

Computer Software and Copyright

CALVIN N. MOOERS

Rockford Research, Inc., Cambridge, Massachusetts 02138



The method of copyright, although neglected in the past, is a method to consider for legal protection of commercial computer software and programs. Copyright is particularly directed to written text, and it is therefore applicable to the text form or listing of computer programs and software, and to all derivatives thereof. When the simple formal requirements for copyright have been met, it immediately provides claim to exclusive rights in the published text. Unauthorized copying, translation, performance, or making derivative versions, of a copyrighted text then constitutes a copyright infringement with legal penalties. Included among the things protectable by copyright are: computer programs, documentation and diagrams, the meticulous details of a debugged program, program translation such as from FORTRAN to ALGOL, compiling of programs, running a program, software systems and their command languages, languages and their compilers, and the microprogramming required for a computer to mimic a defined instructional set. Copyright may be more suitable than patent or trade secret for commercial protection of most typical programs and software. Copyright of computer software may speed the achievement of program portability, and standardization of software systems. Utilization of, and acceptance of, copyright protection in the industry is expected to increase as computer professionals learn of its features and advantages.

Keywords and Phrases: property rights in software, software copyright protection, program copyright, computer languages and copyright, patents.

CR Categories: 2.3, 4.0, 4.5

INTRODUCTION

Traditional literary copyright has quietly become an important form of legal protection for valuable computer software. Such protection is needed in order that software may be readily traded and marketed in business and commerce. An indication of need for such protection is given by the fact that the current investment in new software development exceeds that for new hardware development, and is expected to become increasingly more important in the future.

It is the purpose of this paper to provide a survey and tutorial of what copyright is,

how it applies to software and programs, and how it affects the practice of our profession. In particular, it will show the reader how copyright can be employed in various ways for the protection of property rights in software creations. It will also outline the obligations under copyright that we all have in regard to the property rights of others. An unexpected by-product of this survey is the discovery of at least two important unsuspected areas of application of copyright to computer technology.

Some of the salient observations of this study concerning software and copyright are the following: A computer program, when

Copyright © 1975, Association for Computing Machinery, Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that ACM's copyright notice is given and that reference is made to this publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery.

CONTENTS

INTRODUCTION

WHAT IS COPYRIGHT?

- What is Not Protected by Copyright?
- How is Copyright Infringed?
- The Penalties Are Severe

COPYRIGHT APPLICATIONS TO SOFTWARE

- Software Documentation
- Protection of Programs
- Writing Similar Programs
- Debugging Protected
- Systems, Compilers and Languages
- Computer Families

HOW COPYRIGHT AFFECTS THE COMPUTING COMMUNITY

- Why Was Copyright Neglected?
- Copyright is Being Accepted
- Program Portability and Standardization
- Are These Conclusions Valid?
- Strategy for the Copyright Owner
- Journal Publication

CONCLUDING REMARKS

REFERENCES

of copying, translation, and the like, for each step of execution. Copyright protection is particularly suitable for the protection of the "investment in debugging" of a well-tested, finished program. Copyright indirectly protects property rights in programming languages and their compilers, because a text description which defines a programming language is copyrightable, and any translation of such a defining text into a derived text which is a computer program for a compiler for that language may be an infringement. In a similar sense, computer "families" may be protected indirectly by copyright (when desired), since computer families are characterized by a common instructional set (e.g., the IBM 360 and 370 series) and copyright of the text defining an instructional set also appears to give rights to the making of derived texts which are microprograms for carrying out the individual instructions.

In comparison to the patent method of protection of software, copyright has the great advantage in that it is very inexpensive, is much easier and more rapid to secure, and is applicable to almost all computer programs and software, rather than to just the tiny fraction of programs which meet the requirement of embodying a patentable new concept.

In comparison to the trade secret method of protection, copyright provides legal rights even after the software has been sold to many customers, and is in widespread use. The trade secret method requires close maintenance of secrecy. Strictly viewed, the trade secret method is meaningful when used by only one, or a very few companies. Trade secret status may be lost by general marketing of the secret.

One possible disadvantage to the copyright method of protection is that absolute "secrecy" of any new software method is lost, since copyright requires "publication" (as through sale or lease) of the software. However, as the following discussion will show, really new methods and ideas in software seldom occur, while copyright may provide a wide scope of protection for the actual text of computer software after publication. In addition, copyright may be ad-

considered in the form of the text of the source language listing (such as in FORTRAN), is a "literary work" in the eyes of the Copyright Law. Such a literary work has property attributes under the law, and the owner of such a literary property can secure a variety of "exclusive rights" by means of the Copyright Law. A copyrighted work is protected against a variety of unauthorized acts, such as copying, translation, the making of versions from it, as well as performance or execution of it in a computer. Copying includes photocopying as well as tape-to-tape copying. Translation includes the translation of a program to another computer language, such as from FORTRAN to ALGOL. The making of other versions includes the assembly or compilation of the text of the source program. Performance of the program entails many kinds

vantageously used in conjunction with software licenses requiring nondisclosure of the texts or tapes of programs, as is practiced by IBM with their licensed program products.

Another disadvantage of the copyright manner of protection is that judicial enforcement of certain parts of copyright is currently in a puzzling state.

The analysis of copyright in this paper is based on a professional understanding of the nature of computers and of their programmed operation. For this reason, the analysis is qualitatively different from that of previous discussions of the subject, which have been written by copyright and patent lawyers. It is my feeling that these previous studies have suffered because their authors have attempted to apply the notions of copyright to an understanding of computer programming derived secondhand from computer professionals. As we all know, computer professionals can give quite varied and misleading answers when questioned philosophically about "what a program really is." The resulting inadequate understanding of computer software has had several undesirable effects. It has resulted in several important areas being overlooked, or in being considered only superficially. It has also led to an unfortunate belief in, or overemphasis of, the importance of the "idea" aspect of a computer program, as contrasted with the key element, which is the "expression" of a program by means of written text.

This paper is intentionally broader and less conservative than the typical legal paper on the subject. Thus it discusses not only rights in copyright which are well-proved in the law, but it also explores additional, legally untested rights which derive from a careful logical analysis of the subject. This broader scope of discussion is appropriate for the new computer field. The owner of computer software deserves to be acquainted with the widest scope of rights which he might claim—even though eventually such rights may not be totally supported in the courts. Similarly, the users of software deserve to be acquainted with the fullest extent of their possible obligations.

It is reassuring to me that the earlier critical studies of software and copyright [1, 2, 3, 4], which have been addressed primarily to the legal profession, reached many of the conclusions advanced in this paper. Where there is not such a corroboration, it appears that the particular computer topic area dealt with in this paper was unknown to, or was not considered by, those earlier, legal commentators. The main topics analyzed here which were not considered in previous studies, are "systems," "languages," and "microprogramming."

In one sense, this article is a prognostic paper by a concerned computer professional. The conclusions presented here may be taken as my personal prediction of how the existing Copyright Law may be applied in the future to the protection of computer programs, software systems, and documentation.

In another sense, it is a motivational paper. It is my expectation that this discussion of copyright will stimulate many readers to investigate how they can apply copyright to their own problems of software protection and its commercialization.

In any event, the reader is advised to seek the counsel of his lawyer before acting upon any of the matters discussed in this paper.

WHAT IS COPYRIGHT

The legal property right that an author has in his original writings goes by the name of "copyright." It is the right to make copies. In the case of published writings, these rights derive from a provision of the US Constitution (Article I, Section 8):

The Congress shall have Power . . .

To promote the Progress of Science and the useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.

On this authority, in 1909, Congress enacted the most recent Copyright Law [5]. This law gives to any author who has satisfied certain formal requirements a "copyright" on his work. The copyright secures

a variety of exclusive rights for the author during a period of 28 years, and during an optional renewal period of 28 additional years.

The purpose of the Copyright Law is to provide a means to prevent an author's loss of his property rights in a piece of writing as a consequence of its publication. The law is designed to encourage open publication by authors of their new writings, with the ultimate intent of promoting "the progress of science and the useful arts."

The property rights which are of particular interest to software are specified by the Copyright Law as follows:

Section 1. Exclusive Rights as to Copyrighted Works—Any person entitled thereto, upon complying with the provisions of this title, shall have the exclusive right:

a) To print, reprint, publish, copy, and vend the copyrighted work;

b) To translate the copyrighted work into other languages or dialects, or to make any other version thereof, if it be a literary work; . . .

c) To deliver, authorize the delivery of . . . or present the copyrighted work in public for profit if it be a . . . nondramatic literary work; to make or procure the making of any transcription or record thereof by or from which, in whole or in part, it may in any manner or by any method be exhibited, delivered, presented, produced, or reproduced; . . . and to exhibit, represent, produce, or reproduce it in any manner or by any method whatsoever. . . .

The exclusive property rights which are derived from copyright are like any other property rights. There is an "owner" of the copyright. The owner may sell all or part of his rights, transfer the rights to others, use his rights, or give temporary permission to others to use some or all of his rights under copyright.

The effect of copyright is to transform a writing into a legally protected article of commerce. This is most familiar in the case of books. The owner of the copyright for a book has the sole right to print and to make original sale of copies of the book. (He has

no control over resale.) The same transformation into a legally protected article of commerce can take place with computer software, as will be described, making items of software into commodities that may be marketed and traded like any other valuable commodity.

The formal requirements for securing copyright under the Copyright Law are surprisingly simple. The key requirement is that an author must publish his work with a legal notice of copyright *placed on the work* at the time of first publication. Such a legal notice is of the form:

Copyright © 1975 Calvin N. Mooers

This notice must be placed on the title page, or the page immediately following, of the work in which copyright is being claimed. The purpose of the copyright notice is to inform the public that the author of the work bearing the notice is claiming legal rights in the work according to Copyright Law. Simply stated, the notice is a "No Trespassing" sign.

To repeat, it is the simple act of *publication with notice* that actually secures for the author his legal copyright protection!

Later, a public record is made of the author's claim to rights in copyright by submitting copies of the published work to the Copyright Office, together with an Application Form and a \$6 registration fee. In due time the Office issues a certificate of "Registration to Claim for Copyright." Under our Federal laws this kind of copyright is called "statutory copyright."

If the author does not publish his work, and does not publicly circulate or distribute it without restriction, then the author's work is protected by "common law copyright" under the provisions of the laws of his state. These laws serve to retain ownership of the work for the author until he does publish.

All private papers, letters, memoranda, and other unpublished documents are protected by common law copyright. Strict secrecy is not a requisite for common law copyright [15].

While software is under development, and even though there may be a number of copies which have been shown to various friends or colleagues, or even to potential

customers, it is protected by common law copyright. No formalities are required for such protection, although for precaution such copies may be marked with the legend "All rights reserved" or "Proprietary." Protection by common law copyright ends with "publication." Publication occurs at the time when unrestricted distribution or sale of copies of the work is offered to the public. It may occur through the lease or license of the copyrighted work, even though there are restrictive clauses in the license agreement. Publication requires a show of intent by the author to make his work available to the public.

If publication occurs without the copy bearing the legal copyright notice, then all rights in copyright are lost forever, without recourse. Rights in both common law copyright and in statutory copyright are lost. The writing is then "dedicated to the public" and is "in the public domain." No one owns it. To prevent such loss of copyright from occurring through inadvertent publication, many people place the legal copyright notice on all important documents. Then if publication occurs, statutory copyright is secured and ownership is not lost.

In this paper, we shall be primarily concerned with statutory copyright.

Are computer software, systems, and documentation eligible for protection under the Copyright Law? This question will be examined in detail in the remainder of the paper. In general, the answer is "Yes."

To begin with, Copyright Law is surprisingly broad. It states that copyright may be secured on "all the writings of an author." It turns out that the "writings of an author" include such things as: musical compositions, statuary, graphic works of art, photographs, scientific models, maps, etc., as well as the more familiar copyrighted works like books (novels, poetry, etc.), whose natural form of expression is by alphanumeric characters on a printed page.

One main focus of copyright is on written text. The logic of the application of copyright to software relies mainly on this fact. All forms of computer software, computer systems, and documentation, at some stage of their development or application,

make use of alphanumeric text. In the case of software documentation, this is self-evident, because such documentation is simply technical prose. It is more surprising (perhaps because no one had ever given it very much thought) that most or all of the rest of software technology is also concerned with "text" taken in the broad sense. This dependence upon text or "writing" puts software technology directly within the scope of copyright protection.

In terms of the language of the Copyright Law, software in its text form is classed as a "nondramatic literary work." You can write it, and you can read it—therefore it is "literary." Furthermore, since literary works are divided by Copyright Law into the categories "dramatic" and "nondramatic," it is clear that writings in computer software are "nondramatic literary works." Another important consideration in analyzing the implications of Copyright Law is the fact that programs are run or "performed" for the satisfaction or edification of people—just as lectures, sermons, or motion pictures are. This performance aspect may lead to another class of rights in software having to do with text dialog, visual effects, or amusing actions.

What is Not Protected by Copyright

Although copyright secures to the copyright owner a variety of rights, there are several important things that copyright does not protect. The first is that copyright by itself does not in any way give continued control or ownership of the physical object which carries the writing. The law expresses it in this way: "Copyright is distinct from the property in the material object copyrighted." (Copyright Law, Section 27) Since this distinction sounds peculiar, an illustration will help to make it understandable.

When a person has legitimately purchased a copyrighted book, he may thereafter resell, loan, or give the physical book to whomever he pleases, or he may even destroy it, without any recourse whatsoever from the copyright owner. The book may be sold and resold. However, the customer,

or anyone else having the book, may not (without permission) make a photocopy of the book, or any substantial part of it. Even though a person owns the book, he may not translate the writing of the book into another language, nor may he publicly present or perform the contents of the writings of the book for a profit. Thus, although the copyright owner does not have continued control of the physical object which is the book, he does have continued control—irrespective of the book's ownership—of certain things that can be done with the writings contained in the book.

It is this kind of continued control over certain uses of the writings that makes copyright so applicable to computer software.

The second thing which copyright does not protect is the "idea" contained in the copyrighted writing. Again we meet a peculiar distinction which requires explanation.

We saw that continued control of the physical object (the book or paper) is not protected by copyright. We know that the writing contained in a book or paper has as its purpose the conveying of information, a message, or some ideas. Therefore, if these "ideas" are not protectable, what is left for copyright to protect?

The answer is that copyright protects the "expression" of the "ideas" contained in the writings. By "expression" is meant the sequence, arrangement, and choice of the marks, words, sentences, or alphanumerical characters (letters and digits) used to represent or to convey the ideas of the copyrighted word. Also included in the "expression" is the sequence, choice, and arrangement of descriptive elements (names, incidents, or other things used to explain, develop, or state the ideas), as well as the choice of any factual elements brought forth (the particular sentences or quotations used, any diagrams used, and so on). Thus, "expression" involves all the descriptive elements used to convey the ideas of the copyrighted writing.

Since copyright protects the expression, it provides protection against making copies of the marks on paper which compose the elements of expression; it protects against

translating the marks or elements of expression into any other language or system of marks and elements of expression; it protects against making any other version of marks or elements of expression (even for conveying the same ideas) which are directly derived from these marks; it protects against exhibiting, producing, or reproducing the marks or elements of expression, or any version derived therefrom; and it protects against the giving of presentations or performances in public for profit from these marks or elements of expression.

Where does the "expression" leave off, and the "idea" take over? The best insight into this matter comes from discussions of copyright as applied to novels and dramatic productions. In these, "expression" is considered to include the choice of incident, the personalities and development of character, the choice of names, the elaboration of the plot, the choice of locale, and the many other minor details and gimmicks used to build the story. In other words, "expression" is considered to include not only the marks, words, sentences, and so on in the work, but also all these other details or structures as they aggregate into larger and larger units to make up the expression of the entire story.

In other words, after the bare abstract "idea" has been chosen (e.g., boy meets girl, boy loses girl, boy wins girl), the "expression" to which copyright applies covers the remaining elements of original choice and artistry which are supplied by the author in order for him to develop, express, and convey his version of the bare idea.

The third thing which copyright does not protect are certain objects which are primarily articles of utility. However, this is a shaky area of exclusion, having many exceptions. Thus an artistically designed article of clothing, by long precedent, is not protected in its design by copyright, since garments are considered to be articles of utility. Yet, an artistic statuette is protectable by copyright, even though it is used in a utilitarian fashion as a lamp pedestal [8]. If an original writing is for "explanation," it has been considered to be copyrightable, irrespective of its other attributes. This dis-

tion concerning "explanation" versus "utility" is relevant to our consideration, since one of the very important functions of a source language computer program, and its supporting documentation, is to provide "explanation." In fact, "explanation" is the only thing it can directly provide, without a great deal of transformation. On the other hand, the binary object program, resulting from many machine transformations of the source program, can be said to have actual "utility"—as well as having a limited value for "explanation" (for debugging). Since the object program is a transformation from a copyrightable writing (the source program), it is protected by the copyright of the source writing.

A fourth thing which copyright does not protect are writings which do not contain any "creative" expression of the author. Blank forms, and mechanically-generated lists of words or numbers fall in this category. However, lists or writings which are the result of artistic or creative selection, arrangement, or labor by the author are copyrightable. Thus mathematical tables, since they embody many elements of choice, and are the result of labor, are copyrightable. Subsequent mechanical re-arrangements of such lists or writings do not lose the copyright privilege, since they are "other versions" of copyrighted writings.

A fifth thing which copyright does not protect are writings which are in the "public domain." These include writings for which the copyright has expired, most governmental publications, and writings which were never copyrighted. Such writings may be copied, translated, or used freely by anyone. However, an author may make a new compilation, adaptation, abridgement, dramatization, or translation of works in the public domain, and such a work is then considered to be a new work which is copyrightable. The new copyright does not remove the original works from the public domain. Instead, the copyright applies only to the new elements of authorship, which consist merely of the new arrangement, adaptation, translation, etc., of the old public domain writings.

HOW IS COPYRIGHT INFRINGED?

Since the copyright owner has certain exclusive rights to his property under the law, actions taken without the permission of the copyright owner which transgress these rights constitute "infringement of the copyright."

Referring to the listing of the exclusive rights (see page 48), we see that there are a wide variety of actions which constitute infringement when undertaken without permission. These actions include (in Section 1.a) making copies of the copyrighted work, reprinting it, or selling such copies or reprints; they include (Section 1.b) making any manner of translation or derivative adaptation of the work; and they include (Section 1.c) presenting or exhibiting the work in public for profit and making records by which it can be presented or performed. Later in this paper, the particular application of these provisions to the different software situations will be explored in detail.

The question is, how strictly are these provisions of the Copyright Law interpreted? Is it all right to "infringe just a little"? For example, is it all right to make a handwritten copy? Is it all right to copy the language of a document providing some little changes are made in it here and there? Is it all right if the copy is to be made from memory? Is it all right to make the copies providing they are not to be sold? Is it all right to extract chunks of text here and there and to work them into your own (additional) writing? Is it all right to make a copy of the complete text, just so long as you give full credit to the source?

The answer to each of these questions is that if permission of the copyright owner has not been secured, you are probably infringing his copyright. In each case, the copyright owner—if he so desired—could bring suit in a court of law to recover damages from your infringement of his copyright prerogatives. The reason that a copyright owner does not usually go to court because of small acts such as these is that in most cases the infringement is so minor and of so little consequence to the owner's rights that it is simply not worth his trouble and cost to go to court.

On the other hand, there is no guarantee that a copyright owner will always overlook such acts of infringement, even though they may seem minor to the transgressor. There will be instances where the copyright owner may feel that it is well worth the trouble and expense to go to court. This may happen when the nature of the infringement is troublesome or damaging to him or to his commercial interests, where the nature of the infringement appears to flout his rights in copyright, or where it would tend to establish a public habit of infringement which, if continued, would jeopardize his copyright and commercial interests. In such cases the owner might go to court to "make an example" of the offender.

This has been particularly true in the case of copyright in musical compositions. Currently there is pressure by the publishers to stop all casual copying of sheet music by commercial photocopy establishments. The performance of copyrighted plays, even single performances by amateurs with free admission, is strictly controlled by the copyright owners. With copyrighted musical compositions, the strict control of the ASCAP and BMI organizations in monitoring and charging for musical performances is well known.

You might wonder "at what instant does infringement occur?" Does legal recognition of its occurrence wait until the actual publication, or when the derivative work is presented to the public? The answer is that infringement occurs at *the very moment* the infringing act takes place: when the copy is made, when the paraphrase is written, when the transformation is written out, when the extracts are put on paper, and so on. Further infringements may then occur at each successive stage of making additional copies and getting them ready to circulate to friends, customers, pupils, or the public. The fact that some or all of the acts are taken in private does not mitigate the fact of infringement.

There is a narrow exception to these comments on infringement. To quote from the explanation offered by the Copyright Office [14]:

... the courts have recognized certain limited uses of copyrighted materials as "fair use." In the broadest terms, the doctrine of "fair use" means that in some circumstances where the use is reasonable and not harmful to the copyright owner's rights, copyrighted material may be used to a limited extent without obtaining permission. For example, under this doctrine scholars and critics have been held free to publish short extracts or quotations from copyrighted works, without the permission of the copyright owner, for the purposes of illustration or comment. The line between "fair use" and infringement is unclear and not easily defined. There is no specific number of words, lines, or notes that can safely be taken without permission. Acknowledging the source of the copyrighted material does not avoid infringement.

With computer software, if there is any possibility that the copyright owner may feel damaged by your actions, your reliance upon "fair use" as a defense or excuse may be dangerous.

At the time of preparation of this paper, a most important case involving copyright infringement and the definition of "fair use" has just been ruled upon by the US Supreme Court. The case was brought by the Baltimore firm of Williams and Wilkins Co., publishers of various scholarly medical and scientific journals. They alleged copyright infringement and sought relief from open, large-scale, systematic photocopying of nearly a million pages a year of its journals, without permission, by the National Library of Medicine, an agency of the US Government. Photocopies of copyrighted journal articles were supplied without cost and upon demand to researchers and to other libraries. Some of the recipients appear to have terminated their subscriptions to the firm's journals a result of the library's services. Under the particular circumstances of the case, a lower court ruled that use of the copyrighted material by the Library was "fair" and denied relief. Williams & Wilkins Co. appealed to the Supreme Court.

Despite the importance of the *Williams & Wilkins Co.* case in clarifying the scope of "fair use" in copyright, and in clarifying the scope of copyright protection to authors and publishers, the Supreme Court ruling did nothing to settle matters or to give legal guidance. The Court split 4-4 in rendering its decision, and no opinions were presented for either side. The effect of the split is merely to sustain the decision of the lower court, and to allow photocopying under the very narrow circumstances of the particular case.

Congress has been considering, during the past dozen years, a copyright revision bill which does define the limits of "fair use," and which does deal with many of the problems of current technology, such as photocopying, computers, and television. In view of the inconclusive ruling of the Supreme Court, it now appears more essential than ever before that Congress act to bring our Copyright Law up to date. Until clarification of the muddled area of "fair use" of copyrighted material is provided from some source, the constitutional rights of authors to their writings, and the protection of their property in the marketplace will remain clouded.

The Penalties Are Severe

If the copyright owner feels that he is injured by an act of copyright infringement, he can take his complaint to court. The reader will probably be surprised at how stiff the penalties are in US law for infringement of copyright. The copyright in a work is viewed as a valuable property. Infringement is viewed as theft or damage to that property. Thus the penalties are both civil and criminal!

If infringement has been found to occur (no matter how much), the guilty party can be ordered to stop his offending actions. Furthermore, the law provides for mandatory statutory damages which "shall be not less than the sum of \$250" nor more than \$5,000 for the act of infringement. Alternatively, it provides for payment to the copyright owner of "all the profits which the

infringer shall have made from the infringement." If the infringement is willful and for profit, the law provides the criminal penalties—on conviction the infringer "shall be punished by imprisonment for not exceeding one year, or by a fine of not less than \$100 or more than \$1,000, or both, in the discretion of the court. . . ."

The mere facts that the infringement might have been for a "good cause," and that there was no money transaction do not affect the imposition of these penalties. In one sad but instructive case, a young professor of music volunteered his services to a local church as a choir director. He arranged some copyrighted sheet music so that his amateur singers could reach the high notes, made some 48 copies on the school duplicator, and from these he presented two admission-free public choir performances by his group. Afterwards he offered his new arrangement to the copyright owner without charge, saying that his version was more easily sung by amateur choirs. The ungrateful owner took exemplary action and sued the choir director for copyright infringement, 1) for making an unauthorized arrangement of the copyrighted work, and 2) for making copies. (The performance, since it was free and in a religious setting, escaped under one of the exceptions written into the law.) The trial judge excused the action as "fair use," but the owner appealed. The appeal judge found that two instances of infringement had indeed occurred—irrespective of the mitigating circumstances—and was therefore obliged to reverse the decision and to order the assessment of the statutory damages of \$250 for each of the two copyrights infringed [6].

Many literary properties, such as books or stage plays which can be made into movies or TV productions, have a considerable dollar value. In such cases, the assessed damages and penalties for infringement can run into many thousands of dollars. Since the dollar value of computer programs and software systems are of comparable magnitude, we can expect that their infringement will also involve penalties and damages of similar large amounts.

COPYRIGHT APPLICATIONS TO SOFTWARE

The best way to learn what copyright is, and what its implications are regarding infringement, is to examine a number of instances where copyright seems to have significant application in connection with computer software. Therefore I shall discuss several specific kinds of software, and the transgressions that the reader might make in regard to property rights of copyright. The first kind is "software documentation."

Software Documentation

By "software documentation" I mean any manner of written descriptive matter which is provided to instruct, describe, or to support the use or operation of any aspect of a computer system. It may be completely textual, or it may have varying amounts of diagrammatic or graphic material. In either case, it is without question subject to copyright protection. "Diagrammatic material" includes such things as charts and flow diagrams. Whether they are included within the text, or are in separate plates in the publication, they are included within the copyright protection.

Documentation has as its primary intended purpose the function of instruction, support, and the like. Thus studying from it, learning from it, operating a computer console according to it, or writing application programs in accord with it, are all intended uses of such documentation. As such, these uses are free of any suspicion of copyright infringement. Certain other uses of such copyrighted documentation may lead to trouble because of infringement.

The most frequent infringing act with documentation is to make photocopies of the text of some of the pages. Ordinarily nothing will happen. However, if the copyright owner really feels abused by it, he has a clear legal prerogative to bring action against the infringer. It could cost the perpetrator \$250 or more for each instance of such infringement!

Another familiar infringement is to reprint or to republish a section, page, diagram, or even the complete document. Unless you have been given prior permission to

reprint, this is definitely an infringement—even if you give full credit to the source, and even if you include the original copyright notice on your reprinted copy.

You may wish to write a paper, report, or article on the same or similar topic. If so, watch out! You can use without question anything which is already in the "public domain," i.e., which has no copyright. You can copy such public domain material, adapt from it, paraphrase it, or do whatever you wish. You can freely write up your own ideas. However, if you begin to adapt or to follow closely any copyrighted writing, you may be flirting with trouble. The copyright owner has the exclusive right to "make any other version" of his work. Therefore if you paraphrase from, adapt from, extract from, add to, rearrange, translate, expand, edit, make a manual of, or otherwise transform from any portion of his work, or combine it with other works, including your own, the owner may reasonably consider that you have infringed his rights. The courts may agree.

In one instance, an infringer was punished for making an unauthorized quotation of only three sentences from a medical book [7]. The author had strong feelings in the matter, since the infringing quotation was put into a cigarette advertisement without his permission. The author vigorously objected to the implication that he had allowed his name to be used for a fee in such advertising, and he exercised his rights in copyright to stop the publication. He prevailed.

There is a widespread misconception that copyright infringement can be avoided by "merely making a few changes here and there in the copy." Don't you believe it! When brought into court, such instances frequently seem to especially infuriate the judges. Most judges—quite rightly—consider this to be a willful, calculated, sneaky trick. They consider it to be a specific indication that the infringer knew he was committing an illegal act, and that he made the superficial changes in an attempt to evade the charge of infringement. Such an infringer may be given a very hard time in the courts.

In addition to such acts of misappropriation, which are often designated "piracy," there is the matter of plagiarism. Plagiarism consists in taking the work of another (data, ideas, expression) and passing it off as your own. As such, plagiarism is offensive to professional ethics—quite apart from any possible liability for infringement of copyright.

These considerations on infringement of written documentation apply as well to diagrams and flowcharts. Transformation of the descriptive content of such graphic material into a written text form may infringe the original. Similarly, an infringement may occur with the transformation from a text description to a diagram or flowchart. Infringement is especially likely to be found when the creative work of the first version is carried over, detail by detail, into the transformed version. A particular case occurs when the original document is the source listing of a computer program. Using a copyrighted listing as the source to automatically produce a computer-generated flowchart would infringe the copyright of the original program. This is an infringement even though the transformation is produced by machine, and the flowchart is used in private.

The message that you should get from this section is that if you use someone else's copyrighted documentation in any way to produce new writings of any kind, and if there is any question whatsoever in your mind about the use of the material, be sure to get permission for your intended use from the copyright owner. If you can't get permission, don't use it.

Protection of Programs

The next question is, what is a "program" and, under the terms of the Copyright Law, how might it be infringed? The term "computer program" in the mind of the computer professional may mean any of a number of quite different things: a deck of punched cards, a magnetically-recorded tape, an octal listing, the particular state of the magnetic cores in a computer memory bank, an abstract algorithm, a set of concepts or ideas, and so on. These things, in themselves,

are probably all outside the immediate scope of the Copyright Law.

Yet there is a manifestation of a "computer program" which is indeed most congenial to the intent of the Copyright Law. This is the program as it is written or printed on paper by the programmer, in alphanumeric text, in some source language. The source language for the program may be the assembly language for any one of the machines, or it may be any one of the high-level languages such as FORTRAN, JOVIAL, ALGOL, APL, or the like.

The text form of a computer program is without doubt a copyrightable "literary work." It therefore carries all the forms of protection available to other copyrightable works under the US laws. It is a "literary work" for the pure and simple reason that a person trained in the requisite computer language can "read it" and understand it. It has all the marks of a literary work. It is created by the original intellectual work of one or more persons. It is composed of sequences of alphabetic, numeric, and punctuation characters forming text on a page. It is a manner of expression of an idea. It may even contain interspersed comments in something like the English language. The mere fact that the text form of the program can be transformed, through a process of multiple mechanical steps, into a means for eventually controlling a computing machine, does not detract from its suitability for legal copyright protection.

It is also conceivable that the copyright program may, in addition, be patentable—but that in itself would not affect any prior copyright.

Since the text form of a computer program is a literary work, it can be infringed by any of the various forms of copying, translation, the making of other versions, and exhibition or performance. Consider some of the various kinds of infringement:

To begin with, the use of a keyboard device to copy the copyrighted text of the program into some other machine-readable medium (cards, tape, core) is an instance of copying, and thus it is an infringement. Conversely, the production of a printed version or "listing" from a machine readable me-

dium is another instance of making a copy—or copying. Making a photocopy of a printed listing is copying. Any of these acts of copying, when done without the specific permission of the copyright owner, is an infringing “action” and may cause the owner to object.

These acts are infringing even if you have outright ownership of the piece of paper containing the copyrighted text of the program. The right to make copies (and all the other rights of copyright) is separate and distinct from the ownership of the actual physical medium. While you may legally sell or give the paper to others, you have no right to copy it!

When the text of a program as written in any source language is copyrighted, then any copy or derived version in any machine readable medium shares the copyright protection of the original. In one view, the derived version is a copy, a “translation” or another “version” of the original. In another view, the actual medium containing the copy or the derived version may be considered a form of “... plates, molds, matrices, or other means for making ... infringing copies” according to the language of the Copyright Law. In either case, the production of such copies or transformations in machine media constitutes an infringement of the copyrighted material.

An important manifestation of the copyrighted text of a computer program is the version produced by the action of an assembler or compiler. The act of making such a version is also infringing. In the first place, reading-in the source program text is an instance of making a copy. Production of the binary object code output is clearly within the broad scope of making a “translation” into “another language or dialect.” It is also a copy. It is also an instance of the production of “any other version” of the copyrighted work. In any event, the binary code output produced shares the copyright protection of the original.

In light of this fact, it is good practice for the copyright owner to ensure that each copy of the actual binary code actually *contains* a statutory copyright notice. (See reference 12 for examples.) This notice may be

abbreviated as “COPR. (C) 1975 C. MOOERS” and it should be inserted as a data constant of type “character” at the beginning of each module of the object code. Furthermore, this data constant should always be kept with the operative module of object code, irrespective of its movement back and forth between the central processor and a storage medium.

Any movement of the binary form of the copyrighted computer program from one medium to another, or from one place to another, within a computer system is an instance of copying.

Does running a copyrighted computer program constitute infringement? Very likely it does [3], although the language of the Copyright Law (written in 1909) is understandably vague in these matters. According to one interpretation, because a program is protectable by copyright as a “nondramatic literary work,” then the exclusive rights given to the copyright owner would seem to include the exclusive rights to “exhibit, represent, produce, or reproduce it in any manner or method whatsoever.”

A computer under control of its program, with the flashing lights, spinning reels, snarling line printer, and its fascinated human spectators, could be said to be giving an “exhibition,” “production,” or even “performance” of the program for the spectators and beneficiaries of that computer run. A receptive human audience seems, by implication of the interpretation of the law, to be a necessary ingredient for any infringing “performance” of a copyrighted work. If so, an audience (either directly or indirectly) certainly exists for any computer run. This argument with regard to an appreciative human audience to a computer performance of a copyrighted work is even more persuasive when there is a program running interactively on a time-sharing system, in which case the program most definitely has a “captive audience” for its performance.

If one does not want to accept this “performance” interpretation of copyright implications for running a computer program, then it is possible to view the running of a program as the step-by-step, instruction-

by-instruction, translation internally of the program text by means of which the physical computer is controlled.

If neither of these interpretations is taken (nor any of the other reasonable interpretations which might be advanced), then it can still be shown that the actual running of a program in a modern computer is so dependent upon multiple instances of making and moving copies of the program from here to there, that in practice a program cannot be run without somehow making infringing copies, and thereby incurring copyright liability.

There is an esoteric line of argument that when a copyrighted work is put to the use for which it is intended, such use is not an infringement. This argument is not always true. Dramatic plays are intended to be used by being performed, and such a use is certainly infringing.

Nevertheless, the "use" argument has been raised in regard to running computer programs. The force of this argument has been greatly diminished by a recent decision of the Supreme Court [8] which held that an artistic statuette was protectable by copyright even though the statuette was actually used as an article of utility, as the pedestal of a table lamp. In analogy to this decision, the text form of a computer program (which is the thing copyrighted) has for its first and main use the explanation to and guidance of human beings—such as for their instruction, guidance in debugging, or for further development of the program. Thus it is clearly protectable by copyright. Only after such an explanatory text form has been greatly transformed by assembly or compilation, is a transformed version usable at all for controlling a computer. However, such an eventual "use" of a derived manifestation doesn't destroy the copyrightability of the original program text—nor of copyright protection of the derived form.

Running a program might also be considered merely as "carrying out the idea" of the program, and as such might be considered to be free from infringement. But, what is the "idea" of a program? At one level, the idea of a program may consist of

such nebulous concepts as: 1) "It is a good idea to have a program to compute pay-checks"; or 2) "It would be good to have a program to draw flow diagrams"; or 3) "It would be good to have a program to carry out general interactive string manipulation." However, none of these abstract ideas—devoid of any carefully worked-out *expression* of the idea by a series of program statements—could possibly make a computer go. Thus the expression of the idea, which is, of course, a copyrightable written program or a derivative of a program, is the thing that must be translated, copied, transformed, performed, etc., to eventually make the computer go.

At another level, the "idea" of a program might be considered to be the scheme of abstract manipulation or algorithms of what the program is "really about." Such disembodied abstractions are not copyrightable, they cannot be infringed by computer operation, nor can they make a computer go. Neither the law, nor computer hardware, is responsive to such disembodied abstractions. To be usable, or copyrightable, they must be put into some notation or language, a concrete form of expression, and such written expressions are copyrightable and can be infringed.

In an actual program, the particular sequence and choice of available algorithms, manipulations, etc., as embodied in the expression of the program, also constitute valid elements of expression. They are like the incidents in the development of a story or novel, whose choice and sequence is protected by copyright. In summary, everything but the barest "idea" in a program appears to be protectable by copyright.

Writing Similar Programs

The writing of programs which are adapted from other programs may lead to copyright infringement. The simplest case is the conversion of a program from one source language to another. Consider a copyrighted text of a program written in FORTRAN. A person might take this listing and, with some ingenuity, write a corresponding program in ALGOL. He might follow the general design

of the original program, and make use of any features which might carry over. Much of the original expression, such as the data structures, the choice and sequence of algorithms, the organization of the parts, and the layout of the original program could be carried over to the ALGOL version. Thus the programmer's work in writing the ALGOL program would be greatly expedited. However, he would probably thereby infringe the copyright of the original program, since his work would be closely guided by the original expression—even though he constantly had to make many changes and adjustments in going from one language to the other.

Producing a derived program in this way constitutes infringement because the second program is clearly a "translation" from the first program into "another language or dialect." If it is not to be viewed as a "translation," because of the considerable amount of change from the first, it still constitutes an infringement because it is surely "another version" of the first text. The key question is, "Was the second author materially guided in his choice of expression by the writings of the first author?" If he was so guided, or if he appropriated any substantial part of the work of the first author, then he has certainly infringed, since infringement is "taking the work of another" without permission.

By the same argument, a copyrighted text of a program written in assembly language for one machine may be "translated" into the assembly language of another machine, and this could also constitute infringement. The same applies to translations from assembly language to FORTRAN, etc., and conversely.

On the other hand, it is noninfringing (and quite all right) to be "inspired" by the "idea" of some copyrighted program, and then *completely independently* to write your own program. Thus, if you get the idea that a "pension plan program" would be a fine thing to have (even if you got the idea from hearing about another pension plan program), you are safe to write your own independent pension plan program. (Safe, that is, providing there are no patent problems, providing that you do not copy the sequence

of algorithms, description of data structures, input commands, etc., and providing that no illegal means were used to learn about the first program, and so on.) However, in order to write your program *independently*, your work must, in fact, be *independent*. If any traces of the original program, such as names of identifiers, the nature and sequence of chosen subroutines, the use of stacks, commands, data formats, or arrangement of the sections can be found in your new work, then a judge in the courtroom may rule that your new program was derivative and infringing. In forming his opinion, the judge can be expected to have the use of independent computer experts to advise him in locating any existing similarities, and to explain their role in the program. In making a determination of infringement in this manner, the judge will be following the precedent of case after case of literary plagiarism of copyrighted novels, plays, and other literary and artistic productions.

Debugging Protected

A rock-solid, well-running, completely debugged computer program may be a very valuable property. Often the initial cost of writing such a program is but a tiny fraction of its final cost. The complete lack of bugs may represent its greatest element of value. The lack of bugs depends upon the correctness of a multitude of fine and exacting details in the text of the program. The text of such a smooth-running program may be compared to the valuable text of a well-polished novel. As with a polished novel, the method of copyright is well suited to the protection of a smooth-running program.

The reason for suitability of copyright is that if anyone tries to evade the copyright by making small changes in the program, he will most certainly introduce inadvertent bugs. The more significant his introduced changes are (such as recasting the nature of the subroutines, renaming the identifiers, changing data forms, etc.), the greater will be the number of serious bugs that can be expected in the new version. Therefore, if anyone wishes to avoid copyright problems by just using the bare "idea" of the program

and by changing everything else, or by independently writing a new program, he will find that he still has almost the complete job of debugging on his hands. This is the major part of the work and expense. If he tries to get away with anything less in the way of changes, he is choosing to live dangerously, and he could be found to be an infringer. In other words, the likelihood of his being caught as an infringer is in proportion to the amount of debugging that he attempts to avoid through copying.

There is a curious and possibly useful other side to the matter of program bugs and copyright. The makers of copyrighted maps, directory listings, and the like, have long been known to insert harmless, but purposeful "mistakes" into their copyrighted works. The purpose is simply entrapment. If a brazen infringer makes a copy of such a work, passing it off as his own independent creation, then at any later time the presence of these fictitious insertions in his work can discredit his claims to its independent creation. Owners of valuable software might therefore consider using analogous techniques of inserting obscure code or bugs in order to booby-trap their works against possible infringers.

Systems, Compilers and Languages

Upon initial analysis, one might conclude that copyright has no application to the protection of computer systems such as operating systems, file management systems, interactive calculator systems, compilers, language interpreters, language systems, and the like. The protection of such systems would seem to be limited to patent protection—if and when patent is applicable. The reason for this initial impression is that copyright protects merely the text of the writing as a literary work, i.e., the "expression" but not the "idea," while patent is directed to the protection of ideas as they are embodied in some new arrangement or interaction of physical elements.

For example, consider a hypothetical compiler system which achieves its results through the application of a new idea for

using three stacks, a certain data format with pointers, and a group of recursive parsing routines. Such a system might, in fact, be patentable if the system is new and if it has the requisite inventive cleverness about it [13]. In such a case, patent protection is desirable. If a patent is secured, the inventor would then have the usual exclusive rights by patent to "make, use, and vend" his invention. He would be protected against unauthorized use by others of the same idea for the arrangement of functional modules composed of stacks, data forms, pointers, routines, and the like. He would be protected irrespective of how such a software system was programmed, or irrespective of what command language system was used with it.

In addition, prior to the issue of a patent, he could secure copyright on the actual program text of a particular implementation of his compiler.

Yet, this discussion of protection is still too limited, since it considers only two avenues of protection of a hypothetical "system," namely, 1) patent for the functional arrangement of the system, and 2) copyright for a particular program text for the system.

There is another avenue by means of which protection can be secured. This is through the highly-developed language command structure which is used to control such a system. When statements composed of combinations of symbols from a command structure are presented to the software system, certain definite control actions take place. Data may be moved from one place to another, other programs may be loaded or run, output text or program text or code may be generated, or arithmetic computations may take place with numeric answers being printed out. The actions are usually complex, and control must be exact. When a command statement is given, it must lead to predictable results.

To accomplish this, there must be a high degree of exactness in the definition of the statements and symbols of the command structure, as well as in the corresponding definitions of the functional actions to be taken by the software system. The two

must be very closely related. In fact, each statement of the command structure might be called a "mirror image" or "logical dual" of the corresponding function of the software system.

This "mirror" relationship is, in fact, the case. For example, given the text for a *program* for a software system (e.g., for a compiler), it is then possible to reconstruct a text which is an exact *definition* of the corresponding command structure or language. This reconstructed text of the definition for the command structure is, in fact, a "translation" from the text of the program.

The converse is also true. Given the text which *defines* exactly a command structure or a language, it is then possible to perform a translation from such a detailed text description and to come forth with the text of a *program* for the corresponding software system (e.g., a program for a compiler). The software system resulting from this program will, in fact, accept the described symbols and statements of the command structure, and will operate correctly upon them.

As might be supposed, the reconstruction of a compiler program solely from the description of the command structure or language may result in a compiler program which does not have all the clever features in the use of stacks, pointers, etc., of the original program. However, from the standpoint of its logical or functional performance on the command structure language, such a reconstructed program must be equivalent to the original program.

This kind of translation from a description of a command structure language to a program—and back—is very similar to the kind of transformation involved in the preparation of a dramatic work from a novel, or conversely. The essential elements of plot, the names of the characters, the details of characterization, events, time, place, incidents, and so on, are all carried carefully over from the first work into the second work by the transformation. This need for carrying over the details is even more necessary in the case of a translation between natural languages,

such as translating from English into German, where such sentence of the source is individually translated into the target language.

In close analogy, translating from, or making versions from, the text for a program to the text for defining the command structure of a system (or conversely), also requires a meticulous attention to accuracy and detail. Each essential element of the expression must be carefully carried over into the text of the target version. If any essential element or detail of expression is gratuitously added, or omitted, or changed, the effect will be to destroy the essential "mirror" relationship between the intent of the statements of the command structure and the intended functional actions of the software system. If the mirror relationship is faulty, the system will perform erroneously, or it will do things which were not intended by the submitted commands. Functional deviations of this kind, between the commands and the actions, are clearly unacceptable in any working system.

From the preceding discussion, it is seen that there are two kinds of text involved in any working "system." The first is the text of the program. The other is the text of the detailed definition of the command structure or language. Each detailed element contained in the expression of the one text must have a corresponding detailed element in the expression of the text of the other. Therefore, whenever such a mirror relationship prevails, either text is logically an exacting derivative of the other.

In consequence of this conclusion, if either one of these two texts is copyrighted, then the unauthorized preparation of a mirror image text of the other kind would appear to constitute copyright infringement [9]. For example:

- A) If the definition text for the command structure of a system is copyrighted, then the unauthorized preparation of a text for a computer program to carry out exactly the actions specified for the defined command structure constitutes the making of an infringing version of the copyrighted text.

- B) If the source language program text of a system is copyrighted, then the unauthorized preparation of a text defining and explaining exactly the command structure for the actions inherent in the program text also constitutes the making of an infringing version of the copyrighted text.

In case A, even though a new program text for a compiler involves original programming by a programmer, it still does not avoid infringement so long as the program text is derived from a copyrighted text expression which constitutes the definition of the command structure of the system. In other words, a variety of quite different "original," though derived, programs can all infringe.

In case B, a "systems manual" which is derived from the copyrighted source language text of a system program would also infringe, even though the second author made no use of a legitimate system manual in writing his derived work.

The following "mirror" pairs of texts for different kinds of software systems illustrate these principles:

Mirror Pairs of Texts:

| <u>Defining Text for:</u> | <u>Program Text for:</u> |
|--------------------------------------|--|
| Job Control Language | Operating System |
| File Interrogation Commands | Information Retrieval System |
| Programming Lan- guage Statements | Compiler for the Lan- guage |
| Interactive System Commands | Interpreter for an Inter- active System |

In each of these cases, if the text of one of the pair of "mirror" texts has been copyrighted, then the unauthorized production from it of a derivative text constituting the other member of the pair may be a copyright infringement.

However, for any of these systems, we should note that it is free and available to all to use any of the copyrighted texts or documents in order to write command statements for using the defined system, or to write programs with statements of the defined language. Thus no permission is

needed to make and use job control cards, to interrogate a file system, to write application programs in the defined language, or to give commands to the interpretive system. Such use is, in fact, a use which is intended for the system-defining documents.

Furthermore, any use of the general *ideas* contained in a text which defines the command structure of a system may be used as *inspiration*, and from these ideas a new, *independent* command structure may be defined. Such a new definition will in turn result in a new and different software system—incompatible with the first. In such a case, the resulting text which defines the new command structure and the text for the new program for the software system, if sufficiently different, would not be infringing so far as copyright was concerned.

In conclusion, we see that copyright does not directly protect an abstract "system" or an abstract computer "language" in itself. Abstractions of such kinds are not protectable by copyright. On the other hand, copyright, as interpreted here, does appear to protect the text or writings which define such systems or language, and it does appear to protect the texts which are computer programs for carrying out such systems or languages. Furthermore, in this view, since copyright protects the making of "any other version" from a copyrighted text, it protects the making of a mirror text from either of the texts of a mirror pair. It is in this indirect sense, and only in this sense, that copyright might be said to protect both parts of a computer software system, or a language/compiler system.

Computer Families

A computer—the real hardware physical object—is characterized by an "instructional set." This is the list of instructions and conventions by which the computer is operated. In recent years, we have seen the appearance of a number of "families" of computers, with each member of a family having pretty much the same instructional set. The most evident instance of this is

the IBM 360 and 370 series (or family) of computers. In the 360 family, although there is a common instructional set, the different models are reputed to be quite different mechanically and electrically in their internal structure. Because of this, if the 360 family can be said to have any actual "identity," it is an identity which is mainly provided by an adherence to the 360 series common instructional set.

Similarly, a common instructional set characterizes the family composed of the PDP-5, 8, 8S, 8I, 8L, and other "8" computers put out by the Digital Equipment Corporation. Again, these computers are all quite different mechanically and electrically. The same can be said of several other families of computers marketed by other manufacturers.

A computer family with a common instructional set has considerable attraction for both the manufacturer and his customers because it generates the development of a large pool of programs all written with the common set of instructions. Such programs are operative, for the most part, on any member of the computer family.

In such a situation, it is possible for another manufacturer to create an imitative computer (although internally quite different) which can operate on the same instructional set. When this is possible, the imitating manufacturer and his customers are able to get a partial "free ride" on the programs and documentation created by others. For example, the IBM 360 series was imitated by RCA, and the Digital Equipment PDP-8 series was imitated by Digital Computer Controls, Inc.

A method for prevention of this kind of imitation seems to be provided by the Copyright Law. The reasoning behind the method is analogous to that developed in the last section, where it was shown that a text which defines a command structure is mirrored in the text of a program for executing the commands, and that copyright of one member of the pair appears to protect the other.

In the case of the computer families, the analog of the "command structure" is

the "instructional set." The text which is the descriptive definition of such an instructional set is clearly a "literary work," and as such it definitely can be copyrighted.

All of the defining and descriptive texts, manuals, and other documentation for the instructional set can be copyrighted. When all the original documentation is copyrighted, then the making of any unauthorized derivative text or set of diagrams of any kind whatsoever is a copyright infringement. In the simplest case, the derivative texts might be new versions of documentation for use with the computers of the imitative family.

However, there is a deeper kind of infringement in the case of the mimicking computers. All computers these days carry out their internal operations by microsteps. In some of these machines, the microsteps are sequenced by microcode from read-only memories. In other machines, some form of wire-in logical sequencing is used to control the microsteps. More recently, machines have appeared in which the microcode in the control memory can easily be changed at will.

Whatever the hardware method, in each case there is some notational system in text form (or graphic form) for the description of the microsteps for carrying out each of the instructions of the instructional set of the mimicking computer. The text in such a notational system for specifying the various microsteps, for each of the instructions, is called a "microprogram" for that instruction. All instructions for the computer can be microprogrammed in this fashion.

Finally, the collection of all such microprograms, for all the instructions, can be transformed and inserted into the bare physical computer. The result of such insertion is to cause the computer to mimic or to "emulate" all of the instructions of the target machine. It can be made to mimic completely the behavior of the target machine.

Such a derived text consisting of the collection of microprograms is also a "literary work," and all of the provisions of the Copyright Law apply to such texts and to

the manner in which they are derived. If original, such texts may be copyrighted.

However, if the texts of the microprograms are not original, or are not produced independently, but are produced without authorization by following some copyrighted text describing an instructional set, or from any related documentation, then the text of such a collection of microprograms may be an infringing derivative work. It matters little whether the manner of derivation is called "translation" or is called "making another version" of the original. Even if the derivative work required a great deal of creative effort in order to reproduce the essential aspects of the original instructions, the derived text may still be an infringement.

Infringement is not avoided if the act of translation is done in private, or if only a private handwritten copy is made, or if none of the text copies are ever sold, or if the purpose of the infringing copy was only to make a wire-in logical sequencing unit, or to write into a read-only memory. Moreover, the profits resulting from such infringement (which may include the profits from the sale or use of such computers) could be claimed by the original copyright owner as part of his legal damages!

From this discussion, it appears that if copyright is consistently applied to all of the published documents describing the instructional set of a new computer, or new computer family, then the copyright owner may have exclusive rights to the making of all derived texts which are microprograms for the imitation or emulation of the computer or computer family whose instructional set is so described. This conclusion appears to be true whether the computer relies on wired-in logic, read-only memory, or an emulative program which is loaded just prior to operation. Copyright does not control computers whose microprograms are created completely independently. However, if such microprograms and their instructions are too similar to some copyrighted instructional set, the claim for "independent" creation would be looked upon

very critically and with considerable skepticism if the case ever came to court.

HOW COPYRIGHT AFFECTS THE COMPUTING COMMUNITY

Why has serious consideration of the applicability of copyright to computer software come so late in the game? Is copyright protection really useful to the goals of the profession? To what extent is the profession now using copyright with computer software? Since computers are new, and since the Supreme Court has never ruled on copyright for computer software, what is the likelihood of the validity of these applications of copyright? Some answers to these and other questions are provided in the following sections.

Why was Copyright Neglected?

The applicability of copyright to software was probably overlooked by computer professionals for a number of reasons. In the earliest days, there was no thought of property rights in programs—the big trick was merely to get the hardware machines to run for more than a few seconds or minutes at a time. Later, as realization of the value invested in software began to develop, software creators began to search for some means of protection.

In my opinion, there then occurred a "communications gap" between the lawyers and the computer professionals. The lawyers, although they were expert in patent and copyright law, had to rely upon the computer professionals for their understanding of software and computer technology. As we all know, computer professionals produce some very odd responses when asked philosophically, "What is a program?" Computer people tended to believe (and still do) that a "program" is epitomized by a deck of punched cards, or by a magnetic tape, or by an "algorithm" (whatever that may mean), by some abstract manipulative process, or by some clever "idea" which is the secret ingredient of the program.

While each of these viewpoints may have some element of truth, none of them is a suitable point of reference for consideration of copyright protection. Carelessness in the use of computer terminology, which is indigenous to our profession, did not help. Also, the variety of compiler and assembly transformations (and the means for transforming), which a program must go through, certainly added to the confusion. Since many of the lawyers were specialists in patent law, there was a great tendency to seize upon the "idea" ingredient of a program.

Whatever the reason, the importance of the text form of a program apparently was originally overlooked. As this paper makes clear, once the importance of the text form to copyright is understood, it immediately becomes apparent how the various embodiments of computer software and its various transformations can be dealt with, and be protected by copyright.

The earliest known use of copyright on programs occurred in 1951 on sorting programs due to Betty Holburton at the Eckert-Mauchly Computer Corporation. There was no attempt to secure registration. More than a decade later, John Banzhaf III, then a programmer, began a series of papers discussing copyright of programs [1, 2] (and followed this interest by going on to study law). In 1964 he applied for registration of claim for copyright of a computer program by filing a magnetic tape together with its printed listing. He thereby made history when a registration of his claim for copyright was granted [10].

A second reason why copyright was so long ignored was the great fascination by everyone with the patent method of protection for computer programs. In the layman's mind, patents have a "status" that copyright lacks. Also the alleged "idea content" of the programs had caught the patent lawyers' attentions, and therefore, to them, patent seemed most suitable, because patent can protect an inventive new idea providing that it is embodied in a system. Unhappily, patents for programs were not issued until just recently. Now, after a

confusing series of court rulings, certain software patents have been granted, but the resulting legal situation remains murky indeed [13].

The fact that some programs have been patented does not mean that all programs are patentable. In fact, only a tiny fraction of the computer programs now being produced would ever be able to meet the traditional high standards of patentability—which require that the invention be truly new and that it be based upon an inventively unobvious new principle or method. Certain kinds of software are also unpatentable for the reason that "methods of doing business" are unpatentable. In contrast to patent, almost any program listing or software documentation which is original can be copyrighted.

A third reason for ignoring copyright has been the view of many lawyers that the "trade secret" method of protection is somehow more suitable for the protection of software in the marketplace. The trade secret method was also probably selected as a consequence of the misplaced belief in the importance of the "idea content" of the software, and therefore of a desire to keep such ideas secret, even though the software was marketed.

Such heavy reliance on trade secret is, I believe, erroneous. A trade secret, in the eyes of the law, is some secret article, method, formula, or item of knowledge which gives the company possessing and using the secret a commercial advantage over its competitors. It must truly be a secret, and its secrecy must be maintained. (Consider the formula for Coca-Cola, which is a true trade secret.)

If an article, such as a program, embodying the trade secret is sold or licensed to a large number of customers, it may no longer be a legally recognized "trade secret"—irrespective of any auxiliary sales contract secrecy provisions. For one reason, sale or license of the article to various buyers who have met the asking price may constitute "publication," and the trade secret status is thus lost. For another reason, purchase and use of the article by a num-

ber of competing companies removes it from the definitional status of a "trade secret," since the article then no longer gives the purchasing company a "commercial advantage over its competitors" who may also have purchased the article.

A fourth reason for overlooking copyright was the widespread belief that computers and their programs were some kind of uniquely new technical development. It was believed that programs were so novel and qualitatively different from anything known before, that the established methods and traditions of commerce and law could not cope with their very unusual nature. As a result there has been much speculation about "new methods of protection." The Patent Office has mentioned the idea of some new kind of junior patent for programs. IBM has suggested a new method of protection which includes a formal disclosure of the principle of operation of the software, a secret deposit of the actual program listing, and an official registration of the software.

In matter of fact, computers and their programs are really not significantly different from other established articles of commerce. Computers are merely complex, expensive electronic engines. Programs are complicated texts. New laws for the protection and regulation of engines and texts are not necessarily required. We should first carefully study the laws that we have, and we should thoroughly explore their means of protection. Only after these existing means have truly been found to be inadequate, and only after we know exactly what we want to do, should we propose radically new legal means for the protection of software.

Copyright is Being Accepted

Three events signalled the beginning of an acceptance of copyright as a preferred method for the legal control or protection of computer programs and software:

- 1) In 1964 the US Copyright Office began issuing certificates of Registration for Claim to Copyright for computer programs. It has also issued "Circu-

lar 61" to provide instructions for applying for registration for copyright of programs [10, 11].

- 2) In 1968 Congress enacted the "Standard Reference Data Act" which for the first time in history gave an agency of the government authorization to obtain copyright on published data and to make such data available by means of copyrighted computer tapes and programs.
- 3) In 1969 International Business Machines Incorporated "unbundled" its software, and it is now making use of copyright on computer "program products," including documentation and object code, which it provides to its customers under lease [12].

If the reader has been alert, he has probably also noted during the past year or two that more and more items of computer documentation, as put out by the manufacturers, includes a copyright notice. Many software suppliers, in addition to IBM, are also beginning to use copyright on their own program products. Interactive computer systems now very often print out a copyright notice as the first action in their man/machine dialog.

Textbooks on computer science, when making use of copyrighted source, are beginning to be meticulous in using statements such as "used by permission," together with a repetition of the original copyright notice of the source. Art work for etched circuit boards for computer modules is now being copyrighted. In addition, the art work for the masks for microcircuit LSI (large scale integration) devices is now being copyrighted—and in some cases the statutory copyright notice appears microscopically on the final tiny silicon chip! The largest of the governmental agencies (DOD, NASA, HEW) are now beginning to pay attention to copyright in their research reports.

As more computer people and computer-based businesses become aware of the essential role that copyright can fill in providing needed protection to all kinds of software and documentation, there will be

an accelerating use of copyright in the marketing of computer products.

Program Portability and Standardization

"Program portability" refers to the desirable property of a program or software system which allows it to be moved from one brand of computer to another brand or model with the expectation that it will run satisfactorily with a minimum amount of change, disruption, or grief. As such, program portability is very closely related to "standardization" and to "machine independent programming." Each of these has been named as a desirable objective in our profession.

Early in the computer game, computers were so minimal in their capabilities that, to secure efficiency, programs had to be tightly written around the idiosyncracies of the specific hardware. The resulting programs were anything but portable.

With contemporary computers, and with the use of the high-level programming languages, true program portability is possible—except for the fact that neither the hardware vendors nor many of the software vendors actually seem to want it!

Hardware vendors apparently don't want program portability because lack of portability tends to lock present customers into the vendor's product line. If programs were easily portable, the customers would be, too. Managers of computer installations could switch almost at will from one vendor's computer to another's as the pricing became advantageous. Software vendors don't seem to want program portability either, since the lack of portability provides a certain degree of protection against the complete loss of control of their software in the case of misappropriation.

In my opinion, these reasons have caused both hardware and software vendors to cultivate or encourage the widespread fiction that attainment of true program portability is an extremely difficult or impossible technical goal. Equipment vendors have facilitated this fiction by refusing to standardize their compilers and operating systems in a sufficiently meaningful fashion.

As usual, the users—who pay the bills—have been the victims. True program portability would be very desirable for the users, both to eliminate expensive duplication in the production of software, as well as to allow them to move their work freely from one brand of computer to another, and onto future "new generations" of machines.

It is my belief that copyright, by providing easy and effective protection of computer software, will have two desirable by-product effects, both of benefit to the profession.

The first is that it will accelerate the standardization of programming languages and their compilers. This will come about because languages and compilers can be protected, through copyright of their documentation, and therefore various kinds of unauthorized development of language dialects and deviant compilers can be controlled by the original language developer.

The second desirable byproduct effect of copyright is that it will cause the development of truly portable program packages, which can run on any machine of adequate size or capability. This development will probably be led by the independent software houses. Once they can easily protect their programs, they will be moved by marketing considerations to develop program packages which are easily portable and therefore widely usable. They will then begin to drop their present heavy reliance on various kinds of "confidentiality" and "trade secret" methods of software protection. This will be attractive to all. Secrecy is inherently annoying to the buyer. Also the "trade secret" method is legally very dubious for software which is to be sold to a variety of users.

With copyright, they may also be able to drop their present reliance upon "planned obsolescence" as an indirect means of software control.

Are These Conclusions Valid?

This paper has explored the applicability of Copyright Law to the protection of computer software. The exploration has been conducted with an orientation to the rami-

fications of computer technology. The conclusions presented here all have a reasonable and logical foundation in the law as it has been applied to analogous cases in other fields. Most of the conclusions presented here are supported by the views of other commentators on copyright. A few of the conclusions are novel to this paper, and are a result of its deeper technological orientation.

However, to this date, no actual cases involving copyright and computer software seem to have reached the courts. In such a situation, how should we view the matter of the validity of copyright as applied to software?

Some of the conclusions in this paper have a stronger foundation than others. Perhaps the weakest conclusion is the view that running a program is an infringement of copyright. Although this topic has a number of puzzling legal points, substantial critical opinion seems to support the conclusion that performing a copyrighted program is actually an infringement [3]. Certainly it would be difficult to get a computer ready for running a program without incurring infringement through copying. A similar puzzle appears with respect to the possibility of infringement from running a microprogrammed computer where the microprogram is derived from copyrighted material.

On the other hand, it is logically clear (though startling at first) that the preparation of a microprogram text script would infringe if it were derived from a copyrighted text defining the machine instructions. The infringement of FORTRAN to ALGOL translations is also clear, once the proposition has been stated. It is only a minor further step to conclude that the preparation of a text for a system program from a copyrighted definition text for a command structure is also an infringement. Likewise for language definitions, and the derived texts for their compiler programs.

For the most part, all these conclusions depend upon the observation that the infringing act consists either in the copying of text or in the preparation of some kind of derivative text from an original copyrighted

text. What the texts are "about," what they are to be used for, or how or in what way they are to be used, is generally of little actual concern to the Copyright Law. The fact that the copyrighted texts in this case come from the field of computer technology is therefore of little consequence in the application of the law.

It is fallacious to believe—as many people seem to do—that a law has no force in a specific new situation until the Supreme Court of the United States has finally ruled upon it and has upheld its application to the specific situation. The opposite is nearer the truth. The Supreme Court would never have the time to rule in this fashion on all the new kinds of situations that arise. As a matter of policy, it refuses to review cases which do not in its opinion involve some major national problem or some actual conflict in the legal interpretation of our laws. Such is not the case with software and copyright. For the most part, the words of the Copyright Law are very clear in their application to the text matter of computer software.

Enacted law, such as the Copyright Law, applies to new situations under reasonable interpretation from similar or analogous cases and situations. In our computer field, the courts have an abundance of precedent from closely analogous situations. The most relevant of these involve infringement of literary and dramatic texts, and infringements resulting from their various transformations and manners of performance.

A legal interpretation applies until it is overturned by a consensus of the lower courts, or (and only in a microscopic fraction of the cases) by final review by the Supreme Court.

Neglect of a law does not make the law invalid. Also, ignorance of the law is no excuse. The fact that rights in copyright have not been widely relied upon in the protection of computer software, or have not been widely known or accepted by the computer profession, does not mean, therefore, that Copyright Law is inapplicable to computer software. The reader need only be reminded of the recent rediscovery and revival of the long-neglected 1899 Rivers and Harbors Act

(Refuse Act) which dealt with the dumping of refuse into waterways. Pollution is now being stopped, and heavy fines are being levied under this old act for the first time!

The detailed interpretation of enacted laws by the courts is a dynamic matter. It tends to reflect the desires and practices of the community concerned. This has been true of the Copyright Law in the past, and it will be in the future. In our situation, if the business and user community shows by its practices a desire to see programs and software protected as business property, so that they may be freely traded and marketed like other business goods, we may anticipate that the courts will tend to follow this lead, and will be inclined to interpret the law in a fashion that will secure these goals. On the other hand, if the community as a whole shows a desire to minimize the application of property rights to computer software, the court will still tend to oblige, and will be inclined to interpret the Copyright Laws in a very narrow fashion. I believe that this second alternative would be most unfortunate because it would tend to leave software unprotected, even though it will be the major cost element for computer operation in future years.

Strategy for the Copyright Owner

All predictions about the application of the law have a degree of uncertainty. This is particularly true both at the borderlines of settled law, and in new territories. In varying degrees, such uncertainty affects predictions about copyright law and software.

Yet, despite uncertainties over the future course of the law, there is a useful strategy for the copyright owner to follow. Except for software which must be kept secret, the strategy is to make claim for copyright (by publication with notice) for software and documentation, and then to commercially exploit all the rights (e.g., rights in copies, translations, versions, etc.) that derive from ownership of the copyright.

This strategy is a consequence of the following three points. First, claiming copyright is a very inexpensive thing to do, and

it incurs few, if any, disadvantages. Second, the commercial advantages that may flow from proper claim to copyright can be very substantial. Third, the copyright owner, with his prior claims to rights in copyright, is in an advantageous position with respect to any future clarification of copyright law in any uncertain areas. These points will now be discussed.

The disadvantages to claiming copyright (by publication with notice) are minimal. A frequently cited disadvantage is that copyright requires "publication" and that any "secret idea" contained in the software is disclosed by such publication. This is true, but those offering this objection fail to realize that any general sale or distribution of the software also constitutes publication. Sale also causes loss of any "trade secret" status which the idea contained in the software might have had.

Claiming of copyright incurs no other apparent disadvantage. At any later date, if the owner does not desire to maintain his rights in copyright, he simply does not enforce his rights.

It is fortunate that making claim to copyright is so simple and incurs no real disadvantages, since the claims must be made at the time of initial publication, and at this early date the full value of the software development may not be known. If claiming of copyright were, instead, very complicated or expensive, it might not be done.

As a practical matter, to insure against mishaps (in view of the many photocopy machines and the possibility of unintended publication) it is desirable to make a practice of putting the statutory copyright notice on all internal copies of software and documentation (that are not to be maintained truly secret) even before the time of publication.

The commercial advantages that flow from a claim to copyright depend upon the marketing of any or all of the rights in copyright that have been discussed in this paper. Since software is a very expensive commodity, and because copyright secures its ownership, the commercial advantage of trading in these rights may be considerable. But what about the areas in which copyright is

less certain? Can these rights also be traded?

The copyright holder can trade in such rights. He can do so because his copyright covers all rights, both those which are now certain and well-established, as well as those which are less certain, but which in the future may prove to be valid in some court of law. In the latter case, infringement of the rights through unauthorized use extends from the date of copyright, rather than from the date of any clarifying court decision. Moreover, damages may be collected for any acts which take place within three years preceding the date of any action brought by the copyright owner. It is seen that the advantage is with the copyright owner, even in these uncertain areas.

A business transaction in regard to any specific right in copyright (e.g., permission to make a translation) is therefore logically equivalent to a transaction which frees the buyer, for some monetary payment, from any future claims for infringement resulting from the buyer's use of the right. Of course, where the specified rights are to some degree uncertain, the amount of payment would be somewhat less than where the rights are well-established and certain. In either case, any responsible organization desirous of using the particular rights in copyright would prefer to settle these matters beforehand, that is, before the use of the rights took place. In doing so, they would be free from any anxieties in regard to unexpected future claims for damages, or possible lawsuits, as a consequence of their reactions.

In connection with their "licensed program products," IBM employs a mixed strategy of software protection which should be considered by those seeking maximum security for their investment in software [12]. All programs and documentation of IBM licensed program products, both in printed and machine-readable form, are copyrighted and bear the required statutory copyright notice. Such copyrighted programs and documentation are supplied to IBM customers only after the customers have signed a "License Agreement for IBM Program Products." In addition to specifying the serial number of the particular computer with which these program products must be used,

and the charges for the products, the license agreement binds the customer "not to provide or otherwise make available any licensed program and/or optional material . . . to any person other than Customer . . . without prior written consent from IBM." At the end of the license period, the programs and documents are required to be destroyed. Furthermore, all licensed printed materials bear, at the top and bottom of each page, the legend: "Licensed Materials. Property of IBM". Clearly, ownership of the IBM program products and licensed documents is not transferred.

By this approach, it would appear that IBM attempts to secure the advantages of both copyright protection and the very considerable additional protection which results from the contractual promises of nondisclosure. The validity of IBM's method, especially in regard to certain copyright aspects, does not seem to have been tested in court.

Since the acquisition of statutory copyright requires "publication" of the materials and documents so protected, IBM's method is a little puzzling until it is analyzed. "Publication" of a document or program for copyright purposes occurs at the time when availability to the public of the document or article is announced or takes place. However, Copyright Law makes no legal requirement as to the price of the copyrighted article, or need for transfer of ownership, or the nature of any concomitant legal restrictions which a member of the public may be asked to subscribe to in order to secure access to the published article. In other words, IBM appears to take advantage of the fact that an open sale or free gift of the copyrighted article is not necessary for "publication" to occur.

Strict secrecy is not guaranteed by this method, since the copyrighted materials are available for public inspection at the Copyright Office in consequence of the requirement for deposit of copies as part of the copyright registration. The Copyright Office will not permit copying of the materials on deposit, so that complex, long and detailed listings of programs and operation manuals are, in effect, protected by their very length and complexity. The Copyright Office at any

time can demand registration and deposit (on pain of fine and loss of copyright); however, as a result of judicial interpretations, and if not challenged by the Copyright Office, one can defer registration of claim of copyright and deposit of the copyrighted materials until the copyright must be defended by lawsuit. When such delay is possible, security of the material during the initial period of marketing is enhanced.

Journal Publication

Announcement of new results in software systems and programming methods is often made via publication in one of the scientific journals. Contrary to popular belief, such publication does not dedicate the author's writing to the public, since the contents of almost all scientific journals are now copyrighted. While the "ideas" the author expresses in his paper are made public property by such publication (there is no method other than secrecy to hold on to abstract ideas), the ownership of the actual writings in any paper so published goes to the person or organization which secures copyright on the paper in the course of its publication.

In accordance with standard scholarly practice, an author submits his paper to a publisher for publication with the assurance to the publisher that his paper is "original" and that it "has not been published elsewhere." By this act, the author: 1) indicates that the paper is a valid candidate for copyright, and 2) turns over ownership of his writing to the publisher for purposes of publication. The legal consideration for the transfer of ownership is the value to the author of having his work published. When the issue of the journal is published, with several such papers included in its contents, the publisher's single copyright notice for the issue secures copyright to all the contents of the issue, including copyright to each of the submitted papers. Thus the publisher comes to own the copyright in the author's writing.

An alternative practice, widely used by authors knowledgeable in copyright (e.g., used by lawyers in writing for any sort of scholarly or other publication), is for the author to copyright his paper himself. There are two ways to do this. In either way, the

author supplies his own statutory notice, for example, "Copyright © 1975 Calvin N. Mooers" on the first page of the paper. Then, by the first way, copyright is secured when the paper is published with the author's notice in the journal. By the second way, the author first "publishes" the paper himself, say in an "edition" of ten copies, thereby securing copyright. He then authorizes the journal to reprint his copyrighted paper, including his copyright notice. In either way, the author makes the application for registration of his claim to copyright, and the author comes to own the copyright.

The responsibility for defense of copyright lies with the owner of the copyright. Likewise, the authority for giving permission for use of any or all rights under the copyright falls to the owner of the copyright.

Therefore, if the publisher, rather than the author, is the owner of the copyright in the paper, the author must rely almost totally upon the publisher to protect his interests. Ordinarily, the publisher will attempt by persuasion to discourage gross abuses of the author's interests—providing they come to the publisher's attention in time. However, unless some offense deeply affects the publisher's own interests, the publisher will understandably be very reluctant to engage in any further defense of the author's interests, especially whenever such defense might involve any substantial expense. Because the author has actually transferred all his legal rights in his writing, in this situation he has little more than "moral suasion" to bring to bear upon the publisher, or upon any offender.

For many kinds of papers, this situation need cause little concern to an author or to his organization, since the writings in the paper may have no substantial continuing importance of any commercial (or any other) sort. Papers of this kind, for example, report work in progress, tell of a partially developed idea in some academic project, superficially describe some commercial offering, provide news of the profession, and so on.

Another class of papers does have a substantial continuing importance to the author or his organization, and such papers deserve a very careful treatment of their rights in copyright. Papers of this category include,

for example, such things as a careful study on some topic which is planned for later inclusion as a chapter in a book by the author; a software system description where, in order to convey the "idea" of the system, a substantial portion of the "expression" constituting the definition of the system must be included in the paper; or an announcement of a proprietary new language where the description of the nature of the language (the "idea") is most efficiently handled by including actual definitions (elements of "expression") of the statement types of the new language.

For valuable papers of this kind, several courses of action are open to authors. The simplest, if the publisher is willing, is for the author to supply his own copyright notice for his paper, and thus directly to secure for himself the ownership of the copyright. When this is done, the publisher will usually ask for permission for reprinting the paper, such as for reprinting any out-of-stock issues of the journal, etc. By employing this option, the author can directly protect and deal with his rights in copyright without further complication.

The second course of action is for the author to arrange with the publisher before publication for legal transfer of the copyright back to the author, to take effect after publication. The publisher will probably ask for, and should be given, certain rights for limited reprinting. Arrangements for such assignment of copyright should preferably be made before the author releases his paper for publication. With such assignment, the author is again owner of copyright in his writing, and can directly protect his rights.

If the publisher will not agree to either course of action, nor to any satisfactory alternative, and if ownership of the paper is considered to be of substantial continuing value to the author or to his organization, then the author should seriously consider submitting it for publication in some other journal.

CONCLUDING REMARKS

We all need to give some serious thought to matters of software and copyright. By our practices in the use of copyright, we can es-

tablish the attitudes and bases for future court interpretations. In our current activities, there are obviously numerous situations in which there have been unintended transgressions—situations in which we shall now have to seek prior permission in order not to infringe. Substantial rights in software and documentation have gone unrecognized all too often, even by the owners of such rights, because of the owner's lack of awareness of what his rights might be. Valuable rights have often provided little return, and rights have gone undefended.

There has been widespread complaint and frustration because of the popular belief that we lacked an effective means for control of software property, yet in copyright there has always existed an excellent mechanism, just waiting to be used for such protection.

Users and consumers of software must now familiarize themselves with the nature of property rights in software under the Copyright Law, so they will be aware of their obligations to the rights of others under these laws. During a period of transition, while the computer profession becomes familiar with copyright, infringing activities will be a delicate matter. Most infringement will be unintended or inadvertent. Users of software should take pains to avoid such infringement, and should be prepared to stop activities which are shown to be infringing. Software proprietors should do all they can, through education and persuasion, to steer users of software away from infringing activities. The force of the law and the courts should be used only as a final resort, and only against the most recalcitrant offenders.

The several major messages of this paper can be summed up in a few sentences:

Copyright Law exists; it is broadly applicable to software, and it is ready and waiting to be used. If you are a creator of software systems or programs, the method of copyright provides a wide scope of protection for the textual expression of your creations. Put on a copyright notice. You certainly have nothing to lose, and you may have much to gain. Copyright may provide you with that additional increment of protection you need.

If you are working with someone else's copyrighted piece of text involving software

or programs, be careful. Don't copy, translate, perform, or make any adaptation or version of the copyrighted software, in public or in private, without the permission of the copyright owner. Your alternative might be to learn all about copyright the expensive way.

For definitive advice on what to do, see your lawyer and rely on his counsel. Moreover, when you visit him, it just might be a good idea to bring this article along with you!

REFERENCES

The following references are addressed primarily to the legal readers of this paper. They are far from complete. They are included to give a starting point for further research on this topic.

- [1] BANZHAF, JOHN III. "Copyright protection for computer programs," in *ASCAP Copyright Law Symposium*, Vol. 14, Columbia University Press. (1966), 118-179.

This is a pioneering comprehensive discussion.

- [2] BANZHAF, JOHN III. "Copyright protection for computer programs," in *The Law of Software, 1968 Proc.*, Computers in Law Institute, George Washington Univ., Washington, D. C.

A shorter and more recent discussion by the same author.

- [3] GOLDBERG, MORTON DAVID. "Copyright for computer programs—yes, but," in *The Law of Software, 1969 Proc.*, Computers in Law Institute, George Washington Univ., Washington, D. C.

A penetrating assessment.

- [4] ISKRANT, JOHN "The impact of multiple forms of computer programs on their adequate protection by copyright," in *ASCAP Copyright Law Symposium*, Vol. 18, (1970), 92-134.

Discusses tapes and machine-readable forms.

- [5] Title 17, US Code, Copyright, reprinted in pamphlet form as Bulletin No. 14, "The Copyright Law of the United States of America," available for 45 cents from the Government Printing Office, Washington, D. C. 20402.

- [6] Wihtol v. Crow, 199 F.Supp. 682, reversed 309 F.2d. 777 (8th Circuit 1962).

Defendant made an arrangement and copies of copyrighted music for church performance. A ruling of "fair use" was reversed and statutory damages were assessed.

- [7] Henry Holt & Co., Inc. v. Liggett Meyers Tobacco Co., 23 F.Supp. 302 (E.D.Pa. 1938).

Use of three sentences from a book held infringing.

- [8] Mazur v. Stein, 347 U.S. 201 (1954).

The Supreme Court stated: "We find nothing in the copyright statute to support the

argument that the intended use in industry of an article eligible for copyright bars or invalidates its registration. We do not read such limitation into the copyright law."

- [9] Addison-Wesley Publishing Co. v. Brown, 207 F.Supp. 678, 233 F.Supp. 219 (E.D.N.Y. 1963).

Preparation and publication of a derivative book containing answers to physics problems in copyrighted book held to be an infringement. Court noted that the offending book could be taken as a manner of "translation" of the source, as the plaintiffs argued, but the decision of infringement was rendered on the broader ground that "unfair use has been made by the defendants of the product of the plaintiff's original and useful work," and thus the defense of "fair use" was rejected. This decision contains revealing discussion of the transformations or the "making of other versions" of the content of a technical work.

- [10] *New York Times*, May 8, 1964, news story (p. 43 cols. 4-6).

Tells of first copyright registration for computer program issued to Banzhaf.

- [11] Circular 61, Computer Programs. Available without charge from the Copyright Office, Library of Congress, Washington, D. C. 20540.

Provides directions for copyright registration of computer programs

- [12] International Business Machines Corporation, Form 120-2083 "Instructions on Copyright Notice"; also see Form Z120-2065, "License Agreement for IBM Program Products." Available from IBM.

Gives instructions for placement of copyright notices on program products, maintaining catalog of holdings, notice on object code modules, and other restrictions imposed on licensees.

- [13] *Gottschalk v. Benson* 409 U.S. 63, 175 U.S.P.Q. 673 (1972).

The US Supreme Court considered an appeal of an application for patent for a method of decimal to binary conversion conducted inside a computer. The Court called the method a computer program or a formula, and then denied a patent saying that such a patent "would wholly preempt the mathematical formula and in practical effect would be a patent on the algorithm itself." Two years later a lower court (Court of Customs and Patent Appeals), in the case of *In Re Johnston* 183 U.S.P.Q. 172 (1974), despite this Supreme Court decision, granted a patent by a 3-2 decision to an accounting system invention which is described by means of a computer program. Because of the present confused state of the law, it is difficult to predict the future course of patentability of computer software.

- [14] Circular 20, "Fair Use" of Copyrighted Works." Available without charge from the Copyright Office.

- [15] Circular 64, Letters, Diaries, and Similar Personal Manuscripts. Available without charge from the Copyright Office.