

Lectio doctoralis, Richard Stallman

This is the "lectio doctoralis" that Richard Stallman held when receiving the honorary degree in Engineering ("Laurea ad honorem in Ingegneria Informatica") from the University of Pavia (Italy), on September 24th 2007.

Innovation can create riches and once in a while, those riches can lead to general economic prosperity, especially if you don't have neo-liberal economics to impede the result.

But innovation affects things much more important than riches or even economic prosperity. Democracy was an innovation, fascism was an innovation. Today, in Italy, we see the innovation of placing criminal charges against fishermen for saving people from drowning in the sea. Innovations can directly affect our freedom, which is more important than anything else. Innovation can affect social solidarity, for good or for ill.

So when we consider technical progress in computers or in software, the most important question to ask is: "How does this affect our freedom? How does this affect our social solidarity?". Technically speaking it's progress, but is it really progress in social and ethical terms, or is it the opposite?

During my career in programming, as computers developed from something used by a few specialists and enthusiasts into something that most people used, there has been tremendous technical progress and it was accompanied by ghastly social and ethical regression. In fact, nearly everyone who uses computers began using them under a social system that can only be described as dictatorship.

The developer of the program controls what it does. If you use it, the developer controls what you can do, and what you can't do. And controls what it does to you. So that the software that you think is yours, is not really there to serve you. It is there to control you. Companies such as Microsoft and Apple designed the software specifically to restrict you.

Windows Vista is primarily an advance in how to restrict the user. Which is why we have the `badvista.org` campaign. And when this is over, outside the building I will offer you stickers from that campaign, if you wish to help teach people why they shouldn't downgrade to Vista.

Apple designs software specifically to restrict the users. It's known as "Digital Restriction Management", or DRM. We have helped protests against Apple just as we helped protests against Microsoft. See the site `defectivebydesign.org` for more information and for how to participate.

Google designs software specifically to restrict the user. That's the nature of the Google Earth client: it is made the way it is specifically to restrict the people who use it. Obviously, it's not Free Software, because Free Software develops under the democratic control of its users. With the four freedoms: the freedom to run the program as you wish, to study the source code and change it so the program does what you wish, the freedom to distribute exact copies to others (which is the freedom to help your neighbor) and the freedom to distribute copies of your modified version (which is the freedom to contribute to your community). With these four freedoms the users, individually and collectively, are in charge.

And therefore Free Software cannot be designed to restrict the users. To design software to restrict the user is only possible when there is a dictator, when someone has power to control what the program will do and what it won't do. When the users have the control, when they can control their own computing, then nobody has the kind of power that would enable him to impose malicious features to restrict users or spy on users or attack users. If you use MacOS or Windows Vista, you are completely

at the mercy of that system's developer. Those developers have the power to forcibly change your software in any way they like, whenever the machine is connected to the network. The user no longer has even the chance to say yes or no. The system is one big back-door.

But with Free Software, *you* are in charge of what the computer will do. So it will serve you, instead of subjugating you. The question of Free Software is therefore not a technical question, it's an ethical, social and political question. It's a question of the human rights that the users of software ought to have.

Proprietary software developers say "no rights, we are in control, we should be in control, we demand total power over what your computer does; we will implement certain features and let you use them, but meanwhile we may spy on you as you use them and we can take them away at any time". But Free Software developers respect your freedom, and this is the ethical obligation of every software developer: to respect the freedom of the users of that software. Making proprietary user-subjugating software sometimes is profitable, but it is never ethical, and it should never happen.

But it will be up to you to make that be true. I alone can say these things, but I alone cannot make them reality. We must all work together to establish freedom and democracy for the users of software. And this freedom and democracy is now essential to enjoy freedom and democracy in other aspects of life. Right now, some of the biggest Internet service providers in the United States are carrying out political censorship of email. A major organization called "truth out", whose website you may have seen, truthout.org, is being blocked from sending mail to its subscribers by Yahoo and Hotmail and WebTV. And it has done this for more than a week, despite the complaints from many of the users of those companies. Apparently they think they have gone beyond the point where they have to care what anyone says about them.

All the forms of freedom that we hold dear are transformed when we carry out the relevant activities through computers. We must *re*-found these freedoms in such a way that we can depend on them while we use digital technology. An essential part of this re-foundation is insisting that the software we use be under our control.

Not everyone wants to be a programmer, not everyone will learn personally how to study the source code and change it. But, in a world where your software is free you can, if you feel it necessary, hire someone else to change it for you. You can persuade your cousin programmer to change it for you if you say it's really important. You can join together with other users and pull your funds to hire a programmer. And the simple fact that there are millions of programmers who can study and change the software will mean that if the software is malicious, almost certainly somebody else, who has the requisite skills, will find that and correct it, and you will get the corrected version without any special effort of your own. So we all benefit, programmers and non-programmers alike, from the freedoms that free software grants to us. The freedom to cooperate and the freedom to control our own lives personally. They go together because both of them are the opposite of being under the power of the dictatorial software developer that can unilaterally make decisions that nobody else can change.

Free Software has an especial connection with Universities, and indeed all schools of all levels. Because Free Software supports education; proprietary software forbids education. There is no compatibility between education and proprietary software, not at the ethical level.

The source code and the methods of Free Software are part of human knowledge. The mission of every school is to disseminate human knowledge. Proprietary software is not part of human knowledge. It's secret, restricted knowledge, which schools are not allowed to disseminate. Schools that recognize this exclude proprietary software from their grounds. And this is what every schools should do. Not only to save money, which is an obvious advantage that will appeal immediately to many school administrators, but for ethical reasons as well. For instance, why do many proprietary software

developers offer discounts, or even gratis copies of their non-free software to schools and students.

I'm told that Microsoft offered a discount on those who wish to accept the shiny new chains of Windows Vista to the employees of this University. Why would they do such a thing? Is it because they wish to contribute to education? Obviously not. Rather, Microsoft and other similar companies wish to convert the University into an instrument for imposing the dependency on the user-subjugating software on society as a whole. They figured that if they get their software into schools, then students will learn to use it, and become dependent on it. They will develop a dependency. And thus after they graduate you can be sure that Microsoft and these other companies would no longer offer them discounted copies. And especially, the companies that this former students go to work for, will not be offered discounted copies. So, the software developers push on the schools, then push on the rest of society and push it deep into a pit. This is not something schools should do. This is the opposite of the mission of the school, which is to build a strong, capable, independent and free society. Schools should teach their students to be citizens of a s strong, capable, independent and free society. And this means teaching them to use Free Software, not proprietary software. So none of the classes in this University should teach proprietary software.

For those who will be great programmers, there is another reason why their schools must teach and use Free Software. Because when they get to the age of 13 or so, they are fascinated with software and they want to learn everything about how their computer and their system are functioning. So they will ask the teacher "how does this work?", and if this is proprietary software, the teacher has to say "I'm sorry, it's a secret, you can't find out". So there is no room for education. But if it's Free Software the teacher can explain the basics of it and then say "here is the source code, read this and you'll understand everything". And those budding brilliant programmers will read the whole source code because they are fascinated, and this way they will learn something very important: how to write software well. They don't need to be taught how to program, because for them programming is obvious, but writing good code is a different story. You have to learn that by reading lots of code and writing lots of code. Only Free Software provides this opportunity.

But there is a particular reason, for the sake of education in good citizenship. You see, schools must teach not just facts, not just skills but above all the spirit of good will, the habit of helping your neighbor. So every class, at every level should have this rule: "students, if you bring software to class, you may not keep it for yourself, you must share copies with the rest of the class".

However, the school has to practice its own rule: it has to set a good example. So every school should bring only Free Software to class, and set an example with its software of the practice of disseminating human knowledge while building a strong, capable, independent and free society. And encouraging the spirit of good will, of helping other people. Every school must migrate to Free Software, and I call on you, those of you who are faculty, or staff, or students of this University, to work together to bring about the migration of this University to Free Software, completely to Free Software, within a few years. It *can* be done in a few years: it requires taking a substantial step each year. Other Universities are doing this or have done it: you can do it too. You only have to reject social inertia as a valid reason for going deeper and deeper into the pit.

For those of you who are interested, after we leave this hall and this ceremony outside I will have various things from the Free Software Foundation that you might be interested in. And you can support the Free Software Foundation by going to fsf.org and becoming an associate member. For more information about the Free Software movement and the GNU operating system, and for where to find the entirely free distributions of the GNU/Linux operating system please look at gnu.org.

Thank you.

Transcribed from tape by Alessandro Rubini
Reviewed by Dora Scillipoti and Luca Andreucci

Last modified: Sun Oct 7 19:17:43 CEST 2007