Comments regarding Competition & Intellectual Property
Mark Ellis

Introduction.

We don't know in advance where technology can take us, and we're unlikely to agree on where we want to go. Therefore, no pseudo-rational "top-down" technology policy can succeed. We must rely on the marketplace to evaluate the efforts of self-interested innovators in search of paths that lead to places worth visiting. Under normal circumstances we will be aided in this process by the sheer relentlessness of corporate capitalism. But when the big winners in nominally free markets are offered opportunities to protect their winnings against potential competitors, the pursuit of profit can actually work against technological progress. When innovation is less profitable than making a competitor's innovations irrelevant, the system has failed.

Antitrust law seems likely to fail in the Microsoft case, leaving something like a Federal seal of approval on that corporation's power over the future of information technology. But even if we believe that Judge Jackson's Findings of Fact describe Microsoft's hegemony accurately, we must question not merely his now-abandoned plan for breaking up Microsoft, but also the reasonableness of expecting *any* judge or legislative body to craft an ideal solution. We can easily see many ways in which Microsoft exploits the world's dependence on Windows to extend our dependence on Microsoft into new areas, but there are no obvious lines along which antitrust law and justice demand that we construct legal barriers among Microsoft's products or divisions. Those who would be given the power to dictate which software products must or must not be integrated with an operating system would be very unlikely to have an adequate technical understanding of the consequences of software design decisions; those with the necessary knowledge of the industry could not be trusted to make such decisions impartially. Judge Jackson's proposed remedies were arbitrary, but any others would be equally arbitrary.

Perhaps antitrust law works when the question at issue is something like whether a price-fixing agreement does or does not exist - do, for example, the major record labels act as one to discourage discount retailers from offering “excessively” low CD prices to consumers? That question can be answered (affirmatively, in this case) by considering readily available facts. The Microsoft case, however, invites exactly the kind of ill-informed and arbitrary manipulation decried by people who take Microsoft's side because they're opposed on principle to all antitrust law. Those advocates of free markets claim that the marketplace makes monopolies impossible, that *only through government action can monopolies become so deeply rooted as to deform the marketplace around themselves*.

This is exactly what has happened. By extending patent and copyright law to cover pure information as well as rights to market physical products, and to define the relationships between producers and consumers as well as those among producers, the U.S. government has created an environment in which marketers of information products are free to seek dictatorial powers over their customers. Microsoft has led the way, and the corporations that sell prerecorded music and video are seeking to follow.

The Fallacy of "Intellectual Property Rights".

Why are property rights, which limit our freedom of action by stopping us from using each others' property, necessary in a free society? Because they prevent the conflicts of interests that would otherwise inevitably result when any number of people have an equal right to say what is done with particular things that can't simultaneously serve all those people's purposes. The Marquis de Sade's brand of "take what you can" anarchy collapses to feudal rule by the strongest and most vicious, while the "we can work it out" philosophy of the left-anarchists is the way to a stagnant totalitarian system which must attempt to take everyone's interests into account when allocating any particular thing to any particular use.

"Intellectual property rights" - in the literal sense of unconditional ownership of ideas and their expression - have no such justification, because there need be no conflicts of interests over intangible entities that aren't subject to scarcity. Worse, any regime that tries to enforce such "rights" must become totalitarian; it must seek out "thieves" when any number of people may simultaneously "steal" the same piece of "property" in scattered locations and in secret, without making the "crime" apparent by taking anything away from the "property owner".

Property rights make the rule of law possible by allowing the resolution of disputes over real goods which exist in limited quantities; "intellectual property rights" undermine the rule of law by granting to particular people - or, under current conditions, particular corporations - the privilege of using government power to create shortages for profit.

I might be accused of knocking down a straw man here, because "intellectual property law" has traditionally dealt only with the commercial use of patents, trademarks, copyrights, and trade secrets by companies and individual creators, and has had no direct impact on the private conduct of individuals. Unfortunately, these reasonable limitations no longer exist in certain industries; corporations that sell digital products now use the legal system to protect or increase their profits by controlling their customers' behavior.

Industry organizations representing holders of patents and copyright remind us that the U.S. Constitution is the foundation of "intellectual property rights" in this country: " The Congress shall have Power to... promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries..." However, this falls far short of creating a distinct kind of property right.

- Although Congress is given power to make legislation in this area, it is neither told that it must do so nor given any specific description of the rights to be secured. However, the First Amendment places definite limits on this and all other legislation: "Congress shall make no law... abridging the freedom of speech, or of the press." The theory that copyright somehow "trumps" the First Amendment - which would, if enforced consistently, close all public libraries - has no foundation in the Constitution.

- The fact that these exclusive rights, whatever they might be, exist only for limited times is another clue that what we're discussing is no property right but simply an arrangement under which individuals get a *temporary franchise* in return for the benefits they provide to the community.

- Underlining the previous idea is the fact that the Constitution presents this clause as a means to "promote the Progress of Science and useful Arts"; it states no basic human right, but only an intent to promote the general welfare. Laws that protect the privileges of copyright holders without regard to their effect on the general public - i.e., laws establishing property rights - are not justified by the "Authors and Inventors" clause.

- Finally, we should note that the franchises are to be granted directly to the authors and inventors. This fits the model in which authors and inventors license their creations to publishers and manufacturers, and can be stretched to cover work done by a company's employees, but casts doubt upon the practice of buying and selling copyrights which ends with media corporations "owning" enormous portfolios of works acquired over many decades from many non-employee creators.

I therefore claim that the U.S. Constitution supports no legislation that creates property rights to information, nor does it suggest a rights-holders' exemption to the First Amendment's rejection of laws that block public or private sharing of information. What, then, does the "Authors and Inventors" clause accomplish? Simply what was taken for granted in the days before electronic media: it allows regulation of the *commercial distribution* of patented and copyrighted material. The software and mass communications corporations have been very successful at writing and lobbying for laws that expand their limited temporary monopolies far beyond this, of course, but that has allowed them to develop highly profitable business models at the expense of their customers' rights and the public domain. We must not ignore this fact as we examine the ability of one software company and four or five mass communications companies to suppress potential competition.

All Rights Reversed: The Art of the End User License.

Property can be sold. If I own something, I can use it as I wish; if I sell it to you, *you* can use it as you wish.

If you are an author or inventor, you can no more sell me the status of being author or inventor of a particular work of yours than you can sell me the right to call black white. But if the monetary value of your work comes from the market for widgets embodying your work, and I have a widget factory, and you don't, then we'll both be better off if I can pay you for a license to use your work to make widgets.

In the digital world, where copyrights have been magically promoted to property rights, the term "licensing" has been extended to something beyond negotiated contracts among authors and publishers. The "end user license agreement" (EULA) has become familiar to computer users as a legal-looking document that concedes no rights to software purchasers except the right to non-defective distribution media and the right to remain silent. If EULAs are to be believed, IP rights clearly trump conventional property rights, since users are told how they may and may not use the products for which they've spent perfectly good no-strings-attached dollars. Some EULAs go beyond the usual disclaimers to forbid public criticism of the product, and even to prohibit publication of the results of objective tests of product performance. For example, the users of Microsoft’s Visual Studio .NET are told by the relevant EULA that they "may not disclose the results of any benchmark test of the .NET Framework component of the OS Components to any third party without Microsoft’s prior written approval."

On the other hand, consumers have good reason to doubt that the EULA has any real force since the actual transfer of ownership of the EULA-covered product invariably takes place without reference to the EULA's terms. Common sense suggests that off-the-shelf software products deserve no special exemption from our expectation that the products we buy should do the things they are advertised to do and should do nothing harmful to us or to our organizations. If taking no responsibility for the performance of one's product is so easy, why should Ford, Firestone, or the asbestos companies have to worry about product liability?

Although the very well-funded and well-connected industry associations have had great success in obtaining legislation of the kind they wanted, the situation "on the ground" has been different. The ordinary person's understanding of property rights suggests that he ought to be able to do what he wants with the things he owns. He doesn't bother to read the lawyer-speak that comes with the things he buys, and as law professor Jessica Litman has observed, his usual reaction to being shown exactly what the lawyers have written is "there can't really be a law like that." He is moved very little by tales of the hardships that his unauthorized file sharing inflicts upon the software and entertainment industries. And as technological means are found to loosen that industry's grip on its customers, he will use them.

The Pathology of the Software Industry.

Software is information. A software program consists of statements in some language that collectively describe a way to accomplish some task. Software has practical uses because computers exist to run it, but since "running software" is a process of manipulating symbols according to well-defined rules, it can be modeled in full detail by mathematical statements on paper, a mechanical contraption, or a board game. Text representations of computer programs are, in fact, very important: practically all software of significant complexity is created in text form by its developers. The "source code" that programmers type into their machines - exactly as I'm now typing this essay into mine - becomes the "machine code" that computers execute simply by being translated into that form according to the deterministic rules of another program. David Touretzky's "Gallery of CSS Descramblers", which presents many different symbolic representations of a single chunk of software, illustrates the absurdity of treating the symbol manipulation that software accomplishes as something remote from the symbolic operations of mathematics, formal logic, or ordinary speech.

In one respect, software's legal status as information has generally been accepted: software has long been considered subject to copyright. However, for other purposes, judges have ruled that software may be treated as a kind of useful device, the distribution of which may be limited by means that would violate the First Amendment if applied to other kinds of information. A “CSS descrambler” logically equivalent to those mentioned above - a trade-secret recipe for reversing a scrambling process used by DVD makers to ensure that only members of the DVD cartel can make DVD players - has in fact been declared by a judge to be a copyright circumvention device, and a web link to a site offering CSS descrambling software has been found to be illegal under the Digital Millennium Copyright Act.

In a perverse legal environment like this, the nature of software allows objectively valueless "innovations" to marginalize all products competing with those of a market leader, whether we're talking about Microsoft now or IBM thirty years ago. *Microsoft’s ability to behave as a monopoly depends upon the fact that the patent and copyright laws explicitly give Microsoft monopolies on the distribution of certain information.*

Electronic mail provides a simple illustration of the problem. One computer sends a message - a sequence of bits - over a network to another. If the software used to prepare the message is compatible with the software used to read the message, the message's recipient can read what the sender wrote. Here, "is compatible with" means something like "follows the same conventions for translating between raw network data and readable text." Stating exactly what these conventions should be is boring and specialized work, but we need not be specialists to realize that e-mail users need to be able to communicate with each other regardless of what hardware and software they're using. Agreeing on a set of conventions is essential. "Innovations" that make communication impossible are worse than useless. For example, communication breaks down when a message is sent in a fancy word-processor format that its recipient's e-mail program can't decode.

This is not a hypothetical example. If I, using the basic Internet mail protocol on a machine running the Linux operating system, exchange messages with you and your Microsoft system, you can unknowingly send me a response that I can't read. The software you're using may be able to send and receive plain text e-mail, but will by default be configured to send even simple e-mails in a complex format that can be interpreted correctly only by that software. You are using a trade-secret protocol, encapsulated in machine code, the distribution of which is controlled by copyright. You are sending messages that can be legally interpreted only by Microsoft customers.

E-mail is the simplest example to understand, but computer systems contain many other kinds of software interfaces through which applications software connects to the operating system, query software connects to databases, network software connects to servers, and so on.

One might reasonably expect the marketplace to prevent this problem, since customers will naturally prefer to use e-mail software that doesn't create obstacles to communication. No one wants a telephone that can only call other similar phones. But, given a legal system that is friendly to marketers of shrinkwrap software, what actually happens is an avalanche toward monopoly. In the first stage of such a market, there may be an early innovator, and several competitors pursuing market share by adding features. Adding features subtracts interoperability; if you want to put pictures or sound recordings in e-mails, you'll need more complex file formats, and when several vendors are creating their own formats, it's very unlikely that they'll all happen to stumble into a common format. One format may well be as good as another on technical grounds; what matters more is the market power of the formats' proprietors.

The avalanche toward monopoly is powered by a few simple principles:

- *Where interoperability issues exist, technical standards are essential.* Systems and software have to work together. The interplay between standardization and innovation is too complex and dynamic to be dictated by a formal or political process; standards are set in the marketplace, and when the community of users lacks the sophistication to develop vendor-neutral standards, the competition is between the proprietary "standards" offered by vendors. The openness of the Internet is a direct result of the fact that its core standards were set by its sophisticated and government-funded early users rather than computer or software companies.

- When IBM was in a dominant position like that of today's Microsoft, MIS managers had a saying: "No one ever got fired for buying IBM." The issue is one of avoiding accusations of malpractice: *professionals with responsibilities greatly exceeding their ability to make informed technical judgments can't afford to deviate from accepted professional standards and practices in pursuit of incremental technical improvements.* "Standardization" often consists of choosing the vendor chosen by a majority of one's peers.

- *When "accepted professional standards and practices" dictate the use of the products of a particular company, that company can lock its competition out by deliberately creating partially-incompatible products and, in effect, selling compatibility.* If the "standard" is a

> product which conforms to no openly-defined specification, nothing keeps its proprietor from making that specification a moving target, thwarting the attempts of would-be competitors to reverse engineer the product, and pushing users to upgrade to newer versions of the product to pursue compatibility. When the favored product is a "platform" upon which other goods and services may be built, the platform proprietor enjoys immense advantages over all other vendors seeking to provide those goods and services. Judge Jackson's Findings of Fact - and the experiences of millions of users - illustrate this principle with examples of Windows upgrades that break user-installed software products from Microsoft's competitors. Microsoft is now not merely admitting this, but claiming it's a good thing that enables Windows to provide a consistent experience to users.

It's difficult to imagine that the many billions of dollars that businesses spend on upgrading Microsoft software translates into productivity improvements of comparable size; however, Microsoft assures itself of a continuing source of income by making sure that compatibility between newer and older PCs must be purchased. But with very little market share left to capture, and a shrinking growth rate for PC sales, continuing growth for Microsoft means using its power in the OS market - its ability to decide which products will and will not be compatible with ninety-five per cent of personal computers - to sweep away competition in related markets.

The Menace of Software Patents.

Even in the United States, there has been resistance to the concept of software patents. The collapse of this resistance in recent years is probably due to the fact that computers are now used daily by many millions of people who know very little about how software works. When computers were rare and exotic tools of Big Science, it was easy for the uninitiated to imagine that computer programmers were scientists who used sophisticated mathematics to create their programs. Today, when almost everyone is a computer user and much computer programming is simply clerical work, it's easier to overlook the hard science that makes computing possible. Nevertheless, computing is a technology that rests entirely upon mathematical science.

The electronic technology in today's computers is used because it happens to be the fastest economically reasonable means for performing automatic computations, but the relationship between computation and any particular kind of computer is like the relationship between time and clocks: whether you use electronic oscillators, springs and gears, sand, or shadows to tell time, the nature of time is the same. The nature of automatic computation, as developed by mathematicians like Alan Turing and John von Neumann, was envisioned in thought experiments and formalized in theorems and proofs. Algorithms, the recipes for performing computations, are formally stated and proved as theorems like those of geometry, algebra, and other familiar branches of mathematics.

"Owning" an algorithm is like owning the Pythagorean theorem - yes, the result may be quite useful, but it is also a demonstrable truth, inherent in the mathematical system in which it is expressed. This is not to say that algorithms "exist" before anyone writes them down, but although the creator of an algorithm may be pursuing more or less novel lines of development, his ability to "invent" anything is constrained at every step by the laws of logic and mathematics. In this as in other areas of mathematics, two people working independently with the same basic idea will often use different notation and terminology to accomplish exactly the same things - not by coincidence, but because of the internal self-consistency of mathematics.

In recent years, patents have been awarded to the proprietors of many algorithms for compressing, encrypting, filtering, error-correcting, and otherwise manipulating commercially-important datastreams. All owe their existence to enormous amounts of unpatented mathematical work, and few add very much to the state of the art. The value of these patents doesn't come from the the intrinsic quality of the work; the patented LZW compression algorithm is much inferior to the unpatented gzip algorithm, and the unpatented Vorbis audio compression algorithm is at least as good for most purposes as the multi-patented MP3 algorithm. The monetary value of such a patent comes from its use in "standards" schemes which allow no compatibility without payment or pool membership.

Better known than algorithm patents are the application patents that simply cover the use of ordinary programming methods and algorithms in a particular application. Probably the most notorious patent of this kind is the Amazon.com “one-click” ordering system. The technical foundation of this patent is the idea of a “cookie”, a unique ID number stored on an Internet user’s computer in an interaction with a particular web server and thereafter sent automatically from the user to the server as part of each additional transaction with that same web site. Without cookies, web servers have no way to associate a series of transactions with a single user, but upon receiving a cookie, a web server can query a database to find any or all data pertaining to the user identified by the cookie. This is not Amazon’s invention; it is exactly what the “cookie” mechanism was created for by the early developers of the World Wide Web. Amazon keeps customer data on file - like every mail-order company - and during the one-click order process, an order is created by combining Amazon’s data about the selected product with credit card and address data submitted during earlier transactions with the customer identified by the cookie.

Amazon, by using cookies to do just what they were created for, and by having the audacity to call its process an invention in a patent application, was granted a legal weapon to use against its competitors. Unsurprisingly, Amazon chose not to try to get royalties from all cookie users, but instead to bring an infringement suit against a leading competitor, the online service of Barnes and Noble.

The fact that a machine happens to be involved in a process does not transform a way of doing business into an invention. A promoter’s plan to create a profitable monopoly does not transform methods that any reasonably skilled software architect would use into innovations.

Because computing is a mathematical discipline that builds systematically upon a rather small set of basic concepts, there exists no middle ground between the discovery of significant algorithms that are too fundamental to be patented, and, on the other hand, practical applications of those algorithms that embody too little innovation to deserve patents. Much effort and ingenuity goes into turning algorithms into programs, but this is like the effort and ingenuity that a lawyer or doctor uses while applying knowledge and experience to the solution of a particular client's problem. Would the patent lawyers who support the idea of software patents enjoy having to ensure that they've licensed all precedents and used no other patent lawyer's patented lines of reasoning without permission? Would they like to wait on an operating table for a surgeon to identify and seek licenses for the patents needed to complete an operation?

Software development is quite different from, say, drug development, which demands extensive and costly empirical studies of processes that are not fully understood. If our knowledge of human biochemistry someday becomes so great that creating the next drug is simply a matter of applying well-known methods to whatever problem we decide to solve, patents will cease to have a place in that industry as well.

No Business Like Show Business.

What does a "record company" do? Historically, and as most people understand it, a record company is a manufacturer that has one or more plants for making sound recordings. In this archaic model, there are musicians running loose in the world, performing music that exists only while they perform it, and there are record companies paying musicians for the right to make tangible products - records, tapes, CDs - that serve as containers for their music. The record company is an intermediary between musicians and music listeners, necessary because neither the audience nor the performer has the means to package music for distribution. Looking at the music industry this way, we must wonder why we need record companies when we have a general-purpose information pipeline like the Internet, which carries music exactly as it carries other kinds of data. Of course this is because the record companies - or, rather, the music departments of the multi-media conglomerates - have found ways to change their relationship to the music.

Show business has historically had more than its share of hustlers and racketeers, probably because the value of its products depends entirely on subjective factors. Industries dealing with more tangible goods can use more objective standards to predict which firms and individuals can deliver the goods in question; what the entertainer or promoter has to offer may be known only in retrospect. To offer reasonable and consistent returns on investment in an intrinsically illogical business, the entertainment oligopoly uses a gatekeeper system to ensure that whatever becomes popular can reach the public only through its channels.

Long before there were technological tools for enforcing monopolies, the entertainment industry used all available means to exert control over the paths to show business success. In music, the radio industry leads the way. Station ownership is becoming more highly concentrated, while formats become more numerous and playlists become shorter. By adding to these trends radio's unique scheme of pay-for-play via "independent promotion agents", the system guarantees that any particular song is pushed quickly and at great expense either to national ubiquity within its format, or to oblivion. This system is not one that provides a home for a great variety of subtle or sophisticated music, but it's adequate for its purpose: offering formulaic content to draw a demographically-predictable audience to the advertisements that pay for the whole thing. It works well also for mass-market retailers, who need to add to their stock of proven best-sellers only the limited selection of new music that their customers hear on the radio.

Representatives of the recording industry never tire of telling us how many records and performers must be promoted at what expense for each record that breaks through to become a Big Hit, but they neglect to point out that the high barriers to success serve them well by holding back competition. If recordings could be made and distributed cheaply - as PC-based recording equipment and Internet distribution do in fact allow - there would be no need to spend hundreds of thousands of dollars to launch the next generic record by the next generic "artist". Artists and listeners could find new ways of getting together without the record companies. Hilary Rosen, president of the Recording Industry Association of America, has told Congress that the function of her industry is to "add value" to music by "creating the demand" for it. But the real effect of the absurdly expensive system her industry has created is to push a few lucky performers to great wealth while marginalizing every musician whose talents and vision don't fit the current set of formulas. Neither the public nor the overwhelming majority of musicians benefits from the industry's constriction of the paths to the audience.

We can see easily enough how new relationships between musicians and their fans could replace the radio stations and the record companies. Without the industry’s demand creators to force-feed music to its target audience, financial rewards for the most successful musicians will be less they are today, but a much larger number of musicians will have easier access to potential fans. Even if recorded music becomes entirely free, it will sell concert tickets and create other merchandising opportunities.

For Internet-assisted digital music sharing, the question is no longer how it can be accomplished, but how far the music companies are willing to go to stop it. Napster’s attempt to build a business model by providing access to copyrighted material without dealing with the copyright holders was doomed from the start, but with Napster out of the picture, replaced by “peer to peer” file-sharing software running entirely on end-users’ computers, there is no longer a central facilitator for the record companies to sue. The record companies can still do a lot of harm by harassing Internet service providers and lobbying to have future computers and data storage devices crippled in various ways, but going back to the limited possibilities for music sharing that existed before Napster is no longer an option.

The Internet’s threat to the content companies’ current business models is real. As a Bush administration spokesman said recently about the Enron debacle, business failures are part of “the genius of capitalism”. A company that makes piles of money by serving as an intermediary between musicians and music listeners should - in a truly free market - wither away when its services are no longer required.

Open-source software.

Microsoft's policies create demand for an alternative, which we should expect a free market system to supply. Investors, however, understand that mere supply and demand do not threaten Microsoft’s position in its industry, and will not fund efforts to challenge Microsoft head-on with a new general-purpose operating system or office “productivity” suite.

We can end Microsoft’s domination by choosing to stop using government power in the service of Microsoft’s business model, but how will we replace Microsoft? We know from the Microsoft trial that companies like Netscape, Sun, and Apple have tried and failed to reduce the world's dependence on Windows, but what would have happened if they had succeeded? The next industry giant might simply continue and expand upon Microsoft's business model, just as Microsoft did when IBM's mistakes handed Microsoft the leadership of the personal computer era. When Netscape produced the leading web browser, it tried to become the Web's standard setter, adding proprietary features to its version of the Hypertext Markup Language, and attempting to become a platform provider by positioning Netscape Navigator as the foundation for cross-platform applications. Had it succeeded, would Netscape have hesitated to pursue higher profits by undermining the Web's openness? Sun, having had some success with the Java language, continues to refuse to loosen its control over Java, disappointing the people who expected it to provide a more open alternative to Microsoft technology . The simple fact is that corporate executives do not hold their jobs by leaving money on the table. If my explanation of the rich-get-richer logic of the software industry is correct, we can expect similar tactics from any corporate successor to the throne that Microsoft took from IBM.

The alternative to software monopolies is not a regulated or benevolent software monopoly, or the components of a dismembered monopoly. The real alternative is open-source software. To understand what open-source software is, and why it's such a radical - and necessary -

alternative to proprietary software, we must move beyond the idea that software is a manufactured product in a colorful box to the understanding that software is information that describes methods for solving problems. The main difference between these two viewpoints is in their respective assumptions about the possibility of looking into software's inner workings.

A user with only a machine-code program at his disposal has no way to change what the program does or how well it works or what kind of hardware it works with. The program's source code could be debugged, given additional features, or adapted to new purposes if the user had it, but in the traditional closed-source software business, the source code is its developers' trade secret. This is fine for software companies, which need to sell an unending series of upgrades to preserve their revenue streams, but not so good for software users.

Computer programmers are also users of software. At work or school they use operating systems and programming tools - often "closed-source" tools provided only in machine code form by Microsoft or another vendor. For other purposes, they typically use the same software that non-programmers use. Most of them work not for companies that sell shrink-wrapped software products, but for organizations - manufacturers, banks, science labs, and so forth - that use software as a means to their main ends. They recognize the various strategies by which companies like Microsoft restrict their options, but unlike other software users, they have the means to avoid this manipulation: they can write alternative software. Of course, no single programmer can write all the software he would need to be free of Microsoft. But thousands of Internet-connected programmers, making their work available to each other as source code, *can* produce the hundreds of programs they need. The value of their work is by no means available only to other programmers. Even the busiest open-source programmers have time to work on only a tiny fraction of the programs they use, so, like any other users, they seek out software that doesn't require additional programming to be made useful. The reputations of open-source programmers and projects depend upon the usability of their work.

To many observers familiar only with the concept of software as a manufactured product, open-source software seems to be something vaguely communistic that can't possibly work. However, if we view software as information, nothing is simpler: software persists and evolves as it is being used, exactly as scientific knowledge increases and becomes more reliable by passing among generations of scientists. The open-source process of software development leads to many dead-ends - projects that are never completed, bad software that never becomes usable, and redundant software that never finds users - but the best software becomes still better as programmers and users are drawn to the best projects. This is like the functioning of a healthy conventional marketplace, and in its diversity it resembles the more competitive PC software market of the early 1980s more closely than either resembles today's software industry.

Neither the existence of open-source software nor the demolition of software patents and EULAs can remove the need for a software industry, since nothing works as well as commerce to align the interests of producers with those of consumers. However, a software industry built around a pool of freely-available open-source software would be quite different from one based on monopolies protected by "intellectual property" law. It would be a market for the programming services required to tailor generic free software to users' specifications, and for low-cost convenience products that add value to free software by making it easier to select, install, and use. The value to investors of a shift to such an industry would be far less than the market capitalization of today's software industry, but the value provided to software users worldwide would be immense.

Could Microsoft and its allies succeed in extinguishing the possibilities created by open-source software? In the long run, probably not. But in the next few years, the primary techniques for slowing the acceptance of open-source software will depend on exploitation of “intellectual property” law. Microsoft must seek to make the Linux operating system, the Apache web server, the Mozilla web browser, and other significant open-source software products incompatible with its own mainstream products, and this will depend upon using a mix of patents and the trade secrets hidden within copyrighted closed-source programs to preserve a realm of information to which access must be purchased.

The greatest potential for abuse exists at the point of convergence between the technology industry and the digital content distributors. Laws like the Digital Millennium Copyright Act, which forbids programmers even to talk about means for accessing data to which its proprietors may wish to control access, are powerful weapons for enforcing incompatibility. Under the DMCA, no open-source software can be used legally in the United States to play DVDs, because distribution of the information embodied in that software is viewed as contributory copyright infringement, despite the fact that one can copy a DVD without decoding it or play a DVD without copying it.

Only the mutual suspicion between the computing and content industries has (so far) prevented the U.S. Congress from passing laws requiring copyright enforcement technology to be built into all "interactive digital devices." Two recent bills proposed by Senator Fritz Hollings and supported by the Motion Picture Association of America would have done exactly that. Any “copyright enforcement” scheme requiring cooperation from a computer’s operating system would effectively outlaw open-source operating systems, since the legally-required features put in by one programmer could be removed by another. In the context of the “War on Terror,” future legislation might demand that data storage and communications software be made surveillance-friendly; open-source programs that could be modified to lock out the FBI and NSA {and, probably more to the point, the movie, music, and software trade organizations) could then be banned as terrorist tools. The reaction to the Hollings bills suggests that the limits on how far such measures will go are defined less by concerns about individuals' rights than by the technology companies' unwillingness to use "digital rights management" tools that they don't control.

Conclusion.

It is one thing for "Authors and Inventors" to license their work to corporations producing goods for sale, and another for corporations to license wholly intangible goods to end users. Because the supply of any pure information product is infinite, a proprietor's ability to monetize the intangible depends entirely upon a government's enforcement of his monopoly. To preserve monopolies like those the software and entertainment industries hope to protect, a government must do violence to conventional property rights and the right to communicate both privately and in public. Therefore all such claims of monopoly rights, derived from the false theory of "intellectual property," should be discarded immediately. From where we stand now, this seems like a radical proposal, but that is simply a measure of how far the lawyers and lobbyists have moved the system from where logic and the U.S. Constitution might have taken it.