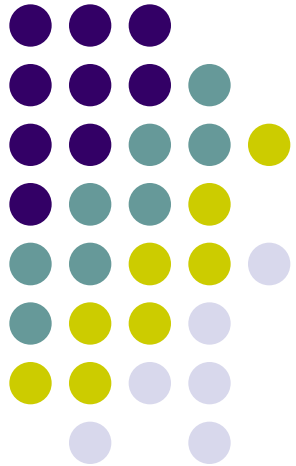
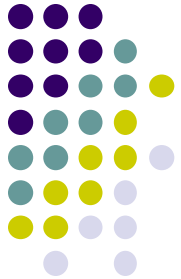
# Legal impacts of OS and Free Software Licensing

Presentation: Dr. Vu Thanh Nguyen
Msc. Nguyễn Công Hoan

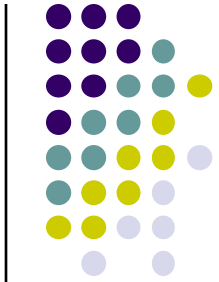


# Content



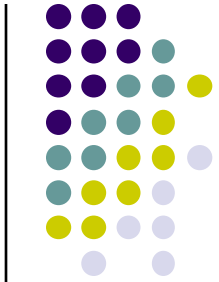
- Entering contracts
- Statutory Developments Related to Software Contracts
- The Self-Enforcing Nature of Open Source and Free Software Licenses
- The Global Scope of Open Source and Free Software Licensing
- The “Negative Effects” of Open Source and Free Software Licensing
- Community Enforcement of Open Source and Free Software Licenses
- Compatible and Incompatible Licensing: Multiple and Cross Licensing

# Introduction



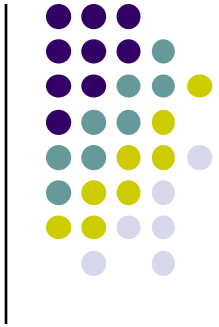
- Two unique problems (in addition to those involved in the enforcement of any contract) affect licensors of software under open source and free software licenses.
  - First, for each license described in previous chapters, the licensor may not even know who the licensees are. These open source and free software licenses do not require notification or other affirmative action to be taken by licensees that would notify the licensor of the fact that the licensee has entered into the contract.
  - Second, while some of these licenses require that the licensee engage in some affirmative action to access the licensed work prior to permitting access of the licensed work, many of them—like the BSD, MIT, and Apache Licenses—do not.

# Introduction



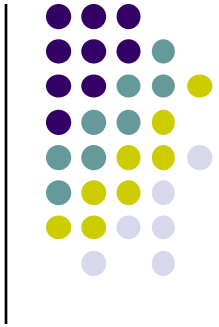
- Both of these problems are substantially addressed by the fact that use of the licensed work is contingent on accepting the terms of the license.

# Entering Contracts



- Any contract between two or more persons rests on two fundamental assumptions: one, that there is some mutual obligation created by the agreement, which is known as the *consideration*; and two, that there is mutual consent, or a meeting of the minds, as to the terms of the contract, usually described as the *offer* and the *acceptance*.
- These concepts are capable of any number of variations and any number of hard cases involing these variations provide the subject matter for first-year law students.
- Basic principles suffice for our purposes.

# Entering Contracts

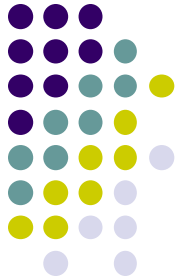


- The idea of consideration turns on the fact that each party is undertaking an obligation, even a very minor one, to the other as part of the transaction.
- Even the most unrestrictive open source license imposes at least a minimal obligation ensuring that consideration in the legal sense is exchanged and an enforceable contract is created through the license.
- Potentially more problematic is the question of mutual consent. In general, mutual consent can be attacked only in relatively unusual circumstances.
- In the classic formulation of a contract, the two parties to the contract have met, negotiated, and reached final agreement, embodied in a formal, signed document.

# Entering Contracts

- While these circumstances appear in numerous variations and can present difficulties in interpreting contracts and adjudicating disputes that arise from them, they are relatively clear cut assaults on the mutual consent to a contract.
- These questions become more difficult when the product and the license both exist in a virtual space and the offer and acceptance both take place there. There are a number of different contexts in which this kind of offer and acceptance can take place, and small differences can be critical in determining whether a contract is formed.
- This issue has obvious application to the open source and free software licenses already discussed. The user takes the code and uses it on his personal computer. The user incorporates the code into a program that he is writing. The user distributes the program, either for profit or not. At no point has the user taken any affirmative, symbolic action that would indicate his consent to the terms of the license that is comparable to the act of signing a contract.

# Statutory Developments Related to Software Contracts

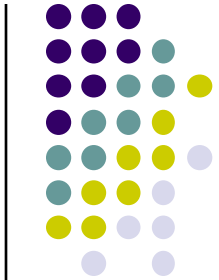


- The Uniform Electronic Transactions Act (UETA), a model law adopted by at least 22 states and under consideration in others, provides as a general matter that a contract may not be denied legal effect simply because the contract is recorded in an electronic medium and not on paper.
- E-Sign, a federal law passed on October 1, 2000, operates to a similar effect, in holding that digital signatures on documents are as effective as ordinary written signatures on paper in memorializing an agreement.
- Another model law, the Uniform Computer Information Transaction Act (UCITA), does modify ordinary state contract laws relating to transactions in software.
- Only two states, Maryland and Virginia, have adopted UCITA; a number of states, however, have adopted anti-UCITA statutes.

# The Self-Enforcing Nature of Open Source and Free Software Licenses

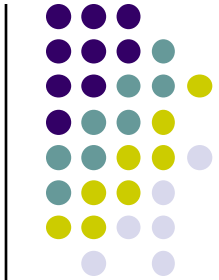
- There is a “savings” logic present in the MIT License (and others) that preserves the effect of the license even in the absence of an affirmative act of consent.
- This is because open source and free software licenses do not impose affirmative obligations on licensees but rather impose restrictions on the rights granted under the license: such restrictions can be relatively straightforward, as is the case with the MIT License’s requirement of reprinting the copyright and permission notice; or somewhat more complex, as with the far-reaching consequences of licensing under the GPL License.
- The GPL License provides a good example of this phenomenon. The typical limitations of proprietary licenses simply do not apply to most applications of GPL-licensed software.

# The Self-Enforcing Nature of Open Source and Free Software Licenses



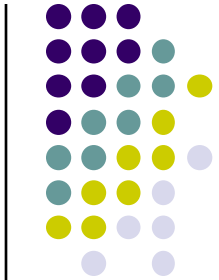
- On the one hand, the user is free to disclaim the obligations of the license, most likely on the grounds that he never affirmatively agreed to be bound by the license. If he does so, he is not obligated to pay royalties or otherwise conform to any affirmative agreements that the license might require.
- A user in such a “default” copyright state is barred from distributing or modifying the work, without the permission of the copyright holder, which permission, by disclaiming the license, he has already refused.
- If, however, the user wishes to exercise rights under the license, he is compelled to accept with it whatever limitations or restrictions may be contained in the applicable software license.
- This feature makes open source and free software licenses remarkably easy to enforce. A licensor can simply tell infringers that infringement vacates their continued rights to the licensed code.

# The Self-Enforcing Nature of Open Source and Free Software Licenses



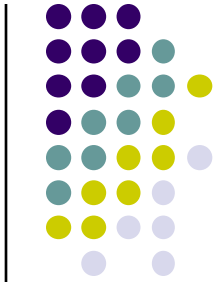
- In addition, because the customer is aware of the difficulties and expense associated with relying on software licensed under what is, at best, a highly questionable license, it is probably sufficient to convince such customers to abandon the use of the work distributed in violation of the license.
- The Free Software Foundation has policed the GPL License in exactly this fashion for many years with consistent success.

# The Global Scope of Open Source and Free Software Licensing



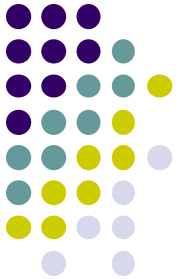
- Another issue for open source and free software licenses is their enforcement in jurisdictions outside the United States. The global nature of commerce and the generally free travel of software across national boundaries implicates the enforcement of open source and free software licenses in a number of jurisdictions, not only those in the United States.
- International enforcement of copyright laws is frequently lax. While many countries are signatories of treaties that provide for the international enforcement of copyright protection (such as the Berne Convention), such treaties are frequently disregarded.
- In many countries, particularly in the “developed” world where most software creation takes place, the enforcement of copyright is routine.

# The Global Scope of Open Source and Free Software Licensing



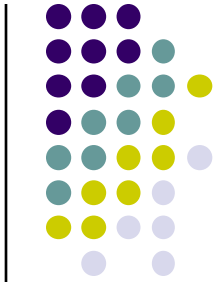
- While the unauthorized distribution of copyrighted material is commonplace, it is nonetheless difficult for any established company or person to reasonably hope to profit from the illegal distribution of copyrighted material.
- This is particularly true of software. Users of software, at least commercial users, are generally more concerned with reliable performance and support than with the incremental cost of software. Users expect to be able to rely on a software maker's products and to receive support for that software's application going forward.
- Providing this reliability and these services requires the existence of a stable, aboveground organization—exactly the kind of organization that is subject to suit and accordingly to the legal enforcement of copyright law.
- The question thus becomes whether open source and free software licenses can reasonably be expected to be enforced, as a legal matter, outside the United States.

# The Global Scope of Open Source and Free Software Licensing



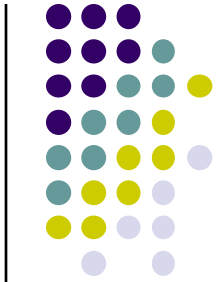
- The answer to this question is a slightly qualified yes.
- Many countries are signatories to the Berne Convention, which provides for copyright protection more stringent in many respects than that provided by United States copyright law. Moreover, as has been the case with the enforcement of proprietary licenses, the existence of some amount of "pirating" or distribution outside the boundaries of a given license, such as with the unauthorized distribution of music, is not fatal to the successful distribution of the licensed work.
- "Pirating" work generally means nothing more than the distribution of the work itself without the payment of royalties to the creator of the work. "Pirating," in this sense, thus does not violate the restrictions applicable to most open source and free software licenses, which generally do not limit the free (i.e., without charge) distribution of unmodified versions.

# The Global Scope of Open Source and Free Software Licensing



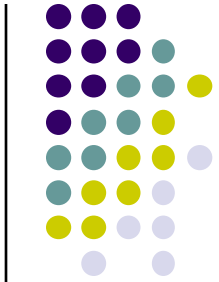
- “Pirating” in this sense is also limited by the fact that the major markets in which software or any other kind of work can be sold at a profit are subject to legal constraints and the enforcement of law.
- While a consumer may be willing to take a chance on a five dollar bootleg CD or DVD that she intends to use just for her personal entertainment, such a consumer is much less likely to take such a chance on software, the stability and functionality of which she really must rely on.
- These dynamics probably explain the relatively small amount of litigation spawned by open source and free software licenses. While these licenses certainly can be infringed upon, market forces and social dynamics tend to limit the extent of such infringement, even in the absence of vigorous legal enforcement of the license by the licensor.

# The “Negative Effects” of Open Source and Free Software Licensing



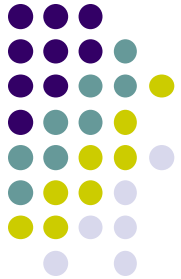
- Another effect of open source and free software licensing that has already been touched upon is the obstacle that violations of applicable licenses create for the violator of that license. Such violators will find that their own ability to enforce copyrights that arise out of or are related to infringements of the terms of an open source or free software license is seriously compromised.
- Violations of such licenses put the violators at risk of surrendering the benefits of any actual, copyrightable work that they may have invested in modifying or improving a licensed program.
- Taking again one of the least restrictive examples of open source licenses as an example, it becomes apparent that violation of its terms undermines any future copyright enforcement relating to the modified work (take an example for discussion).

# Community Enforcement of Open Source and Free Software Licenses



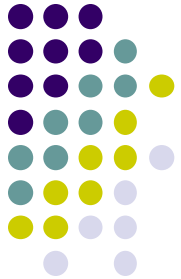
- The open source and free software communities are also critical to the practical enforcement of open source and free software licenses. While the discussion so far has focused on the legal and practical reasons why open source and free software licenses tend to be complied with, there is a more fundamental reason why most programmers comply with such licenses. Non-compliance, or at least knowing noncompliance with the terms of these licenses, is simply wrong.
- **The world of open source and free software licensing is still a relatively small one**.
- The principle is deeply felt by this community. The gross violation of it by taking someone else's work and distributing it as one's own is unthinkable. This moral principle is, by itself, responsible for the largest part for the enforcement of open source and free software licenses, not the texts of the licenses themselves, and not the courts that enforce those licenses.

# Community Enforcement of Open Source and Free Software Licenses



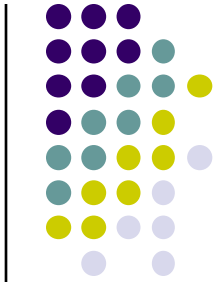
- Even those who have not internalized this principle have good reason to abide by the norms of this community. Violating those norms will incur, at the least, the displeasure of this community.
- ***In sum, while contracts and courts are fundamental to protecting the principles of open source and free software licensing, the real guardians of these principles are programmers (and users) themselves.***

# Compatible and Incompatible Licensing: Multiple and Cross Licensing



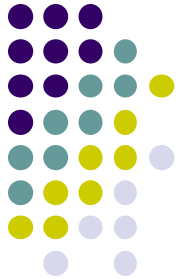
- In writing code, a programmer may find that he wants to fuse elements from two or more programs into a new program. The two programs are under different licenses. The question arises: is it possible to take this code, under different licenses, and combine them in one work without violating the terms of either of the two licenses?
- It is much easier to describe those licenses that are incompatible than to assert with any assurance that two licenses are compatible. There are several scenarios in which the answers are obvious. If either one of the works is licensed under a proprietary license, the code cannot be combined with work under another license.
- As a general matter, under a classic proprietary license, the user has no rights to the work other than to use a single copy of it.
- Another example susceptible to quick analysis is the GPL License. GPL-licensed code is incompatible with code licensed under most licenses.

# Compatible and Incompatible Licensing: Multiple and Cross Licensing



- In general, the “research style” licenses described previous chapter (as MIT, BSD, Apache licenses…) are compatible with each other.
- Beyond these general observations, it is difficult, if not impossible, to provide precise guidance about what licenses may or may not be compatible with each other.
- **Programmers who are considering combining code governed by two or more different licenses should proceed cautiously.**
- As the creator of a work, the original licensor retains all of the rights associated with that copyright, subject only to the sale or licensing of those rights to others.
- Accordingly, such a licensor retains the power and the discretion to license his work under terms other than those contained in the original license.
- Some open source groups will not cross-license works copyrighted by them. The Apache Software Foundation, for example, does not cross-license its works.

# Compatible and Incompatible Licensing: Multiple and Cross Licensing



- Linux is licensed under the GPL and includes the works of thousands of people who made contributions to the project with the belief (assuming that they took the time to develop one) that the resulting work would be licensed under the GPL.
- Because of the rigidity of the monopoly granted by copyright laws, each one of those contributors could argue, legally, that their contribution can only be used in ways consistent with the terms upon which they agreed to participate in the project.
- Cross-licensing such projects, while not impossible, is impractical in all but the most unusual situations.
- Most open source and free software projects, however, do not present such logistical difficulties. They are maintained by a small number of people, frequently just one person, whose permission to distribute that work under another license can often be gained for no more than the cost of asking.