



# CONTENTS



• 오픈소스SW의 개요 | 06



• 오픈소스SW의 지적재산권과 라이선스 | 10

1. SW 지적재산권 | 10
2. 라이선스와 오픈소스SW | 11
3. 오픈소스SW 라이선스의 이해와 활용 | 13



• 오픈소스SW 라이선스의 구체적 내용 | 16

1. 공통적 준수사항 | 16
2. 라이선스별 준수사항 | 18
3. 주요 쟁점 | 29
4. 주요 오픈소스SW 사례 | 32
5. 오픈소스SW 라이선스 위반사례 | 35

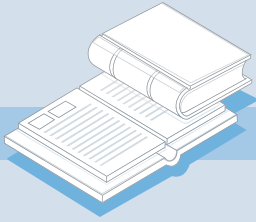


• 기업에서의 오픈소스SW 라이선스 관리/활용 방안 | 38

1. 오픈소스SW 관련 정책의 수립 | 38
2. 오픈소스SW 라이선스 관리를 위한 프로세스 및 조직의 구축 | 40

참고  
자료

- 주요 오픈소스SW와 라이선스 | 48



Korea Software Copyright Committee



오픈소스SW 라이선스 가이드

오픈소스SW의 개요

## I

## 오픈소스SW의 개요

**오픈소스SW**는 소스코드가 공개되어 있는 SW를 말하며, 일반적으로 자유롭게 사용·복제·배포·수정할 수 있다. 오픈소스SW의 대표적인 예로는 Linux 커널 및 관련 GNU SW, 아파치 웹서버, FireFox 웹브라우저, MySQL 데이터베이스시스템, Python/PHP/Perl 언어, Eclipse 툴 등을 들 수 있으며, 그 외에도 많은 오픈소스SW들이 전세계에 걸쳐 수많은 개발자들에 의해 개발되고 있다.

전세계적으로 오픈소스SW는 FSF(Free Software Foundation)<sup>1)</sup>의 자유SW(Free Software)를 포함한 넓은 의미로 사용되고 있다.<sup>2)</sup> 하지만 자유SW와 오픈소스SW는 역사 및 추구하는 이념 등에서 미묘한 차이가 있다.

1980년대에 들어 PC가 널리 보급되기 시작하면서 이전에는 하드웨어의 부속물로만 간주되던 SW가 거대한 부가가치 산업으로 발전하기 시작하였고, 지적재산권 및 라이선스 계약을 통하여 SW의 사용·복제·배포·수정에 일정한 제한을 가하려는 움직임이 나타나게 된다. SW를 둘러싼 이러한 일련의 흐름에 반대하고 SW의 자유로운 사용·복제·배포·수정에 대한 기존의 권리를 지키기 위하여 리처드 스톨만은 FSF를 설립하고 자유SW(FreeSoftware) 운동을 전개하였다.




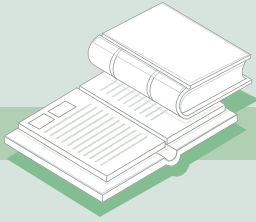
# Open Source SW Licence Guide

1990년대 들어서면서 인터넷과 더불어 GNU<sup>1)</sup> GPL(General Public License)로 배포된 리눅스가 널리 보급되기 시작하였고, MS의 익스플로러에 밀려 어려움을 겪고 있던 Netscape 사가 웹브라우저의 소스코드를 공개하는 결정을 내렸으며, IBM, Sun 등이 자유SW에 대한 지원을 시작하였다. 그러나 자유SW의 '자유(Free)' 라는 단어가 일반인들에게 '무료' 라는 의미로 인식되고 있다는 점, 엄격한 GPL조항 때문에 상용SW 개발에 이용할 수 없어 대다수 기업들이 자유SW운동에 참여를 꺼려한다는 점 등을 고려하여, 소스코드 공개에 보다 많은 참여를 이끌어내기 위하여 에릭 레이먼드, 브루스 페런스 등은 '오픈소스(OpenSource)' 라는 새로운 용어를 제안하였다. 그리고 이러한 '오픈소스' 는 1998년 오픈소스SW 활성화 및 오픈소스SW에 대한 인증을 담당하는 OSI(Open Source Initiative)가 결성되면서 널리 사용되게 되었다. OSI에서는 오픈소스에 해당하는 다양한 종류의 라이선스가 지켜야 할 최소한의 기준을 정의(Open Source Definition, OSD)해 놓고 이 정의의 규정에 따라 인증, 관리 및 촉진시키는 일을 한다. OSI가 인증한 오픈소스SW 라이선스에는 OSI의 인증마크<sup>5)</sup>를 부여한다.



- 1) FSF는 자유SW의 개발과 보급을 위해 리처드 스톨만(Richard Stallman)이 1984년 설립한 비영리 민간단체다. 컴퓨터 프로그램의 복제·배포·개작의 자유와 이를 위한 소스코드의 사용에 대한 제한 철폐 등을 목적으로 하며, 본부는 미국 보스턴에 있다. 출처 : 네이버백과사전 (<http://100.naver.com/100.nhn?docid=784184>)
- 2) 국내에서는 자유SW를 포함한 오픈소스SW를 '공개SW'로 번역하여 사용하고 있다.
- 3) 자유SW는 리처드 스톨만(Richard Stallman)과 FSF에 의해 만들어진 개념으로서, SW의 이용자에게 해당 SW를 실행·복제·배포할 수 있는 자유, 그리고 소스코드에 대한 접근을 통해서 이를 학습·수정·개선시킬 수 있는 자유를 부여하는 SW이다.
- 4) GNU는 Gnu is Not Unix - GNU는 유닉스가 아니다 라는 뜻의 재귀적 약어이다.

5) OSI 인증마크 : 



Korea Software Copyright Committee



오픈소스SW 라이선스 가이드



오픈소스SW의  
지적재산권과 라이선스

- 
1. SW 지적재산권
  2. 라이선스와 오픈소스SW
  3. 오픈소스SW 라이선스의 이해와 활용
-

## II

## 오픈소스SW의 지적재산권과 라이선스

## 2.1 SW 지적재산권

현재 SW는 다음과 같이 저작권, 특허권, 상표권, 영업비밀 등의 지적재산권에 의해 보호받고 있다.

## ● 저작권

저작권(copyright)은 문학·학술 또는 예술의 범위에 속하는 창작물(저작물)의 창작에 의하여 그 창작물에 대하여 창작자(저작자)가 취득하는 권리로서 창작물의 아이디어를 보호하는 것이 아니라 그 표현(expression)의 결과물을 보호하는 것이다. 저작권은 권리의 발생에 있어 등록과 같은 요건이 필요하지 않고 창작과 동시에 권리가 발생한다(무방식주의). 따라서 어떤 프로그래머가 특정 SW를 개발하게 되면 컴퓨터프로그램저작권이 자동적으로 발생하며, 그 권리는 프로그래머 또는 그가 속한 회사에 부여된다. 저작권이 있는 저작물의 경우 누구도 원 저작자나 저작권자의 허가가 없이는 해당 저작물을 사용·복제·배포·수정할 수 없다.

## ● 특허권

특허권(patent)은 발명에 관하여 발생하는 독점적·배타적 지배권으로 법에 정해진 절차에 의해 출원을 하여야 하며, 심사를 통해 부여되는 권리이다. 특허기술을 사용하기 위해서는 반드시 특허권자의 허락을 얻어야만 한다. 특허권자는 자신이 허락하지 않은 사람이 해당 특허를 활용한 제품을 만들거나, 사용하거나, 판매하는 것을 막을 수 있다. 특허는 무엇인가 유용하게 하는 방식(method)이므로 특허받은 방식을 구현하는 SW라면 프로그래밍 언어가 다르거나 소스코드가 다르더라도 해당 특허권자의 명시적인 허락을 받아야 하며 이는 오픈소스SW 뿐만 아니라 상용 SW에도 마찬가지이다.

## ● 상표권

상표권(trademark right)이란 상표권자가 지정상품에 관하여 그 등록상표를 사용할 독점적인 권리로서 일정한 절차에 따라 등록하여야 효력이 발생한다. 이러한 상표를 사용하기 위해서는 반드시 상표권자의 허락을 얻어야 하며 허락받지 않고 상표를 이용할 경우 처벌을 받게 된다. 상표권을 취득한 SW의 경우 상표를 사용하려면 상표권자의 명시적인 허락을 받아야 한다.

# Open Source SW Licence Guide



## ● 영업비밀

영업비밀(trade secret)이란 공연히 알려져 있지 아니하고 독립된 경제적 가치를 지니는 것으로서 상당한 노력에 의하여 비밀로 유지되는 생산 방법, 판매 방법, 기타 영업 활동에 유용한 비밀유지의무가 있음에도 다른 사람에게 이를 누출하는 경우 처벌받게 된다. 공개되지 않은 SW의 경우 영업비밀로서 보호를 받을 수 있으며, 공개된 SW라 하더라도 아이디어에 대한 부분은 영업비밀로 보호를 받을 수 있는 가능성이 있다. 단, 영업비밀로서의 SW보호는 널리 공개되어 유통되는 경우에는 보호받기가 어렵고, 이를 알지 못하고 사용한 제3자에게 법적으로 문제를 삼을 수 없는 한계가 있다.

이상과 같은 지적재산권에 의해 권리자는 SW에 대한 배타적인 권리를 가지게 되며, 원칙적으로 권리자만이 SW를 사용·복제·배포·수정할 수 있다.

## 2.2 라이선스와 오픈소스SW

### 2.2.1 라이선스의 의미

SW는 지적재산권에 의해 자신이 만든 SW를 다른 사람이 사용하지 못하게 하고 자신만이 사용할 수 있는 권리를 가지게 되며, 원칙적으로 이러한 권리자만이 SW를 사용·복제·배포·수정할 수 있다. 하지만 다양한 필요에 의해 이들 권리자가 다른 사람에게 일정한 내용을 조건으로 하여 특정 행위를 할 수 있는 권한을 부여할 필요가 있는데, 이와 같은 권한을 보통 '라이선스(license, 사용허가권)'라고 한다. 예를 들면 우리가 윈도우즈XP를 구입하여 사용하게 되면 이것은 윈도우즈XP 프로그램 전체를 사서 내 마음대로 이용할 수 있는 것이 아니라 SW 권리자인 마이크로소프트와 윈도우즈XP를 한 대의 컴퓨터에 설치하여 사용할 수 있는 라이선스를 받은 것이다. 그러므로 우리가 윈도우즈XP를 구입했다고 해서 이를 다른 사람에게 빌려주거나 이를 복제하여 팔 수 없다. 만약 이를 어기고 윈도우즈XP를 복제하거나 수정하거나 배포하는 등의 경우에는 해당 계약(라이선스)을 위반한 것이 되며 저작권 침해로 인해서 처벌을 받게 된다. 이러한 의미에서 라이선스는 물건을 판매하는 매매와는 차이가 있다.

## 2.2.2 오픈소스SW 라이선스

오픈소스SW 라이선스란 오픈소스SW 개발자와 이용자간에 사용 방법 및 조건의 범위를 명시한 계약을 말한다. 따라서 오픈소스SW를 이용하기 위해서는 오픈소스SW 개발자가 만들어놓은 사용 방법 및 조건의 범위에 따라 해당 SW를 사용해야 하며, 이를 위반할 경우에는 라이선스를 위반함과 동시에 저작권 침해로 인해서 이에 대한 처벌을 받게 된다. 대표적인 라이선스로는 GPL(General Public License), LGPL(Lesser General Public License), BSD(Berkeley Software Distribution), MPL(Mozilla Public License) 등의 라이선스가 있으며, 이런 오픈소스 SW 라이선스는 기본적으로 사용자의 자유로운 사용·수정·배포를 보장하고 있다. 오픈소스 SW가 이와 같은 라이선스를 만들어서 운영하는 이유는 만약 오픈소스SW에 라이선스가 없다면 오픈소스SW를 무단으로 상업적인 SW를 개발하는데 사용하고 해당 소스코드를 공개하지 않을 수 있기 때문에 법의 테두리안에서 소스코드의 공개를 강제할 수 있도록 하기 위한 것이다.

2007년 10월 현재 오픈소스SW 라이선스의 인증을 관장하고 있는 OSI에 따르면 64개의 라이선스가 오픈소스 라이선스로 인정되어 등록되어 있다. 하지만 실제로 많이 사용되는 라이선스의 개수는 한정되어 있다. 오픈소스 프로젝트 개발 포털사이트인 Freshmeat(<http://freshmeat.net>)에 등록되어 있는 프로젝트들 중에서 오픈소스로 분류되는 약 43,722개 프로젝트가 진행 중에 있다. Freshmeat에 있는 “오픈소스SW 라이선스 분포”를 살펴보면 이런 프로젝트 중 약 72%에 해당하는 프로젝트가 GPL과 LGPL 라이선스를 사용하고 있으며, 그 뒤를 BSD가 차지하고 있음을 알 수 있다. 참고로 Mozilla 웹브라우저, Apache 웹서버 등 일부 중요한 오픈소스 SW들은 자체적인 라이선스를 가지고 있으므로 이를 사용할 때는 해당 라이선스에 대한 이해가 필수적이다. 아래표는 Freshmeat에서 진행중인 프로젝트에 사용된 오픈소스SW 라이선스 비율을 나타낸 것이다.

오픈소스SW 라이선스 분포표

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%
Freeware	1,079	2.30%
Freely Distributable	980	2.09%

\* <http://freshmeat.net/stats/#license> 오픈소스SW 라이선스 분포표

## 2.3 오픈소스SW 라이선스의 이해와 활용

오픈소스SW는 독점SW(proprietary software)와 동일하게 저작권 등에 의한 법적 보호를 받고 있으며, 이와 같은 권리에 기반하여 오픈소스SW 저작권자는 오픈소스SW 이용자에게 라이선스를 부여한다. 그러나 오픈소스SW 라이선스는 일반적인 독점SW 라이선스와는 많은 점에서 차이가 있다. 이를 살펴보면 아래와 같다.

기본적으로 오픈소스SW 라이선스는 다음과 같이 사용자의 자유로운 사용·복제·배포·수정을 보장하고 있다.



- 라이선시<sup>6)</sup>는 해당 오픈소스SW를 자유롭게 사용할 수 있다.
- 라이선시는 해당 오픈소스SW를 자유롭게 복제할 수 있으며, 일정한 조건하에 재배포할 수 있다.
- 라이선시는 해당 오픈소스SW를 자유롭게 수정하여 사용할 수 있으며, 일정한 조건하에 수정된 내용을 재배포할 수 있다.
- 라이선시는 해당 오픈소스SW의 소스코드를 자유롭게 획득하고 접근할 수 있다.

오픈소스SW 라이선스는 또한 SW의 사용자에게 일정한 의무를 부과하고 있다. 구체적인 내용은 오픈소스SW와 함께 배포되는 라이선스의 내용을 통해 알 수 있다. 해당 오픈소스SW에 대한 라이선스는 주로 소스코드 내부나 홈페이지 등에 명시되어 있다. 소스코드에서는 주로 최상위 디렉토리에 'COPYING' 이라는 독립된 파일에 라이선스 조항을 기록하기도 하며, 각각의 소스코드 파일 상단에 명시해 두기도 한다.

오픈소스SW 라이선스에서 요구하고 있는 준수사항을 오픈소스SW 이용자가 이행하지 않으면 권리자로부터 저작권 위반(또는 계약 위반)으로 소송을 제기 당할 수 있다. 만약 권리를 침해한 것으로 결론이 내려지면 SW의 배포가 더 이상 불가능할 뿐만 아니라 이미 배포한 SW에 대한 손해배상 등 막대한 책임을 부담할 수 있다. 특히 임베디드 SW의 경우 이를 내장한 제품까지 판매하지 못하거나 리콜(Recall)을 해야 하는 경우도 발생할 수 있으므로 라이선스의 의무사항



6) 라이선시(Licensee)는 라이선스를 받는 자이고, 라이선서(Licenser)는 라이선스를 부여하는 자이다.

을 명확히 이해하여 이와 같은 상황을 사전에 예방하는 것이 필수적이다. 그러나 이러한 위험 때문에 오픈소스SW를 전혀 사용하지 않겠다는 결론을 내릴 필요는 없다. 독점SW 라이선스에서 규정하고 있는 의무사항에 비하면 오픈소스SW 라이선스가 요구하고 있는 내용이 결코 어려운 것이 아니므로, 오히려 이를 잘 이해하고 준수한다면 독점SW를 사용하는 것보다 훨씬 비용을 절감할 수 있으므로 오픈소스SW의 장점을 적극 활용할 필요가 있다. 또한 몇몇 라이선스만이 독자 개발한 소스코드의 공개를 요구하고 있기 때문에 이를 잘 분석한 후 사용한다면 문제 발생 소지는 거의 없다고 봐야할 것이다.

따라서 오픈소스SW를 다운로드받아 개발에 적용할 때는 반드시 라이선스의 요구 사항을 확인하여야 한다. 자체 판단이 불가능할 경우에는 외부 전문가에게 조언을 의뢰하여 개발 시작 전 해당 라이선스의 요구 사항과 오픈소스SW의 사용목적에 확실히 분석하여야 한다. 이렇게 하는 것만으로도 충분히 올바르게 오픈소스SW를 최대한 활용할 수 있으며, 나중에 발생할 수 있는 문제들을 사전에 차단할 수 있다.

오픈소스SW 라이선스 가이드



오픈소스SW 라이선스의  
구체적 내용

- 
1. 공통적 준수사항
  2. 라이선스별 준수사항
  3. 주요 쟁점
  4. 주요 오픈소스SW 사례
  5. 오픈소스SW 라이선스 위반사례
-

## III

오픈소스SW 라이선스의  
구체적 내용

## 3.1 공통적 준수사항

오픈소스SW 라이선스의 의무사항은 각각의 라이선스마다 조금씩 차이가 있지만 크게 나누어 보면 공통적으로 ‘저작권 관련 문구 유지’, ‘제품명 중복 방지’, ‘서로 다른 라이선스의 SW 조합 시 조합 가능 여부 확인’ 등이 있고, 선택적으로는 ‘소스코드 공개’, ‘특허관련 사항 준수’ 등이 있다.

## 3.1.1 공통적 준수사항

아래는 모든 오픈소스SW에 공통적으로 적용되는 조건들로 반드시 지켜야할 사항이다.

## ● 저작권 관련 문구 유지

저작권이란 표현된 결과물에 대해 발생하는 권리이며 저작물의 창작과 함께 자동적으로 부여된다. SW의 경우는 소스코드에 프로그램의 이름과 개발자, 버전, 연락처 등을 포함하고 있는 경우가 많으며 이러한 것들은 성명표시권, 동일성 유지권 등의 저작인격권으로 보호를 받는다. 오픈소스SW의 경우 거의 대부분의 프로그램 소스코드 상단에 개발자 정보와 연락처 등이 기록되어 있으며 이러한 정보는 저작권의 보호를 받기 때문에 개발자 정보를 임의로 수정하거나 삭제하여서는 안된다. 특히 GPL 등 수정된 결과물을 다시 공개하도록 규정하고 있는 ‘상호주의(reciprocal)’ 라이선스들의 경우 만약 소스코드 상에 개발자 정보가 수정·삭제된 채로 외부에 소스코드를 공개하였다가 그 사실이 밝혀질 경우 저작권 침해문제가 발생할 수 있으므로 주의하여야 한다. 저작권 관련 문구 유지는 쉽게 판단이 가능한 사항이므로 항상 준수하여야 한다.

## ● 제품명 중복 방지

SW의 제품명은 상표권에 의해 보호받는다. 따라서 오픈소스SW의 경우에도 이와 동일한 이름을 제품명이나 서비스명으로 사용하면 상표권 침해의 문제가 생기게 된다. 특히 유명한 오픈소스 SW일수록 해당 오픈소스SW의 이름이 상표로서 등록되어 있는 경우가 많기 때문에(예: 리눅스) 더욱 조심하여야 한다. 다만 이러한 제품명 및 서비스명에 대한 결정이 개발자들에 의해 이루어지는 경우는 많지 않으므로 역시 상식적인 수준에서 판단하면 될 것이다.





# Open Source SW Licence Guide

## ● 서로 다른 라이선스의 조합

SW를 작성하고자 할 경우 기존에 만들어진 코드를 재사용하거나 결합하는 경우가 많은데, 이러한 경우 결합되는 각 코드의 라이선스가 서로 상충되는 경우가 있다. 예컨대 MPL 조건의 A코드와 GPL 조건의 B코드를 결합하여 'A+B' 라는 프로그램을 만들어 배포하고자 하는 경우, MPL은 'A+B'의 A부분을 공개하고 MPL로 배포할 것을 요구하는 반면, GPL은 'A+B' 전체를 공개하고 GPL로 배포할 것을 요구하기 때문에 'A+B' 프로그램을 배포하는 것은 불가능하게 된다. 이러한 문제를 가르켜 라이선스의 양립성(Compatibility) 문제라고 한다. 따라서 서로 다른 라이선스로 배포된 오픈소스SW를 결합하는 경우 반드시 두개의 라이선스가 서로 호환되는지를 확인하여야 한다.

## 3.1.2 선택적 준수사항

아래의 준수사항은 라이선스에 따라 다르다. 어떤 라이선스의 경우는 아래 세가지 사항 모두에 관계되는 경우도 있고, 어떤 라이선스는 아래 중 일부만을 요구하는 경우도 있다. 자세한 사항은 라이선스별 준수사항 부분(3.2)을 참고하기 바란다.

## ● 사용 여부 명시

많은 오픈소스SW 라이선스들은 소스코드를 자유롭게 열람하고 수정 및 재배포할 수 있는 권리를 부여하는 한편, SW를 사용할 때 해당 오픈소스SW가 사용되었음을 명시적으로 표기하는 것을 의무사항으로 채택하고 있다. 이것은 마치 논문을 쓸 때 인용을 하는 것과 비슷하여, '이 SW는 오픈소스SW인 무엇 무엇을 사용하였습니다.' 라는 식으로 사용 여부를 명확히 기술하라는 것이다. 사용자 매뉴얼이나 기타 매뉴얼을 대체하는 매체가 있다면 그곳에 기술하면 된다.

## ● 소스코드 공개

오픈소스SW는 라이선스에 따라서 수정하거나 추가한 부분이 있을 때 해당 부분의 소스코드도 공개하여야 한다고 명시하는 경우가 있다. 대표적인 라이선스로 GPL이 있다. 그러나 정확한 공개 범위는 각각의 라이선스에서 정하고 있는 범위와 SW를 개발하는 방법에 따라서도 달라질 수 있다.

● 특허

특허에 대한 기본적인 내용은 만약 어떤 기술이 특허로 보호될 경우 해당 기술을 구현할 때 반드시 특허권자의 허락을 받아야 한다는 것이다. 이러한 조건은 오픈소스SW의 여부와 상관 없이 모든 SW에 공통적으로 해당된다. 그러나 어떤 특허를 오픈소스SW로 구현할 경우 해당 특허의 구현 결과는 오픈소스SW 라이선스를 따라야만 하는 문제 등 오픈소스SW와 관련된 특허권의 문제는 보다 복잡하게 전개된다. 특히 최근 SW특허가 급격히 증가하면서 이와 관련된 문제가 심각해지고 있기 때문에 새롭게 만들어지는 오픈소스SW 라이선스들에서는 특허관련 조항을 포함하고 있는 경우가 많아지고 있다.

## 3.2 라이선스별 준수사항

이제 주요 라이선스 별로 주요내용과 저작권 표시 등 의무사항, 소스코드의 공개범위에 대해 살펴보기로 하자.

### 3.2.1 GPL 2.0

#### 1) 주요내용

GPL 2.0은 현재 가장 많은 오픈소스SW가 채택하고 있는 라이선스이다. 오픈소스SW 라이선스들 중에서 가장 잘 알려져 있고 의무사항들도 다른 오픈소스SW 라이선스에 비해 엄격한 편이다. GPL 2.0의 주요 내용은 다음과 같다.



- SW를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시 및 GPL에 의해 배포된다는 사실을 명시
- SW를 수정하거나 새로운 SW를 링크(Static과 Dynamic linking 모두)시키는 경우 GPL에 의해 소스코드를 제공해야 함
- Object Code 또는 Executable Form으로 GPL SW를 배포하는 경우, 소스코드 그 자체를 함께 배포하거나 또는 소스코드를 제공받을 수 있는 방법에 대한 정보를 함께 제공해야 함
- 자신의 특허를 구현한 프로그램을 GPL로 배포하는 경우에는 그 프로그램을 GPL 조건에 따라 이용하는 이용자에게 특허에 대한 사용료를 받을 수 없으며, 제3자의 특허를 구현한 프로그램인 경우에는 그 특허권자가 GPL 조건에 따라 이용하는 프로그램 이용자에 대하여 특허 사용료를 받지 않을 때에만 그 프로그램을 GPL로 배포하는 것이 가능

## 2) 저작권 표시 등 의무사항

GPL 2.0 SW를 사용하였을 경우 “본 제품(SW)은 GPL 라이선스 하에 배포되는 SW인 ○○○ (사용한 GPL SW 이름)를 포함합니다”와 같은 문구를 매뉴얼 혹은 그에 준하는 매체에 포함시키고, GPL 전문을 첨부해야 한다. 소스코드를 공개하기 위해서는 소스코드를 CD-ROM 등의 매체에 담아서 제품 판매시 함께 배포하거나, 매뉴얼에 소스코드를 요청할 수 있는 연락처를 기입하여 두거나, 혹은 FTP 서버, 웹서버 등에 소스코드를 업로드해 두고 매뉴얼에 해당 주소를 기입하면 된다.

GPL 2.0 라이선스를 보면 GPL 2.0으로 배포하는 경우 소스코드의 시작 부분에 아래와 같은 양식으로 기재하도록 하였다. 각각의 파일에는 최소한 저작권을 명시한 행과 본 사용 허가서의 전체 내용을 참고할 수 있는 위치 정보를 명시해야 합니다.

### (프로그램에 대한 이름과 용도를 간단하게 설명)

ex) Define ISO C stdio on top of C++ iostreams.

### (Copyright (C) 20yy년 <저작권자 이름>)

ex) Copyright (C) 1991, 1994-1999, 2000, 2001 Free Software Foundation, Inc.

이 프로그램은 자유소프트웨어이다. 소프트웨어의 피양도자는 자유 소프트웨어 재단이 공표한 GNU 일반 공중 사용 허가서 2판 또는 그 이후 판을 임의로 선택해서, 그 규정에 따라 프로그램을 개작하거나 재배포할 수 있다.

이 프로그램은 유용하게 사용될 수 있으리라는 희망에서 배포되고 있지만, 특정한 목적에 맞는 적합성 여부나 판매용으로 사용할 수 있으리라는 묵시적인 보증을 포함한 어떠한 형태의 보증도 제공하지 않는다. 보다 자세한 사항에 대해서는 GNU 일반 공중 사용 허가서를 참고하기 바란다.

GNU 일반 공중 사용 허가서는 이 프로그램과 함께 제공된다. 만약, 이 문서가 누락되어 있다면 자유 소프트웨어 재단으로 문의하기 바란다. (자유 SW 재단: Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA)

## 3) 공개의 범위

GPL 2.0의 경우, GPL 프로그램의 소스코드를 개발자가 개발중인 프로그램 코드에 삽입하거나 링크시켜 이를 배포하는 경우에 개발한 프로그램의 소스코드도 공개해야 한다. 다만, 원본 프로그램과 별개의 독립된 프로그램은 GPL 2.0 제2조 후단<sup>7)</sup>에 따라 GPL 프로그램과 단순히 동일한 매체에 저장하여 배포하는 경우 GPL 2.0이 아닌 다른 라이선스 조건에 의해 배포할 수

있다. 하지만 구체적으로 어떠한 경우가 파생물에 해당하는지 또는 독립된 프로그램의 단순한 집합물에 해당하는지를 구별하는 것은 쉽지 않다. 결국 이 문제는 최종적으로는 법원의 판단에 의해 결정될 문제이지만, FSF는 이러한 혼란을 막기 위해 GPL FAQ을 통하여 몇 가지의 구별 기준을 제시하고 있다. 그 기준의 예는 다음과 같다.

- 두개의 모듈이 동일한 실행파일에 포함되어 있거나 공유주소영역(shared address space)에서 링크되어 실행되도록 설계된 경우에는 원 프로그램에서 파생된 프로그램이고, 2개의 프로그램이 파이프(pipes), 소켓(sockets), command-line arguments 형태로 통신하는 경우에는 독립된 프로그램이다.
- 플러그인(plug-ins)의 경우 동적으로 링크되어 함수호출을 하고 데이터구조를 공유하는 경우에는 파생된 프로그램에, fork와 exec를 이용하면 독립된 프로그램에 해당한다.
- 인터프리터(interpreter)나 컴파일러(compiler)의 경우에는 원칙적으로 컴파일된 결과물과는 분리된 것으로 보지만, 컴파일과정에서 라이브러리나 클래스가 결과물에 추가된 경우에는 라이브러리 또는 클래스와 결과물은 하나의 프로그램으로 볼 수 있다.

한편 원칙적으로 GPL 2.0 조건으로 배포하면서도 GPL 2.0 제2조와 관련해서는 다소 완화된 내용의 조건으로 프로그램을 배포하는 경우가 있다. 대표적 사례로 GNU Classpath 프로젝트와 자바(Java) 플랫폼 사례가 있다. GNU Classpath 프로젝트는 자바(java)언어의 가상머신(virtual machines) 및 컴파일러에서 사용되는 핵심 클래스라이브러리(core class libraries)를 자유SW로 대체하기 위한 프로젝트인데, 동 프로젝트의 결과물을 GPL 2.0으로 배포하면서도 이와 링크된 다른 독립된 SW는 GPL 2.0으로 배포할 필요가 없다는 내용의 예외를 인정하였다. 그런데 2006년 말 Sun이 향후 자바 플랫폼을 GPL 2.0 조건으로 배포하겠다는 선언을 하면서, 자바 플랫폼 중 특히 Java SE(Java Platform Standard Edition)와 Java EE(Java Platform Enterprise Edition)의 GPL 2.0 배포조건에 Classpath 예외를 추가한다고 발표하였다. 그 결과 Classpath 예외조항을 포함한 GPL 2.0 조건의 자바 플랫폼을 이용한 응용프로그램도 소스 코드를 공개하지 않고 배포할 수 있다.



7) '원 프로그램으로부터 파생되지 않고 그 자체로 독립적이고 분리된 저작물(separate works)은 다른 라이선스 조건에 의해 배포가 가능하며, 단순집합물(mere aggregation)로서 원 프로그램과 동일한 매체로 배포할 수 있다'고 규정하고 있다.

## 3.2.2 GPL 3.0

### 1) 주요내용

지난 2007년 6월 29일 공식적으로 GPL 3.0을 발표하였다. GPL 3.0의 전체적인 체계를 보면, 서문을 제외하고 제0조부터 제17조까지 총 18개 조문으로 구성되어 있다. 이중 제4조부터 제6조까지, 제8조부터 제10조까지, 제12조, 제14조부터 제17조까지는 기존의 GPL 2.0의 내용을 적절히 수정해서 재구성한 것이다. 새롭게 추가된 내용으로는 제0조와 제1조에서 각종 용어를 새로이 도입하거나 기존의 용어를 수정하였으며, 제3조를 중심으로 DRM과 관련된 내용이 추가되었다. 또한 오픈소스SW 라이선스의 급격한 증가와 양립성 문제를 완화하고자 제7조에서 GPL에 부가적인 조건을 추가할 수 있도록 규정하고 있다. 아울러 제11조 등은 SW특허문제, 제13조에서는 Affero GPL과의 양립성 문제에 대처하고자 새롭게 추가된 내용이다. 기본적인 내용은 GPL 2.0과 동일하지만 GPL 3.0에서는 몇가지 추가되거나 변경된 부분이 있다. 이에 대한 주요 내용은 다음과 같다.



- GPL 3.0의 소스코드를 특정한 제품에 포함시키거나 혹은 그와 함께 배포하는 경우에는 해당 소스에 설치 정보(Installation Information)<sup>8)</sup>를 함께 제공해야 함. 다만 SW가 롬(ROM)에 설치된 경우처럼, 해당제품의 제조업체나 여타 제3자도 수정된 코드를 제품에 설치할 수 없는 경우에는 설치정보를 제공하지 않아도 됨
- DRM(Digital Rights Management)<sup>9)</sup>과 관련하여 각국의 법률에 의해 보호되는 이익을 포기해야 함
- 특허와 관련해서 원래의 소스코드를 개선하여 배포한 기여자의 경우 자신이 기여한 부분에 대해서는 비차별적이고 특허 사용료가 없다는 내용의 라이선스를 제공해야 함
- 특허와 관련해서 라이선시 등으로부터 특허소송이 제기되는 경우 소송을 제기한 날에 특허소송을 제기한 라이선시의 오픈소스SW 라이선스는 종료됨
- Apache License 2.0 및 Affero GPL과 양립 가능함

### 2) 저작권 표시 등 의무사항 및 공개의 범위

의무사항은 GPL 2.0과 동일하다. GPL 2.0을 참고하기 바란다.



8) 설치정보란 SW를 수정하여 해당 제품에 설치하고 실행하는 데 필요한 방법(methods), 절차(procedures), 인증키(authorization keys) 혹은 여타 정보 모두를 의미한다.  
 9) DRM이란 디지털 콘텐츠의 저작권을 보호하기 위한 기술 및 서비스를 말한다.

### 3.2.3 LGPL 2.1

#### 1) 주요내용

LGPL(Lesser General Public License)은 FSF가 일부 라이브러리(Library)<sup>10)</sup>에 대하여 GPL 보다 소스코드의 공개정도를 다소 완화된 형태로 사용할 수 있도록 만든 라이선스이다. FSF에서 LGPL를 만들어 사용하고 있는 이유는 오픈소스SW의 사용을 장려하기 위한 전략적인 차원에서이다. 즉, 상용 라이브러리와 동일한 기능을 제공하는 오픈소스 라이브러리에 GPL과 같은 엄격한 라이선스를 적용하게 되면, 라이브러리를 사용하는 SW의 소스코드를 공개해야하기 때문에 상용SW 개발자들은 오픈소스 라이브러리의 사용을 꺼려할 것이다. 오히려 이미 널리 사용되고 있는 상용 라이브러리와 동일한 기능을 제공하는 오픈소스 라이브러리를 LGPL로 배포하여 원 프로그램의 소스코드는 공개하지 않고 이에 사용된 해당 오픈소스 라이브러리의 소스코드만 공개하게 함으로써 오픈소스 라이브러리의 사용을 장려하고 사실상의 표준으로 유도하는 한편, 관련된 다른 오픈소스SW를 보다 더 많이 사용할 수 있도록 하겠다는 것이 FSF의 전략이다. LGPL은 ‘Library’ General Public License였으나 ‘Library’란 단어가 라이선스 이름에 포함되어 개발자들이 모든 라이브러리를 위한 라이선스로 오인하는 경향이 있었다. 이러한 오해는 일부 한정된 라이브러리에 대해서만 LGPL을 사용하려는 것이 FSF의 의도와는 맞지 않아 LGPL 버전 2.1에서는 결국 이러한 오인을 방지하기 위하여 ‘Library’를 ‘Lesser’로 수정하였다.

LGPL의 주요 내용을 요약하면 다음과 같다.



- SW를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시 및 LGPL에 의해 배포된다는 사실을 명시
- LGPL 라이브러리의 일부를 수정하는 경우 수정한 라이브러리의 소스코드 공개
- LGPL 라이브러리에 응용프로그램을 링크시킴(Static과 Dynamic Linking 모두) 경우 해당 응용프로그램의 소스를 공개할 필요 없음. 다만 사용자가 라이브러리 수정 후 동일한 실행 파일을 생성할 수 있도록 Static Linking시에는 응용프로그램의 Object Code를 제공해야 함
- 특허의 경우 GPL과 동일함



10) 라이브러리는 다른 프로그램들과 링크되기 위하여 존재하는 하나 이상의 서브루틴이나 함수들이 저장된 파일들의 모음을 말하는데, 함께 링크될 수 있도록 보통 컴파일된 형태인 목적코드(object code) 형태로 존재한다. 출처 : 김동근의 텀즈(<http://www.terms.co.kr/library.htm>)

## 2) 저작권 표시 등 의무사항

GPL과 마찬가지로 LGPL로 배포되는 라이브러리를 수정하는 경우에는 이를 다시 LGPL로 공개해야 한다. LGPL 라이브러리를 바탕으로 수정된 라이브러리를 만든 경우에는 파일을 수정할 경우 수정했다는 사실과 수정일자를 명시하여야 하고 라이브러리 전체를 다시 LGPL로 공개해야 한다. 또한 라이브러리에 포함된 모든 함수는 프로그램의 도움없이도 수행 가능해야 한다. 아래는 LGPL 라이선스 예이다.

**(라이브러리의 이름과 용도를 한 줄 정도로 설명)**  
 ex) Define ISO C stdio on top of C++ iostreams.  
 This file is part of the GNU C Library.

**Copyright (C) 20yy년 <프로그램 저작자의 이름>**  
 ex) Copyright (C) 1991,1994-2002,2003 Free Software Foundation, Inc.

이 라이브러리는 자유 소프트웨어이다. 소프트웨어의 피양도자는 자유 소프트웨어 재단이 공표한 GNU 약소 일반 공중 사용 허가서 2.1판 또는 그 이후 판을 임의로 선택해서, 그 규정에 따라 라이브러리를 개작하거나 재배포할 수 있다.

이 라이브러리는 유용하게 사용될 수 있으리라는 희망에서 배포되고 있지만, 특정한 목적에 맞는 적합성 여부나 판매용으로 사용할 수 있으리라는 묵시적인 보증을 포함한 어떠한 형태의 보증도 제공하지 않는다. 보다 자세한 사항에 대해서는 GNU 약소 일반 공중 사용 허가서를 참고하기 바란다.

GNU 약소 일반 공중 사용 허가서는 이 라이브러리와 함께 제공된다. 만약, 이 문서가 누락되어 있다면 자유 소프트웨어 재단으로 문의하기 바란다. (자유 SW 재단: Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA)

## 3) 공개의 범위

LGPL은 링크하는 SW의 소스코드를 공개할 필요가 없다는 점이 GPL과 가장 큰 차이점이다. 어떠한 경우에도 LGPL SW 자체는 공개해야 하지만 LGPL SW와 링크되는 부분의 SW 소스코드는 공개해야 할 의무가 발생하지 않으므로 기업의 입장에서는 LGPL SW를 좀 더 선호하게 된다. 사용 여부 명시 등은 GPL과 동일하게 반영하면 되고 공개해야 할 소스코드의 공개 역시 GPL과 동일한 방식을 이용하면 된다.

LGPL은 일정한 요건을 충족시키는 경우 LGPL 라이브러리를 이용하는 프로그램, 다시 말해 링킹(linking)을 통해 LGPL 라이브러리와 함께 작동하도록 설계된 프로그램을 배포할 경우에는 소스코드를 제공하지 않아도 된다.

### 3.2.4 BSD License

#### 1) 주요내용

BSD(Berkeley Software Distribution) 라이선스는 SW의 소스코드를 공개하지 않아도 되는 대표적인 오픈소스SW 라이선스 중 하나이다. 이렇게 BSD 라이선스는 소스코드를 공개하지 않아도 되기 때문에 GPL 및 LGPL과 비교하여 덜 제한적으로 사용할 수 있으며 허용범위가 넓다. 이렇게 BSD 라이선스의 허용범위가 넓은 이유는 BSD 라이선스로 배포되는 프로젝트가 미국 정부에서 제공한 재원으로 운영되었기 때문이다. 즉, SW에 대한 대가를 미국 국민의 세금으로 미리 지불했기 때문에 사람들에게 그들이 원하는 방식으로 SW를 사용하거나 만들 수 있도록 허가된 것이다. 따라서 BSD 라이선스의 소스코드를 이용하여 새로운 프로그램을 개발하여도 새로운 프로그램의 소스코드를 공개하지 않고 BSD가 아닌 다른 라이선스를 적용하여 판매할 수 있다. 주요 내용을 요약하면 다음과 같다.



- SW를 배포하는 경우 저작권 표시, 보증책임이 없다는 내용을 표시
- 수정 프로그램에 대한 소스코드의 공개를 요구하지 않기 때문에 상용 SW에 무제한 사용가능

#### 2) 저작권 표시 등 의무사항

소스코드를 재배포하기 위해서는 반드시 아래와 같은 저작권 표시와 문구 그리고 면책조항을 유지하여야만 한다.

**(Copyright (c) <해당년도>, <저작권소유자> All rights reserved.)**

ex) Copyright (C) 1993, Digital Equipment Corporation

소스코드와 바이너리 형태의 재배포와 사용은 수정 여부와 관계없이 다음 조건을 충족할 때 가능하다.

- 소스코드를 재배포하기 위해서는 반드시 위의 저작권 표시, 지금 보이는 조건들과 다음과 같은 면책조항을 유지하여야만 한다.
- 바이너리 형태의 재배포는 배포판과 함께 제공되는 문서 또는 다른 형태로 위의 저작권 표시, 지금 보이는 조건들과 다음과 같은 면책조항을 명시해야 한다.
- 사전 서면 승인 없이는 저자의 이름이나 기여자들의 이름을 이 소프트웨어로부터 파생된 제품을 보증하거나 홍보할 목적으로 사용할 수 없다.



본 SW는 저작권자와 기여자들에 의해 “있는 그대로” 제공될 뿐이며, 상품가치나 특정한 목적에 부합하는 묵시적 보증을 포함하여(단, 이에 제한되지 않음), 어떠한 형태의 보증도 하지 않는다. 어떠한 경우에도 재단과 기여자들은 제품이나 서비스의 대체 조달, 또는 데이터, 이윤, 사용상의 손해, 업무의 중단 등을 포함하여(단, 이에 제한되지 않음), 본 소프트웨어를 사용함으로써 발생한 직접적이거나, 간접적 또는 우연적이거나, 특수하거나, 전형적이거나, 결과적인 피해에 대해, 계약에 의한 것이든, 엄격한 책임, 불법행위(또는 과실 및 기타 행위를 포함)에 의한 것이든, 이와 여타 책임 소재에 상관없이, 또한 그러한 손해의 가능성이 예견되어 있었다 하더라도 전혀 책임을 지지 않는다.

BSD 라이선스에 따른 프로그램을 바이너리(binary) 형태로 재배포하는 경우에는 소스코드를 배포하는 경우와 마찬가지로 배포판과 함께 제공되는 문서 또는 그 밖의 매체에 위의 내용을 담아야 한다. 또한 사전 서면 승인없이 저자의 이름이나 기여자들의 이름을 이 SW로부터 파생된 제품을 보증하거나 홍보할 목적으로 사용해서는 안 된다.

### 3) 공개의 범위

BSD 라이선스의 경우 의무사항만 준수한다면 소스코드를 공개하지 않아도 된다.

## 3.2.5 Apache License

### 1) 주요내용

아파치 라이선스(Apache License)는 아파치 웹서버를 포함한 아파치 재단(ASF: Apache Software Foundation)의 모든 SW에 적용되는 라이선스로 BSD 라이선스와 비슷하여 소스코드 공개 등의 의무가 발생하지 않는다. 다만 “Apache”라는 이름에 대한 상표권을 침해하지 않아야 한다는 조항이 명시적으로 들어가 있고, 특허권에 관한 내용이 포함되어 BSD 라이선스보다는 좀 더 법적으로 완결된 내용을 담고 있다. 특히 아파치 라이선스 2.0에서 특허에 관한 조항이 삽입되어 GPL 2.0으로 배포되는 코드와 결합하는 것이 어렵다는 문제가 있었는데, GPL 3.0에서는 이 문제를 해결하여 아파치 라이선스로 배포되는 코드가 GPL 3.0으로 배포되는 코드와 결합하는 것이 가능해졌다.



- “Apache”라는 이름에 대한 상표권을 침해하지 않아야 함
- SW를 배포하는 경우 저작권 표시, 보증책임이 없다는 내용을 표시
- 수정프로그램에 대한 소스코드의 공개를 요구하지 않기 때문에 상용 SW에 무제한 사용가능

## 2) 저작권 표시 등 의무사항

아파치 라이선스로 만들어진 SW를 배포하는 경우에는 저작권 표시와 보증책임이 없다는 표시를 하여야 한다. 예를 들면 아래와 같다.

**(Copyright [해당년도] [저작권소유자의 이름])**

ex) Copyright 1999–2004 The Apache Software Foundation.

이 프로그램은 아파치 라이선스 2.0에 의해 라이선스 되었다 : 당신은 이 라이선스에 따르지 않고 이 파일을 사용할 수 없다. 아래 사이트에서 라이선스의 복사본을 얻을 수 있다.

<http://www.apache.org/licenses/LICENSE-2.0>

적용 가능한 법 또는 합의된 바에서 요구하지 않는 이상, 이 라이선스 하에서 배포될 SW는 묵시적이든 명시적이든 어떠한 종류의 보증 또는 이용조건 없이 “합의된 바”대로에 기초하여 배포된다. 이 라이선스 하에서의 허가사항들과 제한사항들에 대해서는 특정 언어의 라이선스를 참고하기 바란다.

## 3) 공개의 범위

아파치 라이선스의 경우 의무사항만 준수한다면 소스코드를 공개하지 않아도 된다.

### 3.2.6 MPL(Mozilla Public License)

#### 1) 주요내용

MPL은 Netscape 브라우저의 소스코드를 공개하기 위해 개발된 라이선스로 공개하여야 할 소스코드의 범위를 좀 더 명확하게 정의하고 있다. 즉, GPL에서는 링크되는 SW의 소스코드를 포함하여 공개하여야 할 소스코드의 범위가 모호하게 정의되어 있지만 MPL에서는 링크 등의 여부에 상관없이 원래의 소스코드가 아닌 새로운 파일에 작성된 소스코드에 대해서는 공개의 의무가 발생하지 않는다. 따라서 MPL SW 그 자체는 어떻게 하든 공개를 해야 하지만 원래 소스코드에 없던 새로운 파일들은 공개하여야 할 의무가 발생하지 않으므로 GPL에 비해 훨씬 명확하다. 주요 내용을 요약하면 다음과 같다.



- SW를 배포하는 경우 저작권 표시, 보증책임이 없다는 표시 및 MPL에 의해 배포된다는 사실을 명시
- MPL 코드를 수정한 부분은 다시 MPL에 의해 배포
- MPL 코드와 다른 코드를 결합하여 프로그램을 만들 경우 MPL 코드를 제외한 결합 프로그램에 대한 소스코드는 공개할 필요가 없음
- 소스코드를 적절한 형태로 제공하는 경우, 실행파일에 대한 라이선스는 MPL이 아닌 다른 것으로 선택가능
- 특허기술이 구현된 프로그램의 경우 관련 사실을 'LEGAL' 파일에 기록하여 배포

## 2) 저작권 표시 등 의무사항

MPL로 SW를 배포하는 경우에는 저작권 표시, 보증이 없다는 표시와 함께 MPL에 의해 배포된다는 사실을 아래와 같이 명시하여야 한다. 특히, 특허기술이 구현된 경우에는 관련 사실을 “LEGAL” 파일에 기록하여 배포하여야 한다.

이 프로그램은 모질라 공개 라이선스 1.1에 따른다. 당신은 이 라이선스에 따르지 않고 이 파일을 사용할 수 없다. 라이선스의 복사본은 아래 사이트에서 구하실 수 있다.

<http://www.mozilla.org/MPL/>

이 라이선스 하에서 배포될 SW는 묵시적이든 명시적이든 어떠한 종류의 보증 또는 이용조건 없이 “합의된 바” 대로에 기초하여 배포된다.

이 라이선스 하에서의 허가사항들과 제한사항들에 대해서는 특정 언어의 라이선스를 참고하기 바란다.

**(원본SW는 <원본SW이름>)**

ex) The Original Code is mozilla.org code.

**(원본소스의 최초개발자는 <개발자이름>).**

ex) The Initial Developer of the Original Code is Netscape Communications Corporation.

(Portions created by <프로그램저작권 소유자 이름> are Copyright (C) <해당연도>. All Rights Reserved.)

ex) Portions created by the Initial Developer are Copyright (C) 1998 the Initial Developer. All Rights Reserved.

(기여자들 : \_\_\_\_\_.)

ex) 기여자들 : Mozilla.org teams

**(아래의 GPL, LGPL, NPL 등은 자신이 원하는 라이선스로 변경할 수 있음)**

다른 한편, GPL 또는 LGPL의 규정이 위의 것들을 대체하여 적용될 수 있는 경우, 이 파일의 콘텐츠는 GNU GPL 2.0 또는 GPL 아니면 GNU LGPL 2.1 또는 LGPL의 조건에 따라서 사용될 수 있다. 만약 개발자가 이 파일의 개발자 버전이 GPL 또는 LGPL의 조건에 따라서만 사용되기만을 원하고 NPL의 조건에 따라서 사용되는 것을 원하지 않는다면, 위의 규정들을 삭제하고 GPL 또는 LGPL의 고지 사항 및 다른 규정들로 대체함으로써 개발자의 의도를 표시해야 한다. 만약 개발자가 위의 규정들을 삭제하지 않는다면, 사용자는 NPL, GPL 또는 LGPL 중 어느 하나의 조건에 따라 이 파일의 개발자 버전을 사용할 수 있다.

### 3) 공개의 범위

MPL도 한편으로는 GPL과 마찬가지로 수정된 코드의 소스코드를 제공할 것을 요구하면서, 다른 한편으로는 MPL 조건의 코드와 기타의 라이선스 조건의 코드를 결합한 프로그램을 만드는 것을 허용하고, 결합된 프로그램을 MPL이 아닌 다른 라이선스로 배포하는 것을 허용하고 있다. 예를 들면 별도의 파일로 함수(Function)를 추가할 경우 MPL은 기존 코드의 수정 부분에만 적용할 뿐 추가된 함수에는 적용되지 않는다.

주요 오픈소스SW 라이선스 비교

	무료 이용가능	배포 허용가능	소스코드 취득가능	소스코드 수정가능	2차적 저작물 재공개 의무	독점SW와 결합가능
GPL	○	○	○	○	○	×
LGPL	○	○	○	○	○	○
MPL	○	○	○	○	○	○
BSD license	○	○	○	○	×	○
Apache license	○	○	○	○	×	○

## 3.3 주요 쟁점

### 3.3.1 소스코드 공개 여부

앞에서 GPL·LGPL·BSD·MPL·아파치 라이선스 등 많이 사용되는 라이선스들을 간략히 살펴보았는데 이중에서 GPL·LGPL·MPL 등은 수정한 내용에 대한 소스코드를 공개하여야 하지만 BSD·아파치 라이선스 등은 수정하더라도 소스코드를 공개할 의무가 발생하지 않는다. GPL·LGPL·MPL 등 소스코드 공개의무가 발생하는 라이선스는 상호주의(reciprocal) 혹은 Copyleft 라이선스라고 하며, 그 결과물이 원 SW의 소스코드를 이용하였을 경우 소스코드를 공개해야 한다. 그런데 소스코드의 공개범위를 기계적으로 판단할 수 있는 방법은 없으며 라이선스마다 서로 다르게 정의하고 있으므로 잘 판단하여야 한다.

### 3.3.2 특허권

GPL·LGPL·MPL·아파치 라이선스 등의 오픈소스SW 라이선스는 특허와 관련된 조항들을 가지고 있는데, 각각의 경우를 i) 라이선서(Licensor)의 특허인 경우, ii) 라이선시(Licensee)의 특허인 경우, iii) 제3자의 특허인 경우로 구분하여 설명할 수 있다. 다만 LGPL은 특허와 관련해서는 GPL과 동일하게 규정하고 있고, BSD는 특허에 관한 규정을 두고 있지 않기 때문에 생략한다.

#### ● 라이선서(Licensor)의 특허인 경우

SW에 대해 저작권을 가지고 있는 주체가 특허권을 함께 가지고 있는 경우 MPL과 아파치 라이선스는 라이선서가 SW를 오픈소스SW 라이선스로 배포하는 경우 관련 특허권의 라이선스도 무상으로 제공하는 것으로 규정하고 있다. GPL의 경우에는 명문으로 규정하고 있지 않지만 대체적으로 관련 조문(제7조 등)의 해석상 묵시적인 라이선스를 제공하는 것으로 보고 있다. GPL 3.0에서는 단순 재배포자를 제외한 개발자 및 기여자(Contributor)의 경우 자신이 기여한 부분과 관련된 특허권 라이선스를 무상으로 제공하는 것으로 규정하고 있다. 한가지 주의하여야 할 것은 특허권 그 자체는 여전히 유효하다는 것이다. 예를 들면 특허권자가 특허받은 정렬 알고리즘을 GPL로 배포하는 리눅스에 로열티 없이 사용 가능하도록 제공한다고 할지라도 독점 라이선스인 MS 윈도우즈에는 해당 정렬 알고리즘을 사용토록 허가하면서 여전히 로열티를 받을 수 있다.

### ● 라이선시(Licensee)의 특허인 경우

오픈소스SW를 사용하는 이용자가 특허권을 가지고 있는 경우이다. MPL의 경우 이용자가 자신의 특허권을 그냥 사용하는 경우에는 아무런 문제가 없지만, 만약 이용자가 MPL로 배포된 프로그램을 사용하던 중 자신의 특허권을 근거로 소송을 제기하게 되면, 적절한 시일 내에 소송을 철회하지 않는 한 라이선스가 종료된다. 따라서 MPL로 배포된 프로그램 사용자는 그 결과 MPL로 배포된 프로그램을 더 이상 사용할 수 없거나, 그 동안 사용했던 부분에 대하여 로열티 산정을 하는 등 일정한 보복이 가해진다. 아파치 라이선스 2.0도 MPL과 비슷한 취지의 조항을 추가하였으며, GPL 3.0에서도 관련 내용이 추가되었다.

### ● 제3자의 특허인 경우

특허권자와 이를 프로그램으로 구현한 주체가 다른 경우인데, 특허권자가 무상(Royalty-Free) 조건의 특허 라이선스를 허용하지 않는다면 구현자는 이 프로그램을 GPL 조건으로 배포할 수 없다(GPL 제7조). 예를 들면 甲회사가 乙회사의 특허기술을 바탕으로 A라는 프로그램을 만들었을 경우, 乙회사가 그 특허를 모든 사람에게 무상으로 허용하지 않는다면, 甲회사가 라이선스를 무료로 받았다고 할지라도 A프로그램을 GPL 조건으로 배포할 수 없다. 나아가 GPL 3.0에서는 제3자인 특허권자가 이용자들을 차별하여 라이선스를 부여하는 것을 막기 위한 조항이 삽입되었다. MPL은 제3자의 특허인 경우에도 일단 배포는 허용하되, 'LEGAL'이라는 이름의 파일을 추가하여 어떠한 특허가 문제되고 있는지, 어떤 사람이 문제나 이익을 제기하고 있는지에 대한 사항을 자세히 기록하도록 하고 있다.

## 3.3.3 라이선스의 양립성

SW를 작성하고자 할 경우 기존에 만들어진 코드를 재사용하거나 결합하는 경우가 많은데, 결합되는 각 코드의 라이선스가 상충되는 경우가 있다. 따라서 어떤 오픈소스SW에 다른 오픈소스SW를 결합할 경우 반드시 두개의 라이선스가 서로 양립하는지를 확인하여야 한다. 양립성 문제는 오픈소스SW 진영에 심각한 문제점을 제기하였으며, 이를 해결하기 위한 노력도 다양하게 진행되고 있다. 예를 들어 모질라 프로젝트(Mozilla.org)에서는 프로젝트의 결과물을 MPL, GPL, LGPL의 3가지(triple) 라이선스로 배포하는 라이선스 정책을 채택하여, 라이선스의 양립성과 관련된 불확실성을 제거하고 모질라 코드를 GPL 또는 LGPL 기반의 응용프로그램에 사용할 수 있도록 하였다. Trolltech도 Qt 라이브러리에 대한 오픈소스SW 라이선스인 QPL과 GPL의 양립성 문제를 해결하기 위하여 QPL 및 상용라이선스 이외에 GPL을 추가하는 정책을 취하고 있다.

오픈소스SW 라이선스의 양립성 문제는 주로 GPL 조건의 엄격성 때문에 나타난다. GPL 3.0의 초안에서는 이러한 양립성 문제를 해결하기 위해 이용자들이 추가적인 제한·허용 조건을 선택할 수 있도록 폭넓게 규정했었다. 예를 들어 특허보복조항의 포함여부, 네트워크서버 형태로 이용하는 경우 소스코드를 제공하게 할 것인지의 여부 등을 개별적으로 선택할 수 있도록 하였다. 그러나 GNU·리눅스 배포판 업체들은 오픈소스SW 라이선스들의 증가와 그에 따른 양립성 문제는 인정하지만, 추가적인 제한 및 허용 규정에 의해 수많은 프로그램들이 서로 다른 GPL 조건들을 가질 경우, 라이선싱과 관련된 문제들이 더욱 복잡하게 될 것이라고 주장하였다. 이러한 비판을 어느 정도 수용하여 GPL 3.0에서는 특허보복조항을 제10조에 포함시키고, 네트워크서버 형태의 이용과 관련해서는 제14조를 별도로 만드는 등 보다 단순화하였다. 또한 이러한 조치를 통해 GPL 3.0은 아파치 라이선스 2.0과 양립가능하며, 네트워크서버 형태로 이용하는 경우에도 소스코드를 제공하도록 요구하는 Affero GPL 3.0과도 양립가능하게 되었다.

### 3.3.4 듀얼 라이선스

일부 오픈소스SW는 상업용 라이선스와 오픈소스SW 라이선스 또는 오픈소스SW 라이선스들을 조합하여 해당 프로그램을 배포하는 경우가 있다. 이를 주로 듀얼 라이선스(dual license)라고 하며, 이런 경우는 주로 오픈소스SW를 상업적 목적으로 이용할 뿐만 아니라 오픈소스 커뮤니티와의 협력을 위한 경우가 대부분이다. 하나 이상의 라이선스가 있는 오픈소스SW를 이용할 경우, 이용자는 사용 목적에 가장 잘 부합하는 라이선스 하에 배포되는 소스코드를 선택할 수 있다. 대표적인 사례로는 MySQL, Trolltech의 Qt 라이브러리 등이 있다.

MySQL을 예로 들어 듀얼 라이선싱 모델에 대해서 좀 더 살펴보면 다음과 같다.

MySQL은 개발에 있어서는 소스코드가 공개된 GPL 라이선스를 이용하고 있으나 자신이 개발한 프로그램을 공개하지 않기를 원하는 기업이나 개인에게는 이들이 개발한 소스코드를 공개하지 않아도 되는 상업용 라이선스를 주는 듀얼 라이선싱 모델을 취하고 있다. 이러한 개발방식을 통해 MySQL AB사는 전세계의 수많은 오픈소스SW 사용자나 개발자들의 도움을 받아 MySQL의 품질이나 신뢰성, 인지도 등을 높이고 이를 통해 MySQL의 시장점유율을 확대하고 있다. 이렇게 만들어진 MySQL 서버 및 부가제품들은 많은 오픈소스SW 사용자들에 의해서 프로그램을 검증받았을 뿐 아니라 높은 시장점유율을 보이고 있기 때문에 MySQL을 이용을 상업적으로 이용하려는 요구는 점점 많아지게 되고 이것은 MySQL AB사의 수익으로 연결되게 된다. 현재 이러한 듀얼 라이선스 모델은 오픈소스SW를 이용한 비즈니스 모델의 대표적인 예로 자리를 잡았으며, 많은 오픈소스SW기업들이 이러한 유형의 수익 모델을 이용하고 있다.

## 3.4 주요 오픈소스SW 사례

앞서 살펴본 오픈소스SW 라이선스에 대한 이해를 바탕으로 실제 많이 사용하고 있는 오픈소스 SW들 중에서 특별히 기억해 두어야 할 라이선스 관련 이슈에 대해 살펴본다.

### 3.4.1 Linux Kernel

리눅스 커널은 GPL 2.0으로 배포된다. 그런데 리눅스 커널의 'COPYING' 파일에는 GPL 2.0의 전문과 함께 다음과 같은 내용이 맨 위에 추가로 기재되어 있다.

NOTE! This copyright does *\*not\** cover user programs that use kernel services by normal system calls – this is merely considered normal use of the kernel, and does *\*not\** fall under the heading of “derived work”. Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the Linuxkernel) is copyrighted by me and others who actually wrote it.

Also note that the only valid version of the GPL as far as the kernel is concerned is this particular version of the license (ie v2, not v2.2 or v3.x or whatever), unless explicitly otherwise stated.

정상적인 시스템 콜에 의해 커널 서비스를 이용하는 사용자 프로그램(user programs)은 GPL 2.0에 의해 배포하지 않아도 좋다는 뜻이다. 이와 같은 내용에 따라 리눅스 커널을 기반으로 한 라이브러리나 응용프로그램은 GPL 2.0의 영향을 받지 않는 것이다. 그러나 리눅스 커널은 커널의 기능을 커널 모듈이라는 형태로도 이용할 수 있는 인터페이스를 제공하는데 이 커널 모듈도 위의 예외사항에 속하는지에 대한 논란이 계속되고 있다. 즉, 리눅스 커널 모듈도 모두 GPL 2.0으로 소스코드를 공개해야 하는지, 아니면 독점 라이선스를 적용하여 소스코드를 공개하지 않아도 되는지에 대한 논란이다. 이에 대해서는 리눅스 커널 개발자들 사이에서도 의견이 갈리고 있다.

최소한, <http://www.kernel.org>에서 배포되는 공식 커널 버전을 기준으로 모듈 인터페이스를 임의로 수정하지 않은 상태에서 동작이 가능한 자체 개발된 커널 모듈은 GPL이 아닐 수 있다고 주장할 수 있다. 그러나 모듈 인터페이스를 임의로 수정할 경우에는 설득력이 훨씬 약해지며, 커널 모듈이 GPL인지의 여부에 대한 논란은 매우 첨예하게 대립하고 있으므로 커널 모듈은 모두 GPL이라고 가정하고 공개가 불가능한 부분은 사용자 프로그램(user program)에서 구현하는 것이 바람직하다.



### 3.4.2 FreeBSD

FreeBSD는 Unix시스템인 BSD(Berkeley Software Distribution)를 기반으로하여 개발되었고, 다양한 Unix 버전 중 오픈소스 Unix라 할 수 있을 것이다. FreeBSD는 1993년에 처음 발표되었으며 가장 제약사항이 적은 BSD 라이선스에 의해 배포되고 있다. 따라서 FreeBSD 소스코드를 상업적으로 아무런 제한없이 이용할 수 있다. 단, 소스코드 혹은 바이너리 형식으로 재배포할 때는 FreeBSD의 원저작권자인 The Regents of the University of California의 저작권을 명시해야 한다.<sup>11)</sup> 또한, 해당 프로그램이 FreeBSD를 이용하고 있음을 밝혀야 하며<sup>12)</sup> 재배포 시에는 FreeBSD 소스코드에 포함된 라이선스 내용을 그대로 포함시켜야 한다. 요약하면 FreeBSD는 소스코드 공개를 요구하지 않으며, 만약 원본 혹은 수정된 소스코드를 공개하고자 하면 위에서 언급한 사항들만 준수하면 된다.

### 3.4.3 MySQL

MySQL은 현재 가장 인기있는 관계형 데이터베이스 서버로서 사용자는 GPL 라이선스나 일반 상용 라이선스 둘 중 한 가지를 선택할 수 있다. 상용 라이선스는 GPL 라이선스의 여러가지 요구 사항들(소스코드 공개 등)을 지키기 어려운 경우에 선택할 수 있으므로 일반적인 상용 라이선스 판매를 통해 수익을 내고 있다. 이러한 라이선스 모델을 듀얼 라이선스라고 하며 MySQL은 듀얼 라이선스 모델의 대표적인 사례로서 종종 언급된다.

그러나 한가지 주의하여야 할 것은 GPL의 의무사항은 SW를 배포할 때 발생하는 것이므로 만약 MySQL을 다운로드하여 MySQL과 연동되는 웹사이트 등을 만들어 서비스만 하는 경우는 MySQL을 직접 배포하지 않는 것이므로 GPL의 의무사항이 발생하지 않는다는 것이다. 예를 들면 인터넷 포털 업체들이 MySQL의 상용 버전을 구입하지 않고 GPL을 사용하면서 MySQL이나 관련 SW의 소스코드를 공개하지 않아도 법적인 문제가 발생하지 않는다.

### 3.4.4 Apache

Apache SW들은 현재 아파치SW재단(ASF)에서 자체적으로 개발한 아파치 라이선스 버전 2.0하에 배포되고 있다. 아파치 라이선스에 따르면, 누구든 자유롭게 Apache SW를 다운받아 부분



11) 현재 FreeBSD의 소스코드에는 다음과 같이 저작권이 명시되어 있다. "All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by The Regents of the University of California"

12) "This product includes software developed by the University of California, Berkeley and its contributors."를 명시하여야 한다.

또는 전체를 개인적 혹은 상업적 목적으로 이용할 수 있다. 또한 재배포 시에도 원본 소스코드 혹은 수정한 소스코드를 반드시 포함시킬 것을 요구하지 않는다. 다만, 재배포하고자 할 경우에는 아파치 라이선스 2.0을 포함시키고 아파치SW재단에 개발된 SW임을 명확히 밝힐 것을 요구한다. 아울러 아파치SW재단이 승인하지 않는 재단 공식마크를 임의로 사용할 수 없다.

### 3.4.5 Mozilla Firefox

Firefox는 유명한 오픈소스 웹브라우저로서 현재 MPL, GPL, LGPL 세 가지 라이선스 하에 배포되고 있다. 이 세 가지 라이선스는 모두 공통적으로 누구나 소스코드를 보고 수정하며 재배포하는 것을 허용한다. 원래 Firefox는 MPL에 의해 배포되었으나 파생물의 상업적 이용을 제한적으로 허락하는 MPL은 GPL 혹은 LGPL과 호환될 수 없기 때문에 FSF로 부터 많은 비난을 받았다. 이 문제를 해결하기 위해 Mozilla는 Firefox를 MPL, GPL, 그리고 LGPL하에 다시 라이선싱하였다. 이후, 개발자들은 이 세 가지 라이선스 중 자신들의 목적에 가장 잘 부합하는 라이선스를 선택할 수 있게 되었다. 그런데 하나 유의할 점은 Firefox는 일부 상용 컴포넌트를 포함하고 있기 때문에, 이 상용 컴포넌트는 위에서 언급한 세 가지 라이선스의 적용을 받지 않는다는 것이다. 대신 이들은 모질라 최종사용자 라이선스 계약(Mozilla End User License Agreement)의 제한을 받고 있다.

### 3.4.6 Sendmail

Sendmail은 1981년 Eric Allman에 의해 메일전송 에이전트(MTA, Mail Transfer Agent)로 개발되었다. 당시에 그는 UC Berkely에서 일하며 대학의 네트워크와 Arpanet<sup>13)</sup> 사이에 이메일을 주고받기 위한 목적으로 Sendmail을 개발하였다. 처음부터 Sendmail은 메일 프로토콜의 개방성과 라우팅 기능성, 파일의 유연성에 초점을 맞추었기 때문에 오늘날 이메일 서버 시장의 1위 자리를 차지하고 있다. Allman은 여전히 Sendmail 개발의 중심에 있으며, Sendmail의 유지보수와 기능 및 성능 개선 등의 난관을 헤쳐가기 위해 Sendmail, Inc.이라는 회사를 설립하였다. Sendmail 8.9 이전 버전은 오픈소스 형식으로 개발되었으며, 8.9 버전부터는 Allman이 설립한 회사에 의해 개발 및 공개되었다. 이러한 역사적 배경 때문에 Sendmail은 두 가지 다른 라이선스 형식을 취하고 있다. 우선, Sendmail을 단순히 컴파일해서 바이너리만 사용하거나, 아니면 오픈소스 형식으로 소스코드를 제공하며 이용할 경우는 Sendmail License를 적용



13) 미국방부 안에 있는 ARPA(advanced research project agency; 고등연구 계획국)가 미국 각지에 있는 대학과 비영리기관을 중심으로 완성한 전국적인 컴퓨터 네트워크를 말한다.

받는다. 이 라이선스는 Sendmail의 자유로운 이용, 수정 및 배포를 허락한다. 단, 재배포할 경우에는 배포에 필요한 비용 이상을 부과할 수 없으며, Sendmail에 포함된 copyright notice를 반드시 포함시켜야 한다. 그리고 재배포시 소스코드를 포함시키지 않을 경우에는 최대 3년까지 소스코드 제공을 보장하는 문서를 포함시켜야 한다. 반면, 소스코드 제공없이 상업적 용도로 이용할 경우에는 Sendmail, Inc.와 접촉해서 별도의 라이선스를 받아야만 한다.

### 3.4.7 기타 주요 오픈소스SW

그 밖의 수많은 오픈소스SW가 사용되고 있으며 이러한 SW들은 GPL, LGPL, BSD 등 다양한 라이선스로 배포되고 있다. 실제 많이 사용되고 있는 오픈소스SW와 그에 해당하는 라이선스를 『참고자료』에 정리하였다.

## 3.5 오픈소스SW 라이선스 위반사례

### 3.5.1 Bracken 사건

Bracken은 GNU/리눅스 운영체제 제품을 생산하는 회사로 이 회사에서 판매하는 제품은 주로 OEM 벤더들에 판매되며 응용장치에 설치되어 인터넷 브라우징 역할을 하는 장치 같이 단일한 용도로 쓰인다. Bracken의 제품은 거의 100% 자유소프트웨어(Free Software)이며, 대부분 GPL이나 관련된 자유소프트웨어 라이선스에 따라 생산된 것이었다. 그러나 Bracken은 소스코드 제공 없이 단순히 바이너리(binary)만 이용할 수 있도록 하고 있었으며, 최종 사용자 이용 계약(End User License Agreement, EULA)에는 GPL에 의해 승인된 허용범위와는 모순된 내용을 포함하고 있었다.

이러한 사실이 자유소프트웨어 저작권자들에 의해 FSF에 통보되었다. FSF는 Bracken에 자세한 위반사항을 전달했고 Bracken은 즉시 제품 배포를 일시 중단했다. 그리고 FSF와 합의에 따라 최종 사용자 라이선스 계약을 FSF 심사를 거치도록 하고 소스코드를 공개하였다.

### 3.5.2 Fortinet 사건

이 사건은 보안 소프트웨어 개발사인 Fortinet가 자사의 소프트웨어에 리눅스 컴포넌트이자 GPL로 배포된 'initrd'를 사용하였으나 암호 기술을 사용하여 그 사용을 숨긴 데서 발단했다.

이에 gpl-violations.org 운영자인 벨테(Welte)가 이를 개발한 개발자에게 저작권을 양도받아 GPL 위반으로 독일 법원에 제소하였다. 이에 2005년 독일 뮌헨 지방법원은 Fortinet에 GPL 조건을 준수하지 않음을 이유로 배포금지명령을 내렸고, 이로 인해 자사 제품을 유통할 수 없게 되자 결국 Fortinet은 자사 소프트웨어 소스코드의 일부를 공개하였다.

### 3.5.3 Skype 사건

Skype는 인터넷 전화인 P2P VoIP(voice over IP) 소프트웨어로 유명한 회사로 자사가 판매하는 SMC 네트워크 VoIP 전화기가 있었다. 그러나 이 전화기에는 벨테(Welte)가 저작권을 가지고 있는 GPL 2.0으로 배포된 소프트웨어 2개를 포함한 리눅스 운영체제를 사용하였다. 그러나 이 제품은 GPL 2.0의 요구조건인 소스코드를 공개하지 않았을 뿐만 아니라 GPL 라이선스 문구를 붙이지도 않았다.

이에 벨테는 독일 뮌헨 지방법원에 라이선스 위반으로 고소를 했고 승소했다. 이 사건에서 Skype의 위반 내용은 미미하기 때문에 업무정지나 판매정지를 받지는 않고 벌금만 낼 것으로 보인다.

오픈소스SW 라이선스 가이드

## IV 기업에서의 오픈소스SW 라이선스 관리/활용 방안

- 
1. 오픈소스SW 관련 정책의 수립
  2. 오픈소스SW 라이선스 관리를 위한  
프로세스 및 조직의 구축
-

## IV

기업에서의 오픈소스SW  
라이선스 관리/활용 방안

## 4.1 오픈소스SW 관련 정책의 수립

최근 국내·외 많은 기업들이 오픈소스SW를 활용하여 비즈니스를 수행하고 있다. 선진 기업들은 기업 외부의 오픈소스SW를 도입하여 활용할 뿐만 아니라, 오픈소스 개발방법론을 통해 기업에서 필요한 SW를 개발하기도 하며, 그 과정에서 외부에 있는 오픈소스 커뮤니티를 적극 활용하기도 한다. 하지만 국내 기업들의 대다수는 오픈소스SW 관련 정책을 따로 마련하지 않고 개발자들 수준에서 기업외부에 존재하는 오픈소스SW를 단순히 활용하는 수준에 머물러 있는데, 향후 오픈소스SW에 대한 활용의 필요성이 지속적으로 증가할 것이라는 점을 고려한다면 개별기업 차원에서 오픈소스SW 관련 정책을 수립할 필요성이 있다. 오픈소스SW 관련 정책을 수립할 때에는 개별기업이 속한 시장의 상황, 오픈소스SW의 발전정도, 개별기업들의 기술수준 및 비즈니스모델 등을 고려할 필요가 있다.

개별기업에서 오픈소스SW를 활용할 경우에는 오픈소스SW의 장·단점을 명확히 이해한 상태에서 활용 여부 및 방향을 결정하여야 할 것이다. 일반적으로 오픈소스SW는 다음과 같은 장점을 가지고 있다.

## ● 낮은 진입비용

일반적으로 오픈소스SW는 웹상에서 무료로 다운로드 및 소스코드 수정·재배포가 가능한 것이 특징이다. 따라서 초기 개발 비용이 적게 요구된다는 장점이 있다.

## ● 빠르고 유연한 개발

오픈소스SW 커뮤니티는 보통 최신 기술 정보 및 문제점과 해결책을 공유하는 형태로 자유롭게 운영되기 때문에 독점 프로그램에 비해 기술 발전 속도가 빠르다.

## ● 오픈포맷과 프로토콜

오픈소스SW는 주로 오픈포맷 또는 프로토콜을 사용하기 때문에 서로 다른 SW간 상호 연동성이 보장되는 장점이 있다. 모든 기기들이 서로 다른 네트워크를 통해 하나로 연결되는 유비쿼터스 시대에는 필수적인 요소로 볼 수 있다. 또한 오픈소스 운동의 주원인 역시 상용 프로그램들 간의 비호환성을 해결하는 것이다.



# Open Source SW Licence Guide

## ● 신뢰성과 안정성

오픈소스SW의 개발 과정을 볼 때, 전세계에 있는 수많은 우수한 개발자들이 직접 개발과 디버깅 과정에 참여하기 때문에 In-house에서 폐쇄적으로 개발되는 독점 프로그램에 비해 비교적 안정적으로 동작한다는 평이 일반적이다. 하지만 이러한 신뢰성과 안정성은 많은 개발자들의 적극적인 참여가 있을 때에만 가능한 것이기 때문에 사용하고자 하는 오픈소스의 개발 과정을 주의깊게 살펴볼 필요가 있다.

## ● 네트워킹 지원

오픈소스SW가 확산된 가장 큰 이유가 다양하고 강력한 네트워킹 기능 지원이라 해도 과언이 아닐 것이다. 대표적으로 아파치는 전세계 웹 서비스의 절반 이상을 차지하고 있을 정도이다. 또한 오픈소스 네트워킹 솔루션은 대부분 상용 프로그램과 호환되기 때문에 상품화에도 아주 잘 활용될 수 있을 것이다.

반면 오픈소스SW는 다음과 같은 단점이 지적되고 있다.

## ● 애플리케이션의 부족

대부분의 사용자들은 윈도우즈 기반의 GUI(Graphical User Interface)를 갖고 있는 Application에 익숙해 있지만, 이에 버금가는 리눅스 기반의 Application이 많이 부족한 것이 현실이다. 또한 리눅스 기반으로 개발된 많은 Application들은 윈도우즈 기반 Application들과 호환되지 않는 문제점도 있다.

## ● 빈약한 문서

상용 프로그램에 비해 오픈소스SW는 체계적인 문서를 갖고 있지 못한 단점이 있다. 이는 경우에 따라서는 개발과정의 지연을 초래할 수도 있기 때문에 활용하고자 하는 오픈소스가 얼마만큼 문서화가 잘 되었는지도 살펴보아야 한다.

## ● 불확실한 개발 로드맵

오픈소스SW는 영리를 목적으로 하는 회사에서 개발되는 것이 아니라, 자발적 참여를 바탕으로 웹상에서 자유롭게 개발되는 것이 특징이다. 그렇기 때문에 독점 프로그램에서 볼 수 있는 로드맵을

기대하기 힘든 면이 있다. 오픈소스SW의 이러한 단점은 오픈소스SW를 활용하는 개발 과제의 로드맵에 까지 영향을 미칠 수 있기 때문에 활용하고자 하는 오픈소스SW의 향후 개발 계획이 얼마나 체계적으로 세워져 있는지도 고려해야 한다.

#### ● 지적재산권

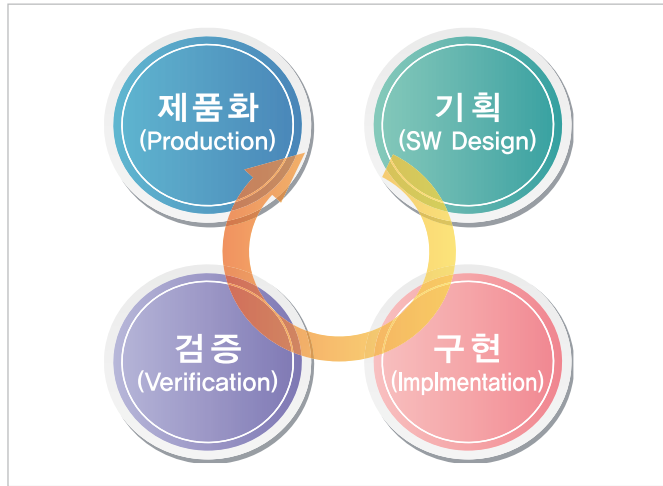
기업이 보유한 특허 및 소스코드가 오픈소스SW에 포함되는 경우 대부분의 오픈소스SW 라이선스에서는 일반적으로 특허에 대한 사용료 없이 배포할 것을 요구하고 있다. 따라서 특허에 대한 사용료 없이 배포하기를 원치 않을 경우에는 해당 오픈소스SW를 사용할 수가 없으며, 또한 사용 후 로열티를 주장하게 되면 해당 오픈소스에 대한 사용권한이 박탈되는 경우가 일반적이다. 따라서 오픈소스를 활용하여 특허를 구현하거나 기존 소스코드를 포함하고자 할 경우, 반드시 특허 사용료에 대한 입장을 명확히 하여야 할 것이다.

## 4.2 오픈소스SW 라이선스 관리를 위한 프로세스 및 조직의 구축

오픈소스SW를 활용하기 위해서는 독점SW와 마찬가지로 반드시 해당 오픈소스SW의 라이선스에 대한 준수가 필수적이다. 하지만 오픈소스SW 라이선스에서 강제하고 있는 내용에 대해서 개발자 및 관리자들의 이해가 아직도 많이 부족한 것이 현실이다. 자칫 잘못하면 라이선스 위반으로 이미 판매중인 제품을 리콜하거나 소스코드를 공개해야 하며, 개발중인 제품을 아예 처음부터 다시 개발해야 하는 상황을 초래할 수도 있으므로 체계적인 프로세스를 수립하고 이를 담당할 관련 조직을 구축하는 것이 필요하다.

개발이 끝난 이후에 오픈소스SW 라이선스 관련 문제가 발견된다면 수정에 많은 시간과 비용이 소요되므로 과제 계획 단계부터 오픈소스SW 라이선스 문제를 고려할 필요가 있다. 또한 개발이 진행되면서도 단계별로 준수해야 할 사항들을 정의하고 반드시 체크해야만 한다. 본 문서에서는 이러한 준수 사항에 대해 구체적으로 설명하기 위해 SW 개발프로세스를 다음과 같이 표준화·단순화하였다.





## 4.2.1 기획(SW Design)단계

오픈소스SW 라이선스 관련 문제를 피하는 가장 좋은 방법은 개발 기획 시점부터 이를 고려하는 것이다. 우선 해당 과제에 오픈소스SW를 활용할 것인지의 여부를 판단하여야 하며, 구체적으로 어떤 프로그램을 사용할 지를 판단하여야 한다. 오픈소스SW의 특성상 Web 상에 여기 저기 흩어져 있지만, 쉽게 오픈소스에 관한 정보를 찾을 수 있는 곳으로는 Freshmeat.net, SourceForge.net, OSDir.com, BerliOS, Bioinformatics.org 등을 들 수 있다. 이와 같은 사이트들은 대부분 라이선스별 오픈소스SW 분류 항목을 두고 있기 때문에 쉽게 해당 프로그램의 라이선스를 확인할 수 있을 것이다.

기획 단계의 마지막으로 해당 SW Component별로 소스코드 공개 가능여부를 판단하여야 한다. GPL 등 소스코드 공개 의무가 발생하는 오픈소스SW를 사용할 경우에는 결과물의 소스코드 공개가 요구되기 때문에, 경우에 따라서는 SW 구현 방법을 달리해야하기 때문이다. 소스코드의 공개가능 여부에 대한 판단 기준으로 다음의 사항을 참조할 수 있을 것이다.

### ● 유지보수

SW의 경우 하드웨어와 달리 개발 후 지속적 Upgrade 및 Debugging과 같은 유지보수 과정이 중요하다. 이러한 유지보수 과정은 상당한 Resource를 요하기 때문에 유지보수를 직접 할 것 인지에 대한 고려가 필요하다. 개발한 소스코드를 오픈소스SW 커뮤니티에 공개하고, 이를 바탕으로 오픈소스 커뮤니티를 통한 유지보수 방법 역시 경우에 따라 아주 효율적일 수도 있을 것이다.

● **빠른 개발**

오픈소스SW의 개발 모델 중 가장 특징적인 것이 바로 ‘Release Early and Often’을 통한 ‘Parallel Development and Debugging’이 가능하므로 오픈소스SW는 빠른 개발 속도를 나타낸다. 이러한 모델을 Resource가 부족한 개발 과제에 적용하면 보다 효율적이고 빠른 개발이 가능할 것이다.

● **신뢰성 확보**

SW의 신뢰성 확보의 가장 좋은 방법은 다양한 사용자들이 다양한 환경에서 해당 프로그램을 사용하면서 발견되는 문제점을 신속히 수정하는 것이다. 이런 측면에서 볼 때 오픈소스SW 커뮤니티를 잘 활용하면 SW 신뢰성 확보에 상당한 도움을 얻을 수 있을 것이다.

● **차별화 유지 어려움**

소스코드를 공개하게 되면 그 소스코드는 경쟁사에게도 공유되는 것이기 때문에 결국 제품의 차별화 확보가 불가능하게 되는 단점이 있다.

● **지적재산권 확보의 어려움**

기업이 보유한 특허를 구현하여 소스코드를 공개하는 것은 결국 모든 사용자에게 Royalty-free의 조건으로 특허를 공개하는 것이나 마찬가지가 된다.

● **특허 침해 소송 제기 가능성 증가**

소스코드가 공개되어 있으면 누구든 그 소스코드를 볼 수 있기 때문에 특허침해 소송 제기 가능성이 증가하게 되는 문제점이 발생할 수 있을 것이다.

**4.2.2 구현(Implementation)단계**

자체 개발한 소스코드를 공개해도 무방한 경우는 특별히 구현 방법에 신경 쓸 필요가 없다. 단, 소스코드를 공개할 경우 회사보유의 지적재산권을 포함시키지 않도록 주의할 필요가 있다. 그러나 소스코드 공개를 원하지 않을 경우는 사용하는 오픈소스SW의 라이선스 의무사항과 활용하고자 하는 형태(Kernel, Application, Device Driver 등)에 따라 다양한 경우가 발생할 수 있기 때문에 상당한 주의가 요구된다.

**1) 라이선스 삭제 금지**

프로젝트 매니저는 개발자에게 오픈소스SW를 사용한 경우에는 해당 라이선스를 삭제하지 않도록 해야 한다. 거의 대부분의 오픈소스SW의 경우에는 소스코드 시작부분에 해당 라이선스를 표시하고 있다. 이러한 라이선스는 오픈소스SW를 검증하는 차원에서 검색을 통해 해당 오픈소스

SW의 사용이 적절한지 판단하기 위해서 꼭 필요하기 때문에 개발자에게 이를 꼭 주의시켜야 한다. 만약 이러한 라이선스를 따르지 않을 경우 저작권 침해로 인한 법적인 문제가 발생한다. 따라서 개발자들이 오픈소스SW를 사용하는 경우 해당 SW에 사용한 오픈소스SW의 라이선스를 반드시 추가하고 이를 삭제하지 않도록 하여야 한다.

## 2) 오픈소스 사용목록 작성

SW를 구현하는 단계에 있어서 오픈소스SW를 사용할 경우에는 오픈소스SW 사용목록을 작성하여 제출하도록 하여야 한다. 이렇게 함으로써 개발자에게 오픈소스SW를 사용하는데 있어 법적인 문제가 있음을 인식시킬 수 있으며, 향후 오픈소스SW 사용으로 인한 법적인 문제가 발생했을 때도 해당 목록을 통해 적극적인 대응을 할 수 있다.

특히 SW의 일부를 아웃소싱하는 경우 SW를 개발하는 업체에 대해 오픈소스 사용목록을 작성하도록 해야 나중에 이로 인한 법적인 문제가 발생시 책임문제를 명확하게 할 수 있다.

## 3) 오픈소스 라이선스 확인

일부 오픈소스SW의 경우 라이선스의 확인이 어려운 경우가 있다. 이 경우에는 다음과 같은 방법을 통해서 라이선스를 확인해야 하고 이를 소스와 오픈소스 사용목록에 추가하여야 한다.

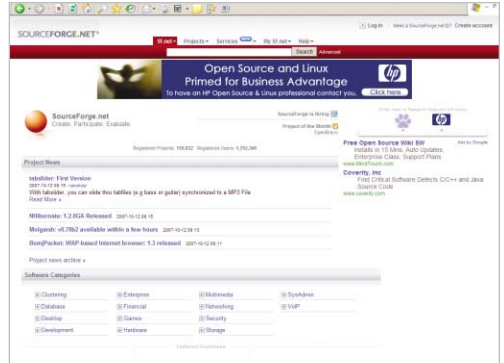
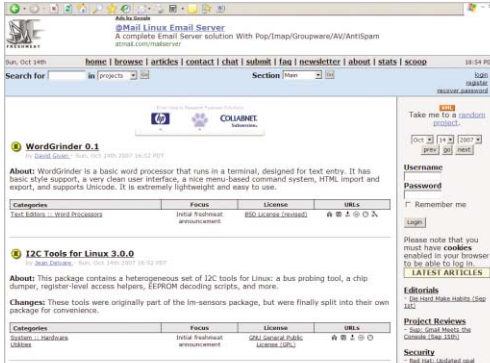
### ● 구글을 통한 방법

구글은 오픈소스SW 코드를 검색하는 코드서치(<http://www.google.com/codesearch>) 사이트를 운영하고 있다. 이 사이트를 이용하여 자신이 인용한 코드의 파일명이나 패키지명을 입력하면 해당 라이선스를 화면에 보여주기 때문에 자신이 사용한 프로그램의 라이선스를 쉽게 확인할 수 있다.



● Freshmeat이나 Sourceforge를 이용한 방법

오픈소스SW의 대부분의 프로젝트가 Freshmeat이나 Sourceforge에서 이루어지고 있기 때문에 이 사이트를 통하여 검색하면 해당 오픈소스SW의 라이선스를 쉽게 확인할 수 있다.



● 소스코드를 통한 방법

소스코드를 다운받아 COPYING, README, LICENSE 등의 파일에서 확인하거나 소스코드의 상위 코멘트에서 확인이 가능하다.

4.2.3 검증(Verification)단계

개발이 완료된 후에는 개발 결과물인 소스코드에 대해 실질적인 검증 작업이 필요하다. 개발 계획서 그 자체로는 라이선스 이슈가 없었더라도 실제 구현과정에서 개발자가 오픈소스SW 라이선스에 대한 검증없이 사용한 경우가 있을 수 있기 때문이다. 최근에는 특정한 소스코드가 오픈소스SW 코드와 일치하는지를 검증하여 주는 프로그램을 활용하는 사례가 증가하고 있다.

그러나 개발자들이 오픈소스SW의 라이선스를 지우지 않았다면 다음과 같이 간단한 스트링(string) 검색으로 라이선스를 확인할 수 있다.

또한 구현과정에서 설명한 오픈소스SW 라이선스 확인방법을 이용하면 된다.

```
$ grep -r 라이선스이름 .
현재 디렉토리( . ) 아래서 recursive하게(-r) 라이선스이름(GPL, LGPL, BSD) 등이 들어간 line을 찾는 것이다.
```

## 4.2.4 제품화(Production)단계

이 단계에서는 사용된 오픈소스SW들을 라이선스별로 분류하고 각 라이선스에서 준수해야 할 사항들이 실제로 제품에 반영될 수 있도록 하여야 한다. 앞에서 오픈소스SW의 라이선스 의무 사항은 크게 ‘저작권 관련 문구 유지’, ‘제품명 중복 방지’, ‘서로 다른 라이선스 조합’, ‘사용 여부 명시’, ‘소스코드 공개’, ‘특허’ 등이 있다고 기술하였는데, 이중에서 ‘저작권 관련 문구 유지’, ‘제품명 중복 방지’, ‘특허’ 등은 기획 및 구현 단계에서 확인되어야 할 사항이고, ‘소스코드 공개’, ‘사용 여부 명시’ 등은 제품화단계에서 확인되어야 할 사항이다.

### 1) 소스코드 공개 방법

소스코드의 공개 방법으로 두 가지 방법이 있다. 첫째는 제품판매 시 제품과 함께 소스코드를 제공하는 것이다. 둘째는 제품과 함께 저작권 정보만을 제공하며 여기에 소스코드에 대한 정보와 실제 소스코드는 웹사이트를 통해 공개하거나 CD-ROM 등과 같은 매체를 통하여 우편으로 전달하는 것이다. 이 중 첫 번째 방법은 소스코드 제공에 있어서 확실한 방법이 되겠지만 라이선스를 유지하고 관리하는데 있어서는 비효율적이라 할 수 있다. 따라서 일반적으로 제품의 라이선스를 제공할 수 있는 웹페이지를 운영하는 것이 체계적이고 효율적인 방법이라 하겠다.

### 2) 소스코드 공개방법 사례

구분	하드웨어	SW
제품명	소니 디지털카메라 DSC-G1	곰플레이어
사진		
저작권 문구	<p><b>On GNU GPL/LGPL applied software</b>                      The software that is eligible for the following GNU General Public License (hereinafter referred to as "GPL") or GNU Lesser General Public License (hereinafter referred to as "LGPL") are included in the camera.                      This informs you that you have a right to have access to, modify, and redistribute source code for these software programs under the conditions of the supplied GPL/LGPL. Source code is provided on the web. Use the following URL to download it.  <a href="http://www.sony.net/Products/Linux/">http://www.sony.net/Products/Linux/</a>                      We would prefer you do not contact us about the contents of source code.                      Read "license2.pdf" in the "License" folder on the CD-ROM. You will find licenses (in English) of "GPL," and "LGPL" software.                      To view the PDF, Adobe Reader is needed. If it is not installed on your computer, you can download it from the Adobe Systems web page:  <a href="http://www.adobe.com/">http://www.adobe.com/</a></p>	<p><b>※ 해당부분 내용</b>                      - 이 프로그램에 포함된 비디오 디코더는 FFmpeg (<a href="http://ffmpeg.sourceforge.net">http://ffmpeg.sourceforge.net</a>)을 수정하여 제작하였으며, FFmpeg의 라이선스는 LGPL을 따릅니다. LGPL 라이선스는 설치 디렉터리의 LGPL.TXT를 참고하시기 바랍니다. 수정된 FFMPEG의 소스는 <a href="http://gomdevel.gomtv.com">http://gomdevel.gomtv.com</a> 을 통해서 다운로드 받으실 수 있습니다.</p>

### 3) 새로운 오픈소스SW 라이선스를 만드는 경우

제품을 상품화하면서 오픈소스로 개발은 하지만 자신의 회사에게 유리하도록 라이선스를 규정할 수가 있다. 그것은 바로 OSI에 정하는 오픈소스SW의 10가지 조건<sup>14)</sup>에 맞추어 승인을 받으면 된다. 11월 1일 현재 OSI에는 60여개의 오픈소스SW 라이선스가 등록되어 있다. 등록절차는 다음과 같다.



1. 다른 라이선스의 제목과 구별되는 자신만의 고유한 라이선스 이름을 만든다. 즉, 기존에 승인 받은 라이선스(제목 또는 카테고리)와 구별되도록 한다.
2. 라이선스를 HTML과 플레인 텍스트의 두 가지 형식으로 제작하고 HTML 형식의 라이선스를 OSI 웹페이지에 올린다. 그러면 OSI에서 이미 승인한 라이선스와 같은 형식으로 변환시켜준다.
3. 라이선스가 오픈소스의 정의규정에 부합하도록 법적인 검토(의견)를 덧붙인다. 라이선스의 각 단락들이 각 오픈소스 정의 규정과 어떻게 부합하는지에 대한 해설이 있어야 한다. 그러한 법적검토(의견)는 공인된 변호사로부터 작성된 것이야 한다. 변호사의 법적검토가 담긴 이메일을 OSI로 보낸다
4. 아래 설명에 따라 세 부분으로 나누어진 이메일을 라이선스-토론 메일링 리스트로 보낸다. 메일 제목에는 라이선스 제목과 함께 “승인요청(승인용)”이라고 적는다.
  - OSI에서 승인받은 라이선스 중 본인의 라이선스와 가장 유사한 라이선스를 표시하고 왜 기존의 라이선스가 본인의 필요를 만족시켜주지 못하는지에 대한 설명을 덧붙인다.
  - 본인의 라이선스의 적용을 받아 배포되는 SW는 다른 오픈소스 라이선스 하에 배포되는 SW와 어떻게 결합될 수 있는지 설명한다.
  - 본인 라이선스의 텍스트 버전을 첨부한다.
5. [license-discuss-subscribe@opensource.org](mailto:license-discuss-subscribe@opensource.org)에 가입하여 라이선스 논의에 참여한다. 만약 라이선스 토론 메일링 리스트의 회원들이 라이선스가 OSI정의에 부합하지 않는다는 것을 발견 한다면 그 문제를 토론을 통해 해결할 수 있다.
6. OSI와 라이선스-메일링리스트를 구독하는 사람들 또는 다른 검토자들이 본인의 라이선스가 오픈소스정의와 부합함이 분명하고 더 이상 특이사항이 없다는 확신이 생기면 라이선스가 승인 되었음을 공지하고 OSI 웹사이트의 오픈소스SW 라이선스 리스트에 등재될 것이다.



14) OSI에서는 오픈소스가 되기 위한 10가지 조건을 만들어놓고 있는데 그 조건은 다음과 같다.

1. 자유로운 재배포, 2. 소스코드 제공, 3. 개작 허용, 4. 저작자의 소스코드 원형유지, 5. 개인 및 단체에 대한 차별 금지, 6. 사용분야에 대한 차별 금지, 7. 라이선스의 재배포, 8. 특정제품에만 유용한 라이선스 금지, 9. 다른 SW를 제한하는 라이선스 금지, 10. 기술종립적인 라이선스 제공

오픈소스SW 라이선스 가이드

## 참고자료

---

주요 오픈소스SW와 라이선스

---

# 참고 자료

# 주요 오픈소스SW와 라이선스

※ 출처 : 한국SW진흥원, 공개SW 가이드, 2005

## 가. Networking

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Web Browser	Mozilla	1) 넷스케이프의 공개SW 버전 2) 웹브라우저, 이메일 리더, 챗 클라이언트 등등 핵심 어플리케이션의 통합	MPL1.1 / NPL GPL2 / LGPL2.1
	Konqueror	1) KDE기반의 기본 웹브라우저 2) 윈도우의 탐색기와 유사한 기능제공(파일 관리, 웹 브라우저, 다양한 뷰어 기능 등)	GPL2
	Galeon	1) GNOME기반의 웹 브라우저 2) 다양하고 편리한 기능 제공	GPL2
Email client	Ximian Evolution	1) 그놈 기반의 개인정보관리 프로그램 2) 이메일, 캘린더, 통합정보관리, 일정관리, 메시지 보드 지원	GPL2
	Mozilla Mail	1) 모질라에 포함되어 있는 전자우편 2) 웹과 통합된 전자우편 기능 지원	MPL
	Kmail	1) KDE를 기반으로 한 이메일 클라이언트 2) 아웃룩 익스프레스와 같이 이메일 관련 기능만 있음.	GPL2
FTP-clients	Gftp	GTK+ 기반의 FTP 클라이언트 프로그램	GPL2
	KBear	KDE 기반의 FTP 클라이언트 프로그램	GPL
	ncftp	콘솔용 ftp 클라이언트. 메타문자 지원, 리커시브 디렉토리 지정 가능, 탭 완성 기능	Artistic
IRC-clients	Xchat	1) 유닉스 기반의 IRC 클라이언트 프로그램(GTK +) 2) 거의 대부분의 플랫폼 지원	GPL
	Kvirc	유닉스 기반의 IRC 클라이언트 프로그램 (QT)	GPL
	bitchX	콘솔 기반의 IRC. 쉬운 사용법, 많은 기능 지원	GPL2 / Others
Instant Messaging clients	Gaim	1) 인스턴스 메시지 클라이언트 2) 대부분의 플랫폼과 메시징 프로토콜을 지원	GPL
	Kmess(MSN)	리눅스에서 MSN 사용자 이용하는 KDE기반의 메시지 클라이언트	GPL
	Jabber	1) 여러 플랫폼에서 사용가능 2) MSN을 비롯한 여러 메신저와 메시징 가능	GPL / JOSL
IDS	Snort	1) 널리 쓰이는 네트워크 침입 탐지 시스템 2) 다양한 플랫폼 지원 및 다양한 플러그인과 룰	GPL2
Peer-to-peer clients	GNUtella	1) 강력한 기능을 갖는 파일 공유 프로그램 2) 방화벽 뒤에서도 사용가능 3) 서로가 서로를 계속 중계	GPL
Remote management	VNC	현재 운영 중인 컴퓨터뿐만 아니라 인터넷으로 연결된 다양한 아키텍처의 컴퓨터 데스크탑 환경을 보면서 제어할 수 있도록 해주는 원격 디스플레이 시스템	GPL
	Open SSH	1) 콘솔 기반의 원격 셸 (telnet, rsh 대응) 2) 매우 강력한 암호화 및 호스트 인증으로 보안에 최적	BSDL



# Open Source SW Licence Guide



소분류	오픈소스SW	제품 설명 및 특징	라이선스
Network monitoring tool	Gkrellm	X 윈도우 상에서 CPU 부하량, 프로세스, 디스크 액세스, 네트워크를 통한 서버 접속상황 등등을 한눈에 모니터링 가능	GPL
	Etherape	네트워크 상태를 그래픽으로 보여줌	GPL
	MRTG	1) SNMP를 지원하는 네트워크장비가 발생하는 트래픽을 모니터링 2) 지정한 시간마다 모니터링 결과값을 GIF(or PNG)이미지 생성하여 HTML 페이지로 뿌려주기 때문에 쉽게 트래픽을 알 수 있음	GPL
	Nagios	1) 웹 기반의 쉬운 작동 2) 각 인터넷 서비스 감시 3) 각 호스트의 상태 점검	GPL2
Video conference	GnomeMeeting	1) H.323과 호환되는 비디오 컨퍼런스 2) MS NetMeeting과 연동 가능 3) 비디오 컨퍼런스에 필요한 모든 기능 제공	GPL
Firewall tool	Kmyfirewall	KDE용 방화벽 구성 툴	GPL
	firewall-easy	1) 강력한 여러 기능 2) 매우 쉬운 설정 (거의 설정 필요 없음)	GPL
Network monitoring tool	Big Brother	웹 기반으로 쉬운 사용 웹 기반으로 어디서나 사용 실시간 네트워크 모니터링	BTFL / 상용 라이선스
Protocols analysing	Ethereal	1) 네트워크 프로토콜 분석기 2) 다양한 프로토콜 지원	GPL
	Tcpdump	네트워크 프로토콜 분석에 가장 기본이 되는 분석기	BSD License
Security scanner	Nessus	1) 원격 보안 스캐너 2) 분류된 결과 보고서와 다양한 포맷 변형 가능	GPL2
	Nmap	포트 스캐너의 가장 기본이 되는 분석기	GPL2
Sharing data/files	Samba	윈도우 시스템과 리눅스의 다양한 퍼미션을 통한 파일서버 및 프린터 서버 관리가능	GPL

## 나. Work with files

소분류	오픈소스SW	제품 설명 및 특징	라이선스
File manager	Konqueror	1) KDE용 파일 관리자 2) 웹 브라우저와 겸용	GPL2
	Nautilus	1) GNOME용 파일 관리자 2) 멀티미디어 파일을 비롯한 파일 쉽게 보기 지원 3) 테마지원으로 모양 변경 가능	GPL2
compressed files	Gnoup	1) 워집과 비슷함. 2) 프론트 엔드 형식으로 다양한 압축파일 지원	GPL2
Console archivers	tar, gzip, bzip2	압축과 아카이빙이 분리되어 있어서 더욱 강력한 작업이 가능	GPL

## 다. System Software

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Text editor	Kedit	KDE용 기본 문서 편집기	GPL2
	Gedit	GNOME용 기본 문서 편집기	GPL2
Console text editor	Vi	1) 유닉스 계열의 필수/기본 편집기 2) 정규식을 이용한 막강한 검색/치환기능 3) 키보드에서 손이 거의 움직이지 않는 편리한 명령키 배열 4) 개발용 문법 표시 기능 및 자동 들여쓰기	GPL
	Emacs	1) GNU 프로젝트의 시초가 된 편집기 2) LISP 언어 사용으로 학습가능 3) Add-on 기능으로 메일, 유즈넷, 게임 등 추가	GPL
Multi-purpose text and source code editor	Kate (KDE)	1) 파워풀한 하이라이팅 기능 2) 킥커 HTML 뷰어	GPL2
	Glimmer	1) 강력한 문법 표시 기능 (HTML을 비롯해 20개 이상의 언어 기본 지원) 및 괄호 검사 기능 2) 여러 파일 동시 검색/치환 기능 3) 단단계 실행 취소 기능	GPL
Viewing PostScript	GhostScript/ GhostView	1) PS와 PDF를 동시 지원 2) 거의 모든 플랫폼에서 사용 가능 3) 인쇄를 비롯한 다양한 기능 지원 및 다른 프로그램에서 불러 쓸 수 있음 4) GhostScript는 해석기, GhostView는 프론트엔드	GPL
Viewing PDF	Xpdf	1) 쉽게 사용 가능 2) 다양한 플랫폼 지원 3) PDF 전용 뷰어	GPL2

## 라. Multimedia(auto/CD)

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Programs for CD burning with GUI	K3b. (KDE)	1) 일반적인 CD 제작을 기본으로 비디오CD, CD 립, DVD 립 등 다양한 CD 제작 가능 2) 직관적인 그래픽 인터페이스 3) KDE 기반으로 만들어짐	GPL
	XCDRoast	1) 사용자나 그룹을 가리지 않고 사용 가능 2) 끌어 놓기를 지원 3) 여러 플랫폼을 지원	GPL2
	Gnome Toaster	1) 끌어 놓기를 지원 2) GNOME 기반으로 작성됨 3) On-the-Fly 방식의 CD 복사 지원 4) Disk-at-Once 모드로 On-the-Fly를 지원하는 유일한 프로그램	GPL
Music mp3 players	XMMS (X multimedia system)	1) 윈앰프와 비슷한 모양 및 비슷한 사용법 2) 윈앰프와 스킨을 공유할 수 있다. 3) 다양한 플러그인으로 쉽고 막강한 기능 확장	GPL

## 마. Multimedia(graphics)

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Graphic files viewer	Xnview	1) 간단한 사용법 2) 이미지 파일 조작 및 변환 가능 3) 썸네일 생성 4) 400개 포맷 보여주기 가능 5) 40개 포맷으로 내보내기 가능 6) 명령행 변환기와 개발자용 라이브러리 제공	Freeware
	Gqview	1) 쉬운 사용법 (끌어 놓기) 2) 기본적인 파일 관리 기능 제공 3) 디렉토리 브라우징 기능	GPL
graphic editor	Gimp	1) Gnome 기반의 그래픽 프로그램 2) 사진 리터칭과 이미지 작업에 최적화 됨	GPL
vector graphics	Sodipodi	벡터 그래픽 제작 기능	GPL
Flash creation	DrawSWF	JAVA2로 만들어져서 플랫폼에 구애받지 않고 실행 가능하며, 쉽고 간단한 플래시 제작 가능	GPL
	Ming	1) 플래시를 만들기 위한 라이브러리 2) C를 기본으로 C++, PHP, Python, Ruby에서 사용 가능	LGPL
3D-graphics	Blender	2002년 7월 톤 루센달에 의해 10원 라이선스를 GPL로 바꾸고, 2003년 2월 블랜더 2.6이 발표되어 블랜더의 부활을 다시 한번 꿈꾸고 있음	GPL
	KPovModeler	KDE기반의 프로그램 POV-Ray 파일을 만들어줌 계층적 객체 구조	GPL2
making screenshots	Ksnapshot	KDE 기반의 화면 캡처 프로그램	GPL

## 바. Multimedia(video and others)

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Video / mpeg4 players	Mplayer	1) 리눅스에서 가장 많이 이용하는 동영상 재생기 2) DivX/AVI/ASF/MPEG1/MPEG2 지원 3) DVD, VCD 지원	GPL2
DVD players	Ogle	1) 첫번째 Opensource DVD 플레이어 2) JPG로 저장, 전체모드 등 다양한 기능 제공	GPL
Professional video creation and editing	Cinella	영상 편집을 위해 필요한 무수히 많은 기능들	GPL2
	Avidemux	영상 편집, 추가, 절단, 필터링, 인코딩 지원	GPL2
Cutting video	Kino	음성, 영상 편집 기능 IEEE 1394 캡션 지원 99회까지 실행 취소 가능	GPL2
	Transcode	영상 편집 기능 제공 영상 크기 변경 가능 DVD를 비롯한 다양한 포맷 지원 포맷 변환 가능	GPL2
Converting video	Fmpeg	영상 포맷을 간단히 변환 가능 영상 스트리밍 서비스도 지원	GPL
	CinePaint	35mm 필름에 최적화된 리터칭 프로그램 해리포터, 스튜어트 리틀을 비롯한 여러 영화에 사용됨 (안정성과 기능이 입증됨)	GPL

## 사. Emulators

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Virtual machine emulator	VMWare for Linux.	1) 복수의 가상 시스템을 운영할 수 있음 2) 다양한 운영체제 지원 3) 사운드 입출력, SCSI, CD-ROM 드라이버 지원	상용
Windows emulator	Wine	1) 다양한 윈도우 어플리케이션 사용 가능 2) Win32 (9x/NT/XP), Windows 3.x and DOS binaries 지원 3) Win32 and Win16 함수 콜 지원	GPL2
	Mame	여러 플랫폼 지원 게임기들을 에뮬레이트	Mame License

## 아. Office and Business

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Office suite	OpenOffice	1) StarOffice에 기반한 오픈소스 오피스 2) MS Office와 완벽한 호환성 3) 높은 신뢰도 4) 유럽시장에서 약 40%의 점유율 보임 5) 다양한 플랫폼 지원	GPL/ LGPL/ SISSL
	StarSuite	1) OpenOffice에 몇 가지 유틸리티를 붙여 판매 2) Sun에서 제품 지원 및 보증 3) 약 11개 언어로 판매되고 있음	상용
	Koffice	1) KDE기반의 오피스 2) 다양한 기능을 갖는 프로그램이 융합되어 있음	GPL
	HancomOffice	1) 글을 리눅스로 포팅한 제품 2) 프리젠티와 스프레드 시트 등을 추가하여 판매 3) 한컴 리눅스에서 판매	상용
	Gnome Office	GNOME기반의 오피스 역시 다양한 프로그램이 섞여 있음	GPL2
Word processor	Abiword	GNOME Office에 포함되어 있는 워드프로세서	GPL
	Ted	많은 기능을 갖고 있는 워드프로세서	GPL
	Kword	KDE Office에 포함되어 있는 워드프로세서	GPL
	OpenOffice Write	OpenOffice에 포함되어 있는 워드프로세서	GPL
Spreadsheets	Gnumeric	GNOME Office에 포함되어 있는 스프레드시트	GPL2
	Kspread	KDE Office에 포함되어 있는 스프레드시트	GPL
	OpenOffice Calc	OpenOffice에 포함되어 있는 스프레드시트	GPL
Graphing / charting data	Kivio	Visio 스타일의 흐름도 프로그램 KDE Office에 포함되어 있음	GPL
	Dia	Visio와 비슷한 다이어그램 프로그램 GNOME Office에 포함되어 있음	GPL2
	OpenOffice Draw	OpenOffice에 포함되어 있는 그리기 프로그램	GPL

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Creating presentations	OpenOffice Impress	OpenOffice에 포함되어 있는 멀티미디어 프리젠테이션 프로그램	GPL
	Kpresenter	KDE에 포함되어 있는 프리젠테이션 프로그램	GPL
Local database	Kexi	KDE Office에 포함되어 있는 통합 데이터베이스 환경 액세스	GPL
	Gnome DB	GNOME Office에 포함되어 있는 데이터베이스 연결 프로그램	GPL2
Software for e-commerce and web business	JBoss	1) J2EE 기반의 WAS(Web Application Server) 중 EJB 컨테이너 2) J2EE 상품으로 기업용 시장에서 두각	LGPL
Personal finances manager	GNUCash	GNOME Office에 포함되어 있는 개인 재정 관리 프로그램	GPL
	GnoFin	1) GNOME에 기반한 개인 재정 관리 프로그램 2) 작고 사용하기 쉬움	GPL
Project management	Mr Project	1) GNOME Office에 포함되어 있는 프로젝트 관리 프로그램 2) 인터페이스가 Ms Project와 유사함	GPL2
	Toutdoux	GNOME Office에 포함되어 있는 프로젝트 관리 프로그램 데이터베이스 백엔드를 갖고 있음	GPL
ERP, CRM	Compiere	CRM과 통합된 ERP 프로그램 제품 지원을 받을 수 있음	MPL 1.1

## 자. Programming and Development

소분류	오픈소스SW	제품 설명 및 특징	라이선스
IDE	CodeForge	디버거가 통합된 환경	상용
	Kdevelop	1) KDE(QT), GNOME, C, C++ 지원 2) 프로젝트 관리 기능 3) 대화상자 편집기 4) 클래스 관리 도구 5) 통합된 디버거 6) 어플리케이션 마법사	GPL2
	Eclipse	IBM, 레드햇, 수세를 비롯한 유수업체 참여 ETRI도 참여하고 있음	CPL
	Glade	1) GTK (GNOME)와 C를 지원 2) XML 설치만으로 C++, Ada95, Python, Perl 지원 3) 쉽고 빠르게 비주얼한 사용자 인터페이스를 만들 수 있도록 해줌	GPL
Visual C++ IDE	Anjuta + Glade + Devhelp	MS Visual Studio와 같이 통합된 환경에서 컴파일, 디버거, 도움말 까지 참조할 수 있도록 됨	GPL2
	Kylix Open Edition	1) 오랜 경험의 볼랜드에서 만든 리눅스용 통합환경 2) 리눅스에서 델파이를 사용할 수 있는 유일한 도구	GPL
	Kdevelop	위 설명 참조	GPL

소분류	오픈소스SW	제품 설명 및 특징	라이선스
C++ IDE	GCC + vi/Emacs...	강력하고 편리한 콘솔 개발환경 쓸수록 빠져드는 매력적인 환경	GPL2
	Eclipse	위 설명 참조	CPL
Pascal	Freepascal	1) Turbo Pascal 7,0과 호환되는 문법 2) 인텔 계열과 모토롤라 계열의 CPU 지원 3) Linux, FreeBSD, DOS, OS/2, BeOS, SunOS, QNX, 아미가 등의 여러 OS 지원 4) 32bit 컴파일러	GPL
	GNU Pascal	1) 32/64bit 컴파일러 2) gcc가 작동하는 모든 시스템에서 사용 가능 3) 모든 GNU 프로그래밍 유틸리티와 호환 (gdb 포함) 4) Turbo Pascal 7,0과 호환되는 문법 5) Borland Delphi의 일부로 사용됨 6) 표준 및 확장 파스칼 문법을 완벽히 지원	GPL
Basic	wxBasic	1) Linux와 윈도우즈에서 사용 가능 2) QBasic과 비슷한 문법 3) 인터프리터 언어	GPL
	Gambus	1) 비주얼한 개발 환경 (Visual Basic과 비슷) 2) 컴파일러와 인터프리터로 모두 사용 가능 3) 통합된 개발 환경	GPL
	ScriptBasic	1) 확장과 임베디드가 수월함 2) 컴파일러와 인터프리터로 모두 사용 가능 3) 멀티 쓰레드를 지원	GPL
	X11-Basic	1) 아타리용 베이직과 비슷한 문법 2) X11 기능을 완전히 사용 가능 3) 인터프리터 언어로 작동 4) 셸이나 CGI로 사용 가능	GPL
	KBasic	1) KDE 용으로 개발된 베이직 2) 비주얼한 쉬운 개발 환경 3) 인터프리터 언어로 작동	GPL
	Blassic	1) 행번호가 반드시 있어야 하는 전형적 베이직 2) 고전적인 PEEK, POKE 명령 사용 가능	GPL
	YaBasic	윈도우즈와 유닉스에서 사용 가능	GPL
	XBasic	1) 윈도우즈와 리눅스에서 사용 가능 2) 통합된 그래픽한 개발 환경 제공 3) 컴파일러 언어로 작동	GPL/ LGPL
Prolog	GNU Prolog	1) 표준 프롤로그 지원 2) 저수준 디버거 제공 3) Native Code는 WAM-CC와 비슷한 속도 4) Byte Code는 WAM-CC보다 5배 정도 빠름	GPL
	Mercury	1) Logic 프로그래밍 언어이면서 2) 동시에 함수형 프로그래밍 언어	GPL or LGPL

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Assembler	NASM	1) 인텔 계열의 어셈블러 2) a.out 파일 형식 지원 3) 리눅스/BSD/COFF/MS16bit/MSwin32 파일 형식 지원	LGPL
	FLAT Assembler	1) 인텔 계열의 어셈블러 2) DOS/Linux/Windows 지원 3) MMX, SSE, SSE2, 3DNow 지원	BSDL
Debugger	gdb	1) 여러 언어에서 사용할 수 있는 디버거 2) 윈도우즈와 대부분의 유닉스에서 작동 가능 3) 로컬뿐 아니라 네트워크를 통해서도 실행 가능	GPL2
WYSIWYG html editor	Netscape / Mozilla Composer.	1) 모질라 웹 브라우저에 기본으로 붙어 있는 편집기 2) 문서 편집하듯 보면서 할 수 있음	MPL
	Openoffice HTML editor.	상동	GPL
HTML editor	Quanta Plus	1) KDE용 웹 개발 환경 2) 디렉토리 트리를 이용해 대규모 웹을 쉽게 제작 가능	GPL
	Bluefish	1) 숙련된 웹 디자이너와 개발자를 위한 에디터 2) 통합 환경을 제공 3) WYSIWYG 환경 제공 4) 다양한 언어를 지원	GPL
Java IDE	NetBeans	1) 자바용 통합 개발 환경 2) 자바 외에 C, C++, XML, HTML을 지원 3) CVS지원 4) 비주얼한 디자인 환경 제공 5) 마법사와 코드 생성 관리 기능 제공 6) JSP, XML, RMI, CORBA, JINI, JDBC, Servlet 지원	SPL
	Eclipse	위의 설명 참조	CPL
Graphical libraries	Tk (Tcl, C)	스크립트 언어인 TCL와 거의 함께 사용	BSD License
	Qt (C++)	C++을 기반으로 하며 윈도우즈용과 비슷한 컴포넌트제공	GPL
	GTK+ (C, C++)	C를 기반으로 하고, C++용 래퍼 있음	GPL2

#### 차. Scientific and Special Programs

소분류	오픈소스SW	제품 설명 및 특징	라이선스
CAD/CAM	Qcad	1) 강력한 2차원 캐드 프로그램 2) 많은 참여로 활발히 개발되고 있음	GPL2
Diagram and chart designer	Kivio (Koffice)	1) KDE 기반의 다양한 흐름도(Flow Chat) 프로그램 2) 기본 표준규격 지원 (letter, A4, etc...)	GPL
	Dia	GTK+ 기반의 다이어그램 생성 프로그램	GPL
	GNUplot	1) 콘솔 기반의 그리기 프로그램 2) 다양한 공식으로 그래프 생성 가능 3) 강력한 그리기 기능	GPL

소분류	오픈소스SW	제품 설명 및 특징	라이선스
GIS (Geographical information system)	Grass	1) 명령행으로 사용하거나 Tcl/Tk 그래픽 인터페이스 사용 가능 2) 윈도우에서도 실행 가능 3) Intel x86, Motorola PPC, SGI MIPS, Sun SPARC, Alpha AXP, HP PA-RISC, CRAY를 비롯한 여러 플랫폼에서 사용 가능 4) Linux/Intel, Linux/PowerPC, Solaris/SPARC, Solaris/86, SGI IRIX, HP UX, Mac OS X (Darwin), IBM AIX, BSD-Unix variants, FreeBSD 등의 운영 체제에서 사용 가능 5) 방대하고 부족함 없는 기능	GPL2
	Quantum GIS.	Unix/Linux에서 작동, PostgreSQL 지원, 지리 파일 화면 출력, 동적인 플러그인 지원, 지도 출력, 스크립트 엔진, 메타 데이터 지원	GPL
HDD testing / benchmarking	Bonnie++.	1) 2G 이상의 스토리지 지원 2) ReiserFS를 비롯한 여러 프로그램 지원 3) 동일 디스크의 다른 Zone의 성능 측정 가능	GPL2

### 카. Server Software

소분류	오픈소스SW	제품 설명 및 특징	라이선스
Web-server	Apache	1) 웹서버 시장 점유율 65% (2003년 3월기준) 2) 아파치 관련된 다양한 모듈 3) 멀티 플랫폼	Apache Software License 1.1
FTP-server	wu-ftp	레드햇 6.x 에서 가장 널리 쓰임	Wu-Ftpd Software License
	proftpd	1) 보안이 강화된 ftp 서버 2) 아파치 스타일 설정 파일로 관리가 쉬움	GPL
Language for Web-development	PHP	Apache 모듈로 포함되면서 Mysql과 함께 널리 쓰이는 서버 스크립트 언어	PHP License
	Perl	과거 시스템 관리 및 CGI 프로그래밍 시에 널리 이용되었던 서버 스크립트 언어	Artistic license
Database engine	PostgreSQL	가장 발전된 공개SW 데이터베이스 엔진	BSD License
	MySQL	1) 가장 많이 쓰이는 공개SW 데이터베이스 엔진 2) 멀티유저, 멀티쓰레드 지원	GPL2
Email server	Sendmail	1) 가장 널리 쓰이는 MTA(Mail Transfer Agent)	Sendmail License
	Qmail	1) 안정성에 가장 큰 기반을 두고 설계 2) 성능, 신뢰성 및 간편성 우수 3) 가장 널리 쓰이는 Sendmail의 아성에 도전	Qmail License
Email / PIM / Groupware server	Evolution	1) 그놈 기반의 개인정보관리 프로그램 2) 이메일, 캘린더, 통합정보관리, 일정관리, 메시지 보드 지원	GPL2
	Kroupware	독일 정부 지원으로 만들어진 KDE기반의 개인정보관리 시스템	GPL2
Proxy server	Squid	가장 널리 쓰이는 프락시 캐시 서버	GPL
	TinyProxy	설정이 필요없는 간단한 프록시 서버	GPL
Server for supporting Java Servlets	Tomcat	1) 자카르타 프로젝트의 하위 프로젝트로 J2EE 기반의 WAS(Web Application Server) 중 서블릿 컨테이너 2) 최신 버전 4.0, 개발버전 5.0.1	Apache Software License 1.1