

Freie Software

Stephan Uhlmann

Potsdamer Linux User Group (upLUG)

26. Januar 2007

(cc) Creative Commons Attribution-ShareAlike

- Computer sind unentbehrlich in einer Vielzahl von Lebensbereichen
 - Kommunikation, Bildung, Arbeit, ...
- ohne Software läuft auf Computern nix
- Software vergleichbar mit Kochrezepten
- Quellcode vs. Binärcode

Quellcode vs. Binärcode

Binärcode

```
7F454C460101010000000000000000
00000000300030001000000105E
010034000000ACAB1300000000
00340020000700280039003600
06000000340000003400000034
000000E00000000E000000000500
000004000000003000000C8D712
00C8D71200C8D7120013000000
13000000040000000100000001
0000000000000000000000000000
000058DD120058DD1200050000
```

Quelltext

```
if (panic_timeout > 0) {
/*
Delay timeout seconds before rebooting
the machine. We can't use the "normal"
timers since we just panicked..
*/
printf(KERN_EMERG "Rebooting in %d
seconds..",panic_timeout);
}
```

- anfangs wurde Hardware mit Software im Quellcode ausgeliefert
- Kultur des Teilens, universitärer/wissenschaftlicher Bereich
- dann: Software wird Ware, kein Quellcode mehr, restriktive Nutzungslizenzen, NDAs
- Richard Stallman am MIT KI-Labor: Problem mit Xerox Netzwerkdrucker auf einmal für ihn nicht mehr lösbar

- 1984 GNU Projekt (GNU's Not Unix), 1985 Free Software Foundation
- <http://www.gnu.org>, <http://www.fsf.org>
- Frei im Sinne von „Freiheit“ nicht „Freibier“
- Freie Software garantiert folgende vier Freiheiten:



Die Freiheit, das Programm für jeden Zweck zu nutzen.

Freiheit 1

Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann.

Der Zugang zum Quellcode ist dafür Voraussetzung.

Die Freiheit, Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann.

Die Freiheit, das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann.

Der Zugang zum Quellcode ist dafür Voraussetzung.

Der Copyright „Hack“

- Copyright/Urheberrecht in Grundform restriktiv
- Lizenzierung: z.B. durch GNU GPL (General Public License)
- GPL-Besonderheit: veröffentlichte Veränderungen müssen wieder unter GPL lizenziert werden
 - Copyleft-Prinzip, sichert die Freiheit des Programms
- viele weitere Lizenzen mit den Freiheiten von Freier Software

- ähnliche Ziele, anders „verkauft“
- weniger Fokus auf Freiheitsaspekt, mehr Fokus auf praktische Vorteile
- Open Source Initiative
- <http://www.opensource.org>



Use the source, Luke!

- „Shared Source“
 - Marketingbegriff der PR-Abteilung von Microsoft
 - Quellcode einsehbar, darf aber nicht ohne Einschränkungen verändert oder weitergegeben werden
- „Closed Source“
 - kein Quellcode
 - proprietäre (geheime) Software

- Linux Kernel → Betriebssystem
- Kernstück, das die Hardware verwaltet
- 1991 von Linus Torvalds gestartet, um seinen 386er PC zu nutzen
- seitdem Hunderte von Mit-Programmierern
- Maskottchen: Tux



- durchaus hierarchisch → Maintainer (Koordinator des Projekts), oftmals auch wechselnd
- Bsp.:
 - Linux: Linus Torvalds „wohlwollender Diktator“, entscheidet was in den Linux-Kernel geht und was nicht
 - KDE: flache Strukturen, jeder Teilbereich eigene Maintainer, wechselnde Release-Manager
- Teilbereiche (Programmierung, Dokumentation, Übersetzung)

Projektorganisation

- Teammitglieder oft global verteilt
- Mischung: Freizeitprogrammierer und Angestellte von Firmen (Bsp. Linux, MySQL)
- Internet sehr wichtig zur Kommunikation und „Vertrieb“ der Software
 - Homepage, Mailinglisten, Chat, Codeverwaltung, ...
- grössere Projekte: rechtliche Dachorganisation (Bsp. KDE e.V.)



- Fokus auf technische Überlegenheit der Software
- keine ablenkenden strategischen Ziele („time-to-market“, Verdrängung der Konkurrenz), keine sinnlosen Features nur fürs Marketing
- Konsenssuche im Team
- Bevorzugung offener Standards (Protokolle, Dateiformate), Kompatibilität erwünscht
- aber: keine Garantien bzgl. Release-Zeiten, Bugfixes, Support

- Lösung eines eigenen Computer-Problems
- Erlernen neuer Fähigkeiten
- Anerkennung, Ego :-)
- Ausleben von Kreativität, Spaß
- Referenzen, Aufbesserung des Lebenslaufs
- Geld verdienen

Vorteile für Benutzer

- offener Quellcode
 - mehr Sicherheit, keine Spyware, Backdoors, ...
 - keine Abhängigkeit von einem Hersteller
 - Verwendung offener Standards und Dateiformate
 - keine Lizenzkosten, Zwangsregistrierungen oder ähnliche Schikanen
- keine Einschränkung der Nutzung und Weitergabe, Kopieren kein Problem
- Community: Support über Mailinglisten, Foren, Newsgroups usw., oft durch Entwickler selber

- Chancengleichheit
- Verringerung der digitalen Kluft
- freier Informations- und Wissensaustausch
- keine Monopolbildung

Warum soll meine Software Freie Software sein?

- Feedback und Verbesserungen durch Gleichgesinnte möglich
 - dadurch schneller zu besserer Software
- interessante meist internationale Kontakte
- Beitrag zur „Community“
 - mehr Quellcode zum Lernen
 - mehr Quellcode zur Wiederverwendung
 - mehr Software die frei ist

- Bugreports
- Übersetzungen
- Dokumentation
- Grafiken, Icons, usw.
- „Danke“ sagen :-)
- Einsteigern helfen, LUG gründen oder einer beitreten

- Installation und Benutzung nicht schwer
- Distributionen: SUSE, Debian, Fedora, Ubuntu, Gentoo, Mandrake, Knoppix, ...
- Programme ausgereift und vielfältig
 - grafische Desktopumgebungen: KDE, GNOME
 - Programme für jeden Bereich: OpenOffice, Firefox, Thunderbird, Gimp, Inkscape, XMMS, MPlayer, Apache, MySQL, PHP, ...

„If I have seen further, it is by standing on the shoulders of giants.“
- Sir Isaac Newton, 1676



- Volker Grassmuck: „Freie Software - Zwischen Privat- und Gemeineigentum“, <http://freie-software.bpb.de/>



- FSF Europe, <http://www.fsf europe.org>

Potsdamer Linux User Group

www.uplug.de

