Study-Guide: X-Window-System

Alles was mit der graphischen Oberfläche (dem X-Window-System) zu tun hat fällt unter dieses Kapitel. Installation und Konfiguration von XFree86 Einrichten eines Display Managers Installation und Anpassung einer Window Manager Umgebung

Seite:	-= LinuxLernSystem =- (http://www.lpi-test.de)
Kurs:	LPIC-1 [101]
Buch:	Study-Guide: X-Window-System
Gedruckt von:	André Scholz
Datum:	Dienstag, 1 November 2005, 10:27 Uhr

Inhaltsverzeichnis

X-Window-System

Alles was mit der graphischen Oberfläche (dem X-Window-System) zu tun hat fällt unter dieses Kapitel.

- Installation und Konfiguration von XFree86
- Einrichten eines Display Managers
- Installation und Anpassung einer Window Manager Umgebung

1.110.1 - Installation und Konfiguration von XFree86

Beschreibung: Prüfungskandidaten sollten in der Lage sein, X und einen X-Fontserver zu installieren und zu konfigurieren. Dieses Lernziel beinhaltet die Überprüfung, ob Grafikkarte und Monitor von einem X-Server unterstützt werden, und das Anpassen und Trimmen von X für Grafikkarte und Monitor. Ebenfalls enthalten ist die Installation eines X-Fontservers, die Installation von Schriftarten und das Konfigurieren von X zur Benutzung des Fontservers (möglicherweise durch manuelles Bearbeiten des Abschnitts "Files" in `/etc/X11/XF86Config`).

Die wichtigsten Dateien, Bezeichnungen und Anwendungen:

- **XF86Setup**
- **xf86config**
- **xvidtune**
- `/etc/X11/XF86Config`
- `.Xresources`

Grundlagen

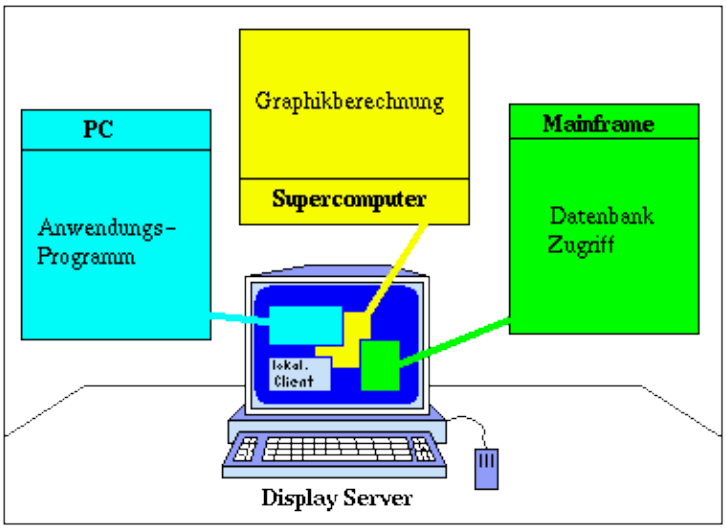
Das X-Window-System oder kurz X11 ist ein netzbasiertes graphisches Fenstersystem, das auf einer *Client/Server* Architektur beruht. Programme, die ihre Ausgaben in Fenstern machen wollen sind Clients, die den Service in Anspruch nehmen, den der *Display-Server* anbietet, nämlich Ausgabe auf dem Schirm zu bringen und Eingaben von der Tastatur und Maus entgegenzunehmen.

Dabei ist wichtig, daß es möglich ist, daß Client und Server über ein Netzwerk miteinander kommunizieren können. Es ist also unerheblich, ob die Programme, die auf einem Bildschirm dargestellt werden auf dem lokalen Rechner laufen oder auf einem anderen Rechner im Netz.

Client und Server verständigen sich mittels einem Protokoll, dem sogenannten *X-Protokoll*. Dieses Protokoll nutzt als Transportbasis meist TCP/IP, aber auch DECNet oder der Betrieb auf reinen Unix-Sockets ist möglich. Technisch gesehen ist das X-Protokoll die eigentliche Definition des X-Window-System.

Durch die Definition eines eindeutigen Protokolls, das alle Fähigkeiten zur Übermittlung von graphischen Ausgaben beinhaltet wird X11 zur hardwareunabhängigen Graphikplattform. Das X11-System ist aufgegliedert in einen Displayserver und in Programme, die den Display-Service in Anspruch nehmen, also Clients. Ein Server kann für jede beliebige Art von Graphiksystemen geschrieben werden, über das vordefinierte X-Protokoll können beliebige Clients dann ihre Ausgaben auf dem Server machen.

Natürlich können Clients (also X11-Programme) auch lokal auf dem selben Rechner laufen, auf dem auch der Display-Server läuft, das ist sogar die Regel. Aber es können eben auch beliebige andere Rechner Programme laufen haben, deren Ausgaben dann auf unserem Server zu sehen sind.



Zusammenfassend können wir also sagen, das X11-System beruht auf drei Einzelelementen:

1. Der **Display-Server**, der alle Ausgaben auf den Bildschirm steuert und die Eingaben von der Tastatur und der Maus entgegennimmt. Er ist ein Programm, das lokal auf dem Rechner läuft und Kontrolle über die Grafik-Hardware hat.
2. Ein oder mehrere **Clients**, also Programme, die ihre Ein/Ausgabe nicht selbst erledigen sondern diese Aufgabe vom Display-Server erledigen lassen. Kurz gesagt sind alle X-Anwendungen (xterm, xsnow, xteddy, ...) Clients. Es ist unerheblich, auf welchem Rechner im Netz diese Programme laufen, solange sie mit dem Display-Server verbunden sind geht ihre Ausgabe auf den Bildschirm dieses Servers.
3. Das **X-Protokoll**, das die Kommunikation zwischen den Clients und dem Display-Server regelt. Dieses Protokoll ist das einzige, das X11 eigentlich ausmacht. Prinzipiell kann für jedes graphikfähige Gerät ein X-Server geschrieben werden, wenn er das X-Protokoll versteht, kann er X-Clients erlauben, ihre Ausgaben auf dem Gerät auszugeben.

Bei der hier gestellten Aufgabe geht es also um die Installation eines X-Servers, also des Programms, das anderen Programmen (den X-Clients) den Dienst anbietet, sich auf dem Bildschirm darstellen zu können und die Eingabegeräte Tastatur und Maus zu nutzen. Beim Umgang mit X11 ist es grundsätzlich wichtig, sich dieser Client-Server-Architektur immer bewußt zu sein, um tatsächlich zu verstehen, was eigentlich vorgeht.

Es gibt X-Server von verschiedensten Herstellern für verschiedenste Hard- und Softwaresysteme. Der hier verwendete X-Server ist der X-Free-86 Server. Dieser Server ist freie Software und für die Verwendung auf 80x86 Systemen (und kompatiblen Prozessoren) gedacht. Aus diesen Eigenschaften bezog er seinen Namen. XF86 ist der unter Linux verbreitetste Server, andere angebotene Server sind meist kommerzielle Produkte und spielen in der Praxis kaum eine Rolle.

Die Konfiguration des XFree86-Servers

Der Display-Server verwaltet die Hardware, deren Verwendung er anderen Clients ermöglichen soll. Er muß daher sehr genau über die installierte Graphik- und Eingabehardware Bescheid wissen. Insbesondere die folgenden Hardware-Elemente muß er verwalten:

- Graphikkarte
- Monitor
- Tastatur
- Maus
- Andere Eingabegeräte (Joystic, Grafiktablett,...)

Die gesamten Angaben über diese Geräte und auch alle anderen Konfigurationseinstellungen befinden sich in einer einzigen Datei. Diese Datei kann manuell bearbeitet werden, es ist eine einfache Textdatei. Sie heißt `XF86Config` und befindet sich in der Regel im Verzeichnis `/etc/X11`. Die Orginalversion von XF86 hatte eigentlich andere Orte für die Konfigurationsdatei vorgesehen, aber im neuen Filesystem Hierarchy Standard ist `/etc/X11` festgelegt worden. Die meisten Distributionen halten sich heute an diesen Standard. Der Vollständigkeit halber seien hier nochmal die ursprünglichen Konfigurationsdateien genannt:

- **~/XF86Config**
  Wenn der User, der den X-Server manuell startet der Systemverwalter ist, so wird zuerst in seinem Homeverzeichnis nach der Konfigurationsdatei gesucht.
- **/etc/XF86Config**
  Das war früher der Standard für die systemweite Konfigurationsdatei des X-Servers.
- ***X-Wurzel*/lib/X11/XF86Config.*hostname***
  Falls in /etc keine Konfigurationsdatei gefunden wurde, so wurde in diesem Verzeichnis danach gesucht. *X-Wurzel* bezeichnet die Wurzel des X-Systems, meist `/usr/X11R6`. Damit geteilte `/usr`-Verzeichnisse benutzt werden konten, konnte für verschiedene Rechner je eine Konfigurationsdatei angelegt werden, die dann den Hostnamen des Rechners als Namenserweiterung trug, für den sie gedacht war.
- ***X-Wurzel*/lib/X11/XF86Config**
  Falls an dieser Stelle keine Datei mit dem passenden Hostnamen gefunden wurde, wurde nach einer Konfigurationsdatei ohne Hostnamenendung gesucht.

Noch heute funktionieren diese Dateien, werden aber kaum noch benutzt.

Weil heute zwei XFree86 Versionen im Umlauf sind (Version 3.x und Version 4.x) und die beiden unterschiedliche Formate ihrer Konfigurationsdatei erwarten, haben verschiedene Distributionen unterschiedliche Mechanismen entwickelt, um einen gleichzeitigen Betrieb beider Versionen zu ermöglichen. Entweder liegen beide Konfigurationsdateien in `/etc/X11`, dann trägt die Datei für die Version 4.x den Namen `XF86Config-4`, oder es wurde die Konfigurationsdatei für Version 3.x in `/etc` belassen und die von Version 4.x liegt in `/etc/X11`. Im weiteren Verlauf werden wir hier keine Unterschiede machen,

sondern die Datei einfach immer `/etc/X11/XF86Config` nennen.

Früher musste die Konfigurationsdatei von Hand erstellt werden und es bestand die Gefahr, daß - bei falschen Monitor-Einstellungen - die Hardware tatsächlich beschädigt werden konnte. Die Errechnung der Bildschirmparameter war eine umständliche und schwierige Arbeit, die nicht immer das erreicht hatte, was gewünscht war. Durch die Architektur moderner Graphikkarten und Monitore ist diese Konfigurationsarbeit heute erheblich erleichtert worden. Zudem gibt es eine ganze Reihe von Hilfsmitteln, um die Konfigurationsdatei erstellen zu lassen.

xf86config
: Ein kleines textorientiertes Programm, das die ganze Abfrage aller notwendigen Hardwareinformation vornimmt, und anschließend eine XF86Config Datei daraus erstellt. Dieses Programm war die erste automatisierte Methode von XFree86 und gehört zur Standard-Ausstattung.

XF86Setup
: Auch dieses Programm wird von XFree86 selbst betreut und ausgeliefert. Im Gegensatz zu **xf86config** arbeitet es aber bereits in einem graphischen Modus. Das Programm startet in einem VGA16 Graphikmodus, der von praktisch jeder Graphikkarte unterstützt wird, und arbeitet mit einer Bildschirmauflösung von 640x480 Pixeln, die jeder Bildschirm darstellen können sollte. Auch dieses Programm erstellt eine XF86Config Datei aus den gemachten Angaben, ist aber dann in der Lage, die Konfiguration gleich auszuprobieren.

SAX und SAX2
: Sind Programme der SuSE-Distribution (SAX heißt SuSE Advanced X-Configuration) und stehen nur dort zur Verfügung. SAX erstellt Konfigurationsdateien für Version 3.x, Sax2 für Version 4.x. Beide Programme arbeiten wie **XF86Setup** bereits im graphischen Modus, versuchen jedoch bereits vor dem Start des Graphikmodus durch einen Scan des PCI-Busses herauszufinden, welche Einstellungen notwendig sind.

Xconfigurator
: ist das X11-Konfigurationsprogramm von RedHat. Es handelt sich um eine textorientierte verbesserte Version von **xf86config**, die auch vorher den PCI-Bus scant, um die angeschlossenen Geräte zu erkennen. Über die SLANG-Library wurde die Benutzerführung verbessert. Das Programm kann im interaktiven Modus laufen oder automatisch aus den gefundenen Informationen eine Konfigurationsdatei erstellen (kickstart).

dexconf
: ist das Programm der Debian Distribution, das aus den gemachten Angaben der Debian-Konfiguration eine entsprechende XF86Config Datei erstellt. Das Programm wird automatisch während der Installation aufgerufen, ist also kein wirkliches usergeführtes Konfigurationsprogramm für XFree86. Sollte unter Debian eine manuelle Konfiguration erwünscht sein, so wird der Befehl

```
dpkg-reconfigure xserver-xfree86
```

verwendet. **dexconf** wird dann abschließend automatisch aufgerufen.

Als distributionsunabhängige Zertifizierung genügt für LPI das Wissen um die ersten beiden Programme, alle anderen sind spezielle Lösungen der Distributionen. All diese Programme erledigen die selben Aufgaben (auf unterschiedliche Weise), die hier nochmal genauer dargestellt werden sollen. Im Wesentlichen handelt es sich um zwei Aufgaben:

1. Abfrage der Hardware und erstellen einer passenden XF86Config Datei.
2. Erstellen eines Links auf den zu verwendenden X-Server.

Beide Aufgaben werden jetzt nochmal im Einzelnen besprochen:

Abfrage der Hardware und erstellen einer passenden XF86Config Datei

Der X-Server steuert die Graphik- und Eingabehardware um sie dann den Clients zur Verfügung zu stellen. Er muß also sehr genau über die verwendeten Geräte Bescheid wissen und benötigt zum Teil sehr technische Informationen. Folgende Informationen werden abgefragt, müssen also bekannt sein:

Tastatur
: Diese Einstellung ist sicher die einfachste. Tastaturen sind stark standardisiert und dem Betriebssystem auch im Textmodus bekannt. Wichtige Einstellungen sind hier
  - Anzahl der Tasten (pc101, pc102, pc104)
  - Tastaturlayout (Sprache - de, en, ...)
  - Zusätzliche Optionen, wie mit Akzent-Tasten umgegangen werden soll (nodeadkeys)

  Normalerweise sollten diese Einstellungen kein Problem darstellen.

Maus
: Bei der Maus haben wir schon wesentlich mehr Fragen zu beantworten. Dazu zählen
  - An welcher Schnittstelle ist die Maus angeschlossen (seriell, PS/2, USB)?

- Welches Maus-Protokoll verwendet die Maus (imps/2, microsoft, logitech, mmseries, ...)?
- Wieviele Tasten hat die Maus?
- Soll (bei 2-Tasten Mäusen) die Emulation der mittleren Taste durch gleichzeitiges Drücken der beiden Tasten aktiviert werden (Emulate3Buttons)?
- Soll ein eventuell vorhandenes Scroll-Rädchen aktiviert werden (ZAxisMapping)?

Die Einstellungen können - zumindest bei den graphischen Setup-Programmen - gleich ausprobiert werden. Wichtig ist, daß zumindestens die Anschluß- und Mausart stimmen, weil XFree86 ohne Maus gar nicht startet.

Graphikkarte

Hier wird in der Regel bei modernen Rechnern alles automatisch erkannt. Durch die Verwendung von PCI/AGP Karten ist eine automatische Konfiguration möglich. Ansonsten müssen hier - bei alten Rechnern - folgende Angaben gemacht werden:

- Chipsatz der Graphikkarte (oder Auswahl aus einer Liste bekannter Karten).
- Größe des Graphikspeichers.
- Zu verwendender X-Server (Nur Version 3.x)
- Clocks (nur bei alten Karten)

Die Angabe der Clocks ist bei modernen Karten nicht mehr notwendig, weil sie diese Angaben auf Anfrage selbst machen. Bei sehr alten Karten, die nicht in der Kartendatenbank vorhanden waren, war diese Angabe sowohl lebenswichtig, als auch nahezu unlösbar...

Bildschirm

Um einen Bildschirm richtig ansteuern zu können, benötigt XFree86 die folgenden Angaben, die im Handbuch des Monitors zu finden sein sollten:

- Horizontale Synchronisation in kHz (entweder eine Liste von festen Frequenzen durch Kommas voneinander getrennt, oder - bei fast allen modernen Monitoren - ein Bereich, der durch Bindestrich verbunden ist, etwa 30-96).
- Vertikale Synchronisation (Refreshrate) in Hz. (Wie bei der horizontalen Synchronisation entweder eine Liste oder ein Bereich).

Diese Angaben müssen stimmen, sonst wird der Monitor falsch angesteuert. Bei modernen Monitoren kann zwar dabei nichts mehr kaputtgehen, aber es wird nichts oder die Fehlermeldung *out of range* ausgegeben.

Neben der Angabe der Monitordaten werden von den Konfigurationsdateien auch noch die gewünschten Bildschirm-Auflösungen und Farbtiefen erfragt. Es können mehrere angegeben werden, die dann später im laufenden Betrieb umgeschaltet werden können.

Neben diesen Hardware-Angaben werden eventuell noch verschiedene Pfade abgefragt, die zu weiteren Informationen führen. Diese Pfade werden später noch einer genaueren Betrachtung unterworfen. Für die Konfiguration reicht hier die Übernahme der Vorgaben.

Erstellen eines Links auf den zu verwendenden X-Server

Der eigentliche X-Server wird später auf verschiedene Art und Weise gestartet. Gemeinsam ist allen Methoden aber, daß sie das Programm `/usr/X11R6/bin/X` aufrufen. Es wird also immer davon ausgegangen, daß der X-Server einfach nur X heißt.

In den XFree86 Versionen vor 4.0 gab es nicht nur einen, sondern sehr viele unterschiedliche X-Server. Jeder Server war für eine bestimmte Graphikkarte (oder eine Klasse von Karten) zuständig. Typische Beispiele dieser Server waren **XF86_Mach64, XF86_VGA16, XF86_SVGA** oder **XF86_S3**. Damit das jeweils verwendete Startprogramm auch den richtigen Server startet, ist das Programm `/usr/X11R6/bin/X` nur ein symbolischer Link auf den entsprechenden Server. Genauer gesagt ist X ein Link auf `/var/X11R6/bin/X` und das ist wiederum ein Link auf den entsprechenden X-Server. Der Grund für diese doppelte Verlinkung ist, daß ansonsten bei einem ReadOnly gemounteten `/usr`-Verzeichnis keine Änderungen möglich wären. Die Verlinkung bei XFree86 Version 3.x sieht also folgendermaßen aus:

```
/usr/X11R6/bin/X => /var/X11R6/bin/X => /usr/X11R6/bin/XF86_Servertyp
```

Diese Links werden auch vom jeweils verwendeten Konfigurationsprogramm angelegt.

Ab der Version 4.0 von XFree86 hat sich diese Architektur geändert. Es war einfach zu viel Aufwand, für jede Graphikkarte einen eigenen X-Server zu schreiben. Aus diesem Grund wurde ein modularisierter X-Server erstellt, der die kartenabhängigen Informationen als Modul nachläd. Der eigentliche X-Server heißt jetzt `/usr/X11R6/bin/XFree86`. Trotzdem existiert meist immer noch der Link von `/usr/X11R6/bin/X` auf `/usr/X11R6/bin/XFree86`. Nur bei Debian ist `/usr/X11R6/bin/X` ein kleines Ladeprogramm, das dann aber auch `/usr/X11R6/bin/XFree86` läd.

Tunen der Konfiguration

Nach abgeschlossener Konfiguration kann der X-Server gestartet werden. Im einfachsten Fall wird das durch die Eingabe des Befehls **startx** erfolgen. (Andere Startmethoden werden im Abschnitt 1.110.2 - Einrichten eines Display Managers beschrieben.)

Falls die Bildschirmeinstellungen nicht zufriedenstellend sind, gibt es zwei Möglichkeiten, daran noch etwas zu verändern. Entweder der Monitor wird mittels seiner Hardwarekonfiguration (Monitormenü oder Drehknöpfe) an die Konfiguration angepasst, oder die Konfiguration wird im Nachhinein noch verändert. Dazu steht uns das Programm **xvidtune** zur Verfügung.

Mit Hilfe dieses Programms können die Bildschirmeinstellungen ähnlich verändert werden, wie mit dem Menü des Monitors selbst. Im Unterschied dazu wird **xvidtune** aber die Einstellungen in der XF86Config Datei verändern, statt sie hardwareseitig einzustellen.

Mögliche Einstellungen sind:

- **Left, Right, Up, Down**
  Verändert den Video-Modus dahingehend, daß sich das sichtbare Bild in die angegebene Richtung verschiebt.
- **Wider, Narrower, Shorter, Taller**
  Verändert den Video-Modus dahingehend, daß das Bild breiter, schmäler, flacher oder höher wird.

Die beste Methode ist es, die Bildschirmeinstellungen mit **xvidtune** soweit wie möglich einzustellen, und dann anschließend das Feintuning mit der Monitoreinstellung vorzunehmen.

Struktur der Datei XF86Config

Alle Konfigurationseinstellungen des X-Servers werden in der Datei `/etc/X11/XF86Config` abgelegt. Diese Datei kann manuell mit einem Texteditor verändert werden. Ihr Format unterscheidet sich zwischen den Versionen 3.x und 4.x erheblich.

Wichtige Einstellungen für die LPI101 Prüfung beziehen sich nur auf einen Abschnitt, den Abschnitt Files. Dieser Abschnitt ist in beiden Versionen vorhanden.

Die weiteren Abschnitte bezeichnen die jeweiligen Konfigurationen der verschiedenen Hardware und Bildschirmauflösungen. Der Vollständigkeit halber seien sie hier kurz aufgezählt:

- **Section "Module"**
  Angabe über ladbare Module.
- **Section "Files"**
  Pfadangaben zu Schriften, Farbinformationen und Modulen
- **Section "ServerFlags"**
  Angaben zu bestimmten Grundeinstellungen des X-Servers
- **Section "InputDevice"**
  Angaben zu allen denkbaren Eingabegeräten (Tastatur, Maus, ...) Erst ab Version 4.0
- **Section "Keyboard"**
  Angaben zur Tastatur. Nur bei Versionen bis 3.x
- **Section "Pointer"**
  Angaben zur Maus. Nur bei Versionen bis 3.x
- **Section "Monitor"**
  Angaben zum Monitor.
- **Section "Device"**
  Angaben zur Graphikkarte
- **Section "Screen"**
  Angaben zur Bildschirmauflösung und Farbtiefe

Optional können noch weitere Abschnitte enthalten sein, die sich auf Extras wie 3D-Beschleuniger (DRI) oder ähnliches beziehen.

Die Sektion "Files"

In diesem Abschnitt werden dem X-Server die Pfade zu drei verschiedenen Informationen übermittelt. Die Pfade zu den Verzeichnissen, in denen die Schriften liegen (FontPath), der Pfad zu der Datei, die die Farbbeschreibungen enthält (RgbPath) und optional der Pfad zu dem Verzeichniss, das die Module für den Server enthält (ModulePath). Diese Angabe muß nur gemacht werden, wenn der Pfad vom Standard abweicht.

Nachdem sowohl der ModulPath, als auch der RgpPath sehr statisch sind, werden wir an diesen beiden Einstellungen eigentlich niemals Veränderungen vornehmen müssen. Die einzigen Einstellungen, die wir tatsächlich verwalten und verändern müssen sind die Pfade zu den Schriftverzeichnissen.

Auf einem X-Server sind normalerweise mehrere Verzeichnisse mit Schriften installiert. Wir können diese Verzeichnisse entweder in einem Eintrag durch Kommas getrennt hintereinanderhängen, oder - was sicherlich zu einer besseren lesbarkeit führt - für jedes Verzeichnis einen eigenen Eintrag vornehmen. Also entweder

```
FontPath="/usr/X11R6/lib/X11/fonts/local/","/usr/X11R6/lib/X11/fonts/misc/"
```

oder

```
FontPath   "/usr/X11R6/lib/X11/fonts/local/"
FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
```

In modernen Versionen wird nur noch die zweite Variante benutzt. Jede der Pfadangaben bezieht sich auf ein Verzeichnis, das Schriftdateien enthält. Die Reihenfolge dieser Angaben ist nicht unwichtig, weil der Server auf der Suche nach einer Schrift immer die Pfade in der angegebenen Reihenfolge durchsucht. Die erste gefundene Schrift, die den Anforderungen der Suche entspricht, wird verwendet. Wenn beispielsweise sowohl TrueType, als auch klassische X11-Schriften installiert sind, und es dabei zu einer Namensüberschneidung kommt (beispielsweise Times und Helvetica), dann kommt es darauf an, ob das Verzeichnis mit den TrueType Schriften vor oder nach dem mit den klassischen Schriften hier angegeben wurde.

Eine Veränderung der Datei XF86Config wird erst wirksam, wenn der X-Server neu gestartet wird. Es gibt aber zumindestens für FontPath-Änderungen auch die Möglichkeit, die neuen Einträge manuell zu aktivieren oder zu deaktivieren. Dazu wird das Programm **xset** benutzt:

```
xset +fp neuer_Pfad
xset fp+neuer_Pfad
```

fügen einen angegebenen neuen Pfad zum FontPath hinzu,

```
xset -fp Pfad
xset fp-Pfad
```

entfernen den angegebenen Pfad aus dem FontPath.

Das Prinzip der Schriftnamen von X11

Schriften können bei X11, wie bei fast jeder anderen graphischen Oberfläche auch, beliebig hinzugefügt oder entfernt werden. Dabei existieren mehrere mögliche Formate, so daß es also möglich ist, daß ein und dieselbe Schrift in einer Installation mehrfach vorhanden ist.

Schriftnamen haben eine etwas gewöhnungsbedürftige Form, hier einmal ein Beispiel, an dem die einzelnen Elemente des Namens erklärt sind.

```
 Hersteller   Gewicht Breite     Pixels  XAufl Sp     Zeichensatz
    |           |        |         |      |     |        |
-b&h-lucida-medium-r-normal-sans-18-180-75-75-p-106-iso8859-1
         |          |          |      |     |     |          |
     Schriftart    Neigung    Stil   Punkte YAufl Durchschn. Optionen des
                                                 Zeichen-   Zeichensatzes
                                                 breite
```

Dabei bedeuten im Einzelnen:

Hersteller
: Meist die Herstellerfirma der Schriftart, z.B. adobe, dec, sony, manchmal aber auch nur eine weitere Schublade wie misc oder bitstream.

Schriftart
: Die "Schriftfamilie" also etwa Times, Symbol, Utopia,...

Gewicht

Das "Gewicht" der Schwärze wie black, bold, demibold oder medium.

Neigung
Die Schräge der Schrift. i steht für italic (kursiv), r für roman (aufrecht).

Breite
Die Breite einer Schrift wie normal, narrow oder semicondensed.

Stil
Zusätzliche Stilangaben

Pixels
Das Höhenmaß der Schrift in Pixel (in einer bestimmten Punktgröße und Auflösung)

Punkte
Die Größe der Schrift in typographischen Punkten (1/72 Zoll) angegeben in Zehntel-Punkten

X-Auflösung
Horizontale Auflösung in Dots per Inch (DPI) in der die Schrift ursprünglich erstellt wurde.

Y-Auflösung
Vertikale Auflösung in Dots per Inch (DPI) in der die Schrift ursprünglich erstellt wurde.

Spacing
Die Beschreibung der Raumaufteilung einer Schriftart. Hier geht es darum, ob eine Schriftart proportional aufgebaut ist, also für ein i einen kleineren Platz als z.B. für ein o benötigt, oder nicht. Mögliche Werte sind hier p für proportional, m für monospaced also nicht proportional und c für charactercell, einer Technik, in der allen Zeichen ein gleichgroßer Rahmen gegeben wird, was sie auch wieder zur nicht-proportionalen Schrift macht.

Durchschnittsbreite
Die durchschnittliche Breite eines Zeichens dieser Schrift.

Zeichensatz
Die ISO-Standard, die diese Schrift enthält.
Die Nummer der ausgewählten Zeichensatzseite des Standards.

Oft werden statt einzelner Angaben auch ein Sternchen angegeben, dann wird die erste passende Einstellung genommen. So wird ein * bei der Angabe der Neigung automatisch zu einem i.

Verwaltung von Schriften unter X11

Wenn neue Schriften installiert werden sollen, dann werden ein paar Einstellungen nötig, damit sie benutzbar werden.

Verzeichnisse, die Schriften für X11 enthalten, enthalten neben den eigentlichen Schriftdateien (Endungen .bdf .snf .pcf . spd .fb .ps .ttf) noch die folgenden Dateien:

fonts.dir
Diese Datei enthält die Angaben, welche Schrift in welcher Datei zu finden ist. Am Anfang der Datei steht eine Nummer, die Anzahl der im Verzeichnis enthaltenen Schriften. Die weiteren Zeilen der Datei enthalten paarweise die Dateinamen mit zugehörigem Schriftnamen, also etwa

```
courBO10.snf  -adobe-courier-bold-o-normal--10-100-75-75-m-60-iso8859-1
```

Damit diese Datei immer auf dem neuesten Stand ist, muß nach jeder Neuinstallation von Schriftdateien in einem Verzeichnis in diesem Verzeichnis das Programm **mkfontdir** aufgerufen werden. Alternativ kann auch der Name des Verzeichnisses dessen fonts.dir Datei neu erstellt werden soll, als Parameter von **mkfontdir** angegeben werden.

fonts.scale
Neben den Bitmap Fonts existieren auch noch skalierbare Schriften. In den Verzeichnissen mit solchen skalierbaren Schriften (z.B. speedo) existiert neben der fonts.dir auch die Datei fonts.scale. Sie enthält je einen Eintrag pro skalierbarer Schrift, mit den Größenangaben jeweils auf 0 gesetzt.

fonts.alias
Diese Datei enthält beliebige Alias-Namen, mit den dazugehörigen Schriftnamen. So kann etwa mit der Zeile

```
school15 -adobe-new century schoolbook-bold-r-normal-*-15-150-75-*-*-105-
iso8859-1
```

die Schrift -adobe-new ... schlicht mit school15 angesprochen werden. Dieser Mechanismus hat aber Vor- und Nachteile. Es ist zwar so für einen User einfacher, eine Schrift auszuwählen, es kann aber bei einer Anwendung, die auf verschiedenen Rechnern läuft fatal sein, wenn es die Schrift bzw. eben das Alibi auf einem anderen Rechner nicht gibt.

Wenn ein neues Verzeichnis mit Schriften also hinzugefügt wird, indem es beispielsweise neu erstellt und mit Schriftdateien

aus dem Internet gefüllt wird, so sind folgende Schritte nötig, damit die Schriften benutzbar werden.

- Ausführen des Programmes **mkfontdir** *Verzeichnis*
- Verzeichnis in den FontPath aufnehmen (indem ein Eintrag in `/etc/X11/XF86Config` vorgenommen wird - siehe oben)
- Neustart des X-Servers oder manuelle Aufnahme des Verzeichnisses mit **xset** +fp *Verzeichnis*.

Ab diesem Zeitpunkt sind die Schriften auf dem X-Server verfügbar.

Installation eines FontServers

Das Schriftenmodell unter X11 erfordert relativ viel Speicherplatz, weil bei den meisten installierten Schriftarten jede Schrift eine eigene Schriftdatei benutzt. Aus diesem Grund gibt es die Möglichkeit, einen Fontserver zu installieren, der beliebig vielen X-Servern im Netz Schriften zur Verfügung stellt. So müssen die Schriftdateien nur auf einem Rechner vorhanden sein und stehen trotzdem allen X-Servern im Netz zur Verfügung.

Ein weiterer Vorteil dieser Technik ist es, daß ein Unternehmen bestimmte Schriften für seine Präsentationen benutzen kann und zentral festlegen kann, daß alle X-Server die selben Schriften benutzen.

Für die Ansprüche der LPI101 Prüfung muß sowohl ein Fontserver eingerichtet werden können, als auch ein X-Server an einen Fontserver angebunden werden, so daß er die vom Server angebotenen Schriften nutzen kann.

Konfiguration eines Fontservers

Der Fontserver für X11 heißt **xfs** und ist ein eigenständiger Daemon. Er wird in der Regel über ein init-Script gestartet und bietet dann seine Dienste an.

Wie so oft bei Linux/Unix-Servern besitzt **xfs** eine einzige Konfigurationsdatei, in der die gesamte Konfiguration des Servers erledigt wird. Diese Konfigurationsdatei liegt standardmäßig unter `/usr/X11R6/lib/X11/fs/config`. Da diese Plazierung dem neuen Dateisystem-Standard widerspricht, kann alternativ dazu dem **xfs** per Kommandozeilenoption eine andere Datei mitgeteilt werden, die er als Konfigurationsdatei nutzt. In modernen Versionen von Linux wird meist `/etc/X11/xfs.conf` oder `/etc/X11/fs/config` benutzt.

Der Fontserver muß die Schriftdateien in Verzeichnissen ablegen, die exakt den oben genannten Ansprüchen der Schriftverzeichnisse eines normalen X-Servers entsprechen. Er kann jetzt diese Verzeichnisse an andere X-Server weitergeben, ohne daß dabei zusätzlicher Verwaltungsaufwand entsteht.

Eine einfache Konfigurationsdatei eines Fontservers könnte folgendermaßen aussehen:

```
# Erlaubt maximal 10 Clients, die sich mit diesem Fontserver verbinden
client-limit = 10

#
# Wenn ein Fontserver die maximale Anzahl Clients erreicht hat,
# startet er einen neuen um weitere Clients bedienen zu können
clone-self = on

#
# Angabe der Schriftverzeichnisse
#
catalogue = /usr/local/fonts/tr_misc/,
            /usr/local/fonts/tr_75dpi/,
            /usr/local/fonts/freefont/,
            /usr/local/fonts/sharefont/,
            /usr/local/fonts/Adobe/,
            /usr/X11R6/lib/X11/fonts/misc:unscaled,
            /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/Type1,
            /usr/X11R6/lib/X11/fonts/Speedo,
            /usr/X11R6/lib/X11/fonts/misc,
            /usr/X11R6/lib/X11/fonts/75dpi
```

```
# Voreingestellte Schriftgröße ist 12 Punkte (120 Dezipunkte)
default-point-size = 120

# 100 x 100 und 75 x 75 Auflösung
default-resolutions = 100,100,75,75

# Keine Meldungen an den Syslogdaemon
use-syslog = off
```

Die Kommentare in der Datei erklären die jeweiligen Zeilen ausreichend. Wenn diese Datei jetzt als `/etc/X11/fs/config` abgespeichert wird, dann lautet der Aufruf des XFontServers:

```
xfs -config /etc/X11/fs/config -daemon
```

Diese Zeile sollte jetzt aus einem init-Script heraus aufgerufen werden. Nachdem sie ausgeführt wurde, ist der Fontserver aktiv und hört auf den TCP-Port 7100.

Soll einem laufenden Fontserver ein neues Schriftverzeichnis hinzugefügt werden, so muß es in die Konfigurationsdatei eingefügt werden und danach dem **xfs**-Prozeß ein HUP-Signal geschickt werden. Das zwingt ihn, seine Konfigurationsdatei neu einzulesen und sich entsprechend zu verhalten.

Anbindung eines X-Servers an einen Fontserver

Damit ein beliebiger X-Server jetzt den Fontserver benutzen kann, reicht es, in der Datei XF86Config in der Section "Files" einen FontPath Eintrag in der Art

Transportprotokoll/*Fontservername*:*Portnummer*

zu machen. Wenn unser Font-Server auf dem Rechner **marvin** auf dem Standard-TCP-Port 7100 läuft, so kann jeder X-Server im Netz den Eintrag

```
FontPath   "TCP/marvin:7100"
```

in die Section "Files" aufnehmen und nach einem Neustart des X-Servers (nicht des Rechners) stehen die Schriften des Font-Servers zur Verfügung. Auch hier spielt die Reihenfolge der Angaben eine Rolle. Der FontPath wird von vorne nach hinten (oben nach unten) durchgearbeitet und die erste passende Schrift wird gewählt. Steht also der Font-Server Eintrag am Anfang des FontPath, so werden wahrscheinlich niemals lokale Schriften benutzt (es sei denn es gibt lokale Schriften, die der FontServer nicht anbietet), steht er am Ende, werden immer lokale Schriften verwendet, außer es gibt eine bestimmte Schrift nicht lokal, dann wird der Font-Server herangezogen.

Verwendung von Schriften

Die in den Prüfungszielen angesprochene Datei `.Xresources` kann benutzt werden, um bestimmte Schriften für bestimmte Anwendungen fest einzustellen. Diese Technik wird im Abschnitt 1.110.4 - Installation und Anpassung einer Window Manager Umgebung genau besprochen.

1.110.2 - Einrichten eines Display Managers

Beschreibung: Prüfungskandidaten sollten in der Lage sein, einen Display Manager einzurichten und anzupassen. Dieses Lernziel beinhaltet das Aktivieren und Deaktivieren des Display Managers und das Ändern der Willkommensmeldung. Ebenfalls enthalten ist das Ändern der voreingestellten Bitplanes des Display Managers. Weiters enthalten ist die Konfiguration des Display Managers für die Verwendung auf X-Stationen. Das Lernziel deckt die Display Manager XDM (X Display Manager), GDM (Gnomde Display Manager) und KDM (KDE Display Manager) ab.

Die wichtigsten Dateien, Bezeichnungen und Anwendungen:

- `/etc/inittab`
- `/etc/X11/xdm/*`
- `/etc/X11/kdm/*`
- `/etc/X11/gdm/*`

Startmöglichkeiten des X-Servers

Es existieren grundsätzlich zwei Möglichkeiten, einen X-Server zu starten. Er kann entweder als Anwenderprogramm gestartet werden, nachdem sich der aufrufende User normal eingeloggt hat, oder er stellt bereits den Login graphisch zur Verfügung.

Die erste Methode wird gewöhnlich dadurch erreicht, daß der Befehl **startx** aufgerufen wird. **startx** ist ein Shellscript, das den Befehl **xinit** aufruft. Dieses Programm startet dann sowohl den X-Server, als auch zusätzliche Programme, wie den Window-Manager oder die Desktop Umgebung. Die Konfiguration von **xinit** wird über die Datei `$HOME/.xinitrc` oder falls diese Datei nicht existiert über `/var/X11R6/lib/xinit/xinitrc` konfiguriert. Der X-Server wird also als normale Anwendung gestartet. Dieses System hat erhebliche Sicherheitslücken, weil "hinter" dem X-Server eine offene Konsole liegt, von der aus der Befehl **startx** eingegeben wurde.

Um diese Sicherheitslücke zu schließen, kann die gesamte Kontrolle des Einloggens auch an X11 weitergegeben werden. In diesem Fall wird der X-Server nicht von einem Anwender per Befehl gestartet, sondern von einem Daemon-Prozess, dem **X-Display-Manager**. Dieser Daemon läuft ständig im Hintergrund und stellt für jeden angeschlossenen X-Server (in der Regel nur einer) ein graphisches Login-Fenster zur Verfügung. Wenn sich ein User über dieses Fenster anmeldet, dann startet der X-Display-Manager gleich den X-Server, der User bekommt also sofort eine graphische Oberfläche, ohne überhaupt auf einem Textterminal eingeloggt zu sein.

Der Display-Manager ist ein Daemon-Prozeß, er sollte also über ein init-Script gestartet werden. Ist der Display-Manager aktiv, so ist es - zumindestens ohne Tricks - nicht mehr möglich, über ein Textterminal mit **startx** den X-Server als normale Anwendung zu starten. Aus diesem Grund wird der Display-Manager in der Regel nur in einem ganz bestimmten Runlevel gestartet.

Diese Einschränkung erlaubt es, den Display-Manager beliebig an- und auszuschalten, indem einfach der Runlevel entsprechend verändert wird. Die meisten Distributionen haben eine entsprechende Voreinstellung, also einen Runlevel, der alle Netzwerkdienste startet und zusätzlich den Display-Manager aktiviert und einen anderen, der alle Netzwerkdienste startet aber keinen Display-Manager.

Welche Runlevel dazu verwendet werden ist nicht standardisiert. Hier ein paar Beispiele für wichtige Distributionen:

Distribution	Runlevel ohne DM	Runlevel mit DM
SuSE bis 7.2	3	5
SuSE ab 7.3	2	3
RedHat	3	5
Caldera	3	5
Slackware	3	4

Nur bei Debian gibt es keinen solchen Runlevel voreingestellt, da die Debian Distribution davon ausgeht, daß sobald der Display-Manager installiert ist, auch sein Einsatz gewünscht wird.

In der Regel sollte in der Datei `/etc/inittab` eine Liste der zur Verfügung stehenden Runlevel als Kommentare stehen, und danach die Einstellung, welcher Runlevel beim Systemstart verwendet werden soll (*initdefault*). Beispielsweise könnte hier stehen:

```
...
#
# runlevel 0  is  System halt   (Do never use this for initdefault)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot (Do never use this for initdefault)
#

id:5:initdefault:
...
```

Dem wäre also zu entnehmen, daß der Runlevel 5 der Runlevel ist, in dem der Display-Manager (xdm) gestartet werden soll und daß dieser Runlevel auch der voreingestellte Runlevel ist.

Um einen Linux-Rechner also zu zwingen, den Display-Manager zu benutzen, muß der Eintrag in `/etc/inittab` entsprechend auf den Runlevel geändert werden, der für den Display-Manager Runlevel steht. Um Linux zu zwingen, den Display-Manager **nicht** zu benutzen, so muß entsprechend der Eintrag geändert werden, so daß in unserem obigen Beispiel die Zeile jetzt lauten müßte:

```
...
id:3:initdefault:
...
```

Der Display-Manager-Runlevel sollte erst aktiviert werden, wenn sichergestellt ist, daß die Konfiguration des X-Servers tatsächlich funktioniert. Ansonsten geht man Gefahr, daß sich das System nicht mehr vernünftig bedienen läßt, weil beim Systemstart gleich die graphische Oberfläche geladen wird und die womöglich nur einen flackernden Bildschirm produziert.

Verschiedene Display-Manager und ihre Konfiguration

Der klassische Display-Manager des XFree86-Projekts war und ist der **xdm**. Dieser Display-Manager stellt ein relativ anspruchsloses Login-Fenster zur Verfügung, das nur je eine Eingabezeile für Username und Passwort anbietet.

Sowohl das KDE-, als auch das Gnome-Projekt haben jeweils einen eigenen Display-Manager im Angebot. Der Displaymanager von KDE heißt **kdm**, der von Gnome ist **gdm**. Der primäre Unterschied zum **xdm** liegt im entsprechenden Look&Feel. Allerdings bieten **gdm** und **kdm** noch einige interessante Zusatzfeatures:

- User werden als Bilder (evt. Passfotos) dargestellt und können angeklickt werden, statt den Usernamen direkt einzugeben.
- Der gewünschte Window-Manager der zu startenden X-Session kann per Menü ausgewählt werden.
- Das System kann heruntergefahren oder neu gestartet werden, ohne sich vorher einloggen zu müssen.

Der **kdm** basiert sehr stark auf dem **xdm**, das heißt, er nutzt sowohl XDM-Konfigurationsdateien, als auch Teile des Quellcodes von **xdm** Im Gegensatz dazu ist der **gdm** komplett neu geschrieben und daher etwas unabhängiger von **xdm**.

Die Konfiguration der jeweiligen Displaymanager findet nach dem neuen Dateisystemstandard immer in den folgenden Verzeichnissen statt:

Displaymanager	**Konfigurationsverzeichnis**
xdm	`/etc/X11/xdm`
gdm	`/etc/X11/gdm`
kdm	`/etc/X11/kdm`

In diesen Verzeichnissen liegen unterschiedliche Konfigurationsdateien. Die Verzeichnisse von **kdm** und **xdm** ähneln sich sehr stark, was auf ihre Verwandschaft zurückzuführen ist, das Verzeichnis von **gdm** ist vollkommen anders aufgebaut.

Konfiguration von xdm

Der optische Aufbau von **xdm** ist sehr simpel gehalten. Ein Login-Fenster enthält eventuell ein Logo und eine einfache Willkommensmeldung. Die einzige Einstellungsmöglichkeit außer der des Logos und der Willkommensmeldung bezieht sich auf den Hintergrund des Bildschirms, auf dem das Login-Fenster erscheint.

xdm ist ein typisches X11-Programm, wird also durch Resources konfiguriert. Die Einstellung der von **xdm** benötigten Resources liegt in der Datei `/etc/X11/xdm/Xresources`.

In dieser Datei können die folgenden Einstellungen getroffen werden:

- Die Willkommensmeldung kann mittels der Ressource `xlogin*greeting` verändert werden. Ein eventuelles Vorkommen des Wortes `CLIENTHOST` wird in der Ausgabe durch den Hostnamen des Rechners ersetzt.
- Die Schriftart der Willkommensmeldung wird über die Ressource `xlogin*greetFont` eingestellt.
- Das darzustellende Logo wird über die Ressource `xlogin*logoFileName` eingestellt. Das Dateiformat des Logos sollte XPM sein.
- Die Farben des Login-Fensters und seine Rahmenbreite können über Ressourcen wie `xlogin*borderWidth, xlogin*frameWidth, xlogin*background, xlogin*foreground` und `xlogin*greetColor` angepasst werden.

Um schließlich den Bildschirmhintergrund zu verändern, kann eine weitere Datei verwendet werden. Im Verzeichnis `/etc/X11/xdm` liegt ein Shellscript `Xsetup`. Dieses Script wird jedesmal abgearbeitet, wenn xdm eine Loginmeldung auf einen Bildschirm schreibt. In dieses Script können jetzt schon Befehle eingetragen werden, die den Bildschirmhintergrund verändern, wie etwa **xpmroot**, **xsetbg** oder jedes beliebige andere Programm, das den Bildschirmhintergrund modifiziert.

Nehmen wir an, im Verzeichnis `/etc/X11/xdm` befindet sich eine Datei `background.xpm`. Diese Datei enthält das gewünschte Hintergrundbild. Dann tragen wir in die Datei `/etc/X11/xdm/Xsetup` die folgende Zeile ein:

```
/usr/bin/xpmroot /etc/X11/xdm/background.xpm
```

Damit wird das gewünschte Bild jedesmal auf den Hintergrund dargestellt, wenn **xdm** ein Loginfenster darstellt.

Konfiguration von kdm

Im Prinzip funktionieren grundsätzlich alle Einstellungen, die für **xdm** gezeigt wurden auch für **kdm**. Der einzige Unterschied ist der, daß die Einstellungen jetzt in `/etc/X11/kdm` vorgenommen werden müssen.

Allerdings gibt es eine zweite Konfigurationsmöglichkeit für **kdm**, die wesentlich mehr Einstellungen erlaubt, als die Möglichkeiten, die wir bereits besprochen haben. Normalerweise werden diese zusätzlichen Einstellungen über ein graphisches Programm erledigt, das Teil des K-Control-Centers ist. Unter dem Menüpunkt *System/Anmeldungsmanager* können die verschiedensten Einstellungen vorgenommen werden. Alle Einstellungen, die über diese Methode vorgenommen wurden, werden in der Datei `/etc/X11/kdm/kdmrc` gespeichert und können dort auch manuell verändert werden.

Einer der wesentlichen Unterschiede zwischen **kdm** und **xdm** ist die Darstellung der User in Bildform zum Auswählen. Auch die Frage, welche Bilder verwendet werden sollen, kann mit dem k-Control-Center eingestellt werden. Allerdings kann auch einfach je ein Bild (im Format XPM) für jeden User in das Verzeichnis *kde-Wurzel*`/apps/kdm/pics/users` gelegt werden, dessen Namen dem des Users entspricht. Das Bild `hans.xpm` würde also für den User **hans** benützt.

Konfiguration von gdm

Der Display-Manager des Gnome-Projektes arbeitet nicht mit dem Quellcode von **xdm**, daher ist seine Konfiguration auch entsprechend anders, als die der beiden anderen Displaymanager. Die Konfiguration wird grundsätzlich mit dem Programm **gdmconfig** vorgenommen. Es handelt sich auch um ein graphisches Programm mit entsprechenden Einstellungsmöglichkeiten.

Die Einstellungen dieses Programmes werden in der Datei `/etc/X11/gdm/gdm.conf` abgespeichert und können hier auch manuell verändert werden. In der Regel wird das aber nie der Fall sein, da die graphische Konfiguration wesentlich komfortabler ist.

Damit auch der **gdm** die Bilder der User darstellt, muß dieses Feature ersteinmal aktiviert werden (über **gdmconfig**). **gdm**

bietet jetzt aber die Möglichkeit, daß jeder User sein eigenes Bild einstellt. Dazu kann er das Programm **gdmphotosetup** benutzen.

XTerminals mit dem Displaymanager bedienen

Die Möglichkeiten des graphischen Logins beschränken sich nicht alleine auf den lokalen Rechner. Es ist auch möglich, daß sich User fremder Rechner auf unserem Rechner graphisch einloggen. Dazu stellt das X-Protokoll ein spezielles Netzwerkprotokoll zur Verfügung, XDMCP (X-DisplayManager Control Protocol). Damit ist es möglich, daß ein fremder X-Server das Bild unseres Displaymanagers auf seinem Bildschirm zu sehen bekommt und sich dann graphisch auf unserem Rechner einloggt. Alle Programme - außer dem X-Server selbst - die während dieser Session auf dem fremden Rechner dargestellt werden, laufen nicht mehr auf seiner CPU sondern auf der des Servers, auf dem er sich eingeloggt hat. So kann z.B. auch noch ein alter 486er als graphischer Arbeitsplatz genutzt werden.

Damit diese Möglichkeit besteht, muß der Displaymanager zunächst einmal so konfiguriert werden, daß er Zugriffe von fremden Rechnern auch zuläßt. Das geschieht bei **xdm** und **kdm** wieder exakt gleich, nur der **gdm** hat wiederum seine eigenen Einstellmöglichkeiten.

XDMCP aktivieren

Bei **xdm** und **kdm** müssen zunächst einmal die grundsätzliche Bereitschaft des Display-Managers aktiviert werden, XDMCP-Pakete anzunehmen. Das geschieht in der Datei `xdm-config` bzw. `kdm-config` im jeweiligen Verzeichnis des Display-Managers. Dort steht - meist in der letzten Zeile - der Eintrag

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

Die Zeile, die den `requestPort` auf 0 setzt muß auskommentiert werden, wenn ein Login von fremden Rechnern aus möglich sein soll. In beiden Fällen wird durch ein Ausrufungszeichen diese Zeile deaktiviert, also:

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

Damit ist die grundsätzliche Bereitschaft zur Bedienung aktiviert.

Um den selben Schritt mit **gdm** zu vollziehen, muß über das Programm **gdmconfig** im Submenü *Expert* im Einstellungstab *XDMCP* das Schaltkästchen *XDMCP aktivieren* angeschaltet werden. Dadurch wird der Eintrag `Enable=true` in der Sektion `[xdmcp]` der Datei `/etc/X11/gdm/gdm.conf` vorgenommen.

Zugriffe erlauben

Für **xdm** und **kdm** existiert jetzt noch die Notwendigkeit, einzustellen, welche Rechner das Recht haben sollen, sich graphisch einzuloggen. Dazu existiert die Datei `Xaccess` im jeweiligen Verzeichnis des Displaymanagers. Dort kann festgelegt werden, welcher Rechner

- ein Login-Fenster bekommen soll
- ein Auswahlfenster weiterer Server, die xdm-Logins anbieten, bekommen soll

Dazu werden in dieser Datei Einträge nach dem Muster

Hostnamensmuster

für Direct und Broadcast Zugriffe gemacht. Der einfachste Eintrag wäre also eine Zeile mit nur einem Sternchen (*), der Zugriff für alle Rechner erlaubt.

Den Zugriff auf die Auswahl anderer Rechner wird mit

*Hostnamensmuster*CHOOSER BROADCAST

geregelt.

Eine solche Datei, die also allen Rechnern der Domaine `foo.bar` die Benutzung unseres graphischen Logins erlauben soll könnte folgendermaßen aussehen:

```
*.foo.bar
*.foo.bar       CHOOSER BROADCAST
```

Clientseitige Einstellungen

Normalerweise - wenn der X-Server entweder über xinit (startx) oder xdm gestartet wird, wird nach dem Start des X-Servers sofort noch ein Fenster-Manager (fvwm2, kwm, gwm, ...) gestartet, der dann wiederum die Oberfläche konfiguriert. Das ist anders, wenn wir uns auf einem fremden System graphisch einloggen wollen.

Zu diesem Zweck wird der X-Server tatsächlich direkt aufgerufen, also weder durch **startx**, noch durch **xdm** sondern durch den Befehl **X**. Dazu kommen drei Techniken in Betracht. DIRECT, INDIRECT und BROADCAST.

Die DIRECT-Methode spricht direkt einen Rechner im Netz an um sich bei ihm einzuloggen. Die Systax dazu ist

```
X :0.0 -query Hostname
```

Jetzt wird der Server 0 gestartet, seine gesammte Konfiguration übernimmt aber der **xdm** des Rechners *Hostname*. Der Rechner, auf dem wir diesen Befehl gestartet haben dient also nur als X-Terminal für den unter *Hostname* angegebenen Rechner.

Die zweite Methode (INDIRECT) geht einen anderen Weg. Sie frägt einen Rechner nicht nach dem Login-Fenster, sondern nach einer Liste von bekannten Servern, die den xdm-Dienst (aso den Dienst sich graphisch einloggen zu können) anbieten. Der gefragte Rechner startet jetzt diese Auswahlliste (den sogenannten CHOOSER) und der aufrufende Rechner kann sich aus dieser Liste aussuchen, wo er sich einloggen will. Um diese Methode anzuwenden lautet der Befehl:

```
X :0.0 -indirect Hostname
```

Die dritte Methode ist schließlich ein Broadcast, der ins lokale Netz ausgesandt wird, um einen Rechner oder CHOOSER zu bekommen. In diesem Fall wird kein Rechnername angegeben, sondern einfach nur ein

```
X :0.0 -broadcast
```

Statt dem ersten zu startenden Server kann bei jedem der drei Methoden auch ein zweiter, dritter, usw gestartet werden, was dann statt der Angabe `:0.0` entsprechend die Angaben `:1.0, :2.0` usw. erfordert.

1.110.4 - Installation und Anpassung einer Window Manager Umgebung

Beschreibung: Prüfungskandidaten sollten in der Lage sein, eine systemweite Desktopumgebung und/oder einen Window Manager einzurichten und ein Verständnis der Anpassungsprozedur für Menüs der Fenstermanager und oder der Desktop-Panele demonstrieren. Dieses Lernziel beinhaltet die Auswahl und Konfiguration des gewünschten X-Terminals (xterm, rxvt, aterm etc.), das Prüfen und Auflösen von Bibliotheksabhängigkeiten von X-Anwendungen und das Exportieren der Ausgabe von X-Anwendungen auf einen Client.

Die wichtigsten Dateien, Bezeichnungen und Anwendungen:

- `.xinitrc`
- `.Xdefaults`
- **xhost**
- `DISPLAY` Umgebungsvariable

Window-Manager

Der X-Server selbst enthält noch keinerlei Mechanismen zum Bewegen, zur Größenveränderung oder zum "Iconifizieren" von Fenstern. Auch die Ausstattung von Fenstern mit Rahmen findet noch nicht statt, wenn nur der Server läuft. All diese Aufgaben übernehmen Fenster-Manager (Windowmanager) wie der Motif Window Manager (mwm), der Open-Look Window Manager (olwm) oder fvwm. Auch die KDE-Desktop-Umgebung bringt einen eigenen Window-Manager mit, der kwm, während die Gnome-Desktop-Umgebung mit vielen Window-Managern zusammenarbeitet und keinen eigenen besitzt.

Erst durch das Starten eines Window-Managers wird der X-Server zur arbeitsfähigen Graphikstation. Erst der Window-Manager erlaubt es dem User, Fenster zu verschieben, in der Größe zu verändern oder in ein Icon zu verwandeln. Er definiert meist auch Menüs, die durch Drücken der verschiedenen Maustasten aktiviert werden können (wenn sich der Mauszeiger nicht innerhalb eines Fensters befindet).

Die Konfiguration der einzelnen Window-Manager ist so unterschiedlich, daß sich keinerlei allgemeine Regeln dazu formulieren lassen außer der, daß fast immer eine persönliche Konfigurationsdatei im jeweiligen Home-Verzeichnis eines Users befindet, die immer mit einem Punkt beginnt und auf rc endet. dazwischen steht meist der Name des Window-Managers. Also hat Motif eine `.mwmrc`-Datei, olwm wird in `.olwmrc` konfiguriert und fvwm2 in `.fvwm2rc`.

Die beiden Desktop-Umgebungen **KDE** und **Gnome** bieten zusätzlich zu den Features der Window-Manager noch eine vereinheitlichte Darstellung der jeweils verwendeten Programme und eine Schnittstelle für Programmierer, um ihre Programme entsprechend für diese Umgebungen auszustatten. Auch Features wie Drag&Drop werden dabei unterstützt.

Diese beiden Spezialfälle, die heute wohl eher als die Regel statt die Ausnahme zu bezeichnen sind, bieten vor allem eine Möglichkeit, ihre Konfiguration auf eine Art und Weise vornehmen zu lassen, wie sie die BenutzerInnen von Windows gewohnt sind. Kontextmenüs über die rechte Maustaste und Controlcenter in denen die gesamte Konfiguration graphisch vorgenommen werden kann, gehören zur Grundausstattung.

Konfiguration von X-Clients

Weil X-Clients (also X-Anwendungsprogramme) auf allen X-Servern in einem Netz dargestellt werden können, muß jede X-Anwendung konfigurierbar sein. So portierbar das X-System auch ist, so unterschiedlich sind doch einige Voraussetzungen auf unterschiedlichen Servern. Bildschirme können unterschiedliche Auflösungen haben, Graphikkarten unterschiedliche Farbtiefen. Ein Programm, das auf einem Server gut dargestellt wird kann auf einem anderen nicht auf den Schirm passen oder farblich falsch dargestellt werden. Eine Taste, die auf einer Tastatur leicht erreichbar ist kann auf einer anderen dazu führen, daß man sich fast den Finger bricht...

Auf anderen Fenstersystemen wie MS-Windows oder dem Macintosh-Finder werden alle Anwendungen auf der Anwendungsebene konfiguriert. Das macht Sinn, weil sowohl DOS als auch das Mac-OS Single-User-Systeme sind. Alle Konfigurationen können zentral gespeichert werden.

X11-Programme müssen anderen Anforderungen standhalten. Erstens ist Unix ein Mehrbenutzersystem und es ist daher

nötig, daß alle User ihre individuellen Einstellungen machen können, zweitens weiß das X11-Programm noch nicht, auf welchem Server es dargestellt werden soll, es muß also auch in dieser Hinsicht konfigurierbar sein.

Es stehen bei allen X11-Clients zwei Konfigurationsmöglichkeiten zur Verfügung, die Kommandozeilenparameter und die Resourcen.

Konfiguration von X-Clients über Kommandozeilenparameter

Anhand des Clients **xterm** kann schön dargestellt werden, wie die klassischen Kommandozeilenparameter eines X-Clients angewandt werden. All diese Parameter werden von allen X-Clients verstanden, **xterm** ist hier also nur ein Beispiel.

Schriftenauswahl

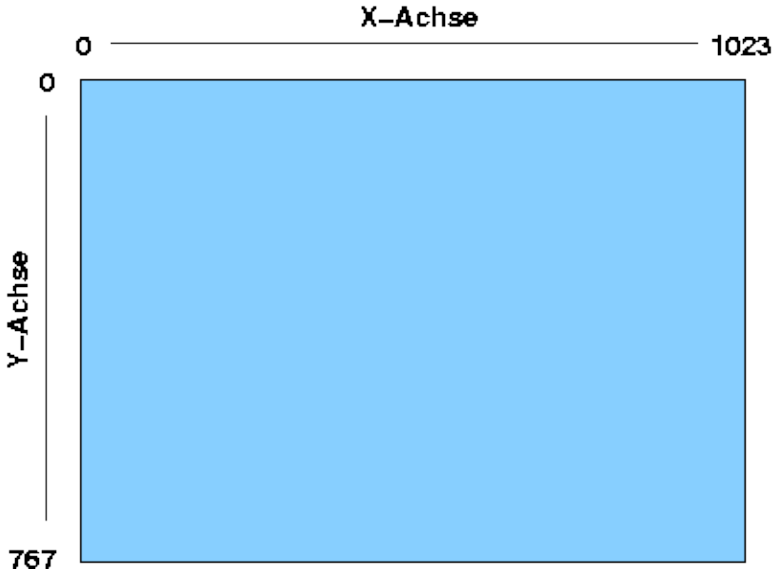
Die Auswahl von Schriften erfolgt mit dem Kommandozeilenparameter -fn. Um ein xterm zu starten, das die Schrift `-misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1` benutzen soll reicht die Angabe

```
xterm -fn -misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1 &
```

Damit ist ein xterm mit anderer Schrift gestartet. Vorsicht ist übrigens angesagt bei der Verwendung von anderen Schriftfamilien. In der Regel ist das kein Problem, aber ein xterm-Fenster simuliert eben ein Terminal. Werden hier Proportionalschriften verwendet, die für ein i weniger Platz als für ein o benutzen, so kann es zu heftigen Problemen beim Arbeiten mit bildschirmorientierten Programmen kommen. Eine Aufteilung in untereinanderliegende Spalten ist ja dann nicht mehr möglich. Xterm sollte also immer mit Schriften der Familie fixed benutzt werden.

Fenstergeometrie

X11 verwaltet den Bildschirm als ein Koordinatensystem. Die obere linke Ecke entspricht der Position 0,0. Die unteren Ecken werden durch die maximale Auflösung bestimmt, genau wie die obere linke Ecke. Bei einer Auflösung von 1024x768 sieht das beispielsweise so aus:



Ein X-Client kann mit dem Parameter `-geometry` auf zweierlei Arten konfiguriert werden. Die Größe und die Position des Fensters auf dem Schirm kann angegeben werden. Zunächst zur Größe:

Wir wollen ein xterm-Fenster, das statt der Standardgröße 80 Spalten in 24 Zeilen jetzt 90 Spalten in 30 Zeilen besitzt. Kein Problem:

```
xterm -geometry 90x30 &
```

Nach dem `-geometry` folgt also die Angabe der Spalten und Zeilen durch ein x getrennt. (Wem das geometry zu lang ist, der kann beruhigt sein - in den meisten Fällen funktioniert auch ein -g statt dessen).

Achtung: Die Angabe von Zeilen und Spalten bezog sich beim xterm auf Zeichen statt auf Pixel. Graphikorientierte Programme rechnen hier mit Pixel, zeichenorientierte Programme mit Zeichen. Um also etwa ein Programm wie xeyes (die wohlbekannten Augen) in der Größe zu verändern, muß mit Pixeln gerechnet werden.

Die andere Aufgabe, die mit `-geometry` erfüllt werden kann ist die Positionierung eines Fensters auf dem Schirm. Dazu wird der Offset zum Rand des Schirms angegeben, ein Pluszeichen meint den oberen bzw. den linken Rand, ein Minuszeichen den unteren bzw. rechten:

```
xterm -geometry +10+50 &
```

positioniert die obere linke Ecke des Fensters also 10 Pixel vom linken Rand entfernt und 50 Pixel vom oberen Rand. Bei dieser Angabe sind übrigens immer Pixel die Maßangabe, nie Zeichen. Um sich auf die untere linke Ecke zu beziehen können wir schreiben:

```
xterm -geometry -30-100 &
```

Wir erhalten ein Fenster, dessen untere rechte Ecke 30 Pixel vom rechten Rand des Bildschirms und 100 vom unteren Rand entfernt ist. Es ist sogar möglich, diese Angaben zu mischen. Die Zeile

```
xterm -geometry +10-100 &
```

meint also ein Fenster, dessen linker Rand 10 Pixel vom linken Rand des Bildschirms entfernt ist und dessen untere Begrenzung 100 Pixel Abstand zum unteren Rand aufweist.

Die Angaben von Größe und Position lassen sich auch kombinieren. Dabei wird immer die Größe zuerst angegeben, danach (ohne Leerzeichen aber mit Vorzeichen + oder -) die Positionsangaben. Um etwa ein xterm mit 90 Spalten in 30 Zeilen an der Position 10,100 zu erhalten schreiben wir:

```
xterm -geometry 90x30+10+100 &
```

Wird keine Angabe zur Position gemacht, so übernimmt der Window-Manager die Positionierung. Dabei kann meist eingestellt werden, auf welche Art und Weise diese Positionierung erfolgt (Userdefiniert mit Maus, zufällig, auf freien Bildschirmteilen, ...).

Farbauswahl

X-Clients haben fast immer die Möglichkeit, ihre Farbdarstellungen einzustellen. Dazu dienen in der Regel die Parameter `-fg` für die Vordergrundfarbe (foreground) und `-bg` für den Hintergrund (background).

Um also ein xterm mit blauem Hintergrund und gelber Schrift zu erhalten benutzen wir die folgende Zeile:

```
xterm -bg blue -fg yellow &
```

Konfiguration von X-Clients über Resources

Die Steuerung der Programmeigenschaften über Kommandozeilenparameter ist zwar eine mächtige Möglichkeit, jedoch kann es auch ganz schön lange Befehlszeilen geben, wenn wir etwa ein xterm an einer bestimmten Position in einer bestimmten Größe und Farbe mit einer anderen Schrift haben wollen. Dadurch entstünden Befehlszeilen wie etwa:

```
xterm -geometry 90x30-10-10 \
>   -fn -misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1\
>   -fg blue -bg bisque2 &
```

Zugegebenerweise wäre das etwas umständlich zu tippen. Um oft gebrauchte Einstellungen fest zu verändern steht uns das Resource-System zur Verfügung. Resources sind einfach nur Variablen, die ein Programm benutzt und die von außerhalb des Programms verändert werden können.

Die Grundeinstellungen dieser Resourcen stehen im Verzeichnis `/etc/X11/app-defaults`. Jeder User kann aber diese Werte verändern, so daß für ihn (und nur für ihn) Programme mit anderen Werten gestartet werden. Das geschieht in der Regel über die Datei `.Xresources` oder `.Xdefaults` im jeweiligen Home Verzeichnis des Users. Dort können Zeilen wie die folgenden eingetragen werden:

```
XTerm*font: -misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1
XTerm*Background: bisque2
XTerm*Foreground: blue
XTerm*geometry: 90x40
```

Diese Werte werden dann beim nächsten Start des X-Servers in den Speicher geladen und aktiviert. Jedesmal, wenn Sie jetzt xterm ohne Parameter aufrufen erscheint ein Fenster mit Hintergrundfarbe bisque2, Vordergrundfarbe blau, fetter Schrift und der Größe 90x40...

Um die Datei während des Betriebs zu laden (ohne den Server neu zu starten) kann der Befehl

```
xrdb -merge .Xresources
```

angegeben werden. Der Parameter -merge sorgt dafür, daß die bestehenden Resourcendefinitionen, die durch die Veränderungen nicht betroffen wurden erhalten bleiben.

Welche Resourcen ein Client versteht kann in der Regel in seiner Handbuchseite nachgelesen werden.

Die grundsätzliche Form der Eingabe ist immer die gleiche:

*Client (oder Clientname)*Resource: Wert*

In der Regel haben Clients sehr viele Resourcen, wesentlich mehr als mögliche Kommandozeilenparameter. So kann einem Xterm beispielsweise durch die Resource

```
XTerm*scrollbar:true
```

eine Scrollbar (Rollbalken) zugefügt werden. Damit alle diese Resourcen auch über die Kommandozeile möglich sind, gibt es den Parameter `-xrm`, dem ein Resourcenstring folgen kann. Damit ist es also auch möglich, Programmeinstellungen über die Kommandozeile zu machen, für die es zwar Resourcen gibt aber keine Parameter.

Starteinstellungen

Wenn der X-Server über den Befehl **startx** gestartet wird, so wird beim Start automatisch die Datei `~/.xinitrc` oder - falls sie nicht vorhanden ist, die Datei `/etc/X11/xinit/xinitrc` abgearbeitet. Dabei handelt es sich um ein einfaches Shellscript, das verschiedene Aufgaben übernimmt:

- Die systemweite und die userbezogene Tastaturdefinition wird geladen (`/etc/X11/Xmodmap, ~/.Xmodmap`).
- Die systemweiten und die userbezogenen Resource-Dateien werden geladen (`~.Xdefaults, ~.Xresources, /etc/X11/Xresources`)
- Beliebige X-Clients können gestartet werden
- Der gewünschte Window-Manager wird gestartet

Einen ähnlichen Vorgang gibt es, wenn der X-Server vom Display-Manager gestartet wird. In diesem Fall wird die Datei `/etc/X11/xdm/Xsession` abgearbeitet.

Wenn im Homeverzeichnis eines Users ebenfalls eine Datei `~/.Xsession` existiert, so wird auch diese Datei abgearbeitet. So kann jeder User seine Einstellungen persönlich verändern.

In vielen Distributionen rufen sowohl xinitrc, als auch Xsession einfach nur ein anderes Script auf, um zu einer gemeinsamen Konfiguration zu gelangen.

X11 im Netz

Da das X-Protokoll ein Netzwerkprotokoll ist, kann jeder X-Client auch entsprechend so aufgerufen werden, daß er seine Ausgaben nicht auf dem lokalen Server macht, sondern auf einem anderen X-Server. Dazu gibt es verschiedene Mechanismen, die sowohl den Aufruf des Clients, als auch die Einstellungen auf dem Server betreffen.

Die Client-Seite

Jeder X-Client kann mit einem Parameter `-display` gestartet werden, mit dem angegeben werden kann, auf welchem Display der Client dargestellt werden möchte. Die Angabe des gewünschten Displays hat die Form:

Hostname:Servernummer.Displaynummer

Der Hostname kann weggelassen werden, wenn das Programm auf dem lokalen Server zugreifen soll. Servernummer und

Displaynummer beziehen sich auf die einzelnen Server und Displays des jeweiligen Hosts. Das ist nur von Bedeutung, wenn auf einem Rechner mehrere Server laufen (etwa wenn mehrere XTerminals angeschlossen sind) - im Normalfall steht hier ein `:0.0` also der Server 0 und das Display 0. In den meisten Fällen kann auch die Angabe der Displaynummer weggelassen werden (Displaynummern fallen nur an, wenn ein Server mehrere Graphikkarten anspricht) so daß einfach ein :0 als Serverbezeichnung reicht.

Damit es unter X11 möglich ist, lokale X-Clients ohne die Nennung eines Displays zu starten, wird beim Starten von X eine Umgebungsvariable `DISPLAY` definiert, die den Wert :0.0 beinhaltet. Dadurch wissen XClients auf welchen XServer sie zugreifen sollen, wenn keiner explizit angegeben wurde.

Ein Beispiel:

Wir sind im Rechner marvin eingeloggt und wollen hier das Programm netscape starten. Die Ausgaben dieses Programmes sollen aber auf dem XServer des Rechners hal ausgegeben werden. Wir schreiben also:

```
netscape -display hal:0.0
```

Natürlich kann auch hier wieder statt dem Rechnernamen die IP-Adresse stehen.

Die Server-Seite

Ein X-Server, auf dem Ausgaben von Clients anderer Hosts zugelassen werden sollen, muß natürlich zunächst diesen Hosts erlauben, auf seinen Dienst zuzugreifen. Dazu existiert eine Hostbasierte Zugriffskontrolle.

Das Programm, mit dem diese Kontrolle realisiert wird heisst **xhost** und ist sehr einfach anzuwenden:

```
xhost [+|-]
xhost Hostname
xhost -Hostname
```

Wenn **xhost** ohne Argumente aufgerufen wird, so zeigt es den Status der Zugriffskontrolle (an- oder ausgeschaltet) und die Liste der authorisierten Rechner an.

Durch den Aufruf von `xhost +` wird alle Zugriffskontrolle ausgeschaltet, jeder Rechner kann jetzt auf den Server zugreifen.

Mit `xhost -` werden die Zugriffskontrollmechanismen wieder aktiviert, nur authorisierte Rechner dürfen auf den Server zugreifen.

Durch die Nennung eines Rechnernamens nach **xhost** (also etwa `xhost hal`) wird der angegebene Rechner in die Liste der authorisierten Rechner aufgenommen. Programme auf diesem Rechner können also jetzt ihre Ausgaben auf dem Server machen.

Die Zeile `xhost -hal` würde den Rechner `hal` wieder von der Liste der authorisierten Rechner streichen.

Selbstverständlich können diese Einstellungen auch beim Start von X11 automatisch ablaufen. Es ist nur nötig, sie in die jeweils verwendete xinitrc Datei zu schreiben. Also entweder in die systemweite in `/etc/X11/xinit/xinitrc` oder in die private `$HOME/.xinitrc`

Eleganter ist allerdings die Möglichkeit, eine Datei namens `/etc/X`*n*`.hosts` zu verwenden, die eine Liste aller Rechner enthält, die Zugriff auf den Server haben sollen. Das *n* im Dateinamen steht für die Servernummer, in unserem Fall also immer die 0. Diese Datei ist nicht standardmäßig vorhanden, sie muß bei Bedarf vom Systemverwalter angelegt werden.

Auflösung von Bibliotheksabhängigkeiten

Zur Prüfung und Auflösung von Library-Abhängigkeiten wurde im Abschnitt 1.102.4 - Verwaltung von Shared Libraries schon alles notwendige erläutert. Es gibt keinen Unterschied zwischen X11-Anwendungen und normalen Linux-Programmen hinsichtlich der Libraries.