

Open-source software (OSS) 라이선스

세종대학교 이은상

라이선스

- 라이선스

- SW는 지식재산권에 의해 보호받으며 저작권자만이 쓸 수 있지만, 권리가 다른 사람에게 일정 조건으로 특정 행위를 할 수 있는 권한을 부여할 수 있음
- 이 권한을 보통 라이선스(license)라고 함
- 예시) Windows
 - Windows를 구입하면 SW 권리자인 Microsoft로부터 Windows 10을 한 대의 컴퓨터에 설치하여 이용할 수 있는 라이선스(권리)를 받은 것에 불과함
 - Windows 정품을 구입했다고 해서 다른 사람에게 빌려주거나 복제하여 팔 수 없음

오픈소스 라이선스

- 오픈소스 라이선스

- 오픈소스는 소스코드가 공개되어 **사용, 복제, 배포, 수정**할 수 있지만 원 저작자의 라이선스 규칙에 따라야함
- 오픈소스 라이선스란 오픈소스 개발자와 이용자 간의 사용 방법 및 조건의 범위를 명시한 계약임. 오픈소스 저작권자가 소스코드를 공개했을 뿐 지적재산권을 보호받고 있으며, 라이선스 종류에 따라 다양한 의무사항을 포함함
- 오픈소스를 이용하려면 오픈소스 개발자가 만든 조건의 범위에 따라 해당 SW를 사용해야하며, 이를 위반할 경우 라이선스 위반 및 저작권 침해로 법적 책임을 져야함

오픈소스 라이선스

- 오픈소스 라이선스 사용 분포
 - OSI에 따르면 현재 80개의 라이선스가 있지만 실제 많이 사용되는 라이선스 개수는 한정되어 있음.
 - 오픈소스 프로젝트 개발 포털사이트 Freshmeat에 등록된 프로젝트 약 43,722개 중 약 72%가 GPL과 LGPL 라이선스임

OSS 라이선스 분포도		
License	Branches	Percentage
GNU GENERAL PUBLIC LICENSE (GPL)	30,299	64.70%
GNU LESSER GENERAL PUBLIC LICENSE (LGPL)	3,067	6.55%
BSD LICENSE (ORIGINAL)	1,366	2.92%
BSD LICENSE (REVISED)	1,354	2.89%
FREWARE	1,079	2.30%
FREELY DISTRIBUTABLE	980	2.09%

오픈소스 라이선스 상세

- 공통적 준수사항(의무사항)
 - 저작권 관련 문구 유지
 - 저작권은 저작물의 창작과 함께 자동적으로 부여됨. SW의 경우 소스코드에 프로그램의 이름과 개발자, 버전, 연락처 등을 포함하는 경우가 많고 이들은 저작권격권으로 보호받음
 - 오픈소스는 거의 대부분 소스코드 상단에 개발자 정보와 연락처 등이 기록되어있으며 개발자 정보를 임의로 수정하거나 삭제해서는 안됨
 - 제품명 중복 방지
 - SW의 제품명은 상표권으로 보호받음. 오픈소스의 경우에 이와 동일한 이름을 제품명이나 서비스명으로 사용하면 상표권 침해에 문제가 생기게 됨. 특히 유명한 오픈소스일수록 해당 오픈소스의 이름이 상표로 등록되어 있는 경우가 많기 때문에(ex 리눅스) 더욱 조심해야함

오픈소스 라이선스 상세

- 공통적 준수사항(의무사항)
 - 서로 다른 라이선스의 SW 조합 시 조합 가능 여부 확인
 - SW를 작성하고자 할 경우 기존에 만들어진 코드를 재사용하거나 결합하는 경우 많음. 이 때 결합되는 각 코드의 라이선스가 서로 상충되는 경우가 있음 (라이선스의 compatibility 문제)
 - 서로 다른 라이선스로 배포된 오픈소스를 결합하는 경우 반드시 2개의 라이선스가 서로 호환되는지를 확인해야함

오픈소스 라이선스 상세

- 선택적 준수 사항(라이선스에 따라 다름)
 - 이용 여부 명
 - 많은 오픈소스 라이선스들은 SW를 이용할 때 해당 오픈소스가 이용되었음을 명시하도록 함. 즉, "이 SW는 오픈소스인 무엇 무엇을 이용하였습니다." 라는 식으로 이용 여부를 명확히 기술하라고 함
 - 소스코드 공개
 - 오픈소스는 라이선스에 따라 수정하거나 추가한 부분이 있을 때 해당 부분이 소스코드로 공개하여야 한다고 명시하는 경우가 있음 (GPL)
 - 정확한 공개 범위는 각 라이선스에서 정하고 있는 범위와 SW를 개발하는 방법에 따라 달라질 수 있음
 - 특허
 - 어떤 기술이 특허로 보호될 경우 해당 기술을 구현할 때 반드시 특허권자의 허락을 받아야함. 이 조건은 오픈소스 여부와 상관없음
 - 특히 최근 SW 특허가 급격히 증가하면서 새롭게 만들어지는 오픈소스 라이선스들에서는 특허 관련 조항을 포함하는 경우가 많음

대표적인 라이선스 (GPL 2.0)

- GPL 2.0에서 GPL은 GNU general public license의 줄임말
- 자유 소프트웨어 재단(FSF)에서 만든 라이선스로서 가장 강력한 라이선스임
- 리처드 스톨만이 GNU 프로젝트로 배포된 프로그램의 라이선스로 사용하기 위해 작성함

```
Copyright (C) yyyy name of author
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation; either version 2  
of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```


대표적인 라이선스 (GPL 2.0)

- 소스 코드 형태로 재배포하는 경우
 - GPL 2.0 오픈소스를 소스 형태로 재배포시 다음 사항을 준수해야함
 - 먼저 고지의 의무가 있음. 저작권 고지 제공, 보증 부인 제공, 라이선스 사본 제공
 - 즉, 소스 코드 내 명시된 저작권/라이선스 정보를 그대로 유지한 상태로 재배포함
- 소스 코드 일부를 추가/수정하고 재배포하는 경우
 - 추가/수정한 부분에 GPL 2.0을 적용함
 - 수정 사항에 대한 고지를 포함함 (예: 수정일, 수정내용을 주석 형태로 포함)

대표적인 라이선스 (GPL 2.0)

- 바이너리 형태로 재배포하는 경우
 - 고지의 의무
 - GPL 2.0 오픈소스를 빌드하여 바이너리 형태로만 재배포시 다음 사항을 준수해야함
 - 먼저 고지의 의무가 있음. 저작권 고지 제공, 보증 부인 제공, 라이선스 사본 제공. 이들 내용을 포함하는 오픈소스 고지문을 생성하여 바이너리 재배포시 동봉해야함

대표적인 라이선스 (GPL 2.0)

- 바이너리 형태로 재배포하는 경우
 - 소스 코드 제공
 - 바이너리에 해당하는 소스 코드를 제공해야하며 다음을 준수해야함
 - GPL 2.0은 파생저작물에 대해서는 GPL 2.0을 적용하며 소스코드를 공개할 것을 요구함.
 - GPL 2.0 파생 저작물 범위
 - 수정 코드
 - GPL 프로그램과 동일한 프로세스에서 동작하는 Module
 - GPL 프로그램과 링크로 연결한 Library
 - GPL 프로그램을 상속한 Class
 - GPL 2.0 파생 저작물 범위에 속하지 않는 것
 - CD와 같은 매체에 함께 존재하지만 GPL 프로그램과 전혀 연동되지 않는 독립 프로그램
 - GPL 프로그램과는 별도의 프로그램으로써 Pipe, Socket, IPC, Command Line Arguments로 GPL 프로그램과 통신하는 경우

대표적인 라이선스 (GPL 2.0)

- 바이너리 형태로 재배포하는 경우
 - 빌드 환경 제공
 - 바이너리 사용자가 공개된 소스코드로 동일한 바이너리를 만들 수 있는 빌드 환경을 제공함
 - tool chain 정보, 빌드 스크립트, 빌드 방법 (README)
 - 서면 약정서 (소스 코드 대신)
 - 소스 코드 대신 서면 약정서 (Written offer)를 제공할 수 있으며 여기에는 아래 진술이 포함되어있음
 - 서면 약정서는 제품 판매 후 3년간 유효함
 - 누구에게나 제공함
 - 비용 청구를 하지 않음
 - 이후 외부로부터 서면 약정서를 근거로 소스 코드 제공을 요청받을 경우, 바이너리에 해당하는 소스 코드를 제공해야함
 - 따라서 회사는 제품 판매 후 최소 3년간 소스 코드를 보관해야함

대표적인 라이선스 (GPL 2.0)

- 라이선스 호환성
 - 서로 요구하는 의무사항이 상충되는 오픈소스 라이선스는 하나의 프로그램에 동시에 존재해서는 안 됨
 - 다음은 GPL 2.0과 충돌하는 라이선스 목록임
 - Apache-1.1, Apache-2.0, BSE-4-Clause, FTL, IJG, OpenSSL, Python-2.0
- GPL 2.0 적용 예
 - 리눅스 커널, Git, 마리아 DB, 워드프레스, 파이어폭스(v2.0) 등

대표적인 라이선스 (GPL 3.0)

- FSF에서 2007년 공개한 라이선스로 GPL 2.0과 유사한 의무사항을 갖지만, 추가로 User product 배포 시 설치 정보 제공을 요구함
 - 바이너리를 User product와 배포한다면 설치 정보 (installation information)을 제공해야함
 - User product: 전자 기기와 같은 Embedded device
 - 설치 정보(installation information): 사용자가 소스 코드를 빌드 하여 다시 제품에 설치하기 위해 필요한 모든 정보 및 방법
- *** 대부분의 user product는 보안상의 이유로 설치 정보를 제공하는 것이 불가능함. 따라서, User product로 배포하는 SW에는 GPL 3.0의 오픈소스를 사용하지 않아야함

대표적인 라이선스 (AGPL)

- Affero GPL의 줄임말
- GPL을 기반으로 만든 라이선스이며 버전 1과 2는 Affero, 버전 3은 FSF에서 만듦
- 어떤 개발자가 프로그램의 소스코드를 수정해서 사용하고 프로그램을 배포하지 않았을 때 발생할 수 있는 소스코드의 공유 불가 현상을 해결하기 위해 만들어짐
- 서버에서 프로그램을 실행해서 다른 사용자와 통신하게 되면 실행되고 있는 프로그램의 소스코드를 사용자들이 다운로드할 수 있도록 해야하는 조항을 포함함
- AGPL 라이선스 적용 예: 몽고DB 등

대표적인 라이선스 (LGPL)

- Lesser GPL의 줄임말
- GPL이 붙은 라이선스를 사용할 때 반드시 소스코드를 다시 GPL로 공개해야하는 부담 때문에 실무에서 사용이 어려운 점을 보완하기 위해 만들어진 라이선스
- LGPL을 따르는 프로그램은 전체 소스코드를 공개하지 않고 사용된 오픈소스 라이브러리에 대한 소스코드만 공개하면 됨
- 초기에는 한정된 라이브러리에만 적용된다는 것을 나타내기 위해 Library GPL이라 불렀으나 이 용어가 모든 라이브러리에 해당된다는 오해를 만들기 쉬워 이후 Lesser GPL로 명칭이 변경됨
- LGPL 라이선스 적용 예시: Firefox(v2.1) 등

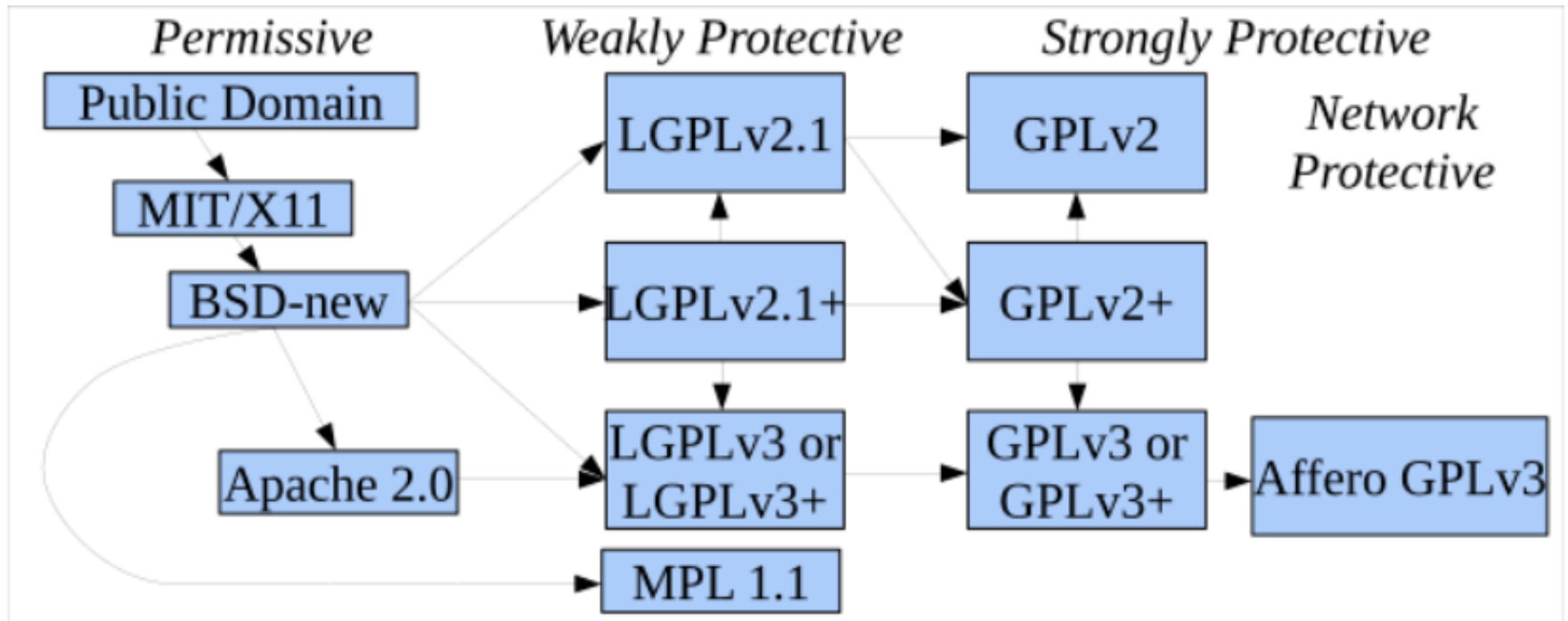
대표적인 라이선스 (MIT)

- MIT에서 학생들을 지원하기 위해 만든 라이선스임
- 라이선스와 저작권 관련 명시만 지켜주면 되는 등 가장 느슨한 조건을 가지고 있어 많은 사람들이 사용하기 용이하다는 특징이 있음

주요 오픈소스 라이선스 비교

구분	무료 이용가능	배포 허용가능	소스코드 취득가능	소스코드 수정가능	2차적 저작물 재공개 의무	독점SW와 결합가능
GPL	●	●	●	●	●	X
LGPL	●	●	●	●	●	●
MPL	●	●	●	●	●	●
BSD license	●	●	●	●	X	●
Apache lilcense	●	●	●	●	X	●

주요 오픈소스 라이선스 비교



위키백과 오픈소스 라이선스 관련 이미지

라이선스 양립성 문제

- 둘 이상의 오픈소스 소스코드를 사용하여 새로운 프로그램을 개발할 경우 각 오픈소스의 라이선스 요구사항이 서로 상충하는지의 문제
- 새 프로그램에서 각 라이선스 요구사항을 모두 충족시키지 못하는 경우 이런 문제 발생함
- 둘 이상의 오픈소스를 결합할 때에는 각 라이선스가 서로 양립 가능한지 미리 조사해야함

라이선스 양립성 문제

- 예시

- 각각 GPL, MPL로 배포되는 오픈소스의 소스코드를 함께 결합하여 사용하는 경우 고려
- MPL은 MPL로 배포된 오픈소스에 해당하는 소스코드로부터 파생된 부분을 동일한 MPL로 배포하기를 요구하고, GPL은 GPL로 배포된 오픈소스에 해당하는 소스코드로부터 파생된 부분 뿐만 아니라, 그와 링크된 전체의 소스코드를 GPL로 배포하기를 요구하기 때문에 이 두 라이선스 요구를 동시에 충족시키는 것은 불가능함

라이선스 양립성 문제

- 특히 GPL과 관련하여 양립성 문제가 많이 발생하는 이유는, GPL은 사용자들에게 GPL에서 규정하는 것 이외의 제한사항을 추가하지 못하도록 엄격하게 통제하고 있기 때문이다
- GPL을 포함하여 FSF가 만든 라이선스 간 양립성에 대해서는 GNU 사이트에서 확인할 수 있음
 - <https://www.gnu.org/licenses/gpl-faq.html#AllCompatibility>

오픈소스 검증 도구

- 개념

- 오픈소스 검증 도구는 어떤 프로그램을 개발했을 때 그 프로그램에서 어떤 오픈소스를 썼는지, 어떤 라이선스 및 저작권이 사용되었는지를 파악해주는 도구임

- Code Eye

- 한국저작권위원회가 개발한 오픈소스 라이선스를 비교, 분석, 검사하는 서비스
- 기업들이 보유한 SW소스코드에 오픈소스 포함여부를 알려주고, 사용자들이 자신의 소스코드 파일이나 폴더에 대한 검사요청 시, 저작권위원회에서는 저작권위원회의 서버에 저장된 데이터베이스와 비교, 검사보고서 제공

오픈소스 검증 도구

- Fossology
 - 오픈소스로 공개된 소스코드 스캔도구
 - <https://www.fossology.org/>
 - 프로젝트에 어떤 라이선스 및 저작권이 사용되었는지 스캔함
 - 웹 UI를 통해 워크플로우 확인
- www.oss.kr
 - 오픈소스 라이선스 검증 서비스 신청
 - 프로젝트 별 소스코드 500MB 이하 단위로 신청할 수 있으며 총 3회까지 신청 가능