

The Role of Intellectual Property in Open Source Software

Dr. Lee A. Hollaar

Professor, School of Computing, University of Utah
www.cs.utah.edu/~hollaar

**IIPI Conference on Strategies for Building
Software Industries in Developing Countries**

May 19, 2004

Overview

- ◆ Open source and downstream requirements
- ◆ The need for strong IP laws
 - Copyright
 - Contracts
- ◆ Similarity in need to proprietary software
- ◆ But not too strong
 - Copying nonliteral aspects
 - Derivative works
- ◆ What about patents?

Open source and downstream requirements

Open Source Definition

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.

Rationale: The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.

The Free Software Foundation (FSF) General Public License (GPL)

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: . . .
 - b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. . . .

The open source *Quid pro quo*

- ◆ You may use open source as the basis for your program, or modify it
- ◆ Those modifications must be made available to the community on the same terms as the original software
- ◆ Definitely *not* “public domain”

Distribution, not use

- ◆ Note that these license requirements only govern the further distribution of the program and its modifications
- ◆ Copies necessary for use are governed by 17 U.S.C. 117, other license terms, or possibly “fair use”

The need for strong IP laws

GPL Preamble

We protect your rights with two steps:

- (1) ***copyright*** the software, and
- (2) offer you this ***license*** which gives you legal permission to copy, distribute and/or modify the software.

Need for strong copyright

- ◆ Gives open source creator the right to control terms for further distributions
- ◆ Without copyright, any unauthorized downstream distribution is, at best, a contract breach
- ◆ Problem for government agencies, if software is available outside of a contract

Need for strong contract law

- ◆ Important open source license terms must be enforceable
 - Downstream distribution
 - Disclaimer of warranties
- ◆ Must allow “take it or leave it” contracts
- ◆ Must allow contract assent by action
- ◆ Consumer protection laws might limit some license terms

Similarity in need to proprietary software

- ◆ Strong copyright laws
- ◆ Enforceable “shrink-wrap” contracts
 - Limitation on warranties
 - Restrictions on use, not distribution
- ◆ Contracts provide the basis for trade secret protection

But not too strong

Reaching new works

GPL Section 2:

- b) You must cause any work that you distribute or publish, that in whole or in part contains or is ***derived from the Program*** or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

Lesser (library) GPL

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

Derivative works

A “derivative work” is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted.

17 U.S.C. 101

Compilations

A “compilation” is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship.

17 U.S.C. 101

So, what is covered?

- ◆ Easy with static linking
 - Open source library is combined with new program to give a compilation
 - Distribution of compilation requires permission of library copyright owner
- ◆ Not clear for dynamic linking
 - No compilation distributed
 - Is a program “based on” a library?

Reimplementations

- ◆ Many open source projects are reimplementing proprietary programs
 - GNU operating system
- ◆ Clean room approach
 - Copyright laws must separate expression from functionality

Unconscious copying

- ◆ You don't have to be looking at program to copy it
 - *My Sweet Lord* and *He's So Fine*
- ◆ It doesn't have to be identical
- ◆ Data structures and overall program organization is also protected

Elements not protectable by

- ◆ The element's expression was dictated by reasons of efficiency, such as when it is the best way of performing a particular function.
- ◆ The element's expression was dictated by external factors, such as using an existing file format or interoperating with another program.
- ◆ The element's expression is a conventional way of writing something in the particular programming language or machine running the program.
- ◆ The element, at the particular level of abstraction, is an unprotectable process and not protectable expression.
- ◆ The element is taken from the public domain or is an unprotectable fact.

What about patents?

GPL Preamble

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

United States Patent [19]

[11]

4,135,240**Ritchie**

[45]

Jan. 16, 1979**[54] PROTECTION OF DATA FILE CONTENTS****[75] Inventor: Dennis M. Ritchie, Summit, N.J.****[73] Assignee: Bell Telephone Laboratories,
Incorporated, Murray Hill, N.J.****[21] Appl. No.: 377,591****[22] Filed: Jul. 9, 1973****[51] Int. Cl.² G06F 11/10; G06F 13/00****[52] U.S. Cl. 364/200****[58] Field of Search 340/172.5;
364/200 MS File, 900 MS File****[56] References Cited****U.S. PATENT DOCUMENTS**

Re. 27,239	11/1971	Ulrich	340/172.5
Re. 27,251	12/1971	Amdahl et al.	340/172.5
3,368,207	2/1968	Beausoleil et al.	340/172.5
3,377,624	4/1968	Nelson et al.	340/172.5
3,469,239	9/1969	Richmond	340/172.5
3,576,544	4/1971	Cordero et al.	340/172.5
3,599,159	8/1971	Creech et al.	340/172.5
3,631,405	12/1971	Hoff	364/200

3,683,418	8/1972	Martin	340/172.5
3,735,364	5/1973	Hatta	340/172.5
3,742,458	6/1973	Inoue et al.	340/172.5
3,761,883	9/1973	Alvarez	364/200

Primary Examiner—James D. Thomas*Attorney, Agent, or Firm*—Stephen J. Phillips**[57]****ABSTRACT**

An improved arrangement for controlling access to data files by computer users. Access permission bits are used in the prior art to separately indicate permissions for the file owner and nonowners to read, write and execute the file contents. An additional access control bit is added to each executable file. When this bit is set to one, the identification of the current user is changed to that of the owner of the executable file. The program in the executable file then has access to all data files owned by the same owner. This change is temporary, the proper identification being restored when the program is terminated.

4 Claims, 2 Drawing Figures

But patents may be the only way to protect a new method

- ◆ Trade secrets are unavailable for open source
- ◆ Copyright does not protect methods or ideas
- ◆ Ideas in code can be used if taken through a clean room

Why patents for open source?

- ◆ Moving beyond reimplementing to new systems and techniques
- ◆ Stop techniques from being used in proprietary software
- ◆ Develop patent portfolio to trade for blocking patents

For more information

Legal Protection of Digital Information,
published by BNA Books

<http://digital-law-online.info>

Also has CONTU Final Report and copyright history
from Patry's *Copyright Law and Practice*