

„Information wants to be free“ – oder: Wem gehört freie Software? Version 2.0

Christian Imhorst

7. November 2004

Inhaltsverzeichnis

1	Einleitung	2
2	Free Software	5
3	Linux	7
4	GNU	9
5	GPL	10
6	Literatur	14

„Anarchie und Quellcode“ ist mein Artikel für das „Open Source Jahrbuch 2005“ und unter anderem auf meiner Homepage <http://www.imhorst.net> zu finden. Der Artikel ist mehr oder weniger ein Abstract von meinem Buch „Die Anarchie der Hacker“. Unter [christian<at>imhorst.net](mailto:christian@imhorst.net) kann man mir schreiben.

„...information wants to be free, because the cost of getting it out is getting lower and lower all the time.“¹

1 Einleitung

Wenn es um Open Source geht, machen viele Journalisten aus dem freien Betriebssystem GNU/Linux häufig einen direkten Gegenspieler von Microsoft. In den Medien wird diese Konkurrenz dann als ein Kampf zwischen Bill Gates, dem Präsidenten von Microsoft, und dem Linux-Erfinder Linus Torvalds dargestellt. Dabei ist freie Software mehr als nur eine andere Methode, Software zu entwickeln. Sie ist auch mehr als eine Auseinandersetzung zwischen dem erfolgreichen Geschäftsmann Bill Gates und fanatischen Hackern, die ihrer Utopie nachhängen. Denn freie Software wurde nicht in Konkurrenz zu Microsoft entwickelt, sondern aus dem Wunsch heraus, die eigene Software wirklich zu besitzen und keiner strengen Lizenz von Microsoft oder anderen Herstellern proprietärer Software mehr unterworfen zu sein.

Für die Programmierer und Anwender freier Software hat proprietäre keinen Nutzen. So wurde Linux von Linus Torvalds entwickelt, weil ihm das Betriebssystem Minix, das er benutzte, nicht ausreichte. Ein großer Nachteil von Minix war nämlich, dass der Quellcode 150 US-Dollar kosten sollte. Für den Preis durfte er noch nicht einmal Änderungen am Quellcode vornehmen, sondern nur hineingucken. Das war dann doch zuviel Geld für einen armen finnischen Informatikstudenten. Aber worum geht es genau bei Open Source, dem offen gelegten Quellcode?

„Technisch gesehen handelt es sich beim Quell- oder Source Code um eine Liste von Computerbefehlen, die in einer für Menschen lesbaren Programmiersprache abgefaßt sind. Diese Liste wird mit Hilfe von Compilern in Bit-Zeichenketten umgewandelt, die Binär- oder Maschinencode genannt werden und nur von Computern (und einigen sehr, sehr begabten Menschen) verstanden werden können.“²

Der Quellcode eines einfachen Programms, das nur den Satz „Hallo Welt!“ auf den Bildschirm schreiben soll, sieht in der Programmiersprache C folgendermaßen aus:

```
/* Datei: "Hallo.c" */  
int main (int argc, char** argv)  
{
```

¹ Stewart Brand auf der ersten „Hackers Conference“ 1984. Vgl. Stewart Brand, *The Media Lab: Inventing the future at MIT*. New York, 1987.

² Peter Wayner, *Kostenlos und überlegen*. Stuttgart, 2001, S. 127

```
printf("Hallo Welt!");  
return 0;  
}
```

Wenn wir unseren Quellcode in der Datei „Hallo.c“ mit einem Compiler in maschinenlesbaren Code übersetzen, erhalten wir ein ausführbares Programm. Solche Programme erkennt man in Windows meist an der Dateieindung EXE, was für executable, ausführbare Datei steht. In unserem Fall hieße die compilierte Datei dann „Hallo.exe“. Schaut man sich den Inhalt von „Hallo.exe“ mit einem Editor an, erhält man scheinbar nur Kauderwelsch. Der Computer weiß durch dieses Kauderwelsch, was er tun soll, nämlich „Hallo Welt!“ auf den Bildschirm schreiben. Will jemand, dass der Computer statt „Hallo Welt!“ den Satz „Carpe diem!“ auf den Bildschirm schreibt, ist das in diesem Kauderwelsch nicht zu ändern. Er braucht den Quellcode, um die Software an seine Bedürfnissen anzupassen, um „printf(„Hallo Welt!“)“, durch „printf(„Carpe diem!“)“ zu ersetzen. Er braucht dazu allerdings auch das Wissen, wie man das macht.

Linux wurde 1991 entwickelt, als Microsoft seinen Siegeszug über die heimischen Desktop-PCs noch nicht vollendet hatte. Dabei entstand Linux als Alternative zum Unix-Ableger Minix. Torvalds benutzte beim Programmieren Werkzeuge, die vom Namen her scheinbar in Konkurrenz zu einem anderen Betriebssystem entwickelt worden waren, die GNU-Tools. GNU ist ein rekursives Akronym, bei dem einer der Buchstaben für das Akronym selber steht, und bedeutet „GNUs Not Unix“. GNU entstand als Reaktion auf die Kommerzialisierung des Unix der amerikanischen Telefongesellschaft AT&T.

Richard Stallman, ein bärtiger Hacker am Massachusetts Institute of Technology, wollte mit GNU nicht nur ein freies Betriebssystem entwickeln, es ging ihm vielmehr um die Durchsetzung seiner politischen Überzeugung. Für ihn hat jede Software in einer doppelten Bedeutung frei zu sein: Zum einen soll der Quellcode eines Programms kostenlos und zum anderen der Allgemeinheit frei zur Verfügung gestellt werden, damit sie für jeden veränderbar ist und von allen geteilt werden kann. Damit soll nicht gesagt sein, dass freie Software nicht verkauft werden dürfe. Doch wenn sie verkauft wird, dann nur mit ihrem Quellcode. Seine Gegner werfen Stallman vor, dass er dadurch das geistige Eigentum abschaffen würde. Doch im Gegenteil, er will Software als geistiges Eigentum schützen, indem verhindert werden soll, dass sie zu proprietärer Software wird. Das politische Mittel für diesen Schutz ist die GNU General Public License (GPL), ein Vertrag, den Entwickler freier Software abschließen, und der besagt, dass jeder, der dieses Programm

benutzt, die gleichen Rechte hat, wie alle anderen auch. Es gibt nur eine Einschränkung: Der Quellcode muss für alle offen bleiben.

Die Idee des GNU-Projekts, dass Software frei sein soll, kommt aus den Anfangszeiten der Informatik. Damals war es genauso üblich, Resultate wissenschaftlichen Forschens zu veröffentlichen, wie in anderen wissenschaftlichen Disziplinen auch. Um diese Idee zu erhalten, mit allen Vorteilen, wie offene Standards, günstige Software und Eigentum an Software, hat sich die Freie-Software-Bewegung gebildet. Von ihr hat sich 1998, sehr zum Ärger Stallmans, die Open Source Initiative abgespalten, um neben der GPL weitere Lizenzen unter einem Markennamen zu vereinen. Der Haken an der Sache: Einige Lizenzen versprechen nur, dass sie quell-offen sind. Die Quellen dürfen aber nicht verändert werden.

Die Freie-Software-Bewegung tritt dagegen für die gemeinschaftliche Entwicklung von Software ein und will mit der GPL ein System freier Software schaffen. Dafür soll Software zum Gemeineigentum werden, damit sie letztendlich Privateigentum des Anwenders werden kann.

Die besondere Produktionsweise freier Software liegt an den Besonderheiten von Software selber: Software kann über das Internet weitergegeben werden, ohne dass ein Mangel entsteht. Dabei entstehen nur Entwicklungskosten. Die entwickelte Software ist sofort einsetzbar, sobald sie aus dem Internet heruntergeladen wird. Da bei der Verteilung von Software kein Mangel entsteht, eben weil sie beliebig reproduzierbar, das heißt, kopierbar ist, muss niemand von ihrem Gebrauch als Privateigentum aus geschlossen werden. Windows wird von Microsoft als proprietäre Software vertrieben. Privateigentümer von Windows ist Microsoft. Von dem Gebrauch von Windows sind alle solange ausgeschlossen, bis sie die Lizenzbedingungen von Microsoft anerkennen, die unter anderem beinhalten, dass die Software nicht weiter kopiert oder verändert werden darf. Nach der Anerkennung der Lizenz ist die Benutzung des Programms gestattet, mehr jedoch nicht. Freie Software, die unter die GPL gestellt ist, schließt niemanden von ihrem Gebrauch als Privateigentum aus. Allerdings schließt diese Lizenz aus, dass freie Software zu proprietärer wird. An einem Programm, das unter der GPL steht, können entscheidende Verbesserungen vorgenommen werden. Von diesen Verbesserungen können alle anderen ausgeschlossen werden, indem sie nicht veröffentlicht werden. Das macht Software zu Privateigentum. Aber sobald das Programm mit seinen Verbesserungen verkauft oder frei verteilt werden soll, muss der Quellcode offengelegt werden. Die GPL schließt dadurch aus, dass sich jemand freie Software nimmt, sie verbessert und dann unter einer neuen Lizenz weiter

verkauft. Ob das Programm nun verkauft oder verschenkt wird, es hat wieder unter der GPL zu stehen, und der Quellcode muss dabei sein.

2 Free Software

Bislang hat Microsoft jeden Konkurrenzkampf gewonnen. Entweder kaufen sie die Konkurrenz auf, oder schreiben ein ähnliches Programm, das sie dann mit ihrem populären Betriebssystem Windows vertreiben. Den Browserkrieg gegen Netscape gewann Microsoft, indem es seine ganze monopolistische Macht dazu einsetzte, mit seinem Betriebssystem Windows 95 gratis den Internet Explorer zu verteilen. Aber noch vor dem Browserkrieg wuchs in den Artificial Intelligence Labs, Computer Laboren an Universitäten, Computer Heimarbeitsplätzen, Internet-Firmen und Jugendzimmern der Welt eine Bedrohung für Microsoft heran, die sich nicht aufkaufen lässt. Miteinander verbunden über das Internet, zeigten sich Programmierer gegenseitig ihre Programme und luden jeden dazu ein, daran mitzuschreiben, Ideen einzubringen oder Programme zu testen und von Fehlern zu berichten. Auf diese Weise entstand unter anderem der Web-Server Apache. Einer der Entwickler von Apache, Brian Behlendorf, studierte in Berkeley und war Chef Entwickler von HotWired, dem ersten Online-Magazin. Damit HotWired über das Internet von Lesern auch gelesen werden konnte, brauchte es ein spezielles Programm, einen Web-Server, der die Verbindung vom Computer des Lesers mit dem Computer von HotWired regelte. Angetan von der Idee des HTTP, dem „Hyper Text Transport Protocol“, und der Software-Lizenz der Universität von Berkeley, schlug Behlendorf der Entwicklergruppe vor, Apache für alle frei verfügbar zu machen. Jeder konnte (und kann heute noch) an Apache mitschreiben – und die Entwicklergruppe wird von Hunderten Programmierern im Internet unterstützt – das Programm verändern und weitergeben. Diese Freiheit, mit dem Programm anzustellen, was man will, und dass es dazu noch umsonst ist, führte zu einer rasanten Verbreitung von Apache. Dazu schreibt Glyn Moody in *Rebel Code*: „Apache 1.0 [...] was released in December 1995. A couple of months later, a year after the Apache group had been formed, Apache became the most widely used Web server on the Internet, according to the survey produced by Netcraft. Since then, its market share has risen almost every month, and as of the year 2000 it outpaces Web servers from Netscape and Micosoft by a huge margin.“³

Zu Beginn des Anti-Trust-Verfahrens gegen Microsoft erschienen 1998 interne Memos von Vinod Valloppillil, einem Mitarbeiter des Unternehmens, die Halloween-Papiere hei-

³ Glyn Moody, *Rebel Code*. London, 2001, S. 129.

ßen. Eric Raymond bekam die Memos an Halloween zugespielt, weshalb er sie gleich so nannte. Raymond ist der selbsternannte *minister of propaganda*⁴ der Open Source Initiative. Die Echtheit der Papiere ist von Microsoft nie bestritten worden. In diesen Memos äußert Valloppillil nicht nur seine Sorge wegen der rasanten Ausbreitung von Apache, sondern von freier Software, oder Open Source Software (OSS), allgemein: „OSS poses a direct, short-term revenue and platform threat to Microsoft, particularly in server space. Additionally, the intrinsic parallelism and free idea exchange in OSS has benefits that are not replicable with our current licensing model and therefore present a long term developer mindshare threat.“⁵

Der Begriff Open Source wurde 1998 geprägt und als Markenname geschützt. Außerdem begann sich IBM für Open Source zu interessieren und stellte Investitionen von Milliarden US-Dollar in Linux und Apache in Aussicht.

Für die Öffentlichkeit wurde Open Source zur Konkurrenz von Microsoft, als das Unternehmen in seinem Anti-Trust-Verfahren von 1998 bis 2002 dazu gezwungen wurde, den Beweis anzutreten, dass es kein Monopol in der Software-Branche habe. Als Konkurrenten bemühten sie Apple, die nur kurz zuvor eine gewaltige Finanzspritze von Microsoft erhalten und einen Vertrag zur Zusammenarbeit unterschrieben hatten. Sie nannten BeOS, ein Betriebssystem, dessen Verbreitung nur einen geringen Marktanteil ausmacht, und sie nannten eben Open Source.

Die Freie-Software-Bewegung, bzw. die Open Source Initiative entwickeln Software, genau so wie Microsoft. Ihr Flaggschiff ist ein Betriebssystem, das nicht viel Ähnlichkeiten mit Windows aufweist. Sieht man von der Bedienung des Betriebssystems und seinen vielen Features ab, ist allein bei der Entwicklung von Linux vieles anders gelaufen. Die erste vorzeigbare Version von Linux wurde im Internet veröffentlicht, war sofort für alle zugänglich und wurde gleich diskutiert. Der Quellcode von Linux wird nicht von hoch bezahlten Software-Ingenieuren in Büros entwickelt, nicht als Geschäftsgeheimnis betrachtet und nicht durch umfangreiche Lizenzen geschützt, die dem User viel Geld kosten, noch bevor er es benutzen kann. Einer der wesentlichen Unterschiede zwischen Windows und GNU/Linux ist, dass GNU/Linux demjenigen gehört, der es sich aus dem Internet heruntergeladen hat. Er kann damit anstellen was er will, er darf nur keine proprietäre Software daraus machen. An Windows, mit der bezahlten Anerkennung der Lizenz, wird nur das Recht gekauft, das Programm benutzen zu dürfen. Windows selbst bleibt eine *black box*, bei der selbst der interessierte User nicht weiß, wie sie funktio-

4 Ebd., S. 152.

5 Vinod Valloppillil, *Halloween I*. URL: <http://www.opensource.org/halloween/halloween1.php> – Zugriff am 10.09.2004

niert. Mit GNU/Linux und freier Software im allgemeinen lässt sich kein Geld durch die Vergabe von Lizenzen verdienen, sondern nur mit Support- und Wartungsverträgen.

3 Linux

Torvalds begann 1991 mit dem Schreiben von Linux, weil er mit dem Betriebssystem für Studenten der Informatik, Minix, von Andrew Tanenbaum unzufrieden war. Zwar hatte Microsoft 1991 seinen Siegeszug über die Desktop-PCs dieser Welt mit MS-DOS und BASIC bereits angetreten, doch vor allem im wissenschaftlichen Bereich triumphierte immer noch Unix, das 1969 von Ken Thompson und Dennies Ritchie in den Bell-Laboratories der amerikanischen Telefongesellschaft AT&T entwickelt wurde. Minix war ein kleiner Ableger von Unix, und Tanenbaum, Professor für Informatik in Amsterdam, schrieb es ausschließlich für Forschungszwecke. Dass Linux als Alternative zu Minix entwickelt wurde, zeigt ein Posting von Torvalds in der Newsgroup comp.os.minix vom 25. August 1991:

„Hallo an alle, die dort draußen Minix verwenden – ich arbeite an einem (frei zugänglichen) Betriebssystem (nur so als Hobby, wird nicht groß und professionell wie GNU sein) für 386er (486er) AT-Kompatible. Die Sache ist seit April am köcheln und nimmt allmählich Formen an. Ich hätte gerne ein Feedback über die Dinge, die euch an Minix gefallen/nicht gefallen, da mein Betriebssystem gewisse Ähnlichkeiten dazu aufweist [...]“⁶

Einer der wesentlichen Unterschiede zu Minix, auch jenseits von technischen Erweiterungen, besteht in der „Freiheit“ des Betriebssystems. Das Software „frei“ ist, hat seit Stallman zwei Bedeutungen in der Software-Welt, die genau der doppelten Bedeutung des englischen Wortes free entsprechen. Zum einen soll das Programm kostenlos zur Verfügung stehen, zum anderen soll der Quellcode des Programms für alle frei einsehbar sein. Wobei Stallman sehr großen Wert darauf legt, dass freie Software mehr mit freier Rede zu tun hat, als mit Freibier.⁷ Das soll nämlich alles nicht heißen, dass man für freie Software kein Geld verlangen dürfe. Geld lässt sich mit GNU/Linux verdienen, wenn man es mit anderer freier Software zusammen auf CDs presst und eine Dokumentation und Support dazu anbietet. Ein ganzer Geschäftszweig von Firmen, wie Red Hat und SuSE, sogenannte Distributoren, hat sich um GNU/Linux entwickelt.

⁶ Linus Torvalds und David Diamond, *Just for Fun*. München, 2001, S. 94.

⁷ „Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer. Richard Stallman, *The Free Software Definition*. URL: <http://www.gnu.org/philosophy/free-sw.html> – Zugriff am 10.09.2004

Linux wurde von Anfang an mit Quellcode veröffentlicht. Hatte jemand Probleme, Linux auf seinem Computer zu installieren, konnte er den Quellcode verändern, das Programm neu compilieren, und es vielleicht dann installieren. Hat jemand etwas an Linux vermisst, was er unbedingt brauchte, konnte er es einfach hinzufügen. Die Kompatibilität zu Unix sollte mit Minix allerdings gemeinsam bleiben. Linux ist ebenfalls wie Unix in der Programmiersprache C geschrieben, weswegen es auf den verschiedensten Computerplattformen (z.B. IBM-PC oder Apple) installiert werden kann.

Weiter ist an dem Posting bemerkenswert, dass Torvalds die Werkzeuge *bash* und *gcc*, den GNU-C-Compiler, bereits in sein neues Betriebssystem integriert hat. Beides sind Werkzeuge des GNU-Projekts. Der GNU-C-Compiler dient zur Übersetzung von Programmen in Binärcode, die in der Programmiersprache C geschrieben worden sind. Mit dem GCC wird der Quellcode von Linux compiliert, damit das Betriebssystem auf einem Computer läuft.

Durch die Entscheidung Torvalds, sein Betriebssystem mit Unix kompatibel zu gestalten, waren mit einem Schlag alle Programme unter Linux lauffähig, die für Unix geschrieben worden waren, darunter alle Programme des GNU-Projekts. Ohne die GNU-Werkzeuge wäre die Entwicklung von Linux fast gar nicht möglich gewesen – Torvalds hatte sich auf die „Schultern von Giganten gehievt“. Deshalb argumentiert Stallman auch, dass nur von Linux zu sprechen zu verkürzt wäre. Besser sei, GNU/Linux zu sagen, wenn man das ganze Betriebssystem meint und Linux, wenn nur vom Kern, das was Torvalds und seine Mitstreiter konkret programmiert haben, die Rede ist.

Stallman und die von ihm mitbegründete Free Software Foundation, eine gemeinnützige Organisation für die Freie-Software-Bewegung, haben den Entwicklern freier Software eine politische Waffe in Form einer Lizenz in die Hand gegeben – die GNU General Public License (GPL). Sie sorgt dafür, dass ihre Software weiterhin frei bleibt. Unter die GPL hat Torvalds auch Linux gestellt. In seiner Biographie *Just For Fun* sagt er, warum er sich für die GPL entschieden hat: „Um Linux nutzbar zu machen, hatte ich mich auf eine Menge Tools verlassen, die frei über das Internet verteilt worden waren – ich hatte mich auf die Schultern von Giganten gehievt. Das wichtigste dieser frei zugänglichen Programme war der GCC-Compiler gewesen. Das Urheberrecht an ihm wurde nach der General Public License geschützt, die weltweit als die GPL (oder das Copyleft“) bekannt und ein geistiges Produkt von Richard Stallman ist. Nach den Begriffen der GPL ist Geld nicht das Thema. Du kannst eine Million Dollar verlangen, wenn du jemanden findest, der sie dir zahlt. Aber du musst den Quellcode verfügbar machen. Und die Person, der du den Quellcode gibst oder verkaufst, muss alle Rechte besitzen, die auch du besitzt. Es ist

ein brillantes Konzept. [...] Deshalb verwarf ich mein altes Copyright und übernahm die GPL, ein Dokument, das Stallman verfasst und von Anwälten hatte überprüfen lassen.“⁸

4 GNU

Der Begriff Free Software existiert erst seit etwa 1984. Damals gründete Stallman das GNU Projekt als Reaktion auf die Proprietarisierung von Software. Noch in den späten 1970er Jahren, war es nicht notwendig, zwischen freier und proprietärer Software zu unterscheiden, da Software nicht als geistiges Eigentum angesehen wurde, mit dem sich Geld verdienen ließe. Software wurde von Firmen umsonst mit zur Hardware dazugegeben und in der Informatik wurde über Quellcodes diskutiert. Die Veröffentlichung von Programmen war so üblich wie die Veröffentlichung von Forschungsarbeiten in anderen wissenschaftlichen Disziplinen. GNU ist ein rekursives Akronym und bedeutet „GNUs Not Unix“. GNU ist der Name für ein vollständiges Software System, das kompatibel zu Unix ist, und das im Gegensatz zu proprietären Unix-Varianten frei an jeden weitergegeben werden soll, der es braucht. Im *GNU Manifesto* schreibt Stallman dazu: „I consider that the golden rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. For years I worked within the Artificial Intelligence Lab to resist such tendencies and other inhospitalities, but eventually they had gone too far: I could not remain in an institution where such things are done for me against my will.“⁹

In Stallmans Biographie *Free as in Freedom* erzählt Sam Williams, wie Stallman auf den Paradigmenwechsel, dass Software als Geschäftsgeheimnis betrachtet und nur noch als Binärcode vertrieben wurde, aufmerksam wurde. Die Firma Xerox hatte dem Artificial Intelligence Laboratory (AI Lab) am MIT einen Laserdrucker gestiftet. Stallman ärgerte sich über einen Fehler in der Software des Druckers. Der Drucker lieferte keine Meldung, wenn er kein Papier mehr oder Papierstau hatte oder der Druckauftrag nicht vollständig ausgeführt wurde. Mit einer Veränderung der Software des Druckers hätte er den Fehler beheben können, doch: „The printer didnt have any software, at least nothing Stallman or a fellow programmer could read. Until then, most companies had made it a form of courtesy to publish source-code files – readable text files that documented the individual

⁸ Torvalds und Diamond, S. 105.

⁹ Richard Stallman, *The GNU-Manifesto*. URL: <http://www.gnu.org/gnu/manifesto.html> – Zugriff am 10.09.2004

software commands that told a machine what to do. Xerox, in this instance, had provided software files in precompiled, or binary, form. Programmers were free to open the files up if they wanted to, but unless they were an expert in deciphering an endless stream of ones and zeroes, the resulting text was pure gibberish.¹⁰

Stallman fand heraus, dass die Software des Laserdruckers aus einem Xerox-Projekt heraus an der Carnegie Mellon Universität entwickelt worden war. Bei einem Besuch an der Carnegie Mellon fragte er den zuständigen Professor, ob er eine Kopie des Quellcodes haben könnte. Doch zu seiner Überraschung weigerte sich der Professor, Stallman den Quellcode zu geben. Er hatte eine Nichtweitergabe-Verpflichtung (Non-disclosure-Agreement) unterschrieben. Das Resultat wissenschaftlichen Forschens stand damit weder der Allgemeinheit noch anderen Wissenschaftlern zur Verfügung. Es gehörte ausschließlich Xerox, die es lizenziert vertrieben.

Ein weiteres Zeichen für den Paradigmenwechsel in der Software-Welt war die Kommerzialisierung von Unix. Bis 1984 hatte die Firma AT&T nur das Monopol darauf, Telefondienstleistungen zu verkaufen. Unix, das in den Bell Laboratories von AT&T entwickelt worden war, war nur ein akademisches Experiment, aus dem sich ein beliebtes Betriebssystem entwickelte. Nach der Zerschlagung von AT&T wurde überlegt, Unix zu lizenzieren, um damit Geld zu verdienen. Dafür wurden Angestellte von Universitäten, die Unix nutzten und weiterentwickelten, gebeten, Geheimhaltungsabkommen zu unterzeichnen.

Gegen diese Kommerzialisierung von Unix entwickelte Stallman das GNU-Projekt. Für GNU begann Stallman eine Reihe von Software zu entwickeln, wie Emacs, einem leistungsstarken Editor, der vor allem unter Programmierern sehr beliebt ist, oder dem GCC. Ein Betriebssystemkern ließ allerdings lange auf sich warten, bis eben Linux kam.

5 GPL

Damit niemand den Quellcode freier Software als Binärdateien verkaufen kann, hat Stallman die GPL entwickelt.

¹⁰ Sam Williams, *Free As In Freedom. Richard Stallmans Crusade for Free Software*. Sebastopol, 2002, S. 4

In seinem Aufsatz *Am Anfang war alle Software frei* schreibt Tilman Baumgärtel, dass es Microsoft in seiner heutigen Form wahrscheinlich gar nicht gäbe, „wenn es bei seiner Gründung nicht auf „Open Source“-Programme hatte zurückgreifen können.“¹¹

Dabei geht es zum Beispiel um so entscheidende Programme wie MS-DOS, welches erstaunliche Ähnlichkeiten zum 1975 entwickelten Betriebssystem CP/M des Universitätsprofessors Gary Kildall aufweist. Es geht um Basic, dessen Quellcode Allen und Gates mit dem frei zugänglichen Basic der Universitätsprofessoren John Kemeney und Thomas Kurtz entwarfen und schließlich geht es um Windows NT (und somit auch um dessen Nachfolger Windows XP), dessen Quellcode eine fast perfekte Unix-Kopie ist.

Die GPL ist ein Vertrag, der garantiert, dass jede Zeile Code frei bleibt. Darüber hinaus garantiert sie, dass jede Software, in der GPL-Code verwendet wird, offengelegt werden muss. Die GPL garantiert, dass Software, die unter ihre Lizenzbedingungen gestellt wird, Gemeineigentum bleibt. Jeder besitzt die gleichen Rechte an der Software wie die Entwickler. Erst dadurch wird Software zu Privateigentum, da GPL-Software ihrem Besitzer vier Freiheiten garantiert, die von der Free Software Foundation in der *Free Software Definition* festgehalten sind:

„Free software is a matter of the users freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. (freedom 3)
- Access to the source code is a precondition for this.“¹²

Software als Privateigentum ist nichts Revolutionäres, denn mit jedem anderen Gegenstand, den ich erwerbe, kann ich verfahren wie ich will. Sie steht in Konkurrenz zur

¹¹ Tilman Baumgärtel, *Am Anfang war alle Software frei*. Microsoft, Linux und die Rache der Hacker. In Alexander Roesler/Bernd Stiegler (Hrsg.): *Microsoft. Medien, Macht, Monopol*. Frankfurt am Main, 2002, S. 106.

¹² Richard Stallman, *The Free Software Definition*. <http://www.gnu.org/philosophy/free-sw.html> – Zugriff am 10.09.2004

proprietärer Software, an der ich nur das Recht erwerben, sie benutzen zu dürfen, die aber nie mein Privateigentum sein wird. Denn das ist gerade der Gebrauch des geistigen Eigentums durch den Lizenzgeber, dass es sein Privateigentum bleibt. Allein Windows an einen Nachbarn zu verleihen, ist nach den Lizenzbedingungen von Microsoft schon ein Verstoß.

Inwieweit die GPL als politisches Kampfmittel Software als Allgemeingut erhalten kann und den Gebrauch als Privateigentum durch nur einen einzelnen Menschen oder einer einzelnen Körperschaft ausschließen wird, wird die Zukunft zeigen. Die wahre Schlacht um die GPL und freier Software steht noch aus, denn bislang triumphiert freie Software über das Monopol von Microsoft „nur“ auf dem Server-Markt.

Microsoft fürchtet bei der GPL nicht nur darum, Millionen US-Dollar an freier Software zu verlieren, es fürchtet um eine Vision. Diese Vision heißt .NET (sprich: Dot Net), und sie ist die Vision des vollständig von Microsofts Software umgebenen Menschen: „.NET ist für Microsoft ein weiterer Schritt auf dem Weg nach vorn. Das Ziel von .NET ist nicht nur ein Computer, der immer mehr die Funktion eines persönlichen Assistenten übernimmt. Das Ziel ist auch eine an den Menschen angepasste Form digitaler Kommunikation, die durch Sprach- und Handschriftenerkennung ermöglicht wird. Und das Ziel ist eine einheitliche Technologie, mit der Benutzer Kommunikationsmedien zusammenführen und Aufgaben aus verschiedenen Lebensbereichen erledigen können – zu jeder Zeit, an jedem Ort und von jedem Gerät aus.“¹³

Freie Software, die unter den Bedingungen der GPL veröffentlicht wird, stört diese Vision, da sie nicht kontrollierbar ist, und die Softwareprojekte sind nicht aufkaufbar. Jeder GPL-Code, der in andere Projekte miteinfließt, hat zur Folge, dass das ganze Projekt offen gelegt werden muss. Die Entwicklung freier Software wird über das Internet delegiert und ihre Ergebnisse im Netz diskutiert. Das schafft zwar leistungsstarke Software, weil sich tausende von Entwicklern an der Software beteiligen, heißt allerdings „nur“, dass die Kooperation von Angestellten eines Softwareunternehmens auf eine höhere Ebene gehievt wird. Das Revolutionäre, oder das Neue daran ist die Masse, wieviel Menschen miteinander mehr oder weniger kontinuierlich kooperieren können. Mit dem Internet ist diese Form der Kooperation erst möglich geworden. Es bleibt jedoch Kooperation, wie sie in jedem Softwareunternehmen zu finden ist.

Die GPL verhindert zwar, dass Menschen von dem Gebrauch freier Software ausgeschlossen werden, aber sie schließt auf der anderen Seite aus, dass jemand aus freier Software

¹³ Microsoft, *.NET – ein Schritt auf dem Weg nach vorn*. URL: http://www.microsoft.com/germany/ms/unternehmensinformationen/corp_profil/25jahre.htm – Zugriff am 10.09.2004

proprietäre macht. Niemand kann daran gehindert werden, Linux zu benutzen, und niemanden kann sein Linux weggenommen werden. Die GPL schließt aus, dass Linux nur einen einzigen Eigentümer hat. Wer Linux aus dem Internet herunter lädt, auf seinen Rechner installiert, verschenkt oder verkauft, dem gehört es auch.

Copyright © 2005 CHRISTIAN IMHORST.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation.

6 Literatur

Baumgärtel, Tilman: Am Anfang war alle Software frei. Microsoft, Linux und die Rache der Hacker. In: **Alexander Roesler/Bernd Stiegler** (Hrsg.): Microsoft. Medien, Macht, Monopol. Frankfurt am Main, 2002.

Brand, Stewart: The Media Lab: Inventing the future at MIT. New York, 1987.

Microsoft: .NET – ein Schritt auf dem Weg nach vorn.

URL: http://www.microsoft.com/germany/ms/unternehmensinformationen/corp_profil/25jahre.htm

Moody, Glyn: Rebel Code. London, 2001.

Stallman, Richard M. 1984: The GNU-Manifesto.

URL: <http://www.gnu.org/gnu/manifesto.html>

Stallman, Richard M. 1996: The Free Software Definition.

URL: <http://www.gnu.org/philosophy/free-sw.html>

Torvalds, Linus & Diamond, David: Just for FUN. München, 2001.

Wayner, Peter: Kostenlos und überlegen. Stuttgart, 2001.

Valloppillil, Vinod 1998: Halloween I.

URL: <http://www.opensource.org/halloween/halloween1.php>

Williams, Sam: Free As In Freedom. Richard Stallmans Crusade for Free Software. Sebastopol, 2002.