데비안 GNU/리눅스 2.1 설치 (Intel x86)

Bruce Perens
Sven Rudolph
Igor Grobman
James Treacy
Adam Di Carlo

version 2.1.11, 28 August, 1999

요약

이 문서는 Intel x86 (”i386”) 아키텍쳐에 Debian GNU/Linux 시스템을 설치하는 방법을 설명한다. 또한 더 자세한 정보가 있는 곳과 새로운 Debian 시스템을 더욱 잘 활용하는 방법에 대한 정보를 포함하고 있다. 그러나 이 문서에서 다루는 과정을 기존의 시스템을 업그레이드 하는데 이용해서는 안될 것이다. 업그레이드를 원한다면 Debian 2.1 Release Notes (`http://www.debian.org/releases/2.1/i386/release-notes/`)를 참고하기 바란다.

저작권

목 차

제 1 장

Welcome to Debian

데비안을 시험해 보고자 하는 여러분의 결정을 기쁘게 생각한다. 여러가지 운영체제 중에서도 데비안은 특별하다는 것을 여러분이 알게 되리라 믿어 의심치 않는다. 전 세계의 훌륭한 자유소프트웨어를 수집하여 일관성을 가진 하나의 것으로 통합한 것이 데비안 이다. 하나로 뭉쳐 놓으면 더 큰 힘을 발휘하게 되는 법이다.

Debian GNU/Linux 배포본은 많은 소프트웨어 *패키지*로 구성되어 있다. 각 패키지는 실행 프로그램, 스크립트, 문서, 그리고 설정 정보로 이루어져 있다. 각 패키지는 *개발자*가 있어서 관리 책임을 지고 있다. 이런 방식으로 데비안은 *눈부시게* 성장하고 있다. Debian Social Contract (`http://www.debian.org/social_contract`)를 지키는데 동의한다면 누구라도 새로운 개발자가 될 수 있다. 개발자는 데비안에 새로운 소프트웨어를 도입할 수 있다. 자유소프트웨어라는 개념에 관한 우리의 기준에 적합한 소프트웨어 이어야 하며, 작성된 패키지는 우리의 품질기준을 만족해야 한다.

자유소프트웨어에 관한 데비안의 기준을 데비안 자유소프트웨어 안내 (`http://www.debian.org/social_contract\#guidelines`)에 명확하고 간결하게 밝히고 있다. 자유소프트웨어 운동에 미치는 이 문서의 영향력은 지대하며, Open Source Free Software Guidelines (`http://opensource.org/osd.html`)의 기초를 제공하였다.

데비안의 품질기준을 상세하게 기술한 명세서는 Debian Policy (`http://www.debian.org/doc/debian-policy/`) 외에는 없다. 이 문서에는 데비안 패키지가 갖추어야 할 품질과 기준을 정의하고 있다.

트로이 목마 그리고 여타 나쁜 의도를 가진 소프트웨어로부터 여러분의 시스템을 보호하기 위하여 각 패키지가 실제로 해당 개발자에 의해 보내진 것인지 검증하고 있다. 또한 패키지 개발자는 패키지를 작성할 때 안전성에 크게 주의를 기울이고 있다. 공개된 패키지에 보안상의 문제가 발생하면 대개는 빠른 시일내에 수정판을 내놓고 있다. 주기적으로 여러분의 시스템을 갱신한다면 보안 문제를 해결한 수정판을 받아서 설치하게 될 것이다.

데비안에 대한 일반적인 정보가 더 필요하다면 데비안 FAQ (`http://www.debian.org/doc/FAQ/`)를 보라.

제 1 절 이 문서의 최신판을 구하려면

이 문서는 꾸준히 갱신되고 있다. 2.1 release에 대한 최신 정보를 얻고자 한다면 데비안 2.1 페이지 (`http://www.debian.org/releases/2.1/`)를 확인하기 바란다. 또한 Official Install Manual 페이지 (`http://www.debian.org/releases/2.1/i386/install`) 에서 이 문서의 최신판을 구할 수 있다.

제 2 절 이 문서의 구성

이 문서는 데비안을 처음 접하는 사용자에게 제공되어져야 하는 매뉴얼이다. 보유하고 있는 하드웨어의 동작에 관한 일반적인 지식을 여러분이 갖고 있다는 가정하에, 되도록이면 전문적 지식이 없는 사용자도 이 문서를 읽고 이해할 수 있도록 노력하였다.

물론 이 문서에는 전문가가 관심을 가지는 부분에 대하여 참고정보를 수록하고 있다. 또한, 최소 설치에 필요한 용량, 데비안 설치 시스템이 지원하는 하드웨어에 대한 자세한 사항 등을 포함하고 있다. 전문가는 이문서에서 필요한 부분만 발췌하여 읽으면 될 것이다.

기본적으로 이 문서의 구성은 사용자가 설치과정을 순서대로 따라갈 수 있도록 되어 있다. 이 문서의 각 절은 설치 과정의 각 단계와 관련지을 수 있다.

1. '시스템 요구 조건' on page 4, 여러분의 하드웨어가 설치 시스템이 요구하는 조건을 만족하는지 확인한다.
2. '시작하기 전에' on page 10, 데비안을 설치하기 전에 필요한 준비사항에 대한 내용을 포함하고 있다. 즉, 시스템의 백업과 시스템 관리를 위해 필요한 정보 및 하드웨어의 설정.
3. '하드 디스크 파티션에 대하여' on page 15, 하드디스크 파티션에 대한 내용. 파티션을 변경하는 것은 쉬운일이 아니기 때문에 매우 중요하다.
4. '데비안 설치에 이용될 수 있는 방법들' on page 21, 데비안을 설치하는 여러가지 방법. 그 방법에 따라 설치 매체를 선택하고 준비한다.
5. '설치 시스템으로 부팅하기' on page 30, 설치 시스템을 부팅시키기 위한 정보. 또한, 부팅에 어려움이 있을 경우의 문제 해결 과정을 포함하고 있다.
6. '`dbootstrap`을 이용한 시스템 초기화 설정' on page 35, '`dbootstrap` 소개' on page 35 에서 "'Configure the Network'" on page 35 까지, 시스템 초기 설정에 대한 내용.
7. "'Install the Base System'" on page 35, 베이스 시스템 설치.
8. '진실의 순간' on page 35, 설치된 베이스 시스템으로 부팅. 베이스 시스템 설치 후 해야 할 작업.
9. '나머지 설치 과정' on page 35, 나머지 설치 과정. `dselect`를 이용하여 필요한 패키지 설치.

설치를 완료한 후 '앞으로 무엇을 할 것인가?' on page 46를 읽을 것을 권한다. Unix, 데비안 그리고 커널의 교체 방법에 대한 더 자세한 정보를 어디에서 찾을 것인지 설명하고 있다. 소스를 이용하여

자신만의 인스톨 시스템을 구축하고자 한다면 '부트 플로피에 관한 기술적 정보' on page 50를 읽어보라.

마지막으로, 이 문서에 대한 정보 그리고 이 문서에 기여하는 방법은 'Administrivia' on page 52에 설명되어 있다.

제 3 절 저작권과 소프트웨어 사용권에 관하여

대부분의 소프트웨어와 함께 오는 사용권에 대한 글을 당신이 읽었을 것으로 생각한다 – 당신은 한 컴퓨터에 한 카피의 소프트웨어만을 사용할 수 있다고 명시되어 있다. Debian GNU/Linux 시스템은 그렇지는 않다. 우리는 당신의 학교 혹은 회사에서 모든 컴퓨터에 한 카피를 갖고 사용하는 것에 대해 개의치 않는다. (오히려 장려한다.) 친구들에게 빌려주고, 친구들 컴퓨터에 설치하도록 도와주기 바란다. 당신은 수천 카피를 만들어서 팔 수도 있다 – 단, 몇 가지 제한이 있기는 하다. 이렇게 할 수 있는 이유는 데비안은 *자유소프트웨어* 이기 때문이다.

자유소프트웨어의 의미가 저작권이 없다는 것은 아니다. 또한, 이 소프트웨어가 담긴 CD를 무료로 배포한다는 의미도 아니다. 그것은 단지 소프트웨어를 복사해서 사용할 권리에 대해 반드시 금전적 지불을 할 필요가 없다는 것을 의미한다. 또한 누구나 소프트웨어를 확장, 개조, 수정할 수 있을 뿐만 아니라 그 결과를 배포할 수 있다는 것을 의미한다. [1]

시스템에 설치된 수많은 프로그램들은 *GNU General Public License*, 혹은 GPL을 따르고 있다. GPL은 프로그램 한 카피를 배포할 때에는 반드시 프로그램의 소스코드를 포함하도록 하고 있다. 이 말은 여러분이 그 소프트웨어를 수정할 수 있다는 것을 보증하는 것이다. 그래서, 우리는 데비안 시스템에 있는 모든 프로그램의 소스코드를 포함하였다.[2] 데비안의 경우, 프로그램에 따라 다른 형태의 저작권과 소프트웨어 사용권이 적용되는 것이 있다. 시스템에 데비안을 설치하면 `/usr/doc/\textit{ÆÐÅřÁö--ÀÌÿğ}/copyright` 디렉토리에서 모든 프로그램들의 저작권과 사용권에 대한 자료를 얻을 수 있다.

어떤 소프트웨어가 있을 때, 그것을 데비안의 주 배포본에 포함시키기 위해서는 그 소프트웨어가 자유소프트웨어인지 아닌지를 판단해야한다. 데비안은 이러한 판단을 어떻게 하는지 그리고 사용권에 대한 더 자세한 정보를 원한다면 Debian Free Software Guidelines (`http://www.debian.org/social_contract\#guidelines`)를 참고하기 바란다.

가장 중요한 법적 사항은 이 소프트웨어는 *어떠한 보증도 없다*는 것입니다. 프로그래머는 공동체의 이익을 위해서 이 소프트웨어를 만든 것이다. 어떤 주어진 목적에 그 소프트웨어가 적합하다는 보장은 할 수 없다. 그러나 이 소프트웨어는 수정이 가능한 자유소프트웨어이기 때문에 필요하다면 여러분의 조건에 맞도록 소프트웨어를 수정할 수 있다. 그리고 이와 같은 방식으로 다른 사람들에 의해 확장된 소프트웨어를 사용하는 즐거움을 느낄 수도 있을 것이다.

[1] 사용 가능한 패키지 중에는 free 이어야 한다는 우리의 조건을 만족하지 않는 패키지들도 많다는 사실를 주지하기 바란다. 이들 패키지는 `contrib` 또는 `non-free` 영역에서 배포된다. 데비안 FAQ (`http://www.debian.org/doc/FAQ/`)의 "Debian FTP archives" 부분을 읽어보라.

[2] 데비안 FAQ (`http://www.debian.org/doc/FAQ/`)를 뒤져보면 데비안 소스 패키지를 구해서 설치하는 방법을 알 수 있다.

제 2 장

시스템 요구 조건

데비안을 설치하기 위해 필요한 하드웨어에 대하여 알아본다. 또한 GNU와 리눅스가 지원하는 하드웨어에 대한 정보를 어디에서 더 많이 얻을 수 있는 지도 언급할 것이다.

제 1 절 지원되는 하드웨어

데비안은 리눅스 커널과 GNU tool-sets이 필요로하는 것 이상의 하드웨어를 요구하지는 않는다. 따라서 임의의 아키텍쳐, 또는 플랫폼으로 리눅스 커널과 libc, gcc 등이 이식되어 있고 해당하는 데비안 배포본이 있다면 그러한 아키텍쳐 또는 플랫폼에 데비안을 설치하여 운영하는 것이 가능하다.

그러나 우리가 제공하는 부트 플로피는 지원하는 하드웨어와 관련하여 몇 가지 제한이 있다. 일부 플랫폼의 경우, 리눅스가 이를 지원한다 하더라도 부트 플로피가 직접적으로 지원하지 않을 수도 있다. 이런 경우에는 여러분에게 맞는 응급 복구 (rescue) 디스크를 만들거나 네트워을 통한 설치를 검토해 보아야 한다.

Intel x86에 대해서 지원하는 모든 종류의 하드웨어 설정을 설명하는 것은 아니며, 그 보다는 일반적인 정보와 함께 추가로 필요한 정보를 어디에서 얻을 것인지 언급할 것이다.

1.1 지원되는 아키텍쳐

데비안 2.1는 네 종류의 아키텍쳐를 지원한다. 그 각각은, Intel x86을 기반으로 한 아키텍쳐; Atari, Amiga, and Macintoshes 같은 Motorola 680x0 머쉰; DEC Alpha 머쉰; Sun SPARC 머쉰 이다. 이들을 *i386*, *m68k*, *alpha*, *sparc*로 각각 표기할 것이다.

이 문서는 *i386*에 설치하는 경우를 다루고 있다. 다른 아키텍쳐에 해당하는 문서는 각각 별도로 존재한다.

1.2 지원되는 CPU, 마더보드, Video

지원되는 하드웨어에 대한 완전한 정보는 Linux Hardware Compatibility HOWTO (`http://www.linuxdoc.org/HOWTO/Hardware-HOWTO.html`)에서 구할 수 있다. 이 절에서는 단지 기본적인 사항만을 훑어볼 것이다.

거의 모든 x86 기반의 프로세서를 지원한다. 물론 AMD 와 Cyrix 프로세서도 지원한다. 그러나 리눅스는 286 이하의 시스템에서는 동작하지 않는다.

시스템 버스는 마더보드의 일부이며 이를 통하여 CPU는 저장 장치등의 주변 장치와 통신을 한다. 여러분의 컴퓨터는 ISA, EISA, PCI, Microchannel Architechure (MCA, IBM PS/2에서 사용하는 버스), 또는 VESA Local 버스 방식을 사용해야만 한다. VESA Local 버스는 VL 버스로도 알려져 있다.

랩탑 또한 지원된다. 대개 랩탑은 나름대로 특수화 되어 있거나 자신만의 하드웨어를 갖고 있는 경우가 많다. 여러분의 랩탑이 GNU/Linux와 원만하게 조화를 이룰 수 있는지는 Linux Laptop 페이지 (`http://www.cs.utexas.edu/users/kharker/linux-laptop/`)를 참고하기 바란다.

콘솔 터미널(console terminal)을 위해 VGA호환 디스플레이 인터페이스를 사용해야 한다. 요즘 나오는 거의 모든 디스플레이 카드들이 VGA와 호환된다. X11(X 윈도우 시스템)이 필요 없다면, CGA, MDA 혹은 HGA (허큘리스 그래픽 어뎁터)도 사용할 수 있다. 이 문서에 설명된 설치 과정 동안에는 X11을 사용하지 않는다는 것을 주지하기 바란다.

어떤 그래픽 카드에 대한 데비안의 지원 여부는 XFree86의 X11 시스템에 의해 결정된다. AGP 비디오 슬롯은 PCI의 것을 수정한 것이며, 대부분의 AGP 비디오 카드는 XFree86과 잘 동작한다. 지원되는 그래픽 버스, 카드, 모니터, 그리고 포인팅 디바이스에 대한 자세한 것은 `http://www.xfree86.org/`를 참고하기 바란다. 데비안 2.1에는 X11 revision 3.3.2.3a가 포함되어 있다.

1.3 다중 프로세서(Multiple Processors)

”symmetric multi-processing” 또는 SMP 라고 불리우는 다중 프로세서 지원이 이 아키텍쳐의 경우에는 이루어 진다. 그러나 표준 데비안 2.1 커널 이미지는 SMP 지원을 포함하지 않는다. 그러나 SMP 지원을 갖추지 못한 커널이라 하더라도 SMP 시스템에 사용할 수 있고, 따라서 데비안을 설치하는 데는 지장이 없다. 이 경우 커널은 단지 첫번째 CPU 만을 이용한다.

다중 프로세서를 활용하고자 한다면, 표준 데비안 커널을 교체해야 할 것이다. ‘새로운 커널 컴파일하기’ on page 46 에 이와 관련된 내용이 있으니 참고하기 바란다. 현재 상태에서 (커널 버전 2.0.36) 다중 프로세서 지원을 가능하게 하려면 커널 소스에 있는 Makefile의 “SMP = 1”이라는 부분의 주석처리를 해제 하고 커널을 컴파일 하면 된다. 다중 프로세서 시스템에서 소프트웨어를 컴파일한다면, `make(1)` 설명서에서 “-j” 옵션을 찾아서 참고 하기 바란다.

제 2 절 설치에 이용할 수 있는 매체

데비안 설치에 이용할 수 있는 매체는 네 가지가 있다. 플로피, 시디롬, 하드디스크, 네트웍을 통하여 각각 설치 가능하다. 뿐만 아니라 이들 방법을 섞어서 사용할 수도 있다. ‘데비안 설치에 이용될 수 있는 방법들’ on page 21을 참고하기 바란다.

선호도가 가장 낮기는 하지만, 플로피 디스크로 설치하는 것이 일반적인 방법이다. 많은 경우에 Rescue Floppy를 이용하여 부팅한다. 일반적으로 필요한 것은 고밀도 (1440 킬로바이트) 3.5 인치 플로피 드라이브가 전부이다.

배밀도 5.25 인치 설치 디스켓 (1200 k) 또한 제공한다.

일부 아키텍쳐의 경우에는 시디롬을 이용한 설치도 지원한다. 시디롬 부팅을 지원하는 머쉰이라면 설치 과정에 플로피 디스크가 전혀 필요없다. 시디롬 부팅을 지원하지 안더라도, 일단 다른 수단으

로 부팅한 후 시디롬을 이용하여 설치할 수 있다. '시디롬을 이용하여 설치하기' on page 21를 참고하기 바란다.

SCSI 와 IDE/ATAPI 시디롬 모두 사용할 수 있다. 게다가 비표준 시디롬이라 하더라도 리눅스가 지원하는 것이라면 부트 디스크도 지원한다. 예를 들면, Mitsumi, Matsushita 시디롬 같은 것 들이다. 이들 시디롬을 사용하려면 특별한 부트 파라메타 또는 다른 처치가 필요하다. 그리고 이들 비표준 시디롬으로 부팅하는 것은 곤란한다. Linux CD–ROM HOWTO (`http://www.linuxdoc.org/HOWTO/CDROM-HOWTO.html`)에는 리눅스에서 시디롬을 사용하는 방법에 대해 자세히 다루고 있다.

하드디스크를 이용하여 설치할 수도 있다. 리눅스를 설치할 파티션 이외에도 비어있는 충분한 크기의 파티션이 있다면 이 방법을 이용하는 것이 훌륭한 선택이다. 일부 플랫폼의 경우, 즉 AmigaOS,TOS, MacOS로부터 부팅이 가능하도록 자체 인스톨러를 갖고 있기도 하다.

마지막으로 네트웍을 통한 설치 방법이 있다. NFS를 이용하여 설치 할 수 있다. 베이스 시스템을 설치한 후에는 FTP, HTTP, NFS를 이용하여 나머지를 설치 할 수 있다.

설치 방법 각각에 대한 자세한 설명과 가장 적당한 설치 방법 선택을 위한 도움말을 구하고자 하면 '데비안 설치에 이용될 수 있는 방법들' on page 21을 참고하기 바란다. 부팅과 설치에 이용할 장치를 데비안 설치 시스템이 지원하는 지 반드시 확인하기 바란다.

2.1 지원하는 저장 장치

데비안 부트 디스크는 최대한 많은 시스템을 구동시킬 수 있도록 제작된 커널을 탑재하고 있다. 불행하게도 이로 인하여 전혀 사용되지 않을 지도 모를 드라이버들이 커널에 포함되어 비대한 커널이 된다 (자신만의 커널을 제작하려면 '새로운 커널 컴파일하기' on page 46를 읽어보기 바란다). 그러나 하드웨어에 상관없이 데비안을 설치할 수 있도록 하기 위해서는 가능한한 많은 종류의 장치를 지원하는 것이 바람직하다.

일반적으로 데비안 설치 시스템은 플로피, IDE 드라이브, IDE 플로피, 병렬 포트 IDE 장치, SCSI 콘트롤러 와 드라이브를 지원한다. 지원하는 파일시스템으로는 MINIX, FAT, Win-32 FAT 익스텐션 (VFAT) 등이 있다 (설치 시스템은 NTFS를 지원하지 않지만 나중에 추가할 수 있다. '새로운 커널 컴파일하기' on page 46를 참고하기 바란다).

지원하는 하드웨어를 일일이 설명하기 보다는, 리눅스가 지원하는 하드웨어이지만 데비안 부트 시스템은 지원하지 않는 하드웨어를 설명하는 것이 더 쉽다.

MFM, RLL, IDE 또는 ATA 라 보통 불리는 "AT" 하드 디스크 인터페이스를 에뮬레이션 해주는 디스크 인터페이스를 지원한다. IBM XT 컴퓨터에서 사용했던 8 비트 하드디스크 콘트롤러는 모듈로서 지원한다. 많은 회사에서 생산되는 SCSI 디스크 컨트롤러도 지원한다. Linux Hardware Compatibility HOWTO (`http://www.linuxdoc.org/HOWTO/Hardware-HOWTO.html`) 문서에 자세한 사항이 있다.

아래의 장치를 포함하여 IDE SCSI 드라이브와 일부 SCSI 컨트롤러는 지원하지 않는다.

- SmartCache III/IV, SmartRAID 콘트롤러 종류, DDP PM2011B, PM2012B 콘트롤러 같은 EATA-DMA 프로토콜을 따르는 SCSI 호스트 어댑터.
- 53c7 NCR 계통의 SCSI 콘트롤러 (53c8 과 5380은 지원한다).

제 3 절 메모리와 필요한 디스크 공간

적어도 4MB이상의 램과 35MB 이상의 하드디스크가 필요하다. X 윈도우 시스템, 프로그램 개발에 필요한 것들, 라이브러리 등을 설치하려면 최소한 300MB 이상의 하드디스크 공간이 필요하다. 이것 저것 필요한 것들을 모두 설치하려면 800MB 가량의 공간이 필요하다. 데비안에서 제공하는 모든 소프트웨어를 설치하려면 2GB 정도가 필요하다. 실제로는 일부 패키지가 다른 패키지와 충돌하기 때문에 모든 소프트웨어를 설치한다는 것은 있을 수 없다.

제 4 절 주변 장치와 여타 하드웨어

리눅스는 마우스, 프린터, 스캐너, 모뎀, 네트웍 카드, PCMCIA 장치 등 너무나도 다양한 하드웨어를 지원한다. 그렇지만, 시스템을 설치하는 동안에는 이런 장치가 사용하지 않는다. 리눅스가 지원한다 하더라도 설치 시스템이 지원하지 않는 주변 장치에 대하여 이 절에서 살펴보겠다.

여러분의 하드웨어를 리눅스가 지원하는지 확인하려면 Linux Hardware Compatibility HOWTO (`http://www.linuxdoc.org/HOWTO/Hardware-HOWTO.html`)를 참고하기 바란다.

데비안 설치 디스크는 일부 네트웍 인터페이스 카드 (NIC)를 지원하지 않으며 그 각각은 다음과 같다. AX.25 카드 와 프로토콜, 3Com EtherLink Plus (3c505), EtherLink16 (3c507), NI5210 카드, 일반 NE2100 카드, NI6510, NI16510 EtherBlaster 카드, SEEQ 8005 카드, Schneider & Koch G16 카드, Ansel Communications EISA 3200, the Zenith Z-Note 내장형 네트웍 카드. Microchannel (MCA) 네트웍 카드 역시 지원하지 않지만 비공식적인 지원을 알아보려면 리눅스 MCA 디스크 이미지 (`ftp://ns.gold-link.com/pub/LinuxMCA/`) 와 리눅스 MCA 토론 아카이브 (`http://www.dgmicro.com/linux_frm.htm`)를 참고하기 바란다. FDDI 네트웍은 카드와 프로토콜 모두 지원하지 않는다.

ISDN의 경우, Gernam 1TR6에 사용되는 D-channel 프로토콜을 지원하지 않으며 또한 Spellcaster BRI ISDN 보드도 지원하지 않는다.

설치 디스크는 사운드 카드 또한 지원하지 않는다.

제 5 절 GNU/Linux 와 잘 맞는 하드웨어 구입하기

이제는 시스템에 데비안 또는 여타 GNU/Linux 배포본을 설치하여 판매하는 곳이 몇 군데 있다. 이 경우 돈이 좀 더 들어 가겠지만, 대신에 하드웨어가 GNU/Linux와 잘 맞을까 하는 불안감에 시달리지 않아도 된다. 윈도즈를 기본 제공하는 머쉰을 사야 한다면, 윈도즈에 딸려오는 소프트웨어 사용권을 잘 읽어보기 바란다. 그 사용권을 거부할 수 있으며, 윈도우즈 가격에 해당하는 금액을 환불 받을 수 있다. 더 자세한 것은 `http://www.linuxmall.com/refund/`를 참고하기 바란다.

리눅스를 기본 제공하는 시스템을 구입하느냐 또는 중고 시스템을 구입하느냐에 상관없이 하드웨어가 리눅스와 잘 맞는지를 점검하는 것은 중요하다. 하드웨어 지원 여부를 점검하기 바란다. 그리고 구입하려는 것이 리눅스 시스템이라는 것을 판매자에게 알려라. 이왕이면 리눅스를 잘 지원하는 하드웨어 벤더의 제품을 구입하기 바란다.

5.1 독점적인 또는 폐쇄적인 하드웨어를 피하라

몇몇 하드웨어 제조업자들은 그들이 만든 하드웨어의 드라이버를 만드는 방법을 가르쳐주지 않거나, 우리가 리눅스 소스 코드를 배포하는 것을 방해하게 하는 '비공개 동의' 조항에 따라 문서를 참조하는 것을 허락하지 않기도 한다. 한 가지 예는 IBM 랩톱 DSP 사운드 시스템(IBM laplop DSP sound system)에 대한 것인데 이는 IBM사의 ThinkPad 시스템에 최근 사용되는 것이다. - 이 시스템의 몇몇은 모뎀에 사운드 시스템을 결합한 것이다. 또 다른 예는 구형의 Macintosh 계열에서 사용된 독점적 하드웨어이다.

우리는 이들 장치에 대한 문서를 참조할 권리를 얻지 못했기 때문에, 리눅스에서는 이들 장치가 작동하지 않는다. 여러분은 그런 하드웨어 제조업자들에게 문서를 배포할 것을 요청함으로써 우리를 도와줄 수 있다. 많은 사람들이 요청한다면, 그들도 자유소프트웨어 공동체의 시장성을 무시하지 못할 것이다.

5.2 마이크로소프트 윈도즈에 맞춰진 하드웨어

현재의 이상스런 추세는 윈도즈 전용 모뎀과 프린터가 우후죽순 처럼 자라나고 있다는 것이다. 일부 경우에, 이들 제품은 마이크로소프트 윈도즈라는 운영체제를 위해 특별히 설계 되어, "WinModem" 혹은 "윈도우 기반의 컴퓨터를 위해 특별히 만들어진 제품"이라는 표제를 갖고 있다. 일반적으로 하드웨어에 내장되어 있는 프로세서를 제거해서 이러한 제품을 만든다. 즉, 내장 프로세서가 해야 할 일을 윈도즈 드라이버에 맡겨 버리는 것이다. 윈도즈 드라이버는 이 일을 하기 위해 CPU를 사용해야 한다. 이러한 정책은 하드웨어 가격을 내렸지만, 절약된 만큼의 가격이 사용자에게 환원되는 것은 아니다. 내장 프로세서를 가진 하드웨어와 비교할 때 오히려 윈도즈 전용 하드웨어의 가격이 비싸기까지 하다.

윈도즈 전용 하드웨어의 사용을 피해야 할 두 가지 이유가 있다. 첫째, 하드웨어 생산자는 일반적으로 리눅스용 드라이버를 만들어내지 않는다. 일반적으로, 하드웨어와 소프트웨어의 연결 드라이버는 저작권이 있고, 문서는 하드웨어의 내부적 사항에 대한 비공개 동의가 없이 허락되지 않는다. 자유소프트웨어의 저자는 소스코드를 모두 공개하기 때문에 이들 자료의 이용을 사전 봉쇄하고 있다. 두번째, 내장 프로세를 제거한 장치의 경우, 내장 프로세서가 해야 할 일을 운영체제가 대신 해야 한다. 이 일은 종종 실시간 우선권을 요구하기 때문에, CPU가 이 장치를 돌리는 동안 또 다른 프로그램을 수행시킬 여력이 없게 된다. 전형적인 윈도즈 사용자는 리눅스 사용자처럼 다중처리를 심하게 하지 않기 때문에, 하드웨어 제조업자들은 그들이 만든 하드웨어가 CPU사용률의 주범이라는 것을 윈도즈 사용자가 깨닫지 못하기를 바라는 것입니다. 그렇기 때문에, 주변장치 제조업자들이 처리 능력이 떨어지는 장치를 만든다면, 윈도즈 95나 NT를 포함한 모든 다중 처리 운영 체제는 CPU에 부하를 많이 받기 때문에 그 수행 능력이 저하된다.

당신은 우리가 하드웨어제조업자들의 하드웨어를 지원하기 위해 필요한 문서나 다른 자원들을 하드웨어 제조자들이 공개하도록 업자들을 독려하는 것으로 우리를 도와줄 수 있지만, 제일 좋은 방안은 Linux Hardware Compatibility HOWTO (`http://www.linuxdoc.org/HOWTO/Hardware-HOWTO.html`) 문서에 "지원되는 하드웨어" 라고 명시될 때까지 이런 종류의 하드웨어 사용을 피하는 것이다.

5.3 가짜 패리티 램 혹은 "가상의" 패리티 램

컴퓨터 가게에서 패리티 램을 사면, 아마도 *실제 패리티* 램 대신에 *가상 패리티* 램을 얻을 것이다. 가

상 패리티 램(SIMM)은 보통 (항상 그런 것은 아니다) 비패리티 램 보다 하나 이상의 칩이 더 있기 때문에 구별 할 수 있고, 그 특별한 칩은 다른 것들보다 크기가 작다. 가상 패리티 램은 비패리티 램처럼 정확히 작동한다. 그런데, 실제 패리티 램은 마더보드 상에서 단일 비트 에러가 있을 때 알려 주지만 가상 패리티 램은 그렇지 않다. 가상 패리티 램에 돈을 들일 필요는 없다. 그 보다는 실제 패리티 램을 산다면 8비트 모두에 대해 한 개의 특수 비트를 실제로 사는 것이기 때문에 실제 패리티 램에 조금 더 돈을 들이는 게 낫다.

Intel x86 용 램에 대한 정보가 필요하다면 그리고 무엇이 가장 좋은 램인지 알고 싶다면 PC Hardware FAQ (`ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/sys/ibm/pc/hardware/systems/`)를 참고하기 바란다.

제 3 장

시작하기 전에

제 1 절　백업을 하자

시작하기 전에, 인스톨 작업 도중 하드디스크의 모든 데이터가 손실될 우려가 있으니 현재 시스템에 있는 모든 파일들을 백업하도록 하자. 설치 과정에 사용되는 프로그램은 상당히 믿을 만하고 수년동안 샤용해 왔지만, 그래도 문제는 있을 수 있다. 백업했다고 해서 안심하지 말고, 어떻게 대답할지 또 어떻게 행동할지에 대해서 생각하고 주의를 기울여야 할 것이다. 2분을 생각하면 불필요한 작업에 허비할 몇 시간을 벌 수 있다.

다중 부팅 시스템을 계획하고 있다면, 현재 설치되어 있는 운영체제의 설치 매체를 자신이 갖고 있는지 꼭 확인하기 바란다. 특히 부팅용 하드디스크를 다시 파티션해야 할 경우라면, 운영체제의 부트 로더를 다시 설치해야 하거나, 일부 경우(즉, Macintosh)에는 운영 체제를 통째로 다시 설치해야 한다.

제 2 절　미리 알아두어야 할 정보

이 문서 이외에도 다음의 자료들이 필요할 것이다.

the cfdisk (`cfdisk.txt`) 매뉴얼,

the fdisk (`fdisk.txt`) 매뉴얼,

dselect Tutorial (`dselect-beginner.html`),

Linux Hardware Compatibility HOWTO (`http://www.linuxdoc.org/HOWTO/Hardware-HOWTO.html`).

컴퓨터가 네트웍에 24시간 연결되어 있다면 (즉, 이더넷 또는 이와 동등한 종류의 연결 그러나 PPP 연결은 아님), 네트웍 시스템 관리자에게 아래의 정보를 물어서 알아 두어야 한다.

- 여러분의 호스트 이름(여러분이 결정할 수도 있다).
- 여러분의 도메인 이름.

- 여러분 컴퓨터의 IP 주소.
- 여러분 네트웍의 IP 주소.
- 여러분의 네트웍에서 사용할 수 있는 넷마스크.
- 여러분의 네트웍 상의 브로드캐스트 주소.
- 여러분의 네트웍에 게이트웨이가 있다면, 여러분이 라우팅해 주어야할 시스템의 디폴트 게이트웨이의 IP 주소.
- 네트웍 상에서 DNS (Domain Name Service) 서버로 사용해야 하는 시스템의 IP 주소.
- 여러분이 이더넷(Ethernet)을 사용하여 네트웍에 연결하는지에 대한 여부.
- 여러분의 이더넷 인터페이스가 PCMCIA 카드 인지에 대한 여부, 만약 그렇다면 사용하는 PCMCIA 콘트롤러의 타입.

여러분이 사용하는 네트웍이 PPP 또는 비슷한 종류의 다이얼업 연결이라면, 베이스 시스템을 네트웍을 이용하여 설치하지는 않을 것이다. 따라서 네트웍 설정을 어떻게 할 것인지는 시스템을 설치하고 나서 걱정해도 늦지 않다. PPP 설정에 대한 정보는 'PPP 설정' on page 35를 참고하기 바란다.

제 3 절　설치 전에 점검해야할 바이오스 및 하드웨어 설정

설치를 시작하기 전에 시스템을 점검해야 할 필요가 있다. x86 플랫폼을 제외한 여타 기종의 경우, 설치 전 하드웨어 설정이 비교적 간단한 반면에 x86 플랫폼의 경우는 그 악명이 자자하다.

데비안을 설치하기 전에 해야 할 필요가 있는 하드웨어 설정에 대해서 알아 보겠다. 일반적으로 시스템의 펌웨어 세팅을 점검하고 수정하는 일이 여기에 포함된다. "펌웨어"란 하드웨어가 사용하는 코어 소프트웨어이며, (전원을 올린 후) 부팅 과정 동안에 기동되는 매우 중요한 것이다.

3.1 바이오스 셋업 메뉴

바이오스는 컴퓨터를 부트시키는데 필요한 기본적인 기능을 제공하며, 궁극적으로 운영체가 하드웨어를 사용할 수 있도록 해 준다. 바이오스 설정을 조정하기 위해 바이오스 셋업 메뉴라는 것이 제공된다. 설치하기 전에 바이오스 설정이 정확한지를 *반드시* 확인해야 한다. 잘못되면 가끔씩 시스템이 정지하거나 또는 데비안의 설치가 불가능해 질 수도 있다.

"CMOS 설정 메뉴로 들어가는 방법은?"에 대한 답변을 PC 하드웨어 FAQ (`ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/sys/ibm/pc/hardware/systems/`)에서 찾을 수 있다. 아래에 그 내용을 인용하였다. 바이오스 (또는 "CMOS") 설정 메뉴로 들어가는 방법은 바이오스에 따라 다르다.

[From: burnesa@cat.com (Shaun Burnet)]

AMI BIOS Del key during the POST (power on self test)

Award BIOS Ctrl-Alt-Esc, or Del key during the POST

DTK BIOS Esc key during the POST

IBM PS/2 BIOS Ctrl-Alt-Ins after Ctrl-Alt-Del

Phoenix BIOS Ctrl-Alt-Esc or Ctrl-Alt-S

[From: mike@pencom.com (Mike Heath)] 일부 286 머쉰은 CMOS 설정 메뉴를 제공하지 않으며 CMOS 셋업 프로그램을 필요로 한다. 이러한 프로그램을 갖고 있지 않다면 쉐어웨어/프리웨어 프로그램의 사용을 시도해 보기 바란다. `ftp://oak.oakland.edu/pub/simtelnet/msdos/`에서 찾을 수 있다.

3.2 부팅 장치 선택

많은 바이오스 셋업 메뉴에는 시스템을 부팅할 때 사용되는 장치를 선택하는 부분이 있다. 부트할 운영체제를 `A:` (첫번째 플로피 디스크)에서 먼저 찾도록 설정하고, 그 다음 첫번째 시디롬 (`D:` 또는 `E:`로 표시될 수도 있음)에서 찾도록 설정한다. 그리고 마지막으로 `C:` (첫번째 하드디스크)에서 찾도록 설정한다. 데비안을 설치하기 위해 일반적으로 사용되는 것이 플로피 디스크와 시디롬이다. 따라서 위와 같이 설정해 놓으면 이들 두 장치중 하나로 부팅할 수 있는 것이다.

여러분의 시스템에서 시디롬으로 부팅하는 기능을 제공하지 않거나 또는 있다 하더라도 정상 작동하지 않는다고 해서 실망할 필요는 없다. 도스에서 `E:\(\backslash\)install\(\backslash\)boot.bat` (`E:`는 시디롬에 할당된 드라이브가 무엇이냐에 따라 바뀌어야 한다)를 실행시켜서 설치과정을 시작할 수도 있다. 더 자세한 것은 '시디롬을 이용하여 설치하기' on page 21를 참고하기 바란다.

또한, FAT (도스) 파티션으로부터 설치할 계획이라면 플로피 디스크가 전혀 필요없다. 이 방법으로 설치하고자 한다면 '도스 파티션을 이용하여 설치하기' on page 21를 참고하기 바란다.

3.3 Extended 대 Expanded 메모리

시스템이 ex*ten*ded 와 ex*pan*ded 메모리를 모두 지원할 때에는, 가능하면 expanded 메모리는 적게 설정하고, extended 메모리를 많이 설정하도록 한다. 리눅스가 요구하는 것은 extended 메모리이며, expanded 메모리는 사용할 수 없다.

3.4 바이러스 방지

바이오스가 제공하는 바이러스 경고 기능을 꺼 놓아야 한다. 만약 바이러스 방지 보드나 다른 특별한 하드웨어를 사용한다면, 그 기능을 꺼 놓거나 GNU/리눅스를 실행시키는 동안에는 그 하드웨어를 물리적으로 제거해야 한다. 이것은 GNU/Linux와 호환되지 않으며, 게다가 파일 시스템 허가권과 리눅스 커널의 보호 메모리 덕분에 바이러스에 대한 전례를 리눅스에서는 찾아보기 어렵다.[1]

[1]설치가 끝난 후에는 부트 섹터 보호 기능을 켜 놓을 수도 있다. 부트 매니저를 셋업한 후에는 마스터 부트 레코드 (MBR)를 건드릴 필요가 없다. 이 기능을 활성화 시킨다고 해서 보안상의 추가적인 잇점이 리눅스에 제공되는 것도 아니다. 그러나 윈도즈도 함께 사용한다면 도움이 될 수 있다.

3.5 셰도우 램

마더보드가 *셰도우 램* 또는 바이오스 캐싱 기능을 제공할 수도 있다. "Video BIOS Shadow", "C800-CBFF Shadow" 등이 그것이다. 모든 셰도우 램 기능을 비활성화 시키도록 한다. 셰도우 램은 마더보드 상이나 몇몇 컨트롤러 상에서 롬(ROM) 억세스를 빠르게 하는 역할을 한다. 롬에 있는 16비트 프로그램들 대신에 더 빠른 32비트 소프트웨어를 리눅스가 제공하기 때문에, 일단 부팅된 후에는 이들 롬을 사용할 필요가 없는 것이다. 셰도우 램 기능을 꺼 놓으면 프로그램이 이 램들을 평범한 메모리로 이용할 수 있다. 셰도우 램을 사용하도록 해놓으면 리눅스가 하드웨어 장치에 접근하는 것을 방해할 수도 있다.

3.6 고급 전원 관리 기능 (Advanced Power Management)

마더보드가 고급 전원 관리 기능(APM)을 지원한다면, APM이 전원 관리를 하도록 설정한다. doze,standby, suspend, nap, 그리고 sleep modes를 끄고, 하드디스크 절전 타이머도 끈다. 리눅스는 이 모드들의 제어를 할 수 있고, 바이오스 보다 전원 관리 기능이 뛰어나다. 하지만, 설치 플로피의 커널에는 APM 기능을 넣지 않았다. 왜냐하면 커널에 APM 기능을 넣을 경우 일부 랩톱 시스템이 작동하지 않는다는 보고 때문이다. 리눅스를 설치한 후, 여러분이 원하는 기능을 가진 커널을 컴파일하여 만들 수 있다. 자세한 것은 '새로운 커널 컴파일하기' on page 46을 참고하기 바란다.

3.7 터보 스위치

많은 시스템에는 CPU의 속도를 제어하는 *터보* 스위치가 있다. 이러한 경우라면 CPU의 속도를 높은 쪽으로 지정하도록 한다. 바이오스에서 터보 스위치를 소프트웨어적으로 끌 수 있다면 그렇게 하고서, 높은 속도로 시스템을 고정 한다. 특정 시스템에서 리눅스가 하드웨어 장치를 찾아내는 동안 터보 스위치의 소프트웨어 제어 부분을 우연히 건드릴 수 있다는 보고가 있었다.

3.8 CPU의 오버 클럭킹

많은 사람들은 90MHz CPU를 100MHz로 동작시키려는 등의 시도를 한다. 보통 작동하긴 하지만, 온도나 다른 조건에 CPU가 민감해지고, 시스템이 실제로 위험할 수 있다. 이 문서의 저자 중 한 사람은 1년 동안 그의 시스템을 오버 클러킹 해 놓고 사용했는데, 그 후 운영체제 커널을 컴파일 하는 동안 시스템의 `gcc`에서 예기치 못했던 일이 나타나기 시작했다. CPU의 스피드를 원래대로 되돌려 놓으니 문제가 해결 되었다.

3.9 불량램

`gcc` 컴파일러는 종종 불량램 (혹은 예측불허로 데이터를 변화시키는 다른 하드웨어 문제) 때문에 죽어버리곤 하는데, 이는 gcc가 반복적으로 왔다 갔다하면서 얽혀진 소스와 오브젝트 파일 사이에서 큰 데이터 구조를 만들어내기 때문이다. 이러한 데이터 구조 내부에서 발생한 에러는 잘못된 명령을 수행하거나 존재하지도 않는 주소의 접근을 유발한다. 이러한 것들로 인해 `gcc`가 죽어버리는 것이다.

최상의 성능을 발휘하는 마더보드는 패리티 램을 지원하고, 실제로 램 안에 단일 비트 에러를 갖고 있는지를 체크하여 그 이상 유무를 알려준다. 하지만 불행하게도 이 에러를 수정할 방법은 없기 때

문에, 일반적으로 불량램에 대해 경고를 하고는 즉시 시스템이 정지된다. 사실 따지고 보면 데이터에 에러가 있다는 것을 말하지 않고 조용히 있는 것보다는 메모리 불량을 경고하는 게 오히려 낫다. 그러므로, 가장 좋은 시스템을 위해서는 패리티와 실제 패리티 램 ('가짜 패리티 램 혹은 "가상의" 패리티 램' on page 4를 참고)을 지원하는 마더보드를 장착하는 것이 좋다.

실제 패리티 램을 가졌고, 마더보드가 지원한다면, 메모리 패리티 에러에 대해 인터럽트를 걸게 하는 옵션을 바이오스 설정에서 반드시 켜 놓아야 한다.

3.10 사이릭스 CPU와 플로피 디스크 에러

사이릭스 CPU를 가진 사람들은 시스템의 캐시를 꺼놓지 않으면 인스톨 도중 발생하는 플로피 디스크 에러 때문에 캐시를 꺼놓아야 했다. 이러한 이유로 캐시를 꺼놓았다면, 설치 후 캐시를 다시 켜놓아야 한다. 왜냐하면 캐시를 꺼놓으면 시스템이 *훨씬* 느리기 때문이다.

이것이 사이릭스 CPU의 문제점이라고 반드시 생각하지는 않는다. 리눅스는 그래도 잘 돌아간다. 기술적으로 이 부분에 대해 우리들은 관심이 있기 때문에, 이 문제에 대해 계속적으로 살펴볼 것이다. 현재 16비트 코드에서 32비트 코드로 이식 후 그러한 문제가 발생하는 것으로 추측하고 있다.

3.11 그 외에 살펴봐야 할 바이오스 설정

바이오스가 "15-16 MB Memory Hole" 같은 기능을 지원한다면, 꺼놓아야 한다. 리눅스는 램이 많으면 거기서 메모리를 찾으려 한다.

현재 우리들에게는 "LFB" 혹은 "Linear Frame Buffer" 라는 옵션이 있는 Intel Endeavor 마더보드에 대한 자료가 있다. 이 옵션에는 "Disable" 과 "1 Megabyte" 라는 두 가지 설정 중 하나를 선택할 수 있는데 "1 Megabyte" 라고 설정해야 한다. 이 옵션을 꺼 놓으면 플로피 디스크 드라이브가 설피 플로피를 제대로 읽지 못하고, 시스템이 갑자기 죽어 버린다. 이런 특별한 장치를 갖고 도대체 무엇을 하자는 것인지 이해 할 수 없다. – 단지 위와 같이 설정하면 작동하고, 그렇지 않으면 작동하지 않는다.

3.12 살펴봐야 할 하드웨어 설정

바이오스 설정 이외에 카드의 설정을 바꾸어야 할 수도 있다. 일부 카드는 소프트웨어적인 설정 메뉴를 제공하기도 하고, 또 다른 일부는 점퍼를 이용하여 설정해야 한다. 이 문서에서 모든 하드웨어에 대한 완벽한 정보를 제공하기를 바랄 수는 없고 유용한 팁이라도 제공되기를 바란다.

어떤 카드들이 "mapped memory"를 제공하면, 메모리는 0xA0000에서 0xFFFFF사이 (640K에서 1M 사이), 혹은 시스템 안의 총 램의 양보다 적어도 1메가바이트 더 큰 메모리 주소 어디에선가 매핑되어야 한다.

제 4 장

하드 디스크 파티션에 대하여

제 1 절 기초 지식

하드 디스크를 파티션 한다는 것은 간단히 말해서 하드 디스크를 여러 조각으로 나누는 것이다. 각 조각은 서로 독립적이다. 이는 집 안에 벽을 세우는 것과 거의 비슷하다. 즉 어떤 방에 가구를 들인다고 해서 그 행위가 다른 방에 영향을 미치지는 않는다.

시스템에 이미 어떤 운영체제 (윈도즈95, 98, NT, OS/2, MacOS, Solaris, FreeBSD)가 설치되어 있고, 같은 하드 디스크에 리눅스도 설치하고자 한다면, 아마도 디스크를 다시 파티션해야 할 것이다. 일반적으로 어떤 파일시스템을 이미 갖고 있는 파티션을 변경하면 그기에 있던 모든 데이타는 파괴되어 버릴 것이다. 그래서 다시 파티션 하기 전에 항상 백업해야 한다. 집에 빗대어 이야기하면, 집안 구조를 변경하기 위하여 벽을 부수기 전에 가구들을 집 밖으로 옮겨야 하는 것과 같은 이치이다.

다행스럽게도 일부 사용자에게는 다른 방법이 있다. '도스, Win-32 또는 오에스투에서 데이타 손실 없이 파티션 나누는 방법' on the current page를 참고하기 바란다.

GNU/리눅스는 최소한 하나의 파티션을 필요로 한다. 하나의 파티션에 운영체제와 응용 프로그램, 개인 파일 모두를 저장할 수 있다. 대부분의 사람은 스왑 파티션도 필수라고 생각한다. 그러나 엄밀하게 따지면 이것은 사실이 아니다. "스왑"은 운영체제가 임시로 사용하는 공간이다. 즉, 비싼 메모리 대신에 값싼 하드디스크를 "가상 메모리"로 이용하는 것이다. 스왑을 하나의 파티션으로 분리하면 리눅스르 이를 훨씬 더 효율적으로 이용할 수 있다. 보통의 파일을 스왑으로 사용하게 할 수도 있지만 추천하고 싶지는 않다.

대부분의 사람들은 GNU/리눅스에 하나 이상의 파티션을 할당하고 있다. 여러개의 파티션에 파일시스템을 나누어 두는 두 가지 이유가 있다. 첫째는 안전성 때문이다. 어쩌다 파일시스템이 엉킨다 하더라도 대개는 해당 파티션 하나에만 국한된다. 따라서 (보관하고 있던 백업을 이용하여) 해당 부분만 교체하면 된다. 최소한 "루트 파티션" 만이라도 따로 만드는 것을 고려해야 한다. 여기에는 시스템의 가장 기본적인 것이 포함된다. 따라서 다른 파티션이 엉킨다고 하더라도 부팅하여 시스템을 보수하는 것이 가능하다. 이렇게 하므로서 시스템을 다시 설치하는 번거로움을 피할 수 있다.

두 번째 이유는 실제 업무와 관련될 때 더욱 중요하다. 그러나 실제로는 컴퓨터 사용의 문제이다. 뭔가가 통제 불능 상태에 빠져 디스크 공간을 잡아먹고 있다고 가정하자. 그리고 그 프로세스가 하필 루트의 권한을 갖고 있다면 (하드 디스크 공간 중 일정량은 일반 사용자가 이용할 수 없고 오로지 루트만이 이용할 수 있다), 어느 순간 디스크 공간이 꽉 차버릴 수도 있다. 이와 같은 상황은 좋지 않다.

왜냐하면 운영체제는 일을 하기 위해서 (스왑 공간 이외에) 실제 파일을 사용해야 하기 때문이다. 이러한 상황의 원인이 내부에만 있는 것은 아니다. 예를 들어, 계속되는 스팸 메일로 인하여 쉽사리 파티션이 차버릴 수 있다. 더 많은 파티션을 이용하면 이러한 많은 문제로부터 시스템을 보호할 수 있다. 다시 메일을 예로 든다면, `/var/spool/mail`을 위하여 따로 파티션을 하나 만들어 놓으면 스팸 메일 공격을 받더라도 시스템 주요 부분은 작동할 것이다.

LBA 모드도 사용하지 않고 오버레이 드라이버도 사용하지 않는 고용량의 IDE 디스크를 갖고 있다면 파티션을 피할 수 없다. 이 경우에, 루트 파티션을 처음 1024 실린더 안쪽 (대개 524 메가바이트 근처)에 두어야 하기 때문이다.

여러개의 파티션을 사용하려 할 때의 곤란한 점은 무엇이 필요하게 될지 미리 알기 어렵다는 것이다. 파티션을 너무 작게 만들면 시스템을 다시 설치해야 하거나 또는 빈 공간을 만들기 위해 데이타를 여기저기로 옮기는 일을 반복적으로 할 수도 있다. 한편, 파티션을 너무 크게 만들면, 다른 곳에 사용할 수 있는 공간을 쓸데 없이 낭비하는 꼴이 될 수 있다. 비록 요즈음 하드 디스크 가격이 싸다고는 하지만, 돈을 낭비할 이유는 없지 않은가?

제 2 절 시스템 사용 계획 세우기

만들고자 하는 시스템의 종류를 결정하는 것은 중요하다. 시스템의 종류에 따라 필요한 하드 디스크 공간이 결정되고 파티션 설계에도 영향을 미친다.

데비안에서는 사용자 편의를 위하여 몇 가지 "프로파일"을 준비해 놓고 있다 ('프로파일의 선택과 설치' on page 35를 참고하기 바란다). 프로파일이란 간단히 말해서, 패키지 선택을 좀 더 쉽게 해주는 것이라고 할 수 있다. 하나의 프로파일을 선택하면 해당하는 수많은 패키지가 자동으로 설치된다.

각각의 프로파일에는 필요로 하는 하드 디스크 공간이 있다. 프로파일을 이용하지 않는다 하더라도 이러한 논의는 의미가 있다. 왜냐하면 이러한 논의를 통하여 몇 개의 파티션이 필요한지 그리고 각 파티션의 크기는 어느 정도로 할지에 대한 감을 잡을 수 있기 때문이다.

선택 가능한 몇 가지 프로파일과 각각의 프로파일이 요구하는 하드 디스크 용량에 대해 아래에 기술하였다.

Server_std (소형 서버) 소형 서버에 대한 프로파일. 계정 사용자의 작업이 별로 없는 경량형 서버에 적당하다. 기본적으로 FTP 서버, 웹 서버, DNS, NIS, POP 서버가 포함되어 있다. 대략 50MB 가량이 필요하다. 물론 이 용량은 단지 소프트웨어 설치에 필요한 공간이며 제공할 서비스에 필요한 데이타를 위한 공간은 포함되어 있지 않다.

Dialup (표준 데스크탑) 표준 데스크탑에 대한 프로파일. X 윈도우 시스템, 그래픽 프로그램, 사운드, 에디터 등을 포함하고 있다. 필요한 공간은 약 500MB 가량이다.

Work_std (경량형 컴퓨터) 경량형 컴퓨터에 대한 프로파일. X 윈도우 시스템 또는 X용 프로그램이 포함되어 있지 않다. 랩탑 또는 모빌 컴퓨터에 적당하다. 필요한 공간은 약 140MB 가량이다. (저자의 경우, 랩탑을 매우 단순하게 구성하였는데 그기에는 X11이 포함되어 있으며 필요한 공간은 100MB 보다 작았다.)

Devel_comp (개발자용 컴퓨터) 개발자용 도구가 모두 설치된 데스크탑에 대한 프로파일. Perl, C, C++ 등이 포함되어 있다. 필요한 공간은 대략 475MB 가량이다. 다른 목적을 위하여 X11 과 몇가지 패키지를 추가로 설치한다면 800MB 가량이 필요할 것이다.

위에서 언급한 하드 디스크 용량은 필요한 소프트웨어 설치에 필요한 공간만을 이야기 한 것이다. 즉 사용자 파일, 메일, 데이타 등을 위한 공간은 포함하지 않았다는 것을 명심하기 바란다. 여러분의 파일과 데이타를 저장할 공간을 항상 고려해야 한다. 특히 주의할 것은, 데비안의 `/var` 파티션에는 여러가지 정보가 저장된다는 것이다. 그 중에서도 `dpkg` 파일들이 (설치된 모든 패키지에 대한 정보를 가지고 있음) 차지하는 공간은 20MB를 쉽게 넘는다. 일반적으로 `/var`에 최소한 50MB는 할당해야 한다.

2.1 PC 디스크 제한 사항

일반적으로 디스크 파티션에는 바이오스에 의한 추가적인 제약이 따른다. 하나의 하드 디스크에 존재할 수 있는 "기본(primary)" 와 "논리(logical)" 파티션의 개수에 제한이 있다. 또한 부팅 가능한 드라이버에 대한 제한도 있다. 여기서는 여러분에게 도움이 될 내용을 간단히 살펴보겠다. 더 자세한 것은 Linux Partition HOWTO (`http://www.linuxdoc.org/HOWTO/mini/Partition.html`)를 참고하기 바란다.

"기본" 파티션은 원래 PC 디스크를 위해 설계된 파티션이다. 그러나 하나의 하드 디스크에 단지 네 개 만이 있을 수 있다. 이러한 제한을 벗어나기 위해 "확장(extended)" 또는 "논리" 파티션이 고안되었다. 기본 파티션을 확장 파티션으로 설정한 다음, 다시 여러개의 논리 파티션으로 나눌 수 있다. 만들 수 있는 논리 파티션의 개수에는 제한이 없으나 확장 파티션은 하나의 디스크에 오로지 하나 만이 있을 수 있다.

리눅스는 하나의 디스크에 있을 수 있는 파티션의 개수를 제한하고 있으며, SCSI 디스크의 경우 모두 15 개의 파티션 (3개의 기본 파티션, 12 개의 논리 파티션)이 가능하고, IDE 디스크는 모두 63 개의 파티션 (3 개의 기본 파티션, 60 개의 논리 파티션)이 가능하다.

마지막으로 알아두어야 할 사항은, 부트 파티션 (즉, 커널 이미지를 포함하고 있는 파티션)은 디스크의 처음 1024 실린더 안에 있어야 한다는 것이다. 일반적으로 루트 파티션은 또한 부트 파티션이기 때문에 루트 파티션이 처음 1024 실린더 안에 있는지 확인해야 한다.

고용량 디스크의 경우, LBA 변환 모드와 같은 실린더 변환 기술을 사용할 필요가 있으며 바이오스에서 설정할 수 있다. 고용량 디스크 사용에 관한 더 자세한 정보는 Large Disk HOWTO (`http://www.linuxdoc.org/HOWTO/mini/Large-Disk.html`)를 참고하기 바란다. 실린더 변환 기술을 사용한다면 1024 번째 실린더의 변환된 값 안에 부트 파티션이 있어야 한다.

제 3 절 리눅스에서 사용하는 장치명

리눅스에서 사용하는 디스크와 파티션의 이름은 다른 운영체제의 그것과는 다르다. 파티션을 만들거나 마운트 하려면 리눅스에서 사용하는 이름을 알아야 한다. 다음은 이름을 붙이는 기본적인 방법이다.

- 첫 번째 플로피 드라이브는 "/dev/fd0"
- 두 번째 플로피 드라이브는 "/dev/fd1"
- 첫 번째 SCSI 디스크 (정해진 SCSI ID에 따름)는 "/dev/sda"
- 두 번째 SCSI 디스크 (정해진 SCSI ID에 따름)는 "/dev/sdb" 등등

- 첫 번째 SCSI 시디롬은 "/dev/scd0", 또한 "/dev/sr0"로도 알려져 있음
- 첫 번째 IDE 콘트롤러에 연결된 마스타 디스크는 "/dev/hda"
- 첫 번째 IDE 콘트롤러에 연결된 슬래이브 디스크는 "/dev/hdb"
- 두 번째 IDE 콘트롤러에 연결된 마스타 디스크와 슬래이브 디스크는 "/dev/hdc" 와 "/dev/hdd"로 각각 명명된다. 더 새로운 IDE 콘트롤러는 두 개의 채널을 가지고 있으며 두 개의 콘트롤러와 같은 작용을 한다.
- 첫 번째 XT 디스크는 "/dev/xda"
- 두 번째 XT 디스크는 "/dev/xdb"

디스크의 이름에 십진수를 붙여서 파티션을 구별한다. 예를 들어, "sda1" 과 "sda2"는 첫번째 SCSI 디스크에 있는 첫번째, 두번째 파티션을 각각 나타낸다.

실질적인 예를 들어 보자. 어떤 시스템에 두 개의 SCSI 디스크가 있으며, 그 중 하나의 SCSI ID 주소가 2 이고 나머지의 SCSI ID 주소는 4 라고 하자. 주소 2의 첫 번째 SCSI 디스크는 "sda"가 될 것이고, 두 번째 디스크는 "sdb"가 될 것이다. "sda"에는 세 개의 파티션이 있으며 각 파티션의 이름은 "sda1", "sda2", "sda3"가 된다. 이는 "sdb" 디스크에도 똑 같이 적용된다.

SCSI 호스트 어댑터 (즉, 콘트롤러)가 두 개라면, 디스크들의 순서에 혼란이 있을 수 있다. 부트 메세지를 잘 관찰하면 이 문제를 해결할 수 있다. 물론 각각의 디스크 모델명 정도는 알고 있어야 한다.

리눅스에서는 디스크 이름에 1 부터 4 까지의 숫자를 붙여서 기본 파티션을 표시한다. 예를 들어, 첫번째 IDE 디스크의 첫 번째 기본 파티션은 `/dev/hda1`과 같이 나타낸다. 논리 파티션은 5 부터 시작한다. 따라서 "hda" 디스크의 첫 번째 논리 파티션은 `/dev/hda5`로 표시한다. 확장 파티션 (즉, 논리 파티션으로 이루어진 기본 파티션) 그 자체는 사용할 수 없다. 이것은 SCSI 디스크와 IDE 디스크 모두에 적용된다.

제 4 절 추천하는 파티션 설계

되도록이면 루트 파티션과 `/usr` 파티션은 따로 분리하도록 한다. 루트 파티션은 좀 작게하고 `/usr` 파티션은 보다 크게 만든다. 대부분의 경우에는 이것으로 충분하다. 특히, 하드 디스크가 하나 뿐이고 그 크기도 작은 경우에 위와 같은 파티션 설계가 적당하다. 왜냐하면, 이 경우에는 여러개의 파티션으로 나누는 것이 결과적으로는 공간 낭비를 초래할 수 있기 때문이다.

데비안에서 패키지 형태로 배포하지 않는 프로그램을 많이 설치할 계획이라면 `/usr/local` 파티션이 따로 필요할 수도 있다. 시스템을 메일 서버로 활용할 계획이라면 `/var/spool/mail`에 따로 파티션을 할당해 줄 수도 있다. `/tmp`에 20 에서 32MB 정도의 파티션을 따로 할당하는 것도 좋은 생각이다. 많은 사람들에게 계정을 발급해 줄 계획이라면, `/home`에 큰 파티션을 하나 주는 것이 좋다. 일반적으로, 컴퓨터를 어떻게 사용하느냐에 따라 파티션 설계 방법이 달라지게 마련이다.

매우 복잡한 시스템의 경우는 Multi Disk HOWTO (`http://www.linuxdoc.org/HOWTO/Multi-Disk-HOWT` `html`)를 참고하기 바란다. 여기에는 서버를 설치하고 설정하려는 사람들과 ISP의 관심을 끌 만한 깊이 있는 정보를 담고 있다.

스왑 파티션 크기와 관련해서는 여러가지 관점이 있다. 경험적으로 볼 때 스왑의 크기를 시스템의 메모리 크기와 같게 하면 원만하게 작동한다. 그러나 대부분의 사용자는 스왑의 크기를 64MB 이상으로 하고 있으며, 그렇다고 해서 별 다른 의의를 제기할 생각은 없다. 대부분의 경우에 스왑의 크기가 16MB 보다 작아서는 안된다. 물론 이러한 규칙에는 예외가 있다. 256MB의 메모리를 가진 시스템에서 10000 개의 연립 방정식을 풀려고 한다면 기가바이트 크기의 스왑도 모자랄 수 있다.

리눅스는 하나의 스왑 파티션에 128 메가바이트 이상의 스왑은 사용하지 않는다. 그 이상의 스왑이 필요하면 설치가 끝난 후에 여러개의 스왑 파티션을 만들고 `/etc/fstab`을 수정하면 가능하다. 그러나 큰 스왑 공간이 필요하면 다른 디스크에 스왑 파티션을 분산시켜야 할 것이다 (”스핀들”이라고 한다). 또는 이러한 제한이 완화된 최근의 리눅스 커널 (2.2 또는 그 이상)을 이용할 수도 있다 (시스템에 어떤 변화를 주어야 할 지도 모르니 주의 해야 한다).

제 5 절 파티션 예

저자가 집에서 사용하는 컴퓨터에는 32MB의 램과 1.7GB의 IDE 디스크 (`/dev/hda`)가 있다.. `/dev/hda1`에 500MB를 할당하여 다른 운영체제를 설치하였으며 (거의 사용하지 않기 때문에 200MB 정도로 만들어야 했다), `/dev/hda3`에 32MB 크기의 스왑 파티션을 할당하고 나머지 (`/dev/hda2`에 1.2GB)를 리눅스 파티션으로 만들었다.

제 6 절 설치하기 전에 파티션 나누기

파티션을 할 수 있는 때가 두 번 있다. 즉, 데비안을 설치하기 전 그리고 데비안을 설치하는 동안. 컴퓨터에 데비안 만을 설치할 계획이라면 설치 과정 중에 파티션 하면 된다 (“‘Partition a Hard Disk”’ on page 35). 그러나 하나 이상의 운영체제를 설치할 계획이라면 해당 운영체제에서 파티션을 만들어 놓아야 한다.

데비안을 설치하기 전에 다른 운영체제에서 파티션하는 것과 관련된 내용을 설명하겠다. 다른 운영체제에서는 파티션 이름을 어떻게 붙이는지 그리고 리눅스에서는 그 파티션 이름을 어떻게 붙이는지 (‘리눅스에서 사용하는 장치명’ on page 15) 알고 있어야 한다.

6.1 도스 또는 윈도즈에서 파티션 나누기

하드 디스크에 이미 존재하는 FAT 또는 NTFS 파티션을 다루고자 한다면 도스 또는 윈도즈에서 제공하는 도구를 이용하거나 아래의 설명을 참고하기 바란다. 그런 것이 아니라면 도스 또는 윈도즈에서 파티션 할 필요는 실제로 없다. 리눅스에서 제공하는 도구를 이용하는 것이 더 좋을 것이다.

제 7 절 도스, Win-32 또는 오에스투에서 데이타 손실 없이 파티션 나누는 방법

일반적으로 도스 (윈도즈 3.1 포함), Win32 (윈도즈 95, 98, NT) 또는 OS/2가 이미 설치되어 있는 하드 디스크에 데비안을 설치하려고 하는 경우가 대부분이다. 물론 기존의 시스템에 손상을 주면 안될 것이다. ‘기초 지식’ on page 15에서 설명했다시피 사전에 어떤 준비를 하지 않는 이상, 기존의 파티

션 크기를 줄이면 그곳에 있던 데이타에 손상이 가는 것은 거의 확실하다. 이제 설명할 방법은 데이타의 완벽한 보호를 보장해 주지는 않지만 실제 상황에서 매우 잘 작동한다. 예방책으로 *백업*을 해두는 것이 좋다.

계속하기 전에, 디스크를 어떻게 나눌 것인지 결정해야 한다. 여기서는 하나의 파티션을 두 조각으로 나누는 방법 만을 설명한다. 한 조각은 기존의 운영체제를 위해 그리고 다른 한 조각은 데비안을 위한 것이다. 데비안을 설치하는 동안에 그 한 조각을 여러분이 원하는 용도 (즉, 스왑 또는 파일시스템)로 사용할 기회가 있을 것이다.

파티션 정보를 바꾸기 전에, 모든 데이타를 파티션의 첫 부분으로 옮겨 놓음으로서 손상을 피할 수 있다. 파티션의 끝 부분에 파일이 있으면 얻을 수 있는 빈 공간이 줄어들기 때문에 데이타를 이동 시킨 후 다시 파티션 하는 사이에 파일을 만드는 일이 없도록 주의해야 한다.

첫번째로 필요한 것은 `fips`이다. 가까운 데비안 ftp 사이트의 `tools/` 디렉터리에서 찾을 수 있다. 압축을 풀면 `RESTORRB.EXE`, `FIPS.EXE`, `ERRORS.TXT`이 나오는데 이들을 부팅가능한 플로피 디스크에 복사한다. 도스에서 부팅 디스크를 만들려면 `sys a:` 명령을 실행시키면 된다. `fips`에는 읽어 볼만한 매우 훌륭한 문서가 포함되어 있다. 디스크 압축 드라이버 또는 디스크 매니저를 사용한다면 분명히 읽어볼 필요가 있다. 디스크를 나누기 *전에* 읽어보기 바란다.

그 다음으로 해야 할 것은 모든 데이타를 파티션 시작 부분으로 옮기는 것이다. 도스 6.0 이상에서 제공하는 `defrag`를 이용하여 쉽게 할 수 있다. 같은 역할를 하는 다른 프로그램에 어떤 것들이 있는지 알고 싶으면 `fips`에 포함되어 있는 문서를 참고하기 바란다. 윈도즈95를 사용할 경우, 도스에서 `defrag`를 실행시켜서는 절대 안되며 윈도즈에서 `defrag`를 실행시켜야 한다. 윈도즈95 이상에서는 긴 파일 이름을 지원하기 위해 VFAT을 사용하며 도스는 이를 알지 못하기 때문이다.

디스크 조각 모음을 완료한 후 (용량이 큰 디스크의 경우 완료될 때 까지 오랜 시간이 걸릴 수도 있다), `fips`를 포함하고 있는 부팅 플로피 디스크를 이용하여 재부팅해야 한다. 그 후 `a:\fips`를 타이핑하여 실행 시키고 이어서 화면에 나타나는 지시에 따르면 된다.

`fips`를 이용한 파티션 나누기가 여의치 않다고 하더라도 그 외의 다른 많은 파티션 관리자를 이용할 수 있다.

제 8 절 도스에 할당할 파티션에 대해서

리눅스에서 제공하는 도구를 이용하여 도스에 할당할 파티션을 만들거나 도스 파티션의 크기를 바꾼 경우에 그 파티션으로 인한 문제점을 경험한 사람들이 많다. 예를 들어, 도스 또는 윈도즈에서 수행 능력이 떨어진다거나, `scandisk`와 문제를 일으킨다거나, 또는 다른 이해 못할 에러 등이 그것이다.

일단은, 도스용 파티션을 만들거나 크기를 변화시킨 후, 그 파티션의 처음 몇 섹터에 제로를 써 넣는 것이 좋다. 도스에서 `format` 명령을 실행하기 전에 아래의 예와 같은 명령을 리눅스에서 실행시키면 된다.

```
dd if=/dev/zero of=/dev/hdXX bs=512 count=4
```

제 5 장

데비안 설치에 이용될 수 있는 방법들

데비안을 처음 설치 할 때 거쳐야 할 몇 가지 단계가 있다. 그 순서는:

1. 설치 시스템으로 부팅
2. 시스템 초기화 설정
3. 베이스 시스템 설치
4. 새로이 설치된 베이스 시스템으로 부팅
5. 나머지 부분을 설치

각 단계에 적용할 수 있는 방법은 여러가지 이다. 플랫폼이 다르면 방법도 달라진다. 여기에서는 Intel x86에 적용할 수 있는 방법만을 설명할 것이다.

첫 번째 단계인 데비안 설치 시스템으로 부팅하기 위해서는 다음 매체를 이용할 수 있다.

- Rescue Floppy
- 부팅 가능한 시디롬
- 다른 운영체제를 실행시키는 부트 로더

이들 각각에 대해서는 '초기 부팅 매체 선택하기' on this page에 설명해 놓았다. 때로는 하드웨어 문제로 인하여 첫번째 단계가 가장 어려울 수도 있다. 그래서 관련된 내용을 '설치 시스템으로 부팅하기' on page 30에 설명해 놓았다.

일단 리눅스로 부팅한 후에는 `dbootstrap` 프로그램이 실행되면서 여러분을 두 번째 단계 (시스템 초기화 설정)로 인도할 것이다. 이 단계에 대한 자세한 내용을 '`dbootstrap`을 이용한 시스템 초기화 설정' on page 35에서 설명할 것이다.

데비안이 독자적으로 돌아가기 위해 필요한 최소한의 핵심적인 패키지를 모아 놓은 것이 "데비안 베이스 시스템"이다. 일단 베이스 시스템을 설치하면 "혼자서 일어 설 수 있게 된 것"이다. 데비안 베이스 시스템을 설치할 수 있는 매체의 종류도 여러 가지이다. 플로피 디스크, 하드 디스크, 시디롬,

NFS 서버를 이용할 수 있다. dbootstrap에 의해 설치가 이루어 진다. “‘Install the Base System’” on page 35에서 설명할 것이다.

마지막 단계는 데비안 시스템의 나머지 부분을 설치하는 것이다. X 윈도우 시스템, 에디터, 쉘, 개발 도구 등, 여러분이 실제로 사용할 응용 프로그램과 문서가 여기에 포함된다. 데비안의 나머지 부분을 설치하는 것은 시디롬을 이용하거나 또는 (HTTP, FTP, NFS를 경유하여) 데비안 아카이브의 미러 사이트를 이용할 수도 있다. 이 때 여러분은 deselect 또는 apt-get과 같은 데비안 패키지 표준 관리 도구를 사용하게 된다. ‘나머지 설치 과정’ on page 35를 참고하기 바란다.

각 단계에서 사용될 매체가 반드시 같을 필요는 없다는 것을 기억하기 바란다. 즉, Rescue Floppy를 이용하여 부팅한 다음, 베이스 시스템은 NFS를 이용하여 설치하고, 나머지 부분 설치는 인터넷을 이용할 수도 있다는 것이다. 아카이브에서 데비안 시스템을 내려 받는 것이 가능한 경우라면, 베이스 시스템 설치 (3 단계)까지는 플로피를 이용하고, 나머지 부분은 인터넷을 경유하여 설치하는 것이 일반적이다.

여러가지 설치 방법과 필요한 파일에 대하여 아래에서 설명할 것이다. 어떤 파일을 사용할 지, 설치 매체를 준비하려면 어떻게 해야 하는지는 여러분이 선택한 설치 방법에 따라 달라진다.

제 1 절 설치 매체 선택하기

먼저, 설치 시스템으로 부팅하기 위해 어떤 매체를 사용할 지 결정해야 한다. 그 다음, 베이스 시스템을 설치할 때 어떤 방법을 이용할 것인지 선택해야 한다. 위에서 설명하였듯이 이러한 결정은 상호 독립적으로 이루질 수 있다.

1.1 초기 부팅 매체 선택하기

다음 방법을 이용하여 설치 시스템으로 부팅할 수 있다: 플로피, 부팅 가능한 시디롬, 또는 non-Linux 부트 로더.

대부분의 플랫폼에 대하여 플로피로 부팅하는 것을 지원한다. 플로피로 부팅하는 것에 대한 설명은 ‘Booting from Floppies’ on the page before에서 할 것이다.

설치하기 가장 쉬운 방법 중의 하나가 시디롬으로 부팅하는 것이다. 좀 운이 없다거나, 시디롬에 있는 커널이 제대로 돌아가지 않는다면 다른 방법을 이용해야 할 것이다. 시디롬으로 설치하는 방법을 ‘시디롬을 이용하여 설치하기’ on the preceding page에서 설명할 것이다.

이미 설치되어 있는 운영 체제로 부팅하여 설치하는 것도 편리한 방법이다. 일부 시스템의 경우에는 이 방법 이외의 다른 설치 방법은 없다. ‘하드디스크를 이용하여 설치하기’ on the page before에 이 설치 방법이 설명되어 있다.

1.2 베이스 시스템 설치에 이용할 매체 선택

여러가지 매체를 이용하여 베이스 시스템을 설치할 수 있다: 플로피 (‘플로피로 베이스 시스템 설치하기’ on the preceding page), 시디롬 (‘시디롬을 이용하여 설치하기’ on the page before), NFS 서버 (‘NFS를 이용하여 설치하기’ on the preceding page), 하드 디스크 (‘하드디스크를 이용하여 설치하기’ on the page before). 이 중에서 가장 편리한 방법을 선택하면 된다.

제 2 절 설치 시스템을 구성하는 파일들에 대한 설명

disks--i386 디렉터리의 파일들에 대하여 간단히 설명하고자 한다. 여기에 있는 모든 파일을 내려 받을 필요는 없으며 부팅 시스템과 베이스 시스템 설치에 어떤 매체를 사용하느냐에 따라 내려 받을 파일이 결정된다.

대부분의 파일은 플로피 디스크 이미지 이며, 필요한 플로피 디스크를 만들기 위해 해당 이미지 파일을 디스크에 써 넣는다. 플로피 디스크의 크기 (1.4MB, 1.2MB, 720KB)에 따라 해당 이미지 파일이 달라진다. 플랫폼에 따라 이용할 수 있는 플로피 디스크의 크기가 달라진다 (예를 들어, Atari에서는 720KB의 디스크를 사용한다). 1.4MB 드라이브에 맞는 이미지 파일은 그 이름 안에 '14'가 들어있다. 1.2MB 이미지 파일의 이름에는 '12'가 들어있고, 720KB 드라이브의 이미지 파일 이름에는 '72'가 들어있다.

여러분의 컴퓨터가 네트웍에 연결되어 있고, 웹 브라우저를 통하여 이 문서를 보고 있다면, 필요한 이미지 파일 이름을 선택하므로서 그 파일을 가져 올 수 있을 것이다. 파일을 내려 받기 하려면, 브라우저에 따라 취해야 할 동작이 다르다. 예를 들어, 넷스케이프에서는 쉬프트 키를 누른 상태에서 마우스를 클릭하면 파일을 직접 내려 받을 수 있다. 이와 같이 이 문서의 URL을 선택하여 파일을 내려 받을 수도 있고, ftp://ftp.debian.org/debian/dists/slink/main/disks-i386/current/에서 내려 받을 수도 있다. 또는 데비안 미러 사이트 (http://www.debian.org/distrib/ftplist)를 이용할 수도 있다.

resc1440.bin, resc1440tecra.bin, resc1200.bin – Rescue Floppy 이미지 이 파일들은 Rescue Floppy 디스크 이미지이다. Rescue Floppy는 초기화 설정에 사용되며, 또한 어떤 이유로 하드디스크로부터 부팅이 불가능하다거나 하는 등의 응급 상황에도 이용된다. 그래서 데비안을 설치할 때 플로피 디스크를 이용하지 않는다 하더라도 Rescue Floppy는 하나 만들어 두는 것이 좋다.

5.25 인치 플로피 드라이브를 사용한다면 이름에 '1200'이 들어간 파일을 사용하고, 3.5 인치 플로피를 이용한다면 '1400'이 들어간 파일을 이용하면 된다. 또한 root.bin 파일이 필요할 것이다.

'tecra' 이미지에 포함되어 있는 커널은 다른 Rescue Floppy 이미지의 커널과는 약간 다르며 표준 이미지 디스크를 이용하면 문제가 발생하는 사람들을 위한 것이다.

drv1440.bin, drv1440tecra.bin, drv1200.bin – Drivers Floppy 이미지 이 파일들은 Drivers Floppy 디스크 이미지이다. 초기 부팅에 아무런 영향을 주지 않기 때문에 커널에 직접 포함될 필요가 없는 모든 종류의 하드웨어에 대한 커널 모듈 (또는 드라이버)가 여기에 들어있다. 설치 과정 중에 필요한 드라이버를 선택하는 단계가 있다.

특별한 Rescue Floppy 이미지를 사용한다면, 그것에 해당하는 Drivers Floppy 이미지를 사용할 필요가 있다.

base2_1.tgz, or base14-1.bin, base14-2.bin, base14-3.bin, base14-4.bin, base14-5.bin, ba 리눅스 파티션에 설치될 베이스 시스템을 포함하고 있는 파일들이다. 전체 패키지 중에서 꼭 필요한 최소한의 패키지로 구성되어 있다. 시디롬, 하드디스크, NFS를 이용하여 베이스 시스템을 설치한다면 base2_1.tgz 파일을 이용하면 된다.

root.bin – 루트 이미지 부팅할 때 메모리에 적재될 임시 파일시스템의 이미지이다. 하드 디스크 또는 시디롬을 이용하여 설치할 때 사용된다.

또한 무슨 이유로 인하여 루트 파일 시스템이 Rescue Floppy에 들어 가지 않을 때 사용된다. 예를 들어, 5.25 플로피를 부팅에 사용한다면 `root.bin` 파일이 필요할 것이다.

`lowmem.bin` – 로우 메모리 부트 이미지 5MB 보다 메모리가 적은 경우에 이용하는 부트 플로피이다. 메모리가 적은 경우의 설치 방법에 대한 것은 '시스템에 메모리가 적은 경우의 설치 방법' on page 21를 참고하기 바란다.

`lowmemrd.bin` – 로우 메모리 루트 파일 시스템 이미지 위와 같은 경우에 사용되는 루트 파일 시스템 이미지. 특히 이 방법은 5.25 인치 드라이버가 있는 경우에 사용된다. `lowmem.bin` 파일이 디스크에 들어가지 않기 때문이다.

`rawrite2.exe` 플로피 디스크 이미지 파일을 플로피에 써 넣기 위한 도스용 유틸리티이다. 이미지를 플로피에 복사하는 것이 아니라, 이 유틸리티를 이용하여 써 넣어야 한다.

`loadlin.exe` – 리눅스 부트 로더 (Linux boot loader) 도스 파티션 또는 시디롬으로 설치한다면 이 부트 로더가 필요하다. '도스 파티션을 이용하여 설치하기' on page 21를 참고하기 바란다.

`install.bat` 도스에서 데비안 설치를 시작하려면 이 도스용 배치 파일을 실행시키면 된다. 하드디스크 또는 시디롬으로 설치하는 경우에 이 배치 파일을 이용한다. '도스 파티션을 이용하여 설치하기' on page 21를 참고하기 바란다.

`linux` – 커널 이미지 하드디스크 또는 시디롬으로 설치하는 경우에 사용되는 리눅스 커널 이미지이다. 파일을 내려 받기 하려면, 브라우저에 따라 취해야 할 동작이 다르다. 예를 들어, 넷스케이프에서는 쉬프트 키를 누른 상태에서 마우스를 클릭하면 파일을 직접 내려 받을 수 있다.

`install.txt`, `install.html` – 설치 설명서 여러분이 지금 읽고 있는 파일이다. 텍스트와 html 문서가 있다.

`fdisk.txt` `cfdisk.txt` 하드디스크 파티션 나누기에 사용할 프로그램에 대한 설명서.

`basecont.txt` 베이스 시스템 내용에 대한 목록.

`md5sum.txt` 이진 파일들의 MD5 체크섬 목록. `md5sum` 프로그램이 있다면, 파일이 엉키지 않았는지 `md5sum -v -c md5sum.txt` 명령을 실행시켜 보면 알 수 있다.

제 3 절 하드디스크를 이용하여 설치하기

이미 설치되어 있는 운영체제로 부팅하여 설치하고자 할 수도 있다. 또는 다른 방법을 이용하여 설치 시스템으로 부팅한 후에 베이스 시스템은 하드디스크를 이용하여 설치하고자 할 수도 있다.

3.1 도스 파티션을 이용하여 설치하기

도스 파티션을 이용하여 데비안을 설치하는 것이 가능하다. 두 가지 선택을 할 수 있다. 하나는 설치 과정에 플로피 디스크를 전혀 사용하지 않는 것이고, 다른 하나는 Rescue Floppy를 이용하여 부팅한 후에 베이스 시스템은 하드디스크를 이용하여 설치하는 것이다.

플로피 디스크를 전혀 사용하지 않고서 설치를 하려면 다음 설명을 따라하기 바란다.

1. 가장 가까운 데비안 FTP 미러 사이트로부터 아래의 파일들을 내려 받아서 도스 파티션의 임의의 디렉터리에 저장한다: `resc1440.bin`, `drv1440.bin`, `base2_1.tgz`, `root.bin`, `linux`, `install.bat`, `loadlin.exe`.
2. 메모리에 드라이버가 적재되지 않도록 도스 (윈도즈가 아님)로 부팅한다. 이렇게 하려면, 부팅 과정의 적당한 순간에 *F8*을 눌러야 할 것이다.
3. `install.bat`를 실행시킨다.
4. 이하의 내용은 무시하고 곧장 '설치 시스템으로 부팅하기' on page 30로 넘어간다.

부팅은 플로피로 하고, 베이스 시스템 설치는 도스 파티션을 이용하고자 한다면, '플로피 디스크에 디스크 이미지 써넣기' on page 21의 설명대로 Rescue Floppy와 Drivers Floppy를 만들어야 한다. 그리고 `base2_1.tgz` 파일을 내려 받아서 도스 파티션에 둔다.

3.2 리눅스 파티션을 이용하여 설치하기

ext2 또는 Minix 파티션을 이용하여 데비안을 설치할 수 있다. 현재 설치되어 있는 리눅스 시스템을 데비안으로 교체하고자 할 경우에 이 방법이 적당하다.

데비안을 설치하기 위해 이용할 파티션과 데비안이 실제로 설치될 파티션이 같아서는 안된다는 것을 명심해야 한다.

기존의 리눅스 파티션을 이용하여 설치하려면, 아래 설명을 따라하기 바란다.

1. 다음에 해당하는 파일들을 내려 받아서 리눅스 파티션에 저장한다. 크기가 가장 큰 파일을 내려 받으면 된다.
   - Rescue Floppy 이미지
   - Drivers Floppy 이미지
   - `base2_1.tgz`
2. 하드디스크를 이용하여 설치할 때, 부팅은 다른 매체를 이용해도 된다. 아래의 설명은 플로피를 이용하여 부팅하는 경우를 가정하였지만 다른 방법으로 부팅해도 된다.
3. '플로피 디스크에 디스크 이미지 써넣기' on page 21를 참고로하여 Rescue Floppy를 만든다. Drivers Floppy는 만들 필요가 없다.
4. 플로피 드라이브에 Rescue Floppy를 넣고, 컴퓨터를 재부팅 시킨다.
5. 이하의 내용은 무시하고 곧장 '설치 시스템으로 부팅하기' on page 30로 넘어간다.

제 4 절 시디롬을 이용하여 설치하기

시디롬으로 부팅하는 것이 가능하고 부팅 가능한 데비안 CD를 갖고 있다면, 플로피 디스크가 필요 없다.

'부팅 장치 선택' on page 10에서의 설명과 같이 하드웨어 설정을 한다.

그 후, 시디롬을 드라이브에 넣고 재부팅한다.

"tecra" 부트 이미지를 필요로 하는 시스템이라면 첫 번째 시디롬 보다는 두번째 시디롬을 넣고 재부팅해야 한다. 이하의 내용은 무시하고 곧장 '설치 시스템으로 부팅하기' on page 30로 넘어간다.

부팅 가능한 시디롬을 지원하지 않는 경우에는 도스로 부팅하여 시디의 `\(\backslash\)boot` 디렉터리에 있는 `boot.bat`를 실행시키면 된다. 이하의 내용은 무시하고 곧장 'dbootstrap을 이용한 시스템 초기화 설정' on page 35로 넘어간다.

시디롬으로 부팅할 수 없다 하더라도, 시디롬을 이용하여 베이스 시스템은 설치 할 수 있다. 일단 다른 방법을 이용하여 부팅한 다음, 베이스 시스템을 설치할 순서가 되면 설치 경로를 시디롬으로 지정하면된다. "'Install the Base System'" on page 35를 참고하기 바란다.

제 5 절 NFS를 이용하여 설치하기

이 설치 방법의 특성으로 인해 베이스 시스템의 설치에만 이용할 수 있다. 응급 복구 디스크와 드라이버 디스크를 만들어 둘 필요가 있다. NFS를 이용하여 베이스 시스템을 설치하려면 'dbootstrap을 이용한 시스템 초기화 설정' on page 35에서 설명하고 있는 설치 과정을 거쳐야 한다. 에더넷 카드 모듈을 설정해야 하고, 또한 NFS를 이용하려면 해당 파일 시스템 모듈도 설정해 주어야 한다.

설치할 베이스 시스템의 경로를 `dbootstrap`이 물어보면 ("'Install the Base System'" on page 35 참고) NFS를 선택하고 지시를 따르면 된다.

제 6 절 플로피로 설치하기

빠르거나 선호되는 방법은 아니지만, 시스템이 지원해 준다면 플로피로 설치하는 것도 괜찮다. 플로피로 설치하려면 여러 단계를 거쳐야 한다.

6.1 Booting from Floppies

플로피로 부팅하는 과정은 간단하다. 먼저 RESCUE-FLOPPY; 이미지와 Drivers Floppy 이미지를 내려 받은 후, 이를 플로피에 복사한다. 어떤 경우에는 이용할 디스크 이미지를 골라야 한다. '설치 시스템을 구성하는 파일들에 대한 설명' on page 21에서 설명한 내용이 도움이 될 것이다. 복사하는 방법은 '플로피 디스크에 디스크 이미지 써넣기' on page 21를 참고하기 바란다.

필요하다면 Rescue Floppy를 수정할 수 있다. 'Rescue Floppy 커널 교체하기' on page 50를 참고하기 바란다.

제 7 절 시스템에 메모리가 적은 경우의 설치 방법

램의 양이 5MB 이하인 경우, 특별한 디스크 이미지를 (lowmem.bin 사용해야 할 필요가 있다. 모두 세개의 플로피가 필요할 것이다.

- Rescue Floppy
- Drivers Floppy
- 로우 메모리 부트 플로피

'플로피 디스크에 디스크 이미지 써넣기' on page 21의 설명을 참고로하여 필요한 디스크 이미지를 플로피에 써 넣는다. 5.25 인치 플로피 드라이브를 사용한다면 부트 플로피 대신에 로우 메모리 루트 디스크 (lowmemrd.bin)를 만들어야 한다. Rescue Floppy로 부팅한 후, 루트 디스크를 넣어라고 할 때 로우 메모리 루트 디스크를 넣도록 한다.

위의 경우가 아니라면, 로우 메모리 부트 플로피로 부팅하면 된다. 다음 진행 과정은 '설치 시스템으로 부팅하기' on page 30와 '메모리가 적은 시스템에서 부팅' on page 30를 참고하도록 한다.

제 8 절 플로피로 베이스 시스템 설치하기

주의: 플로피 디스크는 다른 매체에 비하여 신뢰성이 매우 떨이지기 때문에 플로피를 이용하여 베이스 시스템을 설치하는 것은 권하고 싶지 않은 방법이다. 하드디스크에 여분의 기존 파일시스템이 없는 경우에나 사용하는 방법이다.

아래의 순서대로 따라가면 된다:

1. 다음 디스크 이미지를 내려 받아야 한다 (각 파일에 대한 설명은 '설치 시스템을 구성하는 파일들에 대한 설명' on page 21를 참고하기 바란다):
   - a Rescue Floppy 이미지
   - a Drivers Floppy 이미지
   - 베이스 시스템 디스크 이미지, 즉, base14-1.bin, base14-2.bin, 등.
2. 필요한 이미지를 써 넣을 플로피 디스크를 충분히 준비한다.
3. '플로피 디스크에 디스크 이미지 써넣기' on page 21를 참고하여 이미지를 플로피에 써 넣는다.
4. Rescue Floppy를 플로피 드라이브에 넣고 컴퓨터를 재부팅 시킨다.
5. 이하의 내용은 무시하고 곧장 '설치 시스템으로 부팅하기' on page 30로 넘어간다.

제 9 절 플로피 디스크에 디스크 이미지 써넣기

디스크 이미지란, 어떤 플로피 디스크에 들어있는 전체 내용을 *있는 그대로의 (raw)* 형식으로 만든 파일이다. 일반적인 파일 복사 방법과 resc1440.bin과 같은 디스크 이미지를 플로피 디스크에 써넣는 방법은 다르다. 이미지 파일을 플로피에 *있는 그대로의 형태*로 써넣기 위해 사용되는 특별한 프로그램이 있다. 이미지 파일은 디스크의 내용을 있는 그대로 기록한 것이기 때문이다. 다시 말해서 이미지 파일을 플로피에 써 넣으려면 *섹터 복사*를 해야 하기 때문에 특별한 프로그램이 필요한 것이다.

이미지 파일을 플로피에 써넣는 방법은 플랫폼에 따라 다르다. 이제부터 각각의 방법을 설명할 것이다.

이용하는 방법에 상관없이, 이미지 파일을 플로피에 써넣은 후에는 그 플로피에 쓰기 방지 탭을 붙이는 것이 안전하다. 사소한 실수로 인하여 플로피를 못쓰게 되는 것을 피할 수 있다.

9.1 리눅스나 유닉스 시스템에서 써넣기

플로피 디스크 이미지 파일을 플로피 디스크에 써넣기 위해서 플로피 디스크 드라이브에 공 디스크를 넣은 후 다음과 같은 명령을 사용한다. 명령을 실행시키기 위해 루트의 권한이 필요할 수도 있다.

```
dd if=file of=/dev/fd0 bs=512 conv=sync ; sync
```

*file*은 플로피 디스크 이미지 파일들 중의 하나이다. /dev/fd0는 일반적인 플로피 디스크 장치의 이름으로 사용되는 것이다. 여러분이 사용하는 웍스테이션에서는 그 이름이 다를 수 있다 (솔라리스의 경우에는 /dev/fd/0이다). 유닉스 시스템의 경우, 플로피 디스크에 쓰기를 마치기 전에 프롬프트가 나타날 수 있다. 플로피 디스크 드라이브에서 디스크를 빼내기 전에 반드시 플로피 디스크 드라이브가 사용중임을 나타내는 램프가 꺼졌는지 그리고 드라이브가 완전히 멈추었는지 확인해야 한다. 일부 시스템이 경우,플로피 디스크를 빼기 위해 명령어를 입력해야 하는 경우도 있다 (솔라리스의 경우, eject를 사용한다).

일부 시스템은 드라이브에 플로피 디스크를 넣으면 자동으로 마운트를 시도한다. 웍스테이션에서 플로피 이미지를 써넣기 전에 이 기능을 미리 꺼놓아야 한다. 하지만, 그 방법은 시스템에 따라 다르다. 솔라리스의 경우 vold의 실행을 중단시키면 된다. 그 이외의 시스템에 대해서는 시스템 관리자에게 문의하기 바란다.

9.2 도스, 윈도즈, OS/2 시스템에서 디스크 이미지 써넣기

플로피 디스크 이미지들이 있는 디렉토리에 rawrite2.exe 프로그램이 있다. rawrite2의 자세한 사용법은 같은 디렉터리 내의 rawrite2.txt를 참고하기 바란다.

플로피 디스크에 이미지를 써넣으려면 도스로 부팅해야 한다. 윈도즈로 부팅한 후에 도스 창을 열어서 rawrite2를 사용할 경우 많은 문제가 발생한다고 알려져 있다. 윈도즈 탐색기에서 rawrite2를 더블클릭하더라도 작동하지 않는다고 알려져 있다. 도스로 부팅하는 방법을 모르겠다면, 부팅 도중에 *F8*을 눌러보기 바란다.

도스로 부팅한 후에 다음 명령을 사용한다.

```
rawrite2 -f file -d drive
```

*file*은 플로피 디스크 이미지 파일들의 이름 중 하나이고, *drive*는 'a:' 나 'b:' 이다.

9.3 플로피 디스크의 신뢰성

데비안을 처음 설치하는 사람들이 가장 많이 부딪히는 문제는 플로피 디스크의 신뢰성에 관한 것이다.

Rescue Floppy는 사람들을 가장 괴롭히는 문제들을 가진 것인데, 이는 리눅스가 부트되기 전에 바이오스가 직접 플로피를 읽기 때문이다. 바이오스는 리눅스 플로피 디스크 드라이버 만큼 신뢰성 있게 디스크를 읽지 못하는 것 같다. 그리고 잘못된 데이터를 읽어도 에러 메시지조차 출력하지 않고 멈춰버리기도 한다. 이뿐 아니라, Drivers Floppy와 베이스 시스템 파일 플로피에서도 문제가 일어나기도 한다. 보통 이 때의 에러 메시지는 'disk I/O errors'를 잔뜩 보여준다.

만약 어떤 특정한 플로피에서 인스톨이 멈췄다면, 가장 먼저 할 일은 플로피 디스크 이미지를 다시 다운로드 받고, *다*른 플로피에 써넣는 것이다. 그 디스크를 다시 포맷하여 이미지 파일을 써넣을 때 에러가 발생하지 않았다고 하더라도 문제가 해결될 것이라고 보기는 어렵다. 때로는 다른 시스템에서 디스크 이미지를 써넣는 것이 나을 때도 있다.

어떤 사용자는 제대로 작동하는 플로피를 얻기 위해, 이미지를 플로피에 써넣는 작업을 세 번이나 반복하였으며, 그제서야 문제를 해결할 수 있었다고 한다.

또 다른 사용자는 부팅에 문제가 있었지만 플로피를 새로 만드는 대신에 같은 플로피를 이용하여 몇 번 재부팅을 시도한 결과, 성공적으로 부팅하였다고 한다. 이는 하드웨어 또는 플로피 드라이버의 펌웨어에 문제가 있는 경우이다.

제 6 장

설치 시스템으로 부팅하기

이미 여러분은 어떤 방법으로 부팅할 것인지 결정하였을 것이다. Rescue Floppy를 이용하거나, 시디롬으로 부팅할 수도 있고, 또는 기존에 설치되어 있던 운영체제로 부팅할 수도 있다. 부팅을 조절하는 몇 가지 방법과 부팅 동안에 발생하는 일반적인 문제 그리고 해결 방안 또는 최소한 문제 해결에 도움이 되는 방법에 대하여 여기서 설명할 것이다.

데비안을 설치하기 위하여 응급 복구 디스크, 또는 시디롬을 이용하여 부팅한다면 `Control-Alt-Delete`을 눌러서 컴퓨터를 재부팅 시키는 경우가 있을 것이다. 그러나 `Control-Alt-Delete`에 의해 적절한 리셋이 이루어지지 않는 컴퓨터도 있으므로 ”hard”한 방법으로 컴퓨터를 재부팅 시킬 것을 권한다. 기존에 설치된 운영체제로 (예, 도스) 부팅하여 설치를 시작한다면 물론 그럴 필요가 없다. 그러나 플로피 또는 시디롬 등으로 부팅하여 설치하는 경우라면 반드시 ”hard”한 방법으로 컴퓨터를 재부팅 시키기 바란다.

제 1 절 부트 파라메타

부트 파라메타란 리눅스 커널에 전달되는 파라메타이다. 어떤 주변 장치가 확실하게 작동하도록 하기 위하여 그 장치에 대한 정확한 정보를 커널에 전달해 줄 필요가 있을 때 부트 파라메타를 이용한다. 대부분의 경우, 주변장치에 대한 정보는 커널이 스스로 알아낸다. 그러나 일부의 경우, 커널에 약간의 도움을 주어야 할 것이다.

Rescue Floppy or from CD-ROM 를 이용하여 부팅하면 `boot:`라는 프롬프트가 화면에 나타날 것이다. Rescue Floppy로 부팅한 후 부트 파라메타를 이용하는 자세한 방법은 ‘Rescue Floppy로 부팅’ on this page를 참고하기 바란다. 이미 설치되어 있는 운영체로 부팅한다면 다른 방법을 이용하여 부트 파라메타를 지정해 주어야 할 것이다. 예를 들어, `install.bat` 파일을 편집할 수 있다. 부트 파라메타에 대한 모든 것을 알고자 한다면 Linux BootPrompt HOWTO (`http://www.linuxdoc.org/HOWTO/BootPrompt-HOWTO.html`)를 참고하기 바란다. 여기서는 가장 많이 사용되는 파라메타에 대해 간략히 다룰 것이다.

처음으로 부팅하는 경우에는 디폴트 부트 파라메타를 그대로 적용하는 것이 좋다 (즉, 부트 파라메타를 아무것도 지정하지 말라는 뜻이다). 그리고서 정상적으로 작동하는지 살펴보자. 아마도 문제없이 작동할 것이다. 어떤 하드웨어에 문제가 있다면, 그 장치에 대한 올바른 정보를 커널에 전달해 줄 수 있는 파라메타를 찾아서 다시 부팅할 때 그 파라메타를 이용하도록 한다.

부팅 과정에 `Memory: availk/totalk available` 이라는 메세지가 나타난다. *total*에 해당하는 값과 실제 사용 가능한 램의 총량이 킬로바이트 단위로 일치해야 한다. 일치하지 않는다면 `mem=ram` 파라메타를 사용할 필요가 있다. 여기서 *ram*에는 메모리 양을 킬로바이트 또는 메가바이트 단위로 넣어준다. 그 단위가 킬로바이트이면 숫자 뒤에 "k"를 붙이고, 메가바이트이면 "m"을 붙인다. `mem=8192k` 와 `mem=8m`는 모두 램의 양이 8MB라는 것을 의미한다.

2.0 대의 리눅스 커널에서는 메모리가 제한이 960 MB 까지이다. 이 이상의 메모리를 갖고 있다면 `mem=960m`라고 부트 파라메타를 더해 주어야 할 것이다.

일부 시스템에는 "inverted DCLs"를 가진 플로피가 장착되어 있다. 플로피 디스크는 정상인데 일을 때 에러가 발생한다면 `floppy=thinkpad` 파라메타를 사용해 보기 바란다.

IBM PS/1 또는 (ST-506 디스크 드라이가 장착된) ValuePoint 같은 시스템에서는 IDE 드라이브가 제대로 인식되지 않을 수도 있다. 일단 파라메타를 주지 않은 상태에서 IDE 드라이브가 정확하게 인식되는지 살펴보도록 한다. 제대로 인식되지 않는다면 드라이브 지오메트리를 (실린더, 헤드, 섹터 수) 확인한 후, `hd=cylinders,heads,sectors.` 파라메타를 사용해 보기 바란다.

다시 한번 말하지만, 부트 파라메타에 대한 자세한 것은 Linux BootPrompt HOWTO (`http://www.linuxdoc.org/HOWTO/BootPrompt-HOWTO.html`)를 참고하기 바란다. 잘 알려지지 않은 하드웨어에 대한 팁도 들어 있다.

제 2 절 Rescue Floppy로 부팅

Rescue Floppy로 부팅하는 것은 쉽다. Rescue Floppy를 첫번째 플로피 드라이브에 넣고 *reset*을 누르거나 시스템을 껐다 키면 된다.

즉, 저 위에서 이야기 한 바와 같이 "hard reboot" 하는 것이 좋다.

플로피가 읽혀져야 하고, 그리고 나서 화면에 Rescue Floppy에 대한 소개가 나온 후, "boot:" 프롬프트가 뜨는 것을 볼 수 있을 것이다.

다른 방법을 이용하여 부팅한다면 `boot:` 프롬프트가 나타날 때까지 지시대로 따라하면 된다. 메모리가 5MB 보다 적다면 Low-Memory Boot Disk를 ('시스템에 메모리가 적은 경우의 설치 방법' on page 21를 참고하기 바란다) 이용하여 부팅해야 한다. 1.4MB 플로피 드라이브 보다 적은 용량의 플로피를 이용하여 부팅한다면,

ram-disk boot 방법을 사용해야 하며 루트 디스크가 필요할 것이다.

"boot:" 프롬프트가 떴을 때, 두 가지 일을 할 수 있다. *F1*에서 *F10* 사이의 키를 누르면 유용한 정보를 보거나, 시스템을 부팅시킬 수 있다.

*F4*와 *F5* 키를 누르면 부트 파라메타에 대한 정보를 볼 수 있다. 부트 명령행에 파라미터를 덧붙일 때에는, 부트 방법 (디폴트는 `linux`)을 입력하고, 한칸 띄운 뒤 파라미터를 입력해야 한다 (예, `linux floppy=thinkpad`). 그냥 *Enter*를 누르는 것과 특별한 파라미터 없이 `linux`라고 입력하는 것은 동일하다.

하드디스크로 부팅할 수 없게 되는 경우가 있다면, 이 디스크를 이용하여 시스템을 부팅시켜 문제를 해결할 수 있기 때문에 Rescue Floppy라고 부른다. 그러므로, 데비안 인스톨 후에 이 디스크를 잘 보관하여야 한다. *F3*를 누르면 Rescue Floppy 이용 방법에 대한 더 많은 정보를 얻을 수 있다.

엔터키를 누르면, `Loading...`이라는 메시지가 나타나고 잠시 후 `Uncompressing Linux...`라는 메시지가 뜬 후, 시스템에 대한 정보가 화면을 가득 메울 것이다.

디폴트 부팅 방법을 선택하지 않고 "ramdisk" 또는 "floppy" 와 같은 부팅 방법을 선택하였다면 루트 디스크가 있어야 한다. 루트 디스크를 첫번째 플로피 드라이버에 넣고 *Enter*를 누르도록 한다 (만일, floppy1을 선택하였다면 루트 디스크를 두번째 디스크 드라이브에 넣어야 한다).

제 3 절 메모리가 적은 시스템에서 부팅

만약 램이 5MB 이하인 시스템 이라면, 로우 메모리 부트 플로피를 이용하여 부팅해야 할 것이다. 설치 시스템은 메모리가 충분하지 않으면 특별한 로우 메모리 설치 과정을 따를 것인지 물어 볼 것이다.

다음의 순서대로 메뉴를 선택해 나가면 된다:

- `fdisk`를 이용하여 리눅스 스왑 파티션을 (type 82) 만든다. 설치 작 업은 시스템에 있는 메모리보다 더 많은 메모리를 사용하기 때문에 인스톨 작업 동안 가상 메모리(virtual memory)를 제공할 스왑 파티션이 필요하다. 설치 작업을 할 때 사용할 가상 메모리의 양을 선택한다. 아마도 16메가는 가능한 최저치이고, 하드 디스크에 약간 여유가 있다면 32메가를, 하드 디스크에 여유 공간이 충분하다면 64메가를 선택한다.

  추가로 Minix 파티션 (type 81)을 만든다. 초기 설치 작업 동안 루트 파일 시스템을 여기에 둘 것이다. 그 크기는 적어도 2메가 바이트는 되어야 한다. 설치가 끝나면 이 파티션을 삭제할 수 있다.
- 스왑 파티션을 활성화 시킨다.
- 루트 파일 시스템을 디스크로 카피한다. `root.bin` 파일이 들어있는 플로피 디스크가 필요할 것이다.
- 빠져 나오면, 정상적인 설치 시스템이 시작될 것이다.

제 4 절 시디롬으로 부팅

시디롬 드라이브에 시디롬을 넣고 부팅하면 `boot:` 프롬프트가 나타날 것이다. 이 상태에서 부트 파라메타를 입력할 수 있고, 커널 이미지를 선택할 수 있다.

제 5 절 커널 스타트업 메세지 해석

부팅 중에 `can't find something`, `something not present`, `can't initialize something` 또는 `this driver release depends on something`과 같은 메시지를 많이 볼 수도 있다. 이들 메시지 대부분은 별 의미가 없는 것이다. 설치 시스템에 사용되는 커널은 주변장치가 제각각인 가지각색의 컴퓨터에서도 작동해야 하기 때문에 다양한 주변장치에 대한 지원을 포함하고 있으며 따라서 위와 같은 메세지들이 나타나는 것이다. 주변장치란 주변장치를 모두 달고 있는

컴퓨터란 있을리가 없으며, 따라서 시스템에 없는 장치를 찾다가 그런 장치를 발견할 수 없다고 메시지를 출력하는 것은 당연한 것이다. 잠시동안 시스템이 멈춰 있는 경우도 볼 수 있을 것이다. 이것은 어떤 장치의 응답을 기다리는 경우이다. 그런데 실제로는 그 장치가 없기 때문에 시스템이 잠시 멈추는 것이다. 이와 같은 문제로 시스템 부팅에 걸리는 시간이 너무 길다면, 나중에 시스템에 맞도록 커널을 다시 만들 수 있다 ('새로운 커널 컴파일하기' on page 46 참고).

제 6 절 부팅 과정의 문제 해결

부팅 동안에 멈추거나, 실제로 있는 주변장치를 아예 인식하지 못하거나 또는 엉터리로 인식하는 등의 문제가 있다면 '부트 파라메타' on page 30에서 이야기 하였듯이 부트 파라메타를 먼저 점검해야 한다.

주변 장치를 제거하므로서 문제가 해결되는 경우가 종종 있다. 내장 모뎀, 사운드 카드, 플러그앤플레이 장치는 특히 문제가 될 소지가 많다.

현재의 부트 커널은 Adaptec AHA-2940과 같은 AIC7xxx-based SCSI 카드의 사용에 문제가 있다고 알려져 있다. 일부는 `aic7xxx=no_probe`라고 부트 파라메타를 지정해 주어서 문제가 해결되었다고 한다. 이 문제가 곧 수정 될 것으로 기대한다. 이 문제 해결을 돕고자 한다면 `<debian-testing@lists.debian.org>`로 메일을 주기 바란다.

Tecras를 포함한 일부 노트북의 경우 A20 게이트를 열 때 캐쉬를 비우지 못하는 문제가 있다. 이 문제는 bzImage 커널을 사용할 경우 발생하며 zImage 커널을 사용할 때는 발생하지 않는다. 이 문제가 존재한다면 `A20 gating failed`라는 메세지를 볼 수 있을 것이다. 이러한 경우에는 'tecra' 부트 이미지를 사용해야 할 것이다. '설치 시스템을 구성하는 파일들에 대한 설명' on page 21를 살펴보기 바란다.

루트 파일시스템 (램디스크 이미지)을 메모리에 로딩할 때 `out of memory`와 같은 에러가 발생한다면 로우 메모리 부트 플로피를 이용하여 부팅해야 할 것이다. 로우 메모리 시스템 부팅 과정을 시작하면 '메모리가 적은 시스템에서 부팅' on page 30에 설명된 단계를 따라야 한다.

매우 오래된 머쉰이라면 `Checking 'hlt' instruction...`라는 메시지를 보여주고 멈춰버리는 경우가 있다. 이 때는 부팅 인자로서 `no-hlt`를 입력해 주도록 한다.

여전히 문제가 있다면 버그 리포트를 보내주기 바란다. `<submit@bugs.debian.org>`로 메일을 보내면 된다. 메일의 첫째 줄에는 아래와 같은 내용이 반드시 포함되어야 한다.

```
Package: boot-floppies
Version: version
```

*version*은 여러분이 사용한 boot-floppies의 버전으로 대치되어야 할 것이다. *version*을 모른다면 다운로드 받은 날짜와 어느 배포본에서 받았는지를 (예를 들어, "stable", "frozen") 명시해 주어도 된다.

또한 버그 레포트에는 아래와 같은 정보도 포함되어야 한다.

```
architecture:  i386
model:         your general hardware vendor and model
memory:        amount of RAM
```

```
scsi:          SCSI host adapter, if any
cd-rom:        CD-ROM model and interface type, i.e., ATAPI
network card:  network interface card, if any
pcmcia:        details of any PCMCIA devices
```

버그의 종류에 따라, IDE 디스크에 설치했는지 SCSI 디스크에 설치했는지, 디스크 용량, 비디오 카드 모델에 대한 정보도 알려주면 문제 해결에 도움이 될 것이다.

버그 레포트에는 무엇이 문제인지 설명하고, 커널이 멈추는 경우에는 마지막에 나타나는 커널 메세지를 포함하도록 한다. 그리고 시스템이 문제의 상태로 될 때까지 어떤 단계를 거쳤는지 설명해야 한다.

제 7 장

dbootstrap을 이용한 시스템 초기화 설정

제 1 절 dbootstrap 소개

dbootstrap은 설치 시스템으로 부팅한 후 실행되는 프로그램의 이름이다. 이를 이용하여 시스템 초기화 설정을 하고 "베이스 시스템"을 설치한다.

dbootstrap의 주 임무는 시스템의 핵심 요소를 설정하는 것이다. IP 주소, 호스트 이름 그리고 네트워크 셋업에 필요한 사항 등이 여기에 포함된다. 커널과 연결될 "커널 모듈"을 설정하는 것도 여기에 포함된다. 모듈에 포함되는 것으로는 저장 장치 드라이버, 네트워크 드라이버, 특별한 언어 지원, 그외 주변 장치에 대한 지원 등이 있다.

여러분의 시스템이 제 기능을 발휘하도록 하기 위해 또는 다음 설치 단계로 넘어가기 위해서는 이러한 기본적인 것들을 먼저 설정해 주어야 한다.

dbootstrap는 (그래픽을 표시할 수 없는 시스템을 위한) 간단한 문자 기반의 프로그램이다. 매우 사용하기 쉽다. 길을 따라 걷듯이 설치 과정의 각 단계를 안내할 것이다. 어떤 단계에서 실수가 있었다면 뒤로 돌아가서 그 단계를 다시 밟을 수도 있다.

화살표 키와 *Enter* 그리고 *Tab* 키를 이용하면 dbootstrap 안에서 움직일 수 있다.

유닉스나 리눅스를 어느 정도 사용할 줄 아는 사람이라면, *Left Alt-F2*를 눌러 두 번째 *가상* 콘솔을 열 수 있다. 스페이스 바의 왼편에 있는 *Alt* 키와 *F2*키를 동시에 누른다. Bourne shell과 유사한 ash가 실행되는 또 하나의 분리된 윈도우이다. 이제 RAM 디스크로 부팅된 상태이고, 사용할 수 있는 유닉스 유틸리티에는 제한이 있다. `ls /bin /sbin /usr/bin /user/sbin`을 하면 현재 사용할 수 있는 유닉스 유틸리티를 볼 수 있다. 각 프로그램들이 할 수 있는 어떠한 작업을 수행할 때에는 메뉴를 사용하도록 한다. 셸과 명령어(command)들은 무언가 잘못 되었을 때 사용하기 바란다. 특히, 스왑 파티션을 활성화시키기 위해서는 항상 메뉴를 사용하고 셸을 사용하지 않기 바란다. 왜냐하면, 여러분이 셸에서 무슨 일을 했는지 메뉴 소프트웨어는 모르기 때문이다. *Left Alt-F1*를 누르면 메뉴로 돌아갈 것이다. 리눅스는 가상 콘솔을 64개 까지 제공하지만 Rescue Floppy에서는 단지 몇 개 만을 지원한다.

에러 메세지는 대개 (tty3로 알려진) 세번째 가상 터미날로 출력된다. *Left Alt-F3*를 눌러서 이 터미날을 열 수 있다. dbootstrap으로 돌아가려면 *Left Alt-F1*을 누른다.

제 2 절 "Select Color or Monochrome display"

시스템의 부팅이 끝나면, "Select Color or Monochrome display" 대화 상자가 나타난다.

모니터가 칼라이면, 그냥 *Enter*를 누른다. 이때 화면이 흑백에서 칼라로 바뀌어야 한다. "Next"에서 다시 *Enter*를 눌러 설치를 계속한다.

모니터가 흑백이라면 화살표 키를 이용하여 커서를 "Next" 메뉴 항목에 갖다 놓고 *Enter*를 눌러서 설치를 계속한다.

제 3 절 "Debian GNU/Linux Installation Main Menu"

"The installation program is determining the current state of your system and the next installation step that should be performed."라는 메세지를 볼 수 있다. 어떤 시스템에서는 읽을 수 없을 정도로 빨리 지나갈 수 있다. 주 메뉴의 각 단계 사이에서 이러한 대화 상자를 볼 수 있을 것이다. 설치 프로그램은 (dbootstrap) 각각의 단계 사이에서 시스템의 상태를 체크할 것이다. 이것은 만약 설치 도중 시스템이 중단되는 상황이 일어나도 이미 해놓은 작업에서부터 설치를 다시 할 수 있도록 해준다. 설치를 다시 시작해야만 한다면, 컬러 혹은 모노크롬 여부에 대한 것, 키보드, 스왑 파티션 재활성화를 설정하고, 초기화된 디스크를 다시 마운트(re-mount)해야 할 것이다. 시스템에서 했던 작업은 어떤 것이라도 저장되어 만약의 사태에 대비하는 기능을 데비안은 제공한다.

설치 전 과정에 걸쳐, "Debian GNU/Linux Installation Main Menu"라는 타이틀을 가진 주 메뉴가 나타날 것이다. 메뉴의 제일 위에는 설치가 진행됨에 따라 각 과정에 해당하는 메뉴가 나온다. 리눅스 저널 (http://www.linuxjournal.com/)에서, Phil Hugh는 데비안의 설치 메뉴 방식에 대해 " 닭에게 데비안을 설치하는 법을 가르쳐 줄 수 있을 정도이다!" 라고 이야기 했다. 그는 설치 과정이 *Enter* 키를 *쪼아대*는 것이라고 이야기 한다. 설치 메뉴에서 제일 먼저 나오는 선택 사항은 앞서 이루어진 과정이 무엇인지에 따라 다음에 해야 할 것들이 표시된다. "Next" 라는 항목을 선택하면 시스템 설치의 다음 단계로 넘어가게 된다.

제 4 절 "Configure the Keyboard"

"Next" 항목에 반전 막대가 오도록 한다. 그리고 엔터키를 눌러서 키보드 설정 메뉴로 간다. 여러분의 언어에 맞는 키보드 레이아웃을 선택한다. 해당되는 키보드 레이아웃이 없다면 가장 가까운 설정을 선택하도록 한다. 설치가 끝난 후에도 키보드 레이아웃을 변경할 수 있으며, 이때는 제공되는 선택의 폭이 더 넓다 (설치를 완료한 후에 root의 권한으로 kdbconfig을 실행시킨다).

설치할 키보드에 반전막대를 오게 하고 *Enter*를 누른다. 반전 막대를 움직일 때에는 화살표키를 사용한다. 어떤 언어를 지원하는 키보드 레이아웃이든지 간에 화살표키는 똑같은 위치에 있는, 키보드 설정과는 무관한 사항이다.

제 5 절 마지막 기회!

앞에서 하드디스크를 백업하라는 이야기를 했을 것이다. 지금 이 순간은 하드디스크의 모든 데이타를 지울 수 있는 최초의 기회인 동시에 기존의 시스템을 저장할 수 있는 마지막 기회이기도 하다. 당

신이 디스크를 모두 백업하지 않았다면, 드라이브에서 플로피를 모두 빼고, 시스템을 리셋한 후 백업 프로그램을 실행시키기 바란다.

제 6 절 "Partition a Hard Disk"

리눅스 네이티브와 리눅스 스왑 파일 시스템에 하드디스크 공간을 할당하기 위해 하드디스크 파티션 나누는 작업을 하지 않았다면 "Next" 메뉴 항목은 "Partition a Hard Disk"가 될 것이다. 만약 적어도 하나의 리눅스 네이티브와 리눅스 스왑 파티션을 이미 만들어 놓았다면, "Next" 메뉴는 "Initialize and Activate a Swap Partition"이 될 것이다. 시스템 메모리가 적어서 시스템을 시작할 때 스왑 파티션을 이미 활성화시켰다면 이 단계를 건너뛸 수 있다. "Next" 메뉴가 무엇이든간에, 아래 방향 화살표키를 사용하여 "Partition a Hard Disk"를 선택할 수 있다.

"Partition a Hard Disk" 항목은 분할할 수 있는 디스크 드라이브의 목록을 보여주고, 파티션에 사용되는 프로그램을 실행시킨다. '하드 디스크 파티션에 대하여' on page 15에서 설명한 바와 같이, 최소한 하나의 "리눅스 네이티브" (type 83) 파티션은 반드시 만들어야 하며 "리눅스 스왑" (type 82) 파티션도 최소한 하나 만드는 것이 좋다. 파티션을 어떻게 해야 좋을지 모르겠다면 '하드 디스크 파티션에 대하여' on page 15을 다시 읽어보기 바란다.

시스템의 아키텍쳐에 따라 사용되는 프로그램이 다르다. 아래는 해당 아키텍쳐에서 사용되는 프로그램이다.

fdisk 오리지널 리눅스용 디스크 파티션 프로그램, 전문가용; fdisk 매뉴얼 (`fdisk.txt`) 참고.

cfdisk 사용하기 간편한 풀스크린 디스크 파티션 프로그램; cfdisk 매뉴얼 (`cfdisk.txt`) 참고.

"Partition a Hard Disk"를 선택하면 이 중에서 하나가 디폴트로 실행된다. 디폴트로 실행되는 프로그램이 마음에 들지 않으면 쉘 (tty2)로 빠져 나가서 사용할 프로그램을 직접 실행시킬 수도 있다. 그 후에 "Partition a Hard Disk"를 건너뛰고 다음 단계를 수행하도록 한다.

스왑 파티션을 만들 것을 강력하게 권고하지만, 메모리가 4MB 이상이고 스왑 파티션이 필요없다는 판단이 서면 만들지 않아도 된다. 이때는 메뉴에서 "Do Without a Swap Partition"을 선택하도록 한다.

루트 파티션에 "Bootable" 표시를 해 주어야 한다.

제 7 절 "Initialize and Activate a Swap Partition"

디스크 파티션을 만들고 나면, "Next" 항목은 "Initialize and Activate a Swap Partition"가 된다. 스왑 파티션과 관련하여 세가지 선택권이 주어질 것이다. 즉, 새로운 스왑 파티션을 초기화하고 활성화하거나, 이미 초기화된 스왑 파티션이 있을 경우 그것을 활성화시키거나, 또는 스왑 파티션이 없는 상태로 계속 설치를 진행하는 것이다. 스왑 파티션은 언제든지 재초기화시킬 수 있으므로, 이 단계에서 무엇을 선택할 것인지 확실하게 알고 있는 것이 아니라면 "Initialize and Activate a Swap Partition"을 선택하도록 한다.

대화창이 나타나면서 “Please select the partition to activate as a swap device.”와 같은 메세지를 볼 수 있을 것이다. 그리고 그 밑에 표시되어 있는 것은 앞에서 스왑 파티션으로 선택한 장치의 이름이다. 맞다면 *Return*을 누르면 된다.

그 다음으로 하드디스크 표면의 손상으로 인한 읽을 수 없는 디스크 블럭이 있는지 검사하기 위하여 전체 파티션을 스캔할 것인지 물어볼 것이다. MFM, RLL, 혹은 오래된 SCSI 디스크를 사용한다면 이 옵션은 유용하며, (시간이 소모되기는 해도) 절대 해가 되지는 않는다. 요즘에 생산되는 (제대로 동작하는) 하드 디스크에는 필요없는 일이다. 왜냐하면 요즘의 하드디스크는 손상된 디스크 블럭을 자동으로 표시해 주는 내부 매커니즘이 있기 때문이다.

초기화 시키면 기존에 그 파티션에 있던 데이타가 모두 파괴될 것이라는 확인 메시지가 나타날 것이다. 아무런 문제가 없다면 “Yes”를 선택하도록 한다. 초기화 프로그램이 실행되는 화면으로 전환될 것이다.

제 8 절 “Initialize a Linux Partition”

이제 다음 메뉴 항목은 “Initialize a Linux Partition” 일 것이다. 그렇지 않다면, 디스크 파티션 나누는 과정을 완전히 끝내지 못했거나, 스왑 파티션에 관련된 메뉴 중에서 아무것도 시행하지 않았기 때문이다.

여러분은 리눅스 디스크 파티션을 초기화 시키거나, 혹은 이전에 초기화 시킨 것을 마운트할 수 있다. 기존에 설치되어 있던 데비안 시스템을 업그레이드 시키는 기능이 dbootstrap에는 없다는 것을 명심하기 바란다. 대체로 데비안은 자신을 업그레이드 할 수 있으며 dbootstrap을 사용할 필요는 없다. 데비안 2.1로 업그레드 하는 방법은 upgrade instructions (http://www.debian.org/releases/2.1/i386/release-notes/)을 참고하기 바란다.

따라서 비어있지 않은 기존의 디스크 파티션을 사용하려면 (즉, 기존에 있던 것을 날려버린다면), 기존의 디스크 파티션을 초기화하여야 한다 (모든 파일이 지워진다). 또한 디스크 파티션 나누는 단계에서 만들어진 파티션도 모두 초기화 되어야 한다. 만약, 현재의 시스템에 설치를 하다가 이 단계에서 중단한 적이 있고 그 때 사용하던 설치 플로피 디스크를 이용하여 다시 설치를 한다면 디스크 파티션을 초기화 하지 않고 파티션을 마운트하게 될 것이다.

“Next” 메뉴 항목을 선택하면 / 디스크 파티션을 초기화하고 마운트하게 된다. 여러분이 마운트 또는 초기화 시킨 첫번째 파티션은 / (루트라고 읽는다)로서 마운트 된다. 스왑 파티션을 초기화 할 때와 같이 불량 블럭 존재 여부의 검사를 위해서 디스크를 스캔할 기회를 제공한다. 불량 블럭을 찾기 위한 디스크 스캔 작업은 결코 해로운 것은 아니며, 다만 대용량 디스크의 경우 작업을 마치는 데 10분 이상이 걸릴 수 있다.

/ 파티션을 마운트한 후, 설치 단계를 더 수행하지 않았다면 “Next” 항목은 “Install Operating System Kernel and Modules” 일 것이다. 셋업할 파티션이 더 있다면 화살표키를 사용하여 디스크 파티션의 초기화 또는 마운트 메뉴 항목을 선택할 수 있다. /var, /usr 혹은 다른 파일 시스템에 분리된 파티션을 할당하려 한다면, 지금 초기화하고 마운트해야 한다.

제 9 절 “Mount a Previously-Initialized Partition”

“‘Initialize a Linux Partition’” on page 35 이외에 “Mount a Previously-Initialized Partition”가 있다. 설치를 다시 시작하거나 이미 초기화된 파티션을 마운트하고자 할 때 선택하도록 한다.

제 10 절 "Install Operating System Kernel and Modules"

루트 파티션을 마운트한 다음 단계는 "Install Operating System Kernel and Modules"이다. 먼저 루트 파티션으로서 마운트된 장치가 올바른 것인지 물어볼 것이다. 그 다음, 루트 파티션에 설치할 커널을 어느 장치로부터 읽어들일 지를 선택할 수 있는 메뉴가 제공된다. 커널과 모듈을 어느 장치로부터 읽어서 설치할 것인지 선택해야 한다 ('초기 부팅 매체 선택하기' on page 21 참고).

로컬 파일시스템으로부터 읽어서 설치할 경우, 그 장치가 마운트 되어 있지 않다면 "harddisk"를 선택하고, 마운트 되어 있다면 "mounted"를 선택한다. 그 다음, 데비안 설치 파일들이 저장되어 있는 파티션을 선택한다. 그러면 그 파티션의 어디에 파일을 두었는지 그 위치를 물어볼 것이다. 그 위치는 "/"로 시작한다. 그 후, dbootstrap으로 하여금 실제 파일을 ア [3] 도록 하거나, 필요하다면 직접 입력할 수도 있다.

플로피 디스크를 이용하여 설치를 한다면 Rescue Floppy를 플로피 드라이브에 넣고 (아마도 드라이브에 이미 넣어져 있을 것이다), 뒤이어서 Drivers Floppy를 넣는다.

네트웍을 경유하여 커널과 모듈을 설치하고자 한다면 "nfs" 옵션을 선택한다. 여러분의 네트웍 인터페이스는 표준 커널이 지원하는 것이어야 한다 ('주변 장치와 여타 하드웨어' on page 4 참고). 네트웍 설정을 하지 않아서 "nfs" 옵션이 보이지 않을 수도 있다. 이 때는 "Cancel"을 선택하여 빠져나온 후 "Configure the Network"를 선택하여 네트웍 설정을 하도록 한다 ("'Configure the Network"' on page 35 참고). 그 후, 다시 이 단계를 재실행하면 "nfs" 옵션이 보일 것이다. "nfs" 옵션을 선택한 후, NFS 서버 이름과 경로를 알려주면 된다. Rescue Floppy 와 Drivers Floppy 이미지를 NFS 서버의 적당한 곳에 두었다면, 그 파일에 접근할 수 있어야 커널과 모듈을 설치할 수 있을 것이다.

설치 매체에 따라서 다른 단계를 실행할 필요가 있을 수 있다.

제 11 절 "Configure PCMCIA Support"

"Configure Device Driver Modules" 단계에 앞서서 "Configure PCMCIA Support"라는 단계가 있다. 이 메뉴는 PCMCIA 지원을 활성화하기 위해 사용된다.

PCMCIA를 갖고 있으나 데비안 시스템 설치과정에서 이를 이용하는 것이 아니라면 (즉, PCMCIA 에더넷 카드의 이용), 이 단계를 수행할 필요는 없다. 설치가 끝난 후에 PCMCIA를 설정할 수 있다. 그러나 PCMCIA 네트웍 장치를 이용하여 설치를 하고자 한다면, 반드시 이 단계를 선택하여야 한다. 그리고 네트웍 설정에 앞서서 PCMCIA를 지원하도록 설정하여야 한다.

PCMCIA를 설치할 필요가 있다면 "Configure Device Driver Modules" 아래의 "Configure PCMCIA Support"를 선택한다. PCMCIA 콘트롤러가 무엇인지 물어 볼 것이다. 대부분의 경우 i82365이며 일부는 tcic 이다. 잘 모르겠다면 랩탑을 구입할 때 제공되는 설명서를 보고 확인하기 바란다. 그 다음의 몇 가지 옵션은 대개 빈 칸으로 남겨두어도 된다. 어떤 하드웨어는 특별한 조건을 필요로 한다. 문제가 있다면 Linux PCMCIA HOWTO (http://www.linuxdoc.org/HOWTO/PCMCIA-HOWTO.html)를 참고하기 바란다.

일부 비정상적인 경우에 /etc/pcmcia/config.opts의 수정을 필요로 할 수 있다. 이 때는 두번째 가상 터미날을 열어서 (*Left Alt-F2*) 파일을 수정한 후 PCMCIA를 다시 설정하거나 또는 insmod와 rmmod를 이용하여 강제로 모듈을 올릴 수 있다.

PCMCIA 설정을 제대로 한 후에는 장치 드라이버를 설정해야 한다.

제 12 절 "Configure Device Driver Modules"

"Configure Device Driver Modules"를 선택하여 시스템에 설치되어 있는 장치를 찾아서 해당 드라이버를 설정하면, 시스템이 부팅될 때 메모리에 적재될 것이다.

모든 장치에 대한 설정을 이 단계에서 할 필요는 없으며, 베이스 시스템 설치에 필요한 장치만 설정해 주면 된다 ('베이스 시스템 설치에 이용할 매체 선택' on page 21 참고). 에더넷 카드 드라이버가 여기에 포함된다.

설치가 끝난 후 언제든지 modconf 프로그램을 실행시켜 모듈을 다시 설정할 수 있다.

제 13 절 "Configure the Network"

시스템이 네트웍에 연결되어 있지 않더라도 네트웍 설정은 해야하지만, 이때에는 처음 두 질문에만 답하면 된다 – "Choose the Host name" 와 "Is your system connected to a network? (여러분의 시스템은 네트웍에 연결되어 있는가)".

네트웍에 연결되어 있다면, '미리 알아두어야 할 정보' on page 10에서 알아두라고 했던 정보가 필요할 것이다. 그러나 그 네트웍 연결이 PPP 라면 네트웍 설정을 선택해서는 *안된다*.

여러분이 속해 있는 네트웍에 대하여 몇가지 질문을 할 것이다. 그런 다음, 네트웍 정보를 요약하여 보여주면서 맞는지 재차 확인을 요구할 것이다. 그 다음에는 첫번째 네트웍 연결에 사용되는 네트웍 장치를 지정해 주어야 한다. 일반적으로 그것은 "eth0" (첫번째 에더넷 장치)가 될 것이다. 랩탑의 경우에는 "pcmcia"가 첫번째 네트웍 장치일 것이다.

몇몇 기술적인 사항들이 있다. 프로그램은 시스템의 IP 주소와 넷마스크 사이의 비트 연산(논리곱)을 통해 네트웍 IP 주소를 추측한다. 브로드캐스트 주소는 넷마스크의 1의 보수와 시스템 IP 어드레스간의 비트 연산(논리합)을 통해 추측한다. 그리고, 게이트웨이 시스템을 DNS 서버로 간주한다. 위의 사항들에 대해 아는 것이 없다면, 제시하는 것을 사용하도록 한다. 설치가 끝난 후에도 필요하다면 /etc/init.d/network 파일을 수정하여 이들 설정을 변경할 수 있다. (데비안 시스템에서는 /etc/init.d/에 있는 스크립트에 의해 데몬들이 실행된다.)

제 14 절 "Install the Base System"

설치할 베이스 시스템을 어느 장치로부터 읽어들일지 선택하는 메뉴가 제공되며 그 중의 하나를 선택한다 ('베이스 시스템 설치에 이용할 매체 선택' on page 21 참고).

하드디스크 또는 시디롬을 선택한다면 base2_1.tgz이 있는 경로를 물어볼 것이다. 공식 설치 매체를 이용한다면 디폴트로 주어지는 경로를 사용하면 되겠지만, 그렇지 않다면 정확한 경로를 입력해 주어야 한다. "Install Operating System Kernel and Modules"에서 처럼 dbootstrap으로 하여금 파일을 찾도록 하거나 직접 입력할 수 있다.

플로피 디스크를 이용하여 설치한다면, dbootstrap이 요청에 따라 베이스 플로피를 차례대로 넣는다. 베이스 플로피중 하나를 읽어들일 수 없으면 대치할 플로피를 만들어야 하며, 플로피를 처음부터 다시 순서대로 넣어야 한다. 모든 플로피를 읽은 후, 읽어들인 파일은 설치될 것이다. 느린 시스템에서는 이 과정이 10분 이상 걸릴 수 있다.

NFS를 이용하여 설치한다면, 서버 이름과 공유 디렉터리 그리고 base2_1.tgz이 있는 서브디렉터리를 물어볼 것이다. NFS를 마운트하는데 문제가 있다면 NFS 서버의 시스템 시간과 클라이언트 (현재 데비안을 설치하고 있는 시스템)의 시스템 시간이 일치하는지 확인하기 보도록 한다. 시간을 맞추려면 tty2에서 date 명령을 이용하도록 한다. 사용법은 date(1) 맨페이지를 참고하기 바란다.

제 15 절 "Configure the Base System"

이 단계에서는 최소한의 데비안 시스템 유지에 필요한 파일을 읽어들인다. 하지만, 시스템이 실행되기 전에 미리 설정해 둘 것이 있다.

여러분이 속해 있는 시간대 (time zone)를 물어볼 것이다. 시간대를 지정할 수 있는 많은 방법이 있지만 "Directories:" 메뉴로 가서 여러분의 나라 (또는 대륙)를 선택하도록 하자. 그러면 가능한 시간대로 바뀐다. 계속해서 "Timezones:" 메뉴에서 여러분의 지정학적 위치 (즉, 나라, 주 또는 도시)를 선택한다.

다음으로, 시스템 시간을 GMT로 할지 또는 현재 여러분의 지역 시간으로 설정할 것인지 물어볼 것이다. 시스템에서 리눅스나 유닉스만을 운영 체제로 사용할 것이면 GMT를 선택하고 ("Yes"), DOS 나 윈도즈 같은 다른 운영 체제도 같이 사용할 것이라면 지역 시간을 선택한다 ("No"). 유닉스나 리눅스는 시스템 시간을 GMT로 설정하고 이를 지역 시간대로 변환하여 보여준다. 이것은 일광 절약 시간(써머 타밍)과 윤년을 지원하며 다른 시간대에서 로그인한 사용자 개개인이 그들의 터미널에서 사용할 시간대를 설정할 수 있게끔 해준다.

제 16 절 "Make Linux Bootable Directly From Hard Disk"

하드디스크에서 직접 리눅스로 부팅하고자 한다면, 마스터 부트 레코드에 인스톨할 것인지 물어볼 것이다. 부트 매니저를 사용하고 있지 않다면(그리고, 부트 매니저가 무엇인지에 대해 모르는 경우라면) 그리고 같은 머쉰에 또다른 운영체제가 없다면, 이 질문에 "Yes"라고 답변한다.

예를 들어, 같은 머쉰에 도스가 있다고 하자. "Yes"라고 답변하면 도스로 부팅할 수 없을 것이다. 주의하기 바란다.

"Yes"라고 답변했다면, 시스템을 켤 때 하드 디스크로부터 자동으로 리눅스를 부팅시킬지 물어볼 것이다. 이것은 리눅스를 *bootable partition*으로 지정한다 – 하드디스크로부터 리눅스가 로드된다는 것을 의미한다.

하나의 머쉰에서 다중 운영체제를 각각 부팅시킨다는 것은 쉬운 일이 아니다. 여기서 여러가지 부트 매니저에 대해 설명하지는 않겠다. 자세한 것은 각 부트 매니저의 설명서를 참고하기 바란다. 부트 매니저를 이용한다면 매우 주의를 기울여야 한다는 것을 명심해야 할 것이다.

i386의 표준 부트 로더는 "LILO"이다. 이것은 도스, NT, OS/2 부트 관리를 포함한 많은 기능을 갖고 있는 프로그램이다. 특별히 필요한 것이 있다면 /usr/doc/lilo/ 디렉터리의 설명서를 읽어보기 바란다. 또한 http://www.linuxdoc.org/HOWTO/mini/LILO.html도 참고하기 바란다.

이 단계를 건너뛸 수도 있다. 나중에 리눅스 fdisk 명령어 또는 activate 프로그램을 이용하여 부팅 가능한 파티션을 지정할 수 있다. 문제가 생기거나 도스로 부팅할 수 없게 된다면 도스용 부트

디스크를 사용하여 도스로 부팅한 후, `fdisk /mbr` 명령으로 도스 부트 블럭을 재설치해야 할 것이다. 이것은 데비안으로 다시 부팅하려면 다른 방법을 이용해야 한다는 것을 의미한다.

제 17 절 "Make a Boot Floppy"

하드디스크로 시스템을 부팅시킨다해도 부트 플로피를 만들어야 한다. 그 이유는, 하드 디스크 부트스트랩은 잘못 설치될 가능성이 있지만 부트 플로피는 특별한 일이 없는 한 항상 작동하기 때문이다. "Make a Boot Floppy"를 선택하고, 지시하는 대로 빈 플로피 디스크를 넣어준다. 플로피에 쓰기 방지 탭이 붙어 있지는 않는지 확인하기 바란다. 플로피는 포맷된 후 쓰기를 시작할 것이다. 이 플로피에 '부트 디스크' (Custom Boot)라고 써붙이고, 쓰기 방지 탭을 붙이도록 한다.

제 18 절 진실의 순간

이제 시스템을 재부팅 시킬 차례다. 플로피 드라이브에서 플로피 디스크를 빼고, "Reboot the System" 메뉴를 선택한다. 리눅스 시스템이 시작되지 않으면, 설치 매체 (예를 들면, Rescue Floppy) 또는 전 단계에서 만든 부트 플로피 (Custom Boot floppy)를 넣고 시스템을 리셋시킨다. 부트 플로피로 부팅하는 것이 아니라면 부팅시 입력해 주어야 할 것이 있다. Rescue Floppy 또는 비슷한 방법으로 부팅한다면 `rescue root=`*`root`*라고 적어준다. 여기서 *root*는 "/dev/sda1" 과 같이 여러분이 선택한 루트 파티션을 말한다.

데비안으로 부팅되어야 한다. 설치 시스템으로 처음 부팅할 때 보았던 메세지를 볼 수 있을 것이고 뒤이어서 일부 새로운 메세지도 볼 수 있을 것이다.

제 19 절 루트 비밀번호 설정 (Set the Root Password)

root 계정을 *수퍼 유저 (super-user)*라고도 한다. 이 로그인은 시스템의 모든 보안 방지 장치를 통과할 수 있는 전지전능한 것이다. 루트 계정은 시스템 관리에만 이용되어야 하고, 가능한 짧은 시간 동안 사용하는 것이 좋다.

모든 비밀번호는 6-8자로 구성되어야 하고, 대소문자와 특수 문자도 섞어서 구성하는 것이 좋다. 루트의 강력한 권한을 고려하면, 루트의 비밀번호를 설정할 때는 특히 주의를 기울여야 할 것이다. 사전에 있는 단어 또는 전화번호와 같은 개인 정보를 비밀번호로 설정하면 누출될 가능성이 매우 높다.

루트 비밀번호를 알려달라고 한다면, 극히 조심해야 한다. 여러명이 시스템 관리를 하는 것이 아니라면, 절대 루트 비밀번호를 가르쳐 주어서는 안될 것이다.

제 20 절 일반 사용자 계정 만들기

일반 사용자 계정을 만들 것인지 물어볼 것이다. 이 계정은 개인적인 로그인을 위한 것이다. 일상적인 또는 개인적인 목적으로 루트 계정을 사용해서는 *안된다*.

왜 안되는가? 루트의 권한을 사용하지 말라는 한가지 이유는 치유 불가능한 손상을 입기 쉽다는 것이다. 또 다른 이유는 트로이 목마 프로그램의 트릭에 걸려들지 않기 위함이다. 트로이 목마 프로그램은 여러분이 모르는 사이에 시스템의 보안을 해제하고 수퍼 유저의 권한을 이용하려는 프로그램이다. 유닉스 시스템 관리에 대한 서적은 이에 대해서 더 자세히 다루고 있다. 잘 모르는 내용이라면 한 번 읽어보는 것도 좋을 것이다.

일반 사용자 계정에 붙일 이름은 어느 것이라도 상관없다. 예를 들어, 여러분의 이름이 Jone smith 이고, 자신의 이름을 이용하여 계정 이름을 짓는다면 ”smith”, ”john”, ”jsmith”, ”js” 등을 사용할 수 있을 것이다.

제 21 절 Shadow Password Support

쉐도우 패스워드 (shadow passwords)를 사용할 것인지 물어볼 것이다. 이를 이용하면 여러분의 리눅스 시스템 보안을 좀 더 강화할 수 있다. 쉐도우 패스워드를 사용하지 않는 시스템은 /etc/passwd 파일에 저장된 (암호화된) 비밀번호를 읽을 수 있다. 이 파일에는 또한 반드시 필요한 사용자 정보가 저장되어 있기 때문에 로그인 할 수 있는 사용자는 누구나 이 파일을 읽을 수 있도록 되어 있다. 따라서 누군가가 /etc/passwd을 빼내서 비밀번호를 알아내려고 할 수도 있다.

쉐도우 패스워드를 사용하다면, 비밀번호는 /etc/shadow 파일에 저장되며 루트만이 오로지 읽을 수 있다. 그래서 쉐도우 패스워드를 사용할 것을 권하는 것이다.

shadowconfig 프로그램을 이용하여 쉐도우 패스워드 시스템을 재설정할 수 있다. 더 자세한 것은 /usr/doc/passwd/README.debian.gz 파일을 참고하기 바란다.

제 22 절 PCMCIA 제거

PCMCIA를 사용하지 않는다면, 이제 그것을 제거할 수 있다. 그렇게 하면 좀 더 깨끗하게 시작할 수 있고, 커널을 교체하기도 더 쉽게 된다 (PCMCIA 드라이버의 버전과, 커널 모듈, 커널 사이에 많은 상호 연관이 있다).

제 23 절 프로파일의 선택과 설치

데비안에서 제공하는 pre-rolled 소프트웨어 설정을 사용할 것인지 물어볼 것이다. 설치하고자 하는 패키지를 하나씩 선택할 수도 있다. 이것이 dselect 프로그램의 목적이다. 그러나 데비안에서 제공하는 2250 여개의 패키지를 고려해 보면 만만한 작업이 아닐 것이다.

그래서 패키지를 하나씩 선택하는 대신에 *태스크* (tasks) 또는 *프로파일* (profiles)을 선택할 수 있도록 하고 있다. *태스크*는 ”Perl 프로그래밍”, ”HTML 문서 작성”, ”워드 프로세싱”과 같은 작업을 말한다. *프로파일*이란 여러분 컴퓨터의 용도를 말한다. 다시 말해서, ”네트웍 서버”로 사용될 것인지, 또는 ”개인용 웍스테이션”으로 사용될 것인지 등으로 분류하는 것이다. 여러분이 단지 하나의 프로파일만 선택할 수 있다.

투자할 시간이 넉넉하지 않다면 프로파일을 하나 선택하는 것이 좋다. 시간이 충분하다면 Custom 프로파일을 선택하여 태스크를 하나씩 선택하여 설치하는 것이 좋을 것이다. 시간이 남아돌고, 패키

지의 설치 여부를 신중하게 결정하고자 한다면 이 단계를 건너뛴 후, dselect를 이용하면 좋을 것이다.

이제 곧 dselect를 사용하게 될 것이다. 태스크나 프로파일을 선택했다면, dselect를 사용할 때 ”Select” 단계를 건너뛰어야 한다. 태스크 또는 프로파일을 선택하는 것이 ”Select” 단계에 해당하기 때문이다.

각 태스크의 크기는 그 태스크에 포함되어 있는 패키지를 모두 합한 것이다. 여러분이 두 가지 태스크를 선택하였고, 일부 패키지가 그 두가지 태스크에 모두 포함되어 있다면, 실제 필요한 디스크 용량은 두 태스크를 합한 크기보다 적을 것이다.

루트 계정과 일반 사용자 계정을 만들고 나면, 여러분은 dselect 프로그램 속으로 떨어질 것이다. dselect를 실행하기 전에 dselect Tutorial (http://kldp.org/Translations/html/Debian_Install-KLDP/dselect-beginner.html)를 읽어보는 것이 좋다. dselect를 이용하여 시스템에 설치할 *패키지*를 선택하게 된다. 데비안 패키지가 들어있는 시디롬 또는 하드디스크가 있거나 여러분의 시스템이 인터넷에 연결되어 있다면 이 프로그램을 곧바로 이용할 수 있다. 그렇지 않다면 현재 단계에서 dselect를 사용할 필요는 없다. 시간이 지나서 조건이 충족되면 다시 dselect를 실행 시킬 수 있다. dselect를 실행 시키려면 수퍼 유저 (root) 이어야 한다.

제 24 절 로그인 (Log In)

dselect를 종료하면 로그인 프롬프트가 나타날 것이다. 여러분이 만든 개인용 계정과 패스워드를 사용하여 로그인 하도록 한다. 이제 시스템을 사용할 준비가 다 된 것이다.

제 25 절 PPP 설정

NOTE: 시디롬으로 설치하거나 네트웍에 직접 연결된 경우에는 이 단계를 건너뛸 수 있다. 네트웍이 아직 설정되지 않았다면 설치 시스템이 이 정보를 물어볼 것이다.

베이스 시스템에는 ppp가 들어있다. 이 패키지를 이용하면 여러분이 가입해 있는 ISP에 PPP 연결을 할 수 있다. 기본적인 PPP 설정 방법을 아래에 설명한다. 부트 디스크에 PPP 설정을 도와줄 pppconfig라는 프로그램이 있다. *그 프로그램이 다이얼업 연결의 이름을 물어볼 때, ”provider”라는 이름을 붙이도록 하자.*

pppconfig 프로그램을 이용하면 별 다른 어려움 없이 PPP 설정을 할 수 있을 것이라 기대하지만, 만일에 잘 되지 않는다면 아래에 설명할 방법을 참고하기 바란다.

PPP를 설정하려면, 리눅스에서 파일을 읽고 편집하는 기초를 갖추어야 한다. 파일을 읽으려면 more를 사용하고, .gz 확장자를 가진 압축 파일은 zmore를 이용하면 읽을 수 있다. 예를 들어, README.debian.gz을 읽으려면 zmore README.debian.gz 라고 입력하면 된다. 베이스 시스템은 두 가지 에디터를 제공한다. 사용하기는 간단하지만 기능은 별로 없는 ae와 elvis-tiny라는 기능이 제한적인 vi를 제공한다. nvi, less, emacs와 같은 모든 기능을 갖춘 에디터와 뷰어는 나중에 설치할 수 있다.

/etc/ppp/peers/provider을 에디터로 열어서 ”/dev/modem”을 “/dev/ttyS#”으로 바꾼다. 여기서 #는 시리얼 포트 번호를 의미한다. 리눅스에서 시리얼 포트는 0에서 시작한다. 즉 리눅스에서 첫번째 시리얼 포트는 (즉, COM1) /dev/ttyS0이다. 다음 할 일은 /etc/chatscripts/

provider 파일을 열어서 PPP 연결에 이용하는 전화번호, user-name, 비밀번호를 적어 넣는 것이다. 비밀번호 앞의 "\q"를 지워서는 안된다. 이것의 역할은 로그 파일에 패스워드가 보이지 않도록 하는 것이다.

대부분의 경우, PPP 연결에서 로그인 과정에 PAP 또는 CHAP이 이용된다. PAP 이냐, CHAP 이냐에 따라서 과정이 달라진다. /etc/chatscripts/provider 파일에서 ("ATDT"로 시작하는) 다이얼링 스트링 이하의 모든 주석 처리를 지우고, 위에서 설명한 대로 /etc/ppp/peers/provider 파일을 수정하고 user *name*을 추가한다. 여기서 *name*은 PPP 연결에 사용하는 user-name을 말한다. 이제 /etc/pap--secrets 또는 /etc/chap--secrets 파일을 열어서 비밀번호를 적어 넣는다.

/etc/resolv.conf 파일을 열어서 ISP의 네임 서버 (DNS) IP 주소를 추가해 주어야 할 것이다. /etc/resolv.conf 파일에 다음과 같은 형식으로 한 줄 추가해 주면 될 것이다: nameserver *xxx.xxx.xxx.xxx*. 여기서 *x*는 IP 주소를 구성하는 숫자를 의미한다.

ISP에서 사용하는 로그인 과정이 특별히 다르지 않은 이상, 더 이상 해야 할 일은 없다. 루트의 권한으로 pon을 실행시켜 PPP 연결을 할 수 있으며 plog 명령으로 그 과정을 지켜볼 수 있다. 연결을 끊으려면 루트의 권한으로 poff를 실행시킨다.

제 26 절 나머지 설치 과정

데비안 시스템의 나머지 부분 설치에 대한 정보는 dselect Tutorial (http://kldp.org/Translations/html/Debian_Install-KLDP/dselect-beginner.html)에 있다. 태스크와 프로파일을 이용한다면 dselect에서 "Select" 단계를 건너 뛰어야 한다는 것을 기억하기 바란다 ('프로파일의 선택과 설치' on page 35 참고).

제 8 장

앞으로 무엇을 할 것인가?

제 1 절 유닉스 초보자의 경우

유닉스를 처음 접한다면 아마도 책을 사서 공부를 해야만 할 것이다. Unix FAQ (ftp://rtfm.mit.edu/pub/usenet/news.answers/unix-faq/faq/)는 여러분을 도울 수 있는 책과 유즈넷 뉴스 그룹에 대한 많은 참고자료를 가지고 있다. User–Friendly Unix FAQ (http://www.camelcity.com/~noel/usenet/cuuf-FAQ.htm)를 볼 수도 있다.

리눅스는 유닉스의 일종이다. Linux Documentation Project (LDP) (http://www.linuxdoc.org/)는 리눅스와 관련된 많은 HOWTO 문서와 온라인 문서를 수집한다. 이 문서들의 대부분은 하드디스크에 설치할 수 있다. doc-linux-html 패키지 (HTML versions)를 설치하거나 doc-linux-text 패키지 (ASCII versions)를 설치하면, /usr/doc/HOWTO 디렉터리에서 문서를 찾아 볼 수 있다. LDP HOWTO의 국제화 버전도 데비안 패키지로 제공된다.

데비안에 대한 특별한 정보는 아래에서 찾을 수 있다.

제 2 절 데비안에 적응하기

데비안은 다른 배포본과는 조금 다르다. 다른 배포본의 리눅스에 익숙하다고 하더라도 여러분의 시스템을 양호하면서 깨끗한 상태로 유지하기 위해 데비안에 대해 알아야 할 몇가지가 있다. 여러분이 데비안에 적응할 수 있도록 도울 수 있는 자료를 여기에 포함시키고 있지만 데비안 사용법에 대한 설명서로서 계획된 것이 아니라 단지 매우 급한 사람들을 위해 시스템을 한 번 훑어보는 정도로 그칠 것이다.

파악해야 할 가장 중요한 개념은 데비안 패키징 시스템이다. 본질적으로 여러분의 시스템의 많은 부분을 패키징 시스템의 통제하에 고려한다. 다음과 같은 것들을 포함한다.

- /usr (excluding /usr/local은 제외함)
- /var (/var/local을 만들면 안전할 것이다)
- /bin

- /sbin
- /lib

예를 들면 /usr/bin/perl을 교체한다면, 작동할 것이다. 그러나 perl 패키지를 업그레이드한다면 여러분이 거기에 놓아둔 파일도 교체될 것이다. dselect에서 “hold”에 패키지를 놓음으로써 이 문제를 피할 수 있다.

제 3 절 Further Reading and Information

특별한 프로그램에 대해 정보가 필요하다면 우선 man 프로그램이나 info 프로그램으로 알아보아야 한다.

/usr/doc에도 많은 유용한 문서가 있다. 특별히 /usr/doc/HOWTO와 /usr/doc/FAQ에는 많은 흥미있는 정보를 포함하고 있다.

데비안 웹 사이트 (http://www.debian.org/)는 데비안에 대한 많은 양의 문서를 포함하고 있다. 특히 데비안 FAQ (http://www.debian.org/doc/FAQ/)와 데비안 메일링 이스트 모음 (http://www.debian.org/Lists-Archives/)을 보라. 데비안 공동체는 자급자족(self-supporting)이다. 하나 또는 그 이상의 데비안 메일링 리스트에 가입하려면 메일링 리스트 가입 (http://www.debian.org/MailingLists/subscribe)페이지를 참고하기 바란다.

제 4 절 새로운 커널 컴파일하기

왜 어떤 사람들은 커널을 새로이 컴파일하려 할까? 데비안에 들어있는 기본 커널은 대부분의 설정을 다루고 있기 때문에 커널을 새로이 컴파일 할 필요는 별로 없다. 그러나 다음과 같은 이유 때문에 새로운 커널을 컴파일하는 것은 유용하다.

- 일반적으로 사용하는(stock) 커널이 포함하고 있지 않은 APM나 SMP 같은 하드웨어나 옵션을 다루기 위해
- 쓸모없는 드라이버를 제거해서 커널을 최척화시켜 부팅 시간을 빠르게 하고 커널의 크기를 더 작게 만들기 위해서 (kernel memory cannot be swapped to disk)
- 기본 커널에서 지원하지 않는 커널 옵션을 사용하기 위해(네트웍 파이어월 같은 것)
- 개발 커널을 사용하기 위해
- 친구를 감동시키려고, 새로운 것을 시도하기 위해

커널 컴파일을 시도하는 것을 두려워하지 말 것. 그것은 재미있고 유익하다.

데비안의 방식으로 커널을 컴파일하려면 몇가지 패키지가 필요하다. 커널 패키지, 커널 소스-&커널 버전;(이 글을 쓸 당시 가장 최신 버전), fakeroot와 아마도 이미 설치되어 있을 몇가지 다른 패키지들(완전한 리스트는 /usr/doc/kernel--package/README.gz를 참고하기 바란

다). "데비안의 방식"으로 여러분의 커널을 컴파일할 필요는 없다는 것을 주의하라. 그러나 우리는 커널을 다루기 위해 패키징 시스템을 사용하는 것이 실제로 더 안전하고 더 쉽다는 것을 발견했다. 사실상 커널 소스를 커널 소스-&커널 버전; 대신 Linus로부터 바로 가져올 수 있지만 여전히 kernel-package의 컴파일 도구를 사용한다.

kernel-package를 사용하는 것에 대한 완전한 문서를 /usr/doc/kernel--package에서 찾을 수 있다. 여기에는 단지 간단한 설명서가 포함하고 있다.

앞으로 커널 소스가 /usr/local/src에 있고 커널 버전이 &커널 버전;이라고 가정하자. 루트로 /usr/local/src 아래에 디렉토리를 만들고 소유자를 일반 사용자 계정으로 바꾼다. 일반 사용자 계정으로 커널 소스를 풀고자 하는 디렉토리로 이동해서 (cd /usr/local/src) 커널 소스의 압축을 풀고 (tar xzf /usr/src/kernel-source-2.0.36.tar.gz) 그 디렉토리로 이동한다(cd kernel-source-2.0.36/). 이제 커널을 설정할 수 있다(X11을 설치하고 설정했다면 make xconfig를 이용하고 그렇지 않다면 make menuconfig를 이용하라). 온라인 도움말을 읽고 신중히 선택하라. 의심스러울 때는 불확실한 장치 드라이버(이더넷 카드, SCSI 컨트롤러 등의 주변기기를 관리하는 소프트웨어)도 포함시키는 것이 대체로 더 낫다. 특별한 하드웨어와 관련되지 않은 다른 옵션은 이해가 되지 않으면 기본값으로 놓아두어야 한다. "Loadable module support" (기본으로 선택되지 않는다)에서 "Kernel daemon support (e.g. autoload of modules)"를 선택하는 것을 잊지 말 것. 그렇지 않으면 데비안 설치에서 문제가 생길 것이다. [1]

소스 트리를 정리하고 kernel-package 매개변수를 리셋한다. /usr/sbin/make-kpkg clean를 실행하라.

이제 커널을 컴파일하라. fakeroot /usr/sbin/make-kpkg --revision=custom.1.0 kernel_image. 버전 번호 "1.0"은 마음대로 바꿀 수 있다. 이것은 단지 커널 빌드를 기록하기 위해 사용하는 버전 번호일 뿐이다. 마찬가지로 "custom" 대신에(i.e., host name) 좋아하는 어떤 낱말이든 넣을 수 있다. 커널 컴퍼일은 여러분의 컴퓨터의 성능에 따라 잠시 시간이 걸릴 것이다.

PCMCIA 지원이 필요하다면 pcmcia-source 패키지를 설치할 필요가 있다. 루트로 gzip으로 압축한 tar 파일을 /usr/src 디렉토리에 압축을 푼다(모듈이 있어야 할 곳에 놓여야 한다는 것이 중요하다. 즉, /usr/src/modules에 있어야 한다). 그리고 나서 루트로 make-kpkg modules_image를 실행하라.

컴파일이 끝나면 다른 패키지처럼 여러분의 만든 커널을 설치할 수 있다. 루트로 dpkg -i ../kernel-image-2.0.36-*subarch*_custom.1.0_i386.deb 를 실행하라. *subarch* 부분은 어떤 커널 옵션을 지정했느냐에 따라 선택적인 sub-architecture "i586"처럼, 이다. dpkg -i kernel-image...가 몇가지 다른 훌륭한 지원 파일과 함께 커널을 설치할 것이다. 예를 들면 System.map이 정확히 설치될 것이고(커널 문제를 디버깅하는데 유용한) /boot/config--2.0.36이 현재 설정 세트를 포함한 상태로 설치될 것이다. 새로운 kernel-image-2.0.36 패키지는 부팅할 때 필요한 커널 이미지 정보를 업데이트하기 위해 자동으로 lilo를 사용할 만큼 똑똑하다. 따라서 lilo를 재실행할 필요가 없다. 모듈 패키지를 만들었다면 모듈 패키지도 설치할 필요가 있다.

시스템을 재부팅할 시간이다. 앞 단계에서 나온 어떤 경고라도 주의깊게 읽고 shutdown -r now를 실행하라.

kernel-package에 관한 정보가 더 필요하면, /usr/doc/kernel--package를 읽을 것.

---

[1] kmod가 kerneld를 대신하는 것에 주의하고 "Kernel module loader"를 대신 선택해야 한다. 데비안 2.1는 리눅스 커널 2.2를 완전하게 지원하지 않는다. '데비안 2.1과 같이 리눅스 커널 2.2 사용하기' on page 46를 참고하기 바란다.

제 5 절 데비안 2.1과 같이 리눅스 커널 2.2 사용하기

데비안 2.1는 리눅스 커널 2.2와 함께 사용할 수 있다고 보증할 수는 없다. 그러나 ftp://ftp.debian.org/debian/dists/unstable/에서 일부 패키지를 내려받아서 설치하더라도 시스템이 제 기능을 하게 할 수 있다.

커널 2.2와 호환되지 않는다고 알려진 많은 패키지가 있다. Unofficial Debian GNU/Linux 2.2 Checklist (http://www.debian.org/releases/2.1/running-kernel-2.2)가 이것과 관련해서 도움이 될 것이다. 업데이트에 대해서는 데비안 2.1 페이지 (http://www.debian.org/releases/2.1/)를 참고하기 바란다.

제 9 장

부트 플로피에 관한 기술적 정보

제 1 절 소스 코드

boot-floppies 패키지는 설치 디스켓에 필요한 모든 소스코드와 문서를 담고 있다.

제 2 절 Rescue Floppy

Rescue Floppy는 Ext2 파일시스템이고 (또는 아키텍처에 따라 FAT 파일시스템), Ext2 또는 FAT 디스크를 마운트할 수 있는 모든 것으로부터 Rescue Floppy에 액세스할 수 있어야 한다. The Linux 커널은 linux 안에 있다. root.bin은 gzip으로 압축한 1.4MB의 Minix 또는 Ext2 파일시스템의 디스크 이미지이며, 램디스크에 적재되어 파일시스템으로 사용할 것이다.

제 3 절 Rescue Floppy 커널 교체하기

만약 여러분이 Rescue Floppy의 커널을 교체해야 할 필요가 있다면, 새로운 커널을 만들 때 아래에 열거하는 사항은 반드시 커널안에 포함시켜야 하며 모듈로 만들어서는 안된다.

- 초기화 RAM 디스크
- FAT, Minix, Ext2 파일시스템 (몇몇 아키텍처는 FAT나 Minix 파일시스템이 필요하지 않다 – 소스를 보라)
- ELF 실행파일

새로운 커널을 Rescue Floppy의 linux에 복사하고 나서, 플로피에 들어있는 쉘 스크립트 rdev.sh를 실행하라.

여러분은 또 Drivers Floppy의 modules.tgz 파일도 교체하고자 할 것이다. 이 파일은 단순히 /lib/modules/\textit{kernel--ver}에 있는 모듈 파일들을 gzip으로 압축하여 tar로 묶은 것이다. modules.tgz 파일에는 모듈 파일 뿐만 아니라 그 모듈 파일이 위치하고 있는 디렉터리 정보까지 포함해야 하므로 루트 파일시스템 (/)에서 modules.tgz 파일을 만들어야 한다.

제 4 절 베이스 플로피

base 플로피들의 첫 부분에는 512 바이트 헤더가 있고, 그 뒤에는 gzip으로 압축한 `tar` 아카이브의 일부분이 있다. 만약 헤더를 제거하고서 플로피의 내용물을 연결시키면, 결과는 압축된 tar 아카이브가 될 것이다. 아카이브는 하드 디스크에 설치할 베이스 시스템을 담고 있다. 이 아카이브를 설치하면 반드시 설치 시스템의 “Configure the Base System”과 네트워크 설정을 마쳐야 하고 시스템이 사용가능하도록 운영체제 커널과 모듈을 설치해야 한다.

제 10 장

Administrivia

제 1 절 이 문서에 대해

이 문서는 "DebianDoc" DTD를 사용하여 SGML로 작성하였다. 출력양식은 debiandoc-sgml 패키지의 프로그램으로 만들었다.

이 문서의 유지 및 관리를 용이하게 하기 위하여 entity와 marked section과 같은 많은 SGML 형태를 사용했다. 이들의 역할은 프로그래밍 언어에서 사용되는 변수 및 조건의 역할과 어떤 의미에서 비슷하다. 이 문서의 SGML 소스는 여러가지 아키텍처에 대한 내용을 담고 있다. 문서의 특정 부분이 어떤 아키텍쳐에 한정된 내용임을 표시하기 위해 marked section을 사용한다.

제 2 절 이 문서에 기여하기

만약 이 문서와 관련된 문제나 제안이 있다면 boot-floppies패키지에 대해 버그 보고로 그것들을 제시할 수 있다. bug패키지를 보거나 데비안 버그 추적 시스템 (http://www.debian.org/Bugs/)을 읽어 보기 바란다. 여러분의 문제가 이미 보고되었는지 알아보려면 open bugs against boot–floppies (http://www.debian.org/Bugs/db/pa/lboot-floppies.html)를 확인하면 될 것이다. 이미 보고된 것이라면 여러분은 보충자료(addition corroboration)나 유용한 정보를 <XXXX@bugs.debian.org>에 제공할 수 있다. *XXXX*는 이미 보고된 버그의 번호이다.

더 좋은 방법은 이 문서의 SGML 소스의 복사본을 하나 구해서 이 문서의 패치를 만드는 것이다. SGML 소스는 boot-floppies에서 찾을 수 있으며, 최신 개정판은 unstable (ftp://ftp.debian.org/debian/dists/unstable/)에서 찾을 수 있다. CVS를 이용하는 것도 곧 가능할 것이다.

이 문서의 저자에게 직접적으로 연락하는 것은 *피해주기* 바란다. boot-floppies를 위한 메일링 리스트가 있으며, 이 설명서에 대한 토론도 겸하고 있다. 메일링 리스트는 <debian-boot@lists.debian.org>다. 이 리스트에 가입하는 법은 데비안 메일링 리스트 가입 (http://www.debian.org/MailingLists/subscribe) 페이지를 참고하기 바란다. 데비안 메일링 리스트 모음 (http://www.debian.org/Lists-Archives/)에서 그동안의 토론을 온라인으로 읽을 수 있다.

제 3 절 주요한 기여(Major Contributions)

아주 많은 데비안 사용자들과 개발자들이 이 문서에 기여했다. 특별히 Michael Schmitz (m68k 지원), Frank Neumann (Amiga를 위한 데비안 설치 설명서 (`http://www.informatik.uni-oldenburg.de/~amigo/debian_inst.html`)의 원저자), Arto Astala, Eric Delaunay (SPARC 정보), Tapio Lehtonen, 그리고 St僞慾hane Bortzmeyer에게 수많은 편집과 텍스트를 맡아 준 것에 대해 감사하고자 한다.

많은 문서들 가운데서 매우 유용한 텍스트와 정보를 Jim Mintha의 HOWTO for network booting (URL 사라짐), 데비안 FAQ (`http://www.debian.org/doc/FAQ/`), Linux/m68k FAQ (`http://www.linux-m68k.org/faq/faq.html`), Linux for SPARC Processors FAQ (`http://www.ultralinux.org/faq.html`), Linux/Alpha FAQ (`http://www.alphalinux.org/faq/FAQ.html`)에서 찾을 수 있었다. 이러한 자유롭게 이용 가능한 풍부한 정보를 제공하는 사람들도 반드시 인정받아야 한다.

제 4 절 Trademark Acknowledgement

모든 상표(trademark)는 각 상표(trademark) 소유자의 재산이다.