The monthly GNU Column BRAVE GNU WORLD

GEORG C.F.GREVE

Welcome to another issue of Georg's Brave GNU World. This month, a project aiming to carry GNU into the Apple world will be our starting point.

GNU-Darwin

The GNU-Darwin project works on porting the GNU system to Darwin, the basis of the proprietary MacOS X. Darwin itself is based on FreeBSD and Mach 3.0 and currently runs on PowerPC-based architectures. Apple is working on a port for Intel-architectures, however. So after GNU/Linux and GNU/Hurd there will be a third GNU-based system available.

A very interesting feature of the GNU-Darwin system is to be able to run Macintosh applications parallel to the well-known Unix-programs. This allows the direct comparison of Macintosh & GNU-based programs, something that has been impossible before. So this feature makes GNU-Darwin very well suited for Mac/Unix hybrid-structures and with the port of SAMBA to GNU-Darwin heterogenous networks with Mac/Unix/Windows will be easy to implement.

But of course the main use of GNU-Darwin is not to be able to run proprietary software. It will rather create another bridge between the Macintosh platform and Free Software as the user will quickly realise that there is more Free Software for his computer than there is proprietary software. Software developed on GNU-Darwin is better-suited for interoperability with MacOS X, too, so Free Software can be brought into this world.

The big amount of available Unix-based Free Software and the unique capabilities for heterogenous structures and parallel development are strong arguments for GNU-Darwin instead of MacOS X.

In addition, GNU-Darwin offers several advantages over the LinuxPPC-Project which is the reason why a user-migration to GNU-Darwin can be seen. First of all GNU-Darwin is (like GNU/Hurd) microkernel-based which gives it capabilities that the Linux kernel cannot match. Also the hardware support for Darwin is done by Apple itself as it also is the basis for MacOS X. So a better hardware support is to be expected here.

But not everything is better. Although the contributions of the GNU-Darwin project are of course released under the GNU General Public License, Darwin itself has been released by Apple under the Apple Public Source License (APSL). In version 1.1 this license did not qualify as Free Software for three important reasons.

First of all it was forbidden to make changes for personal use without making those changes public. In the perception of the GNU Project, the right to change things for personal use only is closely related with the right for privacy; that is why the GPL was designed to allow this.

Furthermore the developer and non-scientific user of a modified version was forced to report to a specified institution (in this case Apple). This central control is in direct contradiction to the thought of Free Software.

And finally there was a disclaimer that allowed Apple to terminate the license and stop the further use of the software at any point if copyright or patent claims are being made against Apple. This made every user on this planet dependent on the very problematic U.S. patent-system.

Version 1.2 of the APSL was released in January 2001 and it solves a big part of the problems - the restriction on private modifications and the disrespect for privacy remains, however.

So in a way the APSL takes step after step into the direction of the NPL which is definitely a Free Software license although it does allow a proprietarisation of the source-code. Even if the APSL finally gets to the point of the NPL, it would still be less than satisfactory as it would probably still be incompatible with the most often used Free Software license, the GPL.

So the situation is comparable to that of KDE a few years ago in a way, as it is about a clean Free Software project built on a weak fundament and hence in danger of being cut off at the knees by legal means. Because of this the GNU-Darwin project takes a stand to release Darwin under the GNU General Public License. Especially for this advocacy work, Michael L. Love, one of the participants of GNU-Darwin, asks for support from the community. In the eyes of the GNU-Darwin project, GNU-Darwin will only be truly free once Darwin is GPL'ed.

The current technical issues are porting more packages to GNU-Darwin and the creation of a CD-Distribution. In the long run it is planned to put the special abilities of GNU-Darwin to good use.

As the team currently consists of only six active developers, there is a wide field of possible activities for interested people to choose from. Especially developers with experiences regarding Mozilla, SDL, GNOME & Audio-Support (ALSA) will be received with open arms.

By the way: the origin of GNU-Darwin is the desire of Michael L. Love to use his Apple for protein-crystallography as this is his normal occupation.

Which brings me to a handful of small but rather useful projects.

get_file

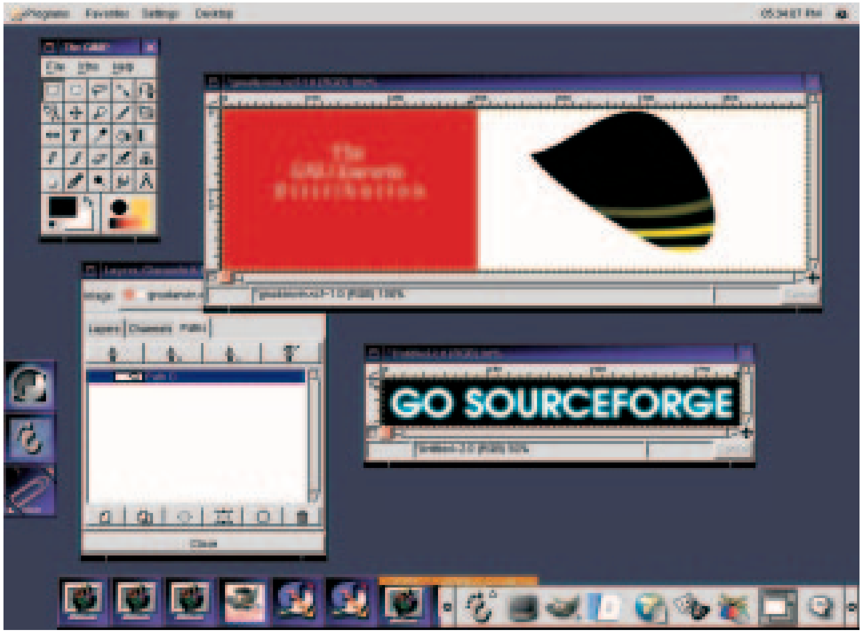
Daniel E. Singer wrote get_file, a small fileselector written in Bourne Shell. The use of this project is apparent to everyone who every tried to have the user select files in a shell-script. Although being called by other scripts is obviously the main use of get_file, it can also be used as a commandline tool.

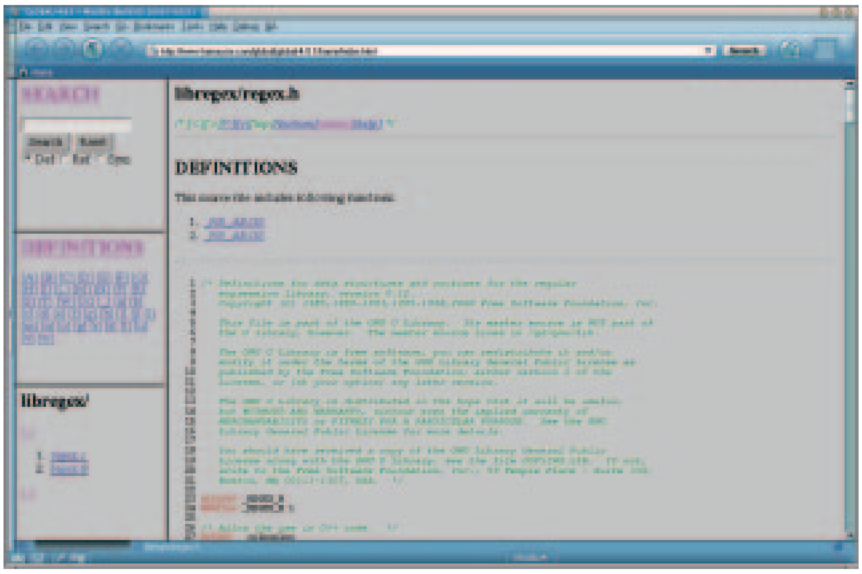
Since it is entirely written as Bash-Shellscript it doesn't have to be compiled and is easy to customise. Additionally it has the capability to associate escape-sequences with certain objects, can change file-modes, execute programs and deal with wildcards. It also has an online-help.

Even if it isn't very efficient because of being a shell-script, this project should prove useful for "on the fly" solutions.

[below]
Gimp is already running on GNU-Darwin

[bottom]
GNU Global





GNU GLOBAL

GNU GLOBAL is a system source-code tag system that makes administration and referencing of source code easy. It has been written by Shigio Yamaguchi who published it under the GNU General Public License, which made its recent declaration as an official GNU project unproblematic.

Big amounts of "flat" source-code distributed over different directories quickly become extremely hard to keep track of. With GNU GLOBAL a user can reference C, C++, Yacc and Java source-codes in order to be used as reference material by shell-commandline, less, nvi, elvis, EMACS or the web. Especially for large projects this makes keeping track of and getting into a project much easier.

GNU GLOBAL already possesses a pretty impressive amount of features - it can not just localize object-definitions but also references, searches the paths of libraries, understands POSIX regular expressions, has a compressed format to save disk space and much more. But of course not every problem is solved already.

The biggest current problem is the fact that the detection of data-type and macro definition does not yet work completely automatic. Enabling this is a primary goal for further development. Support for more languages and editors is also planned.

Since current CVS version of GNU automake already has a GTAGS target (the tag-file format used by GNU GLOBAL), it is to be expected that using GNU GLOBAL will be very easy in the near future.

The next project has a comparable goal but has a different focus.

HeaderBrowser

The HeaderBrowser by Amaury Bouchard is also a project for the better documentation of source-code. It has been developed with the HeaderViewer of NeXT in mind, because Amaury liked it a lot and wanted to reimplement an improved Unix-version.

Similar to GNU GLOBAL, the HeaderBrowser gets rid of the "flat" structure of the source-code and creates a navigationable documentation of the API of a program through processing the header files. Right now the HeaderBrowser supports C and C++ as programming languages and has output-backends to HTML, Texinfo and man-pages.

Plans for the future include adding alphabetical navigation to the functions; and a user-suggested idea of writing a tool that would analyse header-files and create empty HeaderBrowser comments that would only have to be filled by the developer. Since most developers have a rather tense relationship to documentation, this might increase the quality of many programs.

Licenses used for the HeaderBrowser are the GNU General Public License (GPL) and all documentation is released under the GNU Free Documentation License (FDL), so the program is truly Free in the best meaning of the word.

GNUTLS

GNUTLS is a pretty young project with the goal of implementing a transport layer security (TLS) library under the GNU General Public License. This library will grant access to the SSL 3.0 and TLS 1.0 layers and should enable programmers to easily add a security-layer to their programs.

Currently GNUTLS is still inferior to the OpenSSL library as it has not been as thoroughly tested and not really used in real life. But the layout of GNUTLS is thread-safe and a much easier interface. Besides that it is good to have a TLS implentation in the GNU Project that is clearly compatibly with the GNU General Public License.

At the moment GNUTLS is not ready for daily use. First of all an ASN.1 parser for x509 certificates is still missing, but this should change relatively soon as the current author, Nikos Mavroyanopoulos, is working on it at the moment. Since the other developer, Tarun Upadhyay had to leave the project because of timing-problems, help is very much welcome.

Further goals for development are implementation of the extensions specified in the "Wireless Extensions to TLS" internet draft and OpenPGP support is planned, which is something no other free TLS implementation has. For obvious reasons GNUTLS is based on libgcrypt, the library of the GNU Privacy Guard Project by Werner Koch, which has become rather well-known for its support by the German government.

Now I'm coming to a project that is of paramount importance to all of us.

GNU.FREE

The "Free Referenda & Elections Electronically" (FREE) Project, which has originally been started by the University of Warwick (U.S.A.), and is now being maintained by Jason Kitcat, has also been dubbed an official GNU Project recently. The project goal is to create a secure electronic voting system that will be secure and protect privacy.

The concept of the project is already two years old, so it is not a reaction triggered by the election fiasco in Florida as one could be inclined to suspect. But this obvious manipulation of a public election has brought the principles of democracy and the questions about voting and the counting of votes back into the public awareness.

There are a lot of different opinions about e-democracy and I am not sure myself that it is a good idea to establish an even more direct system, as it will most probably give demagogues more power than they already have. As far as I am concerned, the question whether "electronic voting" is desirable has not been answered satisfactory.

But experience shows that we have to expect a strong tendency towards e-democracy and I don't think we can afford to ignore this development just because we have doubts about its sense.

In a conventional system noone would seriously suggest letting a private company do the elections for the government. Imagine: the voter would go to an office of that company, enter his or her vote and in the evening the winner is announced by the company without any possibility of checking those results. Transferred to the digital domain with its much bigger possibilities of manipulation, the majority suddenly doesn't consider this a problem anymore.

I made this experience beginning of February, when I was at the Transmediale in Berlin, Germany where I participated in a panel on "Social Software." An example basically matching the one given above was happily welcomed.

Social software must be Free Software - only that way human rights can be maintained in an increasingly digital culture. So independent of the personal opinion on the sense of e-democracy, it is important that GNU.FREE does exist.

So let's get back to the technical issues.

The system has been designed in a way that lets it scale well. Jason has been thinking of democracies the size in India (the biggest democracy in the world) - with GNU.FREE elections from small board elections to the elections for the European Parliament can be held.

Of course the preservation of privacy and security issues had top priority during development and according to Jason, it does pretty well in these aspects already. If correctly configured, nothing short of a physical destruction of the servers by fire or other catastrophes could endanger an election - and this would also be a problem for more conventional election systems.

The project is relatively small for its rather big goals. Besides Jason there have been contributions to GNU.FREE by Rajagopal C.V., Thomas Muller, Ceki Gulcu, Neil Ferguson and Paul Voller. Help is needed and wanted especially for testing and finding possible security problems.

The improvement of security through additional layers of encryption and bug-fixes is a constant goal of GNU.FREE. Besides this it is planned to implement different election systems as currently only a direct comparison between the amount of votes for certain candidates is possible.

see you...

Okay, that's it for this month. As usual I would like to ask for lots of comments, ideas, questions and project descriptions under the standard address. ■

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
GNU-Darwin home page http://gnu-darwin.org
Apple Darwin home page http://www.publicsource.apple.com/projects/darwin/
Why the APSL is not a Free Software license http://www.gnu.org/philosophy/apsl.html
get_file FTP address ftp://ftp.cs.duke.edu/pub/des/scripts/get_file
GNU GLOBAL home page http://www.tamacom.com/global/
HeaderBrowser home page http://www.headerbrowser.org
GNUTLS home page http://gnutls.hellug.gr/
GNU.FREE home page http://www.thecouch.org/free/

■