

Software Freedom, Open Software and the Undergraduate Computer Science Curriculum

John E. Howland
Department of Computer Science
Trinity University
715 Stadium Drive
San Antonio, Texas 78212-7200
Voice: (210) 999-7364
Fax: (210) 999-7477
E-mail: jhowland@Ariel.CS.Trinity.Edu
Web: <http://www.cs.trinity.edu/~jhowland/>



Trinity University Computer Science

Abstract

The early history of computing records the sharing of computer software in source form. The sharing took place between programmers, and users of programs throughout the computing industry. Later software vendors became rather proprietary with software, using copyright, patent and trade-secret law to restrict the use of programs. In the 1980's, Richard Stallman led a small group of people who were interested in reclaiming the early software freedom enjoyed by programmers and users. These efforts have grown into what is now known as the *Open Source Revolution* and produced such remarkable products as Emacs, gcc and Linux. The relationship of software freedom, open sources and the undergraduate computer science curriculum are presented.



Introduction

Recently, the Linux operating system has become a topic for discussion, even amongst persons outside the computing industry, as the public stock offering of RedHat Software has made three persons associated with RedHat billionaires (at least on paper). RedHat Software distributes a version of the Linux system which is the end result of the efforts of thousands of programmers who have given freely of their time and energy to develop software which is distributed under a license which places few restrictions on those who use the software. The software itself is available at no cost on the Internet or, for a nominal fee, on convenient CDROM packages which include documentation and installation manuals and various levels of customer support.



Introduction

How is it possible that a company could have a stock market value of several billion dollars when the main product sold by the company contains a license which requires that the product must also be given away at no charge? Bob Young, president of Red Hat Software, says that he is selling bits on the basis of brand-name endorsement. Young says that he's in the same kind of business that Evian, a company selling "millions of dollars of French tap water" solely on the strength of its brand and an "irrational fear that the water coming from your tap is not to be trusted". [HAC 1999]



Introduction

There is something more fundamental supporting the Linux software movement and RedHat Software than selling water which is usually free at the tap. It is the concept of software freedom; not free software, but software whose license does not restrict a user's freedom in any significant way.



Software Freedom

The early history of computing records the sharing of computer software in source form. The sharing took place between programmers, and users of programs throughout the computing industry. Later software vendors became rather proprietary with software, using copyright, patent and trade-secret law to restrict the use of programs.



Software Liberty

- Freedom to run a program for any purpose.
- Freedom to modify a program to suit your needs. To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.
- Freedom to redistribute copies of a program, either gratis or for a fee.
- Freedom to distribute modified versions of the program, so that the community can benefit from your improvements.



The GNU System

Richard Stallman decided to create an operating system compatible with the widely used proprietary Unix operating system. He used the recursive acronym GNU which stood for “GNU is not Unix” to name the system.

- Original plan – release a complete system
- System components, compilers, editors, utilities were released when available to augment existing systems
- Kernel – Hurd (collection of servers running on top of CMU micro kernel, Mach) still not reliable
- Linux kernel



The GNU Software License

- Method of software distribution which does not restrict the freedom of software users
- Copyleft – all right reversed
- Run the program
- Copy the program
- Modify the program
- Distribute/sell modified version, modified versions must also be free
- Cannot add restrictions of their own



Open Software

Richard Stallman has been controversial in his unwillingness to relax any of the restrictions in the GNU software license. Many feel that his refusal to use phrases other than *free software* to describe his concept of software freedom hurts the overall acceptance of his ideas.

- Something which is free could not possibly be supported well enough to be relied upon.
- Eric Raymond has been instrumental in establishing a set of licensing standards which are less restrictive than the GNU Public License in the area of what may be done when changes are made to someone else's program.



The Open Source Definition

- *Free Redistribution* The license may not restrict any party from selling or giving away the software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale.
- *Source Code* The program must include source code, and must allow distribution in source code as well as compiled form.
- *Derived Works* The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.



The Open Source Definition

- *Integrity of The Author's Source Code* The license may restrict source-code from being distributed in modified form only if the license allows the distribution of “patch files” with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
- *No Discrimination Against Persons or Groups* The license must not discriminate against any person or group of persons.
- *No Discrimination Against Fields of Endeavor* The license must not restrict anyone from making use of the program in a specific field of endeavor.



The Open Source Definition

- *Distribution of License* The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- *License Must Not Be Specific to a Product* The rights attached to the program must not depend on the program's being part of a particular software distribution.
- *License Must Not Contaminate Other Software* The license must not place restrictions on other software that is distributed along with the licensed software.



Reading to Write

Imagine a college level writing course, taught by the English Department, where students are never given reading assignments of great works of literature. Good writing, like good programming, is partly an art. Most would agree that developing good writing skills involves reading great works of literature as well as doing lots of writing.

- Reading/modifying source code of a compiler or interpreter
- Reading/modifying source code of an operating system



Software Service

The concept of professionals providing their services without compensation for the purpose of improving society, offering aid in a time of need, or to further the goals of the profession is well understood as required behavior for professionals.

- Each Open Source project needs volunteers.
- Documentation writing
- Programming



Open Software and Our Graduates

Does Open Software provide viable employment or business opportunities for our graduates? A number of companies have announced active participation in Open Software projects.

- Netscape
- Apple Computer
- Silicon Graphics
- Sun Microsystems
- IBM
- RedHat, VA Linux and Open Source Support Companies



Open Software and Computing Laboratories

Trinity University has used Linux based systems for several years with excellent results.

- Web, NFS, NIS, Mail and FTP servers
- General Workstations
- Parallel Processing Clusters
- Graphics Workstations



Conclusions

Michael Jensen [Jen 1999] argues that higher education, being not for-profit, should question its relationship with the commercial software industry. He writes “We need to decide what kind of relationship academe should have with the tools that underpin its knowledge bases – that of a huge corporate customer that goes to private industry for software, or of a supporter and underwriter of open and free software tools that serve our needs” .



Conclusions

Peter Linkins remarked, during a meeting at the Online Computer Library Center in Ohio [Jen 1999] : “A for-profit’s mission is to create as much value for its stockholders as possible, within the constraints of society. The non-profits’ mission is to create as much value for society as possible, within the constraints of its money.” Jensen notes [Jen 1999]: “Finally, we should remember that research and scholarship are fundamentally open-source enterprises. The research on which scholarship builds is always cited; the methodology of any study is explained as a repeatable framework; and theoretical preumptions are made clear as part of any published argument. Those principles generally lead authors to be careful, and help other scholars to test an author’s conclusions.”



Conclusions

- Software freedom has the potential of re-kindling some of the excitement and enthusiasm experienced in the early history of computing. Free and open sharing of software ideas and their expression is entirely consistent with the academic freedom of sharing of ideas which all of us within academia so highly cherish. Students benefit from the reading of source code of real programs written by great programmers and designers. They can see first hand the abstractions and data structures which have been designed to deal with the complexity of the program.
- By using Open Software in our courses, we can show the complexity of real programs and instrument them to produce working models of the theory and concepts we teach.



Conclusions

- Our students can benefit from the contributions, no matter how small, they make to an Open Source or GNU project. Such service adds quality to their professional life just as the service performed by professionals in other fields.



References

- [App 1999] Apple Computer, “Open Source Projects at Apple”
<http://www.publicsource.apple.com/apsl/> , 1999.
- [Art 1998] The Artistic License, <http://language.perl.com/misc/Artistic.html> , 1998
- [Bar 1999] Barr, Joe, “IBM’S Secret Summit” , Linux World,
http://www.linuxworld.com/linuxworld/lw-1999-09/lw-09-ibm_2.html ,
September 1999.
- [DiB 1999] *OPENSOURCES, Voices from the Open Source Revolution*, Edited by Chris DiBona, Sam Ockman and Mark Stone, O’Reilly & Associates, Sebastapol, California, 1999.
- [HAC 1999] Hackv an, Stig, “Reverse-engineering the GNU Public Virus” , Linux World,
<http://www.linuxworld.com/linuxworld/lw-1999-09/lw-09-gnu.html?09-21> ,
1999.



- [Hac 1999] Hackv n, Stig, “Radical Software Environmentalism”, Linux World, <http://www.linuxworld.com/linuxworld/lw-1999-09/lw-09-gnusidebar.html?09-21> , 1999.
- [Hui 1992] Hui, Roger, K., *An Implementation of J*, Iverson Software, Toronto, Ontario, 1992
- [Jen 1999] Jensen, Michael, “Information Technology at a Crossroads: Open Source Computer Programming”, The Chronicle of Higher Education, October 29, 1999.
- [Lin 1999] Linux Magazine, Interview with Richard M. Stallman, July, 1999.
- [Net 1998] Netscape, Mozilla & Netscape Public Licenses, <http://www.mozilla.org/MPL/> , January 1998.
- [Ope 1999] “The Open Source Definition”, <http://www.kr.debian.org/OpenSource/osd.html> , 1999.
- [Ray 1997] Raymond, Eric, “The Cathedral and the Bazaar”, <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/> , 1997



- [Ra1 1997] Raymond, Eric, "Homesteading the Noosphere",
<http://www.tuxedo.org/~esr/writings/homesteading/> , 1997
- [Ray 1999] Raymond, Eric, "The Magic Cauldron",
<http://www.tuxedo.org/~esr/writings/magic-cauldron/> , 1999
- [SGI 1999] Silicon Graphics "Linux", <http://www.sgi.com/developers/technology/linux/>
, 1999.
- [SGi 1999] Silicon Graphics "SGI Linux Environment 1.0 with RedHat Linux 6.0",
<http://www.sgi.com/software/linux/index.html> , 1999.
- [Tan 1997] Tannenbaum, Andrew, S., and Woodhull, Albert, S., *Operating Systems: Design and Implementation*, 2nd Edition, Prentice Hall, Upper Saddle River, New Jersey, 1997.
- [WSJ 1999] Hamilton, David, P., "Sun Microsystems, to Parallel Success of Linux, Makes Solaris Code Available", Wall Street Journal, Page B6, October 1, 1999.

