Copyright Protection for Computer Programs in Read Only Memory Chips

Peter D. Aufrichtig
COPYRIGHT PROTECTION FOR COMPUTER PROGRAMS IN READ ONLY MEMORY CHIPS

INTRODUCTION

You walk into a store looking for a computer chess game and see two apparently identical models. One, however, is considerably cheaper than the other. You inquire as to the difference and are told that there is none, but for the price. The reason for this price differential is the lack of copyright protection for a computer program implanted in a Read Only Memory (ROM) chip.

As the computer has developed, the programs which make the computer useful have also changed. To bring the computer out of the temperature and humidity controlled rooms, and into the consumer's palm, various advances have been required. First, the computer itself became smaller, faster, and more affordable than was imaginable twenty-five years ago. Second, the memory spaces for the computers have developed to the stage where they can hold billions of bytes of information in chiclet-sized packages. Third, the programs which run the hardware can be stored so that they are nonvolatile, and

* This note has been submitted in a slightly different form to the Nathan Burkan Memorial Copyright Competition.

1. See, e.g., Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038 (7th Cir. 1980).
2. See infra text accompanying notes 31-37.
5. A byte is the computer's equivalent of a word and is usually eight binary digits long. A. Vazsonyi, Introduction to Data Processing 492 (3d ed. 1980).
6. See C. Evans, supra note 4, at 103-04. The first computers had memories that filled up large rooms. See id. at 74-75 (photo of the Harvard Mark I computer).
7. Hardware is the fixed portion of the computer, consisting of the logic circuits, memory devices, input-output devices and connections. See S. Corbett, supra note 3, at 113. In comparison, software refers to "a set of programs, procedures and possibly associated documentation concerned with the operation of a data processing system." A. Vazsonyi, supra note 5, at 499.

329
can be electronically attached to the hardware, providing a completely functional computer in a box with an on/off switch.

This note will focus on the copyright protection afforded to computer programs implanted in ROM's. First, an exposition of the significance of this area will be undertaken, discussing the ability of individuals and small businesses to take advantage of the time, space and energy saving capabilities of the computer, as well as the computer's recreational, educational and creative uses.

The second portion of this note provides an introduction to computers, software (computer programs) and firmware (the general classification of software implanted in hardware of which ROM's are a part) for the nontechnically oriented reader, with references to more technical source materials.

The third section describes the history of copyright protection extended to computer programs and an analysis of two cases that have dealt directly with this issue. This note will not concentrate on the availability of copyright protection for computer programs in general, as the subject has been addressed elsewhere, and it is now accepted that computer programs, in at least some forms, are a

8. A nonvolatile memory is fixed and, unlike a volatile memory, does not need an electrical current to be saved. See id. at 500.
9. Video game cartridges are just one example of how computers may be used for recreation.
10. Foreign language computer modules provide one example of how computers may be used for educational purposes.
11. There are programming features available to the user and software that allows the user to create computer generated art. See generally N. Graham, The Mind Tool 158-66 (2nd ed. 1980).
12. See infra notes 38-138 and accompanying text.
14. See infra notes 139-300 and accompanying text.
proper subject matter for copyright protection. 16

The fourth section focuses on the uncertain state of the law regarding copyright protection of computer programs imprinted in ROM's, the work of the National Commission on New Technological Uses of Copyrighted Works (CONTU) 17 which led to the computer software related amendments of the copyright law, 18 Commissioner John Hersey's 19 alternative proposal contained in the dissent to CONTU's final report, 20 and recommendations for an optimal approach for copyright protection of proprietary computer programs. 21

In addition, the future of the software and firmware industries and their foreseeable legal protection will be examined. 22

The note concludes by suggesting that the present copyright law be changed so that proprietary computer programs, such as those implanted in ROM's, will be afforded more adequate copyright protection than they are currently given. 23 The changes proposed, if adopted by the legislature and properly interpreted by the courts, will yield decisions that reflect a proper balance of the proprietary interests of software and firmware manufacturers and the American public.

SIGNIFICANCE OF THE AREA

The copyrightability of computer programs implanted in ROM's, 24 and the protection that is afforded them, have a wide and growing importance in our society. Currently, copyright protection of computer programs in this form is relevant to home and small business computers, 25 as well as hand-held 26 and arcade video games. 27

17. CONTU Final Report, supra note 16. CONTU was established to study and compile data on the use of copyrighted works of authorship in conjunction with computers and machine reproduction-xerography and to produce a final report with recommendations for changes in the copyright law or procedure. See Pub. L. No. 93-573, 88 Stat. 1873 (1947).
19. Mr. Hersey has authored 18 books and has won a Pulitzer prize, in addition to being president of the Author's League of America and a commissioner of CONTU. Pope & Pope, supra note 15, at 549 n.132.
21. See infra notes 339-77 and accompanying text.
22. See infra notes 301-24 and accompanying text.
23. See infra notes 286-324 and accompanying text.
24. See supra note 3.
As the computer continues to enter the home, there will be an expanding need for meaningful protection of proprietary computer applications programs, especially those fixed in ROM’s, magnetic tape and cartridge form. Consumers now buy cartridges for their home computers to play a myriad of video games, learn French, Spanish and Italian, and keep track of their checking account balances. In the next few years, as personal-sized computers become less expensive, the software packages will become more sophisticated and the data bases more expansive, with telephone hook-ups to stock exchanges, supermarkets, department stores, libraries and schools. A major component of this information explosion will be proprietary firmware attached to the hardware. Forging ahead in this time of spectacular change, with uncertain protection of proprietary firmware, could result in dire consequences. Without adequate protection, the economic incentive to produce new programs could disappear.

As in the sound recording field, the cost of duplication of firmware is far less than the cost of development. With sound recordings, tape “pirates” were able to buy one record and make copies from that master without paying the artist’s royalties, studio costs, advertising overhead and pre-production costs. When Congress overruled the effect of a sixty-five-year-old decision and


28. A data base represents the information which is accessible to the computer. Data bases are expandable by telephone line connections with other computers. See A. VAZSONYI, supra note 5, at 336-37.

29. Proprietary firmware includes computer programs in a fixed medium such as a ROM or magnetic cartridge which is prepared for sale. See S. CORBETT, supra note 3, at 112; A. VAZSONYI, supra note 5, at 112-13; N. GRAHAM, supra note 11, at 72.

30. See supra note 7.

31. Duplicating a sound recording requires a turntable, a cassette recorder and a cassette. These can be purchased for a few hundred dollars. Assuming the tape pirate already has this equipment, the only costs are buying one copy of a record and as many cassette tapes as copies are desired. A cassette costs approximately three dollars. See Goldstein v. California, 412 U.S. 546 (1973).

“Forging a Pac Man or Centipede [video] game is not much more complicated than pirating a music cassette or videotape.” Skow, Games That Play People, TIME, Jan. 18, 1982, at 50, 56.

32. See Goldstein v. California, 412 U.S. 546, 550 n.5 (1973). California had a state law proscribing tape piracy, but there were no federal, civil or criminal penalties for tape piracy. CAL. PENAL CODE § 653(b) (West 1970).

COPYRIGHT OF COMPUTER PROGRAMS

granted federal copyright protection to sound recordings in 1972,\textsuperscript{34} tape pirates became liable to federal civil and criminal penalties.\textsuperscript{35} The software and firmware industry is faced with a similar economic dilemma: Copying a ROM or magnetic cartridge is very inexpensive when compared to the research and development costs associated with bringing a proprietary computer program in these forms to the market.\textsuperscript{36}

If "program pirates"\textsuperscript{37} are allowed to operate with the sanction of the courts, it would likely weaken the incentive for new companies to enter the field and cause those companies that remained in business to raise their prices in the hope of recouping their research and development costs with fewer sales. Significantly higher prices would discourage consumers from investing as freely in home computer hardware. Furthermore, if the market for home computer hardware were to shrink, then not only would companies that design and manufacture home computer systems suffer, but individuals and small businesses that make use of the time, space and energy saving capabilities of such computer systems would likely alter the ways in which they make use of computers, sacrificing efficiency in the process.

Proper copyright protection for firmware and software could avoid these dire results by providing programmers with security for their creations so that they could recapture their costs and make a profit. This would result from their ability to plan on a longer useful life for their asset—the program—and a lower unit sales price.

A BRIEF INTRODUCTION TO COMPUTERS

It is almost gospel that the most important factor in our technological revolution is the computer. The first computer, as we know it, was built in 1944.\textsuperscript{38} The computer age, however, did not truly begin

\begin{itemize}
\item[37.] A program pirate is an individual or corporation that makes "pirated" or illegal copies of a computer program. Another name for such a person or company is a "knock-off artist." \textit{See} Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171 (N.D. Cal. 1981); Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 854-55 (2d Cir. 1982); Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979), \textit{aff'd on other grounds}, 628 F.2d 1038 (7th Cir. 1980).
\item[38.] M. CAMPBELL-MARTIN, \textit{The Computer Age} 114 (1978).
\end{itemize}
until 1945 to 1946 when John von Neumann wrote on computer architecture and developed a machine with John Mauchly and J. Presper Eckert, Jr., putting his theories to work. The first computer was built with vacuum tubes as the basic building block. The tubes were relatively unreliable. In addition, they generated enormous amounts of heat which posed burdensome cooling problems. Vacuum tubes were bulky and had the additional problem of reacting slowly.

In the late 1940's, the development of the transistor facilitated the second generation of computers. The transistor had many advantages over vacuum tubes: It was cheaper, smaller, faster, more reliable, longer lasting, less costly to operate, and operated over a broader range of temperatures. The transistor was the beginning of the semiconductor industry—based on silicon—and has been the basis of all further advancement in semiconductor products. After only a few years, the transistor was replaced as the basic unit in computers by the printed circuit, which was an entire circuit laid out on a thin board to which transistors, resistors and capacitors were attached in predesignated holes. The printed circuit helped make computer assembly quicker and more automated and allowed circuits to be tested at a functional level. In the event of failure, it was no longer necessary to remove, replace and retest the individual components that had malfunctioned. The printed circuit could be replaced as the smallest component, after being separately tested, thereby saving time and simplifying maintenance. In addition, the

40. See id. at 16-17.
41. See id. at 17; N. Graham, supra note 11, at 27-28.
42. N. Graham, supra note 11, at 28; A. Vazsonyi, supra note 5, at 93.
43. See C. Evans, supra note 4, at 82.
44. Tubes, by their nature, are electronic, comprised of components which must physically be heated, and, therefore, require time to heat up. They require additional time to recover voltage once they have switched, in order to switch again. Transistors, which are purely electrical and direct a flow of electricity without requiring the electricity to be converted into heat, do not require time to “warm up” or recover voltage between switching cycles. See id. at 93.
45. See id. at 92.
46. See id. at 93-96.
47. Id. at 92-93.
48. A semiconductor refers to a silicon product that is doped with various impurities to provide desirable electrical characteristics. See Stonier, What Makes a Micro Tick, in id., at 103, 109-11. Doping is the introduction of an impurity into the silicon. See id., at 103, 109.
49. Id. at 110.
50. See id. at 109-10.
circuits could be individually optimized which led to greater speed, smaller physical size and lower cost. This, however, had barely developed when the next generation of computers based on integrated circuits emerged.\footnote{51}

The integrated circuit surpassed the printed circuit since all of its necessary components and connections were formed in silicon.\footnote{52} At the same time, the design function of the engineer had changed. No longer would a designer use transistors as the base of his design. Instead, he constructed with a new base, using logic devices known as gates (\textit{and, or, nand, nor, not}), which are digital\footnote{53} in nature and take digital inputs (only 0's and 1's) to produce digital outputs.\footnote{54}

The next step in the development of computers involved placing more gates on a smaller silicon chip. By 1976, the industry had succeeded in placing what had once been called a “mainframe” computer\footnote{55} on a single chip.\footnote{56} This process is known as Large Scale Integration (LSI).\footnote{57} In the last few years, smaller chip geometries have been achieved and more has been squeezed onto a single chip—known as Very Large Scale Integration (VLSI).\footnote{58} The computer industry has also shrunk the size of memory devices for storing information.\footnote{59}

The working of a computer has been analogized to a manager

\begin{itemize}
\item \footnote{51}{Computers based on integrated circuits emerged in the late 1960's. N. Graham, \textit{supra} note 11, at 28.}
\item \footnote{52}{Id.}
\item \footnote{53}{Digital refers to a binary representation. Binary is a number system of the base 2. It has only 0's and 1's. Thus, 1 in base 2 would be 1, while 10 in base 2 is 1010 \((1 \times 8) + (0 \times 4) + (1 \times 2) + (0 \times 1)\). This is similar to the base 10 system we deal with daily. In base 10, 253 is \((2 \times 100) + (5 \times 10) + (3 \times 1)\). The multipliers are powers of the base. In base 10 they are \(1 = 10^0, 10 = 10^1, \text{and } 100 = 10^2\). In base 2 they are \(1 = 2^0, 2 = 2^1, 4 = 2^2, 8 = 2^3\). For a more detailed discussion of the binary number system, see W. Davis, \textit{supra} note 3, at 96-101.}
\item \footnote{54}{A digital output would be either a 1 or a 0, corresponding to “Yes” or “No”. \textit{See id.} at 98.}
\item \footnote{55}{A mainframe computer is an entire computer in a single box. \textit{See id.} at 383-87; A. Vazsonyi, \textit{supra} note 5, at 496.}
\item \footnote{56}{M. Campbell-Martin, \textit{supra} note 38, at 114.}
\item \footnote{57}{LSI is the miniaturization of circuits so that thousands and then tens of thousands of components can be placed on a single chip. C. Evans, \textit{supra} note 4, at 104; \textit{see W. Davis, supra note 3, at 141.}}
\item \footnote{58}{VLSI is merely the further miniaturization of LSI chips by squeezing more and more components onto a single silicon chip. \textit{See W. Davis, supra note 3, at 141; C. Evans, supra note 4, at 104; U. Pooch & R. Chattergy, supra note 3, at 1; IEEE Spectrum, Mar. 1980, at 81.}
\item \footnote{59}{\textit{See C. Evans, supra note 4, at 104-05.}}
\end{itemize}
surrounded by two secretaries. William Davis, author of Information Processing Systems, suggests looking at a computer system as a three room suite with the manager in the center office which is accessible to the input secretary on one side and the output secretary on the other. The manager has a stack of cards with instructions on his desk which correspond to the program. On his wall, the manager has a chalkboard with numbered spaces, another chalkboard with three spaces (A, B, and Answer), and two buttons (input and output). The flow of all work is from the input secretary to the manager and, finally, to the output secretary. If someone on the outside wants the manager to do some function, the request has to be given to the input secretary. The manager reads the first card which says “press input button.” Pressing the input button rings a bell in the input secretary’s office, who then takes the top card off his desk and brings it into the manager’s office, where the secretary copies the information onto the numbered chalkboard. The manager then reads the next card which tells him to write the first number on his numbered chalkboard in the space marked A. The third card tells the manager to copy the second number on the numbered chalkboard onto the area marked B. The fourth card tells him to add the amount in A to the amount in B and write the result in the area marked Answer. The fifth card tells the manager to push the output button, which rings a bell on the output secretary’s desk. The output secretary comes in, writes down the number in the Answer area and returns with that number to the output secretary’s office. This is an orderly process in which the input secretary only brings information from the outside world to the manager and the output secretary provides a link from the manager’s office to the outside world.

The manager in the above analogy corresponds to the central processing unit (CPU) of a computer. The CPU performs all arithmetic, receives input data from input devices (input secretary), reads from and writes into the memory (numbered chalkboard), writes data on the accumulator (spaces A and B), and passes data to output devices (output secretary) such as a printer or CRT terminal.

60. See W. Davis, supra note 3, at 48-57.
61. W. Davis, supra note 3.
62. Id. at 48.
63. See id. at 48.
64. For a more detailed description of this analogy, see id. at 48-57.
65. See id. at 58.
66. An accumulator is the register in which the results of logic or arithmetic functions are formed. Id. at 483.
67. A printer is a “device that expresses coded characters as hard copy.” Id. at 493.
Like the manager, the CPU can only do one thing at a time and only what its instruction permits.68

The missing actor in the above analogy is the writer of the cards—the programmer. If the CPU has no instruction, it will wait quietly for one to appear.69 The sequence of cards that the CPU/manager follows is the program. By varying the instructions, the programmer can cause the computer to perform a myriad of different tasks.70

Input devices take various forms. One type of input device is a terminal which has a keyboard similar to a typewriter and either a screen, called a Cathode Ray Tube (CRT), or a printer.71 The user72 types information into the computer, either in the form of programs or data.73 Another type of input device is a card reader which "reads" a deck of punched cards containing coded information (the familiar cards that come with telephone bills and instructions not to fold, spindle or mutilate).74

Output devices include printers, which produce the computer printouts—bills, statements, reports and graphics—and CRT terminals which display information on a television-like screen.75 In addition to these output devices designed to interface with a human user, MODEM's76 allow computers to communicate over the telephone lines.

Memory devices can also take different forms. Memory is the part of the computer that stores information for later use.77 It would be more accurate to refer to memory as "storage," because the information is stored and lays dormant until it is retrieved by the CPU.78 There are two basic types of memory: volatile, or dynamic, and non-volatile. The volatile memories, which, as a group, are faster than

67. A CRT is a Cathode Ray Tube. Less mysteriously, a CRT is a television screen. A CRT terminal is a screen on which the computer can output information. See id. at 82-83.
68. A. VAZSONYI, supra note 5, at 281.
70. See id.
71. See id. at 81-87 (describing various types of terminals).
72. "User" is "a broad term referring to anyone who requires the services of a computer system." A. VAZSONYI, supra note 5, at 500.
73. See W. DAVIS, supra note 3, at 81.
74. See id. at 63-66.
75. See supra note 71.
76. A MODEM is a device which modulates and demodulates digital electrical signals into analog electrical signals and back into digital electrical signals, to allow transmission of information over telephone lines. A. VAZSONYI, supra note 5, at 496.
77. See W. DAVIS, supra note 3, at 491, 496.
78. See A. VAZSONYI, supra note 5, at 496, 499.
the nonvolatile memories, clear themselves of information if the power is shut off.79 The nonvolatile memories, however, do not need refreshing80 to maintain the information they contain.81 Most computer systems have both volatile and nonvolatile memory components. A computer memory system is usually composed of a relatively small “main memory,” which is very quick, and a much larger, nonvolatile “auxiliary memory.”82 This configuration provides the computer with a working memory space which is almost as fast as the CPU and a virtually unlimited memory from which it can draw stored information, programs and data, to perform many different tasks.83

The main memory in every computer is a Random Access Memory (RAM).84 The CPU can access85 any part of the RAM in the same amount of time.86 This is extremely important for the efficient operation of the CPU which has an internal clock regulating its functions.87 The computer’s auxiliary memory can take several different forms, each having different characteristics. Magnetic tapes, somewhat similar to ordinary audio cassettes, are a sequential access memory. To get information stored at the end of the tape, the tape must be advanced.88 With RAM memories having fetch cycles89 operating within millionths of a second, it is obvious that physically winding a tape is a very slow process by comparison. Magnetic tapes are useful, however, when large amounts of information are sought in the same order they were recorded.90 Another type of auxiliary memory device is a magnetic disk. The magnetic disk resembles a

79. Id. at 274.
80. “Refreshing” refers to repetitive electrical surges used to retain the electrically or magnetically stored information in memory devices. Id.
81. See id. at 274.
82. See N. Graham, supra note 11, at 73-74. Auxiliary storage supplements must be contrasted with the main storage. W. Davis, supra note 3, at 484.
83. A virtual memory system is one in which internal memory and auxiliary storage are combined by software to give the illusion of a larger main memory system. A. Vazsonyi, supra note 5, at 274.
84. W. Davis, supra note 3, at 109.
85. A computer accesses a memory location by copying the information at that memory location onto the accumulator. Id at 216-19.
86. Id. at 487 (direct access is the same as random access).
87. U. Poock & R. Chattergy, supra note 3, at 36.
88. N. Graham, supra note 11, at 38.
89. A “fetch cycle” refers to the operation of retrieving an instruction from memory. See id. at 57.
90. Id. at 39.
grooveless phonograph record,\textsuperscript{91} and can either be rigid or flexible (floppy disk).\textsuperscript{92} When the disks are in their "disk drives,"\textsuperscript{93} they are spun at high speed\textsuperscript{94} with tonearm-like access arms which can read or write data from the disk in a few ten-thousandths of a second.\textsuperscript{95} The magnetic disks, unlike the magnetic tapes, are random access, allowing for more consistent, and faster, fetch cycles.\textsuperscript{96} Even these efficient disks, however, are becoming outmoded. A new type of memory is known as Bubble Memory.\textsuperscript{97} The advantages of magnetic bubble memory are numerous. Magnetic bubble memories are smaller than magnetic disks, are faster, and have no moving parts\textsuperscript{98} because they are completely electrical. It is likely that these features will cause bubble memories to replace their less efficient predecessors in many memory applications.

Another type of memory device is called a Read Only Memory (ROM), which, as the name suggests, can only have information read out from, and not changed or written into, by the CPU.\textsuperscript{99} The ROM's, by their very nature, must be nonvolatile, and their fixed memory characteristics make them most useful for permanent computer program storage. In addition, ROM's are particularly important because they may be permanently added to a computer system to provide a system program library.\textsuperscript{100}

As discussed above, the computer program instructs the CPU.\textsuperscript{101} Unfortunately, binary computer circuits\textsuperscript{102} do not understand English sentences and, therefore, instructions must be con-

\textsuperscript{91} Id. at 37.
\textsuperscript{92} Floppy disks are usually used only for home and small business computers because of their relatively limited storage capacity. W. Davis, supra note 3, at 488; N. Graham, supra note 11, at 37.
\textsuperscript{93} Disk drives are machines used to access information on disks. See W. Davis, supra note 3, at 269.
\textsuperscript{94} Disk drives are typically spun at approximately 1500 revolutions per second. N. Graham, supra note 11, at 37.
\textsuperscript{95} See W. Davis, supra note 3, at 268.
\textsuperscript{96} N. Graham, supra note 11, at 37-39.
\textsuperscript{97} Magnetic Bubble Memory is a "type of storage in which data are stored as a series of bubbles in a thin substrate." W. Davis, supra note 3, at 491.
\textsuperscript{98} N. Graham, supra note 11, at 39.
\textsuperscript{100} See N. Graham, supra note 11, at 102 (offering a microprocessor-controlled appliance as an example of an application in which a ROM is permanently added to a computer system to hold a program).
\textsuperscript{101} See supra notes 68-70 and accompanying text.
\textsuperscript{102} Binary circuits is simply another manner of referring to digital circuits, i.e., Base 2 logic. See W. Davis, supra note 3, at 484.
verted into a form which they will comprehend.

The CPU only understands instructions which are written in "machine language." Machine language is a binary language, which means that a single instruction might look like this—010100000110000. Machine language is not convenient for the programmer to work with because the instructions are difficult to become familiar with and each instruction does only a small task—it takes seven machine language instructions to execute C=A+B.

Computer engineers realized early that it was not efficient to write programs of any size or complexity in machine language. It should be noted, however, that all programs which are executed by computers must be in machine language before the computer can understand them.

The next higher level that programs can be written in is called "assembly language." Assembly language statements correspond on a one-to-one basis with machine language statements, but have mnemonics associated with each statement that allow the programmer to write a program in a nonbinary medium. An assembly language statement would look something like these: RD SALES, LD SALES, or SB CUTOFF. Once a program is written, either the programmer or a compiler then converts the assembly language statements into machine language.

Even assembly language is an inefficient level at which to program a computer, because it would still require seven instructions just to add two numbers. Also, virtually the same seven instructions would be used each time two numbers were added together. The same would be true for all arithmetic and logical instructions which require several machine level instructions. The next step, therefore, was to develop a program written on the machine-assembly language level which could convert a statement like C=A+B, written by the

103. Machine language relates to a "computer program in binary form capable of being executed on a computer." Id. at 490-91.
104. See id. at 170, 176.
105. Id.
106. See N. Graham, supra note 11, at 67-68.
107. Id. at 68-69.
108. Assembly language is "a symbolic source programming language using mnemonic, symbolic codes to designate both the operations to be performed and the locations in storage for the data." A. Vazsonyi, supra note 5, at 491. A statement is the same as an instruction. See N. Graham, supra note 11, at 292.
109. See N. Graham, supra note 11, at 68-69.
110. A compiler is a program that translates a higher-level source program into a machine language program. See id. at 70; A. Vazsonyi, supra note 5, at 492.
programmer, into its machine language counterpart. This type of a program is called a compiler. A compiler is written to accommodate a "higher-level" programming language such as FORTRAN, BASIC, COBOL or PL/1. The programmer then writes his program in these higher-level languages without concern for the machine level instructions. Each of the higher-level language statements can correspond to "five, ten, or even more machine language instructions." Once the programmer has completed a program and seeks to test it, the program is inputted into the computer which has been signaled to ready the compiler for the higher-level language that has been used. The compiler then translates the higher-level statements into their machine level counterparts and runs the translated program. This machine level program is also called the "object code." The compiler program is generally a long program which may take several years to write, and will be used frequently by the computer. As a result, many compilers are stored on ROM chips which are attached to the hardware, and are, therefore, considered firmware. In fact, any program, once it is converted to object code can be placed permanently in a ROM, PROM (Programmable Read Only Memory), or EPROM (Erasable Programmable Read Only Memory).

A computer program can take several different forms and a program's manifestation in one of these forms may affect its legal treatment. The first form a computer program takes in its development is a flow chart. A flow chart is, strictly, not a program, but a diagram of the logical operations that will be performed. The flow

111. A "higher-level" programming language is one that is conceptually oriented. See N. GRAHAM, supra note 11, at 69-70.
112. Id. at 70.
113. Id.
114. See supra note 110.
115. See supra notes 103-06 and accompanying text.
116. N. GRAHAM, supra note 11, at 70.
117. The ROM's become part of the electrical circuitry. See W. DAVIS, supra note 3, at 391-92.
118. See supra note 3.
119. Object code is a binary form. W. DAVIS, supra note 3, at 174.
120. Id. at 494. A PROM is a ROM which the user or manufacturer can write into.
121. Skow, supra note 31, at 56. An EPROM is a PROM which can be erased for reuse. See W. DAVIS, supra note 3, at 391.
122. See Data Cash Syss., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063, 1066 n.4 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038 (7th Cir. 1980) (distinguishing between protectability of computer programs in source code and object code).
123. A. VAZSONYI, supra note 5, at 494.
chart is the first expression of the programmer's ideas on the problem that will become a program. Next, a written copy of a higher-level program is made. Another format is a deck of computer punch cards which may have a program coded in a higher-level, assembly or machine language. The program is still in a written form, although it would be difficult for the average person to read. The next form a program could take is as the object code, stored in a ROM or other memory device. In this form, the program, in machine language, is manifested by a binary representation composed of electrical or magnetic bits which are either on (1) or off (0). It would be impossible at this level for even a skilled person, unaided by a machine, to read this program. With the proper machine, i.e., whatever machine the program was written for, however, the program statements could be "read out" in their binary/machine language form.

The cost of developing application programs in higher-level languages is great, and, generally, only medium to large companies can afford to have applications software written for their specific needs. Fortunately for owners of personal and small business computers, most such users have similar needs, which allow software companies to develop standardized packages. Since the software companies expect to sell more than one copy of a program, this reduces the cost of each copy, making applications software available to personal and small business computer users at prices they can afford. If this market in software packages (and firmware, since most software will be delivered as firmware to be "plugged into" small computers) did not exist, most personal and small business computers would not have been sold. Examples of such software packages are those that handle bookkeeping, payroll, check balancing, and inventory control. In addition, home and arcade video games, a recent growth industry generating mass market sales, is based on microcomputer—ROM—technology.

When a software company develops a software package, it can either sell the program to everyone or license the use of it, reserving

124. "Bit" is the abbreviation for Binary digit. Id. at 492.
125. "Read Out" means displayed to the user.
126. "An application program performs a direct function, taking data and processing it into information of greater value." W. Davis, supra note 3, at 310.
127. See Skow, supra note 31, at 58.
Where the program is sold outright, a purchaser can duplicate the program and resell copies without incurring research and development costs inherent in creating a software package. The means of delivery for the machine language program may be magnetic tape, magnetic disk or ROM. The first two mediums of delivery are costly. A user, desiring to run such a software package, must either own or acquire a disk drive or tape reader, both of which are quite expensive. The ROM, on the other hand, is often plugged right into the computer. Another benefit of the ROM is its compact size. Finally, for the software developer who plans to sell many different software packages compatible with a single computer, the most efficient means of delivery and marketing would be in ROM packages.

Currently, ROM's are widely used as software delivery mediums. Software imbedded in a ROM, which becomes part of the hardware, is labelled firmware by the software industry. Firmware is used for personal and small business computers as the operating system and compiler. It is also used by home computers to produce video games. In addition, firmware is the preferred delivery form in arcade video games, especially the more complicated designs.

In the near future, as home computer prices decline—and more consumers purchase home computers—the market for standardized software packages, likely to be in the ROM form, will mushroom. The computer's move from a vast laboratory at a re-

130. For example, a BASIC compiler cost 1500 pounds sterling when it was introduced, but now costs only 15 pounds. In addition, the cost of copying a program into firmware is 1/10,000 of the cost of developing the original program. I. BARRON & R. CURNOW, THE FUTURE OF MICRO ELECTRONICS 102-03 (1979).
131. See infra note 135.
132. A ROM is about an inch by a quarter of an inch by an eighth of an inch—the size of a chiclet.
133. Firmware is "1) Hard-wired computer logic for performing computer functions previously or normally done by programs. 2) Software and hardware that interact so closely and mutually that the functions of both are inseparable." W. DAVIS, supra note 3, at 488.
135. Home computer cartridges are a form of firmware—software plugged directly into the hardwired circuits. See Skow, supra note 31, at 57.
137. Hardware prices have uniformly dropped after their initial appearance on the market. See W. DAVIS, supra note 3, at 22.
138. I. BARRON & R. CURNOW, supra note 130, at 56.
search university to the American home, however, has not ended.

**COPYRIGHTABILITY OF ROM's**

Many cases have influenced the copyrightability of software and firmware. In *Baker v. Selden*, the Supreme Court distinguished between an idea and the expression of that idea. The Court, in this 1879 decision, held that the underlying idea expressed in *Selden's Condensed Ledger, or Bookkeeping Simplified* was not protected under the copyright law, although the expression—the actual description used in the book—was protected. This principle has been applied to computer programs, which some believe are nothing more than an idea. This view, however, has been rejected, because the computer program represents the expression of an underlying idea, the purpose of the program, and hence, the program is a proper subject matter for copyright protection.

The *Baker* Court distinguished between the availability of copyright protection and the scope of that protection. This distinction has assumed great significance in relation to the copyrightability of computer programs, which are generally accepted as copyrightable, although the scope of that protection is not clear.

139. Software is simply another term for computer program. *See W. Davis, supra* note 3, at 495.
141. 101 U.S. 99 (1879).
142. *Id.* at 102-04.
143. *Id.* at 104-05.
144. In *Baker*, the Court held that Selden's "T form" accounts, used in his double entry bookkeeping system, was an idea that was not copyrightable, although Selden was entitled to protection of his explanation of the system. *Id.*
147. 101 U.S. at 107. The *Baker* Court held that Selden's book was copyrightable, but that the scope of his protection was constrained to his explanation of his system. *Id.*
149. *Compare Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982)*

http://scholarlycommons.law.hofstra.edu/hlr/vol11/iss1/8
In *Burrow-Giles Lithographic Co. v. Sarony*, the Supreme Court, in 1884, extended copyright protection to photographs by expanding the definition of a "writing," as used in the Constitution, to include a photograph. Evidently the 1802 copyright law did not include photographs in its list of copyrightable subject matter. The Court observed that photography did not exist when that law was enacted and that a photograph can reflect the necessary work of authorship demanded by the Constitution and the statute.

The Supreme Court was willing to judicially extend the constitutional definition of a writing to include photographs only after Congress evinced an intention to grant photographs copyright protection. The courts, however, have not been similarly influenced by congressional intent in the computer program area.

*White-Smith Music Publishing Co. v. Apollo Co.* was the first major case bearing upon copyright protection of software and firmware. In *White-Smith*, the Supreme Court held that a piano roll was not a copy, for copyright purposes, of the sheet music played by the piano roll. The Court reasoned that the piano roll was actually a part of a machine, rather than a copy of sheet music. A copy, observed the Court, must be visually perceptible to

---

150. 111 U.S. 53 (1884).
151. Id. at 58.
153. 111 U.S. at 58.
154. See id. at 57 (citing 2 Stat. 171 (1802) which amended the copyright law to include etchings, engravings, and their prints).
155. In 1874, Congress passed a law adding photographs as a proper subject for copyright protection, 18 Stat. 78 (1874), but the constitutionality of this law had been called into question by *Burrow-Giles*. See 111 U.S. 53 (1884).
156. See 111 U.S. 53 (1884).
157. See supra note 155.
158. See, e.g., Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038 (7th Cir. 1980) (computer programs in ROM's not entitled to copyright protection).
159. 209 U.S. 1 (1908).
160. A piano roll is the perforated cylinder, placed in a player piano, that plays the keys. In this case, two copyrighted songs were encoded onto a piano roll. Id. at 8-10.
161. Id. at 18.
162. Id.
the human eye and must "give to every person seeing it the idea created by the original."\textsuperscript{163} The Court's insistence that a copy be visually perceptible to warrant copyright protection stymied federal copyright protection for sound recordings until 1972.\textsuperscript{164} Furthermore, it has hampered acceptance of copyright protection for software and firmware, despite the recent changes in the copyright law\textsuperscript{165} extending such protection.\textsuperscript{166}

The argument against copyright protection for computer programs is deceptively simple. Its proponents state the following: The purpose of a computer program is to direct a machine in performing a task.\textsuperscript{167} At some point, before the program can be used, it must be converted into its binary/machine language electrical representation,\textsuperscript{168} at which point the program is no longer a writing as contemplated by the Constitution.\textsuperscript{169} Since a program is of no value until the computer can interpret it, determination of copyrightability should be made at the machine language level.\textsuperscript{170} Therefore, because the program, i.e., a binary, machine-directing electrical component, is part of the machine, and functional, it is not entitled to copyright protection.\textsuperscript{171}

This argument ignores the important fact that a computer program, as it is designed, is the expression of an idea—whatever idea the programmer is trying to put into action through the use of the computer. Furthermore, in view of the fact that the 1976 Copyright Act extended protection to phonograph records\textsuperscript{172} and videotape recordings,\textsuperscript{173} the contention that a computer program does not fall within the parameters of copyright protection is no longer tenable.

At this point, it is necessary to distinguish between operating systems and applications programs. An operating system is a

\begin{footnotesize}
\begin{enumerate}
\item[163.] \textit{Id.} at 17 (emphasis added) (quoting West v. Francis, 5 B. \& A. 743).
\item[166.] 17 U.S.C. § 102(a) (Supp. IV 1980).
\item[167.] See \textsc{A. Vazsonyi}, supra note 5, at 102.
\item[168.] \textit{Id.}
\item[169.] U.S. \textsc{Const.} art. I, § 8, cl. 8.
\item[170.] For a discussion of the use of machine language instructions, see \textsc{A. Vazsonyi}, \textit{supra} note 5, at 102.
\item[171.] \textit{See} 17 U.S.C. § 102(b) (Supp. IV 1980).
\item[172.] \textit{Id.} § 102(a)(7).
\item[173.] \textit{Id.} § 102(a)(6).
\end{enumerate}
\end{footnotesize}
software/firmware package that coordinates the various components of a computer system—the processor, memory and interfaces—so that the system may accept programs in a specified format.\textsuperscript{174} An applications program is written to function in the environment created by the operating system.\textsuperscript{175}

One way of viewing the relationship between an operating system and applications programs is to analogize to a ratchet wrench set having one ratchet piece and many different sized heads for different bolts. Like the operating system, the ratchet is essential for the operation of the wrench with any of the heads. Another analogy is to the autonomic (operating system) and central (applications programs) nervous systems, where the autonomic system controls heartbeat and other involuntary functions without which we would not survive, while the central nervous system controls thought and expression.\textsuperscript{176} Therefore, because of the functional nature of an operating system, additional care must be taken to determine the expression in the operating system. It is this expression which is copyrightable.

It is only the expression of an idea that is protected by copyright law.\textsuperscript{177} An operating system, in a ROM (object code), is a functional part of the whole computer system.\textsuperscript{178} At one time, operating systems and applications programs were wired portions of the computer system.\textsuperscript{179} With the advance of technology, it is now less costly and more efficient to place them in ROM form.\textsuperscript{180} Operating systems and applications programs in ROM form, however, still serve the same purpose as when they were hard wired.\textsuperscript{181}

\textsuperscript{174} An operating system is a "program which supervises the execution of other programs." N. Graham, supra note 11, at 72. The operating system handles program loading, control of peripherals, data and memory, management, accounting and computer security. See id. at 72-74; see also U. Pooch & R. Chattergy, supra note 3, at 171-91.

\textsuperscript{175} A. Vazsonyi, supra note 5, at 104-05.


\textsuperscript{177} 17 U.S.C. § 102(a) (Supp. IV 1980).

\textsuperscript{178} For a discussion of operating systems, see W. Davis, supra note 3, at 317-23. The copyright law specifically excludes from copyright protection "any idea, procedure, process, system, method of operation, concept, principle or discovery." 17 U.S.C. § 102(b) (Supp. IV 1980).

\textsuperscript{179} See A. Vazsonyi, supra note 5, at 281.

\textsuperscript{180} Memorandum of Defendant in Opposition to Motion of Plaintiff for Preliminary Injunction Before Hearing, at 5-6, Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812 (E.D. Pa. 1982).

\textsuperscript{181} "Hardwire-Logic refers to logic designs for control or problem solutions that re-
Computer, Inc. v. Franklin Computer Corp., witnesses for the plaintiff, Apple, were unable to differentiate the expression from the idea in their copyrighted operating systems allegedly infringed upon by the defendant, Franklin. The witnesses referred to the creativity of the programmer and the creativity of the program, but not to an expression separate from the function of the program in the computer system, which would be entitled to copyright protection.

Like a computer program, a phonograph record is not "readable" by humans and is only one part of the total machinery required to produce music. The same is true of videotapes. In 1976, a major revision of the copyright laws changed much of the earlier law, which had developed in an ad hoc manner, into a unified body. In the haste to pass the bill, however, Congress neglected to resolve the matter of the protection to be accorded software. As a stopgap measure, section 117 was drafted, which provided that the law with respect to computer programs, notwithstanding the sweeping changes in the copyright law, was to remain as it stood before the new law took effect. In conjunction with section 117, Congress established the National Commission on New Technological Uses for Copyrighted Works (CONTU), which was to spend three years examining several technical areas, including computer programs, and to develop ideas for revisions of the Copyright Act.

Congress anticipated that section 117 would be changed after the CONTU final report, if it was deemed necessary. That is exactly what occurred. Subsequent amendments to the copyright laws codified CONTU's final recommendations. These amend-
ments added a definition of a computer program\textsuperscript{183} and changed section 117 so that it expressly addresses the limitations on exclusive rights in computer programs.\textsuperscript{184} The effect of these revisions is to establish that a computer program is a proper subject for copyright protection.\textsuperscript{185} In keeping with CONTU's conservative approach,\textsuperscript{186} however, the amended section 117 does not deal with the difficult policy question raised by White-Smith and Commissioner Hersey in his dissent to the CONTU final report:\textsuperscript{187} Is a computer program in a ROM, a machine part,\textsuperscript{188} functional and, therefore, not entitled to copyright protection?\textsuperscript{189}

Congress' unequivocal mandate, that computer programs are to be accepted for copyright registration, is meaningless, unless the courts give computer programs realistic protection. Computer programs imbedded in ROM's were not commercially viable when the 1976 Copyright Act was drafted. CONTU sidestepped the difficult issue of whether machine language programs, in their electrical states, are proper subjects for copyright protection, by deferring to the courts.\textsuperscript{200} Therefore, there exists no clear statement of congressional intent concerning the copyrightability of machine language programs or the extent of copyright protection for computer programs. The courts, however, may glean some indication of legislative purpose from the voluminous congressional hearing transcripts relating to computer software protection legislation, the CONTU reports, and the major statute relating to software copyright protection, the 1976 Copyright Act, including the subsequent amendments. These sources indicate a clear intent to provide software and firmware adequate and meaningful copyright protection, but also reveal an uncertainty as to the best means of achieving this objective. The courts, therefore, have been left with the responsibility of adapting the

\textsuperscript{183} CONTU's final recommendations, see CONTU Final Report, supra note 16, at 1.
\textsuperscript{184} "A 'computer program' is 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 (Supp. IV 1980).
\textsuperscript{185} Id. § 117.
\textsuperscript{186} The revisions included placing a definition of a computer program in § 101 and § 117 which dealt with the making of archived copies of computer programs. See supra note 186.
\textsuperscript{187} CONTU did not suggest major changes in the copyright law (except § 117) and deferred change to the courts. See CONTU Final Report, supra note 16, at 22-23.
\textsuperscript{188} Id. at 27-37.
\textsuperscript{189} See id. at 27 (dissent of Commissioner Hersey).
\textsuperscript{190} Id. at 37.
\textsuperscript{200} See id. at 22-23.
broad copyright protection available to literary works, to computer software and firmware.

Much of the controversy surrounding the copyrightability of software and firmware has centered on three main issues: Is a computer program the "writing" of an author;\textsuperscript{201} is a program, in machine-readable form,\textsuperscript{202} and, therefore, unintelligible to human beings, suitable for copyright protection;\textsuperscript{203} and, if there is copyright protection, what is the extent of that protection?\textsuperscript{204}

Recent articles that have addressed the question of whether a computer program is a "writing" as contemplated by the Constitution have generally concluded that it is.\textsuperscript{205} In the higher-level languages like PL-1, PASCAL, and COBOL, which closely resemble English,\textsuperscript{206} the programs can be analogized to the directions in a cookbook. Like the recipes, there can be no copyright on the individual commands (ingredients), but the creative combination of the commands (the description of the preparation of the ingredients) is protected, because it is an original writing.\textsuperscript{207} The underlying idea in each program, an algorithm,\textsuperscript{208} is not protectible.\textsuperscript{209} Likewise, in the recipe, one cannot protect the underlying idea that an apple pie, made of apples, flour, sugar and water, is one person's property, because the only way to make apple pie is by using these ingredients.\textsuperscript{210}

A problem raised by Commissioner Hersey stems from the belief that a computer program is analogous to the setting of switches between unconnected components of a digital computer.\textsuperscript{211} Commissioner Hersey believes that communication with a machine, no mat-

\textsuperscript{201} See supra note 15.

\textsuperscript{202} Machine-readable is a form in which only binary digits, understood by the computer, are used. See W. Davis, supra note 3, at 169-73.

\textsuperscript{203} See 11 BULL. COPYRIGHT Soc'y 361 (1964).

\textsuperscript{204} See supra note 158.

\textsuperscript{205} See, e.g., Gemignani, supra note 15, at 281-83; Pope & Pope, supra note 15, at 543-47.

\textsuperscript{206} See supra notes 111-13 and accompanying text.

\textsuperscript{207} See Alfred Bell & Co. v. Catalda Fine Arts, 191 F.2d 99, 102 (2d Cir. 1951).

\textsuperscript{208} An algorithm is a "prescribed set of well-defined rules or processes for the solution of a problem in a finite number of steps." W. Davis, supra note 3, at 483-84.

\textsuperscript{209} It is noteworthy that the Supreme Court, in Gottschalk v. Benson, 409 U.S. 63 (1972), held that an algorithm was not suitable for patent protection. Id. at 72-73.

\textsuperscript{210} The Ninth Circuit held that where the expression of an idea is the only way to express that idea, there can be no copyright protection, even for the expression. Herbert Renthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971) (copyright infringement of bee-shaped jewelry).

\textsuperscript{211} See CONTU FINAL REPORT, supra note 16, at 27 (dissent of Commissioner Hersey).
ter how sophisticated and useful, cannot be a writing, because computer programs lack creativity and value to human beings. This is an extremely shortsighted view of the problem, and one which ignores the considerable flexibility and creativity inherent in an effective program. The concern that the computer program has no value to people is not a valid criticism. The purpose of copyright law is not to differentiate between works of authorship intended for people to read and those intended for computers to process, but between human works of authorship that are original or creative and those that are not.

The courts have provided no clear guidance concerning the copyrightability of machine language computer programs or concerning the more general question of the scope of copyright protection for computer programs. Two cases that dealt directly with the issue of the copyright protection available to firmware yielded apparently contradictory results. In the first, Data Cash Systems, Inc. v. JS&A Group, Inc., a federal district court held that a computer program in its object phase is not a copy of the copyrighted source code computer program. The court relied heavily on the White-Smith doctrine of perceptibility to human beings. The plaintiff developed a hand-held computer chess game that was marketed in late 1977. This date is significant because the 1976 Copyright Act became effective on January 1, 1978. The court,

\[\text{References:}\]

212. See id. at 28.
213. This presents the recently arisen difficult problem concerning the creations of computers that are able to self-program to a limited extent. See Keplinger, Computer Intellectual Property Claims: Computer Software and Data Base Protection, 1977 Wash. U.L.Q. 461, 466-67.
216. See supra note 133.
217. See supra note 215.
218. 480 F. Supp. 1063 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038 (7th Cir. 1980).
219. Object phase refers to the program in machine language. See supra text accompanying notes 63-68.
220. 480 F. Supp. at 1068.
221. See id. at 1068-69.
222. Id. at 1065-66.
therefore, considered the case under the 1909 Act and held that *White-Smith* was still good law.225 The court, however, stated in dicta that the result would have been the same under the new law as well.226

The *Data Cash Systems* court differentiated between a computer program in its various forms.227 The plaintiff had claimed infringement of the ROM copy of his copyrighted program.228 The court held that computer programs in their flow chart229 and higher-level language230 forms (source programs) can be considered writings.231 In other forms not humanly readable, however, the computer program is nothing more than "a mechanical tool or machine part,"232 and, thus, not a "writing" and not entitled to copyright protection.233 In a footnote, the court detailed its reasons for believing that even under the 1976 law there would be no copyright infringement for duplication of the computer program in the ROM.234

The *Data Cash Systems* court stated that "the 1976 Act applies to computer programs in their flow chart, source and assembly phases, but not in their object phase, i.e., the ROM,"235 for two reasons. First, the ROM did not satisfy the notice requirement.236 Second, as indicated in a statement attributed to Mr. Keplinger, Assist
tant Executive Director and Senior Attorney of CONTU, the object phase of a computer program is equivalent to "a mechanical device which is . . . an essential part of the mechanical process."\textsuperscript{237}

\textit{Data Cash Systems} was apparently difficult to decide because of the timing of the first sales of the chess games (late 1977)\textsuperscript{238} and the plaintiff's manner of copyright notice.\textsuperscript{239} Had the plaintiff placed a sticker on the ROM or on the chess unit itself, the court might have been satisfied that there was sufficient notice under the Copyright Act of 1976\textsuperscript{240} to provide copyright protection against the "direct copying"\textsuperscript{241} of the program, which the court found had, in fact, occurred.\textsuperscript{242} The \textit{Data Cash Systems} case was affirmed on the ground that the 1909 Act\textsuperscript{243} was applicable\textsuperscript{244} and that under the 1909 Act, the plaintiff's actions, namely, publication without notice of copyright, caused the program to fall into the public domain.\textsuperscript{245} The lower court decision was arbitrary and poorly reasoned because it distinguished between programs in the object phase and subject phase when, in fact, this is not the relevant distinction between idea and expression. In addition, the court ignored the legislative intent of the 1976 Act to extend copyright protection to computer

\textsuperscript{237} 480 F. Supp. at 1065. This quote, however, does not represent Mr. Keplinger's beliefs, or those of CONTU, which were followed in the subsequent amendment of the copyright law, 17 U.S.C. § 117 (Supp. IV 1980), but rather, represent the views of Commissioner Hersey. CONTU FINAL REPORT, supra note 16, at 27 (dissent of Commissioner Hersey).

Commissioner Hersey's thesis is that an entirely new sort of protection should be granted to computer software. He recommended special legislation along the lines of a draft entitled "Computer Software Protection Act." For the text of this draft, see Pope & Pope, supra note 15, at 554. When Congress amended the copyright law in 1980, 17 U.S.C. § 117 (Supp. IV 1980), Commissioner Hersey's "long and thoughtful alternative proposal," Keplinger, supra note 213, at 462, was rejected and the majority decision of CONTU was approved. Compare CONTU FINAL REPORT, supra note 16 with 17 U.S.C. § 117 (Supp. IV 1980). This indicates a belief by Congress that the object program (ROM) is not only copyrightable, but is a copy of the source program.

\textsuperscript{238} 480 F. Supp. at 1066. The sales took place before the effective date of the 1976 Copyright Act, January 1, 1978. 17 U.S.C. § 301(a) (1976).

\textsuperscript{239} See 480 F. Supp. at 1066 n.2 (plaintiff affixed copyright notice to all printouts of program).

\textsuperscript{240} See id. at 1067 n.4 (quoting Proposed Regulation § 201.20(g)(4) (codified at 37 C.F.R. § 201.20 (1981))).

\textsuperscript{241} Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063, 1066 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038 (7th Cir. 1980).

\textsuperscript{242} Id.


\textsuperscript{244} 628 F.2d at 1042.

\textsuperscript{245} See id. at 1043.
programs.\textsuperscript{246} A more recent statement concerning the copyrightability of ROM's is articulated in \textit{Tandy Corp. v. Personal Micro Computers, Inc.}\textsuperscript{247} The court in \textit{Tandy} held that under the 1976 Act,\textsuperscript{248} copying a ROM constituted an infringement of the plaintiff's copyright in the underlying program.\textsuperscript{249}

The plaintiff in \textit{Tandy} manufactured a home computer\textsuperscript{250} and developed an "input-output routine"\textsuperscript{251} which translated information from a high-level programming language into a machine-readable language.\textsuperscript{252} This input-output routine is an example of a compiler.\textsuperscript{253} The program was imprinted directly onto a ROM chip made of silicon, which was then permanently wired into the computer.\textsuperscript{254} The defendant, also a producer of computers designed for home use, was alleged to have copied the plaintiff's ROM, only excising those items which specifically identified the program as that of the plaintiff.\textsuperscript{255}

Again, the timing of the lawsuit was important. The court held that the revised section 117\textsuperscript{256} did not apply to this case and that the 1976 Copyright Act was controlling.\textsuperscript{257} It held that the restrictions imposed by section 117 of the 1976 Act\textsuperscript{258} did not apply to "sections 101 and 102 of the act, which . . . clearly allows a program in this


\textsuperscript{249} 524 F. Supp. at 174-75. The court's holding was in the context of the defendant's motion to dismiss before trial. \textit{Id.} at 172-73.

\textsuperscript{250} \textit{Id.} at 173. The case dealt with the TRS-80 Computer. \textit{See} Newsday, Dec. 1, 1981, at 25. The computer is advertised by appeals to "Give your child a brighter future this Christmas," and lists the following as its attributes: "Helps with Reading, Math, Science, and Other Subjects . . . Huge Library Of Software . . . Includes Intro To Programming . . . Play Exciting Games, Too." The advertised price is $999. \textit{Id.}

\textsuperscript{251} 524 F. Supp. at 173.

\textsuperscript{252} \textit{Id.}

\textsuperscript{253} A compiler is a program which "prepare[s] a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one machine instruction for each symbolic statement, or both, as well as performing the function of an assembler." W. \textit{Davis}, supra note 3, at 486.

\textsuperscript{254} 524 F. Supp. at 173.

\textsuperscript{255} \textit{Id.}

\textsuperscript{256} 17 U.S.C. § 117 (Supp. IV 1980).

\textsuperscript{257} \textit{See} 524 F. Supp. at 173.

form [ROM] to be copyrighted and protected.\textsuperscript{259} The \textit{Tandy} court refused to follow \textit{Data Cash Systems} to the extent that the earlier case held, if only in dicta, that a ROM is not a copy of the copyrighted source code computer program. The court also disagreed with \textit{Data Cash Systems} by holding that a copy of a ROM is an infringement of the copyright in the underlying computer program.\textsuperscript{260} The \textit{Tandy} court analysis began by looking to the 1976 Act as the applicable law.\textsuperscript{261} On the basis of revised sections 101\textsuperscript{262} and 102,\textsuperscript{263} the court was convinced that . . . (1) a computer program is a "work of authorship" subject to copyright, and (2) that a silicon chip [the form a ROM takes] is [a] "tangible medium of expression," within the meaning of the statute such as to make a program fixed in that form subject to the copyright laws.\textsuperscript{264}

The court reached its decision by examining the legislative history of the 1976 Act concerning computer programs.\textsuperscript{265} First, with respect to "works of authorship," the \textit{Tandy} court pointed out that the legislative history strongly indicated that "Congress understood that computer programs were subject to copyright protection under the law as it existed prior to the 1976 Act, as well as under the new statute."\textsuperscript{266} Next, the court looked to section 102(a)\textsuperscript{267} which referred to works "fixed in any tangible medium . . . from which they can be perceived, reproduced or otherwise communicated . . . with the aid of a machine or device."\textsuperscript{268} In reviewing the legislative history of section 102(a), the court found a clear statement of congressional intent:

\begin{quote}
Under the bill it makes no difference what the form, manner or medium of fixation may be—whether it is in words, numbers, notes, sounds, pictures or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or other stable form, and
\end{quote}
whether it is capable of perception directly or by means of any machine or device "now known or later developed."\textsuperscript{269}

The \textit{Tandy} court examined section 117 of the 1976 Act\textsuperscript{270} and determined that its language only addressed the "problems surrounding the input into computers of properly obtained copyrighted materials."\textsuperscript{271} The language of the section refers to the "use of the work \textit{in conjunction with}"\textsuperscript{272} computers and not the \textit{duplication} of a copyrighted work.\textsuperscript{273} The court concluded that Congress did not intend to provide such a large loophole which would permit the copying of a copyrighted program on a silicon chip (ROM).\textsuperscript{274}

Before concluding that there could be copyright infringement for the duplication of ROM's imprinted with a copyrighted program, the court additionally noted that it was possible that the defendant printed out the contents of the ROM and used the program to imprint the silicon chip.\textsuperscript{275} This alternative basis for the court's conclusion weakens the ruling that all programs on ROM's are copyrightable, because a copy of the higher-language source program would be an infringement of the copyright in the plaintiff's program. Even a possibility of this activity should be sufficient to defeat a motion to dismiss. The court's analysis of the copyright problem was clear, yet it was also limited. The court did not go so far as to provide, or even suggest, the availability of effective protection by granting a preliminary injunction to prevent the defendant from continuing to use the allegedly infringing program.

The significance of the \textit{Tandy} decision is more evident when one notes that the forum was the Northern District of California.\textsuperscript{276} The jurisdiction of this court includes Silicon Valley,\textsuperscript{277} the location of a...

\begin{footnotes}
\item[270] 524 F. Supp. 174-75.
\item[271] \textit{Id.} at 174.
\item[273] 524 F. Supp. at 175.
\item[274] \textit{Id.}
\item[275] \textit{Id.}
\item[276] \textit{Id.} at 171.
\item[277] Silicon Valley is, in reality, the Santa Clara Valley of Northern California, located just south of San Francisco. It has been nicknamed Silicon Valley because of the great concentration of semiconductor industries located there. Silicon is the basic component of virtually all semiconductors due to its unique chemical and electrical properties. \textit{See Stonier, supra} note 48, at 111.
\end{footnotes}
large percentage of the American semiconductor\textsuperscript{278} industry.\textsuperscript{279} A court in this particular geographic area, aware of the economic issues and capable of handling the technical problems inherent in this application of the copyright law, understandably would strive toward effective protection of this new work of authorship.

Since the \textit{Data Cash Systems} and \textit{Tandy} decisions, the computer industry has continued to grow.\textsuperscript{280} This growth has caused more friction and new cases have been litigated in this and related areas.\textsuperscript{281} \textit{Williams Electr}onics \textit{v. Artic International, Inc.}\textsuperscript{282} and \textit{Apple Computer, Inc. v. Franklin Computer Corp.}\textsuperscript{283} have been particularly relevant to copyright protection for computer programs in ROM's.

In the \textit{Williams} case, the Third Circuit held that the plaintiff Williams Electronics' Defender video game, a copyrighted computer program stored in a ROM, was infringed by a blatant copy.\textsuperscript{284} The \textit{Williams} court accepted with approval the reasoning of the \textit{Tandy} court and rejected, rather harshly, the dicta in \textit{Data Cash Systems}.\textsuperscript{285} Looking first to the copyright law,\textsuperscript{286} then to the legislative history\textsuperscript{287} and the \textit{CONTU} Report,\textsuperscript{288} the \textit{Williams} court made several well reasoned holdings. First, the court held that there is no difference, for legal purposes, between a program in source code and object code.\textsuperscript{289} Second, the court held that Congress did not intend to provide a loophole for infringers and that a program in ROM form is

\textsuperscript{278} See supra notes 48, 277.

\textsuperscript{279} There are at least 786 electronics firms in Silicon Valley, the “most explosive area for growth companies,” which “produced $8.7 billion worth of goods” in 1980. Taylor, \textit{Striking it Rich}, \textit{Time}, Feb. 15, 1982, at 36, 38.

\textsuperscript{280} See infra notes 301-11 and accompanying text.


\textsuperscript{282} 685 F.2d 870 (3d Cir. 1982).

\textsuperscript{283} 545 F. Supp. 812 (E.D. Pa. 1982).

\textsuperscript{284} \textit{Williams}, 685 F.2d at 877.

\textsuperscript{285} \textit{Id.}

\textsuperscript{286} \textit{Id.} at 873-74.

\textsuperscript{287} \textit{Id.} at 875.

\textsuperscript{288} \textit{Id.}

\textsuperscript{289} \textit{Id.} at 876-77.
a copy of the copyrighted source code program. Finally, the court held that an injunction was a proper remedy. The Williams court, following the lead of the Tandy court and various other decisions dealing with copyrights in video games, converted the obvious congressional intent to provide copyright protection for computer programs into a reality.

The second recent case, Apple Computer, Inc. v. Franklin Computer Corp., held, on a motion for preliminary injunction by the copyright owner, Apple, that the operating system programs, unlike applications programs, are possibly functional and that under section 102(b) of the copyright law, would not be entitled to copyright protection. In Apple, Franklin successfully argued that Apple was seeking to protect under the copyright law what should properly have been patented. The court denied the motion for a preliminary injunction. It questioned the possibility of separating the expression from the idea in an operating system program. The court correctly looked to the substance of the operating system, taking care not to be confused by the form it took—that of a computer program in a ROM. This court's sensitivity to copyright issues, in the face of an extremely technical factual situation, bodes well for proper and effective protection of statutorily protectible computer programs embedded in ROM's.

Starting with Tandy and continuing with Williams and Apple, a favorable trend toward effective protection of computer programs appears underway. A major remedy still needed is rapid preliminary injunctive relief in favor of a copyright holder.

**FUTURE DIRECTIONS**

As the computer industries—hardware, software, and

290. Id. at 877.
291. Id. at 878.
292. See supra note 281.
293. Williams, 685 F.2d at 877.
297. Id. at 824.
298. Id. at 825.
299. Id. at 821.
300. Id. at 815-21.
301. See supra note 7.
302. Software means computer program.
firmware—continue to develop, their products will become smaller, faster and less expensive. This will increase the demand for consumer applications for computer products.

The consumer of the not too distant future could be roused from sleep by a recording of his voice telling him to get out of bed or else he will miss his train. Then, while still in bed, the same consumer might be able to set the temperature in his shower, turn on the heat in the halls of his house, and start his breakfast. The technological basis for all of this electronic gadgetry exists today. Its use depends on whether the electronics industry can reduce the cost by developing a market large enough to justify full scale production. In the next few years, though, the market for home computer software and, especially, firmware will expand greatly. The growth of the Apple Computer Company serves as a good example of the potential in this field. Apple expanded from a basement business to a major corporation, with sales in excess of $600 million, in the span of only a few years. As more and more home and small business computers are purchased or leased, the demand for more programs to utilize the existing hardware will grow. Especially in the case of personal computers, needs of users will be homogeneous enough to provide software and firmware companies with the incentive to develop newer and more elaborate software packages for a broad market.

As the computer industry grows, the law must adapt to meet the changing needs of the American consumer and the industry it-

303. See supra note 133.
304. See supra note 4.
305. I. Barron & R. Curnow, supra note 130, at 105-06.
309. See id.
310. There is a large difference between the needs of small and large corporations. Large corporations generally require large systems that fit their infrastructure and well defined needs. These needs are not necessarily similar to those of other large corporations, due to the diversity in management techniques, corporate structure and existing computing facilities. With small businesses, however, it is simpler to adjust internal operations to fit standardized bookkeeping, accounts receivable, accounts payable, billing and inventory software packages. Within this group having homogeneous needs, small businesses can be grouped with in-the-home consumers. See N.Y. Times, Nov. 8, 1981, § 3, at 1, col. 2.
311. A software package is a collection of computer programs that is designed to perform particular tasks. It is usually sold with instruction manuals. Examples include video game cartridges, the Tandy case's input-output routine, and the Data Cash Systems program.
self. The hardware industry is protected by the patent laws.\textsuperscript{312} The ever-changing firmware and software industries, however, are not adequately protected. At present, most proprietary programs\textsuperscript{313} are not protected by copyright.\textsuperscript{314} This is, in part, a result of the difficulties of discovering and proving infringement of a copyrighted program, where the infringing article is in a different form than the copyrighted program. Finally, computer programs straddle, in many cases, the dividing line between process,\textsuperscript{315} which may be protected by patent,\textsuperscript{316} and the expression of that process, which may be protected by copyright.\textsuperscript{317}

Before determining the proper role of copyright law with respect to the protection of computer programs, it is instructive to reexamine the goals of the copyright and patent laws as expressed in the Constitution.\textsuperscript{318} The Constitution gives Congress the power "[t]o 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."\textsuperscript{319} The two key concepts that are embodied in the Constitution are the monopoly for the author-inventor and the limited duration of the monopoly. The balanced policy considerations behind the constitutional language provide society, on one hand, the opportunity to share technological information, which allows us "to stand on the shoulders of giants"\textsuperscript{320} to advance technology and understanding. On the other hand, it allows the author-inventor to be adequately compensated for his work. Since the 1976 Copyright Act,\textsuperscript{321} the period of copyright protection has been extended to the author's life plus fifty years.\textsuperscript{322} The corresponding

\begin{itemize}
\item \textsuperscript{313} See supra note 29.
\item \textsuperscript{314} See N.Y. Times, Nov. 8, 1981, § 3, at 28, col. 1.
\item \textsuperscript{315} "The term 'process' means process, art or method and includes a new use of a known process, machine, manufacture, composition of matter, or material." 35 U.S.C. § 100(b) (1976).
\item \textsuperscript{316} Id. § 101.
\item \textsuperscript{317} 17 U.S.C. § 102 (Supp. IV 1980).
\item \textsuperscript{318} U.S. Const. art. I, § 8, cl. 8.
\item \textsuperscript{319} Id.
\item \textsuperscript{320} See Letter from Sir Isaac Newton to Robert Hooke (Feb. 5, 1675), quoted in J. Bartlett, Bartlett's Familiar Quotations 379b (14th ed. 1968).
\item \textsuperscript{322} An "author," as defined by the statute, is either the creator or, in most corporate cases, the corporation. 17 U.S.C. § 201 (Supp. IV 1980).
\item \textsuperscript{323} The life-of-the-author-plus-50-years period applies to works created on or after January 1, 1978. 17 U.S.C. § 302(a) (Supp. IV 1980). If the author is a corporation, the
\end{itemize}
period for patents is seventeen years.\textsuperscript{324} To provide adequate protection for proprietary software and firmware, the law must be able to protect more than just direct copying. This is necessary because of the ease of substitution of a handful of program statements without changing the manner in which a program operates. As the copyright law has been able to protect the plots,\textsuperscript{325} themes\textsuperscript{326} and characters\textsuperscript{327} of conventional authors, so the law must protect against more than a rote copy of the copyrighted program.

There is a constraint on this approach—the programs cannot be protected at the algorithm level. The Supreme Court recently held that an algorithm cannot be protected by a patent, because it is an idea.\textsuperscript{328} An algorithm is not copyrightable because only the expression of the idea, and not the bare idea, is copyrightable.\textsuperscript{329} It is difficult to accurately draw the line between an algorithm and a protected expression in the computer program area. This results from the difficulty in defining an algorithm and the algorithms which computer programs incorporate. A higher-level language statement could reasonably be assumed to constitute the equivalent of a concept, whereas the machine language statements that compose the higher-level instruction might be considered an algorithm. An algorithm could, however, also be a series of higher-level instructions.

There are several basic problems with the current copyright law as applied to computer software and, especially, firmware. First, the period of copyright protection, the author's life plus fifty years,\textsuperscript{330} is so long that, for most practical purposes, the protection is "forever." Second, it is difficult to police the copyright, because copying may be carried out in private corporate factories and the ROM is wired inside a computer system. A third problem is the extent of copyright protection that will be given to firmware. This problem comprises two related issues: At what level may a computer program be protected and what degree of similarity is required for there to be ac-

\begin{footnotesize}
\begin{itemize}
\item[324.] See 35 U.S.C. § 302(c) (Supp. IV 1980).
\item[326.] See Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930), cert. denied, 282 U.S. 902 (1931) (dicta).
\item[327.] See Reyher v. Children's Television Workshop, 533 F.2d 87 (2d Cir. 1976), cert. denied, 429 U.S. 980 (1976) (dicta).
\item[328.] See King Features Syndicate v. Fleischer, 299 F. 533 (2d Cir. 1924).
\end{itemize}
\end{footnotesize}
tionable infringement?

To appreciate the significance of fifty-year copyright protection in the computer industry, it is instructive to remember that the first programmable computer was built less than forty years ago. If a copyright had been granted on a program written in 1945 and the author (programmer) died in 1980, the copyright would be in force until 2030.

In our technological age many products become obsolete almost before they come to market. This is especially true of the computer industry, where the lead time between an idea and the final product can be a period of years, during which time new ideas may make earlier ideas outdated. It is wasteful to grant copyright protection for periods of fifty, seventy-five, or more years to a product that will be obsolete within just a few years.

The period of protection for a patent is seventeen years. This term was intended to strike a balance between the incentive for an inventor to publicize his or her invention and the American antithema for monopolies. It is assumed that in seventeen years the inventor can earn sufficient profit to make it worthwhile to reveal his or her invention. Copyright protection has a similar motivation and represents a balance between the proprietary right of the author and the interests of the general public.

In the software and firmware industries, the market interest in a product is of brief duration. It is estimated that a video game is popular for only six months, after which time the market dissipates. To give a monopoly for more than fifty years on a prod-

331. The Harvard Mark I, built in 1944, was the first program-controlled computer. M. CAMPBELL-MARTIN, supra note 38, at 114.
332. It would not have been possible to obtain copyright protection for a computer program in 1944 under the 1909 Copyright Act, 17 U.S.C. § 5 (1909) (repealed January 1, 1978), but it is illustrative of the length of copyright protection. Cf. White-Smith Music Publishing Co. v. Apollo Co., 209 U.S. 1 (1908).
333. See C. EVANS, supra note 4, at 104 (advances in computer field are made practically monthly).
334. See supra note 308.
335. See supra note 333.
340. See id.
uct that is only valuable for six months is both wasteful and a bad bargain on behalf of the American public. Although the video games market is particularly active, the entire software and firmware industry is marked by this rapid obsolescence of products. Therefore, the copyright law, with respect to software and firmware, should be amended to shorten the period of protection. The shortened term should be on the order of five years, which would be sufficient to permit a return on capital and profit.

The shortened term would serve several important purposes. First, it would bring the copyright law into line with the reality of this peculiar industry. Second, it would prompt the courts to more readily find infringement. This would be so because such a finding would not be inhibited by the fear that it would shackle development for long periods of time. Furthermore, the harm to the copyright owner—which would accrue over a shorter period of time—would be easier to calculate. Third, it would provide more of a disincentive to infringement, because the copyright would fall into the public domain within five years. Fourth, the author would be more likely to apply for copyright protection if he or she believed that the courts would effectively enforce the law.

The problem of policing the copyright is a great one, particularly for small companies that cannot afford to have staffs available to continuously monitor their competitors. Small companies have similar problems enforcing their patents. The policing of process patents is especially difficult because it is the method of production and not the final product which is protected. There are, for example, often several ways to produce a certain chemical or composition of matter. One, or several, may be in the public domain. Therefore, the patent owner cannot be certain that the process being performed behind locked doors infringes upon its patent. Once a patent suit is filed, through discovery, the patent owner can ascertain if the process used infringes the owner's patent. This is a very expensive procedure. Costs and attorneys' fees can be taxed to the patent owner if it is determined that there was no infringement and that the

341. See supra note 323.
342. See Skow, supra note 31, at 58.
343. See C. Evans, supra note 4, at 73; supra note 333 and accompanying text.
344. When property is in the public domain, all are free to make and sell copies of it. B. Kaplan & R. Brown, Cases on Copyright 2 (3d ed. 1978).
plaintiff acted in bad faith in initiating the suit. In fact, a patent infringement suit, including reasonable discovery and an appeal, can cost between $500,000 and $1,000,000.

It is equally as difficult to discover the infringement of a software or firmware copyright. The only way to do so is for the copyright holder to obtain a copy of the possibly infringing program and compare it with the copyrighted program. Therefore, infringement cannot be ascertained until the copy has come to market and the copyright holder’s sales have been diminished. Another difficulty is that a ROM may be part of an entire computer system and, therefore, the copyright owner must purchase the whole system and dismantle it in order to determine if the ROM’s copyright has been infringed. If only a few new computers were introduced each year, the cost might not be prohibitive. There are, however, literally thousands of new computers of all sizes, shapes and complexities entering the market, with price tags ranging from $100 to the millions of dollars. Therefore, a small software company may not have the resources to effectively police its copyright.

If a copyright holder has sufficient reason to believe that his software or firmware has been pirated, the court should order the alleged infringing party to submit its program to a court-appointed master, who could then determine the likelihood that an infringement had occurred. If the master were to determine that infringement was likely, the court could enjoin the alleged infringing party from further sales of the offending program, until resolution of the case. If, however, the master were to find it unlikely that there was an infringement of the copyrighted program, the copyright

350. It is well within the court’s power to provide for a master to make determinations of fact in areas that are outside of the court’s expertise. See Wisconsin v. Illinois, 449 U.S. 48 (1980); Texas v. Environmental Protection Agency, 499 F.2d 289 (5th Cir. 1974), cert. denied, 427 U.S. 905 (1976); 28 U.S.C. § 53(b) (1976).
holder could proceed with the litigation, but without a preliminary injunction preventing the other party from manufacturing or selling the allegedly infringing software or firmware.

Invoking an injunction when there is an initial determination of a likely copy would effectively protect a copyright holder from an infringement occurring early in the program's effective life, when most of the sales and profits are made. This procedure is far more efficient than determining infringement after several years and then trying to calculate damages which correspond to the infringing party's profits.352

The selection of a master would not be too difficult, as there are many qualified programmers in the marketplace. The most objective masters, however, would be academics from universities who have no ties to either of the parties. The court could send copies of the two programs and the court's instructions to the master, and the work could be done wherever he or she works. The cost of the master's services could be taxed to the plaintiff, passing to the defendant only if the master determines there is likely infringement or if the plaintiff prevails in the lawsuit. This could serve to deter frivolous suits by plaintiffs.

Once the procedural system for litigating copyright infringement is settled, the more important substantive question—what constitutes copyright infringement—must be explored. This question can be divided into two related issues: the level at which a computer program may be protected and the degree of similarity required for infringement.353 The answers to these questions will form the instructions the court should give to the master for his or her preliminary determination.

The first issue concerns the language-level and number of statements at that language-level. The lowest level is the machine language statement that corresponds to one command to the computer.354 The next level is at the higher-level language statement, which is composed of five to ten machine language statements.355 The highest level would be the routine level, which consists of a series of higher-level language statements that perform certain tasks.356

354. See supra note 103.
355. See supra note 11, at 70.
356. See infra note 359.
It would not be feasible to protect individual machine language statements. First, every program contains the same limited number of machine language statements. If permitted, the manufacturer of the hardware would copyright the statements and sell them with the hardware. Second, there is nothing original contributed by the programmer using a machine language instruction. It is also not feasible to copyright individual high-level language statements. When working in a particular language (e.g., PASCAL, FORTRAN or BASIC), the instruction set is predetermined and there is nothing original contributed by the programmer.

The next level is comprised of a collection of high-level instructions that perform a specified task or set of tasks. This can be called a routine. An example of a routine is a series of instructions that takes in data and plots it on a graph. A program, especially one which can realistically be sold in the marketplace, is often composed of many routines.

One can analogize the three levels as follows: A machine language instruction corresponds to a letter, a high-level language statement corresponds to a word, and a routine corresponds to sentences and paragraphs. Viewed in this light, copyright protection should exist only at the level of routines. Viewed from the point of view of the programmer, this analogy and result is realistic. The programmer builds "sentences and paragraphs" (subroutines and routines) out of the vocabulary words (high-level language statements) he or she has available. Each routine or subroutine is the expression of an idea, and it is that expression which copyright law protects. Therefore,

357. Although the machine language instructions are predetermined, there is much room for programming creativity. The creativity, however, is on the algorithm level. For a discussion of several programming techniques, see N. Graham, supra note 11, at 80-90. See generally U. Pooch & R. Chattergy, supra note 3, at 214-29.

358. For a discussion of several programming languages, see W. Davis, supra note 3, at 179-94; U. Pooch & R. Chattergy, supra note 3, at 334-42.

359. "Routine" refers to a series of instructions in the main program which perform a specific task or tasks. See N. Graham, supra note 11, at 90.

360. For a discussion of the types of routines included in programs, see J. Vles, Computer Fundamentals for Non-Specialists 130-33 (1981).

361. For the programmer who writes in a high-level language, the basic building blocks are the high-level language instructions that must be strung together in a meaningful way. See A. Vazsonyi, supra note 5, at 341-43. See generally id. at 341-445 (detailed description of computer programming, FORTRAN, COBOL, and BASIC computer languages). The terms "tree structure" and "hierarchical data structure" are used to define the directed relationships created by the programmer. See id. at 341.

the copyright law should protect software at this level.\textsuperscript{363}

The second area of inquiry bearing on the determination of copyright infringement is the degree of similarity necessary to find infringement. This is a particularly difficult question and no single clear answer for all applications is possible. In the procedural treatment proposed, with a preliminary determination to be made by a court-appointed master, a slightly narrower standard should be applied by the master than by the court.

The purpose of the master is to speedily determine if it is likely that there was infringement, so that a preliminary injunction can issue to effectively aid the plaintiff. At this stage, it is justified that a blatantly infringing party, one who exactly or almost exactly copied the software or firmware, is stopped. In the interest of fairness, however, close cases ought to be denied preliminary injunctive relief. The fact that a preliminary injunction is denied would not be evidence that there was no infringement. Rather, it would merely indicate that the question, at best, is close and that the case must be tried to completion before any injunctive relief will be granted.\textsuperscript{364}

The master, in his or her evaluation of the two programs, should look to a variety of factors which bear on the question of whether copyright infringement has likely occurred. First, he or she should determine whether the second program is an exact copy of the first\textsuperscript{365} or merely a translation of the first from one programming language to another.\textsuperscript{366} Assuming that the second program is not the same or a translation, the master should determine if large portions (routines) of the first program are found in the second. The final test should be a comparison of the structure and ordering of the two programs. The master should determine if there is a consistent paraphrasing of the first program by the second. If the master believes that any of the above tests are met, he or she should advise the court that it is likely that there is copyright infringement and indicate the basis for finding so. If the master does not believe it likely that there is copyright infringement, i.e., none of the tests are positive, the

\textsuperscript{363} It should be noted that programs are often written in machine language and, therefore, routines could be sequences of machine language statements.

\textsuperscript{364} See Fed. R. Civ. P. 65(a).

\textsuperscript{365} An exact copy would violate the copyright owner’s rights under 17 U.S.C. § 106(1) (Supp. IV 1980). If a second programmer, however, independently developed an exact copy of the program, there is no infringement. This is extremely unlikely. See Tandy Corp. v. Personal Micro Computers, Inc., 524 F. Supp. 171 (N.D. Cal. 1981).

\textsuperscript{366} This is a violation of 17 U.S.C. § 106(b) (Supp. IV. 1980), since a translation is a derivative work. For a definition of “derivative work,” see 17 U.S.C. § 101 (Supp IV 1980).
master should indicate the degree of similarity that was found.

Many cases will end at this stage of the litigation, whichever conclusion the master reaches. This would be due simply to the economic realities of the software and, especially, firmware industries. If the preliminary injunction is granted, then it will not be economically feasible for the alleged infringing party to continue to contest the lawsuit, especially if it knows that it is infringing the plaintiff's copyright. This is understandable, given the brief average economic life of a firmware package. If the preliminary injunction is denied, then, unless the plaintiff is certain that it will prevail at trial, it is not economically sensible to continue the lawsuit.

There are several additional items that the parties should make available to the court at trial to aid in the final determination of copyright infringement. Both parties, in addition to submitting source code copies of their programs, should also provide all supporting documents, including any papers regarding the development of the program. The parties should, additionally, provide documents tending to substantiate the period of time necessary to develop the program. This could provide circumstantial evidence of the defendant's individual effort.

The form of the court's analysis should depend on the master's preliminary decision. If the preliminary injunction was granted, the burden of proving noninfringement should be placed on the defendant. The court would review the defendant's supporting documentation in order to determine whether a realistic amount of time was spent in developing the program in question. If the alleged infringing party was able to market a firmware package strikingly similar to the plaintiff's, within a suspiciously short period of time following the initial marketing of the plaintiff's package, without documentation to prove an independent design effort, the court should find copyright infringement.

If the master preliminarily did not find copyright infringement likely, but the case progressed to trial, the plaintiff should bear the burden of proving that the defendant infringed the plaintiff's copyright. The court, in this instance, would be faced with a very difficult "techno-legal" decision. Unfortunately, most federal judges lack

---

367. See supra note 343.
368. See supra note 347.
369. This word is used to indicate a decision that is both technical and legal.
appropriate technical training and are understandably anxious not to "get stuck" with a case requiring technical expertise. Therefore, both parties should retain expert witnesses to submit reports, written so that the judge, in his or her chambers, can, without embarrassment, examine the technical data addressing the similarities and differences between the two programs. These reports should also indicate the experts' opinions on the time generally required to develop a program of the length and complexity of the program in question. The court should then examine all of the reports submitted, i.e., the master's and those of the parties' experts, to determine if, under general copyright principles, there is infringement.

The cases that have come to court to date have involved direct copying of firmware and software. Were the system proposed in this note adopted, this type of infringement would be deterred. If a potential infringer knows that an exact or near exact copy of a profitable firmware package will not be economically worthwhile, it will be far less likely to develop and market such a copy.

The deterrent imposed by the preliminary injunction system will not, however, prevent a company intent on duplicating a successful firmware package. Such a company could analyze the successful program and redesign it with sufficient dissimilarities to avoid the preliminary injunction and, possibly, a judgment in favor of the copyright holder. In such a case, the courts will have to develop means of analysis which will allow them to distinguish between copyright infringement and an original work. The issues that will be involved are the unprotectability of pure ideas and the originality of the contributions of the second programmer. The courts have not yet been called upon to address these very subtle issues and their treatment


373. The copyright law excludes from protection any idea in a work of authorship. 17 U.S.C. § 102(b) (Supp. IV 1980).

374. Originality is necessary for copyright protection to be accorded a work of authorship. 17 U.S.C. § 102(a) (Supp. IV 1980).

375. To date, all of the cases dealing with infringement of copyrights in computer programs have been clear examples of copying. See supra note 372. In the future, if a complete copy is held to be an infringement, potential copiers will be forced to make substantial revisions to the programs they wish to market. If this occurs, the courts will have to examine the
will depend upon the courts' resolution of more obvious infringement issues.

CONCLUSION

The copyright system is the most appropriate method to protect proprietary software and firmware. The present copyright act provides the means to achieve adequate protection. The responsibility rests upon liberal and farsighted courts, such as the district court in *Tandy Corp. v. Personal Micro Computers, Inc.*, to effect realistic and equitable protection for computer programs in all of their forms, including ROM's. The courts must be more liberal in granting preliminary injunctions to allow copyright owners to police their programs effectively. Finally, the term of copyright protection for computer programs must be shortened considerably.

As our technological society continues its headlong dash toward the future, the computer will become more of a necessity than merely an aid. For the computer to continue its growth and assist people at home and in small businesses, proprietary computer programs must be protected so as to provide sufficient economic incentives for continued development. Accordingly, the courts must provide effective copyright protection for computer programs in ROM's.

*Peter D. Aufrichtig*

---

underlying reasons for protecting computer programs and whether, and to what extent, a monopoly has been created by granting copyright protection to computer programs.

376. *See supra* notes 15-20 and accompanying text.