————————————————————현준—————————————————————

1-3

과거의 자유 소프트웨어

혀거 1950년대와 1960년대엔 대부분의 소프트웨어가 공동으로 일하는 학자 및 기업 연구원에 의해 개발되었다. 그리고 개발된 소프트웨어는 주로 공용 소프트웨어(Public Domain Software)의 형태로 공유되었는데, 이는 학계에서 오랫동안 확립되어온 개방과 협력의 원칙에 따라 배포된 것이었다. 이렇듯 소프트웨어를 상품으로 여기는 경향은 존재하지 않았고, 오히려 소스코드가 소프트웨어와 함께 공유되는 것이 일반적이었다. 이는 유저들이 소프트웨어를 자주 수정했기 때문이었다. 그 당시에는 어떤 하드웨어 또는 운영체제를 사용하는지에 따라 소프트웨어가 작동할 수도, 하지 않을 수 있었다. 따라서 유저들은 자신의 필요에 따라 소프트웨어를 수정했으며 때로는 버그를 수정하기도 했고 새로운 기능을 추가하기도 했다.

자유 소프트웨어의 하락세

1960년대 후반에는 운영체제와 컴파일러가 발전하면서 소프트웨어의 개발 비용이 하드웨어에 비해 크게 증가했다. 점차 성장하는 소프트웨어 산업은 하드웨어 제조업체의 번들 소프트웨어 제품과 경쟁하고 있었는데, (당시 번들 소프트웨어 제품의 비용은 하드웨어 비용에 포함되었다.) 소프트웨어와 관련해 더 많은 지원이 필요하면서 이에 대한 비용을 지불할 여건이 되는 소비자들은 제조업자들의 소프트웨어 제품 비용이 하드웨어 비용에 포함되는 것을 원하지 않았다. 그리고 1969년 미국 정부가 IBM과의 독점금지 소송에서 번들 소프트웨어가 반경쟁적이라고 기소하는 사건이 있었고, 이후 제한적인 라이선스 하에 판매되는 소프트웨어들이 점차 증가했다. 물론 몇몇 소프트웨어들은 여전히 무료로 제공되었다.

이러한 자유 소프트웨어의 감소를 보여주는 예로 UNIX를 들 수 있다. 1970년대 초 AT&T는 정부와 학술 연구자에게 Unix의 초기 버전을 무료로 배포하였다. 하지만 이 버전은 수정된 버전을 배포할 수 있는 권한이 없었기 때문에 자유 소프트웨어라고 할 수 없었으며, Unix가 널리 보급 된 1980년대 초반에 AT&T가 무료 배포를 중단하고 시스템 패치를 진행하면서 대부분의 연구원들은 상용 라이선스를 지불해야 했다.

1970년대 후반과 1980년대 초반에 컴퓨터 판매 회사와 소프트웨어 개발사들은 소프트웨어 라이센스를 판매하기 시작했고 저작권, 상표 등을 이용해 소프트웨어의 활용을 법적으로 제한하기 시작했다. Microsoft의 설립자인 빌 게이츠가 1976년에 쓴 ‘Open letter to Hobbyists’ 라는 제목의 에세이는 애호가(Hobbyists)들이 라이선스 비용을 지불하지 않고 자사의 소프트웨어를 공유하는 것에 대한 불만을 표현하는 내용이었다. 그리고 AT&T는 Unix의 판매를 통해 수익을 얻기 위해 라이선스를 적용했으며 1983년엔 IBM이 더 이상 소스코드를 배포하지 않겠다는 정책을 발표했다.

비공식적인 소프트웨어 공유

이러한 기업들의 정책에도 여전히 소스코드를 다른 사람들과 무료로 공유하고자 했던 사람들이 있었는데, ‘hobbyist’ 또는 ‘hackers’ 라고 불리는 사람들이었다. 이들은 공유를 위한 그룹을 설립하기도 했는데, 1955년에 설립된 ‘SHARE’ 라는 그룹은 자유 소프트웨어를 수집하고 배포하기 시작했다. 인터넷이 도입되고 널리 사용되기 이전에는 소스코드를 공유하는 여러 가지 방법들이 존재했는데, 이 그룹이 1955년 10월에 최초로 배포한 문서는 자기 테이프에 소프트웨어와 정보를 담은 것이었다. 이 외에도 컴퓨터 관련 잡지나 프로그래밍 서적과 같은 형태로도 공유되었다.

1980년대에는 자유 소프트웨어 운동(뒤에서 자세히 설명할 것이다.)과 나란히 소스 코드가 있는 소프트웨어가 BBS(게시판 시스템) 네트워크에서 공유되었다. BASIC 및 기타 언어로 작성된 소프트웨어는 소스코드로만 배포할 수 있었는데, 사용자가 그 소스코드를 수집하고 수정에 대해 논의하기 위해 BBS 네트워크를 이용한 것은 사실상 오픈 소스라고 할 수 있다. 1980년대 초 Usenet과 UUCPNet 의 출현으로 프로그래밍 공동체는 더욱 커졌고 프로그래머들에겐 자신의 소프트웨어를 공유하고 다른 사람들의 소프트웨어 개발에 기여할 수 있는 더욱 간단한 길이 열렸다.

————————————————————현준—————————————————————

**--------------------------------------석형원---------------------------------------**

1-4 자유 소프트웨어

1) 자유 소프트웨어의 등장

1979년 제록스사는 최초의 레이저 프린터 중 하나를 MIT에 있는 인공지능연구소에 기증하였다. 그 프린터는 자주 고장을 일으켰고 인공지능연구소의 한 해커는 제록스에게 프린터를 제어할 수 있는 코드를 제공해 줄 것을 요구했다. 프로그램을 수정해서 인쇄종이가 프린터에 끼어 중단되는 것을 사용자에게 경고 메시지를 보여줄 수 있도록 만들려 하였다. 이렇게 된다면 프린터는 빠르게 복구될 수 있을 것이다. 이런 작업을 하기 위하여 해커에게는 프린터 프로그램의 소스 코드가 필요했다. 다른 사람의 코드를 빌리고 수정하기도 하면서 함께 연구하는 학구적인 분위기의 인공지능연구소에서 일하는 프로그래머인 그에게 이러한 요청은 별로 유별난 것이 아니었다. 게다가 이전에 제록스사는 같은 문제가 있었던 프린터의 소스코드를 그에게 준 적이 있기도 하였다. 그러나 이번에 제록스사는 이 요청을 거부했다. 담당자는 기밀 유지 협약에 서명하였고 회사는 소스코드의 판권을 독점했다는 것이었다. 해커는 소유권이 프로그램을 더 좋게 향상시키려는 그를 가로막고 있다는 사실에 화가 났다. 그는 제록스사가 프로그램을 독점했고 황금률을 위반했다고 말했다. 그러나 제록스는 혼자가 아니었다. 소프트웨어 산업이 점점 커지면서 실리콘 밸리는 수많은 인공지능연구소의 인재들을 빼가기 시작했고 이 프로그래머들은 소프트웨어 회사에서 일할 때 기밀 유지 협약에 서명하도록 강요받았다. 이제 코드는 저작권이 생기며 더 이상 공유될 수도 없고 향상될 수도 없는 것이라는 사실을 그 해커는 알게 되었다. 그리고 저작권이 프로그래밍 사회를 파괴하고 있다고 결론지었다. 이 해커는 바로 처음으로 자유소프트웨어 운동을 시작한 리처드 스톨만이다.

“소프트웨어를 판매하는 사람들은 사용자를 각각 구분하고, 그들 위에 군림하고, 사용자 서로가 프로그램을 공유하는 것을 막고자 한다. 나는 이런 식으로 사용자 간의 결속이 깨지는 것을 거부한다. 나는 올바른 양심으로 비공개 협정이나 소프트웨어 사용권 계약에 서명할 수 없다. 여러 해 동안 인공지능 연구소에서 일하면서 이러한 경향과 다른 박정한 일들에 저항해 보았지만 결국 그들의 승리로 끝나고 말았다. 내 의지에 역행하는 이런 일들이 일어나는 연구소에 나는 더 이상 머무를 수가 없었다.”(GNU Manifesto)

2) 자유 소프트웨어 운동과 개념의 확립

1984년 리처드 스톨만은 자유 소프트웨어 재단을 창설했다. 이 재단은 소프트웨어의 본래 생산 유통 방식인 정보 공유의 방식을 복원하기 위해 자유 소프트웨어 운동을 시작하였다. 이는 소스 코드 공개를 통해 누구나 소프트웨어를 수정할 수 있게 하며, 자유로운 복제와 배포를 허용하는 것이다. 또한 이 운동의 목표는 운영 체제만이 아닌 모든 소프트웨어를 자유소프트웨어로 만드는 것이며 이 프로젝트의 핵심 작업은 운영 체제를 만들어 여러 사람들의 손을 거쳐 더 완성도 높은 소프트웨어를 만드는 것이다. 이를 위한 선결 과제는 컴퓨터 네트워크에 이용되는 유닉스의 소스 코드를 사용하지 않으면서 유닉스와 같은 향상된 운영체제를 개발하는 것이었다. 유닉스는 1969년 벨연구소의 두 연구원에 의해서 개발되어 IBM이나 Compaq, 썬 등 12개의 다른 버전으로 사용되고 있었다. 스톨만은 Unix에 대항해 GNU 프로젝트를 통하여 유닉스 계열 컴퓨터의 새로운 운영체제를 개발하여 GNU라고 명명했는데 이는 'GNU is Not Unix'의 준말이다. GNU의 도전은 대단한 것이었다. 운영체제란 두 숫자를 더한다든지, 정보를 하드디스크에 옮긴다든지 하는 일에 필요한 프로그램들과 그러한 일을 직접적으로 하드웨어에 전달하는 방식을 말한다. 그러나 윈도우를 운영한다거나 프린터나 기타 장치들과 통신하는 등의 특정한 작업을 수행하는 수많은 보조 프로그램들이 없다면 그 운영체제는 무용지물이 되고 만다. 효과적인 시스템을 구축하기 위해서 GNU 프로젝트는 이러한 모든 프로그램들을 만들어 내야 했다. 페렌즈는 "이것은 창고에서 비행기를 만드는 일에 비유될 만한 것이다"고 말한다. 사람들은 이것이 불가능하다고 생각했다. 그러나 이 일을 스톨만은 해내었고 스톨만 보다 덜 뛰어난 사람이 담당했었더라면 불가능했었을 것이다. GNU 프로젝트는 자유 소프트웨어 운동에 대해 사용자가 자유 소프트웨어를 이용하여 자유롭게 컴퓨터를 사용할 수 있도록 하는 것이 목적이라고 설명하며 자유 소프트웨어를 사용하면, 사용자 스스로가 컴퓨터를 사용하는 환경을 구성할 수 있으며 독점 자유 소프트웨어는 소프트웨어 개발자에 의존해야 할 수밖에 없음을 강조한다. 자유 소프트웨어 운동은 소프트웨어를 협업적인 방식으로 공동으로 개발하는 것을 넘어서 더 나은 사회를 만들기 위한 도덕적 활동이다. 공유와 협동이라는 사회적 결속을 증가시키기 때문에 자유 소프트웨어 운동이 사회 전체를 위한 본질적 가치라고 설명한다. 따라서 자유 소프트웨어의 4가지 요건에 대한 다음과 같이 설명한다.

\* 프로그램을 원하는 어떠한 목적으로도 실행할 수 있는 자유

\* 프로그램이 어떻게 동작하는지 학습하고, 자신의 필요에 맞게 개작할 수 있는 자유. 이것을 위해서는 소스 코드에 대한 접근이 전제되어야 한다.

\* 이웃을 도울 수 있도록 복제물을 재배포할 수 있는 자유.

\* 프로그램을 개선시킬 수 있는 자유와 개선된 이점을 공동체 전체가 누릴 수 있게 그것을 발표할 자유. 이를 위해서는 역시 소스 코드에 대한 접근이 전제되어야 한다.

리처드 스톨만은 자유 소프트웨어 운동에 대해 정식화하고 인증하기 위해 GNU 일반 공중 사용허가서와 카피레프트 개념을 제시하였다. 일반 공중 사용허가서는 모든 사람에게 프로그램을 실행하고 복사하고 수정하고 수정된 버전을 배포할 수 있는 권리는 부여하지만 자신만의 규제를 더할 수 있는 권리는 부여하지 않는다고 설명한다. 스톨만의 모든 코드는 카피레프트 되었다. 다른 사람들이 또 다른 사람들로 하여금 그들의 수정에 똑같은 수정을 가할 수 있도록 한다면, 사용자들은 누구나 자유롭게 소프트웨어를 바꿀 수 있었다. 스톨만은 카피레프트가 그들 자신의 도구를 이용하여 소프트웨어 독점자들에게 대항하는 방법이라고 말한다. GNU 운영체제를 위한 텍스트 편집기와 컴파일러를 직접 작성했다. 하지만 운영체제의 중심 모듈로서 소프트웨어 프로그램의 요청을 처리하고 요청을 컴퓨터 CPU가 이해할 수 있는 명령으로 변환하는 역할을 하는 커널(Kernel:운영체제의 핵심적인 부분)을 완성하는데 많은 어려움을 겪었다. 범용으로 사용 가능한 여러 가지 프로그램들을 만들어 내긴 했지만 GNU 운영체제의 커널을 만들어내지 못했다. 그 이유 중의 하나는 스톨만이 유닉스의 커널을 복제하지 않고 GNU 시스템의 기초를 카네기 멜런대학에서 개발한 진보적이고 경험적인 커널로 선택한 것이다. 스톨만은 새로운 커널을 개발한 몇 안 되는 사람 중의 한명이었고 이 일을 할 수 있다고 생각하는 거의 유일한 사람이었다. 그러나 혼자서 많은 코드를 타이핑하는 것이 애초부터 무리였고 결국 포기하게 되었다. 몇 년간의 고통은 키보드와 씨름하기 힘들게 만들었고 커널을 개발하기 위한 작업은 중단되었다. 스톨만은 MIT 학생들을 고용해서 계속해 나가려고 노력했지만 얼마안가 기계적으로 컴퓨터 코드를 번역하는데 염증을 느껴 그만두게 되었다.

3) Linux의 등장

1991년 헬싱키 대학의 21살 대학생 토발즈는 결코 뛰어난 프로그래머가 아니었다. "난 내가 모르는 것에 대해서는 전혀 알지 못한다." 그러나 그는 유닉스가, 마치 질질 새는 펜으로 쓰기를 강요하는 것과 같은 MS의 MS-DOS 운영체제보다 충분히 우위에 있다는 것을 잘 알고 있었다. 토발즈는 프로그래밍을 원했지만 수없이 많은 명령어들로 괴로워해야 했다. 또 그의 4메가의 메모리를 갖춘 386급의 컴퓨터는 유닉스를 돌리기에 역부족이었다. 그러나 그는 형편없는 소프트웨어를 받아들이는 것을 거부했다. DOS를 무시하고 그의 선생님들이 준 코드 뭉치들을 잘게 쪼게 가며 그의 숙제들을 해 나갔다. 예상치 못하게 토발즈는 유닉스의 커널과 비슷한 무언가를 얻어냈다. GNU 프로젝트가 필수적인 프로그램들을 만들어 놓았기 때문에 그는 커널을 그것들에 부합하도록 변형했다. 이게 어찌된 일인가? 그는 완전한 운영체제를 만든 것이다. 이것은 유닉스의 유연함과 안정성을 가졌고, 성능이 그리 좋지 않은 기종의 컴퓨터에서도 아주 유용했다. 그는 그의 운영체제를 Freax 라고 불렀다. 그 이름이 세련되지 못하다고 생각한 그의 친구들이 그 운영체제의 이름을 바꿔 부르기를 Linux라 하였다.

**--------------------------------------석형원---------------------------------------**

**--------------------------------------**김재현**----------------------------------------**

오픈소스운동

오픈소스 운동이란 몇개의, 또는 모든 소프트웨어에 대해 오픈소스를 지원하게 하는 운동을 말한다. 오픈소스 운동은 오픈소스 소프트웨어의 개념과 생각을 널리 퍼트렸다. 오픈소스 운동을 지지하는 프로그래머들은 소프트웨어 개발에 있어서 프로그램 코드를 자발적으로 쓰고 고침으로써 오픈소스 커뮤니티에 기여했다. 오픈소스는 이미 편집된 작업을 편집하는 것에 있어서 누구도 방해할 수 없다는 것 또는 누구도 작성된 코드를 차별없이 공유하는 것을 필요로 한다. 이러한 소프트웨어 개발에 대한 접근 방식은 모두가 오픈소스 코드를 수정할 수 있게 한다. 이렇게 수정한 것은 오픈소스 커뮤니티의 개발자들에게 배포된다. 이러한 방식으로 인해 코드에 코드 수정에 참여한 모든 사람들의 개성이 나타나고 코드의 변화가 시시각각 작성된다. 이 방식은 코드의 소유권을 확실히 하기 어렵지만 오픈소스 운동의 개념에 알맞다.

오픈 소스 소프트웨어 운동의 기원은 1980년대 초반 소프트웨어의 상용화 및 그에 따른 여러 가지 제한에 반대하여 리차드 스톨만(Richard Stallman)이 자유소프트웨어 재단(Free Software Foundation)을 만들고 자유 소프트웨어 운동을 시작하면서부터이다. 그 이후 1990년대 들어서면서 인터넷과 더불어 리눅스가 성공하기 시작하였으며, 이를 토대로 상용 소프트웨어 업체들의 참여를 이끌기 위해 1998년 Open Source Initiative(OSI)가 만들어졌다. 때를 같이 하여 MS의 익스플로러에 밀려 어려움을 겪고 있던 넷스케이프사가 웹브라우저의 소스 코드를 공개하는 결정을 내리게 되었으며, Sun, IBM 등이 오픈 소스 소프트웨어에 대한 지원을 시작하였다. 최근 들어서는 아시아와 유럽을 중심으로 각국 정부에서 미국 중심의 소프트웨어 산업을 극복하고 자국의 소프트웨어 기술 수준 향상 및 산업을 성장시키기 위해 오픈 소스 소프트웨어에 대한 지원정책을 내놓고 있다.

**Launch of the free software movement**

*Main article: free software movement*

In 1983, Richard Stallman launched the GNU Project to write a complete operating system free from constraints on use of its source code. Particular incidents that motivated this include a case where an annoying printer couldn't be fixed because the source code was withheld from users. Stallman also published the GNU Manifestoin 1985 to outline the GNU Project's purpose and explain the importance of free software. Another probable inspiration for the GNU project and its manifesto was a disagreement between Stallman and Symbolics, Inc. over MIT's access to updates Symbolics had made to its Lisp machine, which was based on MIT code.[17] Soon after the launch, he used the existing term "free software" and founded the Free Software Foundation to promote the concept. *The Free Software Definition* was published in February 1986.

In 1989, the first version of the GNU General Public License was published. A slightly updated version 2 was published in 1991. In 1989, some GNU developers formed the company Cygnus Solutions.The GNU project's kernel, later called "GNU Hurd", was continually delayed, but most other components were completed by 1991. Some of these, especially the GNU Compiler Collection, had become market leaders in their own right. The GNU Debugger and GNU Emacs were also notable successes.

오픈소스소프트웨어개념

소스 코드가 공개(Open)된 프로그램이다. 대부분의 오픈 소스 소프트웨어는 무료로 사용 가능하기 때문에 프리웨어와 헷갈리는 경우가 많지만, 프리웨어는 무료로 사용 가능한 프로그램이고, 오픈 소스는 소스 코드가 공개된 프로그램이기 때문에 엄연히 다른 개념이다(예를 들어, 오픈 소스 소프트웨어를 돈 받고 파는 경우도 있다.). 자유 소프트웨어(Free Software)와 비슷하지만, 오픈 소스 소프트웨어가 자유 소프트웨어보다 조금 더 상위 개념이다.  
  
일반 사용자 입장에서는 프리웨어나 오픈 소스 소프트웨어나 단순히 공짜로 사용할 수 있다는 점에서는 비슷할 수 있지만, 소스코드를 보고 이해할 수 있고, 수정할 수 있는 개발자 입장에서는 크게 다르다. 예를 들어, 상용 또는 프리웨어 프로그램을 사용하는 사람들은 버그를 발견했다 하더라도 소스 코드를 모르니 수정할 수 없고, 사용자가 새로운 아이디어가 떠올랐다 해도 그것을 곧바로 프로그램에 적용시킬 수도 없다. 비교적 간단한 프로그램은 리버스 엔지니어링으로 어셈블러 수준에서 뜯어고칠 수는 있으나 코드가 공개된 것보다 몇 백 배는 어렵기도 하고, 저작권 같은 문제가 얽히고설키기에 하려는 사람은 없다고 보면 된다.  
  
하지만 사용자가 프로그래밍 언어를 아는 경우 소스가 공개되어 있다면 본인이 직접 소프트웨어의 문제를 수정하거나 개선을 할 수 있게 되는 것이다. 또한, 개발하던 프리웨어가 개인적인 사정이나 회사의 사정에 따라 개발이 중지되면 그대로 사장되는 경우가 종종 있는데, 오픈 소스 소프트웨어는 소스가 공개되어 있기 때문에 다른 개발자/개발사에서 이를 이어 받아서 새로이 개선해 나가면서 개발하는 것이 된다. 그래서 개발자와 사용자가 일치하는 개발 및 시스템, 네트워크 분야에는 웬만한 클로즈드 소스 상용 소프트웨어는 명함도 못 내밀 정도로 고품질의 오픈 소스 소프트웨어가 넘쳐난다. 그러나 그러지 않는 분야에선 말 그대로 취미 수준에 머물러 있는 경우도 많다.

오픈소스개념  
 어떤 제품을 개발하는 과정에 필요한 소스 코드나 설계도를 누구나 접근 해서 열람할 수 있도록 공개하는 것. 보통 소스가 공개된 소프트웨어를 오픈 소스 소프트웨어라고 하며, 소프트웨어 말고도 개발 과정이나 설계도가 공개되는 경우 하드웨어에도 오픈 소스 모델이 적용 가능하며, 글꼴과 같은 데이터에도 오픈 소스 개발 모델이 적용되는 경우가 있다.  
  
단순히 소스를 공개만 하는 것이 아니라, 이를 2차 창작하는 것을 허용하기도 하고, 나아가 조건 없이 상업적 용도로도 사용할 수 있게 하는 경우가 있다.

**Open-source software (OSS)** is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose. Open-source software may be developed in a collaborative public manner. According to scientists who studied it, open-source software is a prominent example of open collaboration. The term is often written without a hyphen as "open source software".

Open-source software development, or collaborative development from multiple independent sources, generates an increasingly more diverse scope of design perspective than any one company is capable of developing and sustaining long term. A 2008 report by the Standish Group states that adoption of open-source software models has resulted in savings of about $60 billion (£48 billion) per year to consumers.

**The launch of Open Source**

In 1997, Eric S. Raymond published The Cathedral and the Bazaar, a reflective analysis of the hacker community and free software principles. The paper received significant attention in early 1998 and was one factor in motivating Netscape Communications Corporation to release their popular Netscape Communicator Internet suite as free software. This code is today the basis for Mozilla Firefox and Mozilla Thunderbird.

Netscape's act prompted Raymond and others to look into how to bring free software principles and benefits to the commercial-software industry. They concluded that FSF's social activism was not appealing to companies like Netscape, and looked for a way to rebrand the free software movement to emphasize the business potential of the sharing of source code.

The label "open source" was adopted by some people in the free software movement at a strategy session held at Palo Alto, California, in reaction to Netscape's January 1998 announcement of a source code release for Navigator. The group of individuals at the session included Christine Peterson who suggested "open source", Todd Anderson, Larry Augustin, Jon Hall, Sam Ockman, Michael Tiemann, and Eric S. Raymond. Over the next week, Raymond and others worked on spreading the word. Linus Torvalds gave an all-important sanction the following day. Phil Hughes offered a pulpit in Linux Journal. Richard Stallman, pioneer of the free software movement, flirted with adopting the term, but changed his mind. Those people who adopted the term used the opportunity before the release of Navigator's source code to free themselves of the ideological and confrontational connotations of the term "free software". Netscape released its source code under the Netscape Public License and later under the Mozilla Public License.

The term was given a big boost at an event organized in April 1998 by technology publisher Tim O'Reilly. Originally titled the "Freeware Summit" and later named the "Open Source Summit", The event brought together the leaders of many of the most important free and open source projects, including Linus Torvalds, Larry Wall, Brian Behlendorf, Eric Allman, Guido van Rossum, Michael Tiemann, Paul Vixie, Jamie Zawinski of Netscape, and Eric Raymond. At that meeting, the confusion caused by the name free software was brought up. Tiemann argued for "sourceware" as a new term, while Raymond argued for "open source". The assembled developers took a vote, and the winner was announced at a press conference that evening. Five days later, Raymond made the first public call to the free software community to adopt the new term. The Open Source Initiative was formed shortly thereafter. According to the OSI Richard Stallman initially flirted with the idea of adopting the open source term. But as the enormous success of the open source term buried Stallman's free software term and his message on social values and computer users' freedom, later Stallman and his FSF strongly objected the OSI's approach and terminology. Due to the rejection of Stallman of the term open source software, the FOSS ecosystem is being divided in its terminology, see also Alternative terms for free software. For example, a 2002 FOSS developer survey revealed that 32.6% associate themselves with OSS, 48% with free software, and 19.4% in between or undecided. Stallman still maintained, however, that users of each term were allies in the fight against proprietary software.

On 13 October 2000, Sun Microsystems released the StarOffice office suite as free software under the GNU Lesser General Public License. The free software version was renamed OpenOffice.org, and coexisted with StarOffice.

On the end of 1990s the term "Open source" gained much traction in public media and acceptance in software industry in context of the dotcom bubble and the open-source software driven Web 2.0.

**Java**

Since its first public release in 1996, the Java platform had not been open source, although the Java source code portion of the Java runtime was included in Java Development Kits (JDKs), on a purportedly "confidential" basis, despite it being freely downloadable by the general public in most countries. Sun later expanded this "confidential" source code access to include the full source code of the Java Runtime Environment via a separate program which was open to members of the public, and later made the source of the Java compiler javac available also. Sun also made the JDK source code available confidentially to the Blackdown Java project, which was a collection of volunteers who ported early versions of the JDK to Linux, or improved on Sun's Linux ports of the JDK. However, none of this was open source, because modification and redistribution without Sun's permission were forbidden in all cases. Sun stated at the time that they were concerned about preventing forking of the Java platform.

However, several independent partial reimplementations of the Java platform had been created, many of them by the open source community, such as the GNU Compiler for Java (GCJ). Sun never filed lawsuits against any of the open source clone projects. GCJ notably caused a bad user experience for Java on free software supporting distributions such as Fedora and Ubuntu which shipped GCJ at the time as their Java implementation. How to replace GCJ with the Sun JDK was a frequently asked question by users, because GCJ was an incomplete implementation, incompatible and buggy.

In 2006 Jonathan I. Schwartz became CEO of Sun Microsystems, and signalled his commitment to open source. On 8 May 2007, Sun Microsystems released the Java Development Kit as OpenJDK under the GNU General Public License. Part of the class library (4%) could not be released as open source due to them being licensed from other parties and were included as binary plugs.[citation needed] Because of this, in June 2007, Red Hat launched IcedTea to resolve the encumbered components with the equivalents from GNU Classpath implementation. Since the release, most of the encumbrances have been solved, leaving only the audio engine code and colour management system (the latter is to be resolved using LittleCMS).

https://en.wikipedia.org/wiki/History\_of\_free\_and\_open-source\_software#Java

https://en.wikipedia.org/wiki/Open-source\_software#History

**--------------------------------------**김재현**----------------------------------------**

————————————————————현준—————————————————————

2-4

앞에서 오픈 소스에 대해 소개했으나, 아직 먼 나라 이야기처럼 들릴 수 있다. 하지만 오픈 소스는 프로그래머라면 반드시 다뤄야 할 소재이며 절대 어렵지 않다는 것을 알려주고 싶다. 처음으로 오픈 소스 활동에 참여하고자 하는 스스로에게 3가지 질문을 해 보자.

1. Github에 가입할 줄 아는가?

2. Git의 사용법을 아는가?

3. 나의 관심 분야는 무엇인가?

물론 당신은 스스로가 Github에 가입할 수 있다고 믿어 의심치 않을 것이다. 하지만 Git의 사용법도 알고 있을까? 이 책에서 Git의 사용법은 소개하지 않겠지만, 인터넷에 검색하면 이와 관련하여 많은 양의 자료들이 있으므로 충분히 사용법을 익힐 수 있을 것이다. Git의 사용법까지 익혔다면 이제 자신의 관심 분야에 대해 생각해 보아야 한다. 관심이 있는 분야가 오랜 시간 작업하기에 좋고 익히기 쉬울 것이다. 관심이 있는 프로젝트가 있다면 그 프로젝트의 이슈 게시판에 가보자. 그러면 초심자에게 걸맞은 쉬운 과제들이 등록되어 있을텐데, 이것부터 시작해서 점차 자신의 영역을 넓혀가면 된다. 조급하게 행동하지 않아도 괜찮다. 오픈 소스는 먼 미래에도 당신의 옆에 있을 것이다.

그러면 앞으로 동반자가 될 오픈 소스를 어떤 태도로 대해야 하는가? 오픈 소스는 개발자 채용에도 중요한 요소가 되었고 앞으로도 중요할 것이다. 오픈 소스 활동을 소홀히 한다면 당연히 취업과 멀어지겠지만, 너무 오픈 소스 활동에만 치중하다 보면 기본기에 소홀해 질 수 있다. 개발자에겐

————————————————————현준—————————————————————