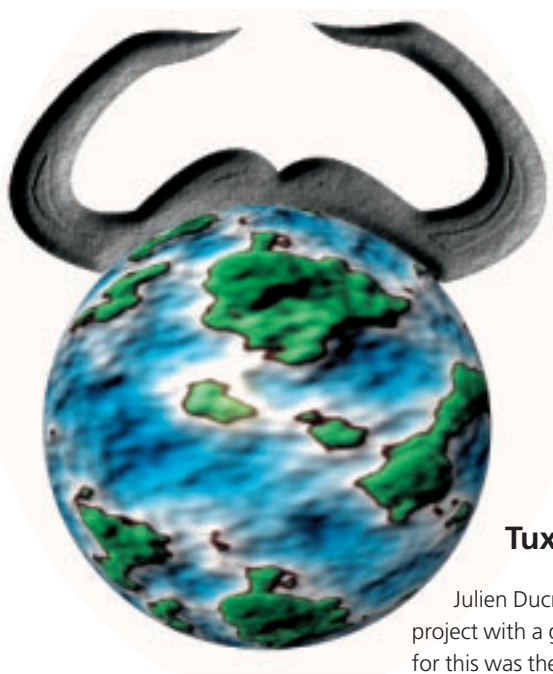


The monthly GNU Column

BRAVE GNU WORLD

GEORG C.F. GREVE



Welcome to another issue of Georg's Brave GNU World. This month I'll make the philosophical background a central issue focusing on the question of commercial Free Software. But first I'd like to introduce a few projects.

TuxFamily.org

Julien Ducros has started the "TuxFamily.org" project with a group of volunteers. The inspiration for this was the American SourceForge project, which provides a central infrastructure for development and presentation (like Web server, FTP server, CVS server and so on) to projects.

It is the belief of Julien Ducros that the European and African community also need such a service on their continent. This is what the French TuxFamily project wants to provide.

TuxFamily itself is clearly oriented strongly towards Free Software. The software used for hosting (vhffs) is published under the GNU General Public License itself. Additionally the TuxFamily only accepts projects that qualify as Free Software.

So projects looking for a new home might consider joining the TuxFamily.

Chrooted SSH CVS-Server HOWTO

Very often relatively small companies are doing good work for Free Software whilst being almost completely unknown outside their respective countries. The French company Idealx is one of them.

On the community homepage of Idealx several interesting modules and documents about Free Software can be found. They solve several standard problems and of course everything is available under the GNU General Public License and the GNU Free Documentation License.

Among the packages is a Python module which allows creating calendars in CGI-scripts. A XML-customizable CVS-notify script to automatize actions during check-in of new versions or a binding of Erlang to Python can also be found.

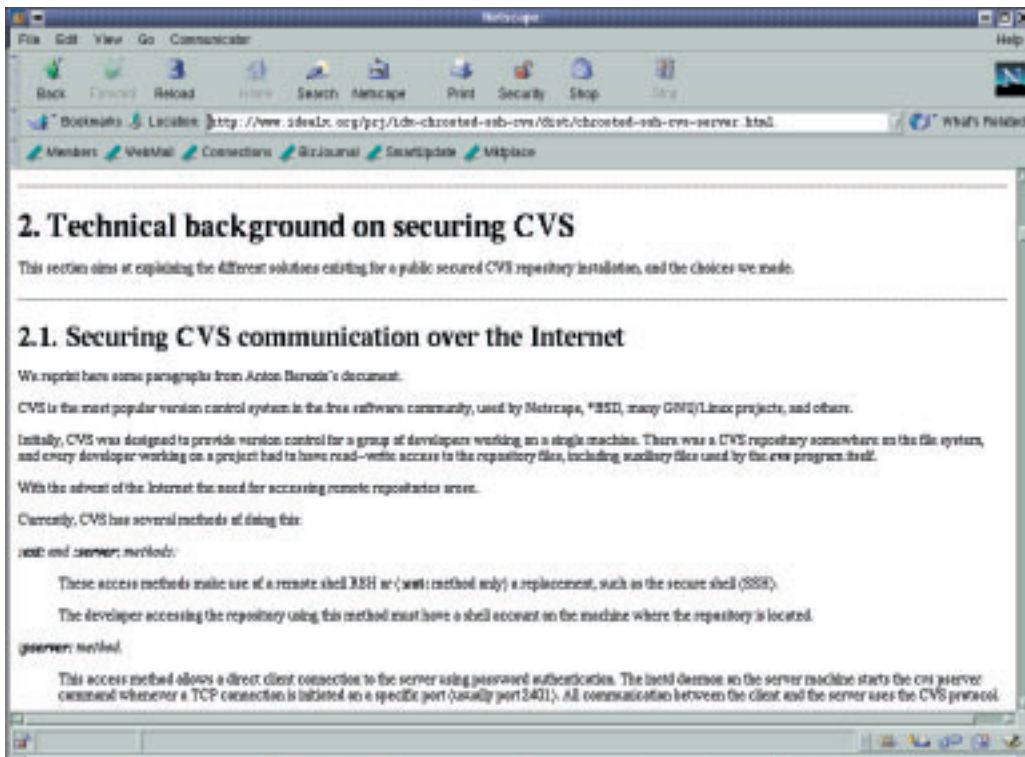
A very interesting document is also a Howto written by Olivier Berger and Olivier Tharan, which deals with the set-up of a very secure and well-insulated CVS server.

Programmers normally know of the advantages of the "Concurrent Version System" (CVS), but CVS is a tool many people underestimate, so I'll give a short introduction.

It should be common knowledge that software is normally written in the form of source code. This source code is improved by one or several authors during the development process. When doing this, the problem of coordination of several developers arises because normally every developer will be sitting at his personal machine to make changes there. This means that source code often changes simultaneously at different places.

To solve this problem, CVS (like other "version control systems") has a central gathering point, the so-called "repository." Every developer communicates directly with the repository, receiving updates by others or submitting their own changes. Of course it can happen that two authors change the same part, but the different ways to resolve or avoid such conflicts are not important right now.

What is important is that the CVS repository does not just contain the current version, it also saves every change. That way the development



The Chrooted web site

process can be tracked step by step later. Also, it is possible to go back to old versions in order to fork a project and have different development processes run in parallel with the option to merge them again later.

All this is not just relevant to developers. Considering the fact that source code are simple ASCII files in most cases, it is immediately obvious that this can be used for any sort of data, especially if it can be expressed in text-only form.

Web sites, documents, email-archives and much more are perfect areas for the use of CVS.

The neuralgic point of a CVS server are its access rights. There are several possibilities for this that have different levels of security. In most cases an account on the CVS server requires an account on the system and some methods of authentication transmit passwords in clear text, so they can easily be read by others.

Even when this is being avoided by using SSH, it is very often not desirable to give every CVS-user full access to the system.

The Howto mentioned above describes very well how to configure several CVS servers in parallel



Autogen

on a server while only granting users the necessary rights to access the specific repository. This should allow the majority of intermediate users to install a secure CVS server on their machine in order to use the advantages of CVS mentioned above.

AutoGen

It is always a pleasure to introduce new GNU-Projects. The one I would like to introduce this month is AutoGen by Bruce Korb. It is a tool designed to make creating and maintaining programs with large amounts of repetitive text quite comfortable.

A classic example for this are loops for the commandline evaluation. In the worst case, text is repeated with cut & paste for every option in order to put the values in some place where other functions that need them will be able to find them. Since this is very much a standard problem, AutoGen has template called "AutoOpts" for this.

AutoGen does bear certain similarities with m4, the traditional UNIX macro processor. But it is superior to m4 in many aspects like a simplified way of adding new parameters to functions. Also AutoGen supports nested collections of values. Because of this, Gary V. Vaughan, the most active developer of AutoGen besides Bruce Korb, would like to see autoconf start using AutoGen instead of m4.

According to Bruce one of the special strengths of AutoGen is the clear separation of templates and definitions, because this makes the templates much more flexible. Also all data is address by names, not positions, which allows restructuring and resorting files. On top of this old definitions can become obsolete without having to change old files which

increases compatibility. And of course all definitions can be nested.

Through variables marking the locations for replacements it can be defined with the help of key words which parts are being left out or repeated. AutoGen also has much better ways of controlling the output than the C-preprocessor.

When asked about the disadvantages of AutoGen, Bruce Korb says that it is widely unknown and it is also much too easy to let templates look like hieroglyphs. The static evaluation of definitions is currently the biggest limitation of AutoGen, but it is planned to make this dynamic in future versions.

Besides this AutoGen is pretty much finished and so mostly feedback to increase the portability is needed. So far it is known to run on GNU/Linux, BSD, SVR4-5, HPUX, SCo OpenServer and Solaris as well as Windows NT provided CygWin is installed.

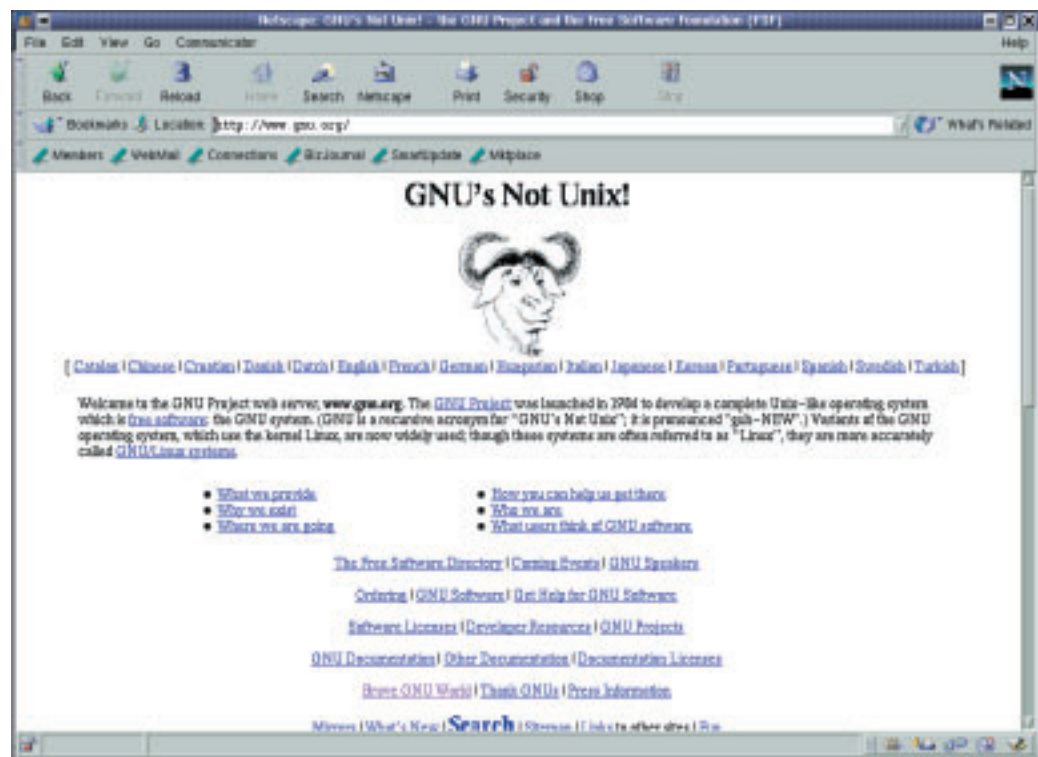
Although AutoGen itself is licensed under the GNU General Public License, some add-ons are under the LGPL and FreeBSD license or public domain.

Free Software and Commercialization

I have received mail about this topic for instance by Tommy Scheunemann, who asked about "Industry and GPL" and on the discussion mailing list of the FSF Europe a pretty controversial debate about whether it was legitimate to make money with Free Software has taken place. This made me aware that there are still some open questions in this area.

Before the question about commerce and the interaction with industry can be understood, it is

Where all your GNU fun should begin



mandatory to take a look at the definition of Free Software. The first step is always to understand that "free" in Free Software does not stand for "gratis" but rather for "freedom."

But what freedom does this refer to? The most precise definition for Free Software are the four freedoms of the Free Software Foundation. The "Debian Free Software Guidelines" were derived of these and have provided the base for the "Open Source Definition". Technically, all three definitions were written to describe the same licenses. Because the four freedoms are the most compact definition I will only be talking about them, however.

The first freedom is to be able to use a program for any purpose. Restricting the use in any way would immediately mean a program does not qualify as free software.

The second freedom allows to study a program in order to learn how it works and to change it accordingly to ones own needs. Having access to the source code is a precondition for this.

Freedom No.3 allows to make copies and pass them on while Freedom No.4 is the sum of freedoms 2 and 3. It says that you must have the freedom to pass on improvements for the benefit of others. Like freedom 2 this requires access to the source code.

It is important to be aware that having the freedom to do something also includes the freedom to not do it - which applies especially to freedoms 2, 3 and 4. There is no obligation to copy or modify a program and also no obligation to pass it along. In fact the requirement to make changes public was what let the "Apple Public Source License" fail to qualify as a Free Software license.

The question whether a piece of software is Free Software or not is decided by its license. Of the licenses for Free Software, the GNU General Public License is the most widely used. Besides the GNU Lesser General Public License, which is a variation of the GPL, the FreeBSD license has the biggest practical importance.

So what about commercialisation? Free Software very deliberately does not make a difference between commercial and non-commercial use. A limitation for commercial purposes would even violate the first freedom - so Free Software is always also commercial. Combined with the knowledge that there is no requirement to pass it on, it immediately becomes clear that Free Software can even be sold.

It should be clear now that Free Software can be commercial although it doesn't have to.

The currently predominant commercial model for software in the industry is the proprietary one, where the price is artificially raised by limiting these freedoms. So a company could be tempted to take away the freedoms to shortly increase its bottom line.

This happens in two ways. Licenses like the FreeBSD license are based on the assumption that noone

Info

Send ideas, comments and questions to Brave GNU World column@brave-gnu-world.org

Home page of the GNU Project <http://www.gnu.org/>

Home page of Georg's Brave GNU World <http://brave-gnu-world.org>

"We run GNU" initiative <http://www.gnu.org/brave-gnu-world/rungnu/rungnu.en.html>

"Tuxfamily.org" home page <http://www.tuxfamily.org>

Sourceforge home page <http://www.sourceforge.net>

Idealx community home page <http://www.idealx.org>

Chrooted SSH CVS-server HOWTO <http://www.idealx.org/prj/lidx-chrooted-ssh-cvs/dist/>

Concurrent Versions System (CVS) home page <http://www.gnu.org/software/cvs/>

AutoGen home page <http://autogen.sourceforge.net>

GNU m4 home page <http://www.gnu.org/software/m4/>

The FSF Free Software Definition <http://www.fsf.org/philosophy/free-sw.html>

Debian Free Software Guidelines

http://www.debian.org/social_contract#guidelines

Open Source Definition <http://www.opensource.org/docs/definition.html>

Free Software licenses <http://www.fsf.org/philosophy/license-list.html>

would put his personal interests above the interests of the public. They deliberately allow proprietary relicensing.

The GNU licenses have a "proprietarisation protection" to avoid this. If you directly base your success on the work of others, you cannot release the result under a proprietary license.

The only program to go back to the proprietary scheme in this case is to write modules that are technically isolated from the original program, which requires additional work and is not always possible.

In both cases the proprietarised end-product is normally sold as "value-added" software with the goal to convince the user to give up his or her freedoms. This can happen commercially or non-commercially.

So despite the protection offered by the GPL, in the end only the awareness of the users can prevent the return to the proprietary model.

To summarize: without question there is commercial Free Software just as there is non-commercial Free Software. It is a question of choice. Instead of being distracted by this question we should rather make sure to not lose track of the freedom as it is the fundament of the whole movement.

Enough for this month

Okay, that's it for this issue. I hope the question of commercial Free Software has become clearer now and I managed to explain things in an understandable way.

As usual I am requesting comments, questions, ideas, inspirations and project introductions to the usual address. ■