

**오픈소스란 무엇일까?**

오픈소스(open source) 누구나 한번쯤 들어 봤을 말이다. 그런데 오픈소스가 무엇인지 정의하라고 하면 대부분의 사람들은 “말 그대로 열려 있는 소스(source) 아니야? 그냥 자유롭게 사용 할 수 있는 소스잖아” 라고 할 것이다. 아주 틀린 말은 아니지만 그렇다고 완전히 다 맞는 말은 아니다. 오픈소스는 프리 소프트웨어(free software)를 포괄하는 개념이다. 프리 소프트웨어는 단순히 ‘자유롭게 사용 할 수 있는’ 소프트웨어를 의미한다. 그러나 오픈소스는 자유롭게 사용 할 수 있을 뿐만 아니라 (프로그램의 설계도 같은) 소스코드까지 공개되어 있어 수정 및 배포까지 자유로운 소프트웨어를 말한다. 또한 오픈소스라고 하면 어떤 사람들은 “리눅스가 오픈소스 아니야?” 라고 한다. 그러나 이 또한 맞는 말이지만 틀린 말이다. 리눅스는 오픈소스를 이용해 만들어진 운영체재(OS)일 뿐 오픈소스 그 자체는 아니기 때문이다.

자 이제 오픈소스에 대해 제대로 알아 보도록 하자. 일단 오픈소스에 대해 알려면 위에서 얘기한 프리 소프트웨어에 대해서도 알아야 하는데 프리 소프트웨어는 자유 소프트웨어 라고도 한다. 자유 소프트웨어에 대해 말하자면 1984년으로 거슬러 올라간다. 1984년 MIT 인공 지능 연구소의 연구원이었던 리처드 스톨만 (Richard Stallman)은 GNU 프로젝트를 시작하였다. GNU 프로젝트의 목적은 간단히 말해서 누구도 소프트웨어에 비용에 대해 지불하지 않도록 하는 것이었다. 스톨만은 소스 코드가 공개되어야 한다고 생각했기 때문에 GNU 프로젝트를 시작하였다. 만약 그렇게 되지 않으면 컴퓨터를 다룰 수 있는 소수만이 컴퓨팅을 지배하는 폐해를 낳을 것이기 때문이다.

독점적인 상업용 소프트웨어 업체는 과학적 지식이 철저히 보호되고 비밀이 유지되어야 한다고 주장하지만 리처드 스톨만은 널리 공유하고 배포해야 하는 것으로 본다. GNU 프로젝트와 자유 소프트웨어 재단(Free Software Foundation) 의 기본 신조는 소스 코드는 컴퓨터 과학을 발전시키는 데 기본이며, 변화가 계속되기 위해 정말로 필요한 것은 바로 자유롭게 사용할 수 있는 소스 코드라는 것이다.

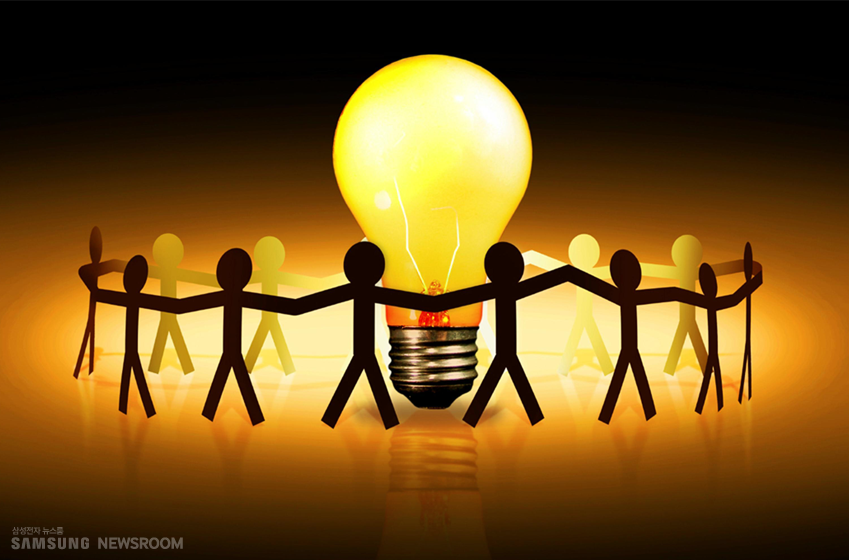
지금까지 자유 소프트웨어에 대해 알아 보았으니 본격적으로 오픈 소스 소프트웨어란 무엇인지에 대해 알아 보도록 하자.

1997년 봄, 자유 소프트웨어 공동체의 리더 그룹들이 캘리포니아에서 회합을 했다. 이 그룹에는 에릭 레이먼드(Eric Raymond),팀 오라일리(Tim O’Reilly), VA 리서치사 사장인 래리 오거스틴(Larry Augustin)이 포함되어 있었다. 그들의 관심사는 기존 자유 소프트웨어라는 개념에 대해 거부감을 가진 사람들에게 자유 소프트웨어를 둘러싼 아이디어들을 진흥시킬 방법을 찾는 것이었다.

에릭 레이먼드의 끈질긴 주장으로, 회합에 모인 사람들은 시장 점유 뿐만 아니라 대중들의 인식까지도 점유할 수 있는 마케팅 캠페인이 부족했다는 점에 동의했다. 회의 도중 새로운 용어인 오픈 소스(Open Source)가 나왔다. 그리고 오픈소스라고 말할 수 있는 소프트웨어에 대한 가이드라인을 만들었다.

브루스 패런스(Bruce Parens)는 오픈 소스 정의에 대하여 많은 기초 작업을 해두었다. GNU 프로젝트의 커다란 목표 중 하나는 GNU 소프트웨어를 실행할 수 있는 플랫폼으로써 자유롭게 사용할 수 있는 운영체제를 만드는 것이다. 소프트웨어 부트 과정과 관련하여 리눅스가 그 플랫폼이 되었으며, 이 리눅스는 GNU 도구의 도움을 받아 만들어졌다.

지금까지 오픈소스에 대해 설명했는데 좀 어려웠던 것 같다. 좀 더 쉽게 다시 한 번 설명하자면 ‘백지장도 맞들면 낫다’라는 속담 다들 들어보았을 것이다. 소프트웨어 개발 작업은 대게 팀 단위로 진행되는 게 일반적인데, 실제로 소프트웨어를 협업 형태로 개발하면 여러 장점을 누릴 수 있다. 토론을 통해 더 올바른 방향을 선택할 수 있는 것은 물론, 어려운 문제도 함께 해결할 수 있다. 코드 안에 숨겨진 잠재적 오류들을 찾아내는 일도 훨씬 더 빠르고 수월해진다.

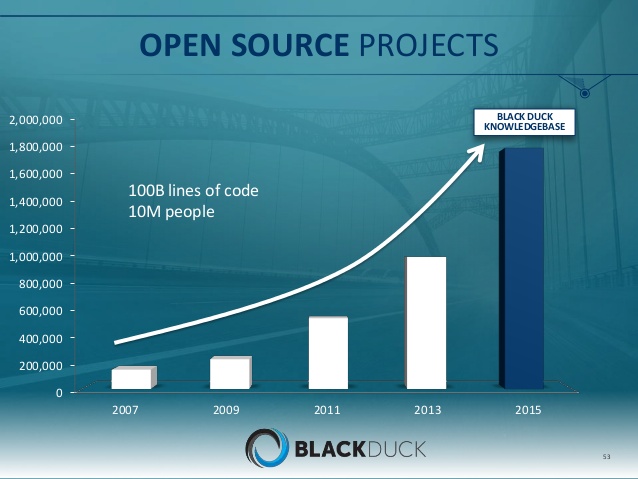


이처럼 소프트웨어 개발에서 협업은 구성원의 능력을 하나로 모아 집단지성과 창의성을 발현시키고 그 과정에서 보다 나은 결과물을 탄생시킬 수 있는 데 기여한다. 그리고 오픈소스는 이러한 장점들을 극대화한 형태라고 할 수 있다. 하나의 소프트웨어를 개발하기 위해 적게는 몇 명, 많게는 수백 수천명의 개발자가 협업에 참여하기 때문이다.

목표가 같은 개발자가 많아지면 집단 지성과 창의성도 위력도 그에 비례해 커진다. 그 결과는 결국 ‘오픈 소스의 빠른 업그레이드’로 나타난다. 기여자의 수가 늘수록 오픈소스의 코드는 간결해지고 오류 발생 빈도는 줄어든다. 더 안정적이고 유용한 소프트웨어가 완성되는 것이다. 실제로 유명 클라우드 컴퓨팅 오픈 소스 중 하나인 오픈스택엔 4000명 이상의 개발자가 참여하고 있다. 그 덕에 안정적 코드와 빠른 성장을 바탕으로 클라우드 컴퓨팅 분야의 중심에 우뚝 설 수 있었다.

오픈소스 소프트웨어의 장점은 또 있다. 청소년들의 교육에 영향을 끼칠 수 있다는 것이다. 청소년들이 무료로 쉽게 정보에 접근 할 수 있기 때문이다. 또 개발도상국에 많은 영향을 끼칠 수 있다. 오픈소스들이 점차 많아지면 세계적인 기술의 혜택을 받지 못하는 개발도상국의 브레인들이 정보격차를 줄일 수 있기 때문이다. 오픈소스의 장점들이 굉장히 많은데 더 자세한 것은 뒤에 서술 해 놓았다. 

지금까지는 오픈소스의 장점들에 대해만 설명한 것 같은데, 오픈 소스에 제약이 전혀 없는 것은 아니다. 가장 큰 문제는 라이선스 이다. 각 오픈 소스가 지향하는 라이선스 정책에 따라 일부 오픈소스는 사용 시 그에 상응하는 책임이 따르기도 한다. 예를 들어 GPL 라이선스를 지닌 오픈소스를 활용해 소프트웨어를 개발하면 소스코드 공개의 의무가 따르므로 라이선스를 충분히 고려하는게 중요하다. 반면 BSD와 같은 라이선스는 저작권과 라이선스만 명시하면 되므로 책임 문제에 관한 한 비교적 관대한 편이다. 자세한 내용은 뒤에서 다루도록 하겠다.



오픈소스의 향후에 대해 짧게 얘기하자면 옆의 표와 같이 오픈소스의 장점에 대해 알고서 오픈소스를 사용하는 개발자들이 점차 더 많아진다는 것이다. 이렇게 많은 사람들이 사용하다보면 점차 더 발전하고 보완되어 더 나은 오픈소스 소프트웨어들이 될 것이다.

여기까지 오픈소스에 대해 설명하고 소개했다. 이제 뒤의 내용에서 오픈소스의 역사에 대해 훨씬 더 자세하게 알아보고, 장단점 앞으로의 향후 등을 소개할 것이다. 뒤의 내용에도 흥미있는 내용들이 많으니 집중하시고 다음 챕터를 읽어보길 바란다.