What's in a Name?

GNU philosophy

To learn more about this issue, you can read our GNU/Linux FAQ, at http://gnu.org/gnu/gnu-linux-faq.html, our page on Linux and the GNU Project, at http://gnu.org/gnu/linux-and-gnu.html, which gives a history of the GNU/Linux system as it relates to this issue of naming, and the article "GNU Users Who Have Never Heard of GNU," at http://gnu.org/gnu/gnu-users-never-heard-of-gnu.html.

This essay was originally published on http://gnu.org, in 2000.

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

What's in a Name?

Names convey meanings; our choice of names determines the meaning of what we say. An inappropriate name gives people the wrong idea. A rose by any other name would smell as sweet—but if you call it a pen, people will be rather disappointed when they try to write with it. And if you call pens "roses," people may not realize what they are good for. If you call our operating system Linux, that conveys a mistaken idea of the system's origin, history, and purpose. If you call it GNU/Linux, that conveys (though not in detail) an accurate idea.

Does this really matter for our community? Is it important whether people know the system's origin, history, and purpose? Yes—because people who forget history are often condemned to repeat it. The Free World that has developed around GNU/Linux is not guaranteed to survive; the problems that led us to develop GNU are not completely eradicated, and they threaten to come back.

When I explain why it's appropriate to call the operating system GNU/Linux rather than Linux, people sometimes respond this way:

> Granted that the GNU Project deserves credit for this work, is it really worth a fuss when people don't give credit? Isn't the important thing that the job was done, not who did it? You ought to relax, take pride in the job well done, and not worry about the credit.

This would be wise advice, if only the situation were like that—if the job were done and it were time to relax. If only that were true! But challenges abound, and this is no time to take the future for granted. Our community's strength rests on commitment to freedom and cooperation. Using the name GNU/Linux is a way for people to remind themselves and inform others of these goals.

It is possible to write good free software without thinking of GNU; much good work has been done in the name of Linux also. But the term "Linux" has been associated ever since it was first coined with a philosophy that does not make a commitment to the freedom to cooperate. As the name is increasingly used by business, we will have even more trouble making it connect with community spirit.

A great challenge to the future of free software comes from the tendency of the "Linux" distribution companies to add nonfree software to GNU/Linux in the name of convenience and power. All the major commercial distribution developers do this; none limits itself to free software. Most of them do not clearly identify the nonfree packages in their distributions. Many even develop nonfree software and add it to the system. Some outrageously advertise "Linux" systems that are "licensed per seat," which give the user as much freedom as Microsoft Windows.

People try to justify adding nonfree software in the name of the "popularity of Linux"—in effect, valuing popularity above freedom. Sometimes this is openly admitted. For instance, *Wired* magazine said that Robert McMillan, editor of *Linux Magazine*, "feels that the move toward open source software should be fueled by technical, rather than political, decisions."[1] And Caldera's CEO openly urged users to drop the goal of freedom and work instead for the "popularity of Linux."

[1] Michelle Finley, "French Pols Say, 'Open It Up,'" 24 April 2000, http://wired.com/politics/law/news/2000/04/35862.

Adding nonfree software to the GNU/Linux system may increase the popularity, if by popularity we mean the number of people using some of GNU/Linux in combination with nonfree software. But at the same time, it implicitly encourages the community to accept nonfree software as a good thing, and forget the goal of freedom. It is not good to drive faster if you can't stay on the road.

When the nonfree "add-on" is a library or programming tool, it can become a trap for free software developers. When they write free software that depends on the nonfree package, their software cannot be part of a completely free system. Motif and Qt trapped large amounts of free software in this way in the past, creating problems whose solutions took years. Motif remained somewhat of a problem until it became obsolete and was no longer used. Later, Sun's nonfree Java implementation had a similar effect: the Java Trap, fortunately now mostly corrected.

If our community keeps moving in this direction, it could redirect the future of GNU/Linux into a mosaic of free and nonfree components. Five years from now, we will surely still have plenty of free software; but if we are not careful, it will hardly be usable without the nonfree software that users expect to find with it. If this happens, our campaign for freedom will have failed.

If releasing free alternatives were simply a matter of programming, solving future problems might become easier as our community's development resources increase. But we face obstacles that threaten to make this harder: laws that prohibit free software. As software patents mount up, and as laws like the Digital Millennium Copyright Act are used to prohibit the development of free software for important jobs such as viewing a DVD or listening to a RealAudio stream, we will find ourselves with no clear way to fight the patented and secret data formats except to *reject the nonfree programs that use them.*

Meeting these challenges will require many different kinds of effort. But what we need above all, to confront any kind of challenge, is to remember the goal of freedom to cooperate. We can't expect a mere desire for powerful, reliable software to motivate people to make great efforts. We need the kind of determination that people have when they fight for their freedom and their community—determination to keep on for years and not give up.

In our community, this goal and this determination emanate mainly from the GNU Project. We're the ones who talk about freedom and community as something to stand firm for; the organizations that speak of "Linux" normally don't say this. The magazines about "Linux" are typically full of ads for nonfree software; the companies that package "Linux" add nonfree software to the system; other companies "support Linux" by developing nonfree applications to run on GNU/Linux; the user groups for "Linux" typically invite salesmen to present those applications. The main place people in our community are likely to come across the idea of freedom and determination is in the GNU Project.

But when people come across it, will they feel it relates to them?

People who know they are using a system that came out of the GNU Project can see a direct relationship between themselves and GNU. They won't automatically agree with our philosophy, but at least they will see a reason to think seriously about it. In contrast, people who consider themselves "Linux users," and believe that the GNU Project "developed tools which proved to be useful in Linux," typically perceive only an indirect relationship between GNU and themselves. They may just ignore the GNU philosophy when they come across it.

The GNU Project is idealistic, and anyone encouraging idealism today faces a great obstacle: the prevailing ideology encourages people to dismiss idealism as “impractical.” Our idealism has been extremely practical: it is the reason we have a free GNU/Linux operating system. People who love this system ought to know that it is our idealism made real.

If “the job” really were done, if there were nothing at stake except credit, perhaps it would be wiser to let the matter drop. But we are not in that position. To inspire people to do the work that needs to be done, we need to be recognized for what we have already done. Please help us, by calling the operating system GNU/Linux.