Linux and the GNU system

GNU philosophy

The original version of this essay was published as the file `etc/LINUX-GNU` in the GNU Emacs distribution.

This document is part of GNU philosophy, the GNU Project's exhaustive collection of articles and essays about free software and related matters.

Copyright © 1996, 2002 Richard Stallman

Verbatim copying and distribution of this entire documentare permitted worldwide, without royalty, in any medium, provided this notice is preserved.

Linux and the GNU system

The GNU project started in 1984 with the goal of developing a complete free Unix-like operating system: GNU. “Free” refers to freedom, not price; it means you are free to run, copy, distribute, study, change, and improve the software.

A Unix-like system consists of many different programs. We found some components already available as free software—for example, X Windows and TeX. We obtained other components by helping to convince their developers to make them free—for example, the Berkeley network utilities. This left many missing components that we had to write in order to produce GNU—for example, GNU Emacs, the GNU C compiler, the GNU C library, Bash, and Ghostscript. The GNU system consists of all these components together.

The GNU project is not just about developing and distributing some useful free software. The heart of the GNU project is an idea: that software should be free, that software users should have freedom to participate in a community. To run your computer, you need an operating system; if it is not free, your freedom has been denied. To have freedom, you need a free operating system. We therefore set out to write one.

In the long run, though, we cannot expect to keep the free operating system free unless the users are aware of the freedom it gives them, and value that freedom. People who do not appreciate their freedom will not keep it long. If we want to make freedom last, we need to spread awareness of the freedoms they have in free software.

The GNU project’s method is that free software and the idea of users’ freedom support each other. We develop GNU software, and as people encounter GNU programs or the GNU system and start to use them, they also think about the GNU idea. The software shows that the idea can work in practice. Some of these people come to agree with the idea, and then they are more likely to write additional free software. Thus, the software embodies the idea, spreads the idea, and grows from the idea.

Early on in the development of GNU, various parts of it became popular even though users needed proprietary systems to run them on. Porting the system to many systems and maintaining them required a lot of work. After that work, most GNU software is easily configured for a variety of different platforms.

By 1991, we had found or written all of the essential major components of the system except the kernel, which we were writing.[1]

That was the situation when Linux came into being. Linux is a kernel, like the kernel of Unix; it was written by Linus Torvalds, who released it under the GNU General Public License. He did not write this kernel for GNU, but it fit into the gap in GNU. The combination of GNU and Linux included all the major essential components of a Unix-compatible operating system. Other people, with some work made the combination into a usable system. The principal use of Linux, the kernel, is as part of this combination.

The popularity of the GNU/Linux combination is success, in the sense of popularity, for GNU. Ironically, the popularity of GNU/Linux undermines our method of communicating the ideas of GNU to people who use GNU.

[1] This kernel consists of the Mach microkernel plus the GNU HURD. The first test release was made in 1996. Now, in 2002, it is running well, and Hurd-based GNU systems are starting to be used.

When GNU programs were only usable individually on top of another operating system, installing and using them meant knowing and appreciating these programs, and thus being aware of GNU, which led people to think about the philosophical base of GNU. Now users can install a unified operating system which is basically GNU, but they usually think these are "Linux systems". At first impression, a "Linux system" sounds like something completely distinct from the "GNU system," and that is what most users think.

This leads many users to identify themselves as a separate community of "Linux users", distinct from the GNU user community. They use more than just some GNU programs, they use almost all of the GNU system, but they don't think of themselves as GNU users. Often they never hear about the GNU idea; if they do, they may not think it relates to them.

Most introductions to the "Linux system" acknowledge that GNU software components play a role in it, but they don't say that the system as a whole is a modified version of the GNU system that the GNU project has been developing and compiling since Linus Torvalds was in junior high school. They don't say that the main reason this free operating exists is that the GNU Project worked persistently to achieve its goal of freedom.

As a result, most users don't know these things. They believe that the "Linux system" was developed by Linus Torvalds "just for fun", and that their freedom is a matter of good fortune rather than the dedicated pursuit of freedom. This creates a danger that they will leave the survival of free software to fortune as well.

Since human beings tend to correct their first impressions less than called for by additional information they learn later, these users will tend to continue to underestimate their connection to GNU even if they do learn the facts.

When we began trying to support the GNU/Linux system, we found this widespread misinformation led to a practical problem—it hampered cooperation on software maintenance. Normally when users change a GNU program to make it work better on a particular system, they send the change to the maintainer of that program; then they work with the maintainer, explaining the change, arguing for it, and sometimes rewriting it for the sake of the overall coherence and maintainability of the package, to get the patch installed. But people who thought of themselves as "Linux users" showed a tendency to release a forked "Linux-only" version of the GNU program and consider the job done. In some cases we had to redo their work in order to make GNU programs run as released in GNU/Linux systems.

How should the GNU project encourage its users to cooperate? How should we spread the idea that freedom for computer users is important?

We must continue to talk about the freedom to share and change software—and to teach other users to value these freedoms. If we value having a free operating system, it makes sense to think about preserving those freedoms for the long term. If we value having a variety of free software, it makes sense to think about encouraging others to write free software, instead of proprietary software.

However, it is not enough just to talk about freedom; we must also make sure people know the reasons it is worth listening to what we say.

Long explanations such as our philosophical articles are one way of informing the public, but you may not want to spend so much time on the matter. The most effective way you can help with a small amount of work is simply by using the terms "Linux-based GNU system" or “GNU/Linux system”, instead of “Linux system,” when you write about or mention such a system. Seeing these terms will show many people the reason to pay attention to our philosophical articles.

The system as a whole is more GNU than Linux; the name “GNU/Linux” is fair. When you are choosing the name of a distribution or a user group, a name with “GNU/Linux” will reflect both roots of the combined system, and will bring users into connection with both—including the spirit of freedom and community that is the basis and purpose of GNU.