano la perfezione, allora sono usati anche meno livelli (simboli), risultando in velocità sotto i 56k. Negli Stati Uniti, anche a causa di regole governative che proibiscono gli alti voltaggi sulle linee telefoniche, certi alti livelli di ampiezza non posssono essere usati, risultando quindi in soli 53.3k circa al massimo per i modem a 56k.

Notate che la parte digitale della rete telefonica è bi-direzionale. Questi due circuiti sono usati per una chiamata telefonica, uno in ciascuna direzione. Il segnale a 56k viene usato solamente in una di queste direzioni: dal vostro ISP al vostro PC. Nell'altra direzione dalla casa/ufficio verso l'ISP è usato lo schema convenzionale di modulazione fase-ampiezza con un massimo di 36.6k (e non 53.3K). Inoltre grazie a sofisticati metodi di cancellazione (non spiegati qui) consente di inviare simultaneamente in entrambe le direzioni.

19.2 Full duplex (bidirezionalità) in un circuito

I modem moderni sono capaci di ricevere ed inviare segnali contemporaneamente. Si potrebbe chiamare questa capacità bidirezionale o full duplex. Una volta la cosa era fatta usando una frequenza per inviare ed un'altra per ricevere. Oggi, la stessa frequenza viene usata sia per la trasmissione che per la ricezione. Come funzioni la cosa non è facile da capire.

La maggior parte delle linee principali del sistema telefonico sono digitali con due canali in uso quando si fa una chiamata telefonica. Quello che dite va su un canale digitale e quello che dice l'altra persona va sull'altro (inverso) canale digitale. Sfortunatamente, la porzione del sistema telefonico che va alle case (e molti uffici) non è digitale ma è un singolo canale analogico. Se entrambi i modem fossero direttamente connessi con la parte digitale del sistema telefonico allora la comunicazione bidirezionale (inviare e ricevere allo stesso tempo) non sarebbe un problema visto che sarebbe disponibili due canali.

Ma le porzioni finali del percorso del segnale passano su un solo circuito. Come è possibile che ci sia comunicazione simultanea nei due sensi? Funziona grossomodo così. Supponete che il vostro modem stia ricevendo un segnale dall'altro modem e non stia trasmettendo. In questo caso non c'è problema. Ma se si iniziasse a trasmettere (mentre l'altro segnale in ricezione sta ancora scorrendo nel modem) il segnale in ricezione sarebbe soffocato. Se il segnale in trasmissione fosse un'onda di voltaggio solida applicata alla fine della linea non ci sarebbe alcuna possibilità che un qualsiasi segnale in ricezione possa essere presente a quel punto.

Ma il trasmettitore ha una impedenza interna ed segnale in trasmissione applicato alla fine della linea non è solido (o abbastanza forte) da eliminare completamente il segnale in ricezione che arriva dall'altro capo. Quindi mentre il voltaggio alla fine della linea è per la maggior parte il più forte segnale in trasmissione, una piccola parte di esso è il segnale in ricezione richiesto. Tutto quello che serve è filtrare il segnale in trasmissione, più forte, e quello che rimane sarà il segnale dall'altro capo che vogliano. Per fare questo, basta prendere il puro segnale in trasmissione direttamente dal trasmettitore (prima che sia applicato sulla linea), amplificarlo per un determinato ammontare, quindi sottrarlo dal segnale totale presente alla fine della linea. Facendo questo nei circuiti di ricezione rimane un segnale che per la maggior parte proviene dall'altra parte della linea.

19.3 Eliminazione dell'eco

Un segnale che viaggia attraverso una linea in una direzione potrebbe incontrare cambiamenti nella linea che farà si che parte del segnale venga riecheggiato all'indietro nella direzione opposta. Visto che viene usato lo stesso circuito per il flusso bidirezionale di dati detti eco genereranno ricezioni sporche. Un modo di migliorare le cose è inviare segnali di prova ogni tanto per determinare le caratteristiche di eco della linea. Questo consentirà di predirre gli eco che potrebbero essere generati ad ogni segnale. Quindi il metodo di predizione è usato per predirre quali eco il segnale in tramissione provocherà. Poi questo eco presunto del segnale viene sottratto dal segnale ricevuto. Questo elimina gli eco.

20 Appendice B: Digital Modem Signal Processing (non fatta)

21 Appendice C: baud contro bps

21.1 Un semplice esempio

baud e bps sono forse due dei termini più abusati nel campo dei computer e delle telecomunicazioni. Molte persone usano questi due termini indifferentemente, quando in realtà essi sono diversi! bps è semplicemente il numero dei bit trasmessi per secondo. Il baud rate è la misura di quante volte per secondo un segnale cambia (o potrebbe cambiare). Per una comune porta seriale il bit 1 è -12 volt e il bit 0 è +12 v (volt). Se 38.400 bps sono una sequenza di 010101 ... dovrebbero essere anche 38.400 baud visto che il voltaggio cambia avanti e indietro da positivo a positivo a negativo ... e ci sono 38400 cambiamenti per secondo. Per un'altra sequenza, diciamo 111000111... ci saranno minori cambiamenti di voltaggio visto che per i tre 1 in sequenza il voltaggio rimane a -12 volt, eppure diciamo che abbiamo ancora 38.400 baud visto che esiste la possibilità che il numero di cambiamenti per secondo raggiunga quel valore.

Vista in altro modo, mettiamo un immaginario marcatore che separi ogni bit (anche se il voltaggio potrebbe non cambiare). 38.400 baud quindi significa 38.400 marcature per secondo. La marcatura scatta all'istante del cambiamento permesso e sono in realtà marcati da un segnale di un clock sincronizzato generato dall'hardware ma non inviato attraverso il cavo esterno.

Supponiamo che un cambiamento possa avere più dei 2 possibili risultati dell'esempio precedente (di +- 12 volt). Supponiamo che abbia 4 possibili risultati, ognuno rappresentato da un diverso livello di voltaggio. Ogni livello potrebbe rappresentare un paio di bit (come 01). Per esempio, -12 v potrebbe essere 00, -6v 01, +6b 10 e +12v 11. Ecco che la velocità di bit è doppia rispetto alla velocità di baud. Ad esempio, 3000 cambiamenti per secondo genereranno 2 bit per ogni cambiamento risultanti in 6000 bit per secondo (bps). In altre parole 3000 baud equivalgono a 6000 bps.

21.2 Esempi reali

L'esempio di cui sopra è oltremodo semplice. Esempi reali sono molto più complicati ma si basano sullo stesso concetto. Questo dimostra come un modem che va a 2400 baud possa inviare 14400 bps (o più). Il modem acquisisce una velocità di bps maggiore di quella di baud, codificando molti bit per ogni cambio di segnale (o transizione). Quindi, quando 2 o più bit sono codificati per baud, la velocità in bps supera quella in baud. Se la vostra connessione modem-a-modem è di 14400 bps, si trasmetteranno 6 bit per segnale (o simbolo) di transizione a 2400 baud. Una velocità di 28000 bps è ottenuta da 3200 baud a 9 bit per baud. Quando la gente usa in modo equivoco il termine baud, probabilmente intende la velocità del modem (tipo 33.6k)

La velocità di bps del normali modem erano precedentemente 50, 75, 110, 300, 1200, 2400, 9600. Esse erano anche le velocità bps nei cavi da porta seriale a modem. Oggi le velocità (massime) bps da modem a modem sono di 14.4K, 28.8K, 33.6K, and 56K, ma le velocità dalla porta seriale al modem non sono le stesse ma sono: 19.2K, 38.4K, 57.6K, 115.2K. Usando modem con una compressione V.42bis (massima compressione 4:1), le velocità fino a 115,2K bps sono possibili per modem a 33.6K (230.4K possibile per i modem a 56K).

Eccettuati i modem a 56K, la maggior parte dei modem girano a 2400, 3000 o 3200 baud. Anche i modem a 56k usano questi baud per trasmettere e talvolta precipitano a questi in ricezione. A causa delle limitazioni di ampiezza di banda nelle linee telefoniche voice-grade, velocità in baud superiori a 2400 sono difficili da raggiungere e solo lavorando in buone condizioni di qualità della linea telefoniche.

Come inizia questa confusione tra bps e baud? Bene, torniamo all'epoca nella quale i vecchi modem a bassa velocità erano considerati modem ad alta velocità, la velocità bps in effetti era uguale alla velocità baud. Un bit era codificato per ogni cambiamento di fase. La gente usava bps e baud intercambiabilmente, visto che rappresentavano lo stesso valore. Ad esempio, un modem a 300 bps aveva pure una velocità in baud di 300. Tutto questo cambia quando i modem più veloci entrarono in circolazione e la velocità in bit superò la velocità in baud. "baud" deriva da Emile Baudot, l'inventore della stampante telegrafica asincrona. Un modo per risolvere questo problema consiste nell'usare il termine symbol rate invece di baud, quindi evitando di usare il termine baud. Comunque quando si parla di velocità tra il modem e la porta seriale (velocità DTE) baud e symbol rate sono uguali. Ed anche velocità è improprio visto che in realtà vuol dire velocità di flusso.

22 Appendice D: Connessione Terminal Server

Questa sezione è adattata da Text-Terminal-HOWTO. Un server di terminale è qualcosa come un interruttore intelligente che può connettere molti modem (o terminali) ad uno o più computer. Non è un interruttore meccanico così esso può modificare le velocità ed i protocolli del flusso di dati che gli passano attraverso. Diverse compagnie costruiscono terminal server: Xyplex, Cisco, 3Com, Computone, Livingston, ecc. Ci sono di diversi tipi e capacità. È necessario un altro HOWTO per confrontarli e descriverli (inclusa la possibilità di crearsi il proprio terminal server con un PC Linux). La maggior parte sono usati per connettere modem piuttosto che connettere direttamente terminali.

Un uso per essi è quello di connettere molti modem (o terminali) ad una rete ad alta velocità la quale si connette a dei computer host. Naturalmente il terminal server deve avere la potenza di calcolo ed il software per far girare protocolli di rete come se fosse per certi versi un computer. Il terminal server potrebbe interagire con l'utente e chiedergli quale computer connettere a quale altro, ecc. oppure potrebbe connettere senza chiedere. Si potrebbe qualche volta inviare dei jobs ad una stampante attraverso il terminal server.

Oggi un PC ha sufficiente potenza di calcolo per agire come un terminal server eccetto che ogni porta seriale dovrebbe avere il proprio interrupt hardware. I PC hanno solo pochi interrupt di riserva per questo scopo e visto che sono hard-wired non potete creare più di tanto tramite software. Una soluzione è usare un'avanzata scheda seriale multiporta che ha il suo proprio sistema di interrupt (sui modelli a basso costo, condividono uno degli interrupt del PC tra diverse porte). Vedere Serial-HOWTO per maggiori informazioni. Se un PC ha Linux e fa girare getty su molte porte seriali, si potrebbe pensare ad esso come ad un terminal server. Esso è in effetti un terminal server se è collegato ad altri PC attraverso una rete e se il suo compito è principalmente quello di far passare dati e gestire gli interrutp della porta seriale ogni 14 (circa) byte. Un software chiamato radius è talvolta usato.

Oggi i veri terminal server servono più che da semplici terminali. Essi servono dei PC che emulano terminali e sono talvolta collegati ad una batteria di modem connessi alle linee telefoniche. Alcuni hanno anche i modem costruiti al loro interno. Se un terminale (o un PC che lo emula) è connesso direttamente ad un modem, il modem dall'altro capo della linea potrebbe essere connesso ad un terminal server. In alcuni casi il terminal server per default si aspetta che i chiamanti usino pacchetti PPP, qualcosa che i terminali testuali reali non generano.

23 Appendice E: Altri tipi di Modem

Questo HOWTO attualmente tratta dei comuni tipi di modem usati per connettere un PC ad una linea telefonica analogica ordinaria. Ci sono diversi altri tipi di modem, inclusi quei dispositivi chiamati modem ma che non sono veramente modem.

23.1 Modem Digitale-a-Digitale

La definizione standard di un modem è talvolta allargata fino ad includere i modem digitali. Oggi servizi diretti digitali sono forniti in diverse case ed uffici così che un computer possa inviare all'esterno segnali digitali direttamente (beh, quasi) alla linea telefonica. Ma un dispositivo è comunque richiesto per convertire il segnale digitale del computer in un tipo permesso sui circuiti telefonici e questo dispositivo è talvolta chiamato modem. Questo HOWTO non tratta detti modem ma alcuni collegamenti a documenti che lo fanno potrebbero essere trovati all'inizio di questo HOWTO. Le successive 3 sezioni: ISDN, DSL e 56k, trattano dei modem digitale-a-digitale.

23.2 Modem ISDN

Il modem è in realtà un Terminal Adapter (TA). Un pacchetto Debian chiamato isdnutils è disponibile. C'è un ISDN Howto in tedesco con traduzione in inglese:

`http://www.suse.de/Support/sdb_e/isdn.html` . Si trova nella distribuzione SuSE di Linux e presumibilmente riguarda dei driver disponibili in quella distribuzione. C'è anche un pacchetto isdn4Linux ed un newsgroup: de.alt.comm.isdn4linux. Molti dei messaggi sono in tedesco. Potreste provare ad uasre un motore di ricerca (tipo DejaNews) per cercare isdn4linux.

23.3 Digital Subscriber Line (DSL)

DSL usa l'esistente doppino telefonico dalla vostra casa (ecc) alla locale centralina telefonica. Questo può essere usato se la vostra linea telefonica può accettare velocità superiori a quelle che un modem ordinario (diciamo 56k) invia. Rimpiazza il convertitore analogico-digitale nella locale centralina telefonica con un convertitore che può accettare un flusso di dati molto più veloce (in un formato differente naturalmente). Il dispositivo che converte i segnali digitali dal vostro computer al segnale usato per rappresentare i dati digitale sulla linea telefonica locale è anch'esso chiamato modem.

23.4 Modem digitali a 56k

Perché un qualsiasi modem a 56k possa lavorare effettivamente a 56k nella vostra casa o ufficio, l'altro capo deve essere connesso direttamente al sistema digitale della compagnia telefonica. Quindi gli ISP dall'altro capo della linea devono procurarsi speciali modem digitali che forniscano ai propri clienti il servizio a 56k. C'è di più visto che batterie di molti modem sono multiconnessi ad un cavo telefonico ad alta capacità che trasporta un gran numero di chiamate telefoniche simultaneamente (tipo T1, E1, ISDN PRI, o linee migliori). Occorre quindi un concentratore o remote access server (server di accesso remoto). Erano in genere costituiti da unità a sè stanti (come i PC, ma costano molto di più ed hanno un sistema operativo proprietario). Ora ci sono alcune schede che si possono inserire in un bus PCI di un PC per svolgere questo compito.

23.5 Modem a linea affittata

Questi sono analogici e non digitali. Questi modem speciali sono usati su linee affittate dalla compagnia telefonica o talvolta da un semplice connessione tramite un lungo cavo diretto. I modem comuni che vanno su una linea telefonica in genere non funzionano su questo tipo di linee. Un linea telefonica normale ha circa 40-50 volt (conosciuta come la batteria) su di essa quando non è in uso ed i modem convenzionali usano questo voltaggio per la trasmissione. Oltretutto, la compagnia telefonica ha segnali speciali che indicano uno squillo, linea occupata, ecc. I modem convenzionali si attendono e rispondono a questi segnali. Connettere due modem normali tramite un lungo cavo non genererà segnali telefonici lungo il cavo, quindi i modem non funzioneranno.

Una tipica linea affittata usava due coppie di cavi (uno per ciascuna direzione) usando una modulazione V.29 a 9600 baud. Alcune marche di modem per linee affittate sono incompatibili con altre.

24 Appendice F: Pixel (punti) dei fax

```
Carta A4:    216mm (orizzontale)  * 297mm (verticale)
modo normale       8punti/mm      * 3.85punti/mm
modo fine                         * 7.7punti/mm
modo extra fine                   *15.4punti/mm
```

Ogni punto può essere sia bianco che nero e quindi rappresenta 1 bit. Un foglio A4 che usa il modo fine è (216*8) * (297*7.7) = circa 4 milioni di punti. Con un rapporto di compressione di 8:1 occorrono circa 50 secondi a 9600bps per la trasmissione.

FINE DEL Modem-HOWTO