

# XFree86 - Hinter den Kulissen ...

[Jana Jaeger](#)

Ausser einigen wenigen Puristen, die für die Arbeit mit ihrem Linuxsystem ausschließlich mit der Textkonsole auskommen, dürfte mittlerweile jeder Linuxuser X (oder besser: XFree86) schon bei der Arbeit gesehen haben. Auch das Browserfenster, in dem dieser Text angezeigt wird, nutzt die Routinen des X-Servers...

Dirk Hohndel, [XFree86](#)-Entwickler, Vize-Präsident von The XFree86 Project, Inc. und - nicht zu vergessen - CTO der [SuSE Linux AG](#), stellte sich einigen Fragen der Portalredaktion zu Hintergründen und Besonderheiten von XFree86.

**Web Portal : Anders als viele andere Open Source Projekte steht XFree86 nicht unter der GPL. Was sind die Unterschiede zwischen der GPL und der XFree86 Lizenz? Warum eine eigene Lizenz und nicht die GPL? Macht die Entscheidung für den Sonderweg rückblickend Sinn?**

**Hohndel :** XFree86 steht unter einer Lizenz, die der MIT / X Consortium Lizenz des X Window Systems entspricht. Kern dieser Lizenz ist es, dass jeder die Software unbeschränkt benutzen, verändern und weiterverbreiten kann. Hierin besteht der wesentliche Unterschied zur GPL: während dort jede abgeleitete Arbeit ("derived work") automatisch wieder unter die GPL fallen muss, und daher auch mit samt Sourcen frei verfügbar sein muss, ist dies bei der XFree86 Lizenz explizit nicht der Fall.

Der historische Grund für diese Lizenz an sich und für die Übernahme dieser Lizenz für XFree86 ist recht einsichtig. Am MIT und natürlich besonders im X Consortium haben Firmen mit kommerziellen Unix Versionen sich an der Entwicklung des X Window Systems beteiligt. Diese Firmen wollten natürlich ihre kommerziellen Produkte auf den X Sourcen aufbauen können, ohne dabei die Sourcen dieser Produkte offenlegen zu müssen. Oder, noch trivialer gesprochen, diese Firmen wollten einen Standard etablieren, für die die Sample Implementation einfach nur ein einfacher Einstiegspunkt war, und jeder sollte basierend auf dieser Sample Implementation arbeiten können, ohne wesentliche Einschränkungen (es gibt natürlich die üblichen Disclaimers zum Thema "zugesicherte Eigenschaften" sowie zur Werbung mit der Tatsache, dass dieser Code verwendet wird).

XFree86 als Projekt, das ein Teil der "X Community" war, hat sich ganz selbstverständlich dieser Lizenz angeschlossen. Man muss dazu verstehen, dass X in der Tat wohl das erste wirklich global erfolgreiche Open Source Projekt war, lange bevor die GNU Idee weiter verbreitet wurde. Aus unserer Sicht macht die Lizenz extrem viel Sinn, da ja auch wir möglichst vielen Leuten und Firmen erlauben wollen, unseren Code als Basis für Entwicklungen und Projekte zu nehmen, egal, unter welcher Lizenz diese Projekte dann stehen.

Viele Leute behaupten immer, eine Lizenz wie die unsrige würde automatisch zu einer Aufsplitterung des Projekts und zu vielen Versuchen eines "taking closed source" führen, das heißt zu Versuchen, basierend auf dem Open Source Projekt ein Closed Source Projekt zu bauen, das einige weitere Features bietet, und damit dem Open Source Projekt die "Kunden" wegnimmt. Dieser Gedanke zeigt aus meiner Sicht nur mangelndes Vertrauen in die entscheidenden Vorteile des Open Source Konzepts. Denn niemand hätte einen echten Vorteil davon, dies zu tun, denn er würde sich damit ja nur von der ständigen Weiterentwicklung von XFree86 abkoppeln.

Und interessanterweise kam es dann genau andersrum. Aufgrund unserer Lizenz hat einer der beiden kommerziellen Anbieter im Bereich X Server für PC Unix Systeme, die Firma Metro Link, ihr eigenes Produkt umgestellt, so dass es auf den XFree86 Sourcen basiert (und im Zuge dessen

einiges an Technologie in XFree86 eingebracht). Es kam also nicht zu einer Aufsplitterung, sondern viel mehr zu einer Zusammenfassung verschiedener Richtungen auf eine Source Basis.

Nach nunmehr bald neun Jahren XFree86 ist die Antwort also ganz eindeutig. Diese Lizenz macht für XFree86 absolut Sinn, die Entscheidung war eindeutig richtig.

**Web Portal : Sind dem X-Server Grenzen gesetzt? Gibt es eine obere Grenze, ab der die Performance nicht mehr weiter zu steigern ist? Ist der X-Server schon zu komplex, um z.B. auf Embedded Systemen zu laufen? Macht er auf der anderen Seite riesig aufwendige 3D-Anwendungen mit?**

**Hohndel :** Viele viele Fragen auf einmal. Also, der Reihe nach:

Ich sehe sicherlich keinen Punkt und keinen Grund in der Architektur von X oder XFree86, wo es eine obere Grenze der Performance gäbe. Es stellt sich natürlich immer die Frage, wie viel 2D Performance braucht der mit X arbeitende Mensch (und die Antwort ist für 99% der Anwendungsfälle "weniger, als heute schon in preiswerten Grafikkarten zur Verfügung steht"). Aber im 2D Bereich (und X ist eindeutig ein 2D Protokoll) sehe ich da keine echten Grenzen. Da stellen sich schon eher Fragen zum Thema des Rendering-Modells, also dazu, welche Möglichkeiten X Anwendungen bietet, um zweidimensionale Objekte darstellen zu können. Hier hat das alte RasterOps Modell von X sicher seine Grenzen erreicht, weswegen wir ja in den aktuellen XFree86 Versionen ein neues Rendering-Modell, basierend auf Image-Composition eingeführt haben, dass X einfach zeitgemäß erweitert, ohne die Kompatibilität mit alten Applikationen aufgeben zu müssen.

Und gerade letzte Woche hat Troll Tech, basierend auf diesem neuen Rendering-Modell eine neue Version der Qt-Libraries freigegeben, die Anti-Aliased-Fonts beinhaltet, also Zeichensätze, bei denen die durch die Rasterung bedingten "Treppeneffekte" abgemildert sind.

Die Frage nach der Komplexität von X ist natürlich berechtigt und sie wird auch immer wieder mal gestellt. Aus meiner Sicht bietet X aber unglaubliche Vorteile gegenüber den anderen aktuell in Erwägung gezogenen Grafik Infrastrukturen. Die Standardisierung, die Netzwerk Transparenz, die umfassende Kompatibilität zu existierenden Applikationen, all das macht X zu einer idealen Umgebung auch und gerade im Embedded Bereich. Und es ist ohne weiteres möglich, einen kompletten X Server samt einiger Zeichensätze in nur 800kB Speicher laufen zu lassen. tinyX ist da ein schönes Beispiel für, und natürlich läuft X schon heute auf den Vorzeige-Devices, also zum Beispiel auf Compaqs iPAQ oder auf IBMs berühmter Linux Uhr.

3D ist ein völlig anderes Thema, denn X ist wie gesagt eine 2D Infrastruktur. 3D unter Linux heißt zumeist OpenGL, und der X Server kommt hierbei nur ins Spiel, wenn mittels GLX OpenGL durch das X Protokoll hindurch "getunnelt" wird, so dass OpenGL auch über das Netzwerk benutzt werden kann (natürlich mit gewissen Nachteilen in der Performance), und natürlich dann, wenn X der OpenGL Rendering-Engine "aus dem Weg gehen muss", denn es kann immer nur eine Kontrollinstanz über den Grafik-Prozessor geben. Da kommt im Zusammenhang mit XFree86 natürlich die Direct Rendering Infrastructure DRI zum Zug, die genau diese Kooperation zwischen den OpenGL Treibern und den X Server sicherstellt.

Die Qualität und Performance der OpenGL Implementation unter XFree86 ist danach dann auch nur wieder eine Frage der Qualität der OpenGL Treiber für die entsprechende Grafik Hardware.

**Web Portal : Mit XFree86 4.0 kam ein grundlegend neues Konzept. Jetzt gibt es einen X-Server für alle Kartentypen, der je nach Modell um die entsprechenden Treibermodule erweitert wird. Hardwarehersteller können jetzt selbst solche Module einreichen. Wie kommt das bei den Herstellern an? Wie läuft die Zusammenarbeit? Und vor allem: wie ist die Qualität der abgegebenen Module?**

**Hohndel :** Die Idee war richtig, es hat bekanntlich zu lange gedauert, XFree86 4.0 fertig zu

stellen, aber was lange währt wird endlich gut (oder so) :-). Im Ernst, die Hersteller haben recht schnell das Prinzip verstanden, sind allerdings zunächst unterschiedlich damit umgegangen. Während manche von sich aus Open Source Treiber als Module geschrieben haben, haben andere lieber Closed Source Treiber (von zum Teil zweifelhafter Qualität) herausgebracht.

Deutlich überlagert wurde das alles allerdings vom brutalen Kampf in diesem Segment, dem, wie alle wissen, viele Hersteller zum Opfer gefallen sind, und in dem die übriggebliebenen Hersteller sich immer wieder neu zum Thema Linux und XFree86 positionieren mussten. Die großen Hersteller, die noch aktiv im Markt tätig sind, haben inzwischen alle XFree86 Module herausgebracht oder deren Entwicklung finanziert, so dass heute eigentlich für alle gängigen Karten auch Treiber existieren. Und damit hat sich unser Konzept als richtig erwiesen, denke ich.

**Web Portal : XFree86 4.0 ist noch recht jung ;-): Was waren seine Kinderkrankheiten? Wenn es welche gab, leidet es immer noch darunter oder hat sich das "ausgewachsen"? Wie sieht es mit der Performance von 3.3 vs. 4.0 aus?**

**Hohndel :** Kinderkrankheiten... eigentlich war 4.0 schon ein Heranwachsender, bis es endlich released wurde... aber trotzdem war noch das eine oder andere Problem zu finden, und eigentlich hätten wir doch besser noch ein bisschen länger gewartet... aber irgendwann mussten wir einfach damit mal "raus".

Angesichts der wirklich fundamentalen Änderungen von 3.3 zu 4.0 war 4.0 recht stabil und hatte überraschend wenig Probleme. Die Performance war bei den weiter verbreiteten Chipsets von Anfang an besser als bei 3.3, und auch die Korrektheit der Treiber stand der in 3.3 nur in ganz wenigen Ausnahmefällen nach. Die eigentlichen Probleme lagen darin, dass 4.0 noch zu wenig Treiber hatte und dass der eine oder andere dieser Treiber Stabilitätsprobleme hatte. Das hat 4.0.1 im Wesentlichen behoben. 4.0.2 hat dann noch einige kleine Irrwege im Design behoben und als große Neuerung die neue Rendering Architektur eingeführt. Die Buglist für 4.0.2 ist eigentlich lächerlich kurz für ein Projekt dieser Komplexität.

**Web Portal : Vielen Dank für das Interview.** 

LinuxKP.org 12.03.2001