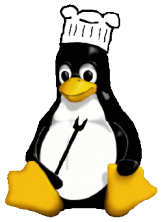
The Open Pitt

What's cooking in Linux and Open Source in Western Pennsylvania



Issue 11 April 2005 www.wplug.org

Stallman Returns to Pittsburgh

On April 7, the University of Pittsburgh chapter of the Association for Computing Machinery hosted a lecture by Richard M. Stallman. In addition to being the founder and president of the Free Software Foundation, he wrote key pieces of software infrastructure including the GCC compiler and the Emacs text editor.

Stallman first formalized the definition of Free Software and created the GNU General Public License, the license which covers the Linux kernel and thousands of other software projects. And it all grew out of an incident which occurred twenty years ago right here in Pittsburgh.

In a Jam

In the late '70s and early '80s, Stallman worked at the Massachusetts Institute of Technology's computer labs. The community of programmers was a small one indeed, and code was freely shared among them in the interest of getting the system to do useful work. Software wasn't much thought of as a product; people bought hardware and support services, not code.

Xerox had donated one of the first laser printer models to MIT, and while very useful, it was subject to frequent paper jams. A similar problem with an earlier printer had inspired Stallman to alter its code to alert users to paper jams so someone could walk over and clear it. Unlike before, he didn't have the source code for this printer to make the necessary changes.

Having heard that someone at Carnegie Mellon University had the software he wanted, he dropped into that individual's office on a trip to Pittsburgh. He left empty-handed; the person had signed a non-disclosure agreement with Xerox and could not reveal the code. Accustomed to a culture of sharing and cooperation, Stallman walked away from the encounter shocked and angry.

A GNU Beginning

In his view, this arrangement turned users into prisoners. Without source code, they were unable to improve the software or fix if it broke. He resolved to create a system which guaranteed the user's freedom. At the time, Unix was a popular and well-defined development environment, so he decided to follow the same design. Since everything would be built from scratch, he called the project GNU, which stands for "GNU's not Unix."

Leaving MIT to work on the GNU project, he defined the four essential conditions of Free Software:

0. The freedom to run the program, for any purpose.
1. The freedom to study how the program works and adapt it to your needs.
2. The freedom to redistribute copies so you can help your neighbor.
3. The freedom to improve the program and release your improvements to the public.

Stallman explained each of these freedoms. A program that doesn't respect freedom 0 may only run for a limited period of time, or have certain features disabled unless you pay the author for a secret key to unlock it. Without freedom 1, you can't even be sure of what your computer is doing, since you can't see the code that makes it run. Freedom 2 means you never have to choose between helping a friend by giving him a copy of a useful program and obeying the copyright restrictions an author has placed on it. And finally, freedom 3 allows you to share any improvements you've made to the program with everyone.

Anyone can exercise freedoms 0 and 2, but what about 1 and 3? If you're not a programmer, how do they help you? As Stallman pointed out, "This works like freedom of the press. Most people do not write articles for publication, they don't exercise freedom of the press—but everybody gets the benefit of freedom of the press, they get the benefit of seeing different viewpoints expressed."

See **STALLMAN**, *p. 2*

March Roundup

Mar. 19 General User Meeting: **Bill Moran** of **Potential Technologies** gave an overview of POSIX-like kernels, mainly BSD and Linux. He described how the kernel handles running multiple tasks, and the use of the *nice* command to assign priority levels to processes, as well as memory management and caching data from disk. By popular demand, he described the timeworn "forkbomb" attack and preventing it using resource limits.

Coming Events

Apr. 30: Special Presentation with PghSAGE, Topic: Flying Linux. 1PM to 5PM, 1507 Newell-Simon Hall, CMU

May 21: Installfest. 10AM to 5PM, 1507 Newell-Simon Hall, CMU

May 28: General User Meeting, Topic: Version Control with Subversion. 10AM to 2PM, 1507 Newell-Simon Hall, CMU

Jun. 4: General User Meeting, Topic: MythTV. 10AM to 2PM, 1507 Newell-Simon Hall, CMU

Jun. 11: Tutorial, Topic: Regular Expressions. 10AM to 2PM, 1507 Newell-Simon Hall, CMU

The public is welcome at all events

STALLMAN, *from p. 1*

Throughout the 1980's, work on the GNU system moved slowly. Many did, and still do, deride Stallman as a hopeless idealist, but his attitude is "When you're trying to do something big, there's nothing more practical than idealism." Free versions of most Unix utilities were created, but critical pieces were missing. One of these was a kernel, which talks directly to the computer hardware and manages the programs running on the system. An effort called the Hurd was under way, but progress was glacial. So at that point, the only people who could use the GNU tools were the few who already had access to an expensive Unix system to run them on.

Enter the Penguin

In 1991, Finnish student Linus Torvalds announced a project he was working on, a primitive kernel for the IBM PC. It rapidly improved, and the following year he made it Free Software. Before long, one could combine the Linux kernel with the GNU tools and have a fully-functional operating system on inexpensive personal computer hardware.

While pleased by this development, Stallman became concerned at the tendency of users to describe their computers as "Linux systems," which he felt neglected the GNU contributions. Even today, he pointedly refers to the "GNU/Linux" operating system and chastises anyone who leaves off the GNU prefix.

Free, or Open?

Another ideological fault line opened up several years later. Stallman was steadfast that freedom was the prime need: "The question is not how much Free Software are you using—it's how little non-Free Software are you using, and when are you going to get that amount down to zero?" Others were more pragmatic, and while they felt closed software was inferior to openly-developed code, they did not consider it odious. They also found the "Free" label off-putting to companies which might otherwise contribute to the growing pool of software. So the term "Open Source" was coined.

Despite what seem like great similarities between Free Software and Open Source, Stallman continues to inveigh against the latter phrase, which he feels devalues the user's freedom.

Education

Stallman concluded his talk with a discussion of the use of Free Software in education. He laid out the reasons why schools should use and teach with Free Software. First is to save money, although proprietary companies have taken to providing copies of their software at no charge to get students used to using their programs.

Second is their duty to, in his words, "teach students to live lives as citizens of a free society, not lives of serfdom." Third is that proprietary software forbids studying it and learning how it works, contrary to the mission of schools to educate. Last is his view that schools should not teach students the ethically and morally repugnant practice of using and creating non-Free software.

On the Lighter Side

After about an hour of this deep philosophical talk, Stallman lightened the mood by donning a halo and appearing as St. iGNUtius of the Church of Emacs. He encouraged all in attendance to exorcise their computers of proprietary software.

Q&A

An audience member asked about the threat of software patents to Free Software. Unsurprisingly, he strongly opposed software patents, and noted that developing a good and useful program requires combining hundreds or thousands of techniques and methods. Software patents sabotage this by locking up certain techniques for the exclusive use of one person.

In response to a question on source code control, Stallman took the opportunity to talk about BitKeeper, the proprietary revision control system used for the Linux kernel until just the day before. While happy that its use was coming to an end, he attacked the decision to use it in the first place, and said that the presence of a proprietary program in the center of such an important project caused serious damage to the cause of Free Software.

The next person asked Stallman to name the important trends of the next ten to twenty years. Joking that his crystal ball was cloudy, he instead listed the items he felt were important now: Free Software to create and view Flash files, a Free replacement for the Java platform, and an end to the practice of having to load non-Free firmware into a device to get it working.

In Conclusion

Whatever your view of Free Software and Open Source, this was an engaging and thought-provoking lecture. One hopes that it doesn't take another twenty years for Richard Stallman to visit the city again. A recording of the speech can be downloaded from <http://www.wplug.org/meetings/one-meeting?wp_meeting_id=3184>.

The Open Pitt is published by the Western Pennsylvania Linux Users Group
<http://www.wplug.org/top/>

Editors: Elwin Green
Vance Kochenderfer

What is Linux?

Linux is a *kernel*, the core of a computer operating system, created by Linus Torvalds. It is typically packaged as a *distribution*, which includes the extra programs necessary to make a computer functional and useful. Since 1991, it has grown from a one-man project which ran on one computer to one with thousands of contributors running on everything from personal organizers to million-dollar supercomputers.

What are Open Source and Free Software?

Open Source and Free Software provide you, the user, with the opportunity to see the source code of the programs you use. You are free to use it, share it with others, and even make changes to it if you wish. While the Free Software and Open Source communities differ in their philosophical approach, in practical terms they share nearly identical goals. Learn more at <http://www.opensource.org/> and <http://www.gnu.org/>.

This newsletter was produced using Open Source and Free Software.