

SelfLinux-0.10.0



Was ist X?

Autor: Erwin Dogs (*edogs@t-online.de*)

Autor: Florian Fredegar Haftmann (*florian.haftmann@stud.tum.de*)

Formatierung: Alexander Fischer (*Selflinux@tbanus.org*)

Lizenz: GFDL

Inhaltsverzeichnis

1 Was ist X?

2 Was leistet X und wo liegen seine Grenzen?

3 Was ist ein Desktop Environment?

1 Was ist X?

Wenn man mit Linux arbeitet und sich damit beschäftigt, wird man früher oder später über den Buchstaben **X** stolpern - kurz für **X Window System**; dabei handelt es sich um das Kernstück der grafischen Oberfläche.

Linux **an sich**, der Kernel, weiß nichts von Grafik; im Gegensatz zu weitverbreiteten anderen Betriebssystemen, welche die grafische Oberfläche fest in ihrem Kern integriert haben, ist das bei UNIXen nicht der Fall. Vielmehr läuft das *X Window System* völlig unabhängig vom Kernel, es ist sogar völlig unabhängig davon entstanden:

Anfang der 80er-Jahre kam man am  [Massachusetts Institute of Technology](#) (MIT) darauf, dass es ja ein bisschen schade ist, wenn man ein echtes Mehrbenutzer- und Multitasking-System hat, darauf aber nur mit einer Konsole zugreifen kann. Die naheliegende Idee:

Es sollten mehrere Anwendungen in unterschiedlichen, bewegbaren Fenstern auf dem gleichen Schirm angezeigt werden können. So entstand **X** als akademisches Projekt; der Quellcode wurde von vielen kommerziellen UNIX-Anbietern aufgegriffen, fortentwickelt und in ihre UNIX-Varianten integriert. In den 90ern startete das  [XFree-Projekt](#) mit dem Ziel, den damaligen Quellcode auf freier Basis auf den neuesten Stand zu bringen und so **X** für Linux verfügbar zu machen - erfolgreich.

2 Was leistet X und wo liegen seine Grenzen?

Entsprechend der UNIX-Philosophie ist **X** keine **eierlegende Wollmilchsau** ohne eine durchschaubare innere Struktur, sondern in einzelne Komponenten aufgegliedert:

- * Der **X-Server**; er ist das Programm, das Tastatur- und Mauseingaben entgegennimmt und die Resultate auf dem Bildschirm anzeigt. Er enthält dafür die passenden Gerätetreiber. Die Eingaben werden von ihm nicht ausgewertet, sondern an die betreffenden **X-Clients** (s. u.) weitergeleitet, die wiederum die Resultate an den **X-Server** zurückliefern.
- * Die **X-Clients**; sie sind praktisch die Anwendungen (Textverarbeitung, Editor, Browser, ...), welche die grafische Oberfläche benutzen wollen. Vom **X-Server** bekommen sie die Tastatur- und Mauseingaben mitgeteilt, die sie betreffen, und melden ihm zurück, was auf dem Bildschirm erscheinen soll. Dazu wird das so genannte X-Protokoll verwendet.
- * Der **Windowmanager**; er kümmert sich um das **Drumherum**, z.B. Fensterrahmen und Fenstermenüs für die Anzeigen der **X-Clients**, Menüs auf dem Desktop, Minimieren und Maximieren der Fenster usw.

Das Zusammenspiel lässt sich in folgender Grafik zusammenfassen (beachten Sie dabei, dass auch ein Window-Manager nichts anderes als ein X-Client ist):

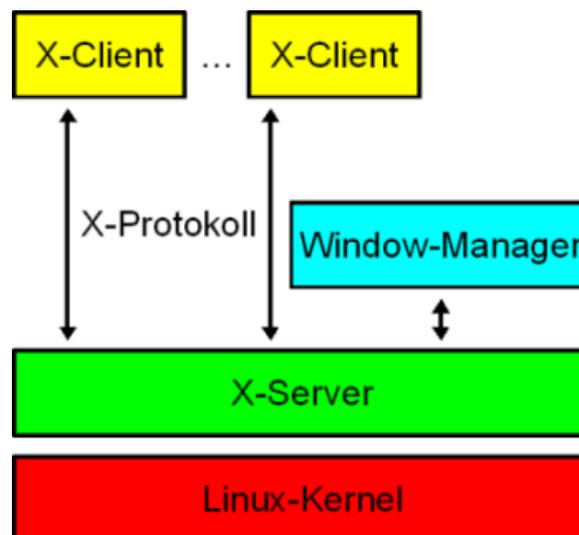


Diagramm der Architektur

Diese Architektur hat eine ganze Reihe von Vorteilen:

- * Auf Systemen, die keine grafische Oberfläche brauchen (z.B. Webserver), kann sie einfach weggelassen werden, das sie ja vom Kernel unabhängig ist; das spart Speicher- und Prozessorressourcen.
- * Das grafische System kann beliebig herauf- und heruntergefahren werden (z.B. zu Konfigurationszwecken), ohne das das System verändert oder angehalten werden muss.
- * Die Textkonsolen arbeiten unabhängig von der grafischen Oberfläche; mit `Strg + AltF1` bis `F6` kann

man auf sie umschalten. Das ist z.B. sinnvoll, falls man schnell an der Systemkonfiguration was ändern will und sich somit auf einer Konsole als root einloggt. Zum X-Bildschirm zurück geht es mit `Strg + Alt + F7`.

- * Das X-Protokoll setzt auf dem [Internet-Protokoll TCP/IP](#) auf; das bedeutet, dass Clients und Server auf unterschiedlichen Rechnern laufen können(!). In einem lokalen Netzwerk, in dem z.B. eine rechenintensive Ingenieurberechnungssoftware nur von einem einzigen Rechner verkraftet werden kann, weil alle anderen zu wenig Hauptspeicher haben, kann man sich einfach von seinem Arbeitsplatz aus dort einloggen, das Programm starten, und es erscheint auf dem eigenen Bildschirm, ohne dass man merkt, dass es auf einem ganz anderen Rechner läuft.
- * Die einzelnen **X-Clients** müssen **nur** das X-Protokoll beherrschen, die Hardware kann ihnen relativ egal sein; das ist zwar heute allgemein so, aber zur Zeit der Einführung von **X** war diese Idee sehr fortschrittlich.
- * Der Windowmanager ist beliebig wählbar; das macht das Erscheinungsbild individuell konfigurierbar.

X ist ein barocker Dinosaurier in jeder Linux-Distribution: barock, weil es sehr viele Features und Tools liefert, die man eigentlich (außer zum Rumspielen und sich darüber Amüsieren) nicht braucht; Dinosaurier, weil auf der anderen Seite seine interne Struktur auf den technischen Möglichkeiten aufbaut, die vor zwanzig Jahren bestanden:

- * Das X-Protokoll kennt nur einfache Anweisungen wie: **Zeichne Linie von A nach B**; moderne Grafikkarten haben viele Möglichkeiten, um den Bildschirmaufbau durch die Hardware zu beschleunigen, diese können aufgrund dieser **Kleinkariertheit** von **X** nicht genutzt werden und liegen brach. Solange **X** nicht durch ein neues Konzept abgelöst wird, wird sich daran auch nichts ändern.
- * Die Programmierung von **X** ist damit auch recht umständlich; mittlerweile gibt es allerdings so genannte GUI-Toolkits, die auf **X** aufsetzen und dem Programmierer das Leben recht einfach machen. Als **Altlast** gibt es aber viele alte Programme für **X**, von denen jedes eine andere Bedienphilosophie verfolgt und die einen kunterbunt aussehenden Zoo bilden.
- * **X** hat absolut nichts mit dem Drucksystem zu tun; das ist zwar nicht unbedingt eine Designschwäche, hat aber zur Folge, dass es zwei paar Stiefel sind, z.B. eine Schriftart am Bildschirm anzeigen zu können oder sie auch ausdrucken zu können.
- * **X** ist nur ein Verfahren, um mehrere (Text-)fenster anzuzeigen; von einer grafischen Oberfläche wird aber mehr verlangt. Den Anwender braucht dies aber wenig zu kümmern, denn im Zusammenspiel mit den modernen [Desktop Environments](#) von Linux liefert **X** eine Benutzeroberfläche, die kaum Wünsche offen lässt.

3 Was ist ein Desktop Environment?

Was macht ein modernes GUI (*Graphical User Interface*) aus? Funktionen wie Copy & Paste, Kontextmenüs, einheitliches Look & Feel, Drag & Drop, Desktop Panel - alles Dinge, die **X** selbst nicht bereitstellt. Dazu wurden die Desktop Environments oder Desktopumgebungen ins Leben gerufen. Desktopumgebungen bringen darüberhinaus eine ganze Ansammlung nützlicher Anwendungssoftware bereits mit (z.B. Web-Browser, Office-Paket, Spiele, Editor...), und bilden somit ein ganzes Software-Bundle. Zudem verwenden sie einheitliche Grafikbibliotheken (GUI-Toolkits), mit denen diese Anwendungen programmiert sind, was einerseits das Aussehen für den Anwender gefällig macht, andererseits dem Programmierer eine Menge Arbeit abnimmt. Eine Desktopumgebung ist der Schlüssel zum Erfolg, um ein Betriebssystem einer breiten Masse an Anwendern zur Verfügung zu stellen, da kaum ein Computeranwender gerne auf Komfort bei der Benutzung seines Computers verzichten will. Er möchte sich im Wesentlichen auf den Kern seiner Bedürfnisse konzentrieren (z.B. Briefe schreiben, im Internet surfen, Mails senden und empfangen). Dies mit einem Mausklick schnell und bequem erledigen zu können, das Kopieren von Textpassagen von einer Anwendung in eine andere, Icons auf der Oberfläche zum vereinfachten Start häufig verwendeter Anwendungen - all dies und noch viel mehr gehört zu den Funktionen, welche ein Desktopsystem bereitstellt. Die beiden wohl verbreitetsten Desktopsysteme sind

- *  [KDE](#) (Kool Desktop Environment), basierend auf dem GUI-Toolkit QT
- *  [Gnome](#) (GNU Network Object Model Environment), basierend auf dem GUI-Toolkit GTK

In unseren Breiten ist durch die starke Präsenz eines Nürnberger Linux-Distributors KDE das am stärksten verbreitete Desktopsystem. Es ist so bekannt, dass KDE oft mit Linux gleichgesetzt wird.

Beide Desktop Environments werden aber mit jeder größeren Distribution mitgeliefert und gleichen sich sehr im Funktionsumfang.

Welches ist die am besten geeignete Oberfläche für mich? - Diese Frage wird häufig gestellt, und dem Einsteiger würden wir erstmal dazu raten, sich mit der vorinstallierten Oberfläche vertraut zu machen. Ansonsten gilt der Grundsatz: ausprobieren! Die Geschmäcker sind verschieden, und es gibt genug Alternativen, um die Arbeitsumgebung an die persönlichen Wünsche und Bedürfnisse des Anwenders anzupassen. Neben KDE und Gnome gibt es alternativ auch noch die Möglichkeit, einen **klassischen** Windowmanager wie FVWM zu verwenden - diese bieten zwar geringeren Komfort, schonen aber die Ressourcen, was z. B. auf einem Rechner, der überwiegend als Server eingesetzt wird, sinnvoll sein kann.