

217
76.9
M65
F67
1994

Computer Ethics
Cautionary Tales and Ethical Dilemmas in
Computing

Second Edition

Tom Forester and Perry Morrison

NOTICE: THIS MATERIAL MAY BE
PROTECTED BY COPYRIGHT LAW
(TITLE 17, U.S. CODE)

Also by Tom Forester

- The Microelectronics Revolution* (ed.) (1980)
- The Information Technology Revolution* (ed.) (1985)
- High-Tech Society: The Story of the Information Technology Revolution* (1987)
- The Materials Revolution* (ed.) (1988)
- Computers in the Human Context* (ed.) (1989)
- Silicon Samurai: How Japan Conquered the World's IT Industry* (1993)

© 1994
The MIT Press
Cambridge, Massachusetts
London, England

strange circumstances, and in 1992 the U.S. House of Representatives Judiciary Committee voted to hold an independent investigation into the ten-year-old affair.⁸

In the Middle East, Latin America, and Eastern Europe, software piracy is part of the way of life. Lotus claims to lose as much as \$25 million annually in foregone revenue as a result of rampant copying in the Middle East alone, while in Latin America pirate programs are openly on sale in markets from Peru to Paraguay. In communist Eastern Europe, virtually all software was illegal as a result of Western trade bans, state restrictions on usage, and lack of money. But in the wake of the fall of the Berlin Wall, IBM offered an amnesty to software pirates in the form of a modest fee to legalize all their pirate programs and an exchange service that enabled users to trade in their pirate programs for legitimate copies purchased at a discount. In Australia, an antipiracy campaign by the Business Software Association of Australia (BSAA) began to bear fruit with successful raids on a Sydney-based dealer, Computer Exchange Corporation, and transport giant Mayne Nickless. In a later court settlement, Mayne Nickless agreed to pay sums of money to Lotus and WordPerfect Pacific.⁹

But in Asia copying is king. Most estimates suggest that about 90 percent of all software in China, Taiwan, Thailand, India, and Pakistan has been copied. In Hong Kong, about seven or eight copies of well-known packages are thought to exist for every legitimate copy sold, while the counterfeiting of goods of all kinds is estimated to be worth \$1 billion a year to the Singapore economy alone. It is common practice in Asia to purchase a computer complete with a variety of bootlegged programs already installed on the hard disk. Asian companies routinely buy one or two legitimate copies of a piece of software and duplicate hundreds of copies for their employees. Stolen software sells in street markets for a fraction of the normal retail price. Very often bootlegged versions of new packages are available before the packages have been officially released in the Asian market. Most people in Asia see copying software as a legitimate way to cut costs, and antipiracy laws are almost nonexistent. Where such laws do exist, they are rarely enforced and fines are usually minimal.

In China, copying of all Western goods, including software, has been condoned for years by the communist government. The BSA estimates

that U.S. software companies lose \$400 million a year because of Chinese software piracy. In a recent case, thousands of fake holograms on counterfeit Microsoft software being sold from Taiwan were traced to a Chinese government-owned factory in the special Shenzhen economic zone near Hong Kong. In 1991 China announced that it would join the Berne Convention, the international pact governing copyright protection, and in 1992 China agreed to U.S. demands that it outlaw the theft of software when it amended its copyright law. (As a direct result, Microsoft entered the Chinese market for the first time.) But doubts remained about what precisely both moves would mean in real terms.

If mainland China is tops for the quantity of software copied, Taiwan is probably tops for quality. Long known as the counterfeiting capital of the world, Taiwan's capital of Taipei is home to master forgers who apparently are able to produce flawless copies of Western computer manuals, diskettes, packaging, and even licensing agreements that have fooled all but expert company investigators. Taiwan runs a huge balance of payments surplus with the rest of the world—much of it due to piracy—and seems unlikely to take action to curb counterfeiting when it knows that the results are so rewarding. Meanwhile, in Thailand, where an estimated 97 percent of software has been copied, a recent Association of Thai Computer Industry survey put personal computer hardware sales in 1992 at \$284 million and software sales at a revealing \$9.8 million only.¹⁰

Revenge of the Nerds? Intellectual Property Rights and the Law

The idea of intellectual property rights has been around since the Middle Ages, and current forms of legal protection have evolved over subsequent centuries. According to the United Nations' Patent Office, the World Intellectual Property Organization (WIPO), intellectual property is defined as the rights to, among other things, the results of intellectual activity in the industrial, scientific, literary, or artistic fields. Generally speaking, copyright law has traditionally protected forms of literary expression, patent law has protected mechanical inventions, and contract law has covered trade secrets. Trademarks also are protected by law.

But computer software is a wholly new kind of entity that presents major new challenges for the law. Chief among these are how we define

ownership of this form of intellectual property and how the rights of ownership can and should be protected. Current laws are outdated and confusing: we are not sure whether copyright, patents, or trade secrets apply or should apply to this strange new thing called software. As Pamela Samuelson has argued, digital media such as software have several distinct qualities, such as compactness, plasticity, ease of replication, ease of transmission, and multiple uses. All these properties mean that software presents a challenge to the doctrines of existing intellectual property systems.¹¹ And as the confusion continues, the gap between legal precedent and everyday behavior on the part of computer professionals and users grows still wider.

Anne W. Branscomb has argued that what suited the age of print and mechanical inventions is proving inadequate in the age of the computer program, the expert system, and the distributed database. Copyright, patents, and contract law worked relatively well in the industrial era, but recently three factors have eroded the effectiveness of these traditional protection mechanisms. First, the development of new information and communications technologies has blurred the boundaries between media. Second, intellectual assets have become increasingly abstract and intangible. Third, the globalization of the world economy has multiplied both the incentives for international violations of intellectual property rights and the economic harm of such violations. Fourth, privatization and the growing trend toward using market mechanisms to gather and disseminate information have disrupted the traditional public infrastructure for sharing intellectual assets. For example, in the United States, Bell Laboratories, federal government agencies, and universities are now turning to patent rights and copyright royalties to recoup their research and development costs. Thus, she says, "at the very moment when information is becoming a valuable commodity, protecting the economic value of intellectual assets is proving more difficult."¹²

U.S. companies filed more than 5,700 intellectual property lawsuits in 1990 (up from 3,800 in 1980). In the same year, no less than 175,000 patents were filed in the United States (up 39 percent over 1985) and a staggering 643,000 copyrights were registered (compared with 401,000 registered over a five-year period in the 1970s). In Japan, the number of patent applications also doubled between 1980 and 1988 (and, incidentally, Texas Instruments was finally awarded a patent on the integrated

circuit in Japan—twenty-nine years after it had applied). Protecting intellectual property has become big business for law firms. Some New York law firms now specialize entirely in patent, copyright, and trademark work, while others have employed many more intellectual property experts. Computer companies also retain patent attorneys; IBM tops the list with 140. Some companies, such as Refac Technology, have come into existence purely to pursue lawsuits on behalf of customers or lawsuits concerning rights to technologies they have purchased from their inventors.¹³

One potentially worrying new trend is that big IT companies now tend to see the law as a way of increasing profits rather than as a defensive move to safeguard technology. For example, IBM and Texas Instruments have collected hundreds of millions of dollars from other companies (overwhelmingly Japanese firms, such as Hitachi and Fujitsu) that clearly have infringed their patents and copyrights over the years. These settlements may seem like justice, but some people, such as Mitch Kapor, the former boss of Lotus, have recently warned that litigation is now being used too often as a business tactic rather than as a move of last resort. This trend effectively could mean that wealth in the IT industry in the future will go more to the talkers and arguers than to the innovators and doers—which cannot be a good thing for the long-term interests of the industry.¹⁴

Among the thousands of lawsuits in the United States over the past decade, the cases involving Lotus Development versus other companies and Apple Computer versus others have been among the most interesting and significant. For example, in the mid 1980s Lotus sued two smaller companies, Paperback Software and Mosaic Software, for copying the "look and feel" of 1-2-3 (Mosaic's program was unashamedly called "Twin"). In turn, Lotus was sued for \$100 million by the Software Arts Products Corporation (SAPC), developers of the original VisiCalc program, which claimed that Lotus had copied many of the commands and keystrokes as well as the screen displays of VisiCalc in 1-2-3. SAPC claimed that Lotus founder Mitch Kapor "misappropriated" copyrighted and confidential aspects of the VisiCalc program while he was an employee of the exclusive marketing agent for VisiCalc. SAPC further alleged that later, as a product tester for an advanced version of VisiCalc, Kapor "had access to copyrighted and confidential aspects" of the pro-

gram. He “deliberately sought to make the 1-2-3 program look and feel like VisiCalc.”¹⁵

SAPC lost, but Lotus won against Paperback in 1990 when U.S. Federal District Court Judge Robert Keeton found that Paperback had unlawfully copied the 1-2-3 method of invoking commands using a command line interface. Days after its victory over Adam Osborne’s Paperback, Lotus filed suit against Philippe Kahn’s Borland for infringing its copyright on 1-2-3 in Borland’s Quattro program. Later, in 1992, Judge Robert Keeton, sitting in Boston, Massachusetts, again found in favor of Lotus by declaring that Quattro had infringed Lotus copyrights. But he also ruled that a further jury trial would be necessary to assess any monetary damages that might be due to Lotus.¹⁶

Although Lotus (rightly or wrongly) won its major cases, Apple Computer has been less successful in the courts. Apple sued both Microsoft and Hewlett-Packard to prevent them from using a Macintosh-style user interface in their products, although Apple had allegedly borrowed the mouse-and-icons concept from Xerox’s Star user interface in the first place. Apple appeared to win a first-round court victory against Microsoft early in 1989, despite Microsoft’s attempts to put a brave face on the judgment. Some industry observers warned that a clear victory for Apple in later rounds would put Apple in a commanding position in the marketplace and would drastically reduce both competition and innovation. But this didn’t happen. Although U.S. Federal Court Judge Vaughn Walker in San Francisco gave further hope to Apple in some minor rulings in 1991, Apple’s main claim that Microsoft had copied the look and feel of the Macintosh interface in its Windows software came crashing down in April 1992 when Judge Walker ruled in favor of Microsoft and rejected Apple’s claim for \$5.5 billion in damages. In some ways, this judgment appeared to contradict aspects of Judge Keeton’s rulings in the Lotus cases.¹⁷

Meanwhile, Xerox’s somewhat belated attempts to gain compensation from Apple for allegedly borrowing the look and feel of Macintosh from Xerox’s Star user interface also failed in the courts. Although Xerox appeared to have a strong case for \$150 million in damages based on the circumstances at the time (such as Steve Jobs’s visit to Xerox’s Palo Alto Research Center and his subsequent hiring of Xerox personnel), Apple was able to convince the same Judge Walker that the company

had done so much work itself on the Macintosh interface that it was substantially different from the Star interface. In addition, the fact that Xerox had waited many years before claiming it had been badly wronged counted against the company in court. Ironically, while Apple was using the “substantially different” argument against Xerox, in the Microsoft case Apple appeared to be arguing the opposite—that Windows was not substantially different from Macintosh.¹⁸

Many other intellectual property cases, some more significant than others, have gone through the U.S. courts in recent years. In 1988, for example, Ashton-Tate, then the third-largest producer of PC software in the world, launched a suit against two smaller companies, Fox Software and Santa Cruz Operation, for violating copyright laws supposedly protecting its best-selling dBaseIII database management and development program. Announcing the suit, Ashton-Tate chairman and chief executive Edward Esber said, “The issue is simple: a company like ours spends hundreds of millions of dollars making a brand name and a family of products, and we intend to protect our rights.”¹⁹

In 1991 Wang Laboratories had a major win in its patent infringement suit against Japan’s Toshiba and NEC, which both were found to have blatantly copied Wang’s single in-line memory module (SIMM) device. Wang followed up with a further suit against NMB and Mitsubishi in 1992. The Japanese firm Hitachi was forced to pay Motorola for copying its popular 68030 microprocessor, and Honeywell collected \$96 million from Minolta and \$45 million from Nikon in 1992 after the two Japanese camera companies were found guilty of stealing Honeywell’s automatic focusing technology in the mid 1980s. Honeywell was also suing Konica, Kyocera, Canon, Matsushita, Premier, and Kodak for the same offense. All of this activity followed U.S. inventor Clay Jacobson’s claim that Kawasaki had stolen his designs for the jet-ski.²⁰

It Looks and Feels Like the Law Is a Mess

As the previous anecdotes illustrate, the current legal position on software in the United States is very confusing, partly as a result of the seemingly contradictory judgments handed down in recent cases and partly because the U.S. legislature has never made up its collective mind about how to cope with this new thing called software.

The U.S. Copyright Office tentatively began accepting computer programs for registration back in 1964, but for many years the computer industry relied primarily on trade secrets regulations to protect its software. It was not until 1980 that the Copyright Act of 1976 was amended to include software in the form of the Computer Software Copyright Act of 1980. Under this legislation, programs are considered copyrightable as “literary works.” Meanwhile, the U.S. Patent Office decided that most computer programs were collections of algorithms (mathematical formulae designed to carry out a specific task or to solve a particular problem) and thus, like other mathematical equations, were excluded from patent protection. Beginning in 1980, the U.S. courts began to steadily extend copyright protection for software—extending it, first, from embracing a program’s source code to include the object code. Later, the logic and sequence of the program were also included.

By far the most significant case of this period was *Whelan v. Jaslow*. In 1986, the Third Circuit Court of Appeals (which covers New Jersey, Pennsylvania, and Delaware) found in favor of the plaintiff, Elaine Whelan, by declaring that copyright protection included the basic structure of a program, its lines of written code, and the nonliteral aspects of a program, such as the screen design and the commands. This decision helped set the scene for the “look and feel” cases launched in the period 1987–1989, which, as we have seen, resulted in victories for Lotus and defeats for Apple and Xerox. But on 23 June 1982, the Second Circuit Court of Appeals in New York handed down a judgment in *Computer Associates v. Altai* that declared that the basic structure of a program is not copyrightable. As such, this ruling marked a considerable retreat from *Whelan v. Jaslow* and the prevailing tendency to broaden rather than narrow the coverage of copyright law.²¹

At the same time, the legal waters have been further muddied by the growing tendency of the U.S. Patent Office to issue patents on software, including quite small bits of software. Beginning with a 1983 memorandum from the Patent & Trademark office, computer companies rushed to stake a claim to patents on all kinds of software devices—some innovative and significant, some quite commonplace and trivial. For example, Teknowledge Inc. received patents on two new artificial intelligence products in 1986, while in 1989 Quarterdeck Office Systems was awarded a patent for Desqview, a Windows-type multitasking operating

environment. The patent had been applied for in 1984, the same year that Desqview and Microsoft’s Windows package first came on the market. Quarterdeck’s persistence was seen as an attempt to get back at Microsoft and to help recoup their \$6 million start-up costs, but some industry observers, such as office systems analyst Andrew Seybold, were already warning that such decisions would do great damage to the software industry in the long run.²²

Cases cited by the critics of software patents include the award to Hayes Microcomputer, the modem maker, of a patent on a program that simply switches a modem from transmit mode to receive mode. Hayes apparently now has the exclusive rights to any program that performs the same function until the year 2002. In another case, Merrill Lynch was awarded a patent on their cash management accounting system, a procedure for moving investment funds between different types of accounts. A federal district court upheld the patentability of the system, even though it accepted that the system was essentially a method of doing business. If the transactions were executable using pencil and paper, the system would not be patentable, but because it made use of a computer, the patent was upheld. The U.S. Patent Office has even awarded patents for other familiar processes, such as generating footnotes and comparing documents. And in 1992 the Patent Office awarded a patent to a company called Arrhythmia Research for the calculations its software performed to analyze electrocardiograms. This award effectively knocked down the final barriers to patenting almost any kind of software. Software developers now live in fear that they will be caught for accidentally using a process that is already patented.²³

Regardless of widespread industry concerns, the U.S. government, through its Patent Office, is continuing to grant ever-increasing numbers of patents on software. The number of computer-related patent applications doubled between 1987 and 1990. A leading critic of this policy, attorney Brian Kahin, argues that the Patent Office is ill-equipped to make judgments on software patent applications, largely because it does not have enough patent examiners with a deep knowledge of computing. He says that the quality of many of the patents awarded has concerned lawyers and that the spread of patenting to software will place an intolerable cost burden on the software industry. This burden will hit small companies hardest and will also provide a field day for patent

lawyers. Most worrying of all, the growth of patenting will slow the pace of innovation in the industry by demoralizing developers. "The software industry was not broke," writes Kahin, "but it is in the process of being 'fixed.'"²⁴

What we have, then, is a situation in the United States in which three kinds of law may or may not apply to programs or even to different aspects of a program. In short, the law on intellectual property as it applies to computer software is in a mess. The irony is that while the scope for applying copyright law to software appears to have been narrowed as a result of recent court decisions, the much stronger patent law is being extended to more and more areas. It seems that the basic issues will not be sufficiently resolved until the U.S. Supreme Court makes a final ruling on the matter.

Much the same sort of legal confusion exists in most other leading nations around the world, which are still struggling with their laws in an attempt to take account of the new phenomenon of computer software. In 1985 a joint UNESCO-WIPO conference in Geneva, Switzerland, called to discuss the problem of software piracy, broke up in confusion, with the international delegates coming to no firm conclusions about how to protect programs. Since then, many countries have gone their own way with amendments to their copyright laws. For example, the United Kingdom passed the massive Copyright, Designs and Patents Act in 1988. Germany, France, Spain, Denmark, and Canada have also extended copyright protection to software in recent years.

Australia extended copyright protection to programs as early as 1984, following the case of *Apple Computer v. Wombat*, in which the Australian company was accused of pirating the operating systems used in Apple's chips. But in 1986, in *Apple Computer v. Computer Edge*, the Australian High Court held that although source code was deemed to be a literary work and therefore covered by copyright, this protection did not extend to the object code, which was not covered. This ruling created considerable dissatisfaction in the industry, and so a new review of copyright law as it applies to software was launched by the federal Attorney-General's office in 1988 and continued through 1992. In what became the most celebrated, and drawn-out Australian software copyright case, *Autodesk v. Autokey Lock*, the Australian High Court ruled in February 1992 in favor of Autodesk, thus effectively extending the

law of copyright in Australia. The judgment had its supporters and its critics: the latter were particularly vocal in a country where consumers are forced to pay 50–70 percent more for software packages than consumers pay in the United States.²⁵

Perhaps the biggest controversy over extending copyright to software occurred in Europe, where the European Community (EC) in Brussels published in 1989 a tough directive that called for complete copyright protection for all software. Critics immediately accused the EC of taking its advice almost wholly from a group of large U.S.-owned computer firms, including IBM, Digital Equipment Corporation, Apple, and Microsoft, which were represented in Brussels by SAGE (the Software Action Group Europe). Pitted against SAGE was ECIS (the European Committee for Interoperable Systems), formed by Italy's Olivetti, France's Bull, Finland's Nokia, and most other nonaligned European computer companies. A fierce battle for the hearts and minds of EC commissioners and parliamentarians ensued, the upshot being that in May 1991 the EC finally adopted a compromise software copyright directive that was much softer than the original proposal. Both sides declared themselves satisfied with the outcome, although most observers seemed to agree that the final directive had moved more in the direction of ECIS than SAGE. But the big worry was that the onset of patenting in the United States and elsewhere in the global marketplace would increase the dominance of large, established computer firms.²⁶

Software Piracy and Industry Progress

What, then, is the answer to the software piracy problem? The central dilemma facing lawmakers, the IT industry, and society at large is how to adequately reward innovation without stifling the creativity that has moved the IT revolution forward. This issue is fundamental to the future of the IT industry.

Without adequate legal protection, innovatory individuals and companies might wonder whether the meager rewards for their efforts really justify the time and money expended on original R&D. On the other hand, intellectual property owners might try to stake too large a claim for their innovations in order to squelch new ideas and get the jump on their competitors. This strategy could strengthen the hand of large,

established firms over small, entrepreneurial firms, which have been the traditional innovators of the industry. The key question is whether the developmental work justifies the influence innovators may gain over both users and competitors. There is a clear need to strike a balance between the competing interests of these groups as we tread the fine line between piracy and progress.²⁷

Mitch Kapor, founder and former chairman of Lotus and now a director of the Electronic Frontier Foundation, argues that any ban on borrowing techniques and ideas from other software developers will strangle creativity in the software business. Tightening copyright law too much and granting patents on software will only serve to increase corporate profits. Many other programmers also argue that software developers should be free to “decompile” the object code of other people’s programs—that is, to work backward to the original source—because it is the only way to pick up new programming ideas. Without such freedom, they say, progress in the industry would grind to a halt. In this respect, company programmers often find themselves at odds with company managers and their legal advisers, who are more concerned with profits than technical progress. This difference of focus is also reflected in the split between large companies and small companies on this issue.²⁸

What laws, if any, should apply to software? Deborah Johnson and John Snapper rule out trade secrets as a means of protecting software on the grounds that they could apply only to research done under extreme security conditions. The publication of technical results would be restricted and employees would find that they could not change jobs with ease or freely discuss their work with other researchers. These drawbacks, they say, might point in the direction of patent or copyright protection. On the other hand, Paul Marette has argued that contractual relationships rather than copyright law may prove to be more important in the future, at least in the world of electronic publishing, and that we should think in terms of developing an entirely new field of informatics law.²⁹

Anne W. Branscomb has argued that because information technology makes a product’s form easy to separate from the intellectual assets that go into it, copyright law, with its focus on the expression of an idea rather than on the idea itself, is inappropriate for protecting what is

valuable in the new kinds of intellectual property. This view is generally supported by Stuart Hemphill and Paul Marette. Branscomb suggests that a modified form of patent rights—or “soft patents”—with registration procedures, monopoly time limits, and rules for licensing shaped to the realities of the IT industry, may be the answer.³⁰

Pamela Samuelson has made a powerful case for using patent law to protect innovation in the computer industry. She says that *Whelan v. Jaslow* (which, as we saw, favored protecting software by copyright law) was particularly bad news for innovators, and she rejects attempts to modify copyright law or to create new laws treating software as a special case. The existing system of patent law, she says, is still the best vehicle for protecting software. Samuelson argues, in particular, that the look and feel of user interfaces should not be protected by copyright law but by patent law primarily because “it is more consistent with legal tradition.”³¹

In complete contrast to those advocating the use of laws and yet more laws to protect software, maverick Massachusetts Institute of Technology programmer Richard Stallman argues that all software should be freely available to be copied. Stallman, who founded the Free Software Foundation in 1985, puts forward a philosophical argument that all information should be free and that “the full fruits of information technology can be realized only when everyone has the freedom and ability to copy and change programs.” Stallman argues that programmers should have access to the source code of programs so that they might tinker with it and improve it. Proprietary software obstructs IT progress, he says, and companies should not be allowed to keep their source code secret. He says that too many software developers are merely motivated by greed—unlike Stallman, who distributes software, such as a text editor called EMACS, free. But Stallman is fortunate in that he is financially supported in his work by groups such as the McArthur Foundation; other, more ordinary programmers have to eat.³²

No doubt this important but complex debate will continue, and while it does, the gap between the law and individual everyday behavior in the real world will remain large and probably grow larger. When asked about their personal copying policy, individuals locate themselves somewhere on a continuum of views ranging from a hard “never copy under any circumstances” position through a soft “sometimes copy in certain cir-

cumstances" position to a completely open "copy everything" position of the kind favored by Stallman. Because copying software is so easy and so hard to detect, the issue will, in the final analysis, be determined largely by social attitudes and individual consciences. In this respect, it probably won't matter what new laws are passed. As Anne W. Bronscomb has put it, "Although disputes about technology and intellectual property are usually cast in narrow legal terms, they are intimately related to public attitudes. Realistic legal rules depend upon a social consensus about what kind of behaviour is acceptable and what is not."³³ This consensus on software theft seems to be missing at present.

Busting the Pirates

Apart from legal remedies, various technical devices and administrative schemes have been proposed as possible solutions to the problem of software theft. The technical devices, sometimes called "dongles," take the form of programmed chips or electronic locks that are physically attached to a computer; in theory, only those in possession of the correct code or key are able to gain access to the protected program. But clever fraudsters have found ways to reengineer the copy locks and have sold large numbers of lock-breaking devices. Some software companies also have supplied their disks with copy protection devices that allow the disks to be copied only once. But software pirates have found ways of opening and rewriting the files to allow multiple copies to be made.

Among other proposals to cope with the problem of piracy are site-based or company-based licensing schemes, which channel royalties back to the program originators. Another suggestion is to further popularize the concept of shareware, or honorware, by which users are invited to send a donation to the authors named at the beginning of the program. Shareware has become fairly popular in recent years, though users often complain that the range of quality software available as shareware is somewhat limited. The Copyright Clearance Center of Salem, Massachusetts, which collects royalties for magazine and book publishers when their copyrighted material is duplicated by large corporations, has suggested a similar scheme for computer software. Each time a copy of a program is made, the company or the institution would send a royalty

payment back to the center, which would collect a commission and pass the rest on to the software supplier.

Despite these suggestions and the efforts of organizations such as the SPA and the BSA, some software companies apparently have thrown in the towel on software copying. In particular, those pursuing the lucrative business computing market say that abandoning protection is the best protection from competitor companies. In 1985 MicroPro International gave up all pretense of protecting its programs, and Microsoft did the same with Word. In 1986 Software Publishing Corporation followed suit, saying, "If you want to get into the corporate market, they won't even look at you if you're copy-protected." Ashton-Tate decided to do likewise, claiming that new legislation around the world now strengthened the hands of software suppliers. The Washington-based Association of Data and Processing Services Organizations (ADAPSO) announced that it was dropping its plans to get software publishers to adopt a voluntary protection standard because it ran against current industry trends. And the small specialist Californian software company Cygnus Support announced that it was giving away its products for nothing, with no restrictions on their reproduction. Although its products (mostly tools for programmers) were free, Cygnus charged for tailoring its programs to customer's needs.³⁴

Nevertheless, the battle against piracy seems to be meeting with more success in the 1990s. In the United States, Congress passed the Software Copyright Protection Act in late 1992. The act defines commercial software pirates as individuals who willfully copy software for commercial advantage or for private financial gain. Prison terms of up to five years and fines of up to \$250,000 can be imposed on people convicted of making at least ten illegal copies of a software program held under copyright or any combination of programs worth more than \$2,500. The act is thus targeted at the professional pirates and users who make multiple copies for employees or friends.³⁵

The international battle against the pirates has resulted in a spectacular series of BSA-led raids in recent years in Taiwan, Singapore, Hong Kong, and other Asian nations. In Taiwan, raids in January 1990 netted more than 5,000 counterfeit packages of MS-DOS; 6,000 counterfeit MS-DOS manuals in English, French, and German; and 12,500 disks with bogus