

Free Software. Free People.

Text für eine Mini-Broschüre im Rahmen der „Free People“-Kampagne der
Österreichischen HochschülerInnenschaft zum Thema „Freie Software“,
www.oeh.ac.at/free

Wien, 2004

Mag. Leonhard Dobusch
Teistlergutstraße 1a
4040 Linz
leonhard.dobusch@reflex.at

Free Software. Free People.

„Free Software is not a matter of free beer, it's a matter of free speech”

(Richard Stallman, Free Software Foundation)

Warum beschäftigt sich die Österreichische HochschülerInnenschaft (ÖH) mit einem Thema wie „Software“? Und warum bezieht sie gleichzeitig eindeutig Stellung für freie bzw. Open-Source-Software? Auf diese Fragen gibt es gleich mehrere Antworten:

- Software spielt in der Wissensgesellschaft eine immer größere Rolle: Neben dem offensichtlichen Einsatz in Computern finden sich in nahezu jedem technischen Gerät auch Mikrochips und dazugehörige Software. Je umfassender der Einfluss von Software auf unser Leben ist, desto entscheidender und mächtiger wird auch die Kontrolle über Software. Und so macht es sehr wohl einen Unterschied, ob die Kontrolle zentralisiert in einem Monopol-Konzern (wie bei herkömmlicher Software) ist, oder dezentralisiert in der Hand tausender unabhängiger EntwicklerInnen (wie bei Open-Source-Software).
- Gerade Studierende haben mit ihrer Entscheidung für oder gegen freie Software einen wesentlichen Einfluss darauf, ob sich Open-Source-Software letztendlich gegen monopolistische Alternativen durchsetzen kann: Hochschulen waren und sind die wichtigsten Quellen für Fortschritt und Weiterentwicklung im Bereich von Software und auch sämtliche größeren Open-Source-Software-Projekte entstammen dem Hochschulbereich (Linux, die Free-Software-Bewegung, das auf dem offen TCP/IP-Standard basierende Internet usw.)
- Wie schon das Zitat von Richard Stallman, einem der Begründer von Open-Source-Software, andeuten soll, geht es bei freier Software wie Linux nicht so sehr darum, dass sie gratis verfügbar ist. Kernfrage hinter freier Software ist die nach dem freien Zugang zum Quellcode, dem freien Zugang zu diesem Wissen und damit zumindest die theoretische Möglichkeit, das Funktionieren der technischen Geräte zu verstehen und beherrschen, die ohnehin das Leben des/der Einzelnen beherrschen. Die Entscheidung für oder gegen freie Software ist also auch eine politische Entscheidung.

Ein Vergleich von Open-Source-Software wie Linux und herkömmlicher Software wie Windows & Co artet – zumindest hat es für Laien oftmals den Anschein – bisweilen in einen regelrechten Glaubenskonflikt aus. Doch ist die Entscheidung für oder gegen den Einsatz von Open-Source-Software keineswegs eine reine Glaubensentscheidung. Und spätestens seit ganze Städte wie München ihre EDV auf Open-Source-Software umstellen und sogar das deutsche Bundeswirtschaftsministerium eine offizielle Broschüre für den Einsatz von freier Software in kleinen und mittelständischen Unternehmen herausgegeben hat¹, wird den VerfechterInnen von Open-Source-Software mehr und mehr Gehör geschenkt.

Stabilität und Sicherheit als Vorteil

Gerade auch für Studierende, die meist nicht die finanziellen Ressourcen aufbringen können, sich proprietäre Software (auf legalem Wege) anzuschaffen, stellt Open-Source-Software eine sinnvolle Alternative dar. Nicht nur, weil Open-Source-Software prinzipiell kostenlos zur Verfügung steht, sondern auch weil sie in den Bereichen Funktionsumfang, Stabilität und Datensicherheit herkömmliche Software in der Regel in den Schatten stellt. Und obwohl auch Open-Source-Software mittlerweile bereits sehr einfach zu bedienen ist, haftet ihr oftmals noch das Image an, schwierig und umständlich zu bedienen zu sein.

Worin unterscheiden sich nun Open-Source-Software und herkömmliche, sogenannte „proprietäre“ Software? Bei proprietärer Software wie Microsofts „Windows XP“ oder „Office“ bleiben sowohl das Ergebnis der Programmierarbeit in Form des Quellcodes als auch die Arbeitsprozesse, -fortschritte und Zwischenergebnisse nicht offen zugänglich. Vervielfältigung, Weiterverbreitung und Modifizierung sind untersagt und Eigentümer der Software bleibt der/die HerstellerIn, der/die das UrheberInnenrecht und die vollständige Kontrolle über das Produkt hat.

Was macht Open-Source-Software aus?

Im Gegensatz dazu nennt www.opensource.org die wesentlichen (formalen) Eigenschaften von Open-Source-Software:

- Jeder hat das Recht, die Software nach eigenem Ermessen zu nutzen.
- Der Quelltext muss jedem Benutzer offengelegt werden oder es muß auf eine frei zugängliche Stelle verwiesen werden, wo er erhältlich ist.
- Der Benutzer hat das Recht, die Software zu modifizieren und in modifizierter Form

¹ oss-broschüre.berlios.de/kontakt.php

weiterzuverteilen.

- Die Lizenz darf niemanden im Verkauf oder der Weitergabe der Software in Form einer Softwarezusammenstellung einschränken.

Open-Source-Software-Entwicklung gleicht unter diesen Voraussetzungen in vielen Punkten einem wissenschaftlichen Prozess, was DiBona et al.² mit der Formulierung „Science, after all, is ultimately an Open Source enterprise“ sehr schön formuliert haben. Mit den Prozessstufen Entwicklung/Entdeckung, Dokumentation und Validierung/Rechtfertigung in Form eines öffentlichen Dialogs wird die prinzipielle Nachvollziehbarkeit und Überprüfbarkeit des Quellcodes ermöglicht. Erst das Teilen der gewonnen Erkenntnisse mit anderen ForscherInnen macht den wissenschaftlichen Forschungsprozess so effektiv und ermöglicht, auf den Erkenntnissen anderer die eigenen Entwicklungen voranzutreiben ohne das Rad ständig zwei-, drei- oder x-mal neu erfinden zu müssen.

Das gleiche gilt nun für Open-Source-Software-Entwicklung: Eben weil jedeR Interessierte zu jeder Zeit Einsicht in den Quellcode nehmen kann, müssen gute und wenig fehleranfällige Programmteile nicht neu erfunden werden, sondern können einfach übernommen werden. Wie eine Theorie, die sich bewährt hat, kommen so bewährte Programmier-Algorithmen immer wieder zum Einsatz. Und gleichzeitig sind die Programme fehlerresistenter, weil ständig in den verschiedensten Anwendungsumgebungen nach Fehlern gesucht werden kann und wird, während bei herkömmlicher Software die Fehlersuche nur in sehr zeitlich- und ressourcenmäßig beschränkten Beta-Test-Phasen abläuft.

Die Anfänge von Open-Source-Software

Die Ursprünge der Open-Source-Bewegung liegen bereits in den 60er Jahren des 20. Jahrhunderts, als Software regelmäßig freien Zusatz zur (teuren) Hardware darstellte. Die Entwicklung hin zur ausschließlichen kostenpflichtigen Lizenzierung von Software in Verbindung mit Unzulänglichkeiten insbesondere im Bereich der Hardware-Unterstützung führte in der Folge zu ersten Open-Source-Software-Projekten. Eine große Rolle spielte in diesem Zusammenhang Richard Stallman mit der Gründung der Free Software Foundation (FSF) und der Entwicklung der General Public License (GPL), der damit einen institutionellen und rechtlichen Rahmen für Entwicklung und Vertrieb von Open-Source-Software schuf.

² DiBona/Ockman/Stone (1999): Open Sources: Voices of the Open Source Revolution.

Open-Source-Software-Lizenzen

Denn Lizenzen bilden auch bei Open-Source-Software – so wie bei herkömmlicher Software – den rechtlichen Rahmen der Nutzung. Wie bei proprietärer Software regeln sie die Nutzungsrechte am Programm und am Quellcode. Bei Open-Source-Software-Lizenzen steht aber im Gegensatz zu proprietären Lizenzen der Schutz der weiteren freien Verfügbarkeit des Sourcecodes im Vordergrund. Die verschiedenen Lizenzierungstypen erlauben aber bisweilen stark verschiedene Freiheiten im Umgang mit dem Quellcode. GPL (General Public Licence) zwingt beispielsweise bei Verwendung von Quellcode dieser Lizenzierungsform auch zur Freigabe des damit neu erstellten, gesamten Programms und verunmöglicht somit die Integration von GPL-Quellcode in proprietäre Software-Projekte. BSD-Lizenzen (Berkley Software Distribution) im Gegensatz dazu überlassen es weitestgehend dem Urheber des Quellcodes, welche Einschränkungen er für die Benutzung des Quellcodes auferlegen möchte und erlaubt beispielsweise oftmals den Einsatz von BSD-Quellcode in proprietärer Software (Prominentestes Beispiel hierfür ist Apples Betriebssystem MacOS X, das auf einem BSD-Unix-Kernel basiert.)

Was kann Open-Source-Software?

Mehr als 30 Jahre nach den Anfängen von Open-Source-Software im Zeitalter der Großrechner hat die Entwicklung und Verbreitung von Open-Source-Software durch das Internet einen neuen Höhepunkt erreicht: Nie war es einfacher, eine derart große Zahl an Menschen an einem gemeinsamen Software-Projekt teilhaben zu lassen, wie das beispielsweise bei Linux der Fall ist. Aber Linux ist nur ein Beispiel, das gerade wegen seiner Konkurrenz zu Windows im Bereich der Betriebssysteme für die größten Schlagzeilen sorgt. Mittlerweile gibt es Open-Source-Software-Anwendungen für nahezu jeden Bereich: „Open Office“ ersetzt mit Tabellenkalkulation, Schreib- und Präsentationsprogramm Microsoft Office, Mozilla ersetzt den Internet-Explorer und so weiter.

Auch schafft Open-Source-Software teilweise Kunststücke, die sich mit herkömmlicher Software niemals umsetzen lassen würden: Mit „Knoppix“ gibt es beispielsweise eine Linux-Version, die sich direkt von CD starten lässt, keine Installation erfordert und gleichzeitig noch Platz für unzählige Programme zu jedem erdenklichen Zweck findet: Neben dem bereits

erwähnten Open Office noch Multimedia-Programme, verschiedene Internet-Browser, wissenschaftliche Programme, Editoren und Spiele.³

Einen Versuch wert?

Gerade Varianten wie Knoppix zeigen einen weiteren Vorteil und damit gleichzeitig eine Chance von Open-Source-Software, sich auch gegen die geballte Monopol-Marktmacht auch eines Microsoft-Windows zu behaupten: Die unglaubliche Flexibilität und die Möglichkeit für jedeN EinzelneN, sein/ihre kreativen Ideen einzubringen. Und mit einsteigerInnenfreundlichen Varianten wie Knoppix, die das einfache Ausprobieren ohne Gefahr für Leib und Rechner ermöglichen, ist freie Software erst recht einen Versuch wert.

³ Eine kostenlose Knoppix-CD kann unter www.oeh.ac.at/free bestellt werden. Solange der Vorrat reicht.

Kurz-Info: Kampf (mancher) Konzerne gegen Open-Source-Software

Während zahlreiche große Unternehmen und Konzerne (z.B. IBM) bereits selbst stark auf Open-Source-Software setzen, um sich aus der Abhängigkeit von Microsoft und deren Preisdiktat zu befreien, setzen andere (allen voran eben Microsoft) wiederum auf den Einsatz vor allem der juristischen Keule, um gegen Open-Source-Software vorzugehen.

Die Vorgehensweise basiert auf einer Doppelstrategie: Einerseits soll die Verletzung von Copyright-Verletzungen im Linux-Quelltext nachgewiesen werden und dadurch Linux-NutzerInnen gezwungen werden, doch für das Produkt zu zahlen. Andererseits wird weltweit Lobbying für die Einführung und Ausweitung von Software-Patenten gemacht. Diese Software-Patente stellen mit Sicherheit die größte Bedrohung von Open-Source-Software dar, das durch sie selbst dann für einzelne patentierte Algorithmen und Programmteile gezahlt werden müsste, wenn sie völlig neu geschrieben und entwickelt würden. Mehr zur Gefahr durch ein Ausweiten der Patentrechte findet sich unter www.oeh.ac.at/free.

Neben diesen juristischen Mitteln gegen Open-Source-Software wurden und werden auch zahlreiche Studien von Software-Unternehmen in Auftrag gegeben und finanziert, die die angebliche Unrentabilität von Open-Source-Software „beweisen“, sich aber regelmäßig durch klar voreingenommen Vorgehensweise und falsche Grundannahmen auszeichnen.

Kurz-Überblick: Unterschiede zwischen Klassischer und Open-Source-Software:⁴

	Klassische Softwareprodukte	Open-Source-Entwicklung
Zugänglichkeit des Quellcodes	Nur für Entwickler / Unternehmensmitglieder	Offen für jeden Interessierten
Größe der Entwicklergruppe	Kleine Entwicklungsteams	Kleine bis sehr große Entwicklergemeinden (z.B. über 3000 Entwickler bei Linux)
Steuerung des Entwicklungsprozesses	Zentral, Top-Down	Dezentral, Bottom-Up
Zeitpunkt des Release	Erst nach umfassenden Tests	Möglichst unmittelbar nach Programmierung
Prominente Beispiele	Windows, Office (Word, Excel), Internet-Explorer, Filemaker	Linux, Open Office, Mozilla, Apache Webserver

⁴ Nach Schneidewind et. al. (2002): Mythos Linux? Zur Übertragbarkeit der Koordinations- und Anreizmechanismen der Linux-Entwicklung auf Unternehmen, In: Zeitschrift Führung + Organisation, 71/4, S 228