# Free and Open Source Software for Librarians

## Why Software Freedom Matters

Scott Dexter
Samir Chopra

Computer and Information Science
Brooklyn College

# Overview

- Some foundational ideas; a bit of history
  - Definitions and dualities
- Copyright and Licensing
- Perspectives and Implications
- Reading List

# Duality: Code

Most executable software is instructions represented in 0s and 1s (binary)

- easy/fast for computer hardware to process
- but hard for humans to read/write

Programmers usually write source code in a language based on English and mathematics

- translated to binary by compiler software

# Duality: Distribution

## Distribution as only the executable binary

- Provides functionality;
  conceals many details of implementation and innovation
- Perceived by many to be cornerstone of competitive advantage in contemporary software market

## Distribution as source code (binary optional)

- Provides all details of implementation
- Allows possibility of others using/building on ideas

# A One-Slide History of Software

- Early mainframes: software just part of the package; each installation wrote, customized, and often shared
- 1968: IBM "unbundled" under pressure from US Department of Justice
- 1976: Gates to Homebrew Computer Club: sharing software is "theft"
- Early 1980s: spread of proprietary ideas about software hampers scientific research;
  Free Software Foundation (FSF) born

# Free Software

The FSF defines free software in a way that reflects its mission to

> preserve, protect, and promote
> the freedom to use, study, copy, modify, and redistribute computer software

Social goals:

- user agency: full control over all software on your computer
- community: encourage sharing software with your neighbors
- education

# From the Free Software Definition:

**Zero** The freedom to run the program, for any purpose.

**One** The freedom to study how the program works, and adapt it to your needs.

**Two** The freedom to redistribute copies so you can help your neighbor.

**Three** The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

# What about "Open Source"?

The Open Source Initiative (OSI) formulated the Open Source Definition (OSD) to make a pitch to industrial software engineers

> "Open source is a development method for software that harnesses the power of distributed peer review and transparency of process"

Technical goals:

- Quality, reliability, flexibility
- Cost
- "An end to predatory vendor lock-in"

# Duality: Advocacy

## Free Software Definition/Foundation

Ethical thrust: users' freedom, social good

## Open Source Definition/Initiative

Technical thrust: better code, more flexibility

## This duality can be traced back (at least) to the Phone Phreaks of the early 1970s

- first newsletter, *Youth International Party Line*, was political
- *Technological American Party* had a strong technical focus

# Duality: Property

Most commercial software is private property

- Not yours. . . the creator's
- You get a very limited license to use it in ways the creator chooses to allow
- Copyright law underwrites this perspective

Free software is (essentially) common property

- Intent is to build, protect, and expand a software commons that all can use
- Copyright law underwrites this perspective, too

# Licensing

So, what's the connection between copyright and the FS/OS Definitions?

Software licenses

FSF and OSI each have long lists of licenses they judge to meet the terms of the FSD/OSD

- How do these licenses work?
- How can there be so many?
- Where do they come from? Why would a hacker want to write something as boring as a license?

# Licensing and Copyright

Law gives copyright holder exclusive rights for a limited time

- to produce (and sell) copies
- to make derivative works
- to sell or assign these rights to others

Copyright originates with author (or employer)

FOSS licenses do more-or-less creative things with how copyrights are assigned to others

# “Permissive” licenses

## The meat of the BSD license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

## What does this “permit”?

# “Copyleft” Licenses

All rights reversed

A challenge:

- Suppose I provide some free software.
- I want people to have all four freedoms.
- But if someone distributes an improved/modified version, I want the source of *that* version to be available to me (and the world) too.

This is copyleft.

# The GNU Public License

The original copyleft license.

```
You may convey a work based on the Program . . . in the form
of source code . . . provided that you also meet all of
these conditions:

c) You must license the entire work, as a whole, under this
   License to anyone who comes into possession of a copy. . .
```

And if you distribute binary code, you have to make the source available, too.

# Duality: Freedoms

So, I'll let you use my program for any purpose, in any way.

But if you have modifications you want to distribute, my license constrains your options.

Copyleft explicitly restricts one freedom in order to support others.

# Political Science and Economics

I haven't talked much about the process by which free software is created . . .

How does a geographically dispersed community, often largely volunteer, produce such useful (= valuable) stuff?

- How are they governed/coordinated?
- What are their individual incentives?
- How is innovation possible?

# Ethics

Free software is the earliest experiment with "property" in the digital age.

- It's not clear that any previously conceived notion of property fully captures what free software is.

Free software is an incredible laboratory for the study of freedom

- How do a license's freedoms interact with community expectations?

Democracy, discourse, and community

- The community collectively updated the GPL in 2006–2007!

# Aesthetics

Free software is a vast collaborative creation open to public view.

- What can we learn about the creative process?
- Does software have an aesthetics?
- What can free software production tell us about artistic collaboration?
- How are software artisans trained?
- Is there a connection between aesthetic quality and "code quality"? Is free software better? Is it "more beautiful"?

# Scientific Inquiry

Philosophers suggest that science benefits from openness, mutual critique, and public testing

- These are certainly characteristics of free software
- Are they characteristics of computer science?

Software is increasingly the object of copyright, patent, trademark, and trade secret law

- What effect does this have on the scientific practice of computer science?
- What about other sciences, which are increasingly dependent on software and digitized data?

# Political Philosophy

Computing technology increasingly mediates our interactions with our environment.

- "Cyberspace" and "real" space are melding.
- Some serious scholars think it's reasonable to call ourselves cyborgs, in a very literal sense.

The software that controls this technology increasingly affects our modes of expression, our political actions, our very agency.

*The technical is political:*
*to free software is to free our selves.*

# Reading List

2004. Steven Weber. *The Success of Open Source*. Harvard University Press. [political science]

2005. J. Feller et al., Eds. *Perspectives on Free and Open Source Software*. The MIT Press.

2007. Samir Chopra and Scott Dexter. *Decoding Liberation: The Promise of Free and Open Source Software*. Routledge.

2007. Johan Söderbergh. *Hacking Capitalism: Free and Open Source Software*. Routledge. [economics]

2008. Chris Kelty. *Two Bits: The Cultural Significance of Free Software*. Duke University Press. [anthropology]

http://www.gnu.org/philosophy/philosophy.html