

Enforceability of the GPL

Michael Labowicz
May 5, 2004

University of Illinois at Urbana-Champaign
Department of General Engineering

Introduction

Millions of people around the world utilize the web search engine known as Google. Its name has become synonymous with the term “Internet search.”¹ Students and professionals swear by its accuracy, but few are actually aware of the technology present behind the scenes. Google would not be possible in its current form without a software license called the GNU General Public License (GPL). This is because Google uses thousands of computers running the GNU/Linux² operating system³ to process the huge numbers of search requests it receives every day.⁴

GNU/Linux’s open source nature allows thousands of programmers to contribute to its development. Their work is protected by the GPL. This license helps ensure that the programmer’s work is protected from being exploited and thus encourages programmers to share their source code⁵ and collaborate. However, the GPL has never been tested in a court of law, and therefore its enforceability is still questionable. This paper will explore the enforceability of the GPL.

Sharing the Source

The right to share and modify software source code is at the heart of the GPL. Software source code has sometimes been compared to recipes that are used in cooking.⁶ A recipe is a list of instructions that a cook has to follow in order to prepare a finished food dish. In a similar manner, software source code is a list of instructions that a program, known as a compiler,⁷ must follow in order to produce a working program. This working program consists of a series of binary digits⁸, which is easily understood by a computer.

In the 1970’s, when computer programming was a relatively young profession, software source code was traded in much the same way as recipes are traded. The recipe for a great dish is often traded between cooks, who want to try out and improve the recipe. At that time, software programmers behaved in much the same way. If an elegant way were found to solve a problem in a program, the solution would be available to anyone who would need it. This led to a free exchange of ideas, which in turn led to higher quality programs.⁹

¹ Google Corporate Information (accessed May 03, 2004)

<<http://www.google.com/corporate/index.html>>

² GNU/Linux is an open-source operating system similar to the UNIX operating system.

(The American Heritage® Dictionary)

³ Operating System - Software designed to control the hardware of a specific data-processing system in order to allow users and application programs to make use of it. (The American Heritage® Dictionary)

⁴ Glyn Moody, Rebel Code, 284 (2001)

⁵ Source Code - Code written by a programmer in a high-level language and readable by people but not computers. (The American Heritage® Dictionary)

⁶ Richard Stallman, Open Sources: Voices from the Open Source Revolution (January 1999)

⁷ Compiler - A program that translates another program written in a high-level language into machine language so that it can be executed. (The American Heritage® Dictionary)

⁸ Binary Digit - Either of the digits 0 or 1, used in the binary number system. (The American Heritage® Dictionary)

⁹ Glyn Moody, Rebel Code, 16 (2001)

As the software industry matured, companies, which specialized in selling software, appeared. These companies strove to ensure that their programmers were not allowed to share their source code. Their code was considered to be a company secret and therefore was closely guarded. While this provided short-term benefits to the companies themselves, it robbed programmers of the ability to freely exchange ideas without fear of losing their jobs. Freelance programmers were also affected. If the programmers shared code, it would be free to be incorporated into another company's product without compensation to the programmer or to the community in general. Under these circumstances, programmers had little incentive to share their work. The cooperation, which had fostered giant leaps in the art of programming, was in danger of being eliminated.¹⁰

The tools with which the programmers created their source code also became more limited. The operating systems and editors that were available came with restrictive licenses and non-disclosure clauses that prevented programmers from using the systems as they wished and collaborating.¹¹ The price of these systems also rose quite significantly, which further prevented programmers from being able work effectively.

The GNU Project

A programmer at the Massachusetts Institute of Technology's Artificial Intelligence Lab named Richard Stallman decided started a project that would provide programmers with the environment and tools needed to be able to continue to share code and collaborate on their projects. Stallman decided to name the project by following a hacker¹² tradition of recursively including a project's name as part of that project's name. He named the project GNU, which stands for "GNU's Not Unix."¹³

The project's main goal was to create a "free" operating system based on the popular UNIX operating system that was widely used at the time. Stallman's definition of "free" was different than the popular meaning of no monetary cost. According to Stallman "free" software meant that its users would have the freedom to run the program in any way that they desired, have access to its source code, modify it to suit their needs, improve its functionality, and redistribute copies of it so that others could benefit.¹⁴

Under Stallman's definition of free software, users had the right to use software in any way that they saw fit. This was a drastically different way of looking at software because, at the time, the majority of software was considered to be the sole property of its copyright holder and its users had to have a license in order to use it or express permission to modify and redistribute it.¹⁵

However, often times a company would not be willing to license and share the source code under any circumstances. An example of this can be found in Stallman's experiences with a printer

¹⁰ Id.

¹¹ Richard Stallman, *Open Sources: Voices from the Open Source Revolution* (January 1999)

¹² Hacker - One who is proficient at using or programming a computer; a computer buff. (The American Heritage® Dictionary)

¹³ Richard Stallman, *Open Sources: Voices from the Open Source Revolution* (January 1999)

¹⁴ Id.

¹⁵ Id.

manufacturer. The printer's software did not have the features that Stallman needed, so he contacted the company to get the source code so that he could add them. The company refused to share the source code, and did not have the resources to implement the requested features. As a result, Stallman could not use the printers in the way he needed to, and found himself out of luck. This situation would have been easily avoided if the source code had been made available.¹⁶

The Free Software Foundation

In order to help spread his message of free software and protect his work on GNU, Stallman created the GPL. It is a unique software license that protects the rights of programmers who decide to share their source code with the community. In essence, it covers the redistribution of software by insisting that software covered by it be distributed with the source code made available. Everybody has the right to look at the software's source code and modify and redistribute it as long as they share any changes that they make to the code. Stallman went on to create the Free Software Foundation (FSF) to help enforce the GPL and to promote free software.¹⁷

Today, software licensed under the GPL has broken into the mainstream. The most widely recognized GPL licensed software is the Linux kernel. It forms the basis of the GNU/Linux operating system, which is used to run devices from set-top DVD players to cell phones to web servers. GNU/Linux levels the playing field between software developers, and has backing from companies such as Novell, IBM, and Sun Microsystems.¹⁸

In addition to Google, a great number of programs and web services would not be possible without the existence of the GPL and the idea of free software. An important example is a program called sendmail. It was the first and still is, the most popular program used by Internet Service Providers to send email across the Internet. It helped create the Internet's first truly universal "killer app," or indispensable application.¹⁹ While it does not use the GPL license, it uses a similar license, which is heavily influenced by the GPL. The freedom of its license allows it to be continuously developed by a dedicated community of programmers. Other important free software programs include BIND²⁰, which is used to manage Internet domain names, and Apache, which is the most popular program used to serve webpages on the Internet.²¹

Enforcing the GPL

This freedom comes at a price. The FSF has to constantly be on the lookout for potential violations of the terms of the license. Despite that fact that most parties comply with all the terms of the license, the FSF deals with dozens of GPL violation cases a year²². Offenses are usually reported to the FSF by programmers working on the violated open source project, or through the FSF website at <<http://www.gnu.org>>. Provided that the FSF owns the copyright to

¹⁶ Id.

¹⁷ Id.

¹⁸ Glyn Moody, *Rebel Code*, 237 (2001)

¹⁹ Id.

²⁰ BIND is an acronym for Berkeley Internet Name Domain.

²¹ Glyn Moody, *Rebel Code*, 127-130 (2001)

²² Eben Moglen, *Free Software Matters: Enforcing the GPL*, *LinuxUser Magazine* (September 10, 2001)

the source code in question, it is legally entitled to seek compliance with the GPL. Otherwise it is up to the original owner of the copyright to seek compliance, usually with help from the FSF.

The FSF is entitled to the copyright of the source code to programs that are part of the GNU project. Additionally, freelance programmers are also encouraged to grant the copyright of their GPL licensed programs to the FSF, so that it can use its resources to directly enforce the GPL in cases involving the program. This is usually the best way for these programmers to enforce their rights under the GPL because of the heavy costs associated with litigation. Corporations are usually reluctant to transfer the copyright to the FSF, and in those cases, the FSF works with the company to help enforce the GPL.

Once the FSF receives notice of a GPL violation and it owns the program's copyright, it first approaches the offending party and notifies them of their violation of the GPL license. If the violation is unintentional, the offending party is provided with steps that it can follow to bring it into compliance. Usually the party then complies, and this is where the process ends. However, as with any aspect of life, there are always people looking to take advantage of the generosity of others. If the violation has been ongoing for a long period of time and it is intentional or particularly blatant, the FSF has been known to work with the offending party to setup a GPL enforcement committee that supervises that party's compliance with the GPL license.²³

If the party is unwilling to cooperate, the FSF would be forced to make the story public or pursue litigation. Most companies go to great lengths to avoid bad publicity; therefore it is an effective weapon for GPL enforcement. However, there may come a time when bad publicity isn't enough of a deterrent to ensure GPL compliance. If that occurs, litigation would be the only option. Surprisingly, it has never come to that. As of March 2004, the GPL has not been openly tested in the court of law. This is because most parties would rather comply with the GPL than be subject to court fines and/or bad publicity. However, the untested nature of the GPL has led some to question the validity of the license because of the lack of legal precedent surrounding its enforcement.²⁴

The General Public License

Overview

In order to be able to examine the enforceability of the GPL, it is necessary to look at the provisions and requirements of the license itself. The GPL is a software license which is included with software that is redistributed. Its preamble states that "if you distribute copies of ... a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code"²⁵ This ensures that any modifications made to the original are published publicly in the form of source code.

Like most software licenses, the GPL consists of a set of Terms and Conditions that cover copying, distribution, and modification. However it differs from most licenses in the fact that its terms additionally protect the rights of the users of the software program instead of only the party distributing the software. The GPL contains thirteen terms and conditions which cover copying,

²³ Id.

²⁴ Id.

²⁵ GNU General Public License. Version 2, June 1991

distribution, and modification of the program which the license covers. The following paragraphs will cover the terms which are important to the enforceability of the GPL.

Section 0 specifies that the license applies to “any program or work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License.”²⁶ It goes on to further specify that “activities other than copying, distribution and modification are not covered by this License.”²⁷ This limits the scope of the license to those activities, meaning that parties participating in them are subject to the terms of the license.

Section 1 covers the rights of the user to copy and distribute copies of the program’s source code, as long as the appropriate copyright notice is preserved, and the GPL license is included. It also specifies that a fee may be charged for the service of transferring a copy of the source code or for providing a warranty.²⁸ This is important because it preserves the right of people to be compensated for their efforts, which has enabled GPL licensed software to be used in the corporate world (most notably by companies such as Hewlett Packard, Red Hat and IBM).

Section 2 specifies that any changes that are made to the source code of the program covered must be given back to the community and proper copyright notice must be preserved throughout the source code and program. It is important to note that this restriction applies to the work as a whole. If even one part of the program’s source code is modified, the source code of the entire work must be redistributed.²⁹

Section 5 is particularly important with regards to the enforceability of the GPL. This is because it explains why a user is bound to the terms of the GPL. It does so by stating that a user is not required to accept the GPL because they have not signed it, but the act of redistributing a modified version of the source code indicates acceptance of the license and all of its terms.³⁰ The license justifies this reasoning by stating that the rights of redistribution and modification of the program and its source code are not specified anywhere else and are not inherently given to a software user.³¹ This aspect of the GPL sets it apart from other, more common, software licenses. In most cases, a user is prompted to indicate their acceptance by clicking “yes” to the license agreement, which appears in a dialog box when the program is installed. The open source nature of programs covered by the GPL usually means that users often use and modify the source code without executing the program. This means that acceptance in the standard manner is impossible.

These conditions contribute to a document which is designed to protect the freedoms of the software user and the copyright holder. In addition, its requirement that the source code be made available ensures that everybody can benefit from the work done on the program. Like cooks trading recipes, or scientists sharing ideas, the GPL brings collaborative advancement to the software world.³²

²⁶ Id.

²⁷ Id.

²⁸ Id.

²⁹ Id.

³⁰ Id.

³¹ Id.

³² Glyn Moody, *Rebel Code*, 26-28 (2001)

Contract vs. Copyright

Contract

According to Linuxplanet.com columnist Dennis E. Powell, “software licenses are generally considered to be contracts.”³³ This means that lawyers arguing a case of GPL violation would be compelled to argue whether or not it fulfills all the elements of a contract. According to law, a contract must be made up of the following five elements in order to be valid: Agreement consisting of Offer and Acceptance, Competent Parties, Consideration from Both Sides, Lawful Purpose, and Proper Form.³⁴

A contract is valid as long as there is an agreement in terms of an offer and acceptance between the two parties to the contract, the offeror and offeree.³⁵ In the case of a software license, such as the GPL, the offer is the software license itself. The offeror is the original copyright holder (usually the original programmer), while the offeree is the user of the software covered by the license. As stated in the previous section, acceptance is carried out under the GPL when a user of a program distributes or modifies it.

Competence is also required in order for a contract to be valid.³⁶ The GPL assumes that both parties are competent from a legal standpoint. A party that has enough awareness to modify or distribute a piece of software should be considered competent enough to realize that he is agreeing to the software license by doing so. An interesting exception could occur if a minor³⁷ would be involved in the process. Since the minor would not be in the age of majority, he should not be subject to the terms of the agreement because his word is non-binding.³⁸ However, this should not be an issue; most computer programmers need several years of training to be able to work with source code at a point where they could infringe on the terms of the GPL.

There must also be proper consideration from both parties in order for the contract to be valid.³⁹ In the case of the GPL, the users of the software gain consideration in the form of rights that are granted to them by the copyright holder of the software, while the copyright holder gains a promise from its users to follow the terms and conditions set by the GPL.

Lawful purpose must also be fulfilled for a valid contract to exist.⁴⁰ In the case of the GPL, it is fulfilled as long as there are no unlawful acts that occur as a result of the actions of the copyright holder or the software user. It can be argued that if the software covered by the GPL is used for unlawful purposes, the contract is void, and therefore, the parties would not be subject to the terms and conditions set by the GPL. However, if this is the case, the authors of the GPL licensed software would have much more pressing criminal issues to deal with instead of enforcing their license.

³³ Dennis E. Powell, .comment: Judgment Day for the GPL? June 26,2000 (accessed May 03, 2004)
<<http://www.linuxplanet.com/linuxplanet/reports/2000/1/>>

³⁴ Richard C. Vaughn, Legal Aspects of Engineering, 48 (1999)

³⁵ Id.

³⁶ Id.

³⁷ Minor - Being under legal age; not yet a legal adult. (The American Heritage® Dictionary)

³⁸ Richard C. Vaughn, Legal Aspects of Engineering, 48 (1999)

³⁹ Id.

⁴⁰ Id.

Finally, form must also be followed in order for there to be a valid contract between the two parties.⁴¹ In the case of the GPL, the contract is expressed in written form from the standpoint of the copyright holder and in terms of the actions of the software users. In this case, the software users accept the contract by either distributing or modifying the program, and therefore, they accept by performance.⁴² However, it could be argued that this method of acceptance is invalid because a user might not have read the terms of the license before distributing the program. The GPL deals with this in Section 5. As mentioned before, the user does not have the inherent rights to redistribute and modify a program. The GPL grants these rights under the condition that the user redistribute the program along with the source code for any changes made.

Copyright

In addition to the GPL being a contract, the source code that it covers is also protected under copyright law.⁴³ The copyright-ability of source code was decided in a case between Apple Computer, Inc. and the Franklin Computer Corporation.⁴⁴ In this case, it was found that software source code is covered under copyright law; similar to the way that written work and music are covered. This means that the copyright owner of the work is free to grant and restrict other people's rights to use and modify their work. The GPL takes this into account and specifies that any modification to the source code results in a derivative work that must be shared in any redistribution.⁴⁵ By including the GPL along with the software, the copyright holder specifies that this is the way in which they want their copyright utilized, if at all.

The GPL is sometimes referred as being a “copyleft” license. Richard Stallman describes how the name came about in the following passage from Open Sources: “In 1984 or 1985, Don Hopkins ... mailed me a letter. On the envelope he had written ... ‘Copyleft--all rights reversed.’ I used the word “copyleft” to name the distribution concept I was developing at the time.”⁴⁶ “Copyleft” is an interesting concept. In copyright law, a party is usually prevented from distributing works made by the copyright owner. Under the concept of copyleft, the party is required to distribute the works made by the copyright owner. This concept forms the basis of the GPL. Since it is an odd way of looking at copyright, difficulties might arise if the GPL is ever tested in court. A jury or judge may not agree with the GPL's definition of copyright, and rule against it. However, if explained correctly, the GPL is relatively straight forward and therefore easy to understand.⁴⁷

As mentioned before, only the copyright owner can enforce their rights under the GPL.⁴⁸ The author of a program covered by the GPL is the one who owns the copyright, unless they have given the copyright to the FSF, which is quite common in the free software community. If the FSF does not have the copyright, GPL enforcement falls directly on the shoulders of copyright owner. This can be a very expensive and time-consuming task, and therefore the FSF advises open source programmers to avoid drawn out legal proceedings whenever possible. The

⁴¹ Id.

⁴² Richard C. Vaughn, *Legal Aspects of Engineering*, 48-50 (1999)

⁴³ Dennis E. Powell, .comment: Judgment Day for the GPL? June 26,2000

⁴⁴ *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, United States Court of Appeals, Third Circuit (Aug. 30, 1983)

⁴⁵ GNU General Public License. Version 2, June 1991

⁴⁶ Richard Stallman, *Open Sources: Voices from the Open Source Revolution*, January 1999

⁴⁷ Eben Moglen, *Free Software Matters: Enforcing the GPL*, *LinuxUser Magazine* (September 10, 2001)

⁴⁸ Id.

preferred method of enforcement is through cooperation with the offending party, or bad publicity. The FSF also follows its own advice by saving its resources, so that it can concentrate on the cases that will most likely prove that the GPL is valid.⁴⁹

Legal Precedence

Importance of Legal Precedence

It is important for the FSF to participate in a case in which it has a very good chance of winning.⁵⁰ If the GPL is enforced in court, it will become a legally binding document. This means that lawyers in other cases involving the GPL would be able to cite the first case to bolster their case, under the common law concept of legal precedence.⁵¹ Additionally, the fear of legal action would keep other parties from infringing upon the rights of the copyright owners of other GPL licensed programs.⁵²

However, if a case is found where a judge rules against the GPL it will have the opposite effect. The GPL will cease to be enforceable, and will have to be either reworded or abandoned. If this occurs, the work of thousands of programmers will cease to be protected under the GPL. The next two cases are examples of pending cases that involve the GPL.

Linksys GPL Violations

An example of the FSF's work in enforcing the GPL can be found in an ongoing dispute with a company named Linksys. Linksys manufactures network routers⁵³ that are often used to split Internet connections, so that they can be used by more than one computer. The routers use GPL software including GNU/Linux to provide their routing functionality. Linksys has modified many of the GPL licensed programs in order for them to perform all the functions necessary for the router. However, the company did not make all of the required source code available, which is required under the GPL.

The GPL specifies that Linksys should share any and all of the modified source code that resulted from any changes made to the original program along with the program's executable files⁵⁴. After repeated requests, Linksys refused to release all of the source code in compliance with the terms of the GPL.⁵⁵ Linksys is probably trying to keep its changes to the source code secret so that other network router manufacturers would not be able to benefit from their modifications. This directly opposes the spirit of the GPL.

The FSF is currently in contact with the copyright holders of all of the GPL licensed programs involved, and is working with Linksys to resolve the issue. The story has already been

⁴⁹ Id.

⁵⁰ Id.

⁵¹ Richard C. Vaughn, *Legal Aspects of Engineering*, 22-26 (1999)

⁵² Eben Moglen, *Free Software Matters: Enforcing the GPL*, *LinuxUser Magazine* (September 10, 2001)

⁵³ Router -A device in a network that handles message transfer between computers. (*The American Heritage® Dictionary*)

⁵⁴ Executable Files - Files that contain the executable binary instructions used to run a computer program. (*The American Heritage® Dictionary*)

⁵⁵ Andrew Miklas. "Linksys WRT54G: Part 2" Newsgroup Posting. Sun Sep 28 2003 - 18:16:29 EST (accessed May 03, 2004) <<http://www.ussg.iu.edu/hypermil/linux/kernel/0309.3/0904.html>>

mentioned in the media⁵⁶ and therefore the threat of bad publicity has become invalid. If these violations continue, a legal dispute will be the next course of action for the copyright owners of the GPL licensed software.

SCO vs. IBM

Another case involving the GPL is currently being heard by the U.S. District Court in Utah. It is a civil case between two companies: the Santa Cruz Operation (SCO) and International Business Machines (IBM). The SCO Group, formerly known as Caldera Systems, Inc. is a software company that specializes in UNIX solutions for small to medium-sized businesses.⁵⁷ It also sold and distributed a version of Linux before entering into the lawsuit with IBM.

IBM is a multi-national \$104.4 billion company, which sells a variety of software solutions including UNIX and Linux. It contributes programmers and source code to various open source projects.⁵⁸ It has also licensed UNIX code from AT&T (when it owned the rights to UNIX) to create its version of UNIX called AIX.⁵⁹

SCO alleges that IBM breached the terms of the UNIX license regarding the non-disclosure of UNIX Code when it introduced UNIX code and methods into Linux. As a result of these transgressions, SCO is seeking three billion dollars in damages from IBM. As of May 2004, SCO has not publicly disclosed any of the code that it says is infringing. However it has release source code to certain members of the media, and most accounts say that the code is similar but not exactly the same. However, in *USL v. BSD Inc.*⁶⁰ the court ruled that similar code does not necessarily mean that the code is the same. Often time's programmers use similar methods to implement solutions to problems. Therefore in this case, SCO will have to prove that IBM programmers used their UNIX code, not just similar programming techniques.

IBM has disputed SCO's claims "citing lack of standing, statute of limitations, economic-loss and independent-duty doctrines, laches, delay, unclean hands, waiver, estoppel, federal law preemption, and improper venue."⁶¹ This multitude of reasons used to dispute SCO's claim ensures that IBM will have plenty of grounds with which to appeal the case if the verdict is not in its favor. IBM has additionally alleged that SCO violated the GPL by including Linux source code into its UNIX product without sharing the modified source code. Regardless of the outcome of the rest of the case, the ruling on SCO's negligence of the terms of the GPL is critical.⁶²

Eric Raymond, an open source software activist, has written a position paper about the SCO vs. IBM case. In it he asserts that since the case is being tried in federal court, the court may rule on

⁵⁶ Slashdot. "Linksys Still In Violation of the GPL?" September 29, 2003. (accessed May 03, 2004) <<http://slashdot.org/article.pl?sid=03/09/29/118235>>

⁵⁷ SCO Group Company Information Website <<http://www.thescogroup.com/company/>>

⁵⁸ About IBM Website (accessed May 03, 2004) <<http://www.ibm.com/ibm/us/>>

⁵⁹ AIX is an acronym for Advanced IBM Unix.

⁶⁰ *UNIX SYSTEM LABORATORIES, INC. v. BERKELEY SOFTWARE DESIGN, INC.*, UNITED STATES DISTRICT COURT OF NEW JERSEY. Civ. No. 92-1667

⁶¹ SCO vs. IBM Executive summary (accessed May 03, 2004)

<<http://twiki.iwethey.org/twiki/bin/view/Main/SCOvsIBMExecutiveSummary>>

⁶² Id.

general issues of intellectual property law. This means that the court can find that the “GNU General Public License under which Linux is issued is in fact a valid license and does create binding legal obligations on those who accept its terms.”⁶³ If a judge were to make a ruling like this, the GPL would be considered completely enforceable.

In addition to IBM, SCO has sued AutoZone, DaimlerChrysler and Novell in hopes that it could further support its case that Linux contains UNIX source code. In doing so, it is hoping to start collecting royalties from all users of the free operating system.⁶⁴

Conclusion

The question of the enforceability of the GPL is still not certain. As of May 2004, it has yet to be validated by a court ruling. However there are a number of cases ongoing that may provide the GPL with the validation that it needs to be enforceable. The most important is the IBM vs. SCO case. It is one of the few cases involving the GPL that has actually gone to trial. The case may last for many years, but IBM has the resources to see the case through to the very end, and it has too much invested in Linux and the GPL to settle out of court.

However, GPL enforcement is not entirely about legal validation. It is important to note that the FSF enforces the GPL all the time outside of court. Companies and individuals comply with the license to avoid the bad publicity associated with taking advantage of free software programmers. Complying with the GPL also has the added benefit of granting these parties the ability to reap the benefits that open source collaboration has to offer.

⁶³ Eric Raymond. “OSI Position Paper on the SCO v. IBM Complaint”
< <http://www.opensource.org/sco-vs-ibm.html> >

⁶⁴ SCO Source Website (accessed May 03, 2004)
<<http://www.sco.com/scosource/>>

References:

1. GNU General Public License, Version 2;
Free Software Foundation, Inc, June 1991
2. Glyn Moody, Rebel Code: Linux and the Open Source Revolution;
Perseus Publishing; 1st Edition (July 15, 2002)
3. Various, Open Sources: Voices from the Open Source Revolution
O'Reilly; 1st Edition (January 1999)
4. Eben Moglen, Free Software Matters: Enforcing the GPL, I and II;
Column in LinuxUser Magazine (August 2001)
5. Linux Planet -. comment: Judgment Day for the GPL, Dennis E. Powell
(Accessed May 03, 2004) <<http://www.linuxplanet.com/linuxplanet/reports/2000/1/>>
6. The GPL Compliance Lab: How to keep Free Software Free
Presentation by David Turner, Free Software Licensing Guru and GPL Compliance Engineer
(Accessed May 03, 2004) <<http://web.novalis.org/lsm-talk/slide-0.html>>
7. Richard C. Vaughn, Legal Aspects of Engineering;
Kendall/Hunt Publishing Company; Sixth Edition 1999
8. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240,
United States Court of Appeals, Third Circuit (Aug. 30, 1983)
9. UNIX SYSTEM LABORATORIES, INC. v. BERKELEY SOFTWARE DESIGN, INC.,
UNITED STATES DISTRICT COURT OF NEW JERSEY. Civ. No. 92-1667
10. Eric Raymond. OSI Position Paper on the SCO v. IBM Complaint
(Accessed May 03, 2004) < <http://www.opensource.org/sco-vs-ibm.html>>