

오픈소스sw의 이해

Lecture #1: Introduction to OSS

Software Engineering laboratory
Yeungnam University



■ Open Source Software (OSS)

- OSS is a type of computer software
 - Source code is released under a license that allows users to study, change, and distribute the software to anyone for any purpose.
 - This promotes collaborative and transparent development.
 - It encourages innovation and knowledge sharing as the code is accessible to the public.
 - This model has led to the development of a wide variety of software, from operating systems like Linux to web browsers like Mozilla Firefox.

<-> Closed source, Proprietary source

개인의, 사유의



What is Open Source Software?

■ 오픈 소스 소프트웨어(Open Source Software, 이하 OSS)

- 소스 코드가 공개되어/누구나 접근하고/사용할 수 있는 소프트웨어
 - 이는 개발자 혹은 사용자가/해당 소프트웨어의 내부 코드를 열람하고 수정할 수 있게 함으로써 사용자 간의 협력과 소프트웨어의 발전을 촉진하는 개념을 포함
- 누구나 자유롭게 사용, 복제, 배포, 수정, 활용 가능
- 저작권자의 권익을 보호할 수 있도록 제도화된 소프트웨어
- 일반 대중의 공동 연구로 개발, 시험 및 개선
 - 동료 평가와 커뮤니티 기반 소프트웨어 개발
 - 분산된 지역의 여러 개발자의 협업 방식으로 개발

What is Open Source Software?

■ OSS의 특징 요약

● 소스 코드 공개

- 소프트웨어의 소스 코드가 공개되어 있어 누구든지 접근 가능
- 소프트웨어의 동작 방식을 이해하고 필요에 따라 수정 가능

● 자유로운 수정 및 배포

- OSS는 사용자가 소스 코드를 자유롭게 수정하고, 해당 변경사항을 다시 공개할 수 있는 권리를 제공
- 수정된 소프트웨어를 누구든지 무료로 사용하거나 배포 가능

● 커뮤니티 협력

- 다수의 개발자가 참여하여 의견을 나누고 코드를 개선함으로써, 커뮤니티 기반의 협력 가능

● 다양한 라이선스

- 다양한 라이선스가 존재하며, 각 라이선스는 소프트웨어를 어떻게 사용하고 공유할 수 있는지 규정함.

오픈소스SW, 비공개SW, 상용SW

■ 오픈소스SW와 비공개SW

● 공통점

- 저작권은 작성과 동시에 저작권자에게 귀속되므로 오픈소스SW와 비공개 소프트웨어 공통으로 저작권이 존재

● 차이점

- 오픈소스SW의 저작권자들은 일반적으로 소스코드를 공개하여 누구나 자유롭게 사용, 복제, 배포, 수정, 활용할 수 있는 권리를 사용자에게 부여하지만,
- 비공개SW의 저작권자들은 일반적으로 소스코드를 공개하지 않으면서 사용자가 자유롭게 사용, 복제, 배포, 수정, 활용 하는 것을 금지하고 사용에 대한 대가를 요구

오픈소스SW, 비공개SW, 상용SW

■ 상용SW

- 상용 소프트웨어는 수익 창출을 목적으로 만들어진 소프트웨어 제품
- 일반적으로 유료로 소스코드를 공개하지 않고 제품 자체를 판매
- 때때로 소스코드를 공개하고 오픈소스SW 라이선스를 적용하여 듀얼 라이선스 정책을 가지고 사업을 하기도 함

확 인

코드가 공개된 상용 소프트웨어도 있기 때문에 오픈소스SW의 반대 개념은 상용 소프트웨어가 아니라 비공개 소프트웨어이다. 오픈소스SW는 Copyleft라고 해서 저작권자가 사용권리를 무상으로 제공하면서 왼쪽과 같은 심볼을 사용하고 있고, 비공개 소프트웨어는 Copyright이라고 해서 오른쪽과 같은 심볼을 사용하며 저작권자가 독점적 권리를 주장하고 있다.



[오픈소스SW 심볼]

Copyleft



[비공개 소프트웨어 심볼]

Copyright

OSI의 10가지 오픈소스SW 정의

공개SW(Open Source Software)

저작권은 존재하지만 저작권자가 **소스코드를 공개하여 누구나 자유롭게 사용, 복제, 배포, 수정, 활용할 수 있는 SW**를 지칭

공개SW 반대 개념은 상용SW가 아닌(소스코드)비공개SW



*OSI(Open Source Initiative) : 공개SW 이용 장려를 위해 만들어진 비영리단체

OSI의 10가지 오픈소스SW 정의

■ (1) 자유 배포 (Free redistribution)

공제, 판매

- 특정한 소프트웨어의 라이선스에 배포나 판매상의 제한을 설정할 수 없음.
- 배포판에 대한 판매나 양도에 있어서 별도의 라이선스 비용을 징수할 수 없음.
- 즉, 오픈소스SW를 재배포하는 사용자에 대한 제한을 할 수 없으며, 재배포 소프트웨어에 별도 라이선스 비용을 징수할 수 없다.

OSI의 10가지 오픈소스SW 정의

■ (2) 소스코드 공개 (Source Code Open)

- 프로그램 저작물에는 반드시 소스코드가 포함되어야 함.
- 가장 바람직한 방법은 인터넷을 통해서 소스코드를 무료로 다운로드 받을 수 있도록 하는 것.
- 프로그래머들이 개작하기에 용이한 형태로 제공되어야 함.
- 고의로 복잡하고 혼란스럽게 만들어진 형태와 선행 처리기나 번역기에 의해서 생성된 중간 형태의 코드는 허용되지 않음.
- 즉, 오픈소스SW는 소스코드 형태로 배포되어야 한다.

OSI의 10가지 오픈소스SW 정의

■ (3) 2차적 저작물 (Derived works) 허용

- 프로그램 원저작물의 개작이나 이를 이용한 2차적 프로그램의 창작이 허용되어야 함.
- 이러한 파생적 프로그램들은 최초의 프로그램이 갖고 있던 라이선스의 규정과 동일한 조건 하에서 재배포될 수 있어야 함.
- 즉, 오픈소스SW를 사용하여 2차적 저작물을 만들 수 있고 이를 배포할 수 있다.

OSI의 10가지 오픈소스SW 정의

■ (4) 소스코드 수정 제한(Integrity of The Author's Source Code)

- 빌드 과정을 통해서 프로그램을 제작할 목적으로 소스 코드와 패치 파일을 함께 배포할 경우에는, 정상적인 빌드를 보장하기 위해서 라이선스 안에 소스코드의 수정을 제한하는 항목을 추가할 수 있음.
- 그러나 이러한 경우에도 수정된 소스 코드를 이용해서 만들어진 소프트웨어에 대한 자유로운 배포를 허용해야 함.
- 즉, 오픈소스SW의 정상적 빌드와 저작권자의 저작권을 보호하기 위해 수정을 제한하는 항목을 추가 할 수 있으나 자유로운 배포를 허용해야 함.

OSI의 10가지 오픈소스SW 정의

- (5) 개인이나 단체에 대한 차별 금지 (No Discrimination Against Persons or Groups)
 - 라이선스는 모든 개인과 단체에 대해서 동일한 기준으로 적용되어야 한다.
 - 즉, 모든 라이선스는 구분 없이 동일하게 적용되어야 한다.

OSI의 10가지 오픈소스SW 정의

- (6) 사용 분야에 대한 제한 금지 (No Discrimination Against Fields of Endeavor)
 - 라이선스 안에 특정한 분야에 종사하는 사람에 대한 프로그램 사용상의 제한을 설정할 수 없음.
 - 예를 들면, 유전연구나 상용 사업체에서는 해당 프로그램을 사용할 수 없다는 등과 같이 특정한 분야에 대한 사용을 금지하는 제한을 설정해서는 안됨.
 - 즉, 사용 분야에 대한 제한을 해서는 안된다.

OSI의 10가지 오픈소스SW 정의

■ (7) 라이선스의 배포 (Distribution of License)

- 프로그램에 대한 권리는 반복되는 배포에 따른 별도의 라이선스 승인이나 양도 과정 없이도 프로그램을 배포 받은 모든 사람에게 동일하게 적용됨.
- 즉, 라이선스 준수를 하여 사용하는 경우 별도의 승인 혹은 양도가 없어도 사용 가능하다.
└ 사용, 수정, 복제, 배포, 활용에 대한 규칙과 조건을 명시하는 법적인 도구

OSI의 10가지 오픈소스SW 정의

- (8) 라이선스 적용상의 동일성 유지 (License must not be specific to a product)
 - 프로그램에 대한 권리는 반복되는 배포 과정의 모든 단계에서 동일한 효력 발생.
 - 만약, 특정한 배포판에 포함되어 있던 프로그램을 독립적으로 사용하거나 재배포한다면,/
 - 해당 프로그램을 배포 받는 사람은 프로그램이 포함되어 있던 최초의 배포판 상태에서 발생한 권리와 동일한 권리를 지님.
 - 즉, 한번 적용된 라이선스는 몇 번이 재배포 되더라도 동일하게 권리와 의무를 갖는다.

OSI의 10가지 오픈소스SW 정의

- (9) 다른 라이선스의 포괄적 수용 (License must not contaminate other software)
 - 라이선스에 오픈소스SW와 함께 배포되는 소프트웨어에 대한 제한을 설정해서는 안 됨.
 - 예를 들면, 동일한 매체를 통해서 배포되는 소프트웨어는 모두 오픈소스SW이어야 한다는 제한으로 인해서 다른 라이선스 기준을 따르는 소프트웨어가 함께 배포될 수 있는 형태를 금지해서는 안 됨.
 - 즉, 함께 배포되는 소프트웨어에 제한을 두지 말아야 한다.

OSI의 10가지 오픈소스SW 정의

- (10) 라이선스의 기술적 중립성 (License must be Technology-Neutral)
 - 라이선스의 어떠한 규정도 개별기술 또는 인터페이스 형태에 기초하여 규정하지 말아야 함.
 - 즉, 라이선스의 조항은 모든 기술 혹은 형태 등에 동일하게 적용될 수 있어야 한다.

OSS Licenses

확 인

OSI에서는 오픈소스SW 10가지 정의를 라이선스에 명문화하여 사용자에게 요구하고 있는 라이선스에 대해 인증을 주고 있다. OSI의 인증을 받지 못한 라이선스도 저작권자가 사용자에게 사용 허가를 제공하는 저작권자의 권리 요구사항으로서 라이선스는 그대로 유효하다.

Academic Free License 3.0 (AFL-3.0)	Microsoft Public License (Ms-PL)
Affero GNU Public License: See "GNU Affero General Public License 3.0 (AGPL-3.0)"	Microsoft Reciprocal License (Ms-RL)
Adaptive Public License (APL-1.0)	MIT license (MIT)
Apache License 2.0 (Apache-2.0)	Motosoto License (Motosoto)
Apple Public Source License (APSL-2.0)	Mozilla Public License 2.0 (MPL-2.0)
Artistic license 2.0 (Artistic-2.0)	Multics License (Multics)
Attribution Assurance Licenses (AAL)	NASA Open Source Agreement 1.3 (NASA 1.3)
BSD 3-Clause "New" or "Revised" License (BSD-3-Clause)	NTP License (NTP)
BSD 2-Clause "Simplified" or "FreeBSD" License (BSD-2-Clause)	Naumen Public License (Naumen)
Boost Software License (BSL-1.0)	Nethack General Public License (NGPL)
Computer Associates Trusted Open Source License 1.1 (CATOSL-1.1)	Nokia Open Source License (Nokia)
Common Development and Distribution License 1.0 (CDDL-1.0)	Non-Profit Open Software License 3.0 (NPOSL-3.0)
Common Public Attribution License 1.0 (CPAL-1.0)	OCLC Research Public License 2.0 (OCLC-2.0)
CUA Office Public License Version 1.0 (CUA-OPL-1.0)	Open Font License 1.1 (OFL 1.1)
EU DataGrid Software License (EUDatagrid)	Open Group Test Suite License (OGTSL)
Eclipse Public License 1.0 (EPL-1.0)	Open Software License 3.0 (OSL-3.0)
Educational Community License, Version 2.0 (ECL-2.0)	PHP License 3.0 (PHP-3.0)
Eiffel Forum License V2.0 (EFL-2.0)	The PostgreSQL License (PostgreSQL)
Entessa Public License (Entessa)	Python License (Python-2.0) (overall Python license)
European Union Public License, Version 1.1 (EUPL-1.1) (<i>links to every language's version on their site</i>)	CNRI Python license (CNRI-Python) (CNRI portion of Python License)
Fair License (FAIR)	Q Public License (QPL-1.0)
Framework License (Framework-1.0)	RealNetworks Public Source License V1.0 (RPSL-1.0)
GNU Affero General Public License v3 (AGPL-3.0)	Reciprocal Public License 1.5 (RPL-1.5)
GNU General Public License version 2.0 (GPL-2.0)	Ricoh Source Code Public License (RSCPL)
GNU General Public License version 3.0 (GPL-3.0)	Simple Public License 2.0 (SimPL-2.0)
GNU Library or "Lesser" General Public License version 2.1 (LGPL-2.1)	Sleepycat License (Sleepycat)
GNU Library or "Lesser" General Public License version 3.0 (LGPL-3.0)	Sun Public License 1.0 (SPL-1.0)
Historical Permission Notice and Disclaimer (HPND)	Sybase Open Watcom Public License 1.0 (Watcom-1.0)
IBM Public License 1.0 (IPL-1.0)	University of Illinois/NCSA Open Source License (NCSA)
IPA Font License (IPA)	Vovida Software License v. 1.0 (VSL-1.0)
ISC License (ISC)	W3C License (W3C)
LaTeX Project Public License 1.3c (LPPL-1.3c)	wxWindows Library License (WXwindows)
Lucent Public License Version 1.02 (LPL-1.02)	X.Net License (Xnet)
MirOS Licence (MirOS)	Zope Public License 2.0 (ZPL-2.0)
	zlib/libpng license (Zlib)

OSS Licenses

■ 정의 (License: 사용권, 사용허가)

- 오픈소스SW 저작권자가 사용자에게 자신의 저작물에 대한 사용을 허락하는 명문화된 문서

■ 종류

- 좋은 라이선스나 나쁜 라이선스는 없으며 어떤 라이선스도 다른 라이선스보다 낫지 않음.
- 누구나 자신에게 맞는 오픈소스 라이선스를 만들 수 있음.
 - 이것이 바로 오픈소스 라이선스가 너무 많은 이유.
 - 이로 인해 오픈 소스 라이선스를 선택하는 것이 복잡한 비즈니스가 될 수 있음.
- OSI에서는 가장 일반적으로 사용되는 80개가 조금 넘는 오픈 소스 라이선스로 구성된 승인된 라이선스 목록을 제공함.

OSS Licenses

■ Top open source licenses explained

License	Type	Must distribute source code 원래 소스 코드	Distribute <u>modified</u> code under same license	Distribute <u>entire project</u> that uses licensed code under same license	Other notable items
Apache 2.0 License	Permissive 관여없음	No	No	No	Requires notice of significant modifications 큰 수정
Berkeley Software Distribution (BSD) 3-Clause	Permissive	No	No	No	
Common Development and Distribution License (CDDL)	Weak copyleft	Yes	Yes	No	Requires notice identifying author of modifications 수정의 작성자
Eclipse Public License (EPL) v2.0	Weak copyleft	Yes	Yes	No	
GNU Affero Public License (AGPL) v3.0	Strong copyleft	Yes	Yes	Yes	Interaction with software over a network counts as distribution 상호작용 통해 간주한다 배포로
GNU General Public License (GPL) v2.0	Strong copyleft	Yes	Yes	Yes	
GNU General Public License (GPL) v3.0	Strong copyleft	Yes	Yes	Yes	
GNU General Public License with Classpath Exception	Weak copyleft	Yes	Yes	No	
GNU Lesser General Public License (LGPL)	Weak copyleft	Yes	Yes	No	
Microsoft Public License (MS-PL) ≠ MPL Mozilla Public License	Weak copyleft	No	Yes, if source code is distributed	No	
MIT License	Permissive	No	No	No	

* All licenses in this chart permit commercial use and all require a copy of the license in distribution.

Apache, BSD, MIT permissive
CDDL, EPL, LGPL, MPL Weak copyleft
AGPL v3.0 GPL v2.0, 3.0 Strong copyleft

OSS Licenses

■ 분류

- 개별 오픈소스SW 저작권자들의 저작권리 주장 및 사용허가를 제공하는 라이선스라는 명문화된 형태로 반영되어 코드 공개여부 및 범위에 따라 분류

- Permissive 계열
허용적인
- Copyleft 계열
모두 공개
 - Weak copyleft 계열
 - Strong copyleft 계열

Permissive	Apache-2.0	3-clause / New BSD	2-clause / Free BSD	MIT/X11	Zlib/libpng
Weak copyleft	LGPL-2.1	LGPL-2.1+	LGPL-3.0 or LGPL-3.0+	MPL-1.1	MPL-2.0
	EPL-1.0	EPL-2.0			
Copyleft	GPL-2.0	GPL-2.0+	GPL-3.0 or GPL-3.0+	AGPL-3.0	

* 오픈소스SW는 해당 라이선스 의무사항만 준수한다면
아무나 자유롭게 상업적으로 이용 가능함.

OSS Licenses

■ 분류 (cont'd)

● Permissive 계열

- 사용함에 있어서 별다른 요구사항을 부여하지 않고 **사용자에게 광범위한 사용권한을 부여**하는 오픈소스SW 계열
- 기본적으로 **저작권 고지** 및 **라이선스 사용 고지** 정도만을 요구
- BSD, MIT 등이 대표적

● Copyleft 계열

- 소스코드 공개 및 원 저작물뿐만 아니라 원저작물을 기반으로 만들어지는 **파생저작물에도 동일한 라이선스로 공개**되어야 한다는 소스코드 공유 개념 계열
 - Weak copyleft 계열
수정 부분을 포함하는 코드에 대한 공개를 요구하는 라이선스들로서 MPL, EPL, LGPL 등이 대표적.
 - Strong copyleft 계열
결합된 모든 코드에 대한 동일조건의 소스코드 공개를 요구하는 라이선스로서 GPL, AGPL 등이 해당.

OSS Licenses

■ 오픈소스SW의 고지의 목적

● 저작권 고지

- 모든 오픈소스SW 라이선스에서 공통적으로 요구하고 있는 의무사항.
- 오픈소스SW의 저작자를 알림으로서 사용자에게 대한 사용 권리를 인지시키는 동시에 질문이나 토론 사항이 있을 경우 저작권자를 특정하는 기능.
 - 협업 개발을 위한 채널로서 활용.
- 저작자 표시 문구는 해당 부분 소스코드의 라이선스 변경 요청을 할 경우에도 필요함.

● 라이선스 고지

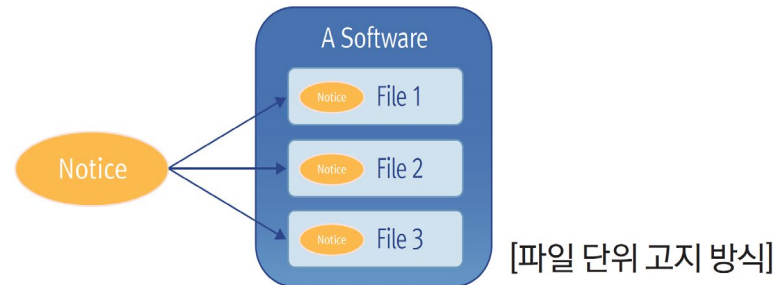
- 오픈소스SW를 사용한 경우 사용자들의 사용 권리를 알리기 위해 오픈소스SW 라이선스를 고지해야 함.
- 대부분의 오픈소스SW 라이선스에서 의무사항으로 규정.
- 만일 이 라이선스 문구가 없는 코드의 경우에는 사용이 불가하다는 암묵적 정보를 포함함.

OSS Licenses

■ 오픈소스SW의 고지 이행 방법

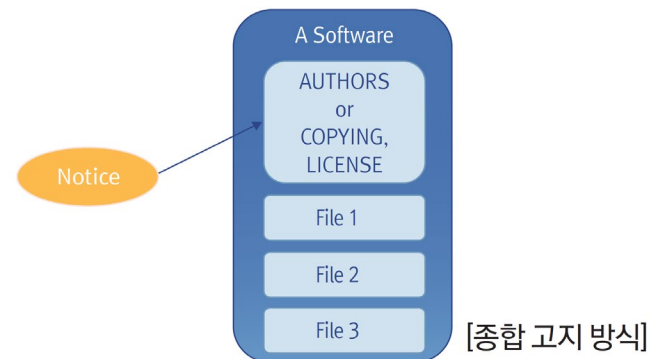
● (1) File scope approach (파일 단위 고지)

- 파일에 여러 명의 기여자가 있을 경우 그 각 파일의 모두 부분에 저작권 표시를 하는 방식.



● (2) Centralized approach (종합 고지)

- 프로젝트 최상위에 AUTHORS or COPYRIGHT 파일을 만들고 저작권 정보를 모두 합쳐서 기재하는 방식.

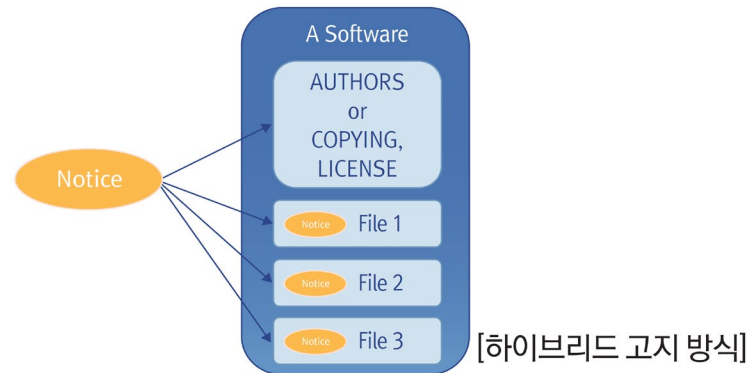


OSS Licenses

■ 오픈소스SW의 고지 이행 방법

● (3) Hybrid approach (하이브리드 고지)

- 파일 단위 고지 방식과 종합 고지 방식을 적절히 혼합하여 사용하는 방식.



* 세 가지 고지 방식 중 가장 권장할 고지방식은 하이브리드 방식이다.

* 오픈소스SW 사용자들이 특정 프로젝트의 전체를 사용할 수도 있지만 일반적/부분적으로 특정 모듈이나 라이브러리 소스코드들만을 취사선택해서 사용할 경우, 또 개별 파일에 고지문이 없을 경우 해당 파일은 출처가 모호해지게 됨에 따라서 가급적 개별 파일의 헤드라인에 고지문을 넣고 전체 프로젝트에 종합 고지문을 함께 작성하여 배포하는 것을 권장함.

OSS Licenses

■ 고지문 작성 방식

● 공통사항

- 원래의 소스코드에 있는 저작자 표시 내용을 그대로 표시
- 저작권 문구에는 통상 웹 주소가 동시에 표시되는데, 이는 저작권 표시가 그 오픈소스SW의 내용에 대한 문의처를 알리는 기능을 한다는 점에서 매우 중요함.

Copyright 2030, Kevin Seo
* Dual licensed under the MIT or GPL Version 2 licenses.
* <http://jquery.org/license>

[고지문 작성 예]

permissive

strong copyleft

OSS Licenses

■ 고지문 작성 방식

● 변경사항 고지

- GPL-2.0 라이선스나 Apache-2.0 라이선스 같은 특정 라이선스에서는 소스코드 수정 시 변경사항에 대한 고지의무를 요구함.
strong copyleft *permissive*
- 변경사항에 대한 고지를 할 경우에는 원 저작권자의 고지문을 유지하고 수정한 저작권자의 고지를 추가하여 배포함으로써 저작권의 범위를 정확히 제공해야 함.

Original Work Copyright © 2024, YU Company,
Modified Work Copyright © 2030-12.31 ABC Company

[변경사항 고지문 작성 예]

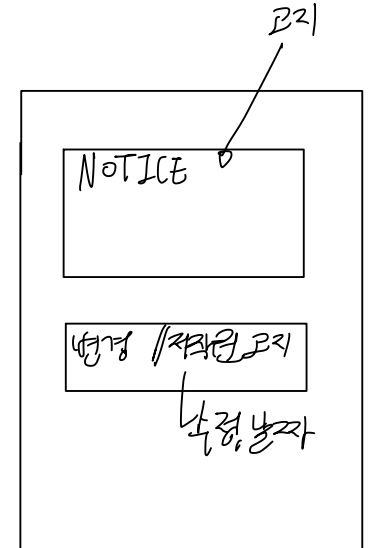
OSS Licenses

■ 고지문 작성 (예)

● GPL-3.0 고지문 적용 방식

- ^{strong copyleft} 오픈소스SW 해당 파일 주석부분/or NOTICE.txt에 상기 영문 고지한다.
- 오픈소스SW를 변경 했을 경우 해당 파일 주석 부분에 저작권 고지한다.
- NOTICE.txt는 통상적으로 소스코드 최상위 수준에 위치하도록 고지한다.
- 파일을 수정할 때는 파일을 수정했다는 사실과 그 날짜를 파일에 명시해야 한다.

Original Work Copyright © 2024, XXX Company,
Modified Work Copyright © Year, Company Name 유저



OSS Licenses


■ 오픈소스SW 라이선스 양립성(Compatibility)

- 복수의 저작물에 적용된 상이한 라이선스가 새로운 저작물을 만들기 위하여 해당 저작물들의 소스코드나 내용물을 결합시키는 것을 불가능하게 만드는 조건을 포함할 때 발생하는 문제.
- 기본적으로 오픈소스SW를 사용함에 있어서 하나 이상의 오픈소스SW 라이선스를 사용할 경우에 검토해야 될 사항
- ‘적어도 두 패키지 중의 하나의 저작자의 직접적인 허가가 없이는 두 패키지의 결합을 합법적으로 배포할 수 없는 경우’두 개의 라이선스는 호환되지 않는다 or 양립할 수 없다고 지칭함.

OSS Licenses

■ 오픈소스SW 라이선스 양립성(Compatibility) 발생 예

- 'A+B'라는 프로그램을 만들어 배포하고자 하는 경우,
 - MPL-1.1 조건의 A코드: MPL-1.1은 'A+B'의 A부분을 MPL로 배포할 것을 요구
weak copyleft
 - GPL-2.0 조건의 B코드: GPL-2.0은 'A+B' 전체를 GPL-2.0으로 배포할 것을 요구
strong copyleft
- 'A+B' 프로그램의 배포를 강행하였다면,
 - 적어도 GPL-2.0 혹은 MPL-1.1중 어느 한쪽 라이선스를 위반하는 결과를 초래.
 - 따라서 해당 A 혹은 B 프로그램의 원 저작자로부터 별도의 허가를 받지 않는 한, 'A+B'를 배포한 자는 저작권법 내지 계약 위반에 따른 책임.

 배포 불가

* MPL-2.0에서는 Larger Work 배포 시에 부차적 라이선스로 GPL-2.0, LGPL-2.1, AGPL-3.0 ~~혹은~~ 이후 버전으로 배포할 수 있게 함으로써 라이선스 호환성 문제를 개선함. *strong copyleft*

OSS Licenses

■ 오픈소스SW 라이선스 양립성(Compatibility)

● 해결방안

- 한쪽의 오픈소스SW를 제거 혹은 다른 라이선스로 대체
- 해당 라이선스에서 명시하고 있는 라이선스 적용의 예외에 해당하는 결합 형태 고려
- 라이선스 충돌이 방지하지 않도록 결합형태를 분리된 저작물로 변경을 검토

● 오픈소스SW 활용시 오픈소스SW 라이선스 검증 작업이 반드시 필요함.

Benefits of Open Source Software

개발효율성

- 공개SW는 SW개발을 위한 핵심 자산
- 상용SW의 96%가 공개SW 활용을 통해 개발기간 단축 및 비용절감 실현

기업 저변확보

- 소스코드 개방을 통해 SW저변 밑바탕 확산 용이



인재양성

- 공개SW 통한 SW교육의 효과적 활용 및 실무 SW기술 학습

최신기술 도입

- 소스코드 원천기술에 대한 접근으로 선진SW기술습득 용이
- 개방적인 SW개발문화에 따른 기술혁신과 아이디어 수용 가능

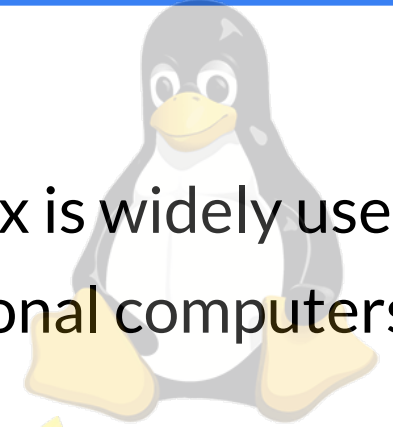
독점제조사 의존 벤더 종속성 탈피

- 국내 시장에서의 독과점 SW 대체재 역할

Examples of Popular Open Source Software Projects

■ Linux

- One of the most well-known open source software projects/ Linux is widely used in servers, embedded systems, and as an operating system for personal computers.



■ Mozilla Firefox

- A popular, open source web browser known for its speed, privacy features, and extensive support for web standards.



■ WordPress

- This content management system (CMS) is extensively used for creating websites and blogs due to its ease of use, flexibility, and extensive plugin and theme ecosystem.



테마 생태계

Challenges and Limitations of Open Source Software

■ Maintaining Quality

- Ensuring quality in a large, distributed development ecosystem, with diverse contributors and varying levels of expertise, can be a challenge for open source projects.

■ Security Concerns

- Open source software may face security vulnerabilities if not rigorously maintained and reviewed, potentially leading to exploitation or data breaches.

■ Complex Governance

- Establishing effective governance models and decision-making processes in open source projects can be complex, potentially leading to conflicts and disagreements among contributors.

History of OSS (cited from IBM)

■ ~1970년대

- 1970년대 중반까지 컴퓨터 코드는 컴퓨터 하드웨어 운영에 포함되는 것으로 여겨졌으며, 저작권법의 보호 대상인 고유한 지적 재산이 아니라고 간주.
 - 조직들은 자체 소프트웨어를 프로그래밍했으며, 코드 공유는 일반적인 관행.
- CONTU(Commission on New Technological Uses of Copyrighted Works)가 1974년에 제정되면서, 소프트웨어 코드는 저작권법의 보호를 받을 수 있는 창작물의 범주에 속한다고 결론
위원회 저작물
- 이로 인해 독점적 소스 코드를 주된 수입원으로 삼는 독립적 소프트웨어 회사가 하나의 산업으로 급성장!
 - 개인용 컴퓨팅으로 모든 기업과 많은 가정에서 애플리케이션을 사용하게 되면서 소프트웨어 시장 경쟁 과열

History of OSS (cited from IBM)

■ 1980년대

- 독점적 소프트웨어의 제한 및 한계에 대한 일종의 반항은 1983년에 시작
- 프로그래머인 Richard Stallman¹⁾이 Free Software Foundation을 설립
 - 자신의 작업을 수행하기 위해 적합한 형식으로 독점적 소프트웨어를 맞춤화할 수 없음
 - “소프트웨어는 맥주가 아니며 표현처럼 자유로워야 한다”고 생각
 - 무료로 맞춤화에 사용할 수 있는 소프트웨어라는 개념을 지지
 - 애플리케이션 중에서 무엇보다 AT&T가 소유한 Unix 운영 체제에 대한 오픈 소스 대안의 개발을 추진
 - 그는 최초의 카피레프트 소프트웨어 라이선스인 GNU GPL(General Public License)을 혁신
 - GPL 라이선스에 따라 소스 코드를 개선한 사람이면 누구나 그와 마찬가지로 모두에게 무료로 편집된 버전을 게시해야 함.



History of OSS (cited from IBM)

■ 1990년대

- 1999년에 "오픈 소스"라는 용어가 채택
 - 많은 사람들이 Stallman이 사용한 "Free software"라는 용어가 부적절하게 가격적으로 "무료"라는 개념을 강조한다고 느낌.
 - The Open Source Initiative (OSI)가 이를 옹호하기 위해 설립
 - 이 조직은 또한 오픈 소스 정의를 통해 이 산업을 위한 기본 규칙을 제정
 - 호환되는 오픈 소스 라이선스를 호스팅

Every free software is open source.
Every open-source software is not free software.

OpenUp (오픈소스 소프트웨어 통합지원센터)

■ 개요

- 오픈소스SW를 개발, 공유, 활용하기를 원하는 개발자, 기업, 정부 등을 체계적으로 지원
- 수요(기업, 공공기관)및 공급(개발자, 기업)의 선순환 구조 구축
- 오픈소스SW 활성화의 허브 역할을 수행
- 정보통신산업진흥원(NIPA) 오픈소스SW사업

■ 목적

- 오픈소스SW를 활용함에 있어 기술지원, 인력양성, 창업지원, 저변 확대 등 기업, 개발자, 커뮤니티 지원기능 통합을 통한 시너지 극대화

OpenUp (오픈소스 소프트웨어 통합지원센터)

오픈소스SW 개발·공유·활용 원스톱 지원



○ 오픈소스SW 매니지먼트 아카데미

- 국내기업, 기관 담당자 대상 오픈소스SW 관리 전문가 양성 집중과정
- 기업 및 기관의 맞춤형 단기 교육과정

○ 오픈소스SW 라이선스/보안 검증

- 개발 소프트웨어 내 오픈소스SW 활용 식별 및 라이선스/보안취약점을 확인하여 리스크 사전예방

○ 오픈소스SW 라이선스/거버넌스 컨설팅

○ 오픈소스SW 확인서 발급 지원

○ 오픈소스SW 기업성장 지원

○ 오픈소스 컨트리뷰션 아카데미

- 오픈소스SW 프로젝트에 참여하여 참여·공유·개방·협업의 다양한 오픈소스 기여(Contribution)를 직접 경험

○ 오픈 프런티어(오픈소스SW 개발자) 지원

- 오픈소스 프로젝트 개발 활동을 위한 개발 인프라 및 연구활동 지원

○ 오픈소스SW 커뮤니티 지원

○ 오픈소스SW 교육

○ SNS 채널 운영 및 뉴스레터 제공

OpenUp (오픈소스 소프트웨어 통합지원센터)

■ 오픈소스SW 라이선스 자동 검증의 필요성

- 소프트웨어 개발시 활용한 오픈소스SW를 일일이 파악하는 것은 매우 힘든 작업.
 - 소프트웨어를 개발이 진행되고 배포에 이르기까지 수많은 변경 작업이 발생.
 - 수동으로 관리했다고 하더라도 미처 확인하지 못한 오픈소스SW의 소스코드가 포함될 수 있음.

■ 오픈소스SW 라이선스 검증 지원



전세계의 오픈소스SW 유관단체

■ 오픈소스SW 라이선스 자동 검증의 필요성

- Linux Foundation(<http://linuxfoundation.org>) : 리눅스 표준 재정 등 오픈소스SW 선도
- Software Freedom Law Center(<http://www.softwarefreedom.org>) : 법적 자문 기관
- GPL-Violations(<http://gpl-violations.org>) : GPL 위반 모니터링
- Open Source Initiative(<http://opensource.org>) : 오픈소스SW 라이선스 인증 관리
- Open Invention Network(<http://openinventionnetwork.com>) : 리눅스 및
오픈소스^{발명품}SW 특허 관리 등