

# UIC John Marshall Journal of Information Technology & Privacy Law

---

Volume 4  
Issue 2 *Computer/Law Journal - Fall 1983*

Article 6

---

Fall 1983

## Copyrighting Object Code: Applying Old Legal Tools to New Technologies, 4 Computer L.J. 421 (1983)

Benjamin S. Kaufman

Follow this and additional works at: <https://repository.law.uic.edu/jitpl>



Part of the [Computer Law Commons](#), [Internet Law Commons](#), [Privacy Law Commons](#), and the [Science and Technology Law Commons](#)

---

### Recommended Citation

Benjamin S. Kaufman, Copyrighting Object Code: Applying Old Legal Tools to New Technologies, 4 Computer L.J. 421 (1983)

<https://repository.law.uic.edu/jitpl/vol4/iss2/6>

This Comments is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in UIC John Marshall Journal of Information Technology & Privacy Law by an authorized administrator of UIC Law Open Access Repository. For more information, please contact [repository@jmls.edu](mailto:repository@jmls.edu).

# COPYRIGHTING OBJECT CODE: APPLYING OLD LEGAL TOOLS TO NEW TECHNOLOGIES\*

## I. INTRODUCTION

Since 1976, when the first microcomputer<sup>1</sup> was marketed,<sup>2</sup> the industry has increased to such a point that over 1.44 million units were sold worldwide in 1982.<sup>3</sup> Predictions are that in the next five years over twenty million additional personal computers will be sold.<sup>4</sup> This tremendous growth has led to an equivalent explosion in the marketing of programs for these computers.<sup>5</sup> Many difficult legal questions have been raised by the rapid expansion of this new technology.

Most computer programs commercially available are in a language known as object code, the only language that the computer understands. Few people can read object code. Some programs in object code are imprinted on computer circuits, in which case they are given the name firmware. Recently,<sup>6</sup> federal courts have had to deal with the question of whether such programs were protected by the federal copyright statute.<sup>7</sup> So far no clear consensus has emerged.<sup>8</sup>

---

\* A version of this Note won the 1983 Nathan Burkan Memorial Competition at the University of Southern California Law Center.

1. For the purposes of this Note, a microcomputer is a computer with an eight or sixteen bit processor which uses only one terminal.

2. The first personal computer was the Altair, which was sold through the mail, unassembled.

3. More than one million were sold in the United States alone. Blundell, *Personal Computers in the Eighties*, *BYTE*, Jan. 1983, at 166, 167.

4. *Id.* at 168.

5. In 1981, it was estimated that over 15,000 computer programs were written each day in the United States. Schmidt, *Legal Proprietary Interests in Computer Programs: The American Experience*, 21 *JURIMETRICS J.* 345 (1981). Since then, the number has undoubtedly increased.

6. The first such case was *Data Cash Sys. v. JS&A Group*, 480 F. Supp. 1063 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980).

7. 17 U.S.C. (1982).

8. This has also been true among commentators. See, e.g., Lawlor, *A Proposal for Strong Protection of Computer Programs Under the Copyright Law*, 20 *JURIMETRICS J.* 18 (1979); Stern, *Another Look at Copyright Protection of Software:*

This Note will examine the copyrightability, under federal law, of object code, both as software and as firmware, in programs whose purpose is the conveyance of information.<sup>9</sup> Further, this Note will address the question of whether there is some better system than copyright protection for protecting object code.

## II. DEFINITIONS

One of the major reasons that legislators and courts have had so much difficulty in finding a clear set of rationales for the rules that should apply to this area is that legal experts have been unable to understand the technical aspects of the problem. Those in the computer industry rarely bother to define the terms they use. Instead, they expect the novice to pick up their meanings by osmosis. Therefore, it is important to define certain technical terms, although they are in common use.

1. **Hardware:** The physical body of the computer, e.g., the keyboard, video monitor, printer, and central processing unit. Hardware may be patented if it meets the statutory criteria.<sup>10</sup>
2. **Software:** The instructions that make up the brain of the computer. This includes not only computer programs but also the documentation that comes with the programs.
3. **Firmware:** Software that is stored in hardware form. The term is generally applied to a small integrated circuit (IC) that has been imprinted with a program (i.e., software) and is wired into a computer.
4. **Program:** "A set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."<sup>11</sup> Programs may be stored on magnetic tape, magnetic disks, punched disks, punched cards, punched tape, or on ROM's. When programs are stored on ROM's they are known as firmware.
5. **Integrated circuits:** For the purposes of this Note, it is only necessary to discuss the two main types of integrated circuits in a computer, central processing units (CPUs) and memory circuits. The CPU does the actual manipulations that carry out the instructions of the program. There are several types of memory circuits, including:

---

*Did the 1980 Act Do Anything for Object Code?*, 3 *COMPUTER/L.J.* 1 (1981); Note, *Copyright Protection of Computer Object Code*, 96 *HARV. L. REV.* 1723 (1983).

9. Thus, the question of the copyrightability of firmware used for entertainment in video games will not be considered.

10. 35 U.S.C. §§ 100-104 (1982).

11. 17 U.S.C. § 101 (1982).

**Read Only Memory (ROM):** This is memory containing programs or data that the computer may only read, not add to or change. This memory is imprinted on a chip in an elaborate and expensive procedure that cannot be altered.

**Programmable Read Only Memory (PROM):** This is a chip that is manufactured blank and may be imprinted by a relatively easy process, i.e., a process that an average computer user can perform. Once imprinted, the information on the chip cannot be altered.

**Erasable Programmable Read Only Memory (EPROM):** This is a PROM that can be erased by exposure to intense ultra-violet light. Some EPROMs can be erased by exposure to an electric current. These are called EEPROMs.

**Random Access Memory (RAM):** The preceding types of memory are all non-volatile, i.e., they retain the information imprinted on them even when the computer's power source is turned off. RAM, however, loses whatever information is stored in it when the power is turned off. It is often referred to as "scratchpad" memory. The information stored in RAM can be changed easily whenever the user wishes.

6. **Object code:** This is a binary code, consisting of a series of zeros and ones. The CPU of a computer recognizes each statement in object code<sup>12</sup> and in response performs one simple operation. Thus, it takes hundreds of object code statements to execute even a very simple program. When computers were first invented, all programs were written in object code. However, within a few years assembly language programming replaced object code. Still, object code remains the only language that the CPU understands.<sup>13</sup> All other programming languages must be reduced to object code to be processed by the CPU. Very few, if any, programmers can read object code today.
7. **Assembly language:** The first language invented that allowed a computer programmer to program without having to remember exactly which series of zeros and ones did what. Assembly language is a coding system that records the instructions to the computer in a mnemonic style.<sup>14</sup> Assembly language is referred to as a "low-level" language because it is very close to object

---

12. For an eight bit computer, a statement in object code consists of eight zeros or ones, e.g., 10110111.

13. Several manufacturers are currently working on CPUs that would be able to execute directly programs written in a high-level language. Stryker, *The Next Generation of Microprocessor*, BYTE, Jan. 1983, at 128.

14. A command to add a number in memory location X to register R1 might be expressed as "ADD R1,X."

code. Each statement in assembly code corresponds to a single object code statement. A program called an assembler translates assembly code into object code. Many programmers still program directly in assembly language.

8. High-level language: The type of computer language most often used by programmers. Although some commentators have defined a "high-level language" as one that is close to ordinary English,<sup>15</sup> the term actually means that the language is removed from the machine-dependence imposed by assembly language programming. Thus, a function which takes many assembly language statements to order the CPU to perform may be programmable in a single high-level language statement. Examples of high level languages include FORTRAN, PL/1, BASIC, APL, and Pascal. Programs written in a high-level language are often called "source code" programs, but the latter term properly refers to the language the program was first written in (i.e., its source).
9. Compiler: A program that converts source code programming into object code. Many computers have a compiler built in as firmware. Often the conversion from a high-level language to object code is a two-step process. The middle level language produced is assembly code. Since this process is necessary for the CPU to execute the program, all programs in a high-level or assembly language must be converted to object code before they may be run. Thus, programs written in a high-level or assembly language take longer to run than programs written in object code.
10. Disk Operating System or Operating System (DOS): This is the program that provides the instructions necessary to control the operation between the disk drive (the piece of hardware that reads the disks on which the programs are recorded) and the computer. It tells the CPU, for instance, where to look on the disk for the program being sought. There are only a handful of popular DOS's. Programs written under one DOS will not run under another.
11. Application programs: These are programs that, unlike the DOS or the compiler, perform only one specific task. For example, a word processing program is only used to prepare written text, while a spread-sheet program is used to prepare and maintain records.
12. Operating programs: Programs that are designed to be used with all application programs to make the computer easier to

---

15. *E.g.*, Stern, *supra* note 8, at 2.

use. The operating program controls the interaction between the application program and the computer hardware. Virtually all computers come with some operating programs as firmware. The DOS is an example of an operating program. Each computer generally has its own operating programs. As with the DOS, application programs written to work with one computer's operating programs will not run on another computer.

These are not legal definitions, but the author's attempt to come to grips with what is essentially jargon. The fact that these terms have no clear definition poses a major problem in the case law.

### III. THE PROBLEM

The United States Constitution states: "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."<sup>16</sup> As Justice Reed stated in *Mazer v. Stein*:<sup>17</sup>

The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors in "Science and the Useful Arts." Sacrificial days devoted to such creative activities deserve rewards commensurate with the services rendered.<sup>18</sup>

The problem of pirating object code arises in two areas. First, there are those application programs that are written in object code since this allows them to be processed in less time, makes them harder to decipher, and allows them to run on more computers.<sup>19</sup> The second and more serious problem is the pirating of operating programs.

The actual cost of building a computer is small compared to its retail price.<sup>20</sup> The main expense for a computer manufacturer is research and development. Much of this expense goes into developing the operating programs. If these programs are not protected, a rival manufacturer can purchase a company's computer, copy all of the operating programs, and produce a competing computer that is identical in all significant respects, but may be sold for less because no

---

16. Art. I, § 8, cl. 8.

17. 347 U.S. 201 (1954).

18. 347 U.S. at 219.

19. A program written in a high-level language can run on a computer only if the user has a copy of the language's compiler. A program written in object code avoids this extra expense and thus has a greater potential market.

20. For example, a Zilog Z-80 CPU has a retail price under \$10.

research and development costs were incurred.<sup>21</sup>

The computer industry is software generated. A potential customer will first determine which software programs will best serve his needs and then purchase a computer that can run that software. If two identical computers can run the same programs, many buyers will buy the cheaper computer. Thus, allowing operating programs to be pirated will result in a tremendous disincentive to the development of any new computer system.<sup>22</sup>

There are, however, several difficulties in extending copyright protection to programs written in object code. Since object code is not readable by people, there is a question of whether such a program is a "writing" within the meaning of the Constitution. Also, as these programs are normally written in source code and then compiled by a separate program into object code, there is a question as to whether these programs are authored by a person, rather than by a machine. A further problem is whether operating programs are not better perceived as a machine part, rather than as a work of authorship.

A final problem is whether the copyright system should be extended at all to object code programs. The copyright system was created in a world where the only "writings" were the written word, maps and charts, and paintings. Since then the law has been extended to cover technological advances such as photographs, sound recordings, and telecommunications devices. With each of these technological advancements over the last two hundred years, the copyright law has been bent more and more out of shape. The language of the statute has left the area of common sense and descended into "terms of art."<sup>23</sup> At some point the language loses all semblance of meaning. Perhaps it is at the question of copyrighting object code that this point is reached.

#### IV. THE COPYRIGHT ACT

##### A. THE 1976 COPYRIGHT REVISION

The Copyright Act of 1976<sup>24</sup> (Act) completely revised the federal copyright system. Under the Act, copyright protection exists for "original works of authorship [including literary works] fixed in any

---

21. Nearly all personal computers use "off-the-shelf" hardware.

22. One expert has likened software to razor blades. "That is where the money is—not so much in making the razor, or hardware. People are making fortunes stealing the blades." N.Y. Times, Apr. 20, 1983, at D19, col. 3.

23. In revising the Copyright Act in 1976, Congress intentionally left many of these terms (e.g., "fair use") undefined.

24. Pub. L. No. 94-553, 90 Stat. 2541 (1976) (codified as 17 U.S.C.).

tangible medium of expression now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."<sup>25</sup> The definition of literary works includes works "expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as . . . tapes, disks, or cards, in which they are embodied."<sup>26</sup> The legislative history of the Act states that literary works include "computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves."<sup>27</sup>

Section 102(b) of the Act codifies this distinction.<sup>28</sup> The legislative history states that, with respect to computer programs, this section was intended to "make clear that the expression adopted by the programmer is the copyrightable element in a computer program."<sup>29</sup>

The Act allows copyright protection for pictorial, graphic, or sculptural works.<sup>30</sup> These works are only copyrightable to the extent that their design "incorporates pictorial, graphic, or sculptural features that can be identified separately from . . . the utilitarian aspects of the article."<sup>31</sup>

Section 117<sup>32</sup> of the Act was enacted to maintain the status quo with respect to the computer uses of copyrighted works.<sup>33</sup> It provided that, notwithstanding the other provisions of the Act, the owner of a copyright was not granted greater or lesser rights with respect to the use of the work in conjunction with a computer than the owner had under the law before the Act. However, section 117 did not apply to the copyrightability of computer programs.<sup>34</sup> This section was included to maintain the status quo until Congress could receive the final report of the National Commission on New Technological Uses of Copyrighted Works (CONTU), which Con-

---

25. 17 U.S.C. § 102(a) (1982).

26. 17 U.S.C. § 101 (1982).

27. H.R. REP. NO. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S. CODE CONG. & AD. NEWS 5659, 5667 [hereinafter cited as H.R. REP.].

28. "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b) (1982).

29. H.R. REP., *supra* note 27, at 57; 1976 U.S. CODE CONG. & AD. NEWS at 5670.

30. 17 U.S.C. § 102(a)(5) (1982).

31. 17 U.S.C. § 101 (1982).

32. In 1980, Congress amended § 117. Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015. See *infra* text accompanying note 40.

33. H.R. REP., *supra* note 27, at 116; 1976 U.S. CODE CONG. & AD. NEWS at 5731.

34. *Id.*



gress had set up in 1974 to study this problem and make recommendations.

This treatment of the scope of copyright protection for computer programs and the legislative history of the Act make it clear that Congress intended to include computer software within copyright protection.<sup>35</sup> The language of the Act does not distinguish between software and firmware.<sup>36</sup> The Act does state that works of authorship can be fixed "in any tangible medium of expression now known, or later developed."<sup>37</sup> Although this language is broad enough to cover programs imprinted (i.e., fixed) on a ROM, this presupposes that a computer program written in object code would be considered a "work of authorship."

While a program imprinted on a ROM may be described as a three dimensional work of art, it is unlikely that firmware would fall within the Act's allowance of copyrighting of sculptural works.<sup>38</sup> The art inherent in the design of the chip cannot in any sense be separated from the utilitarian function that it serves.<sup>39</sup>

#### B. THE COMPUTER SOFTWARE COPYRIGHT ACT OF 1980

Following the final report of CONTU, Congress passed an amendment<sup>40</sup> to the 1976 Copyright Act, which (1) amended section 101 of the Act to include a definition of a computer program,<sup>41</sup> and (2) amended section 117 to define the permissible uses of copyrighted works when used with a computer. The new section 117 provides that:

[I]t is not an infringement for the owner of a copy of a computer program to make . . . another copy or adaptation of that program provided: (1) That such a new copy or adaptation is created as an essential step in the utilization of the computer program . . . or (2) That such new copy or adaptation is for archival purposes only. . . .<sup>42</sup>

In listing only these two exceptions as permissible purposes in copying a computer program, Congress implicitly made any other

---

35. M. NIMMER, *CASES AND MATERIALS ON COPYRIGHT AND OTHER ASPECTS OF LAW PERTAINING TO LITERARY, MUSICAL AND ARTISTIC WORKS ILLUSTRATED* 39 (2d ed. 1979).

36. It is unlikely that the term "firmware" existed in 1976.

37. 17 U.S.C. § 102(a) (1982).

38. 17 U.S.C. § 102(a)(5) (1982).

39. *Mazer v. Stein*, 347 U.S. 201 (1954) (artistic articles are protected in form but not in mechanical or utilitarian aspects).

40. Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015.

41. "[A] set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1982).

42. 17 U.S.C. § 117 (1982).

copying of a program an infringement of copyright protection.<sup>43</sup>

## V. CASE LAW

### A. *WHITE-SMITH MUSIC CO. v. APOLLO CO.*<sup>44</sup>

In *White-Smith Music Co. v. Apollo Co.*, the Supreme Court held that a player-piano roll is not an infringing "copy" of a protected musical composition because a piano roll is not intelligible to a human being. After pointing out that the rolls are not intended to be read, as is sheet music, the Court stated that "these perforated rolls are parts of a machine, which, when duly applied and properly operated . . . produce musical tones . . ."<sup>45</sup>

Under this case, there is no doubt that object code and firmware would not be protected by federal copyright law since neither is intended to be read by a human being. This case has never been explicitly overruled by the Court but has fallen into disfavor in all areas except computer program cases.<sup>46</sup>

The 1976 Copyright Act states that a work can be copyrighted if it can be "perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."<sup>47</sup> This would seem to destroy the remnants of *White-Smith*, yet some courts have imposed a similar requirement,<sup>48</sup> without mentioning *White-Smith*.

### B. *DATA CASH SYSTEMS, INC. v. JS&A GROUP, INC.*<sup>49</sup>

In 1977, Data Cash Systems, a computer game manufacturer, introduced a computerized chess game called Compuchess. The program for the game was encoded as firmware. Thinking that it would be impossible to unload the chip (i.e., retrieve the object code from the IC), Data Cash did not attempt to copyright the program. JS&A Group's agent managed to unload the program from the chip and copied it for another chess game which JS&A marketed. Data Cash brought suit, claiming infringement of the copyright of their source

---

43. Section 106 of the Act states the rights granted to holders of a copyright, while §§ 107-118 express the limits on those rights.

44. 209 U.S. 1 (1908).

45. *Id.* at 18.

46. See *Data Cash Sys. v. JS&A Group*, 480 F. Supp. 1063 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980).

47. 17 U.S.C. § 102(a) (1982).

48. *E.g.*, *Apple Computer, Inc. v. Franklin Computer Corp.*, 545 F. Supp. 812, 825 (E.D. Pa. 1982), *rev'd*, No. 82-1582 (3d Cir. Aug. 30, 1983).

49. 480 F. Supp. 1063 (N.D. Ill. 1979), *aff'd on other grounds*, 628 F.2d 1038 (7th Cir. 1980). The case was decided under the Copyright Act of 1909. Because of the rapid advances in computer programs, virtually all commercially available personal computer programs are protected under the 1976 Act.

code program. The trial court judge held that reproduction of a ROM could not be an infringement of a copyright (an issue that neither party had briefed or argued) because a copy had to be "in a form which others can read."<sup>50</sup>

On appeal, the Seventh Circuit dismissed the trial judge's rationale in one line,<sup>51</sup> but affirmed his decision on the ground that, by publishing without notice, Data Cash had forfeited its right to copyright protection.<sup>52</sup>

Some commentators have taken the position that the appellate court's ruling implies that firmware can be copyrighted. Under this view, the court could not have reached the issue of forfeiture for lack of notice without first having found that, with the proper notice, the program would have been protected by the copyright statute. This is highly speculative reasoning. The court probably found that the notice issue was a simple means to decide the case, and ignored the copyrightability issue because neither party had argued it to the trial court.

C. *TANDY CORP. V. PERSONAL MICRO COMPUTERS, INC.*<sup>53</sup>

Tandy Corporation's TRS-80 personal computer uses an IC containing a compiler program in object code. Tandy had copyrighted the source code version of the program. The defendant unloaded the contents of the IC and copied it onto a chip which it installed in its own computer. On the defendant's motion to dismiss Tandy's claim for copyright infringement, the court held that the broad language of the Act brings firmware within copyright protection. Specifically, it held that (1) a computer program in object code is a "work of authorship,"<sup>54</sup> and (2) the silicon chip upon which the program is imprinted is a "tangible medium of expression," so that a program so fixed is subject to federal copyright law.

Unfortunately, the court here was not long on scholarship. The court dismissed the *Data Cash* case with little more than a "But see" notation.

D. *APPLE COMPUTER, INC. V. FORMULA INTERNATIONAL, INC.*<sup>55</sup>

In its continuing effort to protect its retail market from "copy-

---

50. Note the similarity between this standard and the *White-Smith* test, *supra* text accompanying note 43.

51. *Data Cash Sys., Inc. v. JS&A Group, Inc.*, 628 F.2d 1038, 1041 (7th Cir. 1980).

52. *Id.* at 1042.

53. 524 F. Supp. 171 (N.D. Cal. 1981).

54. *Id.* at 172.

55. 562 F. Supp. 775 (C.D. Cal. 1983).

cat" computers, Apple brought suit against Formula International for copyright and trademark violation. The defendant sold a computer kit, the Pineapple, which, when assembled, resulted in a computer with "uses and capacities very similar, if not identical," to the Apple II computer.<sup>56</sup> The court granted Apple's motion for a preliminary injunction against the continued sale of the Pineapple.

In dealing with the question of the copyrightability of the object code programs that the defendant had copied for use with its computer, the court found that these programs were copyrightable under the current copyright statute. The court stressed the breadth of the language of section 102(a) of the Act, and of the definition of a computer program in section 101 of the Act.<sup>57</sup> The court considered it clear that the purpose of section 102(b) is to distinguish copyrightability from patentability.<sup>58</sup>

The court further found that the majority of the CONTU members supported the copyrightability of all computer programs and that Congress adopted this recommendation and embodied it in the Software Protection Act of 1980. Finally, the court held that public policy required the protection of Apple's programs, and that copyright, though it may not be the most appropriate means, is preferable to no protection at all.<sup>59</sup>

While aware of the district court decision in *Apple Computer, Inc. v. Franklin Computer Corp.*,<sup>60</sup> the court distinguished it on two bases. First, the *Franklin* court had applied a different standard for preliminary injunctions.<sup>61</sup> Second, that court had merely expressed doubts concerning the copyrightability arguments espoused by Apple; its holding was that in balancing the equities the motion for preliminary injunction should be denied.<sup>62</sup>

#### E. *APPLE COMPUTER, INC. V. FRANKLIN COMPUTER CORP.*<sup>63</sup>

Apple brought suit against Franklin, alleging copyright infringement of fourteen object code operating system programs developed by Apple and included with its Apple II computer, some as software

---

56. *Id.* at 777.

57. *Id.* at 779.

58. *Id.* at 780.

59. *Id.* at 782-83.

60. 545 F. Supp. 812 (E.D. Pa. 1982), *rev'd*, No. 82-1582 (3d Cir. Aug. 30, 1983).

61. *Formula*, 562 F. Supp. at 784. A discussion of the different standards applicable to the issuance of a preliminary injunction is beyond the scope of this Note.

62. *Id.* at 787.

63. 545 F. Supp. 812 (E.D. Pa. 1982), *rev'd*, No. 82-1582 (3d Cir. Aug. 30, 1983). In January of 1984, the parties announced the settlement of Apple's suit for \$2.5 million. L.A. Times, Jan. 5, 1984, at IV-1, col. 1.

and some as firmware. These copied programs allowed the defendant to sell a computer (the Ace 1000) that is software and hardware compatible with the Apple II and the Apple II plus.<sup>64</sup> The district court denied Apple's motion for a preliminary injunction on the grounds that the plaintiff had failed to show a reasonable probability of success on the merits, or, alternatively, that Apple was better suited to withstand any injury incurred during litigation than Franklin was suited to withstand the effects of an injunction.<sup>65</sup>

The basic ground for denying the injunction was that it was not clear to the district court that computer programs in object code could be copyrighted. The court stated that "the scope of copyright is limited to material that can claim an underlying expressive or communicative purpose . . . . [T]he question must be: is the expression directed to a human audience?"<sup>66</sup> Noting that object code is incomprehensible to human beings and is intended for communication with the CPU, the district court stated that to protect these programs would be to "step into the world of Gulliver where horses are human because they speak a language that sounds remarkably like the one humans use."<sup>67</sup> The court also quoted with approval a description of firmware as a machine part.<sup>68</sup> The distinction, according to the district court, is that printed instructions explain *how* to do something while programs are *able* to do something.<sup>69</sup>

On appeal, the Third Circuit reversed the denial of the preliminary injunction and remanded the case for determination of unresolved issues.<sup>70</sup> The court interpreted the decision of the district court as the expression of "a series of generalized concerns which may have led the court to its ultimate conclusion,"<sup>71</sup> rather than as a holding. Nevertheless, the court found that the district court was concerned with four legal issues:

- (1) whether copyright can exist in a computer program expressed in object code, (2) whether copyright can exist in a computer program embedded on a ROM, (3) whether copyright can exist in an operating system program, and (4) whether independent

---

64. Thus the buyer of the Franklin computer could use anything that was sold for use with the Apple.

65. Although the court did state that its judgment was based on these alternative grounds, the exposition of the second rationale filled only one paragraph of the twelve-page opinion.

66. 545 F. Supp. at 824.

67. *Id.* at 825.

68. *Id.* at 824.

69. *Id.*

70. For example, whether Apple had complied with registration and notice requirements under the Act.

71. Slip op. at 11.

irreparable harm must be shown for a preliminary injunction in copyright infringement actions.<sup>72</sup>

For the first two issues, the court relied on its decision in a recent video game case, *Williams Electronics v. Artic International*.<sup>73</sup> The court rejected the defendant's claims, reiterating its statement from *Williams* that "[t]he answer to defendant's contention is in the words of the statute itself."<sup>74</sup> Thus, the court held that the broad language of section 102 covers computer programs in both source code and object code, and that fixation of a program in ROM falls within the statutory requirement of "fixation."

As to the third issue, which the court viewed as the "heart of Franklin's position on appeal,"<sup>75</sup> the court found no difference between application programs and operating programs. The court stated that the district court's focus on the physical characteristics of the instructions had been mistaken because the definition of computer programs in the Act makes no distinction between these two types of programs.<sup>76</sup> In addition, on the crucial issue of the dichotomy between expression and idea, the court found that the question of whether a program for an operating system is an idea or an expression must be answered pragmatically, and that the Apple operating programs will be copyrightable unless the idea inherent in the program can only be expressed in the way Apple has done.<sup>77</sup>

Finally, the court ruled that the district court had erred in its finding that an injunction was not warranted because Apple was better suited to withstand whatever injury it might sustain during litigation than was Franklin to withstand the effects of a preliminary injunction.<sup>78</sup> The court adopted the "prevailing view" that "[a] copyright plaintiff who makes out a prima facie case of infringement is entitled to a preliminary injunction without a detailed showing of irreparable harm."<sup>79</sup>

## VI. DISCUSSION

The courts and commentators fail to understand how the personal computer industry functions. They assume that the issue is

---

72. *Id.* at 12.

73. 685 F.2d 870 (3d Cir. 1982).

74. *Franklin*, slip op. at 17; *Williams*, 685 F.2d at 875 (3d Cir. 1982).

75. *Franklin*, slip op. at 19.

76. *See supra* text accompanying note 41. The court focused on the phrase "directly or indirectly."

77. The court adopted the test of "merger" between idea and expression from *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st cir. 1967).

78. *Franklin*, 545 F. Supp. at 825.

79. Slip op. at 28.

whether code in a program communicates with a human audience, either directly or indirectly. They have no problems with programs written in source code, because source code programming is understood by many people. However, some have decided that object code fails this test. What the courts and commentators do not realize, however, is that, except for a few hobbyists, no one reads a program's code. Most of the microcomputer programs purchased today are bought by people who do not even understand source code. They buy the program to perform a task—anything from keeping business records to playing Star Raiders. To the user, so long as the program works it is irrelevant what language it is written in.

In *Apple Computer, Inc. v. Franklin Computer Corp.*,<sup>80</sup> the district court asked: "is the language directed to a human audience?" Based on this test, the court refused to find that object code is copy-rightable. However, source code programming is not directed to a human audience either, but to the compiler program. Therefore, under the court's test a program written in source code would not be copyrightable. Since Congress has undeniably extended copyright protection to computer programs of some type, the district court's test must be wrong.

The trend in the microcomputer industry is to write more programs in object code and to imprint more programs on firmware. All of the recently introduced, very powerful data base management programs are written in object code. So are the best of the new computer games. The newer computers have included many more operating systems as firmware.<sup>81</sup> For example, Apple's new Lisa computer has over 500K bytes of firmware. The legal system should be aware of these trends and ensure that creators are given protection for their creations.

Patent protection would be inappropriate for these types of programs. Under present case law, patent protection cannot be extended to pure programs, but the Supreme Court has ruled that the presence of firmware does not render a machine unpatentable.<sup>82</sup> Although the patent system may be appropriate for protecting a program used to control a machine that performs a task to which patent protection is traditionally extended, this is not the case with firmware in the microcomputer industry. The firmware, as well as other object code programs, are used to process information. It

---

80. 545 F. Supp. 812, 824 (E.D. Pa. 1982), *rev'd*, No. 82-1582 (3d Cir. Aug. 30, 1983).

81. The inexpensive, portable computers such as the Osborne I and the Kaypro II have run counter to this trend. Each includes only 2K bytes of information as firmware.

82. *Diamond v. Diehr*, 450 U.S. 175 (1981).

would stretch the meaning of a patent to protect these programs under that body of law.<sup>83</sup>

Trade secret protection is inappropriate for the protection of software for two reasons. First, this type of law was not intended to cover situations in which thousands of copies of the "secret" are made available to the public. Second, where it is effective, trade secret protection defeats the economic purpose of the copyright clause<sup>84</sup> by shutting off the exchange of ideas.

The comparison of firmware to a machine part<sup>85</sup> is too simplistic. In the cases that have come up so far the IC's have been imprinted with the program at the factory and then soldered into a circuit board. It is easy conceptually to refer to this as a machine part.

Some commentators have perceived a distinction between software and firmware on the basis that "the production of firmware requires the resources of a relatively large, technically sophisticated establishment," while software can be "manufactured" with a pencil and paper.<sup>86</sup> But firmware too may be produced by someone buying a copyrighted software program (in source code) and then imprinting the program on a PROM, EPROM, or EEPROM. Today this is a simple process that can be done in one's own home using a machine that is available for a few hundred dollars. Since they are non-volatile, these ICs may be removed from the circuit and transferred to another computer or copied by a relatively simple process. At this point, firmware looks less like a machine part, and more like just another way to fix a work of authorship, comparable to having a manuscript published in microfiche form rather than as a book.

The problem with copyrighting computer programs is that some of them do not seem to fit our traditional notions of copyrightable subject matter. For this reason some critics, such as Commissioner Hersey, believe that no computer programs should be copyrighted.<sup>87</sup> This approach goes too far. However, there are some programs that should not be copyrightable. Specifically, it is operating programs that should be excluded.

The real distinction lies in the use for which the program is in-

---

83. For a discussion of patent protection for firmware and the *Diehr* case, see Moskowitz, *The Metamorphosis of Software Related Invention Patentability*, 3 COMPUTER/L.J. 273 (1982); Rose, *Protection of Intellectual Property Rights in Computers and Computer Programs: Recent Developments*, 9 PEPPERDINE L. REV. 547 (1982).

84. See *supra* text accompanying note 16.

85. In the CONTU final report, Commissioner Hersey compared firmware to a cam in a machine. CONTU, FINAL REPORT 29-30 (1978).

86. Schmidt, *supra* note 5, at 362 (1981).

87. CONTU, FINAL REPORT 28 (1978).



tended. Operating programs serve only to facilitate the use of application programs. They are "user transparent," i.e., the user is unaware of their operation. It is these operating programs that do not communicate with the human operator. Rather, they operate as a medium between the application program and the hardware. It is to these operating programs that copyright protection should not be extended.<sup>88</sup> While the Third Circuit opinion in *Apple Computer, Inc. v. Franklin Computer Corp.*<sup>89</sup> correctly points to the language of the 1976 Act as being broad enough to cover any type of computer program, the Constitution protects only "writings."<sup>90</sup>

However, even for those computer programs that are conceded to be copyrightable, copyright protection is too limited. Since copyright protects only the expression and not the idea,<sup>91</sup> pirates are free to steal the program's underlying algorithm. For example, it is unlikely that an author of a copyrighted program in one language can successfully bring an infringement action against someone who translates the program into another computer language, because of the differences between languages. Most courts would rule that only the underlying idea, i.e., the non-protected part of the program, had been copied.<sup>92</sup> There is, however no other good way to protect an algorithm; it is not patentable.<sup>93</sup>

In *Apple Computer, Inc. v. Formula International, Inc.*<sup>94</sup>, the court rejected this distinction between operating programs and application programs, stating that:

Essentially, *all* computer programs are designed to operate a machine in such a way as to ultimately produce some useful communication to the user—that is their purpose. It is difficult to understand how they can be classified into two categories for copyright purposes, with protection afforded to one category and not the other, based on whether they directly generate that communication or whether they merely direct certain machine functions which eventually result in that expression. Either all computer programs so embodied are within the terms "idea, procedure, system,

---

88. It is not the author's argument that these programs should be unprotected. Instead, it is that copyright protection is inappropriate for this type of program because it would make a mockery of the copyright system.

89. 545 F. Supp. 812 (E.D. Pa. 1982), *rev'd*, No. 82-1582 (3d Cir. Aug. 30, 1983).

90. *See supra* text accompanying note 16.

91. 17 U.S.C. § 102(b) (1982).

92. There is a continuing argument as to whether translating a copyrighted program is an infringement. Some experts would call such a translation a "derivative work," and protect it from infringement. 17 U.S.C. § 106(2) (1982).

93. *Gottschalk v. Benson*, 409 U.S. 63, 71 (1972).

94. 562 F. Supp. 775 (C.D. Cal. 1983).

method of operation<sup>95</sup> and are excluded, or all of them are outside those terms and thus protectable. There is nothing in any of the statutory terms which suggest a different result for different types of computer programs based upon the function they serve within the machine.<sup>96</sup>

The problem with this approach is that it ignores a real problem. The Constitution grants protection only to "writings." A program of which the user is totally unaware simply cannot be called a "writing." Further, something can facilitate communication, or eventually lead to the expression of a work of authorship, while being itself entirely unsuited to copyright protection.

In a poem or a novel, it is the author's expression that is of paramount importance, and the copyright system as it currently stands is adequate to protect this interest. However, in computer programs, the creator's approach to the problem is his most important contribution. Any competent programmer can take that idea and produce a program from it. Different programmers will produce different programs, but they will appear identical to the user.<sup>97</sup> It is this underlying idea which the law should protect, not the creator's form of expression.

Copyright protection should cover application programs, whether they are sold in source code or object code form. Some new form of protection, however, should be extended to operating programs written in object code. In addition, some type of protection should be granted to an author's algorithm, which is his essential contribution.

## VII. PROPOSAL: AN "IDEA COPYRIGHT"

Since it is his unique approach to the problem which is the creator's chief contribution, he should be granted a property right in this as well as in his expression. A new section should be added to the Act, creating an "idea copyright" for creators of computer programs.<sup>98</sup> Under this section, the ideas contained in all programs copyrighted previously would enter the public domain. However, upon registration of any program after the section became effective,

---

95. 17 U.S.C. § 102(b) (1982) (author's footnote).

96. *Apple Computer, Inc. v. Forrrula Int'l, Inc.*, 562 F. Supp. 775, 780 (C.D. Cal. 1983).

97. Except for inconsequential matters like the speed with which the program runs.

98. The author makes no claim that such a system should be used for any field other than computer programs. Such a suggestion could be singularly inappropriate in other areas, e.g., the entertainment industry.

both the expression and the ideas of the program would be protected.

Rather than grant a complete monopoly to a creator for his ideas, the idea copyright would grant a compulsory license to anyone who wishes to copy the creator's idea. The Copyright Royalty Tribunal would set the value of the compulsory license.<sup>99</sup> This would serve to protect the creator's property right while promoting the availability of his ideas to the public so that they could generate more ideas. Because most ideas are commercial for only a short period, the idea copyright would last for only a relatively short period.<sup>100</sup>

Once an idea copyright system is in place, authors of computer programs would no longer fear that their ideas would be infringed. As a result, they could be required to register the source code of their programs with the Register of Copyrights.<sup>101</sup> For the same reason, the authors would be willing to give the user the source code of the program. This would allow much more efficient use of programs and the ideas they contain.

Rather than employing a novelty standard, an idea copyright would protect any new idea, algorithm, or approach. For those ideas that were especially meritorious, the licensing fee would be high. For those ideas that were obvious, though original, the one who first registered the idea would receive a relatively small license fee from those who use the idea.

For questions of infringement, the idea copyright should protect not only the expressions that the author has chosen but also his underlying idea. While it is difficult to determine precisely the bounds of an author's idea, for the purposes of infringement actions, the case law that has developed in the motion picture and entertainment area could be employed.<sup>102</sup> While there is no perfect answer

---

99. See 17 U.S.C. § 801 et seq. (1982).

100. The seventeen year patent period would, by analogy, be appropriate to protect the idea copyright. 35 U.S.C. § 154 (1982).

101. Mandatory deposit of source code would be an essential source of proof that one's idea had been used.

102. The author does not claim that an idea in its simplest form (e.g., the "idea" of a computerized spreadsheet) should be protected. It is impossible to state clearly the point at which protection should attach in the continuum between copying bare ideas and copying the exact expression used by a creator. However, cases involving the infringement of works of fiction (i.e. books, plays, motion pictures) can be instructive. In these cases more than the exact expression is protected. See, e.g., *Sheldon v. Metro-Goldwyn Pictures*, 81 F.2d 49 (2d Cir. 1936). The most respected tests developed in this area are Judge Learned Hand's abstraction test, *Nichols v. Universal Pictures*, 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931), and Professor Chafee's pattern test, Chafee, *Reflections on the Law of Copyright: I*, 45 COLUM. L.

to this problem, employing the tests developed in this context has the advantage of familiarity.

### VIII. CONCLUSION

The preceding discussion has shown that a program's ability to "convey information"<sup>103</sup> to the user is independent of whether it was written in source code or object code, that courts have no real notion of why firmware should or should not be copyrightable, and that copyright protection of computer programs fails to protect that which makes them especially valuable.

Even those commentators who claim that the Act does not protect object code admit that copying object code is "unfair . . . because doing so violates the spirit of the copyright laws."<sup>104</sup> The question is not whether all computer programs should be protected, but rather what form of protection will best serve the economic philosophy that lies behind the copyright clause of the Constitution.

The copyright laws are a creation of the eighteenth century. For the last hundred years courts have served short term interests by stretching the fabric of copyright protection to cover new technologies that the law was never designed to protect.<sup>105</sup> To do this with

---

REV. 503, 513 (1945). These tests could easily be adapted to determine whether an idea copyright has been infringed.

103. 17 U.S.C. § 101 (1982).

104. Stern, *The Case of the Purloined Object Code: Can It Be Solved?*, *BYTE*, Sep. 1982, at 420.

105. See, e.g., *Burrow-Giles Lithographic Co. v. Sarony*, 111 U.S. 53 (1884) (photographs); *White-Smith Music Co. v. Apollo Co.*, 209 U.S. 1 (1908) (piano-rolls); *Buck v. Jewell-Lasalle Realty Co.*, 283 U.S. 191 (1931) (secondary transmission of radio broadcasts); *Fortnightly Corp. v. United Artists Television, Inc.*, 392 U.S. 390 (1968) (secondary transmission of television signals via cable television); *Goldstein v. California*, 412 U.S. 546 (1973) (phonorecords); *Duchess Music Corp. v. Stern*, 458 F.2d 1305 (9th Cir. 1972) (pirated tape recordings); *Williams & Wilkins Co. v. United States*, 487 F.2d 1345 (Ct. Cl. 1973), *aff'd by an equally divided court*, 420 U.S. 376 (1975) (photocopying); *Universal City Studios, Inc. v. Sony Corp. of Am.*, 659 F.2d 963 (9th Cir. 1981) (videotape recorders).

This trend may well have come to end. Just as this Note was going to press, the United States Supreme Court decided the "Betamax" case. The Court held in this case, which is the most important Supreme Court decision on copyright in at least fifteen years, that the home videotaping of television programs broadcast freely over the airwaves is not an infringement of the plaintiffs' copyrights. Writing for the 5-4 majority, Justice Stevens stated that:

"Sound policy, as well as history, supports our consistent deference to Congress when major technological innovations alter the market for copyrighted materials. Congress has the constitutional authority and the institutional ability to accommodate fully the varied permutations of competing interests that are inevitably implicated by such new technology. In a case like this, in which Congress has not plainly marked our course, we must be circumspect

the types of computer programs discussed in this Note, however, would finally cause the Copyright Act to lose all semblance of meaning. In addition, applying copyright protection to all computer programs would be both too broad and too limited. Congress should, therefore, provide a new means to protect the rights of the creators of computer programs by granting them an idea copyright. Congress has the power to do so under both the copyright clause and the commerce clause. An idea copyright could protect not only the actual program, whether written in source code or in object code, but also the creator's problem solving approach, which is, ultimately, his true creation.

*Benjamin S. Kaufman*

---

in construing the scope of rights created by a legislative enactment which never contemplated such a calculus of interests."

*Sony Corp. of America v. Universal City Studios, Inc.*, No. 81-1687, slip op. at 12-13 (Jan. 17, 1984).

The Court's decision in the *Betamax* case undercuts the argument that due to the expansive language of section 102(a) of the Act, copyright protection is granted to new developments (such as firmware) which were not anticipated by Congress when it passed the Act. The majority of the Court supports a policy of judicial "reluctance to expand the protections afforded by the copyright without explicit legislative guidance. . . ." *Id.* at 12.

In the wake of the *Betamax* decision it is no longer sufficient for a court to find a copyright infringement in the context of a copied operating program on firmware based merely on the combination of the broad language of section 102(a) and the undeniable need to protect all computer programs in some manner. Because this problem has not been considered by Congress, in the future courts may well be compelled to follow the reasoning of Justice Stevens and yield the problem to Congress for a legislative solution.