

ISDN mit OpenBSD 2.8

ANDREAS KREBS

Leider gibt es OpenBSD nicht mit standardmäßiger ISDN-Unterstützung, aber das heißt ja noch lange nicht, daß das unmöglich ist. Dank einiger fleißiger Leute gibt es das unter BSD bekannte isdn4bsd in einer angepaßten Version für OpenBSD 2.8 und ermöglicht vollen ISDN-Support.



Das ISDN-Paket für diverse BSD-Varianten nennt sich `isdn4bsd` (kurz »i4b«) und wurde von Hellmuth Michaelis geschrieben. Es ist Bestandteil von FreeBSD seit der Version 3.0 des Betriebssystems. Leider gibt es keine Unterstützung für OpenBSD 2.6 oder neuer, da diese Portierung aus Zeitmangel vom Entwickler eingestellt wurde. Doch glücklicherweise hat Jörg Bornschein das i4b-Paket gepatcht, so daß man auch unter OpenBSD 2.8 volle ISDN-Unterstützung hat.

An Hardware werden fast alle halbwegs bekannten ISDN-Karten unterstützt:

ISA-Karten:

- Teles S0/8 oder Niccy 1008.
- Teles S0/16 oder Creatix ISDN-S0 oder Niccy 1016 (getestet).
- Teles S0/16.3 (getestet).
- AVMA1 oder AVM Fritz!Card (getestet).
- USRobotics Sportster ISDN TA intern.
- Teles S0/16.3 PnP.
- Creatix ISDN-S0 P&P.

PCI-Karten:

- ELSA Quickstep 1000.
- AVM Fritz!Card PCI (getestet).

Kernel-PPP funktioniert noch nicht, dafür bietet der Userlevel-pppd-Daemon mehr Features (wie Kanalbündelung und Dialin-Support) und vor allen Dingen: er funktioniert.

Allerdings muß man auch den pppd-Daemon patchen und selbst kompilieren, da die von OpenBSD mitgelieferte Version weder HDLC noch isdn-Signaling unterstützt. Beides wird aber gebraucht. HDLC ist das ISDN-Protokoll, das vom PPP-Daemon benutzt wird, ähnlich zum Beispiel wie V.34 oder V.90 bei analogen Modems. Daraufsetzt dann ppp auf.

Die Dateien und die Installation

Die benötigten Patches oder gleich die fertig gepatchten Pakete für i4b und PPP findet man unter »<http://www.zilium.de/~joerg/i4b/>«. Hier befinden sich auch noch Dateien für Beispiel-Konfigurationen, die man sich auf jeden Fall ansehen sollte. Da aber auch der Kernel gepatcht wird, sollte man sicherstellen, daß die Kernelsourcen vorhanden sind. Man findet sie zum Beispiel unter »<ftp://ftp.de.openbsd.org/unix/OpenBSD/2.8/srcsys.tar.gz>«. Die aktuellen Versionen der benötigten Programme befinden sich auch auf der CD-ROM zu dieser Ausgabe der freeX.

Es wird davon ausgegangen, daß alle nachfolgenden Befehle mit `root`-Rechten ausgeführt werden und die Pakete unter »/root« liegen. Mit folgenden Befehlen werden die benötigten Verzeichnisse angelegt und die Quellen des Kernels und des bereits gepatchten i4b-Paketes entpackt:

```
# mkdir -p /usr/src/sys
# cd /usr/src/sys
# tar -xzf ~/srcsys.tar.gz
# cd /usr/src
# tar -xzf \
~/i4b-0.90.0-OpenBSD-20010116.tar.gz
```

Anschließend wechselt man in das i4b-Verzeichnis, liest sich die INSTALL-Anleitung durch und führt die nachfolgenden Befehle aus, um die Kernelquellen zu patchen und die erforderlichen ISDN-Devices anzulegen:

```
# cd i4b/OpenBSD
# more INSTALL
# sh ./install-openbsd.sh
# cd /dev
# sh /usr/src/i4b/OpenBSD/MAKEDEV.i4b \
all
```

Nun kommt der interessante Teil des Kernelbauens. Hier sollte man auf jeden Fall die Online-Dokumentation zu diesem Thema durchgelesen haben (»<http://www.openbsd.org/faq/faq5.html>«). Ich werde hier nicht näher darauf eingehen, sondern die Vorgehensweise exemplarisch am i386-GENERIC-Kernel und einer Fritz!Card PCI zeigen.

```
# cd usr/src/sys/arch/i386/conf
# vi GENERIC
```

Hier die Zeilen, die einfach am Ende der Datei »GENERIC« eingefügt werden:

```
# ISDN stuff
#
option AVM_A1_PCI

isic0 at pci ?

pseudo-device "i4b"
pseudo-device "i4btrc" 2
pseudo-device "i4bctl"
pseudo-device "i4brbch" 4
pseudo-device "i4bte1" 2
```

Anschließend wird die Datei gespeichert, der Editor verlassen und dann *config*, *make depend* und *make* eingegeben, um den angepaßten Kernel zu kompilieren. Dies kann eine Weile dauern. Anschließend wird eine Sicherheitskopie des alten Kernels erstellt (man kann ja nie wissen) und der neu kompilierte Kernel an die Stelle des alten kopiert:

```
# config GENERIC
# cd ../compile/GENERIC
# make depend
# make
# cp /bsd /bsd.old
# cp /sys/arch/i386/compile/GENERIC/bsd \
/bsd
```

Schließlich müssen noch die Quellen für den gepatchten ppp-Daemon entpackt werden. Anschließend wird er kompiliert und installiert:

```
# cd /usr/src/i4b
# cp -Rp machine /usr/include/i4b
# cd /root
# tar -xzf ppp-20000608-OpenBSD.tar.gz
# cd ppp-OpenBSD
# make && make install
```

Jetzt kommt der spannende Moment des Neustarts. Lief alles glatt, sollten die Kernmeldungen Sie mit ein paar Meldungen zur gefundenen ISDN-Karte beglücken. Damit ist die Karte jetzt im System aktiv.

Die Konfiguration

Um die Karte auch nutzen zu können, muß dem ppp-Daemon mitgeteilt werden, daß er jetzt die ISDN-Karte verwenden soll. Erst einmal sollte man sich allerdings noch ein paar Man-Pages zu Gemüte führen, und zwar zum Thema »ppp«, »ppp.conf«, »isdnd« und »isdnd.rc«. Eine ISDN-ppp-Verbindung einzu-

richten ist fast so einfach wie über Modem, es müssen nur zwei zusätzliche Einträge in der »/etc/ppp/ppp.conf« gemacht werden:

```
set device /dev/i4brbch0
set speed sync
```

Hier in paar Auszüge aus einer Beispielkonfiguration in der »/etc/ppp/ppp.conf«.

```
default:
set device /dev/i4brbch0
set speed sync
set login
set hangup
add default HISADDR
```

```
isp:
set authname Loginname
set authkey Passwort
set phone ISP-Telefonnummer
```

Hier werden unter *default* zuerst das zu verwendete Device und die Geschwindigkeit festgelegt, ferner die Benutzung des Login- und Hangup-Skripts. Die letzte Zeile legt den Provider als Defaultrouter fest. Der

```
#####
# SYSTEM section: isdnd global configuration parameters
#####
system
# accounting
acctall = on # generate info for everything
acctfile = /var/log/isdnd.acct
useacctfile = yes # generate accounting info to file
# monitor
monitor-allowed = off # global switch: monitor on/off
# realtime priority section
rtprio = 25 # modify isdnd's process priority
#####
# example configuration file for the isdn daemon and userland ppp.
# (see the ppp setup examples in i4b's user-ppp dir and FAQ item 22 !)
#####
entry
name = uppp0 # name of entry
usrdevicename = rbch # use the raw b channel access device
usrdeviceunit = 0 # unit 0
isdncontroller = 0 # use isdn controller 0
isdnchannel = -1 # use any channel on this controller
local-phone-incoming = 123456789
remote-phone-incoming = *
local-phone-dialout = 123456789
remote-phone-dialout = * # ppp(8) will override this
remdial-handling = first
dialin-reaction = ignore # no dialins allowed
dialout-type = normal
blprotocol = hd1c # b channel layer 1 protocol
idletime-incoming = 0 # Should be larger than ppps timeout
idletime-outgoing = 0 # Should be larger than ppps timeout
ratetype = 0
unitlength = 90
unitlengthsrc = rate
dialretries = 1
dialrandincr = on
recoverytime = 25
usedown = off
downtries = 2
downtime = 30
```

Listing 1: Ein Beispiel der Datei »/etc/isdn/isdnd.rc«

EuroBSD, eine neue Plattform

Die BSD-Organisationen werden ab sofort eine gemeinsame Kommunikationsplattform betreiben: »<http://www.eurobsd.org>«. Die EuroBSD.org soll zu einer Anlaufstelle für alle Aktivitäten rund um BSD werden, die im europäischen Raum stattfinden.

Das Ziel von EuroBSD ist, User zusammenzubringen. Dabei geht es um den Zusammenschluß von Usergruppen und die Vorbereitung der Planung gemeinsamer Aktivitäten. Dies betrifft beispielsweise Stände auf Veranstaltungen wie den Linuxtagen und Informationsveranstaltungen rund um freie Software. Des weiteren sollen Kontakte zur Wirtschaft gepflegt werden – Hardware-Leihstellungen, Jobsuche und kommerzielle Veranstaltungen im Unix-Umfeld. BSD in diesem Sinne sind die drei freien Open-Source-BSDs (Free-, Net-, OpenBSD), sowie das »alte« Berkeley BSD-Unix, sowie die kommerziellen Varianten BSD/OS und MacOS X, sofern die technischen Aspekte des BSD-basierten Untergrunds betroffen sind. Entwicklungen, die auf BSD basieren, sind natürlich ebenfalls herzlich willkommen. Dabei steht ausdrücklich der Aspekt der Kommunikation im Umfeld der Open-Source-Community unter dem Aspekt einer freien Software im Vordergrund. Kooperation und Gedankenaustausch mit Usern anderer freier Software werden hierbei großgeschrieben.

Je nach Nachfrage der Userschaft von EuroBSD.org ist es denkbar, daß

- eine europaweite Liste der Veranstaltungen rund um freie Software gepflegt wird (hierbei ist selbstverständlich die Mithilfe der Community erforderlich),
- Diskussionsforen für Mitglieder entstehen,
- eine Kontaktbörse (beispielsweise für Jobsuche/Angebote, Freiberufler oder Merchandise-Versand) betrieben wird,
- Tausch und Kauf von Hardware, beispielsweise alter Workstations, stattfindet.

Natürlich ist die Mithilfe der Community erforderlich, um beispielsweise Termine von Veranstaltungen wie Linuxtagen oder ähnlichem zu erfahren, aber auch deren aktive Mitarbeit an der inhaltlichen Gestaltung ist erwünscht.

Kontakt: info@eurobsd.org

Abschnitt *isp* enthält die benötigten Zugangsinformationen des Providers wie Loginname und Paßwort für PAP/CHAP-Authentifikation und die Telefonnummer.

Ein Beispiel der Datei »*/etc/isdn/isdn.rc*« finden Sie in Listing 1. Die Erklärungen zu den einzelnen Einträgen finden sich als Kommentar hinter den jeweiligen Optionen.

Auch das Logging sollte noch eingerichtet und der *isdnd*-Daemon in die Boot-Sequenz aufgenommen werden. Folgender Eintrag ist für das Logging in der »*/etc/syslog.conf*« zuständig:

```
local2.info                /var/log/isdnd          #ifconfig -a
```

anschließend wird noch die Datei für das Logfile selbst mit dem Befehl

```
# touch /var/log/isdnd
```

angelegt. Für die Logfile-Rotation wird noch ein Eintrag in »*/etc/newsyslog.conf*« benötigt, damit das Logfile nicht überläuft, sondern bei einer bestimmten Größe ein neues angefangen wird. Folgende Zeile ist hierfür zuständig:

```
/var/log/isdnd 1 600 10 * 168 Z
```

Schließlich wird der *isdnd*-Daemon noch in die »*/etc/rc.local*« eingetragen, damit er beim Booten startet:

```
# Starting isdnd.
if [ -x /usr/local/bin/isdnd ]; then
    echo -n ' isdnd';
    # -d0x95 is a quite verbose
    # debugginglevel
    # -s 18 corresponds to syslog
    # facility local2
    # -s 3 would enable logging to
    # /var/log/daemon just like ppp
    /usr/local/bin/isdnd -d0xf9 -s 18
fi
```

Ob alles fertig konfiguriert ist, muß man nur noch einmal mit einem Aufruf von

```
#ppp -auto isp
```

testen. Hiermit sollte jetzt die Verbindung aufgebaut werden (*isp* bezieht sich übrigens auf den genauso benannten Abschnitt aus der »*/etc/ppp/ppp.conf*«, man kann also auch mehrere verschiedene Provider dort eintragen und einfach durch die Angabe bei der Einwahl auswählen), was durch

überprüft werden kann. Hier sollte jetzt zusätzlich das Device »*tun0*« mit der zugewiesenen IP-Adresse zu sehen sein.

Fazit

Der *ppp*-Daemon bietet eine Unzahl an Konfigurationsmöglichkeiten, die hier nicht alle erläutert werden können, da dies den Umfang des Beitrags sprengen würde. Deshalb ist es sehr sinnvoll, sich die entsprechenden Man-Pages genau durchzulesen und gegebenenfalls die Konfiguration den Bedürfnissen entsprechend anzupassen. Zusätzlich finden sich auf den Webseiten noch mehr Informationen zu den Themen Dialin-Support und Callback.

Durch den Einsatz von Hellmuth Michaelis und Jörg Bornschein ist die Nutzung von ISDN also auch unter OpenBSD möglich. Allerdings sind die Entwickler auch auf die Mithilfe der Benutzer angewiesen, damit die Software in Zukunft noch besser wird und mehr Hardware unterstützt werden kann.

Adressen

ISDN on OpenBSD von Alex Hoffmann: »<http://www.unrest.de>«.

Homepage des isdn4BSD-Projekts: »<http://www.freebsd-support.de/i4b>«.

Homepage des PPP-Dämons: »<http://www.awfulhak.org/ppp.html>«.

OpenBSD, Manpages usw.: »<http://www.openbsd.org>«.