Open source Software
오픈소스 소프트웨어
19. 오픈소스 소프트웨어 개요

학습 개요

1. 자유 소프트웨어와 오픈소스 소프트웨어
2. 오픈소스 소프트웨어 개발 지원 서버
3. 오프소스 소프트웨어 저작권

학습 목표

1. 오픈소스 소프트웨어의 의미를 이해할 수 있다.
2. 오픈소스 소프트웨어 개발을 지원하는 서버인깃허브에서의 협업 방식을 이해할 수 있다.
3. 대표적인 오픈소스 소프트웨어를 알 수 있다.
4. 오픈소스 소프트웨어의 저작권을 이해할 수 있다.

LESSON 01

오픈소스 소프트웨어 개요

1 오픈소스 소프트웨어 개요

오픈소스 소프트웨어

OSS: open source software 오픈소스

- 누구나 특별한 제한 없이 그 코드를 보고 사용할 수 있는 오픈소스 라이선스를 만족하는 SW
  - 소스코드를 공개, 공개 소스의 복제도 가능
  - 통상 간략하게 오픈소스, OSS라 칭함

Open Source Initiative(OSI)

- opensource.org
- 공개 소스 정의(OSD)의 관리 및 촉진을 담당하는 비영리 조합
  - OSI가 인증하는 공개 소스 소프트웨어(OSS) 인증 마크
    - ➡ 소프트웨어가 실제로 공개 소스라는 것을 증명

의미

- 소프트웨어의 소스코드를 자유롭게 읽고, 수정 및 재배포가 가능
  - 소프트웨어의 향상과 발전
  - 여러 사람들이 고치고 쓰고 버그를 개선하는 것이 보다 빠를 수 있음

자유 소프트웨어에서 출발한 OSS

자유 소프트웨어(Free Software)

- 리처드 스톨먼(Richard Stallman), 자유 소프트웨어 재단 설립
- GNU 프로젝트와 관련된 소프트웨어에서 자유를 중시
  - GNU["GNU's Not Unix!"]: 자유 소프트웨어의 운영체제이자 SW
  - 요리의 기본 레시피를 보고 더 나은 요리를 위한 레시피를 만들듯이 소프트웨어 개발도 그렇게 이루어질 수 있다는 것이 바로 GNU(GNU is Not Unix) 정신의 기본
  - Free는 무료라는 의미가 아닌 자유라는 의미
- 카피레프트(Copyleft, Copyright의 반대되는 개념)
  - 제작자에게 저작권은 인정하면서 어느 누구나 소프트웨어를 복제해 사용할 수 있는 권리를 주어야 한다는 개념을 갖는 SW 의미

자유 소프트웨어에서 출발한 OSS

자유 소프트웨어(Free Software)

- 공개된 무료 유닉스 운영체제 Free BSD(Berkeley Software Distribution)
  - 매킨토시나 아이폰 등의 OS로 발전되어 사용
- 일반적으로 소스코드를 변경하고, 변경한 소프트웨어를 공유하려면
  - 원래 프로젝트에 해당 변경 사항을 다시 공유해야만 함

자유 소프트웨어에서 나온 오픈소스 소프트웨어

- 소스의 형태 자체를 중시하는 의미

오픈소스 방식의 의미

오픈소스 지원 관리 서버

- 소스코드를 통해 여러 개발자가 협업하고 공유하며 이를 지원하는 방안을 마련
  - github, gitlab, bitbucket
- 오픈소스 커뮤니티 내의 사고 및 협업 양식
  - 커뮤니티에서 개발된 아이디어와 소프트웨어를 교환
  - 창의적이고 과학적이며 기술적인 발전을 이끌어낼 수 있음

협업 방식

- 서로 간에 소스코드를 사용, 변경 및 공유할 수 있도록 커뮤니티 프로덕션과 동료 평가에 의존
- 개발자들은 보다 혁신적인 소프트웨어 솔루션을 만들기 위해 인사이트, 아이디어 및 코드를 공유
- 모든 사용자가 재활용성과 접근성을 향상시키기 위해 이를 수정, 개선 및 재분배할 수 있도록 보장
- 대규모 협업의 기본 원칙으로 운용
  - 일반 사용자를 위한 보다 지속 가능한 소프트웨어 개발을 구축

오픈소스 소프트웨어 장단점

소스코드를 공개

- 누구든 보다 쉽게 연구하여, 새로운 프로그래밍 기술을 개발
  - 학생은 오픈소스 라이센스를 통해 코드를 편집하고,
  - 친구 또는 보다 광범위한 오픈소스 커뮤니티와 공유
  - 기존 오픈소스 프로젝트를 수정해 실질적으로 커뮤니티에 기여하는 등 실습이 가능

커스터마이징과 혁신 지원

- 기업이 구체적인 요구 사항에 맞게 커스터마이징 가능
- 원래 소스코드에는 포함되지 않았던 새로운 사용자 경험으로 혁신할 수 있도록 지원

단점

- 공개의 의무
- 품질보증 및 유지보수, 보안 등의 어려움

오픈소스 개발 모델

대부분의 웹을 지원하는 서비스 스택 모델: LAMP

- Linux: 오픈소스 운영 체제(OS)이자 세계 최대 규모의 오픈소스 프로젝트
- Apache: 초기 웹에서 핵심 역할을 한 오픈소스 크로스 플랫폼 웹 서버
- MySQL: 대부분의 데이터베이스 기반 웹 애플리케이션에서 사용하는 오픈소스 관계형 데이터베이스 관리 시스템
- PHP: 소프트웨어 개발에 사용되는 범용 스크립팅 언어 (가끔 Perl 또는 Python 언어로 대체됨)

다양한 오픈소스 소프트웨어

빅데이터, 인공지능 분야

- Python
  - https://github.com/python/cpython
- Scikit-learn
  - https://github.com/scikit-learn/scikit-learn
- Tensorflow
  - https://github.com/tensorflow/tensorflow
- Pytorch
  - https://github.com/pytorch/pytorch

다양한 오픈소스 소프트웨어

다양한 오픈소스 개발 모델

- Android 모바일 OS
- Mozilla Firefox 웹 브라우저
- Office 제품군인 OpenOffice 및 LibreOffice
- 버전 제어 시스템인 Git
  - https://github.com/git/git

LESSON 02

오픈소스 소프트웨어 저작권

2 오픈소스 소프트웨어 저작권

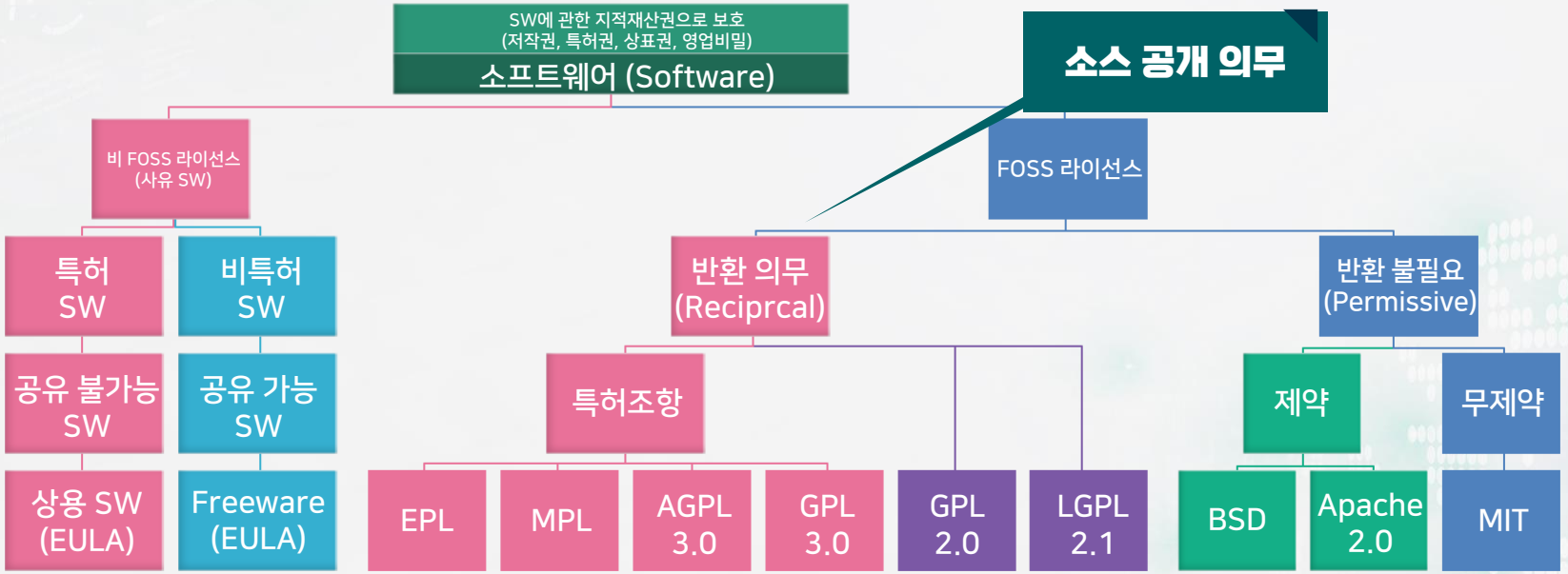
오픈소스 소프트웨어 라이선스 종류

- GNU General Public©(GPL)
  - AGPL, LGPL
- MIT License©
- Apache®
- BSD
- MySQL™:
- SUSE
- Ubuntu®

2 오픈소스 소프트웨어 저작권

오픈소스 라이선스 저작권

- 오픈소스 SW 개발자가 만들어 놓은 저작권의 범위에 따라 해당 소프트웨어를 사용
  - 위반할 경우, 라이선스 위반 및 저작권 침해로 이에 대한 법적 책임이 따름



<출처> : https://www.oss.kr/oss_guide/show/044d96fd-413d-4ab3-ad72-4776d2b7e002

2 오픈소스 소프트웨어 저작권

소스코드 반환의무

GPL, AGPL, LGPL, MPL, EPL 등

- 링크되거나 코드가 포함된 SW의 소스코드를 공개
- 특허, 영업비밀, 핵심 기술 등이 외부로 유출될 가능성이 있으니 주의
- 저작권 고지의무
  - 저작권법에 따른 법적 권리를 보장하는 것으로 공개 SW 라이선스에서 준수하지 않을 시 법정 분쟁의 가능성이 생김

2 오픈소스 소프트웨어 저작권

1. GPL(GNU - General Public License)

자유소프트웨어재단(FSS)에서 만든 라이선스

- 자유소프트웨어재단 설립자인 리처드 스톨만에 의해서 만들어졌음
- 가장 많이 알려진 카피레프트에 속한 라이선스
- 프로그램은 목적이나 형태의 제한없이 사용이 가능
- 프로그램을 이후 수정하고 배포하는 모든 경우에 무조건 GPL로 공개를 해야 함
  - GPL을 따른 프로그램을 사용해서 2차 저작물을 만들면 2차 저작물도 반드시 GPL 라이센스를 명시해야 함

GPL 라이센스 적용 예

- 리눅스 커널, Git, 마리아 DB, 워드프레스, 파이어폭스(v2.0) 등

2. AGPL(GNU - Affero GPL)

GPL을 기반으로 만든 라이선스

- 네트워크로 상호 작용하는 소프트웨어의 소스코드도 공개해야 한다는 조항을 추가한 라이선스
  - 어떤 개발자가 프로그램의 소스코드를 수정해서 사용하고 프로그램을 배포하지 않았을 때 발생할 수 있는 소스코드의 공유 불가 현상을 해결하기 위해 만들어짐
- 서버에서 프로그램을 실행해서 다른 사용자들과 통신 중
  - 실행되고 있는 프로그램의 소스코드를 사용자들이 다운로드할 수 있도록 해야 하는 조항을 포함

AGPL 라이센스 적용 예

- 몽고DB 등

3. LGPL(GNU - Lesser GPL)

보다 완화된 GPL 라이선스

- GPL 라이센스를 사용할 때 반드시 소스코드를 다시 GPL로 공개해야 하는 부담 때문에 실무에서 사용되기 어려운 점을 보완하기 위해서 만들어진 라이선스
  - 전체 소스코드를 공개하지 않고 사용된 오픈소스 라이브러리에 대한 소스코드만 공개
- 초기에는 한정된 라이브러리에만 적용된다는 것을 나타내기 위해서 Library GPL이라고 불렀으나 이 용어가 모든 라이브러리에 해당된다는 오해를 만들기 쉽기 때문에
  - 이후에 Lesser GPL로 명칭을 변경

LGPL 라이센스 적용 예

- 파이어폭스(v2.1) 등

4. Apache License

- 소스코드 공개에 대한 의무사항은 라이센스에 포함되어 있지 않음
  - 아파치 라이센스 소스코드를 수정하고 배포 시
    - 아파치 라이센스 버전 2.0을 꼭 포함시켜야 하고 아파치재단의 소프트웨어라는 사실을 명시
- Apache License 적용 예
  - 안드로이드, 하둡 등

5. MIT License

- MIT에서 학생들을 지원하기 위해서 만든 라이선스
  - 라이센스와 저작권 관련 명시 의무
  - 가장 느슨한 조건을 가지고 있어서 많은 사람들이 사용하기 용이
    - 사용자가 최소의 제한사항으로 원래 코드를 수정할 수 있도록 허용하는 무료 소프트웨어 라이센스
- MIT License 라이센스 적용 예
  - 부트스트랩, Angular.js, Backbone.js, jQuery

2 오픈소스 소프트웨어 저작권

대표 라이선스 정리

주요 의무사항

라이선스	배포 시 라이선스 사본 첨부	저작권 고지 사항 유지	소스코드 제공 의무와 범위	타 라이선스와 통합 및 배포 허용	수정 시 수정 내용 고지
GPL 2.0	O	O	전체 코드	조건부	X
GPL 3.0	O	O	전체 코드	X	O
LGPL 3.0	O	O	전체 코드	O	O
BSD	X	O	X	조건부	X
Apache2.0	O	O	X	O	X

의무 강도에 따른 분류

카테고리	소스코드 공개 의무	사례
Strong copy	2차 저작물 소스코드 공개 의무 있음	GPL
Weak copyleft	특정 조건에서 2차 저작물 소스코드 공개 의무 없음	LGPL
Non copyleft	2차 저작물 소스코드 공개 의무 없음	Apache, BSD, MIT

Summary

$ 오픈소스 소프트웨어의 의미

- 소프트웨어의 소스코드를 자유롭게 읽고, 수정 및 재배포가 가능

오픈소스 소프트웨어 개발을 지원하는 서버인 깃허브에서의 협업 방식

- github, gitlab, bitbucket

대표적인 오픈소스 소프트웨어

- 파이썬, 사이킷런, 파이토치, 텐서플로, MySql, 몽고DB 등

오픈소스 소프트웨어의 저작권을 이해할 수 있다.

- 의무 강도에 따른 분류
  - GPL, LGPL, Apache, BSD, MIT 순으로 의무 강도가 낮아짐