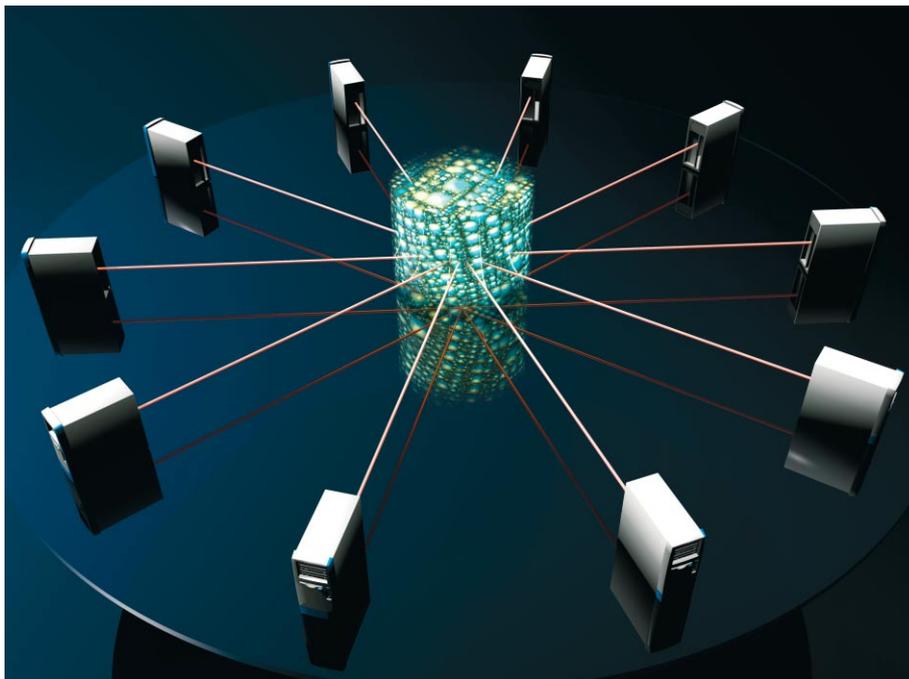


## ZAUBERHAFT GEBOOTET

# Netz-Booter

Wer häufig mit verschiedenen Rechnern hantiert, ab und zu mit Knoppix-CDs Daten retten muss oder einfach eine elegante Boot-Methode für plattenlose Clients sucht, sollte sich mit dem PXE-Boot via Ethernet vertraut machen, das jeder moderne Rechner beherrscht. Wir zeigen, wie der Bootloader *PXELINUX* künftig die Jongliererei mit CDs erspart. VON **MATTIAS SCHLENKER**



**B**ereits in den späten 80ern (des letzten Jahrhunderts) waren festplattenlose Terminals en vogue: Von einem Novell- oder Unix-Server mit einem simplen DOS versorgt, dienten einfache PCs als Terminals an Kassen oder in Bibliotheken. Der größte Vorteil der zentralen Wartbarkeit wurde damals jedoch mit aufwändig herzustellenden Boot-EPROMs erkauft, die für jeden Rechner individuell zugeschnitten sein mussten. Das Verfahren geriet in den 90ern wieder etwas ins Hintertreffen, weil Windows eine lokale Festplatte erforderte. Mit dem Trend zum Thin-Client-Computing erlebte das Booten aus dem Netzwerk aber wieder eine Renaissance. Heutzutage haben praktisch alle Onboard-Netzwerk-Karten PXE-Funktionalität, welche das Brennen eines EPROMs erspart und den-

noch den Start aus dem Netz erlaubt. Das Geheimnis: Bereits die PXE-Komponente des BIOS fordert via DHCP ihre Booteinstellungen an. So genügt auf Clientseite die Aktivierung von PXE in der Bootreihenfolge des BIOS. Dennoch ist der Start per PXE bislang nur in großen Rechenzentren (z.B. für die Rettungssysteme von Rootservern) und Firmennetzen üblich, in denen der Admin auch aus der Ferne Betriebssysteme installieren möchte. Dabei bietet PXE auch im Heimnetz oder in kleinen Firmen ernsthafte Vorteile für Wartungs- und Installationsarbeiten.

## Konfiguriert per DHCP

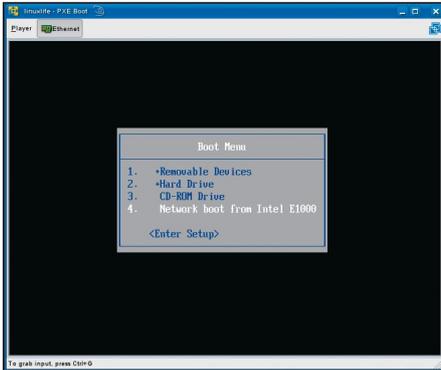
Da bei PXE keine Möglichkeit mehr existiert, die eigene IP-Adresse und die Adressen und Masken von Bootserver, Nameserver und

Gateway fix einzustellen, sondern diese Daten immer per DHCP angefragt werden, muss ein DHCP-Server im lokalen Netz vorhanden sein. Die „kleinen“ DHCP-Daemonen der meisten DSL-Router sind für diese Aufgabe nicht geeignet, da Sie keine Vergabe von Bootdateien zulassen. Schalten Sie den DHCP-Server im DSL-Router deshalb zunächst ab. Sie können später ausprobieren, ob bei gleicher MAC-IP-Adresstabelle Ihre Clients korrekt eine zweite Anfrage stellen, wenn sie die unvollständige Antwort des Routers erhalten haben. Ist dies der Fall, können Sie beide DHCP-Server parallel im Betrieb haben. Quasi-Standard ist der hochgradig konfigurierbare Server des Internet Software Consortiums *ISC-DHCP* in der aktuellen Version 3.0. Unter Ubuntu ist er im Paket *dhcp3-server* vorhanden.

Nach der Installation muss die Konfigurationsdatei `/etc/dhcp3/dhcpd.conf` angelegt/modifiziert werden. Unser Beispielserver residiert auf dem Host `192.168.1.250`, Nameserver und Gateway ist eine unter `192.168.1.252` liegende FRITZ!Box. Den Adressbereich zwischen `192.168.1.180` und `192.168.1.199` haben wir für Clients vorgesehen, deren IP-Adresse egal ist. Damit wir unsere per SSH erreichbaren PCs immer schön unter der gleichen IP-Adresse wiederfinden, vergeben wir fixe Zuordnungen zwischen MAC-Adresse und IP-Adresse. Auf die Option `filename` für die Bootdatei gehen wir später näher ein:

## Drahtlos aus dem Netz booten

➤ Noch geht reines PXE nur mit drahtgebundenen Netzen. Der Grund ist, dass für den DHCP-Request eine Verbindung auf Link-Ebene vorhanden sein muss, der beim WLAN die Assoziierung mit dem Access-Point vorangeht. An einer PXE-kompatiblen Nachrüstung arbeitet das *Etherboot*-Projekt (Name der Software *Etherboot*, demnächst *gPXE*), dessen kompatible Bootsoftware vielseitiger als die in Standard-PXE-ROMs enthaltene ist. Eine einfache Alternative zum direkten Boot per WLAN sind WLAN-Brücken, die mehrere Ethernet-Segmente verbinden und die Authentifizierung und Assoziierung erledigen. In Unternehmen mit mobilen Arbeitsplätzen ist der mögliche Zugriff auf die Fernwartungsfunktionen des BIOS sinnvoll. Einfache Brücken für ein bis vier Rechner gibt es ab 60 Euro.



PXE als temporäres Bootdevice: Bei VMware per ESC, bei den anderen BIOSen per F8 oder F11.

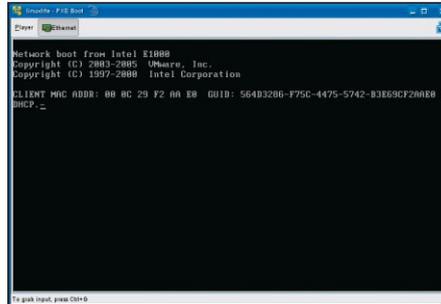
```
authoritative;
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;

filename „pxelinux.0“;

subnet 192.168.1.0 netmask
➔255.255.255.0 {
  option domain-name-servers
➔192.168.1.252;
  option routers 192.168.1.252;
  option domain-name „meinnetz.test“;
  range 192.168.1.180 192.168.1.199;
  option subnet-mask 255.255.255.0;
  next-server 192.168.1.250;
```

## VMware als Testumgebung

► Beim Testen einer PXE-Konfiguration ist es äußerst lästig, ständig zwischen zwei Rechnern hin und her zu rennen. Deutlich komfortabler geht es mit virtuellen PCs wie VMware. Letztere bringt ein gewöhnliches Phoenix-BIOS mit und implementiert den vollständigen PXE-Standard. Um einen echten PC auch auf Link-Ebene zu simulieren, muss der Netzwerkmodus allerdings auf *Bridged* eingestellt sein. Die VMware erscheint dann mit einer automatisch generierten MAC-Adresse an der physisch im Rechner vorhandenen Netzwerkkarte. Den Start per PXE initiieren Sie durch einen Tastendruck *F12* oder *Esc*, während das VMware-BIOS den Selbsttest durchführt. Die PXE-Boot-Funktion ist auch beim kostenlosen VMware-Player vorhanden. Für diesen haben wir eine Beispielkonfiguration auf DVD beigelegt, die auch ein maximal 16 GByte großes Festplatten-Image enthält, auf das Sie per PXE Betriebssysteme installieren können.



Das PXE-ROM sendet eine DHCP-Anfrage und bekommt daraufhin eine IP-Adresse zugewiesen.

```
host amd64-test {
  hardware ethernet
➔00:13:d4:fd:eb:49;
  fixed-address 192.168.1.50;
}
```

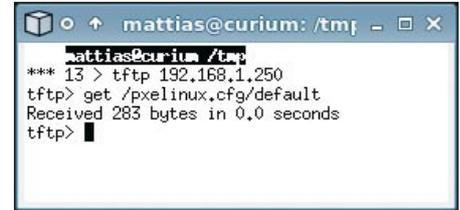
Wenn Sie jetzt den PC mit der angegebenen MAC-Adresse starten und im BIOS den Boot-Agenten einstellen (oder ihn temporär beispielsweise mit *F8* oder *F11* auswählen), wird dieser eine IP-Adresse und den Namen der Boot-Datei vom DHCP-Server anfordern und dies anzeigen. Er wird per TFTP versuchen, die Datei *pxelinux.0* anzufordern, aber irgendwann abbrechen und normal starten.

## Bootdateien per TFTP

Für die Auslieferung der Bootdateien kommt das *Trivial File Transfer Protocol* zum Einsatz. Es hat nicht viel mit seinem großen Bruder *FTP* gemeinsam - außer dass es zur Übertragung von Dateien dient. Im Gegensatz zu *FTP* nutzt *TFTP* das verbindungslose *UDP* statt *TCP*. Was gegenüber der vollständigen *NFS*-Implementierung in der Firmware früher *SPARC*-Systeme als Rückschritt erscheint, hat *PXE* erst möglich gemacht: Der Code für *TFTP* ist kompakt, was dem Kostenargument entgegenkam. Auf dem Testsystem installierten wir den *Advanced TFTP Daemon atftpd*. Da wir beim Start per Meta-Daemon *inetd* auf Schwierigkeiten stießen, veranlassten wir den Start als gewöhnlichen Daemon, der immer im Hintergrund läuft. Hierfür muss die Konfigurationsdatei */etc/default/atftpd* angepasst werden (*OPTIONS=...* ist eine einzige Zeile):

```
USE_INETD=false
OPTIONS="--daemon -port 69 -tftpd-
➔timeout 300 -retry-timeout 5 -mcast-
➔port 1758 -mcast-addr 192.168.1.0-255 -
➔mcast-ttl 1 -maxthread 100 -verbose=5
➔/tftpboot“
```

Die augenfälligsten Änderungen gegenüber der vom System mitgebrachten Version die-



Mit dem Kommandozeilenclient *tftp* prüfen Sie, ob der *TFTP*-Daemon des Bootservers erreichbar ist.

ser Datei ist die Deaktivierung des *Inetd* und die Änderung der Multicast-Adresse auf das eigene Netz. Legen Sie das Verzeichnis */tftpboot* und ein Unterverzeichnis */tftpboot/pxelinux.cfg* an, und achten Sie darauf, dass beide für normale Nutzer les- und betretbar sind (*chmod 0755*). In das Verzeichnis */tftpboot* kopieren Sie den Bootloader *pxelinux.0* und das Bootmenü *menu.c32* (beide aus dem *Syslinux*-Tarball von [www.kernel.org/pub/linux/utils/boot/syslinux](http://www.kernel.org/pub/linux/utils/boot/syslinux)), sowie dem Speichertester *Memtest86+* (von <http://www.memtest.org>). Wir legten zudem Kernel und initiale Ramdisk des selbst gebauten Mini-Linux aus dem letzten Heft hier ab. Beide Dateien (sowie den passenden Artikel aus der letzten Ausgabe) finden Sie auf der Heft-DVD. Der Bootloader benötigt nun noch eine Konfigurationsdatei */tftpboot/pxelinux.cfg/default*. Deren Aufbau erinnert stark an die *Isolinux*-Konfiguration des Mini-Linux:

```
DEFAULT menu.c32
PROMPT 0
TIMEOUT 450

MENU TITLE Linux aus dem Netz

LABEL local
MENU LABEL Von Festplatte starten
MENU DEFAULT
LOCALBOOT 0

LABEL memtest
MENU LABEL Speichertest
KERNEL memtest
APPEND foo
```

## Achtung: Feinheiten!

► Die *LABELS* in der *PXELINUX*-Konfigurationsdatei dürfen maximal acht Zeichen lang sein. Wer besonders viele Einträge im Menü unterbringen muss, ist deshalb gut beraten, einfach durchzunummerieren. Bei der *VNC*-Installation von *openSUSE* muss das verwendete Passwort mindestens zehn Zeichen lang sein.

```

linuxlife - PXE Boot
Player Ethernet
PXELINUX 3.51 2007-06-10 Copyright (C) 1994-2007 H. Peter Anvin
JNDI data segment at: 00099BF0
JNDI data segment size: 4050
JNDI code segment at: 0009E950
JNDI code segment size: 0BBC
PXE entry point found (we hope) at 9E95:0106
My IP address seems to be C0A801C4 192.168.1.196
ip=192.168.1.196:192.168.1.250:192.168.1.252:255.255.255.0
TFTP prefix:
Trying to load: pxelinux.cfg/564d3286-f75c-4475-5742-b3e69cf2aae0
Trying to load: pxelinux.cfg/01-00-13-d4-fd-eb-49
Trying to load: pxelinux.cfg/C0A801C4
Trying to load: pxelinux.cfg/C0A801C
Trying to load: pxelinux.cfg/C0A801
Trying to load: pxelinux.cfg/C0A80
Trying to load: pxelinux.cfg/C0A8
Trying to load: pxelinux.cfg/C0A
Trying to load: pxelinux.cfg/C0
Trying to load: pxelinux.cfg/C
Trying to load: pxelinux.cfg/default
Unable to locate configuration file
Boot failed: press a key to retry, or wait for reset...
To grab input, press Ctrl+G

```

Hier fehlt die Konfigurationsdatei. Gut zu sehen ist die Suchreihenfolge für die Konfigurationsdatei.

```

LABEL minilinux
MENU LABEL Minilinux
KERNEL minilinux.bzi
APPEND initrd=minilinux.dsk rw vga=791

```

Die vollständige Konfigurationsdatei befindet sich auf der Heft-DVD. Bevor Sie den Start aus dem Netzwerk testen, sollten Sie prüfen, ob der TFTP-Server Dateien korrekt ausliefert. Das machen Sie mit dem Kommandozeilenclient *tftp*:

```
tftp 192.168.1.250
```

In der TFTP-Shell können Sie nun eine Datei anfordern:

## Auf DVD

➤ Sie finden folgende Dateien zum Artikel auf der DVD:

- *tftpboot.tgz*: Tarball mit dem kompletten Inhalt des Ordners */tftpboot* unseres Testsystems. Sie können ihn in */tftpboot* entpacken und als Ausgangspunkt für eigene Experimente verwenden. Der Start von *Memtest*, des Mini-Linuxes und der Installation von *openSUSE* und *Ubuntu* sollte ohne Anpassungen funktionieren.
- *dhcpd.conf*: Minimale Konfigurationsdatei für den DHCP-Server.
- *pxevmware.tgz*: VMware-Image für den Start per PXE und die Netzwerkinstallation eines Linux.
- *syslinux-3.51.tar.bz2*: Originaldateien von *SYSLINUX* und *PXELINUX*.
- PDF-Dateien zur Konfiguration des verwandten Bootloaders *EXTLINUX* und zum Selbstbau eines Mini-Linuxes, das alleine aus der *Initrd* läuft.

```
get /pxelinux.cfg/default
```

Wenn die Datei in weniger als einer Sekunde übertragen wird, können Sie den ersten Test mit einem „richtigen Client“ wagen und den Speichertest oder das Mini-Linux aus dem Netzwerk booten. Weil das Bootmenü als *DEFAULT localboot* eingetragen hat, kann es für das gesamte Netzwerk aktiv bleiben - achtet der Benutzer nicht auf das Bootmenü, wird nach 45 Sekunden von der Festplatte gestartet. Dennoch mag es wünschenswert sein, für einzelne Clients bestimmte Standardwerte zu setzen. Um dies zu gewährleisten, sucht *pxelinux.0* zunächst nach einer Konfigurationsdatei, die nach der MAC-Adresse mit vorangestelltem *01* benannt ist, dann nach der in Hexadezimal-Code (!) umgerechneten IP-Adresse, anschließend nach den ersten 7, 6, 5 Ziffern dieser Adresse. Die hexadezimal kodierte IP-Adresse macht nur in Rechenzentren Sinn, in denen ganze Cluster mit gleichen Kernen versorgt werden müssen. Wollen wir den in der DHCP-Konfiguration erwähnten Athlon 64 mit Hardwareadresse *00:13:d4:fd:eb:49* per Default von Festplatte booten, genügt als Konfigurationsdatei „01-00-13-d4-fd-eb-49“:

```

DEFAULT local
PROMPT 0
TIMEOUT 1

```

```

LABEL local
LOCALBOOT 0

```

Analog ist die Konfiguration eines hypothetischen Thin-Client, der ohne Bootmenü immer den gleichen Kernel und die gleiche *Initrd* bekommt:

```

DEFAULT thin
PROMPT 0
TIMEOUT 1

```

```

linuxlife -- Linux aus dem Netz
Von Festplatte starten
Speichertest
Minilinux
openSUSE Netinstall 32 Bit (lokal)
openSUSE Netinstall 32 Bit (UNC)
Ubuntu 7.04 Netinstall 32 Bit
Knoppix 5.1.1
Press [Tab] to edit options
To grab input, press Ctrl+G

```

Mit der Konfigurationsdatei *pxelinux.cfg/default* wird das Auswahlmenü geladen.

```

LABEL thin
KERNEL thinclient.bzi
APPEND initrd=thinclient.dsk weitere
Parameter

```

Soll für Wartungsarbeiten das umfangreiche Bootmenü geladen werden, genügt es, die Datei *01-00-13-d4-fd-eb-49* temporär nach *suspended-01-00-13-d4-fd-eb-49* zu verschieben - und nach getaner Arbeit wieder zurück.

## Fernsteuerinstallation

Bereits die simple Kombination aus DHCP-Server, TFTP-Server und PXELINUX erlaubt vielfältige Einsatzmöglichkeiten: So ist es möglich, standardmäßig eine Konfigurationsdatei zu verwenden, die von Festplatte starten lässt und diese per Cronjob kurz vor Feierabend gegen eine auszutauschen, welche ein Mini-Linux zur Datensicherung (beispielsweise zum Imaging per *ntfsclone*) startet. Der Anwender muss nun nur seinen PC neu starten, um eine Datensicherung durchzuführen. Nach getaner Arbeit fährt das Mini-Linux den PC wieder herunter. Ein zweiter Cronjob am frühen Morgen stellt wieder den Start von Festplatte als Default ein.

Außerdem können Sie mit dieser Konfiguration bereits die Remote-Installation von *openSUSE* veranlassen. Die ist in zwei Modi möglich: Per VNC vom Schreibtisch des Administrators oder lokal am Arbeitsplatz-PC, der installiert werden soll. Bei *openSUSE* genügen Installationskernel und -Ramdisk von der DVD oder aus dem Boot-Verzeichnis - für das Beispiel haben wir diese nach *osuse32.bzi* (Kernel) und *osuse32.dsk* (Ramdisk) kopiert. Alle Einstellungen zur Ferninstallation werden per Boot-Parameter vorgenommen, was in den folgenden Einträgen resultiert:

```

LABEL osuse32
MENU LABEL openSUSE Netinstall 32 Bit
↳(lokal)
KERNEL osuse32.bzi
APPEND initrd=osuse32.dsk splash=silent
↳showopts vga=791 install=serverpfad

```

```

LABEL osu32vnc
MENU LABEL openSUSE Netinstall 32 Bit
↳(VNC)
KERNEL osuse32.bzi
APPEND initrd=osuse32.dsk splash=silent
↳showopts vga=791 vnc=1
↳vncpassword=geheim1234
↳install=serverpfad

```

Geringfügig mehr Aufwand ist für die Installation von Ubuntu aus dem Netz notwendig, da das Installationssystem auf einige zusätzliche, per TFTP angeforderte Dateien angewiesen ist. Immerhin bringen aktuelle Ubuntu-Versionen im Verzeichnis *netboot/386* eine komplette Beispielkonfiguration mit. Verschieben Sie die Datei *pxelinux.cfg/default* nach *pxelinux.cfg/ubuntu* und kopieren Sie dann den gesamten Inhalt von *netboot/386* nach */tftpboot*. Sie können dann die Einträge für Ubuntu per Copy & Paste in Ihr PXE-Bootmenü übernehmen.

### Knoppix herangebeamt

Für den Start kompletter Linux-Distributionen aus dem Netz genügt der TFTP-Server zur Datenübertragung nicht, stattdessen kommt in der Regel ein NFS-Server zum Einsatz, der das Wurzeldateisystem ausliefert. Auf die Details der Konfiguration von dünnen und dicken Clients werden wir in dieser Ausgabe noch nicht eingehen, wollen aber Knoppix als Ausblick verwenden, wie die Konfiguration aussehen kann. Dabei hilft der Knoppix Terminal-Server, der eine komplette TFTP-Boot-Umgebung erzeugt. Starten Sie Knoppix von CD

oder DVD und rufen Sie im KDE-Menü *Knoppix/Dienste/Terminalserver* auf. Konfigurieren Sie den Dienst mit einer großzügigen Anzahl unterstützter Netzwerkkarten. Wenn Sie Knoppix als Wartungssystem verwenden wollen, sollten Sie die Option *secure:Root-Zugang auf den Clients abschalten* deaktivieren. Sie können jetzt aus dem Verzeichnis */tftpboot* den Kernel (*vmlinuz*), die modifizierte Ramdisk (*miniroot.gz*) „ernten“ und auf Ihren regulären Bootserver kopieren. Wie der Eintrag für die Knoppix-PXE-Clients aussehen muss, zeigt ein Blick in */tftpboot/pxelinux.cfg/default*. Auf dem Bootserver muss anschließend der Inhalt einer Knoppix-CD in ein Verzeichnis kopiert werden:

```

rsync -avHP /cdrom/
↳/exports/knoppix/

```

Dieses Verzeichnis geben Sie per NFS frei. Der passende Eintrag in der */etc/fstab* lautet für das Netzwerk *192.168.1.250*:

```

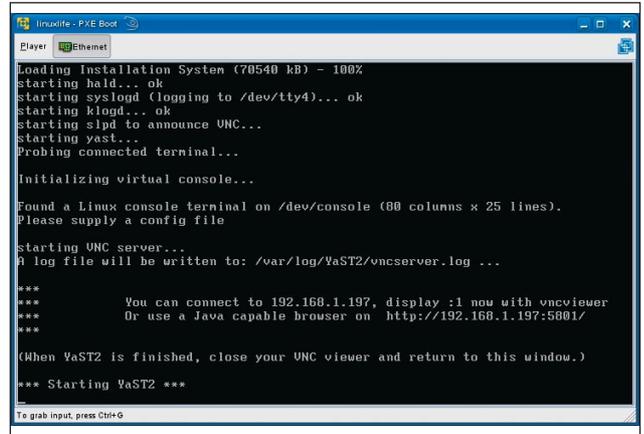
/exports/knoppix
↳192.168.1.0/24(ro,no_
↳root_squash)

```

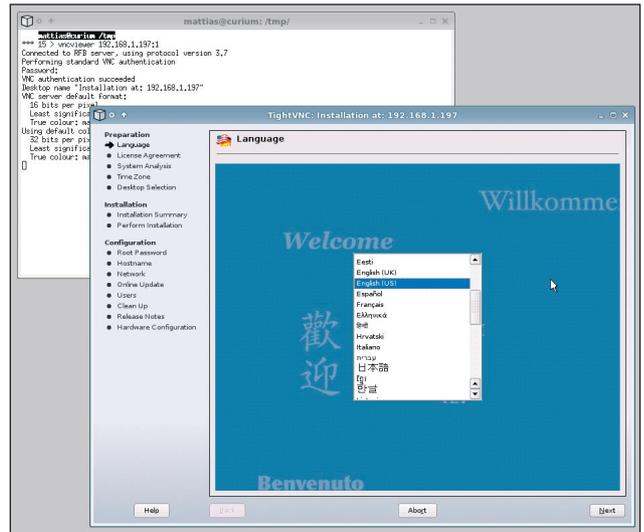
Auf diesen Pfad muss noch der Parameter *nfsroot* in Knoppix PXELINUX-Konfiguration angepasst werden, damit das Live-Linux regulär im Netzwerk verfügbar ist. Starten Sie den Dienst *nfs-kernel-server* und probieren Sie den Start von Knoppix per PXE aus.

### Fazit

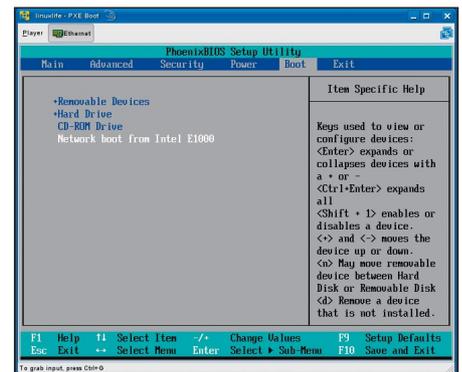
PXE verhilft jedem Netz zu einer enormen Flexibilität. Die Vorteile des praktisch überall vorhandenen Netzwerkboots kommen nicht nur großen Firmen zugute, wo sie dem Admin lange Wege ersparen, es profitieren auch PC-Händler, die häufig mit Memtest oder Knoppix Diagnosen fahren müssen - oder unser Testlabor, in dem häufig verschiedene Linux-



Eine per PXE gestartete openSUSE-Installation...



...ist per VNC vom Schreibtisch des Administrators aus durchführbar.



Wie ein richtiges BIOS: Eine VMware im Bridge-Mode ist ideale Testumgebung

Distributionen auf realer Hardware oder in einer VMware-Instanz benötigt werden. Im nächsten Heft erklären wir die Erweiterung des Prinzips auf plattenlose Clients, die auch ihr Root-Dateisystem aus dem Netzwerk beziehen.

jkn

**Weblinks**

- Die Syslinux-Familie stellt Bootloader für EXT2, ISOs und den Start aus dem Netzwerk zur Verfügung: <http://syslinux.zytor.com>
- Bootet ein Rechner nicht per PXE, kann mit Etherboot ein Diskettenimage mit einem PXE-kompatiblen Bootloader erzeugt werden: <http://www.etherboot.org>
- Der Wikipedia-Eintrag beschreibt den Bootvorgang per PXE im Detail. [http://de.wikipedia.org/wiki/Preboot\\_Execution\\_Environment](http://de.wikipedia.org/wiki/Preboot_Execution_Environment)