

[OSS 라이선스 이슈사례]

오픈 소스 라이선스 실무 적용시 생기는 10가지 오해와 진실

한국소프트웨어진흥원
한글과컴퓨터 컨소시엄

<문서 정보>

일자	VERSION	변경내역	작성자
2008.12.05	1.0	내용 작성	박준완

목 차

1. 문서 개요	4
가. 문서의 목적	4
나. 참고사항	4
2. 공개SW 라이선스 신규 이슈 정보	5
가. 이슈사항	5
다. 이슈내용	5
3. 이슈 사항이 공개SW에 미치는 영향	10
4. 참고사항	11

1. 문서 개요

본 문서는 공개SW 도입컨설팅 사업의 공개SW 라이선스 분석지원 사업을 통해 현 산업에 적용되고 있는 라이선스 사례 및 신규 라이선스 이슈에 대한 정보를 제공하기 위해 제작되었다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- 라이선스 사례 및 신규 라이선스 이슈 정보 제공
- 오픈소스 라이선스에 대한 올바른 내용 홍보

나. 참고사항

- 공개SW 라이선스의 시장 변화에 대해 확인할 수 있다.
- 기존 공개SW 라이선스에 대한 분쟁 사례에 대해 확인할 수 있다.

2. 공개SW 라이선스 신규 이슈 정보

가. 이슈사항

구 분	내 용	비 고
이슈 정보	초보 개발자를 위한 오픈소스 라이선스 길잡이 - 오픈소스 라이선스 실무 적용시 생기는 10가지 오해와 진실	
발표 일자	2007년 12월 11일	
홈페이지	http://www.ibm.com/developerworks/kr/library/s_issue/20071211/	

나. 이슈내용

이 문서는 2007년 12월 11일 IBM developerWorks 임효준님이 기고한 글로써, 국내 개발자들이 개발시 참고해야할 사항들과 오픈소스 라이선스와 관련된 여러 가지 잘못된 인식에 대한 내용을 담고 있다.

오픈 소스 라이선스 실무 적용시 생기는 10가지 오해와 진실

오해 1. 오픈 소스 소프트웨어는 공짜이므로 마음대로 가져다 쓰면 된다

아직도 상당수 개발자들은 오픈 소스 소프트웨어를 아무런 책임 없이 마음대로 가져다 써도 되는 것으로 알고 있다. 오픈 소스 소프트웨어는 소스코드가 공개되어 있어 누구나 접근할 수 있는 것은 맞지만, 소스코드를 사용하는 경우 그에 따른 법적 책임도 뒤따른다. 이러한 법적 책임을 기술한 것이 바로 오픈 소스 소프트웨어 라이선스다.

제품 개발시 상용 소스코드를 구매해 사용하는 경우 가격이나 로열티 등과 기타 법적 의무 사항에 대해 충분히 검토한 후 사용할 것이다. 오픈 소스 소프트웨어도 마찬가지다. 오픈 소스 소프트웨어를 가져올 때에는 반드시 그에 따라 오는 라이선스를 읽어 보고 어떠한 의무사항이 있는지를 확인하고 사용해야 한다. 일반적으로 이러한 의무사항은 수정한 부분의 소스코드 공개, 저작권(copyright) 명시, 특허 등의 내용을 포함하고 있다.

그렇다면 소스코드의 소프트웨어 라이선스는 어떻게 확인할 수 있을까? 오픈 소스 소프트웨어는 대부분 소스코드 최상위 디렉터리에 COPYRIGHT, COPYING, LICENSE 등과 같은 이름의 파일이 존재하며 이 파일 안에 라이선스가 명시되어

있다. 또는 README 파일에 같이 명시되어 있기도 하다. 이러한 파일이 없는 경우에는 인터넷 검색 엔진을 통해 해당 소프트웨어 패키지 이름을 검색해 보면 라이선스에 대한 정보를 알 수 있다.

라이선스 문구는 법률적 용어로 기술되어 있어 일반 개발자가 이해하기에는 좀 어려울 수도 있는데 해당 라이선스 명을 인터넷 검색 엔진에서 찾아 보면 의무 사항을 알기 쉽게 풀이해 놓은 사이트나 FAQ 사이트를 찾아 볼 수 있다. 또 해당 오픈 소스 소프트웨어 홈페이지에도 이러한 라이선스 의무 사항을 명시하고 있는 경우가 많다.

오픈 소스를 가져와 활용할 때 개발자들이 **저지르기 쉬운 실수 중 한 가지는 소스 코드에 포함되어 있는 저작권 관련 문구를 지우는 것이다.** 별 생각 없이 오픈 소스의 일부 함수만 가져와 복사해 사용하면서 이러한 실수를 저지를 가능성이 크다. 그러나 오픈 소스 소프트웨어도 누군가가 공을 들여 작업한 결과물이므로 이러한 저작권 관련 문구는 반드시 포함시켜 두어야 한다. 향후 소프트웨어를 배포할 때 저작권 부분을 삭제한 경우 법률적으로 문제가 생길 수 있기 때문이다. 또 추후에 소프트웨어에 포함된 오픈 소스를 분석할 때에도 이러한 소프트웨어 라이선스 관련 문구를 남겨 두어야 추적이 쉽다.

오해 2. 오픈 소스 소프트웨어는 나와는 상관 없는 일이다

오픈 소스 소프트웨어를 쓰고 있지 않으므로 라이선스 이슈로부터 안전하다고 생각하는 개발자가 많다. 그러나 오픈 소스 소프트웨어는 이미 개발자들에게 깊숙이 침투해 있으며 오히려 오픈 소스 소프트웨어를 사용하지 않는 개발자를 찾아 보기가 힘든 실정이다. 더구나 소프트웨어 크기와 복잡도가 점점 증가하고 있기 때문에 오픈 소스를 사용하지 않고 개발하는 것은 앞으로 더욱 어려운 일이 될 것이다.

특히 오픈 소스 라이선스가 자신과는 별개의 일이라고 생각하는 임베디드 소프트웨어 개발자가 많다. 그러나 최근에 임베디드 시스템에도 리눅스 채택이 늘어나고 있으며 비-리눅스 제품 개발에도 알게 모르게 오픈 소스 소프트웨어가 많이 쓰이고 있다. 특히 네트워크 기능이나 보안, 이미지 처리 등과 관련된 소스코드는 오픈 소스 소프트웨어를 활용하는 경우가 많다. 따라서 이제는 소프트웨어 개발자라면 누구나 오픈 소스 소프트웨어 라이선스에 대한 기본 상식은 가지고 있어야 하는 시대가 온 것이다. 오픈 소스 라이선스에 대해 어느 정도의 이해가 있는 개발자라도 귀찮은 생각에 그냥 무시해 버릴 때가 많다. 그러나 오픈 소스 소프트웨어 의무사항을 준수하지 않으면 소속 회사가 소송을 당하게 되거나 개발 완료 직전에 이러한 문제가 발견되어 개발자가 어려움에 처할 수 있다.

오픈 소스 라이선스 위반 사례를 발견해 소송을 진행하는 단체로는 [gpl-violations.org](http://www.gpl-violations.org)가 대표적이다. 이 단체는 Herald Welte라는 독일인이 운영하며 리눅스에서 필수적으로 사용되는 `initrd` 소프트웨어의 저작권을 양도받아 권리를 행사하고 있다. 특히 역 공학(reverse engineering) 기법을 이용해 소스코드를 공개하지 않은 리눅스 기반 제품에 대해 25건 이상의 소송을 제기해 왔다.

최근에는 국내에서도 오픈 소스 라이선스 위반으로 인해 소송 직전까지 갔다가 결국 소스코드를 공개한 사례를 심심치 않게 볼 수 있다. 이제 오픈 소스 라이선스는 남의 얘기가 아니라 바로 우리 앞에 다가와 있는 것이다.

오해 3. 그렇다면 오픈 소스 소프트웨어를 쓰지 않으면 되겠군

오픈 소스 라이선스에 대해 어느 정도 이해를 하는 사람들이 오픈 소스 소프트웨어 라이선스 이슈를 피하려고 이와 같은 안을 제시하기도 한다. 특히 관리자 위치에 있으면 이러한 의견을 제시할 때가 많다. 그러나 오픈 소스 라이선스 이슈를 피하려고 오픈 소스 소프트웨어를 전혀 쓰지 않는 것은 모기를 잡자고 칼을 빼 드는 격이다. 소프트웨어 크기와 복잡도가 갈수록 커지고 있으므로 개발 효율성 측면에서 오픈 소스는 사용을 장려해야 하는 것이지 피해야 하는 대상이 아니다. 이러한 오해는 상용 소프트웨어 업체나 오픈 소스 라이선스 관련업 종사자들이 라이선스 관련 이슈를 너무 부풀리기 때문에 생기는 경우가 많다. 그러나 오픈 소스 소프트웨어 라이선스에 따르는 의무사항은 그렇게 어려운 것이 아니며 충분히 관리할 수 있는 수준이다.

오해 4. 오픈 소스 소프트웨어 라이선스를 지키는 일은 너무 복잡하고 힘든 일이다

개발자나 관리자들은 오픈 소스 소프트웨어 라이선스와 관련해 구체적으로 어떤 점을 유의해야 하는지 막막해 하는 경우가 많다. 이러한 막막함이 라이선스 준수가 어렵다는 오해로 진행하기도 한다. 그러나 오픈 소스 라이선스 준수는 그렇게 힘든 일이 아니다. 개발자가 오픈 소스를 활용할 때 가장 유의해야 할 점은 다음 세 가지를 들 수 있다.

소프트웨어를 설계할 때 소스코드 공개 범위를 인식하고, 공개하면 안 되는 부분에 대해서는 그에 대한 대응책을 마련해 설계할 것

특히 GNU GPL을 채택한 소프트웨어를 사용하는 경우 오픈 소스와 함께 컴파일되거나 링크되는 소프트웨어는 반드시 소스코드를 공개하여야 한다. 따라서 소스코드를 공개하면 안 되는 부분이 있으면 GPL 소프트웨어와 별도의 실행파일로 분리해 설계하거나 비슷한 기능을 하는 다른 오픈 소스 소프트웨어 사용을 검토하여야 한다.

제품을 출시할 때, 사용된 오픈 소스에 대한 저작권과 소스코드 획득 방법을 사용 설명서에 명기할 것

제품 출시시에는 반드시 사용 설명서에 저작권과 소스코드 획득 방법을 명시하여야 한다. 사용 설명서에 넣을 문구를 어떻게 작성해야 할지 막막하다면 오픈 소스 기반 소프트웨어에 포함된 사용 설명서를 참조하거나 리눅스 기반 제품에 포함된 매뉴얼을 참고하면 될 것이다.

외주 업체로부터 소프트웨어를 받는다면 사용된 오픈 소스 소프트웨어에 대한 리스트를 받을 것

외주 업체가 소프트웨어를 개발해 공급하는 경우에도 어떤 오픈 소스 소프트웨어를 썼는지 반드시 파악하여야 한다.

두 가지 이상의 라이선스로 배포되는 오픈 소스 소프트웨어들이 있다. 이러한 라이선스를 듀얼 라이선스(dual license) 또는 트라이 라이선스(tri-license)라고 한다. 모질라(Mozilla), 프리타입(freetype), 펄(Perl), MySQL, QT, 버클리(Berkeley) DB 등이 대표적인 예다. 이러한 소프트웨어의 경우는 여러 가지 라이선스 중에서 자신의 제품에 가장 적합한 라이선스를 채택하여야 할 것이다. 예를 들어 MySQL을 제품에 탑재한 경우 같이 결합된 소스코드를 공개하여도 무방하다면 GPL을 채택할 수 있을 것이다. 소스코드를 공개하지 않아야 한다면 MySQL 상용 라이선스를 채택할 수 있을 것이다.

오해 5. 오픈 소스를 사용해 만든 제품은 전체 소스를 모두 공개해야 한다

임베디드 시스템 개발에서 오픈 소스 소프트웨어 라이선스가 잘 지켜지지 않을 때가 많은데 바로 이러한 오해에 기인한다. 오픈 소스를 사용해 제품을 만들더라도 전체 소스를 모두 공개하는 것이 아니라 오픈 소스 라이선스에서 명시한 범위까지 공개하면 된다. 예를 들어 리눅스를 사용해 임베디드 시스템을 개발하는 경우에는 일반적으로 다음 부분의 소스코드를 공개하면 된다.

리눅스 커널

셸, 비지박스(busybox) 등 GNU GPL 기반 시스템 애플리케이션

C 라이브러리 및 기타 GNU LGPL 기반 라이브러리들

오픈 소스를 활용한 애플리케이션

즉, **오픈 소스를 활용하지 않은 독자적인 애플리케이션은 소스코드 공개의 의무가 없는 것이다.** 오픈 소스를 활용했더라도 BSD 스타일의 오픈 소스에는 소스코드 공개 의무가 없다. 따라서 초기 설계만 잘 한다면 소스코드 공개 의무를 충실히 지키면서도 회사의 지적 재산권도 지킬 수 있다.

오해 6. 소스코드는 반드시 판매되는 제품에 포함되어야 한다

소스코드 공개 의무가 있는 오픈 소스를 활용했더라도 소스코드를 반드시 제품에 포함하여 배포해야 하는 것은 아니다. 대표적으로 **GNU GPL의 경우는 제품 설명서에 소스코드 획득 방법만 명시하여도 된다고** 기술하고 있다. 따라서 소스코드를 획득할 수 있는 웹 페이지를 설명서에 명시하거나 연락처만 명시해 두고 추후에 사용자가 요청할 때 제공해도 된다. CD 등의 저장매체를 통해 제공하면 저장매체 가격 및 배달료 등 제공에 필요한 실비를 사용자에게 청구할 수 있다.

오해 7. 오픈 소스 소프트웨어 라이선스는 개발이 다 끝나고 대응하면 된다

개발자가 일정에 쫓기면 라이선스 문제는 제쳐 두고 개발이 다 끝난 후 대응하겠다는 생각을 하게 되는 경우가 많다. 그러나 개발을 마친 후 오픈 소스 라이선스

문제로 인해 소프트웨어 구조를 변경해야 한다면 작업량은 상당히 늘어날 수 있다. 따라서 오픈 소스 라이선스 검토는 가능하면 개발 초기에 진행하는 것이 좋다.

오해 8. 외주 업체에서 개발한 소프트웨어의 오픈 소스 라이선스는 내가 검토할 필요가 없다

외주 업체에서 개발한 소프트웨어에 오픈 소스가 포함되어 있고 라이선스 의무를 준수하지 않은 경우 1차적인 책임은 최종 소프트웨어를 배포하는 회사 측에 있다. 따라서 외주 업체를 통해 소프트웨어를 공급받았다면 그에 따른 라이선스 의무사항이 잘 지켜지는지 반드시 확인하여야 한다.

그러나 현실적으로 영세한 외주 업체에서, 개발에 쓴 오픈 소스에 대해 충분히 검토를 하지 못할 수도 있다. 이러한 경우는 소프트웨어에 포함되어 있는 오픈 소스를 자동으로 검출해 주는 Blackduck사의 ProtexIP나 Palamida사의 제품을 사용해 어느 정도 관리할 수 있다.

소스코드에 포함되어 있는 오픈 소스를 찾아 내는 또 한 가지 방법은 소스코드에 포함된 라이선스 관련 문구를 검색해 보는 것이다. 예를 들어 리눅스에서 GNU GPL로 된 소스코드가 있는지 검색해 보고 싶다면 소스코드 최상위 디렉터리에서 다음 명령어를 수행하면 된다.

```
> grep -r GNU *
```

단 이 방법은 개발자가 라이선스 관련 문구를 임의로 삭제하지 않은 경우에만 사용할 수 있을 것이다.

오해 9. 모든 오픈 소스는 마음대로 결합해 사용할 수 있다

일부 오픈 소스 라이선스들은 서로 상충되는 라이선스 문구가 있어 하나의 소스코드에 같이 결합하는 것이 허용되지 않는다. 예를 들어 GPLv2와 아파치 라이선스는 서로 상충되는 라이선스 조건 때문에 하나의 소스코드에 결합하여 사용하는 것이 불가능하다.

이러한 오픈 소스 소프트웨어 라이선스의 비호환성 문제는 오픈 소스 소프트웨어의 2/3 정도를 차지하는 GPL에 대해 가장 잘 분석되어 있다. GPL과 호환되는 오픈 소스 소프트웨어 라이선스에 대해서는 다음 웹 사이트를 참고하기 바란다.
<http://www.gnu.org/philosophy/license-list.html>

오해 10. 내부 목적으로 사용하더라도 소스코드를 외부에 공개해야 한다

일반적으로 오픈 소스 소프트웨어의 **소스코드 공개 의무는 개발 시점이 아니라 배포 시점에 발생**한다. 따라서 외부에 배포하지 않고 내부적인 목적으로만 사용하면 외부에 소스코드를 공개할 의무가 없다. 따라서 사내 인트라넷 시스템 구축 목적으로 오픈 소스를 활용하거나 디버깅 목적으로만 사용하고 배포되는 제품에 탑재되지 않은 오픈 소스는 코드 공개 의무가 없다.

3. 이슈사항이 공개SW에 미치는 영향

위 내용은 2007년 12월 11일자로 IBM developerWorks에 임효준님이 기고한 글로써, 초보 개발자를 위한 오픈 소스 라이선스 길잡이의 Part3 기고 글이다. 이 글에는 오픈소스 라이선스에 대한 설명이 잘 되어있어 초보 개발자 및 오픈소스 라이선스를 사용하려는 기업 또는 사용하고 있는 기업들이 가지고 있던 편견을 벗어 버리는데 상당히 유용한 글이라 생각된다. 아직까지도 많은 개발자 및 기업에서는 오픈소스 라이선스에 대한 위험성을 인지하지 않고 있거나 특별하게 생각하지 않는다. 오픈소스 라이선스에 대해 가장 많은 오해를 가지고 있는 것이 '오픈소스 라이선스를 사용하면 모든 소스를 오픈해야 한다' 라는 생각과 '오픈소스를 차용하더라도 저작권 관련된 내용을 지우면 알기 힘들 것이다' 라는 생각이라고 판단된다. 위 본문에도 나와있듯이 오픈소스 라이선스는 적용되는 소스내에서만 오픈에 대한 의무를 가지고 있으며, 직접 개발한 소스에 대해서는 오픈하지 않아도 된다. 또한 오픈소스를 차용하면서 저작권 관련 내용을 지운다 하더라도 최근 많이 개발되어 있는 여러 가지 솔루션들에 의해 발견되기 쉽다. 실제 여러 임베디드 기업에서 수출을 타진하다 오픈소스 라이선스 문제로 인해 수출을 포기하는 경우도 종종 발생하고 있다. 위에 기고 되어 있는 글만 인식하고 있다면 국내 개발자들도 오픈소스의 장점을 충분히 활용하면서 독자적인 솔루션을 개발할 수 있을 것으로 판단된다.

4. 참고사항

출처

1. http://www.ibm.com/developerworks/kr/library/s_issue/20071211/

(*원저작자 및 IBM Developerworks 담당자의 승인을 통해 인용했음을 알립니다.)