The monthly GNU Column

BRAVE GNU WORLD

GEORG C. F. GREVE

Welcome to Georg's Brave GNU World. As the last issue has shown, Free Software in schools is an interesting area with a lot of potential. Therefore I have another topic from it to start with this month.

GNU/Linux TerminalServer for Schools

The "GNU/Linux TerminalServer for Schools"-Project tries to provide an easy means for the installation and administration of a GNU/Linux based terminal server.

Background info: a terminal server provides the full functionality of a system to several users. Their respective workplace machines are merely terminals. This has a lot of advantages.

First of all old hardware can still serve as a terminal. Even a 486 is sufficient for this, so the solution is very cost-efficient. Also the system administrator only has to administrate a single system, which saves a lot of time. Thanks to the central structure, backups are also easy to make.

The project began on a mailing list of the "Freie Software und Bildung e.V." ("Free Software and Education Association"), where it has been proposed by Hans-Josef Heck. At a congress in November 2000, it was decided to push work on this project. It is mainly maintained by Christian Selig, who is being supported by Georg Baum and Jason Bechtel with this. Jason has joined the team through the "Linux Terminal Server Project" (LTSP), which provides the core functionality of the "GNU/Linux TerminalServer for Schools" project.

The main task of the project is not to write software packages but rather to create a good and easy-to-use installation and configuration system. The administration is based on a webmin module written in Perl for this purpose. A CD is planned for easy installation.

On of the things considered crucial for the CD and the configuration is distribution-independence and internationalization. Translations into French (by Joel Amoros), Swedish (by Michael Habbe) and Spanish (by Angel Eduardo Porras Meza) are already

Using software from the Linux Terminal Server Project (www.LTSP.org), you can turn the lopener into a diskless thin client computer.

available and further translators are quite welcome.

According to Christian, the special strength of the project is that it isn't special, it is rather an easy solution to a common problem. The weak point in his eyes is that there are no statistics about the hardware and network-bandwidth needed for a certain amount of users. Although the LTSP has a lot of users, it is hard to determine authoritative numbers for this in the same way proprietary vendors do it. Although those numbers are very often only rough estimations themselves they present themselves in a very convincing way to the end-user.
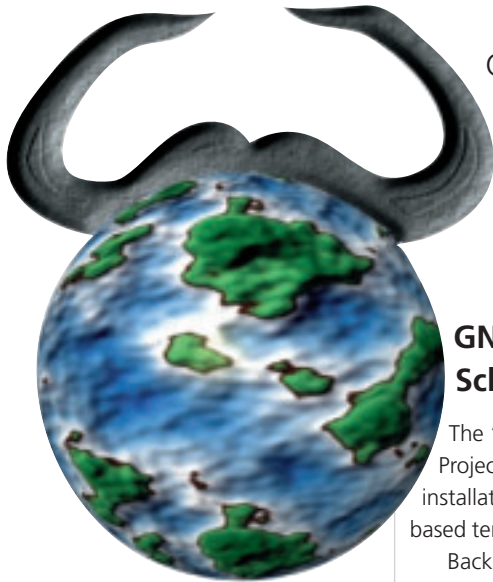
Short-term goal is to complete the CD for installation on all common distributions. To reach this goal, there is still need for beta-testers that are willing to probe the CD and the administration program for weaknesses. In the long term, it is planned to also include other educational software on the CD in order to further spread the use of Free Software in schools. Additional information about this can be found on the Ofset homepage.

The LTSP and the administration module are released under the GNU General Public License. The license of the documentation is not clear yet, but only licenses accepted by the FSF are perceived as acceptable. Personally I would like to see the project make use of the GNU Free Documentation License (FDL).
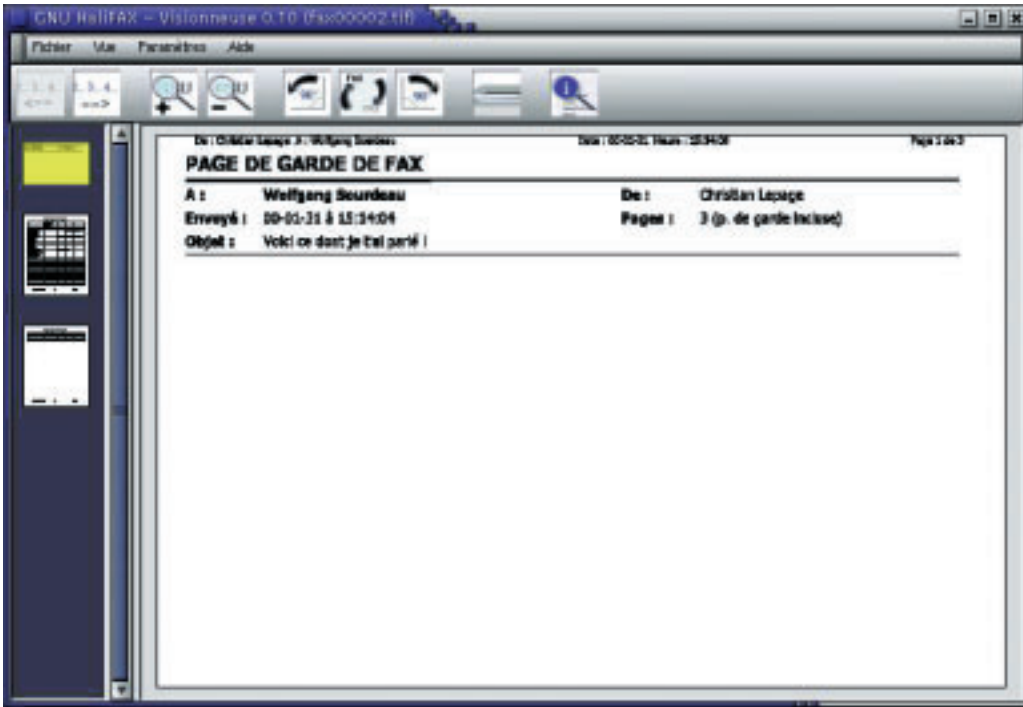
I'll continue with an update about a project that has been mentioned in an earlier issue.

GNU HaliFAX

In Linux Magazine Issue 4 I wrote about the GNU FaXile project, which has the goal to create a complete and comfortable fax environment in the GNU Project. The project has been merged with GNOME-GFax now and been renamed to GNU

HaliFAX with the active page as the yellow thumbnail

HaliFAX. Of the planned functionality (see issue #4), the GNU HaliFAX viewer (ghfaxviewer) and the GNU HaliFAX sender are already usable.

The fax viewer is already pretty advanced and has an easy to use graphical user interface, does anti-aliasing, has an improved zoom algorithm and of course also allows to print faxes.

Some things like a binding of GNU HaliFAX to SANE as well as a project-management part are planned; the maintainers, Wolfgang Sourdeau and George Farris, still have a lot of work to do. They received a lot of help from Till Bubeck who did many things on gfax and also did a German translation for the ghfaxviewer. The original translator into German was Thomas Bartschies, the Chinese translation was done by Kevin Chen and the Polish by Zbigniew Baniewski.

Help in form of funding, allowing Wolfgang and George to concentrate more on this project, would be very welcome.

Although the next project may not be directly relevant to many readers, it is certainly extremely interesting.

GOSSIP

The "GOSSIP Simulation Environment" by Marius Vollmer works on the creation of a simulation environment for use in communications engineering and digital signal processing. It is implemented as an extension to Guile, the Scheme implementation of the GNU Project. The project consists of essentially four parts, which are the simulation-engine (gossip-sim), the tool schematic capture tool of data (gossip-ed), a group of supporting libraries (gossip-lib-*) and an extension that makes it possible to read VHDL-files (gossip-vhdl). The description of the simulation is text-based through Scheme programs which are

being executed by gossip-sim. Currently gossip-sim can only work with synchronous data-flow. Asynchronous data-flow or discrete events should not be impossible, but the implementation of the necessary simulation engines and their interaction has a pretty low priority for Marius Vollmer. The 1.0 release is almost ready: all features planned for it are already implemented in gossip-sim and the other parts are also almost ready for real use. For this the author would like to see constructive criticism especially for the simulation-engine, because he is no expert in simulation-techniques. In his eyes GOSSIP should provide a powerful tool with flexibility and simple structure as its primary attributes. The way some software packages patronize the user is something he especially dislikes, and he would rather give the user the opportunity to determine his own needs himself. The biggest problem is something that manifests as the old chicken and egg problem. GOSSIP is still pretty young, so it lacks simulation modules, which would have to be written by users. But they will only write them if they can use GOSSIP, which requires many good simulation modules. This project opens the possiblity to strengthen the scientific principle of freely exchanging knowledge and its traceability

through Free Software in the field of digital signal processing. Therefore I would like to encourage everyone using such software to start using GOSSIP. Anyone having experience with packages such as COSSAP, SPW or Cocentric SystemStudio should already be familiar with the concepts of GOSSIP, which should greatly simplify a transition and porting existing modules.

Now I'm coming to the next project, which may seem pretty abstract at first but it is dealing with a very important task.

Jude in action
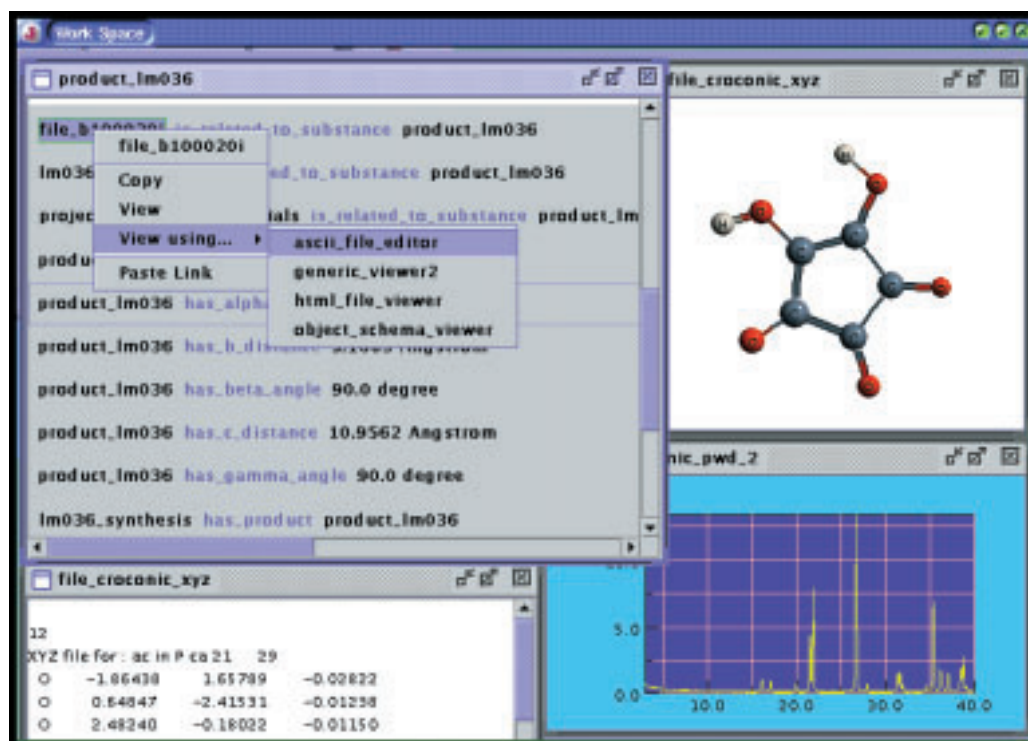
Jude

Jude is part of the thesis of Massimo Zaniboni. He developed it in order to create a "Workgroup Application" for the Crystal Engineering Laboratory of Ciamician, the chemical department of University of Bologna, in Italy. Fortunately he did all this under the GNU General Public License.

Jude is essentially a toolkit or framework for application development that allows the implementation of solutions for the "Data-Management" and "Workgroup Application" areas while being simple to use for users and developers at the same time. The server-side is based on a object-oriented model; the user-side presents itself in a agent-based compound-document way.

The technologies to create such solutions such as relational or object-oriented databases, document-management systems, XML documents, agent-based systems or Java are well-known. Implementation of the desired structure is very often problematic, though, especially if only one or a few of these techniques are being used. Jude tries to provide all the advantages of these technologies in a single coherent development environment.

Jude allows the developer to enter a simplified and abstract representation of the problem in a very high-level declarative and object-oriented language in order to get a fully functional workgroup application.

This allows developers to access many existing modules and makes reusing old code much easier. The user only sees a coherent and simple to use environment based on Java and Swing as the interface to documents and structured information.

Currently Jude is still in the alpha-stage and hard to install, so normal users should better not

give it a try yet. Consequently, making it ready for production use is Massimo's next goal. Afterwards he would like to expand it with a transaction-manager, encryption, offline-capabilities and PDA-support. But it will be a long way until this will be ready. Help or sponsors are very welcome. So if you think this could be an interesting project to spend your time on, feel free to get in touch with Massimo.

I'd say this was abstract enough for this month and I'll now come back to an area that everyone of us has contact with on a daily basis, the Web.

HyperBuilder

HyperBuilder is a project by Alejandro Forero Cuervo who originally began it in order to have a tool to manage big static web sites with more than 100 pages.

In order to allow this, the reusability of information was a crucial design factor since it is rather annoying to insert the same header and footer on 200 pages by hand. Although HyperBuilder is still a good choice for maintaining large static web sites, its real strength is now on the dynamic side.

HyperBuilder runs on the webserver and parses the documents on demand, allowing for maximum dynamics. For this, it is best used as an Apache module, although it is also possible to have it run as a CGI-script, which is much slower.

The web site files themselves are written in a kind of extended HTML that is easily understandable and editable. The HyperBuilder modules can be included as HTML-like tags. Modules for several standard problems like message boards, polls, SQL-backends, user authentication

and more are already implemented. Additionally it is also possible to directly include Perl or Scheme (Guile) code in the files; if desired even both in the same file.

HyperBuilder provides great advantages for non-programmers because it is very easy to learn how to use it. For instance the line <p>Visits: <counter src="file" id="counter_id" inc="1" show="yes"></p> is all that is needed to put a counter on a web page. The alternative would be to write the counter yourself, which will be much longer than the single line above, or download a CGI script by someone else for this purpose and integrate it into the web page. So HyperBuilder lets you get rid of many CGI-scripts.

But programmers also benefit from HyperBuilder. If it is planned to create a web-interface for an application, it is definitely a very good idea to implement this in a HyperBuilder module. This way it become possible for every user to structure and compose the interface according to personal preferences and taste. It also allows the developer to forget about layout and to focus on the functionality instead of the interface and its graphical details.

The HyperBuilder itself has been developed in C under the GNU General Public License with POSIX threads for performance reasons. It runs on Unix-systems and has been tested on different versions of GNU/Linux, Solaris/SunOS, *BSD and Irix. As languages for the dynamic creation of web pages it is possible to use Perl, C and Scheme.

HyperBuilder is fully functional. Right now, the biggest problems right now are the lack of documentation and not enough users to find the remaining problems. Alejandro is especially unhappy about the lack of documentation and offers to coach volunteers for this on the project internals.

Further plans are to write more modules like a module for creation of pages from XML files, a module for communication through XML-RPC, a module to include the whole functionality of GIMP and modules for Ruby and Java.

As a side note I should mention that there is already a complete portal-site called "FuWeb" based on HyperBuilder available under the GNU General Public License - so it can easily be used for own projects.

This should be enough to give you an idea of what this project can do and I can only recommend taking a look at the homepage that also contains examples for the mentioned FuWeb portal.

Brave GNU World internals

Finally I can announce that the Brave GNU World gets translated into another language now. Thanks to Fernando Lozano and Hilton Fernandes who joined the Brave GNU World-family, the Brave GNU World is also available in Portugese now.
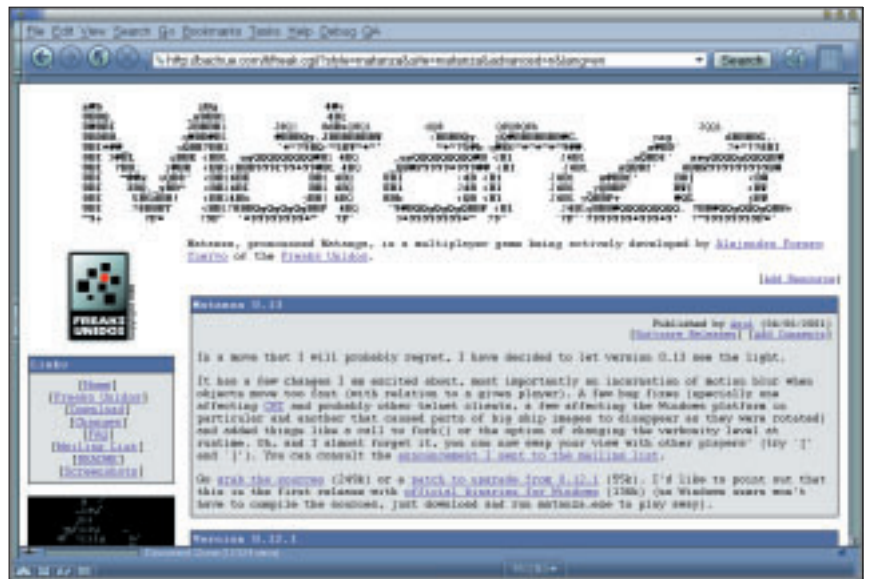
Thank you so much, guys!

Now this column can be read in seven languages (German, English, French, Japanese, Spanish, Korean, Portugese), something I certainly did not expect when I started it.

Also I'd like to welcome Gero Takke, Michael Scheiba and Steven R. Baker in the family. These three have taken over the Brave GNU World web site. Together with Alejandro Forero Cuervo, the author of HyperBuilder, who volunteered to help, they will do the upcoming redesign of the Brave GNU World web site.

Enough for today

Alright, that should have been enough for this month. I hope to have provided some interesting input and as always I'm hoping for tons of email with ideas, feedback, comments and project suggestions to the known address [1]. ∎

Auch die Matanza-Webseite baut auf HyperBuilder.

∎