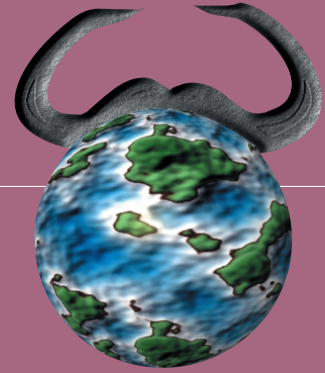


Die monatliche GNU-Kolumne

Brave GNU World



Diese Kolumne berichtet aus der Perspektive des GNU-Projekts und der FSF über Projekte und aktuelle Geschehnisse aus dem Umfeld freier Software. In dieser Ausgabe: K3b, Hardware-Treiber - proprietäre Treiber in freien Betriebssystemen, fünf Jahre Brave GNU World. Georg C.F. Greve

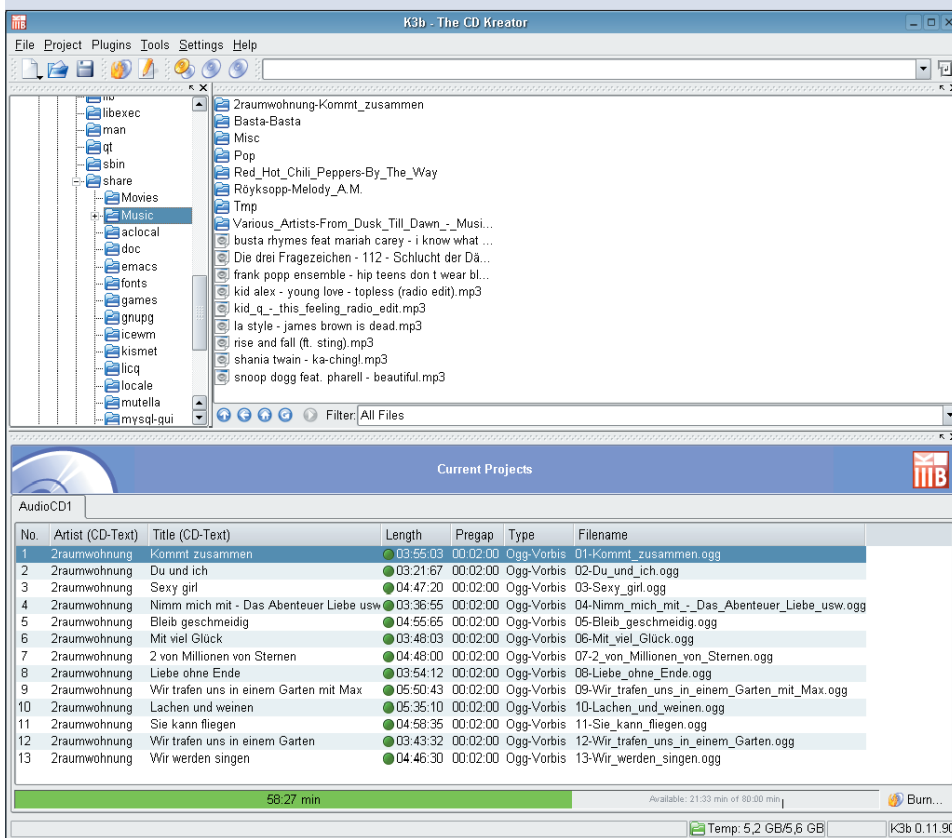


Abbildung 1: Mit der KDE-Brennsoftware K3b erzeugen auch weniger geübte Anwender per Drag & Drop CDs und DVDs. Bei Audio-CDs ist es egal, ob die Dateien als Ogg Vorbis, MP3 oder einem anderen Format vorliegen.

Willkommen zur 60. Ausgabe der Brave GNU World, die gleichzeitig eine Jubiläums-Ausgabe ist: Die Brave GNU World gibt es nun bereits seit fünf Jahren. Doch dazu später mehr. Los geht's mit der KDE-Brennsoftware K3b.

K3b brennt CDs und DVDs

CDs und DVDs brennen gehört bei den meisten Endanwendern zu den alltäglichen Beschäftigungen. Doch unter Linux kam man dabei lange Zeit nicht um die Kommandozeile herum. So tauglich und zuverlässig Programme wie Cdrec-

ord aus den Cdrtools [5] auch für den professionellen Nutzer sein mögen, bei Administratoren und Nutzern führte der Kommandozeilenbetrieb doch regelmäßig zu Schwierigkeiten. Das hat Jörg Schilling, den Autor des Programms, wohl auch dazu veranlasst, den Endnutzer-Support einzustellen.

Die Cdrtools stehen zwar unter der GNU GPL, Jörg Schilling hat aber aus dieser freien Version den DVD-Support entfernt. Um auch DVDs schreiben zu können, müssen Anwender auf eine unfreie Version ausweichen. Glücklicherweise gibt es für diesen Zweck noch die

DVD + RW-Tools [6], doch die arbeiten auch nur auf der Kommandozeile.

Besonders für Cdrecord gab es schon früh einige Frontends, die allerdings fehleranfällig und nicht wirklich schön waren. Zudem war ihr Einsatzgebiet oft auf einen bestimmten Zweck beschränkt, so war das Auslesen von CDs selten vorgesehen. Auch Sebastian Trueg war von den Frontends enttäuscht, als er 1998 endgültig das Kapitel Windows für sich schließen wollte. Keine der vorhandenen Möglichkeiten stellte ihn zufrieden, also begann er mit der Arbeit an K3b [7]. Besonders der Zuspruch, den er dabei für seine Arbeit erfuhr, hat ihn seitdem motiviert weiterzumachen.

Neue Entwickler

Natürlich sind auch weitere Entwickler maßgeblich beteiligt. Der Programmierer Chris Kvasny hat die Video-CD-Funktionalität (das Lesen und Schreiben von VCDs) entwickelt, er gehört seitdem gemeinsam mit Sebastian zu den ständigen Entwicklern. Von Thomas Froescher stammt der Code zum Lesen von Video-DVDs und Klaus-Dieter Krannich hat bei der Geräteunterstützung geholfen.

Einen großen Schub erfuhr das Projekt im Jahr 2003, als Suse es Sebastian Trueg im Rahmen eines dreimonatigen Praktikums erlaubte, Vollzeit an K3b zu arbeiten – dabei entstand die Unterstützung für das Schreiben von DVDs mit Hilfe von Growisofs aus den bereits erwähnten DVD + RW-Tools.

Und schließlich: Was wäre ein so vielseitiges Projekt ohne ein schönes Aussehen? Das außerordentlich coole Originaldesign von K3b stammt von Ayo [8]. Zusätzlich gibt es seit Version 0.11 von

K3b noch ein neues Design von Everaldo, dem Autor des KDE-Icon-Theme „Crystal Icon“.

K3b ist aber nicht nur ein schönes Frontend für Tools wie Cdrecord, Growisofs, Cdrdao und Mkisofs. Es nutzt beispielsweise Cdrecord nur für den eigentlichen Schreibvorgang und erledigt den Rest selbst. Dazu gehört das Auffinden von Geräten und Informationen über das Inhaltsverzeichnis der CD. Außerdem bestimmt K3b die Schreibgeschwindigkeit von Medien und testet vorab die Größe, um die Kapazität vollständig zu nutzen.

Erweiterte Funktionen

Legt man eine Audio-CD ein, erkennt K3b diese, sucht je nach Konfiguration lokal oder in der FreeDB-Datenbank im Internet [9] nach Künstler, Titel sowie Inhaltsverzeichnis und zeigt diese unmittelbar an, wie in **Abbildung 2** zu sehen ist. Mittels eines Kontextmenüs über die rechte Maustaste liest der Anwender dann die gesamte CD oder nur bestimmte Tracks aus und kodiert sie direkt nach Flac, MP3, Ogg Vorbis oder in andere Audioformate.

Beim Schreiben von Audio-CDs funktioniert natürlich auch der umgekehrte Weg. Dateien in beliebigen Formaten können zum Projekt hinzugefügt werden und K3b macht daraus eine Audio-

CD – auf Wunsch inklusive versteckter Bonus-Tracks. Und wem das noch nicht genügt, der nutzt die modulare Architektur von K3b, um zusätzliche Funktionalität einzubinden.

K3b ist in C++ geschrieben und benutzt diverse KDE-Bibliotheken. Mittlerweile ist das Programm im Verzeichnis »kdeextragear-1« des KDE-CVS zu Hause, läuft aber auch problemlos unter Gnome und anderen Oberflächen. Da K3b ebenso wie KDE selbst als freie Software unter der GPL steht, besteht die Möglichkeit, es an andere Oberflächen anzupassen. Für Sebastian steht allerdings die Herausgabe der Version 1.0 im Vordergrund, die noch für dieses Jahr erwartet wird. Die Entwickler sind momentan damit beschäftigt, Fehler zu fixen.

Neue Features sind geplant

Auch neue Features haben die Entwickler in petto. Als Nächstes wollen sie das Kopieren von Video-DVDs einbauen. Im Wesentlichen geht es dabei darum, die Daten so zu komprimieren, dass der Inhalt einer DVD-9 auf eine DVD-5 passt. Es ist auch geplant, ein Video-Transcoding-Modul für alle Video-CD- und -DVD-Projekte in K3b zu programmieren. Damit wird es möglich, aus unterschiedlichsten Videoformaten reguläre Video-CDs und -DVDs zu erstellen.

Freiwillige Helfer sind sehr willkommen: bei der Umstellung von Transcode auf die Libdvread, bei der erwähnten Transcoding-Schicht und bei der Vereinfachung der Menüs. Denn K3b will auch ohne jegliche Dokumentation für Anfänger benutzbar werden. Wer die Entwicklung beschleunigen möchte und nicht die Zeit oder das Wissen hat, sie direkt zu unterstützen, der leistet eine Spende an das Team. Unter [10] findet sich die entsprechende Seite.

Bei diesem Projekt haben die Entwickler fast alles richtig gemacht. Das fängt bereits mit dem sauberen Start als freie Software unter einer wohl bekannten Lizenz an. Das Team steht neuen Entwicklern immer offen gegenüber, orientiert sich stets am Benutzer und bemüht sich die Oberfläche so weit wie möglich zu vereinfachen. Auch Nicht-Programmierer haben die Chance, das Projekt indirekt über Spenden zu unterstützen. Schließlich zählt das Lesen und Schreiben von CDs und DVDs für viele Menschen zum alltäglichen Umgang mit ihrem Computer.

Hardware und Linux-Treiber

Bereits in Ausgabe 48 [11] habe ich die Probleme von Laptops in Verbindung mit freier Software – insbesondere mit GNU/Linux – angesprochen. Dabei ging es unter anderem um die Frage, inwieweit es akzeptabel ist, dass ein Hardwarehersteller seinen Kunden die Wahl der Software und damit das Anwendungsgebiet diktiert. Damit verbunden ist auch die Problematik von proprietären Treibern für freie Betriebssysteme, ein in letzter Zeit offenbar um sich greifendes Phänomen.

Hardware, zu der es keine passenden Treiber gibt, ist nutzlos und sie verkommt im besten Fall zu Ballast. Im Fall eines Laptops kann es sogar so weit kommen, dass bestimmte Komponenten ohne Treiber lediglich Strom und Betriebszeit verbrauchen. Natürlich ist es technisch ohne weiteres möglich, Hardware mit einem proprietären Treiber zu benutzen. Sieht man von einigen Sicherheitsbedenken ab – immerhin läuft hier unbekannter Programmcode, oft an privilegierten Stellen –, könnte man zu nächst davon ausgehen, dass die volle

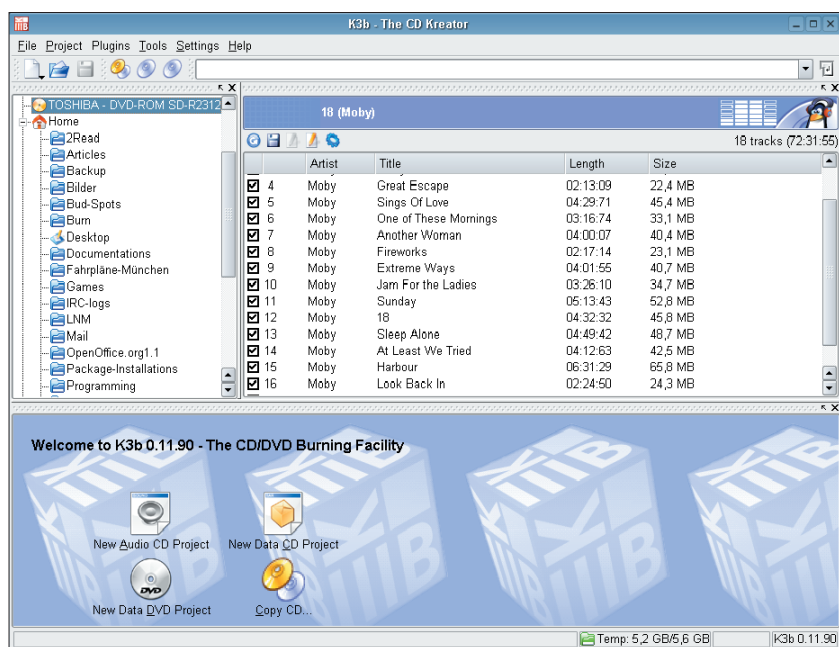


Abbildung 2: K3b ist nicht nur ein Frontend für existierende Programme, sondern liest auch Audio-CDs aus und holt sich im Internet Informationen zu den einzelnen Titeln per CDDb.

Nützlichkeit erreicht ist. Leider ist dies aber aus zwei wesentlichen Gründen nicht der Fall.

Erstens handelt es sich hier um den klassischen Fall, in dem der Nutzer die Hardware nur für jene Anwendungen einsetzen kann, die vom Hersteller vorgesehen und toleriert sind. Tritt ein unvorhergesehenes Problem auf oder ergibt sich eine andere Nutzungsmöglichkeit, besteht keine Aussicht auf Lösung dieses Problems. Ein Beispiel wäre die Idee, die beachtliche Rechenleistung von Grafikkarten als eine Art Zweit-Prozessor zu benutzen.

Unabhängigkeit dank offener Treiber

Zweitens begibt sich der Anwender in ein Abhängigkeitsverhältnis. Die Hardware verliert ihren Nutzen, wenn der Hersteller nicht mehr im Geschäft ist, nicht die benötigte Version oder Art des Betriebssystems unterstützt oder das Interesse an diesem Gerät verliert, um lieber den Absatz neuer Hardware zu forcieren. Ob Anwender ihre Geräte weiterhin einsetzen können, ist also in hohem Maße eine Entscheidung des Herstellers, nicht des Besitzers.

Gibt es Treiber in Form freier Software, bestehen diese Abhängigkeiten nicht mehr. Die Nutzer haben nicht nur die Freiheit, die Treiber in der Art anzupassen (oder anpassen zu lassen), die ihren Bedürfnissen optimal entspricht, sie können diese auch bei Bedarf auf andere Betriebssystemversionen oder sogar andere Betriebssysteme portieren.

Die Hardware ist dann vollständig unter allen Bedingungen einsetzbar. In manchen Fällen entfaltet sie sogar Leistungen, die selbst ihr Hersteller oft nicht für möglich gehalten hätte. Vor allem jedoch ist der Nutzer nicht mehr vom Hersteller abhängig und kann frei entscheiden, in welcher Form und wie lange diese Hardware für ihn nützlich sein soll.

Öffentlich zugängliche Spezifikationen

Auch wenn ein Hersteller selber keine freien Treiber schreiben kann oder will, hat er immerhin die Möglichkeit, die vollständigen Spezifikationen zu seinen

Produkten herauszugeben. Dies erfordert zwar ein bisschen mehr Arbeit als bei bereits vorhandenen Treibern, kommt dem aber in der Nützlichkeit und Freiheit von Abhängigkeit sehr nahe. Betrachtet man also den Wert der Hardware danach, wie nützlich sie ist, ist Hardware mit Treibern als freier Software ganz klar am wertvollsten. Hardware mit proprietären Treibern ist – je nach der Wahl des Betriebssystems – irgendwo zwischen mittelmäßig wertvoll bis wertlos.

Konsequenterweise sollte man also im Falle proprietärer Treiber sein Geld teilweise oder sogar komplett zurückfordern und die Hardware zurückgeben, wenn der Hersteller lediglich proprietäre Treiber anbietet. Im Falle solcher Laptops, bei denen es keine Treiber für integrierte Komponenten gibt, diese aber dennoch zum Gewicht beitragen und Strom verbrauchen, scheint Schadensersatz angebracht.

Proprietäre Treiber in freier Software

Früher kamen proprietäre Treiber nahezu ausschließlich in proprietären Betriebssystemen zum Einsatz, bei denen die zusätzliche Abhängigkeit angesichts der Abhängigkeit von der eigentlichen Software kaum auffiel. Seit kurzem je-

doch gibt es auch zunehmend proprietäre Treiber für freie Betriebssysteme, speziell GNU/Linux. Beispiele sind hier die in Intel-Centrino-Laptops integrierten drahtlosen Netzwerkkarten, die ebenfalls integrierten Modems und die Grafikkartentreiber von Nvidia.

Es ist verständlich, die Verschiebung von „völlig nutzlos“ zu „etwas nützlich“ als Fortschritt zu begrüßen. Das scheint in der Tat weit verbreitet zu sein – proprietäre Treiber werden zunehmend in Foren für GNU/Linux bejubelt und unkritisch empfohlen.

Unkritische Akzeptanz

Über das schrittweise Aufgeben der Freiheit, die nach der Salamataktik verloren geht – „Was ist schon ein kleiner Treiber?“ –, habe ich in dieser Kolumne früher bereits das eine oder andere Mal geschrieben, daher gehe ich darauf nicht weiter ein. Die meisten Nutzer sind sich nicht im Klaren darüber, dass sie mit der Akzeptanz und dem Einsatz proprietärer Treiber ein Klima schaffen, in dem sie den Wettbewerbsvorteil der Hersteller eliminieren, die ihnen die Hardware mit freien Treibern anbieten.

Zusätzlich wird durch diese Akzeptanz der Anreiz zur Entwicklung freier Software stark geschwächt – zum Teil genug, um eine Entwicklung freier Treiber

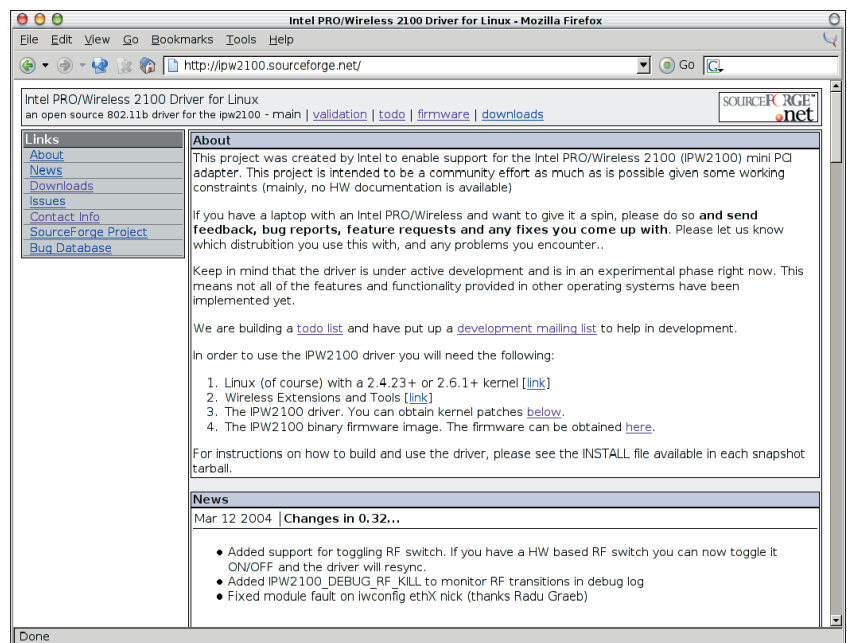


Abbildung 3: Die Homepage des IPW2100-Projekts. Intel hat die Treiber für ihre Centrino-WLAN-Karten teilweise unter die GPL gestellt. Vom Ideal ist diese größtenteils proprietäre Lösung aber noch weit entfernt.

zu verhindern. Gab es vor einigen Jahren noch einen erfolgreichen Boykott von Matrox-Grafikkarten, um an die Spezifikationen zu kommen, damit Programmierer vernünftige freie Treiber schreiben konnten, setzen heute viele Leute unreflektiert die proprietären Treiber von Nvidia ein und begrüßen diese als Fortschritt. Ein anderer Fall ist der proprietäre Driverloader, um Intels Centrino-WLAN-Karten unter Linux einzusetzen. Er hat das äquivalente freie Softwareprojekt Ndiswrapper [12] sicherlich massiv behindert, auch wenn sich genaue Berechnungen dazu wahrscheinlich kaum aufstellen lassen.

Natürlich ist die Argumentation des Driverloader-Herstellers nachvollziehbar, dass sie an der Hardware nichts verdienen und daher anders ihr Geschäft aufbauen müssen. Doch vernachlässigt dies zwei Faktoren: Erstens ist es ein vom Hardwarehersteller geschaffenes Problem, dessen Lösung eigentlich dem Hersteller am Herzen liegen sollte – und es wohl auch würde, wenn er sich einem entsprechenden Druck von der Community ausgesetzt sähe. Und zweitens ist jede Zahlung an die entsprechende Firma effektiv ein Schritt zur Verhinderung eines freien Treibers.

Freie Software muss nicht gratis sein

Gerne würde ich persönlich für Produkte das Doppelte des angegebenen Betrags zahlen, wenn ich dadurch eine freie Treibersoftware erhalte. Wer also darüber nachdenkt, beispielsweise die 15 US-Dollar für den Driverloader auszugeben, dem sei nahe gelegt, lieber 30 US-Dollar an das freie Ndiswrapper-Projekt zu spenden. Es liegt tatsächlich an jedem von uns, ein Klima zu schaffen, in dem Hardwarehersteller bevorzugt werden, die freie Treiber mitliefern oder zumindest ihre vollständigen Spezifikationen offenlegen.

Wer weiterhin proprietäre Treiber auf freien Betriebssystemen einsetzt, sollte sich im Klaren darüber sein, dass er eine Mitverantwortung daran trägt, wenn in Zukunft die frisch gekaufte Hardware nicht unterstützt wird. Tatsächlich gibt es diese Diskussion ja auch nicht erst seit gestern und in ernsthaften Diskus-

sionen wird erfahrungsgemäß niemand den Zusammenhängen und speziell der Einschätzung der Nützlichkeit widersprechen.

Geheimnisse preisgeben?

Vielmehr kommt häufig die Argumentation, dass die Firma ihr Mojo, also ihre Geheimnisse verlöre, wenn sie die Treiber als freie Software herausgäbe oder die Spezifikationen veröffentlichte. Nicht nur fehlt dieser Behauptung bisher der Nachweis, es gibt sogar handfeste Indizien dafür, dass es sich dabei lediglich um einen Mythos handelt. So hat Intel die Geheimnisse der Speed-Step-Technologie in den Centrino-Prozessoren geheim gehalten.

Mittlerweile unterstützt Linux Speed Step, dank Bruno Ducrot. Er musste die Technologie mangels Spezifikationen sehr intensiv erforschen, um überhaupt einen Treiber schreiben zu können. Im Ergebnis wusste Bruno – und damit auch die gesamte Freie-Software-Gemeinschaft – sehr viel mehr über diese Technologie als sich aus den Spezifikationen hätte herauslesen lassen. Hätte er über die Spezifikationen verfügt, hätte es kein Motiv gegeben, sich so intensiv damit auseinander zu setzen.

Die Geheimhaltung hatte also den gegenteiligen Effekt. Es ist durch sie nicht weniger über Intels Speed-Step-Technologie bekannt, sondern mehr. Leider ist dies bei vielen Hardwareherstellern noch nicht bis zum Management durchgedrungen. Denn aus Sicht des Herstellers ist ein freier Treiber eigentlich etwas, was das eigene Produkt wertvoller und damit der Konkurrenz überlegen macht.

Fünf Jahre Brave GNU World

Mit dieser Ausgabe ist die Brave GNU World jetzt also offiziell seit fünf Jahren aktiv. In der halben Dekade gab es 60 Ausgaben und insgesamt 1232 KByte HTML-Text, die in bis zu zehn Sprachen im Internet veröffentlicht sind. Möglich war dies nur aufgrund der Hilfe von vielen Freiwilligen überall in der Welt – von Asien über Europa bis Südamerika. Für diese Hilfe habe ich eigentlich nur ein Wort: danke! Ebenso gebührt mein Dank

allen, die sich mit ihren Projekten und Hinweisen in die Brave GNU World eingebracht haben. Die interessantesten Ausgaben sind zumeist aus euren Vorschlägen hervorgegangen.

Und natürlich danke ich auch den Lesern der Brave GNU World für ihre Treue. Und wer bisher nur als Leser dabei war, den möchte ich gerne auffordern, sich ruhig selber einzubringen. Dafür gibt es mehrere Möglichkeiten. Sei es über Ideen, Kommentare, Fragen, Anregungen oder sonstige Dinge – speziell natürlich auch Projekte. Wie üblich per E-Mail an die bekannte Adresse [1], es bleibt also alles beim Alten. Bis zur nächsten Ausgabe. (mwe) ■

Infos

- [1] Ideen, Anregungen, Kommentare an die Brave GNU World: [\[mailto:column@brave-gnu-world.org\]](mailto:column@brave-gnu-world.org)
- [2] Das GNU-Projekt: [\[http://www.gnu.org/\]](http://www.gnu.org/)
- [3] Georgs Brave GNU World: [\[http://brave-gnu-world.org\]](http://brave-gnu-world.org)
- [4] „We run GNU“-Initiative: [\[http://www.gnu.org/brave-gnu-world/rungnu/rungnu.de.html\]](http://www.gnu.org/brave-gnu-world/rungnu/rungnu.de.html)
- [5] Cdrtools: [\[http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html\]](http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html)
- [6] DVD+RW-Tools: [\[http://fy.chalmers.se/~appro/linux/DVD+RW/\]](http://fy.chalmers.se/~appro/linux/DVD+RW/)
- [7] KDE-Programm K3b: [\[http://www.k3b.org\]](http://www.k3b.org)
- [8] Ayos Homepage: [\[http://www.73lab.com\]](http://www.73lab.com)
- [9] FreeDB CD-Datenbank: [\[http://www.freedb.org/\]](http://www.freedb.org/)
- [10] Spenden an das K3b-Projekt: [\[http://k3b.plainblack.com/index.pl/donations\]](http://k3b.plainblack.com/index.pl/donations)
- [11] Brave GNU World Ausgabe 48: [\[http://brave-gnu-world.org/issue-48.de.html\]](http://brave-gnu-world.org/issue-48.de.html)
- [12] Ndiswrapper: [\[http://ndiswrapper.sourceforge.net/\]](http://ndiswrapper.sourceforge.net/)

Der Autor

Dipl.-Phys. Georg C. F. Greve beschäftigt sich seit etlichen Jahren mit freier Software und kam früh zu GNU/Linux. Nach Mitarbeit im GNU-Projekt und seiner Aktivität als dessen europä-



ischer Sprecher hat er die Free Software Foundation Europe initiiert, deren Präsident er ist. Mehr Informationen finden sich unter: [\[http://www.gnuhh.org\]](http://www.gnuhh.org)