Linux PCMCIA HOWTO

David Hinds (dhinds@hyper.stanford.edu) und Dirk Geschke (geschke@physik.uni-kassel.de)
v1.0-3, 5. September 1997

Dieses Dokument beschreibt die Installation und den Gebrauch des PCMCIA Card Service Pakets für Linux.

Inhaltsverzeichnis

1 Vorwort zur deutschen PCMCIA HOWTO

Wenn man bedenkt, wieviele mögliche PCMCIA Karten es gibt, so erscheint es schon erstaunlich, daß diese doch fast alle auch unter Linux funktionieren. Noch erstaunlicher ist die Einfachheit, mit der solche Karten mit dem *PCMCIA Card Service* Paket erkannt und konfiguriert werden können. Ja sogar die Installation des Paketes erweist sich doch als überaus einfach. Allerdings kann es passieren, daß das Paket den gegebenen Besonderheiten angepaßt werden muß, bevor es richtig läuft. Dazu dient dieses *PCMCIA HOWTO*. Das Originalpaket inklusive der englischsprachigen *PCMCIA HOWTO* stammt von David Hinds.

Bei der Übersetzung habe ich mich meist eng an die Vorlage gehalten. Daher klingt manches noch steif und zum Teil unverständlich. Ich werde mich bemühen, dieses in nächster Zeit, falls ich solche finden werde, zu verbessern. Anmerkungen, Anregungen, Verbesserungen oder gar ganzen neuen Kapiteln stehe ich daher sehr aufgeschlossen gegenüber. Die jetzt hiermit vorliegende Version ist wirklich noch in einer *Rohfassung* und muß noch überarbeitet werden. Allerdings habe ich mir gedacht, lieber eine solche Version bereitzustellen als gar keine.

2 Allgemeine Informationen und Hardwareanforderungen

2.1 Einführung

PCMCIA Card Services realisiert über ladbare Kernelmodule die volle Unterstützung von PCMCIA. Ein Kartenmanager-Daemon (`cardmgr`) überwacht die PCMCIA Slots auf Einschub oder Entfernung von Karten. Gleichzeitig werden die benötigten Treiber geladen oder wieder entfernt. Karten können daher jederzeit gewechselt werden.

Diese Software ist immer noch in der Entwicklung. Sie enthält daher wahrscheinlich noch Fehler und sollte daher mit Vorsicht verwendet werden.

2.2 Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright für die englische *PCMCIA HOWTO*, auf der dieses Dokument basiert, liegt bei David A. Hinds. Das Copyright für die deutsche Version liegt bei Dirk Geschke.

Das Dokument darf gemäß der GNU *General Public License* verbreitet werden. Insbesondere bedeutet dieses, daß der Text sowohl über elektronische wie auch physikalische Medien ohne die Zahlung von Lizenzgebühren verbreitet werden darf, solange dieser Copyright-Hinweis nicht entfernt wird. Eine kommerzielle Verbreitung ist erlaubt und ausdrücklich erwünscht. Bei einer Publikation in Papierform ist das Deutsche Linux HOWTO Projekt hierüber zu informieren.

2.3 Was ist die aktuelle Version und wo kann man sie bekommen?

Die aktuelle Major Version des *Card Service* Pakets ist die Version 2.9; die Aktualisierungs- und Fehlerbeseitigungsversionen (Minor Versionen) sind mit den Nummern 2.9.1, 2.9.2 usw. versehen.

Der Quellcode der neuesten Version ist auf dem FTP-Server

```
hyper.stanford.edu:/pub/pcmcia
```

unter dem Namen `pcmcia-cs-2.9.?.tar.gz` erhältlich. Gewöhnlich befinden sich hier mehrere Versionen. David behält nur die neueste Minor Version einer gegeben Major Version. Er bewahrt ebenfalls die letzte Version der früheren Major Version als Notanker auf, da neue Major Versionen teilweise ungetesteten Code enthalten. Der aktuelle Notanker ist die Version 2.8.23. Es liegt an einem selber, welche Version man verwendet. Die Datei `CHANGES` zeigt die wesentlichen Unterschiede der Versionen auf.

`hyper.stanford.edu` wird auf folgendem Server gespiegelt: `metalab.unc.edu`. Neue Versionen sollten auch über `tsx-11.mit.edu` verfügbar sein.

Wenn man selber die PCMCIA Treiber nicht übersetzen will, so sind die Treiber bereits vorübersetzt in den meisten Linux Distributionen enthalten.

2.4 Welche Systeme werden unterstützt?

Dieses Programmpaket sollte auf fast allen Linux-tauglichen Notebooks laufen. Alle gebräuchlichen PCMCIA Controller, wie z.B. Intel, Cirrus, Vadem, VLSI, Ricoh und Databook Chipsätze werden unterstützt. Controller, die üblicherweise in Notebooks von IBM und Toshiba eingebaut sind, werden auch unterstützt. PCMCIA Karteneinschübe für Desktop-Computer, die direkt an den ISA-Bus angeschlossen werden, sollten eher funktionieren als SCSI-zu-PCMCIA oder IDE-zu-PCMCIA Adapter.

Der Controller 6AHC05GA von Motorola, der in einigen Hyundai Notebooks Verwendung findet, wird nicht unterstützt. Das gleiche gilt für den Controller, der gewöhnlich im HP Omnibook 600 verwendet wird. PCI zu CardBus bridge Controller von SMC, Ricoh, Cirrus und TI sind derzeitig nur im langsamen 16-Bit Modus verwendbar und dieser ist immer noch experimentell.

2.5 Welche PCMCIA Karten werden unterstützt?

Die aktuelle Version enthält Treiber für eine Vielzahl von Ethernetkarten, einen Treiber für Modem- und serielle Schnittstellen-Karten, verschiedene Treiber für SCSI Adapter, einen Treiber für ATA-/IDE-Laufwerkskarten und Speicherkartentreiber, welche die meisten SRAM-Karten und einige Flash-Karten unterstützen sollten. Die `SUPPORTED.CARDS` Datei, die jedem Card Service Paket beiliegt, listet alle Karten auf, von denen bekannt ist, daß sie mindestens in einem aktuellen System laufen.

Die Wahrscheinlichkeit, daß eine Karte, die nicht auf der Liste steht, funktioniert, hängt vom Typ der Karte ab. Im wesentlichen sollten alle Modemkarten mit den mitgelieferten Treibern funktionieren. Einige Ethernetkarten können funktionieren, wenn sie OEM Versionen der unterstützten Karten sind. Andere Varianten von I/O-Karten (Frame Buffer, Soundkarten, etc.) werden solange nicht funktionieren, bis sich einer findet, der einen geeigneten Treiber schreibt.

2.6 Wann wird meine neue Karte unterstützt?

Unglücklicherweise wird gewöhnlich kein Geld für die Entwicklung von Treibern gezahlt. Daher muß man selber Hand anlegen, wenn man einen Treiber für seine bevorzugte Karte haben will. David Hunt arbeitet auf ein Modell wie den Linux-Kernel zu, wo er nur noch für die Entwicklung des Kern-PCMCIA Codes zuständig ist und andere die Entwickelung und Pflege der Treiber für die speziellen Karten übernehmen. Die Datei `SUPPORTED.CARDS` erwähnt einige Karten, für welche die Entwicklung im Gange ist. David will dabei helfen, wo es geht. Allerdings ist die Analyse und Entwicklung von Treibern via E-Mail nicht besonders effektiv.

2.7 Mailinglisten

David bemüht sich um den Erhalt einer Datenbank und Mailingliste von PCMCIA-Anwendern. Aktuell betreibt er eine Webseite mit Berichten zur Installation und Konfiguration verschiedener PCMCIA-Karten unter Linux. Auch finden sich hier Hinweise zur Programmierung und Fehlersuche. Die Linux PCMCIA-Informationsseite ist unter

```
http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html
```

zu finden. Anwender können Benachrichtigung per E-Mail zu Antworten auf besondere Fragen oder alle neuen Nachrichten zu speziellen Kategorien anfordern. David hofft, daß dies eine brauchbare Sammlung an Informationen für die Fragen wird, die außerhalb des Rahmens dieser HOWTO liegen.

Es existiert eine Linux Mailingliste, die dem Thema Notebook gewidmet ist: *linux-laptop*. Für mehr Informationen muß man eine Nachricht, die das Wort `help` enthält, an `majordomo@vger.rutgers.edu` senden. Zum Eintrag in die Mailingliste muß eine E-Mail mit dem Titel `subscribe linux-laptop` an diese Adresse geschickt werden. Die Mailingliste kann ein gutes Forum für die Diskussion von PCMCIA-Themen sein.

Die *Linux Laptop Homepage* enthält zahlreiche Links zu Stellen, die Informationen über die Konfiguration spezieller Notebooks mit Linux enthalten. Außerdem ist hier eine durchsuchbare Datenbank zu finden, die Informationen zur Konfiguration von Systemen enthält.

3 Kompilierung, Installation und Konfiguration

3.1 Vorbemerkungen und Kernelkonfiguration

Vor dem Beginn sollte man darüber nachdenken, ob es wirklich erforderlich ist, das PCMCIA-Paket selber zu kompilieren. Die meisten Linux Distributionen werden mit einem vorkompilierten PCMCIA-Treiberpaket ausgeliefert. Im allgemeinen muß das Paket nur dann selbst kompiliert werden, wenn man eine neuere Version des PCMCIA-Paketes benötigt, weil erst diese einen bestimmten Treiber oder eine bestimmte Funktionalität besitzt, oder man eine andere Kernelversion installiert hat, als die, die mit der Distribution ausgeliefert worden ist. Das Kompilieren des PCMCIA-Pakets ist technisch nicht schwierig, setzt allerdings eine gewisse Vertrautheit mit Linux voraus.

Die folgenden Dinge müssen installiert sein, bevor man mit der Installation des Paketes beginnen kann:

- Einer der folgenden Kernel: 1.2.8 bis 1.2.13, 1.3.30, 1.3.37, 1.3.39 bis 1.3.99, 2.0.x oder 2.1.x.
- Die passende Version der Programme für Module.
- Optional die *XForms* Bibliothek für ein X11 User Interface

Die neueste Version verlangt einen Kernel der Version 1.2.18 oder höher, oder ein Entwickler-Kernel 1.3.30 oder neuer. Version 1.3.38 funktioniert nicht und die Versionen 1.3.31 bis 1.3.36 sind nicht getestet worden. Es werden auch relativ neue Modulprogramme benötigt. Es gibt keine Kernel-Patches speziell für PCMCIA.

Man benötigt einen kompletten Quellbaum des Linux-Kernels. Ein bereits übersetzter Kernel allein genügt nicht. Das PCMCIA Modul enthält einige Referenzen auf Kernel-Quelltexte. Während man einen neuen Kernel übersetzen will, um unnötige Treiber zu entfernen, ist dies für die Installation von PCMCIA nicht notwendig.

Aktuelle stabile Kernelquellen und Patches sind unter folgenden Adressen erhältlich:

- `http://metalab.unc.edu/pub/Linux/kernel/v2.0`
- `http://tsx-11.mit.edu/pub/linux/sources/system/v2.0`

Aktuelle Modulwerkzeuge sind dort unter `modules-2.0.0.tgz` zu finden. Entwicklungskernel sind in den entsprechenden `v2.1` Verzeichnissen zu finden. Einige aktuelle Entwicklungskernel und Compiler arbeiten schlecht mit älteren Modulwerkzeugen zusammen. Bei der Verwendung von 2.1-Kerneln muß man sicherstellen, daß man die richtige Kombination an Libraries und Modulwerkzeugen einsetzt. Die neuesten Modulwerkzeuge sowie Versionen für ältere Kernel sind unter

```
http://www.pi.se/blox/modules
```

zu finden.

Wird der Einsatz einer PCMCIA Ethernetkarte geplant, so ist bei der Konfiguration des Kernels die Unterstützung für Netzwerke zu aktivieren und die normalen Linux Ethernettreiber einschließlich der Treiber für »Pocket« und portable Adapter zu deaktivieren. Die PCMCIA-Netzwerkkartentreiber sind alle als ladbare Module implementiert. Jeder unnötig in den Kernel einkompilierte Treiber verschwendet nur Platz.

Wenn SLIP, PPP oder PLIP verwendet werden sollen, müssen die entsprechenden Treiber entweder in den Kernel einkompiliert werden oder die Treiber als Module geladen werden. In der Kernelkonfiguration der 1.2.x-Kernel ist es nicht möglich, Optionen wie SLIP-Kompression für ladbare Module zu aktivieren. Daher wird es hier wohl besser sein, SLIP direkt in den Kernel zu binden, wenn diese Funktionalität benötigt wird.

Soll ein PCMCIA-Token Ring Karte verwendet werden, so muß der Kernel mit der Option »Token Ring driver support« übersetzt worden sein, wobei die Option `CONFIG_IBMTR` ausgeschaltet sein sollte.

Sollen PCMCIA IDE-Karten verwendet werden, so sollte im Kernel `CONFIG_BLK_DEV_IDE_PCMCIA` aktiviert sein. Diese Funktion gibt es nur bei den Kerneln 1.3.72 bis 2.1.7. Ältere Kernel haben keine Unterstützung für wechselbare IDE-Geräte; neuere Kernel benötigen keine spezielle Konfiguration.

Bei der Verwendung von PCMCIA SCSI-Karten sollte in der Kernelkonfiguration `CONFIG_SCSI` aktiviert sein. Ebenso sollten die Haupttreiber für die geplanten SCSI-Geräte wie Festplatten, Bandlaufwerke, CD-ROMs oder generische Devices aktiviert sein, wohingegen die Treiber für spezielle SCSI-Controller-Karten nicht benötigt werden.

Wenn Treiber des Kernels, die für die PCMCIA-Geräte benötigt werden, als Module übersetzt wurden, so ist die Datei `/etc/pcmcia/config` entsprechend zu modifizieren, so daß jeweils die benötigten Module für eine Karte geladen werden. Wenn zum Beispiel ein serieller Treiber modularisiert wurde, muß die serielle Gerätedefinition wie folgt geändert werden:

```
device "serial_cs"
  class  "serial" module "misc/serial", "serial_cs"
```

Wenn der Kernel mit der Option `CONFIG_MODVERSIONS` (Kontrolle der Versionsnummer von Modulen) kompiliert wurde, so wird das Konfigurationsskript nach der Existenz der Datei `/usr/include/linux/modversions.h`, der Moduldatenbank, suchen. Diese wird erstellt, wenn im Kernelquellverzeichnis der Befehl `make dep` ausgeführt wird.

Dieses Paket enthält eine auf X11 basierte Kartenstatusanzeige mit dem Namen `cardinfo`. Dieses Programm basiert auf der kostenlos vertriebenen Widgetbibliothek XForms, welche installiert sein muß, wenn man `cardinfo` verwenden will. Eine binäre Version ist unter

```
hyper.stanford.edu:/pub/pcmcia/extras
```

sowohl im *a.out* als auch im *ELF* Format erhältlich. Es müssen zusätzlich alle gewöhnlichen X11 Headerdateien und Bibliotheken installiert sein.

3.2 Installation

Hier ist ein Übersicht des Installationsprozesses:

- Entpacken von `pcmcia-cs-2.9.?.tar.gz` in `/usr/src`.
- Starte `make config` in dem neu entstandenen Verzeichnis `pcmcia-cs-2.9.?`.
- Starte `make all`, danach `make install`.
- Anpassen des PCMCIA-Startskripts und der Optionsdatei in `/etc/pcmcia` an die eigenen Bedürfnisse.

Während des Laufs von `make config` werden einige Konfigurationseinstellungen abgefragt und es wird überprüft, ob das System für die Installation der PCMCIA-Unterstützung entsprechend vorbereitet ist. In den meisten Fällen können die Standardeinstellungen verwendet werden. Für den Fall, daß Probleme auftreten, ist es ratsam, auf die Ausgaben dieses Programmes zu achten.

Wird das PCMCIA-Paket für den Gebrauch auf einem anderen Computer kompiliert, so sollte ein anderes Installationsverzeichnis angegeben werden, wenn danach gefragt wird. Dies sollte ein absoluter Pfad sein. Alle PCMCIA-Dateien werden relativ zu diesem Pfad installiert. Danach ist man in der Lage, das ganze Unterverzeichnis in eine tar Datei zu schreiben und zum Zielcomputer zu kopieren, wo das Archiv dann im Root-Verzeichnis entpackt werden muß, um das PCMCIA-Paket an der richtigen Stelle zu installieren.

Wenn das Paket für einen andere Architektur, z.B. für einen Linux Rechner mit einer Alpha CPU, crosskompiliert werden soll, so kann ein anderer Compiler und Linker angegeben werden. Dies kann auch auf Systemen mit gemischtem *a.out*- und *ELF*-Formaten hilfreich sein. Auch wird das Skript nach zusätzlichen Optionen des Compilers für's Debugging fragen.

Einige der Programme (`cardctl` und `cardinfo`) können entweder in der *safe*- oder *trusting*-Form kompiliert werden. Die *safe*-Form verhindert, daß Nicht-Root-Anwender die Kartenkonfiguration ändern können. Die *trusting*-Form erlaubt normalen Anwendern, Befehle abzusetzen, um Karten anzuhalten, wiederanzufahren, zurückzusetzen und die aktuelle Konfiguration zu ändern. Das Konfigurationsskript wird nach einer der beiden Formen fragen. Die Voreinstellung ist *safe*.

Es gibt ein paar Einstellungen der Kernelkonfiguration, die Auswirkungen auf die PCMCIA-Werkzeuge haben. Das Konfigurationsskript kann diese meistens aus dem laufenden Kernel herauslesen. Alternativ, wenn für einen anderen Computer kompiliert wird, kann diese Konfiguration aus einem Kernelquellcode-Verzeichnisbaum herausgelesen werden oder sie können interaktiv abgefragt werden.

Das Durchlaufen von `make all` gefolgt von `make install` wird die Kernelmodule und Anwendungsprogramme erzeugen und installieren. Die Kernelmodule werden in `/lib/modules/`{*version*}`/pcmcia` installiert. Die Programme `cardmgr` und `cardctl` werden in `/sbin` installiert. Wenn `cardinfo` erstellt wird, so wird es in `/usr/bin/X11` installiert.

Die Konfigurationsdateien werden in das Verzeichnis `/etc/pcmcia` kopiert. Wird über eine alte Version installiert, so werden die alten Kofigurationsdateien umbenannt, bevor die neuen installiert werden. Die alten Skriptdateien bekommen Endungen wie *.~1~, *.~2~ und so weiter.

Wenn nicht bekannt ist, welcher PCMCIA Controller Chip in den Rechner eingebaut ist, so kann die `probe` Funktion im Unterverzeichnis von `cardmgr/` verwendet werden, um den Chiptyp zu ermitteln. Es gibt zwei Hauptarten an Chips: der Databook TCIC-2 Typ und der Intel i82365SL kompatible Typ.

Ein Daemon auf Anwenderebene überwacht, ob eine Karte eingeschoben oder entfernt wird. Dieser Daemon heißt `cardmgr`. Dieser ist ähnlich in der Funktion wie Barry Jaspans `pcmdiad` in früheren Versionen. `cardmgr` liest die Konfigurationsdatei `/etc/pcmcia/config`, welche die bekannten PCMCIA Karten beschreibt. Es werden darin ebenso die zusätzlichen Module erwähnt, die für den Gebrauch der PCMCIA-Karte notwendig sind und eventuell auf das eigene System angepaßt werden müssen. Sollten weitere Informationen benötigt werden, sollte die Manual Page weiterhelfen.

3.3 Abschluß der Installation bei Systemen, die BSD Initskripte verwenden

Einige Linux Distributionen, wie z.B. Slackware, verwenden eine BSD-Anordnung der Systemstartskripte. Wenn die Datei `/etc/rc.d/rc.M` existiert, gehört das System zu dieser Gruppe. Das Skript `rc.pcmcia`, das in `/etc/rc.d` installiert ist, kontrolliert den Start und Stopp des PCMCIA-Systems. `make install` verwendet das Kommando `probe`, um den Chiptyp zu ermitteln und das Skript `rc.pcmcia` entsprechend anzupassen. Es sollte dann eine Zeile in der Startdatei `/etc/rc.d/rc.M` eingefügt werden, die das PCMCIA-Startskript aufruft:

```
/etc/rc.d/rc.pcmcia start
```

Es ist nicht wichtig, wo diese Zeile erscheint, solange sie nach dem Start von `syslogd` steht.

3.4 Abschluß der Installation bei Systemen, die den System V Startskripts folgen

RedHat, Caldera und Debian Linux haben Startskripts, die dem *System V* folgen. Wenn ein Verzeichnis mit dem Namen `/etc/init.d` oder `/etc/rc.d/init.d` existiert, gehört die Distribution zu dieser Gruppe. Das `rc.pcmcia` Skript wird dann entsprechend in `/etc/init.d` oder `/etc/rc.d/init.d` installiert. Es besteht hier keine Notwendigkeit, irgendeine Startdatei zu editieren, um PCMCIA zu aktivieren: Dies geschieht automatisch.

Wenn das Verzeichnis `/etc/sysconfig` existiert, wird eine separate Konfigurationsdatei zum Starten als `/etc/sysconfig` installiert. Wenn in diesem Fall irgendwelche Moduloptionen, wie z.B. `PCIC=` oder `PCIC_OPTS=` Einstellungen, geändert werden sollen, so ist dies in dieser Datei einzutragen und nicht im Startskript. Bei nachfolgenden Installationen wird diese Datei nicht überschrieben. Einige Systeme werden mit Systemkonfigurationsdateien ausgeliefert, die PCMCIA per Voreinstellung deaktivieren (wie z.B. S.u.S.E.). Daher sollte der Inhalt solcher Dateien vorher überprüft werden (bei S.u.S.E.: `/etc/rc.config`).

Einige frühere Versionen verwendeten die PCMCIA-Skripte in `/etc/sysconfig` anstelle von `/etc/pcmcia`. Die aktuelle Version und zukünftige werden das Verzeichnis `/etc/pcmcia` auf allen Systemen verwenden. Existierende PCMCIA-Skripte in `/etc/sysconfig` werden nach `/etc/pcmcia` verschoben.

3.5 Computerspezifische Konfigurationsoptionen

Card Services sollte es automatisch vermeiden, auf Ports und Interrupts zuzugreifen, die von anderen Geräten verwendet werden. Es wird ebenfalls versucht, Konflikte mit anderen unbekannten Geräten zu entdecken. Allerdings ist dies nicht absolut zuverlässig. In einigen Fällen müssen bestimmte Bereiche für einen Treiber explizit ausgeschlossen werden. Dies erreicht man über die Datei `/etc/pcmcia/config.opts`.

Hier sind einige Einstellungen für spezielle Notebooks:

- Beim AMS SoundPro vermeide IRQ 10.
- Bei einigen AMS TravelPro 5300 Modellen sollte der Speicherbereich 0x8000-0xcffff verwendet werden.
- Beim Chicony NB5 sollten die IRQs 5 und 9 nicht verwendet werden.
- Die Ports 0x2f8-0x2ff, IRQ 3 und IRQ 5 sollten beim Compaq Presario 1020 ausgeschlossen werden.
- Die Ports 0x300-0x30f sollten beim HP Omnibook 4000C vermieden werden.
- Die IRQs 5 und 9 sollten beim Micron Millenia Transport nicht verwendet werden.
- Beim NEC Versa M schließe die Ports 0x2e0-0x2ff und IRQ 9 aus.
- Beim NEC Versa P/75 sollten die IRQs 5 und 9 ausgeschlossen werden.
- Bei der NEC Versa 6000 Serie sollten die IRQs 9 und 12 ausgeschlossen werden.
- Beim ProStar 9200, Altima Virage und Acquiline Hurricane DX4-100 vermeide IRQ 5, Port 0x330-0x35f. Versuche eventuell den Speicherbereich 0xd8000-0xdffff.
- Verwende auf dem TI TravelMate 5000 den Speicherbereich 0xd4000-0xdffff.
- Bei dem Toshiba T4900 CT schließe IRQ5 und die Ports 0x2e0-0x2e8 und 0x330-0x338 aus.
- Bei dem Twinhead 5100, HP 4000, Sharp PC-8700 und PC-8900 schließe IRQ 9 (Sound) und IRQ 12 aus.
- Bei der MPC 800 Serie sollten IRQ 5 und der Port 0x300-0x30f für das CD-ROM-Laufwerk vermieden werden.

Einige PCMCIA-Controller haben optionale Eigenschaften, die bei einigen Systemen eventuell implementiert sind. Es ist generell unmöglich, für einen PCMCIA-Treiber festzustellen, ob diese speziellen Eigenschaften eingebaut sind. Man lese daher die Manual Page des eingebauten Treibers für die möglicherweise implementierten und aktivierbaren Eigenschaften.

In wenigen Fällen ist das `probe` Kommando nicht in der Lage, den Controllertyp automatisch zu ermitteln. Bei Halikan NBD 486-Systemen, die einen TCIC-2-Controller an einer ungewöhnlichen Stelle haben, muß die Datei `rc.pcmcia` editiert werden, damit das `tcic` Modul geladen wird. Dabei muß der Parameter `PCIC_OPTS` auf `tcic_base=0x02c0` gesetzt werden.

Die Treibermodule *tcic* und *i82365* haben zahlreiche Parameter zur Behandlung der Busgeschwindigkeit, welche für besonders schnelle Prozessoren angepaßt werden müssen. Symptome für Geschwindigkeitsprobleme sind unter anderem Probleme der Kartenerkennung, Blockierung bei hoher Prozessorbelastung, große Fehlerraten oder schlechte Übertragungsraten der Geräte. Man kontrolliere die entsprechenden Manual Pages für nähere Informationen. Hier ist eine kurze Zusammenfassung:

- Cirrus Controller haben zahlreiche anpaßbare Geschwindigkeitsparameter. Der wichtigste scheint die `cmd_time` Option, die die Dauer von PCMCIA-Buszyklen festlegt, zu sein. Schnelle 486-Systeme (insbesondere DX4-100) scheinen oft von einer Erhöhung von 6 (Voreinstellung) auf 12 oder 16 zu profitieren.
- Der Cirrus PD6729 PCI Controller hat eine `fast_pci` Einstellung, welche gesetzt werden sollte, falls die PCI-Busgeschwindigkeit größer als 25 MHz ist.
- Bei Vadem VG-468- und Databook TCIC-2-Controllern ändert die `async_clock` Option die relative Geschwindigkeit des PCMCIA-Busses zum Hauptbus. Wird diese Option gesetzt, so werden extra Wartezyklen bei einigen Operationen eingefügt werden. David hat von einem Notebook gehört, das diese Option benötigte.
- Das `pcmcia_core` Modul besitzt einen `cis_speed` Parameter zum Ändern der Geschwindigkeit, mit der auf den Speicher der Card Information Structure (CIS) zugegriffen wird. Bei einigen Systemen mit schnellen Busgeschwindigkeiten kann die Erhöhung dieses Parameters, d.h. Verlangsamung des Kartenzugriffs, die Erkennung einzelner Karten begünstigen.
- Dies ist keine Geschwindigkeitseigenschaft, ist aber hilfreich, wenn sich mehr als ein PCMCIA-Controller im System befindet oder mehrere Einschübe sich in einer »Docking Station« befinden. In diesem Fall hilft es, wenn beim Laden des *i82365* Moduls die Option `extra_sockets=1` gesetzt ist.

Alle diese Optionen sollten durch Modifizierung der Datei `rc.pcmcia` konfiguriert werden. Zum Beispiel:

```
# sollte entweder i82365 oder tcic sein
PCIC=i82365
# Geschwindigkeitsparameter des Treibers sollten hier
# stehen
PCIC_OPTS="cmd_time=12"
# pcmcia_core Optionen werden hier angegeben
CORE_OPTS="cis_speed=500"
```

Hier sind ein paar Geschwindigkeitseinstellungen für spezielle Systeme:

- Beim ARM Pentium-90 oder Midwest Micro Soundbook Plus sollten die Optionen `freq_bypass=1 cmd_time=8"`

  verwendet werden.
- Beim Midwest Micro Soundbook Elite verwendet man `cmd_time=12"`

  .

- Beim Gateway Liberty versuche man die Option `cmd_time=16"`

  .

Bei einigen Systemen, die den Cirrus-Controller verwenden, wie z.B. der NEC Versa M, versetzt das BIOS den Controller in einen speziellen Ruhemodus während des Einschaltens. Auf solchen Systemen wird das `probe`-Kommando keinen PCMCIA-Controller entdecken. Wenn dies geschieht, muß die Datei `rc.pcmcia` von Hand wie folgt abgeändert werden:

```
# sollte entweder i82365 oder tcic sein
PCIC=i82365
# Geschwindigkeitsparameter des Treibers sollten hier
# stehen
PCIC_OPTS="wakeup=1"
```

3.6 Probleme beim Laden der Kernelmodule

Das Konfigurationsskript stellt normalerweise sicher, daß die PCMCIA-Module mit dem installierten Kernel funktionieren. Daher lassen sich Probleme mit dem Laden der Module meist darauf zurückführen, daß der Installationsvorgang nicht korrekt durchgeführt worden ist. Einige solcher Fehlermeldungen werden direkt auf der Linuxkonsole ausgegeben, andere werden in der Systemlogdatei aufgezeichnet. Diese heißt normalerweise `/usr/adm/messages` oder `/var/log/messages`. Dies hängt von der Konfiguration des `syslogd` Daemons ab und wird in der Datei `/etc/syslog.conf` festgelegt. Um den Fehler einzugrenzen, sollten dieses Log-Dateien genau untersucht werden. Auf diese Weise kann auch herausgefunden werden, welches Modul das Problem verusacht.

Einige der PCMCIA-Module benötigen Kerneldienste, die nur dann vorhanden sind, wenn der Kernel entsprechend konfiguriert wurde. Zum Beispiel verlangt der SCSI-Kartentreiber, daß der Kernel mit SCSI-Unterstützung übersetzt wurde. Analog benötigt der Netzwerktreiber die Netzwerkunterstützung des Kernels. Wenn dem Kernel die notwendigen Treiber fehlen, kann es sein, daß sich `insmod` mit der Begründung von undefinierten Symbolen weigert, ein Modul zu laden.

Wenn `insmod` von »wrong version«-Fehlern berichtet, so bedeutet dies, daß die Module für eine andere Kernelversion als die aktuell auf dem System laufende übersetzt worden sind. Dieses kann passieren, wenn die Module auf einem anderen Computer mit anderer Konfiguration übersetzt worden sind und auf den eigenen kopiert wurden oder wenn der Kernel nach der Installation der PCMCIA-Module neu konfiguriert wurde.

Eine andere Fehlerquelle besteht darin, daß die Module und der Kernel mit unterschiedlichen Einstellungen von `CONFIG_MODVERSIONS` übersetzt worden sind. Wenn ein Modul mit Versionskontrolle in ein Kernel ohne diese Kontrolle geladen wird, so wird `insmod` sich über undefinierte Symbole beschweren.

Zum Schluß sind noch relativ neue Versionen der *binutils* inkompatibel mit den älteren Versionen der Modulwerkzeuge. Dies kann Inkompatibilitäten der Module verursachen. Eine übliche Fehlermeldung in einem solchen Fall ist die Meldung, daß `gcc_compiled` nicht definiert sei. Wenn diese Fehlermeldungen erscheinen, sollte man auf die neuesten Modulwerkzeuge aufrüsten. Diese sind unter

```
http://www.pi.se/blox/modules
```

zu bekommen.

3.7 Interruptprobleme beim Wechsel des Kartenstatus

In den meisten Fällen wird der Treiber (*i82365* oder `tcic`) automatisch einen Interrupt suchen und auswählen, um den Kartenstatus anzuzeigen. Diese automatische Interruptsuche funktioniert bei einigen Intel-kompatiblen Controllern nicht, wie z.B. Cirrus Chips und einige in IBM ThinkPads verwendeten Chips. Wenn ein Gerät während des

Testvorgangs inaktiv ist, kann es passieren, daß der Interrupt dieses Gerätes als frei erscheint. In solchen Fällen kann es passieren, daß dieser Interrupt vom Treiber verwendet wird.

Bei den *i82365* und dem `tcic` Treibern kann die Option `irq_mask` verwendet werden, um die möglichen Interrupts einzuschränken. Diese Maske schränkt den Satz der möglichen Interrupts, die für den Gebrauch mit PCMCIA-Karten oder für die Anzeige von Kartenstatusänderungen verwendet können, ein. Die Option `cs_irq` kann ebenfalls verwendet werden, um explizit den Interrupt festzulegen, mit welchem der Wechsel des Kartenstatus überwacht wird.

Wenn ein funktionierender Interrupt nicht gefunden werden kann, so besteht die Möglichkeit, einen Pollingmodus zu verwenden. Sowohl der *i82365* als auch der `tcic` Treiber akzeptieren die Option `poll_intervall=100`, durch welche festgelegt wird, daß sie jede Sekunde Kartenstatus pollen. Diese Option sollte auch verwendet werden, wenn der Spielraum für freie Interrupts für den Gebrauch durch PCMCIA stark eingeschränkt ist. Insbesondere für Systeme mit mehreren PCMCIA Controllern ist die Zahl der Interrupts zur Anzeige der Statusinformationen der Karten stark eingeschränkt.

Alle diese Optionen sollten in der `PCIC_OPTS=` Zeile in der Datei `rc.pcmcia` gesetzt werden.

3.8 Probleme mit der Konfiguration des Speicherfensters

Per Voreinstellung reservieren sich die PCMCIA-Treiber Speicherfenster im Bereich 0xc0000-0xfffff nach der Überprüfung dieses Bereiches auf den Gebrauch durch ROM oder andere Geräte. Dieses Fenster wird in der Datei `/etc/pcmcia/config.opts` definiert. Die Überprüfung dieses Bereichs wird durchgeführt, wenn ein Treiber versucht, eine neue Karte zu konfigurieren. Dieser Prüfvorgang ist nicht Narrensicher, so daß Probleme auftreten können. Wenn ein Speicherbereich erkannt wurde, der von einem anderen Gerät verwendet wird, so kann es passieren, daß die Karte nicht korrekt erkannt wird. Durch das von einigen Chipsätzen unterstützte BIOS »Shadowing« können ebenfalls Fehler entstehen. Wenn man feststellt, daß alle Karten fälschlicherweise immer als Speicherkarten erkannt werden, sollte man sichergehen, daß das BIOS »Shadowing« beim Computer ausgeschaltet ist. Ein gutes Fenster zu finden, erfordert manchmal einiges Herumexperimentieren. Einige gute Fensteralternativen, die man versuchen kann, sind die Bereiche 0xd8000-0xdffff, 0xc0000-0xcffff und 0xc8000-0xcffff.

Wenn man DOS PCMCIA Treiber besitzt, kann man versuchen, anhand dieser einen guten Speicherbereich herauszufinden. Es ist jedoch zu beachten, daß diese Adressen oft in Segmentform angegeben sind. Es fehlt in diesem Fall die letzte hexadezimale Ziffer. Die absolute Adresse 0xd0000 würde also z.B. als 0xd000 angegeben werden. Man sollte also darauf achten, diese letzte Ziffer hinzuzufügen, falls man solche Werte übernimmmt.

Wenn das Anpassen des Fensters im Speicherbereich das Problem der Kartenerkennung nicht löst, so liegt wahrscheinlich ein Geschwindigkeitsproblem (»timing«) vor.

3.9 Warum werden die PCMCIA-Treiber nicht binär vertrieben?

Das Vertreiben von binären Treibern ist schwierig, da einige Eigenschaften erst beim Übersetzen der Dateien angegeben werden können. Die PCMCIA-Module hängen auch von der richtigen Kernelkonfigutation und -version ab. Daher müssten binäre Versionen zusammen mit passenden Kerneln vertrieben werden. Der größte Bedarf an vorübersetzten Modulen besteht bei der Installation einer Linux-Distribution. In diesem Fall werden die PCMCIA-Module zum Teil direkt für die Installation benötigt, um z.B. über eine PCMCIA-Netzwerkkarte die Pakete der Distribution von einem Server zu beziehen.

PCMCIA ist heute Bestandteil der meisten Linux-Distributionen.

3.10 Warum ist das PCMCIA-Paket so groß?

Eigentlich ist das Paket nicht so groß. Alle Treibermodule zusammen benötigen weniger als 200 KB an Speicherplatz auf der Festplatte. Mit den zusätzlichen Werkzeugen werden es zusätzlich weitere 70 KB und das Verzeichnis

/etc/pcmcia belegt ungefähr 30 KB. Im Betrieb benötigen die Hauptmodule ungefähr 48 KB Hauptstpeicher. Der cardmgr-Daemon wird generell direkt ausgelagert und nur beim Kartenwechsel aktiv. Das gesamte Paket belegt nicht mehr Platz als die DOS-Varianten.

4 Gebrauch und Eigenschaften

4.1 Werkzeuge zur Überwachung der PCMCIA-Geräte

Der cardmgr-Daemon piept normalerweise, wenn eine neue Karte eingeführt wird. Der Piepton zeigt den Status der neuen Karte an. Zwei hohe Töne bedeuten, daß die Karte erkannt und konfiguriert wurde. Ein hoher und ein tiefer Ton zeigen an, daß die Karte erkannt aber nicht konfiguriert werden konnte. Lediglich ein tiefer Ton zeigt an, daß die neue Karte nicht erkannt wurde.

Wenn die Module korrekt geladen wurden, sollte das Kommando lsmod, ohne eingeführte Karten, folgende Ausgabe zeigen:

```
Module:         #pages:  Used by:
ds                  2
i82365              3
pcmcia_core         7    [ds i82365]
```

Alle PCMCIA-Module und der cardmgr-Daemon senden Statusmeldungen an den syslog-Daemon. Diese Meldungen werden dann gewöhnlich in die Datei /var/log/messages oder /usr/adm/messages geschrieben. Diese Dateien sollten bei der Fehlersuche als erstes untersucht werden. Wenn ein Fehlerreport geschrieben wird, sollte der Inhalt dieser Datei mitgeschickt werden. Wenn Probleme bestehen, die Systemmeldungen zu finden, sollte die Datei /etc/syslogd.conf daraufhin untersucht werden, wie die verschiedenen Nachrichtenklassen behandelt werden.

cardmgr speichert einige Informationen der aktuell genutzten Karten in jedem Slot in der Datei /var/run/stab. Hier ist ein Beispiel für den Inhalt einer solchen Datei:

```
Socket 0: Adaptec APA-1460 SlimSCSI
0       scsi    aha152x_cs      0       sda     8       0
0       scsi    aha152x_cs      1       scd0    11      0
Socket 1: Serial or Modem Card
1       serial  serial_cs       0       ttyS1   5       65
```

Im ersten Feld steht der verwendet Slot, das zweite enthält die Geräteklasse, das dritte den Treibernamen, das vierte wird verwendet, um die verschiedenen Geräte, die an den gleichen Treiber angeschlossen sind, durchzunumerieren. Das fünfte Feld ist der Gerätename und die letzten beiden Felder enthalten die Major- und Minor-Gerätenummern, falls diese angebbar sind.

Das Kommando cardctl kann verwendet werden, um den Status der Slots zu ermitteln oder zu sehen, wie sie konfiguriert sind. Hier ist eine Beispielausgabe des cardctl config Kommandos:

```
Socket 0:
Socket 1:
  Vcc = 5.0, Vpp1 = 0.0, Vpp2 = 0.0
  Card type is memory and I/O
  IRQ 3 is dynamic shared, level mode, enabled
  Speaker output is enabled
  Function 0:
```

```
Config register base = 0x0800
  Option = 0x63, status = 0x08
I/O window 1: 0x0280 to 0x02bf, auto sized
I/O window 2: 0x02f8 to 0x02ff, 8 bit
```

Wenn X läuft, wird das Programm `cardinfo` in einer grafischen Anzeige den Status aller PCMCIA-Slots anzeigen, ähnlich wie es `cardctl config` tut.

4.2 Überblick der PCMCIA-Konfigurationsskripte

Jedes PCMCIA-Gerät ist mit einer *Klasse* verknüpft, welche beschreibt, wie es konfiguriert und gehandhabt werden soll. Klassen sind mit Gerätetreibern verknüpft, die in `/etc/pcmcia/config` beschrieben sind. Aktuell sind dort fünf Ein-/Ausgabe-Geräteklassen (Netzwerk, SCSI, CD-ROM, Festplatten und serielle Geräte) und drei Speicherklassen (FTL, memory und pcmem) definiert. Zu jeder Klasse existieren zwei Skripte in `/etc/pcmcia/config`: Ein Hauptkonfigurationsskript, z.B. `/etc/pcmcia/scsi` für SCSI-Geräte, und ein Optionsskript, z.B. `/etc/pcmcia/scsi.opts`. Das Hauptskript für ein Gerät wird aufgerufen, um eine Karte zu konfigurieren, die gerade eingeschoben wird und um das Gerät herunterzufahren, wenn die Karte herausgenommen wird. Für Karten mit mehreren Geräten wird das Skript für jedes Gerät gestartet.

Das Konfigurationsskript entnimmt als erstes einige Informationen aus `/var/run/stab`. Jedes Skript bildet eine *Geräteadresse*, welche eindeutig das Gerät beschreibt, welches es konfiguriert, und speichert sie in der `ADDRESS` Variablen. Diese wird an das `*.opts` Skript weitergegeben. Dieses Skript soll dann die Informationen für die Konfiguration des Gerätes an dieser Adresse liefern. Bei einigen Geräten ist diese Adresse lediglich die Slotnummer, bei anderen enthält sie zusätzliche Informationen, die hilfreich bei der Konfiguration sein können. Zum Beispiel enthalten die Geräteadressen von Netzwerkkarten die Ethernetadressen. Auf diese Weise kann das `network.opts`-Skript diese Karte von anderen Netzwerkkarten unterscheiden und somit zwischen verschiedenen Konfigurationen wählen.

Der erste Teil aller Geräteadressen ist das aktuelle PCMCIA-Schema. Dieser Parameter wird verwendet, um zwischen verschiedene Sätzen von Konfigurationen zu wählen. Eine Anwendung der Schemata könnte es sein, eine Konfiguration für daheim und eine für die Arbeit zu haben, welche verschiedene Parameter für die Netzwerkkonfiguration benötigen. Das aktuelle Schema wird mit dem Kommando `cardctl` ausgewählt. Die Voreinstellung, wenn kein Schema gesetzt wird, ist `default`.

Eine generelle Regel für die Konfiguration von Linux auf Notebooks ist, daß alle PCMCIA-Geräte nur über die PCMCIA-Geräte-Skripte konfiguriert werden. Man sollte nicht versuchen, PCMCIA-Geräte wie permanent angeschlossene Geräte zu konfigurieren.

4.3 PCMCIA-Netzwerkkarten

Ethernetkarten unter Linux haben normalerweise Namen wie `eth0`, `eth1` und so weiter. Token-Ring-Karten werden ähnlich gehandhabt, allerdings haben sie Namen wie `tr0`, `tr1` und so weiter. Das Kommando `ifconfig` wird verwendet, um den Status von Netzwerkkarten zu erfragen oder zu ändern. Eine Besonderheit unter Linux ist, daß diese Netzwerkkarten keine entsprechenden Gerätedateien in dem Verzeichnis `/dev` besitzen. Daher sollte man sich nicht wundern, wenn man dort auch keine findet.

Wenn eine PCMCIA-Ethernetkarte entdeckt wird, so bekommt sie den ersten verfügbaren Schnittstellennamen; dieser wird wahrscheinlich `eth0` sein. `cardmgr` wird das Skript `/etc/pcmcia/network` starten, um die Karte zu konfigurieren.

Es ist nicht ratsam, die Konfiguration der PCMCIA-Ethernetkarte in die Startskripte des Linux-Systems einzutragen, da es passieren kann, daß die Karte noch nicht vorhanden ist, wenn Linux hochgefahren wird. Wenn das System eine automatische Prozedur zur Konfiguration des Netzwerks hat, so sollte hier angegeben werden, daß keine Netzwerkkarte vorhanden ist. Stattdessen sollte die Datei `/etc/pcmcia/network.opts` den Bedürfnissen des Netzwerks angepaßt werden. Die Skripte `network` und `network.opts` werden nur ausgeführt, wenn eine Ethernetkarte anwesend ist.

Die Geräteadresse, die `network.opts` übergeben wird, besteht aus vier durch Kommata getrennte Felder: Schema, Slotnummer, Geräteinstanz und die Hardware-Ethernetadresse der Karte. Die Geräteinstanz wird verwendet, um die Geräte von Karten, die mehrere Netzwerkschnittstellen enthalten, durchzunumerieren. Sie wird daher meistens 0 sein. Wenn mehrere Netzwerkkarten für verschiedene Verwendungen benutzt werden sollen, so besteht eine Möglichkeit darin, die Karten über ihre verschiedenen Slotnummern zu konfigurieren wie z.B. hier:

```
case "$ADDRESS" in
*,0,*,*)
    # Definition der Netzwerkkarte in Slot 0
    ;;
*,1,*,*)
    # Definition der Netzwerkkarte in Slot 1
    ;;
esac
```

Alternativ können diese Karten über ihre Hardware-Adressen konfiguriert werden:

```
case "$ADDRESS" in
*,*,*,00:80:C8:76:00:B1)
    # Definition einer D-Link Karte
    ;;
*,*,*,08:00:5A:44:80:01)
    # Definition einer IBM Karte
esac
```

Um automatisch NFS-Dateisysteme einzubinden oder zu entfernen, ist es sinnvoll, alle diese Dateisysteme als ersten Schritt in die Datei `/etc/fstab` einzutragen. Allerdings sollte im Optionsfeld der Eintrag `noauto` stehen. In der Datei `network.opts` müssen dann die Verzeichnisse, in die die NFS-Dateisysteme gemountet werden, in der Variablen `MOUNTS` aufgelistet werden. Es ist besonders wichtig, entweder `cardctl` oder `cardinfo` zu verwenden, um eine Netzwerkverbindung zu unterbrechen, wenn NFS-Dateisysteme auf diese Weise verwendet werden. Es ist nicht möglich, ein NFS-Dateisystem sauber abzubauen, wenn lediglich die Karte ohne Warnung herausgenommen wird.

Zusätzlich zur gewöhnlichen Netzwerkkonfiguration kann das Skript `network.opts` zusätzliche Funktionen ausführen, wenn die Netzwerkkarte schon konfiguriert worden ist oder bevor die Netzwerkverbindung abgebaut werden kann. Wenn `network.opts` eine Shellfunktion `start_fn` definiert, so wird diese nach der Konfiguration der Karte aufgerufen. Dabei wird der Schnittstellenname als einziges Argument übergeben. Analog wird die Funktion `stop_fn`, wenn sie definiert ist, aufgerufen, bevor die Netzwerkkarte heruntergefahren wird.

4.3.1 Auswahl des Transceivers

Der Transceiver-Typ kann durch Verwendung der Variablen `IF_PORT` bestimmt werden. Dies kann entweder ein numerischer Wert sein, wie er in früheren PCMCIA-Versionen verwendet wurde, oder ein Schlüsselwort, das den Transceiver-Typ bestimmt. Das voreingestellte Verhalten aller Netzwerktreiber ist es, den Typ automatisch zu erkennen, wenn dies möglich ist, oder 10baseT zu verwenden. Das Kommando `ifport` kann verwendet werden, um den aktuellen Typ zu kontrollieren oder zu ändern. Zum Beispiel:

```
# ifport eth0 10base2
```

```
# ifport eth0
eth0    2 (10base2)
```

Aktuelle Versionen des 3c589-Treibers versuchen, die Netzwerkverbindung automatisch zu entdecken. Allerdings scheint es so, als ob dies noch nicht einwandfrei funktioniert. Damit die automatische Erkennung läuft, sollte das Netzwerkkabel an der Karte angeschlossen sein, wenn diese konfiguriert wird. Alternativ kann man nach Anschluß des Netzes den Treiber zwingen, die Verbindung noch einmal zu überprüfen:

```
# ifconfig eth0 down up
```

4.3.2 Anmerkungen zu speziellen Karten

- Bei IBM CCAE und Socket EA Karten muß der Transceiver-Typ (10base2, 10baseT oder AUI) nach Konfiguration der Netzwerkkarte festgelegt werden. In der Systemlog-Datei sollte danach der richtige Transceiver-Typ erscheinen. Dies sollte man überprüfen.
- Die Treiber für SMC-, Megahertz-, Ositech- und 3Com-Karten sollten den angeschlossen Netzwerktyp (10base2 oder 10baseT) automatisch erkennen. Das Setzen des Transceiver-Typs, wenn der Treiber geladen wird, hilft der Karte, den richtigen Typ zu erraten.
- Die Farallon EtherWave-Karte basiert auf einer 3Com 3c589 mit einem speziellen Transceiver. Obwohl die EtherWave die 10baseT-Verbindungen verwendet, verlangt der Transceiver, daß der 3c589-Treiber den 10base2-Modus verwendet.
- Wenn Probleme mit den Karten IBM CCAE, NE4100, Thomas Conrad oder Kingston bestehen, sollte man die Speicherzugriffszeit erhöhen, indem man die Option `mem_speed=\#` beim Modul `pcnet_cs` verwendet. Ein Beispiel, wie dies zu machen ist, findet man in der Standarddatei `config.opts`. Man sollte Zeiten bis zu 1000 Nanosekunden verwenden.
- Für die New Media Ethernetkarte kann es bei einigen Systemen notwendig sein, die Zugriffszeit auf die I/O-Ports zu erhöhen. Dies geschieht mit der Option `io_speed=\#`, wenn das Modul `pcmcia_core` geladen wird. Um diese Option zu setzen, muß die Variable `CORE_OPTS` im Startskript editiert werden.
- Die Multicastunterstützung des New Media Ethernettreibers ist unvollständig. Der neueste Treiber wird mit den Multicast-Kerneln funktionieren, aber sämtliche Multicast-Pakete ignorieren. Der Promiscuous Modus sollte normal funktionieren.
- Der Treiber für IBM und 3Com Token-Ring-Karten scheint sich sehr schlecht zu verhalten, wenn die Karten bei der Initialisierung nicht mit einem Ring verbunden sind. Diese Karten sollten immer mit dem Netz verbunden werden, bevor sie eingeschoben werden.
- Neuere Linksys- und D-Link-Karten haben einen einzigartigen Weg, um den Transceiver-Typ auszuwählen. Bis David einige Informationen von Linksys erhält, ist der einzig gangbare Umweg der, daß zuerst DOS gestartet wird und die herstellertypischen Konfigurationsprogramme verwendet werden, um den Transceiver-Typ auszuwählen. Danach sollte ein Warmstart zu Linux erfolgen.

4.3.3 Untersuchung von Problemen mit Netzwerkkarten

- Wird die Karte als Ethernetkarte erkannt? Überprüfen Sie die Systemlog-Dateien, stellen Sie sicher, daß `cardmgr` die Karte korrekt erkennt und die richtigen Netzwerktreiber startet. Wenn dies nicht geschieht, kann es sein, daß die Karte trotzdem verwendbar ist, wenn sie mit einer unterstützten Karte baugleich ist. Dieses ist am einfachsten, wenn die Karte, was häufig der Fall ist, kompatibel zur NE2000 ist.

- Ist die Karte korrekt konfiguriert? Wenn eine unterstützte Karte verwendet wird und diese von `cardmgr` erkannt wird, aber dennoch nicht funktioniert, so kann ein Interrupt- oder Port-Konflikt mit einem anderen Gerät vorliegen. Dazu sollte man die Ressourcen, die die Karte verwendet, mit Hilfe der Systemlog-Datei herausfinden und versuchen, diese in der Datei `/etc/pcmcia/config.opts` auszuschließen, um die Karte zu zwingen, andere zu verwenden.

- Wenn Meldungen wie »network unreachable« bei Zugriff auf das Netzwerk erscheinen, dann wurde wahrscheinlich die Datei `/etc/pcmcia/network.opts` falsch editiert. Auf der anderen Seite verursachen falsch konfigurierte Karten keine Meldungen.

- Bei der Analyse der Probleme in der Datei `/etc/pcmcia/network.opts` sollte man als erstes mit dem Kommando `ping` ausprobieren, ob andere Rechner im gleichen Subnetz unter Verwendung ihrer IP-Adresse erreichbar sind. Danach sollte man versuchen, das Gateway und zum Schluß Rechner innerhalb anderer Subnetze mittels `ping` zu erreichen. Die Verwendung von `ping` mit Computernamen sollte erst nach diesem simpleren Test erfolgen.

- Man sollte sicherstellen, daß das Problem wirklich ein PCMCIA-Problem ist. Es kann hilfreich sein, zu kontrollieren, ob die Karte unter DOS mit den Treibern des Herstellers einwandfrei läuft. Man sollte die Änderungen in dem Skript `/etc/pcmcia/network.opts` doppelt kontrollieren. Der Anschluß des Kabels, das T-Stück, die Abschlußwiderstände etc. sollten ebenfalls gründlich überprüft werden.

4.4 Serielle Schnittstellen und Modems

Unter Linux wird auf serielle Schnittstellen über die speziellen Dateien `/dev/cua*` und `/dev/ttyS*` zugegriffen. Die `ttyS*` Dateien werden für eingehende Verbindungen (z.B. angeschlossenes Terminal) und die Dateien `cua*` für ausgehende Verbindungen (z.B. Modem) verwendet. Jeder physische Anschluß hat sowohl eine `ttyS*` als auch eine `cua*` Gerätedatei: Es hängt von einem selber ab, welche man verwendet. Jedoch ist ein Trend bei Linux zu erkennen, die `cua*` Devices nicht mehr zu benutzen. Debian GNU/Linux wird bereits ganz ohne die `cua*` Devices ausgeliefert. Die Konfiguration einer seriellen Schnittstelle kann mit dem Kommando `setserial` untersucht und verändert werden.

Wird eine serielle oder Modem-PCMCIA-Karte entdeckt, so erhält sie die erste freie Geräteadresse. Dies wird gewöhnlich, abhängig von der Zahl der im Computer bereits eingebauten seriellen Schnittstellen, `/dev/ttyS1` (`cua1`) oder `/dev/ttyS2` (`cua2`) sein. Die `ttyS*` Geräte sind diejenigen, die in `/var/run/stab` angegeben sind. Das Standardskript für serielle Geräte, `/etc/pcmcia/serial.opts`, wird das entsprechende `cua*` Gerät auf die Datei `/dev/modem` linken.

Man sollte nicht versuchen, das Systemstartskript für serielle Geräte zur Konfiguration von PCMCIA-Modems zu verwenden. Dieses Skript sollte nur zur Konfiguration nicht-entfernbarer Geräte benutzt werden. Für die spezielle Konfiguration eines Modems dient die Datei `/etc/pcmcia/serial.opts`. Auch sollte nicht das Kommando `setserial` verwendet werden, um die I/O-Adresse oder den Interrupt eines seriellen PCMCIA-Gerätes zu verändern. Dies würde den seriellen Treiber anweisen, das Gerät an einer anderen Stelle zu suchen. Dies würde aber nichts an der aktuellen Einstellung der PCMCIA-Karte ändern. Das serielle Konfigurationsskript erlaubt einem, sowohl `setserial`-Optionen anzugeben, als auch festzulegen, ob eine Zeile für diese Schnittstelle in die Datei `/etc/inittab` eingefügt werden soll.

Die Geräteadresse, die dem Skript `serial.opts` übergeben wird, besteht aus drei durch Kommata getrennte Felder: Das erste enthält das Schema, das zweite die Slotnummer und das dritte die Geräteinstanz. Letztere kann für Karten, die mehrere serielle Anschlüsse unterstützen, verschiedene Werte annehmen. Für Karten, die nur einen Anschluß haben, ist dieser Wert immer 0. Wenn gewöhnlich mehrere PCMCIA-Modemkarten verwendet werden, können diese durch die Slotnummer unterschiedlich konfiguriert werden, wie z.B.:

```
case "$ADDRESS" in
```

```
*,0,*)
    # Optionen Modem in Slot 0
    LINK=/dev/modem0
    ;;
*,1,*)
    # Optionen Modem in Slot 1
    LINK=/dev/modem1
    ;;
esac
```

Wenn ein PCMCIA-Modem bereits konfiguriert ist, wenn Linux gestartet wird, kann es passieren, daß das Modem fälschlicherweise als eine gewöhnliche fest eingebaute serielle Schnittstelle identifiziert wird. Dies ist harmlos. Wenn der PCMCIA-Treiber Kontrolle über das Modem übernimmt, wird diesem eine andere Gerätedatei zugewiesen. Es ist daher gut, entweder die Datei `/var/run/stab` zu durchforsten oder `/dev/modem` zu verwenden, anstatt zu erwarten, daß ein PCMCIA-Modem stets der gleichen Gerätedatei zugeordnet wird.

Wenn der Kernel konfiguriert wurde, die Grundtreiber für die seriellen Geräte als ein Modul zu laden, so muß die Datei `/etc/pcmcia/config` editiert werden, damit diese Module geladen werden. Ändern Sie in diesem Fall den seriellen Geräteeintrag auf diese Weise:

```
device "serial_cs"
  class "serial" module "char/serial", "serial_cs"
```

4.4.1 Analyse von Problemen mit seriellen Geräten

- Ist die Karte als ein Modem erkannt worden? Kontrollieren Sie die Systemlog-Dateien und stellen Sie sicher, daß `cardmgr` die Karte korrekt erkennt und den `serial_cs` Treiber startet. Wenn dies nicht funktioniert, muß eventuell ein neuer Eintrag in der Datei `/etc/pcmcia/config` gemacht werden, so daß die Karte richtig erkannt wird. Siehe Abschnitt 4.6 für mehr Details.

- Wurde das Modem erfolgreich durch `serial_cs` konfiguriert? Erneut sollte die Systemlog-Datei untersucht werden. Sind Meldungen vom `serial_cs` Treiber vorhanden? Wenn Einträge wie `register_serial() failed` erfolgt sind, kann es sein, daß Konflikte in der I/O-Adresse mit anderen Geräten vorliegen. Ein anderer Hinweis auf ein Problem ist die Meldung, daß die Karte eine `8250` sei. Die meisten modernen Modemkarten sollten als ein `16550A UART` erkannt werden. Bei dem Verdacht auf einen Adressenkonflikt sollte die Datei `/etc/pcmcia/config.opts` editiert werden und der Adressenbereich, den das Modem belegt, ausgeschlossen werden.

- Besteht ein Interruptkonflikt? Wenn die Systemlog-Datei gut aussieht, aber das Modem trotzdem nicht zu funktionieren scheint, sollte man versuchen, über das `setserial` Programm den Interrupt auf Null zu setzen und zu kontrollieren, ob das Modem so läuft. Dies veranlaßt das Modem, den langsameren Polling-Modus anstelle des Interrupt-Modus zu verwenden. Wenn dies das Problem zu lösen scheint, ist es wahrscheinlich, daß ein anderes Gerät den gleichen Interrupt wie das PCMCIA-Gerät verwendet. In diesem Fall sollte eine Zeile in `/etc/pcmcia/config.opts` eingefügt werden, die diesen Interrupt ausschließt.

- Wenn es scheint, daß das Modem wirklich sehr langsam läuft, ist dies meist ein sicheres Kennzeichen für einen Interruptkonflikt.

- Man sollte sicherstellen, daß es sich wirklich um ein PCMCIA-Problem handelt. Es kann hilfreich sein, sich zu vergewissern, ob die Karte unter DOS mit den Herstellertreibern funktioniert. Ebenso sollte die Karte nicht gleich mit derart komplexen Dingen wie SLIP getestet werden, bis man sicher ist, daß man einfache Verbindungen herstellen kann. Wenn einfache Verbindungen gelingen und SLIP fehlschlägt, so liegt das Problem wahrscheinlich bei SLIP und nicht bei PCMCIA.

4.5 PCMCIA SCSI-Controller

Alle derzeit unterstützten PCMCIA-Karten funktionieren genauso wie folgende ISA-Bus-Karten: Qlogic, Adaptec 152x oder Future Domain TMC-16x0. Die PCMCIA-Treiber werden durch Einbindung von einigem PCMCIA-spezifischen Programmcode (in `qlogic_cs.c`, `toaster_cs.c` oder `fdomain_cs.c`) aus den normalen Linux SCSI-Treibern gebildet.

Wenn ein SCSI-Controller entdeckt wird, wird von dem SCSI-Treiber nach neuen SCSI-Geräten gesucht. Die Systemlog-Dateien sollten Auskunft darüber geben, ob ein Gerät ordentlich erkannt wird. Den neuen SCSI-Geräten werden die ersten verfügbaren SCSI-Devices zugewiesen. Die erste SCSI-Festplatte wird `/dev/sda`, das erste Bandlaufwerk `/dev/st0` und das erste CD-ROM-Laufwerk wird `/dev/scd0` sein.

Mit Kernel 1.3.x und späteren Versionen sind die PCMCIA-Kerntreiber fähig, vom Kernel aus herauszufinden, welche SCSI-Geräte an der Karte angeschlossen sind. Diese werden in der Datei `/var/run/stab` aufgelistet und das SCSI-Konfigurationsskript, `/etc/pcmcia/scsi`, wird für jedes angeschlossene Gerät aufgerufen und zwar entweder um das Gerät zu konfigurieren oder um es herunterzufahren. Das Standardskript unternimmt nichts, um SCSI-Geräte zu konfigurieren, aber gemountete Dateisysteme werden ordentlich aus dem Verzeichnisbaum entfernt, wenn die PCMCIA-Karte aus dem Rechner genommen wird.

Mit Kernel 1.2.x können die PCMCIA-Treiber nicht automatisch erkennen, welches Gerät an welchem SCSI-Controller angeschlossen ist. Wenn man eine normale SCSI-Gerätekonfiguration hat, so kann man diese Geräte in der Datei `/etc/pcmcia/scsi.opts` auflisten. Zum Beispiel, wenn man normalerweise nur eine Festplatte und ein CD-ROM-Laufwerk verwendet, würde man folgendes verwenden:

```
# Kernel 1.2.x: Liste der angeschlossenen Komponenten
SCSI_DEVICE="sda scd0"
```

Die Geräteadressen, die dem Skript `scsi.opts` übergeben werden, sind kompliziert, da eine große Zahl verschiedenartiger Geräte an einen SCSI-Controller angeschlossen werden kann. Die Adressen bestehen entweder aus sechs oder sieben durch Kommata getrennte Felder: Das aktuelle Schema, der Gerätetyp, die Slotnummer, der SCSI-Kanal, die ID, die logische Einheitennummer (LUN) und eventuell die Partitionsnummer. Der Gerätetyp wird `sd` für Festplatten, `st` für Bandlaufwerke, `sr` für CD-ROM-Laufwerke und `sg` für generische SCSI-Geräte sein. Bei den meisten Einstellungen wird der SCSI-Kanal und die Einheitennummer 0 sein. Für Festplatten mit verschiedenen Partitionen wird `scsi.opts` erst für das gesamte Gerät mit einer fünf Felder enthaltenden Adresse aufgerufen. Das Skript sollte dann die Variable `PARTS` als eine Liste anlegen, die die Partitionen enthält. Danach wird `scsi.opts` mit einer sieben Felder enthaltenden Adresse für jede Partition aufgerufen. Hier ist zum Beispiel ein Skript zur Konfiguration einer Festplatte an SCSI ID 3 mit zwei Partitionen und einem CD-ROM-Laufwerk an SCSI ID 6:

```
case "$ADDRESS" in
*,sd,*,0,3,0)
    # diese Festplatte hat zwei Partitionen
    PARTS="1 2"
    ;;
*,sd,*,0,3,0,1)
    # Optionen Partition 1:
    # aktualisiere /etc/fstab und mounte ein ext2 fs
    # auf /usr1
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2"
    OPTS=""
    MOUNTPT="/usr1"
    ;;
*,sd,*,0,3,0,2)
    # Optionen Partition 2:
```

```
        # aktualisiere /etc/fstab und mounte ein MS-DOS fs
        # auf /usr2
        DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
        FSTYPE="msdos"
        OPTS=""
        MOUNTPT="/usr2"
        ;;
    *,sr,*,0,6,0)
        # Optionen CD-ROM an SCSI ID 6
        PARTS=""
        DO_FSTAB="y" ; DO_FSCK="n" ; DO_MOUNT="y"
        FSTYPE="iso9660"
        OPTS="ro"
        MOUNTPT="/cdrom"
        ;;
    esac
```

Wenn der Kernel keine Grundtreiber für besondere SCSI-Geräte wie Festplatten, Bandlaufwerke, etc. enthält, dann wird das Gerät nicht vom PCMCIA-Treiber konfiguriert. Als ein Nebeneffekt wird der Gerätename in `var/run/stab` etwas wie `sd\#nnnn` sein, wobei `nnnn` eine vierstellige Hexadezimalzahl ist. Dies passiert, wenn `cardmgr` nicht in der Lage ist, die SCSI-ID-Nummer in einen entsprechenden Linux-Gerätenamen zu übersetzen.

Es ist möglich, die SCSI-Grundtreiber für den Kernel zu modularisieren, so daß diese nur geladen werden, wenn ein PCMCIA-SCSI-Controller entdeckt wird. Um dies zu erreichen, muß die Datei `/etc/pcmcia/config` editiert werden, damit `cardmgr` weiß, welche zusätzlichen Module geladen werden müssen, um den Controller zu konfigurieren. Zum Beispiel

```
    device "aha152x_cs"
      class "scsi" module "scsi/scsi_mod", "scsi/sd_mod",
        "aha152x_cs"
```

würde das SCSI-Kernmodul und die Grundmodule für SCSI-Festplatten vor dem regulären PCMCIA-Treiber laden. Das PCMCIA-Konfigurationsskript wird nicht automatisch modularisierte SCSI-Module entdecken, so daß die SCSI-Unterstützung manuell in die Konfigurationsskripte eingetragen werden muß.

SCSI-Geräte sollten immer als erstes, vor dem Notebook oder dem Einführen der Karte, eingeschaltet werden, so daß der SCSI-Bus ordentlich terminiert ist, wenn der Controller konfiguriert wird. Man sollte auch äußerst vorsichtig sein, bevor man eine SCSI-Controllerkarte entfernt. In diesem Fall sollte man sicher sein, daß alle angeschlossenen SCSI-Geräte ordnungsgemäß heruntergefahren wurden. Der sicherste Weg dies zu erreichen ist es, entweder `cardctl` oder `cardinfo` zu verwenden und eine Kartenentfernung anzufordern, bevor man die Karte physisch entfernt. Bis jetzt müssen alle SCSI-Geräte eingeschaltet sein, bevor der SCSI-Controller eingeführt wird und sollten solange angeschlossen bleiben, bis der Controller wieder entfernt wird oder das Notebook ausgeschaltet wird.

Es besteht die Möglichkeit von Komplikationen bei Karten, die bei normalen ISA-Bus-Controllern nicht bestehen. Der SCSI-Bus enthält eine »Termination Power« Leitung, welche für den ordentlichen Betrieb von passiven SCSI Terminatoren notwendig ist. PCMCIA-SCSI-Controller speisen diese Leitung nicht. Falls die Leitung benötigt wird, muß sie von einem anderen SCSI-Geräte gespeist werden. Einige externe SCSI-Geräte können so konfiguriert werden, daß sie diese Leitung speisen. Andere, wie z.B. das ZIP-Laufwerk oder das Syquest EZ-Laufwerk verwenden aktive Terminierungen und benötigen diese Leitung daher nicht. In einigen Fällen kann es notwendig sein, einen speziellen externen Termininator wie den *APS SCSI Sentry 2* zu verwenden, der über ein eigenes Netzteil verfügt.

Der *Adaptec APA-460 SlimSCSI*-Controller wird nicht unterstützt. Diese Karte wurde ursprünglich unter dem Namen *Trantor* verkauft. Nach der Übernahme von Trantec durch Adaptec wurde dieser Controller als Teil der Adaptec-Produktserie verkauft. Der *APA-460* ist mit keinem existierenden Linuxtreiber kompatibel. David ist sich nicht sicher, wie schwer es sein wird, einen Treiber für diesen Controller zu schreiben. Er vermutet, daß niemand die dazu notwendigen technischen Informationen von Adaptec erhalten wird.

Der nicht unterstützte *Trantor SlimSCSI*-Controller kann auf folgende Weise erkannt werden:

```
Trantor / Adaptec APA-460 SlimSCSI
FCC ID: IE8T460
Shipped with SCSIworks! driver software
```

Der nicht unterstützte *Adaptec SlimSCSI*-Controller kann auf folgende Weise erkannt werden:

```
Adaptec APA-1460 SlimSCSI
FCC ID: FGT1460
P/N: 900100
Shipped with EZ-SCSI driver software
```

4.5.1 Untersuchung von Problemen mit SCSI-Controllern

- Bei dem `aha152x_cs` Treiber, dieser wird von Adaptec, New Media und ein paar anderen verwendet, scheint die Unterstützung von Disconnect/Reconnect eine häufige Fehlerquelle bei Bandlaufwerken zu sein. Um diese Eigenschaft abzuschalten, muß folgendes in der Datei `/etc/pcmcia/config.opts` hinzugefügt werden:

  ```
  module "aha152x_cs" opts "reconnect=0"
  ```

4.6 PCMCIA Speicherkarten

Das Standardstartskript für Speicherkarten erzeugt *Block-* und *zeichenorientierte* Geräte für den Zugriff auf alle Bereiche der Speicherkarte. Es gibt zwei Treiber für Speicherkarten. Einen älteren Treiber `pcmem_cs`, der gut mit einfachen statischen RAM-Karten zusammenarbeitet, und einen neueren Treiber `memory_cs`, der meistens für den direkten Zugriff auf Flashkarten verwendet wird. Für eine genauere Beschreibung der Gerätenamen sollte man die Manual Pages befragen. Beide, sowohl Block- als auch zeichenorientierte Gerätedateien, werden erzeugt. Die Blockgeräte werden für einen Festplattenähnlichen Zugriff verwendet (Erzeugung und Mounten von Dateisystemen etc.). Die zeichorientierten Gerätedateien werden für *rohe* (raw), nichtgepufferte Lese- und Schreiboperationen an beliebigen Stellen verwendet.

Bei den *FTL* und dem *neuen* Speichertreibern besteht die Geräteadresse, die an die Skripte `ftl.opts` und `memory.opts` weitergegeben wird, aus zwei Feldern, der Slotnummer und der Partitionsnummer. Gewöhnliche Speicherpartitionen werden vor Attributspeicherpartitionen numeriert. Allgemein ist die interessanteste Partition die mit der Nummer 0. Dieses ist die Hauptpartition, auf der die Daten gespeichert werden. Hier ist ein Beispiel eines Skriptes, das automatisch, abhängig vom verwendeten Slot, eine Flashspeicherkarte in das System integriert:

```
case "$ADDRESS" in
*,0,0)
    # Integriere Dateisystem, aber aktualisiere nicht
    # /etc/fstab
    DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2" ; OPTS=""
    MOUNTPT="/ftl0"
    ;;
*,1,0)
    # Integriere Dateisystem, aber aktualisiere nicht
    # /etc/fstab
    DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
```

```
        FSTYPE="ext2" ; OPTS=""
        MOUNTPT="/ftl1"
        ;;
esac
```

4.6.1 Einfache Speicherkarten

Einige ältere Speicherkarten und die meisten einfachen statischen RAM-Karten besitzen keine *Card Information Structure* (CIS), welche das Schema ist, das PCMCIA-Karten verwenden, um sich selbst zu identifizieren. Normalerweise wird `cardmgr` annehmen, daß Karten, die keine CIS aufweisen, einfache Speicherkarten sind und wird den `pcmem_cs` Treiber laden. Auf diese Weise erhält man den Nebeneffekt, daß andere unerkannte Karten fälschlicherweise als Speicherkarten erkannt werden.

Der `pcmem_cs` Treiber erzeugt drei logische Gerätedateien für eine Karte: `pcmem?a` ist eine Zeichengerätedatei zum Zugriff auf Attributspeicher, `pcmem?b` ist eine Blockgerätedatei und `pcmem?c` ist eine Zeichengerätedatei. Da alle PCMCIA-Karten eine Speicherschnittstelle neben allen anderen Funktionen benötigen, kann der `pcmem_cs` Treiber mit allen Karten verwendet werden, um direkten Zugriff auf den Attribut- und allgemeinen Speicherraum zu erhalten.

Der `pcmem_cs` Treiber verwendet ein Verfahren, um die Kapazität einer Karte zu raten. Dieses Verfahren schlägt bei schreibgeschützten Karten fehl und kann in einigen anderen Fällen zu Fehlern führen. Wenn eine Karte falsch erkannt wurde, sollte ihre Größe explizit angegeben werden, wenn man Kommandos wie `dd` oder `mkfs` verwendet.

4.6.2 Verwendung von Flashspeicherkarten

Um eine Flashspeicherkarte wie ein gewöhnliches festplattenähnliches Blockgerät zu verwenden, muß als erstes eine *Flash Translation Layer* Partition auf diesem Gerät mit dem Kommando `ftl_format` erstellt werden:

```
ftl_format -i /dev/mem0c0c
```

Man beachte, daß dieses Kommando auf die Karte über die *rohe* Speicherkartenschnittstelle zugreift. Einmal formatiert, kann die Karte wie ein normales Blockgerät über den `ftl_cs` Treiber verwendet werden:

```
mke2fs /dev/ftl0
mount -t ext2 /dev/ftl0 /mnt
```

Es gibt zwei wesentliche Formate für Blitzspeicherkarten: Das *Flash Translation Layer* und das *Microsoft Flash File System*. Das FTL-Format is generell flexibler, weil es erlaubt, irgendein gewöhnliches, anspruchsvolles Dateisystem wie ext2 oder MS-DOS FAT auf dieser Karte zu verwenden, genauso als ob es eine normale Festplatte wäre. Das FFS ist ein komplett neuer Typ von Dateisystem. Linux kann zur Zeit keine Karten handhaben, die mit diesem Dateityp formatiert sind.

4.7 PCMCIA ATA-/IDE-Karten

Die Unterstützung von ATA-/IDE-Treibern verlangt einen Kernel der Verison 1.3.72 oder höher. Der PCMCIA spezifische Teil des Treibers ist `fixed_cs`. Man sollte sicherstellen, daß `cardctl` oder `cardinfo` verwendet werden, um ATA-/IDE-Karten herunterzufahren, bevor sie entnommen werden.

Die Geräteadressen, die an `fixed.opts` weitergeleitet werden, bestehen aus drei oder vier Feldern: Das aktuelle Schema, die Slotnummer, die Seriennummer des Laufwerks und ebentuell die Partitionsnummer. So wie bei SCSI-Geräten wird `fixed.opts` zuerst für das gesamte Gerät aufgerufen. Wenn `fixed.opts` eine Liste von Partitionen in der Variablen `PARTS` enthält, wird das Skript noch einmal für jede Partition aufgerufen.

Hier ist eine Beispieldatei `fixed.opts`, die die erste Partition einer ATA-/IDE-Karte auf `/mnt` abbildet:

```
case "$ADDRESS" in
*,*,*)
    PARTS="1"
    ;;
*,*,*,1)
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/mnt"
    ;;
esac
```

Beachten Sie, daß die vorgegebene Datei `fixed.opts` diese Zeilen in auskommentierter Form enthält. Wenn es gewünscht wird, können verschiedene Konfigurationen, basierend auf den Seriennummern der Karten, verwendet werden. Um die Seriennummer einer Karte herauszufinden, kann das Kommando `ide_info` verwendet werden. Danach kann ein Teil von `fixed.info` wie folgt aussehen:

```
case "$ADDRESS" in
*,*,Z4J60542)
    # Dies ist eine DOS Platte
    PARTS="1"
    ;;
*,*,Z4J60542,1)
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/mnt"
    ;;
esac
```

4.8 Multifunktionskarten

Beginnend mit dem Linux-Kernel 1.3.73 kann ein einzelner Interrupt mit mehreren verschiedenen Treibern, wie z.B. dem seriellen Treiber und einem Ethernettreiber, geteilt werden. Wenn eine Multifunktionskarte unter einem neueren Kernel verwendet wird, können alle Funktionen ohne ein Ein- und Ausladen von Treibern verwendet werden.

Simultaner Gebrauch von zwei Kartenfunktionen ist trickreich, und verschiedene Hardwarehersteller haben das Teilen von Interrupts auf ihre eigene, nicht kompatible Weise verwirklicht. Die Treiber für einige Karten (*Ositech Jack of Diamonds* , *3Com 3c562*, *Linksys*) unterstützen den gleichzeitigen Zugriff ordentlich, während andere (besonders *Megahertz*) dies nicht tun.

Frühere Kernelversionen unterstützen das Teilen von Interrupts nicht, so daß es für PCMCIA-Treiber nicht möglich ist, gleichzeitig auf eine Ethernetkarte und ein Modem zuzugreifen. Die Treiber für das Ethernet und die serielle Schnittstelle werden automatisch geladen. Wie dem auch sei, der Treiber für das Ethernet besitzt per Voreinstellung den Interrupt der Karte. Um das Modem zu verwenden, muß der Ethernettreiber ausgeladen werden und der serielle neu konfiguriert werden wie z.B. hier:

```
ifconfig eth0 down
rmmod 3c589_cs
setserial /dev/modem autoconfig auto_irq
setserial /dev/modem
```

Das zweite `setserial` Aufruf soll überprüfen, ob der Treiber für das Modem jetzt den Interrupt verwendet, der vorher von dem Ethernettreiber verwendet worden ist.

4.9 Wann ist es sicher, eine PCMCIA-Karte einzuführen oder zu entfernen?

Theoretisch kann eine PCMCIA-Karte jederzeit eingeführt oder entfernt werden. Es ist jedoch eine gute Idee, eine Karte, die gerade von einem Programm in Gebrauch ist, nicht zu entfernen. Kernel die älter sind als Version 1.1.77 bleiben oft hängen, wenn eine serielle Schnittstellenkarte bzw. Modemkarte entfernt wird. Doch dies sollte mittlerweile behoben sein.

4.10 Card Services und Advanced Power Management

Card Services kann mit Unterstützung von APM kompiliert werden, wenn das Paket auf dem System installiert wurde. APM ist Bestandteil von Kernel 1.3.46 und neuer. Es wird derzeit von Rick Faith (`faith@cs.unc.edu`) betreut. APM-Werkzeuge können via FTP von

```
ftp.cs.unc.edu:/pub/users/faith/linux
```

bezogen werden. Die PCMCIA-Module werden automatisch für APM konfiguriert, falls eine kompatible Version auf dem System erkannt wird.

Ohne auf APM zurückzugreifen, kann der Befehl `cardctl suspend` vor dem Anhalten des Notebooks und der Befehl `cardctl resume` nach dem erneuten Anfahren des Notebooks verwendet werden, um die PCMCIA-Karten herunter- und wieder hochzufahren. Dies funktioniert nicht mit einem PCMCIA-Modem zusammen, das in Betrieb ist, da der serielle Treiber nicht in der Lage ist, die Arbeitsparameter des Modems zu sichern und wiederherzustellen.

APM scheint auf einigen System instabil zu sein. Wenn solche Beobachtungen im Zusammenhang mit APM und PCMCIA gemacht werden, sollte versucht werden, den Fehler auf eines dieser beiden Pakete einzuschränken, bevor ein Fehlerreport erstellt wird.

Einige Treiber, ganz besonders PCMCIA-SCSI-Treiber, können aus einem Anhalte- und Wiederanfahrzyklus nicht zurückkehren. Wenn eine PCMCIA-SCSI-Karte verwendet wird, sollte daher das Kommando `cardctl eject` vor einem Anhalten des Systems ausgeführt werden.

4.11 Wie wird eine PCMCIA-Karte abgeschaltet, ohne sie zu entnehmen?

Dazu kann entweder das Kommando `cardctl` oder `cardinfo` verwendet werden. `cardctl suspend \#` wird einen Slot anhalten und herunterfahren. Das entsprechende `resume` Kommando wird die Karte wieder in den ursprünglichen Zustand zurückversetzen.

4.12 Wie wird ein PCMCIA-Treiber entfernt?

Um das vollständige PCMCIA-Paket zu entfernen, muß das Skript `rc.pcmcia` folgendermaßen aufgerufen werden:

```
/etc/pcmcia/rc.pcmcia stop
```

Dieses Skript benötigt mehrere Sekunden, um zu laufen, da jedem Treiber Zeit gelassen wird, sanft herunterzufahren. Wenn ein PCMCIA-Gerät gerade in Benutzung ist, wird das Herunterfahren unvollständig sein und einige Module können im Kernel verbleiben. Um dieses zu vermeiden, sollten mit

```
cardctl eject
```

alle Slots heruntergefahren werden, bevor `rc.pcmcia` aufgerufen wird. Der Endestatus des `cardctl` Kommandos zeigt an, ob irgendein Slot nicht heruntergefahren werden konnte.

5 Anspruchsvollere Themen

5.1 Bereitstellung der benötigten Ressourcen für PCMCIA-Geräte

Theoretisch sollte es egal sein, welcher Interrupt von welchem Gerät verwendet wird, solange nicht zwei Geräte so konfiguriert sind, daß sie den gleichen verwenden. In der Datei `/etc/pcmcia/config.opts` können bestimmte Interrupts, die bei nicht-PCMCIA-Geräten Verwendung finden, ausgeschlossen werden.

Es ist nicht möglich, eine PCMCIA-Karte anzuweisen, eine bestimmte I/O-Adresse zu verwenden. Vielmehr erlaubt die Datei `/etc/pcmcia/config.opts` einem, einen Bereich von verfügbaren Adressen für den Gebrauch durch PCMCIA-Geräte anzugeben oder einen Bereich vom Gebrauch auszuschließen.

Nach der Modifizierung der Datei `/etc/pcmcia/config.opts` kann `cardmgr` mit dem Befehl

```
kill -HUP
```

neu gestartet werden.

Der Interrrupt, der verwendet wird, um den Kartenstatus zu überwachen, wird von dem Grundmodul (`i82365` oder `tcic`) ausgewählt, bevor `cardmgr` die Datei `/etc/pcmcia/config` durchforstet, so daß diese Einstellungen durch diese Datei nicht beeinflußt werden. Um diesen Interrupt zu setzen, muß die `cs_irq=` Option verwendet werden, wenn der Slottreiber geladen wird. Dies wird durch Setzen der Variable `PCIC_OPTS=` im Startskript `rc.pcmcia` erreicht.

Alle darauf aufbauenden Kartentreiber haben einen Parameter, der `irq_mask=` genannt wird und mit dem die Interrupts festgelegt werden, die von dem Treiber verwendet werden können. Jedes Bit von `irq_mask` entspricht einem Interrupt: Bit 0 ist IRQ0, Bit 1 IRQ1 und so weiter. Eine Maske von 0x1200 würde den Interrupts 9 und 12 entsprechen. Um einen Treiber derart einzuschränken, daß dieser nur einen Interrupt verwendet, darf lediglich ein Bit gesetzt werden. Diese Treiberoptionen sollten in der Datei `/etc/pcmcia/config` angegeben werden. Zum Beispiel

```
device "serial_cs"
  module "serial_cs" opts "irq_mask=0x1100"
  ...
```

würde bedeuten, daß nur die Interrupts 8 und 12 verwendet werden dürfen. Unabhängig von der Einstellung der Variablen `irq_mask` wird Card Services niemals einen Interrupt verwenden, der bereits von einem anderen Gerät benutzt oder durch die Datei `config` ausgeschlossen wird.

5.2 Wie können verschiedene Geräteeinstellungen für zu Hause und fürs Büro verwendet werden?

Dies ist wirklich eine einfache Anwendung der Unterstützung von PCMCIA-*Schemata*. Dazu verwendet man am besten zwei Konfigurationsschemata, genannt *Arbeit* und *Heim*. Hier ist ein Beispiel eines `network.opts` Skriptes mit schemaspezifischen Einstellungen:

```
case "$ADDRESS" in
Arbeit,*,*,*)
    # Definition der Netzwerkkarte im Arbeit Schema
    ...
    ;;
Heim,*,*,*|default,*,*,*)
    # Definition der Netzwerkkarte im zu Hause Schema
    ...
```

```
    ;;
esac
```

Der erste Teil einer PCMCIA-Geräteadresse ist immer das Konfigurationsschema. In diesem Beispiel wird der zweite Fall sowohl für das *Heim*, als auch für das *default* Schema verwendet. Wenn also das Schema aus irgendeinem Grund nicht gesetzt ist, wird automatisch das *Heim* Schema verwendet.

Um jetzt zwischen diesen beiden Sätzen von Einstellungen zu wählen, kann man eines dieser Kommandos starten:

```
cardctl scheme home
```

oder

```
cardctl scheme work
```

Das Kommando `cardctl` fährt alle Karten herunter und startet diese neu. Dieses Kommando kann sicher verwendet werden, unabhängig davon, ob das PCMCIA-System geladen ist. Es kann aber fehlschlagen, wenn zur gleichen Zeit andere Geräte verwendet werden. Dies ist unabhängig davon, ob diese Geräte von der Einstellung des aktuellen Schemas abhängen oder nicht.

Um das gerade eingestellte Schema herauszufinden, kann dieses Kommando verwendet werden:

```
cardctl scheme
```

5.3 Booten von einem PCMCIA-Gerät

Das Root-Dateisystem auf einem PCMCIA-Gerät zu haben, ist trickreich, da das PCMCIA-System von Linux nicht dazu gedacht ist, in den Kernel fest eingebunden zu werden. Die Kernkomponenten, die ladbaren Kernelmodule und der `cardmgr` Daemon hängen von einem bereits laufenden System ab. Die `initrd` Möglichkeit des Kernels umgeht diese Voraussetzungen dadurch, daß Linux erlaubt wird, vorübergehend ein RAM-Laufwerk als minimales Root-Dateisystem zu verwenden, die Treiber zu laden und dann ein anderes Root-Dateisystem an dessen Stelle zu mounten. Das temporäre Root-Dateisystem kann dazu verwendet werden, PCMCIA zu konfigurieren und dann ein PCMCIA-Gerät als Root-Dateisystem einzurichten.

Einige Linuxdistributionen erlauben eine direkte Installation auf Geräten, die direkt an einem PCMCIA-SCSI-Controller hängen, als unbeabsichtigten Nebeneffekt der Unterstützung der Installation von einem PCMCIA-SCSI-CD-ROM-Laufwerk. Kein Installationsprogramm unterstützt die Konfiguration von `initrd`, um Linux mit einem PCMCIA-Root-Dateisystem zu booten. Um so ein System einzurichten, benötigt man ein anderes Linuxsystem, um ein `initrd` Startprogramm zu erzeugen. Wenn ein anderes Linuxsystem nicht verügbar ist, kann eine andere Option die temporäre Installation eines minimalen Linuxsystems auf einem Nicht-PCMCIA-Laufwerk, die Erzeugung eines `initrd` Startprogramms und dann die Neuinstallation auf einem PCMCIA-Laufwerk sein.

Das Linux *Bootdisk HOWTO* enthält einige generelle Hinweise zur Erstellung von Bootdisketten aber nichts spezielles zu `initrd`. Die Hauptdokumentation zu `initrd` ist im Quellcode aktueller Kernelversionen in der Datei `linux/Documentation/initrd.txt` enthalten. Bevor man anfängt, sollte man diese Dokumentation lesen. Eine Vertrautheit mit LILO ist ebenfalls hilfreich. Die Verwendung von `initrd` erfordert, daß der Kernel mit den aktivierten Optionen `CONFIG_BLK_DEV_RAM` und `CONFIG_BLK_DEV_INITRD` übersetzt wurde.

5.3.1 Das Hilfsskript pcinitrd

Das Skript `pcinitrd` erzeugt ein `initrd` Grundstartprogramm zum Booten mit einem PCMCIA-Root-Dateisystem. Dies enthält eine minimale Verzeichnishierarchie, ein paar Gerätedateien, einige ausführbare Programme, Bibliotheken und einen Satz von PCMCIA-Treibermodulen. Wenn `pcinitrd` aufgerufen wird, müssen die

Treibermodule angegeben werden, die verwendet werden sollen. Die Kern-PCMCIA-Komponenten, `pcmcia_core` und `ds`, sind automatisch enthalten.

Als ein Beispiel für den Fall, daß das Notebook einen `i82365`-kompatiblen PCMCIA-Controller besitzt und Linux mit dem Root-Dateisystem auf einer Festplatte installiert werden soll, die an einem *Adaptec SlimSCSI* Controller angeschlossen ist, würde man folgenden Befehl verwenden:

```
pcinitrd -v initrd pcmcia/i82365.o pcmcia/aha152x_cs.o
```

Um die Startsequenz von `initrd` anzupassen, kann das Dateisystem mittels des `loopback` Gerätes eingefügt werden:

```
mount -o loop -t ext2 initrd /mnt
```

Danach sollte das Skript `linuxrc` editiert werden. Die PCMCIA-Konfigurationsdateien werden hierauf im Verzeichnis `/etc` installiert und können ebenfalls angepaßt werden. Dazu sollte die Manual Page von `pcinitrd` für weitere Informationen gelesen werden.

5.3.2 Erstellen einer initrd Bootdiskette

Nach der Erstellung durch `pcinitrd` kann durch Kopieren des Kernels, des gepackten `initrd` Startprogramms und einiger Hilfsdateien für LILO auf eine leere Diskette eine Bootdiskette erstellt werden. Im folgenden Beispiel nehmen wir an, daß das benötigte PCMCIA-Root-Dateisystem auf `/dev/sda1` liegen soll:

```
mke2fs /dev/fd0
mount /dev/fd0 /mnt
mkdir /mnt/etc /mnt/boot /mnt/dev
cp -a /dev/fd0 /dev/sda1 /mnt/dev
cp [kernel-image] /mnt/vmlinuz
gzip < [initrd-image] > /mnt/initrd
```

Erzeuge dann `/mnt/etc/lilo.conf` mit diesem Inhalt:

```
boot=/dev/fd0
compact
image=/vmlinuz
    label=linux
    initrd=/initrd
    read-only
    root=/dev/sda1
```

Zum Schluß muß `lilo` aufgerufen werden:

```
/sbin/lilo -r /mnt
```

Wenn `lilo` mit der Option `-r` aufgerufen wird, so wird es alle Aktionen relativ zu dem alternativ angegebenen Root-Dateisystem durchführen. Der Grund für das Erzeugen der Dateien unter `/mnt/dev` ist der, daß `lilo` nicht in der Lage ist, die Dateien in `/dev` zu verwenden, wenn es in diesem alternativen *root* Modus läuft.

5.3.3 Installierung eines initrd Startprogramms auf einem Nicht-Linux-Laufwerk

Eine häufige Verwendung der `initrd` Möglichkeit ist der Gebrauch auf Systemen, wo die eingebaute Festplatte einem anderen Betriebssystem gewidmet ist. Der Linux-Kernel und das `initrd` Startprogramm können auf einer Nicht-Linux-Partition untergebracht werden und LILO oder LOADLIN können so konfiguriert werden, daß sie Linux von dort starten können.

Ausgehend von der Annahme, daß der Kernel für das entsprechende Root-Laufwerk konfiguriert wurde und das `initrd` Startprogramm auf einem anderen System erstellt wurde, ist der einfachste Weg, Linux mit LOADLIN zu starten:

```
LOADLIN <kernel> initrd=<initrd-image>
```

Wenn Linux erst einmal auf der Zielmaschine gestartet wurde, dann kann LILO installiert werden, um Linux direkt zu starten. Als Beispiel sei `/dev/hda1` die Nicht-Linux-Zielpartition und `/mnt` ein freies Verzeichnis zum Mounten der Partition. Als erstes muß ein Unterverzeichnis für Linux auf dieser Partition geschaffen werden:

```
mount /dev/hda1 /mnt
mkdir /mnt/linux
cp [kernel-image] /mnt/linux/vmlinuz
cp [initrd-image] /mnt/linux/initrd
```

Wenn in diesem Beispiel `/dev/sda1`, ein SCSI-Laufwerk, auf welches über einen PCMCIA-Controller zugegriffen wird, die Root-Partition von Linux enthalten soll, so muß zur Installation von LILO eine Datei `lilo.conf` mit folgendem Inhalt erstellt werden:

```
boot=/dev/hda
map=/mnt/linux/map
compact
image=/mnt/linux/vmlinuz
        label=linux
        root=/dev/sda1
        initrd=/mnt/linux/initrd
        read-only
other=/dev/hda1
        table=/dev/hda
        label=windows
```

Die `boot=` Zeile besagt, daß LILO im *Master Boot Record* (MBR) des angegebenen Gerätes installiert werden soll. Die `root=` Zeile kennzeichnet das gewünschte Root-Dateisystem, welches später für den Gebrauch des `initrd` Startprogramms verwendet werden soll, und kann überflüssig sein, wenn der Kernel schon entsprechend konfiguriert worden ist. Der `other=` Abschnitt beschreibt das andere Betriebssystem, das auf `/dev/hda1` liegt.

Um LILO in diesem Fall zu installieren, sollte dies verwendet werden:

```
lilo -C lilo.conf
```

Beachten Sie, daß in diesem Fall die Datei `lilo.conf` absolute Pfadnamen verwendet, die `/mnt` enthalten. David tat dies in diesem Beispiel, da das Zieldateisystem eventuell die Einrichtung von Linux Gerätedateien für die `boot=` und `root=` Optionen nicht unterstützt.

6 Handhabung von Karten, die nicht unterstützt werden

6.1 Konfiguration nichterkannter Karten

Angenommen, daß die Karte von einem bestehenden Treiber unterstützt wird, so ist alles, was getan werden muß, ein Eintrag in die Datei `/etc/pcmcia/config`, um `cardmgr` mitzuteilen, wie die Karte identifiziert und welche Treiber für diese Karte geladen werden müssen. Die Manual Page von `pcmcia` gibt Auskunft über das richtige Format für diese Konfigurationsdatei. Wenn eine unbekannte Karte eingeführt wird, so notiert `cardmgr` normalerweise Identifikationsinformationen in den Systemlog-Dateien, die dazu verwendet werden können, einen `config` Eintrag zu erstellen.

Hier ist ein Beispiel, wie `cardmgr` eine nichtunterstützte Karte in den Systemlog-Dateien eintragen wird:

```
cardmgr[460]: unsupported card in socket 1
cardmgr[460]: version info: "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
```

Der entsprechende Eintrag in der Datei `/etc/pcmcia/config` würde in diesem Fall so aussehen:

```
card "Megahertz XJ2288 V.34 Fax Modem"
  version "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
  bind "serial_cs"
```

Es kann das Zeichen `*` verwendet werden, um Zeichenketten anzugeben, die nicht exakt übereinstimmen müssen, so wie z.B. Versionsnummern. Wenn Einträge gemacht werden, sollte darauf geachtet werden, daß die Zeichenketten exakt kopiert werden, also die Leerzeichen und die Groß- und Kleinschreibung beibehalten werden. Man sollte ebenfalls sicher sein, daß der Eintrag in `config` dieselbe Anzahl an Zeichenketten enthält, wie sie in der Log-Datei stehen.

Nach dem Editieren der Datei `/etc/pcmcia/config` kann `cardmgr` ein Signal gesendet werden, damit die Datei neu geladen wird:

```
kill -HUP `cat /var/run/stab`
```

Wenn ein neuer Eintrag in die `config`-Datei gemacht wurde, so sollte man David eine Kopie davon zuschicken, damit dieser Eintrag in die Standardkonfiugartionsdatei eingefügt werden kann.

6.2 Hinzufügen einer Unterstützung für NE2000 kompatible Ethernetkarten

Als erstes sollte kontrolliert werden, ob die Karte von `cardmgr` erkannt wird. Einige Karten, die nicht in der Datei `SUPPORTED.CARDS` stehen, sind OEM-Versionen von Karten, die bereits unterstützt werden. Wenn so eine Karte gefunden wird, sollte man David dieses mitteilen, damit er diese Karte in die Liste aufnehmen kann.

Wenn die Karte nicht erkannt wird, sollte man den Anleitungen im Abschnitt 4.6 folgen, um einen Konfigurationseintrag für diese Karte zu erstellen. Hierfür sollte die Karte an den Speicherkartentreiber, `pcmem_cs`, gebunden werden. Danach muß `cardmgr` neu gestartet werden, um die neue Konfigurationsdatei zu verwenden.

Man benötigt die Ethernet-Hardwareadresse der Karte. Diese Adresse ist eine Serie von sechs zweistelligen Hexadezimalzahlen, die oft direkt auf die Karte gedruckt ist. Wenn diese nicht direkt auf der Karte steht, kann meist auch der DOS Treiber verwendet werden, um die Kartennummer zu ermitteln. Wenn diese erst einmal bekannt ist, kann man folgenden Befehl aufrufen:

```
dd if=/dev/pcmem0a count=20 | od -Ax -t x1
```

In der Ausgabe dieses Kommandos sucht man jetzt nach der Adresse. Hat man diese gefunden, notiere man sich den Offset des ersten Byte der Adresse. Danach editiere man die Datei `modules/pcnet_cs.c` und finde die `hw_info` Struktur. Man hat nun einen neuen Eintrag für die neue Karte zu machen. Das erste Feld enthält einen beschreibenden Namen, das zweite den Offset mit zwei multipliziert. Die nächsten drei Felder enthalten die ersten drei Bytes der Hardwareadresse. Das letzte Feld enthält einige Einstellungen für spezielle Karteneigenschaften. Als erstes sollte man hier eine 0 versuchen.

Nach dem Editieren der Datei `pcnet_cs.c` muß diese kompiliert und das neue Modul installiert werden. Editieren Sie nun die Datei `/etc/pcmcia/config` erneut und wechseln Sie die Anbindung der Karte vom Modul `pc-mem_cs` zu `pcnet_cs`. Folgen Sie der Anleitungen zum erneuten Laden der Konfigurationsdatei, und alles sollte richtig eingestellt sein. Bitte senden Sie David eine Kopie der neuen `hw_info` und des `config` Eintrags.

Wenn die Hardwareadresse der Ethernetkarte in der hexadezimalen Ausgabe nicht gefunden werden kann, gibt es noch eine letzte Möglichkeit. Es ist möglich, die Adresse direkt anzugeben, wenn das `pcnet_cs` Modul initialisiert wird. Dazu muß die Datei `/etc/pcmcia/config` editiert werden und die Option `hw_addr=` eingefügt werden wie hier:

```
module "pcnet_cs" opts "hw_addr=0x00,0x80,0xc8,0x01,0x02,0x03"
```

Hier muß natürlich die eigene Hardwareadresse an den entsprechenden Stellen eingetragen werden.

6.3 PCMCIA-Schnittstellenkarten für Diskettenlaufwerke

Die Schnittstellenkarte für Diskettenlaufwerke, wie sie im *Compaq Aero* und einigen anderen Notebooks Verwendung findet, wird derzeit nicht unterstützt. Der Haken liegt hier darin, daß der Aero einen modifizierten Controller Chip verwendet, um einen DMA-Zugriff auf das Diskettenlaufwerk zu ermöglichen. Ohne zu wissen, wie dies genau abläuft, kann keine Unterstützung unter Linux bewerkstelligt werden.

Ist diese Gerätekarte für Diskettenlaufwerke anwesend, wenn der Aero eingeschaltet wird, so wird das BIOS des Aero die Karte konfigurieren und Linux wird sie als gewöhnliches Diskettenlaufwerk erkennen. Wenn die Linux-PCMCIA-Treiber geladen werden, so erkennen diese, daß diese Karte bereits konfiguriert und an eine Linux-Gerätedatei angeschlossen wurde. Dieser Slot wird dann in Ruhe gelassen. Auf diese Weise kann das Laufwerk verwendet werden, wenn es zur Bootzeit anwesend war. Aber es ist nicht möglich, diese Karte während der Laufzeit des Systems zu wechseln, zu entfernen und wieder einzuführen.

6.4 Was ist mit der Unterstützung von Xircom Karten?

Ein Treiber für die Unterstüzung von Xircom Ethernet- und Xircom Ethernet/Modem-Karten ist im aktuellen PCMCIA-Paket, dank der Mithilfe von Werner Koch, enthalten. David hat ein HyperNews-Forum speziell zur Diskussion über die Xircom-Treiberentwicklung unter folgender Adresse eingerichtet:

```
http://hyper.stanford.edu/HyperNews/get/pcmcia/xircom.html
```

Lange Zeit wurden Xircom-Karten nicht unterstützt, da Xircom die Firmenphilosophie verfolgte, keine technischen Informationen über ihre Karten zu verbreiten. Wie dem auch sei, sie haben diese Haltung gelockert und geben nun Treiberinformationen weiter.

7 Tips für die Fehlersuche und Informationen zur Programmierung

7.1 Wie kann ein hilfreicher Fehlerreport verschickt werden?

Der beste Weg, einen Fehlerreport zu versenden, besteht darin, die HyperNews-Mitteilungsliste der PCMCIA-Homepage zu verwenden. Auf diesem Weg können andere Anwender die aktuellen Probleme und deren Lösungen oder Vermeidung, falls möglich, sehen. Hier sind einige Dinge, die in einem Fehlerreport enthalten sein sollten:

- Der Systemtyp und die Ausgabe des Kommandos `probe`.
- Welche PCMCIA-Karten werden verwendet?
- Version des Linuxkernels und des PCMCIA-Treibers.
- Alle Änderungen, die in den Startdateien im Verzeichnis `/etc/pcmcia` oder dem PCMCIA-Startskript gemacht wurden.
- Alle im Zusammenhang mit PCMCIA stehenden Mitteilungen in der Systemlog-Datei.

Vor dem Abschicken des Fehlerreports sollte man sicher sein, daß die neuesten PCMCIA-Treiberpakete verwendet wurden. Es ist keine sehr gute Sache, die Zeit mit dem Lesen von Fehlerreporten zu verschwenden, wenn diese Fehler längst behoben wurden.

Wenn das Problem mit einem Kernelfehler einhergeht, so ist der Registerausdruck nur dann hilfreich, wenn die fehlerhafte Adresse, EIP, herausgefunden werden kann. Wenn diese im Hauptkernel liegt, kann die Adresse dazu verwendet werden, mittels der Datei `System.map` die fehlerhafte Funktion herauszufinden. Wenn der Fehler in einem ladbaren Modul liegt, ist diese Funktion ein wenig schwieriger herauszufinden. Von den aktuellen Modulwerkzeugen wird `ksyms -m` die Basisadressen der Module anzeigen. Man nehme dann das Modul, welches die EIP-Adresse enthält und subtrahiere die Basisadresse von der EIP, um einen Offset innerhalb des Moduls zu bekommen. Danach kann das `gdb` Programm auf das Modul angewendet werden. Mit dem `list` Kommando kann dann dieser Offset betrachtet werden. Dies wird aber nur funktionieren, wenn das Modul mit der Option `-g` zum Einbinden von Debugginginformationen übersetzt wurde.

Wenn kein Zugang zum WWW besteht, können Fehlerberichte direkt an David Hinds geschickt werden. David bevorzugt es aber, daß diese Fehlerberichte direkt an die Website geschickt werden, da sie hier auch von anderen gelesen werden können.

7.2 Hilfe zur Fehlersuche auf niedrigster Stufe

Die PCMCIA-Module enthalten eine Menge Debuggingcode, der beim Übersetzen eingebunden werden kann. Der meiste Code steht unter Kontrolle der `PCMCIA_DEBUG` Präprozessordefinition. Wenn diese undefiniert ist, wird der meiste Fehlersuchcode nicht übersetzt. Wenn dieser Wert auf 0 gesetzt ist, wird dieser Code zwar übersetzt, ist jedoch inaktiv. Größere Werte spezifizieren einen höheren Grad der Mitteilungsbereitschaft. Jedes Modul, das mit definiertem `PCMCIA_DEBUG` übersetzt wurde, enthält einen ganzzahligen Parameter `pc_debug`, der die Meldebereitschaft des Moduls bestimmt. Dieser kann während des Ladens eines Moduls modifiziert werden, so daß die Ausgabe auf Modulbasis angegeben werden kann, ohne daß der Code neue übersetzt werden muß.

Es sind einige Werkzeuge zur Fehlersuche in dem Unterverzeichnis `debug_tools` der PCMCIA-Distribution enthalten. Die Programme `dumb_tcic` und `dumb_i365` erstellen komplette Registerauszüge der PCMCIA-Controller und entschlüsseln eine Menge der Registerinformationen. Diese sind besonders hilfreich, wenn man Zugang zu einem Datenblatt des zugehörigen Controller-Chips hat. Das Programm `dump_tuples` listet den Inhalt der CIS (Card Information Structure) auf und entschlüsselt einige interessante Bits. Mit dem Programm `dump_cisreg` können die lokalen Konfigurationsregister einer Karte angezeigt werden.

Der `pcmem_cs` Speicherkartentreiber kann manchmal auch sehr hilfreich sein. Dieser kann an irgendeine Karte gebunden werden und hat keine Wechselwirkungen mit anderen Treibern. Er kann verwendet werden, um direkten Zugriff auf den Attribut- oder allgemeinen Speicher einer Karte zu erlangen.

7.3 Wie schreibt man einen Treiber für eine neue Karte?

Der *Linux PCMCIA Programmer's Guide* ist die beste Dokumentation der Linux-PCMCIA-Schnittstelle. Die neueste Version ist immer über FTP erhältlich bei

```
hyper.stanford.edu:/pub/pcmcia/doc
```

oder per WWW bei

```
http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html
```

Bei Geräten, die den normalen ISA-Geräten nahe stehen, kann man wahrscheinlich einen Großteil der existierenden Linuxtreiber verwenden. In einigen Fällen wird der größte Brocken die Anpassung existierender Treiber sein, so daß diese das Hinzufügen und Entfernen der Geräte nach dem Booten handhaben können. Von den aktuellen Treibern ist der Speicherkartentreiber der einzige eigenständige Treiber, der nicht von den Teilen des Linux Kernels abhängt und so die meiste Arbeit leisten muß.

David hat einen Treiber als Grundgerüst für eigene Treiber geschrieben, der mit einer Menge an Kommentaren erklärt, wie die Kommunikation zwischen dem Treiber und den *Card Services* abläuft. Dieser kann in der PCMCIA-Quelldistribution unter `modules/skeleton.c` gefunden werden.