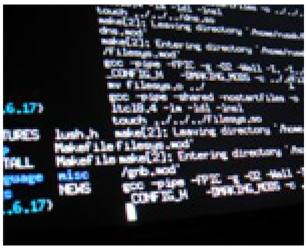
Linux, GNU, and freedom

Just consider: the GNU Project starts developing an operating system, and years later Linus Torvalds adds one important piece. The GNU Project says, "Please give our project equal mention", but Linus says, "Don't give them a share of the credit; call the whole thing after my name alone!''



Since Joe Barr's article criticized my dealings with SIGLINUX, I would like to set the record straight about what actually occurred, and state my reasons.

When SIGLINUX invited me to speak, it was a "Linux User Group''; that is, a group for users of the GNU/Linux system which calls the whole system "Linux''. So I replied politely that if they'd like someone from the GNU Project to give a speech for them, they ought to treat the GNU Project right, and call the system "GNU/Linux''. The system is a variant of GNU, and the GNU Project is its principal developer, so social convention says to call it by the name we chose. Unless there are powerful reasons for an exception, I usually decline to give speeches for organizations that won't give GNU proper credit in this way. I respect their freedom of speech, but I also have the freedom not to give a speech.

Subsequently, Jeff Strunk of SIGLINUX tried to change the group's policy, and asked the FSF to list his group in our page of GNU/Linux user groups. Our webmaster told him that we would not list it under the name "SIGLINUX'' because that name implies that the group is about Linux. Strunk proposed to change the name to "SIGFREE'', and our webmaster agreed that would be fine. (Barr's article said we rejected this proposal.) However, the group ultimately decided to stay with "SIGLINUX''.

At that point, the matter came to my attention again, and I suggested they consider other possible names. There are many names they could choose that would not call the system "Linux'', and I hope they will come up with one they like. There the matter rests as far as I know.

Is it true, as Barr writes, that some people see these actions as an "application of force'' comparable with Microsoft's monopoly power? Probably so. Declining an invitation is not coercion, but people who are determined to believe that the entire system is "Linux'' sometimes develop amazingly distorted vision. To make that name appear justified, they must see molehills as mountains and mountains as molehills. If you can ignore the facts and believe that Linus Torvalds developed the whole system starting in 1991, or if you can ignore your ordinary principles of fairness and believe that Torvalds should get the sole credit even though he didn't do that, it's a small step to believe that I owe you a speech when you ask.

Just consider: the GNU Project starts developing an operating system, and years later Linus Torvalds adds one important piece. The GNU Project says, "**Please give our project equal mention,''** but Linus says, "**Don't give them a share of the credit; call the whole thing after my name alone!''** Now envision the mindset of a person who can look at these events and accuse the GNU Project of egotism. It takes strong prejudice to misjudge so drastically.

A person who is that prejudiced can say all sorts of unfair things about the GNU Project and think them justified; his fellows will support him, because they want each other's support in maintaining their prejudice. Dissenters can be reviled; thus, if I decline to participate in an activity under the rubric of "Linux'', they may find that inexcusable, and hold me responsible for the ill will they feel afterwards. When so many people want me to call the system "Linux'', how can I, who merely launched its development, not comply? And forcibly denying them a speech is forcibly making them unhappy. That's coercion, as bad as Microsoft!

Now, you might wonder why I don't just duck the issue and avoid all this grief. When SIGLINUX invited me to speak, I could simply have said "No, sorry'' and the matter would have ended there. Why didn't I do that? I'm willing to take the risk of being abused personally in order to have a chance of correcting the error that undercuts the GNU

Project's efforts.

Calling this variant of the GNU system "Linux'' plays into the hands of people who choose their software based only on technical advantage, not caring whether it respects their freedom. There are people like Barr, that want their software "free from ideology'' and criticize anyone that says freedom matters. There are people like Torvalds that will pressure our community into use of a non-free program, and challenge anyone who complains to provide a (technically) better program immediately or shut up. There are people who say that technical decisions should not be "politicized'' by consideration of their social consequences.

In the 70s, computer users lost the freedoms to redistribute and change software because they didn't value their freedom. Computer users regained these freedoms in the 80s and 90s because a group of idealists, the GNU Project, believed that freedom is what makes a program better, and were willing to work for what we believed in.

We have partial freedom today, but our freedom is not secure. It is threatened by the CBDTPA (formerly SSSCA), by the Broadcast "Protection'' Discussion Group (see http://www.eff.org/) which proposes to prohibit free software to access digital TV broadcasts, by software patents (Europe is now considering whether to have software patents), by Microsoft nondisclosure agreements for vital protocols, and by everyone who tempts us with a non-free program that is "better'' (technically) than available free programs. We can lose our freedom again just as we lost it the first time, if we don't care enough to protect it.

Will enough of us care? That depends on many things; among them, how much influence the GNU Project has, and how much influence Linus Torvalds has. The GNU Project says, "Value your freedom!''. Joe Barr says, "Choose between non-free and free programs on technical grounds alone!''. If people credit Torvalds as the main developer of the GNU/Linux system, that's not just inaccurate, it also makes his message more influential--and that message says, "Non-free software is ok; I use it and develop it myself.'' If they recognize our role, they will listen to us more, and the message we will give them is, "This system exists because of people who care about freedom. Join us, value your freedom, and together we can preserve it.'' See http://www.gnu.org/gnu/the-gnu-project.html for the history.

When I ask people to call the system GNU/Linux, some of them respond with silly excuses and straw men. But we probably haven't lost anything, because they were probably unfriendly to begin with. Meanwhile, other people recognize the reasons I give, and use that name. By doing so, they help make other people aware of why the GNU/Linux system really exists, and that increases our ability to spread the idea that freedom is an important value.

This is why I keep butting my head against bias, calumny, and grief. They hurt my feelings, but when successful, this effort helps the GNU Project campaign for freedom.

Since this came up in the context of Linux (the kernel) and Bitkeeper, the non-free version control system that Linus Torvalds now uses, I'd like to address that issue as well.

Bitkeeper issue

The use of Bitkeeper for the Linux sources has a grave effect on the free software community, because anyone who wants to closely track patches to Linux can only do it by installing that non-free program. There must be dozens or even hundreds of kernel hackers who have done this. Most of them are gradually convincing themselves that it is ok to use non-free software, in order to avoid a sense of cognitive dissonance about the presence of Bitkeeper on their machines. What can be done about this?

One solution is to set up another repository for the Linux sources, using CVS or another free version control system, and arranging to load new versions into it automatically. This could use Bitkeeper to access the latest revisions, then install the new revisions into CVS. That update process could run automatically and frequently.

The FSF cannot do this, because we cannot install Bitkeeper on our machines. We have no non-free systems or applications on them now, and our principles say we must keep it that way. Operating this repository would have to be done by someone else who is willing to have Bitkeeper on his machine, unless someone can find or make a way to do it using free software.

The Linux sources themselves have an even more serious problem with non-free software: they actually contain some. Quite a few device drivers contain series of numbers that represent firmware programs to be installed in the device. These programs are not free software. A few numbers to be deposited into device registers are one thing; a substantial program in binary is another.

The presence of these binary-only programs in "source'' files of Linux creates a secondary problem: it calls into question whether Linux binaries can legally be redistributed at all. The GPL requires "complete corresponding source code,'' and a sequence of integers is not the source code. By the same token, adding such a binary to the Linux sources violates the GPL.

The Linux developers have a plan to move these firmware programs into separate files; it will take a few years to mature, but when completed it will solve the secondary problem; we could make a "free Linux'' version that doesn't have the non-free firmware files. That by itself won't do much good if most people use the non-free "official'' version of Linux. That may well occur, because on many platforms the free version won't run without the non-free firmware. The "free Linux'' project will have to figure out what the firmware does and write source code for it, perhaps in assembler language for whatever embedded processor it runs on. It's a daunting job. It would be less daunting if we had done it little by little over the years, rather than letting it mount up. In recruiting people to do this job, we will have to overcome the idea, spread by some Linux developers, that the job is not necessary.

Linux, the kernel, is often thought of as the flagship of free software, yet its current version is partially non-free. How did this happen? This problem, like the decision to use Bitkeeper, reflects the attitude of the original developer of Linux, a person who thinks that ``technically better'' is more important than freedom.

Value your freedom, or you will lose it, teaches history. "Don't bother us with politics,'' respond those who don't want to learn.

Update: BitKeeper is no longer used to manage the Linux kernel source tree. RMS discusses the change in his article, Thank You, Larry McVoy.

Author: Free Software Foundation