

# Ein neuer Start

Seit das XFree86-Projekt seine Lizenz geändert hat, findet der X-Server der X.org Foundation immer mehr Beachtung bei Anwendern und Distributoren. Version 6.8 ist seit kurzem veröffentlicht und bringt viele grafische Features, nach denen sich Unix-Anwender schon lange gesehnt haben. René Rebe



www.photocase.de

**Version 6.8** des X.org-Servers [1] bringt nicht nur die Hardware-Unterstützung auf den neuesten Stand, sondern implementiert mit den Damage-, Composite- und Xevie-Erweiterungen moderne grafische Methoden. Um zu verstehen, warum Linux-Anwender und Distributoren in den letzten Monaten verstärkt auf X.org setzen, ist es nötig, ein wenig in die Vergangenheit zu blicken.

Die erste Implementierung des X-Window-Systems für x86-Rechner startete 1989/90 und ist seit 1992 unter dem Namen XFree86 bekannt. Mit der Zeit kamen viele weitere Architekturen hinzu und XFree86 wurde zum Standard auf Linux und den freien BSD-Derivaten. Die Entwickler implementierten stets den X11-Standard des MIT. 1996 übernahm dann die Open Group die Betreuung des Standards und verlangte von dieser Zeit ab Lizenzgebühren für ihre Implementierungen.

Da die Open Group X11 aber kaum weiterentwickelte, verringerte sich ihr Einfluss schnell. Daran änderte auch die

Gründung der offenen X.org-Gruppe im Jahr 1999 nichts, XFree86 wurde zur einzigen Quelle neuer Entwicklungen. Mit der XFree86-Version 4.4 änderte das Team jedoch die Lizenz und ein Großteil der Programmierer verließ das Projekt. Sie versammelten sich aber erneut, um die Entwicklung von X jetzt unter dem Dach von X.org fortzusetzen. Version 6.8 ist das erste Ergebnis dieser neuen Zusammenarbeit.

## Visuelle Effekte satt

Die größte Änderung in 6.8 ist ein alternatives Rendering-Modell, das unter anderem echte Transparenz ermöglicht. Das ist ein großer Fortschritt, denn für grafische Effekte eignete sich X bisher nicht, die Programmierer mussten zu mehr oder weniger hässlichen Hacks für Transparenz, Schatten und Pager-Funktionen greifen: Programme erstellen einfach einen Screenshot des Desktop-Hintergrunds und nutzen diesen dann, um Effekte darauf anzuwenden.

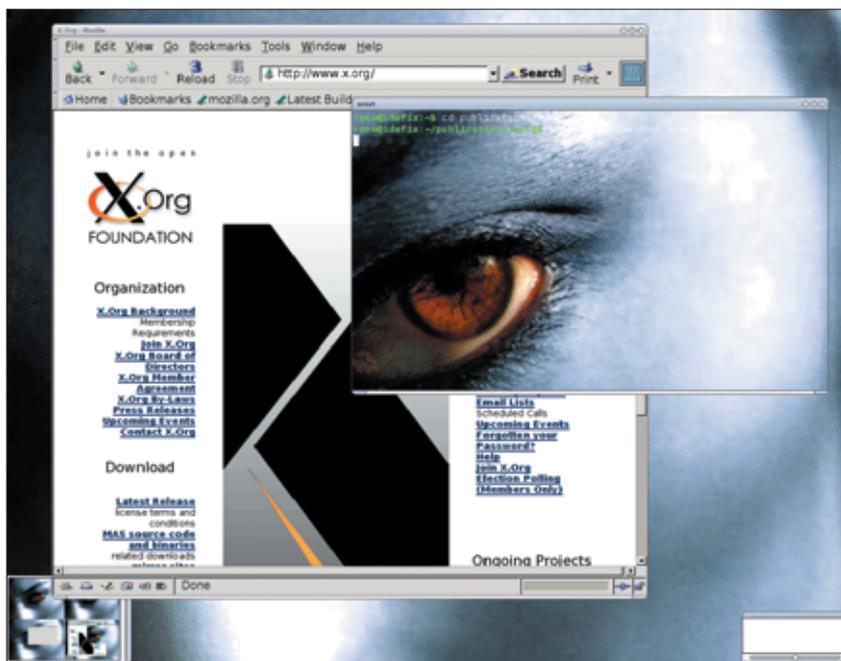
Die Terminal-Emulation Aterm benutzt diesen Trick zum Beispiel für einen pseudo-transparenten Hintergrund. Besonders effektiv ist Pseudotransparenz jedoch nicht. Fenster, die sich hinter dem angeblich transparenten Fenster befinden, sind nicht sichtbar (siehe **Abbildung 1**). Das Gleiche gilt für Pseudoschatten. Pager, die den Desktop in einer Miniansicht darstellen, benutzen ebenfalls nur Screenshots.

Außer den mangelnden grafischen Möglichkeiten stört, dass in X Verzögerungen beim Neuzeichnen von Fenstern wie in **Abbildung 2** auftreten. Zieht ein Benutzer Fenster über ein Pixmap-intensives Programm wie Mozilla oder Open Office, muss dieses seine komplette Oberfläche neu aufbauen. Das erfordert viel Zeit.

## ARGB und Composite

Um all diese Probleme zu lösen, haben die X.org-Entwickler dem Server ein ARGB-Pixelformat hinzugefügt sowie die Erweiterungen Composite und Damage implementiert. In dem neuen ARGB-Format (im X-Jargon Visual) besteht jedes Pixel nicht nur aus einem Rot-, Grün- und Blau-Anteil, sondern hat noch eine Deckungskomponente, den Alphakanal. Der Alpha-Anteil gibt beim Zusammenblenden mehrerer Pixel die Deckkraft an. Programme haben damit die Möglichkeit, transparente Bereiche zu definieren, etwa den Hintergrund eines Terminal-Emulators.

Das neue ARGB Visual allein ermöglicht jedoch noch keine Transparenz von Fenstern, da alle Programme in den sichtbaren Bildschirmspeicher zeichnen und so der Inhalt verdeckter Fenster nicht bekannt ist. Dies ändert die Com-



**Abbildung 1:** Pseudotransparenz wie hier beim Programm Aterm kopiert lediglich den Desktop-Hintergrund. Mit X.org 6.8 ist echte Transparenz möglich, wie in Abbildung 3 zu sehen.

posite Extension. Ist sie aktiviert, zeichnen Programme ihr GUI immer (auch wenn es nach dem alten Modell nicht komplett sichtbar ist) in nicht sichtbare Bereiche des Grafikspeichers. Diese Off-Screen-Pixmapen dienen dazu, die überlappenden Fenster zu repräsentieren. Ein so genannter Composition-Manager übernimmt die Aufgabe, Transparenz und andere Effekte zu erzeugen. Er läuft zusätzlich zum Window-Manager als eigener Prozess.

## Wo gibt's was zu tun?

Damit der Composition-Manager nicht viele Male in der Sekunde den gesamten Bildschirminhalt neu generieren muss, sollte er die Möglichkeit haben, zu beobachten, welche Bildschirmhalte sich wirklich verändert haben. Dafür ist die Damage Extension zuständig. Durch sie erfahren Programme, wo sich Bildschirmhalte verändern. Der Composition-Manager rendert dann gezielt nur diese Bereiche neu.

Da der Code für Composite noch als sehr instabil eingestuft ist, müssen interessierte Anwender die Extension explizit einschalten. Das tun sie entweder beim Starten des Servers per »startx -- +extension Composite« oder in der Konfigurationsdatei »/etc/X11/xorg.conf«:

```
Section "Extensions"
    Option "Composite" "Enable"
EndSection
```

Die Entscheidung, das Zusammenfügen der Off-Screen-Pixmapen einem externen Programm zu überlassen, passt zu den Grundsätzen der X-Architektur, denn die Fensterverwaltung übernimmt seit jeher ein externes Programm, der Window-Manager. Die Anwender haben also freie

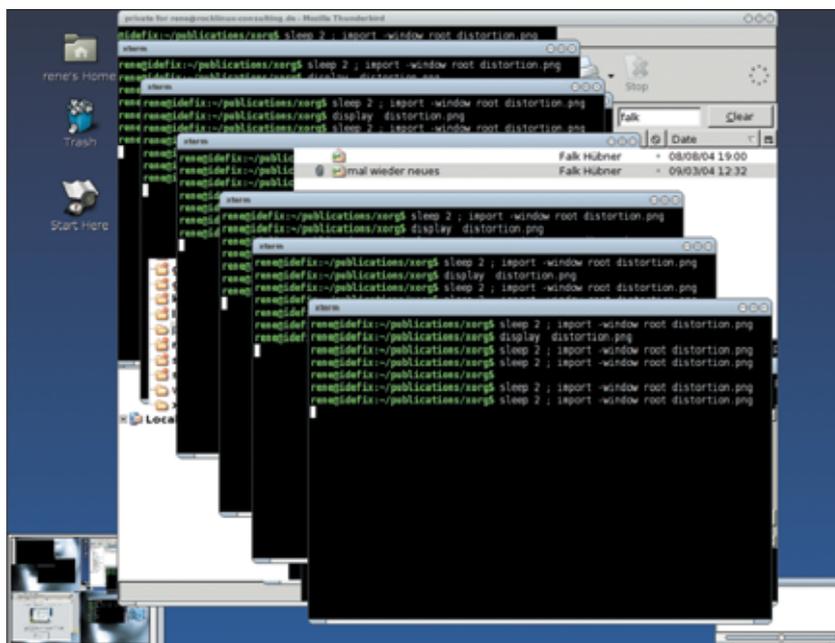
Wahl, welchen Composition-Manager sie verwenden. Außerdem werden wohl in Zukunft viele Window-Manager die Aufgaben des Composition-Managers übernehmen.

Der Composition-Manager lässt sich, genau wie der Window-Manager, einfach nach Belieben starten und beenden, während der X-Server läuft. So testen Benutzer auf einfache Weise ihre Vorlieben bei verschiedenen Effekten.

Der XCompMgr [4] ist eine Beispielimplementierung von Keith Packard. Ohne Argumente rendert er die Off-Screen-Pixmapen der Programme mit echter Transparenz, wie in Abbildung 3 zu sehen ist. Mit der Option »-c« erzeugt er durchsichtige Schatten für alle Fenster (Abbildung 4) und mit »-f« verschwinden Fenster nicht abrupt, sondern werden dezent ausgeblendet. Mit dem Tool Transset [4] legen Anwender die Transparenz einzelner Fenster fest.

## Alles ist möglich

Ein Composition-Manager muss die Fenster nicht Pixel für Pixel aufbauen, denkbar sind stufenlose Zooms und nicht-lineare Transformationen, zum Beispiel Vergrößerungseffekte von Linsen oder aufwändige Animationen beim Minimieren und Maximieren von Fenstern. Genauso ist es möglich, Fenster auf



**Abbildung 2:** Mit alten X-Versionen blieben oft Artefakte von Fenstern übrig, bis ein Programm seinen Inhalt neu zeichnete. Das ist vor allem bei Pixel-intensiven Applikationen wie Mozilla störend.

Knopfdruck neu zu gruppieren, wie Apple es mit Exposé macht.

Pager könnten immer den tatsächlichen Inhalt auch unsichtbarer Fenster zeigen, Videos laufen dann im Pager weiter. Dem Einfallsreichtum der Open-Source-Gemeinde sind in diesem Bereich keine Grenzen gesetzt. Sun Microsystems etwa entwickelt mit Looking Glass [2] an einer 3D-Oberfläche, in welcher der Composition-Manager eine OpenGL-Applikation ist, die die Fenster dreidimensional zusammenblendet.

## Mehr und weniger Performance

Die Composite Extension muss nicht mehr Rechenleistung beanspruchen als konventionelle X-Server. Keith Packards XCompmgr blendet im Auto-Redirect-Modus (Option »-a«) alle Off-Screen-Pixmapen ohne Effekte zusammen. Das verringert die Rechenlast beim Verschieben von Fenstern sogar deutlich.

Auch das Scrollen in einem großen virtuellen Workspace (wie etwa bei Enlightenment) läuft sehr flüssig. Denn ohne Composite bekommen alle Programme eine riesige Anzahl an Exposé-Events, um ihren Inhalt neu zu zeichnen. Das verlangsamt jedoch den kompletten Prozess enorm. Mit Composite kopiert der X-Server lediglich die Off-Screen-Pixmapen in den Screen, Hardware-beschleunigt versteht sich.

Der Rechenbedarf steigt allerdings, wenn viele nicht sichtbare Programme laufen. Während der Server ohne Composite deren Zeichenoperationen nicht ausführt, finden nun alle Operationen statt, da die Off-Screen-Pixmapen immer aktuell sein müssen.

In X.org 6.8 sind transparente Fenster leider nicht Hardware-beschleunigt, da die XAA-XRender-Infrastruktur nach Angaben der Entwickler dafür ungeeignet ist. XAA (X Accelerated Architecture) bezeichnet jene Infrastruktur, in der Treiber die 2D-Hardware-Beschleunigung einbringen. Nach 6.8 will das Team Neuerungen aus Keith Packards KDrive-basiertem Server [3] einarbeiten, um die Performance zu erhöhen.

Die technischen Neuerungen sind natürlich nur interessant, wenn sie auch beim Endanwender ankommen. Die Damage

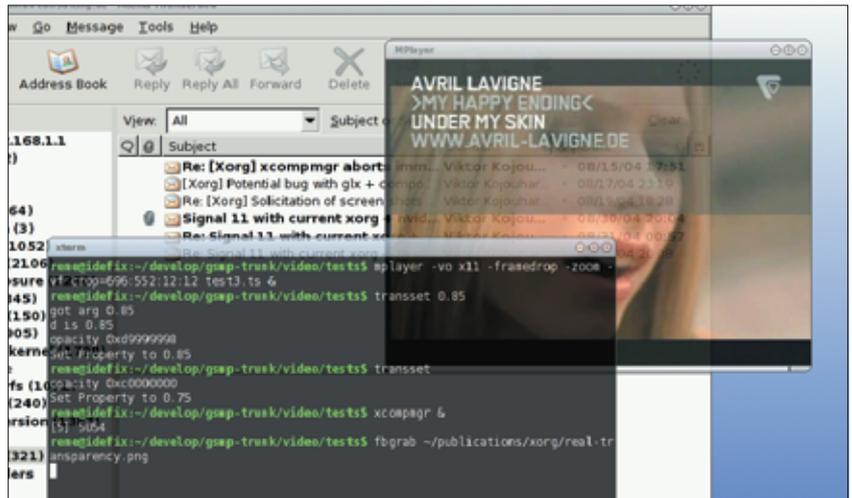


Abbildung 3: Bei echter Transparenz sind auch die hinter der Applikation liegende Fenster sichtbar. Sogar Videos laufen hinter den transparenten Fenstern weiter.

Extension erfordert keine Änderungen an den Grafik-Toolkits, denn Damage-Events generiert der Server automatisch, wenn Programme Zeichenoperationen ausführen. Auch die Composite Extension ist für Programme transparent, theoretisch sollte jeder X-Client weiterhin funktionieren. Es reicht, die Extension einzuschalten und einen Composition-Manager zu starten, um Transparenz und Schatten zu erhalten.

Nur bei Programmen, die selber Kontrolle über diese Features erhalten möchten, sind kleine Änderungen an den verwendeten Toolkits nötig. Denn Applikationen müssen in diesem Fall direkten Zugriff auf die ARGB-Visuals haben. Verhalten sich Programme bei der Auswahl des Visual fehlerhaft, kommt es unter Umständen zu Darstellungsfehlern oder

Abstürzen. Das ist bei einigen älteren GTK-1-Programmen der Fall.

Abhilfe schafft das Setzen der Umgebungsvariablen »XLIB\_SKIP\_ARGB\_VISUALS« auf den Wert »1«. Der X-Server bietet dann den Programmen die neuen ARGB-Visuals nicht an. Sobald Window-Manager einen integrierten Composition-Manager und Unterstützung für das ARGB-Pixelformat bekommen, erledigen sich diese Probleme von selbst.

## Verteilte Köpfe

Abgesehen von schönen Grafikeffekten haben die X.org-Entwickler an der Unterstützung für mehrere Displays gearbeitet. Mit X ist es schon seit einiger Zeit möglich, verschiedene Monitore von einem einzigen laufenden X-Server

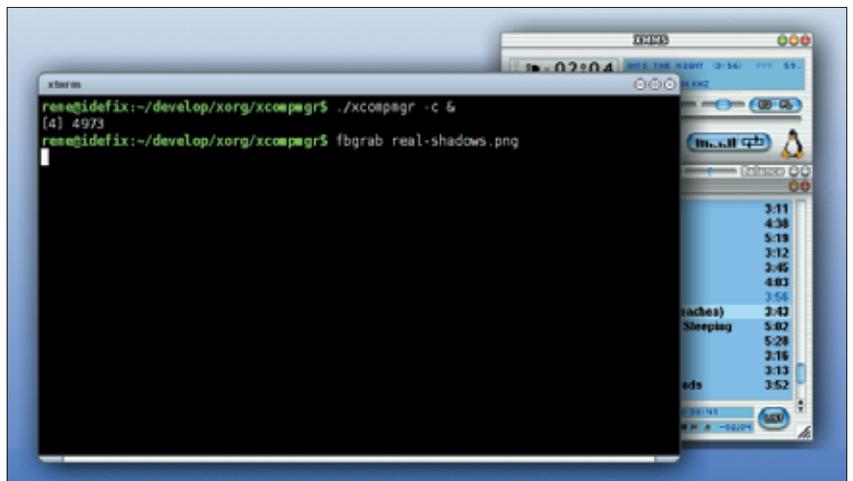


Abbildung 4: Der XCompmgr von Keith Packard ist eine Beispielimplementierung eines Composition-Managers. Er rendert Transparenz, Schatten und Ausblendungen von Fenstern.

aus anzusteuern. Dazu sind mehrere Grafikkarten pro Rechner (oder eine mit mehreren Ausgängen) nötig. Die Xinerama-Erweiterung sorgt dann dafür, dass alle Monitore einen einzigen großen Desktop bilden.

Das neue Distributed Multihead X, kurz DMX, schließt mehrere im Netzwerk laufende X-Server zu einem großen virtuellen Display zusammen. Auf den verteilten Rechnern läuft dabei ein ganz normaler X-Server, der neue XDMX-Server verbindet sich an diese als Client und sendet die Zeichenoperationen an sie. Der XDMX-Server arbeitet also gewissermaßen als Proxy zwischen den Anwendungen (Clients) und den anderen, verteilten X-Servern. Im einfachsten Fall reicht es, XDMX als Argument die zu verwendenden X-Server anzugeben:

```
»Xdmx :0 +xinerama -display Rechner1:0 -display Rechner2:0«
```

Um dann noch programmübergreifend mit den verschiedensten Eingabemethoden auf alle Applikationen zuzugreifen,

gibt es Xevie. Die X Event Interception Extension ist aus dem Gnome-Accessibility-Projekt entstanden. Sie erlaubt es, Keyboard- und Mauseingaben abzufangen, zu verändern oder beliebig zu erzeugen. Es ist mit ihr also auch möglich, programmübergreifend zum Beispiel Bildschirm-Tastaturen, Sprachsteuerung und so genannte Mausgesten zu implementieren.

Die Xevie-Erweiterung besteht lediglich aus einer Hand voll Funktionen zum Anmelden von Programmen und zum Senden von Events. Suns Looking Glass, XStroke und das Gnome-Accessibility-Projekt verwenden sie bereits.

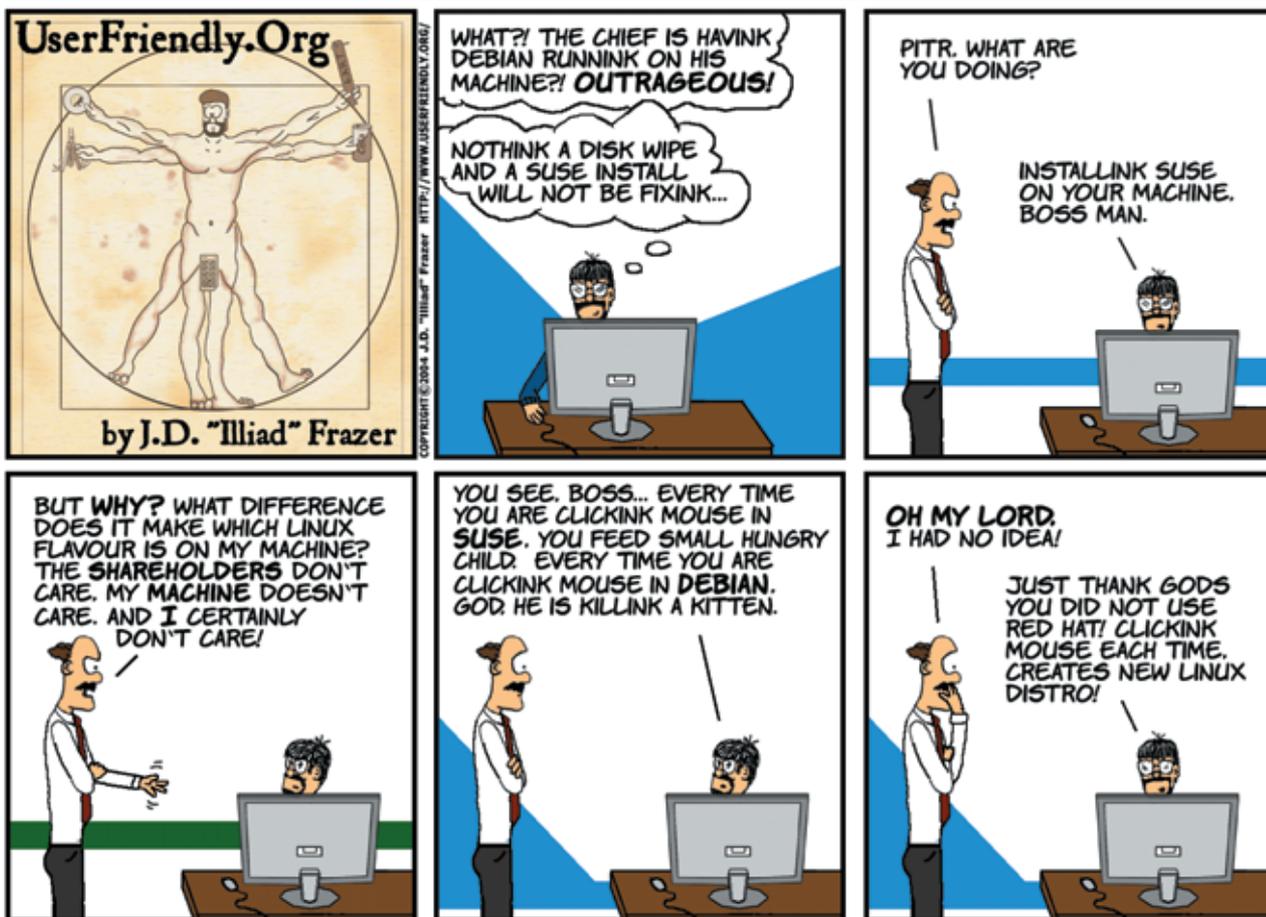
## Schöner drucken

Wer grafische Inhalte nicht nur über den Bildschirm, sondern auch über Drucker, Fax oder PDF ausgeben will, muss sich meist mit mehr oder weniger schlechten Lösungen begnügen. Wenn Programme ihre Inhalte nämlich drucken sollen,

müssen die Programmierer Routinen komplett selbst schreiben. Diese kümmern sich um die Aufbereitung der Daten für Drucker, zum Beispiel in Postscript, und interagieren mit dem Drucksystem. Deshalb sieht das gedruckte Ergebnis auch oft nicht exakt so aus, wie es der Benutzer auf dem Bildschirm hatte. Als Interface zum Drucksystem dient bei vielen Programmen nur das Senden der Daten an »lpr«.

Das ändert jetzt die Extension XPrint, die in X.org 6.8 größere Updates erfahren hat. Sie erweitert das X-Protokoll um Druckerunterstützung und nimmt dem Programm die Interaktion mit Drucksystemen ab. Außerdem bietet sie wie Cups die Auswahl der verwendeten Features eines Druckers (etwa Seitengröße, Auflösung) und erlaubt es Programmen, die gleichen Routinen zu verwenden, die auch für die normale Programmdarstellung vorhanden sind. Zusätzlicher Programmieraufwand, um etwa Postscript zu generieren, ist nicht mehr nötig. ►

### User Friendly - der monatliche Comic im Linux-Magazin



### Neue Features in X.org 6.8

- Überarbeitete 2D-Treiber: SiS, Radeon, Neomagic, MGA, i810, Savage, S3, Chips
- Überarbeitete 3D-Treiber: Mach-64 (neu), Radeon, MGA (Matrox), i810
- Überarbeitete XRender-Unterstützung: SiS, Radeon (neu)
- ARGB-Visual: Pixelformat das neben Rot, Grün und Blau auch einen Deckungsanteil - den Alphakanal - enthält
- XRender: Updates und Beschleunigung durch MMX-Assembler
- XRandR Extension (X Resize, Rotate and Reflection): Verbesserte Unterstützung in einigen Treibern
- XFixes Extension: Events bei Auswahl-Änderungen, Abfrage des Mauszeiger-Icons und benannter Mauszeiger für bessere Theme-Unterstützung. Region-Objekte, die zum Beispiel die neue Damage Extension verwendet
- Damage Extension: Events bei Zeichenoperationen von Programmen, damit andere Programme ihre Darstellung auffrischen
- Composite Extension: Erlaubt das Zeichnen von Programmen in Off-Screen-Pixmap; ein Composition-Manager kann dann diesen Inhalt beliebig transformieren und beliebige Effekte erzielen
- MESA/DRI-Updates: Umfangreichere OpenGL-Implementierung
- XPrint: Drucksystem, das X11-Anwendungen das Auffinden von Druckern und das Drucken ermöglicht
- DMX (Distributed Multihead X): Erlaubt die Verwendung anderer X-Server im Netzwerk für Xinerama - das Display lässt sich so fast beliebig vergrößern
- Xevie (X Event Interception Extension): Erlaubt das Abfangen und Verändern von Maus- und Tastatureingaben. Xevie vereinfacht auf diese Weise Behinderten den Zugang zu Computern

Die Ansteuerung des Druckers übernimmt XPrint, das dann auch hochwertigen Postscript, PCL oder Rasterdaten für den Drucker generiert. Laut Auskunft der Entwickler führt dies zu höherer Druckqualität, da viele Programme nur einen mittelmäßigen Postscript-Generator enthalten. Insbesondere soll der Ausdruck mehr Ähnlichkeit mit der Bildschirmastellung aufweisen.

## Ausblick

Neben dem neuen Rendering-Modell und einigen Erweiterungen sind jetzt auch viele Treiber auf dem aktuellen Stand. X.org 6.8 unterstützt alle neuen ATI-Modelle einschließlich der brandneuen PCI-Express-Hardware (Letztere leider bisher nur in 2D). XRender und OpenGL-Hardwarebeschleunigung haben die Programmierer weiter ausgebaut und die Unterstützung für Laptop-TFT-Bildschirme sowie mehrere Monitor-Ausgänge verbessert.

### X.org kompilieren

X.org verwendet noch immer ein recht altes Build-System namens I-Make (Include Make). Intern verwendet es den C-Präprozessor sowie dessen Makros und Include-Anweisungen, um die eigentlichen Makefiles aus »Imakefile«-Dateien zu generieren.

Nach dem Download der aktuellen Release von [1] und dem Entpacken reicht der Aufruf »make World«, um das gesamte X11-System zu übersetzen. Wer CVS verwendet, um immer die neuesten Features zu erhalten, sollte für jede Übersetzung »make Everything« verwenden. »make World« löscht nämlich zu Beginn alle bereits kompilierten Dateien. Um danach alle Binärdateien und Bibliotheken zu instal-

Die Überarbeitung der in die Jahre gekommenen X-Window-Architektur ist damit aber noch nicht abgeschlossen. Die größte Änderung steht an, wenn die Pläne, einen komplett auf OpenGL basierenden Server zu entwickeln, Gestalt annehmen. Das soll nicht nur die Performance steigern, sondern auch die Treiberentwicklung vereinfachen. 2D-Beschleunigung müssten Programmierer dann nicht mehr als Sonderfall der 3D-Beschleunigung implementieren und die Hersteller wären gezwungen, passende 3D-Treiber zu entwickeln.

Aber auch an weniger drastischen Änderungen arbeitet das X.org-Team. Die 2D-Vektorgrafik-Bibliothek Cairo [5] vereinfacht es Programmen, PDF-1.4-ähnliche Grafikprimitive zusammenzustellen, die der X-Server mit XRender oder OpenGL Hardware-beschleunigt darstellt und in hoher Qualität ausdrückt. Cairo soll nicht nur Vektorprogramme, Postscript- und PDF-Viewer beschleunigen, sondern auch Office-Programme und transparent

markierte Auswahlen, wie sie schon Gnomes Nautilus verwendet.

## Automake & Co.

Wer seinen X-Server oft aus den Sourcen kompiliert, wird sich darüber freuen, dass die Entwickler das Build-System überarbeiten. Sie überführen das monolithische Makefile-System in eine modulare Struktur mit den GNU Auto Tools, um etwa vorhandene Bibliotheken automatisch zu erkennen.

Das aktuelle monolithische Build-System installiert nämlich seine Bibliotheken wie Expat, Zlib, Fontconfig, Freetype und Mesa im System auch dann, wenn sie schon auf dem Rechner vorhanden sind. Geplant ist ebenfalls, die vielen Utilities wie XClock, XDM, XEyes, XMag, XMan, XMH, XEdit, XTerm in einzelne Pakete auszulagern. (mwe) ■

### Infos

- [1] X.org: [<http://www.x.org>]
- [2] Looking Glass: [[http://www.sun.com/software/looking\\_glass/](http://www.sun.com/software/looking_glass/)]
- [3] Keith Packards KDrive-X-Server: [<http://freedesktop.org/Software/xserver/>]
- [4] XCompmgr, Transset und weitere X-Applikationen: [<http://freedesktop.org/Software/xapps>]
- [5] Vektorgrafik-Bibliothek Cairo: [<http://www.cairographics.org>]

### Der Autor

René Rebe studiert Technische Informatik an der TFH-Berlin und kennt Linux leider erst seit 1997. Er arbeitet bei mehreren Projekten mit, zum Beispiel GSMP und Sane.