The Monthly GNU Column

Brave GNU World

Welcome to another issue of the Brave GNU World! Hopefully not too many readers were affected by pyDDR fever and have blamed their sore muscles on the column.

BY GEORG C.F. GREVE

This month we start by taking a look at the drivers for hardware. We then continue with a new 3D viewer library and finish with a discussion on probelms with licensing.

Drivers for Free Software

One particular situation that is among the most difficult one for many users of Free Software operating systems, especially when pure users are concerned. Assembling and installing new hardware can be done quickly, but the included drivers are only for Windows.

Unless you've spent significant time on Internet research before buying hardware, so you could make an informed decision which hardware to buy, precious hours are now lost trying to locate the right driver. In some unfortunate cases you will come to realize that a particular piece of hardware is not supported. Outdated information, unmaintained links and discontinued development of some drivers make the situation worse.

This is the fault of hardware vendors, who do not care about selling their hardware outside the Windows monopoly and so do not take customers outside that monopoly seriously.

Often hardware vendors would even have to develop each and every driver themselves. There are volunteers to help develop drivers for their own use. In many cases, these developers receive very limited support. Even worse, some hardware vendors actively try to defend themselves against this attempt at increasing their market potential.

Of course there are more and more exceptions to the rule Some hardware vendors are beginning to understand the potential of the Free Software market and either try to provide drivers themselves or further their development. However, these are more or less exceptions confirming the rule,. Drivers written and distributed exclusively by hardware vendors are not the best solution.

Better would be a platform on which vendors develop and maintain drivers together with interested companies and volunteers. This would help to keep drivers maintained years after the production has been discontinued and it also becomes possible to use synergies.

Many boards by different hardware vendors use identical chipsets. So a lot of multiple work could be avoided. Also developing drivers for multiple operating systems or allowing porting to a new hard- or software-platform – creating a new sales market – would be possible.

Hardware vendors willing to enter such a process would see much increased usability for their hardware in combination with a higher customer retention based upon the knowledge that support will not be suddenly discontinued in 1-2 years. Not to mention that the potential market would grow.

The entry portal for such a system would have to be a database, which allows searching for products in order to know about the existing drivers and their status. Users would be able to find out whether drivers exist and whether they are, for instance, stable, only suited for developers, orphaned and/or outdated.

Florian Duraffourg from France hopes to build such a database.

Database for Free Software drivers

Based on PHP, HTML and MySQL, Florian Duraffourg is aiming at creating a driver database [5] specifically for GNU/Linux and GNU/HURD. Anyone looking for a driver should then be able to find out whether a driver exists and which status it has.

By means of a simple Web-access, this would also allow checking directly before buying hardware in a store to see how well it is supported. Florian also considers allowing direct server access, so accessing the database would become possible without HTML and browser; for instance with a simple client program on a wireless device like a PDA or mobile telephone.

Thoughts and plans around this program are still under development, so Florian doesn't have a final domain or dedicated server. Among his considerations was to mirror all drivers on a dedicated FTP server to slow linkrot and loss of information.

Right now his biggest concern is finding volunteers that would like to contribute to development, designing the web site, translating documents into different languages and assembling the driver database.

This seems like an extremely useful project and I hope that many volunteers will join the effort. Also some hardware vendors will hopefully recognize their chance and support this project directly.

Those interested in finding business models around Free Software should also have come to realize that another very interesting opportunity has presented itself in this field.

Should a store or mail order business strategically orient itself towards hardware supported by Free Software like GNU/Linux, this added value would certainly be worth a lot to many people.

Customers would know that they could shop as much as they wanted without having to worry about lack of support. As an additional service they could get a CD with the right drivers for

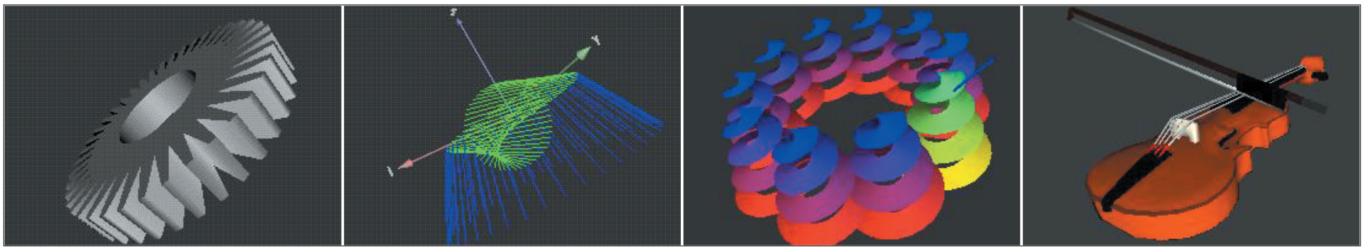


Figure 1: libQGL in action

the hardware they have just bought. Due to customer retention, they are likely to buy other hardware that is often uncritical in terms of being supported, such as hard disks, cases, CPUs, memory from that company. So by supporting the project, even the sales-oriented businesses could gain substantial additional value.

libQGLViewer

The libQGLViewer [6] project by Gilles Debunne has created a 3D viewer library that makes development of 3D applications easier and faster.

According to Gilles, it was a major problem for development of 3D applications that tools like a moving camera, snapshots, simple coordinate systems – although classical and well-known – are rarely part of 3D standard libraries. Even GLUT is using a much lower abstraction level.

In comparison, libQGLViewer provides camera and objects, which can be freely moved with the mouse within a 3D scenario. Also saving snapshots in different formats including vector-based EPS is possible.

The project is based on the QGLWidget class of the Qt library and solely for the purpose of showing the refresh rate it is still using GLUT. This dependency will disappear with the move to Qt 3.1, however. The libQGLViewer itself was – like the Qt library – written in C++ .

The author works in a graphics laboratory in which, previously, everyone was creating their own 3D viewer . Each solution had its strengths, but none was complete. Gilles sought to change this. He also teaches students and sought a way to allow them to become more creative without having to spend much time on the infrastructure.

Therefore Gilles began developing this project. Originally he estimated it to be a weeks worth of work, but in the end it turned out to be a full-time project for a whole year. He now considers the project to be finished, only the documentation needs some polishing by a native English speaker. Also some feedback for the installation routine would be welcome.

Declaring the project finished was a conscious decision, by the way. His declared goal was to keep the viewer as general as possible without focusing on specific applications. Gilles is convinced he has reached that limit and so he has decided to not add any more substantial new features.

According to its author, the most important advantages of the project are its clean, portable and carefully design API, also the library comes with complete documentation and a great amount of commented examples. This makes it possible to generate a simple 3D viewer within one minute and 10 lines of code.

Gilles also sought to emphasize what the libQGLViewer is not. It does not do 3D rendering, because its sole purpose is to provide the viewer – the "draw()" method remains the responsibility of the user and a lot of possibilities exist to generate the scenario. libQGLViewer then allows the user to enter these scenarios and move within them.

The libQGLViewer is published as Free Software under the GNU General Public License (GPL). Apparently there were numerous requests to switch to the Lesser General Public License (LGPL) in order to also allow proprietary applications. However, for personal and political reasons Gilles decided to keep publishing the libQGLViewer under the GPL.

Also he found the generally available information about licenses, possible change of licenses and multi-licensing to be incomplete and sometimes confusing. He is probably not the only person feeling that way, so it might be useful to write a little introduction into the background.

Copyright Introduction

In order to be able to understand the licenses, it helps to understand the background on which they are built. Licenses like the GNU General Public License (GPL) are Copyright-licenses, or Authorship right ("Droit d'Auteur") licenses. Even though the terms Copyright and Droit d'Auteur are almost the same for most practical purposes, some historic differences exist.

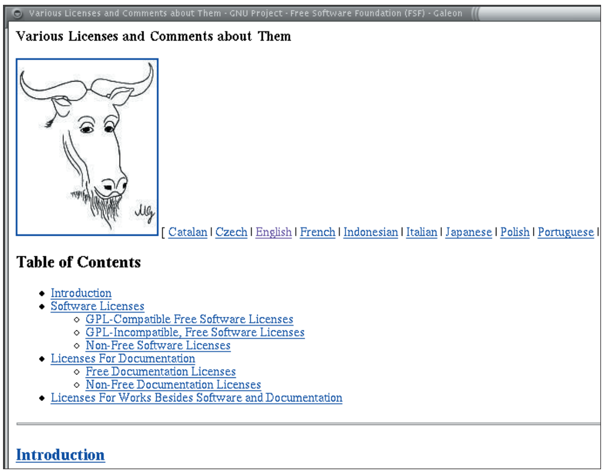
In order to understand these, one has to know that Copyright is an invention of the Gutenberg age, a result of inventing the printing press around 1476. Originally being a pure monopoly for publishers it wasn't intended to give authors any rights in their works. It was not until 1710, when authors could purchase rights in their own works.

The idea of a fundamental right of the author in his or her work – The core of todays Droit d'Auteur – has been propagated mostly by German and French philosophers in the time before the French revolution and its implemented was a major achievement of the French revolution.

This established the Droit d'Auteur as a new legal tradition in order to replace the old, publisher-monopoly oriented system. Today, essentially the continental European countries are following the Droit d'Auteur tradition while the Anglo-American area is still following the Copyright tradition.

To create common international ground, there have been harmonizing processes and agreements that are commonly referred to by the cities of Berne and Stockholm in which they have come to pass in. Because of this, most issues are practically not different despite the very different basis. With one exception.

Contrary to Copyright, Droit d'Auteur knows a personality right of the author, which – like other human rights – is inalienable. Independent of what any



Various Licenses and Comments about Them - GNU Project - Free Software Foundation (FSF) - Galeon

Various Licenses and Comments about Them

[Catalan | Czech | English | French | Indonesian | Italian | Japanese | Polish | Portuguese |

Table of Contents

- Introduction
- Software Licenses
  - GPL-Compatible Free Software Licenses
  - GPL-Incompatible, Free Software Licenses
  - Non-Free Software Licenses
- Licenses For Documentation
  - Free Documentation Licenses
  - Non-Free Documentation Licenses
- Licenses For Works Besides Software and Documentation

Introduction

Figure 2: Licenses galore

contract says, the personality rights of the author can never be limited. In fact, a contract that is (seemingly or for real) trying to do this runs a risk of being declared invalid in court.

Only the so-called exploitation rights are transferable. There are single and exclusive exploitation rights; the proprietor of the exclusive exploitation rights can provide an unlimited amount of single exploitation rights and defend his rights in court. So the exclusive exploitation rights are for most practical purposes similar to the Anglo-American Copyright.

Free Software Licensing

So based upon these exclusive exploitation rights and/or the Copyright the owner of these rights can choose the license under which the software is distributed. In the case of Free Software that means choosing a Free Software license [7].

As the owner of the exclusive exploitation rights/Copyright can issue an unlimited amount of single exploitation rights (and therefore licenses), it is of cause always possible to issue the same software under several licenses. Whether these licenses are Free Software licenses or proprietary makes no legal difference.

Should the exclusive exploitation rights be with several people, because a piece of software has several authors and they have not chosen to combine their rights with one fiduciary, for instance, all of these authors must agree with the licensing.

So the license is given by means of the exclusive exploitation rights – which are more or less the same as the Anglo-American Copyright – as a single exploitation right.

It should be understood that only the owner of the exclusive exploitation rights can also defend the license in court. Also relicensing may become practically impossible if a large number of authors has been working on a project.

For this reason – as presented in Brave GNU World issue #48 [8] – the FSF Europe has published the Fiduciary Licence Agreement (FLA) [9] in February 2003.

Questions about the FLA

In reaction to the publication of the FLA, some questions were raise that should be briefly addressed here.

One question raised a few times was whether the FLA should replace the GPL. That is of course not the case, because the GPL is one Copyright based license granted by means of the exclusive exploitation rights, while the FLA is dealing with transfer of exclusive exploitation rights, which is one level before the license.

Another question was with reference to the keeping the possibility open for the assigning party to do dual-licensing, since the FLA is retransferring an unlimited amount of single exploitation rights back to the author. The question raised was whether it would be possible to then enforce these licenses in court as the author is no longer holder of the exclusive exploitation rights.

If you imagine this case, author A would assign her rights to the FSF Europe and give a single, proprietary license to company B with the contract clause of not passing this license and software on to third parties. Now company C is selling software in which the software of author A is obviously used in a proprietary way. What does this mean?

There are two logical possibilities. Normally the software would be published under the GNU GPL by the FSF Europe, so the first possibility would be that company C has violated the GPL by using the software proprietary. Of course the FSF Europe would investigate that. Only when company C provides written evidence that they did not take the GPL version, but rather the proprietary version of company B, will they be cleared of that suspicion.

In that case, company B has violated its contract with author A, however. And the author still has the right to take legal steps against contract violation. So the author can take legal steps and cut the chain of transferral of rights in the transferral to company B.

Of course I need to state explicitly that this will in doubt always require intensive checking by an accredited and fully trained lawyer, which I am not. But I do hope that I was successful in making the overall picture clearer to non-lawyers without upsetting the experts in legal issues with oversimplifying matters in their eyes.

Closing word

Enough for this issue. As usual I am hoping for numerous suggestions, questions, comments and of course information about interesting projects and new developments to the usual address [1]. ■

INFO

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

[2] Home page of the GNU Project: *http://www.gnu.org/*

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

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

[5] Free Software driver database: *http://drivers.linux.free.fr/index.php*

[6] libQGLViewer home page: *http://www-imagis.imag.fr/Membres/Gilles.Debunne/CODE/QGLViewer/*

[7] Free Software licenses: *http://www.gnu.org/licenses/license-list.html*

[8] Brave GNU World issue #48: *http://brave-gnu-world.org/issue-48.en.html*

[9] Fiduciary Licence Agreement (FLA): *http://fsfeurope.org/projects/fla/*