Introduction

Every generation has its philosopher—a writer or an artist who captures the imagination of a time. Sometimes these philosophers are recognized as such; often it takes generations before the connection is made real. But recognized or not, a time gets marked by the people who speak its ideals, whether in the whisper of a poem, or the blast of a political movement.

Our generation has a philosopher. He is not an artist, or a professional writer. He is a programmer. Richard Stallman began his work in the labs of MIT, as a programmer and architect building operating system software. He has built his career on a stage of public life, as a programmer and an architect founding a movement for freedom in a world increasingly defined by "code."

"Code" is the technology that makes computers run. Whether inscribed in software or burned in hardware, it is the collection of instructions, first written in words, that directs the functionality of machines. These machines—computers—increasingly define and control our life. They determine how phones connect, and what runs on TV. They decide whether video can be streamed across a broadband link to a computer. They control what a computer reports back to its manufacturer. These machines run us. Code runs these machines.

What control should we have over this code? What understanding? What freedom should there be to match the control it enables? What power?

These questions have been the challenge of Stallman's life. Through his works and his words, he has pushed us to see the importance of keeping code "free." Not free in the sense that code writers don't get paid, but free in the sense that the control coders build be transparent to all, and that anyone have the right to take that control, and modify it as he or she sees fit. This is "free software"; "free software" is one answer to a world built in code.

"Free." Stallman laments the ambiguity in his own term. There's nothing to lament. Puzzles force people to think, and this term "free" does this puzzling work quite well. To modern American ears, "free software" sounds utopian, impossible. Nothing, not even lunch, is free. How could the most important words running the most critical machines running the world be "free." How could a sane society aspire to such an ideal?

Yet the odd clink of the word "free" is a function of us, not of the term. "Free" has different senses, only one of which refers to "price." A much more fundamental sense of "free" is the "free," Stallman says, in the term "free speech," or perhaps better in the term "free labor." Not free as in costless, but free as in limited in its control by others. Free software is control that is transparent, and open to change, just as free laws, or the laws of a "free society," are free when they make their control knowable, and open to change. The aim of Stallman's "free software movement" is to make as much code as it can transparent, and subject to change, by rendering it "free."

The mechanism of this rendering is an extraordinarily clever device called "copyleft" implemented through a license called GPL. Using the power of copyright law, "free software" not only assures that it remains open, and subject to change, but that other software that takes and uses "free software" (and that technically counts

as a "derivative work") must also itself be free. If you use and adapt a free software program, and then release that adapted version to the public, the released version must be as free as the version it was adapted from. It must, or the law of copyright will be violated.

"Free software," like free societies, has its enemies. Microsoft has waged a war against the GPL, warning whoever will listen that the GPL is a "dangerous" license. The dangers it names, however, are largely illusory. Others object to the "coercion" in GPL's insistence that modified versions are also free. But a condition is not coercion. If it is not coercion for Microsoft to refuse to permit users to distribute modified versions of its product Office without paying it (presumably) millions, then it is not coercion when the GPL insists that modified versions of free software be free too.

And then there are those who call Stallman's message too extreme. But extreme it is not. Indeed, in an obvious sense, Stallman's work is a simple translation of the freedoms that our tradition crafted in the world before code. "Free software" would assure that the world governed by code is as "free" as our tradition that built the world before code.

For example: A "free society" is regulated by law. But there are limits that any free society places on this regulation through law: No society that kept its laws secret could ever be called free. No government that hid its regulations from the regulated could ever stand in our tradition. Law controls. But it does so justly only when visibly. And law is visible only when its terms are knowable and controllable by those it regulates, or by the agents of those it regulates (lawyers, legislatures).

This condition on law extends beyond the work of a legislature. Think about the practice of law in American courts. Lawyers are hired by their clients to advance their clients' interests. Sometimes that interest is advanced through litigation. In the course of this litigation, lawyers write briefs. These briefs in turn affect opinions written by judges. These opinions decide who wins a particular case, or whether a certain law can stand consistently with a constitution.

All the material in this process is free in the sense that Stallman means. Legal briefs are open and free for others to use. The arguments are transparent (which is different from saying they are good) and the reasoning can be taken without the permission of the original lawyers. The opinions they produce can be quoted in later briefs. They can be copied and integrated into another brief or opinion. The "source code" for American law is by design, and by principle, open and free for anyone to take. And take lawyers do—for it is a measure of a great brief that it achieves its creativity through the reuse of what happened before. The source is free; creativity and an economy is built upon it.

This economy of free code (and here I mean free legal code) doesn't starve lawyers. Law firms have enough incentive to produce great briefs even though the stuff they build can be taken and copied by anyone else. The lawyer is a craftsman; his or her product is public. Yet the crafting is not charity. Lawyers get paid; the public doesn't demand such work without price. Instead this economy flourishes, with later work added to the earlier.

We could imagine a legal practice that was different—briefs and arguments that were kept secret; rulings that announced a result but not the reasoning. Laws that

were kept by the police but published to no one else. Regulation that operated without explaining its rule.

We could imagine this society, but we could not imagine calling it "free." Whether or not the incentives in such a society would be better or more efficiently allocated, such a society could not be known as free. The ideals of freedom, of life within a free society, demand more than efficient application. Instead, openness and transparency are the constraints within which a legal system gets built, not options to be added if convenient to the leaders. Life governed by software code should be no less.

Code writing is not litigation. It is better, richer, more productive. But the law is an obvious instance of how creativity and incentives do not depend upon perfect control over the products created. Like jazz, or novels, or architecture, the law gets built upon the work that went before. This adding and changing is what creativity always is. And a free society is one that assures that its most important resources remain free in just this sense.

For the first time, this book collects the writing and lectures of Richard Stallman in a manner that will make their subtlety and power clear. The essays span a wide range, from copyright to the history of the free software movement. They include many arguments not well known, and among these, an especially insightful account of the changed circumstances that render copyright in the digital world suspect. They will serve as a resource for those who seek to understand the thought of this most powerful man—powerful in his ideas, his passion, and his integrity, even if powerless in every other way. They will inspire others who would take these ideas, and build upon them.

I don't know Stallman well. I know him well enough to know he is a hard man to like. He is driven, often impatient. His anger can flare at friend as easily as foe. He is uncompromising and persistent; patient in both.

Yet when our world finally comes to understand the power and danger of code—when it finally sees that code, like laws, or like government, must be transparent to be free—then we will look back at this uncompromising and persistent programmer and recognize the vision he has fought to make real: the vision of a world where freedom and knowledge survives the compiler. And we will come to see that no man, through his deeds or words, has done as much to make possible the freedom that this next society could have.

We have not earned that freedom yet. We may well fail in securing it. But whether we succeed or fail, in these essays is a picture of what that freedom could be. And in the life that produced these words and works, there is inspiration for anyone who would, like Stallman, fight to create this freedom.

Lawrence Lessig
Professor of Law, Stanford Law School.
