

Foreword

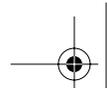
by Lawrence Lessig[†]

Open source resources are completely common within our society. Yet open source and free software remain a mystery. Science, public highways, city parks, language—these are at the core of any free society. Commerce gets built upon them. Culture flourishes through them. Yet the same ideas applied to software puzzle many people. Language can be free, and the results of science open to all, yet to many, software is, or should be, proprietary.

Why code must be proprietary is a question whose answers have changed over the past ten years. At first, the reasons were technical: no free or open source project, it was said, could develop the highly complex and robust code necessary for modern software applications. But when the GNU/Linux project began to produce an operating system that rivaled Microsoft's in robustness and efficiency, this technical argument began to fade.

[†] Professor of Law, Stanford Law School, and author of *Code and Other Laws of Cyberspace* (Basic Books, 2000); *The Future of Ideas: The Fate of the Commons in a Connected World* (Vintage Books, 2002); *Free Culture: How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity* (Penguin Press, 2004).



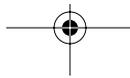


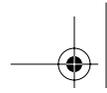
In its place, many offered an argument of commercial necessity: No free or open source project could survive commercially, given the high costs of quality programming, and the inability to exclude others from the benefits of that quality. But again, when companies such as IBM and HP began to invest billions in free software development, this argument too began to weaken. And as the embedded systems market began to take off, built as it is upon open and free software, it became clear to most disinterested observers that open source and free software were elements of a different business model, not opponents to business. Whether this different business model will produce more profits for the technology sector generally is an empirical question we rely upon markets to resolve. But that it does make money for some is no longer subject to doubt.

Having failed to convince the world that propriety software is technically necessary, or commercially necessary, the opponents of free and open source software now argue against it on the basis of legal necessity. At the most extreme (and absurd), SCO President Darl McBride argues that free software licensed under the GPL is “unconstitutional.” At the center are those allied with Microsoft, who argue that the licenses supporting the most popular free and open source projects are “dangerous” and “unproven.”

In this beautifully clear and accessible work, Lawrence Rosen defuses this last, and equally fallacious, argument against open source and free software. While he doesn’t waste trees responding to the ridiculous claims of McBride, this book builds a framework within which the family of free and open source licenses can be understood. And in a rare talent for a lawyer, Rosen succeeds in making these points about the law meaningful and understandable to anyone at all.

This is the great value of this perfectly timed book. As open source software is among the fastest growing and most important software produced, it has become necessary for a wide





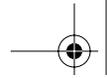
range of people to understand the foundations upon which it is built. Businesses and investors, as well as technologists and scientists, need to understand how pedestrian the legal framework is within which open source and free software are grounded. Policymakers need to see that this distinctive model for creating and spreading knowledge about code is neither communism come to the digital age, nor a binary version of Thomas More's *Utopia*.

The need for this understanding was never more clear to me than when I read about our own government's view about open source and free software. In July 2003, the United States government vetoed a request that the World Intellectual Property Organization (WIPO) hold a meeting to discuss "open collaborative models for producing public goods," including open source and free software. Lois Boland, director of international relations for the U.S. Patent and Trademark Office, explained "that open-source software runs counter to the mission of WIPO, which is to promote intellectual-property rights." As she is quoted as saying, "To hold a meeting which has as its purpose to disclaim or waive such rights seems to us to be contrary to the goals of WIPO."

This statement is astonishing on a number of levels, and Rosen's book demonstrates why. Most obviously, open source and free software is not "counter to the mission of" an organization that "promote[s] intellectual property rights," as open source and free software generally relies upon intellectual property to achieve their effect. The most important open source and free software is not software in the public domain. It is instead, like Microsoft's software, software protected by intellectual property law and licensed to users on the terms chosen by the property owner.

No doubt the property owners in an open source and free software project "disclaim or waive" some of their rights. But again, it is puzzling why property owners choosing how to ex-





ercise their rights could ever be “contrary to the goals of WIPO.” Is it against the property system generally when Bill Gates gives \$20 billion to help the poor in Africa? Are public highways latent communism?

Boland’s view is grounded in a mistaken understanding of the way open source and free software function. In my view, no one who understands what this book teaches could have any principled opposition to this business model competing with any other. There will be a great deal of social wealth created by this family of licensing. There will be an even greater amount of knowledge and freedom that is spread by this legal arrangement. Rosen has done us all a great service by making understandable the legal tools that make these goods possible.

