The Monthly GNU Column

Brave GNU World

This column looks into projects and current affairs in the world of free software from the perspective of the GNU Project and the FSF. In this issue, we celebrate the 60th issue by looking back on five years of Brave GNU World. Also in this month's column: K3b, and hardware drivers. BY GEORG C.F. GREVE

Welcome to the 60th issue of Brave GNU World. In this issue we will be looking at hardware drivers – proprietary drivers on free operating systems, and the KDE burning software, K3b.

Burning CDs/DVDs with K3b

Burning CDs and DVDs has become a daily chore for most users. Linux users had to make do with command-line tools for quite a while. As reliable as programs such as cdrecord from the cdrtools package [5] may be, and although they provide useful support to power users, the command-line seems to be beyond the capabilities of some users (and even administrators). This led Jörg Schilling, cdrecord's developer, to withdraw support for the program.

The cdrtools were released under the GNU GPL, but Jörg Schilling has removed DVD support from the free version. To write DVDs, users need to have the proprietary version. You can avoid this with DVD+RW-Tools [6], although this is back to the command-line.

Front-ends became available for cdrecord at quite an early stage. Most of them were buggy, and not pretty. In many cases, they were restricted to a specific task, and CD ripping was not typically supported.

Sebastian Trueg was disappointed by the front-ends he saw while shopping around for alternatives to Windows in 1998. This led Sebastian to start working on K3b [7]. The appreciation that others have expressed for his work has been a constant source of motivation for Sebastian, who continues to develop the tool.

New Developers

The users' enthusiasm and their feedback were major factors that helped carry the project across the years. Of course, other developers made important contributions. Chris Kvasny developed the video CD functionality (that is, facilities for reading and writing VCDs). Chris has been a major member of the maintenance team ever since. Thomas Froescher contributed the code for reading video DVDs, and Klaus-Dieter Krannich added device support.

The project made a major leap forward in 2003, when Suse gave Sebastian Trueg the opportunity to work full time on K3b within a three-month internship. DVD burning support was implemented during this period, with a little help from growisofs from the DVD+RW tools package mentioned previously.

Finally – a multi-faceted project like K3b would be lost without an attractive appearance. K3b's extremely cool and original design is the work of Ayo [8]. It's well worth visiting his homepage just to check out the fantastic artwork, splash screens and GNU/Linux e-cards. Version 0.11 of K3b also saw the introduction of a new design by Everaldo, the author of the KDE "Crystal Icon" theme.

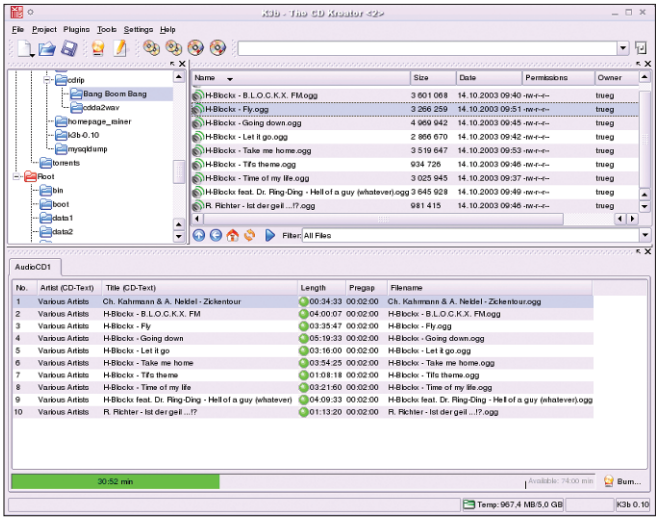

Figure 1: The KDE burning software K3b brings drag & drop to CD and DVD burning, allowing users to create audio CDs from files in Ogg Vorbis, MP3, or other formats.

Added Functionality

K3b is not just a pretty front-end for tools such as cdrecord, growisofs, cdrdao, and mkisofs. Although it uses cdrecord for the burning process, it takes care of everything else itself. The tasks include finding devices, and locating CD directory information, as shown in Figure 2. K3b also determines the burning speed for the medium and tests its size to utilize the medium to the max.

K3b recognizes an audio CD, when a inserted by a user, and depending on the configuration, will either search locally or in the FreeDB Internet database [9] for the artist, title, and tracklist, which it then displays. Users can right-click to pop up a menu that allows them to rip the whole CD, or just specific tracks, and encode them as Flac, MP3, OGG Vorbis, or other audio-formatted files.

You can do the same thing in reverse when burning audio CDs. That is, you can add files in any audio format to a project, and tell K3b to create an audio CD, including hidden bonus tracks if so desired. If all that is not enough for you, K3b has a modular architecture that facilitates the addition of plug-in facilities.

Sebastian wrote K3b in C++, and it uses a number KDE libraries. The program now resides in the *kdeextragear-1* tree of the KDE CVS. As K3b is a free software program under the GPL, it is quite happy to run on Gnome and other window managers.

At present, Sebastian is concentrating on the imminent release of version 1.0. The current version already has the functions for the next release, and developers are concentrating on debugging.

Sebastian anticipates trouble with drives that do not support the "Multi Media Command" (MMC) command set. Although, any drive built in 1999 or later should support this standard. In fact, it might be difficult to dig up a drive that does not have MMC support.

New Features in the Pipeline

The developers have a long wishlist of new features. The next feature on their implementation list, is a facility for copying video DVDs. The major issue here is converting DVD 9 to DVD 5 (by increasing the compression ratio), to use the lower capacity of DVD media more efficiently. A video transcoding module for arbitrary video CD and video DVD projects is also planned. This will allow users to create normal video CDs and DVDs from arbitrary video formats.

The project welcomes developers with experience in video programming, and needs help with migrating transcode to libdvdread for ripping video DVDs. There is also some work to be completed on the Transcode/Mencoder based transcoding layer. This said, help in improving usability is always appreciated.

If development is not progressing quickly enough for you, but you don't have have the time, and/or knowledge, you can always help out with a donation. Check out [10] for more details.

This is a project where the developers have gotten almost everything right, starting with the clean kick-off as a free software project. The team is open to new developers, user-centric, and keen to simplify the GUI as much as possible. After all, reading and burning CDs and DVDs is one of the most basic requirements that most users have of their computers.

Hardware and Linux Drivers

The major question with laptops is to what extent a hardware manufacturer should be allowed to influence a customer's choice of software, and thus dictate that customer's field of applications. The issue is the availability of proprietary drivers for free operating systems.

Hardware without a matching driver is useless ballast. In the case of a laptop, components may even consume power and impact battery life due to a lack of driver support.

Of course, it is technically possible to run hardware with a proprietary driver, and apart from security reservations that one might have about unknown code running with a privileged account, one can normally assume that full hardware support will be available. Unfortunately, there are two things that prevent this.

Firstly, there is the classic case of users only being able to use hardware for the applications envisaged and tolerated by the manufacturer. If an unforeseen problem occurs, or if the user has a different kind of application, there is no simple solution. One example of this is the idea of using the power of graphics adapters as a kind of second processor.

The second thing is that users depend on the manufacturer. The hardware is useless if the manufacturer goes out of business, does not support the required operating system version or type, or loses interest in maintaining a product, preferring to concentrate on pushing newer hardware. Thus, the decision as to whether a user will be able to continue using a hardware product is mainly taken by the manufacturer.

Independence

If free software drivers exist, this problem simply disappears. Users are free to modify the driver (or have it modified by someone who knows what they are doing) to best suit their requirements.

This makes hardware usable, under any circumstances. In some cases, the hardware develops "skills" that the manufacturer may not have thought possible.

The user no longer depends on the manufacturer, and is free to choose, how and for how long to use the hardware.

Freely Available Specs

Manufacturers who are unable or unwilling to write free drivers themselves can at least release a full set of specifications for their products. This means more work than simply maintaining an existing driver base, but it is the next best thing to freedom of choice for the user.

If you evaluate hardware on the basis of its usefulness, hardware with free software drivers is bound to come out on top. Hardware with proprietary drivers is somewhere between fairly useful and useless, depending on your choice of operating system.

Users should be able to insist on a partial or complete refund, and return the hardware, if the manufacturer provides only proprietary drivers. If you ask me, manufacturers of laptops that do not provide drivers for integrated components, despite the added weight and the extra power they need, should be required to pay damages.

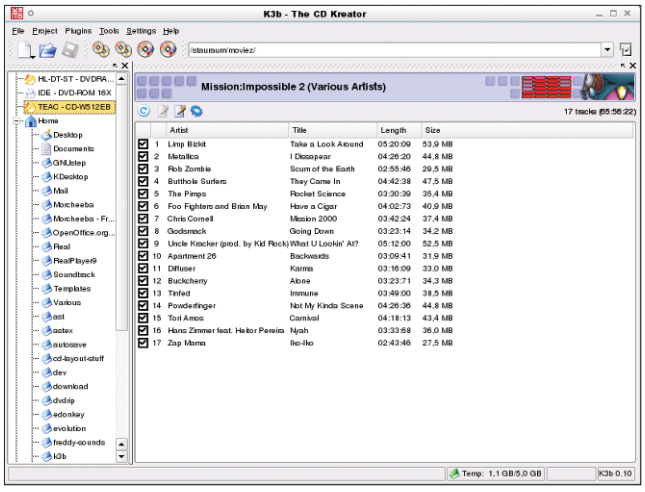

Figure 2: K3b is more than just a front-end for existing programs. It can rip audio CDs and use CDDB to display information on individual tracks.

Proprietary Drivers

It used to be the case that proprietary drivers were supplied almost exclusively for proprietary operating systems, where the additional dependency was more or less invisible against the background of the dependency on the software itself.

Recently, there has been a crop of proprietary drivers. For

example, the wireless network cards in Intel's Centrino laptops, internal modems, and Nvidia graphics drivers.

It might seem logical to welcome the move from "completely useless" to "useful". Many users seem to be welcoming these drivers instead of taking a more critical view.

We have looked into the steady decimation of user freedom, as in "What difference does a little driver make, anyway?" many times in this column. I won't repeat all those arguments.

Most users are not truly aware that, by accepting and using proprietary drivers, they are creating a climate that removes the competitive advantage of manufacturers that supply hardware with free driver support.

Also, this kind of acceptance removes the incentive to free software developers – some development projects have been discontinued for just this reason.

Acceptance of the Status Quo

Just a few years ago, users successfully boycotted Matrox graphics adapters, insisting that the manufacturing provide specifications that would allow developers to program useful free drivers. Today, people are using Nvidia's proprietary drivers without thinking twice and some people even believe that this is progress.

The proprietary driver loader, required to run Intel's Centrino WLAN cards on Linux is a completely different matter. It has been an obstruction to an equivalent free software project, Ndiswrapper [11], although it may be hard to find facts to support this statement.

You can understand the arguments put forward by the manufacturer; it's hard to earn money with hardware, and they need to set up a new line of business. But these arguments ignore two factors.

The problem is artificial, and manufacturers should look to remove it, as they would if there were more pressure from users. Secondly, any money going to the manufacturer effectively prevents the development of a free driver.

Paying for Free Software?

I, for one, would be quite happy to pay twice the price for a product to have a free software driver. If you are thinking about paying $15 for the driver loader, you might like to donate $30 to the Ndiswrapper team instead.

It is up to everyone to create a climate where manufacturers who supply free drivers are more successful than those who do not.

If you insist on using proprietary drivers on a free operating system, you should be aware that you are partly to blame if you purchase a hardware product in future, only to discover that it is not supported by your operating system.

Disclosing Secrets?

Of course, this discussion did not grow up over night, and no-one could reasonably question its usefulness or the ties discussed thus far in this article.

More often, you hear arguments to the effect that manufacturers cannot afford to lose their "Mojo", their secrets, by releasing free software drivers or publishing specs. This kind of argument is unfounded. For example, Intel tried to keep its Centrino Speed Step technology a secret.

This forced Bruno Ducrot to take a very close look at the implementation of this technology, in order to develop GNU/Linux support and create a driver that could handle Speed Step.

As a result, Bruno Ducrot – and thus the whole free software Community – soon knew a lot more about this technology than they would ever have extracted from the specs, if they had not been motivated to take a closer look.

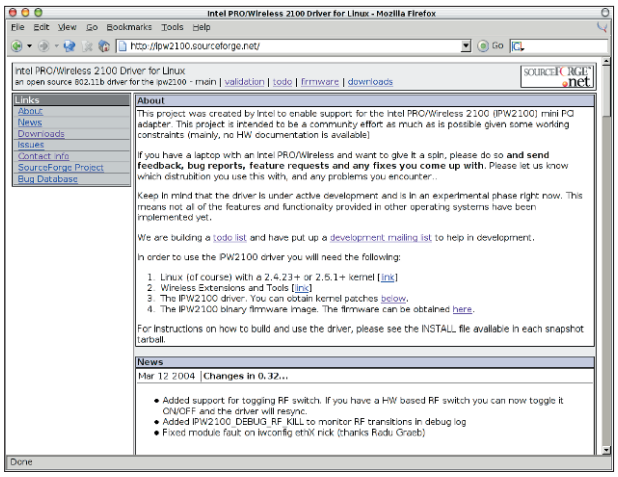

Figure 3: The IPW2100 project homepage. Intel has released drivers for its Centrino WLAN cards partly under the GPL. This mainly proprietary solution is far from ideal.

Trying to keep things secret really backfired on Intel, leading to more, rather than less, information on Intel's Speed Step technology becoming public knowledge. Unfortunately, the management of many hardware manufacturers is not aware of this fact, and management is often responsible for preventing free drivers. From the viewpoint of a manufacturer, a free driver is actually something that adds value to a product, setting it apart from its competitors.

Five Years on

This is Brave GNU World's official fifth birthday. Over the period of half a decade, we can look back on 60 issues, a total of 1232 KBytes of HTML text, which has been translated in up to ten languages and published on the Internet.

This was made possible by volunteers from all over the world. I have only one thing to say to all of you: Thanks!

Of course, thanks to you, the readers of Brave GNU World, for your loyalty. I look forward to your ideas, comments, questions, suggestions, or whatever to the usual address [1]. ■

INFO

[1] Send ideas, comments, and questions to Brave GNU World: *column@brave-gnu-world.org*

[2] The GNU Project homepage: *http://www.gnu.org/*

[3] Georg's Brave GNU World: *http://brave-gnu-world.org*

[4] "We run GNU" initiative: *http://www.gnu.org/brave-gnu-world/rungnu/rungnu.en.html*

[5] Cdrtools: *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/*

[7] K3b, a KDE program: *http://www.k3b.org*

[8] Ayo's Homepage: *http://www.73lab.com*

[9] FreeDB CD database: *http://www.freedb.org/*

[10] Donate to the K3b project: *http://k3b.plainblack.com/index.pl/donations*

[11] Ndiswrapper: *http://ndiswrapper.sourceforge.net/*