

Vom noch einfacheren Umgang mit NEXSTEP

Booten vom Zip auf Intel-NEXSTEP-Rechnern

Bisher sah man ziemlich blaß aus, wollte man ein Komplettdump nach einem Plattencrash wieder von einem DAT-Streamer einspielen, wenn man nur einen Standalone NEXSTEP-Rechner zu Hause hatte, denn dazu braucht man zumindest ein lauffähiges NEXSTEP-System. Dank Zip-Drive wird dazu nun nicht mehr länger eine Festplatte benötigt, da ein ganzes Minimal-System auf so ein kleines Medium paßt (Zip-Drives sind portable Wechsellplattenlaufwerke, die es in einer Parallelport- und in einer SCSI-Variante gibt. Die Medien sind kaum größer als die 3,5"-Disketten, besitzen aber 100 MB Speicherkapazität bei einer Zugriffszeit von 29 ms bei der SCSI-Variante, die problemlos unter NEXSTEP läuft – plug & play vom Feinsten!). Selbst ein Multiuserboot ist möglich – wichtig, da Intel-NEXSTEP-Rechner den SCSI-Tape-Treiber erst im Multiuserboot laden. Mit mittlerweile nur noch ungefähr 20 DM pro Medium ist dies auch die erheblich preiswertere Alternative zu einer Not-Festplatte. Anleitungen, wie so ein Boot-Medium erstellt wird, sucht man (fast) vergebens. Auf den Seiten von *Radical System Solutions* (<http://www.radical.com/TheHome/TheSolutions/RadicalSolution4.html>) befindet sich ein Versuch, diesen Mißstand zu beseitigen. Leider gilt die Anleitung nur für schwarze Hardware [„This has not been tried on Intel hardware (or the other platforms).“] und ist für Intel-NEXSTEP-Rechner unbrauchbar, da die unten beschriebenen Modifikationen fehlen. Laut 3.3 Release Notes ist **BuildDisk.app** nicht für Intel-NEXSTEP-Rechner geeignet: „Auf einem Intel-gesteuerten Computer können Sie BuildDisk nicht verwenden.“ Wie dennoch für diese Plattform ein bootfähiges Zip-Medium mit **BuildDisk.app** erstellt werden kann, möchte ich im Folgenden verraten:



Auswählen der zu installierenden Erweiterungspakete in **BuildDisk.app**

Zip-Disk einlegen und **BuildDisk.app** aufrufen. Wenn es nicht als **root** aufgerufen wird, fragt **BuildDisk.app** nach dem Paßwort, bevor die Zip-Disk initialisiert wird. Auf alle Fälle das Panel vergrößern, damit die Zusatzpakete wie z.B. Deutsch und die neueren Treiber für die seriellen Schnittstellen zu sehen sind und ausgewählt werden können. Als Tipp würde ich noch den 3.3-Patch mit auswählen, sofern man ihn auf der Festplatte installiert hat. Nicht wundern, daß **BuildDisk.app** erstmal das Zip wieder auswirft und sich dann über ein fehlendes Medium beschwert – einfach wieder in das Laufwerk schieben und **BuildDisk.app** arbeiten lassen – das dauert so an die 30–45 Minuten, je nachdem, wie viele Zusatzpakete man installieren will.

Eigentlich fehlt (fast) nur noch *ein* Link, der aber äußerst wichtig ist, da sonst das System überhaupt nicht „hochkommt“. NEXSTEP greift nicht direkt auf die Treiber zu, sondern handhabt dies über mehrere Links. Wichtig

ist es, diesen nicht im Workspace zu erstellen, weil dieser den Link auf den vollständigen Pfad setzt, wir aber nur einen Verweis auf gleicher Ebene brauchen. Folgendes muß daher als **root** in einer Shell im Pfad **/Zip-StartDisk/private** eingegeben werden (Zip-StartDisk steht hier für den Namen des ZIP-Mediums):

```
cd Zip-StartDisk/private
ln -s Drivers/i386 Devices
```

Dann sollte der Link `/usr/Devices` auf dem Zip-Medium nicht mehr ins Nirvana sondern eben tatsächlich nach `/private/Drivers/i386` zeigen und somit ein Hochfahren ermöglichen. Eigentlich zeigt der Link `/usr/Devices` auf `/private/Devices`, was aber wiederum ein Link auf `Drivers/i386` ist. Ferner kopiert **BuildDisk.app** einige Einträge und Treiber nicht mit aufs Zip, was von Hand als **root** nachgetragen werden muß: Der gesamte Inhalt von **System.config** ebenso wie die **Instance0.table** von **SCSITape.config** sowie die von Treibern, die beim Initialisieren ausgewählt wurden (bei mir die **Instance0.table** von **PortServer.config**, **ISASerialPort.config** und **SerialPointingDevice.config**), wie auch die kompletten systemrelevanten Treiber, die in der **Instance0.table** von **System.config** mit aufgeführt sind (bei mir der **SYM53c8.config** für meinen SCSI-Controller und der **miro20SV.config** für meine Grafikkarte). Als letztes muß noch in der `/etc/fstab` auf dem Zip-Medium das Zip als root-device eingetragen werden, da die Option `-a` (Fragen nach Rootdevice) beim Hochfahren leider nicht funktioniert und der Rechner zumindest im Multiuserboot irgendwo hängenbleibt, weil er das Zip sonst wie eine CD-ROM nur read-only mountet.

```
/dev/sdx / 4.3 rw,noquota,noauto 0 1
```

Das *x* in `sdx` steht hierbei für die interne SCSI-Gerätenummer des Zip-Drives und nicht etwa für dessen ID! Auf Grund der Read-Only-Problematik ist es wahrscheinlich auch nicht möglich, daß das Zip eine größere ID als das CD-ROM bekommt, da der Rechner sofort beim Bootvorgang aussteigt. Bei mir hat das Zip ID 5 (es kann ja leider nur 5 oder 6) und das CD-ROM ID 6.

Wer damit fertig ist, sollte mal testen, ob alles klappt. Am Bootprompt muß dafür `sd(x,0,0)mach_kernel` eingegeben werden, wobei *x* wieder für die interne SCSI-Gerätenummer steht. Eigentlich sollte alles klappen und das Sprachpanel erscheinen, wo die Sprache und Tastatur ausgewählt werden sollen und man anschließend als **me** eingeloggt wird. Günstig ist es noch, **me** ein Passwort zu verpassen, damit man sich im Notfall als **root** einloggen kann. Wenn das alles geklappt hat, ist das Notsystem-Zip-StartDisk-Medium fertig und wird hoffentlich nie gebraucht! ;-)

Tip: Für mich war es sehr hilfreich, das Plattenabbild auf dem Zip immer mit dem Original auf der Bootplatte zu vergleichen.

Anke Ehlers (anke@anarcoma.in-berlin.de)