

# Über GPL und das Tuxbox Projekt.

## Table of contents

1 Zusammenfassung.....	2
2 Distanzierung.....	2
3 Die "Hacker's Ethics".....	2
4 GPL: Ein Hackers Ehrencodex in juristischer Spache.....	3
5 Das Hackerprojekt "Tuxbox".....	4
6 Das Tuxbox Forum.....	6
7 Zu den drei "offiziellen Images".....	7
7.1 Die dietmarw-Images.....	7
7.2 YADI-images.....	7
7.3 "Jack the Grabber"-Images.....	8

## 1. Zusammenfassung

In diesem Artikel wird etwas Hintergrundinformation zu der Lizenz für die Software des Tuxbox Projekts, die GPL (General Public License) präsentiert. Dies ist als Formalisierung der Idee von Software Sharing zu verstehen. Es wird auf das Tuxboxprojekt und seine drei "offizielle Images", eingegangen, sowie der Kampf gegen das Schwarzsehen. Es wird gezeigt, dass diese Verhältniss an starken Widersprüchen leidet.

Ich möchte mich hier bei den Forumsbenutzern dietmarw und Feynman für Feedback auf eine frühere Version dieses Artikels bedanken.

## 2. Distanzierung

Ich distanziere mich ausdrücklich vom empfangen und entschlüsseln von verschlüsselten Fernseh- und Radioprogrammen, für die keine gültige Lizenz vorliegt ("Schwarzsehen"). Dies ist sowohl strafbar, als auch ein moralisch verwerfliches Vergehen/Verbrechen. Dass ich im Folgende einige Versuche, das Schwarzsehen zu bekämpfen kritisiere, darf in **keinster Weise** als Unterstützung oder Verharmlosung vom Schwarzsehen verstanden werden.

## 3. Die "Hacker's Ethics"

Es ist sowohl nützlich als auch interessant etwas über die Hintergründe der GPL zu wissen. Dies ist von Stephen Levy in dem Buch "Hackers: Heroes of the computer revolution" sowohl sehr gut als auch sehr lesenswert beschreiben. (Das Buch ist z.B von Amazon erhältlich; leider keine deutsche Übersetzung verfügbar. Das englischsprachige Wikipedia hat einen sehr guten [Artikel](#) über das Buch.) Levy beschreibt dadrin u.a. die "hackers ethics", die sich in folgende Punkten zusammenfassen lässt:

1. Information soll frei sein. Sie darf nicht verborgen werden, oder geheimgehalten.
2. Programmierer/hackers tauschen Information zwischen sich aus. Das Verbergen von Information ist unkooperativ, sowohl gegen andere Hackers, als auch gegen die Menschlichkeit.
3. Quellcode für Programme ist in diesem Sinn nichts anderes als "Information".
4. Jede Art von "Gefängniss" für Informationen stellt eine Herausforderung zum Knacken dar.

Bekanntestes Sprachrohr ist [Richard Stallman](#), der, um diese Ideen zu verteidigen und weitmöglichst zu verbreiten, die [Free Software Foundation \(FSF\)](#) und das [GNU Projekt](#) gegründet hat.

Das Prinzip vom "Sharen" von Software und dessen Quellcode ist vom Prinzip der Freiheit (und die langfristige Gewährleistung der Freiheit) der Information abgeleitet.

## 4. GPL: Ein Hackers Ehrencodex in juristischer Sprache

Die einfachste Möglichkeit ein selbstgeschriebenes Programm zu sharen, ist das Verzichten auf alle Rechte und Einschränkungen für das Programm: die Veröffentlichung als "public domain". Dies heisst, dass jeder sich davon bedienen kann (gut), vielleicht abgeleitete und verbesserte Versionen erstellen kann (auch gut), und vielleicht sie unter weniger freie Voraussetzungen andere zu Verfügung stellen kann (weniger gut). Es war Stallmans Wunsch, dass freie Software frei bleiben sollte, in dem Sinn, dass auch abgeleitete Werke zu Benutzung und Weiterverarbeitung der Öffentlichkeit zu Verfügung stehen. Deswegen hat er einige Regeln formuliert, die dem Empfänger einige Dinge verbieten, Dinge die die abgeleitete Werke unfrei machen würden.

Diese Regeln wurden in der "General Public License" zusammengefasst. Streng genommen ist sie nicht anderes als eine Formalisierung und Präzisierung von dem Prinzip (und Hackers Ehrencodex): "Freie Information (Software) soll frei bleiben".

Hier ist, informell ausgedrückt, die Grundidee in GPL: "Diese Software ist frei (nicht mit kostenlos zu verwechseln). Du darfst sie für alle Zwecke benutzen. Du darfst sie ausserdem weitergeben und für unterschiedliche Zwecke weiterentwickelt, und die modifizierte Versionen weitergeben. Was du nicht machen darfst, ist die Freiheit der Empfänger einzuschränken, in dem du die abgeleitete Software mit restriktiveren Bedingungen versiehst."

Die GPL formuliert dazu einige präzise Anweisungen, wie z.B. Anforderungen wie "Quellcode zur Verfügung zu stellen" zu verstehen sind.

Ein in diesem Sinn freies Programm darf für jeden zweck ("for any purpose") benutzt werden, gut oder böse.

Zu den Quellen eines Programmes zählen auch "Buildscripte", die zum Erstellen der Software erforderlich sind (es sei denn, das sie ganz trivial sind).

Der genaue Text für GPL Version 2 befindet sich [hier](#). Neulich ist eine [Version 3](#) erschienen, die aber sich nicht in dem Sinn von Version 2 unterscheidet. Eine deutsche Übersetzung der Version 2 befindet sich [hier](#). Nur die englische Version ist aber verbindlich. Auch lesenswert ist die FSF [FAQ zum GPL](#) (nur in englischer Sprache).

Die GPL ist neulich von Landgericht München I für juristisch verbindlich befunden.

Niemand kann zum "Sharen" gezwungen werden. So kann jemanden, der alle Rechte für ein Programm besitzen, frei wählen zwischen, u.a. keine Veröffentlichung, eine restriktive Lizenzierung gegen Lizenzgebühr, Veröffentlichung nur in Binärform, Public Domain (in Quell- oder Binärform), oder eine Veröffentlichung unter eine Lizenz wie GPL oder Ähnliches (siehe [diese Liste](#) über verbreitete Lizenzen für freie Software). Für abgeleitete Werke gelten aber andere Regeln, z.B. besitzt der "letzte Author" nicht die Rechte (mit Ausnahme des Public-Domain-Softwares), und kann, sowohl formell als auch moralisch, nicht frei über die Bedingungen für eine Veröffentlichung bestimmen.

Die GPL wurde in der 80-er Jahren formuliert. Während dieser Zeit war z.B. das selbständige Kompilieren von Programmen eine Selbstverständlichkeit. Nicht alle könnten selbständig ein Programm schreiben, aber man könnte mindestens C-Quellen von Usenet Newsgruppen wie z.B. [comp.sources.unix](http://comp.sources.unix) runterladen und mit dem Compiler übersetzen. Das Konzept von Software ohne Quellcode, oder Betriebssysteme ohne einen C-compiler war einfach fremd. Hacker (hier benutze ich das Wort als in Levys buch) haben hauptsächlich Programme für UNIX-Plattformen (vorzugsweise BSD) geschrieben, selten für andere Plattformen wie VMS, Amiga, Atari, Macintosh, C-64. Die Microsoft Betriebssysteme MS-DOS und Windows galten als das uncoolste überhaupt.

Seitdem hat sich vieles geändert... Das Wort [Hacker](#) hat leider in der moderne Nachrichtensprache die ursprüngliche Bedeutung verloren, und wird eher im Sinn von Computerkriminalität benutzt.

## 5. Das Hackerprojekt "Tuxbox"

In Jahr 2000 wurde ein interessantes Hackerprojekt gestartet: das Tuxbox projekt. Der PayTV-Sender [Premiere](#) hat, um das digitale Bezahlfernsehen zu stimulieren, sehr interessante high-end Hardware (mit dem Standards von etwa Jahr 2000) stark subventioniert auf den deutschen Markt veräußert (die [dBox2](#)). Das darin enthaltene Betriebssystem, Betanova, war, wie wir alle wissen, überhaupt nicht in der Lage das Potential der Hardware und der digitale Fernsehsendungen auszuloten. Die Hackerseele stellt sich dabei die Frage, wie die Hardware "befreit" werden kann.

Die ursprünglich entwickelte Software wurde dabei (mit Ausnahme von `mkflfs`, siehe unten) unter die GPL gestellt.

Eine interessante Beschreibung nicht nur von konkrete Tatsachen, sondern auch wie ein Hardwarehacker "tickt", befindet sich in dem Buch von Andrew "bunnie" Huang, ["Hacking the X-Box"](#), das sich mit dem X-Box von Microsoft befasst. (Leider ist das Buch nicht in Deutsch erhältlich.)

Auch wenn es (mit aller Wahrscheinlichkeit) niemals die Bestrebung der ursprünglichen Entwickler war, hat es sich gezeigt, dass die dBox2/Tuxbox-Kombination von relativ flexibler und leistungsfähiger Hardware, zusammen mit der offenen Natur der Software eine sehr attraktive Sammlung von Tools für das Entwickeln von Schwarzseherprogramme darstellte.

(Offiziell) aus diesem Grund hat sich das Tuxboxprojekt für ein Open-Source Projekt etwas merkwürdig verhalten. Eine wichtige Komponente (`mkflfs`) wurden nicht veröffentlicht, sondern geheimgehalten. Auch wenn die meisten anderen Teile im Quellcodeverwaltungssystem CVS veröffentlicht wurden, war es nicht möglich ausschliesslich mit veröffentlichten Code ein vollständiges Image zu erstellen und zu flashen. "Offizielle" Flashimages wurden von AlexW zur Verfügung gestellt. Erst später, u.a wegen der nicht veröffentlichte Tools, war es möglich ohne umfassende Einarbeitung ein Image zu erstellen.

Die offizielle Motivierung für die, in dieser Art, teilweise nicht freie Software war, dass man somit das erstellen von Schwarzseherimages auf Basis der Tuxboxsoftware verhindern wollte. Es hat sich gezeigt, dass dies nicht erfolgreich war. Das geheimhalten von `mkflfs` hat nichts gebracht: Es hat einfach gereicht, die passende Version (für 1 bzw. 2 Flashchips) von einem AlexW-Image in Binärform zu extrahieren und unverändert wiederzuverwenden, genau so wie die [Anleitung](#) zu Erstellen von "sauberen" Images das Verfahren beschrieben hat.

Zum Flashen, und für einige andere Operationen auf der dBox war es nahezu notwendig, ein nur für Windows verfügbares programm mit geheimem Quellcode zu verwenden: der [Bootmanager](#). (Laut Gerücht sind die Quellen nicht nur niemals veröffentlicht worden, sondern sogar in einem Plattencrash verlorengegangen :-). Ein moderneres ähnliches Programm ist Alexander Hallenbergs ("Gurgel") [Flashassistent](#), auch mit unveröffentlichtem Quellcode.

Wesentlich für den Erfolg des Projekts war das "reverse Engineering" von Teilen der ursprünglichen Software. Insbesondere wird aus der (als legal erworbene, z.B. bei Kauf der Hardware) Software einige Binärdateien, die Firmware für einige custom Chips (AViA und CAM-modul) extrahiert, und in der neuen Software benutzt. Diese Firmware ist nur in vorhandene, verschlüsselte Form erhältlich. Eine decompilierung oder Analyse ist (meines Wissens nach) niemals publiziert worden. Es wird angenommen, dass sobald die Originalsoftware legal erworben ist, und die Firmware nicht zu Weiterverbreitung angeboten werden, alle gesetzlichen Anforderungen erfüllt sind, und, implizit, dass das Verfahren juristisch unproblematisch ist. (Siehe z.B. [Tuxbox Wiki](#).)

Unter anderem aus diesen Gründen ist das Tuxbox-Projekt (leider) mehr verbunden mit bereitstellen von Flashimages, als mit dem eigentlichen Kern: die Quellen im CVS.

Seit ein Paar Jahren existiert eine große Anzahl von unterschiedlichen Tuxbox-basierenden Images. Fast alle sind für Schwarzsehen ausgelegt, auch wenn man oft das eigentliche "Scharfmachen" (installation von besondere Softwarekomponente, z.B. sogenannte Emulatoren ("Emus"), die eine lizenzierte Entschlüsselung emuliert, und installation von Entschlüsselungsschlüsseln ("Keys")) dem Benutzer überläßt. Fast alle diese Images verletzen die GPL in dem Sinn dass sie:

1. Quellcode für Erweiterungen, sowie ggf. Buildscripte etc. nicht den Benutzern zu Verfügung stellen, und/oder
2. Die Images werden anderen Regeln für Weiterverbreitung und Modifikation unterlegt als GPL.

Die meisten Imagebauer legen großer Wert darauf, ihre Images mit individuellen Verbesserungen zu versehen. Dies betrifft natürlich die Schwarzseherfähigkeiten, aber auch andere Eigenschaften wie GUI-Menüs, Plugins, Logos etc. Die "Verbesserungen" diesbezüglich sind in fast allen Fälle ziemlich überflächlich, und tragen nur selten zu wirklichen Funktionalitätsverbesserungen oder -Erweiterungen bei. Auch zu den Modifikationen, die nicht mit Schwarzsehen verknüpft sind, wird Quellcode geheim gehalten. Dies gilt auch für verwendete Buildscripte etc. In einem Fall wird eine

Weiterdistribution untersagt (sowohl von unmodifizierte, als auch für modifizierte Images), sogar Anleitungen zu Modifikation (egal für welche Zweck) werden verboten!, angeblich um die Benutzung für Schwarzsehen zu verhindern...

Es ist verständlich, dass die Imagebauer in gewissem Sinn markieren wollen, was sie erschaffen haben — trotzdem ist es ja eine kreative Tat. Durch 1. oder 2. oben macht man aber dadurch freie Software unfrei, indem man weiteres "Sharing" (sowohl von eigene Beiträge als auch von den "99%" der Programmcodes, der unverändert durchgereicht wird) untersagt. Traurig ist, dass das Sharen von Software keinen Stellenwert hat; man ist sogar stolz über geheim gehaltene ("non-public") Teile. (Vielleicht hat man Angst, dass eine Veröffentlichung schlechten Programmierstil, oder "geklaute" Teile verraten würde?)

Formell sagt man, dass 1. und 2. oben die GPL verletzt. Wie oben beschrieben, sehe ich dies nicht als ein Verletzung irgenwelcher langweiligen und uncoolen Bestimmungen, sondern als ein *unkooperatives Verhalten*: Es werden Programme, frei im Sinn der GPL zu Verfügung gestellt, und man bedankt den Autoren und der Welt damit, dass man abgeleitete Werke unfrei macht. Es ist auch eine zweifelhaftes Verständniss von "intellektuellem Eigentum" (sowohl in formellem als auch in moralischem Sinn). Den ursprünglichen Authoren und Copyrightinhabern verneint man ihre Rechte indem man ihre Lizenzbestimmungen ignoriert. Selbst fordert man aber, dass der Rest der Welt die eigenen Bestimmungen (die sich oft die ursprüngliche widersprechen) unbedingt respektiert.

## 6. Das Tuxbox Forum

Für das Tuxbox-projekt, wie vom [Forum](#) und [Wiki](#) definiert, gelten folgende Regeln (in meiner Formulierung):

1. Schwarzsehen wird nicht toleriert, auch nicht Diskussion darüber. Die Foren, die sich mit Schwarzsehen befassen, oder dulden, werden (in Anlehnung an Star Wars) "die dunkle Seite" genannt.
2. Die Images von dietmarw, YADI und Jack the Grabber ("JtG") gelten als die drei "[offiziellen Images](#)" des Tuxbox-projekts.
3. Alles andere als die drei "offizielle Images", und natürlich selbstgebautes, also auch "illegale Images" wo das "Schaftmachen" (in Sinn von oben) nicht stattgefunden hat, gelten als illegal.
4. Support für "illegale Images" werden verweigert, weil sie illegal sind, und weil sie (nicht a priori, aber in allen bekannten Fällen) "GPL verletzen".
5. Die "offiziellen Images" unterstützen out-of-the-box nicht den Empfang von Premiere, auch nicht mit gültiger Lizenz und Smartcard. Der Grund ist dass die AGB von Premiere den Empfang mit nicht autorisierter Hard- oder Software untersagt. In diesem Fall wird das "Scharfmachen" dem Benutzer überlassen: Entweder durch Austausch von einem enthaltenen Programm durch ein anderes, in Internet erhältliches, oder eine triviale Sourcecodemodifikation wird die Software in der Lage

sein Premiere, mit einem gültigen Smartcard, zu empfangen.

6. Die oben genannten Firmwarefiles (ucodes) dürfen nicht verbreitet werden. Die "offiziellen Images" werden ohne sie zur Verfügung gestellt. Das Extrahieren und das von Betanova unabhängige Benutzen wird als unproblematisch angesehen, solange sie aus dem original legalen Betanovaimage des Benutzers gewonnen wurden.

Im Tuxboxforum haben sich besondere Sitten entwickelt. Ablehnung vom offensichtlich illegalem Schwarzsehen, und man "hat" GPL. Man supportet "die drei offiziellen Images". Support für andere (ausser selbstgebastelte, ohne Schwartseherzusätze) wird verweigert, nicht nur aus gesetzliche Gründen, sondern (völlig konsequent und richtig) weil sich die Images "der dunklen Seite" sich auf, (zumindest im Detail) in geheimen Art und Weise modifizierte Quellen, und deswegen korrekte Antworten nicht möglich seien. Dabei geht man sehr "pragmatisch" mit den Mangel von Quellen zu dem JtG-Image um (siehe unten).

Das Verständniss von GPL scheint auch sehr "pragmatisch" zu sein. GPL wird als eine Sammlung feine, aber schwierig zu verstehende Regeln zu sein. Z.B. wurde einmal die Frage, falls das Bereitstellen der Quellen einer Modifikation durch Bereitschaft, auf Anfrage die Quellen zu mailen, abgedeckt ist, als "Auslegungsinterpretation" (fehlerhaft) eingestuft. Die Grundcharakter, GPL als Inkarnation der Idee der Sharing von Software wird nicht verstanden.

Viele Teilnehmer des Tux-Forums betätigen sich auch, mit den gleichen Nicknames, in Foren der dunklen Seite. Dabei sind sie fast ohne Ausnahme sich der unterschiedlichen Regeln und Sitten der Foren bewusst.

Die naive Idee, dass man das Böse (hier: das Schwarzsehen) durch Verbote (Einschränkungen der Freiheit der Software) bekämpft, ist sehr verbreitet (siehe z.B. die (teilweise recht amüsante) Diskussion über den Einsatz von Filesysteme mit effizientere Kompression). Es ist dabei mehr als unwahrscheinlich, dass irgendwelche Teilnehmer der dunkeln Seite sich von irgendetwas "bösem" durch ein Verbot (oder durch binary-only Programme) aufhalten lassen.

## 7. Zu den drei "offiziellen Images"

### 7.1. Die dietmarw-Images

Die [dietmarw-Images](#) werden nach Änderungen im CVS jede Nacht automatisch gebaut und zum Download zur Verfügung gestellt. Sie sind aus den tagesaktuellen Quellen, mit im CVS verfügbaren Werkzeugen gebaut.

### 7.2. YADI-images

Als die AlexW-Images eingestellt wurde, gab es keine automatische und reproduzierbare Art, Images zum Tuxboxprojekt zu bauen. Das Zeil des [YADI-Projekts](#) war, (und ist noch laut [YADI Homepage](#)): "Yadi versucht den Prozess der Imageerstellung basierend auf

dem GNU DBox2 Software Projekt, durch diverse Scripte und Patche zu vereinfachen bzw. zu automatisieren. Zusätzlich gibt es hier fertige Images im SquashFS und JFFS2 Format". Werkzeuge (shellskripte) wurde geschrieben, um ein Image automatisch und reproduzierbar zu erstellen. Die so erstellte Werkzeuge ("[das YADI-Skript](#)") waren aber in keinsten Weise ein Entwurf für ein richtiges Buildsystem für die Tuxbox, sondern eher als eine laienhaftige Sammlung von Shellskripte zu sehen. In 2004 könnte trotzdem das YADI-Skript ein benutzbares Image erstellen. Es wurde auch unterhalten. Eher als "Nebenprodukt" wurde das so erstellte Image zur Verfügung gestellt: das YADI-Image. Seitdem, scheint es, haben ursprüngliche Mitglieder das Projekts verlassen, und es scheint so, dass "YADI" nur noch aus ein bis zwei Personen besteht. Seit geraumer Zeit wird das YADI-skript nicht mehr gepflegt, und kann auch nicht mehr ein Image erzeugen. Die ursprüngliche Zielsetzung ist also inzwischen nicht mehr existent und statt dessen wird gelegentlich ein "YADI-Image" ins Netz gestellt.

### 7.3. "Jack the Grabber"-Images

Ursprünglich als Image optimiert um zusammen mit dem Closed-Source Programm [Jack the Grabber](#) zu funktionieren, wurde das sogenannte "Jack The Grabber"-Image (JtG) veröffentlicht. Bei JtG scheint man nicht viel vom Sharing der Software zu halten. (Siehe [Haftungsausschluss und Nutzungsbedingungen](#), siehe auch [Regeln für den JtG-Image Bereich](#).) Das Image verletzt die GPL in mehrfacher Weise: Es sind einige Modifikationen dadrin enthalten, z.B. an Neutrino, wobei die Quellen nicht zur Verfügung gestellt werden. (Wobei die Modifikationen oft früher oder später in irgendeiner Form ins CVS einfließen.) Die Verpflichtung, das abgeleitete Werk gleich frei wie die Quellen zur Verfügung zu stellen wird völlig ignoriert: Laut JtGs Bestimmungen darf das Image nicht weiterverbreitet werden, ein Recht zu Weiterentwicklung (egal für welche Zweck) wird dem Benutzer verneint, und sogar das Veröffentlichung von Anleitungen zu Modifikation (egal zu welchen Zweck) ist untersagt! Das Letzte ist nicht nur eine Verletzung der Softwarelizenzen sondern auch ein Versuch, die Recht der freien Meinungsäußerungen einzuschränken. (Nicht einmal Microsoft traut sich sowas zu...) Werkzeuge ("Buildskripte") zu Erstellen des Images werden nicht veröffentlicht. (Ich habe ganz klare Indizien, dass der Imagebauer das Imagebauen als wohlgehetes Geheimniss halten möchte.)