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.