

Dieser Artikel ist Teil des
Open Source Jahrbuch 2005



erhältlich unter <http://www.opensourcejahrbuch.de>.

Das Open Source Jahrbuch 2005 enthält neben vielen weiteren interessanten Artikeln ein Glossar und ein Stichwortverzeichnis.

Open Code Worlds

WOLFRAM RIEDEL



(CC-Lizenz siehe Seite 463)

1. Einleitung zum Kapitel „Technik“

Unser Buch vereint Betrachtungsweisen aus sehr unterschiedlichen Disziplinen und beinhaltet auch Thematiken, die mit Technik nur noch am Rande zu tun haben. Und doch lohnt es sich, zunächst noch einmal zu den technischen Wurzeln der neuzeitlichen Open-Source-Bewegung zurückzukehren, bevor man den Versuch wagt, die dort bereits funktionierenden Modelle auch auf andere Bereiche zu übertragen. Das Kapitel „Technik“ bietet somit nicht nur eine interessante Lektüre für technisch interessierte Leser, sondern auch eine bereichernde Grundlage für die weiteren fachwissenschaftlichen Vertiefungen.

Innerhalb des letzten Jahres ist viel passiert, und wie immer wird das vergangene Jahr gerne als das beste, aktivste und erfolgreichste Jahr der Open-Source-Geschichte bejubelt (Lindner 2005). Und natürlich ist es nicht möglich, alle aktuellen technischen Entwicklungen in diesem Kapitel abzudecken, jedoch hoffe ich, dass es gelungen ist, die interessantesten Themen herauszupicken. Nach einem Blick auf den Open-Source-Desktop und seine Benutzbarkeit widmen wir uns der Evolution freier Software und der Entwicklung von Linux auf eingebetteten Systemen.

2. Open Source auf dem Desktop

Nach dem stetigen Erfolg von Linux im Server-Segment ist inzwischen immer öfter von Umsteigern bei Desktop-Rechnern zu hören: Die Migrationsberichte häufen sich, bei denen auch die Arbeitsplätze mit Linux „gefahren“ werden. Maßgebliche Faktoren für diese Entwicklung sind die beiden beliebten Desktop-Umgebungen GNOME und KDE, die in den rasch vorangetriebenen Versionen des Jahres 2004 schon sehr weit gereift sind.¹ Und auch OpenOffice, die wohl beliebteste freie Office-Suite, hat mit Sicherheit keinen geringen Anteil am Erfolg von Open Source auf dem Desktop.

¹ KDE 3.2 erschien im Februar, KDE 3.3 im August. Vor allem seine PIM-Applikationen wurden stark weiterentwickelt und arbeiten untereinander besser zusammen. GNOME 2.6 ist seit April, GNOME 2.8 seit September 2004 verfügbar. Sowohl das *Panel* als auch der Dateimanager Nautilus sind noch übersichtlicher geworden. Ein weiteres Highlight ist die Groupware-Applikation Evolution 2.0.

2.1. Rückgrat des Desktops

Eine sehr verbreitete und zentrale Komponente auf freien, unixoiden Desktop-Systemen ist der X-Server. Ihm wird im Vergleich zu den darauf aufbauenden Oberflächen KDE und GNOME oft weniger Aufmerksamkeit geschenkt.

Für Wirbel in der Szene sorgte 2004 eine Lizenzänderung der freien Implementation XFree86, deren Auswirkungen und GPL-Kompatibilität unklar und umstritten bleiben. Zahlreiche Distributionen verwenden XFree86 seitdem nicht mehr. Besonders bemerkenswert ist der zeitnahe Umbau der X.Org-Stiftung², die nun hauptsächlich von Entwicklern der Community geleitet wird und die die letzte XFree86-Version vor der Lizenzänderung „geforkt“ hat und jetzt offen weiterentwickelt.

Als Plattform im Web dient freedesktop.org, die sich als ein zentraler Anlaufpunkt für die Community herauskristallisiert hat, wenn es um Standards, *frameworks* und Backends für den X-Desktop geht. Die Zugewandtheit zur Community sorgte anscheinend für einen Motivationsschub: Die neuen Features der ersten Version nach der Neuaufstellung bescherten dem Projekt den Linux New Media Award³ in der Kategorie „Best Free Project for Hardware Support“.

Im Sommer 2004 erschien erstmals FreeNX, eine auf einem Shellskript basierende freie Implementation des NX-Servers von NoMachine⁴. Im X-Protokoll einer quasi nächsten Generation wurde, vereinfacht gesagt, der Kommunikations-Overhead deutlich reduziert und eine Kompression eingebaut.⁵

„Selbst bei geringer Bandbreite wie mit Modems oder per ISDN erreichen NX und FreeNX eine gute Performanz – und das quer über mehrere Betriebssystem-Plattformen hinweg.“ (Kurt Pfeifle, zitiert bei Ihlenfeld 2004a)

Zügig wurde die Software in Knoppix integriert und steht seit der KDE-Konferenz aKademy 2004 auch auf „temporären“ Linuxrechnern ohne großen Installationsaufwand zur Verfügung.

2.2. Fenster-Notausstieg – bitte Scheibe einschlagen

Nach einer möglichen Entscheidung, die bunten Kirchenfenster durch geschäftige Frackträger zu ersetzen, steht der Migrant zum Glück nicht allein auf weiter Flur: Es gibt zahlreiche Hilfsmittel, die einen Umstieg von Windows auf Linux bzw. die Einarbeitung in Linux erleichtern können. Spezielle Linux-Distributionen bereiten gerade weniger erfahrenen Computernutzern weniger Probleme beim Umstieg und der

2 X.Org <http://www.x.org/>

3 Linux New Media Award 2004 http://www.linuxnewmedia.de/Award_2004

4 NoMachine <http://www.nomachine.com>

5 Die Netzwerk-Fähigkeit erlaubt das Arbeiten auf entfernten Rechnern. Bei NX und FreeNX handelt es sich nicht um einen offiziellen Standard der X.Org-Stiftung. Es ist aber anzunehmen, dass die zukünftige Entwicklung des X-Protokolls von X.Org in diese oder eine ähnliche Richtung gehen wird.

sich anschließenden Benutzung. Neben den schon immer als recht benutzerfreundlich geltenden Distributionen von Novell⁶ und Mandrake⁷ wirbt besonders Xandros⁸ mit einem intuitiven Desktop. Einige Umsteiger, die sehr an die Microsoft'schen Oberflächen gewöhnt sind, tun sich schwer, sobald ein Desktop eine andere Optik anbietet. Linspire⁹ und Freedows¹⁰ sind zwei Distributionen, die diese Nutzergruppe ins Auge gefasst haben und sich explizit als Windows-Ersatz platzieren. Ihre Produkte ähneln einem Windows-Desktop und mindern so Berührungsängste beim Umstieg. Zusätzlich bieten sie wie Xandros teilweise Kompatibilität zu gängigen Windows-Programmen. Weiterhin gibt es im Internet eine Fülle von Anleitungen und Tutorials für den geschäftlichen wie auch den privaten Umsteiger, z. B. die IOSN-Anleitung¹¹ oder das Linux Client Migration Cookbook¹² von IBM. Sogar Schulungs-Videos¹³ sind frei verfügbar.

2.3. Feuerfuchs und Donnervogel

Mozilla, das vom Markt verdrängte Stiefkind des einstigen Browserpioniers Netscape, hat sich zur sprichwörtlichen Eier legenden Wollmilchsau gemauert. Und gleichzeitig ist das Projekt ein leuchtendes Beispiel für die Möglichkeiten und die Flexibilität von OSS: Der nur schwer wartbare Netscape-Code wurde nach der Veröffentlichung 1998 von der Community einem grundlegenden *Re-Design* unterzogen, bei dem sich der Änderungsaufwand auf ein Fünftel und der Koordinationsaufwand auf ein Drittel reduzieren ließen (MacCormack et al. 2004).

Die Aufspaltung des inzwischen übergroßen Projektes war ebenso sinnvoll, denn jedes Teilprogramm hat schon länger den Umfang eines eigenen Projektes, nicht zuletzt, da plattformübergreifend entwickelt wird. Diese Trennung darf man allerdings nicht missverstehen, denn die Entwickler der Mozilla-Suite¹⁴ arbeiten weiterhin eng zusammen. Die Applikationen bauen teilweise auf den gleichen Komponenten auf und sind mit zahlreichen Extentions erweiterbar, ihre Benutzerführung bleibt einheitlich. Die heiß erwartete Version 1.0 des abgespaltenen Browsers „Mozilla Firefox“ erschien im November 2004 und hatte nach nur elf Wochen rekordverdächtige 20 Millionen Downloads zu verzeichnen. Des Öfteren kann man beim Online-Banking Hinweise auf die höhere Sicherheit dieses Browsers im Vergleich zu vorinstallierten Windows-Browsern lesen. Das Mail- und Newsprogramm „Mozilla Thunderbird“ landete im

6 SuSE-Linux <http://www.novell.com/de-de/linux/suse/>

7 Mandrakelinux <http://www.mandrakelinux.com/de/>

8 Xandros-Desktop <http://www.xandros.com/>

9 Linspire (<http://www.linspire.com>) ist eine amerikanische Firma, die ihr Produkt vor einem Rechtsstreit mit Microsoft auch unter dem Namen Lindows anbot. Sie setzt auf kostengünstige Rechner mit vorinstalliertem Linux in Verbindung mit kostenpflichtigen Software-Zusatzpaketen und Support.

10 Freedows <http://www.freedows.com> ist ein Projekt zweier Firmen aus Brasilien, eine englische oder deutsche Übersetzung ist leider noch nicht verfügbar.

11 IOSN User Guide to Using the Linux Desktop <http://www.iosn.net/training/end-user-manual/>

12 Linux Client Migration Cookbook <http://www.redbooks.ibm.com/abstracts/sg246380.html>

13 Video Tutorials <http://www.about-linux.com>

14 Mozilla Project <http://www.mozilla.org/products/>

Dezember und wurde innerhalb eines Monats 2 Millionen Mal heruntergeladen. Gelobt werden sein Spam-Filter und die Integration von RSS-Feeds.

Die Mozilla Foundation hat zusammen mit Adobe, Apple, Macromedia, Opera und Sun Microsystems eine neue, offene Plugin-Schnittstelle für Browser angekündigt, die demnächst in Mozilla und Produkten der genannten Partner eingebaut wird (Mozilla Organization 2004). Dies ist ein wichtiger Schritt, um zu verhindern, dass Interaktivität in Webbrowsern auf einzelne Plattformen beschränkt bleibt.

2.4. Basen und Oasen

Distributionen von Linux sind untereinander nicht automatisch kompatibel, andererseits erfordert ihre Pflege aber oft die gleichen oder ähnlichen Arbeitsschritte der Hersteller. Frühere Versuche, eine gemeinsame Basis-Distribution zu erstellen, müssen als gescheitert angesehen werden.¹⁵ Jedoch wagten Conectiva, Mandrakesoft, Progeny und Turbolinux nun kurz nach der Festlegung der Linux Standard Base 2.0¹⁶ einen neuen Anlauf.¹⁷

Ein noch viel wichtigerer Standard ist der von Dateiformaten. Wer seine privaten oder geschäftlichen Dokumente nicht in einem offenen Format abspeichert, wird immer von der Verfügbarkeit proprietärer Software abhängig bleiben, um darauf zuzugreifen.

Das OpenOffice-Format war von Anfang an offen, jedoch kein allgemeingültiger Standard, bis sich das OASIS-Konsortium dessen angenommen hat und auf Basis des bestehenden OpenOffice-Formats einen neuen Standard für Office-Dateien spezialisierte.¹⁸ Das vor der Tür stehende OpenOffice 2.0 wird ihn unterstützen, natürlich auch kommende Versionen von StarOffice, ebenfalls verwendet wird ihn KOffice 1.4. Die Europäische Union und Sun Microsystems setzen sich dafür ein, dass das Format zum ISO-Standard gemacht wird (Ihlenfeld 2004b).

2.5. Dämon beißt in den Apfel

Anders als Linux-Distributionen, die aus GPL-lizenzierter Software bestehen und eher wenige bis gar keine proprietären Zusätze haben, und anders als Windows, das für den Nutzer vollkommen proprietär daherkommt, geht Apple mit seinem MacOSX einen Mittelweg. Der Kernel ist ein FreeBSD-Derivat und wird als OSS-Projekt entwickelt.

Dem System liegen die bekannten und offenen Tools der BSD-Derivate bei, und die sind, wie von BSD-Systemen gewohnt, wie aus einem Guss:

„Wo sich *Linux* als recht heterogener Baukasten mit diversen Kernel-Varianten und -Versionen, zahlreichen Tools aus unterschiedlichsten Quellen und rund 200 Distributionen mit mehr oder weniger großen

15 Siehe UnitedLinux <http://www.unitedlinux.com/>.

16 Linux Standard Base <http://www.linuxbase.org/>

17 Linux Core Consortium, siehe <http://componentizedlinux.org/lb/>.

18 OASIS Open Office XML Format Technical Committee http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office

Unterschieden präsentiert, legen die mittlerweile knapp 200 NetBSD-Entwickler (wie ihre Free- und OpenBSD-Kollegen) Wert auf ein homogenes System: Es gibt genau eine aktuelle Distribution, und in der steckt alles drin, was zum System dazugehört.“ (Diedrich und Kuri 2004)

Damit erbt OSX die Stabilität eines BSDs, umschiffet aber die für den typischen Apple-Benutzer zu schwierige Installation, Konfiguration und Wartung. Eine Kommandozeile bekommt man zunächst nicht zu Gesicht. Weiß man aber um sie, kann man von ihr vollen Gebrauch machen. Dies ist zwar auch schon bei einigen Linux-Distributionen so gelöst, jedoch ist durch Apple als zentrales Entwicklungs- und Entscheidungsorgan auch die Oberfläche „Aqua“ wie aus einem Guss und bietet die für Apple typische Bedienung. Allerdings erben die Oberfläche und einige Zwischenschichten auch die für proprietäre Software typischen Eigenschaften: Anpassungen und Verbesserungen vornehmen kann ebenso wie Fehler und Sicherheitslücken beheben nur der Hersteller, denn er behält die volle Kontrolle über den Code und damit über seine Software. Und ob man mit OSX besser zurechtkommt als mit Windows, KDE oder GNOME, mag jeder für sich entscheiden: Die Benutzbarkeit eines Systems ist zu einem großen Teil auch vom individuellen Geschmack und der persönlichen Gewöhnung des Benutzers abhängig.

2.6. Open-Source-Usability

Der erste Artikel des Technik-Kapitels beschäftigt sich mit der Usability (Benutzungsfreundlichkeit) von Open-Source-Software. Er trifft genau ins Schwarze, um ihre charakteristischen Stärken und Schwächen für den Einsatz auf dem Desktop auszuloten. Grundlage für den Artikel sind eine Menge praktischer Erfahrungen mit der Materie, unterfüttert von einer Usability-Studie. Autor des Artikels ist Jan Mühlig, Gründer und Vorstand der relevantive AG, einem IT-Unternehmen aus Berlin, das insbesondere der KDE-Community nahe steht.¹⁹ Neben der Mitarbeit im KDE-Projekt und einer Usability-Kolumne im Magazin „LinuxUser“ hat das Team von relevantive den Usability-Track auf der KDE-Konferenz aKademy 2004 maßgeblich gestaltet und ist somit bestens qualifiziert, Erfahrungen aus erster Hand mitzuteilen.

3. Freie Software-Entwicklung

Kein Thema ist so nah dran an Open Source wie seine Entwicklung, denn man kann mit dem Begriff „Open Source“ nicht mehr nur eine Gruppe von Lizenzen bezeichnen: Mit der Community hat sich eine ganz neue Art der Software-Entwicklung etabliert. Auch wenn viele OSS-Projekte leichte Unterschiede im Entwicklungsprozess aufweisen, so haben sie alle doch eine gemeinsame Grundlage: den ungehinderten Austausch von Code.

¹⁹ Zum Team von relevantive gehört noch Ellen Reitmayr. An der Linux-Usability-Studie haben außerdem Eva Brucherseifer und Ralf Ackermann von basysKom und Jutta Horstmann mitgewirkt.

3.1. Blick über den Tellerrand

Ein sehr interessantes Thema, dessen Beobachtung sich über die nächsten Jahre hinweg sicher lohnen wird, ist das der APIs. Ein *Application Programming Interface* ist sicherlich die wichtigste Schnittstelle für einen Programmierer und kann indirekt entscheidend sein für den Erfolg und die Verbreitung eines Softwaresystems und ist damit auch ein wichtiges Instrument im Konkurrenzkampf für Closed-Source-Giganten wie Microsoft und Apple. Ein ehemaliger Microsoft-Mitarbeiter und Gründer einer Software-Firma erklärt in seinem Artikel, warum er davon ausgeht, dass Microsoft den „API War“ bereits verloren hat (Spolsky 2004). Er führt dies auf die radikale Einführung neuer Windows-APIs zurück, die in zukünftigen Windows-Versionen keine Rückwärts-Kompatibilität mehr liefern werden. Obwohl die neuen Schnittstellen technisch durchaus gut entworfen seien, fingen die Programmierer in Softwarefirmen mehr oder weniger bei null an und müssten sich zudem mit zwei konkurrierenden Grafik-APIs herumschlagen. Programmierer mit der entsprechenden Expertise seien entsprechend knapp verfügbar und trieben die Personalkosten nach oben.

Beim Nennen des Begriffs *Registry* gehen bei vielen Menschen haufenweise rote Lichter an. Der Grund ist die schlechte Implementierung der *Registry* unter Microsofts Windows, die ihren Ruf als Geschwindigkeitsbremse und Wartungsproblem verdient hat. Dennoch hat es sich das Projekt Elektra²⁰ zum Ziel gesetzt, eine *Registry* für Linux zu entwerfen und zu implementieren, allerdings ohne die bekannten Design-Schwächen dabei nachzubauen.

Während Longhorn noch lange auf sich warten lässt, ist 64-Bit-Computing in der Open-Source-Welt bereits Realität. Mono 1.0 ist im Juni 2004 angekommen und hat einen wichtigen Grundstein dafür gelegt, dass dot.NET-Applikationen auch unter Linux laufen werden. Schon länger verbreitet sind Java-Applikationen, die auch unter Linux ihren Dienst wie gewohnt verrichten,²¹ trotzdem Sun sich auch 2004 nicht entschließen konnte, sein Java unter eine echte Open-Source-Lizenz zu stellen. Sun verbreitet den Code zwar, lässt Modifikationen aber nicht frei zu.²² Immerhin wird es von Suns Betriebssystem Solaris demnächst eine offene Version namens „OpenSolaris“²³ geben, die eine interessante Alternative zu Linux und *BSD werden könnte. Besonders bemerkenswert ist die angestrebte Kompatibilität zu Linux-Applikationen, vorerst aber sollen wichtige Teile wie der Kernel nicht unter eine Lizenz gestellt werden, die eine Weitergabe von Modifikationen erlaubt (Ziegler 2004).

3.2. „BUGS!“

Fehler im Code sind jedem Programmierer ein Greuel. Im Jargon werden sie Bugs genannt. Kurz nach Anfang der (damals noch langsameren) computertechnischen Geschichtsschreibung finden wir einen kleinen Störfall, der sich 1945 während eines

20 Elektra Initiative <http://elektra.sourceforge.net/>

21 Java Linux <http://www.blackdown.org/>

22 Eine freie Java-Implementation im Sinne der GPL wird derzeit im GNU-Projekt Classpath erstellt, (<http://www.gnu.org/software/classpath/>).

23 OpenSolaris <http://www.opensolaris.org/>

Testlaufs des Urgesteins Mark II an der Harvard University ereignete (Naval Historical Center 1999). Eine Motte verfiel sich in einem der zahlreichen Relays. Das von Grace Hopper in das Logbuch geklebte Insekt kann nicht nur im Museum bestaunt werden, sondern gilt als der Ursprung der Bezeichnung Bug für Fehler bzw. der Tätigkeit des *Debugging*, bei der man Fehler in Programmen aufstöbert und sie entfernt.

In Open-Source-Software können Bugs leicht gefunden und beseitigt werden. Durch den offenen Programmcode ist diese Eigenschaft implizit immer vorhanden, eine Garantie, dass diese Möglichkeit auch genutzt wird gibt es aber nicht. Eine Belohnung²⁴ für sachdienliche Hinweise zum Aufspüren von Fehlern weckt bunte Assoziationen, angefangen bei den „Wanted“-Steckbriefen im Wilden Westen, über Kopfgeldjäger jenseits des Amazonas bis hin zu den Fahndungsplakaten des BKA. Diese Methode ist jedoch nicht für alle OSS-Projekte gleichermaßen geeignet.

3.3. Kern der Dinge

Die vermutlich bekannteste freie Software ist der Linux-Kernel. Seine Entwicklungsmethode stand in der Vergangenheit immer wieder in der Kritik, da auch in den „stabilen“ Kernel-Versionen bereits schwerwiegende Änderungen vorgenommen und damit auch teils schwere Fehler eingebaut wurden. Linus Torvalds (2004b) selbst stellte in einem längeren Text einige Dinge über seinen Management-Stil als Entwicklungsleiter klar und ging auch ausführlich auf den persönlichen Umgangston ein, der nicht rüde werden dürfe: „don't call people d*ckheads (at least not in public)“. Beim Kernel Summit 2004, dem traditionellen Treffen der Linux-Kernel-Entwickler, wurden die Änderungsmöglichkeiten erörtert. Die Diskussion verlief überwiegend zugunsten gradueller Änderungen und paralleler Test-Kernels (LWN.net 2004).

Der Finne Ilkka Tuomi analysiert in seinem Artikel „Evolution of the Linux Credits“ (2004) die Entwickler-Gemeinde des Linux-Kernels. Er beschreibt dabei eine automatische Auswertung der Datei „CREDITS“, die sich als schwierig entpuppt aufgrund der teils sehr unterschiedlichen Weise, wie die vorgesehenen Felder (z. B. Beschreibung oder Adresse) ausgefüllt wurden.

Die Datei war gedacht, um die Teilnahme am Kernel-Projekt festzuhalten, jedoch eignet sie sich nicht für einen juristisch verwendbaren Nachweis über die Urheberschaft bestimmter Code-Teile. Deshalb und aus gegebenem Anlass – die Diebstahlsbehauptungen von SCO standen im Raum – wurde das „Signieren“ von Patches²⁵ eingeführt, damit kein Code aus unbekanntem Quellen in den Kernel aus Versehen oder absichtlich eingeschleust wird (Torvalds 2004a).

Vor allem über Anzahl und Wohnsitz lassen sich dennoch Daten aus dem Credits-File gewinnen: So waren bis zur Linux-Version 2.5.25 Mitte 2002 über 400 Personen

24 Mozilla Foundation Announces Security Bug Bounty Program <http://www.mozilla.org/press/mozilla-2004-08-02.html>

25 Der „unterzeichnende“ Entwickler bestätigt mit einem „Developer's Certificate of Origin“, dass der Code von ihm selbst oder einem ihm bekannten Entwickler unter einer Open-Source-Lizenz geschrieben wurde oder auf einem solch lizenzierten Code aufbaut.

am Projekt beteiligt, davon 154 aus den Vereinigten Staaten und 69 aus Deutschland. Dahinter lagen Großbritannien, Kanada, Australien und die Niederlande mit 32–18 Entwicklern. Nimmt man die Anzahl aus allen EU-Ländern zusammen, so überstieg diese mit 187 sogar die Zahl der amerikanischen Entwickler. Auch wenn in vielen anderen Staaten meistens weniger Entwickler beheimatet sind, so ist Linux dennoch ein sehr internationales Projekt, das – vereinfacht gesagt – alle etwai- gen Probleme überwunden hat, von nur einem einzigen Entwickler auf über 400 Beitragende zu skalieren.

3.4. Software-Evolution

Mit dem Thema der Software-Entwicklung beschäftigt sich der zweite Artikel des Technik-Kapitels: Markus Pizka und Andreas Bauer von der Technischen Universität München haben sich die Mechanismen der Open-Source-Entwicklung einmal genauer angesehen und erklären, warum sie oft besser skalieren und funktionieren als unter den Bedingungen kommerzieller Entwicklung von Closed-Source-Software.²⁶ Bei der Software-Evolution geht es um die kontinuierliche und kontrollierte Weiterentwicklung von Software und insbesondere um die Veränderung des Entwicklungsprozesses an sich.²⁷

4. Open Source im Jahrbuch

Es ist nur konsequent, wenn in einem Buch über Open Source auch Open Source drin ist. Richtig, gemeint ist Programmcode, den man lesen und mit ausreichend fachlicher Kenntnis auch verstehen kann. Wer nicht selber programmiert, hat vielleicht noch nie gesehen, wie die „Kochrezepte“ der Informationstechnologie aussehen. Wer nicht allerengste Beziehungen zu einer „Softwareschmiede“ pflegt, für den ist Open Source die einfachste Möglichkeit, sich einen Eindruck zu verschaffen.

Im Rahmen dieses Buches können wir natürlich nur ein kleines Programm stellvertretend für die Massen an offenem Sourcecode abdrucken, die täglich geschrieben, verbessert und verwendet werden. Um die Auswahl aus dieser schier unendlichen Fülle zu erleichtern, haben wir, die Redaktion des Jahrbuchs, im Herbst 2004 einen Wettbewerb ausgerufen. Aus den Einsendungen wurde auf unserer Internetseite²⁸ das Programm Snowbox von Oliver Feiler zum Gewinner gewählt. Gratulation!

Oliver Feiler hat für seinen mit OpenBSD betriebenen Server einen einfachen und sicheren POP3-Server gesucht. Nachdem er mit bereits vorhandenen Lösungen nicht zufrieden war, hat er sich dazu entschlossen, selber einen zu schreiben. Sein eher minimalistischer Ansatz legt den Schwerpunkt auf wenig Code, Sicherheit und einfache Installation und Konfiguration. Er hat Snowbox in Perl geschrieben:

26 Einen flüchtigen, aber interessanten Einblick in die proprietäre Softwareproduktion gibt z. B. auch das Blog von Rick Schaut (http://blogs.msdn.com/rick_schaut/archive/2004/05/19/135315.aspx).

27 Nicht gemeint sind genetische Algorithmen, siehe dazu „Linux: Tuning The Kernel With A Genetic Algorithm“ <http://kerneltrap.org/node/4493>.

28 Open-Source-Jahrbuch <http://www.Think-Ahead.Org>

„Perl, weil es sehr einfach ist, damit Text zu verarbeiten, und weil es die Kriterien 'so wenig Code wie möglich' und 'so sicher wie möglich' am ehesten erfüllt. Ein Server sollte vor allem erstmal sicher sein, an der Geschwindigkeit kann man immer noch drehen. Und da es unmöglich ist, mit Perl klassische Buffer Overflows, wie wir sie alle von C 'lieben', zu programmieren und ich Perl recht gut kenne, fiel die Wahl darauf. Darüber hinaus sieht man mit Perl recht schnell Resultate. Die erste funktionierende Version war bereits nach wenigen Stunden fertig.“
(Oliver Feiler)

Das Programm hat er unter die GPL gestellt, da diese Lizenz seinen Vorstellungen entspricht und weithin bekannt ist. Aus „Snow“ – er liebt Schnee und Schneecoparden – und „box“ vom englischen Wort Mailbox für Briefkasten wurde schließlich der Name Snowbox.²⁹

5. Eingebettete Systeme

Von digitalen Assistenten, die bald in unsere Kleidung eingearbeitet sein könnten, über Bordcomputer für Autos oder Kühlschränke, bis zur Technik, die die International Space Station im Orbit hält, besteht zunehmend Bedarf an eingebetteten Systemen.

„I get the biggest enjoyment from the random and unexpected places. Linux on cellphones or refrigerators, just because it's so not what I envisioned it.“ (Linus Torvalds, zitiert bei Dudley 2004)

Eine primäre Aufgabe des Betriebssystems ist die Verwaltung von Hardware-Ressourcen. Unabdingbar sind dazu Treiber, mit denen die Hardware angesteuert werden kann.

5.1. Linux-Treiber

Bei Linux sind sie ein heißes Eisen. Die Treiber sind zumeist fest im Kernel eingebaut, der Vorteil ist eine sehr gute Integration der Treiber in das System. Der Nachteil ist, dass Treiber nicht beliebig hinzugeladen werden können. Zwar können ausgewählte Teile des Kernels als Kernelmodule dynamisch geladen werden, diese müssen aber bereits in die Quellen des Kernels eingebettet sein.

Die Integration erfordert die Offenlegung der Treiber-Quellen, ein Schritt, den viele Hersteller nicht gehen wollen, da sie die Funktionsweisen ihrer Geräte oft als Geschäftsgeheimnis ansehen oder nicht gehen können, da Teile ihres Treibers lizenzierten Code von Dritten enthalten. Eine Möglichkeit, dies zu umgehen, sind binäre Kernel-Module, bei denen nur die Einbindung in den Kernel offen liegt. Genau dies ist die Vorgehensweise von NVidia bei den Linux-Treibern ihrer Grafikkhardware.

²⁹ Aktuelle Versionen von Snowbox und andere Software von Oliver Feiler finden sich auf seiner Homepage <http://kiza.kcore.de/software/> oder bei Freshmeat <http://freshmeat.net/~kiza/>.

Binäre Kernelmodule werden in Teilen der Open-Source-Szene nicht gern gesehen, aber ohne diese Lösung gäbe es vermutlich noch weniger Treiber.

„Pranger für Hardwarehersteller?“ (Lindner 2004b) lautete eine provokante Schlagzeile: Als Benutzer von Nischensystemen kann man sich nämlich sehr leicht im Stich gelassen fühlen, wenn die auserwählte Plattform von großen Hardware-Herstellern systematisch übergangen wird. Unverständnis macht sich breit, denn der Benutzer kann nicht verstehen, warum Hersteller freiwillig auf Kunden verzichten. Dies gilt ganz besonders für Betriebssysteme wie Linux, die inzwischen zum „Mainstream“ gehören (IDC 2004). Die teils desolatte Treibersituation bei gängiger Hardware und Peripherie erscheint manchmal absurd und hindert vor allem unbedarfte Privatnutzer am dauerhaften Wechsel zu Linux.

Ein negatives Beispiel ist die WLAN-Unterstützung auf Hardware von Intel und Apple. Es gab zum Erscheinungstermin von Intels Centrino-Komponenten für Notebooks zunächst nur einen Windows-Treiber. Für Linux wurde ein Treiber zwar angekündigt, dann aber auf Eis gelegt. Erst weit über ein Jahr später steht nun ein erster nativer Treiber für eine inzwischen ältere Bauart der Centrino-Chips zur Verfügung, der in Zusammenarbeit mit einem Open-Source-Projekt entwickelt wurde (Lindner 2004a). Wer sich einen neueren Schoß-Rechner von Apple zugelegt hat, kann Airport Extreme³⁰ auch weiterhin nicht unter Linux oder freien BSDs nutzen, ein Treiber oder die Freigabe von Spezifikationen ist nicht geplant. Als kleiner Trost bleibt den üblichen Verdächtigen mit ihren schnuckeligen iBooks nur, dass unter der Haube ihres MacOSX ohnehin schon ein Unix werkelt.

Vor unserem kleinen Treiber-Exkurs wurden einige Ideen für eingebettete Systeme bereits erwähnt, einige konkrete Beispiele und Entwicklungen sollen dem nun folgen.

5.2. Beispiele für OSS auf eingebetteten Systemen

Obwohl die Zahl der mit Linux betriebenen PDAs leider gering ist, scheint sich im restlichen *Embedded*-Bereich eine ganz andere Entwicklung abzuzeichnen. Dort ist der Anteil von Linux-Geräten inzwischen so hoch, dass Linux als erste Wahl für den *Embedded*-Bereich gelten kann (LinuxDevices.com 2004).

So läuft Linux inzwischen nicht nur auf Handys³¹ und Navigationssystemen, es gibt auch ganze Linux-Computer, die komplett auf eine CompactFlash-Karte passen. LinuxDevices.com ist wohl das bekannteste und umfassendste Portal für eingebettete Systeme mit Linux. Von der Fernbedienung bis zum Router finden sich dort fast alle mit Linux betriebenen Geräte.

Da eingebetteten Systemen meistens keine großen Hardware-Ressourcen zur Verfügung stehen, macht es Sinn, den Kernel und die verwendeten Tools daran anzupassen und nicht benötigte Treiber und Programme zu entfernen. So gibt es jetzt z. B. auch

30 Das Airport-Extreme-Modul verwendet einen 802.11g-Chipsatz von Broadcom.

31 Motorola richtet den Mittelpunkt seiner Mobiltelefon-Strategie auf Linux aus: „There’s a need to move to a more open, flexible architecture for the majority of handsets and [to provide] a more open developer platform [. . .] The great thing [about Linux] is that some of the brightest minds are helping develop it“ (Andrew Till, zitiert bei Computer Business Review Online 2005).

von der Linux-Distribution Gentoo einen Ableger für eingebettete Systeme.³² Eine auf Sicherheit ausgerichtete *Embedded*-Distribution ist AMSEL.³³

Unter den bereits angesprochenen Problemen mit nicht vorhandenen Treibern leidet nicht nur Linux, auch die BSDs sind betroffen. Das bewog das OpenBSD-Projekt, eine Kampagne für die freie Bereitstellung von Firmware zu starten, da die benötigten Laderoutinen für die Firmware vieler Geräte oft in geschlossenen Windows-Treibern integriert sind (Andrews 2004).

Die OpenIB-Allianz hat sich zum Ziel gesetzt, einen performanten und skalierbaren Linux-Stack für die InfiniBand-Architektur³⁴ zu entwickeln. Dafür hat sie im November 2004 eine Förderung vom amerikanischen Energieministerium erhalten (HPCwire 2004). Für diese Zukunftstechnologie wird Linux also gut gerüstet sein:

„the embedded market space [...] may not get as much notice, and might not be as in-your-face as the desktop, but it's clearly a growing area for Linux, too.“ (Linus Torvalds, zitiert bei Dudley 2004)

5.3. Embedded Linux Development

Der Beitrag von Joachim Henkel und Mark Tins über die offene Entwicklung bei kommerziellen Herstellern von Systemen mit eingebettetem Linux schließt unseren technisch angehauchten Artikel-Reigen und bildet einen guten Anschluss an das Kapitel „Ökonomie“. Die Autoren zeigen u. a. auf, dass Hersteller von Hardware unter bestimmten Marktbedingungen ihre Programmquellen doch offen legen können, ohne Nachteile im Wettbewerb befürchten zu müssen, sondern von dieser Offenlegung sogar noch profitieren werden.

Literatur

Andrews, J. (2004), 'Feature: OpenBSD Works To Open Wireless Chipsets', <http://kerneltrap.org/node/4118>.

Computer Business Review Online (2005), 'Motorola promises greater Linux phone focus', Artikel vom 21. Januar, http://www.cbronline.com/article_news.asp?guid=65BFD343-7E33-4DB0-8F3E-F7F2E6EAF712.

Diedrich, O. und Kuri, J. (2004), 'Das Allerwelts-Unix: 10 Jahre NetBSD', heise online <http://www.heise.de/newsticker/meldung/35544/>.

Dudley, B. (2004), 'Q&A: Linus Torvalds, inventor of Linux', The Seattle Times http://seattletimes.nwsourc.com/cgi-bin/PrintStory.pl?document_id=2002059632&zsection_id=268448455&slug=linus11&date=20041011.

HPCwire (2004), 'OpenIB Alliance Members Win DOE Grant', **13**(45B). LIVEwire Edition <http://www.tgc.com/hpcwire/hpcwireWWW/04/1110/108788.html>.

³² Gentoo Embedded <http://www.gentoo.org/proj/en/base/embedded/index.xml>

³³ AMSEL (Advanced Modular Secure Embedded Linux) <http://www.amselinux.de/>

³⁴ InfiniBand <http://infiniband.sourceforge.net/>

- IDC (2004), 'The Linux Marketplace – Moving From Niche to Mainstream',
http://www.osdl.org/docs/linux_market_overview.pdf. OSDL Pressemitteilung:
http://www.osdl.org/newsroom/press_releases/2004/2004_12_15_beaverton.html.
- Ihlenfeld, J. (2004a), 'Interview: NX – die Revolution des Netzwerk-Computing?',
Golem IT-News <http://www.golem.de/0408/33026.html>.
- Ihlenfeld, J. (2004b), 'Setzt die EU künftig auf Dateien im OpenOffice.org-Format?',
Golem IT-News <http://www.golem.de/0409/33806.html>.
- LWN.net (2004), 'Kernel Summit: Development process', LWN.net
<http://lwn.net/Articles/94386/>.
- Lindner, M. (2004a), 'Intel Centrino Treiber 1.0.0', Pro-Linux
<http://www.pro-linux.de/news/2004/7502.html>.
- Lindner, M. (2004b), 'Pranger für Hardware-Hersteller?', Pro-Linux
<http://www.pro-linux.de/news/2004/7203.html>.
- Lindner, M. (2005), 'Das Linux-Jahr 2004 – Ein kleiner Rückblick', Pro-Linux
<http://www.pro-linux.de/berichte/rueckblick2004.html>.
- LinuxDevices.com (2004), 'Linux Now Top Choice Of Embedded Developers',
<http://www.linuxdevices.com/news/NS2744182736.html>.
- MacCormack, A., Rusnak, J. und Baldwin, C. (2004), 'Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code',
Harvard Business School Working Paper Number 05–016,
<http://opensource.mit.edu/papers/maccormackrusnakbaldwin.pdf>.
- Mozilla Organization (2004), 'Mozilla Foundation Announces More Open, Scriptable Plugins',
Pressemitteilung <http://www.mozilla.org/press/mozilla-2004-06-30.html>.
- Naval Historical Center (1999), 'Rear Admiral Grace Murray Hopper, USNR, (1906–1992)',
<http://www.history.navy.mil/photos/pers-us/uspers-h/g-hoppr.htm>. Department of the NAVY, Photo #:NH 96566-KN (Color).
- Spolsky, J. (2004), 'How Microsoft Lost the API War',
<http://www.joelonsoftware.com/articles/APIWar.html>.
- Torvalds, L. (2004a), 'Linus on documenting patch provenance', LWN.net
<http://lwn.net/Articles/86436/>.
- Torvalds, L. (2004b), 'Linus on kernel management style', LWN.net
<http://lwn.net/Articles/105375/>.
- Tuomi, I. (2004), 'Evolution of the Linux Credits file: Methodological challenges and reference data for Open Source research', *First Monday* 9(6).
http://firstmonday.org/issues/issue9_6/tuomi/index.html.
- Ziegler, P.-M. (2004), 'Start von OpenSolaris verzögert sich', heise online
<http://www.heise.de/newsticker/meldung/52953/>.