Free Software Matters: Patently Controversial

Eben Moglen*

October 20, 2001

The patent system is heralded by some around the world, particuarly pharmaceuticals companies, as essential to innovation. As I have written here before, patents not only do not assist in the production of innovative software, they can potentially destroy the free software production system, which is the world's most important source of software innovation.

Copyright law permits authors to control the copying and distribution of their expressions, but copyright law does not permit the monopolization of ideas. Patent law works differently: once someone has been permitted by a patent office to "claim" a technical process or invention, no one in the place where that patent has issued may "practice" of the patent, even if he independently invented, without a license.

Until the 1970s it was assumed everywhere in the world that this form of idea monopolization could not be used to own an algorithm, any more than it could be used to own particular ideas in mathematics. But as the political system in the United States became more corrupt, and the owners of "intellectual property" became more directly able to convert their wealth into political results through "campaign contributions," patent doctrine in the United States began to shift towards permitting all sorts of ideas, including obvious and basic concepts in computer programming, to be patented. The weight of the US government has been put behind the globalization of those new principles, resulting in a worldwide move to give unfair advantages to "owners," damaging freedom of invention in many areas.

*Eben Moglen is professor of law and legal history at Columbia University Law School. He serves without fee as General Counsel of the Free Software Foundation. You can read more of his writing at http://moglen.law.columbia.edu.

When someone makes an unfree computer program that behaves usefully, the free software movement encourages people to write a program of their own that accomplishes similar goals, and then give other people the fundamental rights to use, copy, improve and share the program. The result is the rapid innovation and engineering improvement that has produced—in less than twenty years and for the most part in less than a decade—an entire software environment for computers of all sorts, reliably and flexibly competing on equal or superior terms with proprietary products that have cost immense amounts to produce, and are sold at enormous prices. The body of those programs, collectively, is the single greatest technical teaching tool in the world, permitting any motivated student, anywhere to get right to the state of the art in any area of computer programming or design by reading free code.

But if useful computer behavior is "claimed" by a patent, independent reinvention by others is prohibited, and free software is excluded altogether. We cannot just buy a patent license, because though free software isn't always free like free beer, it cannot exist at all unless it is free like free speech: everyone has to be allowed to take free code from one place and use it in another, or build on it, so long as she is willing to share and share alike.

Sometimes, perhaps increasingly, even those who have succeeded in obtaining patents on software ideas recognize the value of permitting the idea's use in free software. A license that allows anyone to practice the patent in GPL'd software, for example, can be fully consistent with the inventor's other uses of the patent. An inventor who makes such a license knows that free use can only be made of her invention in programs that can never be made proprietary, in any version, because the GPL permanently protects those programs' freedom. So the inventor too benefits from those uses of the invention, while retaining the power to exclude proprietary users of the technology who don't take licenses and pay royalties. The Free Software Foundation recently negotiated such a license for a patent that discloses a means to make the Linux kernel effective as a platform for "hard real-time" applications. This license, granted by an inventor who also engages in proprietary licensing and sells software of its own, is a very important example of arrangements we shall see more widely in the near future. It also resolves an important source of legal uncertainty about the use of GNU/Linux for real time applications, and will stimulate an explosion of new work in the area.

But the patent problem may be growing worse in a more dangerous way. The W3 Consortium, which standardizes how the Web technically works, has been considering a proposal to permit patented software technology to be included in W3C standards recommendations. The resulting Web could have basic services covered by royalty-requiring patents, which could never be implemented in free software at all. This proposal has been under intense public scrutiny for the last several weeks, and as a result the W3C invited me and Bruce Perens—long a distinguished entrepreneur of free software—to form part of the Working Group considering its patent policy. If the W3C moves towards a patent policy that is friendly to free software as well as patent holders, it will have a profound and positive influence on standards throughout the industry. If it fails, and does not grasp why free software matters, the Web could become a very different, more corporate, more monopolized, and ultimately unfree place.