00-9185

IN THE UNITED STATES COURT OF APPEALS
FOR THE SECOND CIRCUIT

UNIVERSAL CITY STUDIOS, INC., *et al.*

Plaintiffs-Appellees

v.

ERIC CORLEY, A/K/A EMMANUEL GOLDSTEIN AND 2600 ENTERPRISES, INC.

Defendants-Appellants

SHAWN C. REIMERDES, ROMAN KAZAN

Defendants

On Appeal From The United States District Court
For The Southern District Of New York

BRIEF OF *AMICI CURIAE*

DR. HAROLD ABELSON; DR. ANDREW W. APPEL; DR. DAN BONEH; DR. EDWARD W. FELTEN; DR. ROBERT HARPER; MR. ANDY HERTZFELD; DR. BRIAN KERNIGHAN; DR. MARVIN MINSKY; DR. JAMES MORRIS; DR. P.J. PLAUGER; DR. JAMES C. REYNOLDS; DR. RONALD RIVEST; DR. AVI RUBIN; DR. BARBARA SIMONS; DR. EUGENE H. SPAFFORD; MR. RICHARD STALLMAN; AND DR. DAVID S. TOURETZKY

**IN SUPPORT OF APPELLANTS AND
REVERSAL OF THE JUDGMENT BELOW**

JAMES S. TYRE
California State Bar Number 83117
LAW OFFICES OF JAMES S. TYRE
10736 Jefferson Blvd., #512
Culver City, California 90230-4969
310-839-4114

Counsel for *Amici Curiae*

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases:

Statutes and Constitutional Provisions:

<u>Other Authorities, Internet Sites:</u>

INTERESTS OF *AMICI CURIAE* [1]

All parties have consented to the filing of this brief. FRAP 29(a).

Dr. Harold Abelson is Professor of Computer Science and Engineering at MIT, a Fellow of the IEEE, and a winner of the IEEE Computer Society's education award. Together with Gerald Sussman, Prof. Abelson developed MIT's introductory computer science subject, "Structure and Interpretation of Computer Programs," which has had a world-wide impact on university computer-science education. This subject is organized around the notion that a computer language is primarily a formal medium for expressing ideas, rather than just a way to get a computer to perform operations. Professor Abelson was deposed in the case, but did not testify at trial.

Dr. Andrew W. Appel is Professor of Computer Science at Princeton University, where he has been on the faculty since receiving his Ph.D. in 1985. He does research in computer security, virus prevention, programming languages, and compilers. He is a Fellow of the Association for Computing Machinery ("ACM") and served for several years as Editor in Chief of the journal, ACM Transactions on Programming Languages and Systems. Dr. Appel testified as a non-party witness.

---

[1] Affiliations are listed only to identify the *amici*, whose views expressed herein do not necessarily coincide with those of their respective universities or employers.

Dr. Dan Boneh is a Professor at the Department of Computer Science at Stanford University. His research focuses on cryptography, specifically the security of cryptographic primitives and their application in real world systems. At Stanford he is leading a number of systems security projects on topics such as intrusion tolerance and security applications for handheld devices.

Dr. Edward W. Felten is an Associate Professor of Computer Science, and Co-Director of the Secure Internet Programming Lab, at Princeton University. His main research interest is in computer security, especially relating to the security of software running on computers and electronic devices used by consumers. He served as the primary technical expert for the Department of Justice in the U.S. v. Microsoft case, and testified as an expert witness in this case. He frequently uses computer code as a medium of expression in both teaching and research.

Dr. Robert Harper is a Professor of Computer Science at Carnegie Mellon University. His research is on programming language design and implementation. He is a principal co-designer of the Standard ML programming language and a co-inventor of the LF Logical Framework.

Andy Hertzfeld received a B.S. from Brown University and a M.S. from UC Berkeley, both in Computer Science. He became a key member of Apple's original Macintosh team in February 1981, designing and implementing a substantial portion of the Macintosh's breakthrough system software. After leaving Apple, he co-founded three technology companies: Radius (1986), General Magic (1990) and Eazel (1999). He works at Eazel to help make free software easier to use for a non-technical audience.

Dr. Brian Kernighan is a professor in the Computer Science Department at Princeton University. He was until recently head of the Computing Structures Research Department at Bell Labs, where he did research in programming languages, software tools, and user interfaces. He is the co-author of a number of widely-used computer books and programs.

Dr. Marvin Minsky is Professor in the MIT Media Laboratory and CS departments, where he has made major advances in computer science, artificial intelligence, physics, psychology, and mathematics, and is a member of both the National Academy of Engineering and National Academy of Sciences. He invented the widely used Confocal Microscope. He has won the Japan Prize, the Rank Prize, the ACM Turing award, and the Franklin Medal.

Dr. James Morris is the Dean of the School of Computer Science at Carnegie Mellon University. He was also on the faculty of the University of California at Berkeley. He worked at the Xerox Palo Alto Research Center for ten years and founded a consulting firm specializing in interactive product design.

Dr. P.J. Plauger is President of Dinkumware, Ltd., which develops and licenses standard-conforming libraries for C, C++, and Java. He has authored or co-authored over a dozen books on computer programming, some of which are considered classics. He has also written numerous articles on programming for various trade publications. He helped develop the current standards for Posix, C, and C++.

Dr. John C. Reynolds is a Professor of Computer Science at Carnegie Mellon University, a member of the IFIP Working Group 2.3 on Programming Methodology, and a Fellow of the ACM (induction in March 2001). He has written two books, "The Craft of Programming" and "Theories of Programming Languages", and numerous papers. His research interests are programming languages, formal program specification and proof, and mathematical semantics.

Dr. Ronald Rivest is the Viterbi Professor of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, and a leader of MIT's Cryptography and Information Security research group. Dr. Rivest is an inventor of the RSA Public Key Cryptosystem, and a founder of RSA Data Security.

Dr. Avi Rubin is a Principal Researcher at AT&T Labs - Research and a member of the board of directors of USENIX, the Advanced Computing Systems Association. He also has an appointment as an adjunct professor in the Computer Science department at NYU.

Dr. Barbara Simons was President of ACM from 1998 - 2000. ACM, with about 80,000 members internationally, is the oldest educational and scientific society of computer professionals in the world. Simons is a Fellow of the American Association for the Advancement of Science and of ACM. She founded the U.S. Technology Policy Committee of ACM in 1993, and has served as its chair since that time, except for the period of her presidency. Among other posts, Simons served as a Research Staff Member at IBM Research. She is a member of the President's Export Council's Subcommittee on Encryption. Dr. Simons was deposed, but did not testify at trial.

Dr. Eugene H. Spafford is a professor of computer sciences and of philosophy at Purdue University. He is the founder and director of the Purdue CERIAS (Center for Education and Research in Information Assurance and Security), a national center of excellence and the nation's foremost academic center in this field. Dr. Spafford is a Fellow of the ACM, the IEEE, and the AAAS, and he was the year 2000 recipient of the NIST/NSA National Computer Software Security Award. He currently serves on the Computing Research Association board of directors, and on the U.S. Air Force Scientific Advisory Board.

In 1984, Richard Stallman launched the development of the free operating system, GNU. In 1992, the free kernel, Linux, was combined with GNU; the resulting

GNU/Linux system has some 20 million users. Stallman is the principal author of the GNU C Compiler, GNU Emacs, and other GNU programs. He received the Grace Hopper Award from the Association for Computing Machinery for 1991. In 1990 he was awarded a MacArthur Foundation fellowship, and in 1996 an honorary doctorate from the Royal Institute of Technology in Sweden.

Dr. David S. Touretzky is a Principal Scientist in the Computer Science Department at Carnegie Mellon University, and the author of a popular textbook on the Lisp programming language. Dr. Touretzky also is the creator of the "Gallery of CSS Descramblers" web site, and he testified as an expert witness at trial.

SUMMARY OF ARGUMENT

> It cannot seriously be argued that any form of computer code may be regulated without reference to First Amendment doctrine. The path from idea to human language to source code to object code is a continuum. As one moves from one to the other, the levels of precision and, arguably, abstraction increase, as does the level of training necessary to discern the idea from the expression. Not everyone can understand each of these forms. Only English speakers will understand English formulations. Principally those familiar with the particular programming language will understand the source code expression. And only a relatively small number of skilled programmers and computer scientists will understand the machine readable object code. But each form expresses the same idea, albeit in different ways.

Universal City Studios v. Reimerdes, 111 F.Supp.2d 294, 326 (S.D.N.Y. 2000).

The court below properly recognized that computer code, whether in the form of source code or object code, is a form of expression, subject to First Amendment scrutiny. However, based in part on the court's ill-considered concerns about the "functionality" of code, *id.* at 332-33, the court ruled that code is entitled only to the intermediate level of scrutiny set forth in *United States v. O'Brien*, 391 U.S. 367 (1968), rather than to the strict judicial scrutiny which should be afforded to this constitutionally protected speech.[2] These *amici*, a cross section of some of the leading computer scientists and programmers in the country, will argue that the "functionality" of computer code does not defeat its essential expressive nature, and that minimizing the level of judicial scrutiny finds no support in the law or in how code actually is used. Specifically, these *amici* argue that:

1. Computer codes are text languages, as expressive as any text language, with dialects, grammar structures and nuances, and thus are entitled to the same level of First Amendment scrutiny as any natural text language, such as English.

2. Among academics and programmers, communicating in computer code (in addition to or in lieu of a natural language) is essential "[t]o promote the Progress

---

[2] *Sable Communications of Cal., Inc. v. FCC*, 492 U.S. 115, 126 (1989) ("The Government may, however, regulate the content of constitutionally protected speech in order to promote a compelling interest if it chooses the least restrictive means to further the articulated interest").

of Science and useful Arts ...", the core purpose of copyright. U.S. CONST., art. I, § 8 (Copyright Clause). Minimizing First Amendment protections given to code would deter, not promote, the progress of science.

3. Copyright law already recognizes that code, both source and object, may be copyrighted as a literary work or an original work of authorship. 17 U.S.C. § 102(a)(1). It does not take a leap of faith to say that a copyrightable literary work is a work entitled to full First Amendment protections, regardless of its functionality.

4. Based on all of the foregoing, the functionality of code should not, and does not, limit the First Amendment protection to which it is entitled.

ARGUMENT

I. A Quick Primer on "Source Code" and "Object Code."

Generally, source code and object code are the languages used by humans to express human ideas in forms understandable to and usable by computers. Source code languages, in which most programmers write, involve a series of letters and symbols, with specific vocabulary, syntax and expository conventions.[3] Source code languages sometimes are referred to as "high level" languages, as they are not far

---

[3] Just as there is no single language spoken by humans, there is no single computer programming language. Among the thousands of programming languages, widely used ones include Fortran, Cobol, C, Visual BASIC, Perl, Java and JavaScript.

removed from the languages which humans speak. Three simple examples of the Visual BASIC programming language, each of which should be understandable by those who do not know programming, are as follows:

```
If Month(Date) = 2 And Day(Date) = 12 Then
  Print  "Don't forget that today is your anniversary."
End If

If warning_message_type = "hurricane" Then
  SendToAllEmployees("Warning:  hurricane detected.  Proceed to
     storm shelter immediately!!!")
End If

If fldEmployeeName = "John Doe" Then
  fldEmployeeSalary = fldEmployeeSalary + 100000
End If
```

While an example of the Perl programming language might look like this:

```
    my ($xor_len) = $key_length{$request};
    my ($file_key) = substr($cipher_key, 0, $xor_len);

    while (read(FILE, $xor_block, $xor_len)) {
        $plain_text = $file_key ^ $xor_block;
        $plain_text =~ s/\s+$//;
        $plain_text =~ tr/A-Z/a-z/;
        print $plain_text,"\n";
    }
}
```

We chose this snippet of Perl for two reasons. First, when compared to the Visual BASIC, it is apparent that more symbolic characters and fewer natural-looking words are used, illustrating the variety of programming languages. Though it looks less like a natural text language, its meaning is as clear to those who read Perl as is this sentence to those who read English.[4]

The second reason is the more thematic. The snippet is from a program which decrypts the encrypted list of URLs blocked by the filtering software application known as X-Stop, the use of which in a public library was found to be unconstitutional in *Mainstream Loudoun v. Board of Trustees*, 24 F.Supp.2d 552 (E.D.Va 1998). In part by showing that X-Stop blocked valuable Internet speech, Plaintiffs prevailed, but on the face of 17 U.S.C. § 1201(a)(1), such a decoder might be unlawful as a circumvention measure. In October 2000, the Librarian of Congress completed the first rulemaking mandated by § 1201(a)(1). One exemption was for "[c]ompilations consisting of lists of websites blocked by filtering software applications." 37 CFR Part 201. If the LOC rulemaking now legitimizes the use of a program such as the X-stop decoder to compile such lists, it makes no sense under any theory of copyright law that § 1201(a)(2) proscribes the distribution of such a program, but that is the effect of § 1201(a)(2), the section at issue in this action.

---

[4] If "$plain_text = $file_key ^ $xor_block" seems unapproachable, consider what those not trained in the language of legal citation would make of "111 F.Supp.2d 294, 326 (S.D.N.Y. 2000)." Each is meaningless to those unfamiliar with the language; but each is more precise and compact for those who do understand than would be an English narrative equivalent.

Object code, or machine code, sometimes referred to as "low level" languages, have a more simple structure than source: a sequence of instructions, each of which is a sequence of fields, each of which has a fixed size. However, there is no necessary bright line between source and object code. In Defense Trial Exhibit BBE, Dr. David Touretzky of Carnegie Mellon University, a sponsor of this brief, wrote:[5]

> 1) All computer code is human readable. Some forms are simply more convenient to read than others.
>
> 2. All computer code is expressive. Many of the ideas expressed in C code are also expressed in the assembly language code that results from compiling that C code, and again in the binary machine language that is the output of the assembler. Some content may be lost, e.g., source code comments are typically not preserved in object code, although variable names may be. But some ideas that are only implicit in the source code may be made more apparent in the object code, such as how a particular sequence of actions should be best expressed in terms of processor operations in order to obtain maximum performance from the machine.
>
> 3) All computer code is executable. In some instances it may be advantageous to transform the code into another form first, but transformation is by no means mandatory. An interpreter can be employed instead. Interpreters are in common use in computer systems.

[5] Dr. Touretzky's related trial testimony is at Tr. 1086-91. At 111 F.Supp.2d 326, n.183, the court expressed indebtedness to Dr. Touretzky for his explanation of the interplay between various kinds of computer languages, though it differed with Dr. Touretzky on the applicability of the First Amendment to DMCA and the suppression of computer programs.

4) "Source" and "object" are relative terms, not absolute categories.[6]

Indeed, even “source code” itself is a relative term. The notion of what it is changes rapidly as developments in computer technology occur, and it is particularly important for the Court to understand that point as it considers even how to define the term, let alone how, if at all, it may be restricted. A more proper definition of source code may be simply that it is that subset of human expression which computers can interpret and execute. As time progresses, computers can interpret increasingly larger subsets of human expression; thus, restricting the expression of source code restricts increasingly larger subsets of human expression.

Consider the following: if one has watched even a few episodes of “Star Trek”, one is familiar with the concept of humans speaking directly to and interacting with computers. In such a scenario, the source code for the computer is the human speech itself. But the concept is not just science fiction, it is what is now being worked on at the Spoken Language Systems (SLS) group at MIT’s Laboratory of Computer Science. SLS has developed a system where humans can speak with a computer by telephone, using conversational English, to obtain information about weather forecasts,

---

[6] See also Tr. (Felten) 757-62.

airline schedules and Boston area restaurants. See <http://www.sls.lcs.mit.edu/sls/whatwedo/> (visited January 5, 2001).

What SLS has developed is not as advanced as the computer interface on the starship Enterprise, of course, but it is a working example of human speech as the "source code" for a computer. It is easy to extrapolate, in the not so distant future, to computers being able to process far more complex and detailed instructions issued directly in English. The line between human speech and historic understandings of "source code" is becoming at least as blurred as the line between source code and object code. The lower court was absolutely correct when it stated that "[t]he path from idea to human language to source code to object code is a continuum." Drawing legal lines in the sand is, at best, a risky venture. *Cf. Denver Area Educational Television Consortium v. FCC*, 518 U.S. 727, 768 (1996) (Stevens, J., concurring) ("[I]t would be unwise to take a categorical approach to the resolution of novel First Amendment questions arising in an industry as dynamic as this").

II. **Computer Code is Expressive Speech.**

At root, computer code is nothing more than text, which, like any other text, is a form of speech. The Court may not know the meaning of the Visual BASIC or Perl texts set forth on page 9, *supra*, but the Court can recognize that the code is text.

There are rules for writing code, rules of grammar, naming conventions and logical organization, just as there are rules (which are not always followed) for writing in English. Indeed, a seminal text, Kernighan & Plauger, The Elements of Programming Style (McGraw-Hill, 1974, 1978) deliberately takes its name from the Strunk & White classic, The Elements of Style. Dr. Plauger describes the book as "[a] guide to writing more readable computer programs...."[7] Some of the rules are specific to the writing of code, but many have general application to well-constructed writing in any language, for example:

> Write clearly - don't be too clever.
> Say what you mean, simply and directly.
> Write clearly - don't sacrifice clarity for "efficiency."
> Parenthesize to avoid ambiguity.
> Don't stop with your first draft.
> Use the good features of a language; avoid the bad ones.

That code is a mode of textual expression (often better for its purpose than a written natural language) perhaps is best illustrated by Defense Trial Exhibit CCP, in which Dr. Touretzky describes in English the DeCSS code, interspersing in his English description the C code itself. We commend the Court to look at the whole exhibit, but just a few lines amply illustrate that code is just another form of language, text which more precisely conveys the speaker's message than does English:

---

[7] <http://www.plauger.com/books.html> (visited December 27, 2000).

The body of procedure CSStitlekey1 is as follows:

1.Take byte 0 of im, OR it with the hexadecimal constant 0x100, and store the result in t1.

```
t1=im[0]|0x100;
```

2.Take byte 1 of im and store it in t2.

```
t2=im[1];
```

3.Take bytes 2-5 of im and store them in t3.

```
t3=*((unsigned int *)(im+2));
```

Those who understand the C language could write the code directly from Dr. Touretzky's English description. Code is an expressive language, just as are mathematical formulae, music scores and dance choreography. See *Hurley v. Irish-American Gay, Lesbian and Bisexual Group*, 515 U.S. 557, 569 (1995).

III. **Academics and Programmers Must Have the Freedom to Communicate Fully in Code.**

The core purpose of copyright is "[t]o promote the Progress of Science and useful Arts ...." U.S. CONST., art. I, § 8 (Copyright Clause). Not only in the field of computer science, but also in other fields which have become increasingly dependent on computers, progress often is best promoted when humans can speak to each other in code, since that language more effectively conveys that which they seek to communicate. As long ago as the 1970s, economist Wassily Leontief, who won the Nobel Prize in 1973 for his input-output model of national economies, published the Fortran code implementation of his model. Computer models, including publication of the underlying code, are increasingly used by physicists, engineers, biologists, neuroscientists and computational linguists, to name just a few disciplines.

Scholarly print journals may have a general discussion in English of the code, with snippets being included on paper, but with a URL (Uniform Resource Locator) to a web site where the full code can be obtained. Alternatively, scientific papers appear exclusively on the Internet, and include both discussion and the full code.

A good example, one of thousands which could be given, is the 1992 Request for Comments on the MD5 Message-Digest algorithm by Dr. Ronald Rivest, which may be found on the Web at <http://www.ietf.org/rfc/rfc1321.txt?number=1321> (visited December 27, 2000). As described by Dr. Rivest:

> The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

Dr. Rivest uses a mixture of code and English to describe the algorithm and what it does. Then, at the end, he sets forth the full source code for the algorithm. The reason is simple: Dr. Rivest could have spoken forever about the algorithm, but obviously the best way to see if it does what he believed to be the case is to run the code, test it, probe for weaknesses, determine if strengths can be added and so forth. This simply cannot be done without access to the full code. Further, for those who can read code fluently, the code itself is a precise description of what is intended, more than any amount of English. Whether the code is for a cryptographic algorithm or a macroeconomics model, the ability to communicate in code helps to best promote the progress of science and the useful arts. Interfering with the ability of academics and professionals to speak freely in code will chill scientific discourse and force the risk-averse to communicate in a less-preferred form.

And make no mistake, there is science in the study of CSS and DeCSS, well beyond what the studios would have us believe. The undisputed testimony of Dr. Touretzky and Professors Ramadge, Felten and Appel made that abundantly clear.[8] Yet, because of the court's ruling, those best positioned to do scientific work in this area feel unable to communicate freely. See, *e.g.*, Tr. (Touretzky) at 1084-86.

---

[8] Tr. (Touretzky) at 1066-68; Tr. (Ramadge) at 896-99; Tr. (Felten) at 757; Tr. (Appel) at 1096-98.

IV. **<u>Source Code And Object Code Are Copyrightable And Thus Entitled To Full First Amendment Protection.</u>**

The law does not allow for the copyright of abstract ideas, procedures, processes, methods of operation, facts, or even "sweat of the brow" compilations of facts. *See* 17 U.S.C. § 102(b) and *Feist Publications, Inc. v. Rural Telephone Service Co., Inc.*, 499 U.S. 340 (1991). Rather, as set forth in 17 U.S.C. § 201(a):

> Copyright protection subsists, in accordance with this title, <u>in original works of authorship</u> fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. (Emphasis added.)

Stated otherwise, copyright is proper only for particular expressions of ideas, including ideas expressed in "literary works." 17 U.S.C. § 201(a)(1).

17 U.S.C. § 101 defines a computer program as "... a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result," a definition which necessarily includes object code as well as source, since object code is what is used directly by a computer to "bring about a certain result." Further, 17 U.S.C. § 117 expressly provides for certain non-infringing uses of computer programs, thus making it clear that computer programs, whether expressed in object code or source code, are proper subjects of copyright.

That fact is well-established in case law, which should be no surprise. *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3rd Cir. 1983) ("Thus a computer program, whether in object code or source code is a 'literary work' and is protected from unauthorized copying, whether from its object or source code version."); *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, 797 F.2d 1222, 1233 (3rd Cir. 1986) ("It is well, though recently, established that copyright protection extends to a program's source and object codes."); *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989) ("Source and object code, the literal components of a program, are consistently held protected by a copyright on the program."); *NLFC, Inc. v. Devcom Mid-America, Inc.*, 45 F.3d 231, 234-35 (7th Cir. 1995) ("Both the source and object codes to computer software are also individually subject to copyright protection.")

The Second Circuit reaches the same conclusion. Citing to *Whelan*, *Apple Computer* and other cases, in *Computer Associates International, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2nd Cir. 1992), the Court stated that "[i]t is now well settled that the literal elements of computer programs, *i.e.*, their source and object codes, are the subject of copyright protection."

The fact that computer code is copyrightable means that it is protected by the First Amendment.[9] Copyright law, of which DMCA is a part, confirms the point:

> In view of the First Amendment protections already embodied in the Copyright Act's distinction between copyrightable expression and uncopyrightable facts and ideas, and the latitude for scholarship and comment traditionally afforded by fair use, we see no warrant for expanding the doctrine of fair use ....

Harper & Row, Publishers, Inc. v. Nation Enterprises, 471 U.S. 539, 560 (1985).

Thus, a First Amendment analysis must occur if the work is a copyrightable expression. As shown, source and object code surely are that. Nor does it matter that code is a relatively novel form of expression, understood by comparatively few, and the worth of which may not be fully appreciated by this Court:

> As Justice Holmes explained, "[i]t would be a dangerous undertaking for persons trained only to the law to constitute themselves final judges of the worth of [a work], outside of the narrowest and most obvious limits. At the one extreme some works of genius would be sure to miss appreciation. Their very novelty would make them repulsive until the public had learned the new language in which their author spoke." *Bleistein v. Donaldson Lithographing Co.*, 188 U.S. 239, 251, 23 S.Ct. 298, 300, 47 L.Ed. 460 (1903)

Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 582-83 (1994).

9

No legislation can diminish First Amendment rights, but the DMCA expressly provides for such rights. 17 U.S.C. § 1201 (c)(4) states that "[n]othing in this section shall enlarge or diminish any rights of free speech or the press for activities using consumer electronics, telecommunications, or computing products."

We acknowledge that no court has held expressly that a copyrightable original work of authorship must, necessarily, be entitled to full First Amendment protection, including strict judicial scrutiny, but the conclusion is ineluctable.[10] Code is a literary work, and there is no basis for distinguishing between the levels of Constitutional review given to differing types of literary works.[11]

V. <u>The Protection Given To Code Cannot Be Limited On Account of Functionality.</u>

The court below gave lip service to the notion that code is protected speech, but then promptly minimized the protection it was willing to give because of the functional aspects of code:

---

10 *Bernstein v. Dep't of State*, 922 F.Supp. 1426 (N.D.CA 1996) ("*Bernstein I*") was decided on pure First Amendment grounds, but after discussing the nuances of copyright and the First Amendment at some length, the court stated "[c]opyright protection, designed to protect original expression, 17 U.S.C. § 102(a), supports the likeness of a computer program to speech as defined by First Amendment law." *Id.* at 1436.

11 That code is literary also is illustrated by the many devotees of Perl poetry, *see, e.g.*, <http://www.itknowledge.com/tpj/contest-poetry.html> (visited January 5, 2001); or, on the other side of the coin, by the International Obfuscated C Code Contest <http://www.ioccc.org/> (visited January 5, 2001), which seeks to illustrate through the irony of functional but poorly written code (deliberately so, for the purpose of the contest) the importance of good writing style. The IOCCC was, to some extent, inspired by the Bulwer-Lytton Fiction Contest for wretched English prose. <http://www.bulwer-lytton.com/> (visited January 10, 2001).

> Thus, [DeCSS] has a distinctly functional, non-speech aspect in addition to reflecting the thoughts of the programmers. It enables anyone who receives it and who has a modicum of computer skills to circumvent plaintiffs' access control system. 111 F.Supp.2d at 329.
>
> [DMCA] is a content neutral regulation in furtherance of important governmental interests that does not unduly restrict expressive activities. In any case, [DeCSS'] particular functional characteristics are such that the Court would apply the same level of scrutiny even if it were viewed as content based. *Id*. at 332-33.

The court's concerns about the functionality of code were deeply misplaced. It is precisely that functionality which makes computers compute; yet, as we have seen, computer code is fully protectable under both copyright law and the First Amendment.

Indeed, most computer code does <u>not</u> function in a manner legally different from how a cake recipe or a music score "function", but there can be no doubt that the latter are entitled to full First Amendment protection. A cake recipe or a music score will give one all the information one needs to prepare the dessert or play the concerto, but in the absence of human intervention (baking, playing) they are nothing but pieces of text, which do nothing on their own except to inform the reader. So it is with most computer code: a human must give the command to interpret or compile the source code, and even if the code has been compiled as a binary executable file, a human must give the command to execute it. Further, a human must operate the device which makes the code functional – here, a computer outfitted with a DVD drive into which a disk bearing an encrypted movie has been inserted.[12] The law is clear that the protection afforded to speech is not dependent on whether a device is needed to "execute" the speech. *Freedman v. Maryland*, 380 U.S. 51 (1965).

In *Junger v. Daley*, 209 F.3d 481 (6th Cir. 2000), the Court expressly rejected the premise that the functionality of computer code defeats First Amendment protection:

> The district court concluded that the functional characteristics of source code overshadow its simultaneously expressive nature. The fact that a medium of expression has a functional capacity should not preclude constitutional protection. Rather, the appropriate consideration of the medium's functional capacity is in the analysis of permitted government regulation. *Id.* at 484.
>
> Because computer source code is an expressive means for the exchange of information and ideas about computer programming, we hold that it is protected by the First Amendment. *Id.* at 485
>
> The functional capabilities of source code, and particularly those of encryption source code, should be considered when analyzing the governmental interest in regulating the exchange of this form of speech. *Id.*

It has been suggested that, because *Junger* cites to *O'Brien* and *Turner Broadcasting System v. FCC*, 512 U.S. 622 (1994), the Sixth Circuit was deciding that

[12] Since code, without human intervention, is no more functional than a dessert recipe, we are troubled that the court did not articulate what it meant by "functional".

intermediate scrutiny, that purportedly employed below, is the appropriate level of judicial review of code.[13] That is not the case. Rather, a careful reading of *Junger* discloses that the Sixth Circuit did not decide on the appropriate level of scrutiny. Instead, it decided that even if intermediate scrutiny is more appropriate than strict scrutiny, Professor Junger's challenge must be allowed to proceed:

> Before any level of judicial scrutiny can be applied to the Regulations, Junger must be in a position to bring a facial challenge to these regulations. In light of the recent amendments to the Export Administration Regulations, the district court should examine the new regulations to determine if Junger can bring a facial challenge. *Id.*

In *Bernstein I*, 922 F.Supp. at 1435, the Court stated:

> Nor does the particular language one chooses change the nature of language for First Amendment purposes. This court can find no meaningful difference between computer language, particularly high-level languages as defined above, and German or French. All participate in a complex system of understood meanings within specific communities. Even object code, which directly instructs the computer, operates as a "language." When the source code is converted into the object code "language," the object program still contains the text of the source program. The expression of ideas, commands, objectives and other contents of the source program are merely translated into machine-readable code. (Footnote omitted.)
>
> [....] Thus, even if Snuffle source code, which is easily compiled into object code for the computer to read and easily used for encryption, is

---

13

Appellants argue that even if intermediate scrutiny is appropriate, the court did not correctly utilize that level of review. We do not address that issue here.

> essentially functional, that does not remove it from the realm of speech. Instructions, do-it-yourself manuals, recipes, even technical information about hydrogen bomb construction, see *United States v. The Progressive, Inc.*, 467 F.Supp. 990 (W.D.Wisc.1979), are often purely functional; they are also speech.

The rejection of functionality as a reason to deny First Amendment protection to code was put even more squarely by a panel of the Ninth Circuit. In *Bernstein v. Dep't of Justice*, 176 F.3d 1132, 1141-42, *reh'g en banc granted and opinion withdrawn*, 192 F.3d 1308 (9th Cir. 1999) ("*Bernstein IV*"), the Court stated:[14]

> The government, in fact, does not seriously dispute that source code is used by cryptographers for expressive purposes. Rather, the government maintains that source code is different from other forms of expression (such as blueprints, recipes, and "how-to" manuals) because it can be used to control directly the operation of a computer without conveying information to the user. In the government's view, by targeting this unique functional aspect of source code, rather than the content of the ideas that may be expressed therein, the export regulations manage to skirt entirely the concerns of the First Amendment. This argument is flawed for at least two reasons.
>
> [...]
>
> Second, and more importantly, the government's argument, distilled to its essence, suggests that even one drop of "direct functionality" overwhelms any constitutional protections that expression might otherwise enjoy. This cannot be so. The distinction urged on us by the

---

14 The Opinion was withdrawn pending *en banc* review. After the Administration changed significantly the regulations applicable to the code in that case, the review was mooted. We do not normally cite to withdrawn opinions, but note that the court below thought *Bernstein IV* worthy of mention. 111 F.Supp.2d at 327 n.186.

government would prove too much in this era of rapidly evolving computer capabilities. The fact that computers will soon be able to respond directly to spoken commands, for example, should not confer on the government the unfettered power to impose prior restraints on speech in an effort to control its "functional" aspects. The First Amendment is concerned with expression, and we reject the notion that the admixture of functionality necessarily puts expression beyond the protections of the Constitution. (Footnote omitted.)

There can be little doubt but that the functionality of computer code can not limit the constitutional protections to which it is entitled. The First Amendment, of course, does not guarantee that all speech can be made without legal consequence, we are mindful of the law concerning matters such as obscenity, defamation and fighting words; but there is no basis to restrict computer code without strict judicial scrutiny.

The lower court relied on the content v. conduct distinction in *United States v. O'Brien* 391 U.S. 367 (1968), but that distinction has no bearing here. Code is simply text, which has no conduct inherently associated with it.[15] In the absence of human intervention, code does not function, it engages in no conduct. It is as passive as a cake recipe.

15

At 176 F.3d 1143 n.18, *Bernstein IV* continued:

Of course, source code may be functional as well as expressive. We are not persuaded, however, that that fact transmogrifies the distribution of scientific texts from "expression" into "conduct" deserving of diminished First Amendment protection.

The lower court obliquely acknowledged this, see page 22, *supra*, but then proceeded to enjoin the expression of the content based on the (unproven) fear that others, not before the court, might conduct themselves in a manner which the court perceived as unlawful. *O'Brien* intermediate scrutiny applies only when the content and conduct are merged in the same actor and the same act. Here, DeCSS itself is merely content, not conduct, which content was expressed by Appellants. The only evidence of conduct before the court was evidence of the use of DeCSS by expert witnesses in preparation for trial. There was no merger of conduct and content, *O'Brien* does not apply.

Nor can the lower court's use of intermediate scrutiny be justified by *Turner Broadcasting*, *supra*. "[T]he intermediate level of scrutiny [is] applicable to content-neutral restrictions that impose an incidental burden on speech." 512 U.S. at 662. "Our precedents thus apply the most exacting scrutiny to regulations that suppress, disadvantage, or impose differential burdens upon speech because of its content.... In contrast, regulations that are unrelated to the content of speech are subject to an intermediate level of scrutiny...." *Id.* at 642. (Citations omitted.) We have shown that code is pure textual speech, regardless of its supposed "functionality". The DeCSS code itself is pure textual content, and it simply cannot be regulated without the most exacting judicial scrutiny.

Indeed, a bone-chilling footnote suggests that the court's concern may have been something rather different than the functionality of code. At 111 F.Supp.2d 345 n.275, the court stated:

> During the trial, Professor Touretzky of Carnegie Mellon University, as noted above, convincingly demonstrated that computer source and object code convey the same ideas as various other modes of expression, including spoken language descriptions of the algorithm embodied in the code. Tr. (Touretzky) at 1068-69; Ex. BBE, CCO, CCP, CCQ. He drew from this the conclusion that the preliminary injunction irrationally distinguished between the code, which was enjoined, and other modes of expression that convey the same idea, which were not, *id.*, although of course he had no reason to be aware <u>that the injunction drew that line only because that was the limit of the relief plaintiffs sought.</u> With commendable candor, he readily admitted that the implication of his view that the spoken language and computer code versions were substantially similar was not necessarily that the preliminary injunction was too broad; rather, the logic of his position was that it was either too broad or too narrow. *Id.* at 1070-71. <u>Once again, the question of a substantially broader injunction need not be addressed here, as plaintiffs have not sought broader relief.</u> (Emphasis added.)

We cannot know what the court would have done had it been asked to enjoin a purely English narrative of the DeCSS source, but the note strongly suggests the court's willingness to do so. It is unfathomable that English prose could be a circumvention measure under the terms of DMCA, yet the court intimated that it might have so found. And if DMCA could be read to proscribe English as a circumvention measure, the Constitutional ramifications come into even sharper focus. *Cf. Bernstein I*, 922 F.Supp at 1435 (page 25, *supra*) and *American Booksellers Ass'n v. Hudnut*, 771 F.2d 323, 333 (7th Cir. 1985), *aff'd mem* 475 U.S. 1001 (1986). ("Much speech is dangerous. Chemists whose work might help someone build a bomb, political theorists whose papers might start political movements that lead to riots, speakers whose ideas attract violent protesters, all these and more leave loss in their wake. Unless the remedy is very closely confined, it could be more dangerous to speech than all the libel judgments in history.")

It is inconceivable that English text could be proscribed with the application of anything less than the strictest judicial scrutiny. But the *raison d'etre* of this brief is to show that, legally and functionally, programming languages are indistinguishable from natural language text.

We leave this Court with these disturbing ideas, because we suggest that the lower court's note evinces an attitude that Appellees' economic interests must be protected at all costs. The "functionality" of code was, we think, a means to an end, not a legal basis for decision. If code is to be proscribed at all, it must be only after the attempt has withstood strict judicial scrutiny.

CONCLUSION

For the reasons stated, the District Court's Judgment should be reversed.

Respectfully submitted,

____

JAMES S. TYRE
California State Bar Number 83117
LAW OFFICES OF JAMES S. TYRE
10736 Jefferson Blvd., #512
Culver City, California 90230-4969
310-839-4114

Counsel for *Amici Curiae*

Date: January 23, 2001

CERTIFICATE OF COMPLIANCE

Pursuant to Fed. R. App. Pro. 29(d) and 32(a)(7)(B), the undersigned certifies that this brief, exclusive of the exempted portions, contains 6,998 words. The brief has been prepared in proportionally spaced typeface using: WP 9; Times New Roman, 14 point.

Dated: January 23, 2001

____

JAMES S. TYRE

CERTIFICATE OF SERVICE

I, James S. Tyre, hereby certify that on this the 23rd day of January, 2001, two correct copies of the Brief of *Amici Curiae* were served via Federal Express, overnight delivery, upon each of the following parties:

Martin Garbus, Esq.
FRANKFURT, GARBUS, KLEIN & SELZ, P.C.
488 Madison Avenue
New York, New York 10022
Attorneys for Defendants-Appellants

Cindy A. Cohn, Esq.
ELECTRONIC FRONTIER FOUNDATION
454 Shotwell Street
San Francisco, CA 94110
Attorneys for Defendants-Appellants

Leon P. Gold, Esq.
PROSKAUER ROSE LLP
1585 Broadway
New York, New York 10036-8299
Attorneys for Plaintiffs-Appellees

On said date and by said manner, the original plus nine (9) copies of said Brief will be delivered to the Clerk of the Court for filing.

____
JAMES S. TYRE