Linux Modem-HOWTO

David S. Lawyer (dave@lafn.org) und Claus Poltermann (Claus.Poltermann@gmx.de) v0.04, August 2001

Dieses HOWTO bietet Hilfe bei Auswahl, Anschluss, Konfiguration von Modems, erläutert Grundlagen und gibt Tipps zur Fehlerbehebung. Für Hinweise zum Einsatz von seriellen Multiport-Karten sei hier auf das Serial HOWTO verwiesen.

Inhaltsverzeichnis

1 Einleitung

Dieses HOWTO enthält viele Informationen aus dem *Serial HOWTO* von Greg Hankins und behandelt externe und interne Modems. Bei den internen Modems liegt der Schwerpunkt auf der Behandlung von ISA-Bus-Modems, obwohl das meiste auch für PCI-Bus-Modems gelten sollte. Für PCMCIA-Modems siehe das (englische) *PCMCIA HOWTO: »PCMCIA serial and modem devices«*. Dieses HOWTO behandelt weder

PPP (Point-To-Point Protocol, es wird verwendet, um mit Hilfe eines Modems eine Verbindung zum Internet herzustellen) noch Kommunikationsprogramme. Es erklärt aber, wie Kommunikationsprogramme verwendet werden, um zu testen, ob Ihr Modem korrekt funktioniert und eine Verbindung aufgebaut werden kann. Wenn Sie ein Modem verwenden möchten, um einen Zugang zum Internet zu bekommen, dann ist es notwendig, PPP einzurichten. Es gibt eine Menge Dokumentation über PPP (siehe z.B. im Verzeichnis `/usr/doc/ppp`; es gibt auch ein *PPP HOWTO*).

1.1 Copyright

1.2 Hinweise zur deutschen Übersetzung

Dieses HOWTO basiert auf dem englischen *Modem HOWTO* von David S. Lawyer. Die Übersetzung habe ich nicht immer wortwörtlich durchgeführt, um dem Text einen deutschsprachigen Charakter zu verleihen. An manchen Stellen habe ich den Text gegenüber dem Original auch erweitert, wenn ich der Ansicht war, dass einige zusätzliche erklärende Worte zu einem besseren Verständnis beitragen könnten. Hinweise aller Art und Fragen richten Sie bitte daher zuerst an mich (`Claus.Poltermann@gmx.de`); falls das Problem auch das englische Original betrifft, werde ich die Hinweise weiterleiten. Die meisten der in diesem Text enthaltenen Rechtschreibfehler wurden allerdings durch die Rechtschreibreform legalisiert. Entgegen dem im Deutschen üblichen unpersönlichen man-Stil findet sich der Leser häufig direkt, wenn auch höflich distanziert, mit »Sie« angesprochen - das schulterklopfende »Du« erschien mir zu persönlich für die Übersetzung des englischen »you«. Wider besseres Wissen habe ich für den Plural von Modem »Modems« statt »Modeme« verwendet, weil sich diese Bezeichnung in der Praxis durchgesetzt hat. Und schließlich mögen mir Benutzerinnen von Linux verzeihen, wenn sie in diesem Text als »Benutzer« oder »Anwender« angesprochen werden.

1.3 Für zukünftige Versionen ist Ihre Mitarbeit erwünscht

Wenn Sie in diesem Text sachliche oder logische Fehler, Rechtschreibfehler oder Unklarheiten feststellen, oder wenn Sie eine andere Ansicht haben als hier dargestellt, dann lassen Sie es mich bitte wissen. Bitte vergewissern Sie sich aber vorher, dass Sie die neueste Version haben. Bitte senden Sie mir auch jede andere Information, von der Sie glauben, dass Sie in diesem Zusammenhang behandelt werden soll.

1.4 Neue Versionen dieses HOWTOs

Neue Versionen dieses Modem-HOWTOs werden alle paar Monate veröffentlicht. Vielleicht finden Sie Hinweise zu Ihrem Problem in der aktuellsten Version. Sie ist als Online-Version oder zum Herunterladen auf der Homepage des Deutschen Linux HOWTO Projekts

```
http://www.tu-harburg.de/dlhp/
```

erhältlich.

1.5 Was ist ein Modem?

Ein Modem ist ein Gerät, welches es ermöglicht, digitale Signale über gewöhnliche Telefonleitungen zu senden, die nicht für die Übertragung digitaler Signale ausgelegt sind. Solche Leitungen werden im folgenden als »analog« bezeichnet. Analoge Telefonleitungen sind geeignet, um vor allem die zur sprachlichen Verständigung notwendigen niedrigen Frequenzen (bis ca. 3 kHz) zu übertragen. Eine »digitale« Telefonleitung ist dagegen technisch auf die bei der Übertragung von (hochfrequenten) digitalen Signalen auftretenden Effekte abgestimmt. Wenn alle Telefonleitungen digital wären, dann bräuchte man kein Modem. Es erlaubt dem Computer sich mit dem Rest der Welt zu verbinden und mit ihr zu kommunizieren. Wenn man ein Modem verwendet, dann nutzt man normalerweise auch ein Kommunikationsprogramm oder einen Web Browser, um über eine Telefonleitung eine Wählverbindung aufzubauen und Daten zu übertragen. Ein fortgeschrittener Modem-Benutzer kann sein System so einrichten, dass andere sein Modem (natürlich wiederum mit Hilfe eines Modems) anrufen und den angeschlossenen Computer verwenden können. Dies wird als »Einwahl« bezeichnet (engl. »Dial-in«).

Es gibt zwei Grundtypen von Modems für einen PC: externe Modems und interne Modems. Das externe Modem steht auf Ihrem Schreibtisch ausserhalb des PC, während das interne Modem sich nicht sichtbar im Inneren des PC befindet. Das externe Modem wird mit einem Anschluss an der Rückseite des PCs verbunden, der als serieller Port (oder auch serielle Schnittstelle) bekannt ist. Das interne Modem ist eine Steckkarte, die im Computer installiert wird und über einen (unsichtbaren) eingebauten seriellen Port verfügt. Ein detaillerter Vergleich ist im Abschnitt 6.1 (Externe und interne Modems) zu finden. Wenn Sie sich also für ein internes Modem entscheiden, dann entscheiden Sie sich auch für einen dedizierten seriellen Port; das ist ein Port, der nur für dieses Modem verwendet werden kann und nicht für ein anderes Gerät, wie z.B. ein anderes Modem oder einen Drucker. Unter Linux heißen die seriellen Ports `ttyS0`, `ttyS1`, usw.; dies entspricht üblicherweise den unter DOS/Windows bekannten Schnittstellen COM1, COM2, usw.

Der serielle Port ist nicht zu verwechseln mit dem »Universal Serial Bus« USB. Der USB verwendet einen speziellen Anschluß und es werden vermehrt Peripheriegeräte wie z.B. Modems mit einer USB Schnittstelle angeboten. Im Abschnitt 2 (Grundwissen über Modems und serielle Ports) sind Modems und serielle Ports genauer beschrieben.

Viele Modems können auch Faxe versenden (Fax Modems). Der Abschnitt 13.2.3 (Fax) enthält eine Übersicht über Fax Software. »Voice Modems« können als Anrufbeantworter eingesetzt werden und sprachliche Nachrichten verarbeiten, siehe 10.6 (Sprachnachrichten).

1.6 Hinweise zur schnellen Installation

1.6.1 Installation eines externen Modems

Verbinden Sie das Modem mithilfe eines Modemkabels mit einem freien seriellen Anschluß am PC. Den Namen des seriellen Ports müssen Sie wissen: In den meisten Fällen entspricht COM1 `ttyS0` und COM2 `ttyS1`. Eventuell müssen Sie im BIOS Setup nachsehen. Außerdem muss das Modem noch mit Strom versorgt werden; dies geschieht meist über ein passendes Steckernetzteil. Weitere Hinweise finden Sie im Abschnitt 1.6.3 (Alle Modems).

1.6.2 Interne Modems (ISA Bus)

Wenn sich das Modem nur unter MS Windows betreiben läßt, haben Sie leider Pech und Sie können es unter Linux nicht benutzen.

Wenn Sie bereits zwei serielle Ports haben, konfigurieren Sie den internen Modemport als dritten seriellen Port (`ttyS2` entspricht COM3). Dazu benötigen Sie eine freie IRQ Nummer. Oft können Sie dafür IRQ5 verwenden, wenn Sie keinen zweiten parallelen Port oder keine Soundkarte haben. Dann setzten Sie die Jumper (das sind winzige Anschlussstifte auf der Platine, die mit einer kleinen Brücke verbunden werden oder kleine Schiebeschalter von der Größe eines ICs) auf dem internen Modem auf diesen IRQ und die I/O-Adresse auf 3E8 (`ttyS2`).

Dies ist oft leichter gesagt als getan. Wenn das Modem ein PnP (Plug and Play) Modem für den ISA-Bus ist, kann diese Einstellung vielleicht auch mit Hilfe des Programms `isapnp` aus den »isapnptools« durchgeführt werden. Weitere Hinweise dazu erhalten Sie mit

```
man isapnp
```

oder in den entsprechenden FAQs. Es gibt auch ein *Plug-and-Play HOWTO*. Bei einem PnP-BIOS können Sie im CMOS-Setup einstellen, dass Sie kein PnP-Betriebssystem haben; das BIOS kann dann passende Werte für IRQ- und I/O-Adresse in der Modemkarte einstellen. Falls Sie das Modem unter Windows 9x konfiguriert haben, habe ich einmal geglaubt, dass Windows so freundlich sein würde, die passende Konfiguration (IRQ- und I/O-Adresse) in den BIOS-Flash-Speicher zu schreiben, so dass sie vom BIOS verwendet werden kann, wenn Linux gestartet wird. Aber Windows verhält sich nicht immer so; probieren Sie dies nur als letzte Möglichkeit. Vielleicht gibt es auch einen Weg, die PnP-Funktionalität mithilfe von Windows-Software, die mit dem Modem geliefert wurde, zu deaktivieren.

Schließlich müssen Sie noch die Datei finden, von der aus `setserial` aufgerufen wird und etwa folgende Zeile hinzufügen:

```
setserial /dev/ttyS2 irq5
```

Weitere Informationen finden Sie im nächsten Abschnitt.

1.6.3 Alle Modems

Verbinden Sie das Modem mit der Telefonleitung. Starten Sie ein Terminalprogramm wie `minicom` und rufen Sie das Konfigurationsmenü für den seriellen Port auf. Weisen Sie ihm eine hohe Baudrate zu, die ein mehrfaches der Bitrate Ihres Modems beträgt; siehe auch 12.4 (Geschwindigkeitstabelle). Geben Sie den vollständigen Namen des seriellen Ports an wie z.B. `/dev/ttyS1`. Schalten Sie die Hardwareflusskontrolle ein (RTS/CTS). Sie müssen diese Einstellungen nun abspeichern und `minicom` beenden. Starten Sie `minicom` erneut, geben Sie `AT` ein, um zu überprüfen, ob Ihr Modem erkannt wird und mit OK antwortet. Rufen Sie dann das Nummernverzeichnis auf und wählen Sie eine Nummer.

2 Grundwissen über Modems und serielle Ports

Um ein Modem anzuschliessen und zu verwenden, brauchen Sie nicht unbedingt ein Grundwissen über Modems oder serielle Ports. Falls Fehler auftreten, ist es aber für die Problemlösung sehr hilfreich, wenn Sie die Grundlagen verstehen. Wenn Sie diesen Abschnitt gelesen haben und noch mehr über Modems wissen möchten, finden Sie weitere Informationen im Abschnitt 17 (Wie ein Modem technisch funktioniert) (dieser Abschnitt ist allerdings noch nicht fertiggestellt). Im *Serial HOWTO* finden Sie weitere Informationen über serielle Ports.

2.1 Ein Modem konvertiert digitale Signale in analoge (und umgekehrt)

Die meisten Telefon-Hauptleitungen sind bereits digital, aber die Kabel, die zu Ihrem Haus oder Ihrer Firma führen, sind normalerweise analog. Das bedeutet, dass sie eine Welle in Form einer Spannungsschwankung übertragen, die ein Abbild der Schallwelle ist, die von Ihrer Stimme erzeugt wird. Eine solche Spannungsschwankung heißt »analog«. Würde man sie auf einem Oszilloskop sichtbar machen, dann würde sie aussehen wie eine Sinuskurve mit wechselnder Frequenz und Amplitude. Ein digitales Signal hat dagegen einen rechteckigen Verlauf. Bsp: 3 V (Volt) könnte einer digitale Eins und 0 V könnte einer digitalen Null entsprechen. Bei den meisten seriellen Ports, die von einem externen Modem verwendet werden, entspricht +12 V einem 0-bit und -12 V einem 1-bit (einige verwenden auch + oder -5 V).

Um Daten von Ihrem Computer über eine Telefonleitung zu senden, nimmt das Modem die digitalen Signale vom Computer entgegen und wandelt sie in ein analoges Signal. Dies wird erreicht, indem es eine Spannung in Form einer analogen Sinuskurve erzeugt und diese dann »MOduliert«. Das Ergebnis repräsentiert noch immer digitale Daten, es könnte auch digitales Signal anstelle von analoges Signal heißen. Aber es sieht so ähnlich aus wie ein analoges Signal und beinahe jeder nennt es auch so. Am anderen Ende der Telefonleitung »DEModuliert« ein anderes Modem dieses Signal und das rein digitale Signal ist wiederhergestellt. Ein Modem ist daher ein »MOdulator-DEModulator«.

2.2 Was ist ein serieller Port?

2.2.1 Einführung in die serielle Datenübertragung

Nachdem sich zwischen dem Modem und dem Computer ein serieller Anschluß befindet, ist es notwendig, neben dem Modem auch die Arbeitsweise eines seriellen Ports zu verstehen. Der serielle Port ist ein I/O (Input/Output, Eingabe/Ausgabe) Gerät.

Die meisten PCs haben ein oder zwei serielle Ports. Jeder hat einen 9-poligen (manchmal auch einen 25-poligen) Anschlussstecker auf der Rückseite des Computers. Einer der Anschlussstifte ist für das Senden von Daten zuständig, ein anderer für den Empfang. Die anderen Pins sind entweder auf Masse gelegt oder für Kontrollaufgaben vorgesehen.

Der serielle Port ist mehr als nur ein einfacher Anschlussstecker. Er wandelt parallele Daten in serielle Daten und ändert ihre elektrische Darstellung. Innerhalb des Computers fließen die Datenbits parallel (und brauchen viele parallele Leitungen gleichzeitig). Ein serieller Datenstrom fließt über eine einzige Leitung (wie die Leitung, die mit dem Pin für das Senden oder Empfangen des Anschlusssteckers verbunden ist). Damit der serielle Port solch einen Datenstrom erzeugen kann, muß er die parallelen Daten (aus dem Inneren des Computers) in serielle Daten umwandeln, um sie zu senden. Für den Empfang gilt genau das umgekehrte Verhalten.

Der größte Teil der Elektronik eines seriellen Ports befindet sich auf einem Chip (oder in einem bestimmten Abschnitt eines Chips), der unter dem Namen UART bekannt ist. Im *Serial HOWTO* finden Sie weiter Informationen über UARTs.

Im restlichen Teil dieses Kapitels wird die Rolle des UART im Gesamtsystem vorgestellt.

2.2.2 Anschlussstifte und Anschlusskabel

Ältere PCs verwendeten 25-polige Anschlüsse, aber nur höchstens 9 Pins waren tatsächlich belegt, deshalb sind die meisten Anschlussstecker heutiger PCs nur 9-polig. Jeder der neun Anschlussstifte (Pins) ist mit einer Leitung verbunden. Neben den beiden Leitungen, die für das Senden und Empfangen der Daten verwendet werden, ist ein weiterer Pin bzw. eine weitere Leitung mit der Signalmasse belegt. Die Spannung auf jeder Leitung wird in Bezug auf diese Signalmasse gemessen. Es müssen also mindestens drei

Leitungen verwendet werden, um Daten gleichzeitig senden und empfangen zu können. Es ist aber auch möglich, ohne die Masseleitung zu arbeiten, allerdings langsamer und manchmal treten Fehler auf.

Die übrigen Leitungen sind nur für die Kontrolle des Datenaustausches vorgesehen und nicht für die Übertragung von Bytes. Alle diese Signale könnten über eine einzige Leitung gesendet werden, aber für jedes Signal gibt es eine eigene dedizierte Leitung. Diese Leitungen werden Modem-Steuerleitungen genannt. Diese Steuerleitungen befinden sich entweder im aktiven Zustand (Ein) mit einer Spannung von +12 V oder im inaktiven Zustand (Aus) mit einer Spannung von -12 V. Eine der Leitungen signalisiert dem Computer, das Senden von Bytes an das Modem zu unterbrechen. Umgekehrt signalisiert eine andere Leitung dem Modem, das Senden von Bytes an den Computer einzustellen. Wieder andere Leitungen veranlassen das Modem, aufzulegen, oder teilen dem Computer mit, dass eine Verbindung hergestellt wurde oder dass ein Klingelsignal auf der Leitung liegt (wenn jemand versucht anzurufen).

2.2.3 Ein internes Modem enthält einen seriellen Port

Für ein internes Modem gibt es keinen 9-poligen Verbindungsstecker, aber das Verhalten ist exakt so, als würden die oben erwähnten Leitungen existieren. Anstelle eines 12-Volt Spannungspegels auf einer Leitung, der den Status auf einer Modemsteuerleitung angibt, verwendet das interne Modem einfach ein Statusbit in seinem eigenen Speicher (ein Register), um den Status dieser nicht existierenden Steuerleitung zu bestimmen. Der interne serielle Port des Modems sieht für den Computer aus wie ein richtiger serieller Port. Für ihn gelten sogar die selben Geschwindigkeitsgrenzen für die Datenübertragung, die man bei gewöhnlichen seriellen Schnittstellen einstellt, z.B. 115200 Bits/s. Unglücklicherweise arbeiten viele interne Modems nicht auf diese Weise, sondern nutzen MS Windows Software für ihre Funktion und arbeiten daher nicht unter Linux. Siehe auch den Abschnitt 6.4 (Interne Modems, die zu meiden sind (Winmodems etc.)).

2.3 I/O-Adressen und IRQ

Weil der Computer mit jeder seriellen Schnittstelle kommuniziert, muss das Betriebssystem (Operating System, OS) wissen, dass die seriellen Schnittstellen existieren, wo sie sind (ihre I/O-Adressen) und welche Leitung die serielle Schnittstelle verwendet, um einen Dienst der CPU des Computers in Anspruch zu nehmen. Deshalb muss innerhalb der seriellen Schnittstelle sowohl die Ein-/Ausgabe-Adresse (I/O) als auch die Interrupt ReQuest Nummer (IRQ) in einem nicht flüchtigen Speicher abgelegt werden. Der IRQ bestimmt, welche Leitung für ein Interrupt Signal verwendet wird, um einen Dienst der CPU zu beanspruchen. Siehe auch Abschnitt 2.5 (Interrupts).

Beim PCI-Bus funktioniert das alles etwas anders, da der PCI-Bus über eine eigene Interrupt-Verarbeitung verfügt. Das BIOS auf PCI-Boards bildet die PCI-Interrupts jedoch auf IRQs ab, so dass sich nach außen hin alles so verhält wie oben beschrieben, mit der Ausnahme, dass sich zwei oder mehr Geräte einen IRQ teilen können.

I/O-Adressen sind nicht dasselbe wie Adressen im Hauptspeicher. Wenn eine I/O-Adresse auf dem Adressbus des Computers anliegt, wird eine andere Leitung aktiviert. Beides bedeutet für den Hauptspeicher, die Adresse zu ignorieren, und es bedeutet für alle angeschlossenen Geräte, die über I/O-Adressen-verfügen (wie z.B. der serielle Port) die Adresse auszuwerten und sie mit der eigenen zu vergleichen. Wenn die Adresse übereinstimmt, liest das I/O-Gerät die Daten auf dem Datenbus.

2.4 Namensgebung

Die seriellen Schnittstellen heißen unter Linux `ttyS0`, `ttyS1`, usw. (die korrespondierenden Schnittstellen unter DOS/Windows heißen normalerweise COM1, COM2, usw). Im `/dev` Verzeichnis gibt es eine spezielle Datei für jeden Port. Um sie anzuzeigen, können Sie folgenden Befehl eingeben:

```
ls /dev/ttyS*
```

Nur weil es z.B. eine Datei `ttyS3` gibt, muss das aber nicht heißen, dass auch ein physischer serieller Port existiert.

Welcher dieser Namen (`ttyS0`, `ttyS1`, etc) zu welchem physikalisch Port gehört, wird wie folgt bestimmt: die Treibersoftware des seriellen Ports verwaltet eine Tabelle, welche die Zuordnung von I/O-Adressen und ttyS-Namen enthält. Diese Abbildung von Namen auf I/O-Adressen und IRQs kann mit Hilfe des Befehls `setserial` sowohl gelesen als auch gesetzt werden, siehe 8.1 (Was ist setserial?). Dieser Befehl setzt *nicht* die I/O-Adressen und IRQs in der Hardware selbst (dies wird durch Jumper oder Plug-and-Play Software vorgenommen). Welcher physikalische Port also z.B. mit `ttyS1` korrespondiert, hängt folglich sowohl davon ab, was die Treibersoftware (per `setserial` Befehl) denkt und wie die Hardware konfiguriert ist. Falls ein Fehler gemacht wurde, kann es vorkommen, dass der physikalische Port keinem einzigen Namen (wie z.B. `ttyS2`) entspricht und daher auch nicht verwendet werden kann. Siehe auch 7 (Gerätedateien für die serielle Schnittstelle /dev/ttyS2, usw.).

2.5 Interrupts

Bytes kommen über die Telefonleitung zum Modem, werden vom Modem von analog nach digital umgewandelt und werden auf dem Weg zu ihrem Ziel innerhalb des Computers der seriellen Schnittstelle übergeben. Wenn die serielle Schnittstelle z.B. 8 Bytes vom Modem empfängt (diese Anzahl kann auch auf 1, 4, 8 oder 14 eingestellt sein), signalisiert sie der CPU, diese Daten abzuholen, indem sie ein elektrisches Signal sendet. Dieses Signal liegt auf einer ganz bestimmten Leitung an und ist eine Aufforderung an die CPU, ihre momentane Arbeit kurz zu unterbrechen und sich um die serielle Schnittstelle zu kümmern (engl: Interrupt Request, abgekürzt IRQ). Ältere serielle Schnittstellen senden so ein Unterbrechungssignal für jedes einzelne empfangene Byte.

Jede Interrupt-Leitung innerhalb des Computers hat eine Nummer (im allgemeinen ebenfalls als IRQ bezeichnet) und die serielle Schnittstelle muss wissen, auf welcher Leitung sie ihre Anforderung zu signalisieren hat. Z.B. verwendet `ttyS0` in der Regel die IRQ Nummer 4 (IRQ4). Eine Liste dieser Nummern (und mehr) ist in der Manual Page des Kommandos `setserial` zu finden (`man setserial`, im Abschnit »Configuring Serial Ports«). Ein Interrupt Signal wird ausgelöst, wann immer die serielle Schnittstelle die Aufmerksamkeit der CPU benötigt. Es ist wichtig, dass dies rechtzeitig geschieht, weil der Speicherpuffer der seriellen Schnittstelle nur 16 empfangene Bytes (in älteren Modems ist es nur ein Byte) aufnehmen kann. Wenn die CPU es nicht schafft, diese emfangenen Daten rechtzeitig abzuholen, dann bleibt für weitere ankommende Bytes kein Platz mehr übrig und der kleine Buffer kann überlaufen (Overrun). Es gehen dann Bytes verloren. Für ein externes Modem gibt es keine Möglichkeit, den Datenfluss schnell genug zu stoppen, um diese Situation zu verhindern. Bei einem internen Modem ist der 16 Byte Buffer auf derselben Karte und ein gutes Modem wird nicht in diesen Buffer schreiben, wenn er voll ist. Deshalb wird ein gutes internes Modem den Buffer nicht zum überlaufen bringen und dies ist ein Vorteil eines internen Modems.

Interrupts werden auch ausgelöst, wenn die serielle Schnittstelle gerade alle 16 Bytes ihres kleinen Sendebuffers zum Modem übertragen hat. Es ist dann wieder Platz für 16 weitere, zu sendende Bytes. Der Interrupt teilt dies der CPU mit, so dass sie weitere Bytes in den Sendebuffer schreiben kann. Ein Interrupt wird auch ausgelöst, wenn eine Modem-Steuerleitung ihren Status ändert.

Die oben genannten Buffer sind alle in Form von Hardware realisert. Die serielle Schnittstelle hat auch große Buffer im Bereich des Hauptspeichers. Dies wird später erklärt.

Interrupts übertragen eine Menge Informationen, allerdings nur indirekt. Der Interrupt selber signalisiert lediglich einem Chip, dem Interrupt Controller, dass ein bestimmter serieller Port um Aufmerksamkeit bittet. Die CPU wird darüber vom Interrupt Controller unterrichtet. Die CPU führt dann ein bestimmtes Programm aus, um den seriellen Port zu bedienen. Dieses Programm wird Interrupt Service Routine genannt und ist Bestandteil der Treibersoftware für den seriellen Port. Es versucht herauszufinden, was im seriellen Port passiert ist und behandelt dann das Problem, wie z.B. die Übertragung von Datenbytes aus oder in den Buffer des Ports. Die Service Routine kann leicht feststellen, was passiert ist, weil der serielle Port über spezielle Register verfügt, deren I/O-Adressen der Treibersoftware bekannt sind. Diese Register enthalten Statusinformationen über den seriellen Port. Die Software liest diese Register aus, erhält dadurch Information, was passiert ist und kann entsprechende Maßnahmen einleiten.

2.6 Datenkompression (durch das Modem)

Bevor wir mit den Grundlagen zur seriellen Schnittstellen fortfahren, müssen wir noch eine weitere Funktion des Modems verstehen: Datenkompression. In manchen Fällen wird diese Aufgabe in Wirklichkeit von der Software ausgeführt, die auf der CPU des Computer läuft, aber unglücklicherweise funktioniert diese Software bislang nur unter MS Windows. Wir beschäften uns hier mit dem Fall, dass das Modem selbst diese Aufgabe erledigt, denn dies ist eine Voraussetzung, damit das Modem auch unter Linux funktioniert.

Um Daten schneller über die Telefonleitung übertragen zu können, kann man sie komprimieren, indem man ein maßgeschneidertes Kodierungsverfahren verwendet, welches selber wieder von der Art der Daten abhängt. Die kodierten Daten sind kleiner (sie enthalten die gleiche Information in weniger Bytes) und können in kürzerer Zeit über das Internet übertragen werden. Dieser Vorgang wird als »Datenkompression« bezeichnet.

Wenn Sie Dateien aus dem Internet herunterladen, dann sind sie wahrscheinlich bereits komprimiert und es ist für das Modem unmöglich, sie noch weiter zu komprimieren. Ihr Modem kann evtl. feststellen, dass die Daten bereits komprimiert wurden und einen weiteren Kompressionsversuch unterlassen. Wenn Sie Daten empfangen, die von dem anderen Modem komprimiert wurden, so wird Ihr Modem sie dekomprimieren und sehr viel mehr Bytes erzeugen, als über die Telefonleitung gesendet wurden. Daher wird der Datenfluss vom Modem in den Computer höher sein als der Fluss über die Telefonleitung zu Ihnen. Das Verhältnis beider Flussgrößen wird Kompressionsrate genannt. Kompressionsraten bis zu einem Wert von 4 sind möglich, aber sehr selten.

2.7 Fehlerkorrektur

Ähnlich wie Datenkompression können Modems auch eine Fehlerkorrektur durchführen. Eigentlich ist der Name »Fehlerkorrektur« falsch, denn das Modem kann einen Fehler nur erkennen und den fehlerhaften Datenblock ein zweites Mal anfordern. Das Modem kann den Fehler aber nicht selbständig korrigieren. Wenn man es genau nimmt, müsste man also von »Fehlererkennung« sprechen.

Die serielle Schnittstelle erwartet, dass jedes zu übertragende Byte von zwei zusätzlichen Bits eingerahmt ist: ein Start-Bit und ein Stop-Bit. Ohne Verwendung einer Fehlerkorrektur werden diese zusätzlichen Bits vom Modem über die Telefonleitung übertragen. Obwohl bei den Fehlerkorrekturverfahren zusätzliche Steuer- und Prüfsummeninformationen übertragen werden, die den effektiven Datendurchsatz (in Byte/s) reduzieren, führt die Tatsache, dass jetzt auf die Start- und Stop-Bits verzichtet werden kann, in Wirklichkeit zu einer leichten Erhöhung des Datendurchsatzes. Die Bytes werden jetzt in einzelnen Paketen übertragen, und die Start- und Stop-Bits werden überflüssig.

2.8 Datenflussrate (Übertragungsgeschwindigkeit)

Daten (Bytes, aus denen Briefe, Bilder usw. bestehen) fließen vom Computer zum Modem und dann über die Telefonleitung (und umgekehrt). Durchsatzraten wie z.B. 56k (56000 Bits/s) werden fälschlicherweise auch als »Übertragungsgeschwindigkeit« bezeichnet, dieser Begriff hat sich jedoch weitgehend durchgesetzt. Gäbe es keine Datenkompression, dann würde die Durchsatzrate zwischen Computer und Modem etwa mit der Durchsatzrate auf der Telefonleitung übereinstimmen.

In Wirklichkeit sind jedoch zwei verschiedene Übertragungsgeschwindigkeiten an Ihrem Ende der Telefonleitung zu unterscheiden:

- Die Übertragungsgeschwindigkeit auf der Telefonleitung zwischen beiden Modems (DCE Geschwindigkeit)
- Die Übertragungsgeschwindigkeit zwischen der seriellen Schnittstelle und dem Modem (DTE Geschwindigkeit)

Wenn Sie eine Verbindung mit einem Modem auf der anderen Seite der Telefonleitung aufbauen, schickt Ihnen das Modem oft eine Nachricht wie z.B. `CONNECT 28000` oder `CONNECT 115200`. Was bedeuten diese Angaben? Nun, damit kann entweder die DCE- oder die DTE-Geschwindigkeit gemeint sein. Wenn der Wert höher ist als auf dem Modem angegeben, muss es die DTE Geschwindigkeit zwischen Modem und Computer sein. Dies ist der Fall für den oben genannten Wert 115200. Der Wert 28800 kann nur die DCE-Geschwindigkeit sein, weil die serielle Schnittstelle diesen Wert nicht kennt. Das Modem kann so konfiguriert werden, dass es die eine oder die andere Geschwindigkeit meldet. Einige Modems melden beide Werte und zeigen den DCE-Wert z.B. als `CARRIER 28800` an.

Wenn Sie ein internes Modem verwenden, würden Sie vielleicht nicht erwarten, dass es eine Begrenzung für die Übertragungsgeschwindigkeit zwischen Ihrem Modem und Ihrem Computer gibt, weil sich das Modem innerhalb des Computers befindet und fast ein Teil Ihres Computers ist. Aber es gibt diese Grenze, weil das Modem einen fest zugeordneten seriellen Port enthält.

Es ist wichtig zu verstehen, dass die durchschnittliche Geschwindigkeit häufig kleiner ist als die angegebene Geschwindigkeit, besonders auf der kurzen Übertragungsstrecke zwischen Computer und Modem. Die geringere Durchschnittsgeschwindigkeit ist durch Wartezeiten bedingt. Wegen der 2.9 (Flusskontrolle) können durchaus Wartezeiten von etwa einer Sekunde auftreten. Andererseits können auch sehr kurze Verzögerungen von wenigen Mikrosekunden vorkommen, die das Ende des einen Bytes und den Anfang des nächsten kennzeichnen. Außerdem können Modems auf kleinere Geschwindigkeiten umschalten, wenn die Qualität der Telefonleitung während der Verbindungsdauer schlechter wird.

Im Abschnitt 12 (Welche Geschwindigkeit sollte ich verwenden?) wird erläutert, welche DTE Geschwindigkeit eingestellt werden sollte.

2.9 Flusskontrolle

Unter Flusskontrolle wird die Fähigkeit verstanden, den Fluss von Bytes auf einer Übertragungsleitung zu unterbrechen und ohne Datenverlust wieder aufzunehmen. Mittels Flusskontrolle kann das Modem die momentane Durchsatzrate der Daten beeinflussen.

2.9.1 Ein Beispiel für Flusskontrolle

Nehmen wir einmal an, Sie verbinden ein 33,6k Modem, welches einfach Bytes mit genau 33,6 kBits pro Sekunde (bps) über die Telefonleitung sendet und empfängt, mit dem seriellen Port. Das Modem soll keinerlei Datenkompression oder Fehlerkorrektur durchführen. Und nehmen wir weiter an, dass Sie die Geschwindigkeit der seriellen Schnittstelle auf 115.200 Bits/s (bps) eingestellt haben und Daten vom Computer zum Modem übertragen. Dann beträgt der Datenfluss vom Computer zum Modem 115,2 kbps. Der Datenfluss des Modems über die Telefonleitung beträgt im besten Fall jedoch nur 33,6 kbps. Weil ein höherer Datenfluss in das Modem hineingeht als von ihm wieder herauskommt, speichert das Modem die

Differenz (115,2k - 33,6k = 81,6k) in einem seiner Buffer. Dieser Buffer kann schnell überlaufen, wenn der 115,2k-Fluss nicht gestoppt wird.

Und nun kommt die Flusskontrolle zu Hilfe. Wenn der Buffer des Modems fast voll ist, sendet das Modem ein Stop-Signal zum seriellen Port. Der serielle Port leitet dieses Signal zum Gerätetreiber weiter der 115,2 kbps Fluss wird gestoppt. Das Modem sendet weiterhin Daten mit 33,6 kbps aus seinem vorher angefüllten Buffer. Weil keine Daten mehr in diesen Buffer hineinfließen, beginnt er sich zu leeren. Wenn fast keine Bytes mehr im Buffer vorhanden sind, sendet das Modem ein Start-Signal zum seriellen Port und der 115,2 kbps Datenstrom beginnt wieder zu fließen. Flusskontrolle bewirkt also im Ergebnis einen durchschnittlichen Datendurchsatz (in unserem Fall 33,6 kbps), der deutlich unter dem Durchsatz von 115,2 kbps im eingeschalteten Zustand liegt. Dies ist die so genannte »Start-Stop« Flusskontrolle.

Das eben geschilderte ist ein einfaches Beispiel für Flusskontrolle für einen Datenfluss vom Computer zum Modem, aber auch für die entgegengesetzte Übertragungsrichtung (vom Modem oder einem anderen Gerät zum Computer) kann Flusskontrolle genutzt werden. Bei jeder Richtung des Datenflusses spielen 3 verschiedene Buffer ein Rolle: 1. innerhalb des Modems, 2. innerhalb des UART-Bausteins im seriellen Port und 3. in dem Hauptspeicher, der vom Gerätetreiber des seriellen Ports verwendet wird. Flusskontrolle schützt diese Buffer vor einem Überlauf. Die kleinen Hardware-Buffer des seriellen Ports sind durch diesen Mechanismus nicht direkt geschützt, sie vertrauen statt dessen auf die schnelle Reaktion auf die von ihnen ausgelösten Interrupt-Signale.

Dies war das Wichtigste über Flusskontrolle, aber es gibt natürlich noch mehr Details.

Für die Übertragungsrichtung vom Modem zum PC wird Flusskontrolle nur selten notwendig sein. Im Kapitel »Complex Flow Control Example« des *Serial HOWTO* wird ein Beispiel behandelt, in dem dies notwendig ist. Aber wenn Sie die Übertragungsrate zwischen Modem und Computer (Geschwindigkeit des seriellen Ports) nicht hoch genug eingestellt haben, dann ist es notwendig, den Datenfluss vom Modem zum Computer zu reduzieren. Um dies zu erreichen muss der über die Telefonleitung hereinkommende Datenfluss unterbrochen werden können. Ihr Modem muss das andere Modem veranlassen, weiter Daten zu senden. Siehe 2.9.4 (Modem-zu-Modem Flusskontrolle)

2.9.2 Hardware und Software Flusskontrolle

Falls möglich, ist es das Beste, Hardware-Flusskontrolle zu verwenden; dabei werden zwei der Modem-Kontrollleitungen verwendet, um das Start- und Stop-Signal zu übermitteln. Software-Flusskontrolle nutzt die normalen Sende- und Empfangsleitungen, um die Start- und Stop-Signale zu übertragen. Für diesen Zweck werden die ASCII-Kontrollzeichen DC1 (Start) und DC3 (Stop) verwendet. Sie werden einfach in den normalen Datenstrom eingefügt. Software-Flusskontrolle ist nicht nur langsamer hinsichtlich der Reaktionszeit, auch die Übertragung binärer Daten ist nicht möglich, ohne dass besondere Vorsichtsmaßregeln getroffen werden. Weil beliebige binäre Daten sehr wahrscheinlich die Bitmuster DC1 und DC3 enthalten, muss mit besonderen Mitteln zwischen einem DC3-Zeichen, das das Flusskontroll-Zeichen »Stop« bedeutet und einem DC3-Zeichen, das Teil der binären Daten ist, unterschieden werden. Das gleiche gilt für DC1. Eine funktionierende Software-Flusskontrolle für binäre Daten setzt sowohl Modem- (Hardware) als auch entsprechende Software-Unterstützung voraus.

2.9.3 Anzeichen fehlender Flusskontrolle

Die Theorie der Flusskontrolle zu verstehen, kann von praktischem Nutzen sein. Z.B. habe ich mein Modem für einen Internet-Zugang verwendet, und alles schien gut zu funktionieren. Nach einigen Monaten versuchte ich, große Dateien von meinem PC zum ISP (Internet Service Provider) zu übertragen, dabei kam es zu vielen Übertragungswiederholungen und Fehlern. Das Empfangen von Dateien aus der anderen Richtung (von meinem ISP zu mir) funktionierte einwandfrei. Es stellte sich heraus, dass das Problem ein Hardwarefehler in meinem Modem war, der dazu führte, dass die Flusskontrolle deaktiviert wurde. Der Buffer des Modems lief über, wenn große Dateien gesendet wurden, weil nie ein Stop-Signal zum Computer gesendet wurde, um den Datenfluss zum Modem zu unterbrechen. Für die entgegengesetzte Übertragungsrichtung vom Modem zum Computer trat kein Problem auf, weil die Endgerätegeschwindigkeit (z.B. 115,2 kBit/s) immer höher war als die Übertragungsgeschwindigkeit auf der Telefonleitung. Die Lösung des Problems bestand darin, ein Kommando zur Aktivierung der Flusskontrolle am Ende des Modem-Initialisierungsstrings anzuhängen.

2.9.4 Modem-Modem Flusskontrolle

Dies ist die Flusskontrolle für die Daten, die zwei Modems über die Telefonleitung austauschen. In der Praxis gibt es diese Flusskontrolle nur, wenn die Fehlerkorrektur eingeschaltet ist. Es ist aber tatsächlich möglich, Software-Flusskontrolle zwischen zwei Modems auch ohne Fehlerkorrektur einzuschalten, dies kann aber das Senden von binären Daten stören und daher wird sie selten verwendet.

2.10 Weg des Datenflusses; Puffer

Obwohl viel von diesem Thema schon erklärt wurde, z.B. Flusskontrolle, ein Paar 16 Byte Puffer der Hardware des seriellen Ports und ein Paar von größeren Puffern innerhalb des Modems, so gibt es doch noch ein anderes Pufferpaar. Dies sind größere Puffer (z.B. 8 kByte) im Hauptspeicher, die so genannten Schnittstellenpuffer des seriellen Ports. Wenn ein Anwendungsprogramm Bytes zum seriellen Port (und damit zum Modem) sendet, werden sie zuerst im Sendepuffer des seriellen Ports im Hauptspeicher zwischengespeichert. Das Pufferpaar besteht aus diesem Sendepuffer und einem Empfangspuffer für die entgegengesetzte Richtung des Datenstroms.

Der Gerätetreiber des seriellen Ports holt sich aus diesem Sendepuffer z.B. 16 Bytes, ein Byte nach dem anderen, und legt sie im 16 Byte Hardware-Sendepuffer zur Übertragung ab. Wenn die Bytes erst einmal in diesem Sendepuffer gelandet sind, gibt es keine Möglichkeit mehr, ihr Versenden zu verhindern. Sie werden dann zum Modem übertragen, welches wiederum über einen genügend großen Puffer (z.B. 1 kByte) verfügt. Wenn der Gerätetreiber (durch die Flusskontrolle angefordert) den ausgehenden Datenfluss des Computers unterbricht, wird in Wirklichkeit nur der Datenstrom aus dem großen Sendepuffer des Hauptspeichers unterbrochen. Selbst nachdem dies geschehen ist und der Datenfluss zum Modem gestoppt wurde, kann ein Anwendungsprogramm weiterhin Daten in diesen 8k-Sendepuffer hineinschreiben bis er voll wird.

Wenn dieser Puffer gefüllt ist, kann das Anwendungsprogramm keine Daten mehr in diesen Puffer schreiben (der `write()`-Befehl in einem C-Programm blockiert) und das Anwendungsprogramm stoppt vorübergehend in seinem Ablauf und wartet, bis wieder ein wenig Speichplatz im Puffer verfügbar wird (synchrone Arbeitsweise). Daher kann ein Flusskontroll-»Stop« Signal im Extremfall das Programm anhalten, welches die Bytes sendet. Obwohl dieses Programm steht, muss das nicht bedeuten, das auch der Computer steht. Andere Prozesse können die CPU Zeit beanspruchen, während er auf ein »Start«-Signal von der Flusskontrolle wartet.

Die eben geschilderten Vorgänge wurden ein wenig vereinfacht dargestellt, da es für das Anwendungsprogramm auch die Alternative gibt, etwas anderes zu tun, während es auf die »write«-Anweisung wartet (asynchrone Arbeitsweise).

2.11 Modem Befehle

Befehle werden vom Kommunikationsprogramm über die selben Leitungen an das Modem gesendet, die auch für die Übertragung von Daten genutzt werden. Die Befehle bestehen aus kurzen ASCII-Strings, z.B. »AT&K3« um die Hardware-Flusskontrolle (RTS/CTS) zwischen Computer und Modem zu aktivieren, oder »ATDT5393401« um die Nummer 5393401 anzuwählen. Beachten Sie, dass alle Befehle mit »AT« beginnen. Einige Befehle, wie z.B. der Befehl zum Einschalten der Flusskontrolle, helfen dabei, das Modem zu konfigurieren. Andere Befehle wie z.B. der Wählbefehl bewirken, dass tatsächlich etwas passiert. Es gibt etwa 100 verschiedene möglich Befehle. Wenn Ihr Kommunikationsprogramm startet, sendet es zuerst einen Initialisierungsbefehl an das Modem, der aus mehreren AT-Befehlen bestehen kann, um es richtig einzustellen. Alle Befehle werden über die gewöhnlichen Datenleitungen gesendet, bevor das Modem wählt oder einen Anruf erhält.

Wenn das Modem mit dem anderen Modem verbunden ist (Online Modus), wird alles, was vom Computer zum Modem gesendet wird, direkt zum anderen Modem übertragen und vom Modem nicht als Befehl interpretiert. Es gibt eine Möglichkeit, diesen Modus zu verlassen und zurück in den Kommandomodus zurückzukehren, in dem alles, was an das Modem gesendet wird, als Befehl interpretiert wird. Der Computer sendet einfach die Zeichenfolge »+++«, wobei innerhalb eines bestimmten Zeitfensters vor und nach dem Senden dieses Befehls keine anderen Daten übertragen werden. Wenn das Zeitfenster eingehalten wird, kehrt das Modem in den Befehlsmodus zurück. Eine andere Möglichkeit dies zu erreichen ist ein Signal auf einer bestimmten Modemsteuerleitung.

Es gibt eine Reihe von Tabellen dieser Modem-Befehle im Internet. Der Abschnitt 16.4 (Seiten im Internet) enthält Links zu einigen solchen WWW Adressen. Verschiedene Modem-Modelle und -marken verwenden nicht genau den gleichen Satz von Befehlen. Was bei einem Modem funktioniert, kann bei einem anderem Modem versagen. Einige gemeinsame Befehle (es wird nicht garantiert, dass sie für alle Modems funktionieren) sind in diesem HOWTO im Abschnitt 5 (Modem Konfiguration) aufgelistet.

2.12 Gerätetreibersoftware für den seriellen Port

Der Gerätetreiber für den seriellen Port ist die Software, die die Hardware des seriellen Ports anspricht. Der Linux Kernel bedient sich der Funktionen der Treibersoftware, um die serielle Schnittstelle zu nutzen. Dieser Gerätetreiber wird nun als ladbares Modul angeboten. Das Modul wird normalerweise vom Kernel automatisch geladen, wenn es benötigt wird. Kernel ab der Version 2.2 unterstützen dies. In älteren Kernel-Versionen musste der Prozess `kerneld` laufen, um Module bei Bedarf automatisch zu laden. Anderenfalls musste das Modul explizit im Verzeichnis `/etc/modules` zu finden sein. Bevor unter Linux Module verwendet wurden, waren Gerätetreiber üblicherweise Teil des Kernels (»monolithischer Kernel«).

Wenn das Modul mit dem seriellen Gerätetreiber geladen wird, werden Informationen über die vorhandenen seriellen Ports auf dem Bildschirm angezeigt (oft mit einem falschen IRQ). Wenn aber mithilfe von `setserial` dem Gerätetreiber der hoffentlich richtige Wert für den IRQ mitgeteilt wurde, sollten Sie eine weitere Bildschirmausgabe mit den richtigen Werten sehen. Der Abschnitt 8.1 (Was ist setserial) enthält mehr Informationen über den `setserial`-Befehl.

Man kann den Treiber modifizieren, wenn man den Kernel-Quellcode ändert. Vieles, was den seriellen Treiber betrifft, findet sich in der Datei `serial.c`. Informationen über die Entwicklung von Programmen für den seriellen Port gibt das *Serial Programming HOWTO*.

3 Konfiguration von Modems (und des seriellen Ports)

Wenn Sie ein Modem nur unter MS Windows/DOS nutzen möchten, können Sie fast jedes Modem verwenden und es wird funktionieren. Mit einem Linux PC ist es normalerweise nicht so einfach, außer Sie verwenden ein externes Modem. Alle externen Modems sollten funktionieren; selbst wenn es »Plug-and-Play«-Modems sind. Aber die meisten neueren internen Modems sind Plug-and-Play-Modems (PnP) und verfügen über serielle PnP-Ports. Sie müssen vielleicht das Linux-Programm `isapnp` verwenden, um diese seriellen PnP-Ports zu konfigurieren. Mehr Informationen dazu finden sie im *Plug and Play HOWTO*.

3.1 Übersicht über die Konfiguration

Weil jedes Modem über einen zugeordneten seriellen Port verfügt, besteht die Konfiguration eines Modem aus zwei Teilen:

- Der Konfiguration des Modems; dies wird vom Kommunikationsprogramm übernommen.
- Der Konfiguration des zugeordneten seriellen Ports; dies wird nur teilweise vom Kommunikationsprogramm durchgeführt.

Die meisten der Konfigurationsschritte (aber nicht notwendigerweise auch die meiste Arbeit) werden vom Kommunikationsprogramm erledigt, welches Sie zusammen mit Ihrem Modem einsetzen: z.B. von `minicom`, `seyon`, oder vom PPP Daemon (welcher normalerweise notwendig ist, um eine Internet Verbindung aufzubauen). Wenn Sie Ihr Modem verwenden, um einen Dial-In-Zugang zur Verfügung zu stellen, wird Sie das Programm `getty`, welches dem externen Benutzer einen login-Prompt zur Verfügung stellt, bei der Konfiguration unterstützen. Um also das Modem und große Teile des seriellen Ports zu konfigurieren, müssen Sie das Kommunikationsprogramm (oder den PPP Daemon oder `getty`) richtig einstellen.

Bitte beachten Sie jedoch, dass nicht die gesamte Konfiguration des seriellen Ports vom Kommunikationsprogramm (oder `getty` übernommen wird. Wie der verbleibende Rest zu konfigurieren ist, ist einfach festzustellen (aber häufig nicht so einfach durchzuführen). Die Arbeit besteht hauptsächlich darin, die I/O-Adresse und den IRQ einzustellen. Plug-and-Play könnte dies völlig für Sie übernehmen. Aber es gibt ein schwerwiegendes Problem: Linux (Stand: Anfang des Jahres 1999) unterstützt Plug-and-Play nicht besonders gut. Dies kann für Sie ein schwieriges Problem bedeuten. Der nächste Abschnitt behandelt die Konfiguration des seriellen Ports.

4 Konfiguration des seriellen Ports

4.1 Überblick über die Konfiguration

In vielen Fällen wird die Konfiguration automatisch passieren und Sie selbst müssen gar nichts tun. Aber manchmal müssen Sie etwas einstellen oder Sie möchten die aktuellen Einstellungen feststellen. In diesem Fall müssen Sie mit den beiden Teilen der Konfiguration des seriellen Ports unter Linux vertraut sein:

Der erste Teil (Low-Level-Konfiguration) weist dem Port eine I/O-Adresse, einen IRQ und einen Namen (z.B. `ttyS2`) zu. Die I/O-Adresse und der IRQ müssen sowohl in der Hardware eingestellt als auch dem Gerätetreiber mitgeteilt werden. Im Folgenden werden wir dies kurz als »io-irq«-Konfiguration bezeichnen. Der `setserial`-Befehl wird verwendet, um die Werte dem Gerätetreiber mitzuteilen. Über PnP, Jumper usw. werden die Hardware-Einstellungen festgelegt. Die Einzelheiten hierzu werden später beschrieben. Wenn Sie diese Werte einstellen müssen, aber einige wichtige Details nicht verstehen, können Sie leicht in Schwierigkeiten geraten.

Der zweite Teil (High-Level-Konfiguration) weist dem Port eine Geschwindigkeit (z.B. 38,4 kBit/s) zu, schaltet die Flusskontrolle ein usw. Dies wird häufig von Kommunikationsprogrammen wie z.B. `pppd`, `minicom` oder `getty` übernommen. Diese High-Level-Konfiguration kann auch mithilfe des `stty`-Befehls durchgeführt werden. `stty` ist auch nützlich, um den momentanen Status zu ermitteln, falls Probleme auftreten oder das Terminal-Interface auf den Normalzustand zurückzusetzen, indem Sie z.B.

```
stty sane
```

eingeben.Siehe auch den Abschnitt »Stty« im *Serial HOWTO*. Wenn Linux gestartet wird, so versucht das System herauszufinden, welche seriellen Ports existieren und für diese Ports wird eine Low-Level-Konfiguration durchgeführt. Was genau passiert, hängt vom BIOS, der Hardware, der Linux Distribution usw. ab. Wenn die serielle Schnittstelle funktioniert, gibt es für Sie evtl. keinen Grund, irgend etwas zu konfigurieren. Anwendungsprogramme führen die High-Level-Konfiguration oft selbständig durch, aber Sie müssen eventuell zuvor einige Informationen eingeben, um dies zu ermöglichen. Mit seriellen PnP-Ports, wie sie oft in internen Modems enthalten sind, ist die Situation noch komplexer geworden. In folgenden Fällen müssen Sie eine Low-Level-Konfiguration durchführen (d.h. IRQ und I/O-Adresse einstellen):

- Sie möchten mehr als zwei serielle Schnittstellen verwenden.
- Sie möchten eine neue serielle Schnittstelle installieren (z.B. ein internes Modem).
- Es treten bei einem (oder mehreren) seriellen Ports Probleme auf.

Ab der Kernelversion 2.2 können Sie mehr als zwei serielle Ports verwenden, ohne eine Low-Level-Konfiguration durchführen zu müssen, indem mehrere serielle Ports sich einen Interrupt teilen (Interrupt Sharing). Aber dies muss nicht einfacher sein als die Low-Level-Konfiguration. Siehe auch 4.5.2 (Gemeinsame Verwendung von Interrupts ab Kernel 2.2).

Die Low-Level-Konfiguration (d.h. das Einstellen des IRQ und der I/O-Adresse) verursacht offensichtlich mehr Probleme als die High-Level-Konfiguration, obwohl sie meist automatisch abläuft, ohne dass man etwas tun müßte. Deshalb beschäftigt sich der überwiegende Teil dieses Abschnitts mit diesem Thema. Der serielle Port wird u.U. überhaupt nicht funktionieren, solange der Gerätetreiber nicht die richtigen Werte für den IRQ und die I/O-Adresse weiß. Es ist sehr wahrscheinlich, dass er nichteinmal von Linux gefunden wird. Selbst wenn der Port gefunden wird, kann es sein, dass er nur sehr langsam arbeitet, falls der IRQ falsch eingestellt ist. Siehe auch 14.10 (Text erscheint auf dem Bildschirm nur langsam und nach langen Wartezeiten).

In der Wintel-Welt werden die I/O-Adresse und der IRQ als »Ressourcen« bezeichnet, daher konfigurieren wir bestimmte Ressourcen. Zur Wiederholung: die Low-Level-Konfiguration besteht daraus, zwei Werte (für den IRQ und die I/O-Adresse) an zwei Orten abzulegen:

1. im Gerätetreiber (häufig, indem beim Booten der `setserial`-Befehl ausgeführt wird
2. in den Registern der Hardware des seriellen Ports

Zur Kontrolle können Sie die Meldungen verfolgen, die während des Bootvorgangs am Bildschirm ausgegeben werden. Normalerweise sind die Angaben richtig. Aber wenn Sie Probleme haben, ist es sehr wahrscheinlich, dass einige dieser Meldungen nicht die wahre Konfiguration der Hardware angeben (und das kann man auch nicht erwarten); siehe 4.3 (I/O-Adresse und IRQ: Meldungen beim Booten).

4.2 Häufige Fehler bei der Low-Level-Konfiguration

Hier ist eine Auflistung von häufigen Fehlern:

`setserial`-Befehl

Er wird aufgerufen (ohne die Option »autoconfig«) und es wird angenommen, dass dabei die Werte aus der Hardware gelesen wurden, was aber nicht der Fall ist.

`setserial-Meldungen`

Man geht fälschlicherweise davon aus, dass die beim Booten angzeigten Meldungen angeben, wie die Hardware wirklich konfiguriert ist.

`/proc/interrupts`

Wenn der serielle Port nicht verwendet wird, werden seine Interrupts hier nicht angezeigt, und fälschlicherweise wird angenommen, dass der serielle Port nicht gefunden werden kann (oder kein IRQ eingestellt ist).

`/proc/ioports`

Es wird häufig angenommen, dass hier die Hardware-Konfiguration angezeigt wird, während in Wirklichkeit etwa die gleichen, eventuell fehlerhaften Informationen angezeigt werden wie beim `setserial`-Befehl.

4.3 I/O-Adressen und IRQ: Meldungen beim Booten

In vielen Fällen wird während des Bootens eine automatische aber nicht immer richtige Low-Level-Konfiguration der seriellen Ports durchgeführt. Die Meldungen, die beim Booten auf den Bildschirm geschrieben werden, geben darüber Aufschluss. Vernachlässigen Sie auch nicht die BIOS-Meldungen; sie erscheinen noch bevor Linux geladen wird. Meldungen, die nicht mehr auf dem Bildschirm sichtbar sind, können Sie mit der Tastenkombination »Shift-Bild-nach-oben« wieder zurückholen. Der Befehl `dmesg` kann verwendet werden, um jederzeit die Boot-Meldungen anzuzeigen, häufig erscheinen aber wichtige Meldungen nicht. Das Folgende ist ein Beispiel der Boot-Meldungen (Stand: Mitte 1999). Beachten Sie, dass »ttyS00« das selbe bedeutet wie `/dev/ttyS0`.

Zunächst sehen Sie, was von Linux erkannt wurde; aber der IRQ ist nur geraten:

```
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
```

Etwas später können Sie sehen, welche Werte abgespeichert wurden, aber sie sind ebenfalls nicht unbedingt richtig:

```
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
```

Beachten Sie, dass es hier einen Widerspruch gibt: die erste Meldung zeigt für `ttyS2` den IRQ=4 an, während die zweite Meldung für diesen Wert IRQ=5 ausgibt. Vielleicht sehen Sie nur die erste Meldung. Meistens ist die letzte Meldung die richtige. Wenn Sie Schwierigkeiten haben, kann dies sehr verwirrend sein. Bevor Sie die detaillierten Erklärungen im Rest dieses Abschnittes durcharbeiten, ist es vielleicht das Beste, Sie probieren Ihren seriellen Port einfach aus. Wenn alles funktioniert, ist es nicht unbedingt notwendig, hier weiterzulesen.

Die zweite Meldung ist die Ausgabe des `setserial`-Befehls, der beim Booten ausgeführt wird. Sie zeigt an, was der Gerätetreiber für die korrekte Konfiguration hält. Aber auch dies kann falsch sein. Z.B. könnte IRQ=8 in der Hardware eingestellt sein (in diesem Fall sind beide Meldungen falsch). Der IRQ=5 wird angezeigt, weil irgend jemand fälschlicherweise diesen Wert in einer Konfigurationsdatei (o.ä.) angegeben hat. Die Tatsache, dass Linux manchmal von falschen Werten für den IRQ ausgeht liegt darin begründet, dass Linux diese IRQs nicht ausprobiert. Linux geht einfach von den Standardwerten aus (erste Meldung) oder akzeptiert die Werte, die Sie während der Konfiguration angegeben haben (zweite Meldung). Keiner der beiden Angaben ist notwendigerweise richtig. Wenn der serielle Gerätetreiber mit einem falschen IRQ arbeitet, ist der serielle Port sehr langsam oder funktioniert evtl. überhaupt nicht.

Die erste Meldung ist das Ergebnis des Versuchs, die seriellen Ports anzusprechen. Wenn hier ein Port auftaucht, heißt das, dass er existiert, aber der angegebene IRQ kann falsch sein. Linux überprüft IRQs nicht, weil das nicht sehr sicher ist, sondern geht einfach von den Standardwerten aus. Sie können sie mit dem `setserial`-Befehl und der `autoconfig`-Option manuell prüfen, aber auch hier gibt es keine Garantie für die Richtigkeit der Werte.

Die Hardwareeinstellung wird durch die als erstes sichtbaren BIOS-Meldungen angezeigt. Wenn Sie über einen seriellen PnP-Port verfügen, ist es möglich, dass das Programm `isapnp` aufgerufen wird und diese Einstellungen geändert werden. Beachten Sie hierzu die Meldungen, nachdem Linux gestartet wurde. Die letzte Meldung des seriellen Ports im obigen Beispiel sollte mit den BIOS-Meldungen übereinstimmen. Wenn dies nicht der Fall ist, dann müssen Sie entweder die Hardware-Einstellungen ändern, oder Sie verwenden `setserial`, um dem Treiber die tatsächliche Hardwareeinstellung mitzuteilen.

Wenn Sie serielle PnP-Ports verwenden, wird Linux ausserdem diese Ports nicht finden, solange der IRQ und die I/O-Adresse der Hardware nicht durch die PnP-Software eingestellt wurden. Dies ist häufig der Grund, dass die Meldungen beim Start keinen existierenden seriellen Port anzeigen. Eventuell kann die PC Hardware (ein PnP BIOS) diese Low-Level-Konfiguration automatisch vornehmen. Das Vorgehen für die PnP-Konfiguration wird später erklärt.

4.4 Wie ist die richtige I/O-Adresse und der richtige IRQ für meinen seriellen Port?

Der letzte Abschnitt zeigte, wie man diese Informationen ermitteln kann, indem man die Meldungen beim Start des Systems auswertet. Eine andere Überschrift für diesen Abschnitt könnte auch heißen: »Andere Wege, um die I/O-Adresse und den IRQ herauszufinden«. Falls Ihnen die Meldungen beim Start ausreichend Information liefern, brauchen Sie diesen Abschnitt nicht unbedingt zu lesen.

Es gibt zwei mögliche Antworten auf die Frage »Wie ist die richtige I/O-Adresse und der richtige IRQ für meinen seriellen Port?«:

- Die Werte, die aus Sicht des Gerätetreibers gesetzt wurden (dies wird durch `setserial` gesetzt und angezeigt).
- Die Werte, die tatsächlich in der Hardware eingestellt sind.

Beide Werte sollten übereinstimmen. Wenn Sie Schwierigkeiten haben (eingeschlossen Kommunikationsprogramme, die nicht kommunizieren können), kann dies bedeuten, dass diese beiden Einstellungen nicht übereinstimmen. Dies heißt mit anderen Worten, dass der Treiber falsche Informationen über die physikalisch vorhandene Hardware hat. Wenn der Gerätetreiber von einer falschen I/O-Adresse ausgeht, wird er versuchen, Daten an einen nicht vorhandenen seriellen Port zu senden, oder gar an ein anderes Gerät als an einen seriellen Port. Wenn die Treibersoftware auf einen falschen IRQ hört, wird sie keine Unterbrechungsanforderungen vom seriellen Port empfangen, was zu einem Überlauf des Buffers im seriellen Port und zu sehr langen Antwortzeiten führen kann (siehe auch 14.10 (Text erscheint auf dem Bildschirm erst nach langen Verzögerungen)). Falls der Treiber von einem anderen als dem installierten UART Typ ausgeht, können sich ebenfalls Probleme ergeben. Um festzustellen, ob beide I/O-IRQ Wertepaare identisch sind, müssen Sie Antworten auf die Fragen der nächsten beiden Abschnitte finden.

4.4.1 Von welchen Werten geht der Gerätetreiber aus?

Das ist leicht herauszufinden. Sehen Sie sich einfach die Meldungen beim Systemstart an oder verwenden Sie folgenden Befehl:

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

Wenn alles funktioniert, stimmen die so ermittelten Werte höchstwahrscheinlich mit den in der Hardware konfigurierten Werten überein. Es gibt auch einige andere Möglichkeiten, diese Informationen zu erhalten, indem Sie sich bestimmte »Dateien« im `/proc`-Verzeichnis ansehen. Wichtig ist es aber, immer daran zu denken, dass nur die Werte angezeigt werden, von denen der Gerätetreiber ausgeht. Manche sehen sich bestimmte Dateien im `/proc`-Verzeichnis an und denken, dass das, was sie sehen, auch in der Hardware eingestellt ist, dies muss aber nicht notwendigerweise so sein.

`/proc/ioports` zeigt die I/O-Adressen, die die Treiber verwenden. `/proc/interrupts` zeigt die IRQs, die von Treibern laufender Prozesse (welche eine Gerätedatei geöffnet haben) verwendet werden. Beachten Sie, dass Sie in beiden o.g. Fällen nur sehen, was der Treiber denkt, und nicht unbedingt, was wirklich in der Hardware eingestellt ist. `/proc/interrupts` zeigt auch, wie oft die Interrupts für ein bestimmtes Gerät bereits ausgelöst wurden (häufig viele tausend Male). Dies kann Hinweise zur Fehlersuche geben, denn wenn Sie eine große Zahl von bereits ausgelösten Interrupts sehen, bedeutet dies, dass es irgendwo auch Hardware gibt, welche diesen Interrupt verwendet. Wenn nur einige wenige Interrupts angezeigt werden, bedeutet das nicht, dass diese Interrupts wirklich durch einen seriellen Port erzeugt wurden. Wenn Sie also fast keine Interrupts für einen Port feststellen, den Sie versuchen zu verwenden, könnte dieser Interrupt nicht auf der Hardware eingestellt sein und dies bedeutet, dass der Treiber einen falschen Interrupt verwendet. Um sich den Inhalt von `/proc/interrupts` anzusehen, während ein Programm läuft, das die zu überprüfenden Interrupts verwendet (wie z.B. `minicom`), sollten Sie versuchen, eine Shell auszuführen, ohne das Programm zu verlassen (z.B. auf einer anderen virtuellen Konsole oder in einem zweiten xterm-Fenster).

4.4.2 Wie ist die Hardware des seriellen Ports eingestellt?

Wie können Sie herausfinden, auf welche I/O-Adressen und auf welchen IRQ die Hardware tatsächlich eingestellt sind? Vielleicht erhalten Sie aus den BIOS Meldungen einige Information, bevor Linux startet. Verwenden Sie die »Shift-PageUp«-Taste, um rückwärts durch die Meldungen zu blättern, und sehen sie sich die allerersten Meldungen an, die vom BIOS stammen, als Linux noch nicht gestartet war. Die Werte können durch `setserial` nicht verändert werden, wohl aber durch `isapnp` oder durch »pciutils«.

Eine ziemlich grobe Methode besteht darin, `setserial` mit der »autoconfig«-Option zu verwenden und eine angenommene Adresse einfach auszuprobieren (siehe 8.1 (Was ist setserial?)). Bei einem seriellen PCI-Port können Sie sich den Inhalt von `/proc/pci` oder `/proc/bus/pci/devices` ansehen. Falls Ihr seriellen Port ein Plug-and-Play Port ist, finden Sie weitere Informationen in den nächsten beiden Unterabschnitten.

Verfügt der Port über Jumper, gibt die Stellung der Jumper die Konfiguration wieder. Wenn es sich nicht um einen PnP-Port handelt, der Port aber durch ein DOS-Programm konfiguriert wurde, dann sind die Einstellungen wirksam, die mit diesem Programm gesetzt wurden.

4.4.3 Wie ist meine PnP Hardware eingestellt?

PnP Ports speichern ihre Einstellungen nicht innerhalb der Hardware, wenn der Rechner abgeschaltet wird. Ports mit Jumpern verhalten sich genau entgegengesetzt, denn hier bleiben die Einstellungen auch im abgeschalteten Zustand erhalten. Ein ISA PnP-Port kann sich in einem Zustand befinden, in dem er überhaupt keine I/O-Adresse oder IRQ hat und daher deaktiviert ist. Es sollte jedoch immer noch möglich sein, den Port mithilfe des Programms `pnpdump` (Teil der `isapnptools`) zu finden. Wenn Sie die Option »–dumpregs« verwenden, sollte die tatsächliche I/O-Adresse und IRQ, die im Port eingestellt sind, angezeigt werden.

Bei PnP Ports macht es nicht viel Sinn, nachzusehen, wie sie unter DOS/Windows konfiguriert sind. Windows speichert die Konfiguration in der Registry, die von Linux nicht verwendet wird. Falls Sie vom PnP BIOS die Konfiguration automatisch erstellen lassen, wenn Sie Linux starten (und dabei im BIOS eingestellt haben, dass Sie kein PnP Betriebssystem haben), dann sollte Linux die Konfiguration verwenden, die im nicht flüchtigen BIOS Speicher enthalten ist.

4.5 Die Wahl des seriellen IRQs

Wenn Sie echtes PnP verwenden, wobei entweder das Betriebssystem oder ein PnP BIOS alle Geräte konfiguriert, brauchen Sie keine IRQs einzustellen. PnP stellt die optimalen Einstellungen fest und weist diese Werte zu. Aber wenn Sie Linux-Werkzeuge für PnP verwenden (`isapnp` und `pcitools`), dann müssen Sie explizit Werte auswählen. Wenn Sie bereits wissen, welchen IRQ sie verwenden möchten, können Sie diesen Abschnitt überspringen, es sei denn, Sie möchten wissen, dass der IRQ 0 eine besondere Bedeutung hat.

4.5.1 IRQ 0 ist kein IRQ

Während der IRQ 0 dem Hardware Timer zugeordnet ist, hat dieser Wert für den seriellen Port eine spezielle Bedeutung, wenn er mit `setserial` eingestellt wird. Damit wird dem Treiber mitgeteilt, dass es keinen Interrupt für den Port gibt und der Treiber wird Polling verwenden, d.h. den Port selbständig in regelmäßigen Zeitabschnitten nach neuen Daten abfragen. Das ist nicht sehr effizient, kann aber verwendet werden, wenn es einen Interrupt-Konflikt oder einen falsch eingestellten Interrupt gibt. Der Vorteil dieser Vorgehensweise liegt darin, dass Sie nicht zu wissen brauchen, auf welchen Interrupt die Hardware eingestellt ist. Sie sollte nur als Notlösung verwendet werden, bis Sie einen richtigen Interrupt gefunden haben, den Sie verwenden können.

4.5.2 Gemeinsame Verwendung von Interrupts ab Kernel 2.2

Im allgemeinen gilt, dass jedes Gerät seinen eigenen IRQ verwenden sollte, eine gemeinsame Nutzung sollte vermieden werden. In einigen Fällen ist dies aber erlaubt, z.B. bei Verwendung der meisten Multi-Port Karten. Aber selbst wenn dies möglich ist, ist es nicht sehr effizient, denn jedesmal, wenn ein gemeinsam verwendeter Interrupt auftritt, muss zunächst festgestellt werden, welches Gerät ihn ausgelöst hat. Falls möglich, sollte also jedem Gerät ein eigener Interrupt zugeordnet werden.

Bei älteren Kernelversionen konnten serielle IRQs nur bei den meisten Multi-Port Karten gemeinsam genutzt werden. Ab der Kernelversion 2.2 können serielle IRQs in manchen Fällen auch gemeinsam von allen seriellen Ports verwendet werden. Damit die gemeinsame Nutzung funktioniert, muss der Kernel mit der Option »CONFIG_SERIAL_SHARE_IRQ« übersetzt worden sein, und die Hardware der seriellen Ports muss die gemeinsame Nutzung unterstützen. Selbst wenn Sie also einen Kernel ab Version 2.2 einsetzen, ist es das Beste, gemeinsam verwendete Interrupts zu vermeiden.

4.5.3 Welchen Interrupt soll man wählen?

Die Hardware der seriellen Ports läßt sich häufig nur auf eine kleine Auswahl von möglichen IRQs einstellen. IRQ Konflikte sollen auf jeden Fall vermieden werden. Ihr PC wird üblicherweise so eingestellt sein, dass `ttyS0` und `ttyS2` auf IRQ 4 und `ttyS1` und `ttyS3` auf IRQ 3 liegen. Ein Blick in `/proc/interrupts` zeigt, welche IRQs von den gerade laufenden Programmen verwendet werden. Sie werden wahrscheinlich keinen dieser IRQs verwenden wollen. Bevor IRQ 5 für Soundkarten genutzt wurde, wurde er oft für serielle Ports verwendet.

Das folgende Beispiel zeigt, wie Greg (Autor des *Serial HOWTO*) die Interrupts in `/etc/rc.d/rc.serial` gewählt hat. `rc.serial` ist ein Shellskript, das beim Start aufgerufen wird (es kann evtl. auch einen anderen Namen haben oder in einem anderen Verzeichnis stehen). Bei neueren `setserial`-Versionen (ab 2.15) werden die IRQs nicht mehr auf diese Art eingestellt, aber dieses Beispiel soll die Wahl der IRQs zeigen:

```
/sbin/setserial /dev/ttyS0 irq 3        # serielle Maus
/sbin/setserial /dev/ttyS1 irq 4        # Wyse Terminal
/sbin/setserial /dev/ttyS2 irq 5        # Zoom Modem
/sbin/setserial /dev/ttyS3 irq 9        # USR Modem
```

Standard IRQ Zuordnungen:

```
IRQ  0    Systemtaktgeber
IRQ  1    Tastatur
IRQ  2    programmierbarer Interrupt Controller
IRQ  3    serielle Schnittstelle ttyS1
IRQ  4    serielle Schnittstelle ttyS0
IRQ  5    Soundkarte oder paralleler Druckerport 2
IRQ  6    Diskettenlaufwerk
IRQ  7    paralleler Druckerport 1
IRQ  8    Echtzeitsystemuhr
IRQ  9    löst IRQ 2 aus
IRQ 10    frei
IRQ 11    frei
IRQ 12    PS/2 Mausanschluss
IRQ 13    mathematischer Koprozessor
IRQ 14    primärer IDE Kanal
IRQ 15    sekundärer IDE Kanal
```

Die Lösung für die Auswahl der IRQs gibt es nicht. Aber Sie sollten sicherstellen, dass der serielle IRQ nicht von der Hauptplatine oder anderen Karten verwendet wird. Die IRQs 2, 3, 4, 5, 7, 10, 11, 12 oder 15 sind mögliche Kandidaten. Beachten Sie, dass IRQ 2 und IRQ 9 identisch sind. Sie können diesen Interrupt 2 oder 9 nennen, der serielle Gerätetreiber versteht beide Namen. Wenn Sie eine sehr alte serielle Karte einsetzen, kann es sein, dass Sie IRQs ab 8 nicht verwenden können.

Die IRQs 1, 6, 8, 13 oder 14 sollten Sie niemals verwenden. Diese IRQs werden von der Hauptplatine verwendet. Nach Abschluss der Konfiguration sollten Sie `/proc/interrupts` überprüfen, während die Programme aktiv sind, die die Interrupts nutzen. So können Sie Konflikte erkennen.

4.6 Die Wahl der Adressen - Grafikkarten-Konflikte mit ttyS3

Die I/O-Adresse der Grafikkarte »IBM 8514« (und ähnliche) ist angeblich 0x?2e8, wobei ? für 2, 4, 8 oder 9 steht. Dies kann mit der I/O-Adresse von `ttyS3`, 0x02e8, Konflikte ergeben, falls der serielle Port die führende Null der hexadezimalen Adresse ignoriert; dies sollte eigentlich nicht so sein, kommt aber häufig vor. Wenn Sie versuchen, `ttyS3` mit dieser I/O-Adresse zu betreiben, könnte es also Schwierigkeiten geben.

Falls möglich, sollten Sie Standard-Adressen verwenden. Die Adressangabe bezeichnet die erste Adresse eines 8 Bit langen Bereichs. Z.B. bedeutet 3f8 in Wirklichkeit den Bereich 3f8-3ff. Jedes serielle Gerät (und ebenso andere Gerätetypen, die I/O-Adressen verwenden) benötigt seinen eigenen eindeutigen Adressbereich. Es sollten keine Überlappungen existieren. Dies sind die Standardadressen für die serielle Schnittstelle:

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

4.7 Hardwarekonfiguration von I/O-Adressen und IRQs (i.a. für PnP)

Vergessen Sie nicht, auch den Gerätetreiber mithilfe von `setserial` auf die entsprechenden Werte einzustellen, nachdem Sie die Hardware fertig konfiguriert haben. Bei normalen seriellen Schnittstellen (kein PnP) werden I/O-Adressen und IRQs entweder direkt auf der Karte durch kleine Steckbrücken, sog. Jumper, oder mithilfe eines DOS-Programms eingestellt. Der Rest dieses Abschnitts behandelt nur serielle PnP Ports. Hier ist eine Liste der Möglichkeiten, um serielle PnP Ports zu konfigurieren:

- Verwendung eines PnP BIOS Setup Menüs (normalerweise nur bei externen Modems an `ttyS0` (COM1) oder `ttyS1` (COM2)).
- Automatische Konfiguration durch ein PnP BIOS, siehe 4.7.1 (Verwendung eines PnP BIOS zur I/O-IRQ Konfiguration).
- Gar nichts ist zu tun, falls Sie sowohl einen seriellen PnP Port als auch ein PnP Linux System verwenden (siehe auch *Plug-and-Play HOWTO*).
- Verwendung von `isapnp` für einen PnP Port (kein PCI-Bus).
- Verwendung von pciutils (pcitools) bei Einsatz eines PCI-Busses.

I/O-Adressen und IRQs müssen durch PnP bei jedem Systemstart in den Hardware-Registern eingetragen werden, da PnP Hardware sich die Werte nicht selbständig merkt, wenn das System abgeschaltet wird. Eine einfache Möglichkeit, dies zu erreichen, ist, in einem PnP BIOS einzustellen, dass Sie nicht mit einem PnP Betriebssystem arbeiten. Das BIOS wird dann automatisch bei jedem Start die richtigen Werte einstellen. Dies kann aber Probleme mit Windows verursachen (Windows ist ein PnP Betriebssystem), wenn Windows unter einem BIOS gestartet wird, das davon ausgeht, dass Windows kein PnP Betriebssystem ist. Weitere Informationen hierüber finden Sie im *Plug-and-Play HOWTO*.

PnP wurde entwickelt, um die I/O-IRQ Konfiguration zu automatisieren, aber die Arbeit mit Linux wird damit (zumindest nach dem gegenwärtigen Stand der Entwicklung) nicht einfacher. Die Standard-Kernels für Linux unterstützen PnP nicht besonders gut. Wenn Sie einen Patch für den Linux Kernel verwenden, um daraus ein PnP Betriebssystem zu erstellen, dann sollten, wie oben erwähnt, alle Einstellungen vom Betriebssystem vorgenommen werden. (Ein Patch ist ein kleines Programmpaket, welches den Quellcode vor dem Compilieren modifiziert, um Fehler zu beseitigen oder neue Funktionalitäten zu ermöglichen.) Aber wenn Sie andere Geräte als die serielle Schnittstelle auf diese Weise konfigurieren möchten, werden Sie feststellen, dass Sie manche Gerätetreiber dennoch von Hand konfigurieren müssen, weil viele Treiber nicht für ein PnP Linux geschrieben wurden. Wenn Sie `isapnptools` oder das BIOS verwenden, um eine PnP Konfiguration durchzuführen, werden dadurch nur die beiden Werte in die Hardware-Register des seriellen Ports eingetragen, und wahrscheinlich werden Sie dennoch `setserial` verwenden müssen. Nichts davon ist einfach oder gut dokumentiert (Stand: Anfang 1999). Weitere Informationen finden Sie im *Plug-and-Play HOWTO* und in den FAQs zu den `isapnptools`.

4.7.1 Verwendung eines PnP BIOS zur I/O-IRQ Konfiguration

Während die Verwendung eines PnP Betriebssystems oder der `isapnptools` in der mitgelieferten Dokumentation erläutert wird, ist dies nicht der Fall, wenn Sie ein PnP BIOS zur Konfiguration verwenden möchten. Nicht alle PnP BIOS Versionen unterstützen dies vollständig. Das BIOS bietet normalerweise ein Menü, um die ersten beiden seriellen Schnittstellen im CMOS zu konfigurieren. Häufig sind die Einstellmöglichkeiten sehr begrenzt. Wenn in den Menüs nichts anderes angeboten wird, werden die ersten beiden seriellen Ports lediglich auf die Standard I/O-Adressen und IRQs eingestellt (siehe 7.1 (Namen und Nummern von Devices am seriellen Port)).

Ob Sie es nun mögen oder nicht, ein PnP BIOS wird die PnP-Konfiguration (I/O-IRQ) der Hardware beim Einschalten des PC durchführen. Dies wird vielleicht nur zum Teil erledigt, und der Rest muss vom PnP Betriebssystem erledigt werden (das Sie vielleicht nicht haben). Oder das BIOS geht davon aus, dass Sie kein PnP Betriebssystem haben; in diesem Fall werden alle PnP Geräte vollständig konfiguriert, nicht jedoch die Gerätetreiber. Das ist, was Sie wollen, aber es ist nicht immer einfach herauszufinden, was genau das PnP BIOS getan hat.

Wenn Sie im BIOS einstellen, dass Sie kein PnP Betriebssystem verwenden, sollte das PnP BIOS alle seriellen PnP Ports konfigurieren, und nicht nur die ersten beiden. Wenn Sie MS Windows 9x auf dem selben PC installiert haben, könnte es möglich sein, dass das BIOS die I/O-IRQ Werte im nicht-flüchtigen Speicher gesichert hat, die für den Betrieb von MS Windows benötigt werden. Wenn Sie gerade die aktuelleste Version dieses HOWTOs lesen, lassen Sie mich wissen, ob Windows die Werte dort abspeichert. Falls ja, werden vom BIOS die gleichen Werte auch für Linux verwendet. Wenn Sie herausfinden, welche Werte von Windows gewählt wurden, sollten sie auch für Linux gültig sein.

Wenn Sie eine neue PnP Steckkarte einbauen, sollte das BIOS seine PnP Konfiguration entsprechend anpassen. Es kann sogar nötigenfalls die I/O-IRQ Zuordnungen der exisitierenden Geräte ändern, um Konflikte zu vermeiden. Zu diesem Zweck verwaltet das BIOS eine Liste der nicht-PnP Geräte und ihrer jeweiligen Konfiguration. Voraussetzung dafür ist allerdings, dass Sie im BIOS eingestellt haben, wie diese nicht-PnP Geräte konfiguriert sind. Eine Möglichkeit, dies im BIOS einzustellen ist z.B. die Verwendung des Programms ICU unter DOS/Windows.

Aber wie können Sie herausfinden, was das BIOS genau getan hat, so dass Sie den Gerätetreiber entsprechend konfigurieren können? Das BIOS selbst kann einige Informationen hierzu bereitstellen, entweder in den Einstellungsmenüs oder durch Bildschirmmeldungen beim Start des Rechners, siehe auch 4.4.2 (Wie ist die Hardware des seriellen Ports eingestellt?)

4.8 Verwendung von setserial mit den richtigen Werten

Wenn die richtigen Werte für I/O-Adresse und IRQ auf der Hardware eingestellt sind, müssen Sie auch sicherstellen, dass der `setserial`-Befehl bei jedem Systemstart aufgerufen wird. Siehe 8.1.3 (Systemstart-Konfiguration).

4.9 Weitere Einstellungen

4.9.1 Hardware Flusskontrolle (RTS/CTS)

Dieser Begriff wird im Abschnitt 2.9 (Flusskontrolle) näher erläutert. Falls möglich, sollten Sie immer Hardware Flusskontrolle verwenden. Das Kommunikationsprogramm oder `getty` sollte eine Möglichkeit anbieten, Hardware Flusskontrolle einzustellen (und wenn Sie Glück haben, ist dies die Standardeinstellung). Dies muss sowohl im Modem (durch einen Initialisierungs-String oder standardmäßig) als auch im Gerätetreiber eingestellt sein. Ihr Kommunikationsprogramm sollte beides richtig behandeln.

Sollte dies nicht der Fall sein, müssen Sie die Hardware Flusskontrolle selbst einstellen. Beim Modem sollten Sie sicherstellen, dass diese Option entweder durch einen Initialisierungs-String oder standardmäßig aktiviert wird. Wenn Sie den Gerätetreiber entsprechend konfigurieren wollen, sollte dies am besten durch eine Datei durchgeführt werden, die beim Booten ausgeführt wird. Siehe hierzu 8.1.3 (Systemstart-Konfiguration). Sie müssen für jeden seriellen Port, für den die Hardware Flusskontrolle eingeschaltet werden soll, folgende Zeile einfügen:

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

Wenn Sie überprüfen wollen, ob die Flusskontrolle eingeschaltet ist, können Sie dies wie folgt tun: In einem Terminalprogramm wie `minicom` oder ähnlichen geben Sie »AT&V« ein, um zu sehen, wie das Modem konfiguriert ist. Suchen Sie nach »&K3«, dies bedeutet Flusskontrolle. Dann überprüfen Sie, ob der Gerätetreiber richtig eingestellt ist, indem Sie

```
stty -a < /dev/ttyS2
```

eingeben. Suchen Sie in der Ausgabe nach »crtscts« (ohne ein führendes Minuszeichen, dies würde bedeuten, dass die Flusskontrolle ausgeschaltet ist).

5 Modem Konfiguration (ohne seriellen Port)

5.1 AT Befehle

Genau wie der serielle Port, den das Modem verwendet, richtig konfiguriert werden muss, ist auch für das Modem selbst eine Konfiguration notwendig. Ein Modem wird durch Senden von AT Befehlen konfiguriert, die auf der selben seriellen Leitung übertragen werden, die auch für die Übertragung von Daten verwendet wird.

Die meisten Modems verwenden einen AT Befehlssatz. Dies sind kryptische kurze ASCII Befehle, wobei alle Befehle mit den Buchstaben »AT« beginnen, z.B. »ATZ&K3«. In diesem Beispiel sind gleich zwei Befehle enthalten: »Z« und »&K3«. Leider gibt es viele verschiedene Variationen dieses AT Befehlssatzes, so dass ein Befehl, der bei einem Modem funktioniert, bei einem anderen Modem vielleicht nicht angewendet werden kann. Daher kann nicht garantiert werden, dass die AT Befehle, die in diesem Abschnitt vorgestellt werden, auch mit Ihrem Modem funktionieren. Ausserdem muss jeder Befehl mit einem CR abgeschlossen werden, damit das Modem den Befehl ausführt.

Diese Befehlssequenzen werden entweder automatisch durch das Kommunikationsprogramm gesendet oder explizit durch Sie. Die meisten Kommunikationsprogramme verfügen über eine Möglichkeit, um direkt Befehle an das Modem zu senden. Dies eignet sich sehr gut, um das Modem zu konfigurieren. Sie können das Modem z.B. anweisen, sich eine bestimmte Einstellung zu merken, selbst wenn zwischendurch die Stromversorgung abgeschaltet wurde.

Wenn Sie ein Handbuch zu Ihrem Modem haben, können Sie wahrscheinlich alle AT Befehle darin nachschlagen. Anderenfalls können Sie versuchen, Informationen über das Internet zu bekommen. Man kann dazu eine Suchmaschine verwenden, und als Suchwörter einige Befehle verwenden, um zu vermeiden, dass Sie Seiten finden, die zwar Informationen über diese Befehle enthalten, die Befehle selber aber nicht auflisten. Sie können auch einige der Seiten ausprobieren, die im Abschnitt 16.4 (Seiten im Internet) angegeben sind.

5.2 Initialisierungs-Strings: Abspeichern und Wiederherstellen

Die Beispiele in diesem Abschnitt stammen aus dem Hayes AT Modem-Befehlssatz. Alle Befehle müssen mit den Buchstaben »AT« beginnen (z.B. »AT&C1&D3«). Wenn ein Modem eingeschaltet wird, dann stellt es sich automatisch auf eine der Konfigurationen ein, die in seinem nicht flüchtigen Speicher enthalten sind. Wenn Sie mit dieser Konfiguration zufrieden sind, brauchen Sie nichts weiter zu tun.

Falls die Konfiguration nicht zufriedenstellend ist, kann man die abgespeicherte Konfiguration entweder direkt verändern, oder das Modem jedes mal, bevor es verwendet wird, durch einen »Init-String« (= Intialisierungs-Sequenz) einstellen. Ein Kommunikationsprogramm nutzt normalerweise diese Möglichkeit. Was genau gesendet wird, hängt davon ab, wie Sie das Kommunikationprogramm konfiguriert haben oder welche Skripts Sie dafür geschrieben haben, wenn Sie Kermit verwenden. Üblicherweise können Sie den Init-String, den Ihr Kommunikationsprogramm verwenden, editieren und Ihren Bedürfnissen (bzw. den Ihres Modems) anpassen. Häufig können Sie auch innerhalb des Kommunikationsprogramms einen bestimmten Modemtyp auswählen, dann wird ein Init-String verwendet, von dem das Kommunikationsprogramm denkt, dass es der bestmögliche ist.

Die Konfiguration, die das Modem verwendet, wenn es eingeschaltet wird, (ein sogenanntes Profil) kann durch einen Init-String festgelegt werden. Wenn ein Kommunikationsprogramm dem Modem einen Init-String sendet, dann wird dieser Befehl die Standardkonfiguration ändern. Wenn z.B. der Init-String nur zwei Befehle enthält, dann werden nur die betreffenden beiden Parameter verändert. Andere Befehle können hingegen ein bereits im Modem gespeichertes Profil aktivieren, so dass der Init-String in diesem Fall die komplette Konfiguration ändern könnte.

Moderne Modems verfügen über einige verschiedene Profile, aus denen man auswählen kann. Diese Profile sind im nicht flüchtigen Speicher des Modems abgelegt; der Inhalt dieses Speichers bleibt auch dann erhalten, wenn das Modem abgeschaltet wird. In meinem Modem gibt es z.B. zwei werkseitig eingestellte Profile (0 und 1, keines von beiden kann verändert werden) und zwei benutzerdefinierte Profile (0 und 1), die der Anwender verändern und abspeichern kann. Ihr Modem verfügt u.U. über eine größere Anzahl an Profilen. Welches der benutzerdefinierten Profile beim Einschalten aktiv wird, hängt von einem anderen Parameter im Profil ab. Falls der Befehl »AT&Y0« eingegeben wird, dann wird in Zukunft das Profil 0 beim Einschalten verwendet. Beim Befehl »AT&Y1« wird dagegen Profil 1 verwendet.

Es gibt auch Befehle, um jedes der vier gespeicherten Profile ab sofort zu aktivieren. Man könnte so einen Befehl in einem Init-String verwenden. Wenn das selbe Profil aktiviert wird, das beim Einschalten geladen wurde, wird natürlich nichts geändert, es sei denn, das aktive Profile wurde seit dem Einschalten verändert. Da dies nicht ganz auszuschliessen ist, ist es eine gute Idee, einen Init-String zu verwenden, selbst wenn dabei nichts anderes bewirkt wird, als ein bereits gespeichertes Profil zu aktivieren.

Das gespeicherte Profil 0 aktivieren:
»ATZ0« aktiviert benutzerdefiniertes Profil 0 und bewirkt einen Reset
»AT&F0« aktiviert das werkseitig eingestellte Profil 0

Wenn Sie das Modem nach Ihren Wünschen konfiguriert haben (einschließlich der Aktivierung eines werkseitigen Profils mit kleinen Änderungen), möchten Sie es vielleicht als benutzerdefiniertes Profile ablegen:
»AT&W0« speichert die aktuelle Konfiguration als Anwenderprofil 0

Viele Modemanwender kümmern sich nicht darum, eine gute Konfiguration in ihrem Modem zu speichern. Statt dessen senden sie dem Modem jedesmal, bevor es verwendet werden soll, einen längeren Init-String. Eine andere Möglichkeit besteht darin, die Werkseinstellung am Beginn des Init-Strings zu aktivieren, und dann einige kleinere Änderungen mit zusätzlichen Befehlen am Ende des Init-Strings durchzuführen. Bei diesem Verfahren werden Probleme vermieden, falls jemand das benutzerdefinierte Profil, welches beim Einschalten geladen wird, verändert hat.

Einige Kommunikationsprogramme verfügen über eine Bibliothek, aus der für jeden Modem-Typ ein geeigneter Init-String ausgewählt werden kann. Die schwierigste Methode (bei der Sie jedoch auch eine Menge über Modems lernen können) besteht darin, das Modem-Handbuch zu studieren und selbst einen Init-String zu schreiben. Sie können auch die geänderte Modem-Konfiguration dauerhaft als benutzerdefiniertes Profil abspeichern, so dass künftig eine Inititalisierung des Modems nicht mehr notwendig ist.

Wenn Sie die von Kommunikationsprogrammen verwendeten Init-Strings näher untersuchen, werden Sie vielleicht Symbole finden, die keine gültigen Modem-Befehle sind. Diese Symbole stellen Befehle für das Kommunikationsprogramm dar und werden nicht an das Modem übertragen (wie z.B. das Zeichen »˜«, das für eine kurze Pause steht).

5.3 Weitere Modem-Befehle

Alle Modem-Befehl müssen mit der Zeichenfolge »AT« beginnen. Die folgenden Befehle für Hayes-kompatible Modems stellen sinnvolle Werte ein:

```
»E1«    Befehlsecho EIN
»Q0«    Modemmeldungen werden ausgegeben
»V1«    Modemmeldungen erscheinen im Klartext
»S0=0«  Modem hebt bei ankommenden Anrufen nicht ab
```

Die folgenden Befehle betreffen die Modemsteuerleitungen DCD und DSR:

```
»&C1«  DCD leuchtet nur bei bestehender Verbindung
»&S0«  DSR ist immer gesetzt, wenn das Modem eingeschaltet ist
```

Das Verhalten der DTR Steuerleitung läßt sich auch beeinflussen, aber das ist ein wenig komplizierter.

6 Modems für den Linux-PC

6.1 Externe und interne Modems

Es gibt zwei Typen von Modems für einen PC: externe und interne Modems. Das interne Modem wird im PC installiert (dazu müssen Sie u.a. das Gehäuse öffnen), während das externe Modem einfach an die serielle Schnittstelle angeschlossen wird. Interne Modems sind häufig preisgünstiger, verbrauchen in der Regel weniger Energie und beanspruchen keinen Platz auf Ihrem Schreibtisch. Externe Modems sind sehr viel einfacher zu installieren und zu konfigurieren, und verfügen über Kontrollanzeigen. Sie lassen sich leicht an einen anderen Computer anschliessen. Die meisten externen Modems haben keinen Ausschalter, und verbrauchen daher auch Energie, wenn sie nicht gebraucht werden. Es empfiehlt sich daher, das Modem-Netzteil aus der Steckdose zu ziehen, um Energie und Kosten zu sparen. Achten Sie beim Anschluss eines externen Modems auch darauf, dass der serielle Port des PC in der Lage ist, die Daten mit der notwendigen Geschwindigkeit an das Modem zu liefern. Nach Möglichkeit sollte ein UART 16550 verwendet werden, da die CPU hierdurch entlastet wird.

Für Linux stellen interne Modems ein besonderes Problem dar. Sie funktionieren aber genauso gut wie externe Modems, unter der Voraussetzung, dass Sie keines der Modems verwenden, die nur unter MS Windows funktionieren und dass Sie bereit sind, u.U. eine Menge Zeit in die Konfiguration zu investieren. Vorsicht: einige der Modems, die nur unter MS Windows laufen, sind nicht unbedingt auch als solche gekennzeichnet. Vereinbaren Sie beim Kauf ein Rückgaberecht, falls das Modem nicht unter Linux funktioniert.

Die meisten modernen Modems sind PnP Modems. Sie haben folgende Möglichkeiten:

- Verwendung des `isapnp` Programms
- Konfiguration durch ein PnP BIOS
- Verwendung eines PnP Linux (Kernelpatch)

Jede Vorgehensweise hat ihre Nachteile. Die Dokumentation zu `isapnp` ist schwer verständlich, das *Plug-and-Play HOWTO* bietet aber eine Hilfestellung. Wenn das PnP BIOS die Konfiguration durchführen soll, müssen Sie lediglich dafür sorgen, dass das BIOS weiß, dass Sie kein PnP Betriebssystem verwenden. Dennoch können dabei Fehler auftreten (siehe 4.4.2 (Wie ist die Hardware des seriellen Ports eingestellt?)). Ein Kernelpatch kann ebenfalls kompliziert sein, und für die neuesten Kernel ist u.U. kein Patch verfügbar.

Viele Linux Anwender sind der Ansicht, dass es am einfachsten ist, sich ein externes Modem zu besorgen und anzuschliessen. Aber da moderne Peripheriegeräte meistens PnP Geräte sind, müssen Sie sich sowieso früher oder später mit diesem Problem auseinandersetzen. Dennoch ist die zweckmäßigere (aber auch teurere) Lösung der Einsatz eines externen Modems.

6.2 Externe Modems

6.2.1 Externe PnP Modems

Viele externe Modems sind als »Plug-and-Play« gekennzeichnet, aber sie sollten genauso gut funktionieren wie nicht-PnP Modems. Weil Sie normalerweise das Modem mit dem seriellen Port verbinden, der seine eigenen Werte für IRQ und I/O-Adresse hat, braucht das Modem keine PnP Funktionalität für diese Einstellungen. Dennoch muss natürlich der serielle Port entsprechend konfiguriert werden, falls die Standardeinstellung nicht übernommen werden kann.

Wie kann ein externes Modem ein PnP-Modem sein, wenn es nicht durch PnP konfiguriert werden kann? Nun, es hat eine spezielle PnP Identifikation eingebaut, die von einem PnP Betriebssystem durch den seriellen Port abgefragt werden kann. Ein PnP Betriebssystem weiß dann, welcher Modemtyp an welchem seriellen Port angeschlossen ist. In diesem Fall brauchen Sie in Anwendungsprogrammen den Port nicht einzustellen (wie z.B. `/dev/ttyS2` oder COM3). Aber wenn Sie kein PnP Betriebssystem verwenden, müssen Sie diese Einstellungen manuell vornehmen.

6.2.2 Anschlusskabel und Installation

Der Anschluss eines externen Modems an eine serielle Schnittstelle ist vergleichsweise einfach. Modems benötigen einfache Modemkabel (keine »Null-Modem Kabel«) ohne gekreuzte Leitungen, wie sie im Fachhandel erhältlich sein sollten. Für den Anschluss am seriellen DB-9 oder DB-25 Stecker des Computers benötigen Sie eine Buchse am Modemkabel (keinen Stecker). Verbinden Sie das Modem mit dem seriellen Port. Wenn Sie die Standardeinstellung für IRQ und I/O-Adresse übernehmen, können Sie jetzt ein Kommunikationsprogramm starten und das Modem selbst konfigurieren.

6.2.3 Die Bedeutung der Kontroll-LEDs

- **TM:** Modemtest
- **AA:** automatische Anrufannahme
- **RD:** Daten auf der Empfangsleitung = RxD
- **SD:** Daten auf der Sendeleitung = TxD

- **TR:** Rechner ist betriebsbereit = DTR
- **RI:** jemand ruft das Modem an
- **OH:** Modem hat »abgenommen«
- **MR:** Modem betriebsbereit
- **EC:** Fehlerkorrektur ein
- **DC:** Datenkompression ein
- **HS:** Hochgeschwindigkeit

6.3 Interne Modems

Zur Installation eines internen Modems wird das Gehäuse des PC geöffnet und die Modemkarte wird in einen freien Steckplatz der Hauptplatine gesteckt. Es gibt Modems für ISA Steckplätze und für PCI Steckplätze. Während externe Modems mit einem kurzen Kabel mit der seriellen Schnittstelle verbunden werden, ist die serielle Schnittstelle bei internen Modems bereits eingebaut. Mit anderen Worten: Die Modemkarte ist sowohl ein Modem als auch eine serielle Schnittstelle.

Die I/O-Adresse und der IRQ für die serielle Schnittstelle werden vor der Installation durch sogenannte »Jumper« eingestellt. Dies sind winzige Steckbrücken, die Kontakt zwischen bestimmten Pins der Karte herstellen. PnP Modems (oder genauer der serielle Schnittstellenanteil der Modemkarte) verwenden keine Jumper zur Konfiguration, sondern werden durch spezielle Konfigurationsbefehle eingestellt. Solche Befehle können von einem PnP BIOS, dem `isapnp`-Programm (nur für ISA-Bus!) oder von einem PnP Betriebssystem gesendet werden. Windows 9x beherscht diese Art der Konfiguration. Mit Linux haben Sie folgende Möglichkeiten der Konfiguration (keine der Möglichkeiten ist immer einfach):

1. Verwenden Sie `isapnp` (evtl. automatisch beim Booten)
2. Verwenden Sie ein PnP BIOS
3. Führen Sie einen Kernelpatch für Linux durch, um aus Linux ein PnP Betriebssystem zu machen

6.4 Interne Modems, die zu meiden sind (Winmodems etc)

Die Mehrheit der seit etwa Mitte 1998 hergestellten internen Modems arbeiten *nicht* unter Linux (Winmodems o.ä.). Andere übliche Bezeichnungen sind HSP (Host Signal Processor), HCF (Host Controlled Family) und Soft-... Modem. Diese Modems überlassen die meiste (oder die gesamte) Arbeit dem Prozessor (CPU) des Rechners. Sie können keine Daten komprimieren, sie besitzten keine Möglichkeiten der Fehlererkennung und können Daten nicht packetieren. Diese Arbeit muss von der CPU geleistet werden. Weil hierfür nur Windows-Software zur Verfügung steht (bisher wurde diese Software nicht auf Linux portiert), wird ein solches Modem unter Linux nicht funktionieren. Dazu muss das Modem einen seriellen Port simulieren. Eine Liste der Modems, die unter Linux funktionieren, finden Sie hier:

```
http://www.o2.net/~gromitkc/winmodem.html
```

»Software Modem« wäre eigentlich die bessere Bezeichnung für »Winmodem«. Diesem Namen kann man wenigstens entnehmen, dass diese Software nur unter MS Windows funktioniert. Der Name »Winmodem« ist übrigens auch ein geschütztes Warenzeichen für einen besonderen Modemtyp.

Wie können Sie herausfinden, ob Ihr internes Modem unter Linux funktioniert? Zunächst müssen Sie den genauen Typ Ihres Modems in Erfahrung bringen. Falls Sie den Typ nicht wissen, und Sie auch MS Windows auf Ihrem Linux-PC installiert haben, können Sie dies mit einem Mausklick auf das Modem-Symbol in der Systemsteuerung herausfinden. Sehen Sie dann am besten in der ein paar Zeilen weiter oben erwähnten Modemliste nach. Wenn dies nicht weiterhilft oder nicht durchführbar ist, können Sie auch in der Dokumentation nachsehen, die mit dem Modem geliefert wurde. Suchen Sie nach einem Abschnitt wie »Systemvoraussetzungen« (engl. »System Requirements«). Vielleicht ist dieser Abschnitt klein gedruckt, lesen Sie ihn sorgfältig. Falls Windows als eine der Systemvoraussetzungen aufgelistet ist, wird es unter Linux wahrscheinlich nicht funktionieren.

Andererseits könnte das Modem unter Linux funktionieren, wenn nicht ausdrücklich erwähnt ist, dass Sie Windows benötigen. Falls von »Designed for Windows« die Rede ist, bedeutet das u.U. nur, dass es Microsofts Plug and Play Spezifikation unterstützt. Das wäre in Ordnung, da diese auch von Linux unterstützt wird (bis jetzt allerdings nicht besonders gut). »Designed for Windows« gibt also keinerlei Hinweis darauf, ob das Modem unter Linux funktioniert oder nicht. Sie können die Internetseite des Herstellers zu Rate ziehen oder dort per E-Mail nachfragen. Ich habe mal eine Internetseite gesehen, in der ausdrücklich stand, dass das eine Modell unter Linux funktioniert, während dies für ein anderes Modell nicht zutraf.

Soweit dem Autor bekannt, gibt es im Moment keine Bestrebungen, Winmodems unter Linux zu unterstützen. Vielleicht hilft es, wenn Sie den Hersteller auffordern, die Treiber auf Linux zu portieren.

6.5 Welche internen Modems sollte man meiden?

Folgende interne Modems sollten gemieden werden:

- 6.4 (winmodem) oder ähnlich. Diese Modems werden überhaupt nicht funktionieren.
- 6.5.3 (PCI Modems) funktionieren unter Linux selten
- 6.5.1 (MWave und DSP Modems) funktionieren vielleicht, aber nur, wenn Sie jedesmal nach dem Einschalten des PCs zunächst Windows/DOS starten
- 6.5.2 (RPI (Rockwell)) Treiber funktionieren, aber nur mit eingeschränkter Leistungsfähigkeit

6.5.1 MWave und DSP Modems

Dieser Modemtyp verwendet DSPs (digitale Signalprozessoren). Die Software für diese Prozessoren muss unmittelbar vor dem Gebrauch des Modems von der Festplatte des PC in den Speicher des DSP übertragen werden. Unglücklicherweise wird diese Übertragung durch DOS/Windows-Programme bewerkstelligt, unter Linux ist dies nicht möglich. Normale Modems, die unter Linux arbeiten, verfügen häufig ebenfalls über einen DSP (auf der Verpackung wird damit Werbung gemacht), aber die Software für den Betrieb ist im Modem gespeichert. In diesem Fall handelt es sich nicht um ein »DSP Modem« im Sinne dieses Absatzes, und das Modem sollte unter Linux funktionieren. Ein Beispiel für ein DSP Modem ist »Aptiva MWAVE« von IBM.

Wenn ein DSP Modem eine serielle Schnittstelle simuliert, ist es unter Linux verwendbar, da Linux zur Kommunikation mit einem Modem ebenfalls die serielle Schnittstelle verwendet. Wenn Sie zusätzlich zu Linux auch noch DOS/Windows auf dem selben PC installiert haben, können Sie das Modem wie folgt nutzen: Zuerst installieren Sie den Treiber unter DOS (verwenden Sie den DOS-Treiber, nicht den Treiber für Windows). Starten Sie dann DOS/Windows, stellen Sie sicher, dass das Modem initialisiert wird, und starten Sie Linux, ohne den PC abzuschalten. Eine Möglichkeit hierfür ist der Aufruf von `loadlin.exe`. Dies ist ein DOS-Programm, welches Linux von DOS aus bootet (siehe auch das *Config HOWTO*). Eine andere einfache Möglicheit ist, gleichzeitig die Tasten STRG-ALT-ENTF zu drücken. Das Modem bleibt an der selben seriellen Schnittstelle angeschlossen (und an der selben I/O-Adresse), die auch unter DOS verwendet wird.

6.5.2 Rockwell (RPI) Treiber

Modems, die einen Rockwell RPI Treiber benötigen, sind unter Linux nicht voll nutzbar, da die Treibesoftware nicht unter Linux läuft. Der RPI Treiber sorgt für die Datenkomprimierung und Fehlererkennung, indem er Software für MS Windows verwendet. Wenn Sie das Modem ohne Treibersoftware verwenden wollen (und auf die Datenkomprimierung und Fehlererkennung verzichten), können Sie die Verwendung des RPI Treibers sehr einfach deaktivieren, indem Sie dem Modem bei jedem Einschalten einen entsprechenden Initialisierungsbefehl (Init-String) senden. Bei Davids Modem lautet der Befehl »+HO«. Auf die Datenkomprimierung zu verzichten, ist kein so großer Nachteil, wie man zunächst meinen könnte. Die meisten großen Dateien, die Sie über das Internet herunterladen könnnen, sind bereits komprimiert, und jeder Versuch, sie noch weiter zu komprimieren, verlangsamt die Datenübertragung eher.

6.5.3 PCI Modems

Ein PCI Modem ist ein Modem, dass sich auf einer Steckkarte befindet, die in einen freien PCI Steckplatz auf der Hauptplatine des PC eingesteckt wird. Unglücklicherweise scheint es so, als ob beinahe alle PCI Modems nicht unter Linux funktionieren, auch wenn hin und wieder davon berichtet wird, dass es wieder jemandem gelungen sei, ein PCI Modem unter Linux erfolgreich einzusetzen.

Wenn Sie ein PCI Modem installiert haben, sehen Sie sich den Inhalt der Datei `/proc/pci` an. Wenn Sie eine hohe Speicheradresse sehen (z.B. »0xfebfff00«), nutzt das Modem eventuell einen gemeinsamen Speicherbereich. Das gemeinsame Nutzen eines Speicherbereichs wird unter Linux nicht unterstützt. Wenn es keine hohe Speicheradresse gibt und die einzige Adresse, die Sie sehen, ist eine I/O-Adresse kleiner als »0xffff??«, dann könnte das Modem funktionieren. Ich bin nicht sicher, ob diese Methode immer geeignet ist, festzustellen, ob ein Modem unter Linux betrieben werden kann oder nicht. Falls Sie erfolgreich ein PCI Modem unter Linux verwenden, lassen Sie mir bitte Informationen darüber zukommen, wie die Datei `/proc/pci` aussieht oder anhand welchen Eintrags man feststellen kann, ob ein Modem unter Linux läuft oder nicht.

7 Gerätedateien für die serielle Schnittstelle /dev/ttyS2, usw.

Gerätedateien (engl.: Devices) enthalten im Unterschied zu »normalen« Dateien keine Daten. Ein Zugriff des Betriebssystems auf eine Gerätedatei bedeutet einen Zugriff auf das entsprechende Peripheriegerät, welches durch die Gerätedatei dargestellt wird. Häufig wird zwischen Gerät und Gerätedatei gar nicht mehr unterschieden, für beide wird einfach der Begriff »Device« verwendet.

Informationen über das Anlegen von Gerätedateien im Verzeichnis `/dev` finden Sie im Kapitel »Creating Devices in the /dev directory« der *Serial HOWTO*.

7.1 Namen und Nummern von Devices am seriellen Port

Devices haben unter Linux Major und Minor Numbers. Jeder serielle Port kann durch zwei gültige Gerätedatein im `/dev`-Verzeichnis repräsentiert sein: `ttyS` und `cua`. Die entsprchenden Gerätetreiber (d.h. die Software, die auf diese Dateien zugreift) verhalten sich leicht unterschiedlich. Das `cua`-Device ist überholt und wird in Zukunft nicht mehr verwendet werden (siehe auch 7.3 (Die cua-Gerätedatei)).

DOS/Windows verwendet den Namen »COM« für die seriellen Schnittstellen, während das `setserial`-Programm die Namen »tty00«, »tty01«, ... gebraucht. Verwechseln Sie diese Bezeichner nicht mit `/dev/tty0`, `/dev/tty1`, usw., die für Ihre Konsole verwendet werden, aber keine seriellen Ports sind. Die standardmäßigen DOS-Namen (COM1, usw) und I/O-Adressen sind in folgender Tabelle aufgeführt (die Werte auf Ihrem System können evtl. davon abweichen):

```
       set-                                                    IO
DOS    serial          major minor          major minor   Adresse
COM1   tty00  /dev/ttyS0  4,  64;  /dev/cua0  5,  64        3F8
COM2   tty01  /dev/ttyS1  4,  65;  /dev/cua1  5,  65        2F8
COM3   tty02  /dev/ttyS2  4,  66;  /dev/cua2  5,  66        3E8
COM4   tty03  /dev/ttyS3  4,  67;  /dev/cua3  5,  67        2E8
```

In allen Linux-Distributionen sollten die `ttyS`-Devices bereits angelegt sein (und auch die `cua`-Devices, bis sie endgültig abgeschafft werden). Sie können dies durch folgende Befehle überprüfen:

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

7.2 Sollte ttySn auf /dev/modem gelinkt werden ?

Bei einigen Installationen werden zwei zusätzliche Devices angelegt, `/dev/modem` für das Modem und `/dev/mouse` für die Maus. Beides sind symbolische Links auf die entsprechende Gerätedatei in `/dev`, die Sie während der Installation ausgewählt haben (wenn Sie eine Bus-Maus haben ist `/dev/mouse` ein Link auf das Bus-Maus Device).

Es gab einige Diskussionen um den Sinn von `/dev/mouse` und `/dev/modem`. Von der Verwendung dieser symbolischen Links wird abgeraten. Insbesondere könnten sich Probleme mit den Lock-Dateien ergeben, wenn Sie das Modem für eine Einwahl von aussen zur Verfügung stellen wollen und `/dev/modem` verwenden. Sie können diese Links verwenden, aber stellen Sie sicher, dass sie auf das richtige Device verweisen. Andererseits müssen Sie u.U. andere Applikationen umkonfigurieren, wenn Sie diese Links verändern oder löschen.

7.3 Die cua-Gerätedatei

Jedes `ttyS` Device hat eine korrespondierendes `cua` Device. Aber das `cua` Device wird in Zukunft abgeschafft, und so empfiehlt es sich, `ttyS` zu verwenden (solange `cua` nicht explizit notwendig ist). Es gibt einen Unterschied zwischen `cua` und `ttyS`, aber ein findiger Programmierer kann es erreichen, dass sich ein `ttyS` Port wie ein `cua` Port verhält, so dass es keine Notwendigkeit mehr für `cua` gibt. Eine Ausnahme bilden einige ältere Programme, die `cua` verwenden müssen.

Worin besteht nun der Unterschied? Der hauptsächliche Unterschied zwischen `cua` und `ttyS` besteht darin, was in einem C-Programm passiert, wenn ein normaler `open()`-Aufruf versucht, einen Port zu öffnen. Ein `cua` Port kann einfach geöffnet werden, selbst wenn die Modem Signalleitungen überprüft werden und die DCD Leitung ein Öffnen eigentlich verbietet (weil sich niemand eingewählt hat und es daher auch kein Trägersignal gibt). Durch schlaue Programmierung kann man es aber erreichen, dass sich ein `ttyS` Port genauso verhält wie ein `cua` Port. Dies ist der Grund, warum `cua` früher für abgehende und `ttyS` für eingehende Verbindungen verwendet wurde.

Ab Linux Kernel 2.2 wird eine Warnmeldung in die Kernel-Logdatei geschrieben, wenn ein `cua` Port verwendet wird. Dies ist ein Zeichen dafür, dass `cua` Ports nicht mehr lange unterstützt werden.

8 Interessante Programme, die Sie kennen sollten

8.1 Was ist setserial?

8.1.1 Einführung

Eine wichtige Bemerkung gleich am Anfang: Verwenden Sie niemals `setserial` mit Laptops (PCMCIA).

Mit Hilfe des Programms `setserial` können Sie dem Gerätetreiber die I/O-Adresse des seriellen Ports mitteilen, auf welchen Interrupt (IRQ) die Hardware des Ports eingestellt ist, über welchen UART Typ Sie verfügen, etc. Das Programm kann Ihnen auch die aktuellen Einstellungen anzeigen. Zusätzlich kann es die Hardware prüfen, wenn bestimmte Optionen angegeben werden. Informationen über die Anwendung von `setserial` in Ihrer Distribution finden Sie mit

```
man setserial
```

sowie in den Info-Seiten unter `/usr/doc/setserial.../` (o.ä.).

`setserial` wird oft beim Systemstart durch ein Start-Skript aufgerufen. Es wird nur funktionieren, wenn das entsprechende Kernel-Modul geladen ist. Sollten Sie aus irgendeinem Grund das Modul zu einem späteren Zeitpunkt wieder aus dem Speicher entfernen, gehen vorgenommene Änderungen wieder verloren. In diesem Fall müssen Sie `setserial` nochmals aufrufen. Neben dem Aufruf durch ein Start-Skript wird `setserial` aufgerufen, wenn das Kernel-Modul geladen wird.

Mit Hilfe entsprechender Optionen kann `setserial` den seriellen Port zu Testzwecken an einer bestimmten I/O-Adresse ansprechen (engl.: »Probing«). Allerdings müssen Sie die I/O-Adresse wissen oder raten. Wenn Sie z.B. `/dev/ttyS2` angeben, wird die I/O-Adresse angesprochen, von der `setserial` annimmt, dass sie dem Port `/dev/ttyS2` zugeordnet ist (»0x2F8«). Wenn Sie `setserial` mitteilen, dass sich `ttyS2` an einer anderen Adresse befinden, wird es versuchen, diese andere Adresse anzusprechen.

`setserial` konfiguriert weder IRQs noch I/O-Adressen auf der Hardware des seriellen Ports. Sie müssen `setserial` genau die Werte mitteilen, auf die die Hardware mittels Jumper oder Plug-and-Play eingestellt ist. Verwenden Sie für `setserial` keinerlei aus der Luft gegriffenen Werte. Wenn Sie aber die I/O-Adresse wissen, aber nicht den Wert für den IRQ, können Sie versuchen, mit Hilfe von `setserial` den IRQ zu bestimmen.

Sie können sich eine Liste der möglichen Befehle für `setserial` ausgeben lassen, indem Sie einfach `setserial` ohne weitere Argumente eingeben (mit Ausnahme der Optionen, die nur aus einem Buchstaben bestehen, wie z.B. »-v«, die Sie bei der Fehlersuche verwenden sollten). Beachten Sie, dass `setserial` für eine I/O-Adresse die Bezeichnung »port« verwendet. Wenn Sie eingeben:

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

sehen Sie einige Informationen über die Konfiguration des Gerätetreibers für Ihre Ports. Mit der Option »-gv« erhalten Sie noch mehr Informationen. Aber die Ausgabe verrät Ihnen nicht, ob die Hardware auch wirklich auf die angezeigten Werte konfiguriert ist. Sie können `setserial` aufrufen und eine rein fiktive I/O-Adresse, irgendeinen IRQ und einen beliebigen UART-Typ zuweisen. Wenn Sie setserial das nächste mal aufrufen, wird es diese falschen Werte anstandslos ausgeben. Beachten Sie, dass Wertzuweisungen verlorengehen, wenn der PC ausgeschaltet wird, daher wird setserial automatisch beim Start von Linux aufgerufen.

8.1.2 Probing

Um herauszufinden, ob eine bestimmte serielle Hardware installiert ist, müssen Sie zunächst die I/O-Adresse wissen bzw. raten. Um die vorhandene Hardware anzusprechen, können Sie den Befehl `setserial` mit der Option »-v« und dem Parameter »autoconfigure« verwenden. Wenn die Bildschirmausgabe einen UART Typ wie z.B. 16550A anzeigt, ist alles in Ordnung. Wenn statt dessen »unknown« für den UART Typ angezeigt wird, ist wahrscheinlich überhaupt kein serieller Port an der angegebenen I/O-Adresse. Einige Billig-UART Typen identifizieren sich nicht korrekt; wenn Sie »unknown« sehen, besteht also noch eine geringe Chance, dass Sie dennoch einen seriellen Port installiert haben.

Neben dem UART-Typ kann `setserial` auch versuchen, den IRQ herauszufinden, aber auch das funktioniert nicht immer einwandfrei. In Programmversionen >= 2.15 kann das Ergebniss des zuletzt durchgeführten Tests gespeichert und in der Konfigurationsdatei `/etc/serial.conf` abgelegt werden, die beim nächsten Start von Linux gelesen wird. Das Skript, welches `setserial` beim Booten startet, führt üblicherweise kein Probing durch, aber das können Sie ändern. Weitere Informationen dazu finden Sie im nächsten Abschnitt.

8.1.3 Systemstart-Konfiguration

Wenn der Kernel das Modul mit dem Gerätetreiber für die serielle Schnittstelle lädt (bzw. wenn der Gerätetreiber fest zum Kernel hinzugebunden wurde), werden nur die Devices `ttyS0` bis `ttyS3` automatisch gefunden und der Treiber wird auf die IRQs 4 und 3 konfiguriert (unabhängig davon, auf welche Werte die Hardware eingestellt ist). Sie erkennen das an den Meldungen, die beim Systemstart ausgegeben werden, es sieht so aus, als ob `setserial` gestartet worden wäre. Wenn Sie drei oder mehr Ports verwenden, können sich Probleme ergeben, falls ein IRQ von mehr als einem Port genutzt wird.

Um diese oder ähnliche Problem zu lösen, muss `setserial` ein zweites Mal aufgerufen werden, diesmal allerdings mit den richtigen Werten für die IRQs. Eines der Skripte, die beim Systemstart ausgeführt werden, ist genau dafür zuständig. Dieser zweite Aufruf von `setserial` muss möglichst früh während des Boot-Vorgangs erfolgen, noch bevor irgendein anderer Prozess versucht, den seriellen Port zu verwenden. Eigentlich sollte bei Ihrer Linux Distribution bereits alles so eingestellt sein, dass `setserial` beim Start von Linux automatisch aufgerufen wird. Unter `/usr/doc/` bzw. `/usr/share/doc/` sollten Sie Informationen finden, was speziell bei Ihrer Distribution eingestellt ist und wie Sie diese Einstellungen ggf. verändern können.

Eine neue Konfigurationsmöglichkeit unter Verwendung von /etc/serial.conf Bis zur Programmversion 2.14 wurde `setserial` konfiguriert, indem das Shell-Skript mit dem Aufruf von `setserial` editiert wurde; oft war dies die Datei `/etc/rc.d/rc.serial`.

Ab der Version 2.15 liest dieses Shell-Skript die Konfigurationsdaten für `setserial` aus der Datei `/etc/serial.conf`. Aber selbst diese Datei muss nicht mehr manuell editiert werden, normalerweise bewirkt die Verwendung von `setserial` auf der Befehlszeile, dass `/etc/serial.conf` automatisch entsprechend angepasst wird.

Die Absicht dahinter ist natürlich, es Ihnen so einfach wie möglich zu machen: Sie müssen keine Dateien editieren, und `setserial` wird bei jedem Start von Linux automatisch für eine richtige Konfiguration sorgen. Aber es gibt dennoch einige Fallstricke: Zum Einen ist es nicht `setserial` selbst, welches die Konfiguration in `/etc/serial.conf` schreibt. Zum Anderen können sich unterschiedliche Distributionen unterschiedlich verhalten. Und schließlich können Sie dieses Verhalten auch noch beeinflussen.

Häufig passiert etwa folgendes: Wenn Sie ihren PC herunterfahren, wird das Skript, welches `setserial` beim Systemstart aufgerufen hat, nochmals ausgeführt, diesmal allerdings auf andere Weise. Beim Herunterfahren des Systems werden nur die Teile des Skripts durchlaufen, die für den Shutdown verantwortlich sind. Das Skript sorgt nun dafür, dass `setserial` aufgerufen wird, um die aktuellen Einstellungen zu ermitteln, die dann in `/etc/serial.conf` gespeichert werden. Wenn Sie also `setserial` aufrufen, werden die Änderungen nicht sofort in `/etc/serial.conf` gespeichert, sondern dann und nur dann, wenn Sie einen normalen Shutdown durchführen.

Sie können sich sicher schon vorstellen, welche Probleme auftreten können. Vielleicht haben sie keinen richtigen Shutdown durchgeführt (z.B. den PC einfach ausgeschaltet). Oder Sie haben ein wenig mit `setserial` experimentiert und vergessen, beim letzten Aufruf die richtigen Werte anzugeben. Dann werden Ihre »experimentellen« Werte gespeichert.

Wenn Sie dagegen `/etc/serial.conf` manuell editieren, werden diese Einstellungen beim Shutdown mit den aktuellen Einstellungen von `setserial` überschrieben. Es gibt eine Möglichkeit, um zu verhindern, dass `/etc/serial.conf` beim Shutdown angepasst wird. Dazu muss die erste Zeile von `/etc/serial.conf`, »###AUTOSAVE###«, entfernt werden. Von mindestens einer Distribution ist bekannt, dass diese erste Zeile automatisch beim ersten Shutdown unmittelbar nach der Installation gelöscht wird. Hoffentlich enthält `/etc/serial.conf` in Ihrer Distribution genügend Kommentare, die Ihnen weiterhelfen.

Das Skript, welches `setserial` beim Systemstart aufruft, ist in den meisten Fällen `/etc/init.d/setserial` (Debian) oder `/etc/init.d/serial` (Redhat), es sollte aber nicht editiert werden. Bei Redhat 6.0 muss die Datei `/usr/doc/setserial-2.15/rc.serial` nach `/etc/init.d/` kopiert werden, wenn `setserial` beim Booten ausgeführt werden soll.

Um einen Port zu deaktivieren, können Sie ihn mithilfe von `setserial` auf »uart none« setzen.

Das Format der Konfigurationsdatei `/etc/serial.conf` stimmt mit dem Format überein, mit dem die Parameter für `setserial` auf der Befehlszeile angegeben werden. Für jeden Port wird eine Zeile verwendet.

Seit Juli 1999 gibt es einen bekannten Fehler: Mit der »###AUTOSAVE###«-Funktion werden nur diejenigen Parameter gesichert, die mit dem Aufruf von

```
setserial -g /dev/ttyS?
```

angezeigt werden (das »?« steht dabei für 0, 1, 2, ...). Die anderen Parameter werden nicht abgespeichert (wird `setserial` mit der Option »a« aufgerufen, werden alle Parameter angezeigt). Von diesem Fehler werden nur wenige Anwender betroffen sein, weil diese Parameter nur selten verändert werden müssen und die Standardeinstellung in den meisten Fällen richtig ist. Der Fehler ist schon länger bekannt und evtl. inzwischen behoben.

Um die aktuellen, von `setserial` eingestellten Werte in der Konfigurationsdatei `/etc/serial.conf` zu speichern, ohne dass ein Shutdown ausgeführt wird, können Sie das Skript aufrufen, welches sonst beim Shutdown aufgerufen wird: `/etc/init.d/{set}serial stop`. Die »stop« Option speichert die aktuelle Konfiguration, aber die seriellen Ports arbeiten ohne Unterbrechung weiter.

Die alte Konfigurationsmethode: Editieren eines Skripts Vor der Programmversion 2.15 (1999) gab es keine `/etc/setserial.conf`-Datei. Daher war es notwendig, dasjenige Skript zu finden, welches beim Booten `setserial` aufruft und es anzupassen. Wenn ein solches Skript nicht vorhanden ist, müssen Sie eine neue Skript-Datei schreiben oder die entsprechenden Befehle in ein anderes Skript einfügen, welches während der Boot-Phase sehr früh ausgeführt wird. Wenn dieses Skript exisitiert, befindet es sich wahrscheinlich irgendwo im `etc` Verzeichnisbaum. Bei Redhat < 6.0 befindet sich eine Datei unter `/usr/doc/setserial`, die Sie jedoch zuerst unter `etc` kopieren müssen. Um ein passendes Skript zu finden, können Sie `locate` verwenden, z.B. können Sie dazu folgendes eingeben:

```
locate "*serial*"
```

Früher wurde üblicherweise das Skript `/etc/rc.d/rc.serial` verwendet. Die Debian-Distribution verwendete `/etc/rc.boot/0setserial`. Eine andere häufig verwendete Datei war `/etc/rc.d/rc.local`, allerdings sollte diese Datei nicht editiert werden, da sie eventuell nicht früh genug ausgeführt wird. Es ist bekannt, dass andere Prozesse veruchen können, auf den seriellen Port zuzugreifen, noch bevor `rc.local` aufgerufen wurde. Die Folge ist ein Fehler bei der seriellen Kommunikation.

Falls ein solches Skript mitgeliefert wird, sollte es eine Reihe von auskommentierten Beispielen enthalten. Durch entfernen der Kommentarzeichen und/oder Änderung dieser Beispiele sollte es einfach möglich sein, eine richtige Konfiguration zu erreichen. Stellen Sie sicher, dass Sie den richtigen vollständigen Pfadnamen für `setserial` und die richtigen Namen für die Gerätedateien angeben. Sie können die Änderungen testen, indem Sie das Skript von der Kommandozeile aus aufrufen (als User »root«). Das geht schneller, als das System wiederholt zu booten.

Wenn sie wollen, dass `setserial` automatisch den UART-Typ und den IRQ für das Device `ttyS3` bestimmt, können sie etwa folgendes eingeben:

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

Wiederholen Sie diese Anweisung für jeden seriellen Port, der automatisch konfiguriert werden soll. Es muss auf jeden Fall ein Device-Name angegeben werden, der auf Ihrem System auch wirklich vorhanden ist. In einigen Fällen wird dieser Befehl nicht funktionieren (bei bestimmter Hardware); wenn Sie den UART-Typ und den IRQ kennen, können Sie diese Werte auch explizit zuweisen:

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

8.1.4 IRQs

In der Standardkonfiguration teilen sich `ttyS0` und `ttyS2` den IRQ 4, während sich `ttyS1` und `ttyS3` IRQ 3 teilen. Aber das gemeinsame Verwenden eines Interrupts für die seriellen Ports ist nicht erlaubt, es sei denn, Sie haben 1. einen Kernel 2.2 oder höher, und Sie haben 2. die entsprechende Kernelfunktion auch einkompiliert und 3. die Hardware Ihrer seriellen Ports unterstützt die gemeinsame Nutzung von Interrupts. Weitere Informationen hierzu finden Sie im Kapitel 4.5.2 (Gemeinsame Nutzung von Interrupts ab Kernel 2.2).

Wenn Sie ein internes Modem installieren, sich aber die Ports `ttyS0` und `ttyS1` erhalten möchten, dann sollten Sie versuchen, einen nicht verwendeten IRQ zu finden und diesen sowohl auf dem seriellen Port (bzw. der Modem-Karte) einzustellen, als auch mithilfe von `setserial` diesen IRQ dem Gerätetreiber zuzuweisen. Wenn IRQ 5 nicht für Ihre Soundkarte verwendet wird, kann er für das Modem genutzt werden. Um die Hardware auf diesen IRQ einzustellen, können Sie `isapnp` verwenden, oder ein PnP BIOS, oder Sie machen aus Linux ein PnP-fähiges Betriebssystem. Um herauszufinden, welche freien IRQs es auf Ihrem System gibt, können Sie in der Online-Dokumentation (Manual Page) von `setserial` z.B. nach »IRQ 11« suchen.

8.2 Was ist isapnp?

`isapnp` ist ein Programm, um Plug-and-Play Geräte auf dem ISA Bus zu konfigurieren, dazu gehören auch interne Modems. Es ist im Paket »isapnptools« enthalten und enthält als weiteres Programm `pnpdump`. `pnpdump` findet alle ISA PnP Geräte und zeigt die erforderliche Konfiguration in einem Format an, welches direkt in die PnP Konfigurationsdatei `/etc/isapnp.conf` eingetragen werden kann. Mit der Option

»–dumpregs« kann es auch verwendet werden, um die aktuelle I/O-Adresse und den IRQ des vom Modem verwendeten seriellen Ports zu bestimmen. Der `isapnp`-Befehl kann auch von einem Start-Skript aufgerufen werden, so dass er bei jedem Start des Computers ausgeführt wird, um die ISA PnP Geräte richtig zu konfigurieren, selbst wenn das BIOS Ihres PCs kein PnP unterstützt. Weitere Informationen finden Sie im *Plug-and-Play HOWTO*.

8.3 Was ist wvdialconf?

`wvdialconf` versucht zu bestimmen, an welchem seriellen Port ein Modem angeschlossen ist. Es erzeugt auch eine Konfiguration für das Programm `wvdial`, das verwendet werden kann, um auf einfache Weise eine PPP-Verbindung mit einem Internet Service Provider (ISP) herzustellen. Sie brauchen aber PPP nicht zu installieren, um `wvdialconf` zu verwenden. Es kann nur Modems finden, die gerade keine Telefonverbindung haben. Es kann ebenfalls eine »passende« Initialisierungssequenz konstruieren, das funktioniert aber nicht in jedem Fall. Da dieser Befehl über keine Optionen verfügt, ist er einfach anzuwenden, aber Sie müssen einen Dateinnamen angeben, unter dem der Init-String und andere Daten gespeichert werden, z.B.:

```
wvdialconf mein_dateiname
```

8.4 Was ist stty?

`stty` ähnelt `setserial`, aber es setzt die Baud-Rate und andere Parameter eines seriellen Ports. Der Befehl

```
stty -a < /dev/ttyS2
```

sollte anzeigen, wie `ttyS2` konfiguriert ist. Die meisten Einstellungen betreffen Parameter, die Sie mit Modems niemals nutzen werden (z.B. bestimmte Einstellungen, die nur für alte Terminals aus den 70er Jahren verwendet wurden). Aber `stty` ist manchmal hilfreich, um Fehler zu finden.

Zwei der Einstellungen, die mit `stty` konfiguriert werden können sind: 1. die Hardware-Flusskontrolle durch »crtscts« und 2. ob das DCD-Signal des Modems ignoriert wird (»clocal«). Wenn das Modem kein DCD-Signal sendet und clocal deaktiviert ist (`stty` zeigt »-clocal«), dann kann ein Programm eventuell den seriellen Port nicht öffnen. Wenn der Port nicht geöffnet werden kann, reagiert das Programm u.U. nicht mehr, weil es (oft vergeblich) auf ein DCD-Signal des Modems wartet.

`minicom` setzt clocal automatisch beim Start, hier sind also keine Probleme zu erwarten. Aber die Version 6.0.192 von `kermit` hängt sich beim Befehl `set line...` auf, wenn -clocal gesetzt ist. Wenn kein DCD Signal anliegt und -clocal gesetzt ist, hängt selbst `stty` und es gibt anscheinend keine Möglichkeit, um clocal zu setzen (ausser `minicom` zu starten). Aber wenn `minicom` beendet wird, werden die ursprünglichen Einstellungen wieder hergestellt. Ein Ausweg besteht darin, mit Hilfe von `minicom` den Befehl `AT&C` zum Modem zu senden (um das DCD Signal zu erzeugen) und `minicom` ohne Reset zu beenden, so dass das DCD Signal erhalten bleibt. Dann kann erneut `stty` aufgerufen werden.

9 Das Modem ausprobieren

9.1 Sind Sie bereit, eine Verbindung aufzubauen?

Wenn Sie das Modem angeschlossen haben und wissen, an welchen seriellen Port es angeschlossen ist, können Sie es ausprobieren. Bevor Sie eine Verbindung mit dem Internet herstellen oder jemanden bitten, Ihr Modem anzurufen, können Sie zunächst etwas einfacheres probieren, etwa die Anwahl einer Nummer, um zu überprüfen, ob Ihr Modem richtig funktioniert. Beschaffen Sie sich eine Telefonnummer, von der Sie wissen, dass sich am anderen Ende ein Modem befindet. Wenn Sie nicht wissen, welche Nummer Sie anrufen können, fragen Sie einfach Ihren Händler nach einer Mailbox-Nummer. Vielleicht haben Sie ja auch einen Bekannten, der selbst ein Modem besitzt und das Sie für Testzwecke anrufen können.

Wissen Sie, an welchem seriellen Port das Modem angeschlossen ist? Das sollten Sie wissen, wenn Sie die Ports mit den richtigen I/O-Adressen und IRQs konfiguriert haben. Haben Sie sich für eine Übertragungsgeschwindigkeit, die Sie mit diesem Port verwenden möchten, entschlossen? Sehen Sie im Abschnitt 12.4 (Geschwindigkeitstabelle) nach, um einen schnellen Überblick zu erhalten oder unter 12 (Welche Geschwindigkeit sollte verwendet werden?), um genauere Informationen zu erhalten. Wenn Sie nicht wissen, welche Geschwindigkeit Sie einstellen sollen, setzten Sie sie einfach auf ein mehrfaches der Modem-Übertragungsgeschwindigkeit. Aktivieren Sie nach Möglichkeit auch die Option »Hardware Flusskontrolle« bzw. »RTS/CTS«. Ist das Modem mit einem passenden Telefonkabel mit einer Telefon-Steckdose verbunden? Sie können das Kabel an ein normales Telefon anschließen, um sicherzugehen, dass Sie einen Wählton hören.

Nun müssen Sie sich für ein Terminalprogramm entscheiden, mit dessen Hilfe Sie eine Verbindung herstellen können. Beispiele dafür sind `minicom`, `seyon` (X11), und `kermit`. Im Abschnitt 13 (Terminalprogramme und Hilfsprogramme) finden Sie weitere Informationen. Zwei Terminalprogramme werden in den nächsten Abschnitten näher behandelt: 9.2 (Anwahl mit Minicom) und 9.3 (Anwahl mit Kermit).

9.2 Anwahl mit Minicom

`minicom` ist in den meisten Linux-Distributionen enthalten. Um das Programm zu konfigurieren, brauchen Sie root-Rechte. Der Befehl

```
minicom -s
```

führt Sie direkt zu den Konfigurationsmenüs. Sie können auch einfach `minicom` starten und dann »ˆA« (drücken Sie »Strg«, halten Sie die Taste gedrückt, dann drücken Sie kurz »a«, und lassen dann die »Strg«-Taste wieder los) eingeben, um die untere Statuszeile einzublenden. Dort wird angezeigt, dass der Befehl für Hilfe »ˆA Z« lautet (»ˆA« haben Sie gerade eingegeben, drücken Sie also einfach »Z«). Vom Hilfemenü aus können Sie das Konfigurationsmenü wählen.

Um `minicom` zu konfigurieren, brauchen Sie nur wenige Parameter richtig einzustellen: der Name des seriellen Ports, an dem das Modem angeschlossen ist (z.B. `/dev/tty0`), und die Übertragungsgeschwindigkeit (z.B. 115200). Diese Werte stellen Sie im Konfigurationsmenü für den seriellen Port ein. Aktivieren Sie ausserdem, falls möglich, die Hardware Flusskontrolle (RTS/CTS). Als Protokoll sollten Sie »8N1« verwenden. Dies bedeutet: 8 Datenbits, keine Parität, 1 Stopp-Bit. Wenn Sie nicht genau die Übertragungsgeschwindigkeit finden, die Sie möchten, wählen Sie die nächst kleinere aus, für Testzwecke reicht dies zunächst aus. Verlassen Sie das Konfigurationsmenü und speichern Sie die Einstellungen als Standardeinstellungen (Default, dfl) ab. Sie können `minicom` jetzt beenden und neu aufrufen, so dass der richtige serielle Port gefunden und das Modem initialisiert wird. Über das Hilfe-Menü können Sie das Modem aber auch sofort initialisieren.

Nun sind Sie bereit für die Anwahl. Vergewissern Sie sich aber zunächst, dass das Modem richtig angeschlossen ist, indem Sie auf der Hauptseite, die unmittelbar nach dem Start von `minicom` angezeigt wird, den Befehl »AT« eingeben und die <Eingabe>-Taste drücken. Als Resultat sollte »OK« angzeigt werden. Ist dies nicht der Fall, liegt ein Fehler vor und es ist keine Anwahl möglich.

Wird »OK« angzeigt, gehen Sie zurück zum Hilfe-Menü und wählen Sie das Anwahl-Verzeichnis aus (engl: Dialing Directory). Sie können nun eine Telefonnummer eingeben und »dial« (engl. für »wählen«) selektieren, um die Nummer anzurufen. Sie können die Nummer auch »manuell« wählen (selektieren Sie »manual« und geben Sie die Nummer über die Tastatur ein). Beachten Sie genau die Fehlermeldungen, falls etwas nicht funktioniert.

9.3 Anwahl mit Kermit

Die neueste `kermit`-Version finden Sie bei:

```
http://www.columbia.edu/kermit/
```

Beispiel: Nehmen wir an, Ihr Modem ist an `ttyS3` angeschlossen, und die Übertragungsgeschwindigkeit ist auf 115200 bps eingestellt. Der Dialog am Bildschirm sieht wie folgt aus:

```
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                          ; dies ist Ihre Eingabe
OK                                ; das ist die Antwort des Modems
```

Wenn das Modem auf die »AT«-Befehle antwortet, können Sie davon ausgehen, dass das Modem funktioniert. Versuchen Sie nun, ein anderes Modem anzurufen, indem Sie folgendes eingeben:

```
ATDT7654321
```

wobei »7654321« eine Telefonnummer ist. Verwenden Sie »ATDP« anstelle von »ATDT«, wenn Sie zum wählen Pulswahl anstelle von Tonwahl verwenden müssen. In Deutschland sind alle Vermittlungsstellen der Deutschen Telekom mittlerweile auf Tonwahl umgestellt, es sollte keinen Grund mehr geben, noch die langsamere Pulswahl zu verwenden. Herzlichen Glückwunsch, wenn die Verbindung aufgebaut werden kann.

Um zur `kermit`-Eingabeaufforderung zurückzukommen, drücken Sie »Strg«, dann »Alt Gr - ß« (den umgekehrten Schrägstrich), lassen Sie »Strg« wieder los und drücken schließlich noch die »C«-Taste:

```
Strg-\-C
(zurück zu Linux)
C-Kermit>quit
#
```

Dies war nur ein Test mit der einfachen, »manuellen« Wählmethode. Die normale Methode besteht darin, `kermit` mit seiner eingebauten Modem-Datenbank die Anwahl durchführen zu lassen. Im folgenden Beispiel wird ein US Robotics (USR) Modem verwendet:

```
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        ; Modem-Typ auswählen
C-Kermit>set line /dev/ttyS3       ; serielle Schnittstelle auswählen
C-Kermit>set speed 115200          ; Geschwindigkeit auswählen
C-Kermit>dial 7654321              ; Anwahl
 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:
```

10 Konfiguration eines Einwahldienstes

10.1 Übersicht

Wenn Sie mit Ihrem Modem eine Verbindung zu einem anderen Rechner aufbauen, nutzen Sie damit gleichzeitig einen Einwahldienst auf dem anderen Rechner. Ohne diesen Dienst käme keine Verbindung zustande.

Umgekehrt können Sie es anderen Personen ermöglichen, Ihr Modem anzurufen und Ihren PC zu nutzen. In der Praxis sieht das so aus, dass jemand mit einem Modem Ihre Nummer wählt. Ihr Modem nimmt ab und beide Modems versuchen, eine Verbindung aufzubauen, indem sie sich auf ein Übertragungsprotokoll einigen. Wenn die Verbindung steht, kann Ihr PC z.B. eine Aufforderung zum Einloggen senden. Der Anrufer kann sich auf Ihrem PC anmelden und dann auf Ihrem PC arbeiten, ganz so, als ob er sich direkt an einer Konsole angemeldet hätte. Das setzt natürlich voraus, dass der Anrufer mit seinem Benutzernamen auf Ihrem PC bekannt ist und für ihn ein entsprechender Benutzer eingerichtet ist. Eine andere Möglichkeit besteht darin, dass der Anrufer mit seinem PC Zugang zum Internet erhält, falls Ihr PC eine Verbindung zum Internet hat. Das Programm, dass diese Dienste zur Verfügung stellt, heisst `getty`.

Häufig werden nach einem erfolgreichem Login weitere Programme gestartet, z.B. solche Programme, die den Anrufer mittles PPP mit dem Internet verbinden. In diesem HOWTO werden diese Programme nicht behandelt.

Der Login-Vorgang kann auch automatisiert werden, so dass das Terminalprogramm automatisch den Benutzernamen und das Passwort überträgt.

10.2 getty

getty stellt einen Einwahldienst zur Verfügung. Sie brauchen es nicht, wenn Sie mit Ihrem Modem eine andere Nummer anwählen. Neben der Fähigkeit, einen Login-Prompt zu übertragen, kann es auch auf den Telefonanruf reagieren. Ursprünglich wurde getty zum einloggen in einen Computer von einem einfachen

Terminal aus verwendet. Es wird jetzt noch verwendet, um sich an einer Linux-Konsole anzumelden. Es gibt einige verschiedene getty-Programme mit leicht unterschiedlichen Namen. Nur bestimmte Programme stellen eine Einwahlmöglichkeit zusammen mit einem Modem zur Verfügung. getty wird normalerweise beim Booten des System gestartet. Es muss von der Datei `/etc/initab` aufgerufen werden. Eventuell finden Sie in dieser Datei bereits einen Aufruf von getty, den Sie aber wahrscheinlich noch anpassen müssen. Insbesondere haben die verschiedenen getty-Programmtypen unterschiedliche Aufrufparameter.

Sie können aus vier verschiedenen getty-Programmtypen auswählen, die alle für die Konfiguration einer Einwahlmöglichkeit in Verbindung mit einem Modem verwendet werden können: `mgetty`, `uugetty`, `getty_em` und `agetty`. Einige Details werden in den nächsten Abschnitten behandelt. `agetty` ist das einfachste der vier Typen, hat allerdings auch nur eine eingeschränkte Funktionalität und wird daher hauptsächlich bei einer direkten Verbindung mit Textterminals verwendet. `mgetty` unterstützt Fax- und Sprachnachrichten, `uugetty` dagegen nicht. Angeblich fehlen `mgetty` dagegen ein paar Leistungsmerkmale von `uugetty`. `getty_em` ist eine vereinfachte Variante von `uugetty`. Wenn Sie sich also nicht ohnehin schon mit `uugetty` auskennen, sollten Sie `mgetty` verwenden.

10.2.1 mgetty

`mgetty` ist der Nachfolger von `uugetty`. Beide eignen sich für den Einsatz mit Modems. `mgetty` kann auch für direkt angeschlossene Textterminals verwendet werden. Neben dem Bereitstellen eines Login-Prompts kann `mgetty` auch mit FAX-Dokumenten umgehen und verfügt über eine automatische PPP Erkennung. Es gibt ein Zusatzprogramm namens `vgetty`, welches zusammen mit einigen Modems eine Anrufbeantworterfunktion übernehmen kann. Die `mgetty`-Dokumentation ist ausgezeichnet und nicht Bestandteil dieses HOWTOs. Um PPP automatisch zu starten, muss die Konfigurationsdatei `/etc/mgetty/login.conf` angepasst werden. Informationen zu `mgetty` finden Sie im WWW unter:

- `http://www.leo.org/˜doering/mgetty/`
- `http://alpha.greenie.net/mgetty`

10.2.2 uugetty

`getty_ps` besteht aus zwei Programmen: `getty` wird für Textkonsolen und Terminalgeräte verwendet und `uugetty` für Modems. Greg Hankins (früher Autor des *Serial HOWTO*) verwendete `uugetty`, daher sind seine Erfahrungen mit diesem Programm im Abschnitt 11 (Uugetty-Konfiguration) enthalten.

10.2.3 getty_em

`getty_em` ist eine vereinfachte Version von `uugetty`. Es wurde von Vern Hoxie geschrieben, weil er die komplexen Konfigurationsdateien für `getty_ps` und `uugetty` verwirrend fand.

Es ist Teil einer Sammlung von mehreren Dienstprogrammen und Informationsschriften für die serielle Schnittstelle von Vern Hoxie. Sie ist via FTP erhältlich unter:

```
ftp://scicom.alphacdc.com/pub/linux
```

Der Name der Datei ist `serial_suite.tgz`.

10.2.4 agetty und mingetty

`agetty` ist eine einfache, aber vollständige Implementierung von `getty`, die sich besser für virtuelle Konsolen oder Terminals als für Modems eignet. Unter günstigen Umständen kann es aber auch mit Modems verwendet werden (allerdings können Sie keinen abgehenden Anruf initiieren, wenn `agetty` aktiv ist und auf einen Anruf wartet). In der Debian-Distribution heisst `agetty` einfach `getty`.

`mingetty` ist eine kleine Version von getty, die nur für Konsolen (bzw. Monitore) funktioniert, mit Modems kann sie nicht verwendet werden.

10.3 Was passiert, wenn sich jemand einwählt?

Der Anrufer verwendet ein beliebiges Terminalprogramm, das Ihre Nummer wählt und Ihr Telefon läutet. Es gibt zwei Möglichkeiten, wie Ihr PC den Anruf entgegennehmen kann. Einmal kann das Modem den Anruf automatisch annehmen. Die andere Möglichkeit besteht darin, dass getty merkt, dass es läutet, und daraufhin einen Befehl an das Modem sendet, den Anruf entgegenzunehmen. Wenn das Modem abgenommen hat, sendet das Modem Töne zum anderen Modem (und umgekehrt). Beide Modems einigen sich auf ein Übertragungsprotokoll, und wenn das erfolgreich ist, sendet ihr Modem eine »CONNECTed«-Nachricht an getty. Wenn getty diese Nachricht erhält, sendet es den Login Prompt über den seriellen Port. Manchmal ruft getty auch ein Programm namens `login` auf, um den Login-Vorgang abzuwickeln. getty startet üblicherweise beim Booten des Systems, aber es wartet mit dem Senden des Login Prompts, bis eine Verbindung aufgebaut ist.

Nun noch einige Informationen über die beiden Möglichkeiten, den Anruf anzunehmen.

Der Wert des S0-Registers im Modem bestimmt, ob und wann ein eingehender Anruf beantwortet wird. Mit einem Terminalprogramm kann der Wert des S0-Registers mit dem Befehl »ATS0?« abgefragt werden. Mit dem Befehl »ATS0=3« wird das S0-Register des Modems auf den Wert 3 gesetzt, das Modem nimmt den Anruf nach dem 3. mal läuten entgegen. Steht das S0-Register auf 0, wird das Modem den Anruf nicht automatisch entgegennehmen, sondern erst, wenn es von getty oder einem Terminalprogramm den Befehl »ATA« (=Antwort) erhalten hat, während es läutet. Diese Möglichkeit sollte bevorzugt werden. Wenn das Modem den Anruf nicht automatisch entgegennimmt, spricht man von manueller Rufannahme, obwohl die Rufannahme von getty automatisch durchgeführt wird.

Bei »manueller« Rufannahme öffnet getty den seriellen Port zum Zeitpunkt des Systemstarts und hört auf die serielle Schnittstelle. Kommt ein Anruf herein, sendet das Modem eine »RING«-Nachricht an getty. Wenn der Anruf entgegengenommen werden soll, sendet getty einen »ATA«-Befehl an das Modem. Das Modem baut dann die Verbindung auf und sendet eine »CONNECT..«-Nachricht, sobald die Verbindung steht. getty kann dann den Login-Prompt zum Anrufer übertragen.

Bei automatischer Annahme des Anrufs durch das Modem wird das DCD-Signal (Data Carrier Detect, d.h. es ist ein Trägersignal zur Übermittlung von Daten vorhanden) der seriellen Schnittstelle verwendet, um zu signalisieren, dass eine Verbindung aufgebaut wurde. Zum Zeitpunkt des Systemstarts versucht getty, den seriellen Port zu öffnen, aber dieser Versuch schlägt im Allgemeinen fehl, da noch kein DCD-Signal des Modems anliegt. getty wartet daraufhin an dieser Stelle im Programm, bis ein DCD-Signal auftritt. Wenn das der Fall ist (u.U. viele Stunden später), wird ein Interrupt ausgelöst, getty wird aktiv, der Port kann nun geöffnet werden und getty sendet den Login Prompt. Während getty auf das DCD-Signal wartet, können andere Prozesse ohne Beeinträchtigung aktiv sein, da Linux ein Multitasking-Betriebssystem ist.

Vielleicht fragen Sie sich, wie getty im »manuellen« Rufannahmemodus den seriellen Port öffnen kann, schiesslich gibt es ja zu diesem Zeitpunkt noch kein DCD-Signal. Nun, es gibt eine programmiertechnische Möglichkeit, das Öffnen des Ports zu erzwingen, selbst wenn kein DCD-Signal anliegt.

10.4 Warum die manuelle Rufannahme bevorzugt werden sollte

Der Unterschied zwischen beiden Möglichkeiten der Rufannahme zeigt sich dann, wenn der PC abgeschaltet ist, das Modem aber noch arbeitet. Im manuellen Fall wird die »RING«-Nachricht an getty gesendet, aber da der PC nicht läuft, gibt es auch keinen getty-Prozess, und der Anruf wird nicht entgegengenommen. Es fallen daher auch keine Telefongebühren an. Im Fall der automatischen Rufannahme hebt das Modem ab, aber es wird nie ein Login-Prompt gesendet, da der PC nicht läuft. Je länger der Anrufer auf den Login wartet, umso teurer wird seine Telefonrechnung. Zudem ist es frustrierend, auf einen Login zu warten, der niemals kommt. `mgetty` verwendet den manuellen Annahmemodus. `uugetty` kann dies ebenfalls, wenn ein Konfigurationsskript verwendet wird.

10.5 Rückruf

Es gibt auch die Möglichkeit eines automatischen Rückrufs. Zuerst ruft jemand Ihr Modem an. Sie erhalten dadurch ein paar Informationen über den Anrufer wie z.B. seine Telefonnummer. Der Anrufer legt aber wieder auf und wird unmittelbar danach von Ihrem Modem zurückgerufen. Welche Gründe gibt es, dieses Verfahren einzusetzen? Ein Grund ist z.B., um Telefonkosten zu sparen, falls Sie den Anrufer günstiger anrufen können als er Sie. Ein anderer Grund ist eine höhere Systemsicherheit. Mit dem Rückruf kann sichergestellt werden, dass der Anrufer wirklich der ist, für den er sich ausgibt.

Es gibt für Linux ein Programm namens `callback` (engl. Callback = Rückruf), das mit `mgetty` zusammenarbeitet. Es ist zu finden unter

```
ftp://ftp.icce.rug.nl/pub/unix.
```

Eine Schrittweise Anleitung zur Installation von `callback` (und PPP) ist zu finden unter

```
<http://www.stokely.com/unix.serial.port.resources/callback.html>
```

10.6 Sprachnachrichten

Die Verarbeitung von Sprachnachrichten ermöglicht dem PC, als Anrufbeantworter zu fungieren. Dazu benötigen Sie ein entsprechendes Modem (»Voice-Modem«) und die passende Software. Anstatt die Sprachnachrichten auf Band zu aufzuzeichnen, werden sie in digitaler Form auf der Festplatte gespeichert. Wenn Sie einen Anruf erhalten, hört der Anrufer eine Begrüßungsansage und kann dann eine Nachricht hinterlassen. Die etwas besseren Systeme erlauben je nach Anrufer individuelle Begrüßungstexte und die Sortierung der Nachrichten in verschiedene Mailboxen. Für Linux gibt es bisher nur frei verfügbare Software für einfache Anrufbeantwortersysteme.

Zwei Anrufbeantworter-Programme sind allgemein bekannt. Das eine implementiert nur sehr einfache Funktionen (siehe 13.2.4 (Anrufbeantworter-Software)). Das andere Programmpaket ist `vgetty`, es besitzt zwar einen größeren Funktionsumfang, ist aber nicht so ausführlich dokumentiert. Es ist eine optionale Erweiterung zum gut dokumentierten und weit verbreiteten `mgetty`-Programm. Es unterstützt ZyXEL-ähnliche Modembefehle. Es sieht so aus, als ob `vgetty` im Moment nicht sehr stabil läuft, aber es wird erfolgreich eingesetzt und seine Entwicklung geht weiter. Falls jemand mehr über `vgetty` weiss, wäre ich für Hinweise bzgl. des aktuellen Entwicklungsstands sehr dankbar.

11 uugetty-Konfiguration (aus dem alten Serial HOWTO)

Vorab ein Hinweis: `mgetty` ist eine (bessere?) Alternative zu `uugetty`. Es ist neuer und weiter verbreitet als `uugetty`. Abschnitt 10.2 (getty) enthält einen kurzen Vergleich zwischen beiden Varianten.

11.1 Installation von getty_ps

Da `uugetty` ein Bestandteil des Programmpakets `getty_ps` ist, muss zunächst `getty_ps` installiert werden. Die jeweils aktuelle Version ist zu finden unter:

```
ftp://metalab.unc.edu/pub/Linux/system/serial
```

Möchten Sie hohe Übertragungsgeschwindigkeiten nutzen, dann verwenden Sie Version 2.0.7j oder höher. Systemvoraussetzung ist libc5.x oder höher.

Standardmäßig wird `getty_ps` gemäß dem Linux FSSTND (File System Standard) konfiguriert. Das bedeutet, die ausführbaren Programme befinden sich im Verzeichnist `/sbin`, und die Konfigurationsdateien heissen `/etc/conf.{uu}getty.ttyS0` usw. Dies geht nicht aus der Dokumentation hervor. Die Dateien zur Verwaltung konkurrierender Zugriffe (Lock-Dateien) befinden sich im Verzeichnis `/var/lock`. Vergewissern Sie sich, dass das `/var/lock`-Verzeichnis vorhanden ist.

Wenn Sie auf FSSTND-Konformität keinen Wert legen, werden die ausführbaren Programme in `/etc`, die Konfigurationsdateien in `/etc/dafault/{uu}getty.ttyS0` usw. und die Lock-Dateien im Verzeichnis `/usr/spool/uucp` installiert. Ich empfehle diese Vorgehensweise, falls Sie UUCP einsetzen, weil UUCP ein Problem hat, wenn die Lock-Dateien nicht da stehen, wo sie von UUCP erwartet werden.

`getty_ps` kann auch `syslogd` verwenden, um Meldungen zu protokollieren. Falls `syslogd` auf Ihrem System nicht bereits aktiv ist, können Sie sich mit `man syslogd` und mit `man syslog.conf` über die Konfiguration informieren. Meldungen werden mit der Priorität »LOG_AUTH«, Fehler mit »LOG_ERR« und Debug-Informationen mit »LOG_DEBUG« protokolliert. Falls sie nicht `syslogd` verwenden wollen, können Sie auch in der Datei `tune.h` des `getty_ps` Quellcodes eine andere Protokolldatei angeben, `/var/adm/getty.log` ist die Voreinstellung.

Entscheiden Sie sich, ob Sie die FSSTND-Konformität und die syslog-Funktionalität haben wollen. Sie können auch entweder das eine oder das andere verwenden. Passen Sie die Dateien `Makefile`, `tune.h` und `config.h` entsprechend an. Übersetzen und installieren Sie dann das Paket, wie in der enthaltenen Anleitung beschrieben.

11.2 Konfiguration von uugetty

Sie können mit Ihrem Modem eine abgehende Verbindung aufbauen, während `uugetty` den Port auf mögliche Login-Versuche überwacht. `uugetty` übernimmt auch wichtige Prüfungen auf Sperrdateien. Editieren Sie `/etc/gettydefs` und fügen Sie einen Eintrag für Ihr Modem hinzu. Mit dem folgenden Befehl können Sie überprüfen, ob Ihre Änderungen syntaktisch korrekt sind:

```
getty -c /etc/gettydefs
```

11.2.1 Moderne Modems

Wenn Sie ein Modem mit mindestens 9600 bps haben, können Sie den seriellen Port auf eine feste übertragungsgeschwindigkeit einstellen, z.B.:

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

Falls Sie die RTS/CTS Hardware Flusskontrolle verwenden, können Sie `CRTSCTS` zu den Einträgen hinzufügen:

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

11.2.2 Ältere, langsame Modems

Wenn Sie ein altes Modem mit weniger als 9600 bps verwenden möchten, müssen Sie anstelle einer Zeile für eine Geschwindigkeit mehrere Zeilen angeben, um verschiedene Geschwindigkeiten zu probieren. Beachten Sie, dass die Zeilen untereinander durch das letzte »Wort« verbunden sind (wie z.B. #38000). Zwischen jedem Eintrag muss sich eine Leerzeile befinden.

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

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

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

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

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

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

11.2.3 Login Banner

Wenn Sie möchten, können Sie `uugetty` interessante Sachen im Login-Banner ausgeben lassen. In den Beispielen von Greg sind der Systemname, die serielle Leitung und die aktuelle Übertragungsrate enthalten. Sie können aber auch weitere Informationen hinzufügen:

```
@B    Die aktuelle Übertragungsrate (bps)
@D    Das aktuelle Datum im Format MM/TT/JJ.
@L    Die serielle Leitung, an der uugetty angeschlossen ist
@S    Der Name des Systems
@T    Die aktuelle Uhrzeit im Format HH:MM:SS (24-Stunden).
@U    Die Anzahl der eingeloggten User. Dies entspricht der
      Anzahl der Einträge in /etc/utmp/, die ein nicht-leeres
      ut_name Feld haben.
@V    Der Wert von VERSION, wie in der Default-Datei angegeben.
```

Um ein einzelnes »@«-Zeichen anzugeben, können Sie entweder »\@« oder »@@« angeben.

11.3 Anpassung von uugetty an eigene Wünsche

Für jeden Port gibt es viele Parameter, an denen Sie herumdrehen können. Diese Parameter befinden sich für jeden Port in eigenen Dateien. Die Datei `/etc/conf.uugetty` wird von allen Instanzen von `uugetty` verwendet, und `/etc/conf.uugetty.ttyS0` usw. wird nur von diesem einen Port verwendet. Beispieldateien sind bei den `getty_ps` Quellcode Dateien zu finden, die Bestandteil der meisten Linux Distributionen sind. Aus Platzgründen sind sie hier nicht aufgeführt. Beachten Sie, dass die standardmäßige

Konfigurationsdatei `/etc/default/uugetty.ttyS0` usw. ist, falls Sie eine ältere Version von `uugetty` (< 2.0.7e) oder nicht den FSSTND-Standard verwenden. Die Datei `/etc/conf.uugetty.ttyS3` von Greg sah folgendermaßen aus:

```
# Beispieldatei zur uugetty-Konfiguration für ein
# Hayes-komptibles Modem, erlaubt ankommende Modemverbindungen
#
# abwechselnd die Sperrdateien überprüfen; wenn die Sperrdatei
# exisitert, wird uugetty neu gestartet, um das Modem nochmals
# zu initalisieren
ALTLOCK=cua3
ALTLINE=cua3
# zu initialisierende Leitung
INITLINE=cua3
# Zeitspanne, nach der die Verbindung unterbrochen werden
# soll, falls keine Daten übertragen werden
TIMEOUT=60
# Modem Intialisierungs-String
# format: <expect> <send> ... (chat sequence)
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# diese Zeile legt die Verzögerungszeit fest, bevor das
# Login Banner gesendet wird.
DELAY=1
#DEBUG=010
```

Fügen Sie die folgende Zeile in die Datei `/etc/inittab` ein, so dass `uugetty` auf einem seriellen Port gestartet wird. Tragen Sie die richtigen Werte für Ihre Umgebung ein: Run-Level (»2345« oder »345«, usw.), die vollständigen Pfadnamen der Konfigurationsdateien, den seriellen Port, die Übertragungsgeschwindigkeit und den standardmäßigen Terminaltyp:

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

Neustart von `init`:

```
init q
```

Als Parameter für die Übertragungsgeschwindigkeit sollten Sie den höchstmöglichen Wert angeben, den Ihr Modem unterstützt.

Nun wird Linux den seriellen Port auf eingehende Anrufe überwachen. Zu Testzwecken können Sie von einem anderen Rechner aus Ihren Rechner anrufen und sich auf Ihrem Linux System einloggen.

`uugetty` hat noch viele weitere Optionen, die in der Manual Page von `uugetty` (häufig auch einfach `getty` genannt) vollständig erläutert sind. U. a. gibt es Leistungsmerkmale für Zeitsteuerung und Rückrufmöglichkeit.

12 Welche Geschwindigkeit sollte verwendet werden?

Mit »Geschwindigkeit« meinen wir eigentlich die »Datenübertragungsrate«, aber beinahe jeder sagt einfach (und nicht ganz richtig) »Geschwindigkeit«. Bei modernen Modems haben Sie bezüglich der Geschwindigkeit, die das Modem auf einer Telefonleitung verwendet, gar keine Wahl, da es automatisch die unter den gegebenen Umständen (z.B. Leitungsqualität) höchstmögliche Geschwindigkeit nutzt. Bei der Geschwindigkeit, die zwischen Modem und Computer verwendet wird, haben Sie aber sehr wohl eine Wahl. Diese Geschwindigkeit wird manchmal auch »DTE-Geschwindikeit« genannt, DTE bedeutet »Data Terminal Equipment« (oder auf Deutsch: Datenendgeräteeinrichung), denn Ihr Computer ist ein DTE. Sie müssen diese Geschwindigkeit hoch genug einstellen, damit dieser Teil der Kommunikationsstrecke keinen Flaschenhals darstellt. Die DTE Geschwindigkeit ist immer auf die höchst mögliche Geschwindigkeit für diese Verbindung eingestellt. Meistens wird aber die tatsächliche erreichte Geschwindigkeit darunter liegen.

Bei einem externen Modem ist die DTE Geschwindigkeit die Datenübertragungsrate (in Bits/s) auf der Verbindung zwischen dem Modem und dem PC. Bei einem internen Modem ist es ganz genauso, da das Modem auch einen seriellen Port emuliert. Vielleicht erscheint es ein wenig seltsam, wenn es eine Geschwindigkeitsbegrenzung für die Kommunikation zwischen Computer und Modemkarte gibt, da die Modemkarte ja mit dem Computer über einen viel schnelleren Bus verbunden ist. Es ist aber so, da das interne Modem einen dedizierten seriellen Port darstellt, der eben Geschwindigkeitsgrenzen und einstellbare Geschwindigkeiten hat.

12.1 Geschwindigkeit und Datenkompression

Welche Geschwindigkeit sollten Sie wählen? Wenn es keine Datenkompression gäbe, könnte man für die DTE Geschwindigkeit exakt den selben Wert nehmen wie für die Modemgeschwindigkeit (auf der Telefonleitung). Die Datenkompression bewirkt, dass die Anzahl der Bytes, die vom Computer zum Modem gesendet werden, in eine kleinere Zahl von Bytes komprimiert werden, der Dateninhalt bleibt aber gleich. Beispiel: Wenn die Geschwindigkeit vom PC zum Modem 20.000 Bytes/s (bps) beträgt, und die Kompressionsrate beträgt 2:1, dann werden nur 10.000 Bytes/s über die Telefonleitung übertragen. Bei einer Kompressionsrate von 2:1 müssen Sie also für die DTE Geschwindigkeit das doppelte der maximalen Geschwindikeit auf der Telefonleitung einstellen. Beträgt die Kompressionsrate 3:1, müssen Sie die dreifache Geschwindigkeit wählen.

12.2 Wo wird die Geschwindigkeit eingestellt?

Die DTE Geschwindigkeit ist normalerweise über einen Menüpunkt in Ihrem Terminalprogramm wählbar, oder über eine Option von `getty`, wenn sich jemand auf Ihrem System einwählt. Die DCE Geschwindigkeit zwischen beiden Modems können Sie nicht einstellen.

12.3 Es kann keine ausreichend hohe Geschwindigkeit eingestellt werden

Sie müssen die höchste von Ihrer Hardware unterstütze Geschwindigkeit herausfinden. Ende 1998 unterstützt die meiste Hardware nur Geschwindigkeiten bis 115,2 kbps. Einige wenige interne 56K Modems unterstützen 230,4 kbps. Neuere Linux Kernels unterstützen hohe Geschwindigkeiten (über 115,2 kpbs), aber aus folgenden Gründen könnte es problematisch sein, diese auch zu Nutzen:

1. Das Anwendungsprogramm oder `stty` arbeitet nicht mit dieser hohen Geschwindigkeit.
2. `setserial` hat eine Standardgeschwindigkeit von 115,2 kbps; aber diese Einstellung ist leicht zu ändern.

12.3.1 Konfiguration der Hardware auf eine Geschwindigkeit: der Divisor und baud_base

Der Gerätetreiber für den seriellen Port stellt die Geschwindigkeit der Hardware durch das Senden eines »Divisors« ein (eine positive ganze Zahl). Dieser »Divisor« teilt die maximale Übertragungsgeschwindigkeit der Hardware, das Ergebnis ist eine kleinere Geschwindigkeit (mit Ausnahme des Divisors 1, mit dem die maximal mögliche Geschwindigkeit eingestellt wird). Hier ist eine Liste von gebräuchlichen Divisoren und den entsprechenden Geschwindigkeiten (unter der Annahme einer maximalen Geschwindigkeit von 115,2 kbps): 1 (115,2k), 2 (57,6k), 3 (38,4k), 6 (19,2k), 12 (9,6k), 24 (4,8k), 48 (2,4k), 96 (1,2k), etc.

Wenn Sie eine Geschwindigkeit von 115,2 kbps im Terminalprogramm oder mit Hilfe von `stty` auswählen, dann stellt der Gerätetreiber in der Hardware den Divisor 1 ein, der offensichtlich die höchstmögliche Geschwindigkeit bewirkt. Wenn Sie Hardware mit einer maximalen Geschwindigkeit von z.B. 230,4 kbps verwenden, dann wird die Auswahl von 115,2 kbps den Divisor 1 zur Folge haben und es wird in Wirklichkeit 230,4 kbps eingestellt. Dies ist das Doppelte der Geschwindigkeit, die Sie ausgewählt haben. Tatsächlich wird für jede Geschwindigkeit, die Sie auswählen in Wirklichkeit die doppelte Geschwindigkeit gesetzt. Wenn Sie Hardware hätten, die 460,8 kbps unterstützen würde, würde die tatsächliche Geschwindigkeit auf das vierfache des von Ihnen ausgewählten Wertes gesetzt.

12.3.2 Möglichkeiten, das Problem der Geschwindigkeitseinstellung zu umgehen

Um diese Berechnungsweise zu korrigieren (aber nicht, um das Problem in allen Fällen zu beheben) können Sie mit Hilfe von `setserial` den »baud_base«-Wert auf die tatsächliche maximale Geschwindigkeit Ihres seriellen Ports setzen, z.B. auf 230,4 kbps. Wenn Sie im Anwendungsprogramm oder durch `stty` eine Geschwindigkeit von 230,4 kbps wählen, wird der Divisor 1 verwendet und es wird die selbe Geschwindigkeit gesetzt, die Sie ausgewählt haben. Dabei gibt es aber ein Problem: `stty` und viele Terminalprogramme bieten maximal 115,2 kbps für die Geschwindigkeit an (Stand: Mitte 1999) und es ist nicht möglich, höhere Geschwindigkeiten auszuwählen. In diesen Fällen bleibt Ihnen nichts anderes übrig, als auf Einstellungen mit Hilfe von `setserial` zu verzichten und im Hinterkopf zu behalten, dass die tatsächlich verwendete Geschwindigkeit stets doppelt so hoch ist wie die von Ihnen ausgewählte Geschwindigkeit.

Es gibt noch einen zweiten Weg, der aber auch nicht viel besser ist. Mit dem Befehl

```
setserial /dev/ttyS2 spd_cust baud_base 230400 divisor 1
```

können Sie z.B. für den Port `ttyS2` einen Divisor 1 (d.h. maximale Geschwindigkeit) erzwingen, wenn Sie im Terminalprogramm eine Geschwindigkeit von 38,4 kbps auswählen. Die Verwendung der Option »spd_cust« ist an die Geschwindigkeit 38,4 kbps gekoppelt, näheres dazu finden Sie in der Manual Page von `setserial`.

Falls Sie zwei oder mehr verschiedene Geschwindigkeiten verwenden wollen, die so hoch sind, dass sie im Terminalprogramm nicht ausgewählt werden können, ist es nicht mehr so einfach wie oben beschrieben. Aber das Prinzip bleibt dasselbe. Sie können den Standardwert für »baud_base« beibehalten und dabei immer bedenken, dass Sie in Wirklichkeit einen Divisor bestimmen, wenn Sie eine Geschwindigkeit auswählen. Die tatsächliche Geschwindigkeit berechnet sich immer aus der maximalen Geschwindigkeit geteilt durch den Divisor, den der Gerätetreiber setzt.

12.3.3 Die Taktfrequenz entspricht nicht dem Wert von »baud_base«

Beachten Sie, dass der Wert für »baud_base« normalerweise wesentlich kleiner ist als die Frequenz des Quarzoszillators der Hardware. Die Oszillatorfrequenz wird häufig durch 16 geteilt, erst dieser Wert ergibt die tatsächliche Maximalgeschwindigkeit. Diese hohe Oszillatorfrequenz wird benötigt, damit jedes Bit mehrfach geprüft werden kann, um zu entscheiden, ob es sich um eine 0 oder um eine 1 handelt.

12.4 Geschwindigkeitstabelle

Für ein 56k Modem eignet sich ein 16650 UART am besten, aber nur wenige interne Modems oder serielle Schnittstellenkarten sind damit ausgerüstet. Am zweitbesten ist die Verwendung eines 16550 UART, der abgeändert wurde, um 230,4 kbps zu unterstützen. Die folgende Tabelle enthält die empfohlenen Geschwindikeiten des seriellen Ports in Abhängigkeit der Modemgeschwindigkeit:

- 56k (V.90): 115200 bps oder 230400 bps (eventuell einige Prozent schneller?)
- 28,8k (V.34), 33,6k (V.34): 115200 bps
- 14,4k (V.32bis), mit V.42bis Datenkompression: 57600 bps
- 9,6k (V.32), mit V.42bis Datenkompression: 38400 bps
- langsamer als 9,6k (V.32): die Schnittstellengeschwindigkeit sollte der Modemgeschwindigkeit entsprechen.

Alle oben angegebenen Geschwindigkeiten können die V.42bis Datenkompression und die V.42 Fehlererkennung verwenden. Wird keine Datenkompression verwendet, kann die Geschwindigkeit auch kleiner als angegeben eingestellt werden, so lange sie über der Modemgeschwindigkeit liegt.

13 Terminalprogramme und Hilfsprogramme

Für den Zugang zum Internet wird am häufigsten PPP verwendet, aber Sie benötigen auch ein Wahlprogramm, welches mit PPP zusammenarbeitet. Ein Wahlprogramm wählt eine Telefonnummmer. Wenn auf der anderen Seite abgehoben wird, passieren drei Dinge: auf beiden Seiten wird PPP gestartet, und Sie werden (oft automatisch) eingeloggt. Die genaue Reihenfolge kann variieren. Wahlprogramme für PPP sind z.B. `wvdial`, `chat`-Skripte, `kppp` und gnome-ppp.

Mit einem Terminalprogramm können Sie auch direkt nach draussen wählen, z.B. zu öffentlichen Mailboxen. Dies ist aber kein Internetzugang. Das beliebteste Terminalprogramm ist `minicom`, gefolgt von `seyon` (unter X11) und `kermit`. Ein Terminalprogramm kann u.U. auch als Wahlprogramm für PPP verwendet werden, das ist aber nicht der Zweck, für den Terminalprogramme geschrieben wurden.

13.1 Vergleich von minicom und kermit

`minicom` ist lediglich ein Terminalprogramm, während `kermit` zusätzlich ein Protokoll zur Dateiübertragung implementiert. Aber man kann das `kermit`-Protokoll von `minicom` aus nutzen (vorausgesetzt, `kermit` ist auf dem PC installiert). `minicom` wird über Menüs gesteuert, `kermit` über Kommandozeilenparameter (bzw. interaktiv über den `kermit`-Prompt). Während das `kermit`-Programm selbst freie Software ist, trifft dies auf die Dokumentation nicht zu. Es gibt kein detailliertes Handbuch. Es wird empfohlen, ein entsprechendes Buch zu kaufen und als Handbuch zu verwenden. Aber `kermit` verfügt über eine interaktive Online-Hilfe, die zwar vollständig ist, aber keine einführenden Hinweise für den Anfänger enthält. Die Befehle können auch von einer Skript-Datei gelesen werden, so dass sie nicht immer wieder von neuem eingegeben werden müssen. `kermit` als Terminalprogramm ist mächtiger als `minicom`.

Die Dokumentation zu `minicom` ist zwar frei erhältlich, aber weniger ausführlich als die `kermit`-Dokumentation. Da eine Genehmigung erforderlich ist, um `kermit` zusammen mit einer kommerziellen Distribution auszuliefern, und da die Dokumenation nicht vollständig frei ist, ist `kermit` bei einigen Distributionen nicht mit enthalten. Meiner Meinung nach ist es einfacher, `minicom` zu verwenden.

13.2 Auflistung verschiedener Terminalprogramme

Es folgt eine Liste einiger Terminalprogramme, die über FTP erhältlich sind, falls sie in Ihrer Distribution nicht enthalten sind. Für Vergleiche zwischen den Terminalprogrammen wäre ich dankbar. Sind die weniger verbreiteten Terminalprogramme eventuell überflüssig?

13.2.1 Weniger verbreitete Terminalprogramme

- `ecu` - ein Terminalprogramm
- `pcomm` - `procomm`-ähnliches Terminalprogramm mit ZModem-Protokoll
- `xc` - das xcomm Kommunikations-Paket

13.2.2 Weit verbreitete Kommunikationsprogramme

- `minicom` - `Telix`-ähnliches Terminalprogramm. Unterstützt Skripts, das ZModem-Protokoll, das Kermit-Protokoll.
- C-Kermit - portabel, Skript-fähig, serielle und TCP/IP-Kommunikation, inklusive Dateitransfer, Zeichencodeumsetzung und ZModem-Protokoll.
- `seyon` - Terminalprogramm unter X11

13.2.3 Fax

- `efax` - ein kleines Faxprogramm
- `hylafax` - ein umfangreiches Faxprogramm, Client-Server basiert
- `mgetty+fax` - stellt Funktionen für FAX und Einwahlmöglichkeiten zur Verfügung

13.2.4 Anrufbeantworter-Software

- mvm ist eine minimale Voice Mail Software für Linux. Das Programm ist zu beziehen unter:

  `http://www-internal.alphabet.ch/~schaefer/mvm/`

- `vgetty` ist eine Erweiterung zu `mgetty`, und kann in Verbindung mit einigen Modemtypen als Anrufbeantworter-Software eingesetzt werden. Es sollte in den neueren `mgetty`-Paketen enthalten sein.

13.2.5 Einwahlserver (unter Verwendung von getty)

- `mgetty+fax` unterstützt Modems und ist gut dokumentiert (außer die Anrufbeantworter-Funktionen, Stand 1999). Es unterstützt auch FAX Funktionen und stellt eine Alternative zu `uugetty` dar. Es beinhaltet auch Anrufbeantworter-Funktionalität; dabei wird `vgetty` verwendet. Siehe auch 10.2.1 (mgetty).
- Auch `uugetty` läßt sich mit Modems einsetzen. Es ist Bestandteil des `getty_ps` Pakets. Siehe auch 11 (uugetty-Konfiguration).

13.2.6 Andere Programme

- Callback-Programme unterstützen folgende Funktion: Sie wählen ein anderes Modem an und dieses Modem legt nach erfolgreichem Verbindungsaufbau auf und ruft Sie zurück, um Ihnen z.B. Telefongebühren zu sparen.
- `SLiRP` und `term` laufen mit einer normlen Shell-Benutzerkennung und bieten, änlich wie PPP, eine Login-Möglichkeit auf einem anderen Computer an. Siehe auch 13.3 (term und SLiRP).
- `ZyXEL` ist ein Administrationsprogramm für ZyXEL U-1496 Modems. Es verwaltet ein- und abgehende Anrufe, Rückruffunktion zur Systemsicherheit, Faxfunktionen und Anrufbeantworter-Funktionalität.
- SLIP und PPP Software ist zu finden unter:

  `ftp://metalab.unc.edu/pub/Linux/system/network/serial`
- Andere Programme für die serielle Schnittstelle sind zu finden unter
  - `ftp://metalab.unc.edu/pub/Linux/system/serial`
  - `ftp://metalab.unc.edu/pub/Linux/apps/serialcomm`

  oder auf einer der vielen Server, die die Seiten in Kopie bereithalten (Spiegelserver, engl. »Mirror«).

13.3 SLiRP und term

`SLiRP` und `term` sind Programme, die dann nützlich sind, wenn Sie auf dem angewählten Rechner nur eine normale Benutzerkennung haben, aber einen ähnlichen Dienst wie PPP nutzen wollen, ohne dazu authorisiert zu sein (z.B. weil Sie dafür nicht extra bezahlen wollen, etc). `SLiRP` ist beliebter als `term`, welches schon fast überflüssig ist.

Um `SLiRP` zu nutzen, müssen Sie es unter Ihrer Benutzerkennung auf dem entfernten Rechner installieren. Auf dem entfernten PC ist dann `SLiRP` aktiv, auf dem lokalen PC dagegen PPP. Sie haben nun eine PPP Verbindung und können z.B. einen Browser wie Netscape auf dem lokalen PC verwenden. Es können sich aber einige Probleme ergeben, da `SLiRP` nicht genauso gut ist wie ein richtiger PPP Zugang. Manchmal wird `SLiRP` auch angeboten, weil es IP Adressen spart (Sie brauchen keine IP Adresse, wenn Sie `SLiRP` verwenden).

`term` funktioniert ähnlich wie `SLiRP`, nur dass `term` auf beiden Rechnern, d.h. dem entfernten und dem lokalen PC, läuft. Es wird auch kein PPP verwendet, da `term` über sein eigenes Protokoll verfügt. Um `term` auf Ihrem PC zu nutzen benötigen Sie für `term` kompatible Programme, z.B. eine `term`-Version von `ftp`, um FTP zu verwenden etc. Daher ist es einfacher, `SLiRP` zu verwenden, da die normalen Programme (z.B. das ganz normale `ftp`) mit `SLiRP` funktionieren. Es gibt auch ein HOWTO für `term`.

14 Fehlerursachen erkennen und beheben

14.1 Mit meinem 56k Modem erreiche ich keine 56k

Die Qualität der Telefonleitung muss schon sehr gut sein, um auch nur in die Nähe von 56k zu gelangen. Einige Telefonleitungen sind so schlecht, dass die erreichbare Geschwindigkeit weit unterhalb von 56k liegt: z.B. bei 28,8k oder noch weiter darunter. Manchmal können zusätzliche Telefonapparate, die an die selbe Leitung angeschlossen sind, Probleme verursachen.

14.2 Dateiübertragungen werden abgebrochen oder sind langsam

Eventuell ist die Flusskontrolle (sowohl am PC als auch am Modem) nicht aktiviert. Wenn Sie eine hohe DTE Geschwindigkeit (z.B. 115,2 kbps) eingestellt haben, dann kann der Datenfluss von Ihrem Modem zum PC gut funktionieren, aber in der anderen Übertragungsrichtung bildet die Telefonleitung einen Flaschenhals. Das Ergebnis sind viele Übertragungsfehler und daher müssen viele Datenpakete wiederholt gesendet werden. Es kann daher sehr lange dauern, eine Datei zu senden. Manchmal ist eine Dateiübertragung auch gänzlich unmöglich. Wenn Sie große, nicht komprimierte Dateien oder Internetseiten herunterladen und Ihr Modem verwendet Datenkompression, oder wenn Sie eine geringe DTE Geschwindigkeit eingestellt haben, dann kann das Herunterladen auch abgebrochen werden, weil die Flusskontrolle fehlt.

14.3 Bei Verwendung von getty erhalte ich die Meldung »line NNN of inittab invalid«

Überprüfen Sie, dass die verwendete Syntax für Ihre Version von `init` stimmt. Unterschiedliche Version von `init` setzen auch eine unterschiedliche Syntax in der Datei `/etc/inittab` voraus. Überprüfen Sie auch, ob die Syntax für Ihre Version von `getty` richtig ist.

14.4 Beim Verbindungsaufbau erscheint die Meldung »/dev/ttySN: Device or resource busy«

Dieses Problem kann auftauchen, wenn die Steuerung der Signale DCD oder DTR nicht richtig implementiert ist. Das DCD Signal sollte nur dann gesetzt sein, wenn eine Verbindung aufgebaut ist (z.B. wenn sich jemand eingewählt hat), und nicht wenn `getty` den entsprechenden Port überwacht. Überprüfen Sie, dass das Modem so konfiguriert ist, dass es nur dann das DCD signalisiert, wenn eine Verbindung besteht. Das DTR Signal sollte immer dann gesetzt sein, wenn ein Programm den Port überwacht, z.B. `getty`, `kermit` oder ein anderes Terminalprogramm.

Eine andere häufige Ursache für einen »device busy«-Fehler besteht in der fehlerhaften Zuordnung eines IRQ an einen seriellen Port, der bereits von einem anderen Gerät verwendet wird. Wenn die Peripheriegeräte initialisiert werden, holen sie sich von Linux die Berechtigung, ihren eingestellten Hardware Interrupt zu verwenden. Linux verwaltet die Informationen, welcher Interrupt von welcher Hardware verwendet wird, und wenn der Interrupt der seriellen Schnittstelle bereits vergeben wurde, kann sich die serielle Schnittstelle nicht richtig initialisieren. Dabei hat sie kaum die Möglichkeit, Ihnen mitzuteilen, dass die Intitialisierung fehlgeschlagen ist, ausser bei dem Versuch, sie zu verwenden, indem sie die »device busy« Meldung ausgibt. Überprüfen Sie die IRQs aller Geräte (serielle Schnittstelle, Ethernet Adapter, SCSI Controller etc). Achten Sie auf IRQ Konflikte.

14.5 Es erscheint die Meldung »Id S3 respawning too fast: disabled for 5 minutes«

Id S3 ist hier nur ein beispielhafter Wert. Suchen Sie in der Datei `/etc/inittab` nach der Zeile, die mit »S3« beginnt. Diese Zeile verursacht das Problem. Überprüfen Sie diese Zeile auf korrekte Syntax, und überprüfen Sie, dass das Gerät ttyS3 existiert und gefunden werden kann.

Stellen Sie auch sicher, dass Ihr Modem richtig konfiguriert ist. Überprüfen Sie die Einstellung der Register »E« und »Q«. Das Problem kann auftreten, wenn das Modem mit getty kommuniziert.

Wenn Sie `uugetty` verwenden, überprüfen Sie die Syntax der Datei `/etc/gettydefs` mit dem Befehl

```
getty -c /etc/gettydefs
```

Dieses Problem kann auch auftauchen, wenn die Initialisierung von `uugetty` fehlschlägt; siehe auch 14.7 (uugetty funktioniert noch immer nicht).

14.6 Mein Modem spielt nach dem Verbindungsende verrückt, oder uugetty startet nicht wieder

Das kann passieren, falls Ihr Modem sich nicht zurücksetzt, wenn das DTR Signal verschwindet. Greg Hankins hat beobachtet, wie die RD und SD LEDs in dieser Situation verrückt gespielt haben. Sie müssen Ihr Modem zurücksetzen. Bei den meisten Hayes-Kompatiblen Modems geht das mit dem Befehl »&D3«, aber bei einem USR Courier Modem brauchen Sie den Befehl »&D2« und »S13=1«. Schlagen Sie den Reset-Befehl im Modem-Handbuch nach.

14.7 uugetty funktioniert noch immer nicht

Zu `getty_ps` gibt es eine »Debug«-Option. Erweitern Sie die Konfigurationsdatei `/etc/conf.{uu}getty.ttyS` und fügen Sie die Option »DEBUG=NNN« hinzu. Dabei bedeutet »NNN« eine der folgenden Ziffernkombinationen, je nachdem, was Sie untersuchen wollen:

```
D_OPT   001          Einstellung der Optionen
D_DEF   002          Standard-Dateiverarbeitung
D_UTMP  004          utmp/wtmp Verarbeitung
D_INIT  010          Leitungsinitialisierung (INIT)
D_GTAB  020          gettytab Dateiverarbeitung
D_RUN   040          andere Diagnosehilfsmittel
D_RB    100          Fehlersuche über Rückruffunktion
D_LOCK  200          uugetty Sperrdatei Verarbeitung
D_SCH   400          Verarbeitung der Zeitzuordnung
D_ALL   777          alles zusammen
```

Setzen Sie zu Anfang die Option »DEBUG=010«.

Falls Sie `syslogd` verwenden, werden Hinweise zur Fehlersuche in die Protokolldateien geschrieben. Falls Sie `syslogd` nicht verwenden, werden Debugging-Informationen von `getty` in der Datei `/tmp/getty::ttyS`*N* und von `uugetty` in `/tmp/uugetty::ttyS`*N* und in `/var/adm/getty.log` protokolliert. Sehen Sie sich diese Informationen an und finden Sie heraus, was passiert. Wahrscheinlich müssen Sie einige Parameter in den Konfigurationsdateien anpassen und Ihr Modem umkonfigurieren.

Sie können es ja auch mal mit `mgetty` probieren. Manchmal hat man damit mehr Glück.

14.8 Mein Modem ist zwar physisch vorhanden, kann aber nicht gefunden werden

Falls Sie die seriellen Ports kannten, die vor der Installation des internen Modems existierten, dann besteht das Problem darin, den neuen seriellen Port zu finden. Dies wird im nächsten Abschnitt behandelt. In diesem Abschnitt soll es darum gehen, wie man herausfindet, an welchem seriellen Port das Modem angeschlossen ist.

Es gibt ein Programm namens `wvdialconf`, das die üblicherweise verwendeten seriellen Ports auf ein angeschlossenes Modem überprüft. Geben Sie einfach ein:

```
wvdialconf <ein-neuer-Dateiname>
```

Die neu angelegte Datei ist eine Konfigurationsdatei für `wvdial`, sie benötigen diese Datei, wenn Sie `wvdial` einsetzen wollen; siehe auch 8.3 (Was ist wvdialconf?).

Vielleicht liegt Ihr Problem darin begründet, dass Sie ein Winmodem verwenden, welches unter Linux nicht verwendet werden kann. Siehe auch 6.4 (Interne Modems, die zu meiden sind). `setserial` kann zwar verwendet werden, um Informationen über serielle Ports herauszufinden, aber nicht, ob an einem seriellen Port auch ein Modem angeschlossen ist. Sie sollten es daher zunächst mit `wvdialconf` versuchen.

Eine andere Möglichkeit um herauszufinden, ob an einem Port ein Modem angschlossen ist, besteht darin, `minicom` auf diesem Port zu starten (mit dem Befehl »^A0« gelangen Sie in das Menü mit den Konfigurationseinstellungen). Geben Sie »AT« ein, und Sie sollten ein »OK« zurückerhalten (bzw. eine »0«, falls das Modem auf numerische Antwortcodes eingestellt ist). Wenn es einige Sekunden dauert, bis Sie eine Antwort erhalten oder wenn sich lediglich der Cursor um eine Zeile nach unten bewegt, sehen Sie im Abschnitt 14.10 (Text erscheint auf dem Bildschirm langsam und nach langer Verzögerung) nach.

14.9 Der serielle Port ist physisch vorhanden, kann aber nicht gefunden werden

Überprüfen Sie die BIOS Einstellungen und die Meldungen des BIOS. Wenn es sich um einen seriellen PnP-Port für den ISA Bus handelt, können Sie es mit dem Befehl

```
pnpdump --dumpregs
```

versuchen und/oder im *Plug-and-Play HOWTO* nachsehen. Beim PCI Bus sollten Sie einen Blick in die Datei `/proc/pci` werfen. Sie können mit Hilfe von `setserial` ein Probing durchführen, siehe auch 8.1.2 (Probing). Falls keinerlei Daten über den Port ausgetauscht werden können, obwohl er vorhanden ist, liegt es eventuell an einem falschen Interrupt, siehe auch 14.10 (Text erscheint auf dem Bildschirm langsam und nach langer Verzögerung).

14.10 Text erscheint auf dem Bildschirm langsam und nach langer Verzögerung

Wahrscheinlich liegt die Ursache in einem falsch eingestellten Interrupt oder einem IRQ Konflikt. Im folgenden sind einige der Symptome beschrieben, die bei dem Versuch auftreten, zum ersten Mal ein Modem, ein Terminal oder einen Drucker zu verwenden.

- In einigen Fällen geben Sie etwas ein, aber es erscheint erst nach einigen Sekunden auf dem Bildschirm.
- Oder es wird nur das zuletzt eingegebene Zeichen ausgegeben (und es kann sich dabei um ein unsichtbares Zeichen handeln, z.B. das <return>-Zeichen, so dass sich der Cursor um eine Zeile nach unten bewegt).
- In anderen Fällen ist nur eine ca. 16 Zeichen lange Zeichenfolge auf dem Bildschirm zu sehen, obwohl eigentlich eine ganze Menge an Daten erscheinen sollte. Dann tritt ein lange Pause von etlichen Sekunden ein, bevor die nächsten ca. 16 Zeichen angezeigt werden.
- Vielleicht erhalten Sie auch »input overrun« Fehlermeldungen oder Sie finden Hinweise auf solche Meldungen in den Protokolldateien.

Weitere Informationen über die Symptome und Gründe für diese Verhaltensweisen finden Sie im (englischen) *Serial HOWTO* im Kapitel »Interrupt Problem Details«.

Um sicherzugehen, dass es sich wirklich um ein Interrupt-Problem handelt, können Sie den IRQ mit Hilfe von `setserial` auf Null einstellen. Der Gerätetreiber wird dadurch angewiesen, statt eines Interrupts das sogenannte Polling zu verwenden (mit Hilfe von Interrupts kann ein Gerät die CPU auf sich aufmerksam machen, wenn z.B. Daten zum Lesen anstehen, während bei der Polling-Methode die CPU von sich aus regelmäßig am Gerät prüft, ob Daten vorhanden sind. In der Alltagswelt hat z.B. Ihre Türklingel die Funktion eines Interrupt. Hätten Sie keine Türklingel und wollen keinen Besucher verpassen, müßten Sie alle paar Sekunden zur Türe gehen und nachsehen, ob jemand davorsteht. Klar, dass Polling ziemlich ineffektiv ist). Wenn das Ihr Problem behebt, liegt die Ursache in einem falschen Interrupt. Sie sollten auf jeden Fall versuchen, die IRQs richtig zu konfigurieren, denn das Polling benötigt extrem viel Resourcen des Rechners.

Einen Interrupt-Konflikt zu finden, ist nicht immer einfach. Auch ein Blick in das `/proc`-Verzeichnis kann u.U. täuschen. Stellen Sie sicher, dass kein IRQ unerlaubterweise gemeinsam genutzt wird. Überprüfen Sie alle Steckkarten (serielle Karte, Netzwerkkarte, SCSI, usw). Überprüfen Sie die Jumper- (oder PnP-)Einstellungen und die `setserial`-Parameter für alle seriellen Geräte. Überprüfen Sie auch die Dateien `/proc/ioports`, `/proc/interrupts` und `/proc/pci` auf Interrupt-Konflikte. Weitere Informationen zu diesem Thema finden Sie im (englischen) *Serial HOWTO*, Kapitel »Interrupt Problem Detail«. Informationen bei Verwendung von PnP-Hardware finden Sie auch im *Plug-and-Play HOWTO*.

14.11 Der Startbildschirm zeigt falsche IRQ-Werte für die seriellen Ports

Beim Systemstart versucht Linux nicht, die richtige Belegung der IRQs herauszufinden. Beim Laden des Gerätetreiber-Moduls für den seriellen Port wird nur das serielle Gerät überprüft. Die zu diesem Zeitpunkt ausgegebenen Meldungen können Sie bezüglich des IRQs ignorieren, weil der Gerätetreiber immer von der Standardbelegung der IRQs ausgeht. Die automatische Erkennung von IRQs ist unzuverlässig und schlägt häufig fehl. Wenn aber `setserial` von einem Startskript aufgerufen wird, werden die IRQs geändert und die neuen und hoffentlich richtigen Werte werden auf dem Startbildschirm ausgegeben. Wenn also ein falscher IRQ bei einer späteren Ausgabe nicht korrigiert wird, haben Sie ein Problem.

Obwohl ich den IRQ für `ttyS2` auf den IRQ 5 eingestellt habe, sehe ich

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

auf dem Bildschirm, wenn Linux bootet (bei älteren Kernelversionen kann »ttyS02« evtl. als »tty02« angezeigt werden). Sie müssen `setserial` verwenden, um Linux über den von Ihnen verwendeten IRQ zu informieren.

14.12 »Cannot open /dev/ttyS?: Permission denied«

Überprüfen Sie die Dateirechte für die seriellen Ports mit dem Befehl

```
ls -l /dev/ttyS?
```

Sie benötigen Lese- und Schreibberechtigung. Die Rechte sollten in Spalte 8 und 9 auf »rw-« gesetzt sein, dies ermöglicht jedem den Zugriff auf den Port. Mit dem Befehl `chmod` können Sie die Rechte verändern. Häufig sind Lese- und Schreibrecht auch nur für eine bestimmte Benutzergruppe z.B »uucp« gesetzt; Sie müssen dann mit Ihrer Benutzerkennung Mitglied dieser Gruppe sein, um Zugriff auf die seriellen Ports zu erhalten.

14.13 Fehlermeldung für ttySx: »Operation not supported by device«

Dies bedeutet, dass eine von `setserial`, `stty`, etc. ausgelöste Operation nicht durchgeführt werden konnte, weil dies der Kernel nicht unterstützt. Früher war dies oft der Fall, wenn das Modul für den seriellen Gerätetreiber nicht geladen war. Aber bei Verwendung von PnP-Karten bedeutet die Meldung wahrscheinlich, dass sich an der Adresse, von der der Gerätetreiber (und `setserial`) ausgeht, kein Modem (oder ein anderes serielles Gerät) befindet. Befehle, die zu dieser Adresse übertragen werden, können dann natürlich nicht ausgeführt werden. Siehe auch 4.4.2 (Wie ist die Hardware des seriellen Ports eingestellt?).

Falls das Modul für den seriellen Gerätetreiber nicht geladen wurde, aber `lsmod` anzeigt, dass dieses Modul geladen ist, dann kann es sein, dass das Modul zum Zeitpunkt der Fehlermeldung nicht geladen war. In den meisten Fällen wird dieses Modul automatisch geladen, wenn es benötigt wird. Um das Laden des Moduls zu erzwingen, kann es in die Datei `/etc/modules.conf` oder `/etc/modules` eingetragen werden. Das Modul selbst sollte sich unter `/lib/modules/.../misc/serial.o` befinden.

14.14 Fehlermeldung »Cannot create lockfile. Sorry«

Wenn ein Port durch ein Programm geöffnet wird, wird zugleich eine Sperrdatei im Verzeichnis `/var/lock/` angelegt. Sind die Rechte für dieses Verzeichnis falsch eingestellt, kann keine Sperrdatei erzeugt werden. Mit dem Befehl

```
ls -ld /var/lock
```

können Sie die Berechtigungen anzeigen lassen, normalerweise sollte »drwxrwxrwx« eingestellt sein. Mit dem Befehl `chmod` können Sie die Berechtigungen ändern. Und natürlich kann auch keine Sperrdatei angelegt werden, wenn das Lock-Verzeichnis nicht existiert. Weitere Informationen über Sperrdateien finden Sie im *Serial HOWTO* im Kapitel »What Are Lock Files« (in Englisch).

14.15 Hilfreiche Software

- `modemstat` und `setserial` zeigen den aktuellen Zustand verschiedener Signalleitungen des Modems an (wie z.B. DTR, CTS, etc.)
- `irqtune` kann an die von seriellen Ports genutzten Interrupts eine höhere Priorität vergeben, um die Geschwindigkeit zu verbessern.
- `hdparm` kann helfen, indem Festplattenzugriffe optimiert werden.

15 Flash Upgrades

Für viele Modems besteht die Möglichkeit eines Upgrades (engl., in deutsch etwa: Ausbau der Leistungsmerkmale), indem ein neues Programm (das Upgrade-Programm), das z.B. aus dem Internet erhältlich ist, im Flash Speicher des Modems abgespeichert wird. Indem dieses »Programm« vom PC über den seriellen Port zum Modem gesendet wird, wird es vom Modem im nicht-flüchtigen Speicher abgelegt; d.h. es befindet sich auch noch dort, nachdem die Spannungsversorgung unterbrochen wurde. Die Anleitungen zur Installationen eines Flash Upgrades beziehen sich gewöhnlich auf die Arbeit unter Windows, Sie müssen also herausfinden, wie das Gleiche unter Linux bewerkstelligt werden kann (wenn Sie das Upgrade nicht unter Windows ausführen). Die Übertragung des Programms zum Modem wird häufig auch mit »Download« bezeichnet.

An dieser Stelle ein Aufruf zur Mitarbeit: wenn es sich bei diesem HOWTO um die aktuellste Version handelt (siehe 1.4 (Neue Versionen dieses HOWTOs)) und Sie Erfahrung mit der Installation eines Flash Upgrades haben, so teilen Sie mir diese bitte mit, falls Sie glauben, dass diese Informationen auch für andere interessant sein könnten.

Im folgenden wird das allgemeine Vorgehen beschrieben: Zunächst gibt es eventuell einen Befehl, mit dem Sie dem Modem mitteilen, dass ein Flash ROM Upgrade eingeleitet wird. In einem Fall war dies der Befehl »AT**«. Sie können diesen Befehl mit Hilfe eines Terminalprogramms (z.B. `minicom`) eingeben. Geben Sie zuerst »AT <Return>« ein, um zu sehen, ob das Modem mit »OK« antwortet.

Als nächstes müssen Sie eine Datei (manchmal sind es auch zwei Dateien) direkt zum Modem übertragen. Terminalprogramme (wie z.B. `minicom`) verwenden häufig das ZModem- oder Kermit-Protokoll, um Dateien zum Modem (und darüber hinaus) zu übertragen. Aber diese Protokolle paketieren die Daten und versehen die einzelnen Pakete mit einem Vorspann, wir wollen aber genau diese Datei zum Modem übertragen und keine veränderte Version. Aber `Kermit` verfügt über ein »transmit« Befehl, der die Datei direkt und ohne Paketierung überträgt. `minicom` besitzt dieses Feature leider nicht (Stand: 1998).

Eine andere Möglichkeit, eine Datei direkt zu übertragen, besteht darin, vom Terminalprogramm direkt in eine Shell zu wechseln (bei `minicom` ist dies der Befehl »^AJ«) und dann einzugeben:

```
cat upgrade_file_name > /dev/ttyS2
```

(setzen Sie statt `ttyS2` Ihren seriellen Port ein). Wechseln Sie zurück zum Terminalprogramm (geben Sie `fg` an der Eingabeaufforderung in `minicom` ein) um zu sehen, was passiert.

Hier ist ein Beispiel für ein bestimmtes Modem von Rockwell (»C-a« ist »^A«):

- Starten Sie minicom.
- Geben Sie AT** ein; Ausgabe: »Download initiated ..«
- »C-a J«
- `cat FLASH.S37 > /dev/modem`
- fg: Ausgabe: »Download flash code ..«
- »C-a J«
- `cat 283P1722.S37 > /dev/modem`
- fg: Ausgabe: »Device successfully programmed«

16 Weitere Informationsquellen

16.1 Verschiedenes

- Manual Pages für `agetty(8)`, `getty(1m)`, `gettydefs(5)`, `init(1)`, `isapnp(8)`, `login(1)`, `mgetty(8)`,`setserial(8)`.
- Das Handbuch zu Ihrem Modem. Einige Modems werden leider ganz ohne Handbuch geliefert.
- Die »Serial Suite« von Vern Hoxie, erhältlich unter

  `ftp://scicom.alphacdc.com/pub/Linux.`

  Dies ist eine Sammlung von Tipps und Hinweisen zur Haltung und Pflege eines seriellen Ports unter Linux, darunter auch einige einfache Programme.
- Die »Linux Serial Mailing Liste« (in Englisch). Um sich in die Mailing Liste einzutragen, senden Sie eine eMail an

```
majordomo@vger.rutger.edu
```

mit dem Text

```
subscribe linux-serial
```

Wenn Sie als Text `help` senden, erhalten Sie eine eMail mit Hilfe-Informationen. Dieser Server führt ausserdem viele andere Linux Mailing Listen. Wenn Sie eine Liste aller Mailing Listen erhalten wollen, senden Sie eine eMail mit dem Text `lists`.

16.2 HOWTOs

Kabelmodem mini-HOWTO

ISDN HOWTO

Beschreibt Treiber für ISDN-»Modems«.

Modems HOWTO

Eine französische Modem HOWTO, die für diese HOWTO nicht verwendet wurde.

NET-3 HOWTO

Enthält alles über Netzwerke; enthält auch Informationen zu den Protokollen SLIP, CSLIP und PPP.

PPP HOWTO

Bietet Hilfe zur Verwendung von PPP, enthält auch Hinweise zur richtigen Modemkonfiguration.

Serial HOWTO

Bietet Informationen über serielle Multiport-Karten, die zum Anschluss von mehreren Terminals oder Modems verwendet werden. Der serielle Port wird detaillierter besprochen als in diesem HOWTO.

Serial-Programming HOWTO

Behandelt einige Aspekte der Programmierung des seriellen Ports.

Text-Terminal HOWTO

Behandelt auch den Verbindungsaufbau über Modems.

UUCP HOWTO

Bietet Informationen zur Konfiguration von UUCP.

16.3 Newsgroups (in englischer Sprache)

`comp.os.linux.answers`

FAQs, HOWTOs, READMEs, usw. über Linux.

`comp.os.linux.hardware`

Diskutiert werden Fragen bezüglich der Hardware-Kompatibilität mit dem Linux Betriebssystem.

`comp.os.linux.setup`

Diskutiert werden Fragen bezüglich der Installation und Systemverwaltung von Linux.

`comp.dcom.modems`

Modems für alle Betriebssysteme.

16.4 Seiten im Internet

- Liste von Modems, die unter Linux funktionieren/nicht funktionieren:

  `http://www.idir.net/~gromitkc/winmodem.html`

- Informationen über PCI-Modems, Home Page der Linux-Treiber für serielle Geräte:

  `http://serial.sourceforge.net`

- Technische Referenz der Hayes AT Modem-Befehle:

  `http://www-dcg.fnal.gov/NET/HYSTRM20.TXT`

- AT Befehle und Modem-Register für analoge Modems (Cisco)

  `http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/analogfw/analogat`

- Steuerung des Modems mit AT Befehlen:

  `http://www.zoltrix.com/modem/USEMODEM.HTM`

- Modem FAQs:

  `http://modemfaq.home.att.net/`

- Curts Seite über Hochgeschwindigkeits-Modems:

  `http://www.teleport.com/~curt/modems.html`

- Viele Informationen über 56K-Modems:

  `http://808hi.com/56k/`

- Links zu verschiedenen Modem Herstellern:

  `http://www.56k.com/links/Modem_Manufacturers/`

- Weitere Links zu Modem Herstellern:

  `http://modmes.rosenet.net/`

- Modems durch die FCC ID identifizieren:

  `http://www.sbsdirect.com/fccenter.html`

17 Anhang A: Wie ein Modem technisch funktioniert (nicht fertiggestellt)

17.1 Details zur Modulation

17.1.1 Einführung in die Technik der Modulation

Unter Modulation versteht man die Konvertierung eines digitalen Signals, das aus binären Bits (0 oder 1) besteht, in ein analoges Signal, etwa eine Sinuskurve. Das modulierte Signal besteht aus einem reinen sinusförmigen Trägersignal (engl. »Carrier«), das verändert wird, um Informationen zu übertragen. Ein reines Trägersignal, das sich weder in der Frequenz noch in der Amplitude verändert, überträgt auch keinerlei Informationen (ausser, dass ein Trägersignal vorhanden ist). Um das Trägersignal Informationen übertragen zu lassen, verändern (oder »modulieren«) wir es. Es gibt drei wesentliche Modulationstypen: Frequenzmodulation, Amplitudenmodulation und Phasenmodulation. Diese Modulationstypen werden in den folgenden Abschnitten erklärt.

17.1.2 Frequenzmodulation

Die einfachste Modulationsmethode ist die Frequenzmodulation. Die Frequenz wird in der Einheit »Schwingungen pro Sekunde« angegeben. Es ist die Anzahl, mit der sich z.B. eine Sinus-Schwingung pro Sekunde wiederholt. Die Frequenz gibt an, wie oft pro Sekunde der maximale Ausschlag erreicht wird. Die Einheit der Frequenz ist »Hertz« (abgekürzt Hz, nach dem deutschen Physiker Heinrich Hertz).

Ein einfaches Beispiel für Frequenzmodulation bestünde etwa in der Festlegung, dass eine bestimmte Frequenz eine 0 bedeutet, und eine andere Frequenz bedeutet eine 1. Z.B. bedeutete bei einigen (inzwischen überholten) 300 Baud-Modems eine Frequenz von 1070 Hz ein binäre 0, während eine Frequenz von 1270 Hz eine binäre 1 darstellte (»Frequency Shift Keying«). Anstelle von zwei möglichen Frequenzen könnten aber auch mehr verwendet werden, um mehr Informationen zu übertragen. Würden wir vier verschiedene Frequenzen verwenden (nennen wir sie A, B, C und D), könnte jede Frequenz ein Bitpaar darstellen. Um z.B. 00 zu senden, könnte man Frequenz A verwenden, für 01 die Frequenz B, für 10 C und schließlich für 11 die Frequenz D. In gleicher Weise könnten wir mit jedem Wechsel der Frequenz 3 Bits an Information übertragen, wenn wir 8 verschiedene Frequenzen verwenden würden. Jedesmal, wenn wir die Anzahl der möglichen Frequenzen verdoppeln, können wir die Anzahl der übertragenen Bits um eins erhöhen.

17.1.3 Amplitudenmodulation

Wenn man das obige Beispiel der Frequenzmodulation verstanden hat, bei dem mehrere Bits mit einem einzigen Frequenzwechsel übertragen werden, ist es einfacher, sowohl die Amplituden- als auch die Phasenmodulation zu verstehen. Analog zur Änderung der Frequenz der Sinus-Trägerschwinung verändert man bei der Amplitudenmodulation die Amplitude (oder die Höhe der Spannung). Als einfachste Möglichkeit könnten nur zwei mögliche Amplituden erlaubt sein, wobei die eine ein 0-Bit, die andere ein 1-Bit darstellt. Wie im Fall der Frequenzmodulation erklärt, kann mehr Information übertragen werden, wenn mehrere erlaubte Amplituden zur Verfügung stehen.

17.1.4 Phasenmodulation

Um die Phase einer Sinusschwingung zu einem bestimmten Zeitpunkt zu verändern, stoppen wir das Senden der alten Sinusschwingung und beginnen gleichzeitig mit dem Senden einer neuen Schwingung der selben Frequenz und Amplitude. Wenn wir die neue Sinusschwingung mit der gleichen Spannung (und der gleichen zeitlichen Änderungsrate, d.h. mathematisch gesprochen mit dem gleichen Wert der 1. Ableitung) starten, mit der die alte Schwingung gestoppt wurde, würde sich keine Änderung der Phase (oder eine andere messbare Änderung) ergeben. Aber nehmen wir mal an, wir würde die neue Sinusschwingung an einem anderen Punkt der Sinuskurve starten. Dann würde wahrscheinlich ein plötzlicher Spannungssprung zu dem Zeitpunkt auftreten, an dem die alte Schwingung gestoppt wird und die neue beginnt. Dies ist eine Phasenverschiebung, die in Grad gemessen wird. Eine Phasenverschiebung von 0 Grad (oder 360 Grad) bedeutet keinerlei Änderung, während eine Phasenverschiebung von 180 Grad die Spannung (und Steigung) der Sinusschwingung invertiert. Anders ausgedrückt, bedeutet eine 180 Grad Phasenverschiebung, dass eine halbe Periode (180 Grad) übersprungen wird. Natürlich könnten wir auch 90 Grad oder 135 Grad überspringen, usw. Wie im Beispiel der Frequenzmodulation erklärt, kann eine Phasenverschiebung umso mehr Bits repräsentierten, je mehr mögliche Werte für die Phasenverschiebung zur Verfügung stehen.

17.1.5 Kombinationen der Modulationsarten

Statt eine der drei möglichen Modulationsarten zu auszuwählen, könnten wir auch mehrere Methoden kombinieren. Nehmen wir an, wir hätten 256 mögliche Frequenzen zur Verfügung und könnten daher ein Byte (8 Bits) bei jedem Frequenzwechsel übertragen (2 hoch 8 ergibt 256). Nehmem wir weiter an, dass wir 256 mögliche Amplituden zu Verfügung haben, so dass jeder Wechsel der Amplitude ebenfalls ein Byte repräsentiert. Nehmen wir auch noch an, dass 256 mögliche Phasenverschiebungen erlaubt sind. Dann könnten wir zu einem Zeitpunkt alle 3 Werte ändern. Bei jedem solchen Übergang würden wir 3 Bytes an Information übertragen.

Tatsächlich funktioniert keine der heute üblichen Modulationsmethoden auf diese Weise. Es würde zu lange dauern, um alle 3 Änderungstypen zu prüfen. Weit verbreitet ist allerdings die gleichzeitige Änderung sowohl der Phase als auch der Amplitude. Dies wird auch als Phasen-Amplituden Modulation bezeichnet (oder quadratische Amplitudenmodulation, abgekürzt QAM). Diese Methode wird bei den üblichen Modemgeschwindigkeiten von 14,4k, 28,8k und 33,6k verwendet. Einzig bei 56k Modems wird diese Methode heute nicht verwendet. Aber selbst 56k Modems verwenden QAM für die Übertragungsrichtung vom PC zur Telefonleitung. Manchmal wird sogar für die andere Übertragungsrichtung QAM verwendet, wenn die Leitungsqualität nicht ausreichend gut ist. Daher ist QAM die am weitesten verbreitete Modulationsmethode für normale Telefonleitungen.

17.2 56k Modems (V.90)

Die Modulationsmethode, die oberhalb von 33,6k verwendet wird, ist vollkommen anders als die übliche Phasen-Amplituden Modulation. Die Signale auf einer normalen analogen Telefonleitung (d.h. kein ISDN-Anschluss) werden in der Ortsvermittlungsstelle der Betreibergesellschaft (in Deutschland ist dies in der Regel die Deutsche Telekom AG) in digitale Signale konvertiert, weil die Ortsvermittlungsstellen untereinander digitale Signale austauschen. Die höchste erreichbare Übertragungsgeschwindigkeit auf einer normalen Telefonleitung ist damit begrenzt auf die Übertragungsgeschwindigkeit zwischen den Ortsvermittlungsstellen, und diese liegt knapp unter 64kbps. Die vollen 64k werden nicht erreicht, da Bits, die eigentlich der Informationsübertragung dienen sollen, für Signalisierungszwecke verwendet werden. Auf einer digitalen Leitung (z.B. ISDN) existieren andere Möglichkeiten für die Signalisierung, so dass hier keine Bits »gestohlen« werden müssen.

Um die höchst mögliche Übertragungsgeschwindigkeit von 64kbps zu nutzen, müsste ein Modem genau wissen, wie die analogen Signale in der Ortsvermittlungstelle in digitale Signale konvertiert werden. Diese Aufgabe ist viel zu kompliziert, wenn der Gesprächspartner auch nur über einen analogen Anschluss an seiner Ortsvermittlungsstelle verfügt und auch dort wieder eine Konvertierung (diesmal von digital nach analog) stattfindet. Falls aber eine Seite über einen digitalen Anschluss verfügt, ist die Aufgabe lösbar (zumindest in einer Übertragungsrichtung). Falls Ihr Internetanbieter (Internet Service Provider, abgekürzt ISP) über einen digitalen Anschluss an das Telefonnetz verfügt, kann er ein bestimmtes digitales Signal über die Telefonleitung zu Ihrem PC senden. Dieses digitale Signal wird in der Ortsvermittlungstelle in ein analoges Signal konvertiert. Wenn Ihr Modem in der Lage ist, dieses Signal korrekt zu interpretieren, ist prinzipiell ein Übertragung mit maximal 64k in dieser Richtung (vom ISP zu Ihrem PC) möglich.

Wie werden in der Ortsvermittlungsstelle der Betreibergesellschaft die analogen Signale in digitale Signale konvertiert? Es wird eine Methode verwendet, bei der die Amplitude des analogen Signals 8000 mal pro Sekunde abgetastet wird. Jeder einzelne dieser Abtastwerte wird in einen 8-Bit Wert umgewandelt (d.h. das analoge Signal wird mit nur noch 256 möglichen Werten erfasst. Daher kommt der »magische« Wert: 8 Bits x 8000 Abtastungen pro Sekunde = 64kbps). Diese Modulationsart heisst Puls Code Modulation (PCM). Die entstehenden Datenbytes werden über die digitalen Leitungen der Betreibergesellschaft gesendet, wobei sich viele Verbindungen eine Leitung teilen. Möglich ist dies durch Anwendung eines bestimmten Schemas, welches vorgibt, zu welcher Zeit eine bestimmte Verbindung die Leitung nutzen darf

(Zeitmulitplex-Verfahren). Schließlich wird in der Ortsvermittlungsstelle ein umgekehrtes Multiplex Verfahren verwendet, um das digitale Signal zu erzeugen, welches ursprünglich durch PCM entstanden war. Dieses Signal wird in ein analoges Signal verwandelt und zu Ihrem Modem geschickt. Jedes Byte erzeugt dabei eine bestimmte Amplitude des analogen Signals. Die Aufgabe Ihres Modems besteht nun darin, das ursprüngliche 8-Bit Muster aus der Amplitude des analogen Signals zu rekonstruieren.

Dies ist eine Art von »Amplituden Demodulation«, aber diese Bezeichnung ist nicht ganz korrekt, weil kein Trägersignal existiert. Tatsächlich wird das Verfahren mit »Modulo-Konvertierung« bezeichnet und ist genau das Umgekehrte wie PCM. Um die Bitmuster des PCM Signals zu rekonstruieren, muss das Modem die Amplitude des analogen Signals exakt zu den selben Zeitpunkten abtasten, die zur Erzeugung des analogen Signals verwendet wurden. Wie kann das Modem diese Zeitpunkte wissen? Die Amplitude des analogen Signals ändert sich 8000 mal pro Sekunde (die ursprüngliche Abtastfrequenz). Diese Änderungen erzeugen auf der Telefonleitung ein 4 kHz Signal, aus dem das Modem die Abtastzeitpunkte entnehmen kann.

Nun ist die Kodierung der Amplitudenwerte beim PCM Verfahren nicht linear und bei kleinen Amplituden sind die Unterschiede zwischen zwei erlaubten Amplitudenwerten ziemlich gering. Um die Unterscheidung zwischen zwei nahe beieinanderliegenden Amplitudenwerten zu vereinfachen, werden bestimmte Werte nicht verwendet. Die Differenz zwischen zwei möglichen Werten wird dadurch größer und die Unterscheidung durch das Modem wird einfacher. Bei V.90 wird die Hälfte der erlaubten Amplitudenwerte nicht verwendet. Dies entspricht einer Kodierung in 7-Bit Werte anstelle von 8-Bit Werten. Daher ergibt sich auch die Übertragungsgeschwindigkeit von 56k: 7 Bits/Wert x 8000 Werte pro Sekunde = 56 kbps. Die erzeugten digitalen Werte bestehen natürlich weiterhin aus 8 Bits, aber nur 128 der 256 möglichen Bitmuster werden tatsächlich verwendet.

Aber es wird noch komplizierter. Wenn die Leitungsqualität nicht ausreichend hoch ist, werden u.U. noch weniger der möglichen Bitmuster zur Übertragung verwendet, und das bedeutet eine geringere Übertragungsgeschwindigkeit als 56k. In den USA gibt es zusätzlich noch gesetzliche Auflagen, die die Übertragung von hohen Amplitudenwerten über Telefonleitungen verbieten, was bestenfalls eine Übertragungsgeschwindigkeit von ca. 53,3 kbps für »56k«-Modems ermöglicht.

Beachten Sie, dass der digitale Teil des Telefonnetzes bidirektional ist. Die Verbindung zu Ihrem besteht ISP in Wirklichkeit aus zwei Übertragungskanälen, einer für jede Übertragungsrichtung. Für die Richtung von Ihrem PC zum ISP wird die übliche Phasen-Amplituden Modulation mit einer maximalen Geschwindigkeit von 33,6 kbps verwendet. Nur in der Gegenrichtung werden maximal 56 kbps erreicht.

18 Anhang B: »Baud« und »bps«

18.1 Ein einfaches Beispiel

»Baud« und »bps« gehören zu den Begriffen aus dem Computer- und Telekommunikationsbereich, die vielleicht am häufigsten falsch verwendet werden. Beide Begriffe werden oft gleichbedeutend gebraucht, sie sind es aber nicht. bps bezeichnet einfach die Anzahl der Bits, die pro Sekunde übertragen werden. Die Baudrate ist ein Maß dafür, wie oft sich ein Signal pro Sekunde ändert (oder ändern könnte). Bei einem typischen seriellen Port bedeutet eine Spannung von -12 V ein 1-Bit und +12 V bedeutet ein 0-Bit. Bei einem Wert von 38.400 bps wird die Bitfolge 01010101... ebenfalls einem Wert von 38.400 Baud entsprechen, weil auch die Spannung von plus auf minus und wieder zurück zu plus wechselt, und sie wechselt 38.400 mal pro Sekunde. Bei einer andere Bitfolge, z.B. 111000111..., wird es weniger Spannungswechsel pro Sekunde geben, weil für die drei 1-Bits die Spannung auf einem Pegel von -12 V bleibt, dennoch sprechen wir immer noch von einem Wert von 38.400 Baud, weil die Möglichkeit besteht, dass die Anzahl von Spannungswechseln pro Sekunde diesen Wert erreichen könnte.

Um das Problem von einer anderen Seite zu beleuchten, denken Sie sich imaginäre Zeitmarken zwischen jedem Bit (auch wenn sich die Spannung nicht ändert). Ein Wert von 38.400 Baud bedeutet dann 38.400 dieser Zeitmarken pro Sekunde. Die Zeitmarken bezeichnen die Zeitpunkte einer erlaubten Spannungsänderung und werden in Wirklichkeit durch ein synchronisiertes Taktsignal der Hardware erzeugt, aber nicht über das angeschlossene Kabel übertragen.

Nehmen wir an, ein Spannungswechsel könnte sich über mehr als zwei erlaubte Werte erstrecken. Nehmem wir weiter an, es gäbe vier erlaubte Werte. Jeder Wert würde ein Bitpaar repräsentieren. Z.B. könnte -12 V für das Paar 00 stehen, -6 V für 01, +6 V für 10 und +12 V für 11. In diesem Fall ist die Bitrate doppelt so groß wie die Baudrate. 3000 Spannungsänderungen pro Sekunde würden 6000 Bits pro Sekunde übertragen, da jede einzelne Änderung gleich 2 Bits überträgt. Mit anderen Worten, 3000 Baud bedeuten 6000 bps.

18.2 Realistische Beispiele

Das obige Beispiel ist stark vereinfacht. Beispiele in der Realität sind zwar komplizierter, aber die Idee ist dieselbe. Das erklärt, warum ein Modem mit 2400 Baud 14400 bps (oder mehr) übertragen kann. Das Modem erreicht eine höhere bps-Rate als die Baudrate, indem es mehrere Bits in eine Signaländerung kodiert. Wenn also 2 oder mehr Bits pro Baud kodiert sind, übersteigt die bps Rate die Baudrate. Wenn auf Ihrer Modem-Modem Verbindung 14400 bps übertragen werden, werden bei 2400 Baud mit jedem Signalwechsel 6 Bits übertragen. Eine Geschwindigkeit von 28800 bps kann bei 3200 Baud mit 9 Bits/Baud erreicht werden. Wenn das Wort »Baud« fälschlicherweise gebraucht wird, ist evtl. die Übertragungsgeschwindigkeit des Modems gemeint (z.B. 33,6 kbps).

Früher hatten die üblichen bps-Raten von Modems die Werte 50, 75, 110, 300, 1200, 2400 und 9600. Das waren auch gleichzeitig die bps-Raten auf der Verbindung zwischen seriellem Port und Modem. Heutzutage hat die Übertragungsrate von Modem zu Modem die Werte 14,4k, 28,8k, 33,6k und 56k, aber die Übertragungsgeschwindigkeit über die serielle Leitung zwischen seriellem Port und Modem hat nicht die gleichen Werte, sondern 19,2k, 38,4k, 57,6k und 115,2k. Bei Modems mit V.42bis Datenkompression (max. Kompressionsrate ist 4:1) sind Übertragungsgeschwindigkeiten bis zu 115,2k bei 33,6k Modems möglich (bei einem 56k Modem sind entsprechend 230,4k möglich).

Mit Ausnahme der 56k-Modems verwenden die meisten Modems 2400, 3000 oder 3200 Baud. Da Telefonleitungen auf Sprachübertragung ausgelegt sind, gibt es Einschränkungen bei der verfügbaren Bandbreite, Baudraten von mehr als 2400 sind deshalb nur sehr schwer zu erreichen und setzen eine sehr gute Leitungsqualität voraus.

Wie begann die Verwirrung zwischen bps und Baud? Nun, in den alten Zeiten, als die langsamen Modems noch als Hochgeschwindikeitsmodems angesehen wurden, stimmte die bps-Rate tatsächlich mit der Baudrate überein. Pro Phasenveränderung wurde ein Bit kodiert. Die Begriffe Baud und bps wurden gleichbedeutend gebraucht, weil sie den selben Wert hatten. Z.B. hatte ein 300 bps Modem auch eine Baudrate von 300. Das änderte sich alles, als schnellere Modems auftauchten, und die Bitrate überstieg die Baudrate. »Baud« ist nach Emile Baudot benannt, dem Erfinder des asynchronen Telegraphen. Eine Möglichkeit, mit der dieses Problem behoben werden kann ist, den Begriff »Symbolrate« anstelle des Ausdrucks »Baud« zu verwenden, um damit die Bezeichnung »Baud« zu vermeiden (Mit »Symbol« ist dann das übertragene Bitmuster gemeint).

19 Anhang C: Anbindung eines Terminalservers

Dieser Abschnitt wurde vom *Text-Terminal HOWTO* übernommen und angepasst.

Ein Terminalserver ist so etwas wie eine intelligente Vermittlungsanlage, die in der Lage ist, mehrere Modems oder Terminals mit einem oder mehreren Computern zu verbinden. Da die Vermittlung nicht mechanisch funktioniert, können die Geschwindigkeiten und Protokolle der Datenströme, die durch den Terminalserver hindurchgehen, verändert werden. Es gibt eine ganze Reihe von Herstellern für Terminalserver: Xyplex, Cisco, 3Com, Computone, Livingstone usw. Es gibt viele Typen und Leistungsklassen. Es würde den Rahmen dieses HOWTOs sprengen, die Typen zu beschreiben und zu vergleichen (ausserdem besteht die Möglichkeit, einen Linux PC als Terminalserver zu betreiben). Meistens werden sie für Modemverbindungen und weniger für direkt angeschlossenen Terminals verwendet.

Ein Einsatzbereich für Terminalserver besteht darin, viele Modems oder Terminals an ein Hochgeschwindigkeitsnetzwerk anzuschliessen, an das viele Host Computer angeschlossen sind. Natürlich benötigt der Terminalserver ausreichend Rechenleistung und die passende Software, um die Netzwerkprotokolle zu unterstützen, und in gewisser Hinsicht kann er als Computer angesehen werden. Der Terminalserver kann einen Dialog mit der Anwender führen und z.B. fragen, welcher Computer verbunden werden soll usw., oder er kann diese Verbindung ohne Rückfrage herstellen. Manchmal werden auch Druckeraufträge über einen Terminalserver gesendet.

Ein moderner PC verfügt über genügend Rechenleistung, um als Terminalserver zu fungieren, in diesem Falle sollte jeder serieller Port allerdings über einen eigenen Hardware-Interrupt besitzen. PCs haben nur wenige Interrupts für solche Zwecke übrig, und weil sie fest verdrahtet sind, können mithilfe von Software auch keine weiteren erzeugt werden. Ein Lösung besteht darin, eine gute serielle Multiport-Karte zu verwenden, die über ein eigenes Interrupt-System verfügt (bei Billigmodellen wird ein Interrupt des PCs von mehreren Ports gemeinsam verwendet). Weitere Information hierzu finden Sie im Serial-HOWTO. Wenn auf einem solchen PC Linux läuft und für mehrere serielle Ports ein `getty`-Prozess gestartet ist, kann er als Terminalserver angesehen werden, falls er über ein Netzwerk mit anderen PCs verbunden ist und seine Aufgabe vor allem darin besteht, Daten weiterzuleiten. Ein solcher »dedizierter« PC kann sich darauf konzentieren, die Interrupts der seriellen Ports zu bedienen, die (je nach Buffer des UART-Bausteins) bei jedem 14. Byte ausgelöst werden. Manchmal wird eine Software namens »radius« verwendet.

Heute werden von Terminalservern mehr als nur Terminals bedient. Sie bedienen auch PCs, die Terminals emulieren und sind manchmal mit einer Modembank verbunden, die wiederum mit mehreren Telefonleitungen verbunden ist. Einige beinhalten sogar eingebaute Modems. Wenn ein Terminal (oder ein PC, der ein Terminal emuliert) direkt mit einem Modem verbunden wird, kann das Modem am anderen Ende der Leitung an einen Terminalserver angeschlossen werden. In manchen Fällen erwartet der Terminalserver vom Anrufer standardmäßig PPP Pakete, die von echten Textterminals nicht generiert werden.

20 Anhang D: Andere Modemtypen

Dieses HOWTO beschäftigt sich in der aktuellen Version mit dem üblichen Modemtyp, der verwendet wird, um einen PC mit einer normalen analogen Telefonleitung zu verbinden. Es gibt verschiedenen andere Modemtypen, auch solche Geräte, die zwar Modem genannt werden, aber in Wirklichkeit gar keine Modems sind.

20.1 Digitale »Modems«

Manchmal wird die Standard-Definition eines Modems erweitert, um auch »digitale« Modems zu erfassen. Viele Privatpersonen und Firmen verfügen heute schon über einen digitalen Anschluss, so dass der Computer seine digitalen Signale direkt (oder beinahe direkt) über die Telefonleitung sendet. Aber wird immer noch ein Gerät benötigt, um die digitalen Signale des Computers in Signale zu konvertieren, die für die Telefonleitung zugelassen sind, und dieses Gerät wird häufig Modem genannt. Die nächsten beiden Abschnitte, ISDN und DSL beschäftigen sich mit solchen digitalen Modems.

20.2 ISDN-»Modems«

Das »Modem« ist in Wirklichkeit ein Terminaladapter (TA). Von Debian gibt es ein »isdnutils«-Paket. Es gibt auch ein deutsches *ISDN HOWTO*. Es behandelt die Treiber, die für diese Distribution erhältlich sind. Es gibt ein Paket »isdn4linux« und auch eine Newsgroup:

```
de.alt.comm.isdn4linux
```

Viele der Beiträge sind in deutsch, da in den USA ISDN nicht so weit verbreitet wie in Deutschland.

20.3 Digital Subscriber Line (DSL)

DSL läßt sich übersetzen als »digitale Teilnehmerleitung«. DSL verwendet das existierende verdrillte Leitungspaar (twisted pair) zwischen Ihrem Telefonanschluss und der Ortsvermittlungstelle. Es kann verwendet werden, wenn die Telefonleitung mit einer höheren Geschwindigkeit genutzt werden kann, als von einem normalen Modem (z.B. 56k) verwendet wird. Der Analog-Digital Konverter in der Ortsvermittlungstelle wird ersetzt durch einen Konverter, der einen sehr viel höheren Datendurchsatz verarbeitet (natürlich in einem anderen Format). Das Gerät, das die digitalen Signale Ihres Computers in diejenigen Signale konvertiert, die auf der lokalen Telefonleitung digitale Daten repräsentieren, wird ebenfalls Modem genannt. Dieses HOWTO geht in der dieser Version nicht auf die speziellen Aspekte dieser Modems ein.