

Retten mit NetBSD

VON JÖRG BRAUN

Wenn man auf einmal ein Betriebssystem nicht mehr booten kann oder mit den Bordmitteln nicht mehr an eine Datenpartition gelangt, ist guter Rat teuer. Manchmal ist der Grund für das Malheur Leichtsinn, aber manchmal auch einfach ein Betriebssystemproblem.



Alle drei freien BSD-Betriebssysteme nutzen als Standard-Dateisystem prinzipiell das selbe Dateisystem, nur einmal trägt es den Namen UFS (für Unix File System), einmal heißt es FFS (für Fast File System). In früheren Zeiten besaßen auch alle die gleiche Partitions-ID A5h. Auch heute, wo OpenBSD eine neue Kennung A6h und NetBSD ein A9h möchte, ist es bei den beiden kleineren BSD-Systemen durchaus möglich, auf Partitionen mit der Kennung A5h zuzugreifen. Man sieht dann bei NetBSD beispielsweise den Hinweis, daß es sich entweder um eine 386BSD/FreeBSD oder eine alte NetBSD-Partition handelt, erst vor wenigen Jahren wurde nämlich auf die neue Kennung gewechselt. Während es von NetBSD aus durchaus möglich ist, FreeBSD-Partitionen zu lesen und auch zu schreiben, geht das umgekehrt normalerweise nicht. Es gibt allerdings einen externen GEOM-Treiber, der als Kernelmodul in FreeBSD geladen wird und dann klappt auch der Zugriff auf NetBSD-Partitionen. Dann wird es problemlos möglich, FreeBSD (das zuerst) und NetBSD (nach FreeBSD) gemeinsam auf einer Festplatte zu installieren. NetBSD kommt mit den dann recht merkwürdigen Label-Bezeichnungen gut zurecht – soviel nur am Rande bemerkt.

Allerdings wird es im Normalfall nicht üblich sein, mehrere BSDs auf einer Festplatte zu installieren. Sollen auf einem PC mehrere BSDs installiert werden, bietet es sich auch eher an, sie auf verschiedenen Festplatten zu

verteilen, was im Regelfall die unerwünschten Nebenwirkungen verhindert. Speziell nach dem Einbau einer neuen Festplatte und dem anschließenden Einrichten eines weiteren Betriebssystems kann man aber dann doch die unangenehmen Überraschungen erleben. Im speziellen Fall sind auf einem recht leistungsfähigen Test-PC auf mehreren SATA- und SCSI-Platten FreeBSD/amd64, FreeBSD/i386, NetBSD, Linux und Windows Vista installiert. Diesem Zoo wurde jetzt OpenBSD 4.4 hinzugefügt. Speziell beim Kompilieren großer Softwarepakete wie OpenOffice.org ist es recht sinnvoll, diese Arbeit dem schnellen Rechner aufzuhalten. Nach der Installation von OpenBSD lag es nahe, diverse Einstellungen von einem der anderen installierten Betriebssysteme zu übernehmen und beispielsweise die X-Konfiguration nicht vom Scratch durchzuführen, sondern nur anzupassen. Dazu wurde die Root-Partition einer der beiden FreeBSD-Installationen eingebunden. Da OpenBSD wie auch NetBSD on-the-fly ein Label auch für fremde Platten vergibt und auch FreeBSD-Partitionen als FFS-formatiert erkennt (siehe Bild 1), ist das kein Problem, sollte man meinen.

Die Überraschung tritt beim nächsten Booten von FreeBSD auf, das nicht mehr will:

```
error 187 1ba 120
error 187 1ba 120
no /boot/loader
```

(und so weiter).

Hat man ein zweites FreeBSD installiert und versucht von dort aus mit *fsck* die UFS-Partition zu reinigen, geht das auch nicht. Das Mounten funktioniert theoretisch, aber dabei stürzt FreeBSD ab und der PC ist anschließend nur noch über den Druck auf den Resetknopf wieder zu beleben. Beim nächsten Booten sieht man dann natürlich die Meldung:

```
WARNING: / was not properly dismounted
```

Handelte es sich bei der zuvor in OpenBSD eingebundene FreeBSD-Partition um eine beliebige, ist, falls die Partition in der */etc/fstab* steht, FreeBSD zwar im Single-User-Modus startbar. Das nutzt aber nichts, denn das Programm *fsck* von FreeBSD stürzt bei Versuchen, die Partition zu reinigen, mit einem Trap ab (Bild 2). Das Problem läßt sich weder mit OpenBSD noch mit FreeBSD lösen und ist jederzeit reproduzierbar: Für OpenBSD ist die Partition in Ordnung und FreeBSD kann sie nicht mehr einbinden. Die Lösung bringt erst das Programm *fsck* von NetBSD. Es stellt den Fehler auf der FreeBSD-Partition fest und behebt ihn (Bild 3). Dazu muß man natürlich aber ein NetBSD-System verfügbar haben. Nachdem in der Redaktion freeX schon mehrere Live-CDs zu NetBSD zusammengestellt wurden, waren das geschilderte Problem und auch die anstehende Version 5 des Betriebssystems mit der vollständigen FUSE-Unterstützung Anreiz genug, wieder einmal eine Live-CD zusam-

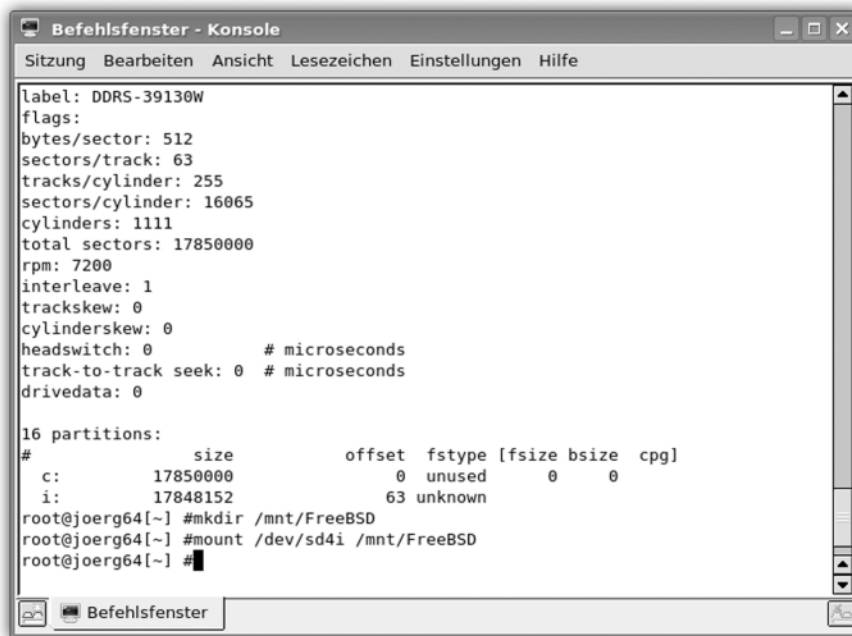


Bild 1: Der Zugriff von OpenBSD ist auch auf die »unknown« markierte FreeBSD-Partition problemlos möglich

menzustellen. Es sollte dieses Mal eine Rettungs-CD für Dateisysteme sein. Mit eingebundenem PUFFS/FUSE sind insgesamt Zugriffe auf FreeBSD-, NetBSD-, Windows- (NTFS, FAT/VFAT) und Linux-Ext2/Ext3-Datenträger möglich, eingeschränkt auch auf XFS und ReiserFS. Es ist nicht nur möglich, die entsprechend formatierten Datenträger einzubinden und vorhandene Daten beispielsweise über das Netzwerk zu einem sicheren Platz zu übertragen, sondern sie im Notfall auch zu reinigen.

Kernel und Bootsystem

Aufgebaut wurde die CD-Distribution auf der Basis des Release Candidates 1 von NetBSD 5.0 für die 32-Bit-Plattform i386. Sie besteht aus einem leicht modifizierten Kernel, den üblichen Programmen und drei zusätzlichen RC-Skripten für das Startsystem. Außerdem wurden einige Verzeichnisse umgehoben und der größte Teil des Systems befindet sich in einer komprimierten Datei. Das brutto über 550 MByte große System belegt dadurch als ISO-Datei nur etwa 220 MByte. Das Booten von NetBSD verläuft in drei Schritten. Der erste besteht im Laden des Urladers. Für das Booten

von CD kann das entweder das NetBSD-eigene Bootsystem übernehmen, mit dem auch von Festplatte gestartet wird und dessen Binärdatei unter `/usr/mdec` zu finden ist, oder der Grub. Letzterer bietet sich speziell für Multiboot-Umgebungen an. Bei der vorliegenden CD fiel die Entscheidung auf das Original von NetBSD. Seine Version für CDs mit dem Namen `bootxx_cd9660` ist genau 2048 Byte groß und muß `mkisofs` beim Ma-

stern des Images als Bootdatei angegeben werden. Der Urlader lädt die Datei `/boot`, die bei CD- und Festplatteninstallationen identisch ist und über die Konfigurationsdatei `/boot.cfg` den Kernel mit den entsprechenden Startoptionen lädt.

Dieser Kernel heißt normalerweise `netbsd`. Die Datei kann aber mit `gzip` gepackt werden. Dann muß dem Bootloader der Name `netbsd.gz` angegeben werden. Hier die `boot.cfg` der CD-ROM:

```

menu=Normal booten:boot netbsd.gz
menu=Single-User-Modus:boot netbsd.gz -s
menu=ACPI abschalten:boot netbsd.gz -2
menu=ACPI und SMP abschalten:boot
netbsd.gz -12
menu=Normal booten (Cardbus-Kernel):boot
cardbus.gz
menu=Single-User-Modus (Cardbus-
Kernel):boot cardbus.gz -s
menu=ACPI abschalten (Cardbus-Kernel):boot
cardbus.gz -2
menu=ACPI und SMP abschalten (Cardbus-
Kernel):boot netbsd.gz -12
menu=Zum Boot-Prompt:prompt
default=1
timeout=5
    
```

Der gepackte Kernel bootet von CD natürlich schneller als der ungepackte, weil bei dieser Art von Datenträgern das Laden von Daten von der CD zeitaufwendiger ist als das Entpacken. Ist der Kernel erfolgreich in den Speicher geladen, lädt er vom

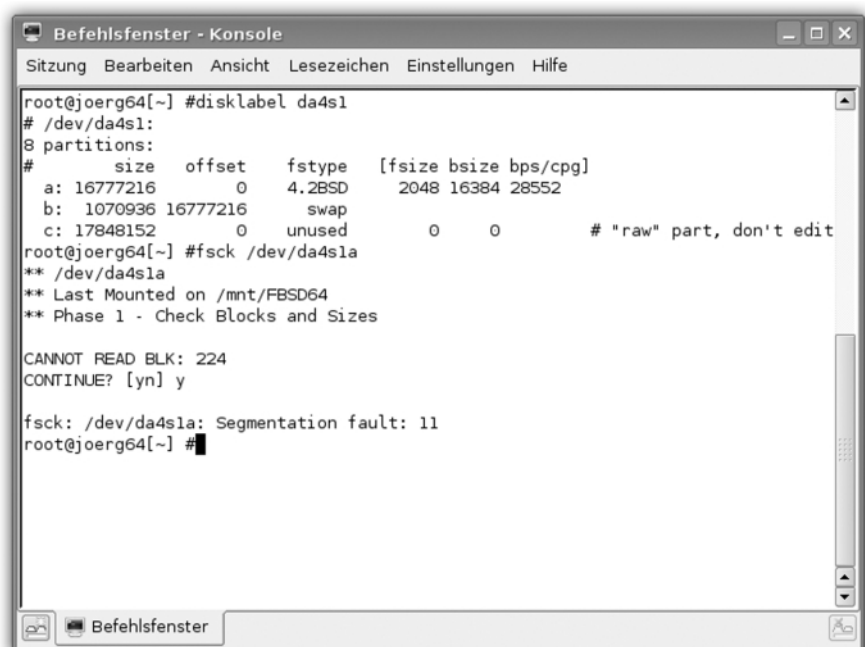


Bild 2: Jetzt streikt FreeBSD, es ist kein Zugriff auf die Partition mehr möglich

Bootmedium das Programm `/sbin/init`. Dazu muß er wissen, von welchem Datenträger gestartet wird. Normalerweise gilt bei NetBSD in der Kernelkonfiguration dafür die Kennung

```
config netbsd root on ? type ?
```

(siehe `/usr/src/sys/arch/i386/conf/GENERIC`). Es wird also versucht, das Bootmedium und auch das Dateisystem selbst zu erkennen. Dieser Kernel ist aber nicht in der Lage, von einer CD zu starten. Hier gilt die Konfigurationszeile

```
config netbsd root on cd0d type cd9660 \
    dumps on none
```

Wichtig ist außer der Angabe des eigentlichen Startmediums und dessen Dateisystems auch die Angabe, daß keine Dumps erlaubt sind. Fehlt sie, kann der Kernel nicht hochfahren, da versucht wird, auf die nicht vorhandene Partition `cd0b` zu dumpen.

Dies ist im übrigen die einzige wirkliche Anpassung, die bei einem NetBSD-Kernel, der von CD booten soll, notwendig ist, alle weiteren Änderungen sind optional. Da das FUSE/PUFFS-System gewünscht ist, dürfen die Zeilen

```
file-system PUFFS
pseudo-device putter
```

nicht fehlen. Um mitzuteilen, daß es sich um einen speziellen Kernel handelt, wurde die Farbe der Kernel-Meldungen vom regulären Grün auf rote Farbe umgeändert.

Regulär erwartet NetBSD, daß auf den Boot-Datenträger und zwar mindestens in die Verzeichnisse `/etc`, `/tmp`, und `/var` geschrieben werden kann. Bei einer CD geht das erst einmal nicht. Das ausgefeilte RC-System, das sich in `/etc/rc.d` befindet, basiert auf einer ganzen Reihe Skripte, die untereinander zusammen- und voneinander abhängen.

Für die CD-ROM-Distribution sollte dieses System möglichst wenig geändert werden. Tatsächlich mußten auch bei den vorhandenen Da-

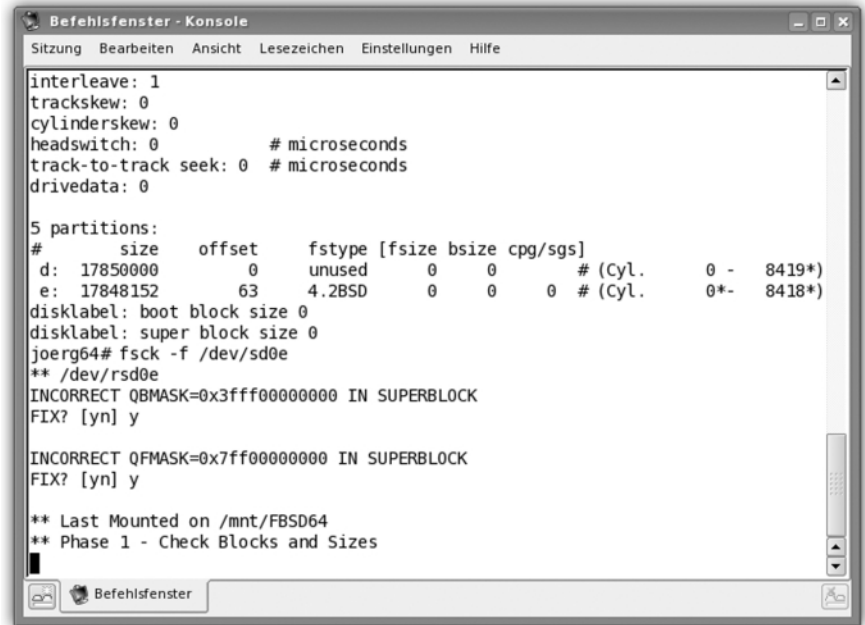


Bild 3: Rettung mit NetBSD

teien nur die Befehle in der Datei `root` auskommentiert werden. Es wurden aber drei Dateien ergänzt. Sie wurden der Version 2007 von NetBSD Live! entnommen und angepaßt beziehungsweise verbessert. Die erste Datei heißt `mfsrtd`; in ihr wird das komprimierte Verzeichnis `/usr` eingebunden, außerdem werden RAM-Laufwerke für `/etc`, `/var`, `/root` und `/home` angelegt. Damit Schreibvorgänge nach `/usr` möglich sind, wird ein Verzeichnis `/var/usr` nach `/usr` per Union übergemountet.

Schreiben auf CD

Ausschnittsweise sieht das Mounten so aus:

```
mount_mfs -s 20m swap /etc
mount_mfs -o noatime -s 25m swap var
/rescue/vnconfig -z vnd0 /stand/LIVE.imz
mount_cd9660 /dev/vnd0d /usr
mkdir /var/usr
mount_union /var/usr /usr
/rescue/tar -xzf /stand/MAIN.tbz -C /
```

Es stellte sich hierbei heraus, daß das neue und als effizient gepriesene `tmpfs` mit Union-Mounts nicht funktioniert, während das ältere und genauso stabile `mfs` damit sehr gut zurechtkommt. Wie die Listing-Zeilen zeigen, befinden sich die Zusatzdateien mit dem Inhalt der RAM-Disks und von `/usr` im Verzeichnis

`/stand`. Dieses Verzeichnis ist der Dreh- und Angelpunkt der Distribution. Hierhin wurden für das Entlasten der RAM-Disks auch die RC-Dateien ausgelagert (`/etc/rc.d` ist deshalb nur ein Link auf `/stand/rc.d`), hier befinden sich auch die Kernel-Konfigurationsdateien und drei Hilfskripte, mit denen die verschiedenen Dateien geschrieben werden. Die Datei wird im RC-System vor `disks` aufgerufen, aus diesem Grund mußte das Skript `root` geändert werden. Die zweite Datei im modifizierten System trägt den Namen `mfsnet`, dieser Teil war bei NetBSD Live! 2007 auch im ersten Skript enthalten und wurde deshalb entfernt, weil es Ärger mit dem Programm `dialog` gab, das eine Bibliothek in `/usr/pkg/lib` und eine funktionierende Konsole benötigt. `mfsnet` besitzt jetzt als Vorbedingung `tty` und wird vor `network` geladen. Hier findet aktuell nur die Konfiguration des Netzwerks statt. Diese ist allerdings leider etwas wackelig, wenn kein DHCP-Server zur Verfügung steht, und es traten Probleme bei PCs mit mehreren Netzwerkkarten auf, da bei einem versehentlichen falschen Tastendruck im Dialog dieser weiterspringt. Außerdem gilt bei fixen IP-Adressen natürlich, daß man Kollisionen vermeiden muß. Die CD enthält außer den Optionen für das Einrichten des

Netzwerks keine weiteren Standard-dialoge. Es wird als Sprache und Länderkennung Deutsch/Deutschland vorausgesetzt, als Zeitzone die CET (Europe/Berlin) bei einer BIOS-Uhr, die auf lokale Zeit gestellt ist. Außerdem wird eine deutsche Tastaturbelegung geladen.

Bei einem Rettungssystem sollten die Angaben entweder passen oder auch ignoriert werden können. Wenn Bedarf für andere Sprachversionen besteht und angemeldet wird, können sie jedoch leicht und schmerzlos nachgerüstet werden. Es sollte aber die Qual vermieden werden, sich jedesmal beim Starten durch eine Reihe immer wieder gleicher Dialoge klicken zu müssen. Das dritte Skript soll hier auch noch kurz erwähnt werden. Es heißt *mkiparts* und versucht, auf den Laufwerken wd0 bis wd3 und sd0 bis sd5 brauchbare Partitionen zu finden und ein-

```

root@netbsd-live-cdrom [/etc] #df
Filesystem      Size      Used      Avail %Cap Mounted on
root_device     218M      218M      0B 100% /
mfs:/8          4,6M      75K       4,3M  1% /dev
mfs:/51         19M       6,8M      12M  37% /etc
mfs:/53         24M       3,1M      20M  13% /var
mfs:/55         23M       20K       22M  0% /root
mfs:/62         23M       1,0K      22M  0% /home
/dev/vnd0d      573M      573M      0B 100% /usr
<above>:/var/usr 96M       75M      20M  79% /usr
<below>:/stand/pkg 51M       30M      20M  60% /var/db/pkg
kernfs          1,0K      1,0K      0B 100% /kern
ptyfs           1,0K      1,0K      0B 100% /dev/pts
procfs          4,0K      4,0K      0B 100% /proc
/dev/puffs      0B        0B        0B 100% /sys
fdesc           1,0K      1,0K      0B 100% /dev
<below>:/stand/rc.d 46M       34M      12M  74% /etc/rc.d
root@netbsd-live-cdrom [/etc] #

```

Bild 4: Mit Union-Mounts wurde ein ausgeklügeltes System aufgebaut, in das auch Anwendungen installiert werden können

zubinden. Dabei wird nach FFS/UFS, LFS, Ext2/Ext3, FAT/VAT und NTFS gesucht. Für jede gefundene Partition wird dann ein Eintrag in die */etc/fstab* geschrieben und gleichzeitig ein Mountpunkt nach */var/mnt* (das auch nach */mnt* gelinkt wird) angelegt. Die Partitionen werden aber

nicht automatisch eingebunden. Wer sie mounten will, muß das jeweils mit dem *mount*-Befehl tun. Davor sollte aber ein Blick in die */etc/fstab* geworfen werden und der Eintrag gegebenenfalls auf die Option *ro* angepaßt werden. Ein Eintrag könnte beispielsweise so aussehen:

```

/dev/wd0a /var/mnt/wd0a ffs rw,noauto \
0 0

```

Die CD bootet durch das RC-System und landet dann am Login-Prompt. Hier gibt es nur einen User *root*, der kein Paßwort besitzt. Die CD-ROM gehört also keineswegs als Standard ins Laufwerk! Als Shell ist die *bash* vorkonfiguriert, was sicher wieder den einen oder anderen NetBSD-Puristen empört, sie ist aber besser als die bei NetBSD vorinstallierte Korn-Shell.

Außer den Reparatursets für die diversen Dateisysteme ist im Live-System nicht viel zusätzliche Software installiert, enthalten sind der Midnight Commander und der von FreeBSD bekannte kleine Editor *easyedit* (bei FreeBSD heißt er *ee*) sowie der *joe*. Bis auf die Include-Dateien handelt es sich aber das komplette NetBSD 5.0-RC1 mit allen dort üblichen Programmen. Auch X ist installiert, aber ohne Zusatzprogramme. Das bedeutet, daß als Window Manager nur der *twm* verfügbar ist. Eine X-Konfiguration befindet sich in */etc/X11*, wobei sie nur den VESA-Treiber in 1024x768 lädt. Wer mehr und besseres möchte, kann sich mit *X-configure*

Manuell ins Netz mit NetBSD

Am einfachsten gelangt man bei NetBSD per DHCP ins Netz. Das Skript *dhclient* macht alles von selbst: Es holt vom Server die IP-Adresse und modifiziert die */etc/resolve.conf*. Damit es in dem geänderten RC-System gestartet wird, muß es allerdings bei der CD-Distribution über die Datei */etc/rc.local* geladen werden. Ein kleiner Schönheitsfehler, der aber dafür sorgt, daß beim Starten manuelle und alternativ automatische Adreßzuweisung funktionieren.

Wenn das Netzwerk manuell eingerichtet werden soll, sind einige Arbeiten mehr erforderlich. Der Name des PCs gehört in die */etc/myname* (das gilt auch bei DHCP), bei einem Namen »meinpc« steht nur das *meinpc* in dieser Datei. Die IP-Adresse des Default-Gateway muß in der */etc/mygate* stehen. Der oder die Nameserver müssen dann in der */etc/resolv.conf* eingetragen werden. Hier stehen dann (hier für Wanadoo) beispielsweise die Einträge

```

nameserver 80.10.246.3
nameserver 81.253.149.2

```

Es kann aber (abhängig vom Fabrikat eines DSL-Routers) auch dessen IP-Adresse in diese Datei gehören.

Zuletzt wird dann bei der manuellen Adreßzuweisung noch eine Datei für *ifconfig* benötigt. Sie besitzt die Endung mit dem Namen der Netzwerkkarte; wie sie heißen muß, verrät der Aufruf *ifconfig -a*. Für die Karte mit der Geräteerkennung *re0* heißt sie *ifconfig.re0* und sie enthält die Zeile mit der Netzwerkkennung, beispielsweise den Eintrag

```

inet 192.168.0.8 netmask 255.255.255.0

```

Die Angaben *inet* und *netmask* sind Pflicht.

Wem das jetzt zu schnell ging, sollte die Manpages zu *ifconfig* und *ifconfig.if* studieren.

Neues Package für dialog

Die CD-ROM nutzt das bekannte Programm *dialog* für die Abfragen beim Start. Die bei NetBSD in den Packages befindliche Version dieses Programms beherrscht über das Gettext-System auch Sprachanpassungen, unverständlicherweise die Version in der Make-Datei in den Package Sources aber nicht. Dem kann aber ganz einfach abgeholfen werden. Das Archiv *dialog.tgz* im Verzeichnis zu diesem Beitrag ergänzt das Verzeichnis um eine erweiterte Make-Datei und eine neue Installationsbeschreibung. Für NetBSD 5 ist auch gleich das Paket im Verzeichnis enthalten. Die Datei *dialog.tgz* muß nach */usr/pkgsrc* entpackt werden, das geht mit `tar -xzf <Pfad>/dialog.tgz -C /usr/pkgsrc`.

und den entsprechenden Nacharbeiten auch etwas anderes erzeugen. Allerdings ist für Tests in VMware und auch für die Reparatur von VMware-Images ein VMware-X in */etc/X11* enthalten, das genauso auch bei regulären Installationen von NetBSD 5 unter VMware eingesetzt werden kann. Durch den Wechsel zu X.org sind jetzt nämlich die Treiber *vmmouse* für die Maus und *vmware* für die Grafik verfügbar. Das Herausklicken aus der Maschine mit [Strg][Alt] ist bei laufendem X-Server dann nicht mehr nötig.

Eine Reparatur-CD macht nur Sinn, wenn sie auf möglichst vielen PCs funktioniert. Deshalb wurde sie nicht mit den DVD-Daten vermischt. Das ISO-Image ist wegen seiner geringen Größe rasch auf eine CD-R gebrannt und kann auch auf Computern ohne DVD-Laufwerke, von denen es noch genügend geben soll, genutzt werden.

Durch drei abgestimmte Union-Mounts ist es auch möglich, im laufenden Betrieb noch notwendige Anwendungen nachzuinstallieren, dafür müssen diese auf dem NetBSD-Server vorkompiliert zur Verfügung stehen. Die Union-Mounts betreffen die Verzeichnisse */var/db/pkg* mit der Paketdatenbank, den */usr*-Baum, in den Programme installiert werden, und */etc/rc.d*, das nach dem Booten im allerletzten Schritt umgemountet wird. Weiterhin ist die Variable *PKG_PATH* gesetzt und zwar auf den schnellen FTP-Spiegelservers in Frankreich. Bild 4 zeigt das gemountete System direkt nach dem Booten. Der letzte Schritt, nämlich

das Umheben von */etc/rc.d* vom Link auf den Union-Mount, erfolgt erst beim ersten Einloggen von *root*, da hier das RC-System abgeschlossen ist und nicht mehr gestört wird. Die entsprechenden Zeilen in der Datei sehen so aus:

```
if [ "$whoami" = "root" ] ; then
if [ -h /etc/rc.d ] ; then
# Link aus der Boot-Zeit entfernen:
rm /etc/rc.d
# Verzeichnis auf dem RAM-Laufwerk
# anlegen:
mkdir /etc/rc.d
# /stand/rc.d per Union unter das
# neue Verzeichnis legen
# (andersherum geht nicht):
/sbin/mount_union -b /stand/rc.d \
/etc/rc.d

# User zuweisen:
/usr/sbin/chown root:wheel /etc/rc.d
# Zum Schluß noch die Rechte-
# Justierung für /root
/usr/sbin/chown root:wheel /root
chmod 700 /root
fi
fi
```

Diese Maßnahme funktioniert natürlich nur, wenn kein anderer User-Account als *root* gestartet wird, was auf dieser CD dadurch gewährleistet ist, daß kein anderer angelegt ist. Würde sie fehlen, wäre das aber auch nicht so schlimm. Das RC-System wird nur dazu gebraucht, Daemons zu starten, neu zu starten und wieder herunterzufahren, was nur *root* kann, und mit dem mitgelieferten Dateien ist es auch durch allein durch die Verlinkung möglich. Interessant ist die Beschreibbarkeit des Verzeichnisses eigentlich nur, wenn entweder ein Startskript angepaßt werden muß (was eher unwahrscheinlich sein wird) oder wenn nach

seiner Installation ein Daemon über ein Skript an der richtigen Stelle und nicht direkt durch seinen Aufruf aufgerufen wird.

Viel Platz ist – wie Bild 4 auch zeigt – aber auf den RAM-Laufwerken nicht. */var* wird durch Protokolle nicht überlaufen, dafür ist dadurch gesorgt, indem in der *syslog.conf* außer Notfallmeldungen alles ausgeschaltet wurde. In */usr* ist aber nur sehr begrenzt Platz, für einen zusätzlichen Editor oder ein dringend erforderliches Schutzprogramm sollten die freien 20 MByte hoffentlich aber reichen.

Eigener Weiterbau

Für Anregungen und Wünschen, was die Distribution anbelangt, ist die Redaktion offen, eine größere Version auf DVD mit einem moderneren Window Manager und Anwendungen für X ist in Planung. Auch mehrsprachige Varianten wären möglich. Die Grundarbeiten sind getan.

Sie können das aber auch selbst tun. Sie benötigen dazu eine NetBSD-5-Installation. Legen Sie hier die Verzeichnisse */usr/pub/live/tree* und */usr/pub/live/comptree* an. Kopieren Sie dann den Inhalt der CD nach *comptree* so, daß keine Dateattribute verlorengehen, am einfachsten im Midnight Commander. Mounten Sie den Knoten der Datei *stand/LIVE.imz* mit *vnconfig* und *mount_cd9660* wie weiter oben schon gezeigt und kopieren Sie die enthaltenen Dateien nach *usr/* des Baums *tree*. Entpacken Sie außerdem die Datei *stand/MAIN.tbz* nach *live*:

```
# tar -xzf <Mountpunkt>/stand/MAIN.tbz \
-C /usr/pub/live/tree
```

Übertragen Sie dann die anderen Dateien so, daß der Baum zweimal vorliegt, in *tree* komplett mit allem und in *comptree* mit einem leeren Verzeichnis *usr/*. Zum Schluß wird das neue CD-Image nach */usr/pub/live/iso* geschrieben. Dieses Verzeichnis muß also auch noch angelegt werden. Im nächsten Schritt müssen Sie in einer *chroot*-Umgebung die ge-

wünschten Anwendungen installieren. Wenn die Pakete lokal vorliegen mit

```
# mount -t union /usr/pkgsrc \
    /usr/pub/live/tree/usr/pkgsrc
# chroot /usr/pub/live/tree
# cd /usr/pkgsrc/packages/All
# pkg_add <Paketname>
```

Oder Sie installieren die Programme aus dem Internet und definieren dafür die Variable `PKG_PATH`, beispielsweise so:

```
export PKG_PATH=ftp://fr.netbsd.org/\
    pkgsrc/packages/NetBSD/i386/5.0/All
```

Vermeiden Sie es, den Hauptserver in Amerika anzugeben, erstens ist hier die Datenübertragung deutlich langsamer als von den europäischen Spiegelservers, zweitens ist es kein guter Stil, vom Primärserver zu laden.

Die Basis eigener Versionen

In der `chroot`-Umgebung kann gebastelt und gearbeitet werden. Zum Schluß müssen dann aber drei Dateien angelegt werden. Dafür gibt es in `stand/scripts` die Skripte `1mkarc`, `2mkimage` und `3mkcd`. Das erste Skript erzeugt die Datei `stand/MAIN.tbz`, die beim Booten nach `/` entpackt wird, das zweite legt das Image für das Verzeichnis `usr/` mit den installierten Anwendungen an, und das letzte das endgültige ISO-Image für die CD oder DVD, das auch die beiden anderen Dateien enthält.

Spätestens vor dem Aufruf von `2mkimage` müssen die `CDRT`ools installiert sein, weil das Image mit `mkisofs` geschrieben und nicht mit dem Systemprogramm `makefs` angelegt wird, das sich als wenig praktikabel erwiesen hat. Weitere externe Anwendungen werden nicht benötigt, in der Umgebung muß aber `dialog` installiert sein, da hiermit der beziehungsweise die Startdialoge angezeigt werden.

Die Lizenz von NetBSD Live! schließt die Weitergabe des Originals durch andere Verlage grundsätzlich aus,

bei anderer kommerzieller Distribution will der Verlag vorher gefragt werden. Ansonsten ist die nicht-kommerzielle Weitergabe erlaubt und auch gewünscht, die Mitglieder der NetBSD-Community können sie auch

gegen Spende oder kostenlos nach eigenem Gusto auf Messen verteilen, das soll hier ausdrücklich festgehalten werden. Was Sie mit einer selbstgebauten Variante tun, ist natürlich Ihre Sache. ◆

Von der CD zurück zum Master

Um aus den CD-Daten wieder die drei benötigten Bäume anzulegen, muß folgendermaßen vorgegangen werden:

Zuerst wird ein Verzeichnis `/usr/pub/live` angelegt. Dabei ist auf ausreichend Platz zu achten, ein GByte muß schon frei sein. In diesem Grundverzeichnis werden die weiteren Unterverzeichnisse `tree`, `comptree` und `iso` angelegt.

Im nächsten Schritt wird die CD gemountet und ihr Inhalt Eins-zu-eins in das Verzeichnis `/usr/pub/live/comptree` übertragen, ebenso in das Verzeichnis `/usr/pub/live/tree`. Die beiden Verzeichnisse sind damit erst einmal identisch. Dabei darf die CD aber nicht aktiv sein, die Arbeit muß in einem regulären NetBSD 5 stattfinden.

Im zweiten Schritt wird der komplette Inhalt des Archivs `stand/MAIN.tbz` in den Baum `tree` übertragen. Dazu wird auf der gemounteten Live-CD in das Verzeichnis `stand` gewechselt und hier

```
# tar -xvpzf MAIN.tbz -C /usr/pub/live/tree/
```

aufgerufen. Der Parameter `p` ist wichtig. Damit bleiben die Rechte und Attribute erhalten.

Im dritten Schritt wird der Inhalt des komprimierten ISO-Images `LIVE.imz` nach `/usr/pub/live/tree/usr` entpackt, wozu das Image zuerst gemountet werden muß:

```
# mkdir /mnt/iso
# vnconfig -z vnd0 /cdrom/stand/LIVE.imz
# mount_cd9660 /dev/vnd0d /mnt/iso
# cp -pR /mnt/iso /usr/pub/live/usr
```

Besser ist es aber, anstelle mit dem `cp`-Befehl die Daten mit dem Midnight Commander zu kopieren. Dabei bleiben auch alle Rechte und Links richtig erhalten.

Ab diesem Moment sind die Ursprungsdaten wiederhergestellt. Das ISO-Image kann nach dem Abschluß der Arbeiten dann wieder ausgebonden werden:

```
# umount /mnt/iso
# vmconfig -u /dev/vnd0
# rmdir /mnt/iso
```

Jetzt wird mit `chroot` in das Verzeichnis `/usr/pub/live/tree` gewechselt und dort mit dem Master gearbeitet. Werden Programme installiert, darf nicht vergessen werden, die Paketdatenbank vor dem Erzeugen des ISO-Images in aus dem `tree`-Pfad (im Unterverzeichnis `stand/pkg`) nach `/usr/pub/live/comptree/stand/pkg` zu spiegeln. Das dient aber nur der Bequemlichkeit. Die Paketdatenbank wird für das ordnungsgemäße Arbeiten mit der Live-CD nicht benötigt. Konfigurationen, die von Programmen im Package-Source-Tree nach `usr/pkgsrc/share` installiert werden, brauchen bei dieser Live-CD nicht mehr berücksichtigt zu werden, durch das Union-Dateisystem von `usr` hat sich das angenehmerweise erledigt. Die Dateien können bei der Arbeit mit dem Live-System normal beschrieben und gelöscht werden und sind nach dem nächsten Start wieder da.