Intellectual property law for software

- **principles**
- **patents, copyrights, trademarks, trade secrets**
- **some patent cases**
- **software copyright**
- **GPL, open source licenses**
- **DMCA and some test cases**
- **disclaimer**
  - an enormous topic
  - IANAL

Patents vs copyrights

- **US Constitution, Article 1, Section 8:**
- **"The Congress shall have Power ...**
  To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries;
- **copyright protects expression but not idea**
  - you can't copy my program
  - but you can implement the same idea in some different form
- **patent protects an idea**
  - you can't use my patented idea
  - but you can achieve the same effect in a different way
- **the meaning of "different" is NOT usually clear**

Basic intellectual property mechanisms

- **patent**
  - granted for useful processes, machines, articles of manufacture, and compositions of matter (also design patents and plant patents)
  - provides exclusive rights to make, use, import, sell
  - for 20 years
- **copyright**
  - works of authorship, such as writings, music and works of art that have been tangibly expressed
  - protection for life of author plus 70 years
  - do not have to be registered
- **trademark**
  - words, names, symbols, sounds, or colors that distinguish goods and services
  - can be renewed as long as in use
- **trade secret**
  - non-disclosure and non-compete agreements, etc.
  - not protected against reverse engineering (probably)
  - protected only if it really stays secret

Software & business method patents

- **patents traditionally for devices and processes**
- **patents have commercial value**
  - cross-licensing
  - right to get royalties from others
  - right to exclude others from practicing your invention
  - right to get damages from infringers
- **software patents date from early 1970s**
- **business method patents from late 1990s**

- **much current dispute**
  - are software & business method patents "obvious"?
  - is the search for prior art good enough?
  - should software be patentable?
    in KSR v Teleflex, Supreme Court raises standard of "obviousness" (4/30/07)
  - should business methods be patentable?
    Bilski (10/30/08) makes such patents harder; Supreme Court hasn't ruled yet

- **prediction: more dispute in the future**
  - different answers in different places

Some notable software patents

- **RSA public key encryption (now expired)**
- **Lempel Ziv compression (GIF, gzip)**
- **MP3**
- **FAT file system**
- **reverse auction online**
- **1-click shopping**
- **JPEG (claimed)**
- **XML (claimed)**

Copyright

- **protects expression, not idea**
- **duration used to be 17 years + one renewal**
- **now life + 70, or 95 for commercial works**
  - the "Mickey Mouse Protection Act", 1998
- **fair use permits limited copying under some circumstances**
  - criticism, comment, scholarship, research, news reporting, teaching
- **very uncertain what fair use really is**
  - case by case decisions
- **considerations:**
  - purpose and character of the use
  - nature of the copyrighted work
  - amount and substantiality of the portion used
  - effect of the use on potential market or value of copyrighted work

Software copyright principles

- **how do we compare two programs?**
- **abstraction—filtration—comparison test**
  - pick the right level to capture expression but not idea
  - filter out parts that are not protectable
  - compare what's left for similarity
- **what's not protectable?**
  - unoriginal, no creativity
    tables of numbers?
  - "mere idea, procedure, process, system, method of operation, concept, principle, or discovery"
    because those can't be copyrighted
  - idea and expression have merged
    "When there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression."
  - dictated by external factors
    standards, compatibility, industry demands, widely accepted practices
  - taken from the public domain
- **what does "similar" mean?**

So what does that mean?

- **can you use my header files?**
  - how about a table of #defines?
  - what if I use the same names but different numbers?
- **can you use my interface specification to make a competing product**
  - e.g., a work-alike operating system
- **how much does a program have to be changed before it's different enough?**
- **how do you prove that I copied your program?**
- **how do we explain our positions to lawyers, a judge, and a jury?**
  - none of them understand programming

Free software, open source, GPL, ...

- **free software**
  - "free" as in "free speech", not as in "free beer"
    Richard Stallman, founder of Free Software Foundation
  - freedom to use a program for any purpose, study and adapt, redistribute copies, improve and distribute improvements
  - source code access needed for all of these
- **GNU General Public License (gnu.org)**
  - copyleft: code is copyrighted, plus
  - license requires that any further distribution must use the same license and include source code
  - does not permit non-disclosure, contract or patent restrictions: rights can't be taken away
  - GPL'd code can be sold, any private use is permitted without disclosure
- **open source software**
  - generally more permissive licenses (BSD):
  - proprietary derivatives permitted
- **Creative Commons (creativecommons.org)**
  - copyright licenses for creative works
  - licenses several kinds of subsequent use/copying

Digital Millennium Copyright Act (1998)

- **US copyright law: www.copyright.gov/title17, Chapter 12**
- **anticircumvention: illegal to circumvent a technological measure protecting access to or copying of a copyrighted work**
  - limited exceptions for reverse engineering for interoperability, encryption research, security testing
- **illegal to remove or alter copyright notices and management information**
- **"safe harbor": protects ISPs from copyright infringement claims if they follow notice and takedown procedures**

No circumvention

Chapter 12 - Copyright Protection and Management Systems

§ 1201. Circumvention of copyright protection systems

(a) Violations Regarding Circumvention of Technological Measures.

— (1)(A) No person shall circumvent a technological measure that effectively controls access to a work protected under this title.

(A) to "circumvent a technological measure" means to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair a technological measure, without the authority of the copyright owner; and

(B) a technological measure "effectively controls access to a work" if the measure, in the ordinary course of its operation, requires the application of information, or a process or a treatment, with the authority of the copyright owner, to gain access to the work.

DMCA test cases

- **DeCSS**
  - does publication of code to defeat content scrambling system used to protect DVDs from copying violate DMCA?
- **SDMI**
  - does a demonstration of how to remove digital watermarking on audio files violate DMCA?
- **SunComm**
  - does showing how to defeat CD copy protection by holding down the shift key while inserting a disk violate the DMCA?
- **Lexmark**
  - cryptography used to prevent 3rd parties from supplying replacement ink cartridges; is reverse engineering a violation of DMCA?
- **Diebold**
  - internal emails reveal flaws in voting machine software; is posting of the emails a violation of DMCA?
- **Florida breathalyzer**
  - does a company that writes breathalyzer software have the right to keep a DUI defendant from seeing the code?