

# Inhaltsverzeichnis

<b>1</b>	<b>XFree86 3.n</b>	<b>3</b>
1.1	Konfiguration des X-Servers	5
	Kleines X-Glossar	6
	Grundlagenwissen zu Monitor und Grafikkarte	8
	Der Konfigurationsprozess	10
	Das Konfigurationsprogramm XF86Setup	13
	Das Konfigurationsprogramm xf86config	16
	Das Programm xvidtune	17
	Der Aufbau von XF86Config	18
	X testen	24
1.2	Maus und Tastatur unter X	25
	Konfiguration der Tastatur	25
	Globale Tastenkombinationen	29
	Eingabefokus	30
	Konfiguration der Maus	30
	Verwendung der Maus	32
1.3	X-Ressourcen	32
	Grundlagen	33
	Ressourcendateien	34



# Kapitel 1

## XFree86 3.n

**HINWEIS:** Einige Abschnitte dieses Kapitel wurden unverändert aus der 5. Auflage des Buchs 'Linux: Installation, Konfiguration, Anwendung', Michael Kofler, Addison Wesley Longman 2000 übernommen.

Dieses PDF-Kapitel beschreibt die Konfiguration von XFree86 3.n. Natürlich gibt es auch in der 6. Auflage meines Linux-Buchs ein X-Kapitel, dieses bezieht sich aber ausschließlich auf XFree86 4.n.

Das hier vorliegende PDF-Kapitel soll eine Hilfestellung für alle Linux-Anwender sein, die – aus welchen Gründen immer – noch mit XFree86 3.n arbeiten. Der Text wird allerdings *nicht* mehr weiter gewartet, und ich kann auch keine Fragen dazu beantworten.

**Mehr Informationen zum Linux-Buch von Michael Kofler finden Sie hier:**

<http://www.kofler.cc>

Das X Window System (kurz X) stellt eine Sammlung von Funktionen und Protokollen dar, mit deren Hilfe grafische Informationen auf dem Bildschirm ausgegeben und Maus und Tastatur verwaltet werden. Diese Funktionen stehen auch für den Netzbetrieb zur Verfügung.

XFree86 ist eine freie Implementierung dieses X Window Systems. XFree86 stand ursprünglich nur für Betriebssysteme mit Intel-Prozessoren zur Verfügung (386, 486 etc., daher die Zahl 86 im Namen), mittlerweile läuft es aber auch auf vielen anderen Prozessoren.

Dieses Kapitel beschreibt verschiedene Aspekte der Konfiguration des XFree86-Servers. Nebenbei lernen Sie einige Grundlagen zum Thema Grafikhardware. Weitere Themenschwerpunkte sind die Verwendung von Tastatur und Maus unter X, der Umgang mit X-Ressourcen sowie einige Interna wie die Installation zusätzlicher Zeichensätze.

X ist die Basis für eine grafische Benutzeroberfläche für Linux. Der Begriff 'Benutzeroberfläche' ist unter Unix/Linux aber mit Vorsicht zu genießen: Im Gegensatz zur MS-Windows- oder Macintosh-Welt werden das Aussehen und die Bedienung von X-Programmen durch die vom jeweiligen Programm benutzte Bibliothek, den gerade aktiven Windows-Manager etc. beeinflusst. X selbst stellt noch keine Benutzeroberfläche zur Verfügung. Mehr Informationen zu diesem Thema sowie zu `fvwm`, KDE und Gnome finden Sie im nächsten Kapitel.

## 1.1 Konfiguration des X-Servers

Die Installationsprogramme der diversen Linux-Distributionen funktionieren mittlerweile so gut, dass X sofort nach (oder bereits während) der Installation zur Verfügung steht. Dieses Kapitel blickt hinter die Kulissen der Konfiguration von XFree86 und zeigt, wie Sie die Konfiguration gegebenenfalls selbst verändern können. Kurz ein Überblick über den Aufbau dieses recht langen Abschnitts:

Glossar	Seite 6
Grundlagenwissen	Seite 8
Konfiguration von XFree86	Seite 10
Unterstützte Grafikkarten	Seite 11
Konfiguration mit XF86Setup	Seite 13
Konfiguration mit xf86config	Seite 16
Optimierung mit xvidtune	Seite 17
Manuelle Konfiguration	Seite 18
Konfiguration testen	Seite 24

### Dokumentation

Zu XFree86 existiert eine Menge Online-Dokumentation. Sehr ausführlich und aktuell sind die man-Seiten zu XFree86: Unter anderem existieren Seiten zu XFree86 (Überblick), zu xserver (allgemeine Optionen für den Betrieb von X) und je nach Installation zu XF86\_Mono, XF86\_VGA16, XF86\_SVGA sowie Texte zu spezifischen Grafikkarten, etwa XF86\_Mach64. Der man-Text zu XF86Config (achten Sie auf die korrekte Groß- und Kleinschreibung) liefert schließlich eine Beschreibung aller in der Konfigurationsdatei erlaubten Schlüsselwörter.

Einen guten Überblick gibt der HOWTO-Text zu XFree86. Schließlich befinden sich im Verzeichnis `/usr/X11R6/lib/X11/doc` viele README-Texte. Eine erste Einführung gibt `Quickstart.doc`, einen Überblick über die Veränderungen gegenüber der letzten Version erhalten Sie in `RELNOTES`. Die Dateien `README.Chip` enthalten Informationen zu bestimmten Grafik-Chips.

#### HINWEIS

Weitere Informationen zum XFree86-Projekt und zur gerade aktuellen Version finden Sie im Internet:

<http://www.XFree86.org>

Bei manchen Distributionen (etwa SuSE) werden die dort verfügbaren HTML-Dateien als Dokumentationspaket gleich mitgeliefert.

### Kommerzielle X-Server

Neben dem freien X-Server von XFree86 gibt es auch kommerzielle Server (MetroX, AcceleratedX), die sich insbesondere durch die Unterstützung von Grafikkarten auszeichnen, die von XFree86 nicht bzw. noch nicht unterstützt werden. (Der Hintergrund ist oft

<http://www.kofler.cc>

ein rechtlicher: Manche Hersteller von Grafikkarten – glücklicherweise immer weniger – sind nicht zu einer Offenlegung der Schnittstellen bereit. Der Quellcode zu XFree86 ist aber frei verfügbar und soll es auch bleiben.)

Der aktive X-Server wird durch den Link von `/usr/X11R6/bin/X` auf das Server-Programm bestimmt. Wenn Sie den X-Server wechseln möchten, müssen Sie also diesen Link ändern. Bei manchen Distributionen zeigt `/usr/X11R6/bin/X` auf `/etc/X11/X`. In diesem Fall zeigt dann erst dieser Link auf den Server. Weitere Informationen zu kommerziellen Servern und deren Konfiguration entnehmen Sie bitte der jeweiligen Produktdokumentation.

## Kleines X-Glossar

**X Window System:** Das X Window System (kurz X) bezeichnet eigentlich nur die Basisfunktionen zum Zeichnen von Punkten, Rechtecken etc. X beinhaltet auch ein Netzwerkprotokoll, das es ermöglicht, ein X-Programm auf Rechner A auszuführen und die Ergebnisse (via Netzwerk) auf Rechner B darzustellen. Die meisten Linux-Distributionen verwenden X11R6.3. (XFree86 4.0 basiert auf dem neueren X11R6.4.) Das X Window System wurde vom MIT (Massachusetts Institute of Technology) entwickelt.

**X-Server:** Der X-Server stellt die Schnittstelle zwischen dem X Window System und der Hardware (Grafikkarte, Maus) her. Unter Linux sind der freie Server von XFree86 und die beiden kommerziellen Server MetroX und AcceleratedX verbreitet.

XFree86 ist Teil von X11R6 und stellt eine freie Portierung des X-Systems für Rechner mit Intel 386-Prozessoren oder dazu kompatiblen Prozessoren dar. XFree86 wurde vom *XFree86 Project, Inc.* entwickelt ([XFree86@XFree86.org](mailto:XFree86@XFree86.org)). XFree86 zeichnet sich vor allem durch die Unterstützung zahlloser Grafikkarten aus. XFree86 ist nicht nur für Linux, sondern auch für andere Unix-Systeme verfügbar.

**Virtueller Bildschirm:** Der X-Server ist in der Lage, einen virtuellen Bildschirm zu verwalten, der größer ist als der tatsächlich am Monitor sichtbare Bildausschnitt. Beispielsweise wird der virtuelle Bildschirm zumindest auf 1024\*768 Punkte eingestellt, selbst wenn Ihr Monitor nur 800\*600 Punkte flimmerfrei darstellen kann. Die Veränderung des sichtbaren Ausschnitts erfolgt vollkommen verzögerungsfrei, sobald Sie die Maus an den Rand des Bildschirms bewegen.

**Windows-Manager:** Der Windows-Manager ist ein X-Programm, das für die Verwaltung der Fenster zuständig ist. Unter X stehen verschiedene Windows-Manager zur Verfügung, die sich durch eine unterschiedliche Bedienung und auch durch eine unterschiedliche optische Gestaltung der Fensterrahmen auszeichnen. Der am weitesten verbreitete Windows-Manager unter Linux war lange Zeit `fwm`; mit der Entwicklung von KDE und Gnome ist allerdings auch in die Welt der Windows-Manager neue Bewegung gekommen. Informationen zu Auswahl, Konfiguration und Bedienung der Windows-Manager finden Sie ab Seite ??.

**Virtueller Desktop:** Ein virtueller Desktop ist eine vergrößerte Benutzeroberfläche, die quasi aus mehreren neben- und übereinander angeordneten Bildschirmen besteht. Da Sie aber tatsächlich nur einen Monitor haben, können Sie immer nur einen dieser virtuellen Bildschirme sehen. Sie können aber sehr bequem zwischen diesen Bildschirmen wechseln und beispielsweise in einem Bildschirm einige Shell-Fenster, in einem zweiten Bildschirm den Emacs und in einem dritten Bildschirm das seitenfüllende Programm `ghostview` laufen lassen.

Virtuelle Desktops werden vom Windows-Manager oder vom Desktop-System (KDE, Gnome) verwaltet (nicht von X). Um die Verwirrung zu steigern, verwendet das Konfigurationsprogramm `ConfigXF86` den Begriff 'virtual Desktop', obwohl dort ein virtueller Bildschirm gemeint ist; die Nomenklatur ist also nicht immer einheitlich.

**X-Ressourcen:** Die meisten Konfigurationsdetails (etwa die Größe des Zeichensatzes, Farben etc.) der unter X laufenden Programme werden durch eine X-Ressourcendatei gesteuert. Diese Datei ist also die zentrale Anlaufstelle zur Konfiguration der X-Programme (aber nicht des X-Basisystems). Details dazu finden Sie ab Seite 32.

**Motif:** Motif ist eine grafische Oberfläche, die auf X aufsetzt. In Motif sind zusätzliche Steuerelemente (Widgets) definiert, die der Oberfläche ihr charakteristisches Aussehen (Look-and-Feel) geben. Zu Motif gehört ein eigenes Set von Anwendungsprogrammen (Editor, Dateimanager etc.). Motif ist lizenzpflichtig, darf also nicht wie Linux frei kopiert werden. Einzelne Programme mit einer Motif-ähnlichen Oberfläche wurden allerdings als Freeware entwickelt und sind auch unter Linux verfügbar.

**OpenLook:** OpenLook ist neben Motif eine weitere Oberfläche, die auf X aufbaut. OpenLook wurde vor einiger Zeit von Sun freigegeben (unter anderem, um so die Verbreitung zu fördern). Deswegen zählen der OpenLook-Windows-Manager und einige OpenLook-Anwendungsprogramme zum Lieferumfang mancher Linux-Distributionen.

**CDE:** CDE steht für Common Desktop Environment. Es handelt sich um eine Benutzeroberfläche, die wiederum auf Motif aufbaut und zusätzliche Komponenten zur Bedienungserleichterung enthält (Hilfesystem, Toolbars etc.). CDE ist der Versuch, eine einheitliche Benutzeroberfläche für diverse UNIX-Systeme zu schaffen. Unter Linux ist die Bedeutung von CDE wegen des relativ hohen Preises (zusätzlich zu dem von Motif) bislang aber gering.

**KDE und Gnome:** KDE und Gnome sind zwei freie Benutzeroberflächen und stellen moderne Alternativen zu Motif, Openlook und CDE dar. KDE besteht aus einem eigenen Windows-Manager und zahllosen Programmen und Utilities, die einen homogenen, einfach zu nutzenden Desktop ergeben. Gnome bietet im Prinzip dieselben Funktionen, basiert aber auf einer anderen Bibliothek und sieht daher ein wenig anders aus. Zudem hat Gnome keinen eigenen Windows-Manager. KDE und Gnome werden im nächsten Kapitel ausführlich beschrieben.

## Grundlagenwissen zu Monitor und Grafikkarte

Es ist vielleicht nicht unbedingt erforderlich, die in diesem Teilabschnitt vermittelten Informationen zu kennen, um eine X-Konfiguration durchzuführen. Es erleichtert aber den Konfigurationsprozess erheblich, wenn Sie nicht blind irgendwelchen Anweisungen folgen, sondern wirklich wissen, was Sie tun. Und bei Konfigurationsproblemen führt an diesen Hintergrundinformationen ohnedies kein Weg vorbei.

**Bildaufbau:** Das Bild am Monitor kommt zustande, indem ein Elektronenstrahl zeilenweise die ganze Bildschirmoberfläche überstreicht und dabei winzige Punkte zum Leuchten bringt (oder auch nicht, wenn der Punkt schwarz dargestellt werden soll). Um ein flimmerfreies Bild zu ermöglichen, wird dieser Prozess mehrere Male in der Sekunde (60 bis 100 Mal) wiederholt. Die Information, welcher Punkt der Bildschirmoberfläche in welcher Farbe dargestellt werden soll, liefert die Grafikkarte. Im Monitor steuert diese Information die Intensität des Elektronenstrahls.

**Horizontaler Rücklauf (HSync):** Die Abtastung des Bildschirms beginnt in der linken oberen Ecke des Bildschirms. Von dort bewegt sich der Elektronenstrahl nach rechts, überstreicht also die erste Zeile. Anschließend springt der Elektronenstrahl zurück zum Beginn der Zeile und bewegt sich gleichzeitig um eine Zeile nach unten. Während dieses horizontalen Rücklaufs hat der Elektronenstrahl die Intensität 0, so dass der Rücklauf am Bildschirm nicht sichtbar ist. Der horizontale Rücklauf wird durch den HSync-Impuls ausgelöst, der von der Grafikkarte an den Monitor übertragen wird. (Physikalisch wird der Rücklauf durch eine Veränderung der elektromagnetischen Felder realisiert, die den Elektronenstrahl im Bereich der Bildfläche ablenken. Solche Veränderungen können nicht unendlich schnell durchgeführt werden, d.h. der Rücklauf nimmt etwas Zeit (einige Mikrosekunden) in Anspruch.)

**Vertikaler Rücklauf (VSync):** Nachdem auf diese Weise alle Zeilen überstrichen wurden und sich der Elektronenstrahl jetzt also in der rechten unteren Ecke des Bildschirms befindet, wird durch den VSync-Impuls der vertikale Rücklauf ausgelöst. Während dieses Rücklaufs bewegt sich der Elektronenstrahl zurück in die linke obere Ecke des Bildschirms. Natürlich ist der Elektronenstrahl auch während dieser Zeit abgeschaltet – sonst gäbe es quer über den ganzen Bildschirm eine Leuchtspur. Der vertikale Rücklauf dauert aus technischen Gründen um Größenordnungen länger als der horizontale Rücklauf. Während dieser Zeit könnten mehrere Zeilen abgetastet werden.

**Frequenzen:** Die Kenndaten des Bildaufbaus werden durch drei Frequenzen charakterisiert (siehe unten). Als Einheit für Frequenzen gilt Hz (Hertz, also Schwingungen pro Sekunde). kHz und MHz geben analog 1000 bzw. 1.000.000 Schwingungen pro Sekunde an.

**Vertikale Bildfrequenz (vertical refresh):** Die vertikale Bildfrequenz gibt an, wie oft (pro Sekunde) die ganze Bildschirmfläche neu gezeichnet wird. Je größer diese Zahl ist, desto stabiler wirkt das Bild. Etwa ab 70 Bildern pro Sekunde (70 Hz) wirkt das Bild flimmerfrei.

**Horizontale Zeilenfrequenz (HSync):** Diese Frequenz gibt an, wie viele Zeilen pro Sekunde vom Elektronenstrahl überstrichen werden. Bei einer Bildauflösung von 640\*480

Punkten (also 480 Zeilen) und bei einer Bildfrequenz von 60 Hz beträgt die horizontale Zeilenfrequenz 31,5 kHz. (Falls Sie mitrechnen: 60 mal 480 ergibt tatsächlich nur 28,8 kHz. Die etwas höhere Frequenz ist notwendig, weil auch etwas Zeit für den vertikalen Rücklauf berücksichtigt werden muss.)

**Pixelfrequenz (Video-Bandbreite):** Diese Frequenz gibt die Anzahl der Bildpunkte (Pixel) an, die während einer Sekunde von der Grafikkarte an den Monitor übertragen werden. Bei 640\*480 Punkten und 60 Hz ergibt sich ein Wert von 18,4 MHz. Wegen des durch den vertikalen und horizontalen Rücklauf bedingten Overheads beträgt die Pixelfrequenz aber tatsächlich 25 MHz.

**Interlace-Modus:** Der Interlace-Modus ist eine zuverlässige Methode, um in kurzer Zeit Kopfweh und in etwas längerer Zeit ein Augenleiden zu bekommen. Dabei wird beim Bildschirmaufbau nicht jede Zeile berücksichtigt, sondern nur jede zweite Zeile. Beim ersten Bild wird also die 1., 3., 5. etc. Zeile aufgebaut, beim zweiten Bild die 2., 4., 6. etc. Auf diese Weise kann eine höhere Auflösung trotz geringerer Pixelfrequenz erzielt werden. Das Ergebnis ist allerdings ein leicht flimmerndes Bild, wenn Sie keinen Monitor mit extrem lange nachleuchtendem Phosphor verwenden.

**Der Monitor:** Der Bildaufbau hängt von zwei Hardware-Komponenten ab, vom Monitor und der Grafikkarte. Von den technischen Daten leichter zu verstehen ist sicherlich der Monitor. Bei modernen Geräten sind drei Eckdaten wichtig:

- der vertikale Synchronisationsbereich (Bildfrequenz, z.B. 50 bis 90 Hz)
- der horizontale Synchronisationsbereich (Zeilenfrequenz, z.B. 30 bis 57 kHz)
- die maximale Bandbreite (Pixelfrequenz, z.B. 75 MHz)

Sehr alte Monitore können sich nicht nach Belieben auf irgendeine Frequenz in diesen Bereichen einstellen, sondern unterstützen nur einige fest vorgegebene Frequenzen.

Synchronisationsprobleme treten dann auf, wenn die Grafikkarte einen Datenstrom an den Monitor liefert, den dieser nicht exakt entschlüsseln kann. Das kann daran liegen, dass Synchronisationsimpulse zu früh, zu spät oder zu wenig ausgeprägt übertragen werden oder wenn die Eckdaten des Monitors nicht eingehalten werden.

VORSICHT

Wirklich gefährlich wird es, wenn die Grenzfrequenzen (insbesondere die maximale Zeilenfrequenz) überschritten werden, d.h. wenn die Grafikkarte zu viele Daten zu schnell anliedert. Dann kann der Elektronenstrahl des Monitors den Daten nicht mehr folgen, einzelne Bildschirmbereiche oder auch der ganze Bildschirm werden zu lange und zu intensiv bestrahlt, dadurch überhitzt und in relativ kurzer Zeit zerstört (innerhalb weniger Sekunden).

Eine Überschreitung der zulässigen Bandbreite ist dagegen ziemlich unkritisch (sofern die maximale Zeilenfrequenz eingehalten wird). Das Ergebnis ist im Regelfall ein unscharfes Bild, weil der Signalverstärker des Monitors überfordert ist. Aus diesem Grund wird die Bandbreite bei neuen XFree86-Konfigurationsprogrammen gar nicht mehr berücksichtigt.

**Die Grafikkarte:** Die Aufgabe der Grafikkarte besteht unter anderem darin, Pixeldaten für den Bildschirmaufbau zur Verfügung zu stellen. (Außerdem soll die Grafikkarte Zeichenoperationen unterstützen – das ist hier aber nicht von Interesse.) Alle zur Zeit gängigen Grafikkarten können dabei mehrere Grafikmodi mit unterschiedlicher Auflösung und Frequenz erzeugen – unter Umständen auch solche Grafikmodi, mit denen Ihr Monitor nicht mehr zurechtkommt.

Auch für die Grafikkarte spielen Frequenzen eine wichtige Rolle. Während der Monitor den in einer bestimmten Pixelfrequenz angelieferten Datenstrom verstehen muss, ist in der Grafikkarte die exakte Erzeugung dieser Pixelfrequenzen ein Problem. Alte Grafikkarten können oft nur wenige vorgegebene Frequenzen generieren, während moderne Karten einen frei programmierbaren Frequenzgenerator besitzen. Bei alten Grafikkarten zählen die verfügbaren Frequenzen zu den wichtigsten Konfigurationsdaten.

**Farben:** Aus der Sicht des Monitors stellen Farben kein Problem dar. Der Monitor bekommt einfach einen analogen Wert, der die Intensität eines Farbanteils (Rot, Grün oder Blau) widerspiegelt. Ob dabei 16, 256 oder mehr Farben dargestellt werden, spielt keine Rolle.

Für die Grafikkarte stellen Farben dagegen sehr wohl ein Problem dar: Zum einen müssen intern umso mehr Daten verarbeitet werden, je größer die Anzahl der Farben ist. Ein Bild mit 256 Farben beansprucht doppelt so viel Speicher wie eines mit 16 Farben. Dieser Speicher muss kontinuierlich in hoher Geschwindigkeit ausgelesen werden. Moderne Grafikkarten kommen dabei auf einen Datenstrom von über 500 MByte je Sekunde!

Zum anderen muss für jedes Pixel dessen digitales Muster in ein analoges Signal umgewandelt werden. Die dafür erforderlichen Digital-Analog-Converter (DAC) sind dazu aber nur bis zu einer bestimmten Frequenz in der Lage. Bei manchen Grafikkarten ist die DAC-Frequenz geringer als die theoretisch von der Karte unterstützte Bandbreite. Außerdem scheint es bei einigen Grafikkarten-Herstellern üblich zu sein, die Chips für den DAC ständig zu ändern, weswegen bei manchen Grafikkarten die exakte Bezeichnung des DAC-Chips genau in die Konfigurationsdatei eingetragen werden muss.

**Beschleunigerkarten:** Hinter diesem unschönen Wort verbergen sich Grafikkarten (Accelerated Cards), bei denen Zeichenoperationen durch einen eigenen Grafik-Chip unterstützt werden. Während bei sehr alten Grafikkarten der Rechner bei einer Grafikoperation jedes einzelne Pixel selbst verändern muss, sind die seit einigen Jahren marktüblichen Beschleunigerkarten in der Lage, manche Operationen wie das Verschieben eines Bildbereichs selbstständig durchzuführen. Der Rechner (die CPU) wird dadurch entlastet, das ganze System läuft schneller. Der Nachteil: Jeder Hersteller verwendet eigene Chips zur Beschleunigung elementarer Grafikoperationen, daher ist auch für jede Kartenfamilie ein eigener Grafiktreiber erforderlich.

## Der Konfigurationsprozess

Das Ziel der folgenden Ausführungen besteht darin, zu einem stabilen, möglichst hochauflösenden und flimmerfreien Bild am Monitor zu gelangen. Weitere Details der

X-Konfiguration (X-Ressourcen, Konfiguration des Windows-Managers) sind dann das Thema weiterer Abschnitte ab Seite ??.

Die zentrale Konfigurationsdatei für XFree86 heißt `XF86Config`. Diese Datei enthält im Textformat alle relevanten Daten Ihrer Grafikkarte und Ihres Monitors. Um diese Datei zu verändern, müssen Sie sich als `root` einloggen. Leider befindet sich `XF86Config` je nach Distribution an unterschiedlichen Orten.

```
/etc/X11/XF86Config      Corel, Debian, Mandrake, Red Hat
/etc/XF86Config          SuSE, XFree86 Default-Position
```

Bei der Erstellung von `XF86Config` ist Ihnen je nach Distribution eine ganze Palette von Konfigurationsprogrammen behilflich:

```
XF86Setup                Seite 13
xf86config               Seite 16
xvidtune                 Seite 17
Steuerzentrum (Corel)
Xconfigurator (Mandrake, Red Hat)
sax (SuSE)
```

VORSICHT

Verwenden Sie nie eine XFree86-Konfigurationsdatei, die Sie von einem Freund, Bekannten, über das Netz oder sonstwoher bekommen haben, ohne vorher den Monitorabschnitt an die technischen Daten Ihres Monitors anzupassen (siehe unten) – Sie riskieren sonst das vorzeitige Ableben Ihres Monitors!

### Unterstützte Grafikkarten

XFree86 unterstützt die meisten gängigen Grafikkarten – mit der Ausnahme ganz neuer Modelle, die inkompatibel zu ihren Vorgängern sind. (Allerdings werden nicht immer alle Fähigkeiten der Grafikkarte ausgenutzt.)

TIP

Informationen darüber, welche Grafikkarten zur Zeit von XFree86 unterstützt werden, finden Sie in der README-Datei des XFree86-Pakets (bzw. bei [www.xfree86.org](http://www.xfree86.org)).

XFree86 stellt für jede Familie von Grafikkarten einen eigenen X-Treiber zur Verfügung – beispielsweise `XF86_Mach64` für alle ATI-Grafikkarten mit einem Mach-64-Chip. Jeder dieser Treiber ist in einem eigenen Paket verpackt. Ein erster Schritt zur Inbetriebnahme von X besteht also darin, das richtige XF86-Treiberpaket zu installieren.

Der oder die Treiber werden in das Verzeichnis `/usr/X11R6/bin` installiert. Der gewünschte Treiber wird durch einen Link von `/usr/X11R6/bin/X` auf die Treiberdatei ausgewählt. Bei manchen Distributionen zeigt `/usr/X11R6/bin/X` auf `/etc/X11/X` oder auf `/var/X11R6/bin/X`. In diesem Fall zeigt dann erst dieser Link auf den Server.

<http://www.kofler.cc>

```

user# ls -l /usr/X11R6/bin/X
lrwxrwxrwx 1 root root X -> ../../../../etc/X11/X
user# ls -l /etc/X11/X
lrwxrwxrwx 1 root root /etc/X11/X -> ../../usr/X11R6/bin/XF86_Mach64

```

Falls Sie nicht wissen, welcher Chip in Ihrer Grafikkarte arbeitet, sollten Sie einen Blick in die Dokumentation zu Ihrer Grafikkarte werfen. Wenn das nichts hilft (oder keine Handbücher zu finden sind), können Sie das Programm `SuperProbe` ausführen. In den meisten Fällen ermittelt dieses Programm zuverlässig Ihre Grafikkarte. Im ungünstigsten Fall kann es aber auch einen Absturz verursachen, den Zeichensatz zerstören oder zu Synchronisationsproblemen führen. Sie sollten das Programm also nur ausführen, wenn sich nicht gerade mehrere Stunden Arbeit in noch ungesicherten Dateien befinden. Deutlicher formuliert: Melden Sie vor der Ausführung von `SuperProbe` alle nicht benötigten Dateisysteme mit `umount` ab, und führen Sie `sync` aus.

Auf meinem Rechner sieht das Ergebnis von `SuperProbe` folgendermaßen aus:

```

root# SuperProbe
First video: Super-VGA
    Chipset: ATI 88800GX-D (Port Probed)
    Memory: 2048 Kbytes
    RAMDAC:  ATI 68860 15/15/24-bit DAC w/pixel-mux
            (with 6-bit wide lookup tables (or in 6-bit mode))
            (programmable for 6/8-bit wide lookup tables)

    Attached graphics coprocessor:
        Chipset: ATI Mach64
        Memory: 2048 Kbytes

```

Eine weitere Möglichkeit, Informationen über die Grafikkarte zu erlangen, bietet die virtuelle Datei `/proc/pci` mit Informationen über die am PCI-Bus angeschlossenen Karten:

```

root# cat /proc/pci
PCI devices found: [...]
    Bus 1, device 0, function 0:
        VGA compatible controller: ATI Mach64 GB (rev 92).
        Medium devsel. Fast back-to-back capable. IRQ 226.
        Master Capable. Latency=64. Min Gnt=8.
        Prefetchable 32 bit memory at 0xe3000000.
        I/O at 0xd800.
        Non-prefetchable 32 bit memory at 0xe2000000.

```

In beiden Fällen ist somit klar, dass der `Mach64`-Treiber benötigt wird. Sollte es für Ihre Grafikkarte tatsächlich keinen geeigneten Treiber geben, hilft im Regelfall einer der drei folgenden Treiber weiter. Diese Treiber laufen auf den meisten VGA-Karten, nutzen aber nicht immer alle Merkmale und Besonderheiten der Karte aus.

<code>XF86_MONO</code>	Monochrome-Treiber (schwarzweiß)
<code>XF86_VGA16</code>	16-Farben, Standard-VGA
<code>XF86_SVGA</code>	Standardtreiber für Super-VGA (inkl. Matrox)

<http://www.kofler.cc>

TIP

Bevor Sie sich eine neue Grafikkarte kaufen, sollten Sie sich unbedingt vergewissern, dass diese Karte bereits von XFree86 unterstützt wird! XFree86 unterstützt zwar viele Grafikkarten, aber durchaus nicht alle (und auch nicht alle mit derselben Zuverlässigkeit). Informationen darüber, welche Grafikkarten unterstützt werden, finden Sie in den README-Dateien zu XFree86 bzw. auf dem XFree86-Webserver: [www.xfree86.org](http://www.xfree86.org)

## Das Konfigurationsprogramm XF86Setup

Seit Version 3.2 gilt XF86Setup als Standardkonfigurationsprogramm von XFree86. Allerdings liefern manche Distributionen stattdessen ihre eigenen Konfigurationsprogramme aus. Damit das Programm verwendet werden kann, müssen allerdings einige Voraussetzungen erfüllt sein:

- Das Programm muss installiert sein. Je nach Distribution ist das Programm in einem eigenen Paket enthalten. Zur Ausführung ist außerdem der VGA16-Treiber erforderlich (noch ein Paket, das oft nicht installiert ist).
- Ihre Grafikkarte und Ihr Monitor müssen den VGA-Standardmodus unterstützen (640\*480 Pixel bei 16 Farben).

### Maus

XF86Setup wird im Textmodus ausgeführt und startet zuerst den VGA16-X-Server, bevor es als X-Programm mit Mausunterstützung, Buttons etc. in Erscheinung tritt. Im Idealfall erkennt das Programm die angeschlossene Maus auf Anhieb, andernfalls sollte diese im ersten Konfigurationsschritt eingestellt werden. Die Bedienung des Programms kann zur Not ausschließlich mit **(Tab)** und den Cursortasten erfolgen. Zur Auswahl einiger Buttons kann auch **(Alt)+(Anfangsbuchstabe)** verwendet werden.

Die Auswahl des Maustyps ist bei No-Name-Fabrikaten oft Glückssache. Im ersten Versuch sollten Sie es mit *Microsoft compatible* (funktioniert auch für manche Drei-Tasten-Mäuse) oder mit *Mouseman* probieren. Die Herstellernamen sind nicht immer aussagekräftig, eine Logitech-Maus kann ohne weiteres das Microsoft- oder Mouseman-Protokoll befolgen. Mit **APPLY** übernehmen Sie die Einstellungen und können diese sofort testen. Dabei sollten Sie alle drei Maustasten drücken – die entsprechenden Felder im Maussymbol (rechts in Abbildung 1.1) sollten dabei schwarz werden.

Wenn die Maus überhaupt nicht reagiert, müssen Sie unter Umständen ein anderes Device auswählen. Am wahrscheinlichsten sind `/dev/mouse` (ein Link auf das tatsächliche Device), `/dev/ttyS0` oder `/dev/ttyS1` (die erste bzw. zweite serielle Schnittstelle) sowie `/dev/psaux` (PS/2-Mäuse).

Falls Sie mit einer Maus mit nur zwei Tasten arbeiten, sollten Sie außerdem die Option `EMULATE3BUTTONS` aktivieren. Sie können dann durch gleichzeitiges Drücken beider Maustasten die fehlende mittlere Maustaste simulieren. (Das ist allerdings nur eine Notlösung. Geben Sie ein paar Mark für eine Drei-Tasten-Maus aus, es lohnt sich!)

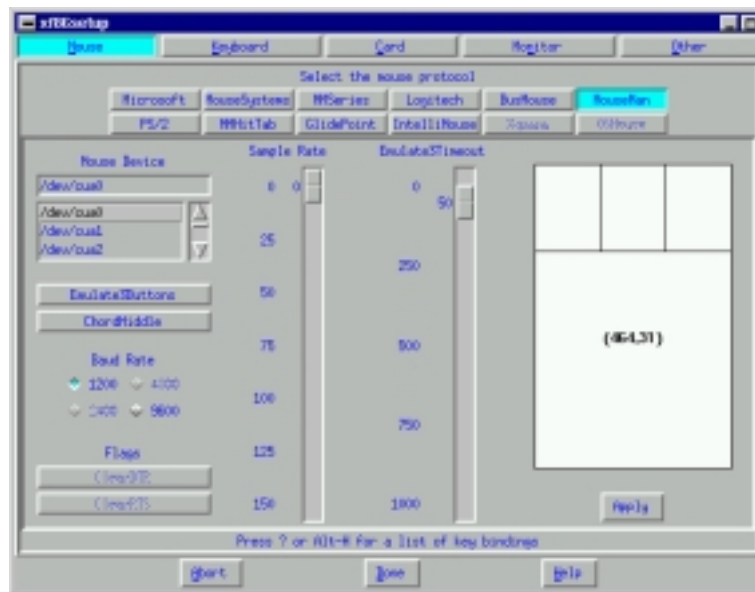


Abbildung 1.1: Einstellung der Maus mit XF86Setup

TIP

Falls Sie X auf einem Notebook mit Trackpoint als Mauseinsatz installieren, sollten Sie es mit dem Maustyp PS/2 und der Schnittstelle /dev/psaux versuchen. Sehr zu empfehlen ist zudem ein Blick auf die Linux-Notebook-Seite im Internet:

<http://www.cs.utexas.edu/users/kharker/linux-laptop>

### Tastatur

Nach der Konfiguration der Maus (vergessen Sie nicht APPLY!) betrifft der nächste Punkt die Tastatur. Die Einstellungen für eine deutsche Tastatur lauten:

MODEL: Generic 102 Key

LAYOUT: German

VARIANT: –

GROUP SHIFT/LOCK BEHAVIOUR: R-Alt switches group while pressed

Im Optionsfeld CONTROL KEY POSITION können Sie die CapsLock-Taste zu einer zusätzlichen Control-Taste machen und so CapsLock deaktivieren – aber das ist Geschmackssache.

### Grafikkarte

Zur Auswahl der Grafikkarte wird Ihnen eine schier endlose Liste angeboten, in der Sie Ihre Grafikkarte suchen müssen. Das klingt einfacher, als es manchmal ist – oder sind Sie sicher, ob Sie eine *ATI Graphics Expression with 68875 RAMDAC* haben oder ob *ATI*

<http://www.kofler.cc>

*Mach 64* die bessere Entscheidung ist? Bevor Sie Ihren Rechner zerlegen und durch einen Blick auf die Grafikkarte zu ergründen versuchen, welche Chips nun wirklich auf der Grafikkarte stecken, entscheiden Sie sich für die am wenigsten spezifische Angabe, von der Sie vermuten, dass sie korrekt ist. Wenn dadurch nur der richtige X-Treiber ausgewählt wird, erkennt dieser im Regelfall die Details selbst.

Nur wenn Sie sehr genau über Ihre Grafik-Hardware Bescheid wissen, können Sie über den Button `DETAILED SETUP` spezifische Angaben über zusätzliche Optionen, den vorhandenen Grafikspeicher etc. machen.

Wenn Sie vollkommen unschlüssig sind, sollten Sie das oben schon erwähnte Programm `SuperProbe` einsetzen, das in vielen Fällen die verwendete Grafik-Hardware erkennt (in seltenen Fällen aber auch einen Absturz verursachen kann!).

### Monitor

Das Konfigurationsprogramm möchte nun die zulässigen Bereiche für die horizontale Zeilenfrequenz (kHz) und für die vertikale Bildfrequenz (Hz) von Ihnen wissen. Sie können dazu entweder einen der vorgeschlagenen Monitortypen auswählen oder die Werte direkt eingeben.

Die erforderlichen Daten finden Sie im Handbuch oder Datenblatt zu Ihrem Monitor (und, wenn Sie Glück haben, sogar auf der Rückseite Ihres Monitors). Für eine ganze Reihe von Monitoren finden Sie diese Daten auch in der Datei `doc/monitors` (bei den Schlüsselwörtern `VertRefresh` und `HorizSync`). Hintergrundinformationen dazu, was diese Daten bedeuten, finden Sie ab Seite 8.

VORSICHT

Wenn Sie für die horizontale Zeilenfrequenz oder für die vertikale Bildfrequenz zu große Werte angeben, riskieren Sie, dass Ihr Monitor zerstört wird! XFree86 verwendet diese Werte, um eventuell nicht mit Ihrem Monitor verträgliche Grafikmodi von vornherein auszuschließen.

### Abschluss der Konfiguration

Im fünften Schritt von `XFree86Setup`, dem Menüpunkt `OTHER`, können Sie die Voreinstellungen unverändert übernehmen. Wenn Sie anscheinend `DONE` anklicken, wird der X-Server gemäß Ihren Einstellungen gestartet. Wenn alles klappt, erscheint auf Ihrem Monitor ein stabiles Bild. Sie können nun per Maus entscheiden, ob Sie die Einstellungen speichern, die Bildlage mit `xvidtune` (siehe Seite 17) noch optimieren möchten oder das Programm mit `QUIT` verlassen.

Wenn es dem Monitor nach dem Start von X dagegen nicht gelingt, sich zu synchronisieren, bestehen zwei Möglichkeiten: Die eine ist die Tastenkombination `(Alt)+(Strg)+(+)`, wobei die Taste `(+)` vom numerischen Tastenblock verwendet wird; damit können Sie in einen weniger hoch auflösenden Grafikmodus wechseln. Hilft auch das nicht, müssen



Abbildung 1.2: Einstellung des Monitors mit XF86Setup

Sie das Konfigurationsprogramm mit **(Alt)+(Strg)+(BackSpace)** abbrechen, anschließend neu starten und alle Konfigurationsschritte wiederholen.

## Das Konfigurationsprogramm xf86config

`xf86config` ist die textbasierte Alternative zu `XF86Setup`. Das Programm ist zwar nicht übermäßig komfortabel, führt aber im Regelfall ebenfalls zum Ziel. Die Einstellungen sind im Prinzip dieselben wie in `XF86Setup` – daher werden hier nur noch einige ergänzende Details beschrieben.

Nach der Monitor- und Grafikkarteneinstellung fragt `xf86config` nach Identifikations- und Beschreibungszeichenketten. Hier reicht es, wenn Sie einfach **(↵)** drücken und so die Defaultvorschläge bzw. leere Zeichenketten übernehmen. Der einzige Zweck dieser Zeichenketten besteht darin, Ihre `XF86Config`-Datei etwas 'lesbarer' zu machen.

Nach einer Rückfrage führt `xf86config` anschließend einen Start des VGA-Treibers mit der Option `-probeonly` durch. Dabei wird `x` nicht wirklich gestartet – es werden lediglich die von der Grafikkarte unterstützten Pixelfrequenzen festgestellt. Der X-Probelauf funktioniert allerdings nicht bei allen Grafikkarten. Daher ist es oft sinnvoll, diesen Punkt mit **(N)** zu überspringen.

Das Konfigurationsprogramm stellt jetzt eine Liste von Grafikmodi zur Auswahl. Dabei gibt die Abkürzung `bpp` die Bits per Pixel an: 8 Bit für 256 Farben, 16 Bit für 32.768 oder 65.536 Farben und 24 Bit für den Echtfarbenmodus ( $2^{24}$  Farben).

In den meisten Fällen kann die Liste so gelassen werden, wie sie ist. Sehr hoch auflösende Modi, die darin aufgezählt werden (etwa 1280\*1024 Pixel), werden beim Start von X automatisch eliminiert, wenn sie von Ihrer Grafikkarte oder von Ihrem Monitor nicht unterstützt werden.

Nach einer abschließenden Rückfrage speichert das Konfigurationsprogramm die soeben erstellte Konfigurationsdatei. Achten Sie darauf, dass die Datei am richtigen Ort gespeichert wird (bzw. kopieren Sie die Datei anschließend an den für Ihre Distribution erforderlichen Ort).

Das Konfigurationsprogramm `xf86config` hat leider die unangenehme Eigenheit, die Konfigurationsdatei mit zahllosen weiteren Abschnitten anzufüllen, die im konkreten Fall gar nicht benötigt werden. Das und die zahllosen Kommentare machen die Konfigurationsdatei länger und unübersichtlicher als notwendig. Details zur manuellen Konfiguration folgen ab Seite 18.

## Das Programm `xvidtune`

Das Programm `xvidtune` ist kein Konfigurationsprogramm, sondern vielmehr ein Hilfsmittel zur Optimierung einer bereits vorhandenen Konfigurationsdatei. (Zu `xvidtune` wird zur Zeit auch eine KDE-Variante namens `kvidtune` entwickelt.)

Mit `xvidtune` können Sie für alle Modi (Buttons `PREV` und `NEXT`) die Größe und Position des Bilds verändern. Die Änderungen der Bildposition können Sie wahlweise mit `TEST` testen (dann wird nach wenigen Sekunden der alte Modus wiederhergestellt) oder mit `APPLY` bleibend einstellen. Mit `(R)` (restore) gelangen Sie auch dann wieder zu einem stabilen Bild, wenn der Monitor nicht mehr synchronisiert und die Maus daher nicht mehr bedient werden kann. Sobald Sie das Programm beenden, werden alle Einstellungen in `XF86Config` gespeichert.

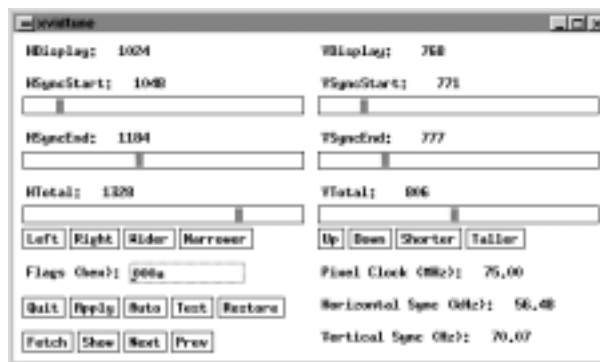


Abbildung 1.3: `xvidtune` zur Feineinstellung der Bildlage

## Der Aufbau von XF86Config

Dieser Abschnitt beschreibt die Konfigurationsdatei `XF86Config`. Sie benötigen diese Informationen, wenn Sie die Konfigurationsdatei manuell optimieren möchten – und das ist trotz der stark verbesserten Konfigurationsprogramme noch immer oft unvermeidlich. `XF86Config` besteht im Wesentlichen aus vier Teilen:

- Im ersten Teil werden globale Optionen (in erster Linie zu Maus, Tastatur und den Orten von Verzeichnissen) eingestellt (`Section Files`, `ServerFlags`, `Keyboard` und `Pointer`).
- Der zweite Teil enthält einen oder mehrere `Monitor`-Abschnitte. Darin werden die Eckdaten des Monitors und die vom Monitor unterstützten Grafikmodi angegeben.
- Der dritte Teil enthält einen oder mehrere `Device`-Abschnitte zur Einstellung der technischen Daten der Grafikkarte.
- Die Konfigurationsdatei endet mit einem oder mehreren `Screen`-Abschnitten. Dort werden einzelne `Monitor`- und `Device`-Abschnitte miteinander kombiniert.

VERWEIS

An dieser Stelle werden nur die Abschnitte `Monitor`, `Device` und `Screen` beschrieben. Die korrekte Einstellung dieser drei Abschnitte reicht aus, um zu einer stabilen Grafikdarstellung zu gelangen. Informationen zu den anderen Abschnitten erhalten Sie ab Seite 25 (Tastatur und Maus) bzw. ab Seite ?? (Zeichensätze).

Für eine korrekte Konfiguration wäre eigentlich nur jeweils *ein* `Monitor`-, `Device`- und `Screen`-Abschnitt erforderlich. Einige Konfigurationsprogramme (insbesondere `xf86config`) liefern allerdings schier endlose Konfigurationsdateien mit redundanten Informationen, die nur Verwirrung stiften. Welche Abschnitte werden nun tatsächlich verwendet?

Am leichtesten fällt die Beantwortung dieser Frage, wenn Sie vom letzten Teil der Konfigurationsdatei ausgehen. Von den `Screen`-Abschnitten ist derjenige gültig, der Ihrem Grafiktreiber entspricht. Wenn Sie mit einem Super-VGA-Treiber arbeiten (und `/usr/X11R6/bin/X` auch auf diesen Treiber zeigt), dann sind die `Screen`-Abschnitte für den Mono-Treiber, den 16-Farben-Treiber und die Gruppe der Accelerated-VGA-Treiber überflüssig. Sie können die Konfigurationsdatei ein wenig 'aufräumen' und diese Abschnitte einfach aus der Datei löschen. (Speichern Sie aber vorher die ursprüngliche Konfigurationsdatei unter einem eigenen Namen.)

Sobald Sie den relevanten `Screen`-Abschnitt identifiziert haben, kennen Sie auch die entscheidenden `Monitor`- und `Device`-Abschnitte. Deren Namen werden nämlich bei den Schlüsselwörtern `Monitor` und `Device` am Beginn des `Screen`-Abschnitts angegeben. Sie müssen jetzt nur noch die `Monitor`- bzw. `Device`-Abschnitte suchen, in denen dieselben Namen nach dem Schlüsselwort `Identifier` angegeben sind.

Wenn Sie mit dem Super-VGA-Treiber arbeiten, enthalten beispielsweise die drei folgenden Abschnitte alle relevanten Daten:

```
Section "Device"
```

<http://www.kofler.cc>

```

    Identifier "sigma legend II"
    ...
EndSection
Section "Monitor"
    Identifier "nec 4d"
    ...
EndSection
Section "Screen"
    Driver      "svga"          # für den Super-VGA-Treiber
    Device      "sigma legend II" # Verweis auf die Grafikkarte
    Monitor     "nec 4d"        # Verweis auf den Monitor
    ...
EndSection

```

**Der Device-Abschnitt:** Im Device-Abschnitt für die Grafikkarte geht es im Wesentlichen darum, die von der Grafikkarte unterstützten Pixel-Frequenzen aufzuzählen und (falls erforderlich) zusätzliche Informationen zu dem auf der Karte befindlichen RAM-DAC anzugeben.

Zuerst zu den Pixelfrequenzen: Im Regelfall werden einfach alle von der Grafikkarte unterstützten Frequenzen durch eine oder mehrere `clock`-Zeilen aufgezählt. Die Angaben erfolgen in MHz. Die Frequenzen können entweder mit `x -probeonly` ermittelt werden oder, falls das nicht funktioniert, aus den Dateien `Devices` oder `modeDB.txt` (Verzeichnis `/usr/X11R6/lib/X11/doc`) entnommen werden.

Die meisten Grafikkarten unterstützen nicht nur eine bestimmte Anzahl starrer Frequenzen, sondern sind mit einem frei programmierbaren Frequenzgenerator ausgestattet. In diesem Fall entfallen die `clock`-Zeilen. Stattdessen muss der Typ des Frequenzgenerators mit dem Schlüsselwort `clockchip` angegeben werden.

Alle weiteren Einstellungen und Optionen hängen stark von der jeweiligen Grafikkarte ab. Sie sind in den `README`-Dateien sowie (sehr ausführlich) in den treiberspezifischen `man`-Seiten (z.B. `man XF86_SVGA`) beschrieben.

Das folgende Beispiel zeigt den Device-Abschnitt für eine Grafikkarte in einem alten Rechner des Autors. (Es handelt sich um eine ET4000 Super-VGA-Karte.) Entscheidend war hier die richtige `clocks`-Zeile (sie wurde aus `modeDB.txt` kopiert) und die Option `legend`, ohne die die Grafikkarte nicht funktioniert (`man XF86_SVGA`).

```

Section "Device"
    Identifier "sigma legend II"
    VendorName "sigma"
    BoardName "legend II"
    Clocks 25 28 0 40 36 40 45 58 32 36 31 35 50 48 33 65
    Option "legend"
EndSection

```

Die folgenden Zeilen gelten für eine weitere Grafikkarte von mir. Entscheidend für die korrekte Funktion war diesmal die Einstellung für die Größe des Grafikspeichers.

```

Section "Device"
    Identifier      "Primary Card"
    VendorName     "Unknown"
    BoardName      "ATI Xpert@Work, 3D Rage Pro"
    VideoRam       8192
EndSection

```

**Monitor-Abschnitt:** Der Monitor-Abschnitt beginnt mit drei Frequenzangaben: Der maximalen Videobandbreite (Pixelfrequenz in MHz), dem zulässigen Bereich für die horizontale Zeilenfrequenz und dem zulässigen Bereich für die vertikale Bildfrequenz. Diese Angaben sollten eigentlich kein Problem darstellen. (Was diese Frequenzen bedeuten, wurde im Grundlagenabschnitt am Beginn dieses Kapitels erklärt.) Erheblich mehr Probleme bereiten dagegen die Grafikmodi, die ebenfalls hier eingetragen werden müssen. Für die Angaben sind zwei Syntaxvarianten möglich:

```

ModeLine "640x480" 25.175 640 664 760 800 480 491 493 525
# oder
Mode "640x480"
    DotClock      25.175
    HTimings      640 664 760 800
    VTimings      480 491 493 525
    Flags
EndMode

```

Die beiden obigen Angaben durch `ModeLine` oder `Mode` sind inhaltlich gleichwertig: Es wird ein Grafikmodus mit 640\*480 Pixeln beschrieben. Die Zeichenkette "640x480" ist gleichzeitig auch der Name dieses Modus. Der Zahlenwert 25.175 gibt die Pixelfrequenz (Videobandbreite) in MHz an. Die Pixelfrequenz muss mit einem der `clock`-Werte aus dem `Device`-Abschnitt übereinstimmen, damit der Modus gültig ist. Wenn mehrere gleichnamige Modi angegeben werden, verwendet X den ersten gültigen Modus.

Die nächsten vier Werte (in der Einheit Pixel) betreffen das horizontale Timing: Eine einzelne Bildschirmzeile mit 640 *sichtbaren* Pixeln wird in Wirklichkeit aus 800 *virtuellen* Pixeln zusammengesetzt. Die ersten 640 Pixel werden tatsächlich angezeigt. Während der verbleibenden 160 Pixel wird der Elektronenstrahl durch den `HSync`-Impuls zurück an den Beginn der nächsten Zeile bewegt. Während dieser Zeit hat der Elektronenstrahl die Intensität 0.

640	640 Bildschirmpixel anzeigen
664	24 weitere Pixel dunkel tasten
760	96 Pixel lang einen <code>HSync</code> -Impuls erzeugen
800	nochmals 40 Pixel dunkel tasten, d.h. insgesamt 800 virtuelle Punkte

Ganz analog wie beim horizontalen Timing sind auch die Angaben für das vertikale Timing (Einheit Bildschirmzeilen) zu interpretieren:

480	480 Zeilen anzeigen
491	11 Zeilen dunkel tasten

493 2 Zeilen lang einen VSync-Impuls erzeugen  
525 nochmals 32 Zeilen dunkel tasten, d.h. insgesamt 525 virtuelle Zeilen

Aus den jeweils letzten Werten der Vierergruppen und der Pixelfrequenz ergeben sich übrigens die horizontale Zeilenfrequenz und die vertikale Bildfrequenz: 25,175 MHz dividiert durch 800 Pixel pro Zeile ergibt eine Zeilenfrequenz von 31,469 kHz. Die Zeilenfrequenz, dividiert durch 525 Zeilen pro Bild, liefert die vertikale Bildfrequenz von 60 Hz. Die `ModeLine`-Angaben geben keine Auskunft über die Anzahl der Farben, die in diesem Modus zur Verfügung stehen – aus der Sicht des Monitors spielt das auch keine Rolle. (Viele Grafikkarten können hoch auflösende Modi allerdings nur mit einer geringen Zahl von Farben darstellen.)

Eine ganze Palette vordefinierter `ModeLine`-Zeilen finden Sie in der von `xf86config` erzeugten Konfigurationsdatei sowie in den Dateien `Monitors` und `README.Config`. Je nach den Fähigkeiten Ihrer Grafikkarte bzw. Ihres Monitors können bzw. müssen Sie diese Werte manuell anpassen, um ein optimales Bild zu erhalten.

Wenn Sie beispielsweise das Bild nach rechts verschieben möchten, müssen Sie den Abstand nach dem `HSync`-Impuls vergrößern. Damit die Gesamtzahl der Pixel (800) gleich bleibt, müssen Sie entweder die Zeit vor dem `HSync`-Impuls oder die Länge des `HSync`-Impulses verkürzen. Sie könnten es beispielsweise mit `640 660 752 800` probieren.

Wenn Sie das Bild verbreitern oder schmaler machen möchten, müssen Sie das Verhältnis zwischen der Anzahl der sichtbaren Pixel und der virtuellen Pixel verändern. Die folgenden Angaben machen das Bild ein wenig breiter: `640 660 752 790`. Allerdings verändern sich dadurch auch die Zeilen- und die Bildfrequenz ein wenig, womit nicht jeder Monitor zurechtkommt.

Bei vielen Grafikkarten können Sie sogar die sichtbare Auflösung verändern. Die folgenden Angaben erhöhen die horizontale Auflösung von 640 auf 656 Punkte: `656 676 772 800`.

Ganz analog können Sie auch die Angaben für das vertikale Timing variieren. Ob es dabei allerdings wirklich zu einem stabilen Bild kommt, hängt stark von den technischen Daten Ihres Monitors und Ihrer Grafikkarte ab.

**TIP**

Zur Feineinstellung können Sie auch das oben beschriebene Programm `xvidtune` oder dessen KDE-Variante `kvidtune` verwenden. Mit diesem Programm können Sie einen Teil der Timing-Werte im laufenden Betrieb von X interaktiv verändern.

An die `ModeLine`-Zeilen können verschiedene Optionen angefügt werden: `Interlace` für den Interlace-Modus (nur bei einer ungeraden Anzahl virtueller Zeilen möglich), `+/-HSync` und `+/-VSync`, um die Polarität des Synchronisationsimpulses zu verändern, sowie `Composite` und `+/-CSync` für ein zusammengesetztes Synchronisationssignal (enthält `HSync` und `VSync`).

Das folgende Beispiel zeigt den `Monitor`-Abschnitt meines fast schon steinzeitlichen NEC-4D-Monitors. Es werden drei Auflösungen definiert, 640\*480 Punkte bei 71 Hz, 816\*612 Punkte bei 75 Hz sowie 1024\*768 Punkte bei 63 Hz.

<http://www.kofler.cc>

```

Section "Monitor"
    Identifier "nec 4d"
    VendorName "nec"
    ModelName "4d"
    BandWidth 75
    HorizSync 30-57
    VertRefresh 50-90
    ModeLine "640x480" 31 640 664 704 832 480 489 492 520
    ModeLine "816x612" 50 816 824 960 1040 612 612 614 635
    ModeLine "1024x768" 65 1024 1065 1065 1260 768 780 785 810
EndSection

```

Als zweites Beispiel folgt der Monitor-Abschnitt für einen moderneren Bildschirm:

```

Section "Monitor"
    Identifier "iiyama 450"
    VendorName "unknown"
    ModelName "unknown"
    HorizSync 31.5-102
    VertRefresh 40-100
    Modeline "1280x1024" 157.50 1280 1344 1504 1728
    1024 1025 1028 1072 +hsync +vsync
    Modeline "1152x864" 137.65 1152 1184 1312 1536
    864 866 885 902 -hsync -vsync
    Modeline "1024x768" 115.50 1024 1056 1248 1440
    768 771 781 802 -hsync -vsync
EndSection

```

**Screen-Abschnitt:** Der Screen-Abschnitt gibt an, welcher Monitor-Abschnitt mit welchem Device-Abschnitt für welchen VGA-Treiber kombiniert werden soll. Der Abschnitt enthält einen oder mehrere Display-Teilabschnitte. Jeder Teilabschnitt gilt für eine bestimmte Farbtiefe (8, 16 oder 24 bpp, d.h. *bits per pixel*).

Jeder Display-Abschnitt enthält eine Liste aller Grafikmodi, die unterstützt werden sollen. Diese Liste wird mit dem Schlüsselwort `Modes` eingeleitet und enthält die Namen der oben in `ModeLine` definierten Grafikmodi. Dabei spielt auch die Reihenfolge eine Rolle – der erste Modus gilt als Defaultmodus. In die anderen Modi können Sie der Reihe nach mit `(Strg)+(Alt)+(+)` oder `(-)` (numerischer Tastenblock) wechseln.

Bei Grafikkarten mit ausreichend Speicher können Sie mit `Virtual` einen virtuellen Bildschirm definieren, der größer als der tatsächlich sichtbare Ausschnitt ist. Mit der Maus können Sie sich dann in diesem virtuellen Bildschirm verzögerungsfrei bewegen. Sie sollten allerdings nicht den gesamten zur Verfügung stehenden Speicher für den virtuellen Bildschirm verwenden, um etwas Platz für einen Font-Cache zu lassen (das führt zu einer signifikanten Beschleunigung). Mit dem Schlüsselwort `ViewPort` wird das erste sichtbare Pixel beim Start von X angegeben (nur von Interesse, wenn der virtuelle Bildschirm größer ist als die reale Auflösung).

TIP

Das Arbeiten mit einem virtuellen Bildschirm ist gewöhnungsbedürftig. Wenn Sie der durch manche Konfigurationsprogramme automatisch eingerichtete virtuelle Bildschirm stört, müssen Sie die `Virtual`-Anweisung auskommentieren. (In diesem Fall orientiert sich die Größe des virtuellen Bildschirms am höchstauflösenden Grafikmodus, der innerhalb des `Screen`-Abschnitts genannt wird.)

```
Section "Screen"          # für eine SVGA-Karte
    Driver "vga256"
    Device "sigma legend II"
    Monitor "nec 4d"
    Subsection "Display"
        Depth      8      # 8 bpp, also 256 Farben
        Modes      "1024x768" "640x480" "816x612"
        Virtual    1024 900
        ViewPort   0 0
    EndSubsection
EndSection
```

Bei einigen Accelerated-VGA-Treibern können gleichzeitig mehrere `Display`-Abschnitte mit unterschiedlicher Farbtiefe (8, 15, 16, 24 oder 32 bpp) angegeben werden (siehe man `XF86_Accel`). Beim Start des X-Servers wird normalerweise der 8-bpp-Modus aktiviert. Die `Display`-Abschnitte mit höheren Auflösungen kommen nur zur Geltung, wenn bei `startx` mit `-- -bpp 16` oder `24` explizit eine höhere Farbtiefe angegeben wird. Alternativ zu dieser Option können Sie im `Screen`-Abschnitt durch `DefaultColorDepth n` den Defaultmodus ändern.

```
Section "Screen"
    Driver      "Accel"
    Device      "Primary Card"
    Monitor     "Primary Monitor"
    DefaultColorDepth 24
    SubSection "Display"
        Depth      8
        Modes      "1280x1024" "1152x864" "1024x768"
    EndSubSection
    SubSection "Display"
        Depth      15
        Modes      "1280x1024" "1152x864" "1024x768"
    EndSubSection
    SubSection "Display"
        Depth      24
        Modes      "1280x1024" "1152x864" "1024x768"
    EndSubSection
EndSection
```

## X testen

Nach dem Erstellen von `XF86Config` wird X normalerweise mit dem Kommando `/usr/bin/startx` gestartet. Dieses Script startet nach dem X-Server auch einen Windows-Manager (z.B. `fvwm`, siehe nächstes Kapitel).

Beim Start von X werden noch in der Textkonsole einige Meldungen angezeigt. Sofern die Konfigurationsdatei keine syntaktischen Fehler enthält oder der vorgefundenen Hardware nicht eindeutig widerspricht, wird unmittelbar danach der Grafikmodus aktiviert. Im Idealfall erscheint jetzt sofort ein stabiles Bild – dann ist die Konfiguration schon so gut wie gelungen.

Je nach Konfiguration werden beim Start von X gleich einige Anwendungsprogramme gestartet. Darüber hinaus können Sie mit den Maustasten ein Menü zum Start weiterer Programme aufrufen. Zu Anfang ist das wichtigste Programm sicherlich `xterm`. Es ermöglicht wie in einer Textkonsole die Eingabe von Kommandos und den Start anderer Programme.

VERWEIS

Weitere Informationen zu den unterschiedlichen Möglichkeiten, X zu starten, den Windows-Manager auszuwählen und zu konfigurieren finden Sie ab Seite ???. Einige X-Programme sind in Kapitel ??? ab Seite ??? beschrieben.

Details zum Start von X finden Sie auch in den man-Seiten zu `startx`, `openwin`, `xdm`, `xinit` und `x`.

Wenn nach der Ausführung von `startx` dagegen kein sichtbares Bild erscheint (d.h. wenn der Monitor schwarz bleibt) oder wenn zwar ein Bild erscheint, der Monitor dieses aber nicht synchronisieren kann (das Bild 'läuft'), können Sie versuchen, in einen anderen Grafikmodus zu wechseln. Normalerweise sind in der Konfigurationsdatei mehrere Grafikmodi (z.B. 640\*480 Punkte, 800\*600 Punkte, 1024\*768 Punkte etc.) vorgesehen, zwischen denen im laufenden Betrieb von X mit `(Strg)+(Alt)+(+)` oder `(-)` (jeweils vom numerischen Tastenblock) gewechselt werden kann. Es ist durchaus möglich, dass je nach Monitor zumindest in manchen dieser Modi ein stabiles Bild angezeigt werden kann.

Gelingt die Synchronisation des Bildschirms nicht, dann sollten Sie X zur Schonung Ihres Monitors möglichst rasch wieder verlassen. Dazu drücken Sie `(Strg)+(Alt)+(Backspace)`. Normalerweise sollten Sie sich jetzt im Textmodus wiederfinden (also in der Textkonsole, in der Sie X gestartet haben). Wenn Sie Pech haben, ist dort allerdings alles schwarz, oder es gelingt auch hier nicht mehr, ein stabiles Bild anzuzeigen. In diesem zum Glück sehr unwahrscheinlichen Fall hilft nur ein Neustart (mit `shutdown` oder `(Strg)+(Alt)+(Entf)`). Mit anderen Worten: Mit einer gewissen Portion Pech müssen Sie nach dem X-Testversuch Ihren Rechner neu booten, und zwar im Blindflug (weil Sie auch im Textmodus nicht mehr arbeiten können). Beenden Sie also alle anderen gerade laufenden Programme, bevor Sie irgendwelche Experimente mit X durchführen.

Damit Sie die Ausgaben beim Start von X auch dann lesen können, wenn es Schwierigkeiten gibt, können Sie diese Ausgaben zur Protokollierung mit `startx >& log` in eine Datei umleiten. Statt `startx` können Sie auch `X -probeonly` ausführen: Dabei versucht

der VGA-Treiber, die Frequenzen Ihrer Grafikkarte zu testen, ohne X wirklich zu starten. Sie kommen also unmittelbar wieder in den Textmodus zurück. Auf diese Weise können Sie relativ problemlos testen, ob die Konfigurationsdatei syntaktisch in Ordnung ist. Sie erhalten dieselben Meldungen wie bei `startx`.

## 1.2 Maus und Tastatur unter X

Dieser Abschnitt beschäftigt sich mit den Besonderheiten von Maus und Tastatur unter X. Dabei werden die vielfältigen Probleme bei der Konfiguration der Tastatur ausführlich erläutert. Außerdem werden wichtige Tastenkürzel und Mausverfahren beschrieben. (Werfen Sie auch einen Blick in den Keyboard-HOWTO-Text!)

### Konfiguration der Tastatur

Die Tastatur ist eine Quelle beständigen Ärgers unter X. Programme, die eigentlich für den Textmodus konzipiert sind, reagieren unter X auf die meisten Funktionstasten (`(Pos1)`, `(Ende)` etc.) nicht mehr korrekt. Bei echten X-Programmen ist es auch eher ein Glücksspiel, ob sie die wichtigsten Tastenkombinationen erkennen oder nicht. Das hängt unter anderem mit der unterschiedlichen Herkunft der Programme zusammen – manche stammen aus der Motif-Welt, andere sind unter OpenLook entstanden etc.

Erstaunlicherweise sind es nicht die deutschen Sonderzeichen, die Probleme verursachen, sondern die Funktionstasten (bis hin zum uralten Unix-Problem, `(Backspace)` und `(Entf)` auseinander zu halten). In diesem Abschnitt wird versucht, die wichtigsten Hintergründe zu erklären und die verschiedenen Dateien aufzuzählen, die einen Einfluss auf die Tastatur unter X haben.

#### Tastaturoptionen in XF86Config

Falls Sie XFree86 als X-Server verwenden, ist die Datei `XF86Config` die zentrale Anlaufstelle auch für die Konfiguration der Tastatur. Bei einer deutschen Tastatur sollte der `Keyboard`-Abschnitt folgendermaßen aussehen:

```
# deutsche Tastatureinstellung in XF86Config
Section "Keyboard"
    Protocol      "Standard"
    XkbRules      "xfree86"
    XkbModel      "pc102"
    XkbLayout     "de"
    XkbOptions    "grp:switch"
EndSection
```

### Die Tilde

Die Tilde ~ wird unter X wie ' ` und ^ zur Komposition zusammengesetzter Buchstaben verwendet.  $\tilde{a}$  führt daher zum Buchstaben ä. Um eine normale Tilde einzugeben (für das Heimatverzeichnis), müssen Sie stets  $\tilde{\text{Leertaste}}$  eingeben, was einigermaßen lästig ist. Das folgende Kommando ermöglicht eine unmittelbare Eingabe der Tilde.

```
user$ xmodmap -e "keycode 35 = plus asterisk asciitilde"
```

Das Kommando gilt nur für das deutsche Tastaturlayout! Wenn Sie möchten, dass es immer ausgeführt wird, müssen Sie die Anweisung in `xinitrc` oder `~/xinitrc` einfügen. (Details zum Startprozess und zu `xinitrc` finden Sie ab Seite ??.)

### Nummerischer Tastenblock

Eingaben mit dem numerischen Tastenblock werden von X standardmäßig im Cursor-Modus ausgewertet (also ohne NumLock). In früheren XFree86-Versionen war es möglich, NumLock automatisch zu aktivieren, indem zum einen im `Keyboard`-Abschnitt in `XF86Config` die Anweisung `X LEDs 1 2 3` und zum anderen in `xinitrc` die Anweisung `xset led 2` oder `3` eingefügt wurde. Seit XFree Version 3.3 funktioniert diese Vorgehensweise nicht mehr (vermutlich wegen des geänderten Protokolls zur Verarbeitung von Tastaturereignissen). Sie müssen `NumLock` also per Tastatur aktivieren.

VORSICHT

Beachten Sie bitte, dass ein aktiviertes NumLock große Probleme im Zusammenspiel mit diversen Programmen und insbesondere mit `fwm2` bzw. `fwm95` bereitet: NumLock wird wie `Shift` oder `Strg` als Zustand betrachtet, der aber in den üblichen `fwm`-Konfigurationsdateien nicht berücksichtigt ist. Die Folge: diverse Fensteroperationen und vor allem der Fensterwechsel per Mausclick bzw. durch Tastenkombinationen funktionieren nicht mehr. Mehr zu diesem unerfreulichen Thema finden Sie bei der Beschreibung von `fwm` ab Seite ??.

### Veränderung der Tastenzuordnung mit xmodmap

Mit `xmodmap` kann die Zuordnung zwischen einer Taste und dem damit erzeugten Zeichen geändert werden. Die Verwendung von `xmodmap` ist bei den X-Servern von XFree86 nur in Ausnahmefällen erforderlich – etwa zur Lösung des leidigen `Backspace`/`Entf`-Problems.

Mit `xmodmap -pke` wird die zur Zeit gültige Tastaturtabelle in der für `xmodmap` erforderlichen Syntax ausgegeben. `xmodmap datei` verarbeitet die in der Datei angegebenen Definitionen; `xmodmap -e definition` ermöglicht die Neudefinition einer einzelnen Taste. Anwendungsbeispiele finden Sie in den folgenden Teilabschnitten.

### Backspace und Entf

Es war einmal vor langer Zeit ein Computer, der hatte eine Tastatur ohne die Taste `Entf`. Zu dieser Zeit waren Computeranwender noch bescheiden, freuten sich darüber, dass sie

ihre Eingaben überhaupt korrigieren konnten, und begnügten sich mit `(Backspace)`. Damals entstand die erste Version von Unix. Seither sind viele, viele Jahre vergangen. Die Rechner wurden immer leistungsfähiger, die Tastaturen etwas komfortabler. Allein die Unterscheidung zwischen `(Backspace)` und `(Entf)` erwies sich als ein derart hartnäckiges Problem, dass selbst die Computermeister ihrer Zeit keine Lösung fanden. Und wenn Unix (und Linux) nicht gestorben sind, dann ärgern sich ihre Anwender noch heute ...

Nach diesem – natürlich frei erfundenen – Märchen zurück in die Realität: Nach dem Start von Linux können Sie mit dem Programm `xev` testen, wie X intern auf das Drücken verschiedener Tasten reagiert. Bei manchen (älteren) Linux-Distributionen werden Sie feststellen, dass das Drücken von `(Entf)` und `(Backspace)` zum gleichen Keysym-Text `Delete` führt. Diese Zeichenkette erhalten X-Programme als Reaktion auf das Drücken einer Taste. Es ist also nicht weiter verwunderlich, dass X-Programme nicht zwischen `(Entf)` und `(Backspace)` unterscheiden können.

Um überhaupt eine Möglichkeit zur Unterscheidung zu schaffen, muss `xmodmap` eingesetzt werden. Mit der folgenden Anweisung erreichen Sie, dass das Drücken von `(Backspace)` in Zukunft die Keysym-Zeichenkette `'BackSpace'` liefert:

```
# Ergänzung in xinitrc (nur bei alten XFree86-Versionen erforderlich)
xmodmap -e "keycode 22=BackSpace"
```

`xinitrc` finden Sie je nach Distribution in `/etc/X11/xinit` oder in `/usr/X11R6/lib/X11/xinit`. Falls das Heimatverzeichnis die Datei `.xinitrc` enthält, hat diese Datei Vorrang gegenüber dem globalen `xinit`.

**HINWEIS**

Die internen Tastennummern im Textmodus (für die Programme `loadkeys` und `showkey`) und unter X (für die Programme `xmodmap` und `xev`) stimmen nicht überein! Im Textmodus hat die Taste `(Backspace)` die Nummer 14, unter X die Nummer 22.

Für einige modernere X-Programme ist diese Anweisung bereits ausreichend; `(Backspace)` und `(Entf)` funktionieren wunschgemäß. Einige weitere Programme werden mit `BackSpace` gar nichts anfangen können und dafür weiterhin die `Delete`-Zeichenkette als `BackSpace` interpretieren. Bei solchen Programmen hilft häufig ein Eingriff in `~/Xdefaults` oder `~/Xresources` weiter (siehe auch Seite 32).

### Tasten in Textmodus-Programmen

Auch unter X werden Sie häufig mit Programmen arbeiten, die eigentlich für den Textmodus konzipiert sind (`jed`, `pine` etc.). Für solche Programme hat die obige Anweisung wenig gebracht. Die tollste Reaktion liefert der Editor `jed`: `(Backspace)` ruft nun die Online-Hilfe auf, während `(Entf)` die Wirkung von `(Backspace)` hat!

Programme für den Textmodus werden unter X in einem Terminal- oder Shell-Fenster ausgeführt, das durch das Programm `xterm` gebildet wird. Dieser Umstand ermöglicht es, mit einer einzigen Veränderung der Ressourcen für `xterm` eine korrekte Un-

terstützung für fast alle herkömmlichen Programme zu erreichen. Fügen Sie die folgenden Zeilen in die Datei `~/.Xdefaults` oder `~/.Xresources` ein (je nach Distribution):

```
! in ~/.Xdefaults oder ~/.Xresources
*VT100.Translations: #override \
    <KeyPress>BackSpace: string(0x7f)\n\
    <KeyPress>Delete: string(0x04)\n\
    <KeyPress>Home: string(0x01)\n\
    <KeyPress>End: string(0x05)
```

Diese Einstellung bezieht sich eigentlich auf das VT100-Widget – dieses ist aber die Basis von `xterm`. Die Keysym-Zeichenkette `BackSpace` wird durch das Zeichen mit dem Code 127 und die Keysym-Zeichenkette `Delete` durch das Zeichen mit dem Code 4 (das entspricht `(Strg)+(D)`) ersetzt. Nach demselben Muster lässt sich auch gleich eine Zuordnung von `(Pos1)` zu `(Strg)+(A)` und von `(Ende)` zu `(Strg)+(E)` durchführen, so dass auch diese Tasten wie im Textmodus verwendet werden können, um an den Anfang bzw. an das Ende einer Zeile zu springen.

Damit werden die vier Tasten von sehr vielen Programmen richtig interpretiert. Getestet wurden diese Einstellungen unter anderem mit dem Emacs im Textmodus (`emacs -nw`), mit `jed`, `jove`, `pine`, `bash` etc.

Damit auch die X-Version von Emacs `(Backspace)` und `(Entf)` erkennt, müssen Sie die folgende Zeile in `~/.emacs` aufnehmen:

```
; in ~/.emacs
(global-set-key [delete] 'delete-char) ;<Entf> Zeichen löschen
```

**HINWEIS**

Hundertprozentig ist die Simulation der Tastatur für Programme im Textmodus auch jetzt noch nicht: Beispielsweise wird in den Emacs-Emulatoren `jed` und `jove` die Taste `(Alt)` nicht erkannt. Wenn diese Programme unter X verwendet werden sollen, muss statt einer Tastenkombination mit `(Alt)` die Taste `(Esc)` verwendet werden. Die obige `.Xdefaults`-Einstellung gilt nicht für die ebenfalls als Terminals verwendbaren Programme `cmdtool` und `rxvt`.

### Tastatureinstellungen direkt durch `xmodmap`

Wenn Sie unter X *nur* mit konventionellen Programmen arbeiten, gibt es noch eine Lösung: Sie könnten den Tasten `(Backspace)`, `(Entf)`, `(Pos1)` und `(Ende)` durch `xmodmap` die Codes 0x7f, 4, 1 und 5 zuweisen (statt der Texte `BackSpace`, `Delete`, `Home` und `End`):

```
# in xinitrc (diese Vorgehensweise ist nicht zu empfehlen!)
xmodmap -e "keycode 22=0x7f" # <Backspace> liefert den Code 127
xmodmap -e "keycode 107=0x04" # <Entf> liefert Strg+D
xmodmap -e "keycode 97=0x01" # <Pos1> liefert Strg+A
xmodmap -e "keycode 103=0x05" # <Ende> liefert Strg+E
```

Textmodusprogramme, die unter `xterm` oder einem anderen Terminalprogramm ausgeführt werden, bekommen jetzt auf jeden Fall die vom Textmodus bekannten Codes.

Allerdings werden jetzt auch echte X-Programme die eingestellten Zeichencodes erhalten und können damit (mit wenigen Ausnahmen) nichts anfangen. Aus diesem Grund stellt diese Vorgehensweise eine Sackgasse dar.

### X und das Euro-Symbol

Bei neueren Distributionen ist die Tastatur bereits so konfiguriert, dass die Tastenkombination  $\text{AltGr}+\text{E}$  dem Unicode-Zeichen 0x20ac zugeordnet ist. Davon können Sie sich mit dem Programm `xev` leicht vergewissern. (0x20ac ist der international vereinbarte Ort für das Euro-Symbol im Unicode-Zeichensatz. Die Tastenkombination  $\text{AltGr}+\text{E}$  gilt für Tastaturen in Deutschland, Österreich und der Schweiz. In anderen Ländern sind andere Tastenkombinationen zur Eingabe des Euro-Symbols üblich.)

Leider gibt es zur Zeit kaum X-Programme und X-Zeichensätze, die vollständig Unicode-kompatibel sind. (Das bessert sich zusehends. Insbesondere soll KDE 2.0 Unicode-kompatibel sein.) Wenn auf die Eingabe von  $\text{AltGr}+\text{E}$  also keine Reaktion erfolgt, können Sie sich mit der folgenden Notlösung behelfen:

```
xmodmap -e 'keycode 26 = e E currency EuroSign'
```

Damit wird  $\text{AltGr}+\text{E}$  dem internationalen Währungszeichen zugeordnet (das normalerweise so ähnlich aussieht wie eine Überlagerung aus einem o und einem x). Jetzt benötigen Sie noch einen Zeichensatz, in dem dieses Währungszeichen durch das Euro-Symbol ersetzt ist. Das ist für neuere Zeichensätze in ISO-8859-15-Codierung der Fall. Um einen derartigen Zeichensatz auszuprobieren, starten Sie `xterm` mit dem folgenden Kommando:

```
user$ xterm -fn -misc-fixed-medium-r-normal--*-ISO8859-15
```

Jetzt können Sie in `xterm` das Euro-Symbol eingeben und sehen. Analog können Sie diesen Zeichensatz natürlich auch im Emacs oder jedem anderen Programm verwenden, bei dem der Zeichensatz konfigurierbar ist. Eine wirklich universelle Lösung zu diesem Problem wird es aber erst geben, wenn X, KDE, Gnome etc. generell das Unicode-Format unterstützen (und nicht wie jetzt nur punktuell und mit vielen Verrenkungen).

VERWEIS

Weitere Informationen zu diesem Thema (auch zur Tastaturkonfiguration für andere Länder) finden Sie im Unicode-HOWTO sowie an den folgenden Adressen im Internet:

```
http://www.cl.cam.ac.uk/ mgk25/unicode.html
http://www.lil8nux.org/
http://www.hp.com/visualize/support/technotes/linux/
    localization.html
http://monitor.blvk.ch/euro
```

## Globale Tastenkombinationen

Wenn Sie unter X arbeiten, gelten zumeist die folgenden Tastenkombinationen:

<http://www.kofler.cc>

**X-Tastenkürzel**

<b>(Strg)+(Alt)+(Backspace)</b>	beendet X
<b>(Strg)+(Alt)+(Fn)</b>	Wechsel in die Textkonsole 1 bis 6
<b>(Alt)+(F7)</b>	Wechsel aus der Textkonsole zurück ins X Window System
<b>(Strg)+(Alt)+(+) / (-)</b>	Wechsel des Grafikmodus

Die Tastenkombinationen **(Strg)+(Alt)+(Backspace)**, **+(+)** sowie **+(-)** können durch die Schlüsselwörter `DontZap` und `DontZoom` im Abschnitt `Server` in `XF86Config` deaktiviert werden.

Die Tastenkombination **(Strg)+(Alt)+(Entf)** kann unter X *nicht* zum Neustarten von Linux verwendet werden. Zum Neustart müssen Sie entweder in eine Textkonsole wechseln und dort **(Strg)+(Alt)+(Entf)** drücken oder das Kommando `shutdown -r now` in einem Terminalfenster eingeben.

Weitere Tastenkürzel werden durch den aktiven Windows-Manager definiert (siehe Seite ??).

**Eingabefokus**

Im X Window System können mehrere Programme gleichzeitig laufen – es kann aber immer nur ein Programm Tastatureingaben empfangen. Je nach Konfiguration des Windows-Managers gilt wahlweise das Programm als aktiv, das zuletzt angeklickt wurde, oder das, über dem sich gerade der Mauszeiger befindet. Vor allem die zweite Variante bereitet all jenen Umstellungsschwierigkeiten, die bisher mit MS-Windows gearbeitet haben. Abhilfe schafft eine andere Konfiguration des Windows-Managers (siehe Seite ??).

Noch ungewohnter ist bei manchen Programmen die Verwaltung des Eingabefokus innerhalb eines Fensters (probieren Sie es beispielsweise bei `xman` mit dem Kommando `SEARCH`): Eingaben können nur in dem Feld durchgeführt werden, über dem die Maus steht. Da die Eingabefelder oft recht klein sind, reicht ein minimales Verschieben der Maus schon aus, um Ihre Eingaben ins Leere zu senden. Viele neue Programme orientieren sich in dieser Beziehung glücklicherweise am Verhalten von Motif-Programmen, wo der Eingabefokus durch einen Maustastenklick in einem Feld fixiert wird.

**Konfiguration der Maus**

Falls Sie einen X-Server von `XFree86` verwenden, ist der Abschnitt `Pointer` in `XF86Config` für die Mauskonfiguration verantwortlich. Die beiden wichtigsten Optionen, `Device` und `Protocol`, geben an, an welcher Schnittstelle die Maus angeschlossen ist (zumeist `/dev/mouse` oder `/dev/ttyS0`) und welches Protokoll die Maus verwendet. Die Liste der möglichen Protokolle entnehmen Sie bitte der Manualseite zu `XF86Config`. Das Protokoll `Auto` sollte die Maus selbstständig erkennen – das hat bei meinen Modellen allerdings nicht funktioniert.

Bei Zwei-Tasten-Mäusen erreichen Sie mit der Option `Emulate3Buttons`, dass das gleichzeitige Drücken beider Tasten zur Simulation der fehlenden dritten Taste verwendet werden kann. `Emulate3Timeout` gibt dabei an, mit welcher Verzögerung das Drücken einer einzelnen Taste verarbeitet wird. Manche Drei-Tasten-Mäuse von Logitech funktionieren nur dann, wenn die Option `ChordMiddle` benutzt wird.

```
# Mauseinstellung in XF86Config
Section "Pointer"
    Device      "/dev/psaux" # Anschluss an PS2-Stecker
    Protocol    "PS/2"      # PS/2-Maus mit 2 oder 3 Tasten
    # Emulate3Buttons      # nur bei Zwei-Tasten-Mäusen
    # Emulate3Timeout 50  # Verzögerungszeit in ms
    # ChordMiddle          # nur für einige Drei-Tasten-Mäuse
                        #   von Logitech
EndSection
```

### Radmaus (IntelliMouse)

Für Radmäuse sind die Protokolle `IntelliMouse` und `IMPS/2` vorgesehen (zweites für Radmäuse am PS/2-Anschluss). Damit funktioniert das Rad zwar noch nicht so wie unter Windows, aber es kann zumindest als dritte Maustaste verwendet werden. (Lediglich einige KDE-Programme können die Maussignale auf Anhub richtig interpretieren, beispielsweise `kvt`.)

Es geht aber noch besser: Fügen Sie in `XF86Config` die unten abgedruckte `ZAxisMapping`-Zeile ein. Damit werden die Bewegungen des Rades den (gar nicht vorhandenen) Maustasten 4 und 5 zugeordnet. Außerdem müssen Sie das Paket `imwheel` installieren und nach dem Start von X mit `imwheel -k` als Hintergrundprozess starten. Das Programm verarbeitet die Maustastensignale für die Tasten 4 und 5 und wandelt diese in Tastatursignale um, die dann bei den meisten Programmen ein Scrollen des Bildschirminhalts auslösen.

```
# IntelliMouse-Einstellung in XF86Config
Section "Pointer"
    Device      "/dev/psaux"
    Protocol    "imps/2"
    ZAxisMapping 4 5
EndSection
```

`imwheel` kann durch `/etc/imwheelrc` oder durch `~/imwheelrc` gesteuert werden. In dieser Datei können Sie angeben, welche Tastatureingaben bei welchen Programmen simuliert werden sollen.

Wenn Sie `imwheel -k` nicht nach jedem X-Start manuell ausführen möchten, können Sie das Kommando natürlich auch in `xinitrc` oder `xsession` angeben. (Diese Dateien werden ab Seite ?? ausführlich beschrieben.)

## Verwendung der Maus

### Texte mit der Maus kopieren und einfügen

In praktisch allen X-Programmen können Sie die Maus dazu verwenden, Textausschnitte zu kopieren und sie anschließend an einer anderen Stelle (oder in einem anderen Programm) wieder einzufügen. Die Bedienung der Maus erfolgt dabei wie im Textmodus, wenn dort das Programm `gpm` verwendet wird. Das Markieren von Textausschnitten erfolgt bei gedrückter linker Maustaste. Der markierte Text wird dabei automatisch in einen Puffer kopiert. Sobald Sie die mittlere Maustaste drücken, wird der Text dort eingefügt, wo der aktive Eingabecursor steht. Mit der rechten Maustaste können Sie die Größe einer Textmarkierung verändern.

### Bedienung von Bildlaufleisten

Viele X-Programme sind mit standardisierten Schiebepalken ausgestattet (beispielsweise `xterm`). Die Größe des grauen Schiebefelds gibt die relative Größe des sichtbaren Ausschnitts an. Während es beim Aussehen noch gewisse Ähnlichkeiten gibt, weicht die Bedienung stark von dem ab, was Sie möglicherweise von MS-Windows gewöhnt sind:

Die Schiebepalken werden mit gedrückter mittlerer Maustaste verschoben. Das Drücken der linken Maustaste bewegt den Schiebepalken nach unten, das Drücken der rechten Maustaste bewegt ihn nach oben. Dabei gibt die aktuelle Position der Maus an, wie weit der Schiebepalken bewegt wird: Wenn die Maus im Schiebepalken ganz oben steht, wird das Schiebefeld nur ganz wenig bewegt (eine Zeile). Wenn die Maus dagegen ganz unten steht, wird das Schiebefeld eine ganze Seite weiterbewegt.

### Maus unsichtbar machen

Normalerweise ist die Maus unter X ständig sichtbar. Gerade bei Programmen, die primär über die Tastatur bedient werden, kann das lästig sein – die Maus verdeckt oft gerade die Zeichen, die man sehen möchte. Das Programm `unclutter` bietet Abhilfe. Es wird als Hintergrundprozess gestartet und macht die Maus automatisch unsichtbar, wenn sie sich über einem Fenster befindet und mehrere Sekunden nicht bewegt wurde. Die Maus wird sofort wieder sichtbar, wenn sie bewegt wird bzw. wenn eine Maustaste gedrückt wird. `unclutter` gehört beispielsweise bei Debian zum Lieferumfang. Bei den meisten anderen Distributionen muss das Paket im Internet gesucht werden.

## 1.3 X-Ressourcen

Ressourcen stellen eine Besonderheit des X-Systems dar. Sie ermöglichen die bequeme Einstellung der Eigenschaften der meisten X-Programme – von der Hintergrundfarbe eines `xterm`-Fensters bis hin zur Schriftart im Emacs. Damit Sie das Konzept der Ressourcendateien verstehen, ist ein kleiner Ausflug in die Interna des X-Systems erforderlich.

Es gibt natürlich auch X-Programme, die zusätzlich oder ausschließlich eigene Konfigurationsdateien verwenden. Ein Motiv für den Verzicht auf X-Ressourcen ist oft der Versuch, ein gleichbleibendes Konfigurationsverfahren für unterschiedliche Rechnerplattformen zu finden. Auf der Suche nach dem kleinsten gemeinsamen Nenner bleiben die X-Ressourcen dann auf der Strecke.

## Grundlagen

X-Programme sind aus sogenannten Widgets zusammengesetzt. Widgets sind grafische Steuerelemente für bestimmte Aufgaben. Es existieren beispielsweise Widgets für Textfelder, für Buttons etc. Widgets haben bestimmte Merkmale – beispielsweise `font` für den Zeichensatz, `background` für die Hintergrundfarbe etc. Widgets können hierarchisch verschachtelt werden – d.h. in einem Rahmen-Widget für ein Menü können sich mehrere Button-Widgets für die einzelnen Menükommandos befinden.

Die vollständige Bezeichnung eines Details eines X-Programms – etwa der Hintergrundfarbe des VT100-Widgets des `xterm`-Programms – ergibt sich einfach aus der Zusammensetzung aller Komponenten. Die Komponenten werden durch Punkte verbunden. Um die Hintergrundfarbe auf Weiß zu setzen, ist in der Ressourcendatei die folgende Zeile erforderlich:

```
xterm.vt100.background: white
```

Dieses Konzept wird in den meisten X-Programmen durch zusätzliche private Ressourcen erweitert. Aus diesem Grund lassen sich nicht nur die Eigenschaften von Widgets verändern, sondern auch viele globale Einstellungen. (Dieser Unterschied ist insofern interessant, als die Einstellung der Widget-Eigenschaften automatisch funktioniert, ohne dass im X-Programm dazu eine einzige Zeile Code erforderlich ist. Aus diesem Grund sind in den `man`-Texten von X-Programmen oft nur die programmspezifischen Erweiterungen dieses Ressourcenkonzepts beschrieben.) Die folgende Zeile ist ein typisches Beispiel für eine programmspezifische Ressourceneinstellung (hier für den Terminalemulator `seyon`).

```
seyon.hangupBeforeDial: off
```

Noch ein Detail der Ressourceneinstellung ist erwähnenswert: Bei der Bildung der Namen darf das Jokerzeichen `*` verwendet werden. Das Zeichen hat dieselbe Bedeutung wie bei Dateinamen, es erstreckt sich aber auch über mehrere Komponenten von Ressourcennamen. Beispielsweise bewirkt die erste der beiden folgenden Zeilen, dass *alle* Hintergrundflächen beliebiger Programme und Widgets gelb eingefärbt werden! Ausgenommen sind nur die Widgets, für die explizit eine andere Farbe eingestellt wurde. Die zweite Zeile färbt alle Widgets des `seyon`-Programms blau.

```
*background: yellow  
seyon*background: blue
```

## Ressourcendateien

Ressourcendateien können an verschiedenen Orten im Verzeichnisbaum gespeichert werden. Automatisch erkannt werden diese Dateien je nach Distribution an den folgenden Stellen. Dabei haben die Einstellungen in den weiter unten angegebenen Dateien Vorrang:

### Orte von X-Ressourcendateien

Defaulteinstellungen ausschließlich für den xdm-Login:

```
/etc/X11/xdm/Xresources oder  
/usr/X11R6/lib/X11/xdm/Xresources
```

Optionale lokale Defaulteinstellungen werden bei vielen Systemen nur von `startx` (nicht aber von `xdm`) berücksichtigt:

```
~/.Xresources
```

Globale Defaulteinstellungen für ein bestimmtes Programm:

```
/usr/X11R6/lib/X11/app-defaults/progname
```

Lokale Einstellungen des jeweiligen Anwenders:

```
~/.Xdefaults (bei Red Hat)  
~/.Xresources (bei SuSE)
```

Die `Xresources`-Dateien werden nur einmal beim Start von X eingelesen (siehe Seite ??). Die Einstellungen im `app-defaults`-Verzeichnis sowie in `.Xdefaults` werden dagegen bei jedem Start eines X-Programms berücksichtigt.

Vorrang gegenüber allen hier angegebenen Ressourcendateien haben Optionen, die unmittelbar beim Programmstart angegeben werden. Das betrifft natürlich auch Programme, die von `xinitrc` oder über den `fvwm` gestartet werden.

Die Syntax innerhalb der Ressourcendateien ist aus den obigen Beispielen schon weitgehend hervorgegangen. Groß- und Kleinschreibung spielt in den meisten Fällen keine Rolle (es gibt aber Ausnahmen!). Ungewöhnlich ist die Kennzeichnung von Kommentaren: Diese werden nicht wie in anderen Programmen mit einem `#` eingeleitet, sondern stattdessen mit einem Ausrufezeichen. Außerdem müssen Kommentare allein in einer Zeile stehen – es ist nicht erlaubt, an eine Ressourceneinstellung einen Kommentar anzuhängen.

Wenn Sie die Ressourceneinstellungen verändern möchten, besteht die einfachste Vorgehensweise darin, dass Sie sich an vorhandenen `.Xdefaults`-Dateien orientieren und daraus die für Sie brauchbaren Zeilen in Ihre eigene `.Xdefaults`-Datei kopieren (und vielleicht einige Einstellungen ändern). Beispiele für Ressourceneinstellungen finden Sie in den Dateien des `app-defaults`-Verzeichnisses.

**TIP**

Detailinformationen zu den zulässigen Ressourceneinstellungen finden Sie in den man-Texten zum jeweiligen Programm. Das grundsätzliche Konzept von Ressourcen ist in den man-Texten zu X und zu xrdp beschrieben.