

Richard Stallman Interview – Steve Hand, Interviewer

SH: In what cases is the use of free software most useful for expanding the adoption of approaches, technologies, or standards?

RS: I don't know., I've never thought of it that way because you see, Free Software is a matter of human rights and by contrast, expanding the adoption of some technical practice seems like a side issue to me. Free Software is software that respects the user's freedom, which means the user has four essential freedoms. Freedom zero is the freedom to run the program as you wish. Freedom one is the freedom to study the source code and change it to make it do what you wish. Freedom two is the freedom to redistribute exact copies and Freedom three is the freedom to distribute modified versions. So these freedoms are necessary so that the users have control over the software they use, have control over their computing, and also so that users can be free to be good members of a community. Now, it can be useful to promote progress and promote other good technical practices; but we shouldn't put the cart before the horse. No use of technology is good if it doesn't respect our freedom.

SH: OK, let's say we have some software that does all that, respects our freedom in the ways you've outlined. As an industry association, we may also want to produce some free software for the purposes of illustrating how to implement a standard.

RH: Not just for the purpose of illustrating it, but some people can actually use the standard. People who value freedom, they may want to use that standard but the only way they can use it and keep their freedom at the same time is if they use a free software implementation of it. So it's very important to have a free software implementation of it and if the only implementations available are proprietary and somebody in the free software movement is going to have to write a free implementation of it to replace them. Well, it's obviously all much more efficient if you write a free implementation and nobody else has to re-do the work.

SH: On a different topic, there are certainly a lot of GNU tools that are of interest to many people who will be showing up at the event. Are there any ones that are new and upcoming that people ought to be taking a look at?

RS: In the level specifically of tools, I don't know if there is anything new that we have. Keep in mind that tools are just one aspect of what GNU is all about because GNU is an operating system, a Unix-like operating system, which we didn't totally complete it, we never got our kernel to work well, so people use Linux as the kernel and so obviously such a system needs tools. I wrote a number of useful tools early on. But now our tools in GNU and in general in the free software world, are pretty goods so that's not the main area that we need to make progress in. One important GNU package that is being worked on a lot now is GNASH which is a flash player, well that's not a tool but its something that lots and lots of users want. But there is

one area where a lot of progress is being made and that is now that SUN has decided to liberate its Java platform and has done so for large parts of it, we are now working on filling in the gaps by combining the GNU implementations of SUN Java libraries with the Sun implementations of other libraries trying to make a complete free platform for Java that will do everything.

SH: Down this path – what contribution in use licenses should developers be considering for their free software efforts?

RS: When you say use licenses, do you have a specific meaning or are you talking about free software licenses?

SH: This would be inbound you know that the contributor of the code...

RS: Inbound, I don't understand what that means.

SH: So, a developer wants to submit some code, they have many of these projects that have some agreements that the submitter ...

RS: Now, those are not licenses, those are copyright assignments typically.

SH: Right, copyrights and the outgoing would be somebody who wanted to run their code or compile it.

RS: Now, I'm not sure if I want to adopt those terms, but at least I now know what you are asking about. So, if you're talking about how to contribute changes to a free software package or are you talking about how a developer should arrange for contributors to contribute changes, is that the question?

SH: No, it's really the agreement itself both for the contribution and for the use of the software that developers should consider if they are starting a new free software project.

RS: Well, I would say they should talk with lawyers to design this of course, because there are various different goals that one might have. What we do in the GNU project is first of all, we decided to have such an agreement for a couple of reasons, for FSF copyrighted packages, the main reason is that we wanted to make it clear that the code really was released for free use. We didn't want to have a situation where somebody gave us some code and then 5 years later, his ex-employer says we never permitted the release of that code. And another thing is we want to keep the copyright status of the program simple, because that facilitates enforcement in court. If somebody violates the GNU GPL or some other free software license and the copyright holders want to go to court, it's easier for them if the copyright status of that program is as simple as possible. So, these are why we decided to get copyright assignments plus disclaimers from peoples employers and in writing them we had two other goals. One was we wanted to make sure that the

software would be free, we wanted to reassure the authors that we were not tricking them, that we couldn't trick them. Of course, they could rely on our honesty and our commitment to our cause, but I wanted to give them something else that could show that we weren't just asking them to put all their trust in us, so, in our assignments, it commits us to using licenses that permit everyone to re-distribute under the same terms, so, once people donate their code to us, we are not allowed to make it proprietary, we just can't, we're contractually bound to them. The other thing is that I thought as a matter of respect for them that I shouldn't ask them to give up all of their rights to use the code that they wrote. I felt that for them to allow us to use it was enough of a gift and that we shouldn't gratuitously ask them to do even more so, our assignments say that they can have unlimited, non-exclusive rights to use their contributions. So they are not actually giving anything up in terms of what they could do with their code by virtue of doing as we ask and signing the assignment.

SH: So another question along these lines, many storage vendors embed GNU/Linux into their storage products but do not release the source for their modifications.

RS: Well, if they do that, then they are violating the licenses and they better start complying. They have no right according to the license, to distribute binaries and not make the source code available.

SH: Could you give us a quick review of the case law around the GPL?

RS: I'm not a lawyer and there is not a lot of case law but there is a little.

SH: Is it recent?

RS: I don't know, I mean I'm not a lawyer and in fact I don't remember how long ago there was a case involving MYSQL which is the only one, there was that and there was another strange case in the US where somebody tried to sue us claiming that we were suppressing competition and the court decided that we were doing just the opposite. I'm not a lawyer so...

SH: I understand that. Now for a small storage software vendor, that does not have the resources to base all their revenue on service and support, why should they release their software under a free software license?

RS: Well I guess I'm not sure if a storage vendor means somebody who is selling hardware right?

SH: Storage software.

RS: I don't understand.

SH: You have vendors that sell software that is exclusively for storage and that would include file systems or ...

RS: Well if they don't respect the users' freedom, it's not ethical at all and they really shouldn't do it. Developing software can be a contribution to society well, I should say more precisely, developing software and making it available to people to use, can be a contribution to society but that's not automatic, if people have to give up their freedom in order to use that software then it's not a contribution, then basically it's a trap to separate people from their freedom and I think it would be better if it weren't done at all. I value my freedom and that's why I won't use any of that software. I hope nobody else will use it either. I hope to teach others to value their freedom just as I do, because if people don't value their freedom, they tend to lose it. If you look at the US today, you can see where that leads.

SH: So, let me ask a question. Let's say, so you're definitely a proponent for all software should be free, what about hardware?

RS: Well, the question doesn't even make sense, remember the definition of free software that I told you. We are not talking about price here, we are talking about respect for certain freedoms. Think of free speech, not free beer. Now, these freedoms don't entirely make any sense for hardware, consider, well first of all freedom zero, the freedom to run the program. Normally, for hardware, if you buy it, you've got the freedom to use it, so freedom we almost can take for granted and what about freedom one, the freedom to modify the hardware, in this case it would turn in to if we try to translate the definition of free software and apply it to hardware, it would be the freedom to modify the hardware. Well most of the time you have that freedom. If you buy a computer in most cases you are free to change it, there are a few exceptions which I think are sort of nasty. But of course the extent to which you can modify hardware is rather limited. Nobody can modify a chip.

SH: Let me go back to the original question.

RS: I want to finish answering your question, because it's an important question so those two freedoms you mostly have, but what about freedom two, the freedom to copy and redistribute the software? Well that becomes the freedom to copy and redistribute the hardware. But there are no copiers for hardware objects, so it's a meaningless question. The only way to have more hardware is to build it. You can't just put it into your copier and have another one come out. So that's a meaningless question. It means a factory. Users can't copy the hardware.

SH: Well they could copy the design right?

RS: Well, you can copy the design but you can't run a design. See the interesting thing about software is that the design is source code and there is a compiler. You with your computer, can turn the design into something that will actually run. And

that's why these freedoms become important, because they are freedoms to do things that you can really do, and I can really do and every computer user can really do. But with hardware, to turn a design into a physical object requires a factory. If it is a chip it costs millions of dollars just to make one and so, these are not freedoms that are relevant to ordinary people. That's why I don't think these two questions are analogous. You can ask the same questions but the issues are not similar.

SH: So, I think I misunderstood you. You said free speech as opposed to free beer?

RS: Right, to understand which meaning of the word free it is. The word free is ambiguous. In most languages, there is one word that means zero price and another word which means it has to do with freedom. But in English, we do have a word that only means zero price, its gratis. But we don't have a word in everyday use that only means free as in freedom, so the best I can do is say free and explain to you which meaning it is.

SH: I'm not clear on which one it is.

RS: It's the meaning that has to do with freedom. It respects the four fundamental freedoms. It respects the users' freedom, the price is irrelevant. That's a different issue which I'm not concerned with.

SH: So, as long as the requirements that you stated previously are met, a developer of the software can sell it?

RS: Well, "sell it," I'm afraid that that wording also brings in the same confusion. To sell the software, that's ambiguous. To sell copies, that's fine. In fact, part of the freedom that you've got with free software is you can copy it and sell copies, and lots of people do. However, when people think of selling software what they usually mean is selling it and restricting the user, denying the user freedom. They don't say that, they treat it as implicit, but it's that act of denying the user's freedom that the free software movement is against. If you want to sell a copy of a program, that's fine. So if you get a copy of say GCC or GNU Emacs or GNASH or the GIMP or Linux cause its also free software, you're free to make copies and sell those copies or give them away or do one of them on Tuesdays and the other on Wednesdays, whatever you like, but if you restrict what the users can do, such that they don't have these four freedoms, then it is not free software.

SH: For the sake of brevity, let's call this and I know it's not accurate and there are certainly a lot of wrinkles in to how one would actually accomplish ...

RS: We can call it freedom respecting software if we want to use a slightly longer term.

SH: OK so, if somebody is able to get this freedom respecting software, without paying for it...

RS: Which you may. Anyone who has got a copy can give you a copy if he wants.

SH: *Now that individual turns around, modifies it, checks in the modification and then sells it to someone else...*

RS: Again, “sells it to someone else” is ambiguous as I explained before. He can be perfectly free to sell a copy, once he’s modified his version, he now has a modified version. Well, he doesn’t have to release that – he’s free to do so if he wants, but it’s not an obligation, it’s just a freedom so he might just keep it for himself and run it. But on the other hand, he might make it available to the public so, if he is going to be consistent with the ideals of free software, then he could either give away copies or he can sell copies but he has got to respect the freedom of others. And, copy-left licenses like the GNU general public license, require that he respect the freedom of others if he chooses to distribute them. Because, as long as he uses it privately, he is not required to ever distribute it. Of course, if you put it inside of a storage product like a disk control or whatever like that, then you are distributing copies. So, if it is GPL covered, you are going to have to follow the requirements of the GPL – but the GPL doesn’t require you to make your software available to the public at all and no free software license requires that. You always have the right to make changes and use your version privately.

SH: *Now if “use that version privately” means distributing a binary compilation?*

RS: No, no, that’s not private, that’s distributing it to others. If you distribute just a binary to somebody and you don’t let him have the source code, then its not free software for him. So, copy-left licenses like the GNU general public license, require that if you make a binary available, that if you distribute binaries, that those who get the binaries, you must provide source code to. Because if you didn’t, you would be denying them freedom number one which is the freedom to study and change the source of the program so they can make it do what they want.

SH: *What is a typical overhead estimate (addressing Bugs, answering mailing-list questions,) when a company takes its software out to a community under a free software license?*

RS: I have no idea, but I should point out that releasing a program as free software doesn’t require you to engage in collaborative development with anybody, so it may cause you to get more bug reports. But, you presumably are grateful for that, because those bugs that they are reporting are bugs in your code anyway. The only difference is whether you know about them. If you know about them, then you have a chance to fix them, so I presume that anybody who cares about using a certain program is glad to get more bug reports and is probably glad if some of them come with proposed patches which of course they don’t have to use. So they don’t lose anything.

SH: So, let's say this company wants to make a profit through their efforts?

RS: Sorry, you're being a bit vague, the company is doing what?

SH: We have a company who develops software...

RS: They just want to develop software for a profit?

SH: Yes.

RS: That may not work with free software, you know in some situations, it does and in some it doesn't. I won't claim that it is always possible to make money and respect peoples' freedom. What I will say is that it's wrong to make money by not respecting peoples' freedom. There are some things that just shouldn't be done.

SH: OK, but is it wrong in all cases?

RS: It's always wrong to trample peoples' important freedoms to make a profit. In some cases, you can make a profit and respect peoples' freedom and in other cases, you just shouldn't do it.

SH: Can you give me a case?

RS: A case of what? I just made a statement that is so general it applies to all business? There are companies that make money by shipping poisonous wastes to third world countries and dumping them on the beach or on the street or in the rivers and they make people sick. Now this is a case where it's profitable and people obviously shouldn't do it. We're talking about a different issue.

SH: We're talking about software.

RS: It's not a matter of life or death I agree, free software is not a matter of life and death, it's just a matter of having freedom in your life or not.

SH: Well, we'll get back to that in a minute because I do have an interesting question in that regard. Let's talk about software here.

RS: I made a general case for all business.

SH: You gave me an either or - you said that in some cases it's possible for companies to respect peoples' freedom and turn a profit.

RS: And in other cases, they can't make a profit and then they go out of business.

SH: So my question to you is in what cases do you know of, is it possible to make a profit and respect peoples freedom with software?

RS: There are a lot of free software companies that are going along that do respect peoples freedom. As examples, I can mention the big ones I know of are things like Adacore technologies and MySQL and TrollTech. Now, they have different business models and there are a lot I believe of small companies that sell support but part of the act of supporting free software is that they improve it.

SH: For me, that's the only business model that comes to mind.

RS: It's not the only one, those three companies have other business models.

SH: What's another one?

RS: I don't know, I don't try to keep up on this so I know of some examples. But I, you know, focusing on business is not what I do, because lots of other people do that so I don't think that's the most important thing for me to do. My focus is on freedom.

SH: Two more questions and the first one is the free software movement is now 23+ years old. How do you evaluate the successes and failures of this effort?

RS: It is incomplete obviously and we've achieved an important milestone, because we have complete free operating systems which are now convenient for ordinary people to use, so we have free graphical interfaces, we have free office suites, we have thousands of free applications. On the other hand, we need thousands more free applications, so we have a long ways yet to go. And in addition, only a few percent of users are using free operating systems and those are often using some non-free applications so it's not as if we have completely given them freedom. So in that respect, we also have a long way to go. I'm in this for the duration.

SH: OK, so in the next few years, it would presumably be getting more of the software that's missing.

RS: Yes, technically that's what we try to do, but the biggest obstacle to the adoption of free software nowadays is not technical its social inertia. So, what the free software foundation focuses on; of course this is just one organization in a larger free software movement, what the FSF focuses on is political activities to educate people about the ethical issues. To teach people to demand freedom and to campaign against obstacles that make it hard for people or inconvenient for people to use free software. Things like hardware devices whose specifications are secret and which therefore it's hard to write free software to operate. And the developers of these hardware devices may provide their specs to Microsoft but not to the public well that's not good and then there are file formats that are secret or patented and then there is the bios problem where for instance there is no laptop you can get which comes with enough specifications that we can make the free bios run on it.

SH: You would be happy to know that that situation is changing.

RS: I would be glad to know. What can you tell me?

SH: SNIA has a sister organization called DMTF, a standards body as well, and there are working on a standard called DASH. Don't ask me what the acronym stands for, can't think of it off the top of my head, and the purpose of that effort is in fact to design a specification that would enable a developer of software to write a single application and interact with many different devices at the hardware level.

RS: Well OK, but the thing is, the problem with laptops is not a matter of, yeah it might be easier if devices were more standardized in their interfaces, but the big problem is not lack of standardization, after all we have got a substantial amount of manpower and we've written free drivers for lots of different things, the problem is that there's hardware that they won't tell us how to support and their refusal doesn't come from the fact that the interface is merely non-standard. I don't know what to do, I don't know what's causing it but...

SH: So if the interface were standardized and well known and they didn't change it...

RS: Then we wouldn't have to find out anything.

SH: That's what I wanted to make you aware of.

RS: Well, I have to find out more, before I rejoice. Depends how much companies such as Intel adopt and follow this specification.

SH: They are actually driving it believe it or not.

RS: Well, maybe it's hopeful.

SH: So it sounds like for the most part, one component of the free software movement is the moral component if you will?

RS: Yes, well that's the foundation for everything, cause that's the reason for doing it.

SH: So I have a moral question, let's say for the sake of interesting discussion artificial intelligence is finally achieved in software, such that that software application can pass the Turin test. It is quickly realized by the inventors that this software can be extended to beyond human conversation to robots taking on complex tasks autonomously, automated complex systems, and otherwise greatly enhancing all of humanity. But, it is also realized that this technology could be used to subjugate whole

communities by restricting travel and putting down resistance to that subjugation and so on.

RS: First of all, I have to say that anyone who kept this under wraps and kept this to himself is most likely to be a participant in that subjugation. Look again at the United States – who has subjugated us? Our mega-corporations together with their pet government in Washington. Its not a matter of some of say some underground organizations, they cannot do near as much damage because they can only operate underground.

SH: So let's say that these inventors for the sake of argument, are completely committed to releasing this software to all of humanity for the very purposes of avoiding such a situation, for the reasons you describe however, this software, this artificial intelligence could be used for malicious purposes – for example developing a robot that could act as a terrorist with a bomb inside.

RS: And not only that, it could be used by the US government to torture people. The US government has committed more terrorism than Al Qaeda.

SH: You have a moral dilemma, you have technology that could be used for very good things and for also very bad things. Should we have the faith that good things will be done and therefore release it or ought we have pessimism that bad things will be done and keep it private?

RS: Keeping it private wouldn't solve the problem. You'd have to kill it, you'd have to bury it and then you'd have to hope that nobody else comes up with it and the history of technology shows that usually when it's time for some development, lots of people develop it. So the problem in this scenario is that the real alternative of burying it probably doesn't exist. The real alternatives are a matter of release it to everybody or release it to just a few but who are those few likely to be? And how can you really trust them? Now the big danger of such a smart AI is that it would be smarter than we are so it would be able to trick us easily and I'm really scared of what would happen if anybody ever develops that and I suggest reading the book "Excellerando," see what some of the danger is. Now the interesting thing in that book is essentially it says that the AI becomes corporations and results in things which are cleverer than human beings and basically push human beings out of the solar system, cause there is no room for them anymore, they can't compete. But these things aren't like humans only smarter, they aren't beings at all- they have no consciousness, they are just good at competition. They are mindless, clever corporations.

SH: There are several interesting fictional works on this topic and a few movies I believe.

RS: Well the movies are probably stupid because they usually are. It's a very dangerous thing and I don't have a solution, I don't see a solution.

SH: Well one possible answer could be if you could figure out who the trusted were in some way?

RS: But you can't and the problem is that the most likely organizations that would be the ones who had the sole access, would be corporations or governments and they are not, they are just the ones we've got to be most afraid of.

SH: No doubt.

RS: I don't claim to have an answer.

SH: It's certainly a moral dilemma. Some day we may have this problem unfortunately.

RS: May be and humanity may be deleted.

SH: Let's hope not.

I wanted to thank you for taking your time and speaking with us and again for speaking at SNIA's Developer Conference in September.