Freie Software – wie alles begann
Das GNU-Projekt und Linux

Seminararbeit im Rahmen des Seminars
„Wissen in der modernen Gesellschaft“
Universität Leipzig, Sommersemester 2009

Verfasser: Marcus Eichhorn (9921592)
Betreuer: Prof. Hans-Gert Gräbe

17.06.2009

Inhaltsverzeichnis

0. Vorwort

„Ohne Moos nichts los!“ – Nach diesem Prinzip hat sich seit Ende der 70er Jahre die Herstellung von Software zu einer eigenen Branche entwickelt. Allerdings ist aufgrund dieses Prinzips eine Monopolbildung fast nicht zu verhindern. Dass der Marktführer alles dafür tut, um die Funktionsweise seiner Software vor den Anwendern und Kongruenten zu verbergen ist selbstverständlich klar. Umgesetzt wird dies, indem man den „Quellcode“, also das lesbare Programm, zurückhält und nicht veröffentlicht. Auf diese Art konnte Microsoft, welches eines der ersten reinen Softwareunternehmen war, seine Monopolstellung erst im Bereich der Betriebssysteme errichten und später auch auf Teile der Anwendungssoftware ausweiten. Einerseits profitieren die Anwender von dieser Marktstellung Microsofts, weil sie zueinander kompatible und funktionierende Programme erhalten. Aber andererseits leiden die Anwender darunter, da Monopolisten zu hohe Preise verlangen und die Qualität der Software vernachlässigen können. Dieser Fakt wird noch verstärkt, weil der Wechsel zu einem anderen Anbieter meist mit hohen Kosten für neue Hardware, Schulungen der Mitarbeiter und eventuelle Support-Kosten verbunden ist.

Aufgrund dieser Tatsachen ist eine Bewegung entstanden, welche ihre Anfänge auch schon in den frühen 70er Jahren hat: die Freie Software. Ihr liegt das Prinzip zu Grunde, dass Software beliebig bearbeitet und weitergegeben werden kann. Dies ist rein kommerziell gesehen nicht sehr reizvoll. Trotz dieses Nachteiles existiert bis heute eine Vielzahl an Freier Software.

Diese Arbeit geht auf die Entwicklung von Freier Software etwas genauer ein. Im ersten Kapitel wird definiert, was man unter Freier Software versteht. Anschließend in Kapitel zwei erhält man einen kurzen Überblick über die Geschichte der Softwareentwicklung von ihren Anfängen bis in die frühen 70er Jahre. Das dritte Kapitel beschäftigt sich speziell mit dem GNU-Projekt als Paradebeispiel für Freie Software. Hierbei werden die wichtigsten Stationen dieses Projektes bis hin zu GNU/Linux dargestellt.

1. Was ist „Freie Software“?

„Freie Software ist Software, die mit der Erlaubnis für jeden verbunden ist, sie zu benutzen, zu kopieren und zu verbreiten, entweder unverändert oder verändert, entweder gratis oder gegen ein Entgelt. Im Besonderen bedeutet das, dass der Quellcode verfügbar sein muss. Wenn es kein Quelltext ist, ist es keine Software. Dies ist eine vereinfachte Definition.“ [1]

Der Begriff „Freie Software“ ist eng verbunden mit der Person des Richard Stallman. Er hat durch seine Erfahrungen in den Anfangszeiten seiner Karriere am Massachusetts Institute of Technology und später durch sein GNU-Projekt diesen Begriff geprägt wie kein anderer. Frei heißt dabei nicht kostenlos. Stallman prägte den Ausspruch: „Free as in freedom, not as in free beer“ (Frei wie in Freiheit, nicht wie in Freibier.). Freie Software ist dabei deutlich zu unterscheiden von dem Begriff Freeware. Richard Stallman und die Free Software Foundation definieren eine Software als frei, wenn sie die 4 Freiheiten erfüllt:

1. Das Programm darf zu jedem Zweck ausgeführt werden.
2. Das Programm darf studiert und verändert werden.
3. Das Programm darf kopiert werden.
4. Das Programm darf verbessert und so verbreitet werden, um damit einen Nutzen für die Gemeinschaft zu erzeugen.

Für die ersten 3 Freiheiten ist der Zugang zu den Quellcode vorausgesetzt, da sonst keine Veränderungen vorgenommen werden können, beziehungsweise dies nur sehr schwer möglich ist. Erfüllt ein Programm diese Freiheiten nicht, so bezeichnet man es als unfreie oder proprietäre Software.

[1] Kategorien freier und unfreier Software, Richard Stallman, URL: http://www.gnu.org/philosophy/categories.de.html, Stand: 14. Juni 2009

2. Die Geschichte der Software-Entwicklung bis Ende der 70er Jahre

Der Beginn der Software-Entwicklung ist eigentlich schon zu Beginn des 18. Jahrhunderts zu erkennen. Zu dieser Zeit gab es die ersten Ideen zur Entwicklung von automatischen Rechenmaschinen. Diese Ideen kann man sozusagen als die primitiven Vorläufer der heutigen Technik bezeichnen. Die ersten Ausführungen für solche „Analytical Engines“ brachte C. Babbage (1792 – 1871) schon im Jahr 1841. Diese Maschine sollte nach seiner Meinung mittels Lochkarten gesteuert werden und bedingte Sprünge mittels „IF-Anweisungen“ durchführen können. Die Ergebnisausgabe sollte dann durch eine direkte Ausgabe über Stahlstempel erfolgen. Allerdings kam es nie zum Bau dieser Maschine.

Diese Idee nahm sich etwas später Ada Byron (1815 – 1852) an, welche auch bekannt war unter den Namen Ada Augusta Byron, Ada King oder Countess of Lovelace. Sie übersetzte die Artikel von Luigi Menabre über die „Analytical Engine“ und entwickelte die ersten Programme für sie. Im Jahr 1997 wurde nach ihr die Programmiersprache ADA benannt. Im Jahr 1944 wurde dann die erste richtige Software entwickelt. Sie befähigte einen Rechner Lochkarten zu lesen und darauf enthaltene Informationen zu verarbeiten.

In den 50er und 60er Jahren begann der Softwarepionier IBM damit eigene Computer zu entwickeln, für diese Rechner wurde natürlich auch passende Software benötigt und entwickelt. Zu dieser Zeit der Software-Entwicklung war alle Software noch quellenoffen und frei. Die Hardwarehersteller lieferten Software als „Gebrauchsanweisung“ zu ihren Rechnern dazu. Weitere Software wurde dann von den Anwendern selber geschrieben. Für diese Anwender war es beim Wechsel einer Rechnergeneration zur nächsten immer ein Problem ihre Software an die neuen Gegebenheiten anzupassen. Wegen diesem Problem taten sich im Jahr 1955 eine Gruppe von Nutzern des IBM 704 zusammen, um die Software des Vorgängers IBM

701 an den neuen 704 anzupassen.[2] Diese Gruppe von Nutzern nannte sich IBM Share[3]. Share bestand aus Mitarbeitern verschiedener Firmen aus der Gegend um Los Angeles. Diese unterschiedlichen Firmen standen zwar teilweise in Konkurrenz, aber der Vorteil der sich aus Erfahrungen und Wissen miteinander zu teilen ergab, war größer als die Konkurrenz. Während IBM an höheren Programmiersprachen arbeitete, entwickelte die ständig größer werdende Share-Gemeinde im Laufe der Zeit eine sehr große und umfangreiche Bibliothek an Routinen für den IBM 704. Was auch ein Grund dafür war, dass sich dieser Rechnertyp relativ schnell verbreitete.[4] Deshalb wurde bald darauf Share wichtiger für die Unterstützung des IBM 704 als der IBM-eigene Support. Eine weitere solche User Group neben IBMs Share war DECs DECUS.

Quelltexte fand man zu dieser Zeit in Zeitschriften, wie zum Beispiel in der Algorithmen Rubrik der Communications of the ACM oder in weiteren Amateurfunkzeitschriften. Anwender entnahmen diese Quelltexte den Zeitschriften, verbesserten und entwickelten sie weiter, um sie dann wieder zu veröffentlichen. Programmierer wurden zu dieser Zeit nicht für ein Programm bezahlt, sondern für das Programmieren. An Universitäten war es noch bis in die 70er Jahre üblich dort entwickelte Software frei zu verbreiten.

Die führenden Hardware-Firmen Anfang der 60er Jahre waren IBM, DEC, Hewlett-Packard und Data General. Bei dem Marktführer IBM konnten Kunden Hardware nur in einem Paket kaufen, welches Software, Peripheriegeräte, Wartung und Schulung einschloss. Im Jahr 1969 begann IBM diese Bündelung aufzugeben. Grund dafür war ein durch das US-Justizministerium eingeleitetes Kartellverfahren. Viele Programme standen jedoch weiterhin frei zur Verfügung, da IBM sie als gemeinfrei erachtete. Außerdem entstanden viele dieser Programme zum Großteil in Nutzergemeinschaften und somit hätte IBM auch kaum einen Urheberrechtsanspruch darauf gehabt.

Aufgrund dieser Entkopplung entstand trotz der Konkurrenz durch Freie Software eine erste eigenständige Softwareindustrie. Allerdings waren die ersten, Anfang der 70er

[2] Vgl. Ceruzzi 2000, S. 87 f.

[3] SHARE stand für Society to Help Avoid Redundant Effort. Vgl. Ceruzzi 2000, S. 330

[4] Vgl. Ceruzzi 2000, S. 88

entstanden Softwarefirmen, meist noch Satelliten von Hardwareherstellern. Software wurde somit erstmals zu einer Ware. In den 1976 verfassten „Open Letter to Fellow Hobbyists“ beklagte Bill Gates unter anderem, dass die meisten Fellows[5] seine Software stehlen würden und somit das Schreiben von besserer Software behindern würden. Dieser Brief und das kurz zuvor begangene Vergehen von Bill Gates, wobei er fast von der Harvard Universität geflogen wäre, weil er öffentliche Mittel missbrauchte, um kommerzielle Software zu schreiben, spiegelt den neuen Zeitgeist wieder, welcher Software als Eigentum sieht. Nachdem Bill Gates von der Harvard Universität gezwungen wurde seine Software frei zur Verfügung zu stellen, verließ er diese und gründete kurz darauf zusammen mit Paul Allen Microsoft „und damit das Zeitalter von Software aus der Schachtel.“[6]

Ende der 70er Jahre brachte IBM seinen IBM-PC (den IBM 5150) auf den Markt. Dieser PC bestand aus Komponenten verschiedenster Hersteller, zum Beispiel kamen die CPU 8088 von Intel und das Betriebssystem DOS von Microsoft. Aufgrund dieser Bauweise entstanden viele Clone, welche ebenfalls mit DOS betrieben werden konnten. Dies brachte den Durchbruch für Microsoft und somit auch für kommerzielle Software.

[5] Fellow bezeichnet „im britischen akademischen Sprachgebrauch (…) einen Gelehrten, der von einer Hochschule oder Universität zum Zwecke der Forschung und/oder Lehre finanziell unterstützt wird“ Quelle: http://de.wikipedia.org/wiki/Fellow Stand 13.Juni 2009

[6] Vgl. Grassmuck 2004, S. 203

3. Das GNU-Projekt

3.1. Richard Stallman

Wenn wir über Freie Software sprechen, dann kommen wir an einer Person nicht vorbei. Richard Stallman ist die Person, die wie keine andere für Freie Software steht. Geboren wurde Stallman am 19. März 1953 in Manhattan, New York City. Seine akademische Laufbahn begann in Harvard, wo er als Experte für Assemblersprachen, Betriebssysteme und Texteditoren arbeitete. Anfang der 70er Jahre wechselte er in das Labor für Künstliche Intelligenz des Massachusetts Institute of Technology (MIT) als Systemprogrammierer. Der Grund für den Wechsel von Harvard nach Massachusetts ist wohl in der hackerfreundlichen Atmosphäre zu suchen, für welches das MIT legendär war. Der Begriff Hacker beschreibt nicht wie heute einen Computerkriminellen. Hacker waren zu dieser Zeit Technikenthusiasten, welche umfangreiche technische, vor allem computertechnische Grundlagenkenntnisse besaßen.

Mitte der 50er Jahre besaß das MIT nur einen IBM 704, welcher Lochkartenstapel verarbeiten konnte. Im Jahr 1959 bekam das MIT einen transistorbetriebenen Rechner mit Kathodenstrahlmonitor. Nun war es möglich an einem PC zu sitzen, während dieser ein Programm durchrechnete und im Anschluss konnte man auf der Stelle neue Anweisungen in die Tastatur hacken.

Während seiner Zeit beim MIT lernte Stallman die Hackerlegenden Richard Greenblatt und Bill Gosper kennen und tauchte in eine Kultur des freien Wissensaustausches ein. Seine Zeit am MIT beschrieb Stallmann während einer Konferenz vom 16. bis 17. Juli 1999 im Haus der Kulturen der Welt in Berlin:

„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."[7]

Neben seiner Arbeit als Systementwickler und am Editor „Emacs", machte Stallman noch einen Magna cum laude-Abschluss in Physik an der Harvard Universität. Der Emacs Editor war Stallmans bekanntestes Werk während dieser Zeit. Er basierte auf einem Lisp Dialekt und war beliebig erweiterbar. Wegen seiner offenen Architektur wurden für ihn viele Erweiterungen und Verbesserungen geschrieben. Dies war auch möglich, da Stallman mit dem Programm in genau diesen Stil verfuhr, wie es am MIT üblich war, er gab das Programm frei an jeden weiter unter der Bedingung, dass alle Erweiterungen der Emacs-Community mitgeteilt werden sollten.

Ende der 70er, Anfang der 80er Jahre musste Stallman jedoch den Untergang dieses Hackerparadieses erleben. Grund dafür war, dass das MIT-Rechenzentrum Passwörter einführte. Das MIT war während dieser Zeit an das ARPAnet angeschlossen und das US-Verteidigungsministerium verlangte die Einführung von Passwörtern, da es sonst damit drohte das MIT vom ARPAnet abzuhängen. Der Grund für das Vorgehen des US-Verteidigungsministeriums war, dass es in einem zunehmend wichtiger werdenden Netz nicht ging, dass jeder ohne Berechtigung auf einen militärischen Rechner zugreifen könne. Stallman verachtete Passwörter, da sie seiner Meinung nach den freien Austausch von Wissen behindern. Zuerst führte Stallman noch das „leere" Passwort ein, doch der Druck wurde immer größer, so dass nach und nach viele Hacker das MIT verließen, um in Computerfirmen zu arbeiten oder eigene zu gründen. Die nachfolgende Generation sah das offene System nicht als Chance. Auch wenn die neuen Programmierer am MIT weiter eifrig Programme schrieben, tauchte doch beim Start von Programmen immer öfters ein Copyright-Vermerk auf. Sie waren nicht in die Hackerethik hineingewachsen und empfanden es nicht als falsch Eigentum an einem Programm zu haben.

[7] Vgl. Grassmuck 2004, S. 219

Gegen diese Form der Schließung kämpft Stallman bis heute. Im Jahr 1983 zitiert ihn Steven Levy wie folgt:

„Ich finde nicht, dass Software Eigentum sein sollte, weil mit dieser Praxis die Menschlichkeit im Ganzen sabotiert wird. Sie verhindert, dass die Menschen aus einem Programm den maximalen Nutzen ziehen.“ [8]

Seit 1975 arbeitet Richard Greenblatt zusammen mit andern Hackern des MIT Labors an einer Lisp-Maschine. Dabei handelte es sich um einen Rechner, welcher auf die ressourcenhungrige Programmiersprache Lisp zugeschnitten war. Während dieser Arbeit entstanden insgesamt 32 Lisp-Rechner. Diese Technologie wollte Greenblatt als Element im Kampf zwischen Amerika und Japan, um die Führung im Bereich der Künstlichen Intelligenz verwenden. Greenblatt sah dieses Potential am besten durch die kommerzielle Verbreitung verwirklicht. Diese Idee stieß natürlich unter den Hackerkollegen auf Wiederstand. Aus dieser Auseinandersetzung ging erst LISP-Machine Incorporated (LMI) und etwas später die hochkapitalisierte Firma Symbolics hervor. Diese beiden Firmen warben fast alle verbliebenen Hacker vom MIT ab. Diese zunehmende Kommerzialisierung bedeutete das Ende der Hackerkultur. Stallman blieb als letzter Überlebender dieser Kultur übrig. Anfang der 80er war fast alle Software proprietär und man konnte sie sogar seit 1982 zum Patent anmelden.

3.2. Das GNU-Projekt startet!

Anfang der 80er Jahre hatte Stallman ein Schlüsselerlebnis. Das MIT besaß einen Xerox-Netzwerkdrucker. Es kam regelmäßig vor, dass jemand einen Druckerauftrag abschickte und dann in den Nachbarraum ging und feststellen musste, dass es einen Papierstau gab oder das Papier alle war. Stallman wollte darauf die Software um eine Funktion erweitern, welche den Druckerstatus direkt am Arbeitsplatz anzeigt. Er fand jemanden bei Xerox, welcher den Quellcode besaß. Dieser wollte, beziehungsweise

[8] Vgl. Grassmuck 2004, S. 220

konnte den Quellcode nicht herausgeben, da er sich zur Nichtweitergabe verpflichtet hatte. Also was sieht man an diesen Ereignis? Es existiert ein praktisches Problem. Dieses könnte gelöst werden, indem man eine bestehende Software um eine Funktion erweitert. Früher hätte man den Autor der Software um den Quellcode gebeten und hätte diesen fortgeschrieben. Die Technologie wäre somit für alle Betroffenen nützlich geworden. Doch jetzt stand vor dem Quellcode und vor einer Kooperation eine Mauer namens „geistigen Eigentums“. Eine Lösung dieses Problems wäre immer noch möglich. Man könnte die Firma darum bitten diese Funktion in einem späteren Update einzubinden, aber dies ist langwierig und unsicher. Eine weitere Möglichkeit wäre die Dekompilierung der Software, dies wäre allerdings sehr mühsam, zeitraubend und nur in engen Grenzen legal. [9]

Stallman fragte sich nach diesem Erlebnis, was man dagegen tun könnte und was für Voraussetzungen man schaffen müsste, um wieder eine Gemeinschaft, wie sie in den frühen Jahren des MIT existierte, zu erschaffen. Stallman kam daraufhin zu dem Schluss, dass man zuerst ein gemeinsames und freies Betriebssystem benötigt. So startete Stallman 1984 das GNU-Projekt. Das rekursive Akronym GNU steht dabei für „GNU´s not Unix“. Ziel war es also ein Betriebssystem zu schreiben, dieses sollte:

- funktional äquivalent zu Unix sein,
- keine einzige Zeile geschützten Code enthalten und
- in freier Kooperation weiterentwickelt werden können.

Doch warum sollte dieses neue Betriebssystem äquivalent zu Unix sein? Für Unix sprachen drei einfache Punkte:

- erstens Unix hatte sich bewährt,
- zweitens Unix ist portabel und
- drittens für Unix gab es bereits eine aktive weltweite Gemeinde, so dass ein Umstieg auf GNU leicht möglich wäre.

[9] Vgl. Grassmuck 2004, S.222

Doch bevor man an ein eigenes Betriebssystem denken konnte, mussten erst einmal alle benötigten Komponenten entwickelt werden. Diese sollten dann später zu einem Betriebssystem zusammengefasst werden können.

Stallman kündigte daraufhin seinen Job beim MIT, da ansonsten alle seine Entwicklungen der Universität gehören würden und diese könnte somit die Betriebsbedingungen bestimmen. Stallman wollte verhindern, dass er erneut eine Menge Arbeit investierte und dann später hilflos zusehen müsse, wie das MIT ein proprietäres Softwarepaket daraus machen würde. Der Leiter des Labors für Künstliche Intelligenz des MIT lud Stallman nach seiner Kündigung jedoch ein, das Labor weiterhin zu nutzen, wofür Stallman sehr dankbar war.

1983 kündigte Stallman sein Projekt erstmals in verschiedenen Unix-Newsgroups an und lud zur Mitarbeit ein. Stallman begann zunächst noch alleine mit dem GNU C-Compiler und dem GNU Emacs Editor. Auf der Konferenz in Berlin 1999 beschrieb er seine Situation und das Projekt wie folgt:

„So fingen wir an, die Komponenten dieses Systems zu schreiben. Die Struktur dieses Systems besteht aus vielen einzelnen Programmen, die miteinander kommunizieren; und es war dokumentiert, so dass man nachvollziehen konnte, wie die Schnittstellen zwischen diesen Programmen aussahen. Wir konnten nach und nach alle Bestandteile als Ersatz für die entsprechenden Teile eines Unix-Systems schreiben und schließlich testen. Wenn alle Teile ersetzt sind, fügt man diese zusammen und erhält so das vollständige System. Und das taten wir. Es war eine sehr dezentralisierte Vorgehensweise, was zu einer eher amorphen und dezentralisierten Community von Freiwilligen, die überall in der Welt hauptsächlich über E-mail kommunizierten, passte.“ [10]

Zu Begin des Projektes stellte Stallman seine Programme, wie zum Beispiel Emacs auf einem anonymen ftp-Server zur Verfügung. Alternativ bot er an, ihm ein leeres

[10] Vgl. Grassmuck 2004, S.223

Datenband sowie einen frankierten Rückumschlag zuzusenden. Da Stallman während dieser Zeit kein Einkommen hatte, stellte er etwas später noch ein Band mit Emacs für 150 US-$ zur Verfügung. Das Interesse an den Emacs-Bändern wurde kurz darauf so groß, dass Stallman 1985 die gemeinnützige Free Software Foundation (FSF) gründete. Die FSF vertrieb die Emacs-Bänder sowie weiter Software und dazugehörige Handbücher und Schulungen. Neben diesen Verkäufen finanziert sich die FSF jedoch hauptsächlich durch Geld- und Sachspenden. Dieses Kapital verwendet die FSF, um Freie Software zu fördern. Außerdem wurden mit diesem Geld Entwickler bezahlt, um fehlende Komponenten für GNU zu programmieren. Seit 1990 beschäftigt sich die FSF hauptsächlich mit rechtlichen Belangen.

3.3. Das GNU-Manifest

1985 verfasste Richard Stallman das GNU-Manifest. In ihm rief er zur Teilnahme und Unterstützung seines Projektes auf. Neben der Frage was GNU überhaupt ist, erläutert Stallman in seiner Schrift auch die Vorzüge sowie die Wichtigkeit seines GNU-Projektes. Desweiteren gibt er Hinweise wie man das Projekt unterstützen kann. Doch der größte Teil des Manifestes beschäftigt sich mit Einwänden gegen GNU und ihre Endkräftigung durch Stallman. In dieser Schrift begründet Stallman auch die Philosophie der Freien Software:

„Ich denke, dass die goldene Regel vorschreibt: Wenn ich ein Programm gut finde, muss ich es mit anderen Leuten, die es gut finden, teilen.“ [11]

Umgekehrt kann man also sagen, wer die Nutzungsmöglichkeit eines Programmes einschränkt, um damit Geld zu verdienen, verwendet zerstörerische Mittel. Wenn alle sich so verhielten, würde die Menschheit durch diese Zerstörung ärmer. Das Bestreben

[11] Vgl. Grassmuck 2004, S.224

für seine Kreativität belohnt zu werden, rechtfertigt nicht, dass man sein Wissen der Welt vorenthält. Stallman kontert der Kritik, welche die fehlende Entlohnung für die Kreativität betrifft, indem er verschiedene Möglichkeiten nennt trotz Freier Software Geld zu verdienen. Diese Möglichkeiten wurden in den 80er Jahren erprobt und setzten sich in den 90ern durch. Er nennt unter anderem die Möglichkeiten mittels Vertrieb von Dokumentationen, Support für unerfahrene Nutzer und Hardwarehersteller, Schulungen, Spenden und Steuermitteln Geld zu verdienen.

„*Das absolute Ziel* [Stallmans] *ist es, frei Software anzubieten, um alle Aufgaben zu erledigen, die ein Benutzer erledigen will - und somit*“ [12] geschützte Software überflüssig zu machen.
Das GNU-Projekt ist somit mehr als eine Sammelstelle für verschiedene freie Programme, das GNU-Projekt wird von einer Vision eines vollständigen Systems geleitet. Nach und nach wurden alle nötigen Programme für ein Betriebssystem in eine Liste eingetragen und Schritt für Schritt abgearbeitet.

3.4. Die GNU General Public License (GPL)

Um diese Vision, welche im GNU-Manifest verdeutlicht wurde zu sichern, wurde von der Free Software Foundation die GNU General Public License (GPL) herausgegeben. Sie wurde von Stallman in Zusammenarbeit mit seinem juristischen Berater Jerry Cohen geschrieben und existiert heute mittlerweile in der dritten Version. Die grundlegende Intention der GPL lautet:

„Die meisten Lizenzen für Software und andere nutzbaren Werke sind daraufhin entworfen worden, Ihnen die Freiheit zu nehmen, die Werke mit anderen zu teilen und

[12] Überblick über das GNU Projekt, übersetzt von Matthias Blazejak, URL: http://www.gnu.org/gnu/gnu-history.de.html , Stand 14.Juni 2009

zu verändern. Im Gegensatz dazu soll Ihnen die GNU General Public License die Freiheit garantieren, alle Versionen eines Programms zu teilen und zu verändern. Sie soll sicherstellen, dass die Software für alle ihre Benutzer frei bleibt. Wir, die Free Software Foundation, nutzen die GNU General Public License für den größten Teil unserer Software; sie gilt außerdem für jedes andere Werk, dessen Autoren es auf diese Weise freigegeben haben. Auch Sie können diese Lizenz auf Ihre Programme anwenden."[13]

Diese GPL kann von jedem Entwickler für seine Software verwendet werden. Sie gewährt den Nutzern:

1. den Zugang zum Quellcode,
2. die Freiheit, die Software zu kopieren und weiterzugeben,
3. die Freiheit das Programm zu ändern und
4. die Freiheit das veränderte Programm unter denselben Bedingungen weiter zu verbreiten.

Die Freiheiten eins bis drei bezeichnet man auch als die individuellen Freiheiten, da sie die Freiheit eines jeden betrifft, welcher die Software verwendet. Die vierte Freiheit nennt man wegen ihrem „Zwang", dass die Software frei bleibt auch kollektive Freiheit, da sie die gesamte Gemeinschaft betrifft.

Diese Philosophie schreibt jedoch nicht vor, dass die Weitergabe kostenlos zu geschehen hat. Für Dienstleistungen, wie Zusammenstellung, Produktion und Vertrieb von CD-ROMs und Handbüchern sowie Schulung und Support ist die Erhebung von Gebühren ausdrücklich erlaubt. Jedoch für die Software selber darf kein Geld verlangt werden.

Mit der Herausgabe der GPL beginnt die FSF zusammen mit ihren Advokaten, wie zum Beispiel dem New Yorker Rechtsprofessor Eben Moglen, sich in den Bereichen Copyright und Vertragsrecht zu engagieren.

[13] Die GNU General Public License, übersetzt von Peter Gerwinski, URL: http://www.gnu.de/documents/gpl.de.html, Stand 14.Juni 2009

3.5. GNU wird zu GNU/Linux

Bis 1990 war das GNU-System nahezu vollständig, das einzige was noch fehlte war ein Kernel. Die Wahl viel hierbei auf den Mikrokernansatz. Favorisiert war dafür der Mikrokernel Match, welcher von der Carnegie Mellon University entwickelt und später dann von der University of Utah übernommen wurde. Über den Match Kernel sollten alle anderen Betriebssystemkomponenten als eine Sammlung von Servern, mit den Namen HURD liegen. Doch das Debugging solcher Server war schwerer als erwartet.

1991 kam der finnische Student Linus Torvalds dem GNU-Projekt mit dem Linux Kernel zuvor. Torvalds nutzte den Minix, wobei es sich um ein Unix-Derivat für 286er PCs handelt. Minix wurde von Andrew Tanenbaum entwickelt, welcher Professor für Informatik an der Freien Universität Amsterdam ist. Tanenbaum entwickelte Minix speziell für Lehrzwecke. Die Entwicklung von Minix konnte man in der Usenet-Newsgroup comp.os.minix mitverfolgen. So gab es Ende der 80er Jahre bereits mehr als 50.000 Minix-Anwender. Torvalds war einer dieser Anwender und ausgehend von Minix begann er einen unixartigen Betriebssystemkern zu schreiben. Er veröffentlichte den Quelltext seiner Arbeit im Netz und lud zur Mitarbeit ein. So eine Art Zusammenarbeit gab es bis dahin noch nicht. Eine Zusammenarbeit ohne eine zentrale steuernde Einheit war revolutionär. Linux ist bis heute ein Paradebeispiel für ein komplexes Softwareprojekt, an welchem tausende von Leuten auf der ganzen Welt verteilt arbeiten.

Durch diese effektive Arbeitsweise dauerte es auch nur knapp ein Jahr bis der erste stabile Linux-Kernel 0.12 im Jahr 1992 erschien. Die Linux Community nutze für ihren Kernel das GNU-System. Ebenfalls übernahm sie die GPL, welche die Freiheit der Software sicherte. Desweitern wurde von XFree86 die grafische Benutzeroberfläche übernommen. Somit erschien im März 1994 die GNU/Linux Version 1.0. Um die Aufspaltung des Projektes in verschiedene Richtungen zu verhindern, etablierte sich ein Core-Team, in welchen Torvalds das letzte Wort darüber hat, was in den Kern aufgenommen wird. Desweitern bildeten sich verschiedene Standardisierungsgremien, wie im August 1993 die Linux File System Standard Group, das Linux Documentation

Project (LDP) und zur Sicherstellung der Kompatibilität der Betriebsumgebungen in verschiedenen Distributionen, die Linux Standard Base (LSB).

Die Verbreitung von GNU/Linux erfolge zunächst noch über das Internet, doch spätestens ab 1993 auch auf Disketten und kurz darauf auf CD-ROM. Ab 1993 stellten kommerzielle Anbieter wie SuSE, Caldera und Yggdrasil Distributionen mit dem neusten Kernel, Tools und Anwendungen zusammen.

3.6. GNU/Linux heute

GNU-Linux ist heute das nach Windows und MacOS am weitesten verbreitete Betriebssystem. Es hat einen Marktanteil von etwa 1%. Der Anteil der GNU/Linux Nutzer wächst ständig. So zeigt die Abbildung 1, dass sich der Anteil der GNU/Linux User von 1995 bis 1998 verachtfacht hat.

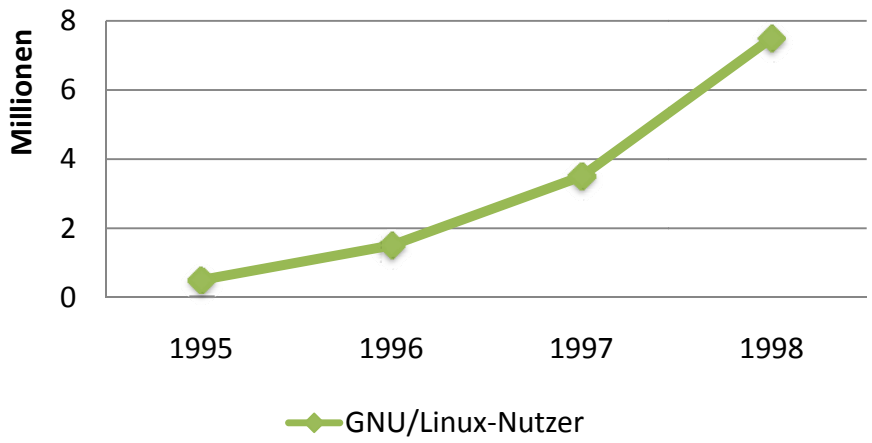

Abbildung 1 Linux-Nutzer Chart, die Daten beruhen auf Grassmuck 2004, S.228

Heute stehen für GNU/Linux mehr als 1000 Anwendungen zur Verfügung, der Großteil davon sogar kostenlos. Es existieren viele verschiedene grafische Benutzeroberflächen, wie zum Beispiel KDE (K Desktop Environment) und Gnome (GNU Network Object Model Environment) mit jeweils verschiedenen Windows-Managern. Auch der anfänglich kritisierte Support wurde durch tausende engagierte GNU/Linux User

wiederlegt, welche oft schon innerhalb von Stunden mit Hilfe von Patches Probleme beheben. GNU/Linux hat besonders nach der Einführung der Distribution ubuntu einen weiteren Schub bekommen, da der Umstieg von Windows auf ubuntu sehr einfach möglich ist. Großanwender, welche heutzutage komplett auf GNU/Linux setzten, sind unter anderem Edeka, Sixt, Debis und Ikea. Der Durchbruch in Hollywood gelang GNU/Linux durch den Film „Titanic“, dessen Computergrafik auf einem GNU/Linux Cluster gerechnet wurde. Seit der Linux-World Expo im März 1999, welche ein Art Volljährigkeitsfeier darstellte, hat Linux sogar eine eigene Industriemesse. GNU/Linux findet heutzutage auch große Anwendung in öffentlichen Einrichtungen. Der öffentliche Sektor ist heutzutage eine treibende Kraft für Freie Software. Vorreiter in Deutschland ist die Stadt München, welche seit Herbst 2006 seine gesamte Verwaltung auf die GNU/Linux Distribution „LiMux“ umstellt und somit Microsoft den Rücken kehrt. *„Sie gelten als anschauliche Beispiele dafür, dass es ernsthafte Alternativen zu den von Redmond vorgegebenen Update-Zyklen gibt, dass sich Lizenzkosten sparen und klamme Haushaltskassen entlasten lassen.“* [14]

Andere Linux Distributionen finden unter anderem noch Einsatz:

- im Bundesamt für Finanzen (mit dem größten Linux-only Mainframe),
- im Bundesarchiv in Koblenz,
- bei der SPD-Bundestagsfraktion,
- im St. Marien-Krankenhaus in Siegen,
- im Max-Planck-Institut für Festkörperforschung Stuttgart,
- im Deutschen Krebsforschungszentrum (DKFZ),
- bei der IHK Bodensee-Oberschwaben und
- bei der Deutschen Bahn.

[14] Vgl. Open Source 2007, S.223

4. Nachwort

Freie Software ist ein immer wichtigerer und größer werdender Markt. Besonders in Zeiten der Wirtschaftskrise und knapper Kassen ist sie ein willkommenes Mittel, um Kosten zu sparen. München war nur der Vorreiter für die Verwendung von Freier Software in öffentlichen Einrichtungen. Auch weitere Ämter stellen auf Linux und Freie Software um oder planen dies zumindest, darunter unter anderem die Stadtverwaltungen von Bonn und Paris. Auch wenn der Anteil von Linux-Nutzer weltweit bei nur knapp einen Prozent liegt, so wird er doch ständig mehr. Neben der Verwendung von Linux und Freier Software in Industriestaaten, leistet Linux einen immer größeren Anteil zur Entwicklungshilfe, vor allem in Afrika. Das Freiburger Open Source Netzwerk startet das Projekt „Linux4Africa“, um vor allem an Schulen in Afrika einen Zugang zu Computern und dem Internet zu ermöglichen. Die dabei verwendeten Programme sind ausschließlich Freie Software.

Somit hat eine Person namens Richard Stallman, mit seiner Vision von Freier Software und den daraus entstandenen GNU-Projekt und der General Public License eine ganze Bewegung ins Leben gerufen. Diese Bewegung hat nicht nur im Bereich der Software-Entwicklung grundlegende Veränderungen hervorgerufen, sondern vor allem aufgezeigt, dass es neben proprietärer Software eine gute und auch alltagstaugliche Alternative gibt.

5. Quellen

Internetverzeichnis:

Deutsche Übersetzung von Peter Gerwinski; 2007; „ GNU General Public License“; URL: http://www.gnu.de/documents/gpl.de.html; Stand: 14. Juni 2009

Deutsche Übersetzung von Matthias Blazejak; 2007; „Überblick über das GNU Projekt“; URL: http://www.gnu.org/gnu/gnu-history.de.html; Stand: 14.Juni 2009

Joachim Korb; 2006; „Geschichte der Softwareprogrammierung“; URL: http://www.perspektive89.com/2006/12/21/geschichte_der_softwareprogrammierung_freie_software_fur_freiheit_und_gerechtigkeit; Stand 13. Juni .2009

Richard Stallman; 1985; „Das GNU Manifest“; deutsche Übersetzung von Peter Gerwinski; URL: http://www.gnu.de/documents/manifesto.de.html; Stand: 14. Juni 2009

Richard Stallman; 1994; „Kategorien freier und unfreier Software“; URL: http://www.gnu.org/philosophy/categories.de.html; Stand: 14. Juni 2009

Literaturverzeichnis:

Ceruzzi 2000 - Paul E. Ceruzzi; Massachusetts 2000; “A history of modern computing”; 2. Auflage; The MIT Press.

Grassmuck 2004 - Volker Grassmuck; Bonn 2004; „Freie Software – Zwischen Privat und Gemeineigentum“; 2. Auflage; Bundeszentrale für politische Bildung.

Open Source 2007 - Bernd Lutterbeck, Matthias Bärwolff, Robert A. Gehring (Hrg.); Berlin 2007; “Open Source Jahrbuch 2007”; Lehmanns Media.