

## 16 Schlanke Endgeräte

Nach der Hochblüte der Mainframes und Zeichen-Terminals erlebten wir die Client-Server-Ära, in der Anwender überwiegend Windows-PCs und Windows-Programme als Schnittstelle zur Datenverarbeitung nutzten.

Solche klassischen Windows-PCs, die von ihrer lokalen Festplatte booten und Programme ausführen, die von ihrer lokalen Platte starten, können durch den hohen Arbeitsaufwand und die vielen Sicherheitsprobleme die Systemverwalter zur Verzweiflung bringen. Diese Client-Server-Umgebungen mit Windows-PCs und Windows-Anwendungen verursachen zudem nach voneinander unabhängigen Untersuchungen amerikanischer Unternehmensberater im Mittel jährlich Kosten von ca. 10.000 € pro Arbeitsplatz. Hauptkostenfaktor sind nicht etwa die Server, um die es hauptsächlich in diesem Buch geht, sondern der Support für die Benutzerarbeitsplätze und die Benutzer. Unternehmen und sonstige Einrichtungen ohne Kostenstellen-/Kostenträgerrechnung nehmen vielleicht nicht wahr, wie viel produktive Arbeitszeit durch Abstürze von Windows-PCs, Reparaturversuche durch Anwender, Installieren von Bildschirmschonern und privaten Anwendungen etc. und die Mühen des Benutzer-Supports, dann wieder funktionsfähige Arbeitsplätze herzustellen, verloren geht. Die Kosten tragen sie, auch ohne zu wissen, woher sie kommen.

Diesen Aufwand versuchen Administratoren, u. a. durch Hardware- und Software-Lösungen zu mindern, wie Steckkarten, die das Ändern von Platteninhalten kontrollieren und durch Fernverwaltungs- und Administrations-Software.

Dieses Buch lenkt den Blick auf radikalere Lösungen, mit denen die Administratoren nachts wieder ruhig schlafen können:

- Terminals für Server-basierte Datenverarbeitung (*Serverbased Computing*), die an den Benutzerarbeitsplätzen Anwendungen an möglichst *schlanken* Endgeräten (so genannten Thin Clients) anzeigen, die auf Servern ablaufen und
- Workstations, die statt lokaler Festplatten zentrale Platten im Netz nutzen.

Zunächst lernen Sie Konzepte für Endgeräte und Appliances kennen. Danach folgen technisch ausgelegte Abschnitte zum Einrichten sehr kostengünstiger Linux-basierter Terminals und plattenloser Workstations.

## 16.1 Konzepte für schlanke Endgeräte

Lesen Sie hier über Lösungen mit schlanken Endgeräten, die entweder nur Arbeitssitzungen auf Terminal-Servern darstellen oder selbst lokal Anwendungen ausführen:

- PCs,
- Terminals,
- Linux-Diskless-Geräte mit Flash-Rom und
- Browser-Appliances sowie
- Diskless Workstations

### 16.1.1 PCs für Terminal-Sitzungen

Terminal-Dienst-Sitzungen von Windows oder Linux-Terminal-Servern oder Mainframes kann man auf traditionell mit Festplatten ausgestatteten PCs mit Windows- oder Linux/Unix-Betriebssystemen nutzen, indem man diese mit Clients für die verschiedenen Transportprotokolle wie

- Remote Desktop Protocol (*RDP*) von Microsoft,
- Independent Computing Architecture (*ICA*) von Citrix für Microsoft,
- Adaptive Internet Protocol (*AIP*) von Tarantella,
- X.11 für Linux/Unix,
- Nomachine NX von Nomachine für Microsoft-Terminal-Dienste und für Linux/Unix,
- 3270 für IBM-Mainframes
- oder Web-Clients ausrüstet.

Dies hat in Organisationen mit unflexiblen Mitarbeitern den praktischen Vorteil, dass diese ihre bisherigen Endgeräte behalten und vielleicht gar nicht merken, wenn sie Server-basierte Anwendungen statt lokaler Programme nutzen.

Technische Herausforderungen sind hier u. a. das Einbinden lokaler Geräte wie Drucker, Laufwerke, Messgeräte etc.

#### *Flash-Rom-Terminals*

Flash-Rom-Terminals halten ihr Betriebssystem in lokalen Halbleiterspeichern statt auf Festplatten. Je nachdem ob man Settop-Boxen für Media-Anwendungen mitzählt oder nicht führen im Markt hier Linux-Geräte vor Windows-Geräten oder umgekehrt. Inzwischen bietet die Terminal-Industrie eine immer größere Aus-

wahl so genannter Thin Clients, die überwiegend von lokalen Speichern, aber auch von Boot-Servern (siehe nächster Abschnitt) oder von benachbarten Terminals booten können.

Microsoft hat Windows-Terminals mit all seinen Varianten, angefangen mit Windows CE bis zu den aktuellsten .NET Windows-Varianten an Terminal-Hersteller wie Wyse, Bounless, IGEL, NCD, Tektronix etc. lizenziert. Linux-Lösungen bieten u. a. die Hersteller Esesix, IGEL, NCD, Neoware, Siemens und Wyse. Diese Linux-Lösung steckt übrigens weltweit in Millionen von Settop-Boxen für den Internet-Zugang und in Receivern für terrestrisches sowie Satelliten-Fernsehen/Rundfunk.

Solche Terminals sind einfach einzurichten und mit Fernverwaltungslösungen zentral administrierbar. Das Setup vieler solcher Terminals fragt nur nach allgemeinverständlichen Daten wie die Adresse des Servers, IP-Adresse oder DHCP und Endbenutzer-Sprache. Übertroffen wird ihre Flexibilität nur noch von Geräten mit Boot-Prom, die ganz darauf verzichten, lokal Programme vorzuhalten.

Browser-Appliances sind Endgeräte, die nur einen Browser als Benutzerschnittstelle für alle möglichen Programme bieten. Appliances können sich in Settop-Boxen für Fernseher, in Bildschirm-Telefone, in Kiosk-Systeme und noch viele andere Geräteformen hüllen. Linux scheint sich immer mehr als Betriebssystem dieser Einfachst-Geräte durchzusetzen, zumal für das Open-Source-Betriebssystem Linux und den Open-Source-Browser Mozilla keine Lizenzgebühren anfallen. In vielen dieser Appliances steckt die hier beschriebene Flash-Rom-Technik.

#### *Linux-Terminals mit Boot-Prom*

Eine noch erheblich sparsamere Form von Linux-Geräten verzichtet ebenfalls auf Fest- und Wechselplatten und bootet Linux übers lokale Netz von Boot-Servern. Hierfür gibt es kommerzielle Lösungen, beispielsweise von Bootix (<http://www.bootix.com>) und zahlreiche freie Linux-Projekte, z. B. vom Linux-Terminal-Server-Projekt (*LTSP*). Bei Boot-Prom-Lösungen veranlasst ein sehr kleines Programm im Boot-Prom der Netzwerkkarte in mehreren Schritten in einem Dialog mit einem Boot-Server, hier im Buch einem Linux-Boot-Server, das Betriebssystem Linux von eben diesem Boot-Server zu laden. Die Grundidee und verschiedenen Version dieser Technologie beschreiben die folgenden Abschnitte.

Sollen die Diskless-Geräte nur als Terminals arbeiten, reicht schon ein Pentium PC ab 16 MB RAM mit passender Netzwerkkarte mit PXE/Boot-Prom. Zwar verbringen Systemverwalter zunächst mehr Zeit mit dem Einrichten, können aber später alle Installationen und Änderungen zentral auf dem Boot-Server pflegen. Während die zuvor erwähnten Lösungen mit Flash-Roms auch Endanwender ohne Computerkenntnisse zum Laufen bringen können, müssen hier erst Systemverwalter Server-seitig fleißig sein. Sie haben zum Glück kaum mehr Aufwand, um 100 Diskless-PCs zu betreiben, als eine Handvoll herkömmlicher PCs einzurichten. Wenn

alles läuft, ist es genauso Endanwender-geeignet wie Flash-Rom-Lösungen. Wer 486er PCs sonst entsorgen müsste oder geschenkt bekommt, zaubert mit weniger als 20 € Materialkosten funktionsfähige Endgeräte ohne Festplatte und Diskettenlaufwerk hervor. Zu den 20 € kommen ggf. noch Kosten für neue Grafik- und Soundkarten hinzu.

### 16.1.2 *Diskless Linux Workstations*

Workstations kann man ähnlich wie Terminals ohne lokale Festplatte betreiben. Je mehr Workstations man zentral bootet und verwaltet, desto höher ist die hiermit verbundene Ersparnis an Hardware, Installationspflege- und Betreuung.

Diskless Workstations kann man u. a. mit spezieller Hardware betreiben, die die Signale des IDE- oder SCSI- Festplatten-Controllers auf Ethernet umsetzt, oder ähnlich wie bei Boot-Prom-Terminals mit dem hier beschriebenen DXS-Projekt von Dirk von Suchodoletz.

### 16.1.3 *Überblick*

Dieses Kapitel zum Booten schlanker Endgeräte erklärt zunächst, wie man Linux-PCs als Linux-X-Terminals verwenden kann, vermittelt dann die Grundlagen von Diskless-PCs, die mit Boot-Proms starten und erklärt schließlich im Detail zwei konkrete Verfahren, um Endgeräte über ein Netz von einem Linux-Server statt von ihrer lokalen Festplatte zu booten: Das Linux-Terminal-Server Projekt (*LTSP*) für Linux-Terminals für Remote-Anwendungen über Terminal-Dienste und das DXS-Projekt für Linux-Workstations, die Anwendungen lokal ausführen, aber statt lokaler Platten solche im Netz verwenden.

## 16.2 **PCs als X-Terminals**

In vielen Organisationen und Kleinbüros gibt es noch unzählige Pentium-Computer mit stabiler Hardware, die für die Anforderungen der aktuellen Ressourcenhungrigen Windows-Versionen einfach zu behäbig sind und zu wenig Arbeitsspeicher bieten.

Viele dieser Geräte kann man noch viele Jahre lang hervorragend an Benutzer-Arbeitsplätzen als X-Terminals einsetzen, um darauf Server-basierte Linux- und Windows-Anwendungen zu nutzen. Wenn die eingebaute Grafikkarte weniger als  $1024 \times 786$  Bildpunkte oder eine Farbtiefe von weniger als 16 Bit bietet, spendieren Sie den alten Pentiums vielleicht noch eine neue Grafikkarte, solange es noch Grafikkarten für den PCI-Bus gibt.

Solche Geräte können Sie von Ihrer lokalen Platte oder von einem CD-ROM-Laufwerk booten oder ohne lokale Laufwerke von Linux-Servern.

Lesen Sie zunächst, wie Sie einen Linux-Server und die Festplatte des Uralt-Pentium-PCs vorbereiten müssen, um die bewährten Geräte als lokal bootende X-Terminals nutzen zu können:

- Auf dem Server konfigurieren Sie das X-System so, dass es Anfragen von anderen Rechnern zulässt.
- Auf den Client-Rechnern installieren Sie ein minimales Linux-System mit X-System, das eine Verbindung mit dem Linux-Server aufnehmen kann.

### 16.2.1 X-Konfiguration auf dem Server

Die Beschreibung geht davon aus, dass auf dem Linux-Server ein X-System mit KDE installiert und konfiguriert ist. Wenn ein Benutzer lokal an diesem Rechner X startet, sollte er seinen KDE-Desktop erhalten.

Mit nur wenigen Anpassungen können sich Benutzer grafisch über das Netz anmelden, ohne dabei unnötigerweise die grafische Oberfläche lokal zu starten.

Den *xdm* kann man durch Änderungen in der Datei `/etc/X11/xdm/Xservers` daran hindern, lokal zu starten:

```
/etc/X11/xdm/Xservers
```

```
# Xservers - local X-server list
#
# This file should contain an entry to start the server on the
# local display; if you have more than one display (not screen),
# you can add entries to the list (one per line).
# If you also have some X terminals connected which do not
# support XDMCP, you can add them here as well; you will want to
# leave those terminals on and connected to the network, else
# kdm will have a tougher time managing them. Each X terminal
# line should look like:
#       XTerminalName:0 foreign
#
# use such a line to enable the console login option in the
# kdm menu
#:0 local@tty1 /usr/X11R6/bin/X vt7
# "reserve" means that the X server gets only started on
# request (only kdm)

#:0 local /usr/X11R6/bin/X -nolisten tcp -br vt7
#:1 local reserve /usr/X11R6/bin/X -nolisten tcp -br :1 vt8
#:2 local reserve /usr/X11R6/bin/X -nolisten tcp -br :2 vt9
#:3 local reserve /usr/X11R6/bin/X -nolisten tcp -br :3 vt10
#:4 local reserve /usr/X11R6/bin/X -nolisten tcp -br :4 vt11
#:5 local reserve /usr/X11R6/bin/X -nolisten tcp -br :5 vt12
```

Dazu deaktivieren Sie die fett hervorgehobenen Zeilen durch ein Kommentar-Zeichen # in der ersten Spalte. Danach startet der X-Server nicht mehr lokal.

Außerdem muss man noch mit einem Kommentar-Ausrufezeichen in der Datei /etc/X11/xdm/xdm-config den Kontakt zum lokalen Displaymanager verhindern.

/etc/X11/xdm/xdm-config

```
!
! xdm-config: Configuration of the xdm
!
DisplayManager.errorLogFile:    /var/log/xdm.errors
DisplayManager.pidFile:         /var/run/xdm.pid
DisplayManager.authDir:         /var/lib/xdm
DisplayManager.keyFile:         /etc/X11/xdm/xdm-keys
DisplayManager.servers:         /etc/X11/xdm/Xservers
DisplayManager.accessFile:      /etc/X11/xdm/Xaccess
DisplayManager.willing:         su nobody -c /etc/X11/xdm/
Xwilling
!
! ATTENTION: `authName' should be in general MIT-MAGIC-COOKIE-1
! For XDM-AUTHENTICATION-1 which is default for xterminals see
! manual page of xdm and the manual coming with the xterminal.
!
DisplayManager.*.authName:      MIT-MAGIC-COOKIE-1
DisplayManager.*.authComplain:  false
!
! All displays should use authorization, but we cannot be sure
! X terminals will be configured that way, so by default
! use authorization only for local displays :0, :1, etc.
!
DisplayManager._0.authorize:     true
DisplayManager._1.authorize:     true
!
! The scripts handling the setup, the startup, the session
! its self, and the reset of an X session.
!
DisplayManager.*.setup:          /etc/X11/xdm/Xsetup
DisplayManager.*.chooser:        /etc/X11/xdm/RunChooser
DisplayManager.*.startup:        /etc/X11/xdm/Xstartup
DisplayManager.*.session:        /etc/X11/xdm/Xsession
DisplayManager.*.reset:          /etc/X11/xdm/Xreset
!
DisplayManager._0.terminateServer: true
!
DisplayManager*resources:        /etc/X11/xdm/Xresources
DisplayManager.*.terminateServer: false
!
```

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals
! with xdm
!
! DisplayManager.requestPort: 0
```

Deaktivieren Sie in dieser Datei die letzte Zeile durch Voranstellen eines Ausrufezeichens ! und Leerzeichens. Dies wird den Kontakt zum lokalen Display-Manager unterbinden.

Damit *xdm* zukünftig bei jedem Start des Rechners automatisch aktiv ist, gehen Sie im YaST-Kontrollzentrum unter *System • Editor für /etc/sysconfig Dateien* auf *Desktop • DISPLAYMANAGER*. Hier achten Sie darauf, dass unter *DISPLAYMANAGER* der Wert *xdm* oder *kdm* eingetragen ist. Wenn Sie mit *kdm*, dem Display-Manager von KDE, arbeiten wollen, muss zusätzlich die Variable *DISPLAYMANAGER\_REMOTE\_ACCESS* auf *yes* stehen.



Abbildung 16.1: *DISPLAYMANAGER*, Einstellungen in YaST

Mit jedem

```
init 5
```

wechseln Sie in den Run-Level mit grafischem Login. Dadurch startet *xdm* oder *kdm*. Der Start des Display-Managers löst nach den beschriebenen Änderungen auf dem Linux-Server keinen sichtbaren Effekt aus. Die zugehörigen Prozesse finden Sie natürlich in der Prozessliste.

### 16.2.2 Installation auf dem Client

Bei der hier beschriebenen Administrator-freundlichen Lösung arbeiten die Benutzer an Linux-Endgeräten, die nur Sitzungen auf dem Linux-Terminal-Server darstellen müssen. Diese Linux-Endgeräte sollen von einem eigenen Laufwerk wie Festplatte oder CD-ROM booten.

Um nur grafische Sitzungen darzustellen, sind die Anforderungen an so einen Linux-Client nicht sehr hoch, er sollte aber über eine Grafikkarte verfügen, die eine Auflösung von 1024 × 768 Punkten bei 16 Bit Farbtiefe darstellen kann.

Auf dem Client-PC bzw. einem Muster eines Client-PCs sollten Sie eine minimale Version von Linux mit X und dem Paket *xdmisc* aus der Serie *System • X11* installieren. Die anderen Auswahlmöglichkeiten wie Büroanwendungen, Dokumentation und KDE sind nicht notwendig.

**Tipp:** Auf älteren Rechnern mit weniger als 64 MByte Hauptspeicher ist auch die Linux-Installation sehr langsam. Bauen Sie einfach die Festplatte dieses Computers in ein aktuelles Gerät und installieren es dort. Nach der Installation bauen Sie die Festplatte zurück und vervollständigen die Konfiguration.

Das hier beschriebene Minimalsystem hat einen Umfang von ca. 350 MByte. Je nach Boot-Gerät können Sie über den Grundbedarf hinaus vorhandenen Speicher verschieden verwenden:

- Wenn Sie es von einer CD-ROM booten, ist noch Platz für kleine Linux-Anwendungen und
- wenn Sie von einer Festplatte booten, steht der ganze darüber hinausgehende Platz für eine Swap-Partition oder Linux-Anwendungen zur Verfügung.

Bei X-Terminal-Clients müssen Sie damit rechnen, dass Benutzer diese einfach ausschalten, statt sie geordnet herunterzufahren. Bei Festplatten sollten Sie daher als Dateisystem *reiserfs* wählen, dem solche Misshandlungen fast nichts anhaben.

Sobald X lokal läuft, können Sie Ihren Client in wenigen Schritten mit YaST zum X-Terminal gestalten.

Die notwendigen Schalter finden Sie im YaST-Kontrollzentrum unter *System • Editor für /etc/sysconfig Dateien* unter *Desktop • XDMSC*. Hier ändern Sie die folgenden Werte:

```
RX_RHOST="192.168.1.2"      die IP-Adresse Ihres X-Servers
RX_XDMCP="query"
START_RX="yes"
```



Um danach das System zu testen, rufen Sie die grafische Oberfläche auf:

```
/etc/init.d/rx tty7
```

Damit starten Sie die grafische Oberfläche und binden sie an die Konsole 7 (tty7).

In ein etwas nüchternes Anmeldefenster können Sie bzw. Ihre Benutzer dann Ihren Benutzernamen und Kennwort eingeben. Wenn alles funktioniert, startet nach dieser Anmeldung die grafische Oberfläche mit KDE, die Programme laufen auf dem Linux-Server, und Sie oder Ihre Benutzer arbeiten am Linux-Client.

Damit das Anmeldefenster beim Login weiterer Benutzer immer wieder erneut starten kann, müssen Sie noch die Datei `/etc/inittab` anpassen.

```
#
# /etc/inittab
#
# Copyright (c) 1996-2002 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Florian La Roche <feedback@suse.de>, 1996
#
# This is the main configuration file of /etc/init, which
# is executed by the kernel on startup. It describes what
# scripts are used for the different run-levels.
#
# All scripts for runlevel changes are in /etc/init.d/.
#
# This file may be modified by SuSEconfig unless CHECK_INITTAB
# in /etc/sysconfig/suseconfig is set to "no"
#

# The default runlevel is defined here
id:5:initdefault:

..... (gekürzt)

# getty-programs for the normal runlevels
# <id>:<runlevels>:<action>:<process>
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
#
```

```
#S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102

#
# Note: Do not use tty7 in runlevel 3, this virtual line
# is occupied by the program xdm.
#

# This is for the package xdmisc, after installing and
# and configuration you should remove the comment character
# from the following line:
#7:3:respawn:+/etc/init.d/rx tty7

# modem getty.
# mo:235:respawn:/usr/sbin/mgetty -s 38400 modem

# fax getty (hylafax)
# mo:35:respawn:/usr/lib/fax/faxgetty /dev/modem

# vbox (voice box) getty
# I6:35:respawn:/usr/sbin/vboxgetty -d /dev/ttyI6
# I7:35:respawn:/usr/sbin/vboxgetty -d /dev/ttyI7

# end of /etc/inittab
```

Durch Entfernen des Kommentar-Zeichens `#` vor der im Listing fett hervorgehobene Zeile aktivieren Sie den Eintrag für die siebte Konsole `tty7`. Diese Änderungen tritt in Kraft, sobald Sie

```
telinit q
```

aufzurufen. Danach sollte das grafische Anmeldefenster starten.

Wenn Sie Ihre Arbeitsplatz-PCs statt von lauten lokalen Festplatte mit geräuschlosen Boot-Proms von Linux-Boot-Servern starten wollen, lesen Sie bitte im nächsten Unterkapitel weiter. Wollen Sie hingegen sofort schauen, wie Sie auf den X-Terminals mit Windows arbeiten, blättern bitte Sie zum Kapitel 18 weiter. Dieses beschreibt, wie Sie auf Linux-Servern auf einer Middleware von NeTraverse das Ihren Anwendern vielleicht vertraute Windows 98 und Windows-Anwendungen laufen lassen.

### 16.3 Grundlagen für Flash-Rom Diskless PCs

Dieser Abschnitt beschreibt grundlegende Protokolle für das Booten laufwerkloser PCs von (Linux-)Boot-Servern. Dabei geht er auf alternative Boot-Proms (PXE und Etherboot) und Dienste wie DHCP, TFTP, ATFTPD, NFS und XDM-Server ein.

### 16.3.1 PCs aus dem Netz booten

Für den ersten Eindruck sollten Sie das BIOS Ihrer Clients prüfen. Vielleicht finden Sie hier schon die Einstellung LAN neben den klassischen Boot-Medien Festplatte, Diskette oder CD-Rom. Diese Einstellung bieten Clients mit modernem BIOS und entsprechendem Netzwerkadapter.

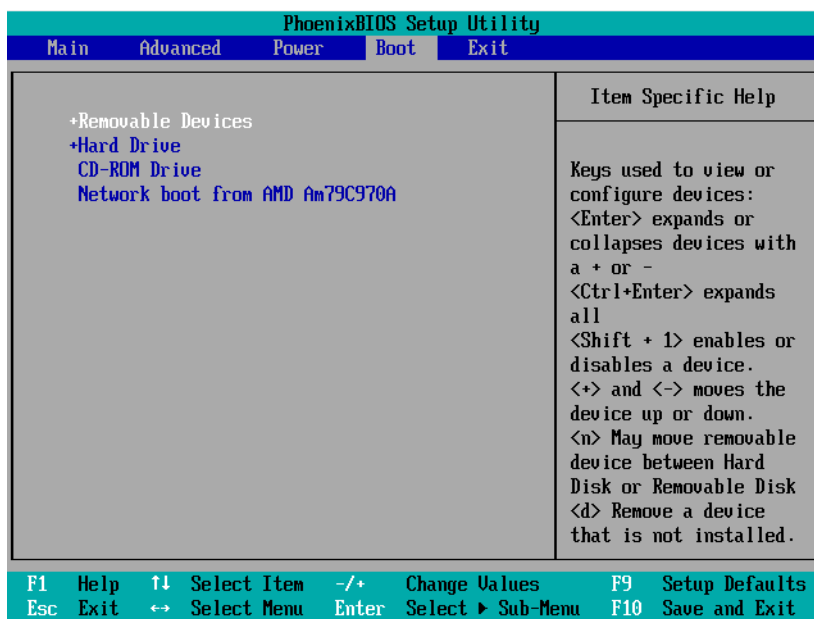


Abbildung 16.2: PXE-Aktivierung

Mainboards benötigen hier eine onboard LAN-Karte auf der Hauptplatine oder einen Netzwerkadapter mit lokalem EPROM. In vielen Fällen steht dann eine proprietäre BIOS-Erweiterung von Intel, die Preboot-eXtension-Environment (PXE™) zur Verfügung. Dieses müssen die Systemverwalter vielleicht noch einschalten. Oft blendet das BIOS beim Start eine Meldung: Press *CTRL+S* for *PXE configuration* oder Ähnliches ein. Diese weist darauf hin, dass Intels PXE vorhanden ist und führt Administratoren zu einer Schnittstelle zum Einrichten der PXE-Erweiterung.

PXE-Menüs sind einfach aufgebaut und erlauben oft nur wenige Einstellungen. Die wichtigste Option ist das Einschalten von Intels PXE, damit das BIOS bei der Suche nach Boot-fähigen Datenträgern das Netzwerk einbezieht. In diesem Menü können Administratoren auch festlegen, ob eine Meldung auf die PXE-Konfiguration hinweisen soll.

Nach der Konfiguration startet der PC automatisch neu und zeigt nach Ablauf der BIOS-Initialisierungsroutinen abhängig von den BIOS-Einstellungen folgendes Bild:

```

Network boot from AMD Am79C970A
Copyright (C) 2003 VMware, Inc.
Copyright (C) 1997-2000 Intel Corporation

CLIENT MAC ADDR: 00 0C 29 6B E8 12  GUID: 564D42F5-495E-ED51-D13A-E123876BE812
DHCP. i

```

Abbildung 16.3: PXE versucht per DHCP eine Adresse zu beziehen

**Tipp:** Hier spielt die im BIOS definierte Boot-Reihenfolge eine entscheidende Rolle: Steht die LAN-Option vielleicht erst an dritter Stelle nach dem Booten von CD-Rom oder Festplatte, bekommen Administratoren PXE gar nicht zu sehen, wenn der PC von einem anderen Datenträger erfolgreich starten konnte. Deshalb sollte man die LAN-Option stets an erster Stelle eintragen.

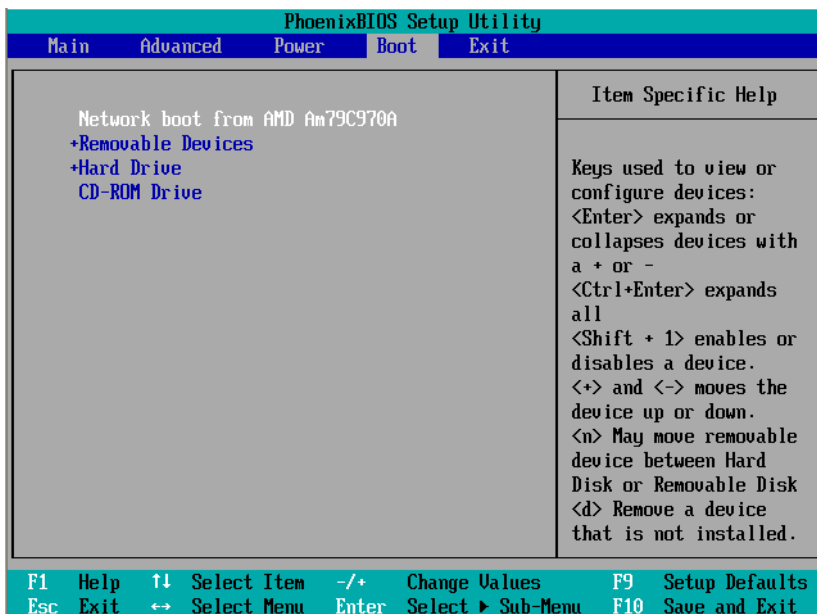


Abbildung 16.4: Einstellen der Boot-Reihenfolge

Läuft im lokalen Netzwerk ein DHCP-Server, der den Client bereits kennt oder diesen akzeptiert, erhält dieser eine IP-Adresse zugewiesen. Dazu muss der PC vielleicht schon im festplattenbasierten Modus erfolgreich eine Adresse vom DHCP bezogen haben oder der DHCP-Server über einen Pool von Adressen verfügen, die er frei verteilen darf. Diese Fragestellungen erörtert das Kapitel 5, *Zugriff von Windows auf Linux-Server*, näher. Hat der PC erfolgreich eine IP-Adresse bekommen, meldet sich PXE üblicherweise so:

```

Network boot from AMD Am79C970A
Copyright (C) 2003  VMware, Inc.
Copyright (C) 1997-2000  Intel Corporation

CLIENT MAC ADDR: 00 0C 29 EF 98 C4  GUID: 564D6859-B3D8-865A-2EB8-4AE452EF98C4
CLIENT IP: 192.168.1.141  MASK: 255.255.255.0  DHCP IP: 192.168.1.1
GATEWAY IP: 192.168.1.1
TFTP.
PXE-T01: File not found
PXE-E3B: TFTP Error - File Not found
PXE-M0F: Exiting Intel PXE ROM.
Operating System not found
-

```

Abbildung 16.5: DHCP erfolgreich, TFTP schlägt noch fehl

PXE konnte eine IP-Adresse von einem DHCP-Server beziehen. TFTP schlug fehl, weil der Server dafür noch nicht eingerichtet war.

Anschließend versucht der PXE-Client, vom Server, der auch die IP-Konfiguration vergeben hat, einen Betriebssystem-Kernel oder eine Startumgebung per Trivial FTP (*TFTP*) zu laden. Dieser wird jedoch auf den wenigsten Servern aktiviert sein. Deshalb meldete PXE wie in der letzten Abbildung einen Fehler und bricht ab.

Der nächste Abschnitt widmet sich daher dem Boot-Server für Diskless Clients.

### 16.3.2 Der Boot-Server

Alle im folgenden beschriebenen Konfigurations- und Installationsschritte erfordern Administratorrechte. Einfache Benutzer können nicht ohne weiteres Systemdienste installieren, konfigurieren oder starten. Ein Server für Diskless-Clients benötigt mindestens die beiden Dienste DHCP und TFTP. Der erste läuft vielleicht schon, der zweite ist eher selten bereits installiert. Die besonders kompakten Linux-Terminal-Projekte ThinStation und PXES kommen mit diesen beiden Diensten

aus. Für Diskless X-Stations (DXS) und LTSP benötigen Sie mindestens noch NFS. DHCP ist hier im Kapitel 2 beschrieben und NFS im Kapitel 8. Für Experimente mit Diskless Clients überprüfen Sie am besten zuerst, ob die notwendigen Dienste bereits auf Ihrem Server installiert sind:

```
s2:/root # rpm -q nfs-utils atftp dhcp-server
nfs-utils-1.0.6-103
atftp-0.7cvs-25
dhcp-server-3.0.1rc13-28.17
```

Sollte dieses nicht der Fall sein, installieren Sie diese Dienste wie gewohnt. Dazu müssen Sie die Pakete mit RPM oder YaST nachinstallieren. Die Dienste lassen sich leicht einrichten, wie die nachstehenden Beispiele zeigen.

Im Beispiel besitzt der Server im Klasse-C-Netzwerk 192.168.1.0 die Adresse 192.168.1.1, das Default-Gateway oder der Router in den Rest der Welt sei die 192.168.1.254. Der Test-Client erhält die Adresse 192.168.1.101.

### 16.3.3 DHCP

Bitte lesen Sie zunächst die allgemeine Einführung zu DHCP im Kapitel 5. In den meisten Fällen genügt es, für DHCP zwei Dateien anzupassen. Welche Clients der DHCP-Server mit welchen Informationen bedient, legt die Datei `/etc/dhcpd.conf` fest. Für einen ersten Test genügen die folgenden Eintragungen:

```
ddns-update-style none;
subnet 192.168.1.0 netmask 255.255.255.0 {
server-identifier 192.168.1.2;
option root-path "/nfsroot/ldc";
filename /nfsexport/ldc/boot/pxelinux.0;
option routers 192.168.1.1;
host ldc-test {
    hardware ethernet 00:11:09:1D:64:E3;
    fixed-address 192.168.1.101;}
}
```

In der `/etc/sysconfig/dhcpd` bestimmen Sie, dass Ihr DHCP-Server auf dem passenden Interface, im Beispiel `eth0`, lauscht und zur Sicherheit in einer CHROOT-Umgebung läuft:

```
DHCPD_INTERFACE="eth0"
DHCPD_RUN_CHROOTED="yes"
[ ... ]
```

Die weiteren Einstellungen können Sie problemlos übernehmen.

```
rcdhcpd start
```

wertet diese Datei beim Start des Dienstes aus.

#### 16.3.4 ATFTPD und PXELinux

Bei dem Advanced Trivial File Transfer Protokoll (ATFTPD) muss man noch weniger einrichten. Hier passen Sie die Datei `/etc/sysconfig/atftpd` durch den Eintrag

```
ATFTPD_OPTIONS="--daemon "
ATFTPD_USE_INETD="no"
ATFTPD_DIRECTORY="/nfsroot"
```

an. Das bedeutet, dass beim Start dieses Dienstes die Clients unterhalb des eingetragenen Pfads Dateien vom Server laden dürfen. Damit der Client auch weiß, wo er suchen muss, teilt ihm der DHCP-Server dies vorher mit. Diese Information tragen Sie in der Konfigurationsdatei des DHCP-Servers in der Option *filename* ein. PXE kann große Kernels von mehreren Megabyte Umfang nicht sofort laden. Für dieses Problem enthält das Syslinux-Paket von Peter Anwin eine PXE-Boot-Option. PXE startet dabei `pxelinux.0`, ein kleines Binärprogramm, das seinerseits alle weiteren Schritte veranlasst. Es sollte an einer per TFTP erreichbaren Stelle liegen. Zum Testen legen Sie die Verzeichnisstruktur mit dem folgenden Befehl an:

```
mkdir -p /nfsexport/ldc/boot/pxelinux.cfg
```

Im DHCP-Server tragen Sie entsprechend `filename /nfsexport/ldc/boot/pxelinux.0` ein, wenn das Programm an dieser Stelle installiert ist. Entsprechend muss die Konfiguration für den TFTP-Dienst das Unterverzeichnis `/nfsexport/ldc/boot` freigeben. Für beide Dienste gilt: Haben eine der Konfigurationsdatei geändert, müssen Sie die Dienste neu starten:

```
rcatftpd restart
rcdhcpd restart
```

**Tipp:** Wenn Sie vergessen, die Dienste neu zu starten, registrieren diese Ihre Änderungen nicht.

Der Start mit `pxelinux.0` arbeitet ähnlich wie `isolinux`, das Sie vielleicht bereits vom Start der Knoppix-CD her kennen: Knoppix ist eine Debian-basierte Linux-Distribution, die nur von CD oder DVD läuft und zum Starten und Ausführen von Programmen keine Festplatten benötigt.

`pxelinux.0` hat mit `isolinux` einige Konfigurationsoptionen gemeinsam und darüber noch eigene Parameter. `pxelinux.0` erwartet seine Konfigurationsdatei im Verhältnis zur eigenen Lage im Verzeichnis `./pxelinux.cfg`. Dort lädt es, wenn es keine speziellen PC-spezifischen Konfigurationsdateien findet, eine Datei namens `default`. In dieser hinterlegen Sie Einstellungen, die etwa so aussehen sollten:

```
prompt 1
timeout 100
default dxs
label dxs
kernel dxs
append vga=normal initrd=initrd-dxs
ipappend 1
```

Die meisten Parameter sind selbsterklärend. Die Einstellung geht davon aus, dass im Unterverzeichnis `/nfsexport/ldc/boot` der Kernel mit dem Namen `dxs` und eine dazu passende RAM-Disk mit dem Namen `initrd-dxs` liegen. Das erreichen Sie am einfachsten wie folgt:

```
cp /boot/vmlinuz /nfsexport/ldc/boot/dxs
cp /boot/initrd /nfsexport/ldc/boot/initrd-dxs
```

Der Eintrag `ipappend` sorgt dafür, dass PXE die von DHCP ermittelte Konfiguration in die Kernel-Command Line schreibt. Diese enthält dann im Beispiel die Option `ip=192.168.1.101:192.168.1.1:192.168.1.254:255.255.255.0`. Den Inhalt dieser speziellen Zeile können Sie auf jedem beliebigen Linux-PC mit dem Kommando

```
cat /proc/cmdline
```

ansehen.

Da die frisch angelegte Verzeichnisstruktur noch nicht sehr viel enthält, passiert erst einmal nichts weiter. PXELinux kann noch keinen Kernel finden. Um den Test fortzuführen, kopieren Sie einfach einen Standard-Kernel und eine Standard-RAM-Disk an die passenden Stellen, im Beispiel bleibend nach `/nfsexport/ldc/boot`. Nach dem nächsten Neustart des Clients ändert sich das Verhalten des Clients.

Nun sieht der Client fast schon wie gewohnt aus: Ein Boot-Loader kümmert sich um den Kernel und die Initial RAM-Disk und sorgt für deren Start. Der Kernel entpackt sich wie gewohnt und die Initial-RAM-Disk führt eine Mini-Linux-Umgebung aus.

Der Prozess bricht jedoch ab, da das Mounten des Root-Dateisystems fehlschlägt, weil Kernel und RAM-Disk von einer klassischen Installation stammen und für den Start von Festplatte ausgelegt sind.

Sind Systemverwalter bis hierher gekommen, könnten sie in einem der nächsten Schritte die vom Entwickler Dirk von Suchodoletz als Diskless X-Stations (DXS) bezeichneten schlanken, plattenlosen Linux-PCs erproben. Zuvor wirft dieser Abschnitt noch einen Blick auf eine Alternative zu kommerziellen PXE-Boot-Proms, auf Open-Source-Etherboot.



### 16.3.5 Etherboot als Alternative

Die bisherige Beschreibung nahm an, dass ein Client für Ihre ersten Schritte in eine Diskless-Umgebung über ein kommerzielles PXE-Boot-Prom von Intel verfügte. Dieses ist leider nicht bei allen PCs der Fall. Clients kann man auch ohne Intels PXE über das Netz booten. Die Entwickler von Etherboot haben schon vor PXE eine hochflexible, freie Boot-Rom-Software geschrieben und hervorragend dokumentiert.

Installieren Sie bitte die Pakete `etherboot` und `mknbi` von der SuSE-DVD nach.

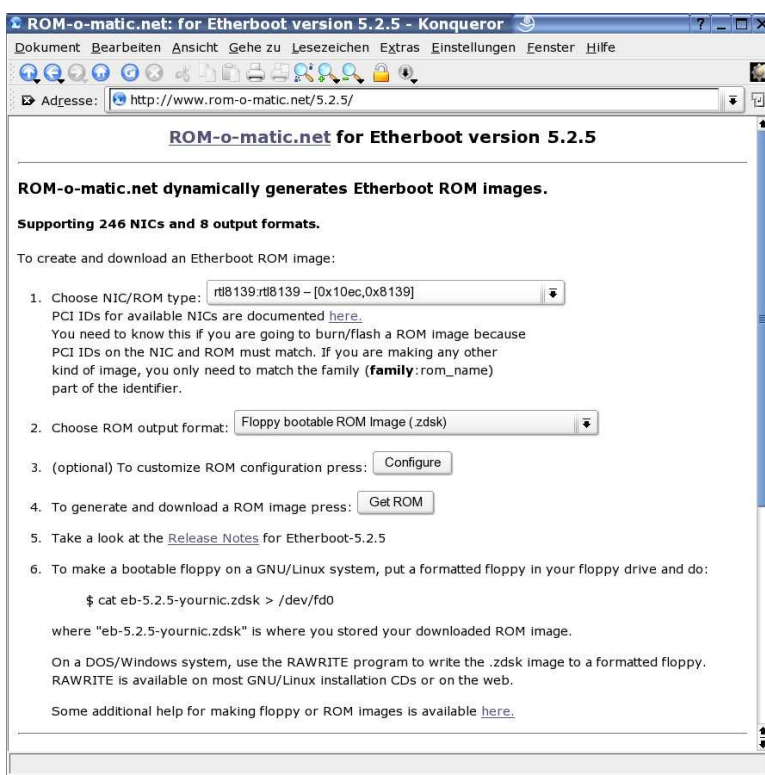


Abbildung 16.6: ROM-o-matic

Von `http://rom-o-matic.net` kann man ein freies, nach eigenen Anforderungen kompiliertes Boot-Rom-Image laden. Das Bild zeigt die Einstellungen für eine einfache Realtek-basierte 100 MBit-Netzwerkkarte mit dem RTL8139 Chip.

Wenn Sie über keinen EPROM-Programmierer verfügen oder das BIOS nicht modifizieren wollen, können Sie dort auch ein Image für Disketten erzeugen lassen. Dieses schreiben Sie mittels

```
dd if=eb-5.2.5-rtl8139.zdisk of=/dev/fd0
```

auf eine leere Diskette. Rufen Sie dabei das Kommando in dem Verzeichnis auf, in dem Sie das Image gespeichert haben. Sie müssen die Diskette nicht einmal formatieren, da obiger `dd`-Befehl das Boot-Rom-Image einfach in den Boot-Block schreibt. Nun stellen Sie in der Boot-Reihenfolge des Test-PCs den Start von Diskette als Erstes ein.

Anders als PXE kann Etherboot direkt etwas mit einem Linux-Kernel anfangen. Damit dieser nach dem Netzwerktransfer auf den Client startet, müssen Sie den Kernel vorher mit dem Programm *mkubi* bearbeiten (*taggen*), das bei SuSE mit auf den Datenträgern vorhanden ist.

Im Boot-Verzeichnis rufen Sie es so auf:

```
mkelf-linux --ip=rom -output=bootimg vmlinuz initrd
```

Dadurch erzeugen Sie eine einzige Datei aus Kernel und RAM-Disk. Der Kernel bekommt dabei einen von Etherboot verwendeten Header. Diese müssen Sie noch an eine Stelle kopieren, an der der TFTP-Server sie erwartet.

Dazu tragen Sie in Ihren DHCP-Server ein: `filename /nfsexport/ldc/boot/dx-simg` und starten ihn neu.

Wenn Sie den TFTP-Server wie im Abschnitt 16.4 konfiguriert haben, müssen Sie ihn nicht anpassen.

Nach einem Reboot startet Ihr Test-PC einen Kernel und führt anschließend ähnlich wie PXE eine RAM-Disk aus. In einem können Sie beide Boot-Verfahren nebeneinander benutzen, wenn Sie den DHCP-Server geeignet anpassen.

### 16.3.6 NFS als Netzwerkdateisystem

Die Diskless X-Stations basieren auf einem Root-Dateisystem, welches Sie mit NFS von einem Server aus einbinden. Dieser Server kann der PC sein, der bereits die Dienste DHCP und NFS für den initialen Teil der Boot-Prozedur zur Verfügung stellt. Kapitel 8 gibt ausführlich Auskunft. Für den Einsatz von Linux Diskless Clients empfiehlt sich der Einsatz Server-Dämons für das Network File System (*Kernel-NFSD*). Diesen Dienst konfigurieren Sie in der Datei `/etc/exports` durch geeignete Freigaben. Sie sieht für dieses Beispiel so aus:

```
/nfsexport/ldc 192.168.1.0/255.255.255.0\  
└ (ro,no_root_squash,async)
```

Durch diesen Eintrag geben Sie das Unterverzeichnis `/nfsexport/ldc` nur lesbar für alle PCs im Subnetz `192.168.1.0` mit der Netzwerkmaske `255.255.255.0` frei. Nach jeder Änderung der Konfiguration starten Sie den Dienst so wie DHCP und TFTP neu:

```
rcnfsserver restart
```

Um auszuprobieren, ob die Freigabe auch funktioniert, geben Sie auf dem frisch gestarteten NFS-Server das Kommando

```
mount -t nfs 192.168.1.1:/nfsexport/lldc /mnt
```

ein. Meldet der Befehl keine Fehler, so sehen Sie im Verzeichnis `/mnt` die angelegte Verzeichnisstruktur der Clients, die bisher nur ein Unterverzeichnis `boot` enthält. In diesem Verzeichnis sollten der Test-Kernel und die RAM-Disk und je nach verwendetem Boot-Rom die PXE-Linux-Konfiguration liegen. Geben Sie in diesem Verzeichnis den Befehl

```
touch boot/test
```

ein. Das Kommando bricht mit einer Fehlermeldung ab, da das Unterverzeichnis ausschließlich lesbar exportiert wurde. Das verhindert später, dass sich verschiedene Clients in die Quere kommen können oder böswillige Nutzer daran etwas ändern können. Das eingehängte Verzeichnis geben Sie anschließend mit

```
umount /mnt
```

wieder frei.

Mit diesen Voraussetzungen können Sie jetzt zwei konkrete Projekte testen:

Terminals à la LTSP wie im nächsten Abschnitt beschrieben dienen dazu, bescheiden ausgebaute, ggf. auch ältere PCs als Terminals von Boot-Servern zu booten. Workstations à la DXS hingegen sollten bis auf Laufwerke wie aktuelle Business-PCs ausgestattet werden, da sie als ebensolche verwendet werden können.

## 16.4 Terminals à la LTSP

Das LTSP-Projekt um Jim Mc Quillan ist zwar etwas jünger als andere Thin Client-Projekte, auch jünger als DXS, hat aber eine sehr starke Community, die die Lösung sehr liebevoll pflegt, erweitert und dokumentiert. Auch der Online-Support ist sehr nett.

Dieser Teil des Kapitel beschreibt, wie Sie plattenlose PCs von einem Boot-Server starten, um sie dann an Terminal-Servern oder Mainframes als Terminal zu betreiben. Solche Endgeräte sollten Anwender von fortgeschrittenen Linux-Benutzern einrichten lassen. Die Installationsarbeit setzt u. a. Grundkenntnisse in NFS (*Network File System*) voraus, wie sie das Kapitel 8 dieses Buchs vermittelt.

Zusätzliche Informationen in deutscher Sprache finden Sie u. a. als Bonuskapitel auf <http://www.linux-terminalserver.de/objects/documents/PDFs/lisp.pdf>.

### 16.4.1 Überblick

Im Laufe dieses Abschnitts lesen Sie Schritt für Schritt, wie Sie X-Terminals (*Clients*) einrichten. Sobald Sie die X-Terminals eingerichtet haben, starten Sie diese in folgenden Schritten:

1. Das Boot-Rom initialisiert die Netzwerkkarte.
2. Das Boot-Rom sendet eine Bootp-Anfrage an den DHCP-Server.
3. Der DHCP-Server beantwortet die Anfrage und teilt dem Client eine IP-Adresse zu. Außerdem teilt er dem Client die Lage und den Pfad des zu ladenden Linux-Kernels mit.
4. Der Client sendet eine TFTP-Anfrage an den Boot-Server. (TFTP: *Trivial File Transport Protocol*).
5. Der TFTP-Server beantwortet die Anfrage und sendet dem Client den Linux-Kernel.
6. Der Client bootet jetzt Linux. Der Linux-Kernel sendet eine Bootp-Anfrage an den Server.
7. Der Server beantwortet diese Anfrage. Mit diesen Daten konfiguriert der Linux-Kernel des Client sein Netzwerkinterface und setzt den Rechnernamen.
8. Der Linux-Client versucht, sein Root-File-System per NFS zu mounten.
9. Der NFS-Server exportiert das angeforderte Verzeichnis an den Client.
10. Der `init`-Prozess auf dem Client startet.
11. Auf dem Client startet der X-Server. Er sendet XDMCP-Anfragen an den Server (XDMCP: *X Display Manager Control Protocol*).
12. Der XDM-Server beantwortet diese Anfragen. Der Client zeigt den Begrüßungsbildschirm des XDM-Servers mit dem Login (XDM: *X Display Manager*).
13. Benutzer können sich auf dem Server einloggen. Ihnen steht jetzt Ihre X-Window-Oberfläche zur Verfügung, und sie können damit auf dem lokalen Linux-System oder auf anderen Linux-Rechnern arbeiten.

Aus dieser Oberfläche können Benutzer einen RDP-, ICA- oder AIP-Client starten, um auf einem Windows-Terminal-Server (s. oben) zu arbeiten.

Das Linux-Terminal-Server-Projekt hat die für das Client-System notwendige Software und einige Installationsskripte zu einem leicht nutzbaren Software-Pakte zusammengestellt.

### 16.4.2 Vorbereitungen für LTSP

Auf einem Linux-Server benötigt man für Terminals à la LTSP einen DHCP-Server, einen TFTP-Server, einen NFS-Server sowie einen XDM-Server. Wenn noch keine DHCP-Server und TFTP-Server eingerichtet sind, müssen Sie diese nachinstallieren.

Überprüfen Sie zunächst, ob folgende Pakete installiert sind:

- tftp bzw. atftp
- dhcp-server
- nfs-utils bzw. nfs-server
- perl-URI
- perl-libwww-perl

Dies geschieht mit dem Befehl:

```
rpm -q Paketname
```

Ein Beispiel für eine derartige Abfrage wäre:

```
rpm -q nfs-utils
```

Das Programm *rpm* gibt dann aus:

```
nfs-utils-1.0.6-113
```

Die letzte Ziffer nennt jeweils das installierte Release. Diese ändert sich, wenn ein Online-Update eine neue Version des Programm-Paketes bereitstellt und kann bei Ihnen anders lauten.

Installieren Sie fehlende Pakete nach.

Der nächste Schritt unterscheidet sich je nach der nfs-Installation:

Damit die Server-Programme automatisch starten, geben Sie nach dem Installieren des Pakets *nfs-utils* ein:

```
insserv nfslock
insserv nfsserver
insserv dhcpd
```

Falls Sie jedoch das Paket *nfs-server* installiert haben, müssen Sie nur die folgenden zwei Zeilen eingeben:

```
insserv nfsserver
insserv dhcpd
```

Falls Sie das Paket *atftp* installiert haben, geben Sie anschließend noch ein:

```
insserv atftpd
```

Falls sie das Paket *tftp* installiert haben, müssen Sie den Dienst *tftp* mit YaST unter Netzwerkdienste Netzwerkdienste (*inetd*) freischalten.

### 16.4.3 LTS-Software

Die weitere Software können Sie von der Web-Site des Linux-Terminal-Server-Projekts beziehen:

Laden Sie bitte von der Homepage des Linux-Terminal-Server-Projekts (<http://www.ltsp.org/>) das Paket

*ltsp-utils-0.10-0.noarch.rpm* herunter.

Dieses Paket können Sie mittels

```
rpm -i ltsp-utils-0.10-0.noarch.rpm
```

installieren. Es beinhaltet ein Installationsprogramm, welches Ihnen das Laden weiterer Pakete und das Konfigurieren des Systems erleichtert.

Mit *root*-Rechten können Sie nach der Installation des Pakets das textbasiertes Administrationsprogramm *ltspadmin* starten:

```
su -
ltspadmin
```

Dieses führt Sie dann durch die Installation.



Abbildung 16.7: LTSP Administration Utility

Wichtig ist hier der erste Punkt, Install/Update LTSP Packages. Die Konfiguration des Installers benötigen Sie nur, wenn Ihr Internet-Zugriff über einen Proxy-Server läuft.

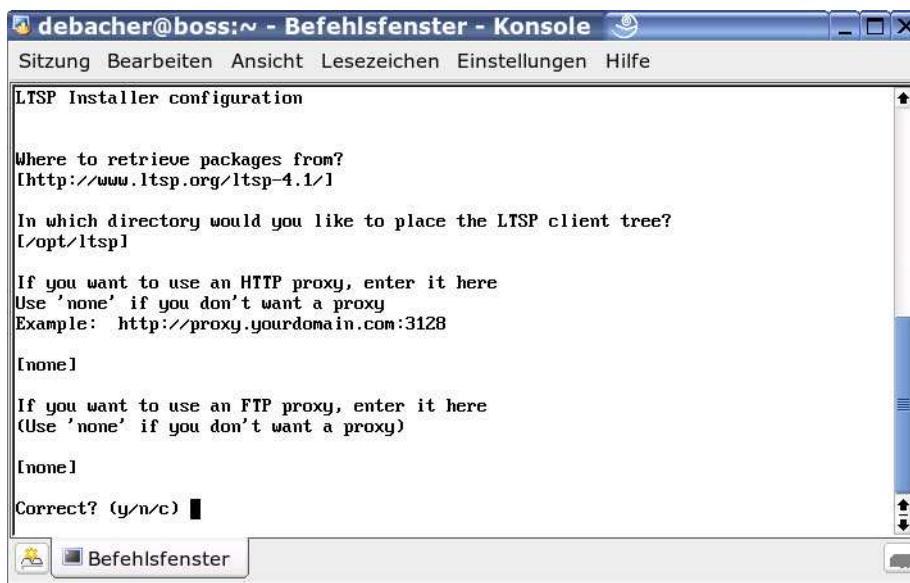


Abbildung 16.8: LTSP Installer configuration

Starten Sie also die Installation der Pakete. Der Installer nimmt nun Kontakt zum LTSP-Server auf und lädt die Liste der vorhandenen Programmpakete. Zur Zeit sind insgesamt 61 Pakete vorhanden, das Laden der Liste dauert etwas.

Wenn die Liste vollständig geladen ist, lässt Sie das Installationsprogramm Pakete auswählen.



Abbildung 16.9: LTSP, Komponenten-Auswahl

Wenn Sie alle Pakete auswählen, sind Sie bei der Installation flexibler.

Ein Druck auf die Taste  startet dann das Laden der Pakete im Umfang von etwa 100 MByte vom LTSP-Server. Dies benötigt etwas Zeit.

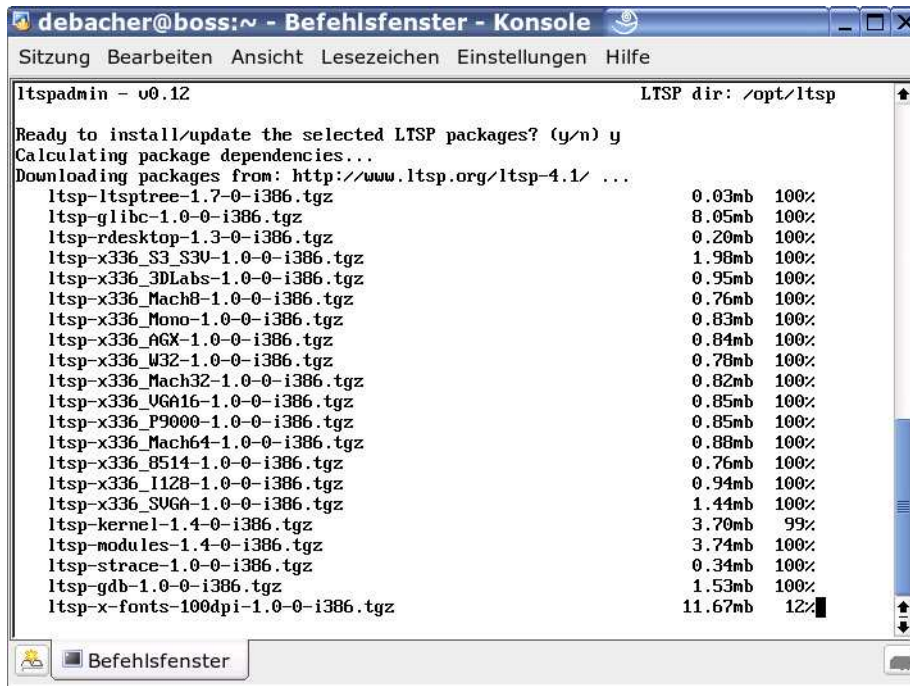


Abbildung 16.10: LTSP, Download der Pakete

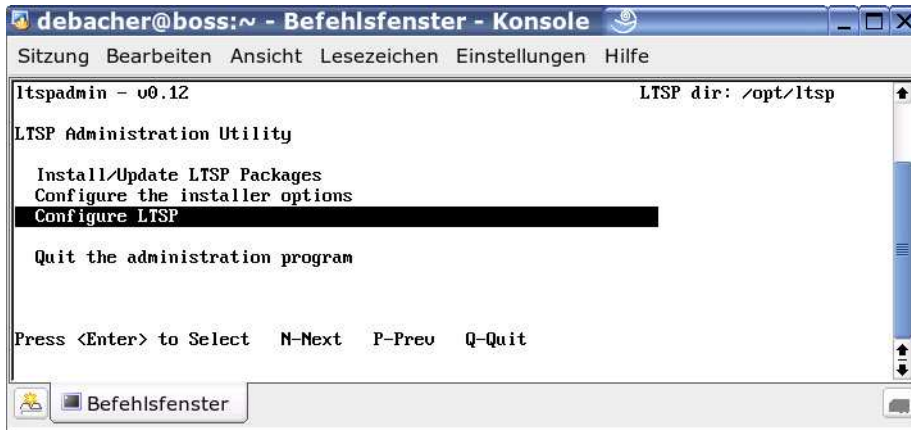
Nach dem Laden installiert das Programm die Pakete sogleich und speichert alle Dateien unterhalb von `/opt/ltsp`. Sie müssen sich also nicht zu sorgen, dass dabei irgendwelche Dateien Ihres Servers überschrieben würden.

#### 16.4.4 Konfiguration der LTSP-Software

Nach dem Laden der Pakete konfiguriert das Skript den Server und verändert dessen installiertes System.

Das *LTSP Administration Utility* stellt auch hierfür einen Menüpunkt zur Verfügung.



Abbildung 16.11: LTSP, *Configure LTSP*

Wenn Sie den Menüpunkt zur Konfiguration von LTSP auswählen, untersucht das Programm die aktuelle Systemkonfiguration und bietet Ihnen für jeden Konfigurationsschritt einen eigenen Menüpunkt an.



Abbildung 16.12: LTSP-Konfiguration

Sie können hier im einfachsten Fall die Punkte der Reihe nach abarbeiten. Lediglich den Punkt 8 `Create /etc/hosts entries` sollten Sie auslassen, wenn Sie bereits einen Name-Server konfiguriert haben.

Ansonsten stellt das Konfigurationsprogramm selbst fest, ob etwas zu tun ist oder nicht. Bei Problemen oder Konflikten informiert Sie das Programm.



Abbildung 16.13: LTSP, DHCP-Konfiguration

Im vorliegenden Fall, der Konfiguration von DHCP, überschreibt das Programm eine vorhandene Konfiguration nicht einfach, sondern erstellt eine Musterdatei, deren Inhalt Sie dann selbst in die Konfigurationsdatei übertragen müssen.

#### 16.4.5 Nachbesserungen

Nachdem das Konfigurationsprogramm von LTSP seine Arbeit abgeschlossen hat, sollten Sie die wichtigsten Konfigurationsdateien prüfen.

Kontrollieren Sie auf alle Fälle die Datei `/etc/exports`:

```

/home/ www.linuxbu.ch(ro) *.lokales-netz.de(rw)
/media/cdrom/ *(ro,root_squash,sync)

### LTSP-begin ###
#
# The lines between 'LTSP-begin' and 'LTSP-end' were added
# on: Fri Jan 14 19:55:06 2005, by the ltspcfg configuration too
#.
# For more information, visit the LTSP homepage
# at http://www.LTSP.org
#

/opt/ltsp          192.168.1.0/
255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles 192.168.1.0/
255.255.255.0(rw,no_root_squash,async)

### LTSP-end ###

```

An dieser Datei können Sie sehen, dass das Konfigurationsprogramm seine Einstellungen an die vorhandenen Einträge anhängt und mit Kommentarzeilen deutlich kennzeichnet.

Wichtig ist hier vor allem, dass die angegebenen IP-Adressen zu Ihrer Netzwerkkonfiguration passen, damit Ihre Client-Systeme auf die Freigaben zugreifen dürfen.

In der Datei `/etc/dhcpd.conf` passen Sie LTSP an Ihre eigenen Gegebenheiten wie Netzadressen und Hardware-Adressen spezieller Netzwerkkarten an.

Hier folgt ein Muster dieser `/etc/dhcpd.conf` :

```
option domain-name "lokales-netz.de";
option domain-name-servers 192.168.1.2;
option routers 192.168.1.2;
option netbios-name-servers 192.168.1.2;
ddns-update-style none;
default-lease-time 14400;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.20 192.168.1.200;
    default-lease-time 14400;
    max-lease-time 172800;
}

option option-128 code 128 = string;
option option-129 code 129 = text;

option root-path          "192.168.1.2:/opt/ltsp/i386";

###
### If you want to use static IP address for your workstations,
### then un-comment the following section and modify to suit
### your network. Then, duplicate this section for each
### workstation that needs a static IP address.
###
### host ws001 {
###     <----- Fix this hostname
###     hardware ethernet 00:11:22:33:44:55; <-- Fix this MAC addr
###     fixed-address 192.168.0.1; <-- Fix this IP addr
###     filename      "/tftpboot/lts/vmlinuz-2.4.26-ltsp-1";
### }

###
### If you want to use a dynamic pool of addresses, then
### uncomment the following lines and modify to match
### your network.
###
### subnet 192.168.0.0 netmask 255.255.255.0 {
```

```

###      range dynamic-bootp 192.168.0.1 192.168.0.253;
###    }
###
#
# If you need to pass parameters on the kernel command line,
# you can do it with option-129. In order for Etherboot to
# look at option-129, you MUST have option-128 set to a
# specific value. The value is a special Etherboot signature
# of 'e4:45:74:68:00:00'.
#
# Add these two lines to the host entry that needs kernel
# parameters
#
#      option option-
128    e4:45:74:68:00:00;      # NOT a mac address
#      option option-129      "NIC=ne IO=0x300";
#
host ws001 {
    hardware ethernet    00:0c:29:ef:98:c4;
    fixed-address        192.168.1.31;
    filename              "/lts/2.4.26-ltsp-2/pxelinux.0";
}

host ws002 {
    hardware ethernet    00:90:f5:1a:a8:7c;
    fixed-address        192.168.1.30;
    filename              "/lts/vmlinuz-2.4.26-ltsp-2";
}

```

Die obige `host`-Deklaration zeigt Beispiele für zwei verschiedene Rechner. Abhängig von der Adresse ihrer Netzwerkkarte weist ihnen der DHCP-Server eine IP-Adresse zu. Die zugehörigen Kernel liegen im Verzeichnis `/tftpboot/lts`. Das `/tftpboot` muss man in der obigen Konfiguration weglassen, da der TFTP-Server im sicheren Modus läuft und ein `chroot` nach `/tftpboot` macht. Außerdem übergibt der DHCP-Server den beiden Rechner ihre Rechnernamen. Die Einträge in der Datei `/opt/ltsp/i386/etc/lts.conf` für die Workstation-Namen müssen mit den hier gesetzten Namen übereinstimmen. Rechner, die mit der Standardkonfiguration der `lts.conf` funktionieren, muss man hier nicht eintragen.

Bitte passen Sie danach die Datei `/opt/ltsp/i386/etc/lts.conf` an. Die Parameter sollten selbsterklärend sein. Die Datei `/opt/ltsp/i386/etc/lts.conf` ist die zentrale Konfigurationsdatei für die LTSP-Clients.

Die Datei `/opt/ltsp/i386/etc/ltsp.conf.readme` erläutert Details dieser Datei.

```
#
# Config file for the Linux Terminal Server Project
# (www.ltsp.org)
#

[Default]
SERVER          = 192.168.1.2
XSERVER         = auto
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE  = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
XkbKeyCodes     = "xfree86"
XkbLayout       = "de"
XkbModel        = "pc104"
XkbRules        = "xfree86"
XkbVariant      = "nodeadkeys"

USE_XFS         = N
SCREEN_01       = startx
SCREEN_02       = shell

[ws001]
XSERVER         = auto

[ws002]
XSERVER         = radeon
```

Um alle Änderungen wirksam zu machen, geben Sie nun ein:

```
init 1
```

und nach einem erneuten Einloggen als *root*:

```
init 5
```

#### 16.4.6 Erste Tests

Images für ein Boot-Rom oder eine Boot-fähige Diskette können Sie aus dem Internet von der Adresse <http://www.rom-o-matic.net/>

laden. Falls Sie zum Testen nur eine Boot-fähige Diskette erstellen wollen, müssen Sie das von dort bezogene Image mit dem folgenden Befehl auf eine Diskette schreiben:

```
cat imagename > /dev/fd0
```

Falls das X-Window-System des Clients nicht starten sollte, versuchen Sie es mit folgender Einstellung in der Datei `/opt/lts/i386/etc/lts.conf`:

```
XSERVER = vesa
```

Wenn Sie den Client-PC mit einem Boot-Rom starten wollen, sollten Sie

- Boot-Roms mit den Treibern des Etherboot-Paketes über das Internet bestellen oder
- sich einen Eprom-Brenner zulegen und die Boot-Roms selbst herstellen oder
- Netzwerkkarten oder Mutterbretter mit PXE-Boot-Roms verwenden.

Im Internet finden Sie Kommentare und Erfahrungen mit Ne2000-kompatiblen Karten, Ne2000Pci-Karten und Karten mit dem rtl8139-Chipsatz mit dem 32Kb (256 Kbit)-Baustein 27256. Wenn man auf einer Netzwerkkarte die Boot-Möglichkeit vom Boot-Rom aktivieren will, versagen die Programme mancher Hersteller der Netzwerkkarten. Hier helfen oft Programme der Hersteller der Chipsätze der Netzwerkkarten.

**Tipp:** Die Netzwerkkartenhersteller geben die Größe des Boot-Roms in KByte an, die Eprom-Hersteller in KBit. Für ein 32 KByte-Bootprom benötigen Sie also ein 256 KBit-Eprom.

#### 16.4.7 Testen mit VMware

Am einfachsten fällt Ihnen das Testen, wenn auf Ihrem Server VMware zur Verfügung steht (siehe Abschnitt 18.2). Die aktuelle Version erlaubt auch das Booten per PXE.

Legen Sie also einfach einen neuen virtuellen Rechner an, und booten Sie diesen ohne ihm eine Diskette oder eine CD/DVD anzubieten. Dann startet dieser virtuelle Rechner seinen PXE-Code.



Für jeden weiteren LTSP-Client müssen Sie immer nur die Datei `dhcpd-conf` um die MAC-Adresse des jeweiligen Rechners erweitern sowie gelegentlich auch die `ltsp.conf` um den passenden Grafiktreiber für die eingebaute Grafikkarte.

Bei den Tests der Redaktion funktioniert die automatische Konfiguration der Grafikkarte nicht immer problemlos.

## 16.5 Flash-Rom-Workstations mit DXS

Der Entwickler Dirk von Suchodoletz will mit dem vom ihm Diskless X-Station (DXS) genannten Projekt die für festplattenbasierte PCs bestimmte SuSE 9.2-Linux-Distribution für Diskless Clients aufbereiten. Da es etwas schwieriger ist, aus dem bootenden Kernel mit geladener RAM-Disk eine lauffähige Linux-Workstation zu machen, stellt der Autor des Projekts einfach zu bedienende Skripten bereit, die die eingesetzte Distribution benutzen. Es fügt ihr lediglich einige Skripten und Konfigurationsdateien hinzu.

Um dies zu testen, laden Sie das aktuelle Paket `ld-v3.2.tgz` von der Webseite des Projekts <http://www.linux-terminalserver.de> und entpacken dieses beispielsweise in das Heimatverzeichnis des Administrators. Das Verzeichnis enthält dann die Dateien:

```
d2d1
.config
README
dxs-inst
y2pmdxs
dxs-specific
dxs-specific/bin
dxs-specific/bin/runvmware
dxs-specific/etc
dxs-specific/etc/sysconfig
dxs-specific/etc/sysconfig/machine-setup.default
dxs-specific/etc/init.d
dxs-specific/etc/init.d/boot.hwsetup
dxs-specific/etc/init.d/boot.servconf
dxs-specific/etc/init.d/halt.ld
dxs-specific/etc/init.d/startgui
dxs-specific/etc/init.d/dhclient-script
dxs-specific/etc/init.d/boot.ld
dxs-specific/etc/init.d/boot.usb
dxs-specific/lib
dxs-specific/lib/klibc
dxs-specific/lib/klibc/bin
dxs-specific/lib/klibc/bin/nfsmount
dxs-specific/lib/README
dxs-specific/lib/mkinitrd
```



```

dxs-specific/lib/mkinitrd/vendor_init.sh
dxs-spedxs-specific/boot
dxs-specific/boot/pxelinux.0
dxs-specific/boot/pxelinux.cfg
dhcpd.example
exclude.local

```

Das Installationsskript für das Client-Dateisystem `dxs-inst` erzeugt in einem leeren Verzeichnis eine komplette Verzeichnisstruktur, die später per NFS an die Clients exportiert wird. Dieses Skript funktioniert nur auf Intel 32 Bit-Servern mit SuSE 9.2-Distribution. Es setzt dabei auf dem SuSE-Tool `y2pms` auf, dessen Vorhandensein es beim Start überprüft. Mit diesem Tools können Sie wie von YaST gewohnt Pakete auswählen. Die Datei `y2pmdxs` enthält ein Hilfsskript für `dxs-inst`.

Genereller arbeitet das Installationsskript `d2d1`. Es clont ein bestehendes SuSE9.X-System in ein Unterverzeichnis des DXS-Servers. Hierzu muss lediglich `rsync` installiert sein.

Das Unterverzeichnis `dxs-specific` enthält Skripten für den festplattenlosen Betrieb. Sie kümmern sich um die angepasste Initial RAM-Disk, die Konfiguration der Dienste auf dem Client, dessen Hardware und grafische Oberfläche. Enthalten sind die notwendigen Dateien für PXE-Boot, sodass Sie dieses Paket nicht installieren müssen. Weiterhin liegt auf der obersten Verzeichnisebene eine Beispielkonfigurationsdatei für den DHCP-Server.

Während der Installation legen `dxs-inst` oder `d2d1` die Dateien `installation.log` und `.config` an. Ersteres nimmt Meldungen auf, die während der Installation anfallen und in der Konsole stören oder verwirren könnten. Die versteckte Datei `.config` speichert Benutzereingaben, sodass Sie bei erneutem Aufruf der Installationsskripten auf vorher eingegebene Daten zurückgreifen können.

```
rpm -root=/nfsexport/ldc -ivh paket.rpm
```

### 16.5.1 Installation aus RPMs

Das Skript `dxs-inst` erlaubt Ihnen eine Installation direkt aus demselben Installationsmedium, aus dem auch der Server, auf dem `dxs-inst` läuft, seine RPMs bezogen hat. Wenn Sie das Installationsmedium ändern wollten, müssten Sie YaST aufrufen und das Untermenü *Installationsquelle wechseln* anwählen.

Vor der eigentlichen Installation stellt `dxs-inst` einige Fragen zur Auswahl des Zielverzeichnisses, des zu verwendenden Subnetzes, für einen Vorschlag der `/etc/exports`, des Client-Root-Passwortes und der zu berücksichtigenden Paketgruppen. Da diese Fragen mit Default-Einträgen vorbelegt sind, führt meist die -Taste zu einem verwendbaren Ergebnis. Für mehrere Parallelinstallationen, beispielsweise mit verschiedenen SuSE-Distributionen, müssen Sie mindestens ver-

schiedene Zielverzeichnisse wählen. Nachfolgend sind Auszüge des Skriptablaufs dargestellt:

```
s2:~/ld-v3.2 # ./dxs-inst
Welcome to version ldc-v3.2 0.8i of dxs-
inst! This installation method
is heavily dependent on the servers software and architecture!!
It works for the most recent SuSE linux versions 9.1 and 9.2.
For older versions use previous script versions or the other
installationmethod (d2d1 - clone installed system).
You will need an installed version of 'y2pmsh' and 'rsync'
- otherwise the procedure would fail

Checking for y2pmsh: y2pmsh-2.10.1-2
This script requires a valid installation source set with
servers YaST - otherwise no package selections could be made!!
See installation.log for errors ...

Checking for rsync: rsync-2.6.3pre1-2

Prechecks passed!!
Please answer the following questions: (Enter takes defaults)

Which dxs root path should be used?
* [ /nfsroot/dxs ] /nfsexport/ldc
Which network do you want to use for DXS? (A.B.C.0)
* [ 192.168.2.0 ] 192.168.1.0
Which netmask should be used? (255.B.C.0)
* [ ] 255.255.255.0
Using 192.168.1.0/255.255.255.0 !
What is your servers IP for NFS, DHCP and TFTP?
* [ ] 192.168.1.1
Where automount home directories from? (A.B.C.D:/home-dir)
* [ 192.168.1.1:/home ]
[ Fortsetzung folgt ... ]
```

Zuerst legen Sie das Ziel-Unterverzeichnis der Installation fest. Anschließend folgt die Netzwerkkonfiguration. Das Skript erkennt die Installationsquelle des Servers und trägt diese ein, wenn Sie nicht mit YaST eine andere Quelle wählen.

```
The following source is used for intallation:
0: [x] SUSE LINUX Version 9.2 (dvd:///;devices=/dev/hdc)
Which SuSE online update source should be used?
* [ http://ftp.gwdg.de/pub/suse ]
Collecting available package selections from source media.
Please wait.
Please select package groups for installation!
Add selection Basis-Devel? [yN]
```

```

[ ... more selections ... ]
Add selection Minimal? [yN]
Add selection Multimedia? [yN]
Add selection Network? [yN]
Add selection Office? [yN]
Add selection SuSE-Documentation? [yN]
Add selection Tcl-Development? [yN]
Add selection default? [yN]
No package selection accepted. Using 'Minimal' as default.
Nacheinander werden Ihnen Paketselektionen angeboten. Diese
können Sie mit y selektieren. Wenn Sie keine ausgewählt haben,
nimmt das dxs-inst eine lauffähige Minimalinstallation vor.
Which debug level should be used?
* [ 2 ]
Which default init you would like to use?
* [ 5 ]

This script sets up the basic filesystem structure for diskless
X-stations now. Please enable the NFS-Share /nfsexport/ldc in
your servers /etc/exports file, i.e.:
/nfsexport/ldc          192.168.1.0/
255.255.255.0(ro,no_root_squash,async)
/tmp/dxs                192.168.1.0/
255.255.255.0(rw,no_root_squash,async)

... and (re)start your nfs server.
Which ethernet card modules should be included in initrd?
* [ e1000 e100 3c59x pcnet32 ] e100 8139too

Welcome to the YaST Package Manager!
This tool is meant for debugging purpose only.

root dir set to /nfsexport/ldc
Known sources:
0: [x] SUSE LINUX Version 9.2 (dvd:///;devices=/dev/hdc)
debug disabled
initializing installation sources ...
[ ... ]
fetching YaST-samba-server (94.5 kB) ..... ok
installing [100%] YaST-samba-server-2.10. (noarch) ..... ok
reading RPM database ..... ok
330 packages installed

```

Anschließend fragt das Skript `dxs-inst` nach dem Run- und Debug-Level der Clients und schlägt Einträge für `/etc/exports` vor, wenn Sie dort noch nichts hineingeschrieben haben. Dann sollen Sie Kernel-Module für Netzwerkkarten nennen, die in Ihren Clients installiert sind. Verwenden Sie Clients mit Intels

EtherExpressPro100 oder Realtek-basierte 100 Mbit-Karten, würden Sie `e100` und `8139too` angeben. Dann beginnt die eigentliche Installation der Pakete aus der angegebenen Quelle. Die Installation selbst dauert je nach Paketumfang und der Geschwindigkeit von Ziel und Quelle mindestens einige Minuten. Die Ausgabe des Paketmanagers zeigt den Installationsfortschritt für jedes Paket durch Punkte. Der Gesamtfortschritt steht in eckigen Klammern vor jedem Paket. Wenn eine Netzwerkverbindung existiert, führt `dxs-inst` danach ein automatisches Online-Update durch.

```

Calling onlineupdate for version: 9.2
Source: http://ftp.gwdg.de/pub/suse
[ ... ]
building file list ... done
bin/
bin/runvmware
boot/
boot/pxelinux.0
boot/pxelinux.cfg/
[ ... ]
generating common host key for all diskless machines
Generating public/private rsa1 key pair.
Your identification has been saved in /nfsexport/ldc/etc/ssh/
ssh_host_key.
Your public key has been saved in /nfsexport/ldc/etc/ssh/
ssh_host_key.pub.
[ ... ]
creating new initial ramdisk 3c59x pcnet32
mknod: `/nfsexport/ldc/dev/null': File exists
Module list:      3c59x pcnet32 af_packet
Kernel version:  2.6.8-24-default (i386)
Kernel image:    /boot/vmlinuz-2.6.8-24-default
Initrd image:    /boot/initrd-2.6.8-24-default
Using /lib/mkinitrd/vendor_init.sh as custom script
Shared libs:     lib/ld-2.3.3.so lib/libacl.so.1.1.0 lib/
libattr.so.1.1.0 lib/libc.so.6 lib/libselinux.so.1
Modules:         kernel/drivers/net/3c59x.ko kernel/drivers/net/
└─ mii.ko kernel/drivers/net/pcnet32.ko kernel/net/packet/
└─ af_packet.ko
Bootsplash:      SuSE (1024x768)
5651 blocks

You may now have to update your boot loader configuration.
./dxe-inst: line 419: mkelf-linux: command not found
mkelf/mknbi is needed only for etherboot enabled
network adaptors
writing /nfsexport/ldc/boot/pxelinux.cfg/default
writing ISC dhcpd configuration example
└─ (/etc/dhcpd.conf.example)

```

Zum Schluss erzeugt `dxs-inst` ein Schlüsselpaar für den Secure-Shell-Services. Ein Kernel-Image für Etherboot generiert das Skript nur, wenn das RPM `mknbi` installiert ist. Dieses benötigen Sie nicht, wenn Sie PXE zum Booten verwenden. Dann können Sie auch die Fehlermeldung ignorieren. Innerhalb der neu erstellten Verzeichnisstruktur wird das SuSE-Konfigurationskript `SuSEconfig` mittels `chroot-shell` aufgerufen:

```
chroot /nfsexport/ldc SuSEconfig
```

Dieses konfiguriert einige Basisdateien und definiert u. a. für X.11 die Fonts.

Ob die Installation korrekt erfolgt ist, können Sie durch Eingabe von `chroot` in das Installationsverzeichnis selbst überprüfen. Mit RPM können Sie innerhalb der `chroot`-Umgebung oder extern durch Angabe des `Root`-Verzeichnisses abfragen, ob bestimmte Programme und Dienste installiert wurden.

Es wird jedoch nicht jede Form der Abfrage funktionieren, da einige Dienste einen echt laufenden PC benötigen. Sie greifen z. B. direkt auf `/proc` zu und funktionieren nicht, wenn es in der `Chroot`-Umgebung nicht gemountet ist. Auch startet beispielsweise `X` nicht, wenn `/dev` leer ist. Diese Verzeichnisse entsprechend zu bevölkern passiert erst beim Start des Clients.

Ebenso ist zu bedenken, dass man zwar auf den Diskless Clients die installierten RPMs abfragen kann, eine Nachinstallation jedoch nicht erlaubt ist. Die Nutzung von `chroot` erlaubt Ihnen, einfach YaST innerhalb des Client-Dateisystems aufzurufen. Damit können Sie dann wie gewohnt Parameter einstellen.

Generell können Sie Pakete auch immer manuell unabhängig von der Erstinstallation und unabhängig von der Paketauswahl des Basisskriptes nachträglich einfügen. Ein Paket installieren Sie so nach:

```
rpm -root=/nfsexport/ldc -ivh paket.rpm
```

Im Unterschied zum gewohnten Benutzen von `rpm` ist hier ein Parameter für das `Root`-Verzeichnis der Installation angegeben. So fügen Sie das Paket nicht dem `Server`-Dateisystem sondern dem `Client`-Dateisystem im Unterverzeichnis `/nfsexport/ldc` hinzu.

Hierfür muss ein Client entsprechend herausgestellt, d. h. beim NFS-Export und dem Bau der Initial-RAM-Disk angepasst werden.

### 16.5.2 Installation durch Klonen

Wenn sich RPM wegen abweichender Rechnerarchitektur oder abweichender Distribution nicht anbietet oder Sie einen PC bereits vorbildlich mit SuSE Linux eingerichtet haben und diese Installation für mehrere Diskless Clients verwenden wollen, möchten Sie SuSE Linux jetzt vielleicht nicht komplett neu installieren.

Mit dem Programm *rsync* können Sie stattdessen Dateien und Verzeichnissen auf Linux- und Windows-Systemen sowohl auf dem lokalen Dateisystem als auch netzwerktransparent synchronisieren. Beim Kopieren von Daten über das Netz kann es die Übertragung automatisch mit der Secure Shell absichern.

Damit Sie keine lange Befehlskette zum Klonen eines bereits installierten Systems eingeben müssen, können Sie mit *d2d1* das Synchronisation weitgehend automatisieren. Nachfolgend sehen Sie Ausschnitte des Skriptes, das nur in einigen Teilen von *dxs-inst* abweicht.

```
linux:/tmp/lid-v3.2 # ./d2d1

Welcome to version lid-v3.2 0.8i of d2d1!
Checking for rsync: rsync-2.6.3pre1-2
See installation.log for errors ...

Checking for rsync: rsync-2.6.3pre1-2

Prechecks passed!!
Please answer the following questions: (Enter takes defaults)
Reading settings from .config file!!

Which dxs root path should be used?
* [ /nfsexport/lidc ]
[ ... ]
Which source should be used for rsync? format [server:]/path
  ↓ (without trailing / or *)
* [ / ] 192.168.75.128:/
[ ... ]
Wait for root password dialog from the rsync source server!
The authenticity of host '192.168.75.128 (192.168.75.128)' can't
be established.
RSA key fingerprint is 69:eb:07:3d:2c:6b:b2:55:33:3d:43:47:
  ↓ 58:81:bc:15.
Are you sure you want to continue connecting (yes/no)? yes
[ ... ]
Warning: Permanently added '192.168.75.128' (RSA) to the list of
known hosts.
Password:
receiving file list ...
[ ... ]
```

Die erste Abfrage bestimmt das Zielverzeichnis, das zukünftige Root-Verzeichnis der Clients. Die Fragen zur IP-Konfiguration sind identisch bis zur Abfrage der Installationsquelle, die kopiert werden soll. Wenn Sie einfach nur das Dateisystem des lokalen Endgeräts kopieren wollen, weil Sie erst einmal auf einer klassischen Workstation eine Testinstallation einrichten, genügt die Angabe des voreingestellten Pfades `/`.

**Tipp:** *rsync* muss sowohl auf der Quell- als auch auf dem Ziel-PC installiert sein. Das können Sie auf dem jeweiligen System mit `rpm -q rsync` nachprüfen.

Andernfalls geben Sie vor dem Pfad den Rechnernamen oder die IP-Nummer des PCs an, der die Daten liefern soll. *rsync* fragt Sie direkt aus dem Skript heraus nach dem Administrator-Passwort des Quellrechners (siehe Beispiel-Listing). Sie kommen nicht umhin, einen Administrator-Account mit weitgehenden Rechten einzusetzen, da Sie sonst einige Dateien vom Quellsystem nicht lesen dürften.

### 16.5.3 Clients via DHCP konfigurieren

DHCP erlaubt, verschiedene Clients individuell einzurichten.

Nach der ersten Installation sollten Sie die DHCP-Konfigurationsdatei erweitern. Ein Beispiel `dhcpd.example` befindet sich im Installationsverzeichnis und kann als Vorlage dienen.

Dabei haben die verschiedenen Variablen folgende Bedeutung:

Viele der klassischen DHCP-Variablen kennen Sie vielleicht bereits. Die Tabelle gibt einen Überblick zu den wichtigsten:

Variablenname	Beispielintrag	Beschreibung
language	de_DE.UTF-8 de-latin1-nodeadkeys	Setzt im ersten Teil die Konsolensprache und im zweiten Teil die Tastaturbelegung in der Konsole und unter X.11.
start-x	kde	Bestimmt, wie die grafische Oberfläche gestartet werden soll, und für den Kioskmode, mit welchem Window-Manager. Werte: <code>direct</code> , <code>indirect</code> , <code>broadcast</code> , <code>kde</code> , <code>gnome</code> ,...
start-snmp	no	Bestimmt, ob der <code>snmpd</code> zur Fernüberwachung des Clients gestartet wird.
start-sshd	yes	Legt fest, ob ein SSH-Server auf dem Client laufen soll. Bei <code>yes</code> ist es dann möglich, sich remote auf dem PC anzumelden.
start-xdmcp	kdm	Bestimmt, ob und welcher Displaymanager gestartet wird. Werte: <code>no</code> , <code>yes</code> , <code>kdm</code> , <code>gdm</code> , <code>xdm</code>
start-cron	yes	Definiert, ob der Scheduler für zeitgesteuerte Abläufe gestartet wird.
Crontab-entries	0 22 * * * root shutdown -h now	Hier können Sie Einträge definieren, die an die Konfigurationsdatei <code>/etc/crontab</code> angefügt werden. Das Beispiel fährt der PC täglich um 22 Uhr herunter.

**Tabelle 16.1:** dhcp-Variablen für Diskless Clients

Variablenname	Beispieleintrag	Beschreibung
start-rwhod	no	Startet den Dienst für LAN-weites <i>who</i> oder nicht.
Netbios-workgroup	tux-net	Setzt den Namen einer Windows-Arbeitsgruppe. Das ist sinnvoll, wenn auf dem PC Samba eingesetzt ist.
Vmware	yes	Lädt die VMware-Kernel-Module. Macht nur Sinn, wenn VMware installiert ist und auf den Clients benutzt werden soll.
hw-mouse	ps/2 psaux	Umgeht die Hardware-Auto-Erkennung. Setzt den Maustyp und die Schnittstelle.
hw-graphic	radeon 24	Umgeht die Hardware-Auto-Erkennung. Setzt den X.11-Treiber (hier für ATI-Radeon-Karten) und die Standardfarbtiefe.
hw-monitor	30-64kHz 50-100Hz 1024x768	Umgeht die Hardware-Auto-Erkennung. Setzt die Monitorhorizontal- und Vertikalfrequenz und die maximale Auflösung unter X.11.

**Tabelle 16.1: dhcp-Variablen für Diskless Clients (Forts.)**

Der Autor des Projekts arbeitet an alternativen Konfigurationskonzepten wie einer Einrichtung per LDAP.

#### 16.5.4 Start der Diskless X-Stations

Nachdem die Installation erfolgreich abgeschlossen wurde, können Sie versuchen, Ihre Clients zu booten. Die bisherigen DHCP-Einstellungen sollten bereits genügen, um Clients weitgehend zu starten. Die Startausgaben ähneln weitgehend denen, die Sie vom Start Ihrer SuSE-Workstation gewohnt sind.

In der Initial RAM-Disk startet das `kinit.sh` das DXS-spezifische Skript `vendor_init.sh`. Beide Skripten finden Sie unterhalb von `/lib/mkinitrd` im Client-Dateisystem. Um zu Beginn des Startvorganges etwas zu ändern, müssen Sie die Datei `vendor_init.sh` bearbeiten und nach jeder Änderung die Initial RAM-Disk neu generieren:

```
chroot /nfsroot/ldc mkinitramfs -b /boot -V \
    /lib/mkinitrd/vendor_init.sh \
    -m "$modulelist af_packet" -s 1024x768
```



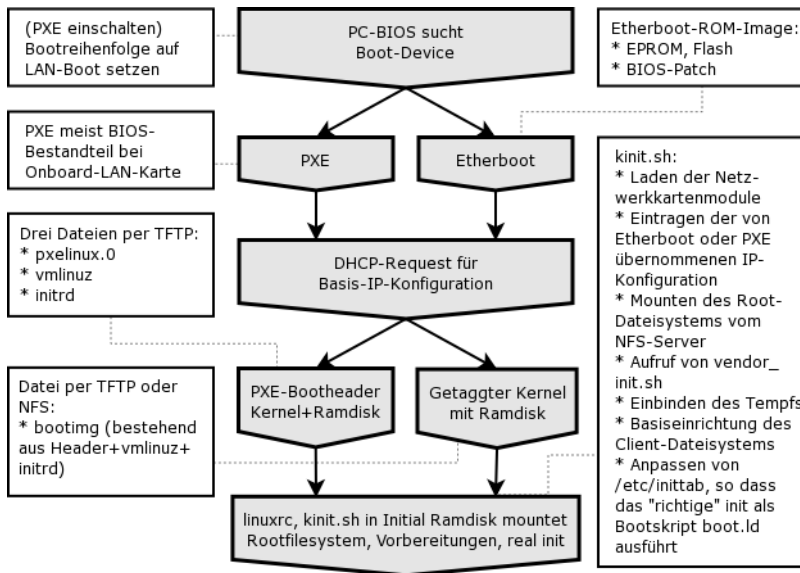


Abbildung 16.16: Ablauf des Boot-Vorganges der DXS bis zum Ausführen der Initial RAM-Disk

Im zweiten Teil des Starts läuft das DXS-Skript `boot.ld` anstelle des klassischen Startskriptes `boot`. Es beschafft die weiteren Konfigurationsdaten per DHCP und lässt einige Schritte, wie den Check festplattenbasierter Dateisysteme weg. Im initialen Boot-Ablauf, vertreten durch Links im Verzeichnis `/etc/init.d/boot.d` kommen noch drei DXS-Skripten hinzu.

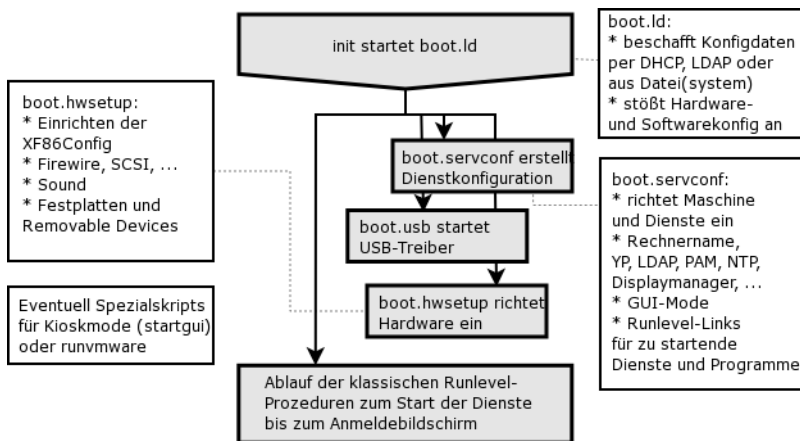


Abbildung 16.17: Anstelle von `boot` übernimmt `boot.ld` den Start einer DXS

`boot.servconf` richtet je nach gewünschter Konfiguration den Start der Dienste ein und schreibt in mehrere Konfigurationsdateien in `/etc`. Die Dateien `boot.hwsetup` und `boot.usb` richten die Hardware ein. Da die Clients ohne Festplatte oder anderen Festspeicher arbeiten, kann man auf den jeweiligen PCs selbst deren Konfiguration nicht speichern. Die automatische Hardware-Erkennung basiert auf dem Hotplug-Mechanismus des 2.6er Kernels und dem `hwinfo`-Tool von SuSE.

`dxs-inst` oder `d2d1` decken bei weitem nicht alle Konfigurationsmöglichkeiten ab. Ein Teil der Konfiguration können Sie über DHCP-Variablen oder durch Bearbeiten der Konfigurationsdateien im Client-Dateisystem beeinflussen. Wenn eine Datei von einem DXS-Skript bearbeitet wird, steht dieses im Kopf der Datei. So wissen Sie immer, welches Skript welche Datei wann angefasst hat.

Alle Änderungen an Konfigurationsdateien sollten Sie auf dem Server ausführen, da Änderungen auf dem Client beim Neustart verloren gehen. Alle Änderungen am Server wirken sich erst beim Neustart der Clients aus.

Die kurzen Beschreibungen können nicht alle Aspekte von Linux-Terminal-Servern beschreiben. Weitergehende Erklärungen, Hintergründe und einen breiteren Überblick finden Sie im Buch *Linux Terminalserver* von Holger Burbach, Bernd Kretschmer und Dirk von Suchodoletz. Informationen und Ergänzungen zu diesem Buch stehen auf der Webseite <http://www.linux-terminalserver.de>.