

Freie Software

GFS im Fach Gemeinschaftskunde

Nicolas Bellm

4. Januar - 6. Januar 2005

Der Inhalt dieses Dokuments steht unter der GNU-Lizenz für freie Dokumentation
<http://www.gnu.org/copyleft/fdl.html>

Inhaltsverzeichnis

1	Definition von freier Software	3
1.1	Vier Freiheiten	3
1.2	Beispiel für freie Software	3
2	Geschichte	4
2.1	Ursprung von freier Software	4
2.2	Richard Stallman und das KI-Labor des MIT	4
2.3	Das GNU-Projekt	5
2.4	GNU/Linux	5
2.5	Streit zwischen Open Source und Freier Software	5
2.6	Freie Software heute	6
3	Kommerzielle Nutzung von freier Software	7
4	Gefahren	8
4.1	Allgemeine Gefahren	8
4.2	Softwarepatente	8

1 Definition von freier Software

Viele Menschen sind der Meinung, daß man unter Freier Software kostenlose Software versteht, was zum einen daher rührt, daß das Wort „frei“ im deutschen zwei Bedeutungen hat¹, und zum anderen, daß Freie Software in den allermeisten Fällen tatsächlich kostenlos ist. Diese Interpretation von freier Software ist jedoch nicht richtig. Es ist durchaus auch möglich, für freie Software Geld zu verlangen. Der Begriff „Freie Software“ sagt also nichts über den Preis der Software aus, sondern die Rechte, die durch die Lizenz der Software garantiert werden. Auf diese Rechte auch Freiheiten will ich im folgenden näher eingehen.

1.1 Vier Freiheiten

Freie Software ist Software, deren Lizenz jedem folgende vier Freiheiten garantiert:

1. Die Freiheit, das Programm für jeden Zweck zu benutzen (Freiheit 0).
2. Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann (Freiheit 1). Der Zugang zum Quellcode ist dafür Voraussetzung.
3. Die Freiheit, Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann (Freiheit 2).
4. Die Freiheit, das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann (Freiheit 3). Der Zugang zum Quellcode ist dafür Voraussetzung. [2]

Software, deren Lizenz diese Freiheiten nicht garantiert, nennt man *proprietäre Software*.

1.2 Beispiel für freie Software

Das bekannteste Beispiel für freie Software dürfte wohl *Linux* sein. Immer größere Verbreitung findet auch die freie Alternative für den Internet Explorer, *Mozilla Firefox*. Häufig vergessen, aber aus dem Schüleralltag nicht mehr wegzudenken ist, *MediaWiki*, die Software, die hinter Wikipedia, der freien Enzyklopädie steckt.

Aber der größte Teil der freien Software wird dem Laien eher unbekannt sein, wie z. B. die freie Bildverarbeitungssoftware *The GIMP* oder die beiden Desktopumgebungen *KDE* und *GNOME*. Man darf auch nicht das freie Officepaket, *OpenOffice.org*, eine brauchbare Alternative für Microsoft Office, vergessen. Wer aber professionelle Texte unter Linux aber auch unter Windows gestalten will, kommt wohl um das Textsatzsystem *LaTeX* nicht herum.

Kurzum, es gibt für fast jedes proprietäre Programm eine freie Alternative, die ich jetzt unmöglich alle aufzählen könnte.

¹1. frei wie in „Freibier“, 2. frei wie in „freie Meinungsäußerung“

2 Geschichte

2.1 Ursprung von freier Software

Am Anfang war alle Software frei und quelloffen. Selbst in der kommerziellen Computerwelt waren die Quellen verfügbar. Die Computerfirmen konzentrierten sich nämlich auf den Verkauf von Hardware und die Software war einfach dabei, um die Hardware benutzen zu können. Es gab also keine Unterscheidung zwischen Hard- und Softwarefirmen. So konnte man auch beim Marktführer IBM Hardware auch nur in einem Paket mit Software, Peripheriegeräte, Wartung und Schulung kaufen. Aber auf Druck des US-Justizministerium mußte IBM diese Bündelung aufgeben. Diese Entkoppelung schuf dann schließlich die Voraussetzung für eine eigenständige Softwareindustrie. Die Programme waren dann auch nicht mehr im Quellcode erhältlich.

2.2 Richard Stallman und das KI-Labor des MIT

Auch *Richard M. Stallman*, der Anfang der 70er-Jahre im KI-Labor des MIT arbeitete, erlebte diese Veränderung mit. Er war am Anfang davon fasziniert, daß jeder, der ein Problem hatte, die Software, die ein anderer geschrieben hatte, benutzen durfte.

„Ich hatte in den 70er-Jahren das Glück, Teil einer Gemeinschaft zu sein, in der die Menschen Software miteinander teilten. Wir entwickelten Software und wann immer jemand ein interessantes Programm geschrieben hatte, wurde es weitergegeben. [...] So arbeitete einer nach dem anderen, um die Software zu verbessern und weiterzuentwickeln. Man konnte in dieser Gemeinschaft immer eine zumindest passive Mitarbeit eines jeden erwarten. Sie mochten zwar nicht bereit sein, ihre Arbeit zu unterbrechen, um stundenlang etwas für dich zu tun, aber das, was sie bereits erledigt hatten, konntest du gerne benutzen.“ (Richard Stallman) [1, S. 219]

Aber nach und nach zerfiel diese Gemeinschaft. Die nachfolgende Generation hatte keine Probleme mehr damit, daß Programme ein Eigentum einer bestimmten Person ist und brachten in ihren Programmen einen Copyrightvermerk an. Die zunehmende Kommerzialisierung war dann das Ende dieser Gemeinschaft. Viele Mitarbeiter des KI-Labors wurden von Firmen abgeworben und mußten Vertraulichkeitsvereinbarungen (*Nondisclosure Agreements*) unterzeichnen und der Quellcode war geschlossen.

Aufgrund dieser Erfahrungen gründete er das GNU-Projekt, auf das ich jetzt zu sprechen komme.

2.3 Das GNU-Projekt

Richard Stallman versuchte nun diesen alten *sharing spirit* wiederherzustellen. Er startete 1984 das GNU-Projekt (GNU's not Unix), dessen Ziel es war, ein Betriebssystem zu schaffen, das funktional äquivalent zu *Unix*¹ ist, aber komplett frei ist. Er gab seine Stelle beim MIT auf, damit das GNU-Projekt nicht das Eigentum des MIT ist. So fing er an einen Compiler, GNU C-Compiler (GCC), und einen Editor, GNU Emacs, schreiben. Stallman hat Emacs auf einem ftp-Server zur Verfügung gestellt, aber auch angeboten, daß die Leute, die daran interessiert waren, ihm ein Band in einem frankierten Rückumschlag schickten, auf das er den Emacs überspielte. Außerdem verkaufte er, da er ja kein eigenes Einkommen mehr hatte, ein Band mit Emacs für 150 Dollar. Hier wird auch deutlich, daß mit „frei“ nicht die Freiheit in Bezug auf den Preis gemeint ist. Als das Interesse am Emacs wuchs, gründete er 1985 die *Free Software Foundation* (FSF), die sich von nun an mit dem Verkauf von Emacs und später auch anderer GNU-Software beschäftigte. Aus dem Verkauf von Software und Handbüchern und auch aus Spenden finanzierte die FSF die Entwickler, die sich damit beschäftigen, die für die Betriebsumgebung notwendigen Programme zu schreiben. Die Programme des GNU-Projekts bekamen kein Copyright, sondern ein Copyleft („Copyleft, all rights reversed“). Daraus entwickelte sich dann die GNU General Public License (GPL), die besagt, daß alle Änderungen der Gemeinschaft wieder verfügbar gemacht werden müssen.

2.4 GNU/Linux

Mit der Zeit wurden alle Teile des Betriebssystems programmiert. Es fehlte nur noch der *Kernel* (Kern), der zentrale Bestandteil des Betriebssystems. Dieser Kernel sollte GNU Hurd werden. Da dieser Kern aber nicht rechtzeitig fertig wurde, nahm man als Ersatz bis GNU Hurd fertig ist, einfach das 1991 von Linus Torvalds entwickelte Linux und nannte das Ganze dann GNU/Linux. GNU Hurd ist immer noch nicht fertig, aber die erste stabile Version soll „in Kürze“ erscheinen.

2.5 Streit zwischen Open Source und Freier Software

Bis 1998 verwendete man ausschließlich den Begriff Freie Software. Doch dann ist plötzlich in dem von Eric S. Raymond geschriebenen Essay „*The Cathedral and the Bazaar*“ der Begriff *Open-Source-Software* aufgetaucht. Der Grund dafür ist nicht nur, daß der Begriff „frei“, wie ich schon im ersten Kapitel erwähnt habe, zweideutig ist, sondern auch, daß der Begriff „frei“ für kommerzielle Firmen wenig schmackhaft klingt. In dem selben Jahr veröffentlichte nämlich Netscape den Quellcode seines Navigators, aus dem dann der Mozilla wurde.

Doch auch der Begriff „Open Source“ ist auch nicht besonders treffend, weil ja die Quelloffenheit zwar eine notwendige Bedingung für Freie Software ist, aber keine hinreichende Bedingung. Es gab nämlich immer mehr Programme, dessen Quelltext zwar offen ist, aber keine Modifikationen zulassen. Diese Firmen schmückten sich zwar mit dem Label „Open

¹Ein von AT&T entwickeltes Betriebssystem

Source“, jedoch handelt es sich keinesfalls um Freie Software. Aus diesem Grund spricht man auch heute wieder mehr und mehr von „Freier Software“ und deshalb nannte ich auch das Thema meiner GFS „Freie Software“ und nicht „Open-Source-Software“.

2.6 Freie Software heute

Immer mehr Menschen nutzen heutzutage freie Software, einerseits, weil sie einfach besser ist, als proprietäre Software, wie z. B. Windows aber auch, weil sie von der Philosophie begeistert sind, die hinter freie Software steht. Viele Menschen widmen ihrer Freizeit der freien Software und arbeiten selbst an einem Softwareprojekt mit. Viele Leute sind auch Angestellte bei Firmen, wie SuSE oder Redhat und lassen sich bezahlen für die Programmierung von freier Software. Darauf werde ich aber noch im nächsten Kapitel näher eingehen. Es gibt auch immer mehr Verwaltungen von Städten und Gemeinden, z. B. die Verwaltung von München, die ihre Software von Windows auf GNU/Linux umstellen.

3 Kommerzielle Nutzung von freier Software

Schon im vorherigen Kapitel wurde deutlich, daß Freie Software eine kommerzielle Nutzung nicht ausschließt. Man kann also durchaus sich die Arbeit, die Software auf CD zu brennen, oder das Leisten von Support bezahlen lassen. Es haben sich auch inzwischen viele Firmen, sogenannte Distributoren, angesiedelt, die sich dafür bezahlen lassen, verschiedene freie Software zusammenzustellen und diese dann in einem Paket vertreiben. Als Beispiel kann man hier SuSE, RedHat und Mandrake nennen.

Diese Firmen verdienen nicht nur an dem Vertrieb dieser Softwarepakete, sondern sie leisten auch Installationssupport oder kümmern sich auch um die Administration der Software.

4 Gefahren

4.1 Allgemeine Gefahren

Aber es gibt auch viele Situationen, die für freie Software eine Gefahr darstellen. Viele Firmen veröffentlichen die Spezifikationen ihrer Hardware nicht und so können auch keine freien Treiber für GNU/Linux entwickelt werden oder nur durch große Anstrengungen (z. B. Reverse-Engineering, was aber nicht legal ist). Eine weitere Gefahr für freie Software sind auch die Softwarepatente, auf die ich jetzt im weiteren eingehen werde.

4.2 Softwarepatente

Normale Patente beziehen sich auf Produktionsabläufe und konkrete Anwendungen, die laut den Naturgesetzen theoretisch möglich sind. Softwarepatente beziehen sich aber auf Konzepte bzw. Ideen. So wird z. B. keine bestimmte Mausefalle patentiert sondern jedes Mittel zum Fangen von Nagetieren. Zurück zur Software: Es wird z. B. nicht Microsoft Word, das ja schon durch das Urheberrecht geschützt ist, patentiert, sondern jegliche Programme zum Erstellen von Texten. Auf diese Weise wurden schon viele zur Softwareentwicklung notwendige Funktionen patentiert, wie z. B. Karteireiter in Dialogfeldern, Fortschrittsbalken und sogar die für das Programmieren unumgängliche Wenn-Abfrage. Die großen Firmen wie Microsoft, IBM haben viele solcher Softwarepatente und sprechen sich ab, diese Patente nicht gegenseitig auszuspielen. Aber mittelständische Unternehmen und Entwickler von freier Software bleiben außen vor. Solche Softwarepatente sind in der USA schon legal und es wurden teilweise schon Firmen verklagt, wegen Verletzung von Patentansprüchen. In der EU sind sie zwar noch nicht legal, was sich durch die Entscheidung des Europäischen Rates bald ändern wird, aber sie sind dennoch zu tausenden schon angemeldet. Dann wird es nicht mehr möglich sein, Software zu entwickeln, ohne Patentansprüche von irgendwelchen Firmen zu verletzen, man bewegt sich als Softwareentwickler also auf einem Minenfeld.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, daß ich die Arbeit selbstständig und nur unter Zuhilfenahme der ausgewiesenen Hilfsmittel angefertigt habe.

Bad Schönborn, den 16. Januar 2005

Nicolas Bellm

Literaturverzeichnis

- [1] Volker Grassmuck: *Freie Software Zwischen Privat- und Gemeineigentum*. Bundeszentrale für politische Bildung, Bonn, 2002.
- [2] <http://www.gnu.org/philosophy/free-sw.de.html>, 6.1.2005
- [3] <http://www.gnu.org/gnu/thegnuproject.de.html>, 6.1.2005
- [4] Wikipedia (6.1.2005): freie Software, Open Source, Softwarepatent