EVERYTHING OF OPENSOURCE

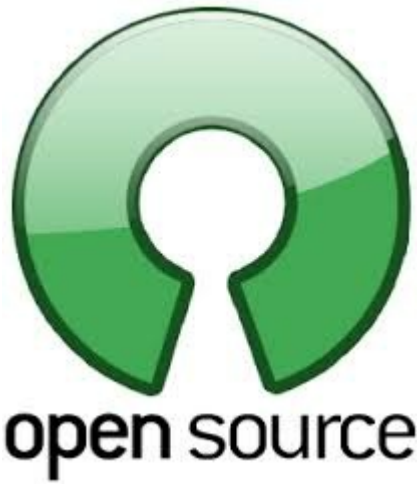
A-07-Error 모르겠다

Table of Contents

- 오픈소스의 소개
  - 오픈소스의 소개
    - 오픈소스가 무엇인가**?**
    - 오픈소스 소프트웨어란 무엇인가**?**
    - 오픈소스 소프트웨어와 상용 소프트 웨어의 차이는 무엇인가**?**
    - 오픈소스의 장점과 단점
    - **OSI(Open Source Initiative)** 오픈소스 조건
    - 자유소프트웨어는 무엇인가**?**
    - 자유 소프트웨어 공동체와의 사회 계약**//** 이게 무엇인지 잘 모르겠음
    - 오픈소스 저장소**(Repository)**
    - **GitHub** 상위 랭킹 프로젝트
    - 오픈소스 기반 서비스 기업 사례
    - 오픈소스 기여 기업
  - 오픈소스의 인물
  - 오픈소스의 재단

오픈소스의 소개

오픈소스가 무엇인가**?**



오픈소스는 소프트웨어 혹은 하드웨어를 누구나 접근할 수 있도록 하여 사람들이 이를 수정 및 공유할 수 있는 소프트웨어 혹은 하드웨어를 일컫는다.

오픈소스 소프트웨어란 무엇인가**?**

오픈소스 소프트웨어, OSS라고도 한다.소프트웨어의설계도에 해당하는 소스코드를 인터넷 등을 통하여 무상으로 공개하여 누구나 그 소프트웨어를 개량하고, 이것을 재배포할 수 있도록 하는 것 또는 그런 소프트웨어를 말한다.

오픈소스 소프트웨어 라이선스

공개SW(Open Source Software)라이선스란 공개 SW개발자와 이용자 간의 사용 방법 및 조건의 범위를 명시한 계약을 말한다. 따라서 공개SW를 이용하려면 공개 SW개발자가 만들어놓은 조건의 범위에 따라 해당 소프트웨어를 사용해야 하며, 이를 위반할 경우에는 라이선스 위반 및 저작권 침해로 이에 대한 법적 책임을 져야한다.

다음은 라이선스 종류들이다.

	무료 이용가능	배포 허용가능	소스코드 취득가능	소스코드 수정가능	2차 저작물 재공개 의무	독점 SW와 결합가능
GPL	O	O	O	O	O	X
LGPL	O	O	O	O	O	O
MPL	O	O	O	O	O	O
BSD license	O	O	O	O	X	O
Apache license	O	O	O	O	X	O

- **GPL** : General Public License.저작권은 개발자에게 귀속되지만 소프트웨어의 복사,수정 및 변경,배포의 자유를 제3자에게 허용.
- **LGPL** : GNU Lesser General Public License. GPL을 변형해 더 허가된 형태로서,소프트웨어 라이브러리를 염두에 둔 것.
- **MPL** : 모질라 공용 허가서(Mozilla Public License).모질라 애플리케이션 스위트, 모질라 파이어폭스, 모질라 선더버드 및 그 외의 모질라 소프트웨어들에 적용.
- **BSD license** : 유닉스(Unix)의 양대 뿌리 중 하나인 버클리의 캘리포니아 대학에서 배포하는 공개 소프트웨어의 라이선스. GPL보다 훨씬 개방적인4개항의 간단한 문구로 되어 있음.
- **Apache license** : 아파치 소프트웨어 재단에서 자체적으로 만든 소프트웨어에 대한 라이선스 규정. 누구나 해당 소프트웨어에서 파생된 프로그램을 제작할 수 있으며 저작권을 양도, 전송할 수 있는 라이선스 규정을 의미.

OSI(Open Source Initiative) 오픈소스 조건

오픈소스는 한마디로 말하면 협업이라 할수있다. OSI(Open Source Initiative)는 다음과 같이 정의를 내리고 있다. "오픈 소스는 독립적인 피어 리뷰와 재빠른 소스 코드 진화를 지원하여 소프트웨어의 신뢰성과 품질을 촉진하고 있다. OSI의 인증을 받기 위해서는,소프트웨어가 무료로 읽혀지고,재배포,변경,수정될 수 있음을 보장하는 라이센스를 통해 배포되어야 한다." OSI(Open Source Initiative)는 오픈소스가 될 수 있는 조건들을 발표했다. 다음은 기본적이지만 궁극적으로 지켜야하며, 오픈소스 소프트웨어가 지향하는 목표들이다.

1.자유로운 재배포

오픈 소스를 이용하여 제작한 소프트웨어의 자유로운 배포를 허용해야 합니다.그리고 사용된 오픈 소스에 대한 어떠한 비용도 받지 않아야 합니다.

-이와 같은 원칙이 지켜지지 않는다면 오픈 소스를 개발하는 사람들의 개발의지가 꺾일 것이며,이로 인해 장기적인 이익도 기대할 수 없기 떄문이다.

2.원시 코드

오픈 소스는 원시코드(Source Code)가 포함되어야 하며,최소의 실비만으로도 원시코드를 구할 수 있도록 하여야 합니다.또한 원시코드는 고의로 알아보기 어렵게 만들어지지 않아야 합니다.

-오픈 소스는 기본적으로 소스의 수정과 변형을 전제로 하기 때문에 최대한 이용하기 쉽게 작성되도록 노력해야 한다.

3.파생 저작물

개작과 파생 저작물을 허용해야 합니다.그리고 이렇게 파생된 저작물에는 원본 오픈 소스에 적용된 라이선스를 똑같이 적용할 수 있도록 해야 합니다.

-이 원칙은1번 원칙과 비슷하게 파생 저작물 역시 빠르고 혁신적으로 발전할 수 있는 바탕이 된다.

4.저작자의 원시 코드 원형 유지

바이너리(2를 기반으로 하는 숫자 체계,컴퓨터에서 데이터를 표현하기 위해 사용됨)를 생성할 시점에서 프로그램을 수정할 목적으로 패치 파일의 배포를 허용한 경우에 한해,변경된 원시 코드의 배포를 제한할 수 있습니다.그러나 변경된 원시코드가 아닌 그것을 통해 만들어진 결과물의 배포는 허용해야만 합니다.파생 저작물에 대해서는 원본 오픈 소스와 다른 이름이나,버전을 적용하도록 할 수 있습니다.

-이러한 조치는 소스코드의 원작자와 수정인의 정보를 나눔으로써 사용자가 피드백이 필요한 부분에 대한 의견을 누구에게 문의할지 명확하게 해준다.

5.개인이나 단체에 대한 차별 금지

라이선스는 특정 개인이나 단체를 차별하지 않아야 합니다.

-개인 또는 단체 모두 오픈 소스의 개발과 발전에 큰 보탬이 될 수 있으므로 그 누구도 차별하지 않고,오픈 소스를 이용할 수 있어야 합니다.일부 국가에서는 특정한 종류의 소프트웨어에 대하여 수출 금지를 하고 있지만 그것에 대한 법률을 준수해야 한다는 경고를 하는 수준에 그치고 있고,라이선스 자체에 금지 규정을 싣지는 않고 있다.

6.사용분야에 대한 차별 금지

라이선스는 소프트웨어가 특정 분야에서 사용되는 것을 금지해서는 안됩니다.예를 들어,영리기업이나 유전학 연구에 프로그램을 사용할 수 없다는 등과 같은 제한을 설정해서는 안됩니다.

-상업적 영역에서도 오픈 소스가 사용될 수 있도록 하여야 한다.

7.라이선스의 배포

프로그램에 첨부된 권리는 별도의 라이선스를 적용하지 않고,재배포를 하는 프로그램의 이용자에게도 동일하게 적용되어야 합니다.

-비공개 동의를 요구하는 것처럼 간접적인 방법으로 소프트웨어를 제한하는 것을 방지한다.

8.특정 제품에만 유효한 사용 허가의 금지

프로그램에 첨부된 권리는 특정한 소프트웨어의 배포에 일부분일 경우,한정적이어서는 안됩니다.만약 배포된 프로그램의 라이선스 내에서 추출·이용·배포되더라도,재배포 받은 이용자는 원본 프로그램의 라이선스와 같은 권리를 가집니다.

-다른 종류의 라이선스를 방지 한다.

9.다른 소프트웨어에 대한 제한 금지

라이선스는 다른 라이선스를 가진 소프트웨어에 대한 제한을 포함해서는 안됩니다.예를 들어 함께 배포되는 모든 소프트웨어가 오픈 소스 라이선스여야만 한다고 강제할 수 없습니다.

-오픈 소스 소프트웨어의 배포자는 자신들의 고유한 소프트웨어 선택권이 있기 때문이다.

10.라이선스는 기술 중립적이어야 합니다**.**

라이선스의 조항은 개인적인 기술이나,인터페이스 스타일에 국한될 수 없다.

오픈소스 소프트웨어와 상용 소프트 웨어의 차이는 무엇인가?

상용 소프트웨어 정의 : 판매를 목적으로 만들어진 소프트웨어 제품이다. 상용 소프트웨어는 대부분 유료이며 실행 코드만 제공하고 소스 코드는 배포하지 않는 것이 특징이다. 사용자는 소스 코드를 받지 못하기 때문에 개량을 못하며 가격을 지불하지 않고서는 복제와 재배포도 허용되지 않는다. 그러나 일반적으로 그렇다는 것일 뿐, 상용 소프트웨어라고 모두 유료는 아니고 그 예로 애드웨어(adware), 셰어웨어(shareware), 프리웨어(freeware)가 있다. 오픈 소스 소프트웨어(OSS: Open-Source Software)와는 달리 제작사는 판매가의 요율에 결정된 가격으로 유지보수 및 기술 지원 서비스를 제공하며 보증(warranty) 활동을 수행한다.

구분	상용 SW	오픈 소스
비용분석	• 초기 도입비용이 높은 • 유지보수 비용 및 시스템 개선비용 높음	• 초기 도입비용이 낮음 • 유지보수 비용 및 기능 추가 비용이 낮음
제약사항	• 라이센스 계약으로 인한 제약사항 • (사용료 지급)	• 사용상의 제약사항 없음 (무료)
배포형태	• 라이센스 계약으로 인한 바이너리 제공 • 소스코드는 기업비밀로 유지함	• 소스코드가 공개 • 라이센스 조건하에 자유롭게 배포,수정
성능분석	• 비교적 큰 시스템 환경에서의 높은 성능 나타냄	• 다양한 환경에서 최적화된 설정으로 높은 성능치를 나타냄 (Intel, PPC, s390등)
기술성	• 문제점 발생시 폐쇄적인 운영으로 취약점 보유	• 소스코드의 공개로 빠른 문제점 해결 • 유지보수 및 업그레이드 용이, 독점피해 방지
확장성	• 시스템 환경에 따라 호환성은 보장 • 높은 적용비용을 지불해야 함 • 제한된 시스템 운영환경에서 용이	• 소프트웨어간 적용비용이 낮음 • 기능추가 비용이 낮음
공급권	• 최초도입 개발업체 또는 벤더에게 공급 • 개발업체 문제시 도입 고객에게도 심각한 영향을 미침	• 동일 솔루션에 대한 다수의 업체로 부터 지원 및 공급이 가능, 사용자의 선택권이 넓음

오픈소스의 장점과 단점

대부분의 사용자들이 오픈 소스 소프트웨어에 끌리는 첫 번째 매력은 가격이다.오픈 소스 프로그램들은 종종 그 작성자가 누구인지 모른다.하지만 무료에 가까운 비용이 오픈 소스 소프트웨어의 근본적인 매력은 아니다. 다음은 오픈 소스 개발을 지지할 수 밖에 없는 오픈소스의 장점들이다.

1.비용절감

오픈소스 소프트웨어는 프리웨어 소프트웨어와 마찬가지로 무료 이용이 가능하다. 여기서 더 나아가 소스코드가 공개 되어 직접 소프트웨어의 개선 또는 수정이 가능해지며, 이로 인해 개발 비용이 적게 드는 편이다. 실제로 오픈소스는 무료 다운로드와 수정,재배포가 가능하여 초기 개발 비용이 새 소프트웨어를 개발하는 것의 절반 정도 되는 것으로 알려져 있다.

2.빠르고 유연한 개발

오픈소스 커뮤니티는 다양한 이용자들에게서 최신 기술 정보와 문제점의 해결책을 공유하여 운영되기 때문에 독점 프로그램에 비해 기술의 발전 속도가 빠른 편이다. 특히 개발자와 사용자가 일치하는 경우 클로즈드 소스 프로그램보다 뛰어난 고품질의 오픈소스 소프트웨어가 개발되기도 하다.

3.호환성/유연성

오픈소스는 주로 오픈포맷 또는 오픈프로토콜(개방형 표준)을 사용하기 때문에 서로 다른 소프트웨어간의 연동이 쉽다. 서로 다른 플랫폼끼리의 상호 연동 또한 가능하다. 또한 특정 기기, 운영체제, 어플리케이션에 종속되지 않고 자유로운 변경이 가능하고, 여러 기기들이 네트워크를 통해 하나로 연결되는 유비쿼터스 시대에 아주 적합한 장점이라고 할 수 있다.

4.신뢰성**/**안정성

전 세계의 수많은 개발자들과 전문가들이 오픈소스의 개발에 참여하기 때문에 폐쇠적으로 개발되는 독점 프로그램에 비해 안정적으로 작동한다. 단, 이는 많은 개발자들이 적극적으로 참여하는 프로그램의 경우에만 가능하므로 해당 오픈소스의 평판과 개발 과정을 주의 깊게 볼 필요가 있다.

이와 같은 오픈소스 소프트웨어에 끌리는 장점들도 있지만 반대로 단점 또한 존재한다. 다음은 오픈소스 소프트웨어의 단점들이다.

1.비숙련 사용자들에게 더 사용이 쉬울 경우

리눅스는 서버 시장에 있어서 많은 영향을 미쳤지만 데스크톱 시장에서는 그렇지 못했다. 지난 수년간 발전을 거듭했음에도 여러 배포판의 UI가 윈도우나 맥 OSX에 비하면 떨어지는 수준이기 때문이다. 기술적으로 리눅스가 이러한 상용 OS보다 떨어질 것이 없다. 그러나 리눅스가 가진 이들 약점은 대부분의 사용자들이 더 사용하기 어려워 현업에서 사용할 가능성이 적어진다는 것을 의미한다. 이는 생산성 저하로 연결된다. 즉 직원들이 친숙해 하는 자체 OS를 사용하는 것보다 더 많은 비용이 발생하는 것이다.

2.사실상의 표준일 때

리브레오피스(LibreOffice)나 아파치 오픈오피스(Apache OpenOffice) 등과 같은 오픈소스 소프트웨어들도 있지만, 써드파티 제품들과의 통합을 위한 UI와 기능, 성능, 플러그인, API의 측면에서 정확히 일치하지 않다. 90%정도의 유사성을 보이기는 하지만, 나머지 차이점이 문제를 발생시킬 가능성도 있다. 특히 공급업체와 소비자간 문서 교환에 있어서 이러한 문제가 발생할 가능성이 있다. 또한 전문 영역에서 사유 소프트웨어(Proprietary Software)의 사용은 당위성을 지니게 된다. 여러 벤더들이 대학에서 자신들의 소프트웨어로 학생들을 교육시키는 상황 등이고, 아파치 소프트웨어 재단(Apache Software Foundation)의 회원이며 NASA 연구소 수석 컴퓨터 과학자인 크리스 매트만은 “이러한 상황이 발생할 때, 학생들은 해당 소프트웨어에 대해 더 잘 알게 된다”라고 말했다.

3.사유 소프트웨어**(Proprietary Software)**가 더 많은 지원을 제공할 때

오픈소스 소프트웨어 영역에서도 훌륭한 지원이 지원될 수 있다. 하지만 자주 그러한 것은 아니며 이것이 문제가 될 수 있다고 카네기 멜론 대학(Carnegie Mellon University)의 토니 와서만 교수는 말했다.

그는 “고객들이 기업 외부의 누군가를 통해 제품 지원을 상시 요청하는 경우도 있고 신속한 대응을 전제하는 서비스 수준 계약에 비용을 지불하고자 하는 사람도 있다. 오픈소스 프로젝트의 포럼 페이지에 질문이 올라오면 이에 대해 신속히 댓글이 달리는 경우가 많다. 하지만 수신자 부담 전화를 통해 확실한 벤더가 지원을 제공하는 것에 비길 수는 없다” 라고 말했다.

4. 서비스로서의 소프트웨어**(Software as a Service)**를 원할 때

클라우드 소프트웨어는 기존의 소프트웨어와는 다르다. 일반적으로 소스코드에 대한 접근이 불가다. 호스트 소프트웨어가 온전히 오픈소스 소프트웨어를 기반으로 구축된 경우에도 마찬가지이다. 따라서 이 경우, 오픈소스의 장점을 제대로 전달해주지 못하고, 또 이러한 관점에서 ‘사용하는 것에 있어 지불하는 방식’을 차용한 SaaS의 장점은 소스코드에 대한 접근이 없다는 것이 가져다주는 단점을 넘어선다.

자유소프트웨어는 무엇인가**?**

"자유 소프트웨어"의 핵심은 구속되지 않는다는 관점에서의 자유에 있는 것이지 무료라는 금전적인 측면에 있는 것이 아니다. 우리가 의도하는 이러한 자유의 의미를 쉽게 이해하기 위해서는 "무료 맥주(free beer)"가 아닌 "언론의 자유(free speech)"와 같은 예를 생각해 볼 수 있다. "자유 소프트웨어"는 사용자가 소프트웨어를 실행시키거나 이를 복제 및 배포할 수 있는 자유와 함께 소스 코드에 대한 접근을 통해서 이를 학습하고 수정, 개선시킬 수 있는 원천적인 자유까지를 모두 포괄하는 것이다. 따라서, 간략히 말하면 다음과 같은 4가지 종류의 자유를 내포한다고 할 수 있다.

-자유 1 : 프로그램을 어떠한 목적을 위해서도 실행할 수 있는 자유

-자유 2 : 프로그램의 작동 원리를 연구하고 이를 자신의 필요에 맞게 변경시킬 수 있는 자유 (이러한 자유를 위해서는 소스 코드에 대한 접근이 선행되어야 한다.)

-자유 3 : 이웃을 돕기 위해서 프로그램을 복제하고 배포할 수 있는 자유

-자유 4 : 프로그램을 향상시키고 이를 공동체 전체의 이익을 위해서 다시 환원시킬 수 있는 자유 (이러한 자유를 위해 서는 소스 코드에 대한 접근이 선행되어야 한다.)

사용자에게 위와 같은 자유를 모두 보장하는 프로그램은 자유 소프트웨어라고 할 수 있다. 따라서, 누구든지 이러한 자유 소프트웨어를 유료 또는 무료로 수정하거나 그렇지 않은 상태로 어느 누구에게나 그리고 어느 곳에라도 자유롭게 복제하고 배포할 수 있다. 이러한 형태로 프로그램을 자유롭게 만든다는 것의 의미는 사용 허가를 받기 위해서 별도로 요청할 필요도 없고 또한 비용을 지불할 필요도 없다는 것을 의미한다. 또한, 여러분은 개인적인 목적을 위해서 프로그램을 자유롭게 개작하거나 이용할 수 있고 이러한 사실을 명시적으로 공지하지 않아도 무방하다. 만약, 자신이 수정한 부분을 공개하고자 할 때에는 특정한 사람이나 방식으로 이를 알려야 할 필요가 없습니다. 개작된 부분을 자유롭게 만들기 위해서 그리고 개선된 버전을 발표하고 이를 의미 있는 것으로 만들기 위해서는 당연히 프로그램의 소스 코드에 대한 접근이 선행되어야 한다. 따라서 소스 코드에 대한 접근은 자유 소프트웨어를 위한 필요 조건이라고 할 수 있다.

GNU 소프트웨어는 유료로 구입할 수도 있고 무료로 얻을 수도 있다. 그러나 어떠한 방법으로 소프트웨어를 구했던 간에 여러분은 해당 프로그램에 대한 복제와 개작의 자유를 항상 갖게 된다. 이러한 종류의 자유가 현실화되기 위해서는 여러분이 소프트웨어에 해악한 일을 하지 않는 한 그러한 자유가 보장되어야 하고, 만약 소프트웨어 개발자가 라이선스를 수정할 수 있는 권리를 갖게 된다면, 당신이 소프트웨어 사용 상의 문제를 갖고 있지 않다고 하더라도 그 소프트웨어는 자유로운 것이 아니다. 그러나 본질적인 자유와 상충되지 않는 한 자유 소프트웨어를 배포할 때 특정한 종류의 배포 방식을 사용할 수 있다.

예를 들면, 카피레프트(copyleft)는 프로그램을 배포할 때 사용할 수 있는 방법 중의 하나이다. 카피레프트를 사용할 경우에는 다른 사람들의 자유를 제한할 수 있는 어떠한 사항도 추가할 수 없도록 하고 있다. 그러나 이러한 규정은 자유 소프트웨어를 구성하고 있는 본질적인 자유에 위배되지 않는다. 오히려 그것을 보호하기 위한 것이다. 만약 개작된 버전을 패키지하는 실제적인 방법이 개작된 버전을 배포하는데 따른 자유를 막는 것이 아니라면 그러한 방법을 위해서 사용되는 규정이나 제한 조건들은 받아들여 질 수 있다. GNU 소프트웨어를 사용하기 위해서 지불된 비용의 유무에 상관없이 일단, 소프트웨어를 입수한 뒤에는 이를 복제하고 수정할 수 있는 자유가 제한 없이 주어진다. GNU 프로젝트는 이러한 자유를 모든 사용자들에게 실질적으로 보장하기 위해서 저작권의 양도에 관한 실정법에 의해서 그 법률적 효력이 인정될 수 있는 '카피 레프트(copyleft)'라는 새로운 개념을 도입하고 있다.

자유 소프트웨어 공동체와의 사회 계약

1. 데비안은 **100%** 자유**(Free)**로 남을 것이다

데비안 자유 소프트웨어 지침 (The Debian Free Software Guidelines)이라 이름 지어진 문서에서 저작물이 자유인지 결정하는 데 쓰는 지침을 제공한다. 데비안 시스템과 모든 구성요소가 이 지침에 따라 자유로울 것임을 약속한다. 데비안에서 자유와 비자유 작업 모두 만들거나 사용하는 사람들을 지원할 것이다. 시스템이 비자유 소프트웨어의 사용을 요구하도록 만들지 않을 것이다.

2. 자유 소프트웨어 공동체에 되돌려 줄 것이다

데비안 시스템의 새로운 구성 요소를 작성할 때, 데비안 자유 소프트웨어 지침과 일치하는 방식으로 라이선스를 부여 할 것이다. 만들 수 있는 한 가장 좋은 시스템을 만들것이며, 그렇게 해서 자유 소프트웨어는 널리 배포되고 사용될 것이다. 버그 수정, 개선 건의, 사용자 제안 사항 등을 우리 시스템이 포함하는 소프트웨어의 "실제 (upstream)" 제작자에게 전달할 것이다.

3. 문제를 숨기지 않을 것이다

모든 버그 보고 데이터베이스를 일반 대중에게 항상 공개되도록 유지할 것이다. 사용자가 온라인으로 제출한 보고는 즉시 다른 사람이 볼 수 있게 된다.

4. 사용자와 자유 소프트웨어가 가장 우선이다

사용자와 자유 소프트웨어 공동체의 요구에 따를 것이다. 사용자들의 이익을 1순위로 할 것이고, 우리 사용자들이 다양한 컴퓨터 운영환경에서 데비안을 운용하는 데 필요한 사항들을 제공할 것입니다. 우리는 데비안 시스템에서 동작시키려는 상업적 소프트웨어에 반대하지 않는다. 다른 이들이 데비안 시스템과 상업적 소프트웨어 두 가지를 포함하는 기능을 좀더 추가한 배포본을 만드는 것도 허락할것이다. 이에 대한 어떠한 대가 지불도 필요 없고, 이러한 목표를 유지하기 위해 우리는 고품질의 통합 시스템을 제공할 것이다.

5. 우리의 자유 소프트웨어 규격에 맞지 않는 프로그램 일부 사용자가 데비안 자유 소프트웨어 지침**(Debian Free Software Guidelines)**에 따르지 않는 프로그램을 사용할 필요가 있다는 것을 알고 있다**.**

우리 아카이브에 이런 소프트웨어에 대한 보관 장소를 contrib와 non-free 영역에 만들었다. 이 디렉토리의 소프트웨어는 데비안 시스템의 일부는 아니지만 데비안에서 원활이 사용할 수 있도록 조정되어 있다. 우리는 CD 제조업자들이 이 디렉토리의 소프트웨어 패키지들 각각의 사용허가(license)를 검토한 후에 그들의 CD로 해당 소프트웨어를 배포할 수 있는지 결정하도록 권고한다. 이런 방식으로 우리는 데비안 시스템의 일부가 아닌 자유롭지 않은 소프트웨어의 사용을 지원하고, 이들 소프트웨어 패키지들에게 기본적인 것(버그 추적 시스템과 메일링 리스트 같은 것)을 제공한다.

오픈소스 저장소**(Repository)**

오픈소스 저장소들로는 대표적으로 6가지정도를 꼽을 수 있다.

GitHub **https://codeonweb.com**

호스팅 서비스에선 선두주자이다. 가장 많은 Integration을 지원하고, 가장 많은 오픈소스 저장소를 보유한다. 또한, 가장 안정적인 서버 상태를 제공하며, Github Status에서 실시간으로 확인 가능하다. 무료 플랜의 경우 Repo를 무조건 공유해야한다.

BitBucket www.atlassian.com

BitBucket Git을 사용하는 사람들에게 가장 익숙할 GUI 툴인 Sourcetree를 만든 Atlassian에서 제공하는 서비스이다. Unity Technologies의 일부 오픈소스는 Bitbucket의 저장소를 사용한다. JIRA, Hipchat과 연동이 편하고, Private Repo를 무료로 제공하지만 용량 제한이 심하다.

Google Code code.google.com

모바일 장치 용 오픈 소스 소프트웨어 스택이며 Google이 주도하는 해당 오픈 소스 프로젝트이다. 이 사이트와 AOSP (Android Open Source Project) 리포지토리는 Android 스택, 포트 장치 및 액세서리의 맞춤 변형을 만드는 데 필요한 정보와 소스 코드를 제공하며 Android 환경을 유지하는 호환성 요구 사항을 충족하도록 장치를 보장한다. 수백만 명의 사용자에게 건강하고 안정적인 환경을 제공한다. 공개 소스 프로젝트로서 Android의 목표는 한 업계 선수가 다른 플레이어의 혁신을 제한하거나 제어 할 수있는 중추적인 실패 지점을 피하는 것이고, 이를 위해 안드로이드는 거의 모든 기기에 공개 할 수있는 사용자 정의 가능한 소스 코드 및 모든 사람이 사용할 수있는 공개 문서 제공한다.

AOSP에 코드를 제공 할 수있는 것처럼 AOSP 문서에도 기여할 수 있다. Android의 융통성과 끊임없이 변화하는 코드베이스는 콘텐츠가 신선하고 정확하며 Android 구현 자와 관련이 있음을 알리기 위해이 사이트가 귀하의 의견을 필요로 함을 의미한다.

Code Project https://www.codeproject.com

SourceForge https://sourceforge.net

오픈 소스 프로젝트의 성공을 돕기 위해 전담하는 오픈 소스 커뮤니티 리소스이다. NAT은 오픈 소스 소프트웨어 개발 및 배포를 위한 초기 리소스를 만들 수 있도록 커뮤니티 협업을 통해 번창한다. NAT이 제공하는 툴을 사용하여 SourceForge의 개발자들은 50만개가 넘는 프로젝트에서 강력한 소프트웨어를 개발하고, 수백만 명의 등록 사용자를 호스팅한다. 인기 있는 디렉토리는 매달 3300만 명 이상의 사용자와 오픈소스 프로젝트를 연결하고 하루에 4백만 명 이상의 다운로드를 제공한다.

Naver nForge

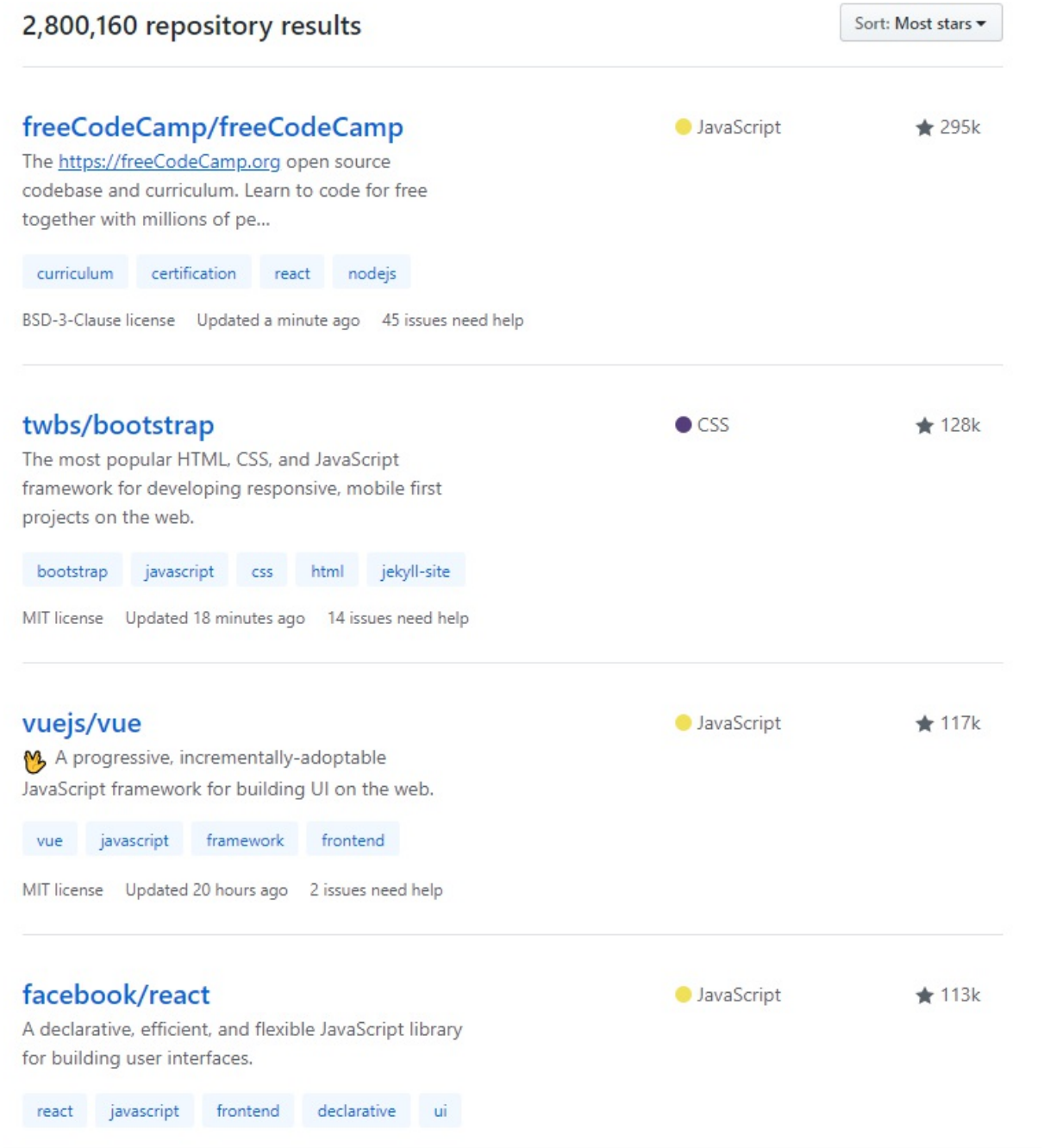
nFORGE는 소프트웨어 개발에 필요한 기능들을 사용하기 편리하게 웹으로 묶은 협업 개발 플랫폼입니다. 협업 개발 플랫폼으로, SW개발을 위한 프로젝트를 관리하고 소스코드를 관리한다. 국내 개발자들이 가장 쓰고 싶어하는 협업 개발환경을 비전으로 삼고있으며 버그나 이슈을 관리할 수 있는 버그 트래커, 각종 문서와 정보를 간편하게 공유할 수 있는 위키, 소스코드의 변경내역을 편리하게 관리할 수 있는 형상관리 툴, 일반적인 용도의 게시판, 그리고 최종 작업 결과물을 공유하기 위한 파일 릴리즈 기능 등을 포함하고 있다.

GitHub 상위 랭킹 프로젝트

흔히들 오픈소스 저장소로 GitHub를 많이 쓰는데 GitHub사이트에서는 상위 랭킹 프로젝트를 볼수있다. 다음은 Most Star와 Most Fork들의 수를 랭킹으로 매겨논 사이트이다.

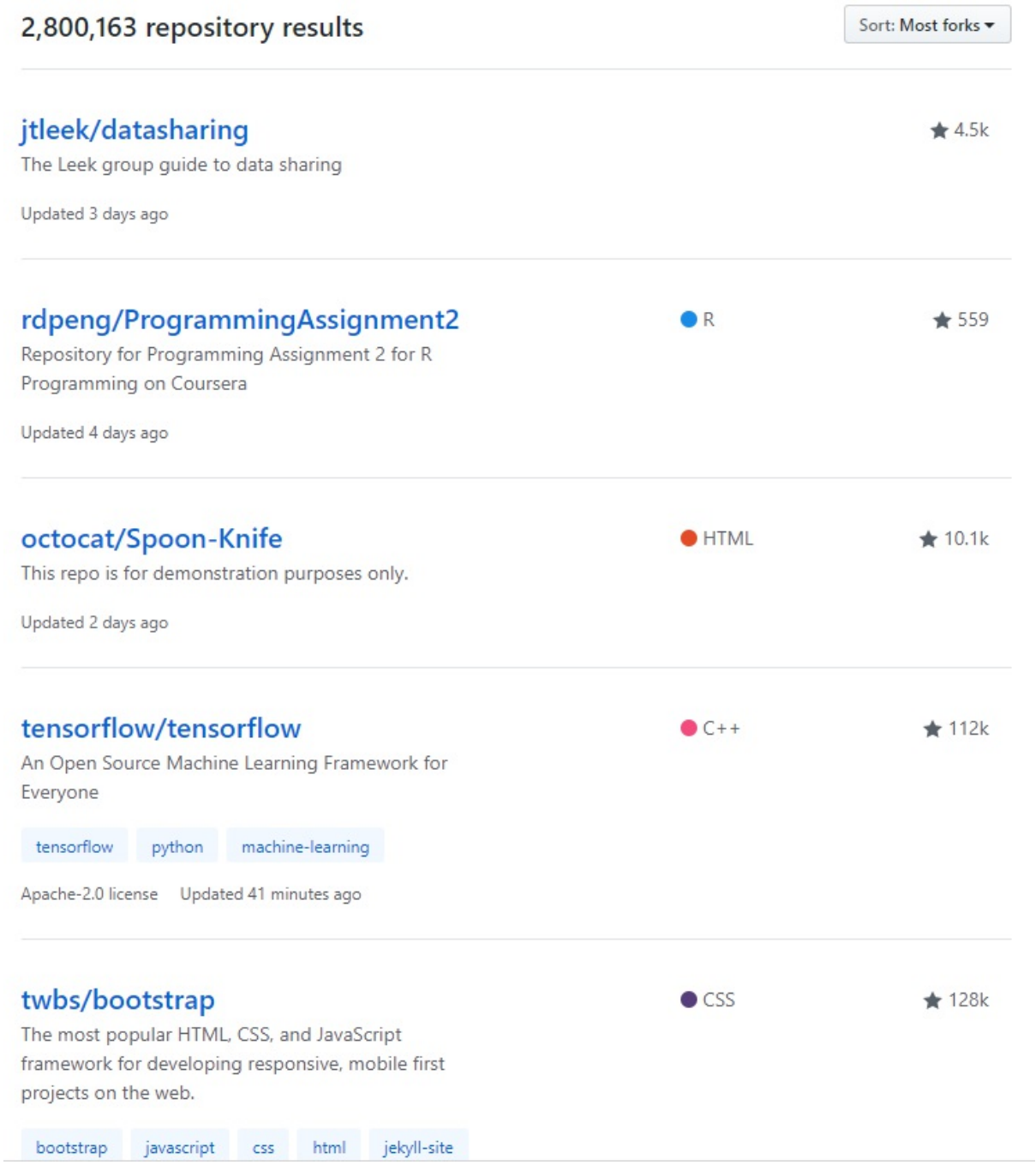
Most Starts

https://github.com/search?q=stars:%3E1&s=stars&type=Repositories



Most Forks

https://github.com/search?o=desc&q=stars:%3E1&s=forks&type=Repositories



오픈소스 기반 서비스 기업 사례

Google

세계 최대의 오픈소스 기업으로 900여개의 프로젝트, 2000만 라인이상의 소스를 공개하였다. 대표적으로 Android, Chrom ium, GWT 등이 있다.

facebook

200여개의 오픈소스 프로젝트를 제공하고 2014년 기준으로 1000만 라인 소스를 공개하였다. 대표적으로 React, Hbase, WebScaleS QL 등이 있다.

NETFLIX

BigData, 개발/빌드, 데이터 저장소 등의 분야에서 수십여개 프로젝트르 제공하였다. 대표적으로 Lipstick, Nebula, RxJs 등이 있다.

KAKAO TALK

S2 Graph, 아파치 재단의 오픈소스 인큐베이터 프로젝트에 채택, MongoDB, OpenSSL, WebRTC 등 오픈소스를 활용하여 주요 서비스 개발하였다.

오픈소스 기여 기업

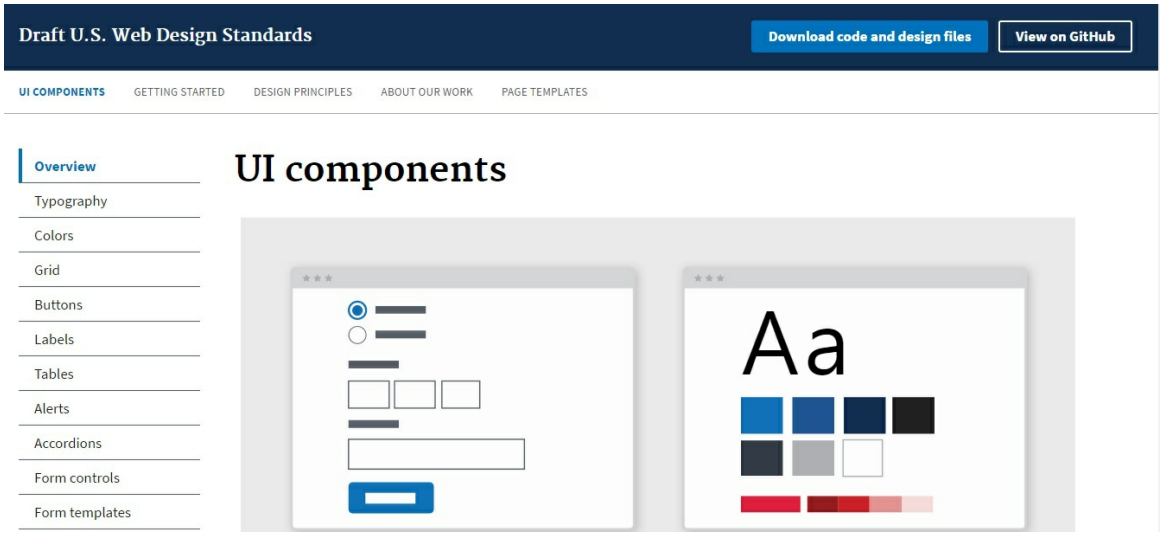
2017년 10, 구글 사용자 펠리페 호파는 오픈소스 프로젝트에 가장 많이 기여하는 기업을 파악하기 위해 깃허브 푸시이벤트를 분석했다. 이 추정에 따르면 기여자 수 측면에서는 마이크로소프트가 약 1300으로 1위이고 구글은 약 900명 정도로 2위를 차지하였다. 저장소로 푸시한 실제 코드 측면에서는 구글이 약1100개, 마이크로소프트가 약 825개로 나타났다. 아래 사진은 깃허브 기여자 랭킹과 깃허브에서 오픈소스 프로젝트를 능동적으로 기여하고 있는 총 직원 수의 개정된 결과이다.

Rank	Company	Employees Contributing
1	Microsoft	4,550
2	Google	2,267
3	Red Hat	2,027
4	IBM	1,813
5	Intel	1,314
6	Amazon.com	881
7	SAP	747
8	ThoughtWorks	739
9	Alibaba	694
10	GitHub	676
11	Facebook	619
12	Tencent	605
13	Pivotal	591
14	EPAM Systems	585
15	Baidu	584
16	Mozilla	469
17	Oracle	455
18	Unity Technologies	414
19	Uber	388
20	Yandex	351
21	Shopify	345
22	LinkedIn	343
23	Suse	325
24	ESRI	324
25	Apple	292
26	Salesforce.com	291
27	VMware	271
28	Adobe Systems	270
29	Andela	259
30	Cisco Systems	233

오픈소스 해외 사례

미국

1. 미국의 오마바 행정부는 끊임없이 공개SW에 지대한 관심을 표명해왔다.재임 초기 백악관의 홈페이지를 공개SW CMS인 워드프레스로 개편한 것을 시작으로,연방조달청 산하의 공공정보화 담당기관인'18F'를 통해 공공부문의 공개SW프로젝트 개발을 추진케하면서 공공에서 개발한SW를 공공,민간이 함께 공유해 코드 재사용성을 높이고, SW관리 효율성을 향상시키고자 했다.



[미 연방 각 기관이 보유한**SW**개발 코드를 공개하고 있는**code.gov**사이트**]**

code.gov사이트는 대표적으로 현재 소비자 금융 보호국, 노동부 등13개 기관이 보유한 기술을 공개하고 있으며,공공 웹사이트의 디자인 가이드라인을 소개하는U.S. Web Design Standards등이 있다. 지난해8월 백악관이 발표한 연방정부 소스코드 정책에 따르면,정부기관이 의뢰해 개발한 프로그램은 최소3년간 코드의20%를 공개해야 된다고 밝혔으며,미 정부 관계자는 이러한 조치가 공공 기술의 공유뿐 아니라 협업 문화 자체를 통한 기술 혁신을 위한 것이라고 한다.

1. 오클라호마대학교(OU)는2007년 이전까지 대규모로Solaris를 운영해왔다.점차 증가하는 유지 보수 비용과 공급업체의 경직성 문제로 타 하드웨어로의 변경을 고려하던 대학은 새로운 하드웨어로 교체 시 웹사이트부터 시스템까지 캠퍼스 내 모든 컴퓨터에 공개SW기반의 엔터프라이즈 리눅스 운영체제로 마이그레이션을 통해 상당한 비용 절감 효과와 사용 편의성을 두루 갖춘 유연한 IT 환경을 구축하게 되었다.

- 기　　관: 미국 오클라호마대학교
- 수행년도 : 2012년
- 도입배경 : 신규로 구축한 하드웨어에 확장성과 안정성을 갖춘 운영체제로 변경 필요성 대두
- 솔 루 션 : RedHat Enterprise Linux(RHEL)
- 도입효과 : IT 비용 절감 효과와 안정성, 사용 편리성을 갖춘 IT 환경 구축

프랑스

프랑스 국립헌병대는 매년 증가하는 IT소요 비용을 줄이고,업무 컴퓨팅 인프라를 보다 효과적으로 지원하기 위해 그간 사용해오던MS기반 시스템의 사용 적정성을 검토했다.문제로 지적된 MS기술에 대한 종속성을 극복하기 위해 2004년 공개 SW기반 PC업무 환경을 조성하는 프로젝트를 시작한다.애초엔MS Office를OpenOffice, Internet Explore를 Firefox로 전환하는 것이었으나 점차 확대되어 2008년엔 우분투OS의 자체 에디션인 ‘GendBuntu’로 전환을 시작하게 된다.이후 2014년 6월 77,000대 PC에 대해 전환을 완료했다. 프랑스 국립헌병대의 사례는 유럽에서 가장 큰 규모의 공개SW전환 프로젝트로 꼽힌다.

러시아

냉전의 시대는 끝이 났음에도 여전히 러시아와 미국 간 견제는 계속된다.러시아 정부의 서방IT기술에 대한 뿌리 깊은 불신과 염려는 미국의 전CIA소속 에드워드 스노든이NSA의 불법 도청 및 감청 폭로로 인해 더욱 가중된 듯하다. 강력한 리더쉽을 발휘하고 있는 러시아의 블라디미르 푸틴 대통령은 이전부터 리눅스 사용을 확대하자는 주장을 피력해 왔다.최근 러시아 의회 듀마(Duma)는 정부 기관이 조달 시 공개SW에 우선권을 부여해 상용SW를 구매를 제한하는 법안의 초안을 준비 중이라고 밝혔다.이는 그간 푸틴 정부가 주장한 온SW주권 그러니까,해외 거대IT기

업에 라이선스 및 SW구매 비용 등에 따른 자본의 국외 유출을 막고, 정보 보안 강화해SW경쟁력을 확보하기 위한 구체적인 조치이다. 지방 정부도 발빠르게 대응하고 있는 가운데 수도 모스크바시는 푸틴 대통령의 디지털 주권 요청에 대한 화답으로 MS Outlook을 버리고 이를 자국SW로 대체할 예정이라고 한다. MyOffice라 불리는 SW는 모스크바시 공무원들이 사용하는 컴퓨터 6천대에 설치되며 현재 사용하는MS(MSFT, Tech30)의 이메일 서비스를 대체하게 된다. MyOffice는 러시아 기업이 개발했으며,향후60만 명의 시 공무원 PC전부 사용하게 된다.새로운 법안은 공공 기관과 국영 기업이 외산SW를 구매하기 전에 유사 기능의 국내 제품에 대한 존재 유무를 먼저 확인하는 조항도 포함하고 있어 30억 달러 규모의 조달 시장의 오라클, MS등 외산 기술에 대한 의존도를 차단함으로써 자국SW시장 성장도 기대할 것으로 내다봤다.러시아 정부 관계자는 이미 일부 지방 정부에서는MS윈도우, 오라클DB를 공개SW기반으로 전환을 시작했다고 밝혔다. 또 지난해12월 당국은 정부의 공식 모바일OS개발을 위해 핀란드 기업 욜라(Jolla)가 개발한‘세일피시(Sailfish) OS’를 선택한다고 발표했다. 2015년 기준 시장점유율 약95%를 차지하고 있는 안드로이드나 iOS를 자국산 모바일OS로 대체해 향후 이들의 비중을 2025년까지 50%선으로 줄인다는 계획이다.

일본

일본의 대형 소매업체인Daiei사는 다양해지는 고객층의 라이프스타일에 맞춘 서비스를 제공하기 위해 주문 프로세스를 개선하기로 결정하고 새로운 시스템 구축에 비용과 시간을 절감하며 운용효율성을 높이기 위한 방법으로 KVM(Kernel-based Virtual Machine)가상화 기술 도입하기에 이르렀다.결과적으로 고객 주문 프로세스가 간소화되고 배송 리드타임이 줄어들었으며 성공적인 가상화 인프라로 운영효율성과 비용 절감을 충족할 수 있게 되었다.

Daiei사는 일본에서 큰 규모의 슈퍼마켓 체인 중 하나이다.현재Daiei의 이름뿐만 아니라 자회사를 통해3,000개 이상의 매장을 운영중으로 식료품,다이 전자 제품,가구 및 옷 등을 취급하며 순 매출액의 측면에서 일본 최대의 소매업체로 널리 알려져 있다.

프로젝트의 현재 초점은 새 시스템을 모든 매장에 성공적으로 설치하고 안정적인 작동을 수행하는 것이지만, Daiei사는 기존 관리 서버를 각 매장에서 선물 주문 관리 시스템을 실행하는 두 서버로 구성된 가상화 인프라로 모을 계획이다.그러면 운영 효율성과 비용 효과가 증가될 것으로 기대하고 있다. 현재 각 매장에서는 관리 서버3대를 작동하고 있으며 나머지3대는 백업으로 두고있다.가상화 환경에서 실행되는 새로운 선물 주문 시스템에는 작동 서버1대와 백업 시스템1대가 있다.앞으로 우리는 이 서버를 선물 주문 시스템이 실행되는 가상화 환경에 모으고 작동 서버1대와 백업 서버1대의 구성으로 모든 관리 애플리케이션을 실행할 계획이다.그로 인해 기존에 비해 한층 운영 효율성이 증가되고 비용 절감과 더불어 고객에게 제공하는 서비스가 향상되었다고 보고 있다.

중국

중국 정부는 자국의 중요한 정보를 보호하고 해외 IT거대 기업에 대항할 수 있는 IT생태계를 조성하기 위해서는 가장 기초적인SW인OS를 제어해야 된다고 보고 오랜 기간에 걸쳐 OS개발에 공을 들이면서 완성도를 높여왔다. 개발 노력과 함께 실질적인 활용에도 적극적인 모습이다. 최근 자국OS의 활용이 빠르게 확산되는 추세로 데스크톱뿐 아니라 모바일, 서버, 임베디드, IoT등 전 영역에 이른다.

중국의 운영체제 Kylin은 2001년 처음 등장했다. FreeBSD를 기반으로 하며 국방과학기술 대학이 공공기관에서 사용할 목적으로 개발했으나 완성도나 호환성이 떨어져 확산에 이르진 못했다. 이후 출시한3.0버전은 국방과학기술 대학과 중국 표준소프트웨어사가 합작 개발한 리눅스 커널 기반인NeoKylin OS를 선보인다.

위 사진은 데스크톱용Ubuntu Kylin 16.04화면이며 중국 날씨 정보, Sogou중국어 입력기, 킹소프트의 오피스 스위트를 포함하며 우분투 Kylin커뮤니티가 그래픽 패키지를 제공한다.

2013년 우분투를 기반으로 한 프로젝트인Ubuntu Kylin 13.04버전도 릴리즈했다.영국 캐로니컬사, 국방과학기술대학, 우분투Kylin커뮤니티가 합작해 만들었다. 현재Ubuntu Kylin 16.04버전이 최신이며 MS오피스를 대체할 수 있는 우분투 기린용 킹소프트 'WPS오피스', 웹브라우저 진산 '례바오', 치후360 '360브라우저', 알리바바그룹 'UC브라우저'등이 있어 우분투 키린의 생태계 구축에 민간도 적극적이다. 중국은 행정용PC의 윈도우8을 금지했으며, 퀄컴사도 반독점과 관련해 과징금 폭탄을 맞는 등 해외IT기업에 대한 강도 높은 제재를 취하고 있다. 북쪽에 자리한 쓰핑시는2014년 행정용PC MS윈도우를 NeoKylin으로 전환하고 다른 지역에도 이와 같은 조치를 확대할 계획이다. 또 중국 시장에 진출한 글로벌IT기업들도 정부의 입장을 수용하는 가운데 미국Dell사는 향후 중국에서 판매하는 PC 42%분량에 대해NeoKylin을 탑재할 예정이라고 밝혔다. 현재 중국내 MS사용 비중은 95%이상을 차지하고 있지만 중국 정부의 자국OS보급 노력이 지속된다면 점유율도 빠르게 성장할 것으로 보고 있다. 또한 중국 정부는 2014년 스마트폰 OS인 COS(China Operating System)를 발표한 바 있다. 중국 정부 산하 중국 과학 아카데미 소프트웨어 연구소(ISCAS)가 리눅스를 기반으로 개발한‘ 중국 정부 공식OS’이다. 야심차게 출시했으나 현재로선 이렇다 할 확산 성과는 없다.그러나 자국IT기업들이 자체 개발한 OS는 좋은 성적을 거두고 있다. 대표적으로 알리바바가 독자적으로 리눅스를 기반해 개발한 모바일OS인 '윈OS'를 탑재한 스마트폰이 안드로이드에 이어 모바일OS 2위로 등극했다.

오픈소스에 관여한 인물들

1) 리처드 스톨먼(1953~)

리처드 스톨먼은 1953년 맨해튼에서 태어났다. 학창시절인 1965년부터 1969년까지 컴퓨터를 접하면서 프로그래밍을 배웠고, 1971년부터 매사추세츠 공과대학 인공지능 연구소에서 '*해커'로 일하기 시작했다. 그는 이 시기를 회고하면서 “요리법을 공유하는 것이 요리의 역사만큼 오래된 것처럼, 소프트웨어를 공유하는 것은 컴퓨터의 역사만큼 오래된 것이었다. 그러나 우리(해커들)는 MIT에서 소프트웨어를 공유하는 이상적인 공동체를 만들었다"고 말했다. (여기서 앞서 언급한 해커의 의미는 시스템 보안을 뚫고 해악한 일을 저지르는 사람이 아닌,‘프로그래밍을 좋아하고 능숙하게 다룰 수 있는 사람'이라는 의미이다.)
그는 인공지능 연구소에서 그의 동료와 함께 ITS라 불리는 운영체제를 DEC사의 PDP-10이라는 메인프레임(다양한 데이터를 처리할 수 있는 대형 컴퓨터)에 탑재하기 위해 업그레이드하는 작업을 맡았다. 기존 PDP-10을 운영할 인적 자원이 부족해진 인공지능 연구소는 ITS가 아닌 새로운 운영체제를 도입하지만 후술될 (오픈소스의 역사-자유 소프트웨어의 초기하락)에 언급됐듯이 이 운영체제는 자유 소프트웨어가 아니였다. 이러한 운영체제를 사용하려면 관련 자료(소스코드 등)를 유출하지 않겠다는 계약 조건에 동의해야만 했는데 리차드 스톨만은 이에 대해 "컴퓨터를 사용하는 처음 단계부터 주위 사람을 돕지 않겠다고 약속하는 것과 같은 의미"라고 언급했다.
리처드 스톨먼은 무너진 공동체를 다시 부활시키길 원했고, 이를 위해 가장 먼저 해야 하는 일은 운영체제를 만드는 것이라고 판단해서 그 결과 리차드 스톨만은 GNU 선언문을 발표했고, 자유 소프트웨어 개발을 위해 자유 소프트웨어 재단을 설립한다. 이후, 1989년에는 일반 공중 사용 허가서(GNU General Public Licence, 이하 GPL)이라는 카피레프트를 발표한다.

2)리누스 토발즈(1969~)

스웨덴계 핀란드인이며 리눅스와 Git의 아버지이다. 운영체제 수업 도중 MINIX를 만나게 되었고 이를 기반으로 리눅스의 최초 프로토타입을 1991년에, 그리고 1.0버전을 1994년에 발표하게 된다. 토발즈는 "오픈 소스만이 소프트웨어를 수행 할 수있는 유일한 방법" 이라고 믿었지만, 독점 소프트웨어가 포함되어 있다고하더라도 "일을 위한 최고의 도구"를 사용한다고도 말한 바가 있다. 그는 리눅스 커널에서 버전 관리를 위한 독점적인 BitKeeper 소프트웨어의 사용과 옹호에 대해 비판을 받게 되었으며 이후에 Git이라는 자유 소프트웨어 대체품을 작성하게 된다.

3)에릭레이먼드(1957~)

1970년 후반 아르파넷 시절부터 인터넷과 해커 문화에 매료돼 참여하고 관찰해온 해커이자 인류학자다. 그의 연구는 리눅스와 인터넷의 발전을 통해 효과적으로 증명된 분산화된 오픈소스 소프트웨어 개발 모델을 설명하는데 기여했다. 그는 컴퓨터에 매혹되기 전에 수학과 철학을 공부했고, 두 장의 앨범에서 플루트를 연주하는 등 음악가로서도 어느 정도 성공을 거뒀다. 직접 만든 여러 오픈소스 프로젝트 중에서 가장 널리 알려진 것은 페치메일이다. 페치메일은 인터넷에서 가장 많이 사용되는 이메일 전송 프로그램으로 모든 주요 리눅스 배포판에 포함돼 있다.

번외의 인물들

1)데니스 리치

데니스 리치가 컴퓨터를 처음 접하게 된 시기는 하버드대에서 '유니박 I' 강의를 들으면서부터였다. 이후 1963년 동 대학에서 물리학과 응용수학 학위를 받고, 메인프레임의 크기와 비용을 줄이는 연구를 처음 시작한 MIT공대에 들어갔다가 1967년 벨 연구소로 들어가게 된다. 당시 벨 연구소는 일괄처리 방식을 상호작용 방식으로 대체하는 아이디어로 멀틱스(Multics)를 개발하기 시작했는데, 후에 벨 연구소가 멀틱스 연구를 포기했음에도 상호작용 방식과 협업과 같은 OS에 대한 핵심 아이디어를 버릴 수 없었던 리치는 유닉스라고 불리는 멀틱스의 후속 OS를 연구하기 시작했다. 이 작업은 동료였던 케네스 톰슨과 같이 진행하게 되었다. 또한 최초의 유닉스를 개발하던 중 새로운 시스템이 필요하게 되어 PDP-7에서 새 기종인 PDP-11로 유닉스를 이식하게 되었다. 이 과정에서 여러 기종에서 유닉스를 이식하기 쉽게 하기 위하여, 케네스 톰슨의 B언어(BCPL을 바탕으로 만든 언어)를 바탕으로 새로운 언어를 만들게 되는데 이것이 바로 C언어이다. 이후 루슨트 테크놀로지스의 시스템 소프트웨어 연구부장 등을 역임하다가 2007년 은퇴하였다.

2)팀 버너스리

그는 CERN의 정보 시스템에서 착안하여 세계의 망을 하나로 묶는 거대한 인트라넷, www를 제안하였다. 월드 와이드 웹의 약자인 www는 흔히 웹이라고도 불리며, 지금의 인터넷 시스템을 이르는 말이기도 하다. 또한 그는 웹과 클라이언트(컴퓨터)를 연결하는 HTTP 프로토콜 방식도 고안하였고 사이트를 갈 수 있는 URL(인터넷 주소)이라는 방식도 고안해 냈다. 이 셋은 2018년 지금까지도 기본적으로 쓰이고 있으며, 고안된 뒤 25년간 큰 틀에서의 변화 없이 소소하게 업데이트가 되어 왔다. 그리고 팀 버너스리는 이 웹과 자신이 고안해낸 다른 여러 가지 기술들을 특허도 내지 않고 무료로 풀어버린다. 이것은 당장 자신의 형편보다는 앞으로의 인터넷의 자유로운 발전을 위해서 한 행동이었으며, 그로 인해 인터넷이 더 빨리 발전하게 된 것은 당연지사. 대중의 찬사를 받으며 2004년 기사작위도 받았다.

오픈소스의 재단**(Open Source Software Foundation)**

1.설립목적

OSSF는 사회 일반의 이익에 공여하기 위하여 민법 제32조의 규정에 따라, 공유와 기여를 강령으로 하는 건전한 오픈소스소프트웨어 및 오픈소스하드웨어 생태계의 조화로운 발전을 위하여 필요한 교육, 연구, 조사, 개발, 보급, 학술, 출판, 교류, 정책 및 입법 활동 지원 등을 목적으로 한다.

2.목적 사업

-오픈소스소프트웨어 및 오픈소스하드웨어 관련 교육 및 훈련 사업 및 지원

-오픈소스소프트웨어 및 오픈소스하드웨어 관련 개발 및 보급 사업 및 지원

-오픈소스소프트웨어 및 오픈소스하드웨어 관련 커뮤니티 국내 및 국제 활동 사업 및 지원

-중소기업에 대한 오픈소스소프트웨어 및 오픈소스하드웨어 관련 컴플라이언스 지원

-오픈소스소프트웨어 및 오픈소스하드웨어 관련 연구, 조사, 학술, 출판, 홍보 사업 및 지원

-오프소스소프트웨어 및 오픈소스하드웨어 관련 국내 및 국제 세미나 및 컨퍼런스 주관 및 지원

-오픈소스소프트웨어 및 오픈소스하드웨어 관련 지식 및 정보의 국내 및 국제 공유 사업 및 지원 -오픈소스소프트웨어 및 오픈소스하드웨어 관련 표준 활동 및 지원

-오픈소스소프트웨어 및 오픈소스하드웨어의 국제적 확산 지원

-기타 재단의 목적달성을 위하여 이사회에서 필요하다고 의결하는 사업

3.비전

-오픈소스 소프트웨어 발전에 기여한다.

-OSS관련 산업 활성화를 위한 종합적이고 실질적인 활동을 전개한다.

-일관성 있고 체계적인 법적 위험관리 방안을 제공한다.

-재단 관련 기관과 기업 그리고 커뮤니티를 대변한다.

-우수한 국내 SW 인력의 세계 활동을 지원한다.

-Asia의 OSS 중심축(Hub)이 되어 OSS의 아시아 브랜드를 만들어 나간다.

4.미션

-아시아 국가를 위하여아시아 국가의 오픈소스SW 주도권 확보를 통한 국가품격 향상한다.

-커뮤니티를 위하여아시아 오픈소스 프로젝트 및 커뮤니티의 활성화를 통한 글로벌 IT 기술 확보한다.

-기업 성장과 보호를 위하여 오픈소스를 통한 글로벌 경쟁력 확보와 라이센스 대응 지원한다.

-SW인력의 성장을 위하여 오픈소스를 통하여 SW 인력의 양성과 글로벌 활동을 지원한다. (오픈소스 사용에서 참여하고 주도하기 위해)

그 밖에 재단들

Apache Software Foundation

가장 대표적인 오픈소스 재단. 350개 이상의 프로젝트 관리

Software Freedom Conservancy

BusyBox, Git, Samba, Wine 등 33개 프로젝트 관리

GPL 의무화를 위한 GPL Compliance Project를 추진

Linux Foundation

Linux Kernel 관리. 최근 SDx, IoT, Embed, Cloud 등의 다양한 이머징 기술 협력

Eclipse Foundation

이클립스 개발툴을 비롯한 배포, 관리 오픈소스 관리

BI(Business Inteligence), IoT 등에 관한 200여 개 프로젝트 지원 주요 오픈소스 재단

Cloud Foundry Foundation

클라우드 구축에 필요한 인프라 소프트웨어들을 관리

EMC, HP, IBM, 인텔, SAP 등의 벤더가 공동 참여

Free Software Foundation

리처드 스톨만이 창립했으며, 소프트웨어의 자유로운 사용에 초점을 맞추고 있음

무료 운영 시스템 구현에 목표를 둔 GNU 프로젝트만 관리

Open Source Initiative

역시 특정 프로젝트가 아닌 '오픈소스 소프트웨어' 운동을 지원

Openstack Foundation

오픈스택 클라우드(Private Cloud) 운영 시스템 프로젝트 주요 오픈소스 재단

- 오픈소스의 역사
  - **1960**년대의 이야기
  - **1970**년대의 이야기
  - **1980**년대의 이야기
  - **1990**년대의 이야기
  - **2000**년대 이후의 이야기

1960년대의 이야기

1960년대 말 이전까지도 소프트웨어는 하드웨어와 함께 제공되는 것 정도로 여겨지고 있었는데 이유는 아직 미분화된 상태였기 때문이었다. 오히려 코드를 무료로 배포하는 경향도 있었는데 그 이유는 소스가 오픈되어야만 컴퓨터 기능이 향상되고, 그래야 더 많은 판매가 이루어질 수 있어서였다. 또한 1950~60년대에 거의 모든 소프트웨어는 공동 연구로 일하는 학자 및 기업 연구원에 의해 제작되었는데 종종 공용 도메인 소프트웨어로 공유되었다. 유닉스도 이런 자유스러운 분위기에서 탄생되었는데 특정한 하드웨어에 종속되지 않는 하나의 운영체제로 저렴한 컴퓨터에 맞게 만들어졌으며 소스코드도 함께 배포되었고 이후, 많은 사람들이 많이 고치고 수정해서 기능을 향상시켜 사용해 나가기 시작한다. 그러한 공동 행동은 나중에 소위 해킹 문화(오픈 소스 프로그래머들 사이에 긍정적인 의미가 있는 용어)의 핵심 요소가 되었다. 이후 1969년 소프트웨어 코드의 교환을 단순화시켜준 대륙 횡단 고속 컴퓨터 네트워크 인 ARPANET (Advanced Research Projects Network)가 구축되게 되었다. 반면, 1960년대 후반부터 운영 체제 및 프로그래밍 언어 컴파일러가 발전하면서 소프트웨어의 생산비용이 하드웨어에 비해 크게 증가하게 되고 이 현상은 이후 70년대의 경향에 크게 영향을 주게 되는데....

1970년대의 이야기

앞서 60년대 항목에서 언급했듯이 1960년대 후반부터 운영 체제 및 프로그래밍 언어 컴파일러가 발전하면서 소프트웨어 생산 비용이 하드웨어에 비해 크게 증가했다. 점차 증가하는 소프트웨어 산업은 하드웨어 제조업체의 번들 소프트웨어 제품(번들 제품 비용은 하드웨어 비용에 포함됨)과 경쟁하고 있었고 임대된 컴퓨터는 소프트웨어 지원을 필요로 하면서 소프트웨어는 수익을 내지 못하게 된다. 그 영향으로 TeX나 SPICE와 같이 1970년대에 개발된 일부 무료 소프트웨어는 계속 개발되어 사용되었지만 제한적인 라이센스 하에서만 판매되는 소프트웨어가 점차 증가하게 된다.

그 일례로 1970년대 초 AT&T는 유닉스의 초기 버전을 정부 및 학술 연구자에게 무료로 배포했지만 이 버전은 수정된 버전을 재배포하거나 배포할 수 있는 권한이 없었기 때문에 현대의 의미에서 자유 소프트웨어가 아니었다. AT&T는 1979년 Unix 시스템을 판매함으로써 수익을 올릴 수 있다고 결정했을 때 라이선스를 시행하기 시작한다. 거기에 수익을 높이기 위해 일반적으로 소스 코드를 배포하지 않고 소스코드에서 컴파일된 실행 가능 머신코드만 배포하기 시작했다.

1974년 미국 저작물의 새로운 기술적 사용에 관한 위원회(CONTU)가 저작자의 원작을 구현하는 범위 내에서 컴퓨터 프로그램이 저작권의 적절한 주제가 되기 전에는 소프트웨어가 저작권으로 간주되지 않았음을 밝혔다. 이에 따라서 소프트웨어에는 라이선스가 첨부되지 않았으며 일반적으로 소스 코드가 있는 공개 도메인 소프트웨어로 공유되었다.

1970년대 후반에 들어서 컴퓨터 벤더와 소프트웨어 전용 회사는 소프트웨어를 "프로그램 제품"으로 마케팅하고 저작권, 상표 및 임대를 통해 자산으로 간주되는 새로운 소프트웨어 개발에 법적 제한을 가하기 시작했다. 또한, 1976년 빌 게이츠는 애호가들에게 “공개서한”이라는 제목의 에세이를 썼는데 그 내용은 애용자가 라이선스 비용을 지불하지 않고 Microsoft의 제품인 Altair BASIC 을 광범위하게 공유하는 것에 대해 크게 낙담한다는 것이었다.

1980년대의 이야기

1970년대의 사건에 이어 IBM은 1983년 2월 8일자 발표에서 구입 한 소프트웨어로 더 이상 소스를 배포하지 않겠다는 정책을 발표한다. 또한, 유닉스가 1980년대 초반에 더 널리 보급된 후 AT&T는 무료 배포를 중단하고 시스템 패치를 부과했으며 다른 아키텍처로 전환하는 것이 매우 어렵기 때문에 대부분의 연구원은 상용 라이센스를 지불했다.

1. 비공식적인 소프트웨어 공유가 계속되다.

앞서 언급했던 소스를 배포하지 않는 상황 속에서도 여전히 소스 코드를 다른 프로그래머 또는 사용자와 무료로 공유하고 싶어하는 사람들이 있었으며 "취미자" 및 " 해커 "라고 불리게 된다.

1) DECUS 테이프

1980년대 초, 소위 DECUS 테이프는 DEC 장비 사용자를 위한 자유 소프트웨어 전송을 위한 전세계적인 시스템이었다. 운영 체제는 일반적으로 독점 소프트웨어였지만 TECO 편집기, Runoff 텍스트 포맷터 또는 List 파일 목록 유틸리티와 같은 많은 도구는 사용자의 삶을 더 쉽게 만들고 DECUS 테이프에 배포하기 위해 개발되었다. 이러한 유틸리티 패키지는 DEC의 독점 운영 체제의 새로운 릴리스에 통합하는 데 도움이 되었고 컴파일러조차도 배포 할 수 있었다. 예를 들어 Ratfor(및 Ratfiv)는 연구원이 Fortran 코딩에서 구조화 프로그래밍으로 이동하는 것을 도왔다(GO TO 문을 억제). 1981년 Decus 테이프는 아마도 사용자가 DEC 16 비트 PDP-11 및 VMS 운영 체제에서 실행 중인 32 비트 VAX 에서 유닉스 계열 시스템 을 사용할 수 있게 해주는 Lawrence Berkeley Laboratory 소프트웨어 도구 가상 운영 체제를 도입함으로써 가장 혁신적인 것이었을 것이며 이는 현재 Windows 용 Cygwin 시스템과 유사하다.

2) 1980년대 온라인 소프트웨어 공유 커뮤니티

80년대에는 자유 소프트웨어 운동과 나란히 소스 코드가 있는 소프트웨어가 BBS 네트워크에서 공유 되었으며 이것은 때로는 필수적으로 요구되었다. BASIC 및 기타 해석 언어로 작성된 소프트웨어는 소스 코드로만 배포 할 수 있었으며 그 대부분은 프리웨어였다. 사용자가 그러한 소스 코드를 수집하고 수정을 논의하기 위해 특별히 보드를 설정하기 시작했을 때 이것은 사실상의 오픈 소스 시스템이었다. 가장 확실한 사례 중 하나는 가장 많이 사용 된 BBS 시스템 및 네트워크 중 하나인 WWIV이다. 처음에는 Wayne Bell이 BASIC에서 개발했다. 자신의 소프트웨어를 "modding"하고 mods를 배포하는 문화는 매우 커져서 소프트웨어가 첫 번째 Pascal, 그 다음 C++ 로 포팅되면 등록된 사용자에게 배포되고 계속해서 mods를 공유하고 자신의 코드를 컴파일하게 된다.

2. 자유 소프트웨어 운동이 시작되다.

새로운 관행에 특히 괴로웠던 사람은 리처드 스톨만(<1>의 2.1항목 참조)이었다. 그는 처음에는 다른 사람들이 작성한 프로그램을 더 이상 연구하거나 수정할 수 없다고 우려했다. 스톨만은 이러한 관행을 윤리적으로 잘못된 것으로 보았고, 그에 대한 응답으로 1983년에 GNU 프로젝트를 설립하여 사람들이 자유 소프트웨어만을 사용하여 컴퓨터를 사용할 수 있게 했다.그는 1985년 비영리기구인 FSF(Free Software Foundation)를 설립하여 보다 공식적으로 프로젝트를 조직했으며 또한 카피 레프트(copy left)를 발명했다. 저작권의 대상이 되는 저작물의 "무료"상태를 보존하는 법적 메커니즘을 제공하고 이를 일반 공중 라이선스에 구현했다.

앞서 언급한 카피 레프트 라이센스는 저작자가 사용자에게 추가 요금 없이 저작물을 사용할 수 있는 권리와 프로그램의 완전한 소스 코드를 입수하고 연구하고 수정할 수 있는 권한을 포함하여 여러 가지 권리를 부여 할 수 있지만 파생 저작물에는 동일한 라이선스나 라이선스가 없는 저작물이 필요하다. 파생 상품에는 다른 원본 프로그램과의 조합이 포함되므로 다운 스트림 저작자는 최초 저작물을 독점 소프트웨어로 전환 할 수 없으며 카피 레프트 공유지에 기여하도록 초청받으며, 나중에 그러한 라이센스의 변형이 다른 사람들에 의해 개발 되었다.

또한, 리처드 스톨먼은 1983년 GNU 프로젝트를 시작하여 소스 코드의 사용에 제약이 없는 완전한 운영 체제를 작성하는 데 성공한다.이 동기를 유발하게 된 특정 사건이 있었는데 그것은 소스 코드가 사용자로부터 보류되었기 때문에 귀찮은 프린터를 수정할 수 없게 된 것이었다. 스톨먼은 또한 GNU 프로젝트의 목적을 설명하고 자유 소프트웨어의 중요성을 설명하기 위해 1985년에 GNU 선언문을 발표했다. GNU 프로젝트와 그 선언문에 대한 영감의 또 다른 영원한 이유는 스톨먼과 Symbolics 그리고 Inc 간에 MIT가 MIT 코드를 기반으로 한 Lisp 머신에 대한 Symbolics의 업데이트에 대한 액세스에 대한 의견 차이였다. 출시된 지 얼마 되지 않아 그는 "자유 소프트웨어" 및 설립 자유 소프트웨어 재단이라는 개념을 촉진 할 수 있었으며 자유 소프트웨어에 관한 정의는 1986년 2월에 출판되었다.

1989년에 GNU 일반 공중 사용 허가서의 첫 번째 판이 출판되었으며, 1989년에 일부 GNU 개발자가 Cygnus Solutions 사를 설립했다.

1990년대의 이야기

1990년대 후반, 포브스와 같은 출판물에서의 Linux의 주류 인식과 넷스케이프 브라우저의 소스 코드 발표와 함께 이러한 현상에 대한 관심과 참여가 두드러지게 증가했다. OSI는 1998년에 협력 개발 역사상 이 중요한 순간에 교육, 지지, 관리 기구로 설립되었다.

1 리눅스의 이야기 (1991~)

Linux는 1991년 Linus Torvalds에 의해 만들어진 유닉스 기반의 무료 오픈 소스 운영 체제이다. 사용자는 컴퓨터와 다른 장치에 대한 배포라고 하는 소스 코드의 변형을 수정하고 만들 수 있다. 가장 일반적인 용도는 서버이지만, 데스크톱 컴퓨터, 스마트폰, 전자책 리더, 게임 콘솔에서도 많이 활용된다.

리눅스 커널에 의해 시작되었으며, 지금까지는 GNU 프로젝트의 커널이 없었기 때문에 완전한 자유 소프트웨어 운영 체제가 존재하지 않았으며 토발즈의 커널 개발로 거의 모든 격차가 좁혀지게 된다. 거의 완성된 GNU 운영 체제와 리눅스 커널의 결합은 최초의 완전한 자유 소프트웨어 운영 체제를 만들었다. 1996년 이래로 리눅스 커널은 독점적인 라이선스 구성 요소를 포함하고 있어 더 이상 완전한 자유소프트웨어가 아니었는데 이에 자유 소프트웨어 재단에서 2008년 리눅스 커널의 수정된 버전을 출시하게 된다. 전체 시스템을 단순히 "Linux"로 언급하는 것은 일반적인 사용법으로 알려져 있다. 하지만 자유 소프트웨어 재단 그리고 많은 다른 사람들은 전체 운영 체제에 대해 보다 정확한 이름을 말하는 용어 "GNU / 리눅스"의 사용을 옹호하는 편이다. 여담이지만, 데비안 GNU는 리눅스 에 의해 시작되었다. 데비안 개발자의 원칙은 데비안 사회 계약에서 표현되는데 처음 시작된 이래로 데비안 프로젝트는 자유소프트웨어 재단과 긴밀하게 연결되어 있으며 1994-1995년에 이들의 후원을 받게 된다. 1997년 전직 데비안 프로젝트 리더인 Bruce Perens는 다양한 자유 소프트웨어 프로젝트를 위한 비영리 기금 및 지원 기관인 Public Interest에서 Software를 찾게 된다.

2 무료 BSD의 이야기 (1993~)

BSD 유닉스는 수년 동안 오픈 소스였고 최초의 오픈 소스 라이센스(BSD 라이센스) 중 하나였다. 하지만 안타깝게도 AT&T가 AT&T를 포함했기 때문에 이를 사용할 수 있는 별도의 라이센스가 필요했다. 이 문제는 마침내 1992년 윌리엄과 린 졸리츠가 386BSD를 출시하면서 해결되었다. 1989년 이후 개발되어 AT&T 라이센스와 무관하게 BSD의 첫 번째 완전 무료 오픈 소스 버전이었다. 그리고 그것은 오늘날에도 널리 사용되고 있는 BSD의 몇몇 버전을 낳게 되는데 그것이 FreeBSD, NetBSD 그리고 OpenBSD이다.

이 중, 1993년에 USL 대 BSDi 소송이 법정에서 해결 되었을 때, FreeBSD 와 NetBSD는 자유 소프트웨어로 배포되었다. 1995년 오픈 BSD는 포크 NetBSD에서, 그리고 2004년 FreeBSD에서 Dragonfly BSD가 나오게 된다.

3 오픈 소스의 출시

1997년, 에릭 레이먼드가 성당과 시장, 그리고 해커 커뮤니티와 무료 소프트웨어 원칙의 반사 분석을 하였다. 그리고 이는 1998년 초에 주목을 받았으며 Netscape Communications Corporation이 인기있는 Netscape Communicator 인터넷 제품군을 무료 소프트웨어 로 출시하도록 유도하는 한 가지 요소였다.

넷스케이프의 행동은 레이몬드와 다른 사람들이 자유 소프트웨어 원칙과 이점을 상용 소프트웨어 산업에 가져 오는 방법을 조사하게 했는데 그들은 FSF의 사회 활동가가 넷스케이프와 같은 회사에 호소력이 없다고 결론을 내리고 소스 코드 공유의 비즈니스 잠재력을 강조하기 위해 자유 소프트웨어 운동을 재편성하는 방법을 모색했다.

1998년 1월 네비게이터용 소스 코드 공개에 대한 넷스케이프의 발표에 대한 반응으로 캘리포니아 전략 세션에서 자유 소프트웨어 운동의 일부 사람들은 "오픈 소스"라는 레이블을 채택했다. 세션의 개인 그룹에는 "오픈 소스", Todd Anderson, Larry Augustin , Jon Hall , Sam Ockman, Michael Tiemann 및 Eric S. Raymond가 제안한 Christine Peterson이 포함 되었습니다. 다음 주가 되자 레이몬드 Raymond)와 다른 사람들은 그 단어를 퍼뜨리는 일을 하였으며 리누스 토발즈는 다음날 모든 중요한 제재를 가했다. Phil Hughes는 Linux Journal에서 강단을 제안하였고 자유 소프트웨어 운동의 개척자인 리차드 스톨만은 이 용어를 채택하면서 마음이 바뀌게 된다. 이 용어를 채택한 사람들은 네비게이터의 소스 코드가 공개되기 전에 "자유 소프트웨어"라는 이데올로기적이고 대결적인 의미를 벗어날 수 있는 기회를 이용했다. 넷스케이프는 Netscape Public License 및 Mozilla Public License에 따라 소스 코드를 발표했다.

이 용어는 기술 출판사 Tim O'Reilly가 1998 년 4 월에 조직한 행사에서 큰 호응을 얻게 된다. "오픈 소스 정상 회의 '라는 이름의 이벤트와 함께 리누스 토발즈를 포함하여 가장 중요한 자유 및 오픈 소스 프로젝트의 많은 지도자들이 참석하게 되는데 그 모임에서 자유 소프트웨어라는 이름으로 인한 혼란이 제기되었다. Tiemann은 "소스웨어"를 새로운 용어로, Raymond는 "오픈 소스"를 주장했으며 조립된 개발자들은 표를 얻었고 그날 저녁 기자 회견에서 승자가 발표된다. 5일 후, 레이몬드는 자유 소프트웨어 공동체에게 새로운 용어를 채택할 첫 번째 공개 전화를 하였다.

1990년대 말, "오픈 소스"라는 용어는 대중 매체에서 큰 인기를 얻었으며, 닷컴 업계의 거품 과 오픈 소스 소프트웨어가 주도하는 Web 2.0 이라는 맥락에서 소프트웨어 산업에서의 인정을 얻게 된다. 1990년대 이래로 오픈 소스 컴파일러 또는 인터프리터 형태의 새로운 프로그래밍 언어가 예외적인 것이 아니라 표준이 되었는데 1991년의 Python , 1995년의 Ruby , 2003년의 Scala 등이 그 예시이며, 최근에는 Java ,ActionScript ,C # 및 Apple의 Swift 등이 추가되게 된다. 또한, Java 런타임의 Java 소스 코드 부분이 1996년 첫 공개가 되게 된다.

4.Red hat의 설립

자체 리눅스 배포에 기반을 둔 회사인 Red Hat은 오픈 소스 큰 사업을 만들었다. 그 회사는 그 핵심에서 자유로운 것으로 높은 수익을 낼 수 있다는 것을 증명했다. Red Hat은 몇 년 동안 오픈 소스의 프로필을 크게 올리게 된다. 1990년대 후반에 Red Hat 주변에 얼마나 많은 소란이 있었는지를 알기 위해, 1999년에 공개되었을 때, 그것은 Wall Street 역사상 가장 큰 첫 날 상승의 하나를 기록한 것이다.

5.MySQL의 개발이 시작되다

Michael Widenius와 David Axmark는 1994년에 MySQL을 개발하기 시작했고 1995년에 첫 번째 버전을 출시했다. 지난 몇 년 동안 MyS QL은 선택의 여지가 있는 오픈 소스 데이터베이스 솔루션이 되었고 Facebook과 Wikipedia 같은 수많은 회사와 웹사이트에서 사용되고 있다. 2009년 현재 1,100만대 이상의 MySQL이 설치되었다.MySQL도 Red Hat과 마찬가지로 오픈 소스가 얼마나 큰 사업이 될 수 있는지를 보여주었다.

6.Apache의 이야기

Apache HTTP 서버는 오픈 소스 제품이 어떻게 시장을 거의 완전히 지배할 수 있는지를 보여 주었다. 최초의 웹 서버 중 하나인 NSCA HTTPd에 따르면 Apache는 1996년부터 지속적으로 인터넷에서 가장 널리 사용되는 웹 서버 소프트웨어이며, 쉽게 그 자리를 다른 웹 서버에게 내주지 않을 것으로 보인다고 한다.

7.Netscape가 자신의 웹페이지를 오픈소스화하다

넷스케이프는 마이크로소프트와 인터넷 익스플로러와의 점점 더 절박한 전쟁에서 마침내 1998년 초에 그것의 웹 브라우저를 열기로 결정하고 지배를 위해 오픈 소스 커뮤니티 모질라를 시작했다. 넷스케이프는 결국 무명으로 사라져 버렸지만, 이런 역사적인 움직임이 없었다면 모질라는 없었을 것이고 모질라가 없었다면 파이어폭스는 없었을 것이고, 웹 브라우저가 얼마나 영향력이 있었는지를 알수 있다.

2000년대 이후의 이야기

2000년 10월 13일, Sun Microsystems는 GNU Lesser General Public License 하에서 StarSuite 오피스 스위트를 무료 소프트웨어로 발표했으며, 자유 소프트웨어 버전은 OpenOffice.org로 이름이 바뀌었고 StarOffice와 공존하게 된다.

5 오픈 소스에 대한 Microsoft의 기여

2006년 Microsoft는 Microsoft 플랫폼을 대상으로 하는 오픈 소스 개발자를 위한 호스팅을 제공하기 위해 CodePlex 오픈 소스 코드 호스팅 사이트를 시작하게 된다. 2002년에 만들어진 Microsoft의 F # 컴파일러도 Apache 라이선스에 따라 오픈 소스로 릴리스된다. (F # 컴파일러는 오픈 소스가 아닌 Microsoft Visual Studio에 통합된 상용 제품이다.)

Microsoft 담당자는 수년 동안 다양한 오픈 소스 및 Linux 컨퍼런스에서 정기적으로 출연했으며 2012년 Microsoft는 오픈 소스 표준을 통해 독점적인 Microsoft 기술과 타사 기술 간의 격차를 해소하기 위해 Microsoft Open Technologies Inc.라는 자회사를 설립하지만 이 자회사는 이후 오픈 소스 및 비 Windows 플랫폼에 대한 Microsoft의 입장이 유리해짐에 따라 Microsoft로 다시 포개지게 된다. 또한, 2016년 1월, Microsoft는 MIT 라이센스 하에 Chakra를 오픈 소스로 출시했으며 이 코드는 GitHub에서 사용할 수 있다.

2 SCO v. IBM 및 관련 나쁜 홍보 (2003-)

2003년 독점적인 유닉스 공급 업체이자 이전의 리눅스 배포 업체인 SCO는 유닉스 지적 재산권이 리눅스 커널에 부적절하게 복사되었다고 주장하면서 IBM에게 책임을 묻는다고 주장하며 IBM을 고소하게 된다. 몇몇 관련 소송 및 반박이 뒤따랐는데, 일부는 SCO에서 비롯되었으며 일부는 SCO에서 소송을 제기했으나 SCO의 주장은 특이성이 결여되었고, 언론 매체의 일부는 신빙성 있는 것으로 보고했지만 SCO의 많은 비판자들은 이와 같은 주장이 의심스럽다고 생각했다.

SCO는 IBM 대 SCO 및 기타 여러 재판에서 패배한 후 2007년 파산 신청을 했으나. SCO가 저작권을 소유하지 않았음을 발견했음에도 불구하고 CEO인 Darl McBride는 더 이상 회사를 운영하지 않았으며 SCO와 파산을 담당하는 파산 관재인은 SCO 대 IBM 소송에서 관련이 있다고 주장하는 일부 부분을 계속 보도하게 된다. 그는 SCO의 주 법률 회사인 IBM과 IBM의 계약이 체결된 지 얼마 안 되는 고정된 금액으로 SCO를 대표하기로 계약을 체결했기 때문에 분명히 이를 감당할 수 있었다고 한다.

2004년 알렉시스 드 토크 빌 기관(ADTI)은 오픈 소스 코드의 '출처'에 관한 Samizdat와 다른 이슈를 발표하면서 리눅스 커널이 유닉스에서 도난당한 코드를 기반으로 한다는 사실을 발표했는데 이 책은 인터뷰한 사람들을 포함하여 셀 수 없이 많은 사람들에게 광범위하게 비난을 당하거나, 조롱당한 이후에는 결코 출판되지 않게 되는 운명을 맞이한다.

많은 사람들은 Linux 커널에 대한 이러한 법적 및 공포, 불확실성 및 의심 공격의 일부 또는 전부가 Microsoft에 의해 은밀하게 처리 된 것으로 의심되는 것으로 나타났으나 ADTI와 SCO는 모두 Microsoft로부터 자금을 지원받는다고 한다.

3 ISO OOXML 논쟁 (2008-)

2008년 국제 표준화기구는 Microsoft의 Office Open XML을 국제 표준으로 발표했는데 이 내용인 즉슨, 법률 및 정책에 따라 개방형 표준을 사용해야 하는 프로젝트에 Microsoft Office를 사용할 수 있다는 것을 의미한다. 프로세스 자체에 관련된 ISO 국가위원회의 일부 구성원을 포함하여 표준화 프로세스 비평가는 프로세스의 불규칙성 및 절차상의 위반을 주장하고 문서화되지 않은 Microsoft Office 동작을 참조하기 때문에 ISO가 OOXML을 표준으로 승

인하면 안 된다고 주장했다. 이는 2012년 현재 OOXML에 대한 올바른 오픈 소스 구현이 존재하지 않으므로 OOXML 구현 및 지정이 어렵다는 비평가들의 발언을 통해 입증된다. 현재 Google은 Office 문서를 자체 독점, Google 문서 형식으로 올바르게 변환 할 수 없는데 이는 OOXML이 진정한 개방형 표준이 아니라 Microsoft Office의 기능을 설명하고 일부 파일 형식만을 포함하는 부분적인 문서임을 의미한다.

4 자바의 이야기

Java 런타임의 Java 소스 코드 부분은 무료로 다운로드 할 수 있음에도 불구하고 "기밀"기반으로 Java Development Kit (JDK)에 포함되었지만 1996년 첫 공개 이후 Java 플랫폼은 오픈 소스가 아니었다. Sun은 JDK의 초기 버전을 Linux로 이식하거나 Sun의 Linux 포트인 JDK를 개선한 자원 봉사자 모임인 Blackdown Java 프로젝트에 기밀로 JDK 소스 코드를 제공했다. 그러나 모든 경우에 Sun의 허락없이 수정 및 재배포가 금지되었기 때문에 이것들 중 그 무엇도 오픈 소스가 아니다. GCJ는 특히 Java 구현 시 GCJ를 출하한 Fedora와 Ubuntu와 같은 배포판을 지원하는 자유 소프트웨어에서 Java에 대한 좋지 않은 사용자 경험을 유발했으며 또한, GCJ를 Sun JDK로 대체하는 방법은 사용자가 자주 묻는 질문이었는데 그 이유는 GCJ는 불완전한 구현 이었기 때문에 호환되지 않고 버그가 있었기 때문이다.

2006년 조나단 슈왈츠(Jonathan I. Schwartz)는 Sun Microsystems의 CEO가되었고 오픈 소스에 대한 헌신을 표명했는데 2007년 5월 8일, Sun Microsystems는 GNU General Public License에 의거하여 Java Development Kit을 OpenJDK로 발표했다. 클래스 라이브러리의 일부(4%)는 다른 당사자의 라이센스로 인해 오픈 소스로 공개될 수 없으며 바이너리 플러그로 포함되게 된다.

5 Ubuntu의 이야기

남아프리카의 백만장자 마크 셔틀워스의 회사인 캐노닉이 2004년에 데비안을 기반으로 한 우분투를 출시했을 때, 그것이 얼마나 큰 성공을 거둘지 거의 예상하지 못했을 것이다. Ubuntu는 지금까지 빠르게, 특히 데스크톱에서 가장 널리 사용되는 Linux 배포판이 되었으며, Linux를 다른 배포와 달리 대중들에게 보급했다.

6 분산 버전 제어 **(2001-)**

최초의 오픈 소스 분산 개정 관리 시스템 (DVCS)은 2001년에 출시된 'tla'(GNU arch로의 개명 이후)였습니다. 그러나 Bazaar가 여전히 계속 사용되고 있었으며, Canonical에 의해 사용 되더라도 그것과 그것의 후계자 'baz'와 'bzr'(Bazaar)는 인기가 없었다. 그러한 이유로 인해 'tla'는 결국 중단되고 말지만 다른 DVCS 프로젝트가 생겨나 고 일부는 상당한 채택을 시작하게 된다.

6 깃의 이야기 **(2005-)**

그 이후로 가장 인기 있는 DVCS가 된 Git은 2005년에 만들어지게 되는데 이것에 얽힌 창작의 이야기는 매우 특이하다. 일부 리눅스 커널 개발자, 특히 리눅스 창시자인 리누스 토발즈(Linus Torvalds)는 비트 커퍼(BitKeeper)라는 독점적인 DVCS를 사용하기 시작했다. 그런데 앤드류 트리 젤(Andrew Tridgell)이 비트 키퍼를 리버스 엔지니어링하여 리눅스 커널 개발과 동일한 기능을 제공 할 수 있는 오픈 소스 툴을 개발하기 시작했을 때, 리눅스 커널 개발이 독점 소프트웨어에 의한 사용과 관련된 비정상적인 상황으로 인해 BitKeeper를 개발 한 회사인 BitMover는 2005년에 특정 커널 개발자에게 부여한 무료 라이센스를 취소하고 만다.

리누스 토발즈는 BitKeeper 라이센스를 잃어버리고 말자 git이라는 자체 DVCS를 작성하기로 결정했는데 그 이유는 기존의 오픈 소스 DVCS가 커널 유지 관리자로서 자신의 특별한 요구에 적합하다고 생각하지 않았기 때문이다. 많은 다른 개발자들이 신속하게 뛰어 들어 도움을 주었고, 시간이 지남에 따라 비교적 단순한 "어리석은 콘텐츠 추적기"에서 오늘날의 정교하고 강력한 DVCS로 성장했습니다. 토발즈는 더 이상 이를 유지하지 않았고 그것은 Junio Hamano를 비롯한 많은 개발자로부터 오랜 기간 동안 계속해서 유지 및 기여를 받고 있다.

이렇게 git과 같은 오픈 소스 DVCS의 높아진 인기, 그리고 DVCS 호스팅 사이트(일례로 2008년 설립된 GitHub)로 인해, 자유 소프트웨어 프로젝트 참여의 장벽은 점차 축소되게 된다. GitHub과 같은 사이트에서는 더 이상 잠재적인 기여자가 소스 코드 저장소의 URL을 찾아야하거나, 패치를 생성하는 방법을 알아내야 할 필요 없이, 필요한 경우 올바른 메일의 링리스트에 가입하여 패치 전자 메일을 올바른 사람들에게 전달 및 단 한 번의 클릭으로 저장소의 사본을 포크하고 변경이 준비되면 해당 지점에서 끌어 오기 요청을 발행할 수 있게 되었다. GitHub는 오픈 소스 소프트웨어를 위한 세계에서 가장 인기 있는 호스팅 사이트가 되었으며, 이는 포크의 용이성 및 포크의 가시성과 함께 기여자가 크고 작은 변화를 창출 할 수 있는 인기 있는 방법이 되었다.

7 최근 개발

작성자가 소프트웨어의 라이센스 준수를 보장하기 위해 사용하는 주요 법적 메커니즘은 저작권이지만 그 외에 법령, 소프트웨어 특허 및 상표와 같은 다른 메커니즘도 사용되는데 자유 소프트웨어 재단은 이러한 특허권과 DMCA에 대한 법적 문제에 대응하기 위해 DMCA의 디지털 저작권 관리\(DRM\) 조항과 특허권을 명시적으로 다루는 2007년 GNU Public License 버전 3\(이하 GPLv3\)을 발표한다.

GNU 컴파일러 콜렉션(GCC) 소프트웨어와 같은 많은 GNU 시스템의 저작권 소유자인 GNU GPLv3의 개발 이후, FSF는 GPLv2에서 GPLv3에 이르는 GNU 프로그램의 라이센스 대부분을 업데이트했다. GCC와 DRM의 주요 사용자인 Apple은 Xcode IDE의 컴파일러를 GCC에서 다른 FOSS 컴파일러인 Clang으로 전환하기로 결정했으나 이는 전부 허용된 라이센스 하에 있으며 LWN은 애플이 GPLv3을 피하려는 욕구에 부분적으로 동기를 부여했다고 추측했다. 삼바 프로젝트 역시 GPLv3으로 전환했으며 애플은 소프트웨어 스위트에서 클로즈드 소스, 독점 소프트웨어 대안으로 교체하게 된다.

또한 최근의 합병은 주요 오픈 소스 소프트웨어에 영향을 미쳤는데 Sun는 2008년에 인기 있던 오픈 소스 MySQL 데이터베이스의 소유자인 MySQL AB를 인수하였으며, 마치 먹이사슬과 같이 오라클은 2010년 1월 sun을 인수하여 저작권, 특허 및 상표권을 획득했는데 이로 인해 오라클은 가장 인기 있는 독점적 데이터베이스와 가장 유명한 오픈 소스 데이터베이스 (MySQL)의 소유자가 되게 된다. 그러나 오라클은 오픈 소스 MySQL 데이터베이스를 상용화하려고 시도했고, 이러한 시도는 OSS 커뮤니티에 우려를 제기하고 만다. 부분적으로 MySQL의 미래에 대한 불확실성에 대응하여 OSS 커뮤니티는 프로젝트를 오라클의 통제 범위를 벗어난 새로운 데이터베이스 시스템으로 전환했다. 이러한 시스템에는 MariaDB, Percona 및 Drizzle이 포함되는데 이것들은 뚜렷한 프로젝트이며 상표등록 된 MySQL을 사용할 수 없다.

8 안드로이드에 관한 이야기 **(2008-)**

2008년 9월, Google은 새로운 스마트 폰 운영체제인 Android의 첫 번째 버전을 오픈 소스로 출시했다. Android에서 번들로 제공되는 일부 Google 애플리케이션은 오픈 소스가 아니었는데 초기 운영체제는 Google에 의해 무료로 배포되었으며 많은 핸드셋 제조업체에서 열심히 채택되었다. 구글은 나중에 모토로라 모빌리티 (Motorola Mobility)를 인수하여 자체 "바닐라 (vanilla)"안드로이드 폰과 태블릿을 생산하면서 다른 제조업체들도 안드로이드를 계속 사용할 수 있게 했다. 안드로이드는 리눅스 커널을 기반으로 하기 때문에, 이것은 리눅스가 현재 안드로이드를 통한 모바일 플랫폼과 슈퍼 컴퓨터, 그리고 서버 운영체제의 주요 플레이어 모두에서 지배적인 커널이라는 것을 의미하는 것이나 마찬가지이다.

9 오라클과 구글의 법정다툼

2010년 8월 오라클은 Android에서 Java 사용이 오라클의 저작권 및 특허를 침해했다고 주장하면서 Google에 소송을 제기했다. 오라클과 Google 사이의 첫 재판은 2012년 5월에 끝났으며 재판관은 Google이 오라클의 특허를 침해하지 않았으며 Google이 사용하는 Java API (Application Programming Interface)의 구조가 저작권이 적용되지 않는다고 판결했다. 배심원단은 구글이 사소한("미니멈 미니") 저작권 침해를 한 사실을 발견했으나 당사자들은 구글이 아무런 손해 배상을 하지 않을 것이라 명언했다. 그러나 오라클은 연방 순회 항소 법원에 항소했으며 Google

은 이에 대한 맞대응으로 문자 그대로 복제 청구에 대해 교차 항소를 제기했다. 연방 순찰 경비대는 Google이 인정한 작은 저작권 침해가 극소주의가 아니라고 판결하고 공정 사용 문제를 재판 판사에게 재검토 요청했으며 2016년 공정 거래 재판을 위해 Google을 통해 배심원이 나오게 된다.

10 Chromium OS의 이야기 (2009-)

최근까지도 리눅스는 여전히 데스크탑과 랩톱을 위한 운영 체제의 비교적 드문 선택이었다. 그러나, 웹 씬 클라이언트인 Chrome OS를 실행하는 Chromebook은 미국의 300 달러 이하 노트북 시장에서 20-25%를 차지하며 Chrome OS는 Linux를 기반으로 하는 오픈 소스 Chromium OS를 기반으로 한다.

참고문헌

- https://www.slideshare.net/JerryJeong2/ss-58804386
- https://moonjewoong.gitbooks.io/os_red_01/Advantage_of_OSS.html
- http://www.ciokorea.com/news/19907
- http://www.gnu.org/philosophy/free-sw.html
- https://translate.google.com/translate?hl=ko&sl=en&u=https://source.android.com/&prev=search
- https://sourceforge.net/about
- https://www.slideshare.net/deview/hello-worldn-forge
- https://ko.wikipedia.org/wiki/%EB%A6%AC%EB%88%84%EC%8A%A4_%ED%86%A0%EB%A5%B4%EB%B0%9C%EC%8A%A4
- https://ko.wikipedia.org/wiki/%EC%97%90%EB%A6%AD_%EB%A0%88%EC%9D%B4%EB%A8%BC%EB%93%9C
- http://ossfadm.wr01.dhrcenter.com/?page_id=1226
- https://github.com/search?q=stars:%3E1&s=stars&type=Repositories
- https://github.com/search?o=desc&q=stars:%3E1&s=forks&type=Repositories
- https://en.wikipedia.org/wiki/History_of_free_and_open-source_software
- http://www.itworld.co.kr/print/108114
- https://www.sciencetimes.co.kr/?news=%EC%9E%90%EC%9C%A0%C2%B7%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4-%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4-%ED%95%B4%ED%82%B9-%EC%97%AD%EC%82%AC

- http://files.idg.co.kr/itworld/image/2018/02/0209_11_%EC%BA%A1%EC%B2%98.JPG