〮오픈소스의 역사

#1980년대

하드웨어 중심 체제로부터 점점 소프트웨어의 비중이 높아지면서 소프트웨어 기업의 소프트웨어에 대한 이용,배포,복제,수정 등에 일정한 제한을 가하려고 하는 추세가 있자, 이러한 ‘독점’체제에 반발해 ‘공유’를 주장하는 운동이 리차드 스톨만에 의해 일어났다.

리차드 스톨만은 소프트웨어 상업화에 반대하고 소프트웨어 개발 초기의 상호협력적인 문화로 돌아갈 것을 주장하며 1984년 GNU(GNU is Not UNIX)프로젝트를 주도했고, 이듬해인 1985년 FSF(프리 소프트웨어 재단, [http://www.fsf.org](http://www.fsf.org/))를 조직하면서, ‘소프트웨어는 공유돼야 하며 프로그래머는 소프트웨어로 돈을 벌어서는 안 된다’는 내용의 GNU 선언문을 제정하기도 했고, 이를 지원하기 위해 저작권(copyright)에 대응하는 카피레프트 운동도 주창했다.

#1990년대

1990년대 들어서면서 인터넷의 보급과 더불어 GNU GPL(General Public License)로 배포된 리눅스의 보급으로 자유소프트웨어 운동이 확산됐고, 1998년에는 MS의 웹브라우저인 익스플로러에 밀려 어려움을 겪고 있던 넷츠케이프(Netscape)사가 웹브라우저 모질라의 소스코드를 공개하는 결정을 하게 됐다.

그 즈음 자유소프트웨어라는 용어에서 오픈소스 소프트웨어로 용어가 변경되는데, 자유(free)란 용어 때문에 일반인들이 무료라고 인식하고 있다는 점, GPL 조항의 엄격성 때문에 소프트웨어 개발이 용이하지 않다는 점을 탈피하기 위해서였다.

#1998년대

이후 1998년 오픈소스 소프트웨어를 인증하는 OSI(Open Source Initiative, [www.opensource.net](http://www.opensource.net/))가 에릭 레이몬드(Eric Raymond) 등에 의해 결성되면서 오픈소스 소프트웨어 운동은 궤도에 오르게 된다.

OSI 단체가 정한 오픈소스 소프트웨어의 기준을 OSD(Open Source Definition)이라 하는데, 이 기준을 만족해야만 오픈소스 소프트웨어로 인정받게 된다.

오픈소스는 서버와 컴퓨터, 디바이스, 그리고 모든 소프트웨어 근간이다. 하지만 불과20년전, 오픈소스는 기술적인 것으로만 남아있을 것처럼 보였던 지극히 주변적이고 급진적인 개념이었다.

자유소프트웨어 운동은 35년 전 리처드 스톨만이 시작했다. OSI의 공동설립자인 브루스 페렌스는 “기존에 있던 자유 소프트웨어 개념을 기업에 홍보하고 또 라이선스를 인증하기 위해 오픈소스가 이 운동에 적합한 이름이다”라고 설명했다.

1998년, 넷스케이프는 넷스케이프 내비게이터의 개방형 개발을 스스로 주도 하지 않고 흘러가는 대로 내버려뒀다. 이것이 모질라 프로젝트로 이어졌다. 이념적인 접근보다는 실용적인 목적이 있다면 많은 기업이 자유 소프트웨어를 도입한다는 것이 분명해졌다.

‘오픈소스’라는 용어가 처음 소프트웨어에 적용된 것은 1998년 2월3일 미국 캘리포니아 마운틴 뷰의 VA리눅스 시스템즈 사무실에서 열린 전략회의에서였다. 오픈소스라는 용어 자체는 포어인사이트 인스티튜드의 시장 크리스틴 페터슨이 만들었다. 포어인사이트는 나노기술에 중점을 둔 씽크탱크 조직이었다.

OSI는 1998년 2월말 새로 등장한 오픈소스의 집사 역할을 위해 설립됐다. 에릭 레이몬드가 초대사장을 맡았고, 브루스 페렌스는 부사장을, 그리고 초대 이사회에는 브라이언 베렌도르프, 이안 머독, 러스 넬슨, 칩살젠버그가 참여했다. OSI로고는 콜린 비브록이 만들었는데, O는 개방성을, 열쇠구멍은 소스코드의 잠금 해제를 암시한다.

1999년 데비안 프리 소프트웨어 지침의 작성자인 브루스 페렌스는 이 지침을 소프트웨어 라이선스가 어떻게 오픈소스로 인식될 수 있는지에 대한 객관적인 정의로 개작했다. 바로 OSD이다.

1999년 데비안 프리 소프트웨어 지침(Debian Free Software Guideline)의 작성자인 브루스 페렌스는 이 지침을 소프트웨어 라이선스가 어떻게 오픈소스로 인식될 수 있는지에 대한 객관적인 정의로 개작했다. 바로 OSD(Open Source Definition)이다.

2000년대 중반, 오픈소스에 대한 인기가 높아지면서 라이선스의 무분별한 확산이 문제가 됐다. 2004~2006년 기간에 OSI는 이 문제를 처리하기 위해 공공 의견 수렴 과정을 진행했다

 2004년 리처드 스톨만이 ‘자바의 함정’이란 글을 통해 그 사악함을 지적했지만, 자바 플랫폼은 2006년 오픈소스화됐다.

2000년대 말 너무나 많은 오픈소스 비영리단체가 생겨나자 OSI는 산하 단체 모델로 전환하고 개인 회원제도도 도입해 새로운 이니셔티브를 가능하게 했다

새로운 경영진과 클라우드 솔루션 구축에서 오픈소스를 피할 수 없게 된 마이크로소프트는 오픈소스에 대한 공개적인 적대감 표현을 중단했다. 물론 마이크로소프트는 임베디드 리눅스에 대한 특허 소송은 계속하고 있다.

지금은 빅스택(Big Stack) 오픈소스의 시대다. 모든 클라우드 플랫폼의 모든 구성요소를 추적해 보면, 오픈소스의 기부나 컨소시엄에서 나온 것이다.

 오늘날 새로운 이니셔티브는 바로 오픈소스로 간다. 오픈소스는 20년만에 위협에서 기회로, 암적인 존재에서 기념비적인 존재로, 파생 기술에서 혁신으로 바뀌었다

[자료&이미지] <http://www.itworld.co.kr/print/108114>

〮유닉스

1969-1970년 사이에 AT&T 벨 실험실의 Kenneth Thompson, Dennis Ritchie 와 동료들은 약간 사용되고 있던 PDP-7 상에서 작은 운영 체제를 개발하기 시작했는데 이는 곧 MULTICS 라 불리는 초기의 운영체제 프로젝트를 빗대어 유닉스 (UNIX) 로 명명되었다. 1972-1973년 사이에 시스템은 실현 불가능했던 드문 방법인 C 프로그래밍 언어로 재작성 되었는데 이 결정 때문에 유닉스는 그 원래의 하드웨어로부터 바꿀 수 있고 그 원래의 하드웨어를 극복할 수 있는 최초의 널리 사용되는 운영체제가 되었다. 다른 혁신적인 것들도 또한 유닉스에 추가되었는데 이는 특히 벨 실험실과 대학 공동체의 상호 협동에 기인했다. 1979년 유닉스 7번째 버전 V7 이 발표되었으며 이는 모든 현존하는 유닉스 시스템의 대부격이다.

이 시점 이후 유닉스의 역사는 다소 복잡하게 되었는데 Berkeley 가 주도한 대학 공동체가 Berkeley Software Fistribution (BSD) 라 불리는 변종을 개발한 반면 AT&T 는 System III 와 나중에는 System IV 라는 이름으로 계속해서 유닉스를 개발했다. 1980년대 후반에서 1990년대 초반에 이러한 두 가지 중요한 변형들간의 전쟁은 최고조에 달했지만 여러 해가 지난 후에는 각 변형이 다른 변형의 중요 특징들 중 많은 부분을 채택하였다. 상업적으로 System V 는 인터페이스의 대부분이 정식 표준에 채택됨으로써 표준 전쟁에서 이겼으며 대부분의 하드웨어 벤더들은 AT&T 의 System V 로 전환하였다. 그러나 System V 도 결국 많은 BSD 혁신들을 받아들이게 되었으며 그래서 결과적으로 시스템은 두 시스템의 합병 이상이었다. BSD 시스템도 사라지지 않았으며 대신 연구, PC 하드웨어와 단일 목적 서버 (예, 많은 웹 서버들은 BSD 에서 파생된 시스템을 사용하고 있다) 용으로 널리 사용되게 되었다.

〮리눅스

1991년 Linus Torvalds 는 ``리눅스"라고 이름지은 운영체제 커널을 개발하기 시작했다 [Torvalds 1999]. 이 커널은 자유로이 수정될 수 있고 매우 유용한 운영체제를 만들기 위해 FSF 산물과 다른 컴포넌트들 (특히 BSD 컴포넌트들의 일부와 MIT 의 X 윈도우 소프트웨어) 과 병합될 수 있었는데 이 책에서 커널 자체는 ``리눅스 커널"로 전체 결합은 ``리눅스"로 지칭할 것이다. 많은 사람들은 대신 ``GNU/Linux" 라는 용어를 사용함을 주목해라.

리눅스 공동체에서 여러 조직들은 사용할 수 있는 컴포넌트들을 다르게 조합하였는데 각 조합은 ``배포판"으로 부르며 배포판들을 개발한 조직들은 ``배포업자"라고 부른다. 일반적으로 널리 알려진 배포판으로는 레드햇 (Red Hat), 맨드레이크 (Mandrake), 수제 (SuSE), 칼데라 (Caldera), 코렐 (Corel) 및 데비안 (Debian) 이 있다. 다양한 배포판들간에는 차이점들이 있지만 모든 배포판들은 동일한 기반인 리눅스 커널 및 GNU glibc 라이브러리들에 기초하고 있다. 두가지 모두는 ``copyleft" 스타일의 라이센스로 다뤄지기 때문에 이러한 기반에 대한 변경은 일반적으로 모든 사람이 사용할 수 있어야 한다. 이는 BSD 와 AT&T 에서 파생된 유닉스 시스템간에는 존재하지 않는 리눅스만의 기반에서 리눅스 배포판들을 통합시키는 힘이 원천이다. 이 책은 특정 리눅스 배포판에 고유한 것은 아니며 리눅스를 논의할 때 이 책은 본질적으로 모든 현재 주요 리눅스 배포판들에 대해 유효한 가정인 리눅스 커널 버전 2.2 이상과 C 라이브러리 glibc 2.1 이상을 가정한다.

〮오픈소스/자유 소프트웨어

자유로이 공유되는 소프트웨어에 대한 관심이 증가함에 따라 이에 대한 정의 및 설명이 더욱 더 필요하게 되었다. 널리 사용되는 용어는 ``오픈 소스 소프트웨어"로 [OSI 1999] 내에 더욱 자세하게 정의되어 있다. Eric Raymond [1997, 1998] 은 다양한 개발 과정들을 고찰한 몇몇의 독창성있는 기사들을 작성했다. 다른 널리 사용되는 용어는 ``자유 소프트웨어 (free software)" 로 ``free" 는 ``freedom" 을 나타내는데 보통 설명은 ``free beer 가 아닌 free speech" 이다. 그러나 두 표현 모두 완벽한 것은 아니다. ``자유 소프트웨어"라는 용어는 대개 실행파일은 무료로 제공되지만 소스 코드를 볼 수 없고, 수정할 수 없고 또는 재배포할 수 없는 프로그램들과 혼동된다. 반대로 ``오픈 소스"라는 용어는 때때로 소스 코드를 볼 수는 있지만 사용, 수정 또는 재배포에 제한이 있는 소프트웨어를 의미하는데 사용 (악용) 되고 있다. 이 책은 보통의 즉, 소스 코드의 사용, 보기, 수정 및 재배포를 자유로이 할 수 있는 소프트웨어에 대해 ``오픈 소스"라는 용어를 사용한다; 더욱 세부적인 정의는 [Open Source Definition](http://www.opensource.org/osd.html) 에 포함되어 있다. 어떤 경우 목적의 차이가 제안되고 있는데 ``자유 소프트웨어"라는 용어를 선호하는 사람들은 자유의 필요성을 강력히 강조하길 바라는 반면 다른 사람들은 다른 목적 (예, 높은 신뢰성) 를 가지거나 단순히 덜 불쾌하게 보이고 싶을 수도 있다.

〮리눅스와 유닉스 비교

이 책은 의도적으로 유닉스와 같은 시스템을 기술하기 위해 ``유닉스 계열 (Unix-like)" 이라는 용어를 사용한다. 특히 유닉스 계열은 모든 주요 유닉스 변형들과 리눅스 배포판들을 포함한다. 많은 사람들이 이러한 시스템들을 기술하기 위해 단순히 ``유닉스"라는 용어를 사용함을 주목해라. 원래 ``유닉스" 라는 용어는 AT&T 에 의해 개발된 특정 제품을 의미했다. 오늘날 오픈 그룹이 유닉스 상표권을 소유하고 있으며 유닉스를 ``worldwide Single UNIX Specification" 으로 정의하고 있다.

리눅스가 유닉스 소스 코드에서 파생되지 않았지만 그 인터페이스는 의도적으로 유닉스와 같다. 따라서 유닉스에서 배운 교훈이 보안에 대한 정보를 포함해 일반적으로 둘 모두에 적용된다. 이 책에서 정보의 대부분은 모든 유닉스 계열 시스템에 적용되는데 리눅스에 특정적인 정보가 리눅스를 사용하는 사람들이 리눅스의 능력을 사용할 수 있게 하려고 의도적으로 추가되었다.

유닉스 계열 시스템들은 많은 보안 메카니즘들을 공유한다. 물론 이 메카니즘들은 미묘한 차이가 있고 모든 시스템들에서 모든 메카니즘을 사용할 수는 없다. 모든 시스템들은 각 프로세스에 대한 사용자 및 그룹 ids (uids 및 gids) 와 user, group 와 other 에 대한 읽기, 쓰기와 실행하기 허가를 갖는 파일시스템을 포함하고 있다.

〮오픈소스의 역사와 상공

소프트웨어가 처음 태동하던 시기에 프로그래밍은 해커와 연구자들 중심으로 발전했다. 그러나 소프트웨어 산업이 발전하면서 점점 더 영리적인 성격을 띠게 되었고, IT 기업들이 소프트웨어를 개발하는 주축이 되었다. 소프트웨어로 돈을 벌려고 할 때 중요한 문제점이 있었는데, 바로 복제가 아주 쉽다는 것이었다.

초기 영리 소프트웨어 기업들은 절대 소스 코드를 공개하지 않고 철저하게 보호했다. 당연했다. 소프트웨어는 누구나 설계도인 소스코드를 가져가면 자기 마음대로 쓸 수 있기 때문이다. 기업들은 특허와 지적재산권을 사용해서 법적으로 소스 코드를 보호했다. 소스 코드에는 권한을 가진 소수의 사람만 접근할 수 있었다. 그러다 보니 IT기업의 개발 조직은 자연스럽게 더 많은 접근 권한이 있는 관리자로부터 가장 적은 프로그래머로 내려가는 하향식(Top-down) 형태가 되었다.

이 과정에서 소프트웨어의 저작권에 관한 치열한 논쟁이 벌어졌다. 해커와 연구자들로 이루어진 그룹은 소프트웨어는 무료로 배포되어야 한다고 주장했고, 소프트웨어 기업들은 소유권이 보호되어야 한다고 주장했다. 대표적인 기업 측 인물이었던 빌 게이츠는 1976년 ‘취미생활자들에게 쓰는 공개편지’라는 글에서 이렇게 말했다

“소프트웨어를 만드느라 애쓴 사람들에게 보상을 주지 않으면 도둑질이다.”

하지만 이에 반대하던 해커들 중 한명인 리처드 스톨먼은 소프트웨어는 자유로워야 한다고 주장하며 자유 소프트웨어 재단(FSF)을 설립했다. 그는 소프트웨어를 자유롭게 수정 배포할 수 있게 하는 GPL이라는 라이선스를 만들었다.

〮리눅스의 등장

이 상황에서 오픈 소스 역사의 획을 긋는 사건이 일어난다. 라이너스 토발즈라는 프로그래머가 나타나서 자신이 개발한 소프트웨어 커널(운영체제에서 핵심적인 역할을 하는 제어 모듈)을 GPL 라이센스로 공개한 것이다. 이것이 그 유명한 리눅스다.

토발즈가 리눅스를 처음 공개했을 때 누구도 리눅스가 역대 최대 규모의 소프트웨어 프로젝트가 될 거라고 생각하지 않았다. 그 때만해도 오픈 소스는 소수의 해커들만이 참여하는 마이너한 문화였기 때문이다. 리눅스는 달랐다. 이전에는 오픈 소스라도 핵심 개발자 집단이 개발 과정을 독점하고, 완성도가 있다고 판단된 후에 소스를 공개하는 방식을 사용했다. 그런데 토발즈는 누구나 소스를 읽은 후 수정한 코드를 보낼 수 있도록 했다. 수정한 코드가 받아들여지면 그 사람은 기여자가 되어 다음 버전에 이름이 박혔다. 이 시스템이 묘한 과시욕을 충족시켰고, 개발자들은 돈을 받지 않고도 커널 버그 수정과 기능 추가에 매달렸다. 점점 더 많은 사람들이 개발에 참여하게 되었고, 그들은 리눅스를 점점 더 정교하게 발전시켜나갔다. 몇 십년이 지난 지금 리눅스는 서버와 모바일에서 가장 많이 쓰이는 운영체제가 되었다. 전세계 수 억대의 스마트폰에 탑재된 안드로이드 OS가 리눅스 커널을 사용하고 있다. 세계에서 가장 빠른 슈퍼컴퓨터 500대의 90%가 리눅스 기반이다.

〮오픈소스, 대세가 되다

리눅스의 성공은 오픈 소스의 협업 모델이 상업적 기업보다 뛰어날 수 있다는 걸 증명했다. 전 세계의 ‘자원봉자사’들이 코드만 100만줄이 넘는, 이 엄청나게 복잡한 컴퓨터 프로그램을 만들고, 기업이 개발한 프로그램을 능가했다. 이는 그 전까지 소프트웨어 역사에서 전례가 없는 일이었다.

이 후로도 계속해서 중요한 소프트웨어들이 오픈 소스로 개발되고 있다. 특히 최근 소프트웨어 업계에서 오픈 소스는 완전히 대세가 되었다. 앞서 말했던 안드로이드 뿐만 아니라 클라우드, 데이터베이스, 웹서버, 웹브라우저, 컨텐츠 관리 시스템, 빅데이터 관리등 거의 모든 분야에서 높은 점유율을 가지고 있다. 현재 사용되는 소프트웨어 중 오픈 소스 기술을 사용하지 않는 것은 거의 없다.

심지어 오픈 소스를 ‘암적인 존재’로 비난하던 영리 기업들도 이제는 너도나도 오픈 소스를 도입하고 있다. 상용 소프트웨어의 상징이었던 마이크로소프트는 최근 리눅스를 지원하고, 자사 소프트웨어를 오픈소스와 연동하고, 일부 소프트웨어를 오픈 소스화했다. 사타냐 나텔라 CEO는 ‘마이크로소프트는 리눅스를 사랑합니다’ 라는 말까지 했다. 이와 같은 변화는 영리 기업들이 수집조의 연구 개발비를 들여서도 만들지 못한 혁신을 오픈소스가 만들어냈기 때문이다. 바햐흐로 오픈소스 는 ‘대세’가 되었다.

[<https://brunch.co.kr/@bumgeunsong/15>]