

# Open Source Software: Economics and Incentives

*"Who can afford to do  
professional work for nothing?"*

- Bill Gates, 1976

Kevin Lacker  
lacker@gmail.com

“Open” + “Source” =  
“Open Source”

```
int main(){  
    printf("hello world\n");  
    return 0;  
}
```

Sometimes interchangeable:  
“Free Software”

But “free” is a tricky word

# The Beginning of Time: 1970

- ◆ Market for software not too large
- ◆ Software strongly tied to hardware
- ◆ MIT, Berkeley, Bell Labs, Xerox PARC
- ◆ Unix and C started the “software layer”
- ◆ So AT&T tried to own Unix

# The Middle Ages: 1983-1991

- ◆ Some people had expected Unix and C to be a public project forever
- ◆ 1983 - Richard Stallman of MIT starts:
  - The Free Software Foundation
  - The GNU's Not Unix project
  - The GNU General Public License

# The GPL: “copyleft”

◆ The GPL licenses anyone to

- See the source code
- Alter the source code
- Distribute identical or altered versions
  - ◆ But you must release under the GPL

◆ “Viral”

◆ ~80% of open source projects use GPL

# The Internet Era: 1991 - ?

Open source software becomes widely used with the advent of the Internet.

- ◆ Linux, a GPL'ed operating system
  - Finally Scandinavia contributes something
- ◆ The Berkeley Software Distribution
  - BSD License, more academic, essentially no restrictions placed on use

# “Hijacking” Open Source Work

- ◆ BSD license permits proprietary improvements and selling the combo
- ◆ Non-free products based on BSD:
  - Darwin (core of Mac OS X)
  - Early Sun OS
  - Windows TCP/IP stack
- ◆ Is this a bad thing?

# Software Example: Apache

- ◆ A “patchy” version of NCSA HTTPd
- ◆ By August 2004, used on 67% of web servers
- ◆ Included in proprietary packages
  - IBM WebSphere application server
  - Oracle database



# Software Example: Mozilla

- ◆ Initiated by Netscape (later AOL)
- ◆ NPL: like GPL but Netscape controls relicensing
- ◆ Also funded by IBM, Sun, Red Hat
- ◆ All of original code scrapped

# Software Example: Perl

- ◆ Programming language invented to make system administration easier
- ◆ Open specifications vs. open source
- ◆ Larry Wall hired by O'Reilly
- ◆ Difficult to write closed source in Perl

# The Fundamental Mystery



Why do skilled professionals donate their valuable time for nothing, to such an extent that non-profit projects can compete with for-profit projects?

Parallels in other industries?

Role of the internet?

# Motivation: Personal Need

- ◆ Apache started by the Wired website operator when NCSA was unresponsive
- ◆ Perl started when Larry Wall got bored of system administration tasks
- ◆ Sendmail started by a system administrator in charge of email
- ◆ Positive feedback?

# Motivation: Career Prospects

- ◆ Working on open source = signaling?
- ◆ Publicly visible work = clearer signals
- ◆ Does no supervisor mean more precise performance measurement?
- ◆ Like academic information production
- ◆ Linus got Red Hat shares

# Motivation: Complementarity

- ◆ Help sell complementary products
- ◆ Originally, software sold the hardware
- ◆ Makes a promise of freedom credible
- ◆ Keep competitors from Microsofting you
- ◆ IBM wants to sell “middleware”
- ◆ Red Hat wants to sell support contracts
- ◆ Doom, ebay.de

# Motivation: Get the Free Ride

- ◆ Attract a free developer base for something with no direct profit
- ◆ Good for internal-use utilities
- ◆ Netscape now just rebranded Mozilla
- ◆ Red Hat package management tools
- ◆ Corporate “shared source”

# From the User's Point of View



Since open and closed source programs are developed in different ways, there are systematic differences between the two. So what can a user expect from open vs. closed source software?

Unsurprisingly, Microsoft claims open source is a naturally inferior product, and not everyone agrees.



# Usability

- ◆ Open source often targets developers
- ◆ User interface motivation is weaker
  - Developers are already skilled users
  - Not much reputation in writing GUIs
  - It's just a really boring thing to work on
- ◆ Analog to funded usability tests?

# How Much Does It Break

- ◆ "Given enough eyes, all bugs are shallow."
- ◆ Is open source more secure?
  - Windows vs. Linux
  - IE vs. Firefox
  - IIS vs. Apache
  - Closed source and "security by obscurity"

# Network Effects

- ◆ Zero Cost = larger potential market
- ◆ Open source permits forks
- ◆ Online peer support is a network
- ◆ Availability of compatible products
  - Proprietary drivers from vendors
- ◆ Often closed source is first mover

# Release Cycle

## ◆ “Release early, release often”

- More feedback
- More up to date product

## ◆ Deadline based major releases

- Necessary for traditional pricing schemes
- Less quality focused?

# Support

- ◆ No official owner, no official support
- ◆ In theory third parties can sell
  - Support
  - Patent indemnification
  - But this can lose on economy of scope
- ◆ Red Hat, Cygnus, Novell

# Abandonment

- ◆ Often no support means no value
  - Just try to secure a Windows 3.1 machine
- ◆ Vendor lock-in, forced upgrades
- ◆ Open source worst case: DIY
- ◆ Open product can't price out small user
  - Red Hat and Fedora

# Concluding Random Points

- ◆ Socially optimal implies free
- ◆ Market shifting towards services
- ◆ Same model for other products?
  - Wikipedia
  - Creative Commons