

Open Source

권순선 <kss@kldp.org>

Open Source License

- Open Source의 개념 소개
 - 오픈소스 활용시 의무사항: Attribution Notice, 소스코드 공개 등
 - 라이선스별로 의무사항의 정확한 범위는 서로 다름
- Open Source와 지적 자산(Intellectual Property)의 관계
 - Source Code: 저작권(Copyright)
 - Method/Algorithm: 특허권(Patent)
- 지적 자산 보호 방안
 - Source Code 공개 조건, 특허권 공개 조건 설명
 - 라이선스별로 조건이 다르고, 판단이 어려운 부분이 많음
 - 따라서 활용시 오픈소스와 소프트웨어 양쪽의 지식을 모두 갖춘 전문가의 확인 필요
- 오픈소스 적용 예시

소프트웨어와 Intellectual Property

내가 가진 저작권과 특허권을 보호
타인의 저작권과 특허권 침해 방지

- 저작권(Copyright)
 - 표현된 결과물: 소스코드
- 특허권(Patent Right)
 - Method: 아이디어
- 상표권(Trademark)
- 기타: 영업비밀(Trade Secret)
- 오픈소스 활용시 IP 이슈는 저작권과 특허권으로 나누어서 생각하여야 함
 - 소스코드 보호: 저작권
 - 아이디어/알고리즘 보호: 특허권



The Definition of Open Source

- Licensee는 해당 오픈소스 소프트웨어를 어떠한 용도로도 사용할 수 있다.
- Licensee는 해당 오픈소스 소프트웨어를 Licensor에게 별도의 로열티를 제공하지 않고도 자유롭게 복사본을 만들고 재배포할 수 있다.
- Licensee는 Licensor에게 별도의 로열티를 제공하지 않고도 해당 오픈소스 소프트웨어의 파생물을 자유롭게 만들고 재배포할 수 있다.
- Licensee는 해당 오픈소스 소프트웨어의 소스코드를 자유롭게 획득하고 접근할 수 있다.
- Licensee는 해당 오픈소스 소프트웨어를 다른 소프트웨어와 결합할 수 있다.



라이선스 준수 사항 Summary

● 라이선스별 준수사항

- 해당 오픈소스 소스코드 공개: Reciprocal License (GPL, MPL 등)의 경우 사용한 오픈소스의 소스코드를 공개하여야 함
 - GPL/LGPL/MPL 등
- 해당 오픈소스 사용여부 명시: 해당 오픈소스를 사용하였음을 배포시에 명시(소비자에게 고지)
 - GPL/LGPL/OpenSSL
- Patent 포함될 경우 확인: 라이선스에 정의된 Patent Termination 조항을 준수하여야 함
 - GPL/LGPL/Apache2.0/MPL/Apache1.0: 수정된 코드에 당사 Patent나 타사 Patent가 포함될 경우 별도 확인 필요



● 공통 준수사항

- Trademark
 - All: 제품 이름에 해당 오픈소스 소프트웨어의 이름을 사용하면 안됨
- 저작권자 표기: 해당 오픈소스 소스코드에 포함된 Copyright Notice는 절대로 임의 수정 불가!
 - All: 소스코드에 표기된 라이선스 관련 주석을 지우면 안됨

소스코드의 저작권 보호 방안 (1/2)

오픈소스 사용시 자체개발 코드를 보호하려면
자체개발 코드가 특정 오픈소스 코드의 파생물이 되지 않게 하여야 함

- 소프트웨어에서 파생물(Derivative Work)의 정의
 - Work based on that program
 - 특정 소스코드를 수정하거나 일부를 차용하면 그 결과물은 파생물로 볼 수 있음

- 파생물과 결합저작물(Combined Work)
 - 소프트웨어에서의 결합저작물은 두 개 이상의 프로그램을 결합한 것을 의미하며 이 둘은 서로의 파생물이 아니어야 함

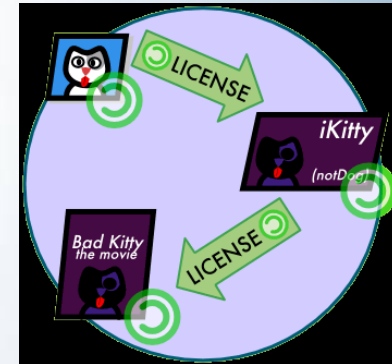


This example is likely a derivative

소스코드의 저작권 보호 방안 (2/2)

오픈소스별로 파생물의 정의가 모두 다르고
일관적이지 않으므로 사전에 확인 필요(특히 GPL)

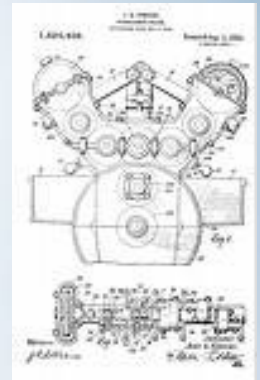
- “Reciprocal(호혜적)” 라이선스에서는 파생물도 오픈소스가 되도록 규정 (예: GPL)
 - BSD License 등 Non-Reciprocal 라이선스는 해당 없음
 - Reciprocal 라이선스는 GPL 이외에도 MPL, EPL 등이 있음
- 라이선스별로 파생물의 정의가 모두 다름
 - GPL, LGPL: 링크 방법 등 모호하고 포괄적으로 정의되어 있음
 - MPL, EPL: 원 소스에 내용을 추가하거나 원 소스를 차용한 경우에만 파생물로 정의
- 저작자(개발자)의 의견도 중요
 - 예: 오픈소스 커뮤니티에서도 GPL에 대해 의견차가 있음



아이디어/알고리즘의 특허권 보호 방안

오픈소스를 사용하면서 특허를 출원하거나,
특허로 보호되는 기술을 오픈소스에 추가/수정할 경우 사전 확인 필요

- 특허 분쟁에 대해 MPL 등 일부 오픈소스 라이선스에서는 명시적으로 관계/권한을 정의하고 있으나 대부분은 모호함
- 오픈소스 코드 적용시 만약 자체 보유 특허가 구현된다면 해당 특허의 중요성과 해당 오픈소스에 대한 라이선스를 고려하여 특허 구현 여부 결정 필요
- 오픈소스 라이선스별 특허 정의 사례
 - GPL: 묵시적 인정, 특허권을 확보하지 못하면 SW 배포 못한다고 명시
 - Apache: 개발자에게 해당 오픈소스에 대해 특허 침해 소송 제기하면 해당 오픈소스 사용 못함
 - MPL, APSL, NOKOS: 개발자에게 어떤 식으로든 특허 침해 소송 제기하면 해당 오픈소스 사용 못함
 - 경쟁사가 개발한 오픈소스를 사용할 경우 주의 필요



사례: SW별 사례

SQLite

- Public domain이므로 아무런 의무사항 없이 자유롭게 사용 가능



FreeType

- Dual License(Freetype License or GPL): FreeType License 채택

- Attribution Notice만 지키면 됨

- Patent 이슈 있음

- Bytecode Interpreter를 Enable하여 사용하려면 Apple에 해당 특허에 대한 로열티 지불하여야 함



Glib

- LGPL 라이선스

- Glib 자체를 수정할 경우 소스코드 공개하여야 함



DPKG

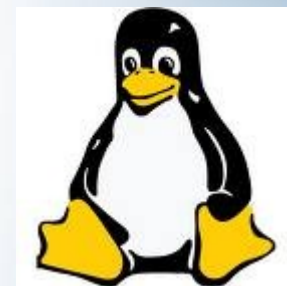
- GPL 라이선스

- 수정시 수정한 부분도 GPL로 릴리즈하여야 함

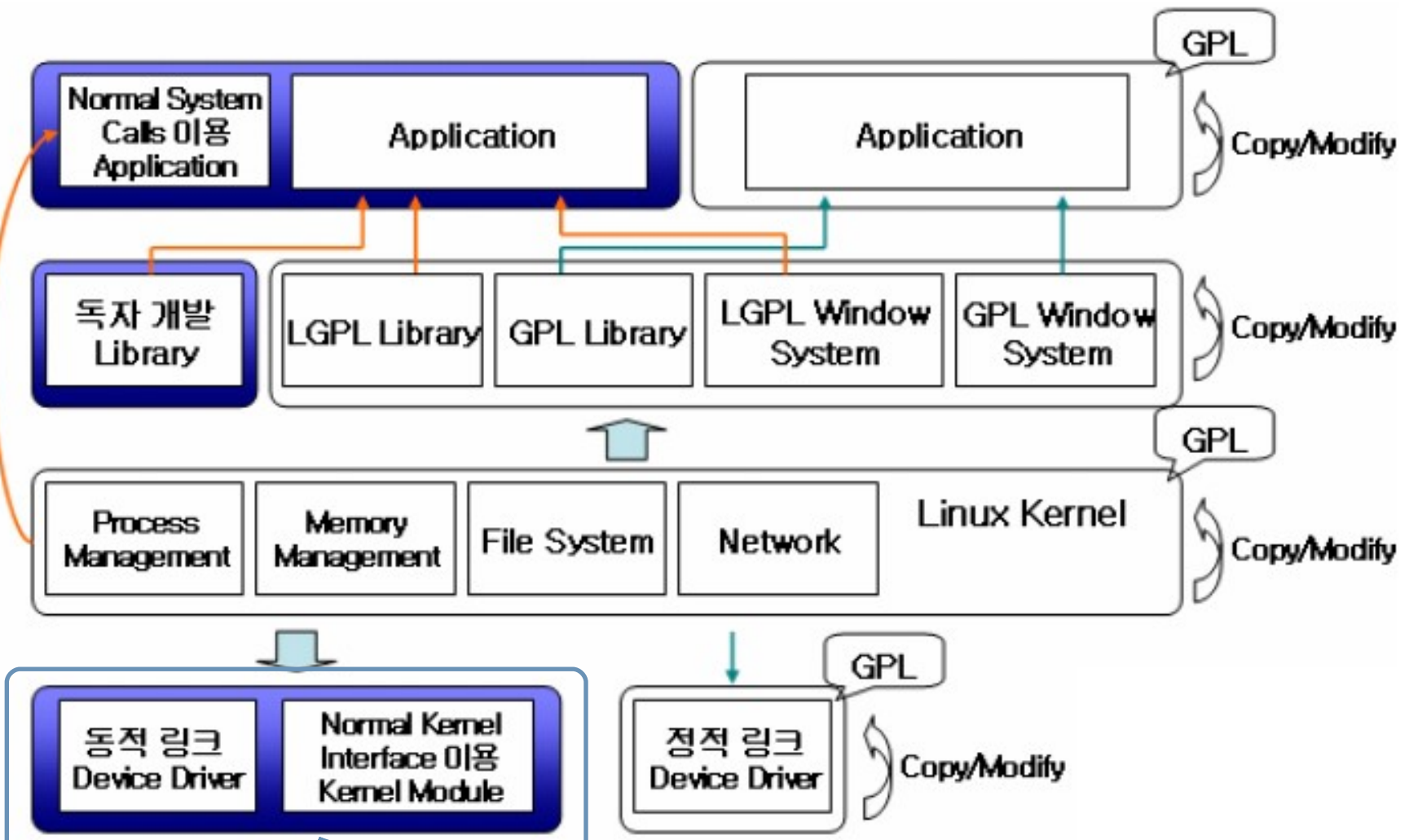


사례: Linux Kernel Module의 GPL 이슈

- Kernel Module은 개발하기에 따라 GPL이 아닐 수도 있음
 - Kernel Module로 뻥다고 해서 ‘무조건’ proprietary는 아님
 - 모듈 인터페이스를 임의로 수정하면 안됨
 - 예: EXPORT_SYMBOL()을 EXPORT_SYMBOL_GPL()로 변경 (X)
 - 즉, 해당 모듈은 <http://kernel.org> 에서 제공하는 커널 모듈 인터페이스만을 가지고도 동작할 수 있어야 함
 - Linus Torvalds의 의견이 우선시됨
 - FSF 및 Richard Stallman은 Linux Kernel의 개발자가 아니므로 크게 관련이 없음



GPL/LGPL Coverage 요약



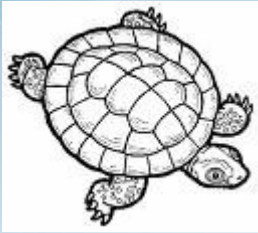
커널 모듈도 GPL? Yes.

Open Source 의 경제적 가치 (Linux Kernel의 사 례)

Linux의 가격은? - Kernel 2.6 기준

- \$176M? (1700억)
 - Total Physical Source Lines of Code (SLOC) = 4,287,449
 - Development Effort Estimate, Person-Years (Person-Months) = 1,302.68 (15,632) (Basic COCOMO model, Person-Months = $2.4 * (KSLOC**1.05)$)
 - Schedule Estimate, Years (Months) = 8.17 (98.10) (Basic COCOMO model, Months = $2.5 * (person-months**0.38)$)
 - Estimated Average Number of Developers (Effort/Schedule) = 159.35
 - Total Estimated Cost to Develop = \$ 175,974,824 (average salary = \$56,286/year, overhead = 2.40).
- OS Kernel의 개발 난이도와 고급 엔지니어 고용 비용을 고려하면 Linux의 개발 비용은 \$612M (6100억)에 달함
 - <http://www.dwheeler.com/essays/linux-kernel-cost.html> (2004)
- Linux Kernel 이외의 System Library, Compiler, Linker/Loader는 제외한 것이므로 실제 비용은 이보다 훨씬 더 큼

Open vs. Closed



Internal

- small group
of developers

- 커뮤니티와 연계 불가능
(자체 패치, backport...etc)
- 문제 자체 해결해야...



Linux Kernel Community

- 2074 committers (tester/user
는 제외한 수치임)
- 766,000 lines were added for
the past 1 year (2006)
- 9,200 lines are changed daily

- 매일 3억, 연간 1,200억 이상의 개발비
투입과 동일
- Library, Compiler 등은 포함하지 않았으
며 Test & Bug Report 비용 역시 포함되지
않은 금액임

Open Source Community를 Leverage하는 것만이 유일한 해결책!

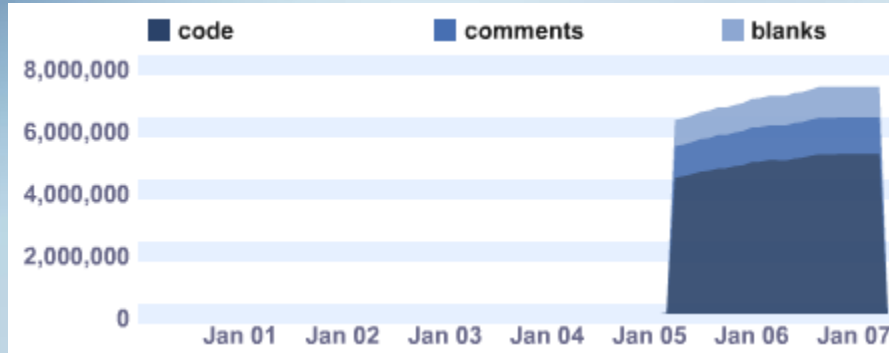
Open Source evolution examples (ohloh.net)

Open Source의 개발 속도와 안정화 사이의 균형 필요

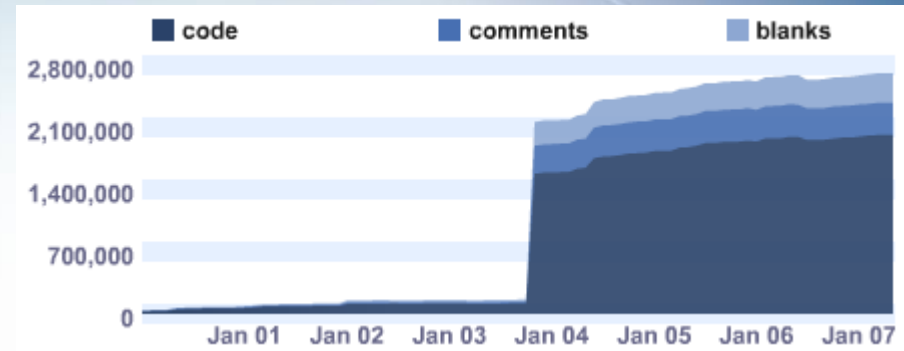
	Total Contributors	Contributors for 1 years	LoC	Person of Years
Kernel 2.6	2,402	1,347	5,137,222	1,575
X.org	286	189	2,015,262	586
Gcc	342	152	4,475,284	1,356
FireFox	148	85	371,869	97

Open Source evolution examples (ohloh.net)

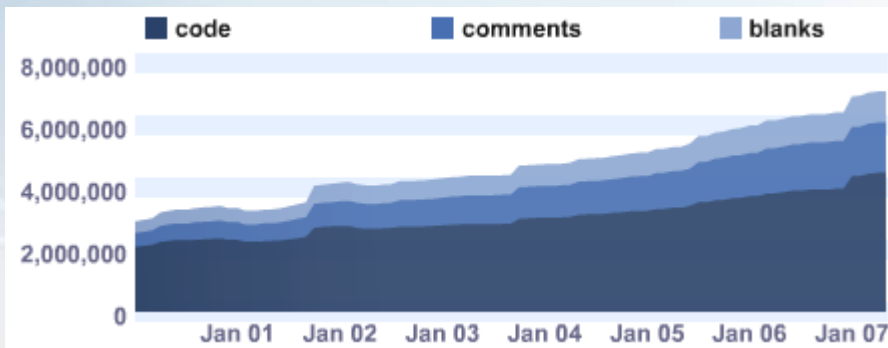
Linux Kernel 2.6



X.org



gcc



FireFox

