

서문

리눅스는 인터넷의 한 현상이다. 리눅스는 한 학생의 취미 프로젝트로 시작해서, 이제는 무 료로 얻을 수 있는 다른 어떤 운영체제보다도 대중적으로 성장했다. 많은 사람들에게 리눅 스는 수수께끼이다. 어떻게 공짜인데도 쓸만한 것일 수 있을까? 몇 안되는 대규모 소프트웨 어 회사들이 판치는 세상에서, 어떻게 한 때의 "해커들"¹ 이 만든 소프트웨어가 이들과 함께 경쟁할 수 있을까? 전세계의 서로 다른 나라의 서로 다른 사람들이 기여한 소프트웨어가 어 떻게 안정적이면서 동시에 효율적일 수 있을까? 하지만 리눅스는 안정적이고 효율적인 동시 에 경쟁까지 펼치고 있다. 많은 대학과 연구기관에서 일 상적인 컴퓨터 업무에 리눅스를 사 용하고 있다. 집에 있는 컴퓨터에서 리눅스를 사용하는 사람도 있 고, 대부분의 회사들도 리 눅스를 사용하고 있다는 것을 실감하지는 못할지라도 어디선가 틀림없이 리눅스를 사용하고 있다. 그들은 리눅스를 웹사이트를 보고, 웹사이트를 구축하고, 이메일을 보내고, 늘 하듯이 게임을 하는데 사용한다. 리눅스는 결단코 장난감이 아니다. 리눅스는 전세계의 애호가들 이 사용하고 있는, 완전히 개발되고 전문적으로 만들어진 운영체제이다.

리눅스의 기원은 UnixTM의 시작까지 거슬러 올라간다. 1969년 벨 연구소(Bell Laboratories) 연구 그룹 (Research Group)의 일원인 켄 톰슨(Ken Thompson)은 놀고 있는 PDP-7을 이용하여 멀티 유저, 멀티 태스킹 운영체제를 실험하기 시작했다. 곧 데니스 리치(Dennis Richie)가 합류하였 고, 둘은 연구 그룹 의 다른 사람들과 함께 유닉스의 초기 버전을 만들었다. 리치는 이전에 수행한 MULTICS라는 프로젝 트에 크게 영향을 받았는데, 유닉스라는 이름 자체가 MULTICS에 빗대어 지은 것이다. 초기 버전은 어 썬블리 코드로 작성하였으며, 세번째 버전 은 새로운 프로그래밍 언어인 C로 다시 작성하였다. 리치 는 운영체제를 만드는데 사용할 목 적으로 C 프로그래밍 언어를 설계하고 만들었다. C로 고쳐 썼기 때 문에 디지털(Digital)의 더 욱 강력한 PDP-11/45와 11/70 컴퓨터로 유닉스를 옮길 수 있었다. 그 이후 는 흔히 하는 말로 역사가 되었다. 유닉스는 연구소에서 나와 컴퓨터 세계의 주류에 편입되었고, 곧 대부분의 주요 컴퓨터 생산업체들은 자신들의 유닉스 버전을 만들어 내놓았다.

리눅스는 단순한 욕구를 만족시켜준 해답이었다. 리눅스의 작성자이자 주관리자인 리누스 토발즈 (Linus Torvalds)가 갖고 놀만한 유일한 소프트웨어는 미닉스(Minix)였다. 미닉스는 유닉 스와 비슷한, 간단한 운영체제로서 교육목적으로 널리 쓰이고 있었다. 리누스는 미닉스의 기 능에 만족하지 못했 고, 자기나름의 소프트웨어를 만들어 이를 해결하려고 했다. 그는 학창시 절 때 익숙한 운영체제인 유 닉스를 모델로 삼고, 인텔 386 PC에서 프로그램을 만들기 시작 했다. 작업은 매우 빠르게 진척되었고, 이에 고무 받은 리누스는 자신의 노력의 결과물을 당 시 막 등장하던 전세계적인 컴퓨터 네트워크를 통 하여 다른 학생들에게 제공하였다. 그리하여 리눅스는 대학 사회에서 주로 쓰이게 되었다. 소프트웨 어를 보고 이에 공헌하는 사람이 나 타나기 시작했다. 그들은 자신이 이전에 가졌던 문제점들을 해결 하는데 사용했던 방법들을 새로운 소프트웨어에 적용했다. 오래지 않아 리눅스는 운영체제의 모습을 갖추게 되었다. 중 요한 점은 리눅스가 유닉스 코드를 단 한 줄도 가지고 있지 않다는 사실이다. 리눅 스는 공 표된 POSIX 표준에 따르는 완전히 새로 짜여진 것이다. 리눅스는 메사추세츠주 캄브리지에 있는 무료 소프트웨어 재단(Free Software Foundation)에서 만든 GNU (GNU's Not UnixTM) 소프 트 웨어로 만들어지고, 많은 GNU 소프트웨어들을 사용한다.

대부분의 사람들은 단지 여러 종류의 잘 만들어진 CD-ROM 배포판 중 하나를 설치하고, 리 눅스를 간 단한 도구로서 사용한다. 상당수의 리눅스 사용자들은 프로그램을 작성하거나, 다 른 사람이 만든 프 로그램을 실행하는데 리눅스를 사용한다. 많은 리눅스 사용자들은 열심히 HOWTO² 문서를 읽고, 시 스템의 한 부분을 제대로 설정하였을 때는 성공의 전율감을 느끼 지만, 동시에 제대로 동작하지 않는

경우 실패의 좌절감을 맛보기도 한다. 소수의 사용자는 디바이스 드라이버를 만들고, 커널을 수정하여 리눅스 커널의 제작자이면서 관리자인 리눅스 토발즈에게 보낼 정도로 열성이다. 리눅스는 어디서든, 누구한테서든 커널 소스에 대한 추가나 수정을 받아들인다. 이것은 얼핏 무정부주의적인 방법처럼 들릴 수도 있겠지만, 리눅스는 새 코드들을 엄격하게 검사하고 모든 코드를 자신이 직접 커널에 추가한다. 그렇긴 하지만, 실제로 어느 한 시점에서 리눅스 커널 소스 작업에 참여하는 사람들의 수는 손에 꼽을 수 있는 정도에 불과하다.

대부분의 리눅스 사용자들은 운영체제가 어떻게 동작하는지, 각 부분들이 어떻게 맞물려 돌아가는지 눈여겨 보지 않는다. 그러나 리눅스를 자세히 살펴보는 것은 운영체제의 동작 원리를 배우는 훌륭한 방법임을 생각해 본다면 이는 부끄러운 일이다. 리눅스 소스는 매우 잘 만들어졌을 뿐만 아니라 그 누구든지 자유롭게 살펴볼 수 있도록 완전히 공개되어 있다. 이것은 작성자들이 소프트웨어에 대해 저작권을 소유하고는 있지만, 무료 소프트웨어 재단의 GNU 공개 라이선스에 따라 그 소스 코드를 자유롭게 배포할 수 있도록 하기 때문이다. 그 령지만 막상 소스 코드를 처음 마주칠 때는 이들이 뭔지 무척 혼동스러울 것이다. kernel, mm, net 등의 디렉토리가 눈에 띄겠지만, 이 안에는 어떤 코드가 담겨 있으며 그 코드들은 어떻게 동작할 것인가? 여기서 필요한 것은 리눅스의 전반적인 구조와 목적에 대해 폭넓게 이해하는 것이다. 이것은 간단히 말하자면 바로 이 책의 목적이기도 하다. 즉 리눅스라는 운영체제가 어떻게 동작하는지 명확히 이해할 수 있도록 도와주는 것이다. 다시 말하면 파일을 복사하거나 이메일을 읽을 때, 리눅스 시스템 내부에서 어떤 일들이 일어나는 가를 머릿 속에서 그려볼 수 있도록 하려는 것이다. 필자는 운영체제가 실제로 어떻게 동작하는지 처음 깨달았을 때 느꼈던 흥분을 아직도 잘 기억하고 있다. 그 흥분이 이 책을 읽는 독자들에게 전해지길 바란다.

내가 처음으로 리눅스와 관계를 맺은 것은 1994년 후반, 알파 AXP 프로세서 기반 시스템으로 리눅스 포팅 작업을 하던 짐 파라디스(Jim Paradis)를 방문했을 때다. 나는 1984년부터 디지털 이큅먼트(Digital Equipment Co, Ltd, 줄여서 DEC 또는 디지털이라고 함)에서 주로 네트 워크 및 통신관련 분야에서 일해 왔었고, 1992년에는 새로 구성된 디지털 반도체 부서에서 업무를 맡게 되었다. 그 부서의 목표는 상용 반도체칩 공급 시장에 전면적으로 진입하여 칩을 판매하는 것으로, 구체적으로는 알파 AXP 계통의 마이크로프로세서와, DEC 외부에서 설계한 알파 AXP 시스템 보드까지도 취급, 판매하는 것이었다. 처음 리눅스에 대해 듣고, 나는 금방 리눅스에 흥미를 갖게 되었다. 나는 짐의 열정에 끌렸고, 포팅 작업을 돕기 시작했다. 일을 계속해 가면서 나는 리눅스 운영체제 자체뿐만 아니라 그것을 만들어 내는 엔지니어들의 공동체에 대해서도 점점 그 진가를 알게 되었다.

그러나 알파 AXP는 리눅스가 동작하는 수많은 하드웨어 플랫폼 가운데 하나에 지나지 않는다. 현재 리눅스 커널의 대부분은 인텔 프로세서 기반 시스템에서 돌고 있지만 비인텔 리눅스 시스템의 수도 점점 늘어나고 있는 추세다. 알파 AXP, ARM, MIPS, Sparc, 그리고 PowerPC 등이 그 예다. 나는 이 책을 쓰면서 이들 플랫폼 중에서 어떤 것이든 이용할 수 있었지만, 나의 리눅스에 대한 기반지식과 기술적인 경험은 주로 알파 AXP에서이고, ARM 에서 도는 리눅스에 대해 어느정도 알고 있는 것이다. 이 책에서 종종 키 포인트를 설명하는데 비인텔 하드웨어를 예로 드는 것은 이 때문이다. 리눅스 커널 소스의 95% 가량은 리눅스가 동작하는 모든 하드웨어에서 그대로 사용되고 있다. 마찬가지로 이 책의 95% 가량은 리눅스 커널 중에서 하드웨어와 무관한 부분에 대한 것이다.

독자의 자세

이 책은 독자가 가진 지식이나 경험에 대해 아무런 가정도 하지 않고 있다. 나는 독자가 주제로 다루는 문제에 관심을 가지고 있다면, 필요한 부분에 대해서 스스로 공부하게 되리라 믿는다. 컴퓨터에

친숙하고(아마도 PC이겠지만), C 프로그래밍 언어에 대해 어느정도 지식을 가진다면, 독자 여러분이 이 글에서 실질적인 이득을 얻는데 도움이 될 것이다.

이 책의 구조

이 책은 리눅스의 내부구조에 대한 매뉴얼로 만든 것이 아니다. 그 대신 일반적인 운영체제, 특히 리눅스에 대한 소개서가 될 것이다. 각 장(章)은 "일반적인 것에서 시작하여 특수한 것으로 작업하는" 나의 규칙을 따르고 있다. 먼저 그 장에서 설명하려는 커널 서브시스템의 개요를 제시하고 나서, 다음으로 끔찍한 상세내용들을 다룬다.

나는 일부러 커널의 작동방식인 알고리즘을, 함수 $X()$ 가 어떤 자료구조의 어떤 항목의 값을 증가시키는 함수 $Y()$ 를 부른다는 식으로 설명하지 않았다. 이런 것들은 코드를 읽어보면 알 수 있는 것이다. 나는 어떤 코드를 이해하거나 다른 사람에게 그것을 설명해야 할 때마다, 종종 칠판에 자료구조를 그리는 일부터 시작하곤 했다. 마찬가지로, 나는 여러개의 서로 관련된 커널 자료구조와 그들의 상관관계를 아주 자세하게 설명했다.

각 장에서 다루는 리눅스 커널 서브시스템이 그렇듯, 이들을 다루는 각 장은 아주 독립적이다. 하지만 이따금 몇몇 장은 연관되어 있다. 예를 들면, 가상 메모리가 어떻게 동작하는지 이해하지 못한 채 프로세스를 설명할 수는 없을 것이다.

1장 "하드웨어의 기초" 장에서는 요즘의 PC에 대해 간략히 소개한다. 운영체제는 자신의 토대가 되는 하드웨어 시스템과 긴밀하게 연결되어 동작해야 하며, 하드웨어만 제공할 수 있는 몇몇 서비스들을 필요로 한다. 리눅스 운영체제를 완전히 이해하려면 관련된 하드웨어의 기본적인 것들을 이해해야 한다.

2장 "소프트웨어의 기초" 장에서는 기본적인 소프트웨어 원칙을 소개하고, 어셈블리와 C 프로그래밍 언어를 살펴본다. 이 장에서 리눅스와 같은 운영체제를 만드는데 사용하는 툴을 살펴보고, 운영체제의 목적과 기능들을 간략히 소개한다.

3장 "메모리 관리" 장에서는 리눅스에서 시스템상의 실제 메모리와 가상 메모리를 관리하는 방법을 설명한다.

4장 "프로세스"장에서는 프로세스란 무엇이며, 리눅스 커널이 어떻게 프로세스를 생성하고 관리하며 삭제하는가에 대하여 설명한다.

프로세스는 그들의 활동을 통합하기 위해 프로세스 사이에, 그리고 커널과 통신한다. 리눅스는 여러 종류의 프로세스간 통신(Inter-Process Communication, IPC) 구조를 지원한다. 시그널과 파이프는 이들 중 일부이며, System V IPC(이것이 처음 등장한 유닉스 버전의 이름을 따라 붙여진 이름이다) 메커니즘 역시 지원한다. 이러한 프로세스간 통신 메커니즘은 5장에서 설명하고 있다.

PCI(Peripheral Component Interconnect) 표준은 이제 저비용 고성능 PC용 데이터 버스로서 확고하게 자리잡았다. 6장 "PCI"장에서는 리눅스 커널이 시스템 내의 PCI 버스들과 장치들을 초기화하고 사용하는 방법을 설명한다.

7장 "인터럽트와 인터럽트 처리" 장에서는 리눅스 커널이 어떻게 인터럽트를 다루는지 살펴본다. 커널이 인터럽트를 처리하는 데에는 일반적인 메커니즘과 인터페이스가 있지만, 세부적인 인터럽트 처

리는 하드웨어와 아키텍처에 따라 다르다.

리눅스의 강점중의 하나는 요즘 PC에서 사용할 수 있는 많은 하드웨어 장치들을 지원한다 는 것이다. 8장 "디바이스 드라이버" 장에서는 리눅스 커널이 시스템에 있는 물리적인 장치 를 제어하는 방법을 설명한다.

9장 "파일 시스템" 장에서는 리눅스 커널이 어떻게 파일 시스템 내의 파일들을 다루는지 설 명한다. 또한, 가상 파일 시스템(Virtual File System, VFS)과 리눅스 커널의 실제 파일 시스템 지원 방법도 설 명한다.

네트워킹과 리눅스는 거의 같은 의미를 가지는 단어이다. 실제로 리눅스는 인터넷, 즉 월드 와이드 웹(World Wide Web, WWW)의 산물이다. 리눅스 개발자들과 사용자들은 웹을 이용하 여 정보와 아이 디어, 코드를 교환하며, 리눅스 자체는 종종 단체들의 네트워킹에 대한 요구 를 지원하기 위해 사용된 다. 10장 "네트워크" 장에서는 어떻게 리눅스가 알려진 네트워크 프 로토콜을 지원하는지 TCP/IP로 총괄하여 설명한다.

11장 "커널 메커니즘" 장에서는, 커널의 여러 부분들이 효율적으로 함께 동작할 수 있도록 리눅스 커 널이 제공하는 몇가지 일반적인 작업과 메커니즘에 대해 살펴본다.

12장 "모듈" 장에서는 리눅스가 어떻게 파일 시스템같은 기능요소들을 동적으로, 필요로 할 때에만 로드할 수 있는지 설명한다.

13장 "프로세서" 장은 리눅스가 포팅되어 있는 여러 프로세서들에 대한 간략한 소개글을 담 고 있다.

14장 "소스" 장은 커널의 특정 기능에 대해 알고자 할 때, 리눅스 커널 소스 코드 어느곳부 터 시작해 야 하는지 설명한다.

본서의 표기법

이 책에서 사용한 활자체 표기법은 다음과 같다.

serif 글꼴 독자가 직접 그대로 입력해야 하는 명령어 또는 문장을 의미한다.

type 글꼴 자료구조나 자료구조내의 항목을 가리킨다.

책 전체에 걸쳐서 리눅스 커널 소스에 있는 코드에 대한 참조 표시가 있다 (그 예로, 본 문 장에 인접 해 있는 테두리가 있는 글상자). 이는 독자가 소스 코드 자체를 살펴보고자 할 경 우를 위해서이며, 참 조하는 모든 파일은 /usr/src/linux 디렉토리를 기준으로 한 상대위 치이다. 예를 들어 foo/bar.c 파일 의 경우 완전한 파일명은 /usr/src/linux/foo/bar.c가 될 것이다. 현재 리눅스를 실행중이라면 (당연히 그래야 하겠지만) 소스 코드를 들여다보는 것은 가치있는 경험이며, 독자는 이 책을 소스 코드의 이해 를 돕고 또 여러 자료구조의 의미를 파악하는 데 유용한 지침서로 사용할 수 있을 것이 다.

등록상표

ARM은 ARM Holdings PLC의 등록상표이다.

Caldera, OpenLinux, 그리고 "C" 로고는 Caldera, Inc.의 등록상표이다.

Caldera OpenDOS 1997 Caldera, Inc.

DEC는 Digital Equipment Corporation의 등록상표이다.

DIGITAL은 Digital Equipment Corporation의 등록상표이다.

Linux는 Linus Torvalds의 등록상표이다.

Motif는 The Open System Foundation, Inc.의 등록상표이다.

MSDOS는 Microsoft Corporation의 등록상표이다.

Red Hat, glint, 그리고 Red Hat 로고는 Red Hat Software, Inc.의 등록상표이다.

UNIX는 X/Open의 등록상표이다.

XFree86은 XFree86 Project, Inc.의 등록상표이다.

X Window System은 X Consortium과 Massachusetts Institute of Technology의 등록상표이다.

저자 소개

나는 1957년, 스푸트니크호가 발사되기 몇 주전 영국 북부에서 태어났다. 나는 대학에서 유닉스를 처음 접했다. 거기서 강사는 유닉스를 커널과 스케줄링 및 다른 운영체제의 개념들을 가르칠 때 예로 사용했다. 나는 졸업년도 프로젝트를 위해 도입된 PDP-11 시스템을 즐겨 사용했다. 1982년 컴퓨터과 학과를 최우등으로 졸업한 뒤, 나는 Prime Computers (Primos)에서 근무하였고, 2년 후 디지털 (Digital)로 옮겼다(VMS, Ultrix). 디지털에 재직하는 동안 다양한 업무를 맡아 일했는데, 마지막 5년간은 알파 및 StrongARM 평가용 보드를 설계하는 반도체 그룹에서 일했다. 1998년 ARM으로 옮겨 로우 레벨 펌웨어를 만들고 운영체제를 포팅하는 작은 엔지니어 그룹을 맡게 되었다. 내 아이들(에스터와 스티븐)은 아빠를 괴짜라고 부른다.

사람들이 종종 직장에서나 집에서 리눅스에 대한 질문을 던지는데, 나는 그저 고맙고 행복 할 뿐이다. 직업상으로 또 개인적으로 리눅스를 쓰면 쓸수록, 나는 점점 더 리눅스 광신도 (zealot)가 되어가고 있다. 독자들은 여기서 맹신도(bigot) 등이 아닌 광신도(zealot)라는 용어를 썼음을 눈여겨 보아주기 바란다. 필자는 '리눅스 광신도'를, 다른 운영체제들의 존재 역시 인식하고 있지만 안쓰는 편을 택한 열성분자로 정의한다. 윈도우즈 95를 쓰는 나의 아내 길 (Gill)이 언젠가 이렇게 말했다. "우리가 남편 운영체제니 아내 운영체제니 하는 말을 쓰게 될거라곤 짐작도 못했어요". 엔지니어인 내게 있어서 리눅스는 나의 요구에 완벽하게 맞아 떨어진다. 리눅스는 내가 집과 회사에서 같이 사용할 수 있는 유연하고 적용하기 쉬운 엔지니어링 도구이며 최고의 운영체제이다. 공짜로 사용할 수 있는 소프트웨어의 대부분은 리눅스에서 쉽게 컴파일되며, 때로는 미리 컴파일 된 실행파일을 다운로드 받거나 CDROM에서 설치할 수도 있다. 공짜로 C++이나 Perl 프로그래밍을 배우고, Java에 대해 공부하는데 사용할 수 있는 다른게 무엇이 있는가!

감사의 글

우선 시간을 내어 이 책에 대해 이메일로 주석을 달아 보내준 많은 분들에게 감사한다. 나는 새 판을 낼 때마다 이들 주석들을 모두 포함시키려고 해 왔으며, 이는 주석을 받는 것보다 더 행복한 작업이었다. 어쨌든 나의 새 이메일 주소를 기억해주면 고맙겠다.

많은 강사들이 이 책의 일부를 컴퓨터를 가르치는데 쓸 수 있는지 편지로 물어왔다. 이에 대한 나의 대답은 당연히 "예"이다. 이는 내가 특히 바랬던 이 책의 용도 중 하나이다. 그 수업받는 학생들 중에 또 다른 리눅스 토발즈가 앉아있을지 누가 알겠는가.

책 전반에 관해서 상세하게 검토해 준 존 릭비(John Rigby)와 마이클 바우어(Miachel Bauer)에 게 특별히 감사드린다. 쉬운 일은 아니었을텐데도 나의 질문에 참을성있게 대답을 해준 앨 런 콕스(Alan Cox)와 스티븐 트위디(Stephen Tweedie)에게도 감사드린다. 각 장을 좀더 즐겁게 하기 위해 래리 에윙(Larry Ewing)의 펭귄을 사용했다. 끝으로, 이 책을 리눅스 문서화 프로젝트(Linux Documentation Project, LDP)로 받아주고 웹사이트에 올려준 그렉 헨킨스(Greg Hankins)에게 감사드린다.

역자 서문

이 책은 리눅스 문서화 프로젝트의(Linux Documentation Project)의 하나인 David A Rusling의 저서를 바탕으로 번역한 것이다. 번역은 서울대 컴퓨터 연구회 졸업생 모임인 돌도끼의 많은 사람들이 참여하여 이루어졌다. 원래는 돌도끼 내의 리눅스 연구 모임에서 리눅스 커널을 공부하던 중, 이 글이 리눅스 커널을 이해하기에 좋은 글이라는 생각 이 들어 번역을 시작하게 되었고 많은 사람들의 도움을 받아서 이루어지게 되었다.

번역은 1차 번역과 2차 번역 두 과정으로 이루어졌으며, 되도록 원문을 뜻을 살리면서 문장을 알기 쉽게 가다듬고, 문맥을 자연스럽게 하려고 했다. 그리고 설명이 필요한 부분에 대해 서는 되도록 많은 주석을 달려고 했으며, 책의 내용에 추가할 내용은 각 장의 끝에 추가하 려고 했다.

원문은 Version 0.8-3을 바탕으로 하였고, 책에 있는 감수 노트(REVIEW NOTE) 등도 그대로 놔두었다. 번역판의 첫 버전은 0.1.0이고, 여기에는 서문을 포함하여 1장부터 14장까지의 원 문의 내용이 들어 있다. 여기에는 부록은 빠져 있으며, 마지막에 용례집이 들어있다.

이 글이 모든 것을 담고 있고 완전한 것은 아니지만, 리눅스를 공부하는 사람들에게 많은 도움이 되리라 고 생각한다. 이 글에 잘못된 부분이 있거나, 보충할 내용이 있으면 언제든지 linux@flyduck.com으 로 메일을 보내주시고, 다음에 개정판을 낼 때 반영하도록 하겠습니다. 이 글의 가장 최신판은 <http://linux.flyduck.com/tlk/>에서 구할 수 있다.

참고로 아직 이 번역에 대해 저자에게서 공식적인 허락을 받진 못했다. 저자에게 편지로 이 문서를 한국어로 번역한다고 이야기를 했지만 답장이 오지 않았기 때문에, 앞의 안내문에 따라 암묵적으로 허가한 것으로 생각하고 있다.

마지막으로 좋은 글을 쓴 David A Rusling과 이 번역과정에 참여해 준 모든 분들께 감사드린다.

1999년 11월 8일
돌도끼

번역 기록

1차 번역 : 1999년 7월 30일 - 1999년 9월 17일

1차번역은 게시판을 통해 사람들이 자유롭게 참여할 수 있는 형태로 이루어졌다. 여기에는 스물명이 넘는 사람들이 참여하였지만, 다음 네 사람이 큰 공헌을 하였다.

이호 (flyduck) : flyduck@flyduck.com

심마로 (maro) :

고양우 (newcat) :

김성룡 :

이외에 다음 사람들이 많은 도움을 주었다.

서창배 (cbsuh)

신문석 (scmoon)

김기용 (gyong)

김진석 (jinsuk)

손은석 (soneus)

정직한 (honest)

윤경일 (kiyoon)

홍경선 (liberty)

이승 (icarus)

이승철 (sclee)

이대현 (donky)

그 밖에 참여한 사람들은 다음과 같다.

이준희 (jhlee), 황태연 (dolphin), 김현석, truejaws, PCK, 김종원, 금화섭, 홍석근

2차 번역 : 1999년 9월 18일 - 1999년 11월 7일

2차 번역에는 다음 세사람이 참여하였으며, 각각 다음과 같은 부분을 맡았다. 2차 번역에는 오역을 잡고, 문맥을 다듬으며, 주석을 다는 일이 포함되었으며, 최종적으로 그림을 추가하고 문서 포맷을 일치시키는 일이 추가되었다.

이호 (flyduck) : 1장 - 8장, 11장 - 14장

고양우 (newcat) : 9장

심마로 (maro) : 10장

Version 0.8-3, 번역판 0.1.0 : 1999년 11월 8일

서문과 1장부터 14장, 용례, 간단한 목차를 포함하고 있다.

역주 1) 원문에는 "hackers" (sic)라고 하고 있는데, sic는 "원문대로"라는 의미로 의심나는 원문을 그대로 인용할 때 쓰는 표기이다. (flyduck)

2) HOWTO는 말그대로 무언가를 어떻게 하면 되는지 적어놓은 문서이다. 많은 HOWTO 문 서들이 리눅스 용으로 존재하며, 이들 모두 대단히 유용하다.