



by Guido Socher ([homepage](#))

Mittels X11 Anwendungen auf anderen Rechnern ausführen



Abstract:

About the author:

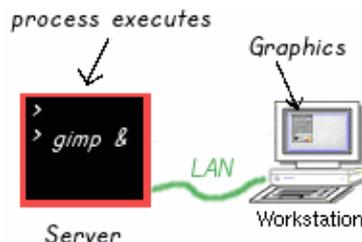
Guido mag Linux nicht nur, weil es interessant ist, zu verstehen, wie Betriebssysteme arbeiten, sondern auch wegen der Menschen, die an der Entwicklung von Linux beteiligt sind.

Viele Linux-Erstbenutzer/innen denken, dass die grafische Oberfläche unter Linux nur ein weiteres "Windows"-System ist, in dem man Anwendungen startet, die dann in eigenen Fenstern erscheinen. Einigen fällt auf, dass man verschiedene Oberflächen benutzen kann, aber das scheint es auch schon gewesen zu sein. Das Linux X-Window-System (X11) ist viel mehr als das! Es ist ein Netzwerk-fähiges Window-System. Wir werden sehen, welche neuen und mächtigen Möglichkeiten es bietet.

Das Display-Konzept

Jede grafische X-Window-Anwendung liest beim Start die Umgebungsvariable DISPLAY aus, um festzustellen, an welchen Computerbildschirm die Grafik-Daten geschickt werden sollen. Dies erlaubt es in Verbindung mit den Netzwerkfähigkeiten des X-Window-Systems, grafische Anwendungen auf entfernten Rechnern zu starten. Das heißt, Sie nutzen die CPU-Leistung einer Maschine, während Sie die Anwendung von einer anderen Maschine aus steuern. Das vollständige GUI (grafische Benutzungsoberfläche) wird auf der Maschine dargestellt, von der aus Sie die Anwendung benutzen. Sie bemerken nicht, dass Sie 2 Computer benutzen.

Die Netzwerkgeschwindigkeit hat natürlich einen Einfluß, aber eine normale 10 Mbit/s LAN-Verbindung ist mehr als genug.



Warum sollten Sie das tun?

Es gibt viele Anwendungen dieser

"Netzwerk-Grafikprogramme". Firmen benutzen sie, um Geräte fernzusteuern, die Tausende von Kilometern entfernt sind und Sie können die gleiche Anwendung zur Steuerung benutzen, als ob Sie direkt vor Ort wären.

Vielleicht haben Sie 2 Computer, eine schnelle 1 Ghz-Maschine und einen alten Pentium 133 Mhz. Sie können sich an der Geschwindigkeit Ihres neuen Rechners erfreuen, obwohl Sie nicht direkt davor sitzen. Vielleicht sitzt gerade Ihre Schwester vor dem schnellen Rechner und ist angemeldet. Das macht nichts, Sie profitieren trotzdem davon.

Wie funktioniert das?

Alle X-Window-Anwendungen, mögen sie nun gimp, xterm, konqueror, netscape ... heißen, sind eigentlich Netzwerk-Clients, die sich mit einem Server, dem X-Server, verbinden. Aufgabe des X-Servers ist es, mit der Grafik-Hardware zu sprechen, die Bilder auf Ihrem Monitor zu zeichnen sowie die Maus- und Tastatur-Eingaben zu lesen. Die Clients (Ihre Programme wie gimp, netscape...) senden dem Server Anweisungen, wie er die Rahmen und Buttons zu zeichnen hat. Im Austausch erhalten sie vom Server die Maus- und Tastatur-Ereignisse übermittelt. Offensichtlich benötigen Sie eine Authentifizierung, weil sonst jedermann den Bildschirm von jedem anderen durcheinander bringen könnte. Es gibt zwei Programme zur Zugriffskontrolle:

- xhost: mittels dieses Programms können Sie jeder Benutzerin auf einer beliebigen Maschine erlauben, Grafikdaten auf Ihrem Monitor anzuzeigen. Beispiel: Sie sitzen vor einem Rechner namens philosophus. Um jedem Programm auf dem Rechner movietux Zugriff auf Ihr Display auf philosophus zu gestatten, würden Sie folgenden Befehl eingeben:

```
xhost +movietux
```

Dies muss in einer Shell auf philosophus eingegeben werden.

- xauth: Dies ist eine Cookie-basierende Authentifizierung und sehr viel mächtiger. Hier können Sie wirklich einzelnen Personen Zugriff gestatten. Es ist viel sicherer als xhost. Die Authentifizierung benutzt ein Cookie, das in der Datei .Xauthority im Startverzeichnis des Benutzers gespeichert wird. Wenn der entfernte Rechner ein gültiges Cookie in dieser Datei hat, wird Zugriff gewährt. Um das Cookie von der Maschine, vor der Sie sitzen (philosophus), auf den Rechner zu kopieren, auf dem Sie das Programm starten wollen (movietux), können Sie eine der beiden folgenden Methoden benutzen:

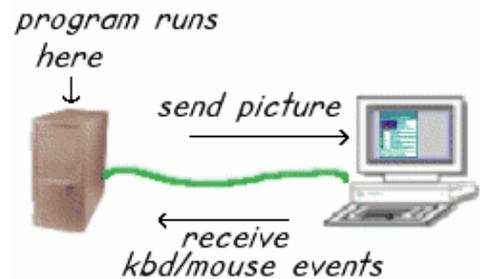
```
xauth extract - philosophus:0.0 | ssh movietux /usr/X11R6/bin/xauth merge
```

oder

```
scp ~/.Xauthority movietux:
```

Die Prozedur, die beim Start eines Programms (Client) abläuft, ist folgende:

1. Der Client überprüft die DISPLAY-Umgebungsvariable, um den Server zu finden, andernfalls wird versucht, den Server auf diesem Rechner zu erreichen.
2. Der Server überprüft, ob der Client berechtigt ist, "Bilder" an ihn zu senden. Wenn der Client autorisiert ist, dann zeichnet der Server das Bild auf den Monitor.



Die DISPLAY-Umgebungsvariable hat folgende Syntax:

```
bash:export DISPLAY=hostname:displaynumber.screennumber
tsh:setenv DISPLAY hostname:displaynumber.screennumber
```

Ich werde hier nicht über die Display- und Bildschirm-Nummer sprechen. Dies ist normalerweise einfach 0:0. Ein Beispiel für die bash-Shell wäre:

```
export DISPLAY=philosophus:0.0
```

Eine Anwendung auf einem entfernten Rechner starten

Nach der ganzen Theorie nun ein praktisches Beispiel. Versuchen Sie es einfach. Wir sitzen wieder an dem Computer philosophus und möchten etwas von dem Rechner movietux starten.

1. Erzählen Sie Ihrem X-Server, dass Clients von movietux berechtigt sind, Bilder auf Ihrem Bildschirm zu zeichnen:
xhost +movietux
2. Melden Sie sich am Rechner movietux an:
slogin -l username movietux
3. Nun sind Sie am Rechner movietux angemeldet und jeder Befehl, den Sie starten, wird auf movietux ausgeführt
4. export DISPLAY=philosophus:0.0
5. Starten Sie das Programm, z. B.
netscape &

Falls movietux eine mächtige Maschine ist, werden Sie bemerken, dass die Wiedergabe von html-Seiten in Netscape um einiges schneller ist als wenn Sie Netscape auf Ihrem lokalen Rechner starten würden. Zur gleichen Zeit werden Sie unter Benutzungsgesichtspunkten kaum bemerken, dass dieser Netscape nicht lokal gestartet wurde, weil Sie ihn in der gleichen Weise mit Maus und Tastatur steuern. Schöne Sache, nicht wahr?

Wie Sie Ihr Display mitnehmen

Obwohl es sehr einfach ist, eine Anwendung entfernt auszuführen, ist immer noch eine zusätzliche Eingabe erforderlich, um die DISPLAY-Variable zu setzen. Es ist möglich, dies zu automatisieren:

- Wenn Sie ssh zur Anmeldung benutzen, dann wird die DISPLAY-Variable automatisch richtig gesetzt. Es gibt noch andere Login-Programme, die die DISPLAY-Variable unterstützen, aber ssh ist sehr verbreitet.
- Wenn Sie slogin oder andere Befehle benutzen, dann können Sie die folgenden Skripte auf dem entfernten Rechner installieren:

Für tsh:

```
# take your display with you at remote login:
# Put it into your ~/.login file
set whoami=`who -ml`
set remhost=`expr "$whoami" : '.*(\(.*\))'`
if ( "$remhost" != "" ) then
```

```
setenv DISPLAY "$remhost":0.0
endif
```

Das Skript holt sich den Namen des fernen Rechners aus dem Befehl "who -ml". Dieser Befehl gibt etwa folgendes zurück

```
>who -ml
movietux!guido pts/3 Oct 26 21:55 (philosophus.tux.org)
```

Wenn Sie die bash benutzen, benötigen Sie das folgende Skript:

```
# take your display with you at remote login:
# Put it into your ~/.bash_profile
whoami=`who -ml`
remhost=`expr "$whoami" : '.*(\.*)"`
if [ -n "$remhost" ]; then
DISPLAY="$remhost":0.0
export DISPLAY
fi
```

OpenGL

Obwohl die Netzwerkfähigkeiten des X-Window-Systems recht gut sind, ist die Grafik etwas langsamer aufgrund der Tatsache, dass Sie die Daten über ein Netzwerk-Protokoll senden. Im Normalfall werden Sie jedoch keinen großen Unterschied bemerken.

Grafik-intensive und schnelle Anwendungen wie grafisch ansprechende Spiele basieren normalerweise auf OpenGL (Open Graphics Library) und GLX (OpenGL Extension to the X Window System). Diese Bibliotheken stellen eine Hardware-unabhängige Programmierschnittstelle bereit, die direkten Zugriff auf 3D-Hardware-Beschleunigung in der Grafik-Karte bietet. Das heißt: Die Anwendung sendet die Beschreibung eines Objektes in der Form von Punkten, Linien und Polygonen an die Grafik-Karte und alle Berechnungen werden innerhalb der Grafik-Hardware erledigt. Dies sorgt für sehr schnelle Grafik-Darstellung.

Derzeit unterstützen die meisten Grafik-Karten-Treiber (X-Server) unter Linux keine GLX/OpenGL-Hardware-Beschleuniger für entfernt laufende Anwendungen. Sie unterstützen Hardware-Beschleunigung für lokale Anwendungen. Der Effekt ist, dass entfernt gestartete Anwendungen gar nicht zu starten scheinen und wirklich sehr langsam sind. Eine Ausnahme sind die NVidia-Treiber, von denen der Quellcode aber nicht veröffentlicht wird. Sie haben ein "Direct-Rendering-Interface", welches indirektes Rendering für entfernte Anwendungen unterstützt.

Schlussfolgerung

Die Nutzung der Rechenkapazität Ihres Netzwerkes ist dank X11 sehr einfach. Sie können mit entfernten Anwendungen in der gleichen Weise arbeiten wie mit lokalen. Der einzige Unterschied, dem Sie begegnen, ist, dass Sie die Dateien und das Startverzeichnis des entfernten Rechners sehen. Mittels installiertem NFS und NIS können Sie sogar diesen kleinen Unterschied verbergen und die CPU-Kapazität der schnellsten Maschinen in Ihren Netzwerken nutzen, ohne darüber nachzudenken..

Links

[X11 system, x.org](http://X11.system.x.org)

xfree86.org das von Linux benutzte X11-System

<p><u>Webpages maintained by the LinuxFocus Editor team</u></p>	
---	--

© Guido Socher

"some rights reserved" see linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher ([homepage](#))

en --> de: Hermann J. Beckers <beckerst@lst-online.de>

2005-01-11, generated by lfparsr_pdf version 2.51