

10 Thin-Clients statt PCs

Nach der Hochblüte der Mainframes und Zeichenterminals erlebten wir die Client-Server-Ära, in der Benutzer überwiegend Windows-PCs und Windows-Anwendungen als Schnittstelle zur Datenverarbeitung nutzten.

Diese Client-Server-Umgebungen mit Windows-PCs und Windows-Anwendungen verursachen nach voneinander unabhängigen Untersuchungen amerikanischer Unternehmensberater im Mittel jährlich Kosten von ca. 10.000 Euro 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 Benutzersupports, dann wieder funktionsfähige Arbeitsplätze herzustellen, verloren geht. Die Kosten tragen sie, auch ohne zu wissen, woher sie kommen.

Hier setzt serverbasierte Datenverarbeitung (Serverbased Computing) an,

- die das Ausführen von Software wieder zentralisiert,
- an den Benutzerarbeitsplätzen Anwendungen nur an *schlanken* Endgeräten (sogenannten Thin-Clients) anzeigt, aber nicht ablaufen lässt und
- durch Spiegeln der Arbeitssitzungen der Benutzer einen kostengünstigen Support von zentraler Stelle aus ermöglicht.

Während beim ausgereiften Betriebssystem Unix Anwendungen auf einem Unix-Server laufen und Benutzer die Anwendung auf einem Zeichen- oder X-Terminal sehen, funktioniert diese Idee jetzt endlich auch einigermaßen problemlos für Windows-Anwendungen.

Zunächst hat Citrix die sogenannte MultiWin-Technologie entwickelt und vor 5 Jahren auf Basis von Windows NT 3.51 ein Multi-User-NT namens WinFrame auf den Markt gebracht. Schon damit war es möglich, Windows-Anwendungen zentral laufen zu lassen und auf Windows-Terminals anzuzeigen.

Mehr Vertrauen in diese Technik bekamen Entscheider, seit Microsoft diese Citrix-Technologie 1997 lizenzierte und 1998 in seine Windows NT 4.0 Terminal Server Edition integrierte.

Zur ganz normalen Windows-Betriebssystem-Technologie ist sie Anfang 2000 durch Windows 2000 geworden, das inzwischen Windows NT 4.0 TSE ersetzt. Alle Windows 2000 (Advanced-) Server enthalten stets die Fähigkeit, Multi-User-Dienste als sogenannte Terminaldienste anzubieten. Die Terminaldienste brauchen Betreiber nur zu installieren und zu lizenzieren.

Diese Terminaldienste können

- PCs mit Microsoft-Client-Software mit beliebigen Windows-Versionen,
- Windows-Terminals mit den Betriebssystemen Windows CE und Windows NT Embedded,
- beliebige Linux-Geräte wie PCs und Terminals mit Linux-Client-Software und
- viele weitere Geräte mit Middleware und Java-Clients von Tarantella und HOB

nutzen.

Zwischen dem Server für Terminaldienste und den Clients vermittelt dabei Microsofts proprietäres Terminaldienstprotokoll, das Remote Display Protocol (RDP).

Deutlich weniger Bandbreite braucht das Citrix-Protokoll Independent Computing Architecture (ICA). Es setzt auf dem Windows 2000 (Advanced-) Server die Verwaltungssoftware *Citrix Metaframe* voraus. Citrix Metaframe bietet darüber hinaus sehr nützliche Verwaltungstools, z.B. zum Zusammenfassen mehrerer Terminal-Server zu sogenannten Serverfarmen.

Citrix unterstützt mit dem ICA-Protokoll über zweihundert verschiedene Betriebssysteme und Endgeräte, darunter viele Unix-Plattformen inklusive Linux.

Zahlreiche Untersuchungen von Unternehmensberatungen zeigen, dass zentralisierte Datenverarbeitung auf Windows-Terminal-Servern und Nutzung dieser Terminaldienste auf Terminals oder Terminal-ähnlichen PCs wesentlich niedrigere Gesamtkosten ermöglicht, weil auf der Benutzerseite weniger Störungen und Supportfälle auftreten und Unternehmen Support von beliebiger Stelle aus statt vor Ort anbieten können.

Die von neuen eBusiness-Anwendungen verlangte Flexibilität und Geschwindigkeit lässt sich mit serverbasierter Datenverarbeitung eher erreichen als mit PC-Umgebungen.

Dieses Kapitel beschreibt, wie man an verschiedenen Typen von Linux-Endgeräten die populären Windows-Anwendungen zentral zur Verfügung stellen kann.

Zunächst lernen Sie Konzepte für Windows- und Linux-Endgeräte und Browser-Appliances kennen

Danach folgt ein technisch ausgelegter Abschnitt zum Einrichten sehr kostengünstiger Linux-basierter Diskless PCs.

Microsoft plant mit seiner dot.net Initiative ganz neue Lösungen, die wesentlich besser skalieren sollen als die derzeitigen Terminaldienste.

10.1 Konzepte für Thin-Clients

Lesen Sie hier über Windows- und Linux-Lösungen mit schlanken Endgeräten, die Arbeitssitzungen von Benutzern auf Terminal-Servern darstellen:

- Windows-PCs,
- Windows-Terminals,
- Linux/Unix-Server und Workstations,
- Linux-Diskless-Geräte mit Flash-ROM und
- Browser-Appliances.

10.1.1 Windows-PCs

Auf PCs mit Windows 3.11 bis Windows 2000 kann man 16-Bit Terminal-Clients von Microsofts Protokoll RDP oder dem Citrix Protokoll ICA und auf PCs mit Windows 95 bis Windows 2000 32-Bit Clients von RDP oder ICA laden und damit Sitzungen auf Citrix Winframe, Windows NT 4.0 Terminal Server Edition oder Windows 2000 Server mit oder ohne Metaframe betreiben. Inzwischen setzen Microsoft und Citrix eher auf Webclients, die sich automatisch installieren. Mit Terminaldiensten wird man zwar die bekannten Probleme mit Windows-PCs nicht los, kann aber wenigstens Software zentral zur Verfügung stellen, administrieren, aktualisieren und betreuen.

Probleme gibt es hier noch bei Druckdiensten, wenn lokale Drucker an den Windows-PCs verwendet werden sollen: Hier führen Druckertreiber, die nicht für dieses Umfeld geeignet sind, zu Systemabstürzen und unkomprimierte Druckjobs belasten das Netz oft so sehr, dass die Benutzersitzungen sehr langsam werden können. Hilfe bieten Kompressionslösungen wie thin print.

10.1.2 Windows-Terminals

Microsoft hat Windows-Terminals mit all seinen Varianten von Windows CE und neuerdings von Windows NT Embedded an Terminal-Hersteller wie Wyse, Bounless, NCD, Tektronix etc. lizenziert.

Die seit Anfang 2000 überholten Single-Session-Windows CE-Terminals kommen ohne Browser und weitere Windows CE-Anwendungen daher, damit Anwender zum Browsen auf Terminal-Server zugreifen müssen und Firmen mehr Verbindungs-Lizenzen brauchen. Windows Terminals mit Windows CE können sich über die Protokolle RDP und ICA mit Windows-Terminal-Servern verbinden. Abbildung 10.1 zeigt ein handgroßes Windows CE-Terminal des Marktführers Wyse.



Abbildung 10.1: Windows CE-Terminal WT32000 LE von Wyse

Erst als der Terminal-Marktführer Wyse im Sommer 1999 ein Linux-Terminal mit Browser und zahlreichen Host-Emulationen auf den Markt brachte, ließ Microsoft Terminals mit Windows NT Embedded und Internet Explorer zu, freilich mit höheren Lizenzgebühren. Abbildung 10.2 gibt einen Überblick über Windows-Terminals. In Abbildung 10.3 sehen Sie ein Windows NT Embedded Terminal von Wyse.

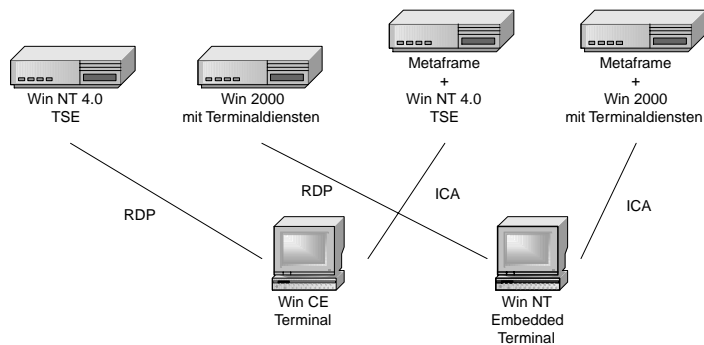


Abbildung 10.2: Windows Terminals mit Windows CE und NT Embedded



Abbildung 10.3: Windows NT Embedded Terminal von Wyse

10.1.3 Linux/Unix-Server und Workstations

Auf Linux/Unix-Rechnern laufen Open Source-RDP-Clients und lizenzfreie ICA-Clients von Citrix. Damit können Anwender gleichzeitig mehrere Arbeitssitzungen auf mehreren Windows-Terminalservern, Mainframes und Unix-Systemen nutzen. An Linux/Unix-Servern arbeiten Anwender selten direkt, wahrscheinlicher ist, dass sie über X-Terminals an diese angeschlossen sind.

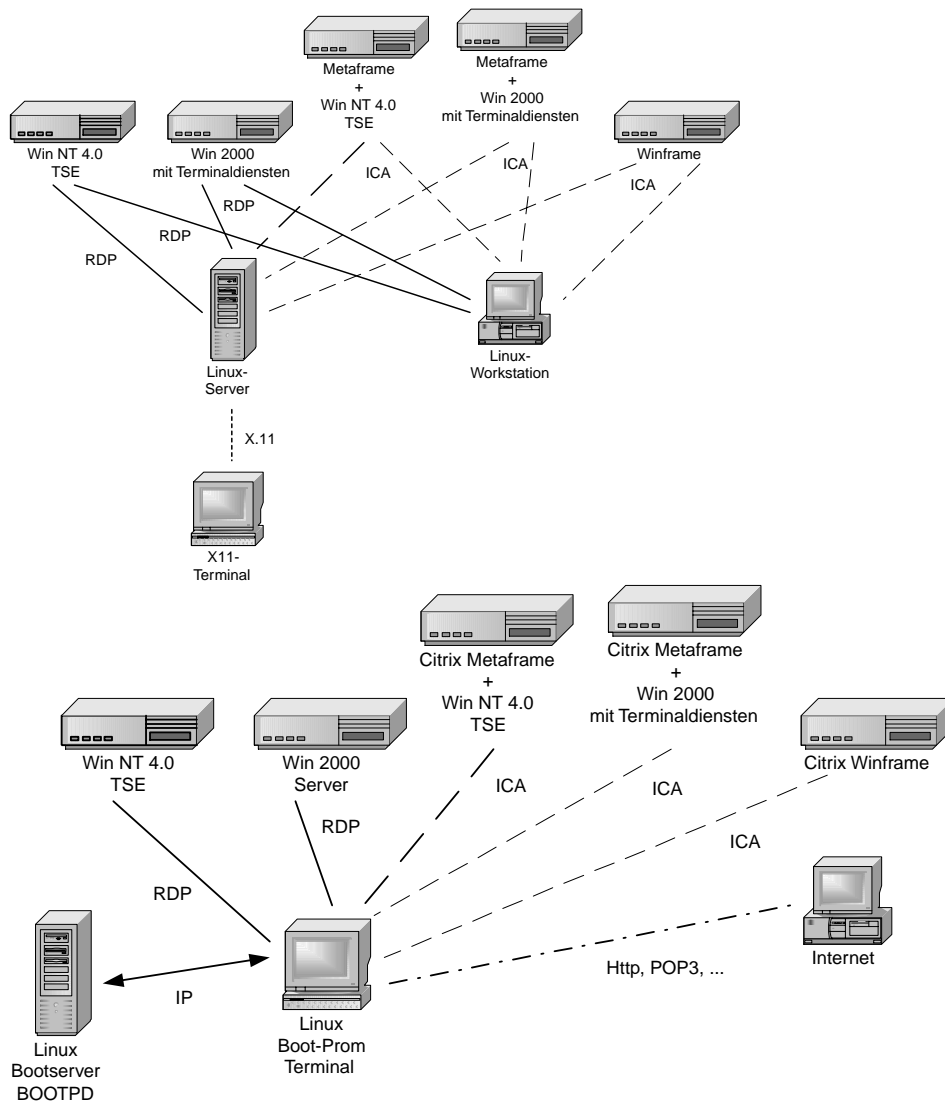


Abbildung 10.4: Linux-Server, Linux-Workstations und Linux-Terminals für Windows-Terminaldienste

10.1.4 Diskless Linux-Geräte mit Flash-ROM

Eine besonders sparsame und stabile Form von Linux-Geräten verzichtet auf Fest- und Wechselplatten und Diskettenlaufwerke und bootet Linux von Flash-ROMs. Weltweit Marktführer für diese Lösung ist die Augsburgener jetzt Infomatec AG, die ihre Linux-Flash-ROM-Lösung u.a. über VXL, Siemens,

Melchers und TechniSat vertreibt. Siemens nutzt die Lösung in PCs, Melchers bietet Linux-Terminals für ca. 500 Euro und Steckkarten für ca. 200 Euro für PCs unter dem Markennamen IGEL an. Die Steckkarten verwandeln PCs ab 486-Prozessoren und 32 MB RAM in vollwertige Linux-Terminals.

Diese Lösung steckt übrigens weltweit in vielen hunderttausenden von Set-Top-Boxen für den Internet-Zugang über funktionsbeschränkte Netscape/Mozilla-Browser und Fernsehgeräte als Bildschirmersatz.

Sie ist wirklich einfach einzurichten; Terminals wie PCs mit den Flash-ROM-Steckkarten haben ein komfortables Setup, das nur nach allgemeinverständlichen Daten wie Adresse des Servers, IP-Adresse oder DHCP und Endbenutzer-Sprache fragt.

Die Thin Clients Netvista N2200 und N2800 von IBM sind noch flexibler. Sie können entweder Turbo Linux vom Flash-ROM oder über Boot Proms von Boot-Servern oder von benachbarten Thin Clients booten.



Abbildung 10.5: Linux-Terminal mit Flash-ROM-Technologie



Abbildung 10.6: Eine Flash-ROM-Karte verwandelt den PC in ein Linux-Terminal

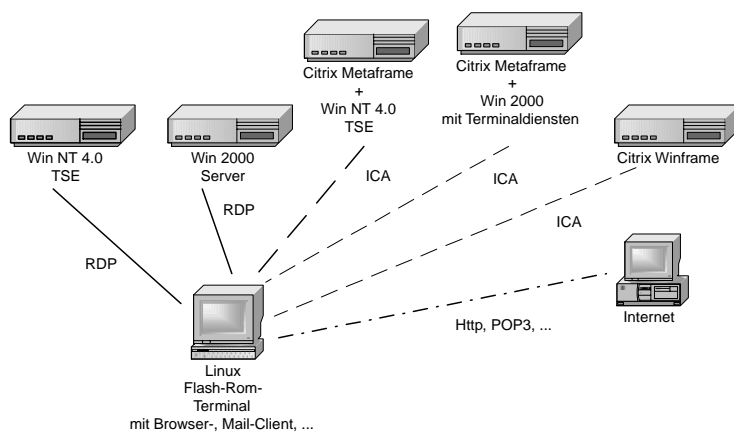


Abbildung 10.7: Serverbasierte Datenverarbeitung mit Linux-Flash-ROM-Terminals

10.1.5 Diskless Linux-Geräte 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 (www.bootix.com) und zahlreichen freien Linux-Projekte, z.B. von Heise, von der Unix Göttingen oder – wie hier dargestellt – vom Linux-Terminal-Server-Projekt. Als Hardware braucht man nur einen 486er PC mit 16 MB RAM und einer handverlesenen Netzwerkkarte mit Bootprom für unter 20 Euro. Zwar verbringen Systemverwalter zunächst mehr Zeit mit dem Einrichten, können aber später alles zentral auf dem Bootserver pflegen. Während die zu-

vor erwähnten Lösungen mit Flash-ROMs Endanwender ohne Computerkenntnisse zum Laufen bringen, müssen hier erst Systemverwalter fleißig sein. Sie haben kaum mehr zu tun, um 100 Diskless PCs einzurichten, als bei 2 herkömmlichen PCs. 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 Euro Materialkosten funktionsfähige Endgeräte ohne Festplatte und Diskettenlaufwerk.

Hier 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.

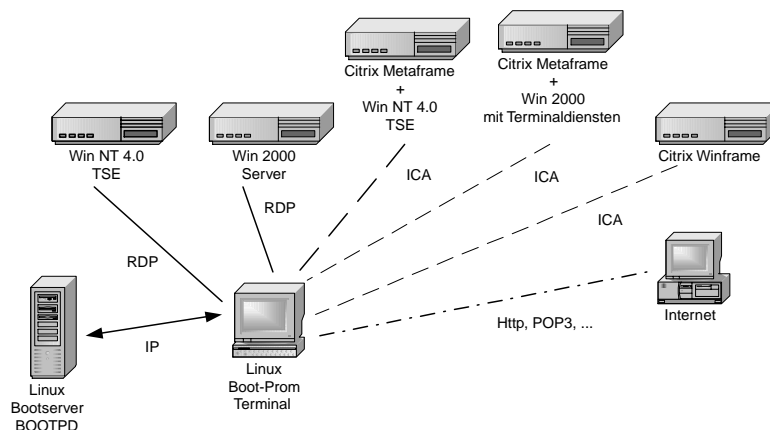


Abbildung 10.8: Linux-Bootprom-Terminals booten von Linux-Bootservern

Eine Version dieser Technologie beschreibt der folgende Abschnitt 10.2.

10.1.6 Browser-Appliances

Nach all den Terminals mit den proprietären ICA- und RDP Protokollen kommen jetzt immer mehr Browser-Appliances auf den Markt, das sind Endgeräte wie die Intel.Dot.Station, die nur einen Browser als Benutzerschnittstelle für alle möglichen Programme bieten. Appliances können sich in Set-Top-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 oben beschriebene Flash-ROM-Technik.

10.2 Diskless Linux-Geräte mit Boot-Prom einrichten

Dieser zweite Teil des Kapitel beschreibt, wie Sie plattenlose PCs von einem Boot-Server starten, um sie dann an Linux- oder Windows-Terminal-Servern zum Darstellen von ICA-, RDP- und X11-Sitzungen betreiben zu können. 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.

10.2.1 Überblick

Im Laufe dieses Abschnitts lesen Sie Schritt für Schritt, wie Sie die X-Terminals (Clients) einrichten. Sobald Sie die X-Terminals eingerichtet haben, starten sie mit folgender Abfolge:

1. Das Bootrom initialisiert die Netzwerkkarte.
2. Das Bootrom 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-Filesystem 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.
12. Der XDM-Server beantwortet diese Anfragen. Der Client zeigt den Begrüßungsbildschirm des XDM-Servers mit dem Login.

13. Der Benutzer loggt sich auf dem Server ein. Ihm steht jetzt seine X-Windows-Oberfläche zur Verfügung, und er kann damit auf dem lokalen Linux-Systeme oder auf anderen Linux-Rechnern arbeiten.

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

10.2.2 Benötigte Softwarekomponenten

Auf einem Linux-Server benötigt man dazu einen DHCP-Server, einen TFTP-Server, einen NFS-Server sowie einen XDM-Server. Der DHCP-Server muss ggf. nachinstalliert werden. Die weitere Software können Sie von der Etherboot-Homepage und von der Homepage des Linux-Terminal-Server-Projekts beziehen:

Laden Sie

- von der Homepage des Etherboot-Projektes (<http://www.slug.org.au/etherboot/>) das Paket `etherboot-4.4.5.tar.bz2` und
- von der Homepage des Linux-Terminal-Server-Projekts folgende Pakete (<http://www.ltsproject.org/>):
`lts_core-1_03-1_i386.rpm`,
`lts_kernel_all-1_03-1_i386.rpm`,
`lts_xmach64-1_1.rpm`,
`lts_xsvga-1_2.rpm`.

Das Paket `lts_core-1_03-1_i386.rpm` enthält das Root-Filesystem der Clients. Das Paket `lts_kernel_all-1_03-1_i386.rpm` enthält den Linux-Kernel für die Clients. In diesen Kernel ist die Unterstützung für viele Netzwerkkarten einkompiliert. Sollte der Treiber für die Netzwerkkarten ihrer Clients nicht dabei sein, so müssen Sie den Kernel neu kompilieren. Beschaffen Sie also lieber Netzwerkkarten, für die Sie fertige Treiber bekommen, wenn Sie sich keine unnötige Arbeit machen wollen. Die Pakete `lts_xmach64-1_1.rpm` und `lts_xsvga-1_2.rpm` enthalten die X-Server. Sie müssen zu den Grafikkarten die benötigten X-Server ermitteln und fernladen. Die folgenden beiden X-Server sind speziell für ein *readonly*-gemountetes Filesystem angepasst.

10.2.3 Softwarekomponenten installieren und Systemdateien anpassen

Installieren Sie zunächst das Root-Filesystem mit dem Befehl:

```
rpm -i lts_core-1_03-1_i386.rpm
```

Die Fehlermeldungen dabei können Sie ignorieren.

Anschließend installieren Sie die anderen Pakete mit dem Befehl:

```
rpm -i rpm-Datei
```

Nun geht es daran, einige Details anzupassen:

Entfernen Sie in der Datei `/etc/inetd.conf` das Kommentarzeichen `#` vor `tftp`, damit der TFTP-Server bei Anfragen auch startet:

```
# Tftp service is provided primarily for booting. Most sites
# run this only on machines acting as "boot servers."
#
tftp dgram udp wait nobody /usr/sbin/tcpd in.tftpd /tftpboot
```

Damit diese Änderungen tatsächlich wirksam werden, muss der Superserver `inet.d` neu gestartet werden. Geben Sie bitte anschließend ein:

```
/sbin/init.d/inetd restart
```

Die Variable für den Start des `inetd` in der Datei `/etc/rc.config` muss auf `yes` stehen:

```
START_INETD="yes"
```

Editieren Sie dann bitte die Datei `/etc/exports`:

```
## LTS-begin ##
/tftpboot/lts/ltsroot
 192.168.1.0/255.255.255.0(ro,no_root_squash)
## LTS-end ##
```

Dies exportiert das Root-Filesystem für die Clients so, dass es *read-only* ist, die Kennung von *root* aber nicht auf `nobody` umgesetzt wird. Wenn Sie den Kernel-NFS-Dämon benutzen, müssen Sie in der Datei `/etc/rc.config` folgende Variablen setzen:

```
START_PORTMAP="yes"
NFS_SERVER="yes"
USE_KERNEL_NFSD="yes"
```

Wenn Sie den Kernel-NFS-Dämon nicht benutzen wollen, müssen Sie das Paket `nfsserv` aus der Serie `n` installieren und die Variable `USE_KERNEL_NFSD="no"` setzen.

Anschließend starten Sie die NFS-Server neu:

```
/sbin/init.d/nfsserver restart
```

Der NFS-Server muss anschließend alle IP-Adressen der Clients in die Rechnernamen auflösen können. Sie müssen also entweder einen DNS-Server konfigurieren oder alle Clients in die `hosts`-Datei einfügen.

Die xdm-Dateien sollten Sie wie folgt anpassen:

Damit auf dem Server nicht auch X-Windows laufen muss, setzen Sie in der Datei `/usr/X11R6/lib/X11/xdm/Xservers` das Kommentarzeichen vor die Zeile:

```
#:0 local /usr/X11R6/bin/X :0 vt07
```

In der Standardeinstellung von SuSE beantwortet der XDM-Server nur lokale Anfragen. Um dies zu ändern, kommentieren Sie die letzte Zeile der Datei `/usr/X11R6/lib/X11/xdm/xdm-config` aus. Ändern Sie also

```
DisplayManager.requestPort: 0
```

in:

```
! DisplayManager.requestPort: 0
```

Damit der XDM-Server automatisch startet, ändern Sie mit YaST die Login-Konfiguration auf graphisches Login mit XDM.

Geben Sie `init 3` ein, um die Änderungen sofort wirksam zu machen.

Setzen Sie die Variable `START_DHCPD="yes"` in der Datei `/etc/rc.config`. Passen Sie danach nach folgendem Muster mit IP-Adressen Ihres lokalen Netzes und den Hardware-Adressen Ihrer eigenen Netzwerkkarten die Datei `/etc/dhcpd.conf` an:

```
option domain-name "lokales-netz.de";
option domain-name-servers 192.168.1.2;
option netbios-name-servers 192.168.1.2;
option routers 192.168.1.2;
option subnet-mask 255.255.255.0;
default-lease-time 1209600;
max-lease-time 12096000;
subnet 192.168.1.0 netmask 255.255.255.0 {
}

group {
    use-host-decl-names on;

    host hh1-20 {
        hardware ethernet 00:80:48:e4:7a:2f;
        fixed-address 192.168.1.20;
        filename "/tftpboot/lts/vmlinuz.all";
    }
    host hh1-21 {
        hardware ethernet 00:00:B4:5D:13:54;
```

```

fixed-address 192.168.1.21;
filename "/tftpboot/lts/vmlinuz.all";
}
host hh1-22 {
hardware ethernet 00:00:E8:90:76:2C;
fixed-address 192.168.1.22;
filename "/tftpboot/lts/bzimage";
}
host hh1-23 {
hardware ethernet 00:50:56:B2:02:0A;
fixed-address 192.168.1.23;
filename "/tftpboot/lts/bzimage";
}
}

```

Bei den obigen `host`-Deklarationen handelt es sich um Beispiele für drei verschiedene Rechner. Abhängig von der Adresse ihrer Netzwerkkarte erhalten sie eine IP-Adresse zugewiesen.

Starten Sie dann mit `/sbin/init.d/dhcp restart` den DHCP-Server neu.

Bitte passen Sie danach die Datei `/tftpboot/lts/ltsroot/etc/lts.conf` an. Die Parameter sollten selbsterklärend sein.

Falls Sie einen X-Font-Server benutzen, können Sie die Variable `USE_XFS` auf `yes` setzen. Der `UI_MODE` gibt an, ob die Workstation ein XDM-Login (mit graphischer Oberfläche, `UI_MODE = GUI`) oder ein Telnet-Login (`UI_MODE = CHAR`) am Server macht. Den für die Graphikkarte Ihrer Workstations notwendigen X-Server entnehmen Sie bitte der Dokumentation von XF86.

```

[Default]
XSERVER = XF86_SVGA
XDM_SERVER = 192.168.1.2
SYSLOG_HOST = 192.168.1.2
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
USE_XFS = N
UI_MODE = GUI

[hh1-20]
XSERVER = XF86_Mach64

[hh1-21]

```

```

XSERVER          = XF86_Mach64
[hh1-22]
XSERVER          = XF86_Mach64
#
# [ws001]
# XSERVER = XF86_SVGA
# X_MOUSE_PROTOCOL = "Microsoft"
# X_MOUSE_DEVICE = "/dev/ttyS1"
# X_MOUSE_RESOLUTION = 50
# X_MOUSE_BUTTONS = 3
# X_MOUSE_BAUD = 1200
# UI_MODE = CHAR
TELNET_HOST = 192.168.0.254
#
# [ws002]
# XSERVER = XF86_Mach64
#
# [ws003]
# XSERVER = XF86_SVGA
# X_COLOR_DEPTH = 24

```

Falls Sie auf den Clients Wert auf eine deutsche Tastaturbelegung legen, editieren Sie noch die Datei `/tftboot/lts/ltsroot/etc/rc.local` und ändern Sie sie, so dass die Sektion `Keyboard` für die `xf86config` wie folgt aussieht:

```

Section "Keyboard"
    Protocol "Standard"
    AutoRepeat 500 5
XkbKeymap      "xfree86(de)"
EndSection

```

Anschließend kopieren Sie das Verzeichnis `/usr/X11R6/lib/X11/xkb/` nach `/tmp`:

```
cp -a /usr/X11R6/lib/X11/xkb/ /tmp
```

Wechseln Sie dann in das Verzeichnis `/tmp/xkb` und löschen die Datei `xkbcomp` und kopieren den Rest in den zu exportierenden Dateibaum:

```

cd /tmp/xkb
rm xkbcomp
cp -a * /tftboot/lts/ltsroot/usr/X11R6/lib/X11/xkb/

```

Wechseln Sie danach in das Home-Verzeichnis und löschen Sie das Verzeichnis `/tmp/xkb`.

10.2.4 Installation des Etherboot-Paketes und erste Tests

Entpacken Sie bitte mit dem Befehl:

```
tar -xIvf etherboot-4.4.5.tar.bz2 -C /usr/src
```

das Etherboot-Paket und wechseln dann in das Verzeichnis `/usr/src/etherboot-4.4.5/src-32` und editieren die Datei `config`. Ändern Sie dann bitte folgende Passage

```
CFLAGS+= -DMOTD -DIMAGE_MENU
```

in:

```
CFLAGS+= -DMOTD -DIMAGE_MENU -DGAS295 -DASK_BOOT=3
```

Der letzte Parameter bietet 3 Sekunden lang die Auswahl an, lokal oder vom Netzwerk zu booten. Wenn Sie ihn weglassen, bootet die Lösung direkt vom Netz. Ein anschließendes `make` kompiliert das Paket. Sie erhalten eine Testdiskette, von der Sie booten können, wenn Sie (für eine Karte mit dem `rtl8139`-Chipsatz) eingeben:

```
make rtl8139.fd0
```

Tipp: Der Treiber für Ne2000-kompatible Netzwerkkarten testet nur die Ports 280, 300, 320 und 340.

Wenn Sie mit einem Bootrom booten wollen, sollten Sie sich Bootroms mit den Treibern des Etherboot-Paketes über das Internet bestellen oder sich einen Eprom-Brenner zulegen und sich die Bootroms selber herstellen. 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. Manchmal machen die Programme der Netzwerkkartenhersteller Schwierigkeiten, wenn Sie die Netzwerkkarte so einstellen wollen, dass ein Booten vom Bootrom möglich ist. Hier hilft manchmal ein Programm des Chipsatzherstellers weiter.

10.2.5 Kompilieren eines Kernels für die Clients

Beim Neukompilieren eines Kernels muss man unbedingt zusätzlich zu den sonst üblichen Parametern setzen:

```
CONFIG_BLK_DEV_RAM=y #
```

Dies bietet die Möglichkeit, eine Ramdisk anzulegen.

```
CONFIG_IP_PNP=y #
```


Der Kernel soll das Netzwerk automatisch konfigurieren können.

```
CONFIG_IP_PNP_BOOTP=y
# Dazu soll er Bootp- Antworten benutzen.
CONFIG_NFS_FS=y
# Das NFS- Dateisystem wird fest in den Kernel kompiliert.
CONFIG_ROOT_NFS=y
# Der Kernel muss sein Root- File- System
# via NFS mounten können.
```

Dazu muss der Netzwerkkartentreiber für Ihre Karte fest in den Kernel eingekompiliert werden. Als Vorlage kann Ihnen die folgende automatisch generierte Konfigurationsdatei für einen Kernel mit Support für den rtl8139-Chipsatz dienen:

```
#
# Automatically generated by make menuconfig: don't edit
#
#
# Sprache der Kernel Konfiguration
#
# CONFIG_CONFIGLANG_ENGLISH is not set
CONFIG_CONFIGLANG_GERMAN=y
#
# Grundsätzliches zur Codegenerierung
#
CONFIG_EXPERIMENTAL=y
#
# Prozessorfamilie und -einstellungen
#
CONFIG_M386=y
# CONFIG_M486 is not set
# CONFIG_M586 is not set
# CONFIG_M586TSC is not set
# CONFIG_M686 is not set
CONFIG_1GB=y
# CONFIG_2GB is not set
CONFIG_MATH_EMULATION=y
CONFIG_MTRR=y
# CONFIG_SMP is not set
#
# Unterstützung für Kernelmodule
#
# CONFIG_MODULES is not set
```

```
#  
# Grundeinstellungen  
#  
# CONFIG_BIGMEM is not set  
CONFIG_NET=y  
CONFIG_PCI=y  
# CONFIG_PCI_GOBIOS is not set  
# CONFIG_PCI_GODIRECT is not set  
CONFIG_PCI_GOANY=y  
CONFIG_PCI_BIOS=y  
CONFIG_PCI_DIRECT=y  
# CONFIG_PCI_USE_RT is not set  
CONFIG_PCI_QUIRKS=y  
# CONFIG_PCI_OPTIMIZE is not set  
CONFIG_PCI_OLD_PROC=y  
# CONFIG_MCA is not set  
# CONFIG_VISWS is not set  
CONFIG_SYSVIPC=y  
CONFIG_BSD_PROCESS_ACCT=y  
CONFIG_SYSCTL=y  
# CONFIG_BINFMT_AOUT is not set  
CONFIG_BINFMT_ELF=y  
CONFIG_BINFMT_MISC=y  
# CONFIG_BINFMT_JAVA is not set  
CONFIG_PARPORT=y  
# CONFIG_PARPORT_PC is not set  
# CONFIG_APM is not set  
#  
# Plug and Play Unterstützung  
#  
CONFIG_PNP=y  
# CONFIG_PNP_PARPORT is not set  
#  
# Blockorientierte Geräte  
#  
# CONFIG_BLK_DEV_FD is not set  
# CONFIG_BLK_DEV_IDE is not set  
# CONFIG_BLK_DEV_HD_ONLY is not set  
# CONFIG_BLK_DEV_LVM is not set  
# CONFIG_BLK_DEV_LOOP is not set  
# CONFIG_BLK_DEV_NBD is not set  
# CONFIG_BLK_DEV_MD is not set
```

```
CONFIG_BLK_DEV_RAM=y
# CONFIG_BLK_DEV_INITRD is not set
# CONFIG_BLK_DEV_XD is not set
# CONFIG_BLK_DEV_DAC960 is not set
CONFIG_PARIDE_PARPORT=y
# CONFIG_PARIDE is not set
# CONFIG_BLK_DEV_IDE_MODES is not set
# CONFIG_BLK_CPQ_DA is not set
# CONFIG_BLK_DEV_HD is not set
#
# Netzwerkeinstellungen
#
# CONFIG_PACKET is not set
# CONFIG_NETLINK is not set
# CONFIG_FIREWALL is not set
# CONFIG_FILTER is not set
CONFIG_UNIX=y
CONFIG_INET=y
# CONFIG_IP_MULTICAST is not set
# CONFIG_IP_ADVANCED_ROUTER is not set
CONFIG_IP_PNP=y
CONFIG_IP_PNP_BOOTP=y
# CONFIG_IP_PNP_RARP is not set
# CONFIG_IP_ROUTER is not set
# CONFIG_NET_IPIP is not set
# CONFIG_NET_IPGRE is not set
# CONFIG_IP_ALIAS is not set
# CONFIG_ARPD is not set
# CONFIG_SYN_COOKIES is not set
# CONFIG_INET_RARP is not set
# CONFIG_SKB_LARGE is not set
# CONFIG_IPV6 is not set
# CONFIG_IPX is not set
# CONFIG_ATALK is not set
# CONFIG_X25 is not set
# CONFIG_LAPB is not set
# CONFIG_BRIDGE is not set
# CONFIG_LLC is not set
# CONFIG_ECONET is not set
# CONFIG_WAN_ROUTER is not set
# CONFIG_NET_FASTROUTE is not set
# CONFIG_NET_HW_FLOWCONTROL is not set
```

```
# CONFIG_CPU_IS_SLOW is not set
#
# Quali of Service
#
# CONFIG_NET_SCHED is not set
#
# Telephony Support
#
# CONFIG_PHONE is not set
# CONFIG_PHONE_IXJ is not set
#
# SCSI Unterstützung
#
# CONFIG_SCSI is not set
#
# I2O device support
#
# CONFIG_I2O is not set
# CONFIG_I2O_PCI is not set
# CONFIG_I2O_BLOCK is not set
# CONFIG_I2O_SCSI is not set
#
# IEEE 1394 (FireWire) support
#
# CONFIG_IEEE1394 is not set
#
# Netzwerkkartentreiber
#
CONFIG_NETDEVICES=y
#
# ARCnet devices
#
# CONFIG_ARCNET is not set
# CONFIG_DUMMY is not set
# CONFIG_BONDING is not set
# CONFIG_EQUALIZER is not set
# CONFIG_NET_SB1000 is not set
# CONFIG_PPPOX is not set
#
# Ethernet (10 or 100Mbit)
#
CONFIG_NET_ETHERNET=y
```

```
# CONFIG_NET_VENDOR_3COM is not set
# CONFIG_LANCE is not set
# CONFIG_NET_VENDOR_SMC is not set
# CONFIG_NET_VENDOR_RACAL is not set
CONFIG_RTL8139=y
# CONFIG_NET_ISA is not set
CONFIG_NET_EISA=y
# CONFIG_PCNET32 is not set
# CONFIG_AC3200 is not set
# CONFIG_APRICOT is not set
# CONFIG_CS89x0 is not set
# CONFIG_DM9102 is not set
# CONFIG_DE4X5 is not set
# CONFIG_DEC_ELCP is not set
# CONFIG_DEC_ELCP_OLD is not set
# CONFIG_DGRS is not set
# CONFIG_EEXPRESS_PRO100 is not set
# CONFIG_EEXPRESS_PRO100_OLD is not set
# CONFIG_LNE390 is not set
# CONFIG_NE3210 is not set
# CONFIG_NE2K_PCI is not set
# CONFIG_RL100ATX is not set
# CONFIG_TLAN is not set
# CONFIG_VIA_RHINE is not set
# CONFIG_SIS900 is not set
# CONFIG_ES3210 is not set
# CONFIG_EPIC100 is not set
# CONFIG_ZNET is not set
# CONFIG_NET_POCKET is not set
#
# Ethernet (1000 Mbit)
#
# CONFIG_ACENIC is not set
# CONFIG_YELLOWFIN is not set
# CONFIG_SK98LIN is not set
# CONFIG_FDDI is not set
# CONFIG_HIPPI is not set
# CONFIG_PLIP is not set
# CONFIG_PPP is not set
# CONFIG_SLIP is not set
# CONFIG_NET_RADIO is not set
#
```

```
# Token ring devices
#
# CONFIG_TR is not set
# CONFIG_NET_FC is not set
# CONFIG_RCPCI is not set
# CONFIG_SHAPER is not set
#
# Wan interfaces
#
# CONFIG_HOSTESS_SV11 is not set
# CONFIG_COSA is not set
# CONFIG_SEALEVEL_4021 is not set
# CONFIG_LANMEDIA is not set
# CONFIG_COMX is not set
# CONFIG_DLCI is not set
# CONFIG_SBNI is not set
#
# Amateur Radio Unterstützung
#
# CONFIG_HAMRADIO is not set
#
# IrDA (infrared) support
#
# CONFIG_IRDA is not set
#
# ISDN Subsystem
#
# CONFIG_ISDN is not set
#
# Alte CD-ROM Treiber (nicht SCSI, nicht IDE)
#
# CONFIG_CD_NO_IDESCSI is not set
#
# Zeichenorientierte Geräte
#
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_SERIAL=y
# CONFIG_SERIAL_CONSOLE is not set
# CONFIG_SERIAL_EXTENDED is not set
# CONFIG_SERIAL_NONSTANDARD is not set
# CONFIG_UNIX98_PTYS is not set
```

```
CONFIG_PRINTER=y
CONFIG_PRINTER_READBACK=y
CONFIG_MOUSE=y
#
# Mäuse
#
# CONFIG_ATIXL_BUSMOUSE is not set
# CONFIG_BUSMOUSE is not set
# CONFIG_MS_BUSMOUSE is not set
CONFIG_PSMOUSE=y
CONFIG_82C710_MOUSE=y
# CONFIG_PC110_PAD is not set
#
# Joystick Unterstützung
#
# CONFIG_JOYSTICK is not set
# CONFIG_QIC02_TAPE is not set
# CONFIG_WATCHDOG is not set
# CONFIG_NVRAM is not set
CONFIG_RTC=y
#
# I2C support
#
# CONFIG_I2C is not set
# CONFIG_AGP is not set
#
# Video für Linux
#
# CONFIG_VIDEO_DEV is not set
# CONFIG_DTLK is not set
#
# Ftape, der Floppy-Tape Treiber
#
# CONFIG_FTAPE is not set
#
# USB support
#
# CONFIG_USB is not set
#
# Dateisysteme
#
# CONFIG_QUOTA is not set
```

```
# CONFIG_AUTOFS_FS is not set
# CONFIG_ADFS_FS is not set
# CONFIG_AFFS_FS is not set
# CONFIG_HFS_FS is not set
# CONFIG_FAT_FS is not set
# CONFIG_MSDOS_FS is not set
# CONFIG_UMSDOS_FS is not set
# CONFIG_VFAT_FS is not set
# CONFIG_IS09660_FS is not set
# CONFIG_JOLIET is not set
# CONFIG_UDF_FS is not set
# CONFIG_MINIX_FS is not set
# CONFIG_NTFS_FS is not set
# CONFIG_HPFS_FS is not set
CONFIG_PROC_FS=y
CONFIG_PROC_CONFIG=y
# CONFIG_QNX4FS_FS is not set
# CONFIG_ROMFS_FS is not set
CONFIG_EXT2_FS=y
# CONFIG_SYSV_FS is not set
# CONFIG_UFS_FS is not set
# CONFIG_REISERFS_FS is not set
# CONFIG_EFS_FS is not set
#
# Netzwerk-Dateisysteme
#
# CONFIG_CODA_FS is not set
CONFIG_NFS_FS=y
CONFIG_ROOT_NFS=y
# CONFIG_NFSD is not set
CONFIG_SUNRPC=y
CONFIG_LOCKD=y
# CONFIG_SMB_FS is not set
# CONFIG_NCP_FS is not set
#
# Partitionstypen
#
# CONFIG_BSD_DISKLABEL is not set
# CONFIG_MAC_PARTITION is not set
# CONFIG_SMD_DISKLABEL is not set
# CONFIG_SOLARIS_X86_PARTITION is not set
# CONFIG_UNIXWARE_DISKLABEL is not set
```



```
# CONFIG_NLS is not set
#
# Konsolentreiber
#
CONFIG_VGA_CONSOLE=y
CONFIG_VIDEO_SELECT=y
# CONFIG_MDA_CONSOLE is not set
# CONFIG_FB is not set
#
# Soundunterstützung
#
# CONFIG_SOUND is not set
#
# Kernel debuggen
#
CONFIG_MAGIC_SYSRQ=y
```

Kompilieren Sie bitte anschließend den Kernel mit `make dep && make clean && make bzImage`.

Danach müssen Sie den Kernel noch weiter bearbeiten.

Kompilieren Sie zunächst das netboot-Paket aus dem etherboot-Paket:

```
cd /usr/src/etherboot-4.4.5/netboot-0.8.1/
./configure
make
```

Wechseln Sie in das Verzeichnis mknbi-linux und geben ein:

```
./mknbi -d /tftpboot/lts/ltsroot/ -i rom \
-k /usr/src/linux/arch/i386/boot/bzImage -o
/tftpboot/lts/bzimage
```

Anschließend sollte ein Client mit diesem Kernel booten können.

