

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 an möglichst *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 die 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 7 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 und deren Nachfolger Windows 2003 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 Anwender mit

- PCs mit Microsoft-Client-Software auf beliebigen Windows-Versionen,
- Windows-Terminals mit den Betriebssystemen Windows CE, Windows NT Embedded, Windows dot.net, Windows 2000 oder Windows XP,
- beliebigen Linux-Geräten wie PCs und Terminals mit Linux-Client-Software und
- vielen Geräten 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).

Dieses Protokoll nutzen Middleware-Server von Tarantella ebenfalls für die Verbindung zwischen den Microsoft Windows Servern und Tarantella Servern, während Tarantella für die Kommunikation zwischen Benutzer-Endgeräten und dem Tarantella-Middleware-Server das Adaptive Internet Protocol (AIP) einsetzt. Mehr zu Tarantella lesen Sie im Abschnitt 11.5.

Etwas weniger Bandbreite als Microsofts RDP braucht das Citrix-Protokoll Independent Computing Architecture (ICA). Es setzt auf dem Windows 2000/2003 (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-Terminalservern 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 .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

Um Sitzungen auf Citrix Winframe, Windows NT 4.0 Terminal Server Edition, Windows 2000 Server oder Windows 2003 Server mit oder ohne Metaframe zu betreiben, kann man

- auf PCs mit Windows 3.11 bis Windows XP 16-Bit Terminal-Clients von Microsofts Protokoll RDP oder dem Citrix Protokoll ICA oder AIP von Tarantella und
- auf PCs mit Windows 95 bis Windows XP 32-Bit Clients von RDP oder ICA laden.

Inzwischen setzen Microsoft, Citrix und Tarantella auch auf Web-Clients, 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 gab es hier lange bei Druckdiensten, wenn lokale Drucker an den Windows-PCs verwendet werden sollten: Hier führten Druckertreiber, die nicht für dieses Umfeld geeignet sind, zu Systemabstürzen und unkomprimierte Druckjobs belasteten 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 angefangen mit Windows CE bis zu den aktuellsten Windows-Abarten an Terminal-Hersteller wie Wyse, Bounless, NCD, Tektronix etc. lizenziert.

Die seit Anfang der Jahrtausendwende überholten Single-Session-Windows CE-Terminals kamen ohne Browser und weitere Windows CE-Anwendungen daher, damit Anwender zum Browsen auf Terminal-Server zugreifen mussten und Firmen mehr Verbindungslizenzen brauchen. Windows-Terminals mit Windows CE können sich über die Protokolle RDP und ICA mit Windows-Terminalservern verbinden.

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.

Inzwischen bietet die Terminal-Industrie eine immer größere Auswahl von Thin-Clients, die von lokalen Speichern, von Boot-Servern oder von benachbarten Thin-Clients mit Windows- und Linux-Betriebssystemen booten können.

10.1.3 *Linux/Unix-Server und Workstations*

Auf Linux/Unix-Rechnern laufen Open Source-RDP-Clients, lizenzfreie ICA-Clients von Citrix und lizenzfreie AIP-Clients von Tarantella. 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.

10.1.4 *Diskless Linux-Geräte mit Flash-ROM*

Eine besonders sparsame und stabile Form von Linux-Geräten verzichtet auf Fest- und Wechsellplatten und Diskettenlaufwerke und bootet Linux von Flash-ROMs. Solche Lösungen bieten u. a. die Thin-Client-Hersteller Esesix, IGEL, NCD, Neoware, Siemens und Wyse zu Preisen ab ca. 250 €.

Diese Lösung steckt übrigens weltweit in vielen hunderttausenden von SetTop-Boxen für den Internet-Zugang über funktionsbeschränkte Netscape/Mozilla-Browser und Fernsehgeräte als Bildschirmersatz und in digitalen Receivern für terrestrisches und Satelliten-Fernsehen/Rundfunk.

Solche Thin-Clients sind einfach einzurichten und mit Fern-Administrationslösungen zentral administrierbar. Sie bieten ein komfortables Setup, das nur nach allgemeinverständlichen Daten wie Adresse des Servers, IP-Adresse oder DHCP und Endbenutzer-Sprache fragt.

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 (<http://www.bootix.com>), und zahlreichen freien Linux-Projekten, z. B. von Heise, von der Uni Göttingen oder – wie hier dargestellt – vom Linux-Terminal-Server-Projekt (LTSP). Als Hardware braucht man nur einen 486er PC ab 16 MB RAM und eine passende Netzwerkkarte mit PXE/Bootprom für unter 20 Euro. Zwar verbringen Systemverwalter zunächst mehr Zeit mit dem Einrichten, können aber später alles zentral auf dem Boot-Server pflegen. Während die zuvor erwähnten Lösungen mit Flash-ROMs auch Endanwender ohne Computerkenntnisse zum Laufen bringen, müssen hier erst Systemverwalter serverseitig fleißig sein. Sie haben kaum mehr zu tun, um 100 Diskless PCs einzurichten, als bei zwei 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 € Materialkosten funktionsfähige Endgeräte ohne Festplatte und Diskettenlaufwerk. Zu den 20 € kommen ggf. noch Kosten für neue Grafik- und Soundkarten hinzu.

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. 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, 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 oben beschriebene Flash-ROM-Technik.

10.2 Linux-Clients mit lokaler Installation

In vielen Unternehmen, Behörden, Bildungseinrichtungen und Klein- und Heimbüros gibt es noch unzählige Pentium-Computer mit stabiler Hardware, die für die Anforderungen der aktuellen Ressourcen-hungrigen Windows-Abarten einfach zu behäbig sind.

Viele dieser Geräte kann man noch viele Jahre lang hervorragend an Benutzer-Arbeitsplätzen als X-Terminals einsetzen, um darauf serverbasierte Linux- und Windows-Anwendungen zu nutzen. Wenn die eingebaute Grafikkarte weniger als 1024×786 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.

Der auf dieses Unterkapitel folgende Abschnitt 10.3 beschreibt, wie Sie diese Altgeräte ohne lokale Festplatten von Linux-Servern mit installiertem LTSP booten und verwenden.

Dieses Kapitel beschreibt, wie Sie einen Linux-Server und die Festplatte des Uralt-Pentium-PCs vorbereiten müssen, um die bewährten Geräte weiterhin wenigstens als X-Terminals nutzen zu können:

Auf dem Server müssen Sie das X-System so konfigurieren, dass es Anfragen von anderen Rechnern wie diesen Client-PCs zulässt.

Auf den Client-Rechnern müssen Sie ein minimales Linux-System mit X-System installieren, welches eine Verbindung mit dem Linux-Server aufnehmen kann.

10.2.1 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 bekommen.

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

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

```
/etc/X11/xdm/Xservers
# $XConsortium: Xserv.ws.cpp,v 1.3 93/09/28 14:30:30
# gildea Exp $
#
# $XFree86: xc/programs/xdm/config/Xserv.ws.cpp,v 1.1.1.1.12.2
# 1998/10/04 15:23
#
# Xservers file, workstation prototype
#
# 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. Each X terminal line should
```

```
# look like:
#
#     XTerminalName:0 foreign
#
# Note: The vt07 is required to start the local X server on the
# virtual console 7. This avoids conflicts with gettys of
# /etc/inittab.
#
#:#0 local /usr/X11R6/bin/X :0 vt07
```

Dazu deaktivieren Sie die fett hervorgehobene Zeile 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 Displaymanager unterbinden.

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

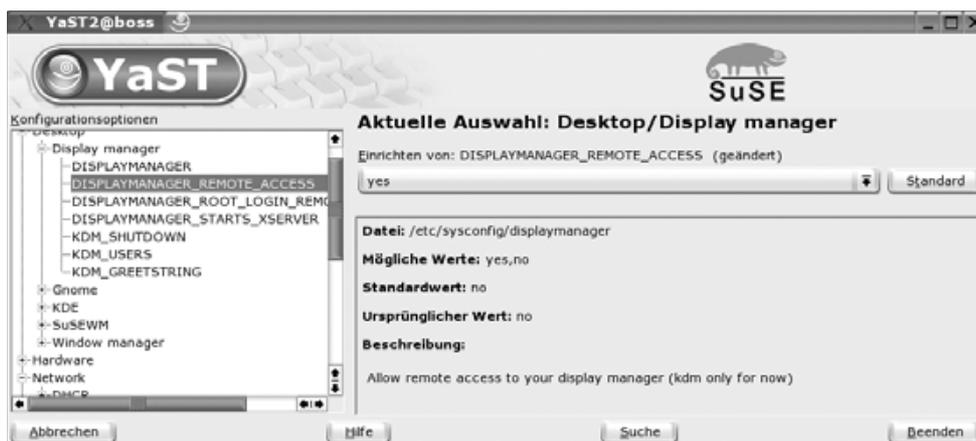


Abbildung 10.1: Displaymanager-Einstellungen in YaST

Mit jedem

```
init 5
```

wechseln Sie in den Runlevel 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.

10.2.2 Installation auf dem Client

Bei der hier beschriebenen administratorfreundlichen Lösung arbeiten die Benutzer an Linux-Endgeräten, die nur Sitzungen auf dem Linux-Terminalserver darzustellen brauchen. 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 *xmssc* aus der Serie *System • X11* installieren. Die anderen Auswahlmöglichkeiten wie Büroanwendungen, Dokumentation und KDE sind nicht notwendig.

Tipp: Auf älteren Rechnern mit weniger als 64MByte Hauptspeicher ist auch die Linux-Installation sehr langsam. Bauen Sie einfach die Festplatte dieses Computers in ein aktuelles Gerät und installieren 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 ausmachen.

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

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 klappt, 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 programm 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 hervorgehobenen Zeile aktivieren Sie den Eintrag für die siebte Konsole tty7. Diese Änderungen treten in Kraft, sobald Sie

```
telinit q
```

aufrufen. Danach sollte das grafische Anmeldefenster starten.

Wenn Sie Ihre Arbeitsplatz-PCs statt von lauten lokalen Festplatten 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 Sie bitte zum übernächsten Unterkapitel 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.

10.3 Diskless Linux-Geräte mit Boot-Prom einrichten

Dieser dritte Teil des Kapitel beschreibt, wie Sie plattenlose PCs von einem Boot-Server starten, um sie dann an Linux- oder Windows-Terminalservern zum Darstellen von ICA-, RDP-, AIP- 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.

10.3.1 Überblick

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

- Das Bootrom initialisiert die Netzwerkkarte.
- Das Bootrom sendet eine Bootp-Anfrage an den DHCP-Server.
- 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.
- Der Client sendet eine TFTP-Anfrage an den Boot-Server. (TFTP: Trivial File Transport Protocol).
- Der TFTP-Server beantwortet die Anfrage und sendet dem Client den Linux-Kernel.
- Der Client bootet jetzt Linux. Der Linux-Kernel sendet eine Bootp-Anfrage an den Server.
- Der Server beantwortet diese Anfrage. Mit diesen Daten konfiguriert der Linux-Kernel des Clients sein Netzwerkinterface und setzt den Rechnernamen.
- Der Linux-Client versucht, sein Root-Filesystem per NFS zu mounten.
- Der NFS-Server exportiert das angeforderte Verzeichnis an den Client.
- Der `init`-Prozess auf dem Client startet.
- Auf dem Client startet der X-Server. Er sendet XDMCP-Anfragen an den Server (XDMCP: X Display Manager Control Protocol).
- Der XDM-Server beantwortet diese Anfragen. Der Client zeigt den Begrüßungsbildschirm des XDM-Servers mit dem Login (XDM: X Display Manager).
- Benutzer können sich auf dem Server einloggen. Ihnen steht jetzt ihre X-Windows-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-Terminalserver (s. oben) zu arbeiten.

10.3.2 Benötigte Softwarekomponenten

Auf einem Linux-Server benötigt man dazu 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. Die weitere Software können Sie von der Website des Linux-Terminal-Server-Projekts beziehen:

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

- ltsp-3.0-de.pdf
- ltsp_core-3.0.9-0.i386.rpm
- ltsp_kernel-3.0.10-0.i386.rpm
- ltsp_x_core-3.0.4-0.i386.rpm
- ltsp_x_fonts-3.0.0-0.i386.rpm
- pxestuff-3.0.5-i386.tgz
- ltsp_initrd_kit-3.0.10-i386.tgz
- Die Datei ltsp-3.0-de.pdf enthält die vollständige Dokumentation. Diese sollten Sie bei Ihrer Installation zu Rate ziehen.
- Das Paket ltsp_core-3.0.9-0.i386.rpm enthält das Root-Filesystem der Clients und
- das Paket ltsp_kernel-3.0.10-0.i386.rpm den Linux-Kernel für die Clients. Dieser Kernel ist modular aufgebaut. Er lädt eine sogenannte »Initial Ram-Disk«. Diese enthält die Treiber für Netzwerkkarten sowie ein Skript zur automatischen Erkennung der Karte und zum Laden der Treiber. Sollte der Treiber für die Netzwerkkarten ihrer Clients fehlen, müssen Sie den Kernel neu kompilieren. Wenn Sie unnötige Arbeit vermeiden wollen, beschaffen Sie sich lieber Netzwerkkarten, für die Sie fertige Treiber bekommen.
- Die Pakete ltsp_x_fonts-3.0.0-0.i386.rpm und ltsp_x_core-3.0.4-0.i386.rpm enthalten die Version 4 des X-Window-Systems XFREE86. Sollte Ihre Grafikkarte nicht mit XFREE86-4 zusammenarbeiten, müssen Sie noch angepasste X-Server von der Homepage des LTSP-Projekts laden.
- Dateien im Paket pxestuff-3.0.5-i386.tgz ermöglichen Booten mit PXE-Boot-Roms.
- Das Paket ltsp_initrd_kit-3.0.10-i386.tgz enthält die dafür notwendigen Kernel- und Initrd-Dateien für den neuesten Kernel.

10.3.3 Softwarekomponenten installieren und Systemdateien anpassen

Überprüfen Sie zunächst, ob folgende Pakete installiert sind:

- tftp bzw. atftp
- dhcp-server
- nfs-utils bzw. nfs-server

Dies geschieht mit dem Befehl:

```
rpm -q Paketname
```

Ein Beispiel für eine derartige Abfrage wäre:

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

Vom Programm rpm bekommt man dann die Ausgabe:

```
nfs-utils-1.0.1-109
```

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

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

Damit die Serverprogramme 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, brauchen Sie nur die folgenden zwei Zeilen einzugeben:

```
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.

Installieren Sie nun zunächst das Root-Filesystem von ltsp mit dem Befehl:

```
rpm -ivh ltsp_core-3.0.9-0.i386.rpm
```

Anschließend richten Sie die weiteren Rpm-Pakete mit dem folgenden Befehl ein:

```
rpm -ivh rpm-Datei
```

Entpacken Sie nun das Archiv `pxe-stuff-3.0.5-i386.tgz`:

```
tar -xzf pxe-stuff-3.0.5-i386.tgz
```

Kopieren Sie die Datei `pxelinux.0` sowie das Verzeichnis `pxelinux.cfg` aus dem neuen Verzeichnis `pxestuff` nach `/tftpboot/lts`

```
cp -r pxestuff/pxelinux.0 pxestuff/pxelinux.cfg /tftpboot/lts/
```

Entpacken Sie jetzt das Archiv `ltsp_initrd_kit-3.0.10-i386.tgz` und kopieren Sie die Dateien `bzimage-2.4.21-ltsp-1` und `initrd-2.4.21-ltsp-1.gz` aus dem neuen Verzeichnis `ltsp_nitrd_kit` nach `/tftpboot/lts`:

```
tar -xzf ltsp_initrd_kit-3.0.10-i386.tgz
cd ltsp_initrd_kit
cp bzimage-2.4.21-ltsp-1 initrd-2.4.21-ltsp-1.gz /tftpboot/lts/
```

Ändern Sie dann die Datei `/tftpboot/lts/pxelinux.cfg/default` wie folgt:

```
prompt=0
label linux
  kernel bzImage-2.4.21-ltsp-1
    append init=/linuxrc rw root=/dev/
      ↵ ram0 initrd=initrd-2.4.21-ltsp-1.gz
```

Diese Datei gibt den Kernel, notwendige Kernelparameter sowie die `initrd` für das Booten mit `Pxe-Bootroms` an.

Nun sind noch einige Details anzupassen:

Wechseln Sie in das Verzeichnis `/opt/ltsp/install_scripts` und editieren Sie die Datei `CONFIG`. Passen Sie die Parameter `IP_NETWORK_BASE`, `IP_NETWORK`, `IP_BROADCAST`, `IP_NETMASK` an Ihre Gegebenheiten an.

```
#!/=====
# Global settings...
#!/=====
#
#-----
# LTSP files and directories
#-----
#
LTSP_DIR=/opt/ltsp
```

```

TFTP_DIR=/tftpboot

TMPL_DIR=${LTSP_DIR}/templates
LOGFILE=/tmp/ltsp.install.log
LIBFILE=/tmp/ltsp.libs
SWAP_DIR=/var/opt/ltsp/swapfiles
ROOT_DIR=${LTSP_DIR}/i386
#
#-----
# Network stuff...
#-----
#
IP_NETWORK_BASE=192.168.1
IP_NETWORK=${IP_NETWORK_BASE}.0
IP_SERVER=${IP_NETWORK_BASE}.1
IP_BROADCAST=${IP_NETWORK_BASE}.255
IP_NETMASK=255.255.255.0
#
#
#-----
# others...
#-----
VERSION=3.0.9
TODAY=`date`

export LTSP_DIR
export SWAP_DIR
export ROOT_DIR
export TMPL_DIR
export LOGFILE
export LIBFILE
export IP_NETWORK_BASE
export IP_NETWORK
export IP_SERVER
export IP_BROADCAST
export IP_NETMASK
export VERSION
export TODAY

```

Geben Sie danach ein:

```
sh suse-8.2.sh
```

Wechseln Sie danach in das Verzeichnis `/opt/ltsp/templates` und geben Sie dort ein

```
./ltsp_initialize
```

Folgen Sie den Anweisungen und geben Sie ein: A
gefolgt von der Eingabetaste.

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

```
### LTS-begin ###
#
# The lines between the 'LTS-begin' and the 'LTS-end' were added
# on: Wed Jul 30 12:31:49 CEST 2003 by the ltsp
# installation script.
# For more information, visit the ltsp homepage
# at http://www.ltsp.org
#
/opt/ltsp/i386          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)
#
# The following entries need to be uncommented if you want
# Local App support in ltsp
#
#/home                192.168.1.0/255.255.255.0
# (rw,no_root_squash,sync)
#
### LTS-end ###
# See the exports(5) manpage for a description of the syntax
# of this file.
# This file contains a list of all directories that are to be
# exported to other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages
# for details on how make changes in this file effective.
```

Falls Sie das Paket `nfs-server` installiert haben, entfernen Sie die Schlüsselworte `sync` bzw. `async`.

Diese Einstellungen exportieren das Root-Filesystem, das für die Clients *read-only* ist, setzen die Kennung von *root* aber nicht auf *nobody* um.

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 Datei `hosts` einfügen.

Damit der XDM-Server automatisch startet, ändern Sie mit dem Runlevel-Editor der YaST Runlevel-Konfiguration auf:

```
5: Voller Mehrbenutzerbetrieb mit Netzwerk und XDM.
```

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 option-128 code 128 = string;
option option-129 code 129 = text;
option space PXE;
option PXE.mtftp-ip code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
option PXE.discovery-control code 6 = unsigned integer 8;
option PXE.discovery-mcast-addr code 7 = ip-address;

class "PXEClient:" {
match if substring (option vendor-class-
identifier, 0, 10) = "PXEClient:";
option vendor-class-identifier "PXEClient";
filename "/lts/pxelinux.0";
option PXE.mtftp-ip 0.0.0.0;
# Der folgende Parameter muesste eigentlich heissen:
# vendor-option-space PXE;
# In diesem Release des Dhcp- Servers heisst er:
vendor-unknown-space PXE;
# Nach einem Update sollten Sie ueberpruefen,
# ob der dhcpd mit diesem Parameter noch arbeitet.
# Nehmen Sie sonst den Parameter:
# vendor-option-space PXE;
}

class "Etherboot" {
match if substring (option vendor-class-identifier, 0, 9) =
↳ "Etherboot";
option vendor-encapsulated-options
↳ 3c:09:45:74:68:65:72:62:6f:6f:74:ff;
filename = "/lts/vmlinuz-2.4.21-ltsp-1";
}

option domain-name "lokales-netz.de";
option domain-name-servers 192.168.1.1;
option netbios-name-servers 192.168.1.1;
option subnet-mask 255.255.255.0;
option routers 192.168.1.1;
ddns-update-style none;
default-lease-time 1209600;
max-lease-time 12096000;
#
option root-path "192.168.1.1:/opt/ltsp/i386";
#
```

```

subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.40 192.168.1.60;
}

group{
  use-host-decl-names      on;
  option log-servers        192.168.1.1;

  host hh2-160 {
    hardware ethernet      00:50:56:40:8e:8f;
    fixed-address           192.168.1.160;
    # filename               "/lts/pxelinux.0";
  }
  host hh2-159 {
    hardware ethernet      00:90:f5:1a:a8:7c;
    fixed-address           192.168.1.159;
    # filename               "/lts/vmlinuz-2.4.21-ltsp-1";
  }
}

```

Die obige host-Deklarationen zeigt Beispiele für zwei verschiedene Rechner. Abhängig von der Adresse ihrer Netzwerkkarte erhalten sie vom dhcp-Server eine IP-Adresse zugewiesen. 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 Rechnern 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, braucht man hier nicht einzutragen.

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          = 192.168.1.1
  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
LOCAL_APPS      = N
RUNLEVEL        = 5

#-----
#
# Example of specifying X settings for a workstation
#
#[ws001]
#   XSERVER      = auto
#   LOCAL_APPS   = N
#   USE_NFS_SWAP = N
#   SWAPFILE_SIZE = 48m
#   RUNLEVEL    = 5

#[ws002]
#   XSERVER      = XF86_SVGA
#   LOCAL_APPS   = N
#   USE_NFS_SWAP = N
#   SWAPFILE_SIZE = 64m
#   RUNLEVEL    = 3

#
# ws004 is my virtual workstation running in a VMware session
#
#[ws004]
#   DNS_SERVER   = 192.168.1.1
#   XSERVER      = auto
#   X4_BUSID     = "PCI:0:15:0"
#   X_MODE_O     = 800x600
#   LOCAL_APPS   = N
#   USE_NFS_SWAP = N
#   SWAPFILE_SIZE = 64m
#   RUNLEVEL    = 5

#-----
#
# Example of a workstation configured to load some modules
#
#[ws001]
#   MODULE_01    = agpgart.o# This is for i810 video
#   MODULE_02    = uart401.o

```

```

# MODULE_03      = sb.o io=0x220 irq=5 dma=1
# MODULE_04      = opl3.o
#-----
#
# Example of ws001 configured for local apps
#
#[ws001]
# LOCAL_APPS      = Y
# LOCAL_WM        = Y
# NIS_DOMAIN      = ltsp
# NIS_SERVER      = 192.168.0.254
#-----
#
# Example of a serial printer attached to /dev/ttyS1 on
# workstation ws001
#
#[ws001]
# PRINTER_0_DEVICE = /dev/ttyS1
# PRINTER_0_TYPE   = S# P-Parallel, S-Serial
# PRINTER_0_PORT   = 9100# tcp/ip port: defaults to 9100
# PRINTER_0_SPEED  = 9600# baud rate: defaults to 9600
# PRINTER_0_FLOWCTRL = S# Flow control: S-Software
# (XON/XOFF),
# # H-Hardware (CTS/RTS)
# PRINTER_0_PARITY = N# Parity: N-None, E-Even, O-Odd
# # (defaults to 'N')
# PRINTER_0_DATABITS = 8# Databits: 5,6,7,8 (defaults to 8)

```

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

```
init 1
```

und nach einem erneuten Einloggen als root:

```
init 5
```

10.3.4 Erste Tests

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

laden. Falls Sie zum Testen nur eine bootfähige 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 Bootrom starten wollen, sollten Sie

- Bootroms mit den Treibern des Etherboot-Paketes über das Internet bestellen oder
- sich einen Eprom-Brenner zulegen und die Bootroms selber 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 Bootrom 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 Bootroms in Kbyte an, die Eprom-Hersteller in Kbit. Für ein 32Kbyte Booteprom benötigen Sie also ein 256Kbit Eprom.

10.4 Win4Lin Terminalserver

Systemadministratoren, die Netze mit Windows-Arbeitsplätzen betreiben, klagen über instabile Windows-Installationen, begründet durch die offenkundigen Qualitätsmängel der Anwendungen und des Betriebssystems und die Möglichkeit für Benutzer, die System- und Anwendungskonfiguration zu verändern.

Die lässt sich bei PC-basierten Windows-Systemen nicht hinreichend mühelos durch Einschränkungen in den Systemeinstellungen verbessern.

Insbesondere wenn, wie in Bildungseinrichtungen oder Internet-Cafes, viele verschiedene Benutzer an einzelnen PCs arbeiten, hinterlassen alle Benutzer irgendwelche Veränderungen.

Wie hier schon eingangs im ersten Unterkapitel beschrieben, helfen sich Systemverantwortliche mit Lösungen wie

- Wächterkarten, die auf den Desktop-PCs bestimmte System-Zustände einfrieren (siehe <http://www.dr-kaiser.de>),
- Booten der Desktop-Rechner von zentralen Boot-Servern mit standardisierter Konfiguration (siehe <http://www.goe.net>),

- Terminal-Dienste mit Windows 2000 oder Windows 2003 Server (siehe <http://www.microsoft.com> und Stefan Herkert, Bernd Kretschmer, Windows 2000 Terminaldienste, Addison Wesley, 2000) oder
- Zentrale Linux-Terminal-Dienste mit darauf aufsetzenden zentralen Windows-Anwendungsschnittstellen für Linux-Clients (siehe Crossover Office).
- Zentrale Linux-Terminal-Dienste mit darauf aufsetzender zentraler Windows-Middleware für Linux-Clients.
- Die letztgenannten Einzelplatz- und Mehrplatz-Lösungen von Netraverse (<http://www.netraverse.com>) zum Versorgen von Linux-Arbeitsplätzen mit Windows 98 und darauf aufsetzenden Windows-Anwendungen hat die Redaktion über zwei Jahre erprobt und als stabil und betriebssicher klassifiziert:
- Bei der Lösung für einzelne Arbeitsplätze setzt die Einzelplatz-Lösung Netraverse Win4Lin auf Linux auf, darauf kommen eine Windows 98-Installation und darauf wie bei nativem Windows die Windows-Anwendungen.
- Bei der Lösung für viele Arbeitsplätze setzt die Mehrplatz-Lösung Netraverse Win4Lin Terminalserver (früher NSSE) auf einem Linux-Terminal-Server auf, darauf für jeden Anwender oder Arbeitsplatz eine (clonbare) Windows 98-Installation und darauf für jeden Anwender oder Arbeitsplatz wie bei nativem Windows clonbare Windows-Anwendungen.

Diese Lösungen von Netraverse sind eine Weiterentwicklung der schon sehr lange im Unix-Umfeld eingesetzten stabilen Ablaufumgebungen für Windows auf Unix, ergrauenden Unix-Gurus auch bekannt unter dem Namen SCO Merge.

In den sehr ausführlichen Tests der Redaktion laufen Windows-Anwendungen stabiler als direkt auf Windows betriebene Anwendungen und im Einzelplatz-Betrieb sind sie obendrein noch schneller als auf baugleichen direkt mit Windows betriebenen PCs.

Praktisches Einsatzgebiet der Einzelplatzlösung *Win4Lin* sind einzelne Windows-Dienste auf Linux-PCs und der Terminalserver-Lösung *Win4Lin Terminalserver*, dagegen sind Windows-Dienste für viele Benutzer geeignet, die ihre Anwendungen über Linux-Terminal-Dienste beziehen. Der folgende Text bezieht sich auf die Mehrbenutzerlösung, auch wenn manchmal von Win4Lin die Rede ist.

Von den Sicherheitsrisiken, wirtschaftlichen Abhängigkeiten und prinzipiellen Bedenken um die Zukunft unserer Weltkultur, Anwendern einen Zugriff auf Windows und auf Windows-Anwendungen mit nicht veröffentlichten proprietären und durch Copy-»Rechte« geschützten Dateiformaten zu ermöglichen, wird der folgende Text schweigen.

Lesen Sie hier, wie Sie Linux-Server und Linux-Client konfigurieren müssen, um Windows mit der serverbasierten Installation von Netraverse Win4Lin Terminal Servern an mehreren Linux-Clients nutzen zu können.

Dazu sind folgende Schritte notwendig:

- Installation von Netraverse Win4Lin Terminalserver
- Vor-Installation und Konfiguration von Windows auf NSSE
- Vor-Installation und Konfiguration von Windows-Anwendungen auf NSSE
- Clonen und Anpassen der benutzer- oder arbeitsplatzbezogenen Installation

10.4.1 *Installation von Win4Lin Terminalserver*

Wie eingangs ausgeführt, ist der Netraverse Win4Lin Terminalserver eine stabile Technologie, um von einem Linux-Server aus Windows-Sitzungen an Linux-Clients zu verteilen.

Die notwendige Software für die Installation auf dem Linux-Server können Sie von <http://www.netraverse.com> fernladen oder bei Netraverse oder deren deutschem Distributor IXSOFT (<http://www.ixsoft.de>) gegen eine Handhabungsgebühr auf CD beziehen. In beiden Fällen bekommen Sie einen kostenlosen Lizenzschlüssel für eine Testperiode von 30 Tagen. Wenn Sie mit dem Produkt glücklich sind, können Sie Lizenzen für ca. 100 € pro Benutzer oder Desktop erwerben.

Falls die CD-Distribution nicht ganz aktuell ist, können Sie diese an der Webseite von Netraverse aktualisieren.

Die Installation der Software ist wirklich sehr ausführlich und klar dokumentiert.

Das Installieren von Win4Lin und NSSE erfordert einen angepassten Linux-Kernel. Netraverse stellt auf seinen Webseiten passende Kernel für die wichtigsten Distributionen zur Verfügung, u. a. auch SMP-Kernel für Computer mit mehreren Prozessoren (Abbildung 10.2).

Sie müssen nur darauf achten, dass Ihr Linux-Server vor der Installation einen Standard-Kernel der jeweiligen Distribution nutzt und keine davon abweichende aktualisierte Version.

Das Installationsprogramm von Win4Lin wählt automatisch den richtigen Kernel aus. Beim Aufruf des Installers bietet dieser zuerst an, sich selbst zu aktualisieren. Nehmen Sie dieses Angebot bitte an (Abbildung 10.3).

Danach kommt ein Dialog für die Online-Registrierung. Hierfür kommen Sie nur nach dem Eintragen eines gültigen Lizenz-Codes weiter.

For instructions on extracting the NeTraverse kernel patches from a source RPM, click here.

NeTraverse-Enabled Kernel Table

Files	Compatible with:					
	Win4Lin	NSSE/ 3.0	3.0.6	3.0.7	4.0+	Win4Lin Server 1.1 1.1.6 1.1.7 2.0+
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.95-01.i686.rpm (1.4M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.95-01.src.rpm (86.6k) Last Updated: 06 Aug 2003			✓	✓		
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.95-01.i586.rpm (1.3M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.95-01.src.rpm (86.6k) Last Updated: 06 Aug 2003			✓	✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.86-01.i686.rpm (1.4M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.86-01.src.rpm (86.4k) Last Updated: 20 Jun 2003				✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.86-01.i586.rpm (1.3M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.86-01.src.rpm (86.4k) Last Updated: 20 Jun 2003			✓	✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.62-01.i686.rpm (1.4M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.62-01.src.rpm (85.6k) Last Updated: 13 May 2003				✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.62-01.i586.rpm (1.3M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.62-01.src.rpm (85.6k) Last Updated: 13 May 2003			✓	✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.100-01.i686.rpm (1.4M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.100-01.src.rpm (86.8k) Last Updated: 15 Aug 2003				✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.100-01.i586.rpm (1.3M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.100-01.src.rpm (86.6k) Last Updated: 15 Aug 2003			✓	✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.01-01.i686.rpm (1.4M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.01-01.src.rpm (85.6k) Last Updated: 23 Apr 2003				✓	✓	
Kernel-Win4Lin3-SuSE8.2smp_2.4.20.01-01.i586.rpm (1.3M) Kernel-Win4Lin3-SuSE8.2smp_2.4.20.01-01.src.rpm (85.6k) Last Updated: 23 Apr 2003			✓	✓	✓	
Kernel-Win4Lin3-SuSE8.2_2.4.20.96-01.i586.rpm (1.3M) Kernel-Win4Lin3-SuSE8.2_2.4.20.96-01.src.rpm (87.1k) Last Updated: 06 Aug 2003			✓	✓	✓	

Abbildung 10.2: NeTraverse – Vorbereitete Kernel

Nach der Online-Registrierung untersucht der Netraverse-Installer Ihren Linux-PC und lädt die notwendigen Pakete von der Netraverse Website.

Das Installationsprogramm lädt die Pakete, installiert sie und aktualisiert den Kernel und die Bootkonfiguration. Sie brauchen hier keine besondere Angst zu haben, da das Boot-Menü danach sowohl den neuen, als auch den alten Kernel anbietet.

Um den neuen Kernel zu aktivieren, fordert der Installer Sie auf, den Rechner neu zu booten.

Nach dem Neustart des Linux-Servers starten Sie den Installer lokal und dann die systemweite Windows-Vorinstallation.

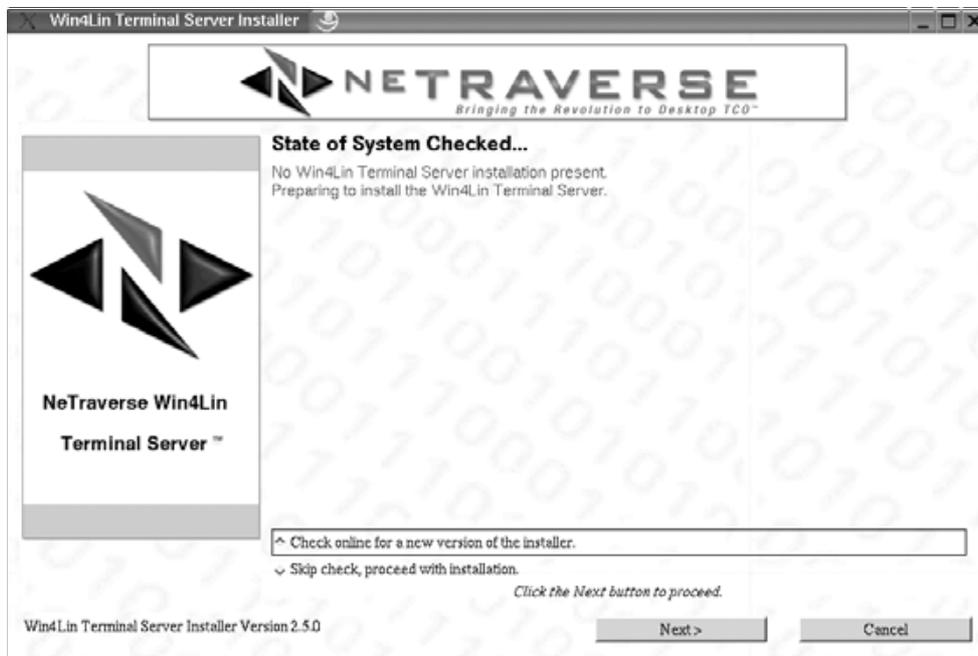


Abbildung 10.3: NeTraverse – Aktualisierung Installer

Sie benötigen dazu eine Produkt-CD und Boot-Diskette sowie Lizenzschlüssel von Microsoft Windows 98. Das Installationsprogramm liest beide Medien ein und speichert sie auf der Festplatte des Linux-Servers.

Im letzten Schritt installieren Sie Windows für alle Benutzer anhand der bereits eingelesenen Medien. Hieraus ergibt sich, dass alle Windows-Installationen mit der gleichen Windows-Version arbeiten müssen.

10.4.2 Windows-Installation für die Benutzer

Wenn der Server über genügend Plattenkapazität und Ihre Organisation über genügend Lizenzen verfügt, können Sie für jeden Benutzer eine eigene Windows-Installation erstellen. Das können die Benutzer auch selbst machen, indem sie unter KDE in einer Shell

```
win
```

eingeben. Wenn das System dann erkennt, dass für einen Benutzer noch kein Windows 98 installiert ist, bietet es an, dieses jetzt einzurichten. Da die Installation auf den bereits eingespielten Windows-Quelldateien aufbaut, brauchen Benutzer dafür keinerlei Medien, sondern nur den Lizenzschlüssel einer Windows 9.x-Lizenz.



Abbildung 10.4: NeTraverse – Windows Benutzerinstallation

Bei der Installation legt Win4Lin drei Verzeichnisse im Home-Verzeichnis des Benutzers an:

win	hier liegt die eigentliche Windows-Installation
.merge	hier liegen Konfigurationseinstellungen
mydata	Verzeichnis für Benutzerdaten, erreichbar als d:

Wenn Sie die nicht besonders zeitraubende Installation für jeden Benutzer ablaufen lassen wollen, können Sie die erste Installation einfach kopieren und anschließend für jede Kopie die Voreinstellung `servername-benutzername` des Rechnernamens ändern.

Für das bequeme Sichern und Zurückspielen der Installationen liefert NeTraverse auf der CD im Verzeichnis `\misc\tools` das kleine Shell-Skript `Nssetemplate`. Dieses kann Installationen gepackt als sog. Templates ablegen. Die Templates kann man dann einfach wieder mit diesem Ordner in ein Benutzerverzeichnis zurückkopieren.

10.4.3 Rechnerbezogene Windows-Installation

In Bildungseinrichtungen, in denen es erheblich mehr Lernende als Computer-Arbeitsplätze gibt, kann man Windows 98 arbeitssparend für jeden Linux-Client-PC statt für jeden Benutzer einrichten.

Prüfen Sie bitte in Microsofts Lizenzbedingungen, ob dies mit den Einzel- oder Campus-Lizenzen Ihrer Organisation für Windows 98 und Windows-Anwendungen erlaubt ist. Wenn ja, gehen Sie bitte wie folgt vor.

Installieren Sie Microsoft Windows 98 und alle erforderlichen Windows-Anwendungsprogramme für einen beliebigen Nutzer auf dem Linux-Server. Kopieren Sie nach der Installation das win-Verzeichnis in eine genügend große Partition (z. B. /opt/win4lin/) und benennen Sie diese so wie einen der Linux-Client-Rechner. Anschließend erstellen Sie für jeden Linux-Client eine Kopie dieses Verzeichnisses, die genauso heißt wie diese Rechner (ohne Domainanteil).

Da Windows »meckert«, wenn bei bestimmten Systemdateien das Datum verändert ist, müssen Sie beim Kopieren die Eigenschaften und insbesondere die Herstellungsdaten/-zeiten der Dateien bewahren. Dazu sollten Sie stets mit dem Parameter `-a` (`cp -a`) kopieren

```
cp -a rechner1 rechner2
```

Nun brauchen Sie noch ein Skript, das

- bei der Anmeldung eines Benutzers feststellt, ob es für den Client-Rechner, an dem er mit Windows arbeiten will, eine Installation gibt und
- wenn ja, im Home-Verzeichnis des Benutzers einen Link namens `win` auf dieses Verzeichnis anlegt.

Mit »geliehenen« Rechten des Systemverwalters `root` muss sich der Benutzer dann noch das Installationsverzeichnis übereignen lassen, um danach Windows zu starten.

Das erste Perl-Skript namens `windows` können Sie im Verzeichnis `/usr/local/bin` speichern. Es ermittelt den Benutzernamen des Aufrufenden und den Namen des Rechners, von dem er es aufruft. Wenn es für den ermittelten Rechner eine Windows-Installation gibt, dann legt es im Home-Verzeichnis einen Link auf das zugehörige Verzeichnis an.

```
/usr/local/bin/windows
```

```
#!/usr/bin/perl -w
# Uwe Debacher und Bernd Burre 2003
# die Verzeichnismomanipulationen erfordern Root-Rechte
```

```

# wo liegen die Windows-Installationen
$winpfad="/opt/win4lin";

# Benutzernamen ermitteln
$uid=$<;
$username=(getpwuid($uid))[0];
if ($username=~ /^(.+)$/) {
    $username = $1;
}
if ($username eq "") {
    die "Fehler: Ich kann den Benutzernamen nicht ermitteln\n";
}

# der volle Hostname steckt in der letzten Spalte
@last= `usr/bin/last -n 1 -a $username`;
$host= (split (/s+/, $last[0]))[9];
if ($host eq "") {
    die "Fehler: lokales Login auf dem Server\n";
}

# den Domainteil verwerfen
$host = (split /\./, $host)[0];
if ($host eq "") {
    die "Fehler: ich kann den Rechnernamen nicht ermitteln\n";
}

if ($host eq "" || $host eq "wtmp") {
    die "Fehler: Ich kann den Anmelde-Rechner nicht ermitteln\n";
}

# ist fuer diesen Rechner eine Installation vorhanden??
if (!-e "$winpfad/$host") {
    die "Fehler: das Verzeichnis $winpfad/$host fehlt\n";
}

# es darf keine Datei und kein Verzeichnis win geben
if (-e "win" && !-l "win") {
    die "Fehler: Ich kann den Link win nicht anlegen, es gibt eine
solche Datei\n";
}

# vorhandene win-Links müssen weg
if (-l "win") {
    print "lösche vorhandenen Link win\n";
    unlink ("win");
}

# den Link win anlegen

```

```

print "Lege Link win an\n";
symlink("$winpfad/$host", "win");

# fuer die Links werden Rootrechte benoetigt
system "wrapper";

# Erfolg testen
if (!-o "win/autoexec.bat") {
    die "Fehler: Die Eigentumsverhältnisse stimmen nicht\n";
}

# weiter mit den normalen Rechten
print "Starte Windows\n";
exec "win"

```

Bevor das Skript den Link `win` anlegt, entfernt es einen eventuell schon vorhandenen Link. Wenn es ein Verzeichnis `win` findet, dann bricht es mit einer Fehlermeldung ab.

Einen wichtigen Teil der Arbeit erledigt das zweite Skript, das Sie über ein kleines C-Programm namens *wrapper* starten.

Das zweite Programm benötigt Root-Rechte. Da man in Perl-Skripten nicht einfach `Setuid` setzen kann, muss ein kleines C-Programm das Perl-Skript starten. Bei diesem C-Programm muss man mit

```
chmod u+s wrapper
```

das `Setuid`-Bit setzen.

Das Perl-Skript muss die Windows-Installation dem aufrufenden Benutzer übereignen, damit Windows schreibenden Zugriff auf seine eigenen Dateien bekommt. Dazu muss es den Benutzernamen ermitteln und den Rechner, von dem aus es aufgerufen wurde.

```

/usr/local/bin/links.pl

#!/usr/bin/perl
# (c) Uwe Debacher und Bernd Burre März 2002

# den Pfad zu den Windows-Installationen bitte anpassen
$pfad="/opt/win4lin";

# wir brauchen Root-Rechte
$uid="$<";
$< = "$>";
$oldpath=$ENV{'PATH'};
$ENV{'PATH'}="/bin:/sbin:/usr/bin:/usr/sbin";

```

```

# Benutzernamen ermitteln
$username=(getpwuid($uid))[0];
if ($username=~ /^(.+)$/) {
    $username = $1;
}

# der volle Hostname steckt in der letzten Spalte
@last= `/usr/bin/last -n 1 -a $username`;
$host= (split (/\/s+/, $last[0]))[9];
if ($host eq "") {
    die "Fehler: lokales Login\n";
}

# den Domainenteil verwerfen
$host = (split /\./, $host)[0];
if ($host eq "") {
    die "Fehler: Ich kann den Rechnernamen nicht ermitteln\n";
}

# den Pfad zum Windowsverzeichnis montieren
$pfad=$pfad . '/' . $host;
if ($pfad=~ /^(.+)$/) {
    $pfad=$1
}

# wenn es den Pfad nicht gibt, dann stimmt was nicht
if ( !-d "$pfad" ) {
    die "Fehler: $pfad nicht gefunden\n";
}

# nur können wir endlich übereignen
print "Übereigne $pfad an $username\n";
system "/bin/chown -R $username $pfad";

```

Ein großer Teil des Skripts dient dazu die Sicherheitsanforderungen von Perl hinsichtlich von Setuid-Programmen zu erfüllen. Perl erwartet, dass die Eingabedaten überprüft werden. Weitere Hinweise hierzu finden sich in der Manpage perlsec.

Wenn Sie gern C-Quelltext lesen, finden Sie hier den kommentierten Code des Wrapper-Skripts.

```

/*
*/
#include <unistd.h>
#include <sys/types.h>
// compilieren: gcc -o wrapper script_suid.c
// danach:      chmod u+s wrapper
int main(argc,argv)

```

```

int argc;
char *argv[];
{
    extern char **environ;
    int c;
    c=execve("/usr/local/bin/links.pl",argv,environ);
    return c;
};
/* *** EOF *** */

```

Damit ist das System funktionsfähig und alle Benutzer können ihre Windows 98-Umgebung starten, indem Sie in einem Terminalfenster unter KDE *windows* eingeben.

Ein bisschen einmaliger Turnschuh-Sport bleibt Ihnen noch. Als Administrator müssen Sie einmal an jeden der Linux-Clients heran und den Namen für den zugehörigen virtuellen Rechner ändern, da diese nach dem Kopieren alle gleich lauten.

Mit etwas Handarbeit können Sie diese Arbeiten von einem Linux-Client aus erledigen, indem Sie die Links passend setzen und die Windows-Rechner nacheinander starten und konfigurieren.

Win4Lin verteilt die MAC-Adressen der virtuellen Netzwerkkarten automatisch jeweils in der Anmeldeihenfolge, um die Eindeutigkeit zu gewährleisten.

10.4.4 Feintuning mit KDE

Um Anwender erst gar nicht mit einem Terminalfenster zu konfrontieren, können Sie allen Benutzern ein Icon namens *Windows* auf ihre Desktops legen, mit dem sie den Emulator und *Windows* starten können.

Erstellen Sie dazu einen Desktop-Link in Form einer einfachen Textdatei.

```

[Desktop Entry]
Comment[de]=
Exec=/usr/local/bin/windows
Icon=HandOnWindowCloseIcon
MimeType=
Name[de]=Windoofs
Path=
ServiceTypes=
SwallowExec=
SwallowTitle=
Terminal=false
TerminalOptions=
Type=Application
X-KDE-SubstituteUID=false
X-KDE-Username=

```

Diese Datei muss für alle Benutzer im Verzeichnis KDesktop ihrer Home-Verzeichnisse stehen. Eventuell müssen Sie diese Verzeichnisse und ebenso die Verzeichnisse .merge und mydata erst noch erstellen.

Als Vorlage kopieren Sie die Verzeichnisse aus dem Home-Verzeichnis des Benutzers, für den Sie Windows installiert haben, in das Verzeichnis /etc/skel und geben Sie für alle Benutzer die Leserechte.

```
cp -r /home/dummyuser/KDesktop /etc/skel
chmod -R a+r /etc/skel/KDesktop
cp -r /home/dummyuser/.merge /etc/skel
chmod -R a+r /etc/skel/.merge
cp -r /home/dummyuser/mydata /etc/skel
chmod -R a+r /etc/skel/mydata
```

Beim Anlegen eines neuen Benutzers kopieren obige Befehle alle Dateien und Verzeichnisse unterhalb von /etc/skel in dessen Home-Verzeichnis. Sind Benutzer schon angelegt, müssen Sie diese Kopiervorgänge bei der nächsten Anmeldung veranlassen.

Dies ist eine herrlich typische Aufgaben für ein Anmeldescript. Bei Linux eignet sich hierfür das Skript /etc/profile, das Linux bei jeder Anmeldung ausführt. Individuelle Erweiterungen bringt man besser in der Datei /etc/profile.local unter.

```
/etc/profile.local
# Erweiterungen von Uwe Debacher und Bernd Burre für Win4Lin
#
if [ $UID -gt 499 ] ; then
    test -d ~/Desktop || cp -r /etc/skel/Desktop ~/
    test -e ~/Desktop/Windows.desktop || cp /etc/skel/Desktop/
└ Windows.desktop ~/Desktop
    test -d ~/.merge || cp -r /etc/skel/.merge ~/
    test -d ~/mydata || cp -r /etc/skel/mydata ~/
fi
```

Um Systembenutzer und Root auszunehmen, kopiert das Skript nur, wenn der Benutzer eine ID größer als 499 besitzt.

Für alle anderen Benutzer richtet es die notwendigen Dateien und Verzeichnisse ein, sofern sie noch nicht vorhanden sind.

10.4.5 Feintuning der Windows-Konfiguration

Bei der Windows-Middleware Netraverse Win4Lin Terminalserver sind einige Parameter, wie Hauptspeicher, Fenstergröße und Laufwerkszuordnungen konfigurierbar.

Die wichtigsten Parameter finden Sie im Home-Verzeichnis jedes Benutzers, im Verzeichnis `.merge`, in der Datei `win.cfw`.

```
# This file was generated by NeTraverse Server Standard Edition
# configuration utilities.
# Use the configuration utilities to make changes to this file.
session=win9x
version=520-1
autoexecglobal=true
autoexeclocal=true
autoexecprivate=
autofreeze=false
autozoom=false
com1=
com2=
configglobal=true
configlocal=true
configprivate=
customdev=
displaytype=auto
dosfont=small
drive=a,unix,HOME/,map_win95
drive=b,unix,HOME/,map_win95
drive=c,unix,HOME/win/,map_win95
drive=d,unix,HOME/,map_win95
drive=e,none
drive=f,none
drive=g,none
drive=h,none
drive=i,none
drive=j,unix,/var/win4lin/dosroot/,map_standard
drive=k,none
drive=l,none
drive=m,none
drive=n,none
drive=o,none
drive=p,none
drive=q,none
drive=r,none
drive=s,none
drive=t,none
drive=u,none
drive=v,none
drive=w,none
drive=x,none
drive=y,none
drive=z,none
```

```

ems=0
installcolormap=false
lpt1=none,0
lpt2=none,0
lpt3=none,0
maxcolors=high
memory=32
mouse=mouse
personaldrive=c
run=
scaledosgraphics=1
showmenukey=shift F12
startpersonal=true
startzoomed=false
unzoomkey=shift F12
windowresizewin=true
windowresizewin=true
windowsdir=windows
windowheight=680
windowsizeauto=false
windowmode=enhanced
windowspaging=false
windowwidth=1016
xcutandpaste=false

```

Der erste Teil der Datei betrifft Einstellungen zu den Laufwerken. Die Diskettenlaufwerke lassen sich hier leider nicht deaktivieren, aber zumindest auf Verzeichnisse umlenken, ähnlich wie c: und d:. Auf das Verzeichnis d: könnte man verzichten, nicht aber auf j:, das für Zugriffe auf die Windows-CD benötigt wird.

Der Speicher war nach der Installation auf 20 MByte eingestellt. 64 MByte ist der höchstmögliche Wert in der aktuellen Installation. Eventuell sind 32 MByte ein sinnvoller Kompromiss.

Die Größe für das Windowsfenster ist auf 800×600 Punkte voreingestellt. Auf einem Desktop mit 1024×768 Punkten scheint eine Fenstergröße von 1016×680 Punkten ein sinnvoller Wert zu sein. Das Fenster bleibt als solches erkennbar, ohne das irgendetwas verdeckt wird.



Abbildung 10.5: NeTraverse – Windows-Fenster

Machen Sie das Windows-Fenster im Zweifelsfall lieber etwas kleiner, da Anwender die Größe dieses Fensters, wie die jedes KDE-Fensters, bequem ändern können.

Innerhalb von Win4Lin läuft der größte Teil der für Windows 98 verfügbaren Programme. Lediglich Software, die direkt auf die Hardware zugreift, bereitet Probleme. Problemlos arbeitet aber weit verbreitete Software wie:

- Microsoft-Office
- StarOffice/OpenOffice
- MediaPlayer
- Internet Explorer
- ...

Die folgende Abbildung zeigt den Windows MediaPlayer mit einem Tagesschau-Bericht innerhalb von Win4Lin.



Abbildung 10.6: NeTraverse – Tagesschau im MediaPlayer

Die Windows-Installationen innerhalb von Win4Lin sind sehr stabil. Falls Windows einmal abstürzt, dann kann man es sofort neu starten, ohne dass zeitaufwändige Festplattenüberprüfungen notwendig wären. Wenn Sie, wie beschrieben, mit Templates arbeiten, dann können Sie auch beschädigte Windows-Installationen sehr schnell erneuern. Diese Umgebung bietet sich daher auch zum Testen von Windows-Programmen an.