Linux XFree86 HOWTO

von Dirk Knabe (knabe@fh-dortmund.de) v2.71828-2, 30. September 1998

Dieses Dokument beschreibt den Bezug, die Installation und die Konfiguration von XFree86 in der Version 3.3.2, ein X-Window-System für Linux-Rechner. Es stellt eine schrittweise Einführung in die Konfiguration von XFree86 dar.

Inhaltsverzeichnis

1 Vorwort

Stellte die Version 1.0 dieses HOWTO vom 13. Dezember 1996 lediglich die deutsche Übersetzung des englischsprachigen Dokuments *The Linux XFree86 HOWTO* in der Version 3.0 vom 15. März 1995 von Matt Welsh dar, so liegt nun in der aktuellen Version ein eigenständiges Werk vor. Dabei wurde der Versuch unternommen, die Vielzahl vorhandener Dokumente über das X-Window-System zu sondieren und didaktisch so aufzubereiten, daß dem Leser durch die schrittweise Anleitung des HOWTO ein erster Einstieg in die Nutzung von XFree86 unter Linux ermöglicht wird.

Zusätzlich sollte die jeweilige Dokumentation der entsprechenden XFree86-Version zu Rate gezogen werden. Sie wird im weiteren Verlauf des Textes an den entsprechenden Stellen benannt. Dieses HOWTO versteht sich nicht als Benutzeranleitung und auch nicht als Anleitung für die individuelle Gestaltung des X-Window-Systems. Zu diesem Zweck sollte man auf eines der vielen guten X-Window-Benutzerhandbücher zurückgreifen.

Allgemeine Informationen über das Linux-Betriebssystem in englischer Sprache findet man auf dem WWW-Server des Linux Documentation Project (LDP) unter der Adresse:

`http://metalab.unc.edu/LDP`

Dieser Server wird von den folgenden Servern in Deutschland täglich gespiegelt:

- `http://www.appl-opt.physik.uni-essen.de/linux/LDP/`
- `http://www.suse.de/doku/LDP/`
- `http://www.mordor.ask.fh-furtwangen.de/LDP/`
- `http://www.dlr.de/linux/LDP/`
- `http://mailer.wiwi.uni-marburg.de/linux/LDP/`
- `http://jurix.jura.uni-sb.de/LDP/`
- `http://www.asta.va.fh-ulm.de/LDP/`

Das Deutsche Linux HOWTO Projekt (DLHP), dem der Autor dieses HOWTO auch angehört, stellt eine Vielzahl von deutschsprachigen HOWTOs zur Verfügung, die entweder aus der Übersetzung des Originals oder als eigenständiges Werk entstanden sind. Den WWW-Server des DLHP erreicht man unter folgender Adresse:

`http://www.tu-harburg.de/dlhp/`

Die Dokumente stehen dort sowohl »online« als auch in den üblichen Formaten (ASCII, SGML, HTML, DVI, PostScript sowie Adobe PDF) zum Herunterladen zur Verfügung. Mittlerweile befinden sie sich aber auch in den meisten Linux-Distributionen der diversen Anbieter, so daß ein Bezug vom DLHP-Server entfallen kann, wenn man nicht gerade auf die aktuellste Version einer HOWTO angewiesen ist.

Gegenstand dieses HOWTO bildet die aktuelle Version 3.3.2 von XFree86 für Linux-Systeme.

Fragen, Bemerkungen und/oder Anregungen zu diesem HOWTO richten Sie bitte per E-Mail an:

`knabe@fh-dortmund.de`

2 Einleitung

Das X-Window-System stellt eine grafische Benutzerschnittstelle dar, die es dem Anwender ermöglicht, seine Programme gemeinsam unter einer grafischen Oberfläche darzustellen. Unter den verschiedenen Unix-Derivaten hat sich das X-Window-System mittlerweile als der Standard für grafische Benutzeroberflächen etabliert. Heutzutage läuft fast auf jedem Rechner mit einem Unix-Betriebssystem irgendeine X-Window-Variante.

2.1 Geschichte

Als Anfang der 80er Jahre grafikfähige Workstations eine schnelle Verbreitung fanden, bestand zunächst das Problem darin, daß für die Entwicklung bzw. Programmierung der erforderlichen grafischen Benutzeroberflächen kaum einheitliche Standards zur Verfügung standen. Dies lag darin begründet, daß das Betriebssystem Unix, bedingt durch die Hardware-Situation zur Zeit seines Entstehens, ein auf zeichenorientierte Ein- und Ausgabe hin ausgelegtes System darstellte. Somit begannen die verschiedenen Hersteller mit der Entwicklung ihrer eigenen grafischen Oberflächen. Schnell erkannte man die Problematik beim Einsatz grafischer Applikationen auf den unterschiedlichsten Hardware-Plattformen. Sollte nämlich eine Anwendung, die von den grafischen Fähigkeiten einer bestimmten Workstation Gebrauch machte, auch auf verschiedenen anderen Plattformen lauffähig sein, so mußte ein großer Teil des Programms mehrfach neu entwickelt und zusätzlich gepflegt werden.

In dieser Situation der unterschiedlichen grafischen Benutzeroberflächen entstand 1984 am MIT (Massachusetts Institute of Technology) in Zusammenarbeit mit der Firma DEC (Digital Equipment Corporation) das Projekt Athena. Das Ziel dieses Projekts sollte der Entwurf einer rechnerunabhängigen, standardisierten Umgebung zur Entwicklung grafischer Applikationen sein. Dies führte schließlich zu der Entwicklung des X-Window-Systems. Durch die Schaffung standardisierter Schnittstellen auf allen Ebenen des Systems war es nun möglich, grafische Anwendungen netzwerkweit und unabhängig von der eingesetzten Hardware zu betreiben.

Anfang 1987 gründeten 12 namhafte Computerhersteller das X-Konsortium mit dem Ziel, die Weiterentwicklung und Standardisierung des X-Window-Systems zu gewährleisten und eine kommerzielle Verwendung zu ermöglichen. Die damalige Liste der Gründungsmitglieder las sich wie ein »Who is who« in der Computerbranche.

Das X-Konsortium fordert in der Dokumentation zu X ausdrücklich die Verwendung folgender Bezeichnungen: X-Window-System, X Version 11, X11 oder einfach nur X.

Im September 1987 verabschiedete das X-Konsortium das X-Window-System in der Version 11, Release 1 (kurz: X11R1 genannt). Es löste die Anfang 1986 freigegebene Vorläuferversion X10 ab. X11R1 war im Gegensatz zu X10 dem Forschungsstadium entwachsen, bot eine größere Flexibilität und höhere Performance als der Vorgänger und ermöglichte somit eine erste kommerzielle Verwendung.

Seit dieser Zeit wurde das X-Window-System intensiv weiterentwickelt. Als herstellerunabhängiges System und durch die kostenlose Verfügbarkeit des Quellcodes des gesamten X-Window-Systems fand es bei den Anwendern eine große Akzeptanz und somit eine schnelle Verbreitung.

Die heutige Version 11, Release 6 (X11R6) des X-Window-Systems wird mittlerweile von fast allen bedeutenden Unix-Anbietern vertrieben und avancierte damit zum Industriestandard für die Steuerung grafischer Benutzeroberflächen unter dem Betriebssystem Unix.

Eine freie Implementierung des X-Window-Systems in der Version 11, Release 6 (X11R6) für PC-basierte Rechner (ab i386) ist von einer Reihe von Programmierern entwickelt worden, die sich im Jahre 1992 zum XFree86-Team zusammenschlossen. Daraus entstand 1994 die nicht-kommerzielle Firma The XFree86 Project, Inc., die sich bis heute ausschließlich durch Spenden finanziert, mit dem Ziel, XFree86 einer breiten Öffentlichkeit zur Verfügung zu stellen. Durch die sich anschließende Mitgliedschaft im X-Konsortium erhielt man zusätzlich die Möglichkeit, an der Weiterentwicklung des X-Window-Systems teilzuhaben. Mittlerweile ist XFree86 für eine Vielzahl unterschiedlicher Betriebssysteme frei erhältlich. Eine Version davon ist für Linux (Intel x86, DEC Alpha/AXP und m68k) konzipiert und frei verfügbar. Sie enthält alle erforderlichen Programme, Konfigurationsdateien, Bibliotheken sowie diverse Hilfsprogramme.

2.2 Konzept

Das X-Window-System besitzt drei besondere Eigenschaften, die es von den anderen herkömmlichen grafischen Benutzeroberflächen unterscheidet. Dazu zählen die Konzipierung und Realisierung als offenes System, die Client/Server-Architektur sowie als wichtigstes Merkmal die sogenannte Netzwerktransparenz.

Durch die Herstellerunabhängigkeit und durch die freie Verfügbarkeit der kompletten Quelltexte konnte das X-Window-System im Gegensatz zu anderen kommerziellen grafischen Oberflächen von Anfang an als offenes System konzipiert werden. Neben Standardschnittstellen, die eine komfortable Entwicklung portabler und hardwareunabhängiger Software erlauben, unterstützt das System auch Schnittstellen für herstellerspezifische Erweiterungen, um somit auch die Anbindung von spezieller Hardware zu ermöglichen.

Das X-Window-System unterscheidet aufgrund seiner spezifischen Architektur zwischen dem X-Server und den X-Clients. Der X-Server stellt ein Programm dar, das hardwareabhängig einen grafischen Bildschirm steuert. Zusätzlich stellt er das Bindeglied zwischen dem Benutzer und den verschiedenen X-Applikationen, den sogenannten X-Clients, dar, indem er Tastatureingaben bzw. Mausbewegungen des Benutzers an die entsprechenden X-Clients leitet und die von ihnen zurückgelieferten, hardwareunabhängigen Ausgabeinformationen grafisch darstellt.

Die Kommunikation zwischen dem Server und den Clients erfolgt ausschließlich über das standardisierte X-Protokoll. Dieses Protokoll ist so flexibel ausgelegt, daß es möglich wird, den X-Server und die X-Clients nicht nur lokal auf einem Rechner zu halten, sondern sie auf beliebige Rechner in einem Netz zu verteilen. Durch diese Netzwerktransparenz besteht somit die Möglichkeit, rechenintensive Programme auf Maschinen mit den leistungsstärksten Prozessoren laufen zu lassen, Ein- und Ausgabe jedoch auf einer normalen Workstation im Netz durchzuführen. Bei einer entsprechend schnellen Netzwerkverbindung kann die Entfernung der beiden Rechner durchaus mehrere tausend Kilometer voneinander betragen, so daß dies im Vergleich zu herkömmlichen grafischen Benutzeroberflächen kaum einen Performanceverlust mit sich bringt.

Die Konzeption als offenes System, die Client/Server-Architektur und die Netzwerktransparenz sind natürlich auch integraler Bestandteil des XFree86-Systems, so daß dem Linux-Anwender mit XFree86 ein hervorragendes X-Window-System zur Verfügung steht, das den Vergleich mit kommerziellen Systemen nicht scheuen muß.

3 Hardwarevoraussetzungen

3.1 Videochipsätze und Grafikkarten

Die aktuelle XFree86-Version 3.3.2, erschienen im März 1998, unterstützt die weiter unten aufgeführten Videochipsätze. Die der Grafikkarte beiliegende Dokumentation sollte Informationen über den verwendeten Chipsatz enthalten. Beim Kauf einer neuen Grafikkarte bzw. eines neuen Rechners sollte darauf geachtet werden, daß der entsprechende Chipsatz der Karte von XFree86 unterstützt wird. Fehlen diese Informationen, so kann mit Hilfe des Programms `SuperProbe` (Version 2.16), Bestandteil der XFree86-Distribution, der Chipsatz der Grafikkarte bestimmt werden. Dieser Vorgang wird weiter unten beschrieben. Zusätzlich besteht die Möglichkeit, den technischen Support des Herstellers der jeweiligen Grafikkarte zu kontaktieren, um die gewünschten Informationen zu erhalten.

Momentan unterstützt XFree86 in der Version 3.3.2 folgende Chipsätze:

Ark Logic

ARK1000PV, ARK1000VL, ARK2000PV, ARK2000MT

Alliance

AP6422, AT24

ATI

18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6, 68800-3, 68800-6, 68800AX, 68800LX, 88800GX-C, 88800GX-D, 88800GX-E, 88800GX-F, 88800CX, 264CT, 264ET, 264VT, 264GT, 264VT-B, 264VT3, 264GT-B, 264GT3 (Diese Liste schließt Mach8, Mach32, Mach64, 3D Rage, 3D Rage II und 3D Rage Pro ein.)

Avance Logic

ALG2101, ALG2228, ALG2301, ALG2302, ALG2308, ALG2401

Chips & Technologies

65520, 65530, 65540, 65545, 65520, 65530, 65540, 65545, 65546, 65548, 65550, 65554, 65555, 68554, 64200, 64300

Cirrus Logic

CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429, CLGD5430, CLGD5434, CLGD5436, CLGD5440, CLGD5446, CLGD5462, CLGD5464, CLGD5465, CLGD5480, CLGD6205, CLGD6215, CLGD6225, CLGD6235, CLGD6410, CLGD6412, CLGD6420, CLGD6440, CLGD7541(*), CLGD7543(*), CLGD7548(*), CLGD7555(*)

Digital Equipment Corporation

TGA

Compaq

AVGA

Genoa

GVGA

IBM

8514/A (und Kopien), XGA-2

IIT

AGX-014, AGX-015, AGX-016

Matrox

MGA2064W (Millennium), MGA1064SG (Mystique und Mystique 220), MGA2164W (Millennium II PCI und AGP)

MX

MX68000(*), MX680010(*)

NCR

77C22(*), 77C22E(*), 77C22E+(*)

Number Nine

I128 (Serie I und II), Revolution 3D (T2R)

NVidia-SGS Thomson

NV1, STG2000, RIVA128

OAK

OTI067, OTI077, OTI087

RealTek

RTG3106(*)

S3

86C911, 86C924, 86C801, 86C805, 86C805i, 86C928, 86C864, 86C964, 86C732, 86C764, 86C765, 86C767, 86C775, 86C785, 86C868, 86C968, 86C325, 86C357, 86C375, 86C375, 86C385, 86C988, 86CM65, 86C260

SiS

86C201, 86C202, 86C205

Tseng

ET3000, ET4000AX, ET4000/W32, ET4000/W32i, ET4000/W32p, ET6000, ET6100

Trident

TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i, TVGA9100B, TVGA9200CXR, Cyber9320(*), TVGA9400CXi, TVGA9420, TGUI9420DGi, TGUI9430DGi, TGUI9440AGi, TGUI9660XGi, TGUI9680, Pro-Vidia 9682, ProVidia 9685(*), Cyber 9382, Cyber 9385, Cyber 9388, 3DImage975(PCI), 3DImage985(AGP), Cyber 9397, Cyber 9520

Video 7-Headland Technologies

HT216-32(*)

Weitek

P9000

Western Digital-Paradise

PVGA1

Western Digital

WD90C00, WD90C10, WD90C11, WD90C24, WD90C24A, WD90C30, WD90C31, WD90C33

(*) Die so markierten Chipsätze besitzen entweder nur eine beschränkte Unterstützung oder ihre Treiber werden nicht permanent gewartet.

Grafikkarten mit den oben aufgeführten Chipsätzen werden von allen gängigen Bustypen unterstützt, und sie sind alle in der Lage, im Grafikmodus mindestens 256 verschiedene Farben darzustellen. Da Speicher heute recht günstig verfügbar ist, verfügen eigentlich alle modernen Karten über so viel Speicher, daß sie auch mit 15, 16, 24 oder 32 Bit Farbtiefe arbeiten können. Allerdings ist die Bandbreite des günstigen dynamischen Speichers für höhere Farbtiefe nur begrenzt brauchbar, so daß die Karten mit diesem Speichertyp bei höheren Farbtiefen, insbesondere bei 24 und 32 Bit, deutlich langsamer werden. Auch die Leistung der DA-Wandler, die das Signal für den Monitor erzeugen, vieler günstiger Grafikkarten ist bei hohen Farbtiefen beschränkt, so daß sich die Bildwiederholfrequenzen schnell in nicht mehr ergonomischen Regionen bewegen. Die Standard-Konfiguration beträgt 16 Bits je Bildpunkt. Das entspricht 65536 (2 hoch 16) verschiedenen Farben.

Der Monochrom-Server unterstützt auch generische VGA-Grafikkarten, indem nur 64 kB des vorhandenen Videospeichers benutzt werden: Die Hyundai HGC1280, die Sigma Laserview sowie die Hercules-, die Visa- und die Apollo-Monochromkarten.

Die XFree86-Distribution enthält Chipsatz-spezifische README-Dateien, die detaillierte Informationen über den Stand der Unterstützung jedes einzelnen Chipsatzes bereitstellen.

Ein Problem, dem die XFree86-Entwickler gegenüberstehen, besteht darin, daß einige Grafikkartenhersteller keine Standardmechanismen zur Bestimmung der Bildpunkt-Taktfrequenz, mit der die Grafikkarte betrieben wird, bereitstellen. Einige dieser Firmen geben die Spezifikationen für die Programmierung ihrer Grafikkarten nicht frei oder sie belegen die Karte mit einem besonderen Schutz, der verhindert, daß diese Informationen ausgelesen werden können. Dies steht dem Gedanken der freien Verteilung bzw. Verbreitung der XFree86-Software offensichtlich entgegen, und das XFree86-Entwicklerteam ist daher nicht bereit, diesen Zustand zu unterstützen. Die Firma Diamond zum Beispiel begann erst mit der Version 3.1 von XFree86 die Zusammenarbeit mit dem Entwicklerteam zur Erstellung von frei verfügbaren Treibern für ihre Grafikkarten.

Es ist wichtig zu wissen, daß auf den meisten Rechnern eine von der Performance eher durchschnittliche, beschleunigte SVGA-Grafikkarte signifikant schneller ist, als eine Standard-SVGA-Karte. Ein Linux-System mit einem XFree86-Server in der Version 3.1 auf einem 80486DX2-PC mit 66 MHz, VESA Local-Bus (VLB) und einer S3-Grafikkarte (864-Chipsatz) mit 2 MB DRAM ist laut X-Benchmark um den Faktor 7 schneller als eine Sun Sparc-IPX-Workstation.

Die Version 3.3 von XFree86 erzielt einen noch größeren Faktor. Im allgemeinen ist XFree86 auf einem Linux-Rechner mit einer beschleunigten SVGA-Grafikkarte leistungsfähiger als eine kommerzielle Unix-Workstation.

3.2 Prozessor, Arbeitsspeicher und Plattenplatz

Mindestens 4 MB physikalischer Speicher und 16 MB virtueller Speicher (z.B. 8 MB für den physikalischen Bereich und 8 MB für den Swapbereich) werden zum Betreiben von XFree86 benötigt. Man beachte, daß mehr physikalischer Speicher ein Ein- und Auslagern von Speicherbereichen auf die bzw. von der Festplatte verhindert, wenn der Hauptspeicher des Systems knapp wird. Da der mit dem Swappen verbundene Zugriff auf die Festplatte deutlich langsamer ist als der Zugriff auf den Hauptspeicher, werden für XFree86 mindestens 8 MB oder mehr RAM empfohlen. Ein System mit nur 4 MB RAM kann bis zu dem Faktor 10 langsamer sein als eines mit 8 MB oder mehr.

Eine empfohlene Hardwareausstattung, um XFree86 unter Linux komfortabel zu betreiben, ist ein PC mit einem i486-Prozessor und mit mindestens 16 MB Arbeitsspeicher sowie eine Grafikkarte mit einem der oben aufgeführten Chipsätze. Für eine optimale Leistung ist eine beschleunigte Grafikkarte erforderlich, wie z.B. eine Karte mit einem S3-Chipsatz. Bevor eine teure Grafikkarte gekauft wird, sollte man sich in der XFree86-Dokumentation davon überzeugen, ob sie auch unterstützt wird. Benchmark-Tests zu den verschiedenen Grafikkarten unter XFree86 finden sich regelmäßig in folgenden Newsgruppen:

- `comp.windows.x.i386unix`
- `de.comp.os.unix.linux.x`

Eine Standard-Installation des XFree86-Systems benötigt minimal ca. 60-80 MB freien Plattenplatz. Dies beinhaltet den eigentlichen X-Server, Bildschirmschriften, Bibliotheken und Standard-Programme. Sollen der Standard-Installation noch weitere X-Applikationen hinzugefügt werden, die ein komfortables Arbeiten mit XFree86 erst ermöglichen, so kann der hierzu benötigte Plattenplatz schnell auf 200 MB und mehr anwachsen.

4 Installation von XFree86

4.1 Bezug der Distribution

Die Linux-Distribution von XFree86 kann von diversen FTP-Servern bezogen werden. Auf dem FTP-Server von XFree86

```
ftp.xfree86.org
```

befindet sich die Distribution in dem Verzeichnis `/pub/XFree86/3.3.2`. Zur Zeit der Erstellung dieses Dokumentes lautet die aktuelle Version von XFree86 3.3.2; neuere Versionen werden regelmäßig folgen.

In den gängigen Linux-Distributionen ist XFree86 bereits enthalten, so daß der Bezug von einem FTP-Server entfällt, falls man nicht gerade eine sehr neue Grafikkarte gekauft hat, die von der XFree86-Version, die der eigenen Distribution beiliegt, noch nicht unterstützt wird.

Falls XFree86 doch direkt per FTP bezogen werden soll, listet die nachfolgende Tabelle vier Server in Deutschland auf, die den FTP-Server der XFree86-Organisation regelmäßig spiegeln:

- `ftp.gwdg.de:/pub/xfree86/XFree86`
- `ftp.cs.tu-berlin.de:/pub/X/XFree86`
- `ftp.uni-erlangen.de:/pub/Linux/MIRROR.xfree86`
- `ftp.uni-stuttgart.de:/pub/X11/Xfree86`

4.2 Benötigte Dateien

Für die Installation von XFree86 in der Version 3.3.2 ist einer der folgenden Server erforderlich:

X3328514.tgz
: Server für die 8514-Grafikkarte

X332AGX.tgz
: Server für die AGX-Grafikkarte

X332Ma64.tgz
: Server für die Mach64-Grafikkarte

X332Ma32.tgz
: Server für die Mach32-Grafikkarte

X332Ma8.tgz
: Server für die Mach8-Grafikkarte

X332Mono.tgz
: Server für die Monochrom-Grafikkarte

X332P9K.tgz
: Server für die P9000-Grafikkarte

X332S3.tgz
: Server für die S3-Grafikkarte

X332S3V.tgz
: Server für die alte S3-ViRGE-Grafikkarte (besser den SVGA-Server benutzen)

X332SVGA.tgz
: Server für viele Super-VGA-Grafikkarten

X332VG16.tgz
: Server für die VGA-Grafikkarte (wird für das Programm XF86Setup benötigt)

X332W32.tgz
: Server für die ET4000/W32- und ET6000-Grafikkarte

Alle nachfolgenden Dateien sind sowohl für eine Neuinstallation als auch für ein Upgrade von Versionen vor 3.3.1 erforderlich:

preinst.sh
: Skript für die Installationsvorbereitung

postinst.sh
: Skript für die Installationsnachbereitung

extract
: XFree86-Extrahierungsprogramm

X332bin.tgz

Rest der XFree86-Programme

X332doc.tgz

Dokumentation

X332fnts.tgz

Basis-Schriftarten (75dpi, misc und PEX)

X332lib.tgz

Zur Laufzeit benötigte Dateien

X332man.tgz

Manual-Seiten

X332set.tgz

XF86Setup-Programm

X332VG16.tgz

Server für die VGA-Grafikkarte (wird für XF86Setup benötigt)

X3322upd.tgz

XFree86 3.3.3 Patch 2

Die folgende Datei ist für eine neue Installation erforderlich, für eine bereits bestehende optional:

X332cfg.tgz

Beispiel-Konfigurationsdateien für `xinit` und `xdm`

Bei einer bereits bestehenden Installation von XFree86 mit benutzerspezifischen `xinit`- oder `xdm`-Konfigurationsdateien besteht keine Notwendigkeit zur Benutzung dieser Datei, da bei einer erneuten Installation von `X332cfg.tgz` diese Dateien überschrieben werden. Möchte man trotzdem `X332cfg.tgz` installieren, sollten benutzerangepaßte `xinit`- bzw. `xdm`-Konfigurationsdateien zuvor gesichert werden.

Die folgenden Dateien sind für die Installation von XFree86 optional:

X332f100.tgz

100-dpi-Bildschirm-Schriftarten

X332fcyr.tgz

Kyrillische Bildschirm-Schriftarten

X332fnon.tgz

Zusätzliche Schriftarten (chinesisch, japanisch, koreanisch und hebräisch)

X332fscl.tgz

Skalierte Schriftarten (Speedo und Type1)

X332fsrv.tgz

Server für Bildschirm-Schriftarten und Konfigurationsdateien

X332prog.tgz

X-Header- und Konfigurationsdateien sowie Bibliotheken zur Übersetzungszeit

X332nest.tgz

Besonderer »Nested«-X-Server

X332vfb.tgz

Virtueller »Framebuffer«-X-Server

X332prt.tgz

X-Print-Server

X332lkit.tgz

Server-Werkzeug für benutzerdefinierte Einstellungen

X332lk98.tgz

Server-Werkzeug für PC98-Server

X332ps.tgz

Dokumentation im PostScript-Format

X332html.tgz

Dokumentation im HTML-Format

X332jdoc.tgz

Japanische Dokumentation (Version 3.2 von XFree86)

X332jhtm.tgz

Japanische Dokumentation im HTML-Format (Version 3.2 von XFree86)

4.3 Anleitung

Das XFree86-Verzeichnis enthält `README`-Dateien und Installationshinweise der jeweiligen aktuellen Version.

Bei einer bereits installierten Version von XFree86 sollte bei einer Neuinstallation zuvor ein Backup (Sicherheitskopie) der bestehenden XFree86-Version erstellt werden, da die Standard-Installationsprozedur die alte Version überschreibt. Zur Erstellung der Sicherheitskopie wechselt man als Benutzer `root` in das Verzeichnis `/usr/X11R6`, archiviert und komprimiert es anschließend:

```
cd /usr/X11R6
tar zcvf XFree86_Backup ./*
```

Für das Backup einer bereits bestehenden XFree86-Version werden ca. 8 MB freier Plattenplatz benötigt.

Bei einer Erstinstallation muß das Verzeichnis `/usr/X11R6` vom Benutzer `root` erstellt werden:

```
mkdir /usr/X11R6
```

Anschließend muß das Skript für die Installationsvorbereitung `preinst.sh` ausgeführt werden. Dieses Skript führt einige Routineüberprüfungen des bestehenden Linux-Systems durch, um zu testen, ob alle Versionen der bei der Installation von XFree86 erforderlichen Systemkomponenten jeweils auf dem aktuellsten Stand sind. Bei Bedarf meldet die Testroutine, die entsprechenden Komponenten vor der Fortführung der Installation zu aktualisieren. Das

Skript löscht auch nicht mehr benötigte Dateien und entfernt symbolische Links vorhergehender XFree86-Versionen, die bei einer Neuinstallation möglicherweise Probleme verursachen könnten. Für den Fortgang der Installation ist es nun erforderlich, daß sich alle benötigten Dateien der XFree86-Distribution in einem temporären Verzeichnis, z.B. `/tmp`, befinden. Zur Ausführung des Skripts `preinst.sh` wechselt man dann in das Verzeichnis `/usr/X11R6` und startet es anschließend:

```
cd /usr/X11R6
sh /tmp/preinst.sh
```

Nach erfolgreichem Abschluß der Installationsvorbereitungen müssen nun im nächsten Schritt die Zugriffsrechte des Installationsprogramms `extract` so gesetzt werden, daß diese Datei auch ausführbar ist. Dazu wird folgendes Kommando ausgeführt:

```
chmod 755 /tmp/extract
```

Das Programm `extract` wird benötigt, um die .tgz-Dateien der XFree86-Distribution zu entpacken (entkomprimieren und -archivieren). Die Dateiendung .tgz besagt, daß es sich um eine »gezipte-tar«-Datei handelt, d.h. ihr Inhalt ist archiviert und komprimiert. Man beachte, daß das normalerweise verwendete Standard-Unix-Programm `tar` zur Installation der XFree86-Distribution nicht geeignet ist. Das Programm `extract` ist eine für die Besonderheiten der XFree86-Installation modifizierte Variante des GNU `tar`-Kommandos der Version 1.12. Daher ist es unbedingt erforderlich, dieses Programm zum Entpacken der notwendigen Dateien der Distribution zu verwenden. Auch darf diese Datei nicht umbenannt werden, da sie sich sonst wie das Standard-Programm `tar` verhält.

Mit Hilfe des Programms `extract` werden die .tgz-Dateien der XFree86-Distribution als Benutzer `root` wie folgt extrahiert:

```
cd /usr/X11R6
/tmp/extract /tmp/X332[8-z]*.tgz
/tmp/extract /tmp/X3322upd.tgz
```

Das Meta-Zeichen in Dateinamen [...] steht für »Eins der in der Klammer vorkommenden Zeichen«; [8-z] steht hier für den Bereich von Zeichen 8 mit ASCII-Wert 56 fortlaufend bis zu Zeichen z mit ASCII-Wert 122.

Man beachte, daß die Update-Datei `X3322upd.tgz` separat zum Schluß extrahiert wird, da sie diverse Bibliotheken (`X11`, `Xt`, `Xaw` und `Xmu`) sowie das `xterm`-Programm mit aktuellen Versionen, die vornehmlich neuen Sicherheitsvorkehrungen dienen, überschreibt.

Nach erfolgreicher Extrahierung der .tgz-Dateien wird das Skript für die Installationsnachbereitung `postinst.sh` ausgeführt:

```
cd /usr/X11R6
sh /tmp/postinst.sh
```

Das Skript aktualisiert das Schriftarten-Verzeichnis in `/usr/X11R6/lib/X11/fonts/misc` und installiert, falls erforderlich, neue `terminfo`-Einträge für die aktuelle Version des Programms `xterm`.

Nach der Ausführung des Nachbereitungsskriptes `postinst.sh` muß der Link `/usr/X11R6/bin/X` so gesetzt werden, daß er auf die Datei des zuwendenden Servers zeigt. Soll z.B. der SVGA-Server benutzt werden, so wird `/usr/X11R6/bin/XF86_SVGA` nach `/usr/bin/X11/X` gelinkt. Falls dagegen der Monochrom-Server verwendet werden soll, findet folgendes Kommando Verwendung:

```
ln -sf /usr/X11R6/bin/XF86_MONO /usr/X11R6/bin/X
```

Entsprechendes gilt bei der Verwendung eines anderen Servers.

Falls Unklarheit über den verwendeten Server bzw. über den Chipsatz der Grafikkarte besteht, kann hier das Programm `SuperProbe` weiterhelfen. Es befindet sich in dem Verzeichnis `/usr/X11R6/bin` und bestimmt den Chipsatz der Grafikkarte und andere wichtige Informationen, die für eine spätere Referenz notiert werden sollten.

Als nächster Schritt muß das Verzeichnis `/usr/X11R6/bin` mit in den Suchpfad aufgenommen werden, falls dieses sich nicht bereits im Suchpfad befindet. Dazu kann bei Verwendung der Bash-Shell die Systemdatei `/etc/profile` ediert werden oder es wird einfach das Verzeichnis dem persönlichen Suchpfad beigefügt, indem die benutzereigene Konfigurationsdatei der Bash-Shell `~/.bashrc` modifiziert wird.

Zum Abschluß der Installation muß sichergestellt werden, daß der Linker `ld.so` das Verzeichnis `/usr/X11R6/lib` lokalisieren kann. Hierzu wird die Zeile `/usr/X11R6/lib` der Datei `/etc/ld.so.conf` hinzugefügt und anschließend als Benutzer `root` das Programm `/sbin/ldconfig` gestartet:

```
/sbin/ldconfig
```

Mit der erfolgreichen Ausführung des Programms `/sbin/ldconfig` ist die Installationsphase abgeschlossen und es kann nun mit der Konfiguration von XFree86 begonnen werden.

5 Konfiguration von XFree86

Das Einrichten von XFree86 ist in den meisten Fällen unproblematisch. Bei der Verwendung von Hardware, deren Treiber sich noch in der Entwicklung befinden oder beim Wunsch, die beste Leistung bzw. Auflösung mit einer beschleunigten Grafikkarte zu erzielen, kann sich die Konfiguration von XFree86 als sehr zeitaufwendig erweisen.

Dieses Kapitel beschreibt die Erstellung und Bearbeitung der Datei `XF86Config`, die den XFree86-Server konfiguriert. In den meisten Fällen hat es sich bewährt, mit einer Basiskonfiguration für XFree86 zu starten, die eine Standardauflösung von 640x480 Bildpunkten einstellt und von fast allen Grafikkarten und Monitoren unterstützt wird. Arbeitet XFree86 erst einmal mit einer Standardauflösung, so kann die Konfiguration jederzeit erweitert werden, um sämtliche Fähigkeiten der Videohardware auszunutzen. Dahinter verbirgt sich die Idee, zu überprüfen, ob XFree86 überhaupt auf dem System funktioniert und ob die Installation auch nicht fehlerhaft ist.

Zusätzlich zu dieser HOWTO sollten die folgenden Dokumente gelesen werden:

- Die XFree86-Dokumentation. Sie befindet sich in dem Paket `X332doc.tgz` und liegt in dem Verzeichnis `/usr/X11R6/lib/X11/doc`.
- Das XFree86-Tutorial. Es befindet sich in der Datei `README.Config`.
- Videochipsatz-Informationen. Zu mehreren Chipsätzen existieren eigene `README`-Dateien, wie z.B.: `README.Cirrus` und `README.S3`, die sich in dem oben genannten Verzeichnis befinden.
- Die Manual-Seiten von `XFree86`.
- Die Manual-Seiten von `XF86Config`, `XF86Setup` und `xvidtune`.
- Die Manual-Seiten über den benutzten Server, wie z.B. `XF86_SVGA` oder `XF86_S3`.

Die Haupt-Konfigurationsdatei ist `/usr/X11R6/lib/X11/XF86Config`. Sie enthält Informationen über die Maus, die Grafikkarte, den Monitor usw. Die mit der XFree86-Distribution bereitgestellte Datei `XF86Config.eg` enthält eine Beispielkonfiguration, die als Ausgangspunkt für die eigene Konfiguration benutzt werden kann. Dazu wird sie nach `XF86Config` kopiert.

Die `XF86Config`-Manual-Seite erläutert das Format dieser Datei im Detail. Bevor mit diesem Dokument fortgefahren wird, sollte die Manual-Seite gelesen werden.

Nachfolgend wird eine Muster-`XF86Config`-Datei schrittweise vorgestellt. Diese Datei kann sich von der Beispieldatei, die in der XFree86-Distribution enthalten ist, unterscheiden, aber ihre Struktur ist dieselbe.

Man beachte, daß das Dateiformat von `XF86Config` mit jeder Version von XFree86 wechseln kann. Diese Angaben beziehen sich auf die XFree86-Version 3.3.2.

Ferner sollte auch der Versuch unterbleiben, die hier aufgelistete Konfigurationsdatei auf das eigene System zu kopieren und anschließend zu benutzen. Eine Konfigurationsdatei, die nicht mit der verwendeten Hardware korrespondiert, kann Schaden an ihr anrichten. Berichten zufolge sind Monitore (besonders Festfrequenz-Monitore) bei der Benutzung einer unkorrekt konfigurierten XF86Config-Datei beschädigt bzw. zerstört worden. Der Grundsatz lautet also: »Vor der Benutzung einer `XF86Config`-Datei sicherstellen, daß sie mit der verwendeten Hardware korrespondiert«.

Jeder Abschnitt einer `XF86Config`-Datei wird von dem folgenden Zeilenpaar umgeben:

```
Section "section-name"
  ...
EndSection
```

Der erste Abschnitt der `XF86Config`-Datei ist `Files`, der wie folgt aussieht:

```
Section "Files"
    RgbPath     "/usr/X11R6/lib/X11/rgb"
    FontPath    "/usr/X11R6/lib/X11/fonts/misc:unscaled"
    FontPath    "/usr/X11R6/lib/X11/fonts/misc"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi"
EndSection
```

Die `RgbPath`-Zeile setzt den Pfad zu der X11R6-RGB-Farbdatenbank. Jede `FontPath`-Zeile zeigt auf das jeweilige Verzeichnis mit den entsprechenden X11-Schriftarten. Im allgemeinen besteht kein Grund, diese Zeilen zu ändern, falls man nicht gerade ein neues Verzeichnis für weitere Schriftarten anlegen möchte. Für jedes Verzeichnis in `/usr/X11R6/lib/X11/fonts` sollte ein Eintrag in der Sektion `Files` zu finden sein.

Der nächste Abschnitt lautet `ServerFlags`, der verschiedene globale Flags des Servers spezifiziert. Er enthält im allgemeinen keinen Eintrag.

```
Section "ServerFlags"
# Diese Option erzeugt genau an der Stelle einen core
# dump, an der das entsprechende Signal empfangen wurde.
# Dieses läßt die Konsole eventuell in einem unbenutz-
# baren Zustand zurück, kann jedoch das Debuggen mit
# diesem core dump erleichtern.
#    NoTrapSignals

# Diese Option schaltet die Möglichkeit ab, den X11-
# Server mit <Strg><Alt><Backspace> abzubrechen.
#    DontZap
EndSection
```

Alle Zeilen innerhalb der Sektion sind daher mit Hilfe des Nummernzeichens # auskommentiert.

Der folgende Abschnitt `Keyboard` beschreibt die Standardeinstellungen für die Tastatur.

```
Section "Keyboard"
    Protocol    "Standard"
```

```
    XkbRules    "xfree86"
    XkbModel    "pc102"
    XkbLayout   "de"
    XkbVariant  "nodeadkeys"
EndSection
```

Für eine Modifizierung der Tastatureinstellungen mit Hilfe zusätzlicher Optionen wird auf die Datei `XF86Config` hingewiesen. Die oben angegebenen Werte funktionieren auf den meisten Systemen.

Die Parameter für die Maus werden im nächsten Abschnitt spezifiziert.

```
Section "Pointer"
    Protocol    "MouseSystems"
    Device      "/dev/mouse"

# Baudrate und SampleRate werden nur für einige Mäuse
# von Logitech verwendet.
#    BaudRate   9600
#    SampleRate 150

# Emulate3Buttons ist eine Option für Microsoft Mäuse
# mit nur 2 Tasten
#    Emulate3Buttons

# ChordMiddle ist eine Option für einige 3-Tasten Mäuse
# von Logitech
#    ChordMiddle
EndSection
```

Von anfänglicher Bedeutung sind hier die beiden Optionen: `Protocol` und `Device`. `Protocol` spezifiziert das von der Maus verwendete Protokoll, nicht aber die Marke der Maus. Gültige Protokolltypen sind:

- `BusMouse`
- `Logitech`
- `Microsoft`
- `MMSeries`
- `Mouseman`
- `MouseSystems`
- `PS/2`
- `MMHitTab`

Man beachte, daß unter Linux zusätzliche Typangaben für andere Betriebssysteme verfügbar sind. Die Auswahl `BusMouse` sollte bei einer *Logitech-Busmaus* verwendet werden. Ferner sollte man beachten, daß ältere *Logitech-Mäuse* das `Logitech`-, dagegen neuere Modelle entweder das `Microsoft`- oder das `Mouseman`-Protokoll benutzen. Dies ist ein Grund, warum das Protokoll nichts mit der Mausmarke zu tun hat.

Die Auswahl `Device` beschreibt die Schnittstelle, an der die Maus angeschlossen ist. Bei den meisten Linux-Systemen ist dies `/dev/mouse`. Gewöhnlich ist `/dev/mouse` ein Link auf die passende serielle Schnittstelle (z.B. `/dev/ttyS0`) für serielle Mäuse bzw. die passende Schnittstelle für Busmäuse. Auf jeden Fall sollte man sich überzeugen, daß die im `Device`-Abschnitt angegebene Gerätedatei existiert.

Der nächste Abschnitt lautet `Monitor` und beschreibt die Kenndaten des Monitors. Er kann, wie andere Sektionen in der `XF86Config`-Datei auch, mehrmals auftreten. Dies ist nützlich, falls mehrere Monitore an ein System angeschlossen sind oder die gleiche `XF86Config`-Datei unter verschiedenen Hardware-Konfigurationen benutzt werden soll. Gewöhnlich wird nur ein einzelner `Monitor`-Abschnitt verwendet.

```
Section "Monitor"
    Identifier  "CTX 5468 NI"

    # Diese Werte sind nur für einen CTX 5468NI Monitor!
    # Versuchen sie nicht, diese Werte für ihren Monitor
    # zu verwenden, wenn sie nicht wirklich dieses
    # Modell haben sollten.

    Bandwidth    60
    HorizSync    30-38,47-50
    VertRefresh  50-90

    # Modes: Name       dotclock horiz                vert

    ModeLine "640x480"  25       640 664 760 800      480 491 493 525
    ModeLine "800x600"  36       800 824 896 1024     600 601 603 625
    ModeLine "1024x768" 65       1024 1088 1200 1328  768 783 789 818
EndSection
```

Der `Identifier`-Eintrag kennzeichnet den Monitor mit einem Namen. Er ist frei wählbar (jede gültige Zeichenkette) und wird in einer späteren Sektion der `XF86Config`-Datei als Verweis auf den `Monitor`-Abschnitt verwendet.

`HorizSync` spezifiziert die gültigen horizontalen Synchronisationsfrequenzen des Monitors in kHz. Bei Multisync-Monitoren können dies ein oder mehrere, durch Komma getrennte Intervalle sein. Bei Festfrequenz-Monitoren ist es eine Reihe von festen Werten, wie z.B.:

```
HorizSync    31.5, 35.2, 37.9, 35.5, 48.95
```

Das Monitorhandbuch sollte im Abschnitt »Technische Spezifikationen« die entsprechenden Werte auflisten. Falls dies nicht der Fall ist, kann man versuchen, die benötigten Informationen beim Monitorhersteller mittels WWW oder beim Händler selbst zu bekommen.

`VertRefresh` gibt die gültigen vertikalen Refresh-Raten oder vertikalen Synchronisationsfrequenzen des Monitors in Hz an. Dies kann, wie schon bei `HorizSync`, ein Intervall oder eine Liste fester Werte sein. Auch hier sollte das Handbuch die entsprechenden Angaben aufweisen.

Mit Hilfe der `HorizSync`- und `VertRefresh`-Einträge wird überprüft, ob sich die angegebenen Bildschirmauflösungen in einem gültigen Bereich befinden. Dadurch wird die Möglichkeit reduziert, den Monitor beim Betreiben einer unzulässigen bzw. zu hohen Frequenz, für die er nicht ausgelegt ist, zu zerstören.

Die `ModeLine`-Zeile wird zur Spezifizierung der einzelnen Auflösungen des Monitors benutzt. Das Format von `Modeline` ist:

```
ModeLine Name dot-clock horiz-Werte vert-Werte
```

`Name` ist eine willkürliche Zeichenkette. Sie dient in der Datei zur Referenzierung auf den jeweiligen Auflösungsmodus. `Dot-clock` bezeichnet die benutzte Bildpunkt-Taktfrequenz. Die Frequenz wird gewöhnlich in MHz angegeben und gibt die Anzahl der von einer Grafikkarte an den Monitor ausgesendeten Bildpunkte je Sekunde bei dieser Auflösung an. Je vier Werte für `horiz` und vier Werte für `vert` spezifizieren den Bildschirmaufbau, d.h., sie geben die Anzahl der Bildpunkte an, bei der der Elektronenstrahl eine Zeile anzeigt, eine Beruhigungsphase und den Synchronisationsimpuls durchläuft.

Zur Bestimmung der `ModeLine`-Werte für den entsprechenden Monitor sei auf die Datei `VideoModes.doc` verwiesen, die zum Umfang der XFree86-Distribution gehört. Das Dokument beschreibt detailliert, wie diese Werte für jede vom Monitor unterstützte Auflösung ermittelt werden können. Zu allererst muß sichergestellt sein, daß der angegebene `dot-clock`-Wert auch mit einer von der Grafikkarte verwendeten Bildpunkt-Taktfrequenz korrespondiert. Man beachte, daß nur solche Videomodi verwendet werden können, deren Bildpunkt-Taktfrequenz von der Grafikkarte unterstützt wird. Im weiteren Verlauf der `XF86Config`-Datei werden die Bildpunkt-Taktfrequenzen weiter spezifiziert.

Zwei Dateien der XFree86-Distribution enthalten Daten für die `ModeLine`-Zeilen des zu verwendenden Monitors. Diese Dateien heißen: `modeDB.txt` bzw. `Monitors` und befinden sich beide im Verzeichnis `/usr/X11R6/lib/X11/doc`.

Anfänglich ist es empfehlenswert, mit `Modeline`-Werten des VESA-Standards zu beginnen, da die meisten Monitore die VESA-Auflösungen unterstützen. Die Datei `modeDB.txt` enthält Werte für die verschiedenen VESA-Standardauflösungen, und ihre Einträge sehen z.B. wie folgt aus:

```
# 640x480@60Hz Non-Interlaced Modus
# Horizontal Sync = 31.5kHz
# Timing: H=(0.95us, 3.81us, 1.59us)
#         V=(0.35ms, 0.064ms, 1.02ms)
#
# Name        clock   horizontales Timing   vertikales Timing     flags
  "640x480"   25.175  640  664  760  800    480  491  493  525
```

Dies ist der VESA-Standard für den 640x480-Videomodus. Er benutzt eine Bildpunkt-Taktfrequenz von 25,175, die die Grafikkarte unterstützen muß, um diesen Modus zu benutzen (später mehr dazu). Um diesen Eintrag in der `XF86Config`-Datei zu benutzen, muß die nachfolgende Zeile eingefügt werden:

```
ModeLine "640x480" 25.175 640 664 760 800 480 491 493 525
```

Man beachte, daß der `Name` der Zeile `640x480` eine frei wählbare Zeichenkette darstellt. Nach Konvention sollte er die verwendete Auflösung bezeichnen. Für `Name` kann theoretisch auch jede andere Zeichenkette gewählt werden, die den Modus beschreibt.

Für jede benutzte `ModeLine`-Zeile überprüft der Server, ob die Angaben des Modus innerhalb des Gültigkeitsbereichs von `Bandwidth` (Bandbreite), `HorizSync` und `VertRefresh` liegen. Falls sie nicht im gültigen Bereich liegen, werden die jeweiligen `ModeLine`-Zeilen vom X11-Server ignoriert.

Stellt sich heraus, daß die Werte für die VESA-Standardmodi nicht funktionieren, so enthalten die Dateien `modeDB.txt` und `Monitors` spezifische Einträge für viele verschiedene Monitortypen. Auf Basis der Einträge aus diesen beiden Dateien können eigene Werte kreiert werden. Man sollte jedoch nur dem Monitormodell entsprechende Werte benutzen. Man beachte, daß jeder Monitor nur eine bestimmte maximale Auflösung darstellen kann. Hier sollte man sich nicht nur auf die Herstellerangaben verlassen, da diese Angaben oft übertrieben sind. Rein physikalisch kann ein 15-Zoll-Monitor maximal 800x600 scharf darstellen. Höhere Auflösungen führen dazu, daß das Bild unscharf wird. Bei 17-Zoll-Monitoren sind maximal 1280x1024 zu empfehlen.

Sind an dieser Stelle immer noch keine gültigen Werte für den Monitor gefunden, kann man den Instruktionen aus der der XFree86-Distribution beigefügten Datei `VideoModes.doc` folgen, um `ModeLine`-Werte aus den im Monitor-Handbuch aufgelisteten Spezifikationen selber zu generieren. Die Datei `VideoModes.doc` beschreibt außerdem sehr ausführlich das Format einer `ModeLine`-Direktive sowie andere Aspekte des XFree86-Servers.

Sind die übernommenen oder selbergenerierten `ModeLine`-Werte noch nicht ganz perfekt, so ist es möglich, durch ein geringes Modifizieren der Werte das gewünschte perfekte Resultat zu erhalten. Ist z.B. während des XFree86-Betriebes das Monitorbild ein wenig verschoben oder scheint es zu laufen, so kann man gemäß der Anleitung in der Datei `VideoModes.doc` versuchen, diese Werte zu verbessern. Auch sollten die Einstellregler des Monitors überprüft werden. In vielen Fällen ist es nur notwendig, die horizontale oder vertikale Bildschirmgröße nach dem Start von XFree86 zu ändern, um das Bild zu zentrieren und auf eine passende Größe anzuordnen. Das Vorhandensein der Kontrollregler an der Frontseite des Monitors macht das Leben sicherlich leichter.

Man sollte weder Monitor-Frequenzen noch `ModeLine`-Werte von Monitoren benutzen, die nicht dem zu verwendenden Modell entsprechen. Versucht man dennoch, den Monitor mit einer Frequenz anzusteuern, für die er nicht konzipiert ist, so besteht die Gefahr, ihn zu beschädigen oder ihn gar zu zerstören.

Der nächste Abschnitt der Konfigurationsdatei `XF86Config` ist `Device`, der die Parameter der Grafikkarte kennzeichnet. Hier ein Beispiel:

```
Section "Device"
    Identifier "#9 GXE 64"

    # bisher nichts; wir werden diese später mit
    # Optionen füllen
EndSection
```

Die Sektion definiert die Eigenschaften für eine bestimmte Grafikkarte. `Identifier` ist wieder eine frei wählbare Zeichenkette, die die Karte beschreibt. Auch hier wird der Name für einen späteren Verweis benutzt.

Anfänglich braucht außer `Identifier` nichts in diesen Abschnitt eingetragen zu werden. Das liegt daran, daß der X-Server dazu benutzt wird, die Eigenschaften der Grafikkarte selber zu ermitteln. Anschließend können die so gefundenen Werte in die `Device`–Sektion eingetragen werden. Der X-Server ist nämlich in der Lage, den Videochipsatz, die Bildpunkt-Taktfrequenzen, den RAMDAC sowie die Größe des Videospeichers auf der Grafikkarte festzustellen.

Bevor diese Werte jedoch zu ermitteln sind, muß die `XF86Config`-Datei durch den letzten Abschnitt komplettiert werden. Er lautet `Screen` und spezifiziert die Kombination von Monitor und Grafikkarte für die Benutzung eines bestimmten Servers.

```
Section "Screen"
    Driver      "Accel"
    Device      "#9 GXE 64"
    Monitor     "CTX 5468 NI"
    Subsection "Display"
        Depth       16
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
EndSection
```

Die Zeile `Driver` beschreibt den zu benutzenden X-Server. Gültige Angaben für `Driver` sind:

- `Accel`: Für die `XF86_S3`-, `XF86_Mach64`-, `XF86_Mach32`-, `XF86_Mach8`-, `XF86_8514`-, `XF86_P9000`-, `XF86_AGX`- und `XF86_W32`-Server.
- `SVGA`: Für den `XF86_SVGA`-Server.
- `VGA16`: Für den `XF86_VGA16`-Server.
- `VGA2`: Für den `XF86_Mono`-Server.
- `Mono`: Für die Nicht-VGA-Monochrom-Treiber der `XF86_Mono`- und `XF86_VGA16`-Server.

Es sollte sichergestellt sein, daß ein symbolischer Link von `/usr/X11R6/bin/X` auf den zu benutzenden Server zeigt.

Die `Device`-Zeile spezifiziert den `Identifier` der `Device`-Sektion, der mit der verwendeten Grafikkarte korrespondiert. In dem oben angegebenen `Device`-Abschnitt ist die Zeile

```
Identifier "#9 GXE 64"
```

eingetragen worden. Daher wird in dieser Sektion die Zeichenkette `#9 GXE 64` in die `Device`-Zeile gestellt.

Ähnliches geschieht mit der Zeile `Monitor`. Sie verweist auf den `Identifier "CTX 5468 NI"` aus der oben beschriebenen `Monitor`-Sektion.

Der Unterabschnitt `Display` definiert verschiedene Eigenschaften des XFree86-Servers mit der Kombination aus Monitor und Grafikkarte. Die `XF86Config`-Datei beschreibt diese Optionen sehr detailliert. Die meisten davon sind jedoch wie der »Zuckerguß auf einem Kuchen« und daher nicht notwendig, um ein System zum Laufen zu bekommen.

Die wichtigsten Optionen sind:

Depth

Definiert die Farbtiefe, d.h. die Anzahl Bits je Bildpunkt. Der Standardwert beträgt 8. Ein VGA-Server benutzt eine Farbtiefe von 4 und ein Monochrom-Server eine Farbtiefe von 1. Bei der Benutzung einer beschleunigten Grafikkarte mit genügend Speicher, um mehr Bits je Bildpunkt zu unterstützen, kann der Wert auf 16, 24 oder 32 gesetzt werden. Falls dabei Probleme auftreten, sollte der Wert auf 8 zurückgesetzt werden. Später kann man dann versuchen, das Problem zu beheben. Der 24 Bit-Modus sollte keine Verwendung finden, da damit einige bekannte Programme wie netscape Probleme haben. Statt dessen sollte der 32 Bit-Modus verwendet werden, wenn eine Echtfarbendarstellung benötigt wird.

Modes

Dies ist die Liste mit Namen der verschiedenen Videomodi, die durch die Verwendung der `ModeLine`-Direktive in der `Monitor`-Sektion definiert worden sind. In dem oben aufgeführten Abschnitt sind `ModeLine`-Zeilen `"1024x768"`, `"800x600"` und `"640x480"` benannt. Deswegen stellt sich die Zeile `Modes` wie folgt dar:

```
Modes    "1024x768" "800x600" "640x480"
```

Der zuerst angegebene Modus in dieser Zeile wird standardmäßig nach dem Start von XFree86 eingestellt. Während des Betriebes von XFree86 kann mit Hilfe der Tastenkombination `Strg-Alt-numerisch +` und `Strg-Alt-numerisch -` zwischen den angegebenen Modi gewechselt werden.

Es hat sich bei der Erstkonfiguration von XFree86 vorteilhaft erwiesen, einen Videomodus mit einer niedrigen Auflösung, wie z.B. 640x480, zu benutzen, da sich diese Auflösung auf den meisten Systemen problemlos darstellen läßt. Ausgehend von dieser Basiskonfiguration kann anschließend die Datei `XF86Config` modifiziert werden, um auch höhere Auflösungen zu erzielen.

Virtual

Kennzeichnet die Größe des virtuellen Desktops. XFree86 besitzt die Fähigkeit, zusätzlichen Speicher der Grafikkarte für die Vergrößerung der Oberfläche zu benutzen. Bewegt sich der Mauszeiger über den Bildschirmrand hinaus, so verschiebt sich der Ausschnitt und der zusätzliche Bereich wird sichtbar. Bei einer Bildschirmauflösung von z.B. 800x600 Bildpunkten kann `Virtual` auf die von der Grafikkarte maximal unterstützten Auflösung gesetzt werden. Eine Grafikkarte mit 1 MB RAM unterstützt 1024x768 Bildpunkte mit einer Farbtiefe von 8 Bits je Bildpunkt, eine Karte mit 2 MB RAM dagegen 1280x1024 Bildpunkte bei einer Farbtiefe von 8 Bits je Bildpunkt oder 1024x768 Bildpunkte bei einer Farbtiefe von 16 Bits je Bildpunkt. Selbstverständlich kann der gesamte virtuelle Desktop nicht auf einmal dargestellt werden, dennoch kann man ihn insgesamt benutzen.

Die Virtual-Option stellt eine gute Möglichkeit dar, um den Speicher der Grafikkarte nutzbar zu machen. Dennoch ist sie ziemlich begrenzt und bietet kaum Bedienungsmöglichkeiten. Zur Nutzung eines wahren virtuellen Desktop sollte statt dessen fvwm2 oder ein ähnlicher Fenstermanager benutzt werden. fvwm2 bietet die Möglichkeit zur Darstellung beliebig vieler virtueller Bildschirme, zwischen denen mit Hilfe des Desktopmanagers umgeschaltet werden kann. So braucht nur der aktuelle Bildschirm und nicht der gesamte Desktop in den Speicher der Grafikkarte gestellt zu werden. Für weitere Details sollte die Manual-Seite über fvwm2 herangezogen werden. Die meisten Linux-Systeme setzen standardmäßig fvwm2 als Fenstermanager ein.

ViewPort

Setzt bei Benutzung der oben beschriebenen Virtual-Option die Koordinaten der linken, oberen Ecke des virtuellen Desktops beim Start von XFree86. Standardkoordinaten für Virtual sind oft 0 0. Werden für Virtual keine Angaben gemacht, erhält man einen zum virtuellen Desktop zentrierten Bildschirmausschnitt.

Für diese Sektion existieren noch viele andere Optionen. Eine komplette Beschreibung enthält die Manual-Seite zu XF86Config. Für die anfängliche Konfiguration von XFree86 sind diese Optionen aber nicht notwendig.

6 Spezifikation der Grafikkarte

Die XF86Config-Datei ist bis auf die vollständigen Informationen über die Grafikkarte soweit fertiggestellt. Zur Ermittlung dieser restlichen Informationen wird der X-Server selber herangezogen. Die so gewonnenen Spezifikationen der Grafikkarte werden anschließend in die XF86Config-Datei eingetragen.

Anstatt den X-Server zur Ermittlung der Informationen zu starten, können auch die Dateien modeDB.txt, AccelCards sowie Devices benutzt werden. Sie enthalten die entsprechenden XF86Config-Werte für eine Vielzahl von Grafikkarten und befinden sich in dem Verzeichnis /usr/X11R6/lib/X11/doc. Zusätzlich existieren verschiedene README-Dateien für bestimmte Chipsätze. Die in diesen Dateien gefundenen Informationen über die zu verwendende Grafikkarte mit Bildpunkt-Taktfrequenzen, Chipsatz usw. können in der Konfigurationsdatei XF86Config natürlich benutzt werden. Fehlende Spezifikationen sind mit der nun nachfolgend beschriebenen Methode zu ermitteln.

Als Beispiel dient die Konfiguration einer #9 GXE 64-Grafikkarte, die über einen S3-Chip verfügt und deshalb den XF86_S3-Server benutzt. Die hier beschriebene Methode kann auch auf jede andere Karte angewendet werden.

Zuerst muß der von der Grafikkarte benutzte Chipsatz bestimmt werden. Dazu kann man das Programm SuperProbe verwenden, das sich im Verzeichnis /usr/X11R6/bin befindet. Man beachte, daß der Name des Chipsatzes, so wie er von dem X-Server verwendet wird, erforderlich ist. Um ihn zu ermitteln, wird das Kommando

```
X -showconfig
```

eingegeben, das sämtliche dem X-Server bekannten Chipsatznamen auflistet. Die Manual-Seiten für den entsprechenden X-Server enthalten auch diese Namen. Für den beschleunigten XF86_S3-Server erhält man z.B. mit Hilfe von X -showconfig folgende Informationen:

```
XFree86 Version 3.3.2 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      mmio_928, s3_generic
```

Die gültigen Chipsatznamen für diesen Server lauten mmio_928 und s3_generic. Die XF86_S3-Manual-Seite beschreibt diese Chipsätze und welche Grafikkarten sie benutzen. Im Fall der #9 GXE 64-Grafikkarte wird mmio_928 verwendet.

Falls der Chipsatzname noch unbekannt ist oder Zweifel bestehen, welcher auszuwählen ist, kann wiederum der X-Server benutzt werden, um ihn zu ermitteln. Bei der Verwendung der Bash-Shell wird das Kommando durch

```
X -probeonly > /tmp/x.out 2>&1
```

gestartet. Wird dagegen die C-Shell benutzt, so muß

```
X -probeonly &> /tmp/x.out
```

eingegeben werden.

Das Kommando sollte abgesetzt werden, solange das System noch nicht geladen ist, d.h., solange sich keine andere Aktivität im System ereignet. Da der Befehl auch die Bildpunkt-Taktfrequenzen der Grafikkarte ermittelt, können zusätzliche Systembelastungen das Ergebnis verfälschen.

Die von dem oben angegebenen Kommando erzeugte Ausgabe, die sich in der Datei `/tmp/x.out` befindet, enthält z.B. die folgenden Zeilen:

```
XFree86 Version 3.3.2 / X Window System
(protocol Version 11, revision 0, vendor release 6000)
Operating System: Linux
Configured drivers:
  S3: accelerated server for S3 graphics adaptors (Patchlevel 0)
      mmio_928, s3_generic
    ...
(--) S3: card type: 386/486 localbus
(--) S3: chipset:   864 rev. 0
(--) S3: chipset driver: mmio_928
```

Der Server, in diesem Fall der `XF86_S3`, kennt demnach zwei gültige Chipsatznamen: `mmio_928` und `s3_generic`, wobei er speziell für die verwendete Grafikkarte den `mmio_928`-Chipsatz ermittelt hat.

Der so gefundene Name des Chipsatzes wird anschließend in die Zeile `Chipset` des `Device`-Abschnitts der `XF86Config`-Datei eingetragen. Zum Beispiel:

```
Section "Device"
    # Den Identifier haben wir bereits ...
    Identifier "#9 GXE 64"
    # Fügen Sie diese Zeile hinzu:
    Chipset "mmio_928"
EndSection
```

Als nächstes sind die von der Grafikkarte benutzten Bildpunkt-Taktfrequenzen zu bestimmen. Die Bildpunkt-Taktfrequenz (dot-clock) gibt die Anzahl der Bildpunkte an, die die Grafikkarte je Zeiteinheit zum Monitor senden kann. Wie bereits an oberer Stelle gezeigt, besitzt jede Bildschirmauflösung ihre eigene Bildpunkt-Taktfrequenz. Die von der Grafikkarte unterstützen Werte gilt es nun im folgenden zu bestimmen.

Zuerst sollte man aber in den bereits erwähnten Dateien `modeDB.txt` usw. überprüfen, ob dort die benötigten Bildpunkt-Taktfrequenzen für die zu benutzende Grafikkarte aufgelistet sind. Sie werden gewöhnlich in einer Liste mit 8 oder 16 Werten angegeben. Die Einheit ist MHz. Die Datei `modeDB.txt` gibt z.B. für die Grafikkarte Cardinal ET4000 folgende Zeilen aus:

```
# Chip    RAM   virtual   clocks                             default-mode  flags
  ET4000  1024  1024 768  25  28  38  36  40  45  32   0    "1024x768"
```

Wie das Beispiel zeigt, sind die Bildpunkt-Taktfrequenzen für diese Karte: 25, 28, 38, 36, 40, 45, 32 und 0 MHz.

In dem `Device`-Abschnitt der `XF86Config`-Datei wird demnach eine Zeile `Clocks` hinzugefügt, die die Liste der Bildpunkt-Taktfrequenzen enthält. Bezogen auf das Beispiel wird die Zeile

```
Clocks 25 28 38 36 40 45 32 0
```

nach der `Chipset`-Zeile in die `Device`-Sektion eingetragen. Man beachte, daß die Reihenfolge der Werte sehr wichtig ist. Die Liste darf weder neu sortiert noch dürfen mehrfach auftretende Werte entfernt werden.

Sind die Bildpunkt-Taktfrequenzen einer speziellen Grafikkarte nicht bekannt, so kann auch hier der X-Server wieder benutzt werden, diese Werte selber zu ermitteln. Mit dem weiter oben beschriebenen Kommando `X -probeonly` erhält die Ausgabe unter anderem eine Zeile mit den gewünschten Werten. Für die Grafikkarte #9 GXE 64 sieht sie z.B. wie folgt aus:

```
(--) S3: clocks:  25.18  28.32  38.02  36.15  40.33  45.32  32.00  00.00
```

Sämtliche ausgegebenen Werte werden der `Clocks`-Zeile hinzugefügt. Passen nicht alle Werte in eine Zeile, da oftmals mehr als 8 Werte angezeigt werden, können sie auf mehrere Clocks-Zeilen in der `XF86Config`-Datei verteilt werden. An dieser Stelle noch einmal der Hinweis, daß die Liste mit den Werten in der ausgegebenen Reihenfolge beibehalten werden muß!

Bei der Ermittlung der Bildpunkt-Taktfrequenzen mit Hilfe des Kommandos `X -probeonly` darf es keine `Clocks`-Zeile in der `Device`-Sektion von `XF86Config` geben. Notfalls kann sie auskommentiert werden. Beim Vorhandensein dieser Zeile wird der Server die Werte *nicht* selber ermitteln, sondern die bereits in der `XF86Config`-Datei aufgeführten Werte benutzen.

Man beachte, daß einige beschleunigte Grafikkarten einen programmierbaren Clockchip benutzen. Für weitere Details kann die `XF86_Accel`-Manual-Seite herangezogen werden. Dies gilt besonders für `S3`-, `AGX`- und `XGA-2`-Karten. Dieser Chip kann jeweils die Bildpunkt-Taktfrequenz erzeugen, die der X-Server für einen bestimmten Mode braucht. Man ist also nicht mehr auf diskrete, vorgegebene Bildpunkt-Taktfrequenzen beschränkt. In diesem Fall findet man in keiner der oben genannten Dateien eine Liste von Werten für die zu benutzende Karte oder sie enthalten nach der Anwendung von `X -probeonly` höchstens 1 oder 2 feste Werte, ansonsten nur Duplikate oder Nullwerte.

Bei der Benutzung von Grafikkarten mit einem programmierbaren Clockchip wird in der `XF86Config`-Datei anstelle einer `Clocks`- eine `ClockChip`-Zeile verwendet. `ClockChip` enthält den Namen des von der Karte benutzten Clockchips. Die Manual Pages der verschiedenen Server enthalten sämtliche Namen. Die `README.S3`-Datei gibt z.B. für verschiedene S3-864-Grafikkarten den ICD2061A-Clockchip an. Dieser Name wird dann anstelle von `Clocks` in die Zeile

```
ClockChip "icd2061a"
```

der `XF86Config`-Datei eingetragen. In dem `Device`-Abschnitt steht diese Zeile, wie auch schon `Clocks` hinter der `Chipset`-Zeile. Einige beschleunigte Grafikkarten erforden zusätzlich die Spezifizierung des verwendeten RAMDAC-Chips, dessen Name in die `Ramdac`-Zeile der `XF86Config`-Datei einzutragen ist. Die Manual-Seite von `XF86_Accel` beschreibt diese Option. Gewöhnlich wird die RAMDAC vom X-Server korrekt ermittelt.

Einige Typen von Grafikkarten erfordern zusätzliche Angaben verschiedener Optionen in der `Device`-Sektion von `XF86Config`. Diese Optionen werden ausführlich auf den Manual-Seiten der entsprechenden Server sowie in den verschiedenen Dateien, wie z.B. `README.cirrus` oder `README.S3`, beschrieben. Sie können mit Hilfe der `Option`-Zeile gesetzt werden. Die #9 GXE 64- Grafikkarte erfordert z.B. zwei Optionen:

```
Option "number_nine"
Option "dac_8_bit"
```

Normalerweise kann der X-Server auch ohne diese Optionen betrieben werden. Jedoch sind sie zum Erzielen der größtmöglichen Leistung notwendig. Es existiert noch eine Vielzahl solcher Optionen, die hier aber nicht weiter aufgeführt werden. Jede dieser Optionen hängt von einer bestimmten Grafikkarte ab. Sowohl die Manual-Seiten über die X-Server als auch die verschiedenen Dateien in `/usr/X11R6/lib/X11/doc` beschreiben diese Optionen ausführlich, falls eine von ihnen benutzt werden muß.

Nachdem alle Einträge in die `Device`-Sektion vollzogen sind, sieht sie für die #9 GXE 64-Grafikkarte folgenderma-ßen aus:

```
Section "Device"
    # Device-Sektion speziell für die #9 GXE 64!
    Identifier "#9 GXE 64"
    Chipset "mmio_928"
    ClockChip "icd2061a"
    Option "number_nine"
    Option "dac_8_bit"
EndSection
```

Die meisten Grafikkarten werden anstelle der `ClockChip`-Zeile eine `Clocks`-Zeile erfordern. Man beachte, daß der oben angegebene `Device`-Abschnitt nur für die spezielle Grafikkarte #9 GXE 64 gültig ist. Er ist hier nur als Beispiel aufgeführt.

7 Starten von XFree86

Nach Fertigstellung der Konfigurationsdatei `XF86Config` kann nun der X-Server gestartet werden. Zuerst sollte sichergestellt sein, daß `/usr/X11R6/bin` im Pfad eingetragen ist.

Das Kommando zum Starten von XFree86 lautet:

```
startx
```

Der Befehl startx startet sowohl den X-Server als auch die Kommandos der sich im Home-Verzeichnis befindenden Datei `.xinitrc`. `.xinitrc` ist einfach ein Shell-Skript, das die zu startenden X-Clients enthält. Falls diese Datei im Home-Verzeichnis nicht existiert, benutzt das System die Standarddatei in `/usr/X11R6/lib/X11/xinit/xinitrc`.

Eine Standard-`.xinitrc`-Datei sieht z.B. wie folgt aus:

```
#!/bin/sh

xterm -fn 7x13bold -geometry 80x32+10+50 &
xterm -fn 9x15bold -geometry 80x34+30-10 &
oclock -geometry 70x70-7+7 &
xsetroot -solid midnightblue &

exec twm
```

Dieses Skript startet zwei `xterm`-Clients, eine `oclock`-Uhr und setzt die Hintergrundfarbe des Root-Fensters auf `midnightblue`. Anschließend wird der Fenstermanager `twm` aufgerufen. Man beachte, daß `twm` mit der Shell-Anweisung `exec` ausgeführt wird. Dieses führt dazu, daß der `xinit`-Prozeß durch `twm` ersetzt wird. Nach Beendigung des `twm`-Prozesses fährt auch der X-Server herunter. Mit Hilfe des Hauptmenüs kann veranlaßt werden, den `twm` zu beenden. Dazu führt man den Mauszeiger auf den Desktop-Hintergrund, drückt die Maustaste 1 und wählt anschließend mit der Maus aus dem eingeblendeten Pop-Up-Menü den Befehl: *Exit Twm*.

Es sollte sichergestellt sein, daß das letzte Kommando in der `.xinitrc`-Datei mit der `exec`-Anweisung gestartet und daß es nicht durch das Hinzufügen eines kaufmännischen UND (`&`) am Ende der Zeile als Hintergrundprozeß aufgerufen wird, da ansonsten der X-Server beendet wird, sobald alle Clients aus der `.xinitrc`-Datei gestartet wurden.

Alternativ kann der X-Server durch die Tastenkombination `Strg-Alt-Rücktaste` beendet werden. Dies beendet den X-Server direkt und somit auch den Fenstermanager.

Die oben angegebene `.xinitrc`-Datei stellt eine sehr einfache Desktop-Konfiguration dar. Viele mächtige Programme und Konfigurationen sind mit einem geringen Arbeitsaufwand an der `.xinitrc`-Datei möglich. Der `fvwm2`-Fenstermanager stellt z.B. einen virtuellen Desktop bereit, dessen Farben, Schriftarten, Fenstergrößen und -positionen usw. vom Anwender nach »Herzenslust« eingestellt bzw. definiert werden können. Erscheint das X-Window-System anfänglich als eher schlicht und einfach, so erweist es sich nach benutzerdefinierter Konfiguration doch als sehr mächtig.

Dem Einsteiger in das X-Window-System sei das Buch *The X Window System: A User's Guide* empfohlen, da die Benutzung und Konfiguration von X zu umfangreich ist, um hier ausführlich behandelt zu werden. Für einen ersten Anhaltspunkt können die Manual-Seiten zu `xterm`, `oclock` und `fvwm2` herangezogen werden.

8 Problemlösungen

Oftmals stellen sich beim erstmaligen Starten des X-Servers Fehler ein. Diese werden fast immer durch falsche Einstellungen in der `XF86Config`-Datei verursacht. Gewöhnlich sind entweder die Monitor- oder die Bildpunkt-Taktfrequenzen nicht korrekt gesetzt. Ein sicheres Anzeichen für falsch gewählte Werte stellt z.B. eine durchlaufende Bildschirmanzeige bzw. verschwommene Bildkanten dar. Außerdem sollte die korrekte Spezifizierung des Chipsatzes der Grafikkarte, sowie der anderen Optionen im `Device`-Abschnitt der `XF86Config`-Datei sichergestellt sein. Auch muß der richtige X-Server verwendet werden und `/usr/X11R6/bin/X` muß mit Hilfe eines symbolischen Links auf diesen Server verweisen.

Schlagen alle Versuche fehl, den X-Server zu starten, so kann er mit Hilfe des Kommandos

```
X > /tmp/x.out 2>&1
```

direkt aufgerufen werden und sämtliche Warnungen und Fehler des Servers werden in die Datei `/tmp/x.out` umgeleitet. Anschließend, nach Beendigung des Servers mit der Tastenkombination `Strg-Alt-Rücktaste`, kann der Inhalt der Datei `x.out` untersucht werden.

Die in der XFree86-Distribution enthaltene Datei `VideoModes.doc` stellt zahlreiche Hinweise bei der Ermittlung der richtigen Werte für die `XF86Config`-Datei bereit.

Mit Hilfe der Tastenkombinationen `Strg-Alt-numerisch +` und `Strg-Alt-numerisch -` kann zwischen den in der `Modes`-Zeile der `Screen`-Sektion von `XF86Config` aufgeführten Grafikmodi umgeschaltet werden. Falls die höchstmögliche Auflösung fehlerhaft ist, besteht somit die Möglichkeit, auf eine niedrigere Auflösung umzuschalten. Dadurch erhält man einen guten Überblick, welche Abschnitte der X-Konfiguration korrekt arbeiten.

Zusätzlich können die Einstellregler des Monitors für die horizontale und vertikale Bildlage bzw. für die vertikale Höhen- und die horizontale Breiteneinstellung überprüft werden. In vielen Fällen ist nur eine Korrektur mit Hilfe der Einstellregler notwendig, nachdem X gestartet worden ist. Scheint das Bild z.B. nach einer Seite hin verschoben, so kann dies durch eine Justierung der Kontrollregler korrigiert werden.

Es existieren verschiedene Newsgruppen zum Thema XFree86, wie z.B.:

- `comp.windows.x.i386unix`
- `de.comp.os.unix.linux.x`

Sie sind eine gute Adresse, wo man eventuell Hinweise bezüglich der zu benutzenden Videokonfiguration erhalten kann, falls man dort »jemandem in die Arme läuft«, der das gleiche Problem hat bzw. es schon gelöst hat.

9 Copyright