오픈소스 소프트웨어 라이센싱의 법적 의미

인하대학교 김병일

1. 서 언

최근 오픈소스 운동(Open Source Movement)은 법률분야에서도 주목을 받고 있다. 다양한 종류의 오픈소스 라이센스가 상업적으로 널리 이용될 수 있다. 이 경우 발생할 수 있는 법적 문제에 대하여 법률가들은 해답을 제시하여야 한다. 그러나 많은 경우에 그 해답을 찾는 것은 쉽지 않다. 왜냐하면 오픈소스 라이센싱은 전통적인 저작권 라인센싱과는 많은 차이가 있기 때문이다. 즉 저작권법 접근과 오픈소스 접근에는 근본적인 차이가 있다.

본 글은 오픈소스 라이센싱에 관한 법적 문제 중 가장 중요한 사안만을 다루고자 한다. 따라서 본 글의 목적은 오픈소스 라이센싱에 관한 모든 문제들을 다루고 있지는 않는다. 예컨대, 오픈소스 프로그래밍 프로젝트 참여와 독점적인 소프트웨어[1] 기업을 위한 작업은 동시에 어려운 문제를 야기하고 있지만, 이것은 매우 광범위한 문제여서 본 글에서 커버하는 것은 불가능하다.

2. 개념정의

2.1 오픈소스 소프트웨어 및 오픈소스 라이센스의 의의

오픈소스의 단순한 정의 - 종종 '자유 소프트웨어'[2]로 언급되는 - 는 소프트웨어의 소스코드가 일반에게 공개되는 것을 말한다. 어떤 프로그램이 오픈소스로써 분류되기 위해서는, 오픈소스 프로그램의 배포조건이 다음과 같은 기준을 따라야 한다.[3]

① 소스 코드

프로그램은 소스코드를 포함하여야 하며, 컴파일된 형태뿐만 아니라 소스코드 형태로도 배포가 허락되어야 한다. 어떤 형태의 제품이 소스코드와 함께 배포되지 않는 경우는 반드시 인터넷 등과 같이 무료로 소스코드를 다운로드 받을 수 있는 공지된 수단이 존재해야 한다. …… 고의로 소스코드를 복잡하게 하는 것은 허용되지 않는다. 전처리기(preprocessor) 또는 번역기(translator)의 결과물 같은 중간형태(intermediate forms)도 허용되지 않는다.

② 저작자의 소스코드의 완전성(integrity)

저작자가 파일, 소스코드나 목적코드가 수정되는 것을 제한할 수 없도록 강력하게 권고되고 있다. 라이센스는 수정된 소스코드로 제작된 소프트웨어의 배포를 명시적으로 인정해야 한다. 라이센스는 2차적 저작물의 이름이나 버전숫자(version number)를 원소프웨어와 다르게 할 것을 요구할 수 있다.

③ 2차적 저작물(소프트웨어)

(오픈소스 소프트웨어) 라이센스는 소프트웨어의 변경과 2차적 저작물을 허용해야 하며, 그것들을 소프원오픈소스 소프트웨어의 라이센스와 동일한 조건하에 배포되는 것을 허용해야 한다.

④ 자유로운 재배포

라이센스는 어느 당사자가 오픈소스 소프트웨어를 무료로 배포하거나 판매하는 것을 제한할 수 없다. '비공개소스 소프트웨어 - 또는 '사적재산으로 보호되는 소프트웨어(proprietary software)'는 상술한

1) 오늘날 소프트웨어라는 용어는 매우 일반적으로 사용되고 있다. 예를 들면, 레코드 플레이어에 대한 레코드·VTR에 대한 비디오 테이프 등도 역시 소프트웨어인 것이다. 그러나, 이하에서는 컴퓨터 하드웨어에 대한 소프트웨어만을 "소프트웨어" 라 지칭하기로 한다.

2) See, e.g., Free Software Foundation, What is Free Software?(visited Oct. 16, 1999), http://www.gnu.org/philosophy/free-sw/html>

3) 그러나 이러한 열거기준은 제한적 열거가 아니고 예시적인 것에 불과하다는 것에 주의해야 한다.

기준을 충족하지 못하는 소프트웨어로 정의된다. 예컨대, '사적재산으로 보호되는 소프트웨어(proprietary software)'의 경우 일반적인 이용자는 소스코드에 접근할 수 없으며, 소스코드를 수정하는 것이 허용되지 않으며, 대개는 소프트웨어를 재배포하는 것도 금지된다.

오픈소스 소프트웨어는 그것에 관련한 다양한 형태의 오픈소스 라이센스를 가지고 있다. 실제로 특정 프로그램이 '오픈소스'로써 정의되기 위해서는 그 프로그램이 일정형태의 오픈소스 라이센스와 함께 배포되어야만 한다. 즉, 오픈소스 라이센스의 이용이 오픈소스 프로그램을 'open source'로 만들 때 가장 중요한 요소이다. 따라서 오픈소스 라이센스가 법적으로 유효한 지의 여부는 오픈소스 개발모델이 성공여부와 관련하여 가장 중요한 관건이다.

2.2 자유(free)와 공개(open)

오픈소스 운동내의 몇몇 그룹은 오픈소스에 대하여 'free' 또는 'free software' 단어를 사용하는 것을 선호하고 있는 것에 유의해야 한다.[4] 그러나 대부분의 오픈소스 공동체는 'open' 또는 'open software' 단어를 선호하고 있는 것으로 보인다.[5] 다만, 주의해야 하는 것은 'free' 용어가 사용되는 경우, 그것이 무료(zero price)와 같은 의미로 사용되는 것이 아니라, 오픈소스 소프트웨어와 관련하여 제한으로부터의 자유(freedom)를 의미한다.[6] 즉 자유 소프트웨어란 누구에게나 이용과 복제, 배포가 자유롭고, 특히 소스코드에 대한 접근을 통하여 개작과 재배포가 자유로운 소프트웨어를 말한다.

3. 오픈소스 운동(Open Source Movement)의 略史

3.1 초기 : GNU 및 Perl - 1983 ~ 1990

1984년 리차드 스톨만(Richard Stallman)은 자유소프트웨어 재단(Free Software Foundation: 이하 FSF)을 설립하였다. FSF의 주요한 기술적 목표는 "GNU"라 불리우는 오픈소스 Unix같은 운영체계를 개발하는데 있었다. 비록 스톨만은 GNU 운영시스템의 'kernel'(central module)을 완성할 수 없었지만, 그와 FSF 관련 프로그래머들은 GNU EMACS 및 GNU C compiler와 같이 매우 유용하고 인기있는 오픈소스 소프트웨어를 개발할 수 있었다. 그러나, 아마도 GNU 프로젝트의 가장 중요한 목적은 소프트웨어 개발에 관한 스톨만의 철학을 촉진하는데 있었다. 스톨만은 "정보는 공동체 재산이고 모든 소프트웨어 소스(코드)는 공유되어야 한다"고 믿었다. 스톨만의 생각에 의하면, '사적재산으로 보호되는 소프트웨어(proprietary software)는' 소프트웨어 개발 촉진에 필요한 협동과 공동체정신을 방해한다고 한다. '사적재산으로 보호되는 소프트웨어(proprietary software)'의 문제점을 해결하기 위한 FSF의 해결책은 스톨만이 'copyleft'라 부르는 개념이었다: 모든 사람이 소프트웨어를 교환하거나 배포할 수 있도록 소스코드를 공개하게 하여야 한다. 'copyleft'의 개념을 실현하기 위하여, 스톨만과 FSF는 최초의 오픈소스 소프트웨어 사례인 GNU General Public License 개발하였다.

한편 1987년에 오픈소스 소프트웨어에 주목할 만한 발전이 있었다. Larry Wall은 텍스트파일을 스캔, 복사 및 출력하기 위하여 Unix에 기초한 프로그램언어인 Perl을 발명하였다. 처음에는 Wall은 GNU GPL하에 이용자넷트워크에 Perl을 설정하였다. 그러나 월은 GPL 조건이 지나지게 제한적이라는 이유로 "Artistic License"라고 불리는 그 자신의 오픈소스 라이센스를 개발하였다.

3.2 오픈소스 공동체(Open Source Community)의 등장 : Linux, FreeBSD, Apache 및 인터넷혁명 - 1990 ~ 1997

1980년대 말까지는 Unix같은 운영체계를 개발하기 위한 FSF의 시도 및 스톨만의 GNU kernel을 실현하고자 하는 것은 달성되기 어려워 보였다. 그러나 Unix같은 오픈소스 운영시스템의 꿈은 21세의 핀란드 대학생인 Linus Torvalds에 의하여 실현되었다. 리누스 및 그 동료들은 오픈소스에 기초한 새로운 운

4) See, e.g., Free Software Foundation, Categories of Free and Non-Free-Software (visited Oct. 16, 1999), http://www.gnu.org/philosophy/categories/html〉

5) See, e.g., Open Source Initiative, Open Source.org(visited Oct. 16.1999) 〈http://www.opensource.org〉

6) See, e.g., Free Software Foundation, What is Free Software?(visited Oct. 16, 1999), http://www.gnu.org/philosophy/free-sw/html〉

영시스템을 개발하여 'Linux'라 명명하였다. 리눅스 개발시, 리누스 토발즈는 주로 GNU 프로젝트에 의해 개발된 수단과 필요한 보조적 프로그램에 주로 의존하였다. 리누스 토발즈는 GNU 프로젝트의 기여를 존중하여 Linux를 'copylefted'하였다. 즉 리누스 토발즈(Linus Torvalds)를 중심으로 개발이 진행되고 있는 리눅스 커널(kernel) 부분이 GPL을 따르고 있다. 또 리눅스에서 작동되는 프로그램도 대부분 GPL을 따른다. 나아가서는 리눅스를 상품화한 기업이 개발한 상업용 프로그램도 많은 부분이 GPL 호환(GPL-compatible) 라이센스에 따라 공개되고 있다.[7] GNU C 라이브러리에는 'GNU Library GPL'이라 불리는 특별한 종류의 카피레프트가 적용된다. 이것은 사적재산으로 보호되는 소프트웨어(proprietary software)와의 링크를 허용하는 것으로서, 지금은 'GNU Lesser GPL(LGPL)'이란 명칭으로 변경되어 사용된다. 이와 같이 소프트웨어의 카피레프트 라이센스가 GPL이고 라이브러리의 카피레프트 라이센스가 LGPL이다.[8] Linux는 1990년대에 개발된 유일한 오픈소스 운영시스템이 아니다. 1993년, 다른 Unix의 오픈소스 버전인 FreeBSD가 인터넷 및 CD-Rom을 통하여 배포되었다.

동시에 오픈소스 소프트웨어 개발모델이 시대적 흐름으로 자리잡기 시작하였는데, 여기에는 인터넷이 중요한 역할을 담당하였다. 인터넷 이용자는 매년 기하 급수적으로 증가하고 있는데, Perl, 1997년 BIND 및 Send-mail과 같은 주요 컴포넌트 및 프로토콜의 대부분은 사적재산으로 보호되지 않는 (non-proprietary) 오픈소스이다.

3.3 오픈소스 발전 : 오픈소스 운동의 관심 증대 및 오픈소스 소프트웨어의 商用化 - 1998 ~ 현재

한편 최근에는 오픈소스 운동, 특히 Linux가 상당히 주목을 받고 있다. 또한 자유 소프트웨어란 용어 대신 '오픈 소스 소프트웨어(open source software)'란 용어가 널리 사용되고 있다. 1997년 에릭 레이몬드(Eric Raymond)는 '성당과 시장(The Cathedral and the Bazaar)'이란 제목의 글에서 리눅스가 어떻게 개발되었는지에 대한 개발방식에 주목하게 된다. 그는 리눅스가 GNU 프로젝트에서 수행하는 그 역할과 사회적 의미보다도 그것이 개발되는 방식에 주목하였다. 이 글에서 그는 몇몇 우수한 해커들에 의해 개발되는 폐쇄적 방식과 리눅스처럼 수많은 사람들이 인터넷을 통하여 자유롭게 개발하고 발전시키는 개방적 방식을 각각 중세 몇몇 뛰어난 건축가들에 의해 만들어지는 성당과 수많은 사람들이 북적대는 시장에 비유하였다. 그는 리누스 토발즈가 리눅스를 만들었다는 사실보다 리눅스 개발모델을 만든 것에 더 큰 의미를 부여하였다.[9] 이와 같이 프로그램의 공유와 사용의 자유라는 측면보다 개발방식의 효율성을 강조하는 사람들은 자유 소프트웨어라는 용어대신에 오픈 소스 소프트웨어라는 용어를 주로 사용한다. 따라서 오픈 소스 운동은 자유 소프트웨어 운동과는 달리 좀 더 현실적이고 실용성을 중시하는 방향을 취하고 있다.

상용화된 오픈소스 소프트웨어(commercialized open source software)의 등장은 오픈소스 라이센스가 어떻게 구성되고 유효성(강제집행성)이 있는지의 여부에 촛점을 던지고 있다.

4. 오픈소스 라이센싱

4.1 소프트웨어의 저작권에 의한 보호

소프트웨어는 영업비밀, 특허, 저작권에 의한 법적 보호가 가능하다. 특히, 현행 저작권법은 컴퓨터 프로그램이 저작권법의 보호대상임을 명문으로 규정하고 있으며(저작권법 제4조 제1항 제9호), 구체적으로는 컴퓨터프로그램보호법에 의해 소프트웨어의 저작권이 보호되고 있다(저작권법 제4조 제2항). 컴퓨터 소프트웨어는 소스코드와 목적코드의 경우 저작권에 의하여 보호를 받을 수 있다. 다만, 오픈소스 소프트웨어와 관련하여 주의하여야 하는 것은, 그것이 소스코드 또는 목적코드 형태이건 간에 전체로써 저작권으로 보호되는 것이지, 모든 컴포넌트가 보호되는 것은 아니라는 것이다. 저작자는 그 저작물을 복제할 권리 (저작권법 제16조)와 저작물의 원작품이나 그 복제물을 배포할 배타적인 권리를 갖는다 (저작권법 제20조). 계약적 관점에서 볼 때, 소프트웨어가 주로 저작권법에 의하여 보호된다는 사실은 매우 중요하

7) 클리프 밀러, 이규원 옮김, 『리눅스 비즈니스.com』, 세종서적, 2000, 130쪽 참조

8) GNU Lesser General Public License, 〈http://www.gnu.org/copyleft/lesser.html〉

9) 주철민, 「자유 소프트웨어 운동:리눅스 운동을 중심으로」, 『다른과학』, 한울, 2000, 봄·여름호, 78-79쪽

다. 왜냐하면 그 결과로서 소프트웨어의 이용은 일반적으로 권리보유자로부터 사용허락 (licence)을 필요로 한다.

한편, 최종 사용자를 일방 당사자로 하는 소프트웨어 거래의 대부분이 매매계약이 아닌 "사용허락계약" 의 형태로 체결되고 있는데, 이러한 계약의 법적 성질을 민법상 매매계약으로 이해할 수 있을지가 문제되고 있다.[10] 오픈소스 소프트웨어 라이센싱의 과정은 비공개 소스 소프트웨어 라이센싱과 여러가지 면에서 유사하다. 그러나 기존 소프트웨어 개발업자와는 달리 오픈소스 소프트웨어 개발자는 그들의 소프트웨어 라이센싱에 다른 동기를 가지고 있기 때문에 '사적재산으로 보호되는 소프트웨어(proprietary software)'에 적용되는 이론을 그대로는 적용할 수 없다. 다만, *'copylefting'*과 관련하여 양자간에는 현격한 차이가 있다.

4.2 주요 오픈소스 라이센스의 사례[11]

(1) GNU GPL

GPL의 핵심은 프로그램을 소스코드의 형태로 전전양도하면서 이에 대한 이용과 복제, 수정, 배포의 자유뿐만 아니라, 수정된 프로그램에 대해서도 동일한 자유가 계속해서 순차적으로 보장되도록 한 것이다. GPL은 저작권을 전제로 하면서, 저작권의 본래의 취지를 반대로 이용하여 소프트웨어를 사적인 재산권의 대상으로 삼는 대신에 자유롭게 이용, 복제, 배포, 수정될 수 있는 수단으로 삼은 것이다.[12]

(2) Artistic License

'Artistic License'는 원래 Larry Wall이 자신의 오픈소스 *Perl* 프로그래밍 언어를 커버하기 위하여 개발되었지만, 다른 오픈소스 소프트웨어를 라이센스하는데 이용되었다. 'Artistic License'는 라이센싱 조건이 GNU GPL보다 완화되어 있다. GNU GPL과는 달리, 'Artistic License'는 라이센시가 라이센스된 프로그램의 개작에 대한 지적재산권을 주장하는 것을 금지하지 않는다. 따라서 'Artistic License'는 원소스코드에서 유래된 2차적 저작물의 경우에 비공개 소프트웨어와 취급에 차이가 없다. 또한 'Artistic License'는 비공개 소스 프로그램과 공개소스 프로그램의 혼합에 대해서도 제한을 두지 않는다. 요컨대, 'Artistic License'는 GNU GPL 만큼 라이센스에 많은 제한을 두고 있지 않은 라이센스 형태이다.

(3) BSD-style Licenses

'버클리 소프트웨어 배포판(Berkeley Software Distribution, BSD)'는 원래 'FreeBSD'로 알려진 BSD Unix의 변형체를 위한 오픈소스 라이센싱 형태로서 이용되었다. 오픈소스 공동체가 성장, 발전함에 따라 BSD 라이센스는 종종 수정된 형태로 Apache Group을 비롯한 많은 오픈소스 개발자에 의하여 이용되고 있다. 자유 BSD 라이센스에서는, 가령 자유 BSD를 컴퓨터에 설치하여 판매하면서 소스 코드를 보여주고 싶지 않다면 비공개로 할 수도 있다. 또한 누군가가 자유 BSD를 수정하여 다른 운영체제를 만들었을 때도 그 소스 코드를 공개하지 않을 수 있다. 즉, BSD 유형의 라이센스들은 원칙적으로 오픈소스 소프트웨어의 무제한적인 상업적 이용을 허용하면서 사적재산으로 보호되는 2차적 저작물(proprietary derivative works)의 개발도 무제한적으로 허용한다.

(4) *Netscape Public License* 및 *Mozilla Public License*

이 밖에 고스트스크립트, 넷스케이프, 센드메일 등의 소프트웨어가 채택하는 라이센스도 주목되는데, 가령 넷스케이프의 소프트웨어에 사용되는 '모질라 공개 라이센스(Mozilla Public License, MozPL)'와 그 변형인 '넷스케이프 공개 라이센스(Netscape Public License, *NPL*)'는 '소프트웨어의 비공개(software hoarding)'를 억제하는 데에는 BSD 유형의 라이센스들보다 뛰어난 것이지만 여전히 개발자들에게 사적재산으로 보호되는 2차적 저작물을 창작할 수 있도록 허용한다.

(5) 비교

다음 표는 주요 오픈소스 라이센스의 차이점과 유사점을 요약한 것이다.

주의해야 하는 것은 여기서 검토한 오픈소스는 오픈소스 라이센스에 필요한 기본적 기준 외에 공통된

---

10) 이에 대한 자세한 설명은, 김병일, 소프트웨어 라이선싱, 계간 저작권, 2001년 가을호 제55호, 14-29쪽 참조

11) 법적 관점에서는 오픈소스는 4 카테고리로 분류할 수 있다. 이 카테고리는 주로 라이센스가 2차적 저작물의 문제에 대해서 어떻게 표현하고 있는가에 관련된다. Kennedy, D.M., A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft, Copyfuture, (http://www.denniskennedy.com/opensourcedmk.pdf)

12) http://www.oreilly.com/catalog/opensources/book/stallman.html 참조

오픈소스 라이센스	2차적 저작물도 반드시 open/freely 이용가능해야 한다. 수정(개작)된 것이 비공개되서는 않된다.	라이센스된 오픈소스 소프트웨어는 비공개소스 소프트웨어와 통합되거나 결합될 수 없다.
GNU GPL	Yes	Yes
Artistic License	No(대부분의 경우)	No(대부분의 경우)
BSD-style licenses	No	No
NPL	Yes	No
MozPL	Yes	No

특징을 하나 가지고 있다. 즉 모든 오픈소스 라이센스는 모든 워런티 책임(warranty)을 부인하고 있다. 이는 오픈소스 라이센스가 비공개소스 라이센스와 공통점을 가지고 있는 것 중의 하나이다.

4.3 오픈소스 라이센스의 법적 유효성(강제 집행성)

오픈소스 라이센스의 가장 중요한 법적 측면은 라이센스 조항의 유효성(validity)과 국내법하에서의 적용가능성이다. 다른 주요한 문제는 오픈소스 라이센스의 실제 사용과 관련된다. 예를 들면, GPL은 소프트웨어 사업에서 모든 상황에 적절한 라이센스는 아니다. 소프트웨어 라이센스의 유효성 문제는 새로운 것은 아니다. 예를 들면 shrink-wrap 라이센스 및 click wrap 라이센스의 유효성에 관하여 논쟁이 계속되고 있다. 전통적 계약법적 원칙은 이러한 새로운 유형의 라이센스에 대하여 대처를 하지 못하기 때문에, 특정 라이센스 규정이 법원에서 강제집행할 수 있는지는 종종 불분명하다. shrink-wrap 라이센스의 기본적 개념은 사용자는 그가 소프트웨어 패키지를 개봉할 때부터 라이센서가 제시한 라이센스 조건들에 구속된다는 데 있다.[13] 또한 웹사이트에서 이용되는 click wrap 라이센스 및 click through 라이센스와 같은 다양한 형태의 shrink-wrap 라이센스가 존재한다. 중요한 문제는 라이센스 계약이 이러한 방식에 의하여 체결될 수 있는가이다. shrink-wrap 라이센스에 대한 일반적인 접근방법은 없으며, 국제조약도 이 문제에 대해서 언급하고 있지 않다. 따라서 shrink-wrap 라이센스의 강제집행성은 국내법 및 법원의 실무에 달려 있다.

shrink-wrap 라이센스의 강제집행성에 관한 문제는 오픈소스 라이센스와 관련해서도 중요하다. 왜냐하면 오픈소스 라이센스 계약은 대부분 shrink-wrap 라이센스와 유사하게 체결되기 때문이다. 그러나 비공개소스 소프트웨어의 shirink wrap · 라이센스 계약과는 달리 오픈소스 소프트웨어의 shirink wrap · 라이센스 계약 방식은 법적으로 유효할 가능성이 높다. 왜냐하면 일반적인 오픈소스 라이센스의 조건은 오픈소스 개발자의 지적재산권을 강화하기 보다는 약화하는 것이기 때문이다.

또한 오픈소스 라이센스의 유효성의 문제는 부당성(unconscionability)의 문제에 의하여 영향을 받지 않는다. 대부분의 비공개소스 소프트웨어와는 달리, 오픈소스 라이센스는 이용자에게 권리를 제한하기 보다는 부여한다. 소스코드를 볼 수 있는 권리, 자유로운 재배포 권리 및 오픈소스 라이센스에 근거한 기타 권리가 이용자에게 불공정한(unfair)한 것이라고 하는 것은 불합리한 주장일 것이다. 비공개소스 매스마켓 소프트웨어 라이센스의 법적 유효성에 대한 위험은 오픈소스 소프트웨어 라이센스에 영향을 줄 것 같지는 않다. 왜냐하면 오픈소스 라이센싱은 이용자 측에 권리를 제한하는 것이 아니라 부여하는 수단이기 때문이다.

4.4 워런티(하자담보 · 보증책임)

비공개소스 소프트웨어와 마찬가지로 모든 오픈소스 라이센스는 모든 워런티 책임(warranty)을 부인하고 있다. 이러한 워런틱 책임의 전면적 부인이 법적으로 유효한지가 문제되고 있다. 라이센스 계약에 포함된 하자담보책임에 관하여 국가들의 입법에는 상당한 차이가 있다. 오픈소스 라이센스는 보통 워런티를 부인하는 조항을 포함하고 있고 많은 것은 책임제한을 시도하고 있다. 이것은 비교적 간단해 보이는 문제이다. 소스코드가 자유로이 이용되는 경우에는, 배포자는 자신의 책임을 제한하고 워런티를 부인하는 것은 이해될 수 있다. 그러나 소비자와의 관계 및 라이센서가 요금을 부과하는 상황에서는 그대

13) Trompenaars, W.M.B., Legal Support for Online Contracts, in Hugenholtz, B., Copyright and Electronic Commerce. Legal Aspects of Electronic Copyright Management, Kluwer Law International 2000, pp. 267.

로 적용될 수 없다. 이러한 형태의 규정은 재산권 있는 소프트웨어 라이센스의 경우에도 대부분 존재하지만, 여전히 고려해야 할 측면이 있다.

4.5 소프트웨어특허와 오픈소스 라이센싱

컴퓨터 관련발명은 발명이 그 실시를 위하여 소프트웨어 또는 하드웨어에 의하여 실현된 논리단계들을 필요로 하는 발명을 말하며, 소프트웨어란 컴퓨터의 동작에 관한 프로그램 또는 절차(procedures)를 말한다. 그러나 소프트웨어는 하드웨어와는 다른 기술적 성격을 가지고 있으므로 특허법에 의한 보호에 대한 검토를 필요로 한다. 미국을 비롯한 대부분의 국가에서는 소프트웨어와 공업제품 등 기계(하드웨어)가 결합되어 어떤 아이디어를 실현하는 경우에는 특허성을 인정해서 특허를 허여하고 있다.

한편, 소프트웨어 특허에 대해 처음부터 조직적인 반대 움직임을 보여온 것은 LPF(League for Programming Freedom)이다. 리차드 스톨만도 이 단체에 참여하고 있는데, LPF의 반대 논리는 독점을 보장하여 기술의 혁신을 촉진한다는 특허제도의 이념이 소프트웨어 산업에는 적절하지 않다는 것으로 요약할 수 있으며, LPF는 특허제도가 다른 산업에 긍정적인 영향을 미쳤다는 점을 부정하지는 않는다. 소프트웨어 특허에 대한 우려의 목소리가 커진 것은 1998년 State Street Bank 사건[14] 이후 인터넷 비즈니스 모델에 대한 특허가 쏟아지면서부터이다. LPF의 주장에 의하면,[15] 소프트웨어의 복합성, 소프트웨어의 추상성, 소프트웨어 기술은 혁신 속도가 빠른 점, 소프트웨어는 소모되지 않는 점, 소프트웨어는 경제 구조가 다른 점 및 소프트웨어의 성공은 시장에서 판가름난다는 점에서 특허제도를 소프트웨어에 적용하는 것 또한, 소프트웨어 특허의 문제점으로는, 소프트웨어 개발의 문제(예컨대, 문자 데이터 압축 분야에는 이제 너무 많은 특허가 존재하여 특허를 하나라도 침해하지 않는 데이터 압축 알고리즘을 만드는 것이 거의 불가능함), 법적 분쟁 문제(특허소송만큼 비용이 많이 드는 소송은 거의 없다), 특허청의 문제(전문가 부족, 종래 기술 검색 불가, 분류의 어려움, 2년 심사기간, 장기간의 존속기간)가 지적되고 있다. 또한 이에 대한 대안으로 소프트웨어 특허를 완전히 폐지할 것, 소프트웨어에 적합하도록 수정된 새로운 특허 제도를 만들 것, 저작권과 특허권 중 하나로만 보호할 것 또는 소프트웨어의 경우 특허요건을 강화하거나 존속기간을 단축하고 소스코드를 공개케 하는 등의 개정이 요구되고 있다.

현재 특허의 보호 대상을 확대하는 방향으로 나아가고 있다. 그러나 소프트웨어 특허의 경쟁정책상 기술개발의 인센티브로서 역할하기 곤란하며, 독점이 형성되고 경쟁제한행위가 발생할 가능성이 높고, 소프트웨어 기초기술의 경우에는 현저한 네트워크효과가 발생할 수 있다. 따라서 소프트웨어 특허와 경쟁정책과의 관계를 생각하는 경우, 소프트웨어 특허에 고유의 특질에만 주목할 것이 아니라, 「소프트웨어 특허에 어떠한 특별한 경쟁정책적 조치가 필요한가」 또는 「앞으로의 IT시대에 있어서 특허제도는 어떻게 존재해야 하는가」라는 근본적인 문제에 대한 해결책을 정책론적인 관점에서 검토해 보아야 할 것이다.

요컨대, 오픈소스 운동의 소프트웨어 대한 관점을 요약하면 다음과 같다. 자신들의 새로운 개발에 관련하여 재정적 문제로 인한 특허출원 및 관리 비용의 부담문제, 광범위한 분야에서 알고리즘의 특허화는 신프로그램의 개발에 대한 자유를 제한, 오픈소스 운동 단체에 의한 특허출원의 가능성이 문제된다. 특히 마지막 문제는 거의 논의 된 적이 없었다. 그러나 최근에 GNU/Linux 운영시스템의 업데이트 버젼 중의 하나인 RTLinux가 특허출원되었다.[16] 앞으로 오픈소스 분야에서 소프트웨어 특허에 대한 입장 변화 여부가 주목되고 있다.

14) State Street Bank & Trust Co. V. Signature Financial Group Inc., 149 F.3d 1368(Fed. Cir. 1998)=47 U.S.P.Q 2d 1596(Fed. Cir. 1998).

15) http://osnome.che.wisc.edu/~epperly/pto-sub/part1.html

16) U.S Patent 5,995, 745.

김 병 일

1990 연세대학교 경영학과(경영학사)
1992 연세대학교 대학원(법학석사)
1998 독일 뮨헨대학교(법학 박사)
1999~현재 인하대학교 법과대학 지적재산권학 전공 조교수
관심분야 : 지적재산권법, 정보법
E-mail : paulkim@inha.ac.kr