

Use of Copyright by Open Source Software Movement on Computer Software and its Implications

Mahesh Madhavan

National Innovation Foundation, B 4, Ravi Niketan, Nehru Park, Vastrapur, Ahmedabad 380 015[†]

(Received 12 September 2002)

Open source software and the movement behind it, the Open Source Software Movement unknown to many has been in existence since long. Discussed in the article is this concept with relevance to the software industry. The US statutes on intellectual property law provide the backdrop of discussion, for this is where the movement, the organizations and the forefathers behind the fostering of the movement owe their origin. Through this article, some of the licensing schemes under open sourcing, which have turned out to be the buzzwords in the computer software industry are also brought into limelight. A look into how the movement assists in the development of the software industry and the long-term implications, the movement could pose for the copyright protection of computer software, are also considered.

The aim of copyright law is to maintain a balance between the diverging interests of the society and the author of a creative work. It tries to maintain this balance by providing to the author adequate compensation for his labour and to the society the liberty to use his work thereby ensuring the free flow of information.

Copyright protection has been applied to computer software in United States since 1976 when Congress amended the federal copyright statutes. The problems in applying the traditional norms of copyright protection to computer software have resulted in the development of a movement known as the Open Source Software Movement.

This article is an effort to go through this movement in analysing and understanding the manner in which the movement has applied copyright law to develop the software industry. The article is structured into three parts. The first part studies the movement by tracing its origin to bring out the significance of copyright to the movement. The second part describes how the movement through the use of copyright law has enabled the development and dissemination of software programs. This part also cites examples of two open source softwares that validate the defense of innovation. Lastly, the long-term implications of this approach on copyright protection of

[†]E-mail: mahesh374@yahoo.com, *Corresponding address: "Sumangali", Palliyil Lane, Kochi 682 016, Kerala*

computer software and on the software industry are also considered.

Copyright Protection of Software

Copyright as an instrument in the protection of a computer program affords protection to the original literary work inscribed in the program. Born out of the statutes, copyright protects the software program by preventing further reproductions of the same thereby preserving the rights of the author in the software. The extent of application of this investment to a software program was one of the disputed issues in the early court decisions on computer software protection in US and UK¹. By the extent of application, what is meant, are the areas of a software program that can be brought within the umbrella of copyright protection.

The object code and the source code being the core program areas, which affect the behaviour of the computer and where the copyright rule applies. It is necessary to understand these concepts to delve into the issues arising in relation to them.

The functioning of a computer's processor is based on the language of instructions provided to it. These instructions which exist in a readable format (higher level) are transformed using a compiler² into binary form³ for feeding into the processor⁴. The high level of instructions, which are non-executable and composed using programming languages is commonly referred to as the source code of a computer program and the executable format of instructions which control the

working of the computer is known as the object code.

From the point of view of a consumer who aims at the use of the software for personal purposes, the availability of the executable version of the software (CD) for running on the computer proves to satisfy his needs. But for a programmer who looks at the software as a means to develop his programming skills by working on the source code, the software does very little. The rationale behind the approach of the software developers in withholding the source code and launching in the market just the object code is based on the impossibility of recreating the source code from the object code⁵.

The software developers preserve their rights even when the source code is made available, with the aid of licensing restrictions. These licensing restrictions may restrict the licensee's right to copying, distribution and modification of the software program⁶.

Copyright Protection of Object Code and Source Code

Based on the recommendations of the National Commission on New Technological Uses of Copyrighted Works (CONTU), the US Copyright Act 1976 was amended to define 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"⁷. The source code rightly falling within the definition of a computer program, to gain protection under copyright, it would have to satisfy the originality requirement of copyright law.

The originality standard was laid down by the Supreme Court of United States in *Feist Publications vs Rural Telephone Service Co*⁸. Justice O'Connor while expressing the following words "original ...means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity", provided for a low level of originality requirement to be satisfied by a majority of works⁹. The source code satisfies this originality test easily.

The originality requirement thus being satisfied, the court's in its later decisions of *Whelan Assoc vs Jaslow Dental Lab Inc*¹⁰ and *Apple Computer Inc vs Franklin Computer Corp*¹¹ have upheld the copyright protection available to a source code¹². The copyright protection available to object codes also was considered by the United States Court of Appeals for the Third Circuit in the *Apple Computer Inc* case¹³, wherein the Court stated that "computer programs, whether object code or source code, are copyrightable as a literary work under the 1976 Act", thus establishing the view that object code and source code were protected by copyright¹⁴.

Open Source Software

The term open source has been defined to include "software which is available to the public in source code form, and which does not have licensing restrictions which limit use, modification or redistribution"¹⁵. Through open sourcing, the community of software developers make freely available to anyone the

source code of software, free for alteration, sharing and distribution.

Open Source Software Movement, a community whose aim is to encourage the free use of software by making available the source code of computer programs has been in existence for long¹⁶. The roots of the movement lie with the hobbyists and other programmers commonly known among the open source software community as the hackers. The community aims to encourage the exchange of information on programming and facilitate the cooperative effort on open source projects, through the distribution of the source code free for use and modification¹⁷. The movement however has emerged as a highly discussed area only in the recent years on account of the open source projects and the increasing use of the Internet¹⁸.

The distinction between open source software and proprietary software lies in the free use of the software and the licensing structure. While the proprietary software dealers release into the market the software program by concealing the source code, under open sourcing the source code is made available with the object code¹⁹. The software released through open sourcing is under a special class of licence known as the GPL (General Public Licence) encouraging and permitting users to use and improve the source code, the proprietary software restricts a users right to use only²⁰. By making available the source code, which exists in the common programming languages of C, C++, Fortran and JAVA free for alteration, use and redistribution, the Open Source Software Movement

argues to promote the idea of innovation and release of improved versions of software. Though the source code is freely made available under open sourcing and free use and modification of the software is permitted, licensing restrictions exist in open source software preserving its open source status and authorship right under copyright²¹.

The advocates of the free software community use the term open source software and free software interchangeably²² in different contexts. The distinction between them as stated by Richard Stallman exists in their principles, the free software adopting a philosophical goal by making software free²³, i.e....to promote the four freedoms and the open source software aiming at a practical goal by making source code available for free under licensing restrictions²⁴. Both the free software movement and the open source software movement, though being two subsets of the free software community, aim at encouraging software development.

Road to Open Source Software Movement

The road to the open source software movement may be traced back to the existence of the right of copyright. The basic reasoning behind the criticisms against software copyright arose from copyright being considered not a good protection measure for the emerging technology²⁵. It has been described to be a protection lacking in the basic ideals of a safeguarding right.

Advocating a *sui generis* approach for copyright law, the weakness of copyright

as a protection measure for software programs has been stated by a writer, "while copyright law can provide appropriate protection for some aspects of computer programs, other valuable aspects of program, such as the useful behaviour generated when the programs are in operation and the industrial design responsible for producing this behaviour, are vulnerable to rapid imitation that, left unchecked, would undermine incentives to invest in software development"²⁶.

The early software community has been described by the father of the Open Source Software Movement, Richard Stallman²⁷, as a community where people shared information with the liberty to use the source code and to work on it for creating newer versions²⁸. The changes in the community which came with the emergence of copyright in the 80's has been considered to be an approach where the sharing culture was prohibited²⁹.

The move for open sourcing of software programs may thus be considered to be two fold, the upper hand (copyright) separating the rights of a user and an author of the source code of the computer program, and the ineffectiveness of the traditional rules of copyright to serve its role as a protective shield.

The GNU Project

The emergence of the GNU project which was the result of attempts by Stallman in making a system compatible to the UNIX known for its features of being portable, flexible and with powerful functional characteristics has been considered by many to be the start of the

Open Source Software Movement³⁰. The GNU project, GNU stands for ‘GNU’s not Unix’ was thus aimed at the development of a software system compatible to the Unix to be distributed free to everyone who could use it³¹.

Stallman established the Free Software Foundation (FSF) aimed at the promotion of free software later in the year 1985 after the GNU in 1984. Though the GNU project was catching up in the race with the proprietary software, GNU lacked the kernel³², which was necessary to activate the hardware of the computer³³. Linux³⁴, developed by Linus Torvalds in 1991, solved this problem by providing the Linux core, the kernel necessary for running the GNU operating system³⁵. Stallman laid emphasis on the importance of the kernel by stating “If I am talking about the kernel, I call it Linux, and otherwise the complete system should be called GNU/Linux since both the GNU system and the kernel must be combined to create a complete whole”³⁶.

Use of Copyright by Copyleft

The central idea of Stallman in the Open Source Software movement was to enable free use of the software by all groups, which comprised its developers and its users. This was recognised by Stallman in the form of providing to every group, the four freedoms of free software³⁷. The four freedoms³⁸ as put forward by Stallman comprised the freedom to use the software program without restrictions, freedom to modify the program by rewriting the software, freedom to share the software which comprised making of copies of the same

and to further the goal of innovation in the software industry, grant the freedom of making alterations to the program and permit its distribution thereon³⁹.

The easiest mode of making the freedom available to all without restrictions was to place the software in the public domain. The most suitable public domain noted for the purpose was the Internet. This mode of granting the freedom, however, posed the dilemma of users modifying the programs and subsequently asserting proprietary rights (copyright) in them, thus obstructing the free flow of the software programs and creating a distinction between the rights of users and developers. A prototypical example noted by Stallman has been the X windows licence where the developers had all the four freedoms and the users with the sole licence of free use (freedom zero)⁴⁰. It was thus the existence of the proprietary right of copyright that paved the way for a method of keeping software free.

So to make available the four freedoms and to ensure that proprietary rights could not be asserted in derivative works, the GNU project came up with the idea of ‘Copyleft’. The term ‘Copyleft’ has been defined as “an agreement allowing the software to be used, modified and redistributed freely on the conditions that a notice to this effect is included with it”⁴¹. The significance of copyright in the copyleft licence has been stated in the jargon file of the hacker lexicon as “1. The copyright notice (General Public Licence) carried by GNU EMACS and other Free Software Foundation software, granting reuse, and reproduction rights to

all comers (but also see General Public Virus). 2. By extension, any copyright notice intended to achieve similar aims⁴².

The goal of the licensing system of copyleft is to keep software free. This goal is attained by copyleft by using copyright law and flipping it over to keep software free, i.e....instead of privatizing software, copyright serves the opposite of this usual purpose⁴³. Copylefting of a software program involves copyrighting the program through a copyright statement, recognizing the authorship right in the program and by adding terms and conditions which give away some of the rights conferred by copyright⁴⁴. Though the distribution terms allow free use, modification and sharing of programs, copyright in the original work is preserved by obliging the users to distribute the program under the same conditions recognizing the original owners right and by not attributing to the original author the changes made subsequently to the program. This mode of licensing through copyright law ensures the free availability of the work and its derivatives to the user public⁴⁵.

The copyleft licence by asserting copyright over the software and by limiting the users rights subject to the terms of the licence, acknowledges the existence of copyright as an integral part of the open source movement⁴⁶. The right of copyright is thereby employed by the open source software developers to grant freedoms as distinguished from the proprietary software developers who use copyright to subtract users freedoms⁴⁷. Also, the source code of software

programs being accessible to the users and the freedom of running, modifying and improving the software being free, the open source software movement uses copyright through the concept of copyleft to guarantee the four freedoms of free software.

General Public Licence (GPL)'s Use of Copyright

The purpose of licensing under copyright law is to enable a licensee to use a copyrighted product with restrictions, which might include the right of the copyright owner to retain his ownership right. The terms of a proprietary licence restrict the use of the source code of a software program by copyrighting the code and by making available the software for use under special licensing agreements⁴⁸. These licensing agreements, which limit the use of the software to customers who have brought the same, rarely make available the source code on payment of royalty and subject to limitations⁴⁹. The proprietary software licenses thus limit the freedoms of users when compared to the open source licences, which grant the freedoms by keeping software free.

Copyleft licenses the software programs under the GNU General Public Licence through its licensing terms and agreements guarantees users the freedom of distribution of free software, freedom to receive the source code and the liberty to modify or alter the software or use it in new free programs⁵⁰.

The preamble to the licence seeks to attain these freedoms by stating:

“When we speak of free software, we are referring to freedom, not price. Our General Public licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive the source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such 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. And you must show them these terms so they know their rights”⁵¹.

The preamble, which is the core of GPL protects the rights of users by preventing others from asserting their rights over the program. It however carries with it certain responsibilities to be shouldered by the user⁵². The responsibilities guarantee the freedoms by obliging the user to pass on to every other user the same freedoms, which would also include the freedom in respect of the derivative versions of the work.

The user of a software program who receives a source code licensed under GPL is therefore under an obligation

while making alterations to the source code to further license the same under GPL. By obligating the user to credit the authorship status of the source code to the original author and that of the subsequent alterations to the user, GPL preserves the originality of the source code. The Python 1.6 Beta 1 CNRI Open Source License, one of the most permissive open source licences requiring the notice of copyright recognizing the authorship right of the author proves to be an excellent example expressing GPL’s use of copyright⁵³.

Development and Dissemination of Software Programs

The open Source Software Movement by making the source code free through its licences for users to share, distribute and modify, increases the number of hobbyist’s programmers working on a program to improve the software. The President of one of the largest open source software commercial distributors, Red Hat, attribute the success of open source software programs to the growing number of engineers of reputed organizations working on the program to develop it⁵⁴. Due to the increased number of programmers accessing the source code and working on it, the Open Source Software Movement argues to promote the development and dissemination of software programs. The advantages of using an open source software licence for a software program, facilitating entry into the market of improved versions of software and thereby leading to innovation in the software industry are manifold.

Access to Source Code

The open source software developers argue that the software developed through open sourcing is much improved in quality compared to that developed by proprietary software developers. This feature of the open source software is attributed to the large number of developers working on the software program to improve it. The source code when not made available along with the software but made available for downloading on the Internet, facilitates easier accessibility and increased use of the software program. The developers of open sources being hobbyists and after-hour programmers; they are able to employ their experience and expertise gained by working for proprietary software companies in developing open source softwares. The open source movement expresses its upper hand over proprietary software by stating:

“When programmers on the Internet can read, redistribute, and modify the source for a piece of software, it evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing. The open source community has learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a few programmers can see source code and everybody else must blindly use an opaque block of bits”⁵⁵

Easy adaptability of open source softwares to the different environments

has been attributed to the continuous and intensive review, testing and debugging of the source code of the open source software⁵⁶. The adaptability feature of the open source software’s has led the public of the software community to encourage the use of open sourcing as a means of extending the life of an application⁵⁷.

The fact of having numerous contributors, process of undergoing an iterative development and following debugging and testing cycles establishes the reliability and stability of open source software over proprietary software⁵⁸. The popularity of Linux as a significant open source software project and as a base providing for the working of different systems of the corporate and mainframe computers arises from its intensive peer review status⁵⁹. This feature has been found lacking in proprietary software’s for the reason of the source code being inaccessible to the developers⁶⁰.

Peer Review and Debugging

The bugs that are found in a software program are two fold, bugs that are detected in the early stages of software development and those which produce the run-time errors (bugs that can be detected only when the program is run on a computer)⁶¹. In closed softwares where a sole or only few developers are allowed to work on the source code, detection and fixing of bugs present a huge task. Due to lack of sufficient technical hands, it is the user who usually detects bugs that escape detection, on running the software program. Though detected, closed software developers mostly refrain from fixing the bugs for the fact that fixing

bugs after release of the software proves to be an expensive affair⁶². It may be noted in this context that 'service packs' released by Microsoft to fix bugs are made available only in response to bulk demands and made available less frequently⁶³.

One of the advantages, which have been noted by the open source advocates in allowing numerous programmers to work on the source code, has been the ability to trace and fix bugs easily⁶⁴. As the number of programmers working on a source code is really high, the bugs are easily detected and fixing of the bug is also easily taken care of by some other programmer in the mean time. This debugging process of the open source software commonly referred to as parallel debugging is considered to be one of quickest approaches in the fixing of bugs⁶⁵.

Describing about the parallel debugging process of Linux, Stallman quotes Linus Torvalds

"My original formulation was that every problem "will be transparent to somebody". Linus demurred that the person who understands and fixes the problem is not necessarily or even usually the person who first characterizes it. "Somebody finds the problem," and "somebody else understands it. And I'll go on record as saying that finding it is the bigger challenge. "But the point is that both things tend to happen quickly"⁶⁶.

Aiding to the detection and fixing of bugs is the process of continuous peer review followed by open source software. The Internet serving as a mode for downloading the fixes as soon as they

become available and the constant and rigorous peer review process of the open source software's ensuring recurrent availability of the source codes, bug detection and fixing works with ease in open source softwares⁶⁷.

Credit-Giving Approach

To be considered as an open source software, apart from allowing access to the source code, the software must also allow free redistribution of the software and the source code distribution under the same terms as the original software. Distribution without discrimination among persons and specific fields, ensuring availability of same rights to all, distribution without affecting the rights of other programs distributed with it and most importantly from the point of view of copyright and software development, acknowledging the author's right of attribution and integrity should be allowed⁶⁸.

The authorship right of attribution and integrity is preserved by affixing the copyright notice acknowledging the authors work and by preventing subsequent developers from crediting the modifications done by them in the name of the author. Also, the terms of the licence oblige a licensee to distribute the source code unmodified under a mechanism that combines the altered code when run on a computer and by prohibiting endorsement of products derived from the code to the original author⁶⁹. This mode of acknowledgement has been referred to as an important characteristic of open source licences; "the right to attribution (to get credit for

his/her work) and right to misattribution (not to have other people's work accredited to him/her)"⁷⁰.

From the standpoint of facilitating the development and dissemination of software, this feature of the open source software holds great relevance. The programmers also known as the hackers working on the software to improve and develop it, receive no monetary gain for the efforts put forth by them. The Open Source Software Community describes the compensation for a hacker's effort to lie in the respect that he/she receives from their peers rather than being monetary⁷¹. This credit giving approach has been stated to encourage the hackers to engage in spending more hours in their programming efforts thereby improving the software programs and by bringing into the market improved and novel versions of software's.

Non-Exclusive Prerogative

Open source software is the result of contributions from many programmers. The proprietary software has a sole manufacturer having a monopoly over the product. The distribution and licensing rights in the product lie with the sole entity. Therefore, the manufacture and future of the software depends on the sole manufacturer.

In the case of software licensed under open source, the ownership of the works exists with different groups, individuals or companies. This non-exclusive right of ownership is significant for the future of the software industry. Once a proprietary software manufacturer ceases the production and distribution of software in

the market, the further development of the program also ceases to operate thereby affecting the software itself and indirectly the user public⁷². The purchaser of proprietary software relies solely on the vendor of the software and the act of stopping production by the vendor affects the business interests of the company. Further, the source code not being accessible to others prevents production of derivative versions of the software.

The advantage of the open source software over proprietary software in this regard exists in not having a sole entity as the owner thereby ensuring the continued availability of the software and its code. Even when a developer from whom the software owes its origin ceases to work on the program, the further availability and improvement of the program is guaranteed by a different developer, who taking advantage of the licensing scheme of open source is able to produce new and improved versions of the same⁷³.

Cost Free Re-distribution of Derivative Works

One of the distribution terms of an open source licence states "the licence shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The licences shall not require a royalty or other fee for such sale"⁷⁴. This permission to distribute software ensures unregulated redistribution of derivative works. The relevance of the distribution term lies in allowing a subsequent developer/licensee to use the derivative work without a fee.

Also, granted to the licensee is the right to create derivative works free of cost⁷⁵. Some of the greatest beneficiaries of this free use licence scheme have been the educational institutions, students and other users who have been able to use and work on the software systems without the hassle of going through the terms of non-disclosure agreements⁷⁶. The success of Linux kernel has been attributed to this free use feature provided by open source⁷⁷. The beneficiaries have been able to use the derivative works through the Internet, which in turn has aided in the widespread propagation of software systems available through the Net.

Another great benefit in allowing this free use has been to the proprietary software industry itself. To compete with the free use software industry of open source, proprietary software manufacturers have had to enhance the technical and other features of their programs thereby enabling release of better and improved softwares⁷⁸. It is worth noting in this context the moves by some of the major software manufacturers. Along with major software corporates like Intel, Motorola, Texas Instruments and Analog Devices funding the development of free GNU compiler for C language, the US Air Force funds the GNU Compiler for the Ada language⁷⁹.

Successful Open Source Softwares

The success of open source as a tool in furthering innovation in the software industry will be evident if one takes a look at some of the existing open source softwares and their widespread

application. The two open source softwares given below provide a good example of how the open source movement has affected the software industry spanning the applications over the Net.

BIND (Berkeley Internet Name Domain)⁸⁰

BIND is an open source software program developed by UC Berkeley supported by the Internet Software Consortium controlling the Domain Name System of the web and e-mail⁸¹. The importance of BIND as an integral part of the Internet lies in its features. The BIND is responsible for enabling the delivery of e-mails to the right destination⁸². The domain name server of BIND provides support to an organization in building its naming architecture on the server and its resolver library serves the role of translating domain names into addresses⁸³.

Sendmail⁸⁴

Sendmail developed at UC Berkeley by Eric Allman is another open source software program, which affects the bulk of the activities of users of the Net⁸⁴. The features of sendmail include "Internet work mail routing facility, featuring aliasing and forwarding, automatic routing to network gateways, and flexible configuration"⁸⁵.

Long-Term Effects of the Movement on Computer Software Protection

The features of open source software discussed earlier facilitate the development and dissemination of

software. Though the open source movement benefits the software industry, it is not free from legal and other risks associated with it.

Copyright Infringement

One of the most evident issues, which would arise in the open sourcing of software relate to copyright infringement. Section 17 U.S.C 106 gives the copyright owner the following exclusive rights:

- “(i) the right to reproduce a copyrighted work,
- (ii) the right to prepare derivative works based on the copyrighted work,
- (iii) the right distribute copies of the copyrighted work,
- (iv) the right to perform the copyrighted work publicly, and
- (v) the right to display the copyrighted work publicly”⁸⁷.

Open Source Software Movement makes some of these exclusive rights non-exclusive, by granting the rights not just to the copyright owner but also to subsequent developers/licensees. The effect of making these rights publicly available poses an issue from the point of view of copyright infringement.

The development of software under open source is the result of countless hours of work by numerous programmers. This being so, the exact owner of a particular piece of code is sometimes untraceable. The way in which this affects copyright is when a programmer introduces an infringing code into an existing code base and due to the

innumerable number of contributors allowed to work on the program, it becomes difficult to trace the infringer⁸⁸. The brunt of this act of infringement is to be borne by the licensee and not the contributors of the code, for, the license has no “intellectual property representations, warranties or indemnity obligations”⁸⁹ benefiting the licensee but it shields the licensor with the existing disclaimer of warranty⁹⁰.

Copyright in Derived Products

With respect to products, which may be derived from software licensed under open source, GPL states that “if identifiable sections of that work are not derived from the program, and can be reasonably considered independent and separate works in themselves, then this license, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the program, the distribution of the whole must be on the terms of this license, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it”⁹¹.

The term of licensing under GPL goes against the very goal of open source movement. It snatches the copyright of a proprietary software dealer or a developer who develops a new software program derived from the open source product. The aim of the Open Source Software Movement since its inception is to encourage free use of software by granting the right of modification,

sharing, etc through the mode of copyright by granting these freedoms and not to subtract the copyright of users/licensors. GPL snatches the copyright of a commercial software dealer or programmer when he by his importunate efforts develops a software product derived from the source code available free under the GPL licence. The copyright in the source code for the new product though derived from open source software, remains with the developer. To distribute the product (the product being a derivative work from an open source software, can be distributed only through GPL or other open source licences) the developer has to make available his source code to others free of cost⁹². By making available his (be it a commercial software dealer who wants to use the free software) source code in the derived product free for distribution, the developer submits his exclusive rights that are granted by copyright in his software product. By making available the source code of his derived product, he loses his proprietary right of copyright in the product. Also, by not being compensated for his creation, the labour of the developer is exploited free. The long-term effect of this term of license could result in destabilizing the commercial software industry by creating an atmosphere where users use the software without paying for the product⁹³.

Copyright in Non-Source Code Files⁹⁴

The application of the term of licensing under GPL with respect to derivative works elicits attention with regard to the copyright of non-source code files that

may be distributed along with a GPL licensed software program. The non-source code files associated with a software program may comprise a graphic file, an image file, etc. The copyright (provided the work is an original work of art) in the file rests with the author.

GPL affects the copyright in the source code of a file, if the file is a product that is derived from a GPL licensed program⁹⁵. With regard to a non-source code file that is derived from a GPL licensed program, the licensee is under an obligation to further license the same under GPL thereby affecting his copyright in the file. This is because the term of licensing of GPL, obligates a licensee to distribute even a non-source code file that may be derived from a GPL'd program under the GPL⁹⁶. The relevant term of licensing stating "when you distributeas a part of a whole which is a work based on the program, the distribution of the whole must be on the terms of the license"⁹⁷ conveys the idea that even non-source code files come under the umbrella of the licensing provision. The license, however, does not affect the copyright in a file that is not derived from a GPL'd program.⁹⁸

Copyright Protection of Softwares on the Net

Open source software is a product of the programming skills of thousands of hackers. Apart from the softwares offered for sale in the market, the Internet technology has made it possible to distribute softwares through the Net. The access to source code being free,

programmers around the world are able to work on open source softwares to modify and distribute it through the net. This provides a leeway for a programmer to make minor changes (there being no exact criteria on what a substantial change or alteration to a software should be under GPL) to the software and to distribute the same under GPL. The result of such a freedom would be to increase the number of software programs distributed on the Net, which might also include those without significant changes and those that are the hobbyist work of after-hour programmers. The ultimate effect would be the cluttering up of softwares distributed over the Net.

For an ISP it would become difficult to monitor all the softwares loaded on the Net, which might also include infringing source codes of the proprietary software dealers increasing his liability for copyright infringement. The user public is thereby able to have a free ride on the source code of commercial software developers. As observed by Microsoft, public would consider copying of software to be morally neutral resulting in the devaluation of copyright and forcing “intellectual property rights into public domain”⁹⁹.

Some Allied Issues

There are some related issues, which may arise in the context of legal implications of open source. An often-discussed implication under open source software’s has been the problem of free riding. Free riding can be quite a common phenomenon when contributions to the code exist from innumerable number of

programmers working on it. Free riding could be two faced – it could exist as an advantage in open source and also pose an eternal problem. Contributing to an existing code base as stated earlier in the paper could improve the program, aid in fixing bugs adding value to the product¹⁰⁰; at the same time by utilizing the work of others, it could hamper development of large projects in open source¹⁰¹.

Another issue relates to the viral nature of GPL. GPL necessitates releasing the new codes added to an existing code base released under GPL to be distributed through GPL itself¹⁰². A classic example is the dilemma faced by Netscape in releasing source codes for branded third-party components of Communicator under GPL, when Netscape uses some of the codes of Communicator in its products¹⁰³.

Other issue is the security concern of most companies on the exposure of open source codes to virus attacks¹⁰⁴, a prominent one being the attack of Trojan Horse virus¹⁰⁵. The legal validity of open source licences is still at bay and debates are still raging, especially on the validity of GPL¹⁰⁶. An interpretation of the licensing schemes while they are tested in the courts could probably open up a Pandora’s box where further implications could come into the picture.

Conclusions

Open Source Movement owes its origin to the existence of the proprietary right of copyright. The very scope and survival of the movement depends on copyright. The movement has employed

copyright law through its licensing scheme of copyleft, which in turn uses GPL and other open source licenses to facilitate the development and dissemination of software programs.

Though the movement is very much beneficial to the software industry in creating software that is much developed compared to that of the commercial software developers, the legal risks involved in the area are very crucial to the existence of the movement and copyright. As the licensing system under GPL involves the surrender of the rights to the Free Software Foundation, the future of softwares licensed through open sourcing depends on this entity. There being no guarantee to the existence of the entity and the chances of dissolution of the same not being improbable, the fate of these softwares and their copyright depends on the Free Software Foundation.

References and Notes

- 1 See Stanley Lai, *The Copyright Protection of Computer Software in the United Kingdom* (Hart Publishing, Oxford, Portland) April 2000, p 1
- 2 A compiler software automatically transforms an object code into a source code.
- 3 The Binary form refers to expression of the instructions in hexadecimal numbers
- 4 See Laddie, Prescott, Vitoria, Speck, Lane, *The Modern Law of Copyright and Designs* (Butterworths, London) 1995, p 1608
- 5 See Mathias Strasser, A new paradigm in intellectual property law?: The case against open source, *Stanford Technology Law Review*, 4, 2001
- 6 Ibid.
- 7 See 17 U.S.C 101; Halpern Sheldon W, Craig Allen Nard, Port Kenneth L, *Fundamentals of United States Intellectual Property Law: Copyright, Patent and Trademark* (Kluwer Law International, The Hague 1999), p 12
- 8 See 797 F.2d 1222, 1233 (3d Cir. 1986)
- 9 See Supra n. 7
- 10 See 797 F.2d 1222, 1233 (3d Cir. 1986)
- 11 See 714 F.2d 1240, 1249 (3d Cir. 1983)
- 12 See Supra n. 5
- 13 The object code not being intelligible to human beings, the court had to decide whether the scope of copyright protection extended to a form of expression (the object code being a form of expression).
- 14 See Teresa Hill, Fragmenting the Copyleft Movement: The public will not prevail, *Utah Law Review*, 797, 1999
- 15 See Open source software: what is it? why use it? and what's gotten into netscape? <http://www.tux.org/~niemi/opensource/customer-case.html>
- 16 See Horne Natasha T, Open source software licensing: Using copyright law to encourage free use, *17 Georgia Street University Law Review* 863
- 17 See Gomulkiewicz Robert W, How copyleft uses license rights to succeed in the open source software, Revolution and the Implications for Article 2B, *36 Houston Law Review* 179
- 18 See Supra n. 16
- 19 See Potter Shawn W, Opening up to open source, *6 Richmond Journal of Law & Technology*, 24
- 20 Ibid.
- 21 See Supra n. 15
- 22 The term open source has often been referred to as a phrase that is ambiguous and is often used synonymously with the term free.
- 23 The word free under open source software may be interpreted in the words of Stallman, "think of free speech, not free beer"
- 24 Stallman condensed, <http://www.gnu.org/events/rms-nyu-2001-summary.txt> Also see <http://www.gnu.org/events/rms-nyu-2001-transcript.txt>
- 25 See Supra n.14
- 26 See Paula Samuelson et al. A manifesto concerning the legal protection of computer programs, *Columbia Law Review*, 2308, 1994

- 27 Though Richard Stallman has been considered to be the father of the open source software by the open source community, Richard Stallman states that he has never used the term open source for free software.
- 28 See Supra n. 14
- 29 Ibid.
- 30 See Supra n. 19
- 31 See Supra n. 5
- 32 Kernel is the fundamental part of a program, it is the part of the operating system that is closest to the machine...it activates the hardware directly or interfaces to another software layer that drives the hardware.
- 33 See Supra n. / 19
- 34 Linux is the core of the free Unix operating system for the PC and other hardware platforms
- 35 See Supra n. 19
- 36 See Supra n. 24
- 37 Ibid.
- 38 The four freedoms have been zero-indexed by Stallman
- 39 See Supra n. 24
- 40 Ibid.
- 41 See *The Concise Oxford Dictionary of current English* (Oxford University Press, Oxford 1990)
- 42 See The Jargon Lexicon <http://www.tuxedo.org/~esr/jargon/html/entry/copyleft.html>
- 43 See The GNU Project <http://www.fsf.org/gnu/the-gnu-project.html>
- 44 See Michael Stutz, Applying copyleft to non-software information, <http://www.gnu.org/philosophy/nonsoftware-copyleft.html>
- 45 See Supra n. 5
- 46 See Lambert Paul, Copyleft, copyright and software IPRs: Is contract still king? *European Intellectual Property Reports*, 165, 2001
- 47 See What is copyleft? <http://www.gnu.org/copyleft/copyleft.html>
- 48 See Supra n. 16
- 49 See Gabriel Richard P & Joy William N, Sun Community Source License Principles, <http://www.sun.com/981208/scsl/principles.html>
- 50 See GNU General Public License, Version 2, June 1991, <http://www.gnu.org/licenses/gpl>.
- 51 Ibid.
- 52 Id.
- 53 See Supra n. 16
- 54 See Bobko Patrick K, Open source software and the demise of copyright, 27 *Rutgers Computer & Technology Law Journal*, 51
- 55 See Introduction to Open Source, <http://www.opensource.org/intro.html>
- 56 See Supra n. 54
- 57 See Free Software / Open Source: Information Society Opportunities for Europe? Working group on Libre Software, <http://eu.conecta.it/paper/paper.html> No binary-only application is believed to have survived beyond 10 years in unmodified form, but open source software systems from the 1980's are still in widespread use (although in many cases conveniently adapted to new environments).
- 58 See Demystifying open source: How open source development works <http://www.linuxcare.com/about-us/collateral/whitepapers/wp-mystify-1099>
- 59 See Supra n. 54
- 60 Ibid.
- 61 See Supra n.5
- 62 Ibid.
- 63 See Frank V W, Why I hate Microsoft, "A personal lengthy, but highly articulate outburst", <http://www.euronet.nl/users/frankvw/IhateMS.html>
- 64 See Supra n.54
- 65 See Open Source Software: A (New?) Development Methodology, Halloween Document I (Version 1.14), <http://www.opensource.org/halloween/halloween1.html> It's been stated that when a Tear Drop IP attack was posted on the web, within 24 hours the Open source community had the fix available for download
- 66 Ibid.
- 67 See Supra n. 54
- 68 See The open source Initiative: open source definition, <http://www.opensource.org/docs/definition.html>
- 69 See Supra n. 17

- 70 See Supra n. 54
 71 See Supra n. 17
 72 See Supra n. 57
 73 Ibid.
 74 See The Open source definition, Version 1.8, <http://www.opensource.org/docs/definition-plain.html> also see Supra n. 67
 75 See Supra n. 16
 76 See Supra n. 57
 77 See An Overview of “Open Source” Software Licenses, A Report of the Software licensing committee of the American Bar Association’s Intellectual Property Section, <http://www.abanet.org/intelprop/opensource.html>
 78 Ibid.
 79 See Stallman Richard, Why software should not have owners <http://www.gnu.org/philosophy/why-free.html>. The US Air Force has ceased funding for the GNU Ada Compiler. It is now being funded commercially.
 80 See Tim O’Reilly, Ten myths about open source software http://opensource.oreilly.com/news/myths_1199.html
 81 Ibid.
 82 See Online Catalog DNS and Bind, 3rd Edition <http://www.oreilly.com/catalog/dns3/desc.html>
 83 See Internet Software Consortium, ISC Bind <http://www.isc.org/products/BIND/> The current versions of BIND are available at <http://www.isc.org/products/BIND/>. The latest version, BIND Version 8.3.0-T2A was released on October 1, 2001
 84 See Supra n. 79
 85 Ibid.
 86 See Gregory Neil Shapiro, *Sendmail*, <http://freshmeat.net/projects/sendmail/> The latest version released on October 1st 2001 is available at <ftp://ftp.sendmail.org/pub/sendmail/>
 87 See Supra n.7
 88 See Supra n. 76
 89 Ibid.
 90 Id. The ‘No Warranty’ section of the license states
 “11. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law.
 Except when otherwise stated in writing the copyright holders and/or other parties provide the program “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.
 12. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages”.
- 91 See Supra n. 47
 92 See Microsoft Press Pass—The Commercial Software Model, <http://www.Microsoft.com/presspass/exec/craig/05-03sharedsource.asp>
 93 Ibid.
 94 See Slashdot! Attorney Dan Ravicher on Open Source Legal Issues <http://slashdot.org/interviews/01/06/05/122240.shtml>
 95 Refer the earlier section, (3.1.2) Copyright in Derived products, to see how GPL affects a source code file
 96 See Supra n. 91
 97 See Supra n. 47
 98 See Supra n. 94
 99 See Supra n. 89
 100 See Weber Steven, The Political Economy of Open Source, BRIE Working Paper 140, E-conomy Project TMWorking Paper 15, June 2000 <http://e-conomy.berkeley.edu/publications/wp/wp140.pdf>

- 101 See Mustonen Mikko, Copyleft—the economics of Linux and other Open Source Software, www.berlecon.de/services/en/iew4/papers/paper3.pdf, p 6
- 102 See Jim Hamerly and Tom Paquin with Susan Walton, Freeing the Source The Story of Mozilla, <http://www.oreilly.com/catalog/opensources/book/netrev.html>, Also see supra n.100
- 103 Ibid.
- 104 See Gatto James, The Impact of Open Source on IP Revenue Opportunities, www.vtip.org/VATEchFinal.ppt
- 105 See Clark Drew, Defense, cybersecurity officials praise 'open source' software, National Journals Technology Daily, October 29, 2002, <http://207.27.3.29/dailyfed/1002/102902td2.htm>
- 106 See Position Paper Sun Microsystems, Open Source Software, <http://www.sun.com/aboutsun/policy/opensource.html>