

Debian GNU/Linux 설치 안내서

December 14, 2005

저작권 © 2004, 2005 데비안 설치 프로그램 팀

법적 공지

이 설명서는 자유소프트웨어입니다; 이 문서를 GNU General Public License에 정해진 바에 의거하여 재배포하거나 수정할 수 있습니다. 라이선스에 관해서는 [부록 E](#)를 참조하시기 바랍니다.

Abstract

이 문서는 Intel x86(“i386”)아키텍처용 Debian GNU/Linux 3.1시스템(코드명 “sarge”)을 설치하는 절차가 쓰여 있습니다. 새로운 데비안 시스템을 만들어 나가는 방법에 대한 정보가 들어 있고 더 많은 정보가 들어 있는 다른 자료들을 안내합니다.

참고

i386용 설치 안내서의 내용은 거의 모두 업데이트되어 있는 상태이지만, sarge 공식 릴리스가 나온 이후에라도 일부 내용을 변경하거나 안내서의 구성을 재조정할 계획입니다. 이 설명서의 최신 버전은 인터넷의 **debian-installer** 홈페이지 <<http://www.debian.org/devel/debian-installer/>>에 있습니다. 이 홈페이지에는 다른 여러 언어의 번역판도 더 있을 것입니다.

i386용 Debian GNU/Linux 3.1 설치하기

우리는 당신이 데비안을 시도해보는 것이 매우 기쁘며 데비안의 GNU/Linux 배포본이 유일하다는 것을 알게 될 것이라고 확신합니다. Debian GNU/Linux은 세상에서 최고품질 소프트웨어를 함께 가져다 줄 것이며 그것을 완벽하게 통합시켜줄 것입니다. 우리는 진실로 부분의 합보다 더 확실한 결과를 가져다 줄 것이라고 믿습니다.

우리는 대부분의 사람이 이 설명서를 읽지 않고 데비안을 설치하기를 원한다고 이해하며 데비안 설치 프로그램은 이를 가능하도록 설계되었습니다. 현재 전체 설치 가이드를 읽을 시간이 없다면, 기본 설치 절차를 안내하며 더 자세한 토픽이나 잘못되었을 때를 위한 설치 하우투 문서를 읽기를 추천합니다. 설치 하우투 문서는 [부록 A](#)에서 찾을 수 있습니다.

우리는 당신이 이 설명서의 대부분을 읽을 시간을 가지고 더 정확하고 더 성공적인 설치 경험을 가질 수 있다고 희망합니다.

Contents

1	데비안에 오신 것을 환영합니다.	11
1.1	데비안이란?	11
1.2	GNU/리눅스란?	12
1.3	Debian GNU/Linux란?	13
1.4	데비안 GNU/Hurd란?	14
1.5	데비안 구하기	14
1.6	이 문서의 최신 버전을 얻는 법	15
1.7	이 문서의 구성	15
1.8	문서에 도움을 주실 분은 언제든지 환영합니다.	16
1.9	저작권과 소프트웨어 라이선스에 대하여	16
2	시스템 요구사항	19
2.1	지원하는 하드웨어	19
2.1.1	지원하는 아키텍처	19
2.1.2	CPU, 메인 보드, 비디오 지원	20
2.1.2.1	CPU	20
2.1.2.2	입출력 버스	21
2.1.3	그래픽 카드	21
2.1.4	노트북	21
2.1.5	다중 프로세서	21
2.2	설치 미디어	21
2.2.1	플로피	22
2.2.2	CD-ROM/DVD-ROM	22
2.2.3	하드디스크	22
2.2.4	USB 메모리	22
2.2.5	네트워크	23
2.2.6	유닉스 계열 혹은 GNU 시스템	23
2.2.7	지원하는 저장 장치	23
2.3	주변 장치 및 기타 하드웨어	23
2.4	GNU/리눅스를 위한 하드웨어 구입하기	24
2.4.1	독점적이거나 폐쇄된 하드웨어 피하기	24
2.4.2	Windows 전용 하드웨어	24
2.4.3	가짜 혹은 “가상” 패리티 RAM	25
2.5	메모리 및 디스크 공간 요구사항	25
2.6	네트워크 연결 하드웨어	25
3	Debian GNU/Linux 설치하기 전에	27
3.1	설치 과정의 개요	27
3.2	기존 데이터를 백업하십시오!	28
3.3	필요한 정보	29

3.3.1	문서	29
3.3.1.1	설치 매뉴얼	29
3.3.1.2	하드웨어 문서	29
3.3.2	하드웨어 정보가 있는 곳 찾기	29
3.3.3	하드웨어 호환성	30
3.3.4	네트워크 설정	31
3.4	최소 하드웨어 요구사항 맞추기	31
3.5	멀티 부팅 시스템에서 미리 파티션하기	32
3.5.1	DOS나 Windows에서 파티션하기	34
3.5.1.1	DOS나 Win-32나 OS/2에서 시작할 때 데이터 손실없이 다시 파티션하기	34
3.5.1.2	DOS용 파티션하기	35
3.6	설치하기 전에 하드웨어 및 운영 체제 설정	35
3.6.1	BIOS 설정 메뉴 들어가기	35
3.6.2	부팅 장치 선택	36
3.6.2.1	IDE 컴퓨터의 부팅 순서 바꾸기	36
3.6.2.2	SCSI 컴퓨터에서 부팅 순서 바꾸기	37
3.6.3	기타 BIOS 설정	37
3.6.3.1	CD-ROM 설정	37
3.6.3.2	Extended와 Expanded Memory	37
3.6.3.3	Virus Protection	37
3.6.3.4	Shadow RAM	38
3.6.3.5	Memory Hole	38
3.6.3.6	Advanced Power Management	38
3.6.4	주의해야 할 하드웨어 이슈	38
3.6.4.1	Turbo 스위치	39
3.6.4.2	Cyrix CPU 및 플로피 디스크 오류	39
3.6.4.3	주변 장치 하드웨어 설정	39
3.6.4.4	USB BIOS 지원 및 키보드	39
3.6.4.5	64 MB RAM 이상	40
4	시스템 설치 미디어 얻기	41
4.1	공식 Debian GNU/Linux CD-ROM 세트	41
4.2	데비안 미러에서 파일 내려받기	41
4.2.1	설치 이미지를 찾을 위치	42
4.3	디스크 이미지에서 플로피 만들기	42
4.3.1	리눅스나 UNIX 시스템에서 디스크 이미지 쓰기	42
4.3.2	DOS, Windows, OS/2에서 디스크 이미지 쓰기	43
4.4	USB 메모리 스틱 부팅에 필요한 파일 준비하기	43
4.4.1	파일 복사하기 — 쉬운 방법	43
4.4.2	파일 복사하기 — 유연한 방법	44
4.4.2.1	Intel x86에서 USB 스틱 파티션하기	44
4.4.2.2	ISO 이미지 추가하기	45
4.4.2.3	USB 스틱 부팅하기	45
4.5	하드 디스크 부팅에 필요한 파일 준비하기	46

4.5.1	LILO 혹은 GRUB 을 이용한 하드 디스크 설치 프로그램 부팅	46
4.6	TFTP 네트워크 부팅에 필요한 파일 준비하기	46
4.6.1	BOOTP 서버 준비하기	47
4.6.2	DHCP 서버 설정하기	48
4.6.2.1	DHCP 설정에서 PXE 부팅하게 만들기	48
4.6.3	TFTP Server 사용하기	49
4.6.4	TFTP 이미지를 적당한 위치에 옮기기	50
4.7	자동 설치	50
4.7.1	데비안 설치 프로그램을 이용한 자동 설치	50
5	설치 시스템 부팅하기	53
5.1	Intel x86에서 설치 프로그램 부팅하기	53
5.1.1	CD-ROM에서 부팅하기	53
5.1.2	LILO 혹은 GRUB 을 사용해 리눅스 부팅하기	53
5.1.3	USB 메모리 스틱에서 부팅하기	54
5.1.4	플로피에서 부팅하기	54
5.1.5	TFTP로 부팅하기	55
5.1.5.1	PXE를 지원하는 NIC 혹은 마더보드	55
5.1.5.2	네트워크 부트로姆이 들어 있는 NIC	55
5.1.5.3	Etherboot	56
5.1.6	부팅 프롬프트	56
5.2	부팅 파라미터	56
5.2.1	데비안 설치 프로그램 파라미터	57
5.3	설치 과정의 문제 해결	59
5.3.1	플로피 디스크 신뢰성	59
5.3.2	부팅 설정	59
5.3.3	자주 발생하는 Intel x86 설치 문제	60
5.3.3.1	PCMCIA 설정 단계에서 시스템 멈춤	60
5.3.3.2	USB 모듈을 읽어들이다가 시스템 멈춤	61
5.3.4	커널 시작 메시지 해석하기	61
5.3.5	버그 보고	61
5.3.6	설치 보고 제출	61
6	데비안 설치 프로그램 사용하기	63
6.1	설치 프로그램이 동작하는 방식	63
6.2	구성 요소 소개	64
6.3	각 구성 요소 사용하기	66
6.3.1	데비안 설치 프로그램 및 하드웨어 설정 준비하기	66
6.3.1.1	사용 가능 메모리 검사	66
6.3.1.2	언어 선택	67
6.3.1.3	국가 선택	67
6.3.1.4	키보드 선택하기	67
6.3.1.5	데비안 설치 프로그램 ISO 이미지 찾기	67
6.3.1.6	네트워크 설정하기	68
6.3.2	파티션하기 및 마운트 위치 선택	68

6.3.2.1	디스크 파티션하기	69
6.3.2.2	논리 볼륨 관리자 (LVM) 설정하기	70
6.3.2.3	멀티디스크 장치 설정하기 (소프트웨어 RAID)	71
6.3.3	베이스 시스템 설치하기	73
6.3.3.1	베이스 시스템 설치	73
6.3.4	시스템을 부팅 가능하게 만들기	73
6.3.4.1	다른 운영 체제 찾기	74
6.3.4.2	하드 디스크에 GRUB 부트로더 설치	74
6.3.4.3	하드 디스크에 LILLO 부트로더 설치	74
6.3.4.4	부트로더 없이 계속	75
6.3.5	첫 단계 마치기	75
6.3.5.1	설치 마치기 및 다시 시작하기	76
6.3.6	기타	76
6.3.6.1	설치 로그 저장	76
6.3.6.2	셸 사용하기 및 로그 보기	76
6.3.6.3	네트워크를 통해 설치	76
6.3.6.4	debian-installer 안에서 base-config 실행하기	78
7	새로운 데비안 시스템으로 부팅하기	79
7.1	진실의 시간	79
7.2	데비안 부팅 후 (기본) 설정	79
7.2.1	시간대 설정	79
7.2.2	사용자와 암호 설정	80
7.2.2.1	Root 암호 설정	80
7.2.2.2	일반 사용자 만들기	80
7.2.3	PPP 설정	80
7.2.3.1	PPP over Ethernet (PPPOE) 설정	81
7.2.4	APT 설정	82
7.2.4.1	네트워크 패키지 소스(Sources) 설정	82
7.2.5	패키지 설치	83
7.2.5.1	aptitude 를 사용한 고급 패키지 선택	84
7.2.6	소프트웨어 설치 중의 프롬프트	84
7.2.7	메일 배달 에이전트(Mail Transport Agent) 설정	85
7.3	로긴	85
8	다음 단계 및 그 다음에 할 일	87
8.1	유닉스를 처음 접한다면	87
8.2	데비안에 익숙해지기	87
8.2.1	데비안 꾸러미 시스템	87
8.2.2	프로그램 버전 관리	88
8.2.3	CRON 작업 관리	88
8.3	DOS 및 Windows 다시 살리기	88
8.4	그 외에 읽을 것과 정보	89
8.5	새 커널 컴파일하기	90
8.5.1	커널 이미지 관리	90

A	설치 방법	93
A.1	사전 준비	93
A.2	설치관리자 시작하기	93
A.2.1	CDROM	93
A.2.2	플로피	94
A.2.3	USB 메모리 스틱	94
A.2.4	네트워크 부팅	95
A.2.5	하드 디스크 부팅	95
A.3	설치	95
A.4	설치에 대한 결과를 보내주세요	96
A.5	마지막..	96
B	데비안의 하드디스크 파티션	97
B.1	파티션을 나눌 때 파티션의 크기 등 고려할 점	97
B.2	디렉토리 구조	98
B.3	권장하는 파티션 구조	99
B.4	리눅스에서의 디바이스 이름	100
B.5	데비안의 파티션용 프로그램	101
B.5.1	Intel x86에서 파티션하기	102
C	여러가지 내용들	105
C.1	미리 설정 파일 예제	105
C.2	리눅스 장치	114
C.2.1	마우스 설정하기	115
C.3	작업마다 필요한 디스크 공간	115
C.4	유닉스/리눅스 시스템에서 Debian GNU/Linux 설치하기	116
C.4.1	시작하기	116
C.4.2	debootstrap 설치	117
C.4.3	debootstrap 실행	118
C.4.4	베이스 시스템 설정	118
C.4.4.1	파티션 마운트하기	118
C.4.4.2	키보드 설정	120
C.4.4.3	네트워크 설정하기	120
C.4.4.4	시간대, 사용자, APT 설정하기	121
C.4.4.5	로케일 설정하기	121
C.4.5	커널 설치	122
C.4.6	부트로더 설정하기	122
C.5	병렬 라인 IP를 (PLIP) 이용해 Debian GNU/Linux 설치하기	123
C.5.1	요구 사항	123
C.5.2	소스 설정하기	123
C.5.3	타겟 설치	124
D	관리	125
D.1	문서 정보	125
D.2	이 문서에 기여하기	125

D.3	중요 기부자들	126
D.4	상표권 표시	126
E	GNU General Public License	127
E.1	Preamble	127
E.2	GNU GENERAL PUBLIC LICENSE	128
E.3	How to Apply These Terms to Your New Programs	132

Chapter 1

데비안에 오신 것을 환영합니다.

이 장에서는 데비안 프로젝트와 Debian GNU/Linux에 대하여 간략히 설명하려고 합니다. 데비안 프로젝트의 역사와 Debian GNU/Linux 배포판에 대하여 이미 알고 있는 분들은 다음 장으로 넘어가셔도 됩니다.

1.1 데비안이란?

데비안은 자유 소프트웨어를 개발하고 자유 소프트웨어 재단(Free Software Foundation)의 이상을 널리 알리는데 전념하는 순수한 자원 봉사 조직입니다. 데비안 프로젝트는 1993년, 이안 머독이 비교적 새로운 커널에 기반한 완전하고 일관된 소프트웨어 배포판을 만드는 데 참여할 소프트웨어 개발자들을 공개적으로 모집하면서부터 시작되었습니다. 처음에는 자유 소프트웨어 재단(Free Software Foundation) <<http://www.fsf.org/>>의 자금 지원을 받았고 GNU <<http://www.gnu.org/gnu/the-gnu-project.html>> 철학에 영향을 받은 이 비교적 작은 규모의 열성적인 집단은, 해를 거듭할 수록 900여 명의 데비안 개발자가 참여하는 조직으로 성장했습니다.

데비안 개발자는 웹 <<http://www.debian.org/>> 및 FTP <<ftp://ftp.debian.org/>> 사이트 관리, 그래픽 디자인, 소프트웨어 라이선스의 법률적 해석, 문서 작성, 소프트웨어 꾸러미 관리를 포함하는 다양한 활동에 참여합니다.

우리의 철학을 전하고 데비안의 원칙을 믿는 개발자들을 끌어 모으기 위해 데비안 프로젝트는 다수의 문서를 발표하였습니다. 이 문서에는 우리의 가치를 개략적으로 서술하고 데비안 개발자가 되는 게 무엇을 뜻하는 지에 대한 지침 역할을 합니다.

- 데비안 Social Contract <http://www.debian.org/social_contract>은 자유소프트웨어 공동체에 대한 데비안의 약속입니다. Social Contract에 따르기로 동의한 사람은 누구나 메인테이너 <<http://www.debian.org/doc/maint-guide/>>가 될 수 있습니다. 어떤 메인테이너든지 새로운 소프트웨어를 데비안에 추가할 수 있습니다. 단 그 소프트웨어가 우리의 자유소프트웨어의 기준에 맞아야 하고, 꾸러미가 우리의 품질 표준에 맞아야 합니다.
- Debian Free Software Guidelines <http://www.debian.org/social_contract#guidelines>는 자유 소프트웨어에 대한 데비안의 기준을 단순명료하게 설명한 것입니다. DFSG는 자유소프트웨어

운동에 있어서 매우 영향력 있는 문서로, 오픈소스 정의(The Open Source Definition) <http://opensource.org/docs/definition_plain.html>의 근간이 되었습니다.

- 데비안 정책 매뉴얼 <<http://www.debian.org/doc/debian-policy/>>은 데비안 프로젝트의 품질 표준을 상세하게 적은 내역입니다.

데비안 개발자는 여러 프로젝트에 참가할 수 있습니다. 어떤 사람은 데비안 고유의 프로젝트에, 또 어떤 사람들은 여러 리눅스 공동체에 참가하기도 합니다. 예를 들면 다음과 같습니다:

- Linux Standard Base <<http://www.linuxbase.org/>>(LSB)는 기본적인 GNU/리눅스 시스템을 표준화하기 위한 프로젝트로, 써드파티 소프트웨어와 하드웨어 개발자들이 특정 GNU/리눅스 배포판이 아닌 일반적인 리눅스에 대한 프로그램과 디바이스 드라이버를 쉽게 설계할 수 있도록 정한 것입니다.
- Filesystem Hierarchy Standard <<http://www.pathname.com/fhs/>>(FHS)는 리눅스 파일 시스템의 구성을 표준화하는 프로젝트입니다. FHS는 개발한 꾸러미가 여러가지 GNU/리눅스 배포판에 어떻게 설치되는 지에 대해 고민할 필요 없이 소프트웨어 개발자는 프로그램 설계에만 전념할 수 있도록 하기 위한 프로젝트입니다.
- 데비안 주니어 <<http://www.debian.org/devel/debian-jr/>>는 데비안 내부 프로젝트로 어린이 사용자들이 사용할 만한 데비안을 만드는 프로젝트입니다.

데비안에 관한 보다 일반적인 정보는 데비안 FAQ <<http://www.debian.org/doc/FAQ/>>를 참고하기 바랍니다.

1.2 GNU/리눅스란?

리눅스는 운영 체제입니다. 여러 프로그램들의 모음으로, 이 프로그램을 이용해 자기 컴퓨터를 사용하고 또 다른 프로그램을 실행합니다.

운영체제는 컴퓨터에 필요한 여러 가지 기본적인 프로그램으로 구성되어 있기 때문에, 사용자와 의사소통을 하거나 사용자로부터 지시를 받습니다. 이를 테면, 하드 디스크, 테이프, 프린터로 데이터를 내보내거나 읽어들이며, 메모리의 사용을 제어하고, 다른 프로그램을 실행합니다. 운영체제의 가장 중요한 부분이 바로 커널입니다, GNU/리눅스 시스템에 있어서 리눅스는 커널 부분을 말합니다. 시스템의 나머지 부분들은 다른 프로그램들로 구성되며, 대부분 GNU 프로젝트에 의해 작성된 것들입니다. 리눅스 커널 그 자체로는 시스템을 구성할 수 없기 때문에, 흔히들 리눅스라고 호칭하는 시스템을 GNU/리눅스라는 명칭으로 사용하시기를 바랍니다.

리눅스는 유닉스 (Unix) 운영체제를 모델로 한 것입니다. 애초부터 리눅스는 멀티 태스킹, 멀티 유저 시스템으로 설계되었습니다. 이러한 사실만으로도 리눅스를 잘 알려진 여타 운영체제와 차별화할 수 있습니다. 그러나, 리눅스는 여러분이 생각하는 것보다 훨씬 다른 점이 많습니다. 다른 운영체제들과는 달리, 어느 누구도 리눅스를 소유하지 않으며 그 대부분이 자원 봉사자들에 의해 개발되었습니다.

후에 GNU/리눅스라고 불리게 된 시스템의 개발은 1984년에 시작되었으며, 이 때 자유 소프트웨어 재단(FSF) <<http://www.gnu.org/>>은 유닉스와 유사한 운영체제의 개발을 시작하면서 그 이름을 GNU 라고 하였습니다.

GNU 프로젝트는 유닉스 (Unix™) 및 리눅스처럼 유닉스와 유사한 운영체제에서 공통으로 사용하기

위한 일련의 자유 소프트웨어 도구를 개발해 왔습니다. 이러한 도구를 이용하여 사용자는 파일을 복사 또는 삭제하는 아주 일상적인 작업부터, 프로그램을 작성하고 컴파일하거나 여러 형태의 문서에 대한 편집에 이르기까지 다양한 작업을 할 수 있습니다.

수많은 그룹과 개인들이 리눅스에 공헌했지만, 최대의 단독 공헌자는 자유소프트웨어 재단으로, 자유 소프트웨어 재단은 리눅스에서 사용하는 도구의 대부분을 개발했고 동시에 리눅스가 생겨나게 했던 철학과 공동체를 만들어내었습니다.

리눅스 커널 <<http://www.kernel.org/>>은, 리누스 토발즈라는 핀란드의 컴퓨터 과학 학생이 1991년에 유즈넷 뉴스그룹인 comp.os.minix에 Minix를 대체하는 커널의 초기 버전을 발표했을 때 처음으로 세상에 그 모습을 드러냈습니다. 자세한 것은 리눅스 인터네셔널의 리눅스 역사 페이지 (Linux History Page) <<http://www.li.org/linuxhistory.php>>를 참고하기 바랍니다.

리누스 토발즈는 몇 명의 신뢰할 수 있는 대표자들의 도움으로 수백명의 개발자들의 작업들을 조율하고 있습니다. 커널 트래픽 (Kernel Traffic) <<http://www.kerneltraffic.org/kernel-traffic/index.html>>은 linux-kernel 메일링 리스트의 토론을 주 단위로 훌륭하게 요약해 놓았습니다. linux-kernel 메일링 리스트에 대한 자세한 정보는 linux-kernel 메일링 리스트 FAQ <<http://www.tux.org/lkml/>>에서 찾아 보기 바랍니다.

리눅스 사용자는 소프트웨어 선택에 있어서 상당한 자유를 갖고 있습니다. 예를 들어, 리눅스 사용자는 12개의 다른 커맨드 라인 셸, 다수의 그래픽 데스크탑에서 원하는 걸 선택할 수 있습니다. 이와 같은 다양한 선택의 폭 때문에, 셸이나 데스크탑 같이 자신들이 변경할 수 있는 것에 대하여 전혀 생각하지 못 했던 사용자들이 당황스러워 하기도 합니다.

또한 리눅스는 여타 운영체제에 비해 시스템이 멈추는 경우가 적고, 동시에 둘 이상의 프로그램을 실행하는 성능이 월등하고, 보안에 강합니다. 리눅스는 서버 시장에서 가장 성장이 빠른 운영체제입니다. 최근에 리눅스는 가정과 업무용 사용자에게도 잘 알려져 나가고 있습니다.

1.3 Debian GNU/Linux란?

데비안의 철학 및 방법론과 GNU 도구, 리눅스 커널, 그리고 기타 중요한 자유소프트웨어들의 조합은 Debian GNU/Linux이라 불리는 독특한 배포판을 형성합니다. 이 배포판은 수많은 소프트웨어 꾸러미로 구성됩니다. 배포판의 각 꾸러미는 실행파일, 스크립트, 문서, 설정 정보가 들어 있으며 메인테이너가 관리합니다. 메인테이너는 각 꾸러미를 항상 최신으로 유지하고, 버그 리포트를 추적하고, 꾸러미로 만든 소프트웨어의 원저자들과 연락을 하고 있습니다. 데비안의 거대한 사용자 기반에 버그 추적 시스템과 결합해 문제점을 빨리 찾아내고 수정합니다.

데비안의 세세한 관심때문에 높은 질의, 안정적이고, 확장 가능한 배포판을 만들어냅니다. 설치 도구를 어떻게 설정하느냐에 따라서 방화벽부터 데스크탑 공학용 워크스테이션과 하이엔드 네트워크 서버까지 다양한 역할을 할 수 있습니다.

데비안은 그 기술적 우수성과 리눅스 공동체의 필요와 기대에 대한 적극적인 참여때문에 고급 사용자들에게 특히 인기가 있습니다. 또한 데비안은 지금은 리눅스에 일반적인 기능이 된 많은 기능들을 새로 도입해 왔습니다.

예를 들어보면, 소프트웨어의 설치와 제거를 쉽게 할 수 있는 꾸러미 관리 시스템을 포함한 최초의 리눅스 배포판이 데비안입니다. 또 재설치하지 않고 업그레이드 할 수 있는 최초의 배포판입니다.

데비안은 리눅스 개발에 있어 여전히 앞서 나가고 있습니다. 데비안의 개발 과정은 오픈 소스 개발 모델이 (완전한 운영체제의 개발 및 관리같은 복잡한 일이라고 해도) 얼마나 잘 진행되는 지를 보여주는 한 예입니다.

데비안과 다른 리눅스 배포판을 구분하는 가장 큰 특징은 데비안의 꾸러미 관리 시스템입니다. 꾸러미 관리 도구를 이용해, 데비안 시스템은 시스템에 설치된 꾸러미들을 완벽하게 제어합니다. 꾸러미 하나 하나를 설치하는 일에서부터 자동으로 전체 운영체제를 업데이트하는 것도 할 수 있습니다. 또한 각 꾸러미마다 업데이트를 하지 않도록 조정할 수도 있습니다. 또 꾸러미 관리 시스템에 직접 컴파일한 소프트웨어에 대해 등록해 놓고 어떤 의존성이 필요한지 지정할 수 있습니다.

여러분의 시스템을 “트로이 목마” 및 여타 악의적인 소프트웨어로부터 보호하기 위하여, 데비안의 서버는 해당 꾸러미에 등록된 데비안 메인테이너들이 업로드한 꾸러미를 검사합니다. 데비안 꾸러미를 만드는 사람들은 안전한 방법으로 자신들의 패키지를 설정하기 위하여 상당한 주의를 기울입니다. 배포된 꾸러미에 보안 문제가 발생하면, 상당히 빨리 수정 버전이 나옵니다. 데비안의 간단한 업데이트 옵션을 이용하여, 인터넷을 통해 보안 수정 버전을 자동으로 다운로드하고 설치할 수 있습니다.

여러분의 Debian GNU/Linux 시스템에 대하여 지원을 받고 데비안 개발자들과 연락을 할 수 있는 주요한 방법, 그리고 최상의 방법이기도 한 방법은 데비안 프로젝트에서 관리하는 많은 메일링 리스트 (이 글이 작성된 시점에 약 160개 이상)를 이용하는 것입니다. 메일링 리스트에 가입하려면, 가장 쉬운 방법으로 데비안 메일링 리스트 가입 페이지 <<http://www.debian.org/MailingLists/subscribe>>를 방문하여 거기에 나타난 양식을 채우면 됩니다.

1.4 데비안 GNU/Hurd란?

데비안 GNU/Hurd는 리눅스 모노리틱 커널을 GNU Hurd로 대체하는 데비안 GNU 시스템으로, GNU 마크 마이크로 커널에서 동작하는 서버의 모음입니다. Hurd는 아직까지 개발이 끝나지 않았고, 일상적인 용도로 사용하기에는 적합하지 않습니다만, 개발은 계속 진행 중입니다. 현재 Hurd는 i386 아키텍처를 대상으로만 개발중입니다만, 일단 시스템이 더 안정되면 다른 아키텍처로 포팅될 것입니다.

보다 자세한 정보는 데비안 GNU/Hurd 포팅 페이지 <<http://www.debian.org/ports/hurd/>>와 <debian-hurd@lists.debian.org> 메일링 리스트를 참고하시기 바랍니다.

1.5 데비안 구하기

공식 데비안 CD를 인터넷에서 다운로드하거나 구입할 수 있는 곳에 대한 정보는 배포판 웹 페이지 <<http://www.debian.org/distrib/>>를 참고하시기 바랍니다. 데비안 미러 목록 <<http://www.debian.org/distrib/ftplist>>에 공식적인 데비안 미러 사이트가 모두 망라되어 있으니, 여러분에게 가장 가까운 미러 사이트를 쉽게 찾을 수 있을 것입니다.

데비안은 설치한 뒤에 쉽게 업그레이드 할 수 있습니다. 일단 설치 과정을 마친 시스템은 필요할 때 언제든지 업그레이드할 수 있습니다.

1.6 이 문서의 최신 버전을 얻는 법

이 문서는 수시로 개정됩니다. Debian GNU/Linux 시스템의 3.1 릴리스에 대한 최신 정보에 대해서는 데비안 3.1 페이지 <<http://www.debian.org/releases/sarge/>>를 확인하기 바랍니다. 이 설치문서의 최신 버전은 공식 데비안 설치 안내서 페이지 <<http://www.debian.org/releases/sarge/i386/>>을 통해서도 구할 수 있습니다.

1.7 이 문서의 구성

이 문서는 데비안을 처음 사용하는 분들을 위한 설명서입니다. 이 문서는 가능한한 읽는 사람이 전문지식을 갖추지 않았다고 생각하고 쓰여졌습니다. 하지만 컴퓨터에 장착된 하드웨어가 동작하는 방법에 대해서는 보편적인 이해를 하고 있다고 가정했습니다.

전문적인 사용자들도 이 문서에서 여러 가지 흥미로운 정보를 찾을 수 있습니다. 예를 들면 최소 설치 크기, 데비안 설치 시스템이 지원하는 하드웨어 등과 같은 내용이 있습니다. 전문적인 사용자는 이 문서를 처음부터 읽지 않고 필요한 부분만 읽어 나가기를 권합니다.

이 매뉴얼은 설치 과정의 처음부터 끝까지, 일련의 과정의 순서로 되어 있습니다. 다음은 Debian GNU/Linux 을 설치하는 과정으로, 각 과정에 관련된 문서 부분이 같이 쓰여 있습니다:

1. **2장**에서, 하드웨어가 설치 시스템을 사용할 때 필요한 요구사항을 만족하는 지 판단합니다.
2. **3장**에서, 데비안을 설치하기에 앞서 여러분의 시스템을 백업하고, 필요한 계획을 세우고 하드웨어를 설정합니다. 멀티 부팅을 생각하고 있다면, 데비안을 설치하려는 하드디스크에 파티션 가능한 공간을 만들어야 할 수도 있습니다.
3. **4장**에서, 설치 방법에 따라 필요한 설치 파일을 구합니다.
4. **5장** 설치 시스템으로 부팅하는 방법을 설명합니다. 이 과정에서 문제가 발생한 경우에 문제를 해결하는 과정도 이 장에서 설명합니다.
5. **6장**에 따라 실제 설치를 진행합니다. 이 과정에서 여러분이 사용하는 언어의 선택, 주변장치 드라이버 모듈 설정, 네트워크 설정, 하드 디스크 파티션 과정 그리고 최소 동작 시스템의 설치 과정을 진행합니다. 특히 CD로 설치하는 경우가 아니라면 네트워크가 설정했을 때 나머지 설치 파일을 해당하는 파일을 모두 데비안 서버로부터 직접 받아 올 수 있습니다. (데비안 시스템의 파티션 설정에 관한 설명은 **부록 B**에 있습니다.)
6. 새로 설치한 베이스 시스템으로 부팅하고 난 후 **7장**를 통해 일부 추가 설정 작업을 하십시오.
7. **7.2.5절**에서 소프트웨어를 추가로 설치하십시오.

일단 시스템을 설치했으면, **8장**를 읽으면 됩니다. 이 장은 유닉스와 데비안에 대하여 더 많은 정보를 얻을 수 있는 다른 읽을 거리를 소개하고, 커널을 교체하는 방법을 설명하고 있습니다.

마지막으로, 이 문서에 대한 정보와 이 문서에 기여하는 방법은 **부록 D**에 있습니다.

1.8 문서에 도움을 주실 분은 언제든지 환영합니다.

어떤 형태든지 이 문서에 대한 도움, 제안, 특히 패치에 대해 깊이 감사드립니다. 현재 작업중인 버전은 <<http://d-i.alioth.debian.org/manual/>>에 있습니다. 이 홈페이지에는 다른 아키텍처용 문서와 다른 여러 언어의 번역판도 더 있을 것입니다.

문서의 소스도 공개되어 있습니다. **부록 D**를 보시면 기여하는 방법에 관하여 더 많은 정보를 얻을 수 있습니다. 제안, 조언, 패치 그리고 버그 리포트는 언제나 환영합니다. (버그를 리포트할 때는 **debian-installer-manual** 꾸러미를 이용하시고, 해당 문제가 이미 보고된 것인지 먼저 확인하시기 바랍니다.)

1.9 저작권과 소프트웨어 라이선스에 대하여

여러분들은 대부분의 상용 소프트웨어에 따라오는 라이선스를 어느 정도라도 읽어 본 적이 있을 것입니다. 그런 라이선스들은 보통 하나의 컴퓨터에 한 카피만 사용하도록 규정하고 있습니다. 데비안 시스템의 라이선스는 이들과 전혀 다릅니다. 얼마든지 한 카피로 학교나 회사의 모든 컴퓨터에 설치하십시오. 설치 미디어를 여러분들의 친구에게 빌려주고 컴퓨터에 설치하는 것을 도와주십시오! 수천장을 복사하고, 판매하십시오. (약간 제한이 있습니다.) 데비안은 자유 소프트웨어를 기반으로 하기 때문에 여러분들은 시스템을 설치하고 사용함에 있어 자유롭습니다.

소프트웨어를 자유라고 부르는 것은 소프트웨어의 저작권이 없다는 것이 아니며, 자유 소프트웨어를 배포하는 CD는 반드시 무료라는 것이 아닙니다. 자유소프트웨어는, 부분적인 의미로는 개개의 프로그램의 라이선스가 배포나 사용에 대하여 비용을 지불할 필요가 없다는 의미입니다. 또 자유 소프트웨어는 누구라도 소프트웨어를 확장하고, 개작하고, 수정할 수 있을 뿐만 아니라, 그 수정한 결과물도 배포할 수 있다는 것을 의미합니다.

참고

데비안 프로젝트는 사용자들의 실용적인 목적으로, 일부 자유소프트웨어의 기준에 맞지 않는 꾸러미도 일부 배포하고 있습니다. 이 꾸러미는 공식 배포판에 포함되지 않으며, 데비안 미러 사이트나 써드파티 CD-ROM에 들어 있는 contrib이나 non-free에서 구할 수 있습니다. 데비안 아카이브의 구조나 내용에 대해서는 “데비안 FTP 아카이브”의 데비안 FAQ <<http://www.debian.org/doc/FAQ/>>를 참고하기 바랍니다.

시스템에 들어 있는 다수의 프로그램의 라이선스는 “GPL”로 알려진 *GNU General Public License*입니다. GPL 라이선스의 프로그램의 바이너리를 배포할 때 “소스 코드”를 같이 제공해야 합니다. 즉 누구라도 소프트웨어를 수정할 수 있도록 하는 것이 이 라이선스의 규정입니다. 그러한 규정때문에 그러한 프로그램의 모든 소스코드¹가 데비안 시스템에 들어 있습니다.

데비안 프로그램에 사용한 저작권과 소프트웨어 라이선스는 GPL 외에도 여러가지가 있습니다. 꾸러미를 설치하면 `/usr/share/doc/꾸러미이름/copyright` 파일을 보면 해당 꾸러미의 저작권과 라이선스를 볼 수 있습니다.

¹데비안 소스 꾸러미를 찾고, 압축을 풀고, 바이너리를 빌드하는 방법에 대해서는 데비안 FAQ <<http://www.debian.org/doc/FAQ/>>의 “Basics of the Debian Package Management System” 부분을 보십시오.

라이선스 및 어떻게 데비안에서 소프트웨어가 메인 배포판에 포함되는 지 여부를 결정하는 지에 대한 정보는 Debian Free Software Guidelines <http://www.debian.org/social_contract#guidelines>를 참고하십시오.

가장 중요한 법적인 고지는, 이 소프트웨어는 어떠한 보증도 하지 않는다는 것입니다. 이런 소프트웨어를 만든 프로그래머는 공동체의 이익을 위해 만든 것입니다. 어떤 목적에 대해서도 소프트웨어의 적합성을 보장하지 않습니다. 하지만, 소프트웨어가 자유소프트웨어이기 때문에 여러분에 목적에 맞게 소프트웨어를 수정하는 권리는 여러분에 있습니다. 또한 이런 방법으로 다른 사람이 소프트웨어를 확장하여 생긴 변화로 얻은 이익을 누릴 권리도 여러분에게 있습니다.

Chapter 2

시스템 요구사항

여기에는 데비안을 처음 설치할 때 어떤 하드웨어가 필요한지에 대한 정보가 들어 있습니다. 또 GNU 및 리눅스에서 지원하는 하드웨어에 관한 더 자세한 정보가 들어 있는 웹페이지 주소도 들어 있습니다.

2.1 지원하는 하드웨어

리눅스 커널과 GNU 툴셋이 필요한 것 외에는 데비안에 특별히 필요한 하드웨어는 따로 없습니다. 그러므로 리눅스 커널, `libc`, `gcc` 등이 포팅되어 있는 아키텍처나 플랫폼 중에서 데비안 포팅이 있는 경우라면 데비안을 돌릴 수 있습니다. 데비안에서 테스트한 Intel x86 아키텍처 시스템에 대한 자세한 정보는 <<http://www.debian.org/ports/i386/>> 페이지에 있는 포팅 페이지를 참고하십시오.

여기서는 Intel x86 아키텍처에서 지원하는 여러가지 하드웨어들을 모두 설명하지는 않고, 일반적인 정보와 추가 정보가 들어 있는 웹사이트를 써 놓았습니다.

2.1.1 지원하는 아키텍처

데비안 3.1 버전은 11개의 아키텍처를 지원하고 각 아키텍처별로 “flavors”라 알려진 몇가지 변종을 지원합니다.

아키텍처	데비안 명칭	서브 아키텍처	변종
인텔 x86 기반	i386		vanilla
			speakup
			linux26
모토로라 680x0	m68k	Atari	atari
		Amiga	amiga
		68k 매킨토시	mac
		VME	bvme6000
			mvme147

아키텍처	데비안 명칭	서브 아키텍처	변종
			mvme16x
DEC Alpha	alpha		
Sun SPARC	sparc		sun4cdm sun4u
ARM 및 StrongARM	arm		netwinder riscpc shark lart
IBM/모토로라 PowerPC	powerpc	CHRP	chrp
		PowerMac	pmac
		PPeP	prep
		APUS	apus
HP PA-RISC	hppa	PA-RISC 1.1	32
		PA-RISC 2.0	64
인텔 ia64 기반	ia64		
MIPS (big endian)	mips	SGI Indy/Indigo 2	r4k-ip22 r5k-ip22
			Broadcom BCM91250A (SWARM)
MIPS (little endian)	mipsel	Cobalt	cobalt
		DECstation	r4k-kn04 r3k-kn02
			Broadcom BCM91250A (SWARM)
IBM S/390	s390	VM-reader 및 DASD에서 IPL	generic
		테이프에서 IPL	tape

이 문서는 *Intel x86* 아키텍처에서의 설치에 대해 다루고 있습니다. 데비안이 지원하는 다른 아키텍처에 관한 정보를 찾고 있다면 데비안 포트 <<http://www.debian.org/ports/>> 페이지를 보십시오.

2.1.2 CPU, 메인 보드, 비디오 지원

지원하는 주변장치에 관한 완전한 정보는 Linux Hardware Compatibility HOWTO <<http://www.tldp.org/HOWTO/Hardware-HOWTO.html>>에 들어 있습니다. 여기서는 대략적인 기본 사항만 다룹니다.

2.1.2.1 CPU

거의 모든 x86 호환 프로세서를 지원합니다. 여기에는 AMD와 VIA (예전의 Cyrix) 프로세서도 포함됩니다.

니다. 또 Athlon XP 및 인텔 P4 Xeon같은 최근의 프로세서들도 지원합니다. 하지만 286이나 그 전에 나온 CPU에서는 리눅스가 동작하지 않습니다.

2.1.2.2 입출력 버스

시스템 버스는 마더보드에서 CPU와 다른 주변 장치들 (저장 장치 등) 사이의 통신을 담당하는 부분입니다. ISA, EISA, PCI, Microchannel Architecture (MCA, IBM의 PS/2 제품군에 사용), 아니면 VESA Local Bus를 (VLB, VL bus라고도 합니다) 사용해야 합니다.

2.1.3 그래픽 카드

콘솔 터미널에 사용할 VGA 호환 디스플레이 인터페이스를 사용해야 합니다. 최근의 거의 모든 디스플레이 카드는 VGA와 호환됩니다. 아주 옛날에 쓰였던 CGA, MDA, HGA도 동작하긴 하지만, X11은 지원하지 않습니다. 이 문서에서 설명하는 설치 과정에서는 X11이 필요없습니다.

데비안의 그래픽 인터페이스 지원은 XFree86의 X11 시스템이 지원하느냐의 여부에 따라 결정됩니다. 대부분의 AGP, PCI 및 PCIe 비디오 카드를 XFree86에서 지원합니다. 지원하는 그래픽 버스, 카드, 모니터, 포인팅 장치에 대한 정보는 <<http://www.xfree86.org/>>에 있습니다. 데비안 3.1 릴리스에는 XFree86 4.3.0 버전이 들어 있습니다.

2.1.4 노트북

노트북 컴퓨터도 지원합니다. 노트북 컴퓨터는 특별한 하드웨어 혹은 독점적인 하드웨어들이 붙어 있기도 합니다. 특정 노트북이 GNU/리눅스에서 잘 동작하는 지 여부를 확인하려면, Linux Laptop 페이지 <<http://www.linux-laptop.net/>>를 참고하십시오

2.1.5 다중 프로세서

다중 프로세서 지원은 (“symmetric multi-processing” 혹은 SMP라고도 합니다)이 아키텍처에서 지원하고, 데비안 커널 이미지에서 지원합니다. 설치 미디어에 따라서, 이 SMP 커널은 기본으로 설치할 수도 있고 아닐 수도 있습니다. SMP가 아닌 표준 커널도 SMP 시스템에서 동작하기 때문에 설치에는 문제가 없습니다. 커널은 첫 번째 CPU만을 사용하게 됩니다.

다중 프로세서의 이득을 보려면, SMP를 지원하는 커널 꾸러미를 설치해야 합니다. SMP 커널이 아니라면, 적절한 커널 꾸러미를 선택하십시오. SMP를 지원하는 자기만의 커널을 빌드할 수도 있습니다. 어떻게 하는 지에 관한 이야기는 8.5절에 있습니다. 현재 (커널 버전 2.6.12) SMP를 켜는 방법은 커널 설정의 “Processor type and features”에서 “Symmetric multi-processing support”를 켜는 것이다.

2.2 설치 미디어

이 부분에서는 여러 가지 미디어 종류 중에서 어떤 미디어로 데비안을 설치해야 할 지에 대해 다룹니다. 예를 들어, 플로피 디스크 드라이브가 있다면, 그 드라이브로 데비안을 설치할 수 있습니다. 미디어

에 관한 장이 있습니다 (4장). 거기에서 각 미디어에 따라 장단점이 쓰여 있습니다. 그 부분을 보시고 다시 이 페이지를 보셔야 할 수도 있습니다.

2.2.1 플로피

어떤 경우 플로피 디스크에서 처음 부팅해야 할 수도 있습니다. 보통 high-density (1440킬로바이트) 3.5인치 플로피 드라이브가 있어야 합니다.

2.2.2 CD-ROM/DVD-ROM

참고

이 안내서에서 “CD-ROM”이라고 할 경우에는, CD-ROM과 DVD-ROM을 모두 포함합니다. 두 가지 기술 모두 운영체제 입장에서는 같은 것이기 때문입니다. 단 예외는 표준 드라이브가 아닌, SCSI도 IDE/ATAPI도 아닌 CD-ROM 드라이브들입니다.

어떤 아키텍처에서는 CD-ROM 설치를 지원합니다. CD-ROM 부팅을 지원하는 컴퓨터에서는, 플로피 없이 완전히 설치할 수 있습니다. 시스템에서 부팅 가능 CD-ROM을 지원하지 않는 경우에도, 다른 방법으로 부팅을 할 수만 있으면, CD-ROM을 사용해서 시스템을 설치할 수 있습니다. 5장 참고.

SCSI와 IDE/ATAPI CD-ROM을 모두 지원합니다. 또, 리눅스에서 지원하는 표준이 아닌 CD 인터페이스도 (예를 들어 Mitsumi나 Matsushita 드라이브) 부트 디스크에서 지원합니다. 하지만 이런 모델을 사용하려면 부트 파라미터나 그 밖의 다른 방법이 필요할 수도 있고, 이런 장치에서 부팅할 수는 없을 겁니다. Linux CD-ROM HOWTO <<http://www.tldp.org/HOWTO/CDROM-HOWTO.html>>에는 리눅스에서 CD-ROM 사용에 관한 자세한 정보가 들어 있습니다.

USB CD-ROM 드라이브도 지원합니다. ohci1394와 sbp2 드라이버에서 지원하는 FireWire 장치도 지원합니다.

2.2.3 하드디스크

하드 디스크에서 직접 설치 시스템을 부팅하는 게 많은 아키텍처에서 할 수 있는 또 다른 방법입니다. 이렇게 하려면 설치 프로그램을 하드 디스크로 읽어들이 수 있는 다른 운영 체제가 있어야 합니다.

2.2.4 USB 메모리

많은 데비안 컴퓨터에서 플로피나 CD-ROM 드라이브는 설치나 응급복구 목적으로만 사용합니다. 서버를 여러 대 운영하는 경우, 이 드라이브들을 없애고 USB 메모리를 사용해 설치와 복구를 한다고 생각하게 될 겁니다. 이렇게 하면 필요한 드라이브를 달 공간이 부족한 작은 시스템에서도 좋습니다.

2.2.5 네트워크

네트워크에서 시스템을 부팅할 수도 있습니다.

네트워크를 사용해 랜과 NFS 마운트에서 부팅하는, 디스크 없는 설치가 또 다른 방법입니다.

운영 체제 커널을 설치한 다음에는, 나머지 시스템은 (어떤 종류이건 간에, PPP 사용 포함) 네트워크 연결을 이용해 FTP나 HTTP를 통해 설치할 수 있습니다.

2.2.6 유닉스 계열 혹은 GNU 시스템

다른 유닉스 계열 시스템이 있다면, 그 시스템을 이용해서 (이 매뉴얼 뒤에서 설명하는) **debian-installer** 없이 데비안을 설치할 수 있습니다. 이러한 방법은 지원하지 않는 하드웨어에 설치할 경우나 다운타임을 용납할 수 없는 호스트의 경우에 유용할 수 있습니다. 이러한 방식에 관심이 있다면, 바로 **C.4 절** 부분으로 넘어가십시오.

2.2.7 지원하는 저장 장치

데비안 부트 디스크에는 될 수 있으면 여러가지 시스템에서 동작할 수 있도록 빌드한 커널이 들어 있습니다. 불행히도, 이렇게 하면 커널의 크기가 커져서 그렇게 많은 드라이버를 사용할 수는 없게 됩니다 (커널을 빌드하는 방법은 **8.5절** 참고). 일반적으로는 가능한 여러가지 장치를 지원하는 게 좋습니다. 그래야 데비안을 여러가지 하드웨어에 설치할 수 있습니다.

보통 데비안 설치 시스템은 플로피, IDE 드라이브, IDE 플로피, 패러렐 포트 IDE 장치, SCSI 컨트롤러 및 SCSI 드라이브, USB, FireWire를 지원합니다. 지원하는 파일 시스템은 여러 가지 중에서도 FAT, Win-32 FAT 확장 기능 (VFAT), NTFS를 지원합니다.

MFМ, RLL, IDE, ATA라고도 부르는 “AT” 하드 디스크 인터페이스를 에뮬레이션하는 디스크 인터페이스를 지원합니다. IBM XT 컴퓨터에서 사용되었던 아주 오래된 8비트 하드 디스크 컨트롤러는 모듈로만 지원합니다. 여러가지 제조사의 SCSI 디스크 컨트롤러를 지원합니다. 자세한 건 Linux Hardware Compatibility HOWTO <<http://www.tldp.org/HOWTO/Hardware-HOWTO.html>>를 참고하십시오.

2.3 주변 장치 및 기타 하드웨어

리눅스는 마우스, 프린터, 스캐너, PCMCIA, USB 장치같은 여러가지 종류의 하드웨어 장치들을 지원합니다. 하지만 시스템을 설치할 때 이런 장치는 보통 필요 없습니다.

USB 하드웨어는 보통 잘 동작합니다. 일부 USB 키보드의 경우에만 설정이 더 필요합니다. (**3.6.4.4절** 참고)

다시 말하지만, Linux Hardware Compatibility HOWTO <<http://www.tldp.org/HOWTO/Hardware-HOWTO.html>>를 참고해 하드웨어를 리눅스에서 지원하는 지 확인하십시오.

2.4 GNU/리눅스를 위한 하드웨어 구입하기

몇몇 업체에서는 데비안 혹은 다른 GNU/리눅스 배포판을 설치한 상태로 시스템을 판매합니다. 그것 때문에 돈을 더 지불해야 할 수도 있지만, 그 때문에 여러분은 안심할 수 있습니다. 구입한 하드웨어가 GNU/리눅스에서 지원하는 하드웨어라는 게 확실하기 때문입니다.

Windows를 번들해서 판매하는 컴퓨터를 구입한 경우라면, Windows와 같이 나오는 소프트웨어 라이선스를 잘 읽어 보십시오. 그 라이선스를 거부하고 업체에서 환불을 받는 게 가능할 지도 모릅니다. 자세한 건 “windows-refund” 사이트를 참고하십시오.

리눅스가 번들된 시스템을 구입하든 그렇지 않은 간에, 아니면 중고 시스템을 구입하든 간에, 리눅스 커널에서 여러분의 하드웨어를 지원하는 지 여부를 확인하는 게 중요합니다. 위에 언급된 참고 자료에 하드웨어가 언급되어 있는지 확인하십시오. 컴퓨터 영업사원에게 (있다면) 여러분이 리눅스 시스템을 구입하려고 한다는 사실을 알려 주십시오. 리눅스와 친한 하드웨어 업체를 지원해 주십시오.

2.4.1 독점적이거나 폐쇄된 하드웨어 피하기

하드웨어 제조업체중에는 하드웨어의 드라이버를 어떻게 만들어야 하는 지 알려주지 않는 경우가 있습니다. 또 어떤 경우에는 비공개 협약을 맺지 않으면 문서조차도 볼 수 없게 해서 소스 코드를 공개할 수 없기도 합니다.

이러한 장치들에 대한 문서를 볼 수 없기 때문에, 그 장치들은 리눅스에서 동작하지 않을 수밖에 없습니다. 그러한 하드웨어의 제조업체에 문서를 공개하라고 요구하십시오. 충분히 많은 사람이 요구한다면, 제조업체들도 자유 소프트웨어 커뮤니티가 중요한 시장이라는 걸 깨닫게 될 겁니다.

2.4.2 Windows 전용 하드웨어

요즘 짜증나는 유형은, Windows 전용 모뎀과 프린터의 범람입니다. 어떤 경우 Microsoft Windows에서 동작하도록 설계되어 “WinModem”이나 “Windows 기반 컴퓨터를 위해 만들어졌습니다”라고 붙어 있습니다. 이런 하드웨어는 보통 하드웨어에 내장된 프로세서를 없애고 프로세서가 하던 작업을 Windows 드라이버로 옮겨서 컴퓨터의 메인 CPU에서 하도록 만들어 놓았습니다. 이 방식은 하드웨어 가격을 저렴하게 하지만, 그 절약한 비용이 항상 사용자에게 돌아가는 건 아닙니다. 심지어 그러한 하드웨어가 프로세서를 내장한 똑똑한 하드웨어들보다도 더 비싸기도 합니다.

Windows 전용 하드웨어는 두 가지 이유로 피해야 합니다. 첫번째는 제조업체에서 리눅스 드라이버를 만드는 데 필요한 자료를 제공하지 않기 때문입니다. 보통 하드웨어와 소프트웨어 사이의 인터페이스는 독점적이고, 문서가 있다고 해도 비공개 협약을 맺지 않으면 볼 수 없습니다. 그것 때문에 자유 소프트웨어에서 사용할 수 없게 됩니다. 자유 소프트웨어 개발자들은 프로그램의 소스 코드를 공개하기 때문입니다. 두 번째 이유로, 그렇게 내장된 프로세서를 빼버린 하드웨어를 사용하려면, 운영체제에서 그 내장 프로세서가 하던 작업을 대신 해 줘야 합니다. 그 작업은 리얼 타임 작업일 수도 있어서, 그 장치를 사용하고 있는 동안은 CPU에서 여러분의 프로그램을 실행할 수 없게 됩니다. 보통의 Windows 사용자들은 리눅스 사용자만큼 멀티 프로세스를 많이 이용하지 않기 때문에, 제조업체들은 하드웨어의 부담이 CPU에 가해진다는 사실을 사용자들이 눈치채지 않기를 희망합니다. 하지만, 어떤 멀티 프로세싱 운영 체제라고 해도, Windows 2000이나 XP라고 해도 주변장치 제조업체가 하드웨어의 프로세싱 파워를 옮겨 놓는 바람에, 효율이 떨어질 수밖에 없습니다.

이 제조업체들에게 하드웨어에 관한 프로그래밍에 필요한 문서와 그 밖의 자료들을 공개하라고 요구하면 도움이 됩니다. 하지만 일단 가장 좋은 방법은 Linux Hardware Compatibility HOWTO <<http://www.tldp.org/HOWTO/Hardware-HOWTO.html>>에서 동작한다고 나올 때까지는 이런 종류의 하드웨어 사용을 피하는 것입니다.

2.4.3 가짜 혹은 “가상” 패리티 RAM

컴퓨터 상점에서 패리티 RAM을 사려고 하면, 아마도 진짜 패리티가 아니라 가상 패리티 메모리 모듈을 사게 될 겁니다. 가상 패리티 SIMM은 (항상 그런 건 아니지만) 일반 패리티 없는 SIMM보다 칩이 1개 더 많고 그 추가 칩이 다른 칩보다 작기 때문에 진짜 패리티 RAM과 구분할 수 있습니다. 가상 패리티 SIMM은 패리티가 없는 메모리와 완전히 똑같이 동작합니다. RAM에서 1비트가 애러가 발생하는 경우에 알 수가 없습니다. 패리티가 없는 SIMM 대신에 가상 패리티 SIMM을 구입하려고 돈을 낭비하지 마십시오. 진짜 패리티 SIMM은 패리티 없는 SIMM보다 조금 더 비싸지는 않습니다. 각 8비트 메모리마다 1개 비트를 더 구입하게 되는 꼴이기 때문입니다.

Intel x86 RAM에 관한 정보를 알고 싶으시거나, 어떤 RAM을 구입하는 게 가장 좋을 지에 대해 알고 싶으시면 PC Hardware FAQ <<http://www.faqs.org/faqs/pc-hardware-faq/part1/>>를 참고하십시오.

2.5 메모리 및 디스크 공간 요구사항

최소한 32MB의 메모리와 110MB의 하드디스크 공간은 있어야 합니다. 최소한의 콘솔 기반 시스템이라면 (모든 표준 꾸러미를 포함해서) 250MB가 필요합니다. X 윈도우 시스템이나 개발 관련 프로그램과 라이브러리까지 쓸만한 정도로 소프트웨어를 설치하려면 400MB가 필요합니다. 완전한 데스크탑이라면 수 기가바이트가 있어야 할 것입니다.

2.6 네트워크 연결 하드웨어

대부분의 PCI와 많은 (오래된) ISA 네트워크 카드를 지원합니다. 일부 네트워크 인터페이스 카드는 데비안 설치 디스크에서 지원하지 않습니다. 예를 들어 AX.25 카드와 AX.25 프로토콜, NI16510 EtherBlaster 카드, Schneider & Koch G16 카드, Zenith Z-Note 내장 네트워크 카드는 지원하지 않습니다. Microchannel (MCA) 네트워크 카드는 표준 설치 시스템에서는 지원하지 않지만, (좀 오래된) 사용 방법을 보시려면 Linux on MCA <<http://www.dgmicro.com/mca/general-goods.html>>를 참고하십시오. FDDI 네트워크도 설치 디스크에서는 카드와 프로토콜을 지원하지 않습니다.

ISDN의 경우 (오래된) 독일 1TR6용 D-channel 프로토콜은 지원하지 않습니다. Spellcaster BRI ISDN 보드도 `debian-installer`에서 지원하지 않습니다.

Chapter 3

Debian GNU/Linux 설치하기 전에

이 장에서는 설치 프로그램을 부팅하기도 전에 해야 하는 데비안 설치 준비에 대해 다룹니다. 데이터 백업, 하드웨어에 대한 정보 모으기, 그 밖에 필요한 정보 찾기와 같은 준비를 해야 합니다.

3.1 설치 과정의 개요

먼저, 다시 설치하는 것에 대해서 설명합니다. 데비안에서는, 처음부터 시스템을 다시 설치해야 하는 상황은 아주 드뭅니다. 다시 설치해야 하는 경우는 아마도 하드 디스크가 물리적으로 망가진 경우가 대부분일 것입니다.

널리 쓰이는 많은 운영체제들에서는 치명적인 문제점이 발생하거나 새로운 운영체제의 버전으로 업그레이드하려는 경우 완전한 설치 과정을 거쳐야 합니다. 처음부터 완전히 새로 설치할 필요가 없다고 해도, 새로운 운영 체제에서 제대로 동작하려면 프로그램들을 다시 설치해야 합니다.

Debian GNU/Linux에서는 운영체제에서 무언가 문제점이 있으면, 운영체제를 바꿔 버리지 않고 바로 잡습니다. 업그레이드할 때 전부 다 설치할 필요는 없고, 항상 그 자리에서 업그레이드할 수 있습니다. 그리고 프로그램은 거의 언제나 뒤의 운영체제 릴리스에서도 호환됩니다. 어떤 프로그램의 새 버전에서 다른 소프트웨어의 새로운 버전이 필요한 경우라면, 데비안 꾸러미 시스템에서 필요한 소프트웨어를 모두 알아서 자동으로 설치합니다. 즉, 다시 설치할 필요가 없도록 많은 노력이 들어가 있기 때문에, 다시 설치하는 마지막 선택으로 남겨 두십시오. 설치 프로그램은 기존 시스템 위에 다시 설치하도록 설계되지 않았습니니다.

다음은 설치 과정에서 해야 할 단계입니다.

1. 설치하려는 하드 디스크에 들어 있는 데이터나 문서를 백업하십시오.
2. 설치를 시작하기 전에, 해당 컴퓨터에 대한 정보와 필요한 문서를 모으십시오.
3. 하드 디스크에 데비안에 쓸 파티션 가능한 공간을 만드십시오.
4. 해당 컴퓨터에 필요한 설치 프로그램 소프트웨어 및 필요한 드라이버 파일을 찾아서 내려받으십시오. (데비안 CD 사용자 제외)

5. 부팅 테이프/플로피/USB 스틱을 만드십시오. 혹은 부팅 파일을 저장해 놓으십시오. (데비안 CD 사용자는 대부분 데비안 CD 중 한장으로 부팅합니다)
6. 설치 시스템을 부팅하십시오.
7. 설치 언어를 선택하십시오.
8. 이더넷 네트워크 연결이 있으면 활성화하십시오.
9. 데비안을 설치할 파티션을 만들고 마운트하십시오.
10. 베이스 시스템의 자동 내려받기/설치/설정을 보십시오.
11. 데비안 그리고/혹은 기존 시스템을 시작할 수 있는 부트 로더를 설치하십시오.
12. 새로 설치한 시스템을 맨 처음 시작하고, 몇 가지 초기 시스템 설정을 하십시오.
13. 필요하다고 생각하는 추가 소프트웨어를 (태스크 및/혹은 꾸러미) 설치하십시오.

설치할 때 문제가 발생할 때를 대비해서, 각 단계가 어떤 꾸러미와 상관이 있는지 알아 두는 게 좋습니다. 이 설치 단계의 주요 소프트웨어들을 소개하면:

설치 소프트웨어인 **debian-installer**는 이 매뉴얼에서 가장 중점적으로 다룹니다. **debian-installer**는 하드웨어를 찾아서 적당한 드라이버를 읽어들이고, **dhcp-client**를 이용해 네트워크 연결을 설정하고, **debootstrap**을 실행해 베이스 시스템 꾸러미를 설치합니다. 이 외에 더 많은 소프트웨어들이 각 단계에서 작은 역할들을 담당하고 있지만, 새 시스템이 처음 시작하기까지 전까지는 **debian-installer**가 작업을 마칩니다.

베이스 시스템을 새로 시작하면, **base-config**를 통해 사용자를 추가하고, 시간대를 설정하고 (**tzsetup**을 통해), 꾸러미 설치 시스템을 설정합니다 (**apt-setup** 이용). 다음에 **tasksel**을 실행해 여러 관련된 프로그램의 모음을 선택하고, **aptitude**를 실행해 각각의 소프트웨어 꾸러미를 선택합니다.

맨 처음 시스템이 시작하기 전에 **debian-installer**가 끝나면, 아주 기본적인 명령행 기반 시스템만 사용할 수 있습니다. 모니터에 창이 여러 개 나오는 그래픽 인터페이스는 설치하지 않습니다. 마지막 단계에서 **tasksel**이나 **aptitude**를 이용해 따로 설치해야 합니다. 이건 많은 Debian GNU/Linux 시스템과 서버에서 그래픽 사용자 인터페이스가 전혀 필요없기 때문입니다.

X 시스템은 **debian-installer**와는 완전히 다르고, 실제로는 훨씬 더 복잡합니다. X 윈도우의 설치나 X 윈도우의 문제점 해결은 이 매뉴얼이 다루는 범위를 벗어납니다.

3.2 기존 데이터를 백업하십시오!

시작하기 전에, 지금 시스템에 있는 모든 파일을 백업해 두십시오. 컴퓨터에 처음으로 네이티브가 아닌 운영 체제를 설치하는 거라면, Debian GNU/Linux의 루트로 쓸 디스크를 다시 파티션해야 합니다. 디스크를 파티션하면 파티션 프로그램으로 어떤 프로그램을 쓰든 간에 그 디스크에 있는 모든 파일을 잃게 됩니다. 설치에 사용하는 프로그램은 상당히 안정적이고 수년동안 사용해 온 프로그램이지만, 그만큼 강력하기도 해서 조금만 잘못하면 막심한 손해가 발생합니다. 백업을 한 뒤에도 대답을 할 때나 어떤 작업을 할 때 신중을 기하십시오. 2분만 더 생각하면 수 시간의 불필요한 작업을 방지할 수 있습니다.

멀티 부팅 시스템을 만든다면, 현재 운영 체제의 배포 미디어를 가지고 있으십시오. 특히 부팅 드라이브를 다시 파티션하는 경우라면, 운영 체제의 부트 로더를 다시 설치해야 하거나, 아니면 더 많은 경우에 운영 체제 전체를 해당 파티션에 다시 설치해야 합니다.

3.3 필요한 정보

3.3.1 문서

3.3.1.1 설치 매뉴얼

지금 읽고 있는 일반 텍스트, HTML, 혹은 PDF 형식으로 된 문서

- <[install.ko.txt](#)>
- <[install.ko.html](#)>
- <[install.ko.pdf](#)>

3.3.1.2 하드웨어 문서

하드웨어를 설정하고 이용하는 법에 관한 유용한 정보가 들어있습니다.

- Linux Hardware Compatibility HOWTO <<http://www.tldp.org/HOWTO/Hardware-HOWTO.html>>

3.3.2 하드웨어 정보가 있는 곳 찾기

많은 경우에 설치 프로그램에서 자동으로 하드웨어를 찾아 냅니다. 하지만 철저히 준비하려면, 설치하기 전에 하드웨어에 대해 잘 알아 두는 게 좋습니다.

하드웨어 정보는 다음에서 얻을 수 있습니다:

- 하드웨어에 같이 들어 있는 매뉴얼.
- 컴퓨터의 BIOS 설정 화면. 컴퓨터가 시작할 때 어떤 키 조합을 누르면 BIOS 설정 화면을 볼 수 있습니다. 보통 **Delete** 키입니다.
- 하드웨어의 케이스 및 포장.
- Windows 제어판의 시스템 창.
- 파일 관리자에서 표시하는 것 등과 같은, 다른 운영체제의 시스템 명령어 및 시스템 도구들. 특히 RAM과 하드 드라이브에 대한 정보를 알아내는 데 유용합니다.
- 시스템 관리자 혹은 인터넷 서비스 회사. 여기서는 네트워크 및 전자메일 설정에 관련된 사항을 알 수 있습니다.

Table 3.1: 설치하는 데 필요한 하드웨어 정보

하드웨어	필요할 수도 있는 정보
하드 드라이브	용량이 얼마나 되는 지.
	시스템에 붙어 있는 순서.
	IDE인지 SCSI인지 (대부분의 컴퓨터는 IDE입니다).
	사용 가능한 빈 공간.
	파티션.
	다른 운영 체제가 설치되어 있는 파티션.
모니터	모델 및 제조사.
	지원하는 해상도.
	가로 출력 주파수.
	세로 출력 주파수.
	지원하는 색상수.
	화면 크기.
마우스	종류: 시리얼, PS/2, 혹은 USB.
	포트.
	제조사.
	단추 개수.
네트워크	모델 및 제조사.
	어댑터 종류.
프린터	모델 및 제조사.
	지원하는 인쇄 해상도.
비디오 카드	모델 및 제조사.
	비디오 RAM 크기.
	지원하는 해상도 및 색상수 (모니터에서도 지원하는 지 확인해야 합니다).

3.3.3 하드웨어 호환성

많은 유명 제품들에서 아무 문제없이 리눅스가 동작합니다. 또 리눅스용 하드웨어가 날이 갈 수록 발전하고 있습니다. 하지만 아직도 리눅스는 다른 운영체제만큼 다양한 종류의 하드웨어에서 동작하지는 못합니다.

특히, Windows가 있어야 동작하는 하드웨어는 리눅스에서 이용하지 못합니다.

일부 Windows 전용 하드웨어는 리눅스에도 쓸 수 있긴 하지만, 보통 추가적인 노력이 필요합니다. 게다가 보통 Windows 전용 하드웨어의 리눅스 드라이버는 한 개의 리눅스 커널 버전에서만 쓸 수 있습니다. 그래서 그러한 드라이버는 금방 사용할 수 없게 됩니다.

이른바 win-modem은 그러한 가장 전형적인 하드웨어입니다. 프린터와 그 외의 장치도 윈도우 전용이 있습니다.

다음 방법으로 하드웨어 호환성을 알아볼 수 있습니다:

- 제조사의 웹사이트에서 새 드라이버 확인하기.
- 웹사이트와 매뉴얼에서 에뮬레이션에 대한 정보 찾아보기. 덜 알려진 상표의 제품이 더 많이 알려진 제품의 드라이버와 설정들을 그대로 사용하기도 합니다.
- 해당 아키텍처에 관한 웹사이트에서 리눅스 하드웨어 호환성 목록 확인.
- 인터넷에서 다른 사용자의 사용담 검색해 보기.

3.3.4 네트워크 설정

컴퓨터가 하루 24시간 내내 네트워크에 연결되어 있다면 (인터넷이나 그같은 연결의 경우 — PPP 아님), 네트워크 시스템 관리자에게 이러한 정보를 알아봐야 합니다.

- 호스트 이름 (직접 결정할 수도 있습니다).
- 도메인 이름.
- 컴퓨터의 IP 주소.
- 네트워크에 사용할 넷마스크.
- 라우팅하는 데 쓸 기본 게이트웨이 시스템의 IP 주소 (게이트웨이가 있는 경우).
- 네트워크에서 DNS (Domain Name Service) 서버로 사용할 시스템.

한편 관리자가 DHCP 서버를 사용할 수 있다고 하고 DHCP 서버 사용을 추천한다고 하면, 이 정보가 필요 없습니다. 설치할 때 DHCP 서버에서 컴퓨터에 이 정보를 알려줍니다.

무선 네트워크를 사용한다면, 다음도 알아봐야 합니다:

- 무선 네트워크의 ESSID.
- WEP 보안 키 (있다면).

3.4 최소 하드웨어 요구사항 맞추기

컴퓨터의 하드웨어에 대한 정보를 모았으면, 하드웨어에서 설치하려는 방법대로 설치할 수 있는 지 확인하십시오.

필요에 따라서는, 아래의 표에 나온 하드웨어보다 못한 하드웨어를 가지고 있을 수도 있습니다. 하지만 아래 제안을 무시할 경우 대부분의 사용자들은 짜증나는 문제를 겪게 됩니다.

데스크탑은 최소 Pentium 100, 서버는 최소 Pentium II-300을 추천합니다.

Table 3.2: 추천하는 최소 시스템 요구사항

설치 종류	RAM	하드 드라이브
데스크탑 없음	24메가바이트	450메가바이트
데스크탑 포함	64메가바이트	1기가바이트
서버	128메가바이트	4기가바이트

많이 쓰이는 데비안 시스템의 예제입니다. C.3절에 보면 관련된 프로그램들이 차지하는 디스크 공간이 얼마나 되는 지 대략 알 수 있습니다.

표준 서버 작은 서버 프로파일입니다. 셸 사용자들이 쓰는 여러가지 프로그램들이 없는, 용량을 줄인 서버에 씁니다. FTP 서버, 웹 서버, DNS, NIS, POP 서버가 들어 있습니다. 100MB의 디스크 공간이면 충분하고, 서비스하려는 데이터에 필요한 공간이 더 필요합니다.

데스크탑 표준 데스크탑 컴퓨터로, X 윈도우 시스템, 완전한 데스크탑 환경, 사운드, 문서 편집기 등이 들어 있습니다. 표준 데스크탑 태스크를 설치하면 약 2GB가 필요하지만, 그보다 적은 용량으로도 가능합니다.

작업용 콘솔 크기를 더 줄인 사용자 컴퓨터로, X 윈도우 시스템이나 X 프로그램이 없습니다. 노트북이나 휴대용 컴퓨터에도 적합합니다. 크기는 약 140MB입니다.

개발자 Perl, C, C++ 등과 같은 모든 개발용 꾸러미들이 들어 있는 데스크탑. 크기는 약 475MB입니다. X11과 그 외에 추가 꾸러미를 더 설치한다고 하면, 이런 컴퓨터에는 약 800MB를 확보해 놓아야 합니다.

이 점을 기억해 두십시오. 위의 용량은 사용자의 파일, 메일, 데이터 따위의 기타 자료들을 포함한 게 아닙니다. 자기만의 파일과 데이터에 필요한 공간을 생각할 때는 항상 넉넉하게 하는 게 좋습니다. 특히, `/var` 파티션에는 여러 가지 데비안의 상태 정보와 로그 파일같은 일반적인 데이터들이 들어 있습니다. `dpkg` 파일들은 (설치 꾸러미에 대한 정보를 합쳐서) 쉽게 20MB를 차지합니다. 또, `apt-get` 은 설치하기 전에 내려 받은 꾸러미를 여기에 저장합니다. 보통 `/var`에 최소 100MB를 확보해 놓아야 합니다.

3.5 멀티 부팅 시스템에서 미리 파티션하기

디스크를 파티션은 디스크를 여러 개의 조각으로 나누는 작업을 말합니다. 각 조각들은 다른 조각과 독립적입니다. 비유 하자면 집에 벽을 놓는 것과 비슷합니다. 어떤 방에 가구를 놓는다고 해서 다른 방에 영향을 끼치지 않습니다.

시스템에 이미 다른 운영 체제가 들어 있고 (Windows 9x, Windows NT/2000/XP, OS/2, MacOS, Solaris, FreeBSD, ...) 같은 디스크에 리눅스도 설치하려는 경우에는 디스크를 다시 파티션해야 합니다.

데비안은 데비안만의 디스크 파티션이 따로 있어야 합니다. 데비안은 Windwos나 MacOS 파티션에 설치할 수 없습니다. 다른 리눅스 시스템과는 일부 파티션을 공유할 수도 있지만, 여기서는 그 방법에 대해 다루지 않습니다. 최소한 데비안 루트에 사용할 전용 파티션이 하나 있어야 합니다.

현재 파티션 상태는fdisk나 PartitionMagic같은 현재 운영 체제의 파티션 도구를 이용해 알아볼 수 있습니다. 모든 파티션 도구는 파티션을 바꾸지 않고도 현재 파티션을 보기만 할 수 있습니다.

이미 파일 시스템이 들어 있는 파티션을 바꾸면 보통 거기에 들어 있는 정보가 모두 망가집니다. 그러므로 파티션 작업을 하기 하기 전에 항상 백업을 만들어야 합니다. 집의 경우에 비유하자면, 벽을 옮기기 전에 모든 가구를 다른 곳에 옮겨 놓아야 가구가 망가지지 않을 겁니다.

컴퓨터에 하드 디스크가 여러 개 있으면, 어떤 한 하드 디스크 전체에 데비안을 설치할 경우가 있습니다. 그러한 경우에는 설치 시스템을 부팅하기 전에 디스크를 파티션할 필요가 없습니다. 설치 프로그램에 포함된 파티션 프로그램이 파티션 작업을 훌륭하게 처리합니다.

컴퓨터에 하드 디스크가 한 개 있고, 지금 운영 체제를 Debian GNU/Linux으로 완전히 덮어 쓰려고 한다면, 설치 시스템을 부팅한 다음에 설치하는 과정에서 파티션을 할 수도 있습니다 (6.3.2.1절). 하지만 설치 시스템을 테이프나 CD-ROM이나 연결되어 있는 다른 기계에서 파일로 부팅할 경우에만 이렇게 할 수 있습니다. 생각해 보면 하드 디스크에 부팅 파일을 넣고 같은 그 하드 디스크를 파티션해 버린다면, 부팅 파일들이 지워질 수밖에 없고, 그것보다는 첫 단계에서 설치를 성공적으로 하는 편이 더 낫습니다. 최소한 이러한 경우에는, 원래 운영체제의 설치 테이프나 CD같은 경로 기계를 복구할 방법을 가지고 있는 게 좋습니다.

기계에 파티션이 여러 개 있다면, 그리고 파티션을 지워서 충분한 공간이 나온다면 기다렸다가 데비안 설치 프로그램의 파티션 프로그램을 사용할 수 있습니다. 이 경우에도 아래의 내용을 읽어야 특별한 상황에 대처할 수 있습니다. 예를 들어 파티션에서 기존 파티션의 순서때문에 설치 전에 파티션해야 할 수도 있습니다.

기계에 DOS나 Windows에서 사용하는 FAT나 NTFS 파일 시스템이 있으면, 데비안 설치 프로그램의 파티션 프로그램을 이용해서 파일시스템 크기를 조정할 수 있습니다.

위의 경우 중 아무것도 아니라면, 데비안에 파티션 가능한 공간을 만들기 위해 설치 전에 파티션을 해야 합니다. 다른 운영체제가 들어 있는 파티션이 들어 있으면, 원래 운영 체제 프로그램의 파티션 프로그램을 이용해 파티션해야 합니다. 다른 운영 체제의 도구로 Debian GNU/Linux의 파티션을 만들지 않기를 권장합니다. 대신에, 계속 유지하려는 원래 운영 체제의 파티션만 만들어야 합니다.

같은 기계에 여러 개의 운영 체제를 설치하려는 경우, 리눅스를 설치하기 전에 다른 운영체제를 먼저 설치해야 합니다. Windows 및 다른 운영 체제를 설치하면 리눅스를 시작하지 못하게 될 수도 있고, 설치 과정에서 네이티브가 아닌 파티션을 다시 포맷하라고 할 수도 있습니다.

이런 동작들은 복구할 수도 있고, 피할 수 있습니다. 하지만 네이티브 시스템을 먼저 설치하면 이런 문제가 없습니다.

파티션이 한 개인 하드 디스크가 한 개 있고 (일반적인 데스크탑 컴퓨터), 원래 운영 체제와 데비안을 멀티 부팅하는 경우, 다음과 같이 해야 합니다:

1. 컴퓨터에 들어 있는 것을 전부 백업하십시오.
2. CD-ROM이나 테이프같은 원래 운영 체제의 설치 미디어에서 부팅하십시오.

3. 네이티브 파티션 도구로 네이티브 시스템 파티션을 만드십시오. Debian GNU/Linux으로 사용할 위치에 파티션을 만들거나 빈 공간을 만들어 놓으십시오.
4. 새 파티션에 원래 운영 체제를 설치하십시오.
5. 네이티브 시스템으로 돌아가서 제대로 되었는지 확인하고, 데비안 설치 프로그램의 부팅 파일들을 내려 받으십시오.
6. 데비안 설치 프로그램으로 부팅해 데비안 설치를 계속하십시오.

3.5.1 DOS나 Windows에서 파티션하기

기존 FAT나 NTFS 파티션을 변경하는 경우에는, 아래의 방법을 사용하거나 Windows나 DOS에 들어 있는 도구를 이용하십시오. 그 밖의 경우에는 DOS나 Windows에서 파티션할 필요는 없습니다. 보통 리눅스 파티션 도구가 더 좋습니다.

큰 IDE 디스크가 있고, LBA 주소나 오버레이 드라이버나 (하드 디스크 제조사에서 제공하기도 함) 큰 디스크 접근 확장 기능을 지원하는 새로운 (1998년 후) BIOS를 사용하지 않는 경우에는 데비안 부팅 파티션 위치를 잘 잡아야 합니다. 이런 경우 부팅 파티션을 하드 드라이브의 첫 번째 1024실린더 안에 놓아야 합니다 (보통 BIOS에서 변환하지 않은 용량으로 약 524 메가바이트입니다). 이렇게 하려면 기존 FAT나 NTFS 파티션을 옮겨야 할 수도 있습니다.

3.5.1.1 DOS나 Win-32나 OS/2에서 시작할 때 데이터 손실없이 다시 파티션하기

설치할 때 가장 많은 경우는 이미 DOS (Windows 3.1 포함), Win32 (Windows 95, 98, Me, NT, 2000, XP), 혹은 OS/2와 같은 시스템이 들어 있고, 이전의 시스템을 망가뜨리지 않고 같은 디스크에 데비안을 설치하려는 경우입니다. 설치 프로그램에는 DOS나 Windows에서 사용하는 FAT와 NTFS 파일 시스템의 크기를 조정하는 기능이 들어 있습니다. 설치 프로그램에서 **수동으로 파티션 테이블 편집** 옵션을 선택하고 크기를 조정하려는 파티션을 선택하고 새로운 크기를 지정하십시오. 이렇게 하면 대부분 아래에 설명한 방법을 쓰지 않아도 됩니다.

계속 진행하기 전에, 디스크를 어떻게 나눌 지 결정해야 합니다. 여기에 쓰여 있는 방법은 파티션을 두 개로 나누는 것 뿐입니다. 한 파티션에는 원래 OS가 들어 있고, 다른 파티션에 데비안이 들어 있습니다. 데비안을 설치할 때 데비안이 사용하는 디스크 부분을 어떻게 사용할 지 (스왑이나 파일 시스템) 결정하는 부분이 있습니다.

하려는 일은 파티션 정보를 바꾸기 전에 그 안의 모든 데이터를 앞쪽으로 옮기는 것입니다. 그래야 아무것도 잃지 않습니다. 데이터 이동과 파티션을 가능하면 조금 하는 게 중요합니다. 그래야 파일이 파티션 끝 부분에 들어 있지 않게 됩니다. 끝 부분에 파일이 들어가면 그 파티션에서 뽑아낼 수 있는 공간이 줄어듭니다.

첫 번째로 필요한 건 데비안 미러 사이트에서 **tools/** 디렉토리에 있는 **fips** 프로그램입니다. 압축 파일을 풀고 **RESTORRB.EXE**, **FIPS.EXE**, **ERRORS.TXT** 파일을 부팅 가능 플로피에 넣으십시오. 부팅 가능 플로피는 DOS에서 **sys a:** 명령으로 만듭니다. **fips**에는 아주 좋은 문서가 들어 있으니 읽어 보십시오. 디스크 압축 드라이버나 디스크 관리자를 사용하고 있는 경우라면 반드시 문서를 읽어 봐야 합니다. 디스크 단편화를 없애기 전에 디스크를 만들고 문서를 읽어 보십시오.

그 다음에 필요한 건 모든 데이터를 파티션의 처음으로 옮기는 일입니다. DOS 6.0 이후에 포함되어 있는 **defrag**가 이 작업을 합니다. 이런 일을 하는 다른 소프트웨어 목록을 보려면 **fips** 문서를 보십시오. Windows 9x를 사용한다면 Windows 9x에서 **defrag**를 실행해야 합니다. DOS는 Windows 95 이상에서 긴 파일이름을 지원하기 위해 사용하는 VFAT를 인식하지 못합니다.

defragmenter를 실행한 다음에 (디스크가 크면 꽤 오래 걸립니다), 플로피 드라이브에 새로 만든 **fips** 디스크를 넣고 다시 부팅하십시오. **a:**“**fips**를 실행해 안내에 따르십시오.

fips에서 파티션 크기 조정을 하지 못하는 경우에도, 다른 파티션 관리자들이 많이 있습니다.

3.5.1.2 DOS용 파티션하기

리눅스 도구를 이용해 DOS 드라이브를 파티션하는 경우나, DOS 파티션의 크기를 바꾸는 경우, 만들어 낸 FAT 파티션에서 많은 문제가 발생했습니다. 예를 들면, 어떤 사람은 속도가 느려지기도 했고, **scandisk**를 사용했을 때 계속 문제가 발생하기도 했고, 그 외에 DOS나 Windows에서 괴상한 오류들이 발생했습니다.

DOS에 사용할 파티션을 만들거나 파티션 크기를 변경한 경우, 처음 섹터 몇 개를 0으로 채워 넣는 게 좋습니다. DOS에서 **format** 명령어를 실행하기 전에 하십시오. 리눅스에서:

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

3.6 설치하기 전에 하드웨어 및 운영 체제 설정

여기서는 데비안을 설치하기 전에 해야 할 하드웨어 설정에 대해 (설정이 따로 필요하다면) 다룹니다. 보통 시스템의 펌웨어 설정을 확인하고 필요하다면 설정을 바꾸는 일들입니다. “펌웨어”는 하드웨어가 사용하는 주요 소프트웨어로, 전원을 켜 다음에 부트스트랩 과정에서 반드시 동작합니다. 여러분 시스템에서의 Debian GNU/Linux 안정성과 관련된 하드웨어 문제들도 다룹니다.

3.6.1 BIOS 설정 메뉴 들어가기

BIOS는 기계를 부팅시키고 운영 체제에서 하드웨어를 접근하도록 하는 기본적인 기능들이 들어 있습니다. 시스템에서 BIOS 설정 메뉴가 있으면, 그 메뉴를 이용해 BIOS를 설정할 수 있습니다. 설치하기 전에, BIOS를 제대로 설정했는지 확인해야 합니다. 그렇지 않으면 갑자기 멈출 수도 있고 데비안을 설치하지 못할 수도 있습니다.

이 장의 나머지 부분은 <<http://www.faqs.org/faqs/pc-hardware-faq/part1/>>에서 가져온 “어떻게 CMOS 설정 메뉴로 들어가는가?”라는 질문에 대한 답입니다. BIOS (아니면 “CMOS”) 설정 메뉴에 들어가는 방법은 BIOS 소프트웨어를 만든 회사에 따라 다릅니다.

AMI BIOS POST (power on self test, 최초 테스트) 도중에 **Delete** 키

Award BIOS POST 도중에 **Ctrl-Alt-Esc**, 아니면 **Delete** 키

DTK BIOS POST 도중에 **Esc** 키

IBM PS/2 BIOS **Ctrl-Alt-Delete** 다음에 **Ctrl-Alt-Insert**

Phoenix BIOS **Ctrl-Alt-Esc** 아니면 **Ctrl-Alt-S** 아니면 **F1**

다른 BIOS를 실행하는 방법에 대한 정보는 <<http://www.tldp.org/HOWTO/Hard-Disk-Upgrade/install.html>>에 있습니다.

일부 Intel x86 기계에는 BIOS에 CMOS 설정 메뉴가 없습니다. 소프트웨어 CMOS 설정 프로그램이 필요합니다. 해당 기계의 설치 및/혹은 진단 디스켓이 없다면, 세어웨어/프리웨어 프로그램을 사용해 볼 수 있습니다. <<ftp://ftp.simtel.net/pub/simtelnet/msdos/>> 사이트를 살펴 보십시오.

3.6.2 부팅 장치 선택

많은 BIOS의 설정 메뉴에서는 시스템을 부팅하는 데 쓸 장치를 선택할 수 있습니다. 여기에서 **A:**에서 운영 체제를 부팅하도록 설정하고, 그 다음에 첫 번째 CD-ROM 장치로 (아마도 **D:**나 **E:**) 하고, 그 다음에 **C:**로 (첫 번째 하드 디스크) 설정하십시오. 이렇게 설정하면 플로피 디스크나 CD-ROM에서 부팅합니다. 대부분 데비안을 부팅할 때 이 둘 중의 하나에서 부팅합니다.

최신의 SCSI 컨트롤러가 있고 SCSI에 CD-ROM 장치가 붙어 있다면 보통 CD-ROM에서 부팅할 수 있을 것입니다. 해야 되는 일은 컨트롤러의 SCSI BIOS에서 CD-ROM 부팅을 켜는 일 뿐입니다.

그 밖의 많이 쓰는 옵션으로, USB 저장 장치에서 (USB 메모리나 USB 키라고도 함) 부팅하는 옵션입니다. 어떤 BIOS에서는 USB 저장 장치에서 직접 부팅할 수 있고, 그렇지 않은 BIOS도 있습니다. BIOS를 설정해서 “Removable drive” 혹은 “USB-ZIP”에서 부팅하도록 하면 USB 장치에서 부팅합니다.

여기서는 부팅 순서를 설정하는 법을 자세히 설명합니다. 리눅스를 설치한 다음에 부팅 순서를 원래 대로 돌려 놓는 걸 잊지 마십시오. 그래야 하드 디스크에서 부팅할 수 있습니다.

3.6.2.1 IDE 컴퓨터의 부팅 순서 바꾸기

1. 컴퓨터가 시작할 때, BIOS 유틸리티에 들어가는 키를 누르십시오. 보통 그 키는 **Delete** 키입니다. 하지만 정확히 어떤 키를 눌러야 할 지는 하드웨어 문서를 참고하십시오.
2. 설정 유틸리티에서 부팅 순서를 찾으십시오. 이 설정의 위치는 BIOS에 따라 다르지만, 드라이브 여러 개를 늘어 놓은 필드를 찾으시면 됩니다.

IDE 기계에서는 보통 C, A, cdrom이거나 A, C, cdrom입니다.

C는 하드 드라이브이고, A는 플로피 드라이브입니다.

- 부팅 순서 설정을 바꿔서 CD-ROM이나 플로피가 첫 번째가 되도록 하십시오. 보통 **Page Up**이나 **Page Down** 키로 여러가지 중의 하나를 선택합니다.
- 바꾼 내용을 저장하십시오. 설정 화면의 안내에 따라 저장하십시오.

3.6.2.2 SCSI 컴퓨터에서 부팅 순서 바꾸기

- 컴퓨터가 시작하면, SCSI 설정 유틸리티로 들어가는 키를 누르십시오.

컴퓨터를 시작하면 메모리 검사를 하고, BIOS 유틸리티를 시작하는 방법을 표시한 다음, SCSI 설정 유틸리티로 들어갈 수 있습니다.

어떤 키를 눌러야 하는지는 유틸리티에 따라 다릅니다. 보통 **Ctrl-F2**입니다. 정확한 키 조합은 하드웨어 문서를 보십시오.

- 부팅 순서를 바꾸는 유틸리티 찾기.
- CD 드라이브의 SCSI ID가 목록의 첫 번째가 되도록 유틸리티를 설정하십시오.
- 바뀐 부분을 저장하십시오. 화면의 안내에 따라 바뀐 부분을 컴퓨터에 저장하십시오. 보통 **F10**을 눌러야 합니다.

3.6.3 기타 BIOS 설정

3.6.3.1 CD-ROM 설정

어떤 BIOS 시스템은 (예: Award BIOS) CD 속도를 설정할 수 있습니다. 속도를 설정하지 말고, 그냥 최저 속도로 놔둬야 합니다. seek failed 오류 메시지가 나오게 된다면 이 문제 때문일 수도 있습니다.

3.6.3.2 Extended와 Expanded Memory

시스템에서 extended와 expanded memory가 모두 있다면, extended memory를 최대한으로 해 놓고 expanded memory를 최소한으로 맞춰 놓으십시오. 리눅스에는 extended memory가 필요하지만 expanded memory를 사용할 수 없습니다.

3.6.3.3 Virus Protection

BIOS에서 제공하는 바이러스 경고 기능이 있다면 끄십시오. 바이러스 방지 보드나 기타 특수 하드웨어가 달려 있다면, GNU/리눅스를 돌리는 동안은 그 기능을 끄거나 물리적으로 제거하십시오. 이러한 기능이나 하드웨어는 GNU/리눅스와 호환되지 않습니다. 또 파일 시스템의 권한 및 리눅스의 보호 메모리때문에 리눅스에서 바이러스는 거의 없습니다.¹

¹설치가 끝난 뒤에 원하신다면 부트섹터 보호를 켤 수 있습니다. 이는 리눅스에 추가적인 보호를 제공하지 않습니다. 그러나 윈도를 같이 사용한다면 재난을 방지할 지 모릅니다. 부트 관리자가 설정된 후에는 마스터 부트 레코드(MBR)에 영향을 주지 않습니다.

3.6.3.4 Shadow RAM

마더보드에서 *shadow RAM*이나 BIOS 캐싱을 제공할 수도 있습니다. “Video BIOS Shadow”나 “C800-CBFF Shadow”와 같은 설정을 볼 수 있을 것입니다. 모든 shadow RAM을 끄십시오. shadow RAM은 마더보드와 일부 컨트롤러 카드에 있는 ROM의 접근 속도를 높이는 데 쓰입니다. 리눅스는 일단 부팅하면 이러한 ROM을 사용하지 않습니다. ROM에 들어 있는 16비트 프로그램 대신에 직접 32비트 소프트웨어가 들어 있기 때문입니다. 어떤 경우에는 shadow RAM을 꺼야 shadow RAM을 다른 프로그램에서 일반 메모리로 사용할 수 있습니다. shadow RAM을 켜 놓으면 리눅스가 하드웨어에 접근할 때 문제가 발생할 수 있습니다.

3.6.3.5 Memory Hole

BIOS에 “15–16 MB Memory Hole”과 같은 게 있다면, 그 기능을 끄십시오. 그 만큼의 메모리가 있으면 리눅스에서는 이 구간의 메모리도 이용합니다.

Intel이 만든 어떤 마더보드에서 “LFB” 혹은 “Linear Frame Buffer”라고 하는 옵션이 있다는 보고를 받았습니다. 두 가지 설정을 할 수 있습니다: “Disabled” 및 “1 Megabyte”입니다. “1 Megabyte”로 설정하십시오. Disabled인 경우에는, 설치 플로피를 제대로 읽지 못하고, 시스템이 결국 멈추었습니다. 현재로서는 이 특정 마더보드에서 무슨 일이 벌어지고 있는 건지 알아내지 못했습니다 — “1 Megabyte”로 설정하면 동작하고 아니면 동작하지 않는 것 밖에 알지 못합니다.

3.6.3.6 Advanced Power Management

마더보드에서 Advanced Power Management (APM) 기능이 들어 있으면, APM에서 전원 관리 기능을 컨트롤하도록 설정하십시오. doze, standby, suspend, nap, sleep 모드를 끄시고, hard disk’s power-down timer 기능을 끄십시오. 리눅스에서 이 모드에 대한 컨트롤을 할 수 있고, BIOS보다 더 전원 관리를 잘 합니다.

3.6.4 주의해야 할 하드웨어 이슈

많은 사람들이 CPU를 90 MHz를 100 MHz와 같이 오버클럭킹해서 돌리고 있습니다. 이렇게 해도 동작하기도 하지만, 온도 및 그 밖의 요인에 민감하고 시스템이 망가질 수도 있습니다. 이 문서의 저자중의 한 명은 1년동안 시스템을 오버클럭킹해서 써 왔는데 gcc 프로그램으로 운영 체제 커널을 컴파일할 때마다 예상치 못한 시그널로 이상 종료하곤 했습니다. CPU 속도를 원래대로 돌려 놓으니까 문제가 해결되었습니다.

메모리 모듈에 문제가 있을 때 (아니면 데이터를 교환할 때 하드웨어 문제가 있을 때) 처음으로 죽는 프로그램은 보통 gcc 컴파일러입니다. gcc는 커다란 데이터 구조를 만들고 그 사이를 계속해서 왔다갔다 하기 때문입니다. 이 데이터 구조에 문제가 생기면 존재하지 않는 주소에 접근해서 잘못된 인스트럭션이 되어 버립니다. 그 증상은 gcc가 예상치 못한 시그널로 죽는 것입니다.

아주 좋은 마더보드는 패리티 RAM을 지원하고 RAM에 한 비트 오류가 발생했을 때 알려줍니다. 아쉽게도, 그 오류를 고칠 수 있는 방법은 없기 때문에, 그 RAM이 있다는 사실을 알자마자 죽어 버립니다. 그래도 데이터에 오류를 집어 넣는 것보다는 메모리가 잘못되었다고 알려주는 편이 낫습니다.

아주 좋은 시스템에는 마더보드가 패리티를 지원하고 진짜 패리티 메모리 모듈이 들어 있습니다. [2.4.3 절](#) 부분을 참고하십시오.

진짜 패리티 RAM이 있고 마더보드에서 패리티 RAM을 지원한다면, BIOS 설정에서 그 기능을 켜서 메모리 패리티 오류가 발생했을 때 인터럽트를 내도록 하십시오.

3.6.4.1 Turbo 스위치

많은 시스템에서는 *turbo* 스위치가 있어서 CPU의 속도를 조절합니다. 고속의 설정을 선택하십시오. BIOS에서 소프트웨어적으로 turbo 스위치를 조정할 수 있다면 (아니면 CPU 속도를 소프트웨어로 조정한다면), 시스템을 고속 모드로 고정해 놓으십시오. 어떤 시스템에서는 리눅스가 하드웨어 장치를 검색하는 동안 turbo 스위치의 소프트웨어 컨트롤 기능을 건드렸다는 보고가 있었습니다.

3.6.4.2 Cyrix CPU 및 플로피 디스크 오류

많은 Cyrix CPU 사용자들은 설치하는 동안 캐시 기능을 꺼야 했습니다. 끄지 않으면 플로피 디스크에서 오류가 발생했습니다. 캐시 기능을 꺼야 하는 경우, 설치를 마치면 캐시를 다시 켜십시오. 캐시를 끈 상태에서는 시스템이 훨씬 더 느리게 동작합니다.

이 현상이 Cyrix CPU의 문제라고 생각하지는 않습니다. 아마도 리눅스에서 이 문제를 피해갈 수 있을 것입니다. 이 문제의 원인은 계속 찾아볼 것입니다. 기술적인 문제가 궁금하신 분들을 위해 말씀드리면, 16비트에서 32비트 코드로 넘어갈 때 캐시가 무효화되면서 발생하는 문제라고 추정됩니다.

3.6.4.3 주변 장치 하드웨어 설정

컴퓨터의 주변 장치 카드의 설정을 바꾸거나 점퍼를 바꿔야 할 수도 있습니다. 일부 카드는 설정 메뉴가 있고, 어떤 카드는 점퍼를 바꿔야 합니다. 이 문서에서는 각각의 모든 하드웨어 장치에 대한 정보가 들어 있지는 않습니다. 그저 유용한 팁들을 전달하길 바랄 뿐입니다.

카드에서 “mapped memory”를 지원한다면, 메모리는 0xA0000과 0xFFFFF 사이에 (640K 위에서 1메가바이트 아래) 들어 있거나, 시스템의 전체 RAM보다 최소 1메가바이트 큰 위치에 들어 있어야 합니다.

3.6.4.4 USB BIOS 지원 및 키보드

AT 방식 키보드가 없고 USB 키보드만 있는 경우에는, BIOS 설정에서 legacy AT keyboard emulation을 켜야 할 수도 있습니다. 설치 시스템에서 USB 모드로 키보드를 사용할 수 없는 경우에만 사용하십시오. 반대로 어떤 시스템에서는 (특히 노트북 컴퓨터의 경우) 키보드가 동작하지 않을 때 legacy USB 지원을 꺼야 할 수도 있습니다. 메인보드 매뉴얼을 살펴 보시고 BIOS에서 “Legacy keyboard emulation”이나 “USB keyboard support” 옵션을 찾으십시오.

3.6.4.5 64 MB RAM 이상

리눅스 커널에서 RAM의 용량을 항상 정확히 알아내지는 못합니다. 그런 경우, [5.2절](#)에 있는 내용을 참고하십시오.

Chapter 4

시스템 설치 미디어 얻기

4.1 공식 Debian GNU/Linux CD-ROM 세트

여러 가지 방법 중에서 Debian GNU/Linux 설치에 가장 쉬운 방법은 공식 데비안 CD-ROM 세트로 설치하는 것입니다. CD 세트는 판매하는 회사에서 (CD 판매 회사 페이지 <<http://www.debian.org/CD/vendors/>> 참고) 구입할 수 있습니다. 네트워크 연결이 빠르고, CD 라이터가 있으면, 데비안 미러 사이트에서 CD-ROM 이미지를 내려 받아서 직접 CD 세트를 만들 수도 있습니다. (CD 만드는 방법은 데비안 CD 페이지 <<http://www.debian.org/CD/>> 참고) 데비안 CD 세트가 있고 CD에서 부팅할 수 있으면, 바로 5장 부분으로 넘어갈 수 있습니다. 이 부팅 CD에는 가능한 범위 내에서 대부분의 사람들에게 필요한 파일들이 들어 있습니다. 바이너리 꾸러미 모음은 전부 여러 장의 CD가 필요하지만, 아마도 3번째 이후의 CD에서 꾸러미가 필요한 일은 없을 겁니다. DVD 버전을 쓸 수도 있습니다. DVD 버전은 보관할 공간도 줄어들고, CD를 이것 저것 교환해야 하는 일도 없습니다.

CD 세트가 있지만 CD 부팅을 지원하지 않는 경우에는, 플로피 디스크, 하드 디스크, USB 스틱, 네트워크 부팅, 아니면 수동으로 CD에서 커널을 읽어들이어서 시스템 설치 프로그램을 맨 처음 부팅할 수 있습니다. 다른 방법으로 부팅할 때 필요한 파일은 CD 안에도 들어 있습니다. 데비안 네트워크 아카이브와 CD의 폴더 구조는 완전히 동일합니다. 그래서 아래에 부팅하는 데 필요한 어떤 파일의 경로가 있으면, CD의 같은 서브 디렉토리 안에서 그 파일을 찾아 보십시오.

설치 프로그램을 부팅하기만 하면, 필요한 다른 파일들은 모두 CD에서 이용할 수 있습니다.

CD 세트가 없으면, 설치 프로그램의 시스템 파일들을 내려 받아서 플로피 디스크나 하드 디스크나 USB 스틱이나 네트워크로 연결된 컴퓨터에 저장해 놓으십시오. 그러면 이 파일을 이용해 설치 프로그램을 부팅할 수 있습니다.

4.2 데비안 미러에서 파일 내려받기

가장 가까이 있는 (그래서 가장 빠를 것 같은) 미러 사이트를 찾으려면, 데비안 미러 목록 <<http://www.debian.org/distrib/ftplist>>을 참고하십시오.

데비안 미러에서 파일을 내려받을 때, 파일을 바이너리 (*binary*) 모드로 받도록 하십시오. 텍스트나

(text) 자동 (automatic) 모드로 받으면 안 됩니다.

4.2.1 설치 이미지를 찾을 위치

설치 이미지는 데비안 미러의 `debian/dists/sarge/main/installer-i386/current/images/` <<http://http.us.debian.org/debian/dists/sarge/main/installer-i386/current//images>> 디렉토리 안에 있습니다. MANIFEST <<http://http.us.debian.org/debian/dists/sarge/main/installer-i386/current//images/MANIFEST>> 파일을 보면 각 이미지 이름과 그 용도가 쓰여 있습니다.

4.3 디스크 이미지에서 플로피 만들기

보통 CD나 다른 방법으로 부팅할 수 없는 하드웨어의 경우, 설치 프로그램을 부팅하는 마지막 방법으로 부팅 플로피 디스크를 사용합니다.

디스크 이미지는 플로피 디스크의 모든 내용이 로우 형태로 들어 있는 파일입니다. `boot.img` 같은 디스크 이미지는 플로피 드라이브에 그냥 복사할 수 없습니다. 이미지 파일을 플로피 디스크에 로우 모드로 쓰려면 특별한 프로그램을 사용합니다. 이미지 파일은 디스크 전체의 내용을 그대로 나타내는 파일이기 때문입니다. 파일의 데이터를 플로피에 섹터 복사해야 합니다.

디스크 이미지에서 플로피를 만드는 방법은 플랫폼에 따라 세 가지가 있습니다. 여기서는 여러 가지 플랫폼에서 디스크 이미지로 플로피를 만드는 방법을 설명합니다.

플로피를 만들 때 어떤 방법을 사용하든 간에, 플로피를 쓴 다음에 쓰기 방지 탭을 걸어 놓아서 플로피 내용이 손상되지 않도록 하십시오.

4.3.1 리눅스나 UNIX 시스템에서 디스크 이미지 쓰기

플로피 디스크 이미지 파일을 플로피 디스크에 쓰려면, 보통은 시스템의 루트 권한이 있어야 합니다. 비어 있는 플로피를 플로피 드라이브에 넣으십시오. 그리고 다음 명령을 사용하십시오:

```
$ dd if=파일이름 of=/dev/fd0 bs=1024 conv=sync ; sync
```

여기서 `파일이름`은 플로피 디스크 이미지 파일의 이름입니다. (`파일이름`을 뭐라고 해야 할지는 4.2절 부분을 참고하십시오) 플로피 디스크 장치 이름으로는 `/dev/fd0`을 가장 많이 씁니다. 워크스테이션에서는 다를 수도 있습니다. 플로피 디스크에 데이터를 다 쓰기도 전에 명령어가 끝나서 프롬프트가 나올 수도 있기 때문에, 플로피 디스크가 사용중이라는 불이 깜박이는 지 잘 살펴 보시고 깜박이지 않을 때 플로피를 드라이브에서 빼야 합니다. 어떤 시스템에서는 플로피를 빼려면 플로피를 꺼내는 별도의 명령을 실행해야 합니다.

어떤 시스템에서는 드라이브에 플로피 디스크를 넣으면 자동으로 마운트합니다. 로우 모드로 플로피를 쓰려면 그 전에 이 기능을 꺼야 합니다. 불행히도, 이 기능을 끄는 방법은 운영 체제마다 다릅니다.

4.3.2 DOS, Windows, OS/2에서 디스크 이미지 쓰기

i386 기계의 경우는, 다음 명령 중의 하나를 이용해 이미지를 플로피로 복사하십시오.

MS-DOS에서는 **rawrite1** 및 **rawrite2** 프로그램을 사용할 수 있습니다. Windows에서는 DOS 창에서 사용하십시오. Windows 탐색기에서 이 프로그램을 두 번 클릭해 봤자 동작하지 않습니다.

rwwrtwin 프로그램은 Windows 95, NT, 98, 2000, ME, XP에서 (그리고 아마도 그 이후 버전에서도) 동작합니다. 사용하려면 같은 디렉토리에서 **diskio.dll** 파일을 풀어야 합니다.

이 도구는 공식 데비안 CD-ROM의 **/tools** 디렉토리 아래에 있습니다.

4.4 USB 메모리 스틱 부팅에 필요한 파일 준비하기

USB 스틱을 준비하려면 GNU/Linux가 동작하고 USB를 지원하는 시스템이 필요합니다. **usb-storage** 커널 모듈을 읽어들이고 (**modprobe usb-storage**) 어떤 SCSI 장치로 USB 스틱이 매핑되었는지 (이 예에서는 **/dev/sda**를 사용) 알아야 합니다. USB 메모리 스틱에 쓰려면, 쓰기 방지 스위치를 풀어 놓아야 할 수도 있습니다.

단 USB 스틱은 크기가 최소 128MB는 되어야 합니다. (4.4.2절에 쓰여 있는 대로 하면 더 작아도 가능합니다)

4.4.1 파일 복사하기 — 쉬운 방법

There is an all-in-one file **hd-media/boot.img.gz** which contains all the installer files (including the kernel) as well as **SYSLINUX** and its configuration file. You only have to extract it directly to your USB stick:

```
# zcat boot.img.gz > /dev/sda
```

주의

Using this method will destroy anything already on the device. Make sure that you use the correct device name for your USB stick.

그 다음 USB 메모리 스틱을 마운트하십시오. (**mount /dev/sda /mnt**) 이 USB 메모리 스틱에는 a FAT filesystem이 들어 있고, 그 안에 데비안 네트워크 설치 ISO 이미지 혹은 비지니스 카드 ISO 이미지가 들어 있습니다. 파일 이름이 **.iso**로 끝나야 한다는 것에 주의하십시오. 다 끝나면 마운트 해제하십시오. (**umount /mnt**)

4.4.2 파일 복사하기 — 유연한 방법

좀 더 유연한 방법이 좋다면, 아니면 무슨 일이 일어나고 있는지 보고 싶기라도 하다면, 다음 방법으로 파일을 USB 스틱에 집어 넣으십시오.

4.4.2.1 Intel x86에서 USB 스틱 파티션하기

다음은 메모리 스틱에서 전체 장치를 사용하지 않고, 첫 번째 파티션을 사용하는 방법입니다.

참고

대부분의 USB 장치는 한 개의 FAT16 파티션으로 미리 포맷되어 있기 때문에, 다시 파티션 하거나 다시 포맷할 필요가 없습니다. 해야 한다면, **fdisk**나 다른 파티션 도구를 이용해 FAT16 파티션을 만들고 다음 명령으로 파일 시스템을 만드십시오:

```
# mkdosfs /dev/sda1
```

USB 스틱의 장치명을 올바르게 사용해야 합니다. **mkdosfs** 명령은 **dosfstools** 데비안 꾸러미에 들어 있습니다.

USB 스틱에서 부팅한 다음 커널을 시작하려면 부트 로더를 USB 스틱에 넣어야 합니다. 그 어떤 부트 로더라도 (예를 들어 **LILO**) 동작하지만, **SYSLINUX**를 사용하는 게 편리합니다. **SYSLINUX**는 FAT16 파티션을 사용하고 텍스트 파일을 편집하는 것만으로 설정을 바꿀 수 있기 때문입니다. FAT 파일 시스템을 지원하는 어떤 운영 체제라도 부트 로더를 설정할 때 이용할 수 있습니다.

SYSLINUX를 USB 스틱의 FAT16 파티션에 넣으려면, **syslinux**와 **mttools** 꾸러미를 설치하고, 다음 명령을 실행합니다:

```
# syslinux /dev/sda1
```

다시 말하지만, 신경 써서 장치 이름을 올바르게 쓰십시오. **SYSLINUX**를 실행할 때 그 파티션을 마운트하면 안 됩니다. 이 명령은 파티션에 부트 섹터를 쓰고, 부트로더 코드가 들어 있는 **ldlinux.sys** 파일을 만듭니다.

파티션을 마운트하고 (`mount /dev/sda1 /mnt`) 데비안 아카이브에서 다음 파일을 스틱으로 복사하십시오:

- **vmlinuz** (커널 바이너리)

- **initrd.gz** (최초 램디스크 이미지)
- **syslinux.cfg** (SYSLINUX 설정 파일)
- 추가 커널 모듈

파일 이름을 바꾸려면, **SYSLINUX**에서는 DOS 파일이름만 (8.3 방식) 사용할 수 있다는 점을 주의 하십시오.

syslinux.cfg 설정 파일에는 다음 두 줄이 들어 있습니다:

```
default vmlinuz
append initrd=initrd.gz ramdisk_size=12000 root=/dev/ram rw
```

부팅하는 이미지에 따라서는 **ramdisk_size** 파라미터 값을 늘려야 할 수도 있습니다. 부팅이 실패하면, **devfs=mount,dall** 파라미터를 “뒤에” 붙여 보십시오.

4.4.2.2 ISO 이미지 추가하기

데비안 ISO 이미지를 (비즈니스 카드, 네트워크 설치 아니면 완전한 버전이라도) USB 스틱에 (들어 간다면) 넣습니다. 이미지의 파일 이름은 **.iso**로 끝나야 합니다.

ISO 이미지 없이 네트워크로 설치하려면, 앞의 단계를 건너 뛰십시오. 또 최초 램디스크로 **hd-media** 디렉토리에 있는 파일 말고, **netboot** 디렉토리에 있는 파일을 사용해야 합니다. **hd-media/initrd.gz** 이미지는 네트워크를 지원하지 않습니다.

다 끝나면, USB 메모리 스틱의 마운트를 해제하고 (**umount /mnt**) 쓰기 방지 스위치를 거십시오?

4.4.2.3 USB 스틱 부팅하기

주의

메모리 스틱에서 부팅하지 못한다면, USB 스틱의 master boot record가 (MBR) 잘못되었을 수 있습니다. 바로잡으려면 **mbr** 꾸러미에서 **install-mbr** 명령을 사용하십시오:

```
# install-mbr /dev/sda
```

4.5 하드 디스크 부팅에 필요한 파일 준비하기

설치 프로그램은 기존 하드 드라이브 파티션에 들어 있는 부팅 파일을 이용해서 부팅할 수 있습니다. 다른 운영체제를 통해서 실행할 수도 있고, 아니면 BIOS에서 직접 부트 로더를 실행할 수도 있습니다.

“네트워크만” 사용해 완전히 설치하려면 이 방법을 사용하면 됩니다. 네트워크 설치를 사용하면 CD 이미지를 찾아서 구울 필요도 없고, 여러 장의 불안한 플로피와 씨름할 필요도 없습니다.

설치 프로그램은 NTFS 파일시스템의 파일에서는 부팅할 수 없습니다.

4.5.1 LILO 혹은 GRUB을 이용한 하드 디스크 설치 프로그램 부팅

여기서는 **LILO** 혹은 **GRUB**을 이용해 기존에 설치한 리눅스에 새로 추가하거나 아니면 기존 리눅스를 덮어 쓰는 방법을 설명합니다.

부팅할 때 부트 로더 두 가지 모두 커널은 물론이고, 디스크 이미지도 메모리에 올리는 기능을 지원합니다. 이 로우 디스크는 커널의 루트 파일 시스템으로 사용할 수 있습니다.

데비안 아카이브의 다음 파일을 하드 드라이브의 편리한 위치로 복사하십시오. 예를 들어 `/boot/newinstall/` 같은 위치로 복사하십시오.

- `vmlinuz` (커널 바이너리)
- `initrd.gz` (램디스크 이미지)

마지막으로 부트 로더를 설정하려면 [5.1.2절](#) 부분으로 진행하십시오.

4.6 TFTP 네트워크 부팅에 필요한 파일 준비하기

LAN에 연결되어 있다면, 네트워크를 통해 TFTP를 사용해서 다른 기계에서 부팅할 수도 있습니다. 다른 기계에서 설치 시스템을 부팅하려고 한다면, 부팅 파일을 특정 위치에 복사해 놓고, 해당 기계의 부팅을 지원하도록 설정해 놓아야 합니다.

TFTP 서버를 설정해야 합니다. 그리고 많은 기계에서 BOOTP 서버, 아니면 DHCP 서버를 설정해야 합니다.

BOOTP는 컴퓨터에게 그 IP 주소 및 네트워크의 어디에서 부팅 이미지를 가져와야 하는 지 알려주는 IP 프로토콜입니다. DHCP는 (Dynamic Host Configuration Protocol) 더 유연하며, BOOTP와 호환되는 확장 기능입니다. 일부 시스템은 DHCP를 이용해야만 설정할 수 있습니다.

Trivial File Transfer Protocol은 (TFTP) 부팅 이미지를 클라이언트에게 넘겨줄 때 사용합니다. 이론상 이 프로토콜을 사용한다면 어떤 플랫폼의 어떤 서버라도 사용할 수 있습니다. 여기서는 SunOS 4.x, SunOS 5.x (Solaris), 그리고 GNU/Linux에서 사용하는 명령을 예로 듭니다.

참고

TFTP 부팅에서 Pre-boot Execution Environment (PXE) 방식을 사용하려면, `tsize`를 지원하는 TFTP 서버가 있어야 합니다. Debian GNU/Linux 서버에서는, `atftpd` 및 `tftpd-hpa` 꾸러미가 이 기능을 지원합니다. `tftpd-hpa`를 권장합니다.

4.6.1 BOOTP 서버 준비하기

GNU/Linux용 BOOTP 서버는 두 가지가 있습니다. CMU `bootpd`가 있고, 그 외의 프로그램은 실제로 DHCP서버인, ISC `dhcpd`가 있습니다. ISC `dhcpd`는 Debian GNU/Linux의 `bootpd`와 `dhcp` 꾸러미에 들어 있습니다.

CMU `bootpd`를 사용하려면, `/etc/inetd.conf`에서 관련된 줄의 코멘트를 없애거나 한 줄을 추가해야 합니다. Debian GNU/Linux에선 `update-inetd --enable bootps`를 실행한 다음 적용하려면 `/etc/init.d/inetd reload` 명령을 실행합니다. 그 줄은 다음과 같은 모양입니다:

```
bootps dgram udp wait root /usr/sbin/bootpd bootpd -i -t 120
```

이제 `/etc/bootptab` 파일을 만들어야 합니다. 이 파일은 과거의 BSD `printcap`, `termcap`, `disktab` 파일과 같이 익숙하면서도 의미를 알기 어려운 형식으로 되어 있습니다. 더 자세한 정보는 `bootptab` 매뉴얼 페이지를 참고하십시오. CMU `bootpd`에서는, 클라이언트의 하드웨어 (MAC) 주소를 알아야 합니다. 다음은 `/etc/bootptab`의 예입니다:

```
클라이언트:\
hd=/tftpboot:\
bf=tftpboot.img:\
ip=192.168.1.90:\
sm=255.255.255.0:\
sa=192.168.1.1:\
ha=0123456789AB:
```

위에서 최소한 “ha” 옵션은 바뀌어야 합니다. 이 옵션은 클라이언트의 하드웨어 주소를 나타냅니다. “bf” 옵션은 TFTP로 클라이언트가 받게 될 파일을 지정합니다. 자세한 정보는 [4.6.4절](#) 부분을 참고하십시오.

반대로, ISC `dhcpd` 설정은 정말 쉽습니다. BOOTP 클라이언트를 DHCP 클라이언트의 일종의 특별한 케이스로 취급하기 때문입니다. 일부 아키텍처에서는 BOOTP로 부팅하게 하려면 복잡한 설정이 필요합니다. 그러한 아키텍처의 경우는 [4.6.2절](#) 부분을 참고하십시오. 그게 아니라면 `allow bootpd`를 클라이언트가 들어 있는 서브넷의 설정 부분에 집어 넣고, `/etc/init.d/dhcpd restart` 명령으로 `dhcpd`

를 다시 시작하면 됩니다.

4.6.2 DHCP 서버 설정하기

자유 소프트웨어 DHCP 서버로 ISC `dhcpd`가 있습니다. Debian GNU/Linux에서 ISC `dhcpd`는 `dhcp` 꾸러미 안에 들어 있습니다. 다음은 간단한 설정 파일 예제입니다 (`/etc/dhcpd.conf`):

```
option domain-name "example.com";
option domain-name-servers ns1.example.com;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
server-name "servername";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.200 192.168.1.253;
    option routers 192.168.1.1;
}

host clientname {
    filename "/tftpboot/tftpboot.img";
    server-name "servername";
    next-server servername;
    hardware ethernet 01:23:45:67:89:AB;
    fixed-address 192.168.1.90;
}
```

주의: 최근의 (권장) `dhcp3` 꾸러미는 `/etc/dhcp3/dhcpd.conf` 파일을 사용합니다.

이 예제에서는, DHCP 서버, TFTP 서버, 네트워크 게이트웨이 역할을 모두 하는 `servername`이라는 서버가 있다고 가정합니다. `domain-name` 옵션도 바꾸고, 서버 이름과 클라이언트 하드웨어 주소도 바뀌어야 합니다. `filename` 옵션은 TFTP로 가져오려는 파일의 이름입니다.

`dhcpd` 설정 파일을 편집한 다음에, `/etc/init.d/dhcpd restart` 명령으로 `dhcpd`를 다시 시작하십시오.

4.6.2.1 DHCP 설정에서 PXE 부팅하게 만들기

다음은 TFTP의 Pre-boot Execution Environment (PXE) 방식을 이용한 `dhcp.conf`의 예입니다.

```
option domain-name "example.com";
```

```

default-lease-time 600;
max-lease-time 7200;

allow booting;
allow bootp;

# 다음 부분은 여러분 경우에 맞춰 바꿔야 합니다
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.200 192.168.1.253;
    option broadcast-address 192.168.1.255;
# 게이트웨이 주소입니다.
# (예를 들어 인터넷에 접근하는 방법에 따라 다를 수도 있습니다)
    option routers 192.168.1.1;
# 사용할려는 DNS
    option domain-name-servers 192.168.1.3;
}

group {
    next-server 192.168.1.3;
    host tftpclient {
# TFTP 클라이언트 하드웨어 주소
        hardware ethernet 00:10:DC:27:6C:15;
        filename "/tftpboot/pxelinux.0";
    }
}

```

PXE 부팅의 경우, 커널 이미지가 아니라 클라이언트 파일 이름 **pxelinux.0**이 부트 로더입니다. (아래의 [4.6.4절](#) 부분 참고)

4.6.3 TFTP Server 사용하기

TFTP 서버를 준비하려면, 먼저 **tftpd** 서버를 사용도록 해야 합니다. 보통 다음과 같은 줄을 **/etc/inetd.conf**에 넣으면 됩니다:

```
tftp dgram udp wait nobody /usr/sbin/tcpd in.tftpd /tftpboot
```

데비안 꾸러미의 경우에는 설치만 하면 기본적으로 이런 설정이 들어갑니다.

이 파일을 보고 **in.tftpd**의 인자로 사용하는 디렉토리를 기억해 두십시오. 아래에서 이 디렉토리 이름이 필요합니다. **in.tftpd** 일부 버전에서는, **-l** 옵션을 쓰면 모든 요청을 시스템 로그에 기록합니다.

이 옵션은 부팅 오류의 원인을 파악하는 데 유용합니다. `/etc/inetd.conf` 파일을 고쳤다면, 고쳤다는 사실을 `inetd` 프로세스에 알려야 합니다. 데비안 기계에서는 `/etc/init.d/inetd reload` 명령을 실행하고, 다른 곳에서는 `inetd`의 프로세스 ID를 찾아서 `kill -HUP inetd-pid` 명령을 실행합니다.

4.6.4 TFTP 이미지를 적당한 위치에 옮기기

그 다음에, 필요한 TFTP 부팅 이미지를 4.2.1절에 쓰여 있는 것처럼 `tftpd` 부팅 이미지 디렉토리에 복사해 놓으십시오. 보통 이 디렉토리는 `/tftpboot`입니다. 이 파일에서, `tftpd`가 특정 클라이언트를 부팅할 때 사용하는 특정 파일로 링크를 만들어야 할 것입니다. 불행히도 그 파일의 이름은 TFTP 클라이언트가 결정하고, 어떤 표준도 없습니다.

PXE 부팅의 경우, 필요한 모든 것이 `netboot/netboot.tar.gz` 압축 안에 준비되어 있습니다. 이 압축을 `tftpd` 부팅 이미지 디렉토리에서 풀기만 하십시오. `tftpd`에 부팅할 파일 이름으로 `/pxelinux.0`을 넘기도록 DHCP 서버에 설정하십시오.

4.7 자동 설치

여러 컴퓨터에 설치하려면 완전 자동 설치를 사용할 수도 있습니다. 이런 목적을 위한 데비안 꾸러미는 `fai` (설치 서버 사용), `replicator`, `systemimager`, `autoinstall`이 있고, 데비안 설치 프로그램 자체를 사용할 수도 있습니다.

4.7.1 데비안 설치 프로그램을 이용한 자동 설치

The Debian Installer supports automating installs via preconfiguration files. A preconfiguration file can be loaded from the network or from removable media, and used to fill in answers to questions asked during the installation process.

Although most dialogs used by `debian-installer` can be preseeded using this method, there are some notable exceptions. You can (re)partition an entire disk or use available free space on a disk; it is not possible to use existing partitions. You currently cannot use preseeding to set up RAID and LVM. Also, with the exception of network driver modules, it is not possible to preconfigure kernel module parameters.

미리 설정해 놓은 파일은 `debconf-set-selections` 명령에서 사용하는 형식으로 되어 있습니다. C.1절에 보면 잘 설명되어 있고 동작하는 예제 파일이 있습니다.

그게 아니면, 미리 설정할 수 있는 값들의 모든 목록이 들어 있는 파일을 만드는 한 가지 방법은, 수동으로 설치한 다음 `debconf-utils` 꾸러미에 들어 있는 `debconf-get-selections`를 사용하는 것입니다. 이 명령은 `debconf` 데이터베이스와 `/var/log/debian-installer/cdebconf` 안에 있는 `cdebconf` 데이터베이스를 한 개 파일로 만듭니다:

```
$ debconf-get-selections --installer > file
```

```
$ debconf-get-selections >> file
```

하지만, 이렇게 만든 파일에는 미리 설정하면 안 되는 것도 들어 있습니다. **C.1절**에 들어 있는 파일이 대부분의 사용자가 시작하기에 더 나은 방법입니다.

미리 설정해 놓은 파일이 있으면, 이 파일을 필요에 따라 편집하고, 웹 서버에 넣든지, 설치 프로그램의 부팅 미디어에 복사해 놓을 수도 있습니다. 이 파일을 어디에 놓든 이 파일을 이용하라는 사실을 부팅할 때 파라미터로 넘겨야 합니다.

설치 프로그램에서 미리 설정해 놓은 파일을 네트워크에서 내려받도록 하려면, `preseed/url=http://url/to/preseed.c` 파라미터를 커널 부팅 파라미터로 넘깁니다. 물론 이 파일은 설치 프로그램에서 네트워크를 설정하고 이 파일을 내려받을 때까지는 효과가 없습니다. 그래서 이런 방법은 설치 프로그램에서 DHCP를 통해 네트워크를 설정할 수 있을 경우에 가장 유용합니다. 설치 우선순위를 "중요"로 해서 네트워크를 설정할 때까지 아무 질문도 하지 않도록 할 수 있습니다. **5.2.1절** 부분을 참고하십시오.

CD에 미리 설정해 놓은 파일을 넣으려면, 미리 설정해 놓은 파일을 포함하도록 ISO 이미지를 다시 만들어야 할 것입니다. 자세한 방법은 `mkisofs` 매뉴얼 페이지를 참고하십시오. 다른 방법으로는, 미리 설정해 놓은 파일을 플로피에 넣고, `preseed/file=/floppy/preseed.cfg` 파라미터를 사용하는 것입니다.

USB 메모리 스틱에서 부팅한다면, 미리 설정해 놓은 파일을 메모리 스틱의 파일 시스템에 복사해 놓기만 하면 됩니다. 그 다음에 `syslinux.cfg` 파일을 편집해서 커널 부팅 파라미터에 `preseed/file=/hd-media/preseed.cfg` 파라미터를 넣으십시오.

Chapter 5

설치 시스템 부팅하기

5.1 Intel x86에서 설치 프로그램 부팅하기

5.1.1 CD-ROM에서 부팅하기

대부분의 사람들에게 가장 쉬운 방법은 데비안 CD 세트를 이용하는 것입니다. CD 세트를 가지고 있고, 기계에서 직접 CD 부팅을 할 수 있다면 매우 좋은 경우입니다. [3.6.2절](#)에 설명되어 있는 대로 CD에서 시스템을 부팅할 수 있도록 설정하고, CD를 넣은 다음, 다시 부팅하고, 다음 장으로 넘어가십시오.

일부 CD 드라이브의 경우에는 특별한 드라이버가 필요하므로, 설치 단계의 앞 부분에서는 CD 드라이브에 접근할 수 없을 수도 있습니다. 하드웨어에서 표준적인 방법의 CD 부팅이 동작하지 않는다면, 이 장을 다시 보시고 이 경우에 동작하는 다른 커널 및 설치 방법을 읽어보십시오.

CD-ROM에서 부팅할 수 없는 경우에도, CD-ROM에서 데비안 시스템 구성 요소와 꾸러미를 설치할 수 있습니다. 일단 플로피같은 다른 미디어를 이용해 부팅하십시오. 운영 체제, 베이스 시스템 및 다른 추가 꾸러미를 설치할 때가 되면 설치 시스템을 CD-ROM 드라이브로 맞춰 놓으십시오.

부팅하는 데 문제가 있으면, [5.3절](#) 부분을 참고하십시오.

5.1.2 LILO 혹은 GRUB을 사용해 리눅스 부팅하기

하드 디스크에서 설치 프로그램을 부팅하려면, 먼저 [4.5절](#)에서 설명한 필요한 파일들을 내려 받아 하드에 저장해야 합니다.

하드 드라이브를 부팅용으로만 사용하고 나머지는 모두 네트워크에서 받으려고 한다면, `netboot/debian-installer/i386/initrd.gz` 파일 및 해당 커널을 받아야 합니다. 설치 프로그램을 부팅한 다음에 하드 디스크를 다시 파티션할 수 있습니다. 단 주의를 기울여야 합니다.

다른 방법으로, 하드 드라이브의 현재 파티션을 바꾸지 않고 유지하려면, `hd-media/initrd.gz` 파일과 그 커널을 받으시고, CD iso 파일도 드라이브에 복사해 놓으십시오 (파일 이름이 `.iso`로 끝나도록 하십시오). 그러면 설치 프로그램이 이 드라이브에서 부팅한 다음 네트워크 없이 그 CD 이미지에서 설치할 수 있습니다.

LILO의 경우, `/etc/lilo.conf`에 있는 두 가지 중요한 부분을 설정해야 합니다:

- 부팅할 때 `initrd.gz` 설치 프로그램을 읽어들이어야 하고
- `vmlinuz` 커널의 루트 파티션으로 램 디스크를 사용하도록 합니다.

여기 예제 `/etc/lilo.conf` 파일이 있습니다:

```
image=/boot/newinstall/vmlinuz
    label=newinstall
    initrd=/boot/newinstall/initrd.gz
    root=/dev/ram0
    append="devfs=mount,dall ramdisk_size=12000"
```

더 자세한 정보는, `initrd(4)` 및 `lilo.conf(5)` 맨페이지를 참고하십시오. 이제 `lilo`를 실행하고 다시 부팅하십시오.

GRUB 사용법은 아주 비슷합니다. `menu.lst` 파일을 `/boot/grub/` 디렉토리에 (어떤 경우는 `/boot/boot/grub/`) 놓고, 다음 줄을 추가하십시오:

```
title New Install
kernel (hd0,0)/boot/newinstall/vmlinuz root=/dev/ram0 ramdisk_size=12000
initrd (hd0,0)/boot/newinstall/initrd.gz
```

그리고 다시 부팅하십시오. 부팅이 실패하면 “kernel” 줄에 `devfs=mount,dall`을 추가해 보십시오.

`ramdisk_size`의 값을 `initrd` 이미지 크기에 따라 조정해야 할 수도 있습니다. 이 이후부터는 **GRUB**와 **LILO** 사이에 차이점이 없습니다.

5.1.3 USB 메모리 스틱에서 부팅하기

3.6.2절 및 **4.4절**의 모든 것을 준비했다고 하면, USB 스틱을 빈 USB 커넥터에 꼽고 컴퓨터를 다시 부팅하십시오. 시스템이 부팅해서 `boot:` 프롬프트가 나와야 합니다. 거기에서 부팅 파라미터를 추가로 넣을 수도 있고, 그냥 **Enter**를 누를 수도 있습니다.

컴퓨터가 USB 메모리 장치에서 부팅을 지원하지 않는 경우, 플로피로 부팅한 다음에 USB로 바꿀 수 있습니다. **5.1.4절**에 설명된 대로 부팅하십시오. 부트 플로피의 커널은 USB 스틱을 자동으로 찾아낼 것입니다. 루트 플로피가 필요한 시점에, **Enter**를 누르십시오. 그러면 `debian-installer`가 시작할 것입니다.

5.1.4 플로피에서 부팅하기

필요한 플로피 이미지를 내려받아서 **4.3절**에 설명한 대로 이미지에서 플로피를 만듭니다.

설치 프로그램의 부트 플로피에서 부팅하려면, 플로피를 첫 번째 플로피 드라이브에 넣고, 평소에 하던 방법대로 시스템을 끈 다음 다시 켜십시오.

플로피를 이용해 LS-120 드라이브에서 (ATAPI 버전) 설치하는 경우, 플로피 장치의 가상 위치를 지정해야 합니다. `root=` 부팅 파라미터를 사용해, `ide-floppy` 드라이버가 어느 장치로 매핑되는 지 장치를 지정합니다. 예를 들어, LS-120 드라이브가 두 번째 케이블의 첫 번째 (master) IDE 장치에 연결되어 있는 경우, 부팅 프롬프트에서 `linux root=/dev/hdc`라고 입력합니다. LS-120에서 설치하는 건 2.4 이후의 커널에서만 지원됩니다.

몇몇 기계에서는 **Control-Alt-Delete** 키를 눌러도 컴퓨터를 리셋하지 않기 때문에, “하드” 리부팅을 권장합니다. 기존 운영 체제에서 설치하는 경우에는 (예를 들어 DOS에서) 다른 방법이 없습니다. 그 외의 경우에는 항상 부팅할 때 하드 리부팅을 하십시오.

플로피 디스크를 읽게 됩니다. 그 다음에 부트 플로피에 대해 소개하는 화면이 나오고 마지막에 **boot:** 프롬프트가 나옵니다.

Enter를 누르면, **Loading...** 메시지 다음에, **Uncompressing Linux...**라고 나오고, 시스템의 하드웨어에 관한 여러가지 정보가 화면 가득히 나옵니다. 이 부팅 과정에 대한 더 자세한 정보는 아래의 [5.3.4 절](#)에 있습니다.

부트 플로피에서 부팅한 다음, 루트 플로피를 요청합니다. 루트 플로피를 넣고 **Enter**를 누르십시오. 그러면 루트 플로피의 내용을 메모리에 읽어들이습니다. 설치 프로그램 **debian-installer**를 자동으로 실행합니다.

5.1.5 TFTP로 부팅하기

네트워크에서 부팅하려면, 네트워크에 연결되어 있어야 하고 TFTP 네트워크 부트 서버가 (DHCP, RARP 혹은 BOOTP) 필요합니다.

네트워크 부팅을 지원하는 설치 방법은 [4.6절](#)에 설명되어 있습니다.

i386에서 TFTP 부팅을 하는 여러 가지 방법이 있습니다.

5.1.5.1 PXE를 지원하는 NIC 혹은 마더보드

네트워크 인터페이스 카드나 마더보드에서 PXE 부팅 기능을 지원할 수도 있습니다. PXE는 Intel™에서 TFTP 부팅을 다시 만든 것입니다. PXE를 지원하는 경우 네트워크 부팅을 하도록 BIOS를 설정할 수 있을 것입니다.

5.1.5.2 네트워크 부트롬이 들어 있는 NIC

네트워크 인터페이스 카드에서 TFTP 부팅 기능을 지원할 수도 있습니다.

어떻게 했는 지 알려 주십시오 (<debian-boot@lists.debian.org>). 이 문서를 언급해 주십시오.

5.1.5.3 Etherboot

etherboot project <<http://www.etherboot.org>>에는 TFTPboot를 할 수 있는 부팅 디스켓과 부트롬이 있습니다.

5.1.6 부팅 프롬프트

설치 프로그램이 부팅할 때, 데비안 로고와 다음 부팅 프롬프트가 표시되는 친근한 그래픽 화면이 나타납니다:

Press F1 for help, or ENTER to boot:

부팅 프롬프트에서 그냥 **Enter** 키를 눌러 기본 옵션으로 설치 프로그램을 부팅할 수도 있고, 특정 부팅 방법이나 부팅 파라미터를 입력할 수도 있습니다.

F2에서 **F7**까지 누르다 보면 유용한 부팅 방법과 부팅 파라미터에 대한 정보를 볼 수 있습니다. 부팅 명령행에 파라미터를 넣으려면, 부팅 방법을 입력하는 걸 잊지 마시고 (기본값은 `linux`), 첫 번째 파라미터 앞에 공백을 넣으십시오 (예를 들어, `linux debconf/priority=medium`).

참고

VGA 콘솔 텍스트 인터페이스 기능이 들어 있는 원격 관리 장치를 이용해 시스템을 설치하는 경우에는, 설치 프로그램이 부팅할 때 나타나는 최초의 그래픽 스플래시 화면을 볼 수 없을 수도 있습니다. 부팅 프롬프트조차 볼 수 없을 수도 있습니다. 그러한 장치는 Compaq의 “integrated Lights Out” (iLO) 및 HP의 “integrated Remote Assistant”가 있습니다. 화면을 보지 못하더라도 F1을 누르면 그래픽 화면을 넘어가고 도움말 텍스트를 볼 수 있습니다. 스플래시 스크린이 넘어가면 도움말 텍스트에서 키를 누르면 그 키가 프롬프트에 쓰여집니다. 설치할 때 프레임버퍼를 사용하지 않으려면 도움말 텍스트에 표시된 것처럼 `debian-installer/framebuffer=false` 옵션을 부팅 프롬프트에 넘깁니다.

어떤 경우 이스케이프 시퀀스를 이용해야 이 키를 누를 수 있습니다. 예를 들어 IRA는 **Ctrl-F, 1**을 사용합니다.

5.2 부팅 파라미터

부팅 파라미터는 리눅스 커널 파라미터로 보통 주변 장치를 제대로 동작하도록 제어하는 데 이용합니다. 대부분 커널에서 주변 장치에 관한 정보를 자동으로 찾아 냅니다. 하지만 몇몇 경우에는 파라미터로 커널을 조금 도와줘야 합니다.

시스템을 첫 번째로 부팅하는 경우라면, 기본 부팅 파라미터를 시도해 보시고 (즉 파라미터를 사용하지 않는 것) 제대로 동작하는 지 보십시오. 보통은 제대로 동작할 것입니다. 제대로 동작하지 않는 경우에 다시 부팅해서 하드웨어에 필요한 특별한 파라미터가 있는 지 찾아 보십시오.

부팅 파라미터에 관한 정보는 Linux BootPrompt HOWTO <<http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html>>에 (여러가지 보기드문 하드웨어에 대한 정보 포함) 있습니다. 여기서는 몇 가지 많이 이용하는 파라미터에 대한 대략만 다룹니다. 많이 발생하는 문제점 몇 개는 5.3절에 들어 있습니다.

커널이 부팅할 때 다음과 같은 메시지가 나와야 합니다:

```
Memory:availk/totalk available
```

여기서 *total*은 전체 RAM의 크기입니다(킬로바이트 단위). 이 숫자가 실제 설치한 RAM의 용량과 다르다면, `mem=ram` 파라미터를 사용해야 합니다. 여기서 *ram*은 메모리의 양이고, 킬로바이트의 경우 “k”, 메가바이트의 경우 “m”을 뒤에 붙입니다. 예를 들어, `mem=65536k`나 `mem=64m`은 모두 64MB의 RAM을 뜻합니다.

시리얼 콘솔로 부팅하는 경우, 보통 자동으로 찾아냅니다. 시리얼 콘솔로 부팅하려는 컴퓨터에 비디오 카드와 (프레임버퍼) 키보드가 붙어 있다면, `console=device` 파라미터를 커널에 넘겨야 할 수 있습니다. 여기서 *device*는 시리얼 장치이고, `ttyS0`과 같이 씁니다.

5.2.1 데비안 설치 프로그램 파라미터

설치 시스템에서는 유용하게 쓸 수도 있는 부팅 파라미터 몇 개를 더 인식합니다¹.

`debconf/priority` 이보다 같거나 높은 우선순위의 메시지만 표시합니다.

기본 설치에서는 `debconf/priority=high`를 사용합니다. 즉, 높음 우선 순위와 필수 우선순위의 메시지를 표시하지만, 보통과 낮은 우선 순위 메시지는 넘어갑니다. 문제가 발생할 경우에는 설치 프로그램에서 우선순위를 필요에 따라 조정합니다.

`debconf/priority=medium`을 부팅 파라미터로 쓴 경우, 설치 메뉴에서 설치 메뉴에서 더 많은 조정을 할 수 있습니다. `debconf/priority=low`라고 쓴 경우, 모든 메시지를 볼 수 있습니다. (*expert* 부팅 방법과 동일합니다.) `debconf/priority=critical`의 경우, 필수 메시지만 표시하고 질문들에 신경 쓰지 않고 설치를 진행합니다.

`DEBIAN_FRONTEND` 이 부팅 파라미터는 설치 프로그램에서 사용할 사용자 인터페이스 종류를 설정합니다. 현재 가능한 파라미터 값은:

- `DEBIAN_FRONTEND=noninteractive`
- `DEBIAN_FRONTEND=text`
- `DEBIAN_FRONTEND=newt`
- `DEBIAN_FRONTEND=slang`

¹ 커널에서는 최대 8개의 명령행 옵션과 8개의 환경 옵션만 사용할 수 있습니다 (설치 프로그램에서 기본으로 사용하는 옵션 포함). 이 개수가 넘어가면 2.4 커널에서는 넘어가는 파라미터를 버리고, 2.6 커널은 멧어 버립니다.

- DEBIAN_FRONTEND=ncurses
- DEBIAN_FRONTEND=bogl
- DEBIAN_FRONTEND=gtk
- DEBIAN_FRONTEND=corba

기본 프론트엔드는 DEBIAN_FRONTEND=newt입니다. 시리얼 콘솔 설치의 경우 DEBIAN_FRONTEND=text가 더 좋을 수도 있습니다. 보통 기본 설치 미디어에서는 newt 프론트엔드만 사용할 수 있기 때문에, 이 파라미터는 현재 별로 쓸모가 없습니다.

BOOT_DEBUG 2 . 3 (. .)

BOOT_DEBUG=0 기본값입니다.

BOOT_DEBUG=1 보통때보다 많은 디버깅 정보.

BOOT_DEBUG=2 디버깅 정보 아주 많이.

BOOT_DEBUG=3 부팅 과정의 곳곳에서 셸을 실행해서 자세히 디버깅을 할 수 있습니다. 부팅을 계속하려면 셸을 나가면 됩니다.

INSTALL_MEDIA_DEV 이 파라미터의 값은 데비안 설치 프로그램을 읽어들이 장치의 경로입니다. 예를 들어 `INSTALL_MEDIA_DEV=/dev/floppy/0`

부트 플로피는 루트 플로피를 찾으면서, 검색할 수 있는 모든 플로피와 USB 저장 장치를 검색합니다. 이 파라미터를 이용해서 장치 한 개만 검색하도록 만들 수 있습니다.

debian-installer/framebuffer 어떤 아키텍처에서는 커널 프레임 버퍼를 사용해 여러 가지 언어로 설치를 지원합니다. 프레임버퍼가 문제가 있는 경우 `debian-installer/framebuffer=false` 파라미터를 통해 프레임버퍼를 끌 수 있습니다. 문제의 증상은 `bterm`이나 `bogl` 관련된 오류 메시지, 검은 화면, 설치를 시작하고 몇 분 후에 멈추는 증상입니다.

`video=vga16:off` 파라미터로 프레임버퍼를 끌 수도 있습니다. 그러한 문제가 Mobile Radeon 카드를 사용하는 Dell Inspiron에서 보고되었습니다.

debian-installer/probe/usb 부팅할 때 USB 검색이 문제가 있는 경우, `false`로 설정해 USB 검색을 하지 않습니다.

netcfg/disable_dhcp 기본값으로, **debian-installer**는 DHCP를 통해 네트워크 설정을 검색합니다. 검색이 성공하면, 그 설정을 검사해 보거나 바꿀 기회가 없습니다. DHCP 검색이 실패할 경우에만 수동 네트워크 셋업을 할 수 있습니다.

랜에 DHCP 서버가 있지만 DHCP를 사용하고 싶지 않을 때 (예를 들어 잘못된 응답을 한다든지 때문에), `netcfg/disable_dhcp=true` 파라미터를 사용하면 DHCP 네트워크 설정을 막고 수동으로 정보를 입력할 수 있습니다.

`hw-detect/start_pcmcia` PCMCIA에 문제가 있을 때 `false`로 하면 PCMCIA 서비스를 시작하지 않습니다. 일부 노트북 컴퓨터에서 이와 관련해 문제가 발생합니다.

`preseed/url` 미리 설정해 놓은 파일을 내려받을 URL을 지정합니다. 이 파일을 이용해 설치를 자동화합니다. [4.7절](#) 참고.

`preseed/file` 미리 설정해 놓은 파일을 읽어들이 URL을 지정합니다. 이 파일을 이용해 자동 설치를 합니다. [4.7절](#) 참고.

`ramdisk_size` 2.2.x 커널을 사용한다면, `ramdisk_size=13000` 로 설정해야 합니다.

5.3 설치 과정의 문제 해결

5.3.1 플로피 디스크 신뢰성

플로피 디스크를 사용해 설치하는 사람이 겪는 가장 큰 문제는 플로피 디스크 신뢰성 문제일 것입니다.

부트 플로피가 가장 문제가 많은 플로피입니다. 부트 플로피는 리눅스가 아니라 하드웨어에서 직접 읽어들이기 때문입니다. 가끔 하드웨어는 리눅스 플로피 디스크 드라이버가 하는 것 만큼 안정적으로 플로피를 읽지 못합니다. 또 잘못된 데이터를 읽은 경우에도 오류 메시지를 표시하지 않고 멈춰버리기도 합니다. 드라이버 플로피에서도 디스크 입출력 오류 메시지만 가득 나오는 문제가 발생하기도 합니다.

설치 도중에 특정 플로피에서 멈추는 현상이 발생하면, 가장 먼저 할 일은 플로피 디스크 이미지를 다시 받아서 다른 플로피에 만들어 보는 것입니다. 기존의 플로피를 다시 포맷하는 것만으로는 (어려없이 포맷이 되더라도) 안 됩니다. 다른 시스템에서 플로피를 만드는 게 도움이 될 수도 있습니다.

어떤 사용자는 플로피가 제대로 동작할 때까지 이미지를 무려 세 번 써야 했다고 하고, 세 번째 플로피에서 제대로 동작했다고 합니다.

또 어떤 사용자는 같은 플로피를 가지고 여러 번 다시 부팅하니 성공적으로 부팅했다고 합니다. 이러한 현상은 모두 문제가 있는 하드웨어나 펌웨어의 플로피 드라이버때문입니다.

5.3.2 부팅 설정

문제가 생겨서 커널이 부팅 과정에서 멈추거나, 주변 장치를 인식하지 못하거나, 드라이브를 제대로 인식하지 못하거나 하는 경우, 먼저 부팅 파라미터가 [5.2절](#)에 쓰여 있는 것처럼 제대로 되었는지 확인하십시오.

설치 프로그램에 들어 있는 커널 대신에 직접 만든 커널로 부팅하려면, CONFIG_DEVFS를 꼭 설정하십시오. 설치 프로그램에서 CONFIG_DEVFS가 필요합니다.

추가 장치나 주변 기기를 없앤 다음에 다시 부팅하면 문제가 없어질 수도 있습니다. 내장 모뎀, 사운드 카드 및 플러그인플레이 장치들이 특히 문제가 많습니다.

기계에 512M 보다 더 많은 메모리가 붙어 있는 경우에 설치 프로그램의 커널이 부팅할 때 멈춘다면, 커널에 보이는 메모리 양을 제한하는 부팅 파라미터를 써야 할 수도 있습니다. 예를 들어 mem=512m과 같이 씁니다.

5.3.3 자주 발생하는 Intel x86 설치 문제

자주 발생하는 설치 문제중에 몇 개는 설치 프로그램에 부팅 파라미터를 넘기는 방법으로 해결할 수 있습니다.

일부 시스템에는 “DCL이 뒤바뀐” 플로피가 달려 있습니다. 플로피가 문제가 없는 게 분명한데도 플로피를 읽다가 오류가 발생하면, floppy=thinkpad 파라미터를 시도해 보십시오.

IBM PS/1이나 ValuePoint같은 (ST-506 디스크 드라이브가 있는) 일부 시스템에서는 IDE 드라이브를 제대로 인식하지 못하기도 합니다. 먼저 파라미터 없이 IDE 드라이브를 제대로 인식하는 지 시도해 보십시오. 제대로 인식하지 못하면, 드라이브 크기를 (실린더, 헤드, 섹터) 알아보고, hd=cylinders,heads,sectors 파라미터를 사용하십시오.

아주 오래된 기계에서 **Checking 'hlt' instruction...**라고 나오면서 커널이 멈춘다면, hlt 인스트럭션 테스트를 하지 않도록 no-hlt 파라미터를 시도해 보십시오.

커널이 부팅할 때 화면에 괴상한 그림이 나온다면, 예를 들어 전부 하얀 색이 되거나 전부 검은 색이 되거나 가지가지 색의 픽셀 쓰레기가 나오거나 하면, 프레임버퍼 모드로 제대로 전환하지 못하는 문제 있는 비디오 카드가 붙어 있다는 뜻입니다. 그러면 debian-installer/framebuffer=false 혹은 video=vga16:off 파라미터로 프레임버퍼 콘솔을 끌 수 있습니다. 콘솔 기능의 한계때문에 설치할 때 언어는 영어만 사용할 수 있습니다. 자세한 건 5.2절 부분을 참고하십시오.

5.3.3.1 PCMCIA 설정 단계에서 시스템 멈춤

Dell에서 제조한 일부 노트북 모델은 PCMCIA 장치를 찾을 때 특정 하드웨어 주소에 접근하는 순간 멈춰 버립니다. 그 외의 노트북 모델도 비슷한 현상이 나타날 수 있습니다. 만약 그러한 문제가 발생하고 설치할 때 PCMCIA 기능이 필요없다면, hw-detect/start_pcmcia=false 부팅 파라미터로 PCMCIA를 끌 수 있습니다. 설치를 마친 다음에 PCMCIA를 설정해서 문제가 발생하는 구간을 제외할 수 있습니다.

다른 방법으로, 전문가 모드에서 설치 프로그램을 부팅할 수 있습니다. 그 다음에 하드웨어에 필요한 리소스 범위를 입력합니다. 예를 들어, 위에서 말한 Dell 노트북을 가지고 있다면, 여기서 exclude port 0x800-0x8ff라고 입력합니다. System resource settings section of the PCMCIA HOWTO <<http://pcmcia-cs.sourceforge.net/ftp/doc/PCMCIA-HOWTO-1.html#ss1.12>>에 보면 리소스 범위 관련 옵션의 목록이 있습니다. 설치 프로그램에서 이 값을 입력할 때, 쉼표를 쓰지 않는다는 것에 유의하십시오.

5.3.3.2 USB 모듈을 읽어들이다가 시스템 멈춤

일부 표준이 아닌 USB 키보드를 지원하기 위해 커널에서 USB 모듈과 USB 키보드 드라이버를 읽어 들입니다. 하지만 일부 문제 있는 USB 시스템에서는 드라이버를 읽어들이는 때 멈춥니다. 한 가지 방법은 메인보드의 BIOS 설정에서 USB 컨트롤러를 끄는 것입니다. 또 다른 방법은 `debian-installer/probe/usb=false` 파라미터를 부팅 프롬프트에 써서, USB 모듈을 읽어들이지 못하게 하는 것입니다.

5.3.4 커널 시작 메시지 해석하기

부팅 과정에서, `can't find something` 혹은 `something not present`, `can't initialize something`, 아니면 심지어는 `this driver release depends on something` 형식의 메시지를 볼 수 있을 것입니다. 이러한 메시지의 대부분은 아무런 문제를 일으키지 않습니다. 설치 시스템은 여러 가지 주변 장치가 달린 컴퓨터에서 동작하도록 만들어졌기 때문에 이런 메시지가 나옵니다. 당연히 그 어떤 컴퓨터라고 해도 모든 주변 장치를 가진 컴퓨터는 없으므로, 운영체제에서는 찾으려는 주변 장치가 없을 때 이러한 메시지를 내보냅니다. 또 시스템이 일시적으로 멈추는 현상이 일어날 수도 있습니다. 이러한 현상은 어떤 장치가 응답할 때까지 기다리는데, 그 장치가 없을 때 발생합니다. 이런 시간이 너무 오래 걸린다고 생각한다면, 나중에 직접 설정한 커널을 사용할 수 있습니다. (8.5절 참고.)

5.3.5 버그 보고

If you get through the initial boot phase but cannot complete the install, the bug reporter menu choice may be helpful. It copies system error logs and configuration information to a user-supplied floppy. This information may provide clues as to what went wrong and how to fix it. If you are submitting a bug report you may want to attach this information to the bug report.

그 외의 설치 메시지는 설치할 때 `/var/log/`에 들어 있고, 설치된 시스템으로 부팅한 다음에는 `/var/log/debian-installer/`에 들어 있습니다.

5.3.6 설치 보고 제출

If you still have problems, please submit an installation report. We also encourage installation reports to be sent even if the installation is successful, so that we can get as much information as possible on the largest number of hardware configurations.

Please use this template when filling out installation reports, and file the report as a bug report against the `installation-reports` pseudo package, by sending it to `<submit@bugs.debian.org>`.

Package: installation-reports

Boot method: <How did you boot the installer? CD? floppy? network?>

Image version: <Fill in date and from where you got the image>

Date: <Date and time of the install>

Machine: <Description of machine (eg, IBM Thinkpad R32)>

Processor:

Memory:

Partitions: <df -Tl will do; the raw partition table is preferred>

Output of lspci and lspci -n:

Base System Installation Checklist:

[O] = OK, [E] = Error (please elaborate below), [] = didn't try it

Initial boot worked: []

Configure network HW: []

Config network: []

Detect CD: []

Load installer modules: []

Detect hard drives: []

Partition hard drives: []

Create file systems: []

Mount partitions: []

Install base system: []

Install boot loader: []

Reboot: []

Comments/Problems:

<Description of the install, in prose, and any thoughts, comments
and ideas you had during the initial install.>

In the bug report, describe what the problem is, including the last visible kernel messages in the event of a kernel hang. Describe the steps that you did which brought the system into the problem state.

Chapter 6

데비안 설치 프로그램 사용하기

6.1 설치 프로그램이 동작하는 방식

데비안 설치 프로그램은 여러 가지 용도의 여러 구성 요소로 구성되어 있고, 각 구성 요소마다 설치 작업을 수행합니다. 각 구성 요소는 해당 작업을 수행하면서 그 작업에 필요한 정보를 사용자에게 물어봅니다. 이 질문들에는 우선순위가 부여되어 있고, 설치 프로그램이 맨 처음 시작할 때 물어볼 질문의 우선순위를 먼저 물어봅니다.

기본 설치를 하면, 꼭 필요한 (우선순위가 높은) 질문만 물어봅니다. 그래서 사용자가 거의 관여하지 않고 아주 자동화된 설치 과정이 가능합니다. 구성 요소는 순서대로 자동 실행합니다. 어떤 설치 방법을 사용하는 지 및 하드웨어가 무엇인지에 따라 어떤 구성 요소를 실행하는 지가 다릅니다. 설치 프로그램에서 물어보지 않는 질문은 기본 값을 사용합니다.

어떤 문제가 발생하면 오류 화면이 나타납니다. 그리고 경우에 따라 설치 메뉴가 나타날 수도 있고, 메뉴에서 그 문제를 피해 다른 작업을 선택할 수도 있습니다. 아무런 문제가 없으면 설치 메뉴를 볼 수 없고, 각 구성 요소에 해당하는 질문에 차례대로 답을 하기만 하면 됩니다. 심각한 오류를 알리는 우선 순위는 '중요'이기 때문에 심각한 오류가 발생하면 반드시 오류 화면이 나타납니다.

설치 프로그램에서 사용하는 기본값 중에 몇 개는 **debian-installer**가 시작할 때 넘기는 부팅 파라미터에 따라 달라집니다. 예를 들어 강제로 고정 네트워크 설정을 하려면 (기본적으로 DHCP를 사용할 수 있으면 DHCP를 사용합니다) `netcfg/disable_dhcp=true` 부팅 파라미터를 추가하면 됩니다. 사용할 수 있는 옵션에 대해서는 [5.2.1절](#) 부분을 참고하십시오.

파워유저라면 메뉴 방식 인터페이스가 더 편할 수도 있습니다. 메뉴 방식에서는 각 단계를 자동으로 진행하지 않고 사용자 입력에 따라 단계를 조정합니다. 설치 프로그램을 수동 메뉴 방식으로 사용하려면, `debconf/priority=medium` 파라미터를 사용하십시오.

커널 모듈을 설치하면서 파라미터를 넘겨야 하는 하드웨어에서는, “전문가” 모드로 설치 프로그램을 시작합니다. 설치 프로그램이 시작할 때 `expert` 명령을 사용하거나, `debconf/priority=low` 부팅 파라미터를 사용하면 됩니다. 전문가 모드는 **debian-installer**의 모든 부분을 마음대로 조정할 수 있습니다.

설치 프로그램의 화면은 문자 단위 화면입니다. (더 널리 쓰이는 그래픽 인터페이스가 아닙니다.) 이

환경에서는 마우스가 동작하지 않습니다. 여러가지 대화 상자에서 왔다갔다할 때 쓰는 키를 설명합니다. **Tab** 혹은 **오른쪽 화살표** 키를 누르면 화면에 나오는 단추들과 선택사항들 중에서 “앞으로” 움직이고, **Shift-Tab** 혹은 **왼쪽 화살표** 키는 “뒤로” 움직입니다. **위쪽** 및 **아래쪽 화살표** 키를 누르면 스크롤 목록에서 선택할 항목을 움직이고, 스크롤 목록을 스크롤합니다. 또 긴 목록에서는, 글자를 하나 누르면 그 글자로 시작하는 항목이 있는 부분으로 직접 이동합니다. 또 **Pg-Up** 및 **Pg-Down** 키로 목록을 스크롤합니다. **스페이스바**를 누르면 확인란 따위의 항목을 토글합니다. 선택한 항목으로 들어가려면 **Enter**를 누릅니다.

오류 메시지는 세 번째 콘솔에서 나옵니다. 세 번째 콘솔은 **왼쪽 Alt-F3**을 누르면 (왼쪽 **Alt**를 누른 상태에서 **F3** 평선 키) 볼 수 있습니다. 설치 메인 화면으로 돌아오려면 **왼쪽 Alt-F1**을 누릅니다.

이 메시지는 `/var/log/messages` 파일에도 들어 있습니다. 설치한 후에는 이 로그는 새로 설치한 시스템의 `/var/log/debian-installer/messages` 파일로 복사됩니다. 그 밖의 설치할 때 나오는 메시지는 `/var/log/` 안에 들어 있고, 컴퓨터가 새로 설치한 시스템으로 부팅하면 `/var/log/debian-installer/` 안에 들어갑니다.

6.2 구성 요소 소개

다음은 설치 프로그램의 구성 요소와 각 구성 요소가 하는 일에 대한 간단한 설명입니다. 특정 구성 요소를 사용하는 방법에 대해 더 자세한 정보는 [6.3절](#)에 있습니다.

main-menu 설치 프로그램이 동작하는 중에 구성 요소의 목록을 표시하고, 구성 요소를 선택하면 그 구성 요소를 시작합니다. 메인 메뉴의 질문은 우선순위가 중간이기 때문에, 우선 순위를 높음이나 중요로 해 놓으면 (기본값은 높음) 이 메뉴를 볼 수 없습니다. 하지만 오류가 발생해서 사용자가 뭔가 작업을 해야 한다면, 사용자가 이 문제를 해결할 수 있도록 우선순위가 일시적으로 낮아지고, 이 경우 메뉴가 나타날 수도 있습니다.

“뒤로” 단추를 계속해서 눌러 현재 실행중인 구성 요소를 나가게 되면 메인 메뉴로 갑니다.

languagechooser 언어 목록을 표시합니다. 언어를 하나 선택하면 설치 프로그램에서는 그 언어로 메시지를 표시합니다. 단 그 언어로 번역이 되지 않았다면, 영어 메시지를 표시합니다.

countrychooser 국가 목록을 표시합니다. 자기가 살고 있는 국가를 선택합니다.

kbd-chooser 키보드 목록을 표시합니다. 여기에서 자기 키보드에 해당하는 모델을 선택합니다.

hw-detect 시스템의 하드웨어 대부분을 자동으로 검색합니다. 네트워크 카드, 디스크 드라이브, PCMCIA 등입니다.

cdrom-detect 데비안 설치 CD를 찾아서 마운트합니다.

netcfg 인터넷을 통해 통신할 수 있도록 컴퓨터 네트워크 연결을 설정합니다.

iso-scan ISO 파일 시스템을 찾습니다. CD-ROM일 수도 있고, 하드 드라이브에 들어 있을 수도 있습니다.

choose-mirror 데비안 아카이브 미러 목록을 표시합니다. 설치할 꾸러미가 들어 있는 위치를 선택합니다.

cdrom-checker CD-ROM이 올바른지 확인합니다. 설치 CD-ROM이 망가졌는지 여부를 확인합니다.

lowmem lowmem은 메모리가 작은 시스템을 검사하고, 여러가지 방법으로 **debian-installer**에서 필요없는 부분을 메모리에서 없앱니다. (그 대신 일부 기능이 없어집니다.)

anna Anna's Not Nearly APT. 선택한 미러 사이트 혹은 CD에서 가져온 꾸러미를 설치합니다.

partman 시스템에 달린 디스크를 파티션하고, 파티션에 파일 시스템을 만들고, 파일 시스템을 마운트 위치에 마운트합니다. 완전 자동 모드 혹은 LVM 지원 기능과 같은 재미있는 기능도 들어 있습니다. 데비안에서 기본 파티션 도구입니다.

autopartkit 사용자가 정하는 방식에 따라 디스크를 자동으로 파티션합니다.

partitioner 여기서 시스템에 달린 디스크를 파티션합니다. 해당 컴퓨터 아키텍처에 적합한 파티션 프로그램을 이용합니다.

partconf 파티션 목록을 표시하고, 사용자 명령에 따라 선택한 파티션에 파일 시스템을 만듭니다.

lvmcfg 여기서 *LVM*(Logical Volume Manager)을 설정합니다.

mdcfg 여기서 소프트웨어 *RAID*(Redundant Array of Inexpensive Disks)를 설정합니다. 이 소프트웨어 RAID는 최근 마더보드에 들어 있는 싸구려 IDE RAID (가짜 하드웨어 RAID) 컨트롤러보다는 대체로 우수합니다.

base-installer 다시 시작했을 때 리눅스가 동작하는 데 필요한 가장 기본적인 꾸러미를 설치합니다.

os-prober 컴퓨터에 지금 설치되어 있는 운영 체제를 찾아서 그 정보를 bootloader-installer에 넘깁니다. 그러면 bootloader-installer에서는 이 운영 체제를 부트 로더 시작 메뉴에 추가할 수 있습니다. 이렇게 하면 부팅할 때 어떤 운영 체제를 시작할 지 쉽게 선택할 수 있습니다.

bootloader-installer 하드 디스크에 부트 로더 프로그램을 설치합니다. 그래야 플로피나 CD-ROM 을 따로 사용하지 않고 컴퓨터에서 리눅스를 시작할 수 있습니다. 컴퓨터가 부팅할 때 많은 부트 로더에서는 여러가지 운영체제중에서 하나를 선택해서 시작할 수 있도록 되어 있습니다.

base-config 사용자 선택에 따라 베이스 시스템을 설정하는 대화 상자를 표시합니다. 컴퓨터를 다시 시작한 다음에 이 대화 상자가 나타납니다. **base-config**는 새로 설치한 데비안 시스템에서 돌아가는 '제1호 프로그램'입니다.

shell 사용자가 메뉴에서 셸을 실행하거나, 두번째 콘솔에서 셸을 실행합니다.

bugreporter 문제가 발생했을 때 관련 정보를 플로피 디스크에 기록합니다. 나중에 설치 프로그램의 소프트웨어 문제를 데비안 개발자에게 정확하게 알리는 데 이 기록을 이용합니다.

6.3 각 구성 요소 사용하기

여기서는 설치 프로그램의 각 구성 요소를 자세히 설명합니다. 이 구성 요소는 사용자 입장에서 몇 단계로 분류할 수 있습니다. 여기서 설명하는 순서는 설치할 때 나타나는 순서입니다. 설치할 때 여기 있는 모듈을 모두 사용하는 건 아닙니다. 이 중에서 실제로 어떤 모듈을 사용하는 지는 설치 방법과 하드웨어에 따라 달라집니다.

6.3.1 데비안 설치 프로그램 및 하드웨어 설정 준비하기

데비안 설치 프로그램이 부팅하고 첫번째 화면이 나왔다고 가정합니다. 이 시점에서는 아직 **debian-installer**의 기능은 매우 제한적입니다. 하드웨어도 모르고, 선호하는 언어도 모르고, 어떤 작업을 해야 할 지도 모릅니다. 걱정하지 마십시오. **debian-installer**는 아주 똑똑하기 때문에, 하드웨어를 자동으로 검색해서, 필요한 구성 요소를 찾아낸 다음에, 다양한 기능을 갖춘 설치 시스템으로 다시 태어납니다. 하지만 자동으로 알아낼 수 없는 일부 정보의 경우는 (선호하는 언어 선택, 키보드 배치, 사용할 네트워크 미리 따위) 여러분이 직접 **debian-installer**에 관련 정보를 입력해야 합니다.

이 단계에서 **debian-installer**는 여러 번의 하드웨어 검색을 합니다. 첫번째는 설치 프로그램의 구성 요소를 읽어들이는 데 필요한 하드웨어를 (예를 들어 CD-ROM이나 네트워크 카드) 검색합니다. 첫번째로 실행할 때는 아직 사용할 수 없는 드라이버가 있기 때문에, 나중 단계에서 하드웨어 검색을 다시 합니다.

6.3.1.1 사용 가능 메모리 검사

debian-installer 가 맨 처음에 하는 일 중의 하나는, 메모리를 검사하는 것입니다. 메모리가 부족할 경우, 이 구성 요소에서는 설치 과정에 약간 수정을 가해서 시스템에서 Debian GNU/Linux을 설치할 수 있게 합니다.

적은 메모리에서 설치할 때는, 모든 구성 요소를 사용할 수 없습니다. 그 부족한 점 중의 하나는 설치에 사용할 언어를 마음대로 선택할 수 없다는 점입니다.

6.3.1.2 언어 선택

설치 첫 단계로, 어떤 언어로 설치를 진행할 지 선택하십시오. 언어 이름은 목록에서 (왼쪽의) 영어와 (오른쪽의) 해당 언어로 (해당 언어의 문자로) 쓰여 있습니다. 이 목록은 영어 이름 순서로 되어 있습니다.

여기서 선택하는 언어를 사용해 나머지 설치 과정을 진행합니다. 단 그 언어로 해당 대화 상자들의 번역이 있어야 됩니다. 그 언어로 된 번역문이 없으면 기본값인 영어를 사용합니다. 여기서 선택한 언어는 적당한 키보드 배치를 선택하는 데도 이용됩니다.

6.3.1.3 국가 선택

만약 6.3.1.2절에서 선택한 언어를 사용하는 국가가 여러 개 있는 경우 (영어, 중국어, 프랑스어 등), 여기에서 국가를 지정할 수 있습니다. 목록의 맨 아래에 있는 기타를 선택하면, 모든 국가의 목록이 대륙별로 구분되어 나타납니다.

여기서 선택한 사항은 나중에 기본 시간대 및 해당 위치에 적합한 데비안 미리를 고르는 데 이용됩니다. 물론 설치 프로그램의 기본값이 마음에 들지 않으면 다른 선택을 할 수도 있습니다. 선택한 국가와 언어는 새 데비안 시스템의 로케일 설정에도 영향을 미칩니다.

6.3.1.4 키보드 선택하기

어떤 키보드는 특정 언어에서 사용하는 문자에 맞게 만들어져 있습니다. 사용하고 있는 키보드에 맞는 키보드 배치를 고르시고, 해당 키보드 배치가 여기 없으면 어느정도 비슷한 키보드 배치를 고르십시오. 시스템 설치를 모두 마치면 더 많은 종류의 키보드 배치 중에서 하나를 고를 수 있게 됩니다. (`kbdconfig`를 실행하십시오.)

반전 표시를 해당 키보드로 옮기고 **Enter**를 누르십시오. 화살표 키로 반전 표시를 움직입니다. 화살표 키는 모든 언어의 키보드에 대해 동일하므로, 키보드 설정과는 상관이 없습니다. 목록에서 '확장' 키보드는 **F1**에서 **F10** 키가 위에 붙어 있는 키보드를 말합니다.

6.3.1.5 데비안 설치 프로그램 ISO 이미지 찾기

`hd-media` 방법으로 설치하면, 데비안 설치 프로그램 ISO 이미지를 찾아서 마운트해야 하는 시점이 있습니다. 이 ISO 이미지에서 나머지 설치 파일을 가져옵니다. `iso-scan` 구성 요소가 정확히 이 일을 합니다.

먼저, `iso-scan`은 알려진 파일 시스템을 사용하는 블록 장치(파티션 등)를 마운트하고, 차례대로 `.iso`로 (혹은 `.ISO`) 끝나는 파일이름을 찾습니다. 단 첫번째에서는 맨 위 디렉토리와 바로 아래 서브 디렉토리에 있는 파일만 찾습니다. (즉 `/아무 개.iso`, `/data/아무 개.iso` 파일은 찾지만, `/data/tmp/아무 개.iso`는 찾지 않습니다.) ISO 이미지를 찾으면, `iso-scan`에서 이미지 내용을 검사해서 이미지가

올바른 데비안 ISO 이미지인지 아닌지 판단합니다. 올바른 이미지가 맞으면 끝나고, 아니면 다른 이미지를 찾습니다.

앞에서 설치 ISO 이미지를 찾는 데 실패하면, **iso-scan**에서는 계속해서 이미지를 찾을 지 여부를 물어봅니다. 그러면 맨 위의 디렉토리만 찾는 게 아니라, 모든 파일 시스템을 뒤져봅니다.

iso-scan에서 설치 프로그램 ISO 이미지를 발견하지 못했다면, 원래 운영 체제로 다시 시작해서 이미지 이름이 올바른 지 (**.iso**로 끝나는 지), **debian-installer**가 인식할 수 있는 파일 시스템에 들어 있는지, 파일이 손상되지 않았는지 (체크섬 확인) 확인하십시오. 경험 많은 유닉스 사용자라면 다시 시작하지 않고 두 번째 콘솔에서 할 수도 있습니다.

6.3.1.6 네트워크 설정하기

이 단계에 들어왔을 때 시스템에서 네트워크 장치를 두 개 이상 찾으면, 어떤 장치를 주요 네트워크 인터페이스로 사용할 지 질문을 받게 됩니다. 주요 네트워크 인터페이스란 설치할 때 사용할 인터페이스를 말합니다. 이 인터페이스 외의 인터페이스는 이 시점에서는 설정하지 않습니다. 설치가 다 끝난 다음에 장치들을 추가로 설정할 수 있습니다. **interfaces(5)** 맨 페이지를 참고하십시오.

기본값으로 **debian-installer**에서는 먼저 DHCP를 통해 컴퓨터 네트워크를 설정하려고 시도합니다. DHCP 검색이 성공하면 네트워크 설정이 모두 끝납니다. 검색이 실패하면 네트워크 케이블을 꽂지 않았거나 DHCP가 잘못되었거나 따위의 여러가지 원인 때문일 수 있습니다. 아니면 LAN에 DHCP 서버가 아예 없을 수도 있습니다. 원인이 무엇이든, 다시 시도할 지 수동으로 설정할 지 여부를 물어봅니다. DHCP 서버는 가끔 응답이 아주 느리기도 하기 때문에, 다른 문제가 없는 게 분명하면 다시 시도해 보십시오.

수동 네트워크 설정에서는 네트워크에 관한 여러가지 정보들을 차례대로 물어봅니다. **IP 주소**, **네트 마스크**, **게이트웨이**, **네임 서버 주소** 및 **호스트이름**을 물어봅니다. 또 무선 네트워크 인터페이스가 있다면, 무선 **ESSID** 및 **WEP 키**를 물어봅니다. [3.3절](#)의 답을 채워 넣으십시오.

참고

알아두면 편리할 수도 있고 아닐 수도 있는 기술적인 정보: 이 프로그램에서는 네트워크 IP 주소가 시스템의 IP 주소와 네트마스크를 비트 AND한 값이라고 가정합니다. 브로드캐스트 주소는 시스템의 IP 주소와 네트마스크의 비트 NOT한 값을 OR한 값이라고 가정합니다. 또 게이트웨이기도 임의로 추정합니다. 수동 설정에서 무슨 값을 써야 할지 잘 모르겠다면, 시스템의 추정값을 사용해 보십시오. 일단 시스템을 설치한 다음에 필요하다면 **/etc/network/interfaces** 파일을 편집해서 이 설정을 바꿀 수 있습니다. 다른 방법으로, **etherconf**를 설치해서, 여기에 나오는 단계에 따라 네트워크 설정을 할 수 있습니다.

6.3.2 파티션하기 및 마운트 위치 선택

여기에서는 마지막으로 하드웨어 검색을 한 다음에, **debian-installer**는 필요한 기능을 모든 갖추고,

사용자의 필요에 맞게 진짜 설치 작업을 할 준비를 갖추게 됩니다. 제목이 말하듯, 다음 몇 개의 구성 요소는 디스크를 파티션하고, 파일시스템을 만들고, 마운트 위치를 지정하고, 또 필요하다면 LVM이나 RAID 장치와 같은 관련 설정을 합니다.

6.3.2.1 디스크 파티션하기

디스크를 파티션할 차례입니다. 파티션하는 게 불안하거나 자세히 알고 싶으시면, **부록 B** 부분을 참고하십시오.

먼저 선택에 따라, 전체 드라이브나 드라이브의 빈 공간을 자동으로 파티션할 수 있습니다. 이 방법을 “단계에 따른” 파티션이라고 합니다. 자동 파티션을 하지 않으려면, 메뉴에서 **수동으로 파티션 테이블 편집**을 선택하십시오.

단계에 따라 파티션을 선택한 경우, 아래의 표의 방식중에서 한 가지 방식을 선택합니다. 모든 방식들은 각자 장단점이 있고, **부록 B**에서 설명합니다. 잘 모르겠으면 첫번째를 선택하십시오. 명심해 뒤야 할 것은, 단계에 따라 파티션할 때는 어느정도 최소한의 빈 공간이 필요하다는 점입니다. 최소 1GB의 공간이 없으면 (방식에 따라 이 최소 용량은 다릅니다) 단계에 따른 파티션은 실패합니다.

파티션 방식	최소 공간	만들 파티션
한 파티션에 파일 모두	600MB	/, 스왑
데스크탑 컴퓨터	500MB	/, /home, 스왑
다중 사용자 워크스테이션	1GB	/, /home, /usr, /var, /tmp, 스왑

파티션 방법을 선택하면, 그 다음 화면에서 새로 만든 파티션 테이블이 나타납니다. 여기에는 파티션을 포맷할 형식과 마운트할 위치에 대한 정보도 들어 있습니다.

파티션 목록은 다음과 같이 나타납니다:

```

IDE1 master (hda) - 6.4 GB WDC AC36400L
    #1 primary   16.4 MB   ext2      /boot
    #2 primary  551.0 MB   swap      swap
    #3 primary   5.8 GB    ntfs
    pri/log      8.2 MB    FREE SPACE

IDE1 slave (hdb) - 80.0 GB ST380021A
    #1 primary   15.9 MB   ext3
    #2 primary  996.0 MB   fat16
    #3 primary   3.9 GB    xfs       /home
    #5 logical   6.0 GB    ext3      /
    #6 logical   1.0 GB    ext3      /var
    #7 logical  498.8 MB   ext3
    #8 logical  551.5 MB   swap      swap
    #9 logical   65.8 GB   ext2
  
```

위의 예에서는 IDE 하드드라이브가 2개이고 여러 개 파티션으로 나뉘져 있습니다. 첫 번째 디스크에는 빈 공간이 있습니다. 각 파티션 줄에는 파티션 번호, 종류, 크기, 플래그, 파일시스템, 그리고 마운트 위치를 (마운트 위치가 따로 있는 경우) 표시합니다.

여기까지가 단계에 따른 파티션입니다. 자동으로 만든 파티션 테이블이 마음에 들면, 메뉴에서 **파티션 나누기를 마치고 바뀐 사항을 디스크에 쓰기**를 선택해서 새 파티션 테이블을 실제로 적용합니다 (이 절의 맨 뒤 부분에서 설명합니다). 마음에 들지 않으면, **파티션에 바뀐 사항을 취소**를 선택해서 단계에 따른 파티션을 다시 실행하거나, 자동으로 만들어 준 파티션을 아래에서 설명하는 것처럼 수동으로 고칠 수도 있습니다.

수동 파티션을 선택하면 위와 비슷한 화면을 표시합니다. 다른 점은 기존 파티션 테이블을 마운트 위치 없이 표시한다는 점입니다. 이 절의 나머지 부분에서는 파티션 테이블을 수동으로 만드는 방법과, 이 파티션을 새 데비안 시스템에서 어떻게 사용할 지에 대해 다룹니다.

파티션도 없고 빈 공간도 없는 새 디스크라면, 새로운 파티션 테이블을 만들게 됩니다. (그래야 새 파티션을 만들 수 있습니다.) 그 다음에 “빈 공간”이라는 줄이 해당 디스크 이름 아래에 나타납니다.

빈 공간을 선택하면, 새 파티션을 만들 지 물어봅니다. 크기, 종류 (주 파티션 아니면 논리 파티션), 위치에 (빈 공간에서 처음 아니면 끝) 대해 답해야 합니다. 그러면 새 파티션에 대한 자세한 정보가 나타납니다. 마운트 위치, 마운트 옵션, 부팅 가능 플래그, 용도 따위의 옵션이 있습니다. 기본값이 마음에 들지 않는다면, 원하는 대로 바꾸십시오. 예를 들어 **용도**: 옵션을 선택해서 이 파티션의 파일 시스템을 선택할 수 있습니다. (파일시스템에는 스왑 파티션, 소프트웨어 RAID, LVM, 아니면 아예 사용하지 않는 것까지도 포함됩니다.) 또 다른 멋진 기능은 기존 파티션의 데이터를 이 파티션으로 복사하는 기능입니다. 새 파티션이 마음에 들면 **파티션 준비를 마쳤습니다**를 선택하고 **partman**의 주 화면으로 돌아갑니다.

파티션에서 뭔가 바꾸려고 한다면, 해당 파티션을 선택하십시오. 그러면 파티션 설정 메뉴가 나타납니다. 새 파티션을 만들 때와 같은 화면이기 때문에, 여기서도 마찬가지로 옵션을 바꿉니다. 첫 눈에 잘 이해가 되지 않을 만한 부분은, 파티션의 크기 항목을 이용해서 파티션의 크기를 바꿀 수 있다는 점입니다. 이 기능이 동작하는 파티션은 fat16, fat32, ext2, ext3, 및 스왑입니다. 이 메뉴에서 파티션을 지워 버릴 수도 있습니다.

최소한 파티션을 두 개 만들도록 하십시오. 한 개는 루트 파일시스템이고 (/에 마운트합니다), 다른 하나는 스왑입니다. 루트 파일 시스템을 마운트하지 않으면, 그 문제를 바로잡기 전에는 다음으로 진행하지 않습니다.

partman의 기능은 설치 프로그램 모듈에 따라 확장되지만, 시스템의 아키텍처에 따라 다릅니다. 모든 기능들을 볼 수 없다면, 필요한 모듈을 모두 읽어들였는지 확인하십시오. (예를 들어, **partman-ext3**, **partman-xfs**, 아니면 **partman-lvm**)

파티션이 마음에 들면, 파티션 메뉴에서 **파티션 나누기를 마치고 바뀐 사항을 디스크에 쓰기**를 선택하십시오. 디스크에 바뀐 점에 대한 요약이 나타나고 이대로 파일 시스템을 만들지 확인합니다.

6.3.2.2 논리 볼륨 관리자 (LVM) 설정하기

시스템 관리자나 “고급” 사용자 입장에서 컴퓨터를 사용한다면, 어떤 디스크 파티션이 (보통 가장 중

요한 파티션) 공간이 부족하고 다른 파티션은 공간이 남아서 데이터를 옮기고 심볼릭 링크를 걸고 하는 등의 작업으로 상황을 해결해야 했던 경험이 있을 것입니다.

이러한 상황을 피하려면, 논리 볼륨 관리자(LVM, Logical Volume Manager)를 사용할 수 있습니다. 간단히 말해 LVM을 사용하면 여러 파티션들을 (물리 볼륨) 하나의 가상 디스크로 (볼륨 그룹) 합칠 수 있고, 그걸 다시 가상 파티션으로 (논리 볼륨) 나눌 수 있습니다. 중요한 점은 논리 볼륨이 (그리고 그 밑에 있는 볼륨 그룹이) 여러 개의 물리 파티션에 걸쳐 있을 수 있다는 점입니다.

기존의 160GB /home 파티션에 용량이 더 필요하다면, 300GB 디스크를 컴퓨터에 붙이고, 이 디스크를 현재 볼륨 그룹에 포함시키고, /home이 들어 있는 논리 볼륨 크기를 늘리면 됩니다. 그러면 파티션은 460GB가 되어 남는 공간이 더 생깁니다. 물론 이 예는 너무 간단하게 설명한 것입니다. 아직 읽지 않으셨다면 LVM HOWTO <<http://www.tldp.org/HOWTO/LVM-HOWTO.html>>를 참고하십시오.

debian-installer의 LVM 설정은 아주 간단합니다. 먼저 LVM의 물리 볼륨으로 사용할 파티션을 표시합니다. (파티션 설정의 partman에서 합니다. 거기서 용도: → LVM의 물리 볼륨을 선택합니다.) 그리고 lvmcfg 모듈을 시작하고 (partman에서 직접 하던가, debian-installer의 메인메뉴에서 합니다) 볼륨 그룹 (VG) 고치기 메뉴에서 물리 볼륨을 볼륨 그룹에 합칩니다. 그 다음에 논리 볼륨 (LV) 고치기 메뉴에서 볼륨 그룹 위에 논리 볼륨을 만들어야 합니다.

lvmcfg에서 partman으로 돌아간 다음, 보통 파티션과 마찬가지로 방금 만든 논리 볼륨이 나타납니다. (또 보통 파티션과 마찬가지로 방법으로 이용하면 됩니다.)

6.3.2.3 멀티디스크 장치 설정하기 (소프트웨어 RAID)

컴퓨터에 하드 드라이브가 여러 개 있는 경우¹, mdcfg 명령으로 드라이브의 성능을 향상시키거나, 데이터 안정성을 높일 수 있습니다. 이렇게 하는 걸 멀티디스크 장치라고 합니다. (더 자주 쓰이는 다른 말로 RAID라고 합니다.)

기본적으로 멀티디스크는 여러 디스크에 있는 여러 개의 파티션으로 하나의 논리 장치를 만드는 걸 말합니다. 그러면 이 논리 장치는 일반 파티션처럼 사용할 수 있습니다. (예를 들어 partman에서 포맷하고 마운트 위치를 지정하고 따위를 할 수 있습니다.)

어떤 종류의 멀티디스크 장치를 만드냐에 따라 어떤 이점이 있는지 달라집니다. 현재 지원하는 종류는:

RAID0 주로 성능 향상이 목적입니다. RAID0는 들어오는 데이터를 스트라이프(stripe)로 나눠서 배열의 각 디스크에 똑같이 분산시킵니다. 이렇게 하면 읽기/쓰기 작업의 속도가 빨라집니다. 하지만 디스크중에 하나라도 망가지면, 모든 데이터를 잃게 됩니다. (망가지지 않은 디스크에 일부 데이터가 남아 있겠지만, 또 다른 부분이 망가진 디스크에 있었기 때문입니다.)

RAID0은 비디오를 편집하는 파티션에 많이 사용합니다.

RAID1 안정성을 최우선으로 할 때 적합합니다. RAID1은 여러 개의 (보통 두개) 같은 크기의 파티션으로 구성되어 있고, 여기서 각 파티션은 정확히 같은 데이터를 담고 있습니다. 이게 근본적으로

¹물론 물리적으로 한 개의 드라이브에 있는 여러개의 파티션에서 멀티디스크 장치를 만들 수도 있겠지만, 그렇게 해 봤자 좋은 점이 전혀 없습니다.

3가지 의미가 있습니다. 첫째로, 하나의 디스크가 망가지더라도, 나머지 디스크에 데이터가 미리 되어 있습니다. 둘째로, 디스크의 전체 용량보다 작은 용량만 사용할 수 있습니다. (좀더 정확히 말해, RAID에서 가장 작은 파티션입니다.) 세번째로, 파일 읽기는 로드 밸런싱으로 서버의 성능을 향상시켜, 파일서버와 같이 디스크 읽기가 쓰기보다 많은 경우 부담이 줄어듭니다.

RAID5 속도, 안정성, 데이터 중복을 적당히 조화시킨 것입니다. RAID5는 들어오는 모든 데이터를 스트라이프로 나누고 각각을 하나의 디스크가 아니라 (즉 RAID0와는 달리) 모두에게 분배합니다. RAID0와는 다르게 RAID5는 디스크에 쓸 정보의 패리티 정보를 계산합니다. 패리티 디스크는 고정되어 있지 않고 (고정되어 있으면 RAID4라고 합니다) 정기적으로 바뀝니다. 디스크중에 하나가 망가지면, 없어진 부분을 나머지 데이터와 패리티를 이용해서 계산해 냅니다. RAID5는 최소한 3개의 파티션이 있어야 합니다. 배열에 디스크를 하나 더 사용해서 망가진 디스크를 대체하도록 만들 수도 있습니다.

이렇게 RAID5는 RAID1과 비슷한 정도의 안정성을 가지면서, 데이터를 덜 중복합니다. 한편 패리티 계산때문에 RAID0보다는 쓰는 속도가 느립니다.

요약하면:

종류	장치 최소 개수	예비 장치	디스크가 망가져도 버티는지?	사용 가능 공간
RAID0	2	아니오	아니오	RAID에서 가장 작은 파티션의 크기 x 장치
RAID1	2	옵션	예	RAID에서 가장 작은 파티션의 크기
RAID5	3	옵션	예	가장 작은 파티션의 크기 x (RAID의 장치 개수)

소프트웨어 RAID의 모든 것을 알고 싶으시면, Software RAID HOWTO <<http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>>를 읽어 보십시오.

멀티디스크 장치를 만드려면, 구성할 파티션을 RAID에 사용한다고 표시해야 합니다. (파티션 설정 메뉴의 **partman**에서 **용도:** → **RAID의 물리적 볼륨**을 선택하십시오.)

주의

설치 프로그램에서 멀티디스크 지원은 비교적 최근에 추가되었습니다. 어떤 특정 RAID 레벨을 특정 부트로더와 함께 루트 (/) 파일 시스템으로 멀티디스크를 사용한 경우에 문제가 발생할 수도 있습니다. 경험 많은 사용자들의 경우, 셸에서 수동으로 설정하거나 수동으로 설치하는 방법으로 이러한 문제를 피해갈 수 있습니다.

그 다음 **partman** 메뉴에서 **소프트웨어 RAID 설정**을 선택합니다. **mdcfg**의 첫 번째 화면에서 **멀티 디스크 장치 만들기**를 선택하기만 하면 지원하는 멀티디스크 장치의 종류 목록이 나옵니다. 거기에서 하나를 (예를 들어 RAID1) 고르십시오. 그 다음은 여기서 어떤 종류의 멀티디스크를 선택했냐에 따라 달라집니다.

- RAID0는 간단합니다. RAID 파티션의 목록이 나타나고 거기에서 멀티디스크를 구성할 파티션을

선택하기만 하면 됩니다.

- RAID1은 약간 더 까다롭습니다. 먼저 멀티디스크를 구성할 활성 장치의 개수 및 예비 장치의 개수를 입력합니다. 그리고 RAID 파티션 목록에서 무엇을 활성 파티션과 예비 파티션으로 할지 결정합니다. 여기서 선택한 파티션 개수는 맨 처음에 입력한 개수와 일치해야 합니다. 걱정할 필요는 없습니다. 실수로 파티션 개수가 틀렸다고 해도, 개수가 맞을 때까지는 **debian-installer**가 다음으로 진행하지 않습니다.
- RAID5는 RAID1과 비슷한 설정을 하지만, 최소한 3개의 활성 파티션을 사용해야 한다는 점이 다릅니다.

여러가지 종류의 멀티디스크를 동시에 사용하는 것도 물론 가능합니다. 예를 들어 3개의 200 GB 하드 드라이브를 멀티디스크에 사용할 때, 각 디스크에 100 GB 파티션이 두개씩 있다고 할 때, 각 3개 디스크의 첫번째 파티션들을 RAID0로 묶고 (빠른 300GB 비디오 편집 파티션) 나머지 3개 파티션을 (2개 활성, 1개 예비) RAID1으로 (/home에 사용할 안정성 높은 100GB 파티션) 사용할 수 있습니다.

원하는 대로 멀티디스크 장치를 만든 다음에, **mdeconf**에서 **마치기**를 선택하고 **partman**으로 돌아가 새로 만든 멀티디스크 장치에 파티션을 만들고 마찬가지로 마운트 위치와 같은 속성을 부여할 수 있습니다.

6.3.3 베이스 시스템 설치하기

이 단계는 문제가 발생할 가능성이 가장 낮지만, 설치할 때 가장 많은 시간을 소모합니다. 여기서는 베이스 시스템 전체를 내려받고, 확인한 다음, 압축을 풉니다. 컴퓨터가 느리거나 네트워크 연결이 느리다면 시간이 좀 걸릴 수도 있습니다.

6.3.3.1 베이스 시스템 설치

베이스 설치 중에 패키지를 풀고 설정하면서 나오는 메시지는 **tty3**에서 표시합니다. 이 터미널은 왼쪽 **Alt-F3**을 누르면 이용할 수 있습니다. 설치 프로그램 화면으로 돌아가려면 왼쪽 **Alt-F1**을 누르십시오.

설치를 시리얼 콘솔을 통해서 하는 경우에는, 베이스 설치할 때 압축 풀기/설정 메시지는 **/var/log/messages** 파일에 저장합니다.

설치 과정의 하나로 리눅스 커널을 설치합니다. 기본 우선순위로에서는 하드웨어에 가장 맞는 커널을 하나 선택합니다. 우선순위가 낮은 모드에서는, 사용 가능한 여러가지 커널 중에서 하나를 선택할 수 있습니다.

6.3.4 시스템을 부팅 가능하게 만들기

디스크 없는 워크스테이션에 설치하는 경우에는, 로컬 디스크에서 부팅하는 건 물론 전혀 가능한 방법이 아니므로 이 단계는 건너 뛴니다.

한 컴퓨터에서 여러 개의 운영 체제를 부팅하는 건 아직까지도 매우 복잡한 기술입니다. 이 문서에서는 모든 부팅 관리자에 대해 다루려고 하지도 않습니다. 부팅 관리자는 아키텍처마다 다르고 서버 아키텍처마다도 다릅니다. 자세한 정보는 부팅 관리자의 문서를 참고하십시오.

6.3.4.1 다른 운영 체제 찾기

설치 프로그램은 부트로더를 설치하기 전에 컴퓨터에 설치되어 있는 다른 운영체제가 있는 지 찾아 봅니다. 부트로더가 지원하는 운영 체제가 있으면 부트로더를 설치하는 단계에서 알려 주고, 데비안 부팅 외에 추가로 이 다른 운영체제를 부팅할 수 있도록 설정할 것입니다.

한 컴퓨터에서 여러 개의 운영 체제를 부팅하는 건 아직도 매우 복잡한 기술입니다. 다른 운영체제를 자동으로 찾아내고 부트로더를 설정하는 건 아키텍처마다 다르고, 서버 아키텍처마다도 다릅니다. 동작하지 않으면 부트로더의 문서에서 더 자세한 사항을 찾아보십시오.

참고

다른 운영체제를 찾을 때, 파티션이 마운트되어 있으면 그 안의 운영체제를 찾지 못할 수도 있습니다. 이 현상은 **partman**에서 다른 운영체제가 들어 있는 파티션에 마운트 위치를 지정했을 경우 (예를 들어 /win) 발생합니다. 아니면 콘솔에서 파티션을 수동으로 마운트했을 경우에도 발생합니다.

6.3.4.2 하드 디스크에 GRUB 부트로더 설치

i386의 주요 부트로더는 “GRUB”입니다. GRUB은 유연하고 안정적인 부트로더이고, 초보자와 오래된 사용자 모두에게 좋은 기본 부트로더입니다.

기본적으로 GRUB은 마스터 부트 레코드에 (MBR) 설치합니다. 그래야 부팅 과정을 완전히 조종할 수 있습니다. 원하신다면 다른 위치에 설치할 수도 있습니다. 완전한 정보를 보려면 GRUB 매뉴얼을 참고하십시오.

GRUB을 아예 설치하지 않으려면, ”뒤로” 단추를 사용해서 메인 메뉴로 이동한 다음에, 쓰고 싶은 부트로더를 선택하십시오.

6.3.4.3 하드 디스크에 LILO 부트로더 설치

두번째 i386 부트로더는 “LILO”입니다. 오래되고 복잡한 프로그램으로 DOS, Windows, OS/2 부트 관리자 등 아주 많은 기능이 있습니다. 특별히 필요한 점이 있으면 /usr/share/doc/lilo/ 디렉토리에 있는 안내 문서를 잘 읽어보십시오. 또 LILO mini-HOWTO <<http://www.tldp.org/HOWTO/LILO.html>>도 참고하십시오.

참고

지금은 LILO를 설치해도 체인로드(chainload)가 가능한 운영체제의 메뉴 항목만 만듭니다. 즉 GNU/리눅스 및 GNU/허드와 같은 운영체제는 메뉴 항목을 수동으로 추가해야 할 수도 있습니다.

debian-installer에서는 LILO를 설치할 위치로 세 가지중의 하나를 선택합니다.

마스터 부트 레코드 (Master Boot Record, MBR) 이 방법으로 LILO가 부팅 과정을 완전히 책임지게 됩니다.

새 데비안 파티션 다른 부팅 관리자를 사용하려면 이걸 선택하십시오. LILO는 새 데비안 파티션의 맨 앞쪽에 설치되어 보조 부트로더로서 동작합니다.

기타 MBR이 아닌 다른 위치에 LILO를 설치하려는 고급 사용자에게 유용한 정보입니다. 이 경우에 LILO를 설치하려는 위치를 물어봅니다. DEVFS 스타일의 이름을 사용할 수 있습니다. (예를 들어 전통적인 방식의 이름으로 /dev/hda 혹은 /dev/sda라고 쓸 수도 있고, DEVFS 방식으로 /dev/ide, /dev/scsi 및 /dev/discs라고 쓸 수도 있습니다.)

이 단계를 거친 다음에 Windows 9x로 (아니면 DOS로) 부팅할 수 없게 된다면, Windows 9x (MS-DOS) 부팅 디스크를 이용해서 fdisk /mbr 명령을 실행해서 MS-DOS 마스터 부트 레코드를 다시 설치해야 부팅할 수 있을 것입니다. 하지만 이 명령을 실행한 다음에 데비안으로 들어가려면 뭔가 다른 방법이 필요합니다! 더 자세한 정보는 [8.3절](#) 부분을 참고하십시오.

6.3.4.4 부트로더 없이 계속

이 옵션은 부트로더를 설치하지 않았지만 설치를 마칠 때 사용할 수 있습니다. 이렇게 하는 경우는 아마도 해당 아키텍처나 서버 아키텍처에 부트로더가 없거나, 부트로더가 필요없는 경우일 (예를 들어 기존 부트로더를 사용) 것입니다.

부트로더를 수동으로 설정하려면, /target/boot에 설치한 커널의 이름을 확인해야 합니다. 또 이 디렉토리에 initrd가 있는 지 확인하고, 있으면 아마도 부트로더가 그 initrd를 사용하도록 해야 할 것입니다. 그 외에 필요한 정보는 / 파일 시스템으로 사용하려는 디스크 및 파티션을 알아야 하고, /boot가 별도 파티션이면 /boot 파일시스템의 디스크 및 파티션도 알아야 합니다.

6.3.5 첫 단계 마치기

다음은 새 데비안 시스템을 다시 시작하기 전에 할 마지막 작업입니다. 대부분은 debian-installer를 끝내고 정리하는 작업들입니다.

6.3.5.1 설치 마치기 및 다시 시작하기

데비안 설치 과정의 첫번째에서 제일 마지막 단계입니다. 설치 프로그램을 부팅할 때 사용했던 부팅 미디어를 (CD, 플로피 등) 빼라는 말이 나옵니다. 여기서 남은 작업을 다 마친 다음에, 새로 설치한 데비안 시스템으로 다시 시작합니다.

6.3.6 기타

여기에 목록이 나와 있는 구성 요소는 일반적인 설치 과정과는 상관이 없습니다. 하지만 백그라운드에서 기다리면서 잘못된 부분이 있을 때 도움이 됩니다.

6.3.6.1 설치 로그 저장

설치가 성공하면, 설치할 때 만든 로그 파일은 새 데비안 시스템의 `/var/log/debian-installer/` 파일에 자동으로 저장합니다.

메인 메뉴에서 **디버그 로그 저장**을 선택하면 로그 파일을 플로피 디스크에 저장합니다. 설치할 때 심각한 문제가 발생할 경우에 다른 시스템에서 로그를 분석한 다음, 이 로그를 설치 보고서에 첨부할 수 있습니다.

6.3.6.2 셸 사용하기 및 로그 보기

메뉴에 **셸 실행** 항목이 있습니다. 셸을 사용해야 하는데 메뉴가 없으면, 왼쪽 **Alt-F2**를 눌러 (Mac 키보드에서는 **Option-F2**) 두 번째 가상 터미널로 전환합니다. 스페이스바 왼쪽에 있는 **Alt** 키와 **F2** 평선 키를 동시에 누르는 걸 말합니다. 이 가상 터미널은 완전히 별도의 창으로 **ash**이라고 하는 본셸 호환 셸이 동작합니다.

이 시점에서는 램디스크에서 부팅했기때문에, 제한적인 유닉스 유틸리티만 사용할 수 있습니다. 어떤 프로그램이 있는지는 `ls /bin /sbin /usr/bin /usr/sbin` 명령 및 `help`를 입력해서 알 수 있습니다. 텍스트 에디터는 **nano**입니다. 이 셸에는 자동완성이나 히스토리같은 훌륭한 기능도 좀 있습니다.

어떤 작업을 수행할 때 메뉴로 할 수 있으면 메뉴로 하십시오. 셸과 명령어들은 무언가 잘못되었을 경우를 대비한 것 뿐입니다. 특히 스왑 파티션을 활성화할 때는 셸이 아니라 꼭 메뉴를 사용해야 합니다. 셸에서 스왑 파티션을 활성화해도 메뉴 소프트웨어에서는 알지 못하기 때문입니다. 왼쪽 **Alt-F1**을 누르면 메뉴로 돌아가고, 메뉴에서 셸을 실행한 경우에는 `exit` 명령을 실행하면 메뉴로 돌아갑니다.

6.3.6.3 네트워크를 통해 설치

재미있는 구성 요소 중의 하나로 `network-console`이 있습니다. 설치 작업의 많은 부분을 네트워크 SSH를 통해 수행하게 되어 있습니다. 네트워크를 사용해야 하기 때문에 최소한 네트워크 설정하기까지의 맨 처음 설치 작업은 콘솔에서 해야 합니다. (이 부분은 [4.7절](#)에 따라 자동화할 수 있습니다.)

이 구성 요소는 주 설치 메뉴에는 기본으로 읽어들이지 않기 때문에, 이 구성 요소를 읽어들이라고 지정해야 합니다. CD에서 설치하는 경우에는 중간 우선 순위로 설치하거나 주 설치 메뉴가 나타나면

CD에서 설치 프로그램 구성 요소를 읽어들이기를 선택하고 **network-console: SSH**를 사용해 원격에서 설치하기 추가 구성 요소를 선택합니다. 성공적으로 읽어들이면 **SSH를 사용해 원격에서 설치하기** 메뉴 항목이 새로 생깁니다.

새로 생긴 이 항목을 선택한 다음에, 설치 시스템에 연결하는데 사용할 새 열쇠글을 입력하게 됩니다. 여기까지 하면 *installer* 사용자로 방금 입력한 열쇠글을 이용해 원격에서 로그인할 수 있는 방법을 알려주는 화면을 표시합니다. 이 화면의 또 다른 중요한 정보는 시스템의 핑거프린트입니다. 이 핑거프린트를 “시스템을 원격에서 설치할 사람에게” 안전하게 전달해야 합니다.

로컬에서 설정을 계속하려고 마음을 바꾸었다면, 언제든지 **Enter**를 눌러서 주 메뉴로 돌아갈 수 있습니다. 주 메뉴에서 다른 구성 요소를 선택하면 됩니다.

이제 네트워크의 다른 한 편으로 가서 할 일입니다. 먼저 필요한 일은, 터미널을 UTF-8 인코딩을 쓰도록 설정하는 일입니다. UTF-8 인코딩이 설치 시스템에서 사용하는 인코딩입니다. UTF-8으로 설정하지 않아도 원격 설치가 가능하지만 창의 테두리라던지, 읽을 수 없는 ASCII가 아닌 문자처럼 표시가 깨질 수도 있습니다. 설치 시스템에 연결하려면 간단히 다음과 같은 명령을 사용하면 됩니다:

```
$ ssh -l installer install_host
```

여기서 *install_host*는 설치할 컴퓨터의 이름이나 IP 주소입니다. 실제로 로그인하기 전에 원격 시스템의 핑거프린트가 표시될 것이고, 이 핑거프린트가 올바른 지 확인해야 합니다.

참고

여러 컴퓨터를 모두 설치하는 경우에 IP 주소 혹은 호스트 이름이 같은 경우, 그런 호스트는 **ssh**에서 연결을 거부합니다. 그 이유는 핑거프린트가 다르기 때문이고, 핑거프린트가 다르다는 건 스푸핑 공격의 징조입니다. 스푸핑이 아니라고 확신하는 경우, `~/.ssh/known_hosts`에서 해당 줄을 지우고 다시 연결하면 됩니다.

로그인한 다음에 최초 화면이 나오면 거기에서 **메뉴 시작**과 **셸 시작** 중의 하나를 선택할 수 있습니다. 전자의 경우는 설치 프로그램의 주 메뉴로 가게 되고, 거기에서 로컬에서와 마찬가지로 설치 작업을 계속할 수 있습니다. 후자는 셸을 실행해서 원격 시스템을 살펴보고 문제점을 수정할 수 있습니다. 설치 메뉴는 한 개의 SSH 세션만 열어야 합니다. 하지만 셸의 경우에는 여러 개를 열어도 됩니다.

주의

SSH을 통해 원격으로 설치를 시작하면, 다시 로컬 콘솔로 돌아가서 설치를 하면 안 됩니다. 그렇게 하면 새로 설치할 시스템에 있는 설정들을 망가뜨릴 수 있습니다. 설정들이 망가지면 설치가 실패하거나 새로 설치한 시스템에 여러가지 문제가 발생할 수 있습니다.

또 SSH 세션을 X 터미널에서 실행하는 경우 연결이 끊어질 수도 있으므로 창 크기를 바꾸지 말아야 합니다.

6.3.6.4 debian-installer 안에서 base-config 실행하기

베이스 시스템을 첫번째 단계의 설치 프로그램에서 (하드 드라이브에서 부팅하기 전에) 설정할 수 있습니다. *chroot* 환경에서 **base-config**를 실행하면 됩니다. 이 방법은 설치 프로그램을 테스트할 경우에만 쓸 만한 방법이므로 보통의 경우 이 방법은 사용하지 말아야 합니다.

Chapter 7

새로운 데비안 시스템으로 부팅하기

7.1 진실의 시간

자체 전원으로 당신의 시스템을 처음 부팅하는 것을 전기 기술자들은 “smoke test” 라고 부릅니다.

만일 직접 데비안으로 부팅 했지만, 시스템이 시작하지 않는 경우, 원래 설치에 사용한 부트 미디어를 사용하거나 직접 제작한 부트 플로피를 넣고 시스템을 재부팅합니다. 이 방법을 사용하면 아마도 `root=root`과 같은 부트 인자를 필요로 할 것입니다. 여기의 `root`은 `/dev/sda1`과 같은 당신 시스템의 루트 파티션입니다.

7.2 데비안 부팅 후 (기본) 설정

부팅 후 기본 시스템의 설정을 완료하도록 요구되고, 설치하기 원하는 추가적인 패키지들을 선택 해야 할 것입니다. 이 과정 동안 당신을 가이드 해줄 어플리케이션을 **base-config**이라고 부릅니다. 이 개념은 첫 번째 단계에서 **debian-installer**와 매우 유사합니다. 실제 **base-config**은 많은 특화된 컴포넌트들로 구성됩니다. 각 컴포넌트는 하나의 설정 작업을 처리하고, “hidden menu in the background” 를 가지며, 동일한 네비게이션 시스템을 사용합니다.

만일 설치가 완료된 후 어느 시점에 **base-config**을 다시 실행하길 원한다면, root 권한으로 **base-config**을 실행 합니다.

7.2.1 시간대 설정

환영 화면이 보여진 후에 시간대 설정을 해야 할 것입니다. 먼저 시스템의 하드웨어 클럭이 지역 시간 (local time) 또는 Greenwich Mean Time (GMT 또는 UTC)으로 설정되는 지를 선택합니다. 다이얼로그에 보여지는 시간은 당신이 정확한 옵션을 결정하는 데 도움을 줄 것 입니다. Dos 또는 Windows를 운영하는 시스템들도 보통 지역 시간으로 설정됩니다. 만일 dual-boot을 원한다면, GMT 대신 지역 시간을 선택하세요.

설치과정의 시작부분에서 선택된 지역에 따라 하나의 시간대 또는 여러 시간대들의 목록을 볼 것입니다. 하나의 시간대만 보여진다면 **Yes**를 선택해 확인하거나 **No**를 선택해 시간대들의 전체 목록에서

선택할 수 있습니다. 만일 시간대의 목록이 보여진다면, 목록에서 당신의 시간대를 선택하거나 전체 목록에서 다른 것을 선택할 수 있습니다.

7.2.2 사용자와 암호 설정

7.2.2.1 Root 암호 설정

root 계정은 또한 *super-user*라고도 불리웁니다. 이 계정은 당신의 시스템의 모든 보안상 보호들을 통과할 수 있습니다. *root* 계정은 시스템 관리를 수행하기 위해서만 사용되어야 하며, 가능한 한 짧은 시간 동안만 사용되어야 합니다.

만드는 어떤 암호라도 적어도 6자 이상 이고, 대문자와 소문자 특수 문자 등이 포함되는 것이 좋습니다. *root* 암호를 설정할 때는 계정 권한이 막강하기 때문에 좀 더 주의를 기울여 주세요. 사전에 나와있는 단어나 추측할 수 있는 개인 정보의 사용은 피하시길 바랍니다.

어떤 사람이 *root* 암호를 말해 달라고 한다면, 극히 조심하시길 바랍니다. 하나의 시스템의 관리자가 여러 명인 경우가 아니라면, 보통의 경우에 *root* 암호를 알려주어서는 안됩니다.

7.2.2.2 일반 사용자 만들기

시스템은 이 시점에서 일반 사용자 계정을 만들 것인지를 물어볼 것입니다. 이 계정은 당신이 주로 사용하는 개인 로그인 계정입니다. 일상적인 사용이나 개인 로그인에서 *root* 계정을 사용하면 안됩니다.

root 권한의 사용을 피하는 이유 중 하나는 *root*로는 복구하기 힘든 손상을 주기가 매우 쉽기 때문입니다. 다른 이유로는 슈퍼유저의 권한을 이용해 숨어서 당신 시스템의 보안을 침해할 수 있는 프로그램인 트로이목마 프로그램을 실행하도록 속을 수 있기 때문입니다. 유닉스 시스템 관리에 대한 좋은 책들은 이 주제에 대해 좀 더 자세하게 다룰 것입니다. 만일 처음 접하는 것이라면 이에 대한 책을 한 권 정도 읽어 보세요.

먼저 전체 이름을 입력하고, 사용자 계정으로 사용할 이름을 입력합니다. 사용자 계정으로 일반적으로 성이나 이와 유사한 어떤 것을 사용하면 충분하고, 실제로 성이 기본 값이 될 것 입니다. 마지막으로 이 계정에 대한 암호를 입력하세요.

설치 후 어느 때에 다른 계정을 만들기를 원한다면, **adduser** 명령을 사용하세요.

7.2.3 PPP 설정

설치의 첫 단계에서 네트워크를 설정하지 않았다면, 시스템의 나머지 설치를 PPP를 사용할 것인지를 물어볼 것입니다. PPP는 모뎀을 사용하여 전화 연결을 만드는 데 사용되는 프로토콜입니다. 이 단계에서 모뎀을 설정한다면, 설치 시스템은 설치의 다음 단계 동안 인터넷을 통해 추가 패키지들이나 보안 업데이트들을 다운로드 할 수 있습니다. 만일 당신의 컴퓨터에 모뎀이 없거나 설치 후에 모뎀을 설정하길 원한다면, 이 단계를 건너 뛸 수 있습니다.

PPP 연결을 설정하기 위해 당신의 인터넷 서비스 제공자(ISP)로 부터 전화번호, 사용자이름, 암호, DNS 서버 (선택적) 에 대한 정보를 필요로 합니다. 몇몇 ISP들은 리눅스 배포판을 위한 설치 안내를

제공합니다. 대부분의 리눅스 배포판들의 설정 인자 값들이나 소프트웨어들이 유사하기 때문에 그들이 데비안에 설치하는 것을 목적으로 하지 않는 다 하더라도 이 정보들을 사용할 수 있습니다.

이 단계에서 당신이 PPP를 설정하도록 선택했다면, **pppconfig**라는 프로그램을 실행할 수 있습니다. 이 프로그램은 당신의 PPP 연결을 설정하는데 도움을 줍니다. 전화 연결의 이름을 물어 볼 때 *provider*로 이름 짓는 것을 확실히 합니다.

pppconfig 프로그램이 문제가 발생하지 않는 PPP 연결을 설정하길 바랍니다. 만일 설정이 동작하지 않는다면, 아래의 자세한 설명을 보세요.

PPP 설정을 하려면 GNU/Linux에서 파일 보기와 편집의 기본을 알 필요가 있습니다. 파일을 보기 위해서 **more**를 .gz 확장자로 압축된 파일을 보기 위해서는 **zmore**를 사용해야 합니다. 예를 들어 **README.debian.gz**을 보기 위해 **zmore README.debian.gz**을 입력합니다. 기본 시스템은 **nano**라는 이름의 매우 간단하고 기능이 적은 편집기를 제공합니다. 당신은 이 후에 좀 더 기능이 많은 편집기와 뷰어인 **jed**, **nvi**, **less**, **emacs** 등을 설치하길 원할 것입니다.

/etc/ppp/peers/provider을 편집기로 열고 **/dev/modem**을 **/dev/ttyS#**으로 바꿔줍니다. 여기에서 #은 당신의 시리얼 포트의 번호를 의미합니다. 리눅스에서 시리얼 포트는 0 부터 시작합니다. 첫 번째 시리얼 포트(즉, COM1)는 리눅스에서는 **/dev/ttyS0** 입니다.

많은 ISP들은 텍스트 모드 인증 대신 로그인 과정에 PAP 또는 CHAP을 사용합니다. 어떤 ISP들은 둘 다 사용하기도 합니다. 당신의 ISP가 PAP 또는 CHAP을 사용한다면, 다른 방법을 따라야 할 것입니다. **/etc/chatscripts/provider**에서 "ATDT"로 시작하는 다이얼링 문자열들을 모두 주석처리하고, **/etc/ppp/peers/provider**을 앞에서 설명된 대로 변경하고, **user name** 를 추가 하세요. 여기에서 **name**은 당신의 ISP에 접속하기위한 이름입니다. 다음으로 **/etc/ppp/pap-secrets** 또는 **/etc/ppp/chap-secrets**을 편집기로 열고, 당신의 패스워드를 입력합니다.

/etc/resolv.conf 파일을 편집기로 열어 당신의 ISP의 DNS 서버 IP 주소를 입력해야 할 것입니다. **/etc/resolv.conf**에 추가될 부분은 다음과 같은 형식을 가집니다. **nameserver xxx.xxx.xxx.xxx** 여기에서 **x**들은 당신의 IP 주소의 숫자들을 나타냅니다. 선택적으로 **usepeerdns** 옵션을 **/etc/ppp/peers/provider**에 추가 할 수 있습니다. 이 옵션을 사용하면 자동으로 반대쪽 호스트에서 주어지는 DNS 서버의 설정 값들을 사용하여 설정을 해줍니다.

대부분의 ISP들이 사용하는 로그인 방법을 사용한다면, 설정이 끝났습니다. **root**로 **pon** 명령을 내려 PPP 연결을 시작하고, **plog** 명령을 사용하여 그 과정을 지켜보세요. 연결을 끊기 위해서는 **root** 권한으로 **poff** 명령을 사용합니다.

데비안에서 PPP 사용하는 것에 대해 더 많은 정보를 원하신다면, **/usr/share/doc/ppp/README.Debian.gz** 파일을 읽어 보세요.

고정 SLIP 연결을 하려면, **slattach** 명령(**net-tools** 패키지에 있음)을 **/etc/init.d/network**에 추가 하세요. 유동 SLIP은 **gnudip** 패키지가 필요할 것입니다.

7.2.3.1 PPP over Ethernet (PPPOE) 설정

PPPOE는 광대역 연결을 위해 사용되는 PPP와 관련된 프로토콜 중 하나 입니다. 기본 설정과정에 당신이 이 연결을 설정하도록 도와줄 부분은 현재 없습니다. 하지만, 필수적인 소프트웨어들은 설치 가 되어 있고, PPPOE를 수작업으로 설정할 수 있습니다. PPPOE 설정을 하려면 이 설치 단계에서

터미널을 VT2로 바꿔 `pppoeconf` 명령을 실행하세요.

7.2.4 APT 설정

주요한 점은 사람들이 시스템에 패키지들을 설치하기 위해 `apt` 패키지에 있는 `apt-get` 프로그램을 사용하곤 한다는 점입니다.¹ 패키지 관리를 위한 다른 프론트엔드로는 `apt-get`를 사용하거나 의존하는 `aptitude`, `synaptic`와 오래된 `dselect`가 있습니다. 이러한 프론트엔드들은 패키지 찾기와 상태 체크 등의 추가적인 기능들을 사용하기 쉽게 제공하기 때문에 새로운 사용자들에게 추천됩니다.

APT는 어디에서 패키지들을 가져와야 할 지 설정되어 있어야 합니다. 설정 작업을 도와주는 `apt-setup` 프로그램이 있습니다.

설정 과정의 다음 단계는 어디에서 다른 데비안 패키지들을 찾을 수 있는 지 APT에게 알려주는 일입니다. 이 작업은 설치 후 언제든지 `apt-setup`를 실행하거나 `/etc/apt/sources.list`을 직접 편집해서 다시 수행 할 수 있습니다.

공식 CD-ROM 이 드라이브에 들어 있다면, 자동으로 CD-ROM 이 apt source로 설정될 것입니다. CD-ROM 이 검색되는 것을 통해 이 작업의 수행을 알 수 있을 것입니다.

공식 CD-ROM 이 없는 사용자를 위해 FTP, HTTP, CD-ROM, 또는 지역 파일시스템 등의 목록을 제공해 데비안 패키지들에 접근할 수 있는 방법을 선택할 수 있도록 합니다.

같은 데비안 아카이브라도 여러 개의 다른 APT source들을 가질 수 있다는 것을 알아 두세요. `apt-get`은 모든 가능한 버전 중에서 가장 높은 버전의 패키지를 자동으로 선택할 것입니다. 예를 들어 HTTP와 CD-ROM APT source가 있다면, `apt-get` 명령은 자동으로 가능 하다면 CD-ROM을 사용하고, 새로운 버전이 있는 경우에만 HTTP를 사용할 것입니다. 하지만, 필요없는 APT source들을 추가하는 것은 새 버전을 네트워크를 통해 검사하기 때문에 시스템을 느리게 할 수 있으므로 그리 좋은 생각이 아닙니다.

7.2.4.1 네트워크 패키지 소스(Sources) 설정

시스템의 나머지 부분을 네트워크를 통해 설치하려고 한다면, 일반적으로 `http` 소스를 선택합니다. `ftp` 소스도 괜찮습니다만, 연결을 만드는 데 약간 느립니다.

네트워크 패키지 소스 설정의 다음 단계는 `apt-setup`에게 당신이 살고 있는 나라를 알려주는 것입니다. 이 작업은 당신이 연결할 데비안 공식 인터넷 미러들 중의 하나를 설정합니다. 선택한 나라에 따라 가능한 서버들의 목록이 보여질 것입니다. 목록의 어느 것을 선택하더라도 괜찮습니다만, 보통 목록의 가장 위의 것을 선택합니다. 미러들의 목록은 데비안의 버전이 릴리즈 될 때 만들어 지기 때문에 어떤 미러들은 더이상 사용가능 하지 않을 수 있습니다.

미러를 선택한 후에 프록시 서버를 사용하는 지 물어볼 것입니다. 프록시 서버는 당신의 HTTP 또는 FTP 요청을 인터넷으로 전달하고 네트워크의 인터넷 접근을 최적화하고 일정하게 하기 위해 사용하는 서버입니다. 어떤 네트워크에서는 프록시 서버의 인터넷 연결만 허용하는 경우가 있습니다. 그러한

¹패키지들을 설치하는 실제 프로그램은 `dpkg` 명령임을 알아두세요. 그러나, 이 패키지는 좀 더 저수준의 도구입니다. `apt-get`은 `dpkg`를 적절히 불러주고, 당신이 설치하려고 하는 패키지가 필요로 하는 다른 패키지들을 알고, CD, 네트워크 또는 다른 곳으로부터 패키지들을 어떻게 받아들 지를 알고 있는 고수준의 도구입니다.

경우에는 프록시 서버의 이름을 입력해야 합니다. 사용자 이름과 암호 역시 입력해야 할 것입니다. 비록 몇몇 ISP들이 프록시 서버를 제공하기는 하지만, 집에서 사용하는 일반 사용자들은 프록시 서버의 설정이 필요 없을 것입니다.

미러 선택이 끝난 후에 새로운 네트워크 패키지 소스가 테스트 됩니다. 잘 된다면 다른 패키지 소스를 추가할 것인지를 물어볼 것입니다. 만일 해당 패키지 소스를 사용하는 데 문제가 있다면, 당신의 나라 또는 세계의 목록에서 다른 미러를 사용해 보거나 다른 네트워크 패키지 소스를 사용해 보세요.

7.2.5 패키지 설치

다음으로 몇개의 데비안에서 제공되는 소프트웨어 설정들을 제공 받을 것입니다. 항상 패키지 별로 당신의 새로운 시스템에 원하는 패키지를 설치할 수 있다는 것을 기억하세요. 이 것이 다음에 설명되는 **aptitude** 프로그램의 목적입니다. 하지만, 이것은 데비안의 15250 패키지들과 함께하는 긴 작업이 될 것입니다.

그래서 **태스크**들을 먼저 선택하고 다음으로 더 많은 개별 패키지들을 추가할 수 있게 되어 있습니다. “desktop environment”, “web server”, “print server”와 같은 태스크들은 당신의 컴퓨터를 사용하고자하는 많은 일이나 작업들을 나타냅니다.² **C.3절** 는 가능한 태스크들을 설치하기 위해 필요한 공간을 보여줍니다.

태스크들을 선택했다면, **Ok**을 선택하세요. 그러면 **aptitude**는 선택된 패키지들을 설치할 것입니다.

참고

아무런 태스크를 선택하지 않았더라도, 시스템에 없는 standard, important, required priority 패키지들은 설치될 것입니다. 이 기능은 명령창에서 `tasksel -ris`을 실행 하는 것과 같고, 현재 약 37M의 아카이브를 다운로드할 것입니다. 설치되는 패키지들의 숫자와 다운로드가 필요하다면 얼마 만큼 받았는 지를 보여줄 것입니다.

당신이 정말로 패키지 별로 설치하기를 원한다면 **tasksel**에서 “manual package selection” 옵션을 선택하세요. 이 옵션과 함께 하나 이상의 태스크를 선택했다면, **aptitude**가 **-visual-preview** 옵션과 함께 실행 될 것입니다. 이 것은 설치될 패키지들을 확인 할 수 있다는 것을 의미합니다.³ 아무런 태스크도 선택하지 않았다면, 일반 **aptitude** 화면이 보여질 것입니다. 선택이 끝난 후에 “g”를 눌러야 패키지들의 다운로드와 설치가 시작 됩니다.

²이 목록을 보여주기 위해 **base-config**는 단지 **tasksel** 프로그램을 실행 시킨 다는 것을 알아두세요. 다른 패키지들을 설치, 제거 하기 위해 설치 후 언제든지 실행 시킬 수 있습니다. 설치가 완료된 후에 특정 패키지 하나를 설치 하려면 `aptitude install package`을 실행 하세요. *package*은 당신이 찾는 패키지의 이름입니다.

³당신은 default 섹션 역시 바꿀 수 있습니다. 다른 추가적인 패키지를 선택하길 원한다면 **View** → **New Package View**를 사용하세요.

참고

아무런 태스크의 선택 없이 “manual package selection”을 선택했다면, 기본으로 아무런 패키지도 설치되지 않을 것입니다. 최소의 시스템 설치를 원한다면 이 옵션을 사용할 수 있습니다. 하지만, 재부팅 하기 전에 기본 시스템의 일부로 설치되어야 하는 패키지들의 선택의 책임도 당신에게 있습니다.

데비안에서는 15250 패키지들이 사용가능 하기 때문에 태스크 설치관리자에서 제공되는 태스크들은 작은 수의 것들만 포함되어 있습니다. 더 많은 패키지들의 정보를 보기 위해서는 `apt-cache search search-string`를 사용해 주어진 문자열을 검색하거나 다음에 설명되는 `aptitude`를 실행하세요. (`apt-cache(8) man` 페이지를 보세요.)

7.2.5.1 aptitude를 사용한 고급 패키지 선택

`Aptitude`는 패키지 관리를 위한 최신 프로그램입니다. `aptitude`는 당신이 개별 패키지나 주어진 기준에 맞는 패키지들의 집합이나 전체 태스크를 선택할 수 있게 해줍니다.

가장 기본적인 키 연결은 다음과 같습니다.

Key	Action
Up, Down	선택을 위/아래로 움직임
Enter	아이템 열기/닫기/활성화
+	패키지 설치하도록 표시
-	패키지를 제거하도록 표시
d	패키지 의존성 보여주기
g	패키지 다운로드/설치/제거를 실제로 수행
q	현재 보여지는 것을 그만둠
F10	메뉴 활성화

더 많은 명령어에 대해 보기를 원한다면 ? 키를 눌러 온라인 도움말을 보세요.

7.2.6 소프트웨어 설치 중의 프롬프트

`tasksel` 또는 `aptitude`를 통해 선택된 각 패키지들은 `apt-get`와 `dpkg` 프로그램을 통해 다운로드 된 후 패키지가 풀리고, 설치됩니다. 특정 프로그램이 사용자로부터 더 많은 정보를 필요로 하는 경우 이 과정에서 당신에게 즉시 물어볼 것입니다. 당신은 설치 오류(패키지 설치 과정에 금지된 에러들이 발생하는 경우 당신에게 알려줄 지를 물어보지만)가 발생하지 않는 지 보기위해 그 과정 동안의 출력을 보고 있기를 원할 것입니다.

7.2.7 메일 배달 에이전트(Mail Transport Agent) 설정

요즘 이메일은 많은 사람들의 생활에서 매우 중요한 부분을 차지하고 있기 때문에 데비안에서 설치 과정의 일부로 메일 시스템 설정을 하도록 하는 것은 그리 놀라운 일은 아닙니다. 데비안에서 기본 메일 배달 에이전트(MTA)는 **exim4** 입니다. 이 프로그램은 상대적으로 작고, 유연하며, 실행시키기 쉽습니다.

당신의 컴퓨터가 네트워크에 연결되어 있지 않는 경우에도 메일 설정이 필요한지 물어볼 수 있습니다. 그 대답은 '예' 입니다. 좀 더 설명하면 **cron**, **quota**, **aide** ... 등의 몇몇 시스템 유틸리티들은 당신에게 중요한 공지를 이메일을 통해 보낼 수 있습니다.

그래서 첫 번째 화면에서 몇 개의 일반적인 메일 사용 시나리오들이 보여질 것입니다. 가장 당신의 환경과 유사한 것을 선택합니다.

인터넷 사이트 당신의 시스템이 네트워크에 연결되어 있고 메일은 SMTP를 사용해 직접 보내고 받게 됩니다. 다음 화면에서 당신 시스템의 메일 이름이나 사용할 도메인들의 목록 등 기본적인 질문을 물어볼 것입니다.

스마트호스트에 의한 메일 보내기 이 시나리오에서는 당신의 나가는 메일은 “smarthost”라고 불리는 다른 머신으로 전달 되고 실제 작업은 여기에서 이루어 집니다. 스마트호스트는 보통 당신의 컴퓨터로 들어오는 메일도 저장합니다. 그래서 당신은 계속 온라인 상태일 필요가 없습니다. 이것은 fetchmail 등과 같은 프로그램을 통해 스마트호스트에서 메일을 다운로드해야 한다는 것을 의미합니다. 이 옵션은 dial-up 사용자들에게 적합합니다.

로컬 배달 당신의 시스템이 네트워크에 있지 않고 메일은 단지 로컬 사용자들 사이에서만 보내고 받아 집니다. 어떤 메시지도 보낼 계획이 없는 경우에도 이 옵션이 적극 추천됩니다. 몇몇 시스템 유틸리티들은 때때로 다양한 알림 메시지를 보냅니다. (e.g. “Disk quota exceeded”) 이 옵션은 더이상 물어보는 것이 없기 때문에 새로운 사용자에게도 편리합니다.

지금 설정 안함 당신이 하는 것에 대해 절대적으로 확신할 수 있는 경우에만 선택합니다. 이 옵션은 당신에게 설정되지 않은 메일 시스템을 남겨 놓습니다. — 당신이 설정하기 전까지는 메일을 보내거나 받을 수 없을 뿐만아니라 시스템 유틸리티들에게서 오는 중요한 메시지들을 놓칠 수 있습니다.

이 시나리오들 중에서 당신에게 적합한 것이 없거나 좀 더 세밀한 설정을 원한다면, 설치가 완료된 후에 `/etc/exim4` 디렉토리의 설정파일을 편집해야 합니다. **exim4**에 대해 더 많은 정보는 `/usr/share/doc/exim4`에서 찾을 수 있습니다.

7.3 로긴

패키지들의 설치가 끝난 뒤에 로긴 프롬프트가 보여질 것입니다. 선택한 개인 로긴 계정과 암호를 사용해 로긴합니다. 당신의 시스템은 이제 사용할 준비가 되었습니다.

새로운 사용자라면 시스템 사용을 시작 할 때 이미 설치되어 있는 문서들을 보기를 원할 것입니다. 현재 몇 개의 문서 시스템들이 있고, 다른 타입의 문서들을 통합하는 작업이 진행 중입니다. 이 문서 역시 몇 개의 시작 지점입니다.

설치된 프로그램들에 따라오는 문서는 `/usr/share/doc/` 안의 프로그램의 이름을 따른 하위 디렉토리에 있습니다. 예를 들면 당신의 시스템에 다른 프로그램을 설치하기 위한 `apt`를 사용하기 위한 APT 사용자 가이드는 `/usr/share/doc/guide.html/index.html` 안에 위치해 있습니다.

추가적으로 `/usr/share/doc/` 계층에는 몇몇 특별한 폴더들이 있습니다. 리눅스 HOWTO들이 `/usr/share/doc/HOWTO/en-txt/` 안에 `.gz` 형식으로 설치되어 있습니다. `dhhelp`를 설치하면 `/usr/share/doc/HTML/index.html` 안에 브라우징 할 수 있는 색인된 문서를 볼 수 있을 것입니다.

이 문서들을 보기위한 쉬운 방법 중 하나는 `cd /usr/share/doc/` 한 뒤에 `lynx` 과 공백 그리고 점 (점은 현재 디렉토리를 의미합니다.)을 입력하세요.

`info command` 또는 `man command`을 명령창에 입력해서 대부분의 명령에 대한 문서를 볼 수 있습니다. `help >`을 입력해서 셸 명령에 대한 도움말을 볼 수 있습니다. 명령의 결과가 화면 위로 지나가 버린 경우에는 `| more`을 명령 뒤에 입력해서 화면 위로 스크롤 되어 지나가기 전에 결과를 멈출 수 있습니다. 특정 문자로 시작되는 모든 명령들의 목록을 보기 원한다면 특정 문자를 입력하고 탭을 두번 누릅니다.

데비안 GNU/리눅스에 대한 완전한 소개를 원한다면 `/usr/share/doc/debian-guide/html/noframes/index.html`를 보세요.

Chapter 8

다음 단계 및 그 다음에 할 일

8.1 유닉스를 처음 접한다면

유닉스를 처음 접한다면, 아마 책을 몇권 사서 읽는 게 좋을 것입니다. 또 여러가지 좋은 정보들이 Debian Reference <<http://www.debian.org/doc/user-manuals#quick-reference>>에 있습니다. list of Unix FAQs <<http://www.faqs.org/faqs/unix-faq/>>에는 역사적인 훌륭한 참고자료들이 들어 있는 여러가지 유즈넷 문서 목록이 들어 있습니다.

리눅스는 유닉스를 구현한 것입니다. Linux Documentation Project (LDP) <<http://www.tldp.org/>> 사이트에는 리눅스에 관한 여러가지 HOWTO와 온라인 서적이 있습니다. 이 문서 대부분은 컴퓨터에 직접 설치할 수도 있습니다. **doc-linux-html** 꾸러미 (HTML 버전) 아니면 **doc-linux-text** 꾸러미를 (ASCII 버전) 설치하시고, `/usr/share/doc/HOWTO` 안의 파일을 읽어보십시오. 여러가지 언어로 되어 있는 LDP HOWTO도 데비안 꾸러미가 있습니다.

8.2 데비안에 익숙해지기

데비안은 다른 배포판들과 약간 다릅니다. 다른 배포판에 익숙해져 있다고 해도 몇 가지 알아둬야 시스템을 훌륭하고 깔끔한 상태로 유지합니다 이 장에서는 데비안에 익숙해지는 방법을 설명합니다. 데비안을 사용하는 방법을 하나하나 설명하지 않고, 성급한 분들을 위해 아주 약간의 시스템 개요만 설명합니다.

8.2.1 데비안 꾸러미 시스템

알아야 할 가장 중요한 점은 데비안 꾸러미 시스템입니다. 원칙적으로 시스템의 대부분을 꾸러미 시스템이 제어하게 됩니다. 이 부분은:

- `/usr` (`/usr/local` 제외)
- `/var` (`/var/local`을 만들면 그 안에서는 제어하지 않음)
- `/bin`

- /sbin
- /lib

예를 들어, /usr/bin/perl 파일을 덮어 썼다고 하면 당장은 덮어 쓴 파일로 동작하긴 하지만, perl 꾸러미를 업그레이드하면 그 파일을 업그레이드하면서 다시 덮어 씹니다. 전문가들이라면 aptitude에서 해당 꾸러미를 “고정” 상태로 놓아서 이 문제를 피해줍니다.

가장 훌륭한 꾸러미 설치 도구의 하나는 APT입니다. 명령행 방식의 apt-get을 사용할 수도 있고, 텍스트 전체화면 방식의 aptitude를 사용할 수도 있습니다. APT를 이용해 main, contrib, non-free 모두에서 설치할 수 있고, 그러면 표준 꾸러미 외에 특정 국가로 수출이 제한되어 있는 꾸러미까지도 설치할 수 있습니다.

8.2.2 프로그램 버전 관리

같은 이름의 프로그램이 여러가지 버전이 있는 경우 update-alternatives에서 관리합니다. 여러 버전의 프로그램을 관리하고 있다면, update-alternative 맨페이지를 읽어 보십시오.

8.2.3 CRON 작업 관리

시스템 관리자가 관리하는 작업들은 모두 /etc 안에 들어 있어야 합니다. 루트 CRON 작업으로 매일, 매주, 밤마다 실행해야 하는 게 있으면, 해당 작업을 /etc/cron.-daily,weekly,monthly” 아래에 넣어 두십시오. 이 디렉토리 아래의 파일들은 /etc/crontab에서 시작되어 알파벳 순서로 차례대로 실행하며 동기화 될 것입니다.

한편, (1) 특정 사용자로 실행해야 할 CRON 작업이 있거나, (2) 특정 시간이나 특정 주기로 실행해야 하는 작업이 있으면, /etc/crontab을 사용하거나, 아니면 더 좋은 방법으로 /etc/cron.d/아무개를 사용할 수 있습니다. 이 파일에는 CRON 작업을 실행할 사용자를 지정하는 필드가 있습니다.

둘 중에 어떤 경우이든 간에, 파일을 편집하기만 하면 CRON에서 자동으로 처리합니다. 따로 명령어를 실행할 필요가 없습니다. 더 자세한 정보는 cron(8), crontab(5), /usr/share/doc/cron/README. Debian을 참고하십시오.

8.3 DOS 및 Windows 다시 살리기

베이스 시스템을 설치하고 마스터 부트 레코드를 쓰면, 리눅스를 부팅할 수 있지만 그 외의 운영 체제는 부팅할 수 없는 경우가 있습니다. 설치할 때 어떻게 했느냐에 따라 달라집니다. 이 장에서는 어떻게 예전의 시스템을 살려서 DOS나 Windows로 다시 부팅할 수 있는지 다룹니다.

부팅 관리자 LILO는 리눅스 외의 다른 PC용 운영 체제도 부팅할 수 있습니다. LILO는 /etc/lilo.conf 파일을 편집해 설정합니다. 이 파일을 편집한 다음에 lilo 프로그램을 실행해야 합니다. 이렇게 해야 하는 이유는 프로그램을 실행해야만 바뀐 사항이 실제로 적용되기 때문입니다.

lilo.conf 파일에서 중요한 부분은 image와 other 키워드가 들어 있는 줄과 그 뒤에 나오는 줄입니다. 각각은 LILO가 부팅하는 시스템 하나하나에 대해 쓰여 있습니다. 여기에는 커널 (image), 루트

파일시스템, 커널 파라미터가 들어 있고, 리눅스가 아닌 다른 (other) 운영 체제를 부팅하는 설정들도 들어 있습니다. 이 `image`와 `other` 키워드는 여러 번 사용할 수 있습니다. 설정 파일에서 시스템의 순서가 중요합니다. 이 순서에 따라 제한시간이 지났을 때 자동으로 부팅하는 시스템이 달라집니다. (제한시간동안 `shift` 키를 누르지 않아서 `LILLO`가 멈추지 않았다고 가정합니다.)

데비안을 처음 설치한 후에, `LILLO`를 이용해 시스템을 부팅하도록 설정했다고 가정합니다. 다른 리눅스 커널로 부팅하려면 `/etc/lilo.conf` 설정 파일을 편집해서 다음 줄을 추가해야 합니다:

```
image=/boot/vmlinuz.new
  label=new
  append="mcd=0x320,11"
  read-only
```

기본적인 설정에서는 처음 두 줄만 필요합니다. 다른 두 옵션에 대해 알고 싶으시면 `LILLO` 문서를 참고하십시오. 이 문서는 `/usr/share/doc/lilo/` 안에 있습니다. 읽어봐야 할 파일은 `Manual.txt`입니다. 부팅에 관해 빨리 알고 싶으시면 `LILLO` 맨페이지를 볼 수 있습니다. 설정 파일의 키워드에 대해 살펴보시려면 `lilo.conf` 맨페이지를, 부트 섹터를 새로 설치하는 설명에 대해서는 `lilo` 맨페이지를 보십시오.

Debian GNU/Linux에는 GRUB (`grub` 꾸러미), CHOS (`chos` 꾸러미), Extended-IPL (`extipl` 꾸러미), loadlin (`loadlin` 꾸러미) 등과 같은 다른 부트로더도 있습니다.

8.4 그 외에 읽을 것과 정보

특정 프로그램에 대한 정보를 보려면, `man` 프로그램 명령을 실행해 보시고, 아니면 `info` 프로그램 명령을 실행해 보십시오.

`/usr/share/doc` 안에도 유용한 문서가 많이 들어 있습니다. 특히 `/usr/share/doc/HOWTO` 및 `/usr/share/doc/FAQ`에 재미있는 정보가 많이 들어 있습니다. 버그를 제출하려면 `/usr/share/doc/debian/bug*` 파일들을 보십시오. 어떤 특정 프로그램의 데비안 관련 정보를 읽으려면 `/usr/share/doc/(꾸러미이름)/README.Debian` 파일을 보십시오.

데비안 웹사이트 <<http://www.debian.org/>>에는 많은 양의 데비안 관련 문서가 있습니다. 특히, Debian GNU/Linux FAQ <<http://www.debian.org/doc/FAQ/>> 와 데비안 레퍼런스 <<http://www.debian.org/doc/user-manuals#quick-reference>>를 보십시오. 많은 데비안 문서들의 인덱스는 데비안 문서 프로젝트 <<http://www.debian.org/doc/ddp/>>를 통해서 얻을 수 있습니다. 데비안 커뮤니티에 참여하면 커뮤니티 안에서 서로 도와줍니다. 데비안 메일링 리스트에 가입하려면 메일링 리스트 가입 <<http://www.debian.org/MailingLists/subscribe>> 페이지를 보십시오. 마지막으로 데비안 메일링 리스트 아카이브 <<http://lists.debian.org/>>에는 데비안에 관한 유용한 정보들이 들어 있습니다.

GNU/리눅스에 관한 정보는 보통 Linux Documentation Project <<http://www.tldp.org/>>에 보면 있습니다. 여기에 GNU/리눅스 시스템에 관한 HOWTO 및 다른 훌륭한 정보가 들어 있는 링크가 있습니다.

8.5 새 커널 컴파일하기

Why would someone want to compile a new kernel? It is often not necessary since the default kernel shipped with Debian handles most configurations. Also, Debian often offers several alternative kernels. So you may want to check first if there is an alternative kernel image package that better corresponds to your hardware. However, it can be useful to compile a new kernel in order to:

- 특별한 하드웨어가 필요할 때, 혹은 기존 커널에서 하드웨어 충돌이 발생할 때
- use options of the kernel which are not supported in the pre-supplied kernels (such as high memory support)
- 필요없는 드라이버를 지워 커널을 최적화해 부팅 시간을 빠르게 하기
- create a monolithic instead of a modularized kernel
- 새로 업데이트된 커널이나 개발버전 커널 사용
- learn more about linux kernels

8.5.1 커널 이미지 관리

커널 컴파일을 꺼려하지 마십시오. 재미있는 일이고 그만큼 좋은 점이 있습니다.

To compile a kernel the Debian way, you need some packages: **fakeroot**, **kernel-package**, **linux-source-2.6** and a few others which are probably already installed (see `/usr/share/doc/kernel-package/README.gz` for the complete list).

이 방법을 이용해 커널 소스에서 .deb 꾸러미 파일을 만들고, 커널에 들어 있지 않은 모듈이 있으면 그 모듈의 .deb 파일도 커널에 맞춰서 만듭니다. 이 방법이 커널 이미지를 관리하는 더 좋은 방법입니다. `/boot` 안에 커널, `System.map` 및 빌드할 때 사용한 커널 설정 파일이 들어 있게 됩니다.

Note that you don't *have* to compile your kernel the "Debian way"; but we find that using the packaging system to manage your kernel is actually safer and easier. In fact, you can get your kernel sources right from Linus instead of **linux-source-2.6**, yet still use the **kernel-package** compilation method.

`/usr/share/doc/kernel-package` 안에 **kernel-package**를 사용하는 방법에 대한 문서가 있습니다. 여기서는 간단히 어떤 단계를 거쳐야 하는지만 설명합니다.

Hereafter, we'll assume you have free rein over your machine and will extract your kernel source to somewhere in your home directory¹. We'll also assume that your kernel version is 2.6.12. Make sure you are in the directory to where you want to unpack the kernel sources, extract them using `tar xjf /usr/src/linux-source-2.6.12.tar.bz2` and change to the directory **linux-source-2.6.12** that will have been created.

그리고 커널을 설정합니다. X11을 설치하고 실행중이라면 `make xconfig`를 실행합니다. 아니면 `make menuconfig`를 실행합니다. (`menuconfig`는 **libncurses5-dev** 꾸러미를 설치해야 합니다.) 온라인 도움말을 읽어 보시면서 주의를 기울여 선택하십시오. 뭔가 의심스러운 게 있으면, 잘 모르더라도 해당

¹There are other locations where you can extract kernel sources and build your custom kernel, but this is easiest as it does not require special permissions.

장치 드라이버(이더넷 카드, SCSI 컨트롤러 등 하드웨어 주변장치를 관리하는 소프트웨어)를 포함하는 편이 더 좋습니다. 주의하십시오: 특정 하드웨어와는 관계없는 옵션이고 그 옵션이 뭔지 잘 모르겠다면 기본값 그대로 놔 두십시오. “Loadable module support”에 있는 “Kernel module loader” 옵션을 꼭 선택하십시오. (기본값으로 선택이 안 되어 있습니다.) 이 옵션이 없으면, 데비안 시스템에 문제가 발생합니다.

소스 트리에 필요없는 파일들을 지워서 **kernel-package** 파라미터를 초기화하십시오. `make-kpkg clean` 명령을 실행합니다.

이제 커널을 컴파일합니다: `fakeroot make-kpkg --revision=custom.1.0 kernel_image` 명령을 실행합니다. 버전 번호 “1.0”은 마음대로 바꿀 수 있습니다. 이 번호는 커널 빌드를 파악하려고 사용하는 버전 번호일 뿐입니다. 마찬가지로 “custom” 대신에 어떤 단어라도 (예를 들어 호스트이름) 넣을 수 있습니다. 커널 컴파일은 컴퓨터 속도에 따라 시간이 좀 오래 걸릴 수도 있습니다.

PCMCIA 기능이 필요하다면 **pcmcia-source** 꾸러미도 설치해야 합니다. tar.gz 파일을 루트 권한으로 `/usr/src` 디렉토리에 푸십시오. (모듈이 `/usr/src/modules`에 들어 있는 게 중요합니다.) 그 다음에 루트로 `make-kpkg modules_image`를 실행하십시오.

컴파일이 끝나면 다른 꾸러미와 마찬가지로 직접 만든 커널을 설치합니다. 루트 권한으로 `dpkg -i ../kernel-image-2.6.12-서브아키텍처_custom.1.0_i386.deb` 명령을 실행합니다. **서브아키텍처** 부분은 커널 옵션을 어떻게 따라 달라지는 “i586같은” 서브아키텍처 이름입니다. `dpkg -i kernel-image. . .` 명령은 커널 이미지를 설치하고, 다른 관련 파일 몇 개도 설치합니다. 예를 들어 **System.map** 파일을 설치하고 (커널 문제를 디버깅할 때 사용), `/boot/config-2.6.12` (커널 설정이 들어 있음) 파일도 설치합니다. **kernel-image-2.6.12** 꾸러미를 설치하면 알아서 해당 플랫폼의 부트로더를 사용해 부팅 설정을 업데이트합니다. 그래서 부트로더를 직접 신경 쓰지 않아도 부팅할 수 있게 됩니다. 모듈 꾸러미를 만들었다면 (예를 들어 PCMCIA가 있다면), 그 꾸러미도 설치해야 합니다.

이제 시스템을 다시 시작합니다. 위의 단계에서 발생한 메시지들을 잘 읽어보시고, `shutdown -r now` 명령을 실행하십시오.

For more information on **kernel-package**, read the fine documentation in `/usr/share/doc/kernel-package`.

Appendix A

설치 방법

이 문서는 새로운 **debian-installer**를 사용하여 Intel x86 용 Debian GNU/Linux sarge (“i386”) 를 설치하는 방법을 설명합니다. 설치하는 동안 필요한 정보에 대한 간략한 설명을 담고 있습니다. 더 많은 정보가 필요할 때에 [Debian GNU/Linux InstallationGuide](#) 안의 더 자세한 설명과 연결할 것입니다.

A.1 사전 준비

debian-installer는 아직 베타 상태 입니다. 만일 설치중 버그를 보게되면, 어떻게 그 버그를 알려야하는 지 [5.3.6절](#)를 참조 해주세요. 이 문서에서 답변을 얻지 못한 궁금한 사항들은 debian-boot 메일링 리스트 (debian-boot@lists.debian.org) 에게 알려주거나 IRC (freenode 네트워크의 #debian-boot 채널) 에 질문해주세요.

A.2 설치관리자 시작하기

CD 이미지들의 링크는 **debian-installer** 홈페이지 <<http://www.debian.org/devel/debian-installer/>>를 방문해 보세요. debian-cd 팀은 데비안 CD 페이지 <<http://www.debian.org/CD/>>에서 **debian-installer**를 사용한 CD 이미지들을 제공합니다. CD들을 어디서 얻을 수 있는 지에 해한 더 자세한 정보는 [4.1절](#)를 보세요.

일부 설치 방법에는 CD 이미지 외의 다른 이미지들을 필요로 합니다. **debian-installer** 홈페이지 <<http://www.debian.org/devel/debian-installer/>>에 다른 이미지들의 링크가 있습니다. [4.2.1절](#) 는 데비안 미러에서 어떻게 이미지들을 찾을 수 있는 지를 설명합니다.

다음 섹션들에는 각각의 가능한 설치 방법들을 위해 어떤 이미지들을 받아야 하는 지에 대해 자세한 설명을 할 것 입니다.

A.2.1 CDROM

debian-installer로 sarge을 설치할 수 있는 두 종류의 netinst CD 이미지들이 있습니다. 이 이미지들은 CD로 부팅해서 추가 패키지들을 네트워크를 통해 설치하기 때문에 'netinst'라는 이름을 가집니다.

두 이미지들의 차이점은 하나는 베이스 패키지들이 포함된 full netinst 이미지이고, 다른 하나는 이 패키지들을 다운로드 받아야하는 business card 이미지라는 것입니다. 원한다면 설치에 네트워크가 필요없는 full size CD 이미지를 받을 수도 있습니다. 설치에는 전체 세트에서 단지 첫 번째 CD만 필요할 것입니다.

선호하는 이미지를 내려 받아 CD를 굽습니다. CD로 부팅하기 위해 3.6.1절에 설명된 대로 BIOS 설정을 바꿔야 할 수도 있습니다.

A.2.2 플로피

만일 CD로 부팅할 수 없다면, 데비안을 설치하기 위해 플로피 이미지들을 받을 수 있습니다. 필요한 이미지들은 floppy/boot.img, floppy/root.img이고, 더해서 드라이버 디스크가 필요할 수 있습니다.

부트 플로피는 boot.img를 가진 플로피 입니다. 이 플로피는 부팅될때 두 번째 플로피 — root.img를 담고 있는 플로피 —를 넣으라고 합니다.

네트워크를 통해 설치하려고 한다면, 일반적으로 floppy/net-drivers.img가 필요할 것 입니다. 이 이미지는 많은 이더넷 카드들과 PCMCIA를 지원하는 추가 드라이버들을 담고 있습니다.

CD가 있지만, 그것으로 부팅할 수 없는 경우 플로피들로 부팅한 뒤에 floppy/cd-drivers.img를 사용하여 CD를 사용한 설치를 수행할 수 있습니다.

플로피 디스크들은 가장 신뢰할 수 없는 미디어들 중의 하나이기 때문에 많은 망가진 디스크들에 대한 대비를 하세요 (5.3.1절 참고). 다운로드한 .img 파일은 각각 한 장의 플로피에 쓰여집니다; /dev/fd0에 기록하기 위해 dd 명령을 사용하거나 다른 방법들을 사용할 수 있습니다 (자세한 내용은 4.3절 참고). 하나 이상의 플로피들을 가질 것이기 때문에 이름을 써놓는 것이 좋습니다.

A.2.3 USB 메모리 스틱

탈착이 가능한 USB 스토리지 장치들로 부터의 설치도 가능 합니다. 예를 들어 USB keychain은 어디서든 구할 수 있는 손쉬운 데비안 설치 미디어 입니다.

USB 메모리 스틱을 준비하는 쉬운 방법은 hd-media/boot.img.gz을 다운로드 받아 압축을 풀어서 128 MB 이미지를 추출하는 것입니다. 추출된 이미지를 적어도 128 mb의 공간이 있는 메모리 스틱에 직접 쓰세요. 이 작업은 메모리 스틱에 들어 있는 기존의 내용을 덮어 쓸 것입니다. 그 다음 메모리 스틱을 마운트 해보면, 그 위에 FAT 파일시스템을 가진 것을 알 수 있습니다. 그 다음 데비안 netinst CD 이미지를 다운로드 받아 그 파일을 메모리 스틱에 복사합니다. .iso로 끝나기만하면 어떤 파일 이름이라도 괜찮습니다.

debian-installer를 사용하는 메모리 스틱을 만드는 좀 더 유연한 방법들이 있으며, 더 작은 메모리 스틱을 사용하는 것도 가능 합니다. 자세한 내용은 4.4절을 보세요.

몇몇 BIOS들은 USB 저장매체에서 직접 부팅할 수 있지만, 몇몇은 하지 못합니다. USB 장치로 부팅하기 위해 “removable drive” 또는 “USB-ZIP”으로 부팅하도록 BIOS 설정을 바꿔주어야 할 것입니다. 직접 부팅하지 못하는 경우 플로피로 부팅하고 나머지 설치과정을 USB 스틱을 사용할 수 있습니다. 도움이되는 힌트와 자세한 설명은 5.1.3절을 보세요.

A.2.4 네트워크 부팅

네트워크로 부터 **debian-installer**를 부팅하는 것 역시 가능합니다. 네트워크 부팅의 다양한 방법들은 아키텍처와 netboot 설정에 의존합니다. **netboot/** 파일들이 **debian-installer**를 네트워크 부팅하는데 사용될 수 있습니다.

설정하기 가장 쉬운 것은 PXE 네트워크 부팅일 것입니다. **netboot/pxeboot.tar.gz** 파일을 **/var/lib/tftpboot** 또는 tftp 서버를 위한 적절한 위치에 풀어 놓으세요. DHCP 서버가 **/pxelinux.0** 파일 이름을 클라이언트에게 넘겨줄 수 있도록 설정하고, 모든 것이 잘 동작하기를 바라며 파일 이름을 넘겨주세요. 자세한 설명은 [4.6절](#) 를 참고하세요.

A.2.5 하드 디스크 부팅

제거 가능한 미디어가 아닌 다른 운영체제를 가지고 있는 기존의 하드디스크를 사용하여 설치관리자를 부팅하는 것이 가능합니다. 하드 디스크의 최상위 디렉토리에 **hd-media/initrd.gz**, **hd-media/vmlinuz**, 데비안 CD 이미지를 다운로드 받으세요. 해당 CD 이미지의 파일 이름이 **.iso**로 끝나는 지 확인합니다. 이제 **initrd** 이미지로 리눅스를 부팅하는 것이 중요한 문제 입니다. [5.1.2절](#) 는 그 방법을 설명합니다.

A.3 설치

설치관리자를 시작하면 초기 화면을 볼 것입니다. **Enter**를 누르거나 다른 부트 방법들과 파라미터들 ([5.2절](#) 참조)을 위한 설명을 읽어 보세요. 만약 2.6 커널을 원한다면, **boot:** 프롬프트에서 **linux26**을 입력하세요. ¹

잠시 후에 언어를 선택하는 부분이 보일 것입니다. 방향키를 사용하여 언어를 선택하고 **Enter**를 눌러 계속 진행합니다. 다음으로 당신의 언어를 사용하는 국가들에 대한 목록과 당신의 국가를 선택하는 부분이 보일 것입니다. 만일 짧은 리스트에 나타나지 않는다면 세계의 모든 국가들의 리스트에서 선택할 수 있습니다.

키보드 설정을 확인합니다. 잘 알지 못한다면 **default**를 선택합니다.

이제 데비안 설치관리자가 하드웨어를 검사하고, CD, 플로피, USB 등에서 나머지 부분을 읽어들이는 동안 기다립니다.

다음으로 설치관리자는 네트워크 하드웨어를 식별하고 DHCP를 통해 네트워크 설정을 할 것입니다. 만일 네트워크에 있지 않거나 DHCP를 가지지 않았다면, 네트워크 설정을 수동으로 할 수 있습니다.

디스크를 파티션 별로 나눌 차례입니다. 먼저 전체 디스크 또는 드라이브의 남은 공간을 자동으로 파티션 나누기 할 수 있도록 기회가 주어집니다. 이 방법은 새로운 사용자나 바쁜 사용자에게 추천됩니다. 만일 자동 파티션 나누기를 원하지 않는다면 메뉴에서 수동(**manual**)을 선택하세요.

만일 보존하기를 원하는 DOS 또는 Windows 파티션이 있다면, 자동 파티션 나누기에 매우 주의 하세요. 수동 파티션 나누기를 선택한다면, 데비안을 설치할 공간을 만들기위해 설치관리자를 사용해 기존의 FAT 또는 NTFS 파티션 사이즈를 조정할 수 있습니다.

¹플로피로부터 부팅하는 것을 제외한 대부분의 부팅 방법에서 2.6 커널이사용가능 합니다.

다음 화면에서 파티션들이 어떻게 나뉘어져 있는 지 어디에 마운트 되어 있는 지를 보여주는 파티션 테이블을 볼 것 입니다. 변경하거나 삭제할 파티션을 선택합니다. 만일 자동 파티션 나누기를 선택했다면, 설정된 대로 사용하기 위해 메뉴에서 단지 **Finished Partitioning**을 선택할 수 있습니다. 적어도 한 파티션은 스왑 공간을 위해 할당하고, 한 파티션은 /에 마운트 해야 한다는 것을 유념하세요. **부록 B**은 파티션 나누기에 대한 좀 더 자세한 정보를 가지고 있습니다.

debian-installer는 파티션을 포맷하고 베이스 시스템 설치를 시작합니다. 이 과정은 약간의 시간이 걸립니다. 이어서 커널 설치가 계속됩니다.

마지막 과정은 부트로더의 설치입니다. 만일 설치관리자가 컴퓨터에서 다른 운영체제를 발견한다면, 그것을 부트메뉴에 추가하고 알려줄 것입니다. 기본으로 GRUB가 일반적으로 좋은 위치인 첫 번째 하드디스크의 마스터 부트 레코드에 설치됩니다. 기본 설정을 바꾸거나 다른 곳에 설치 할 수 있는 기회를 제공합니다.

debian-installer는 설치과정이 끝났음을 당신에게 알려줄 것입니다. cdrom 또는 다른 부트 미디어를 제거하고 재부팅하기 위해 **Enter**를 누르세요. 그러면 **7장**에 설명된 설치과정의 다음 스테이지로 넘어갈 것입니다.

설치과정에 대한 더 많은 정보가 필요하다면, **6장**를 보세요.

A.4 설치에 대한 결과를 보내주세요

debian-installer를 사용하여 설치에 성공했다면, 보고서를 제공해 주는 시간은 내주세요. 새로 설치된 시스템의 /root디렉토리에는 **install-report.template** 템플릿이 있습니다.내용을 채워서 **5.3.6절**에 설명된 대로 **installation-reports** 패키지에 대한 버그로 제출해 주세요.

만일 base-config에 도달하지 못했거나 다른 문제가 발생했다면, 데비안 설치관리자의 버그를 발견한 것일 겁니다. 설치관리자를 개선하기위해 우리는 그것들에 대해 알 필요가 있습니다. 그 사실들을 보고할 시간을 내주세요. 문제를 보고하기위해 설치보고서(installation report)를 사용할 수 있습니다; 설치가 완전히 실패한다면, **5.3.5절**를 참조하세요.

A.5 마지막..

우리는 당신의 데비안 설치가 즐겁고 데비안이 즐만하다는 것을 발견하길 바랍니다. 아마도 당신은 **8장**을 읽기를 원할 것입니다.

Appendix B

데비안의 하드디스크 파티션

B.1 파티션을 나눌 때 파티션의 크기 등 고려할 점

GNU/Linux는 전용으로 쓸 수 있는 파티션을 최소한 하나 필요로 합니다. 각종 응용프로그램과 사용자 파일, 데비안 전체를 하나의 파티션에 둘 수 있습니다. 스왑 파티션도 반드시 필요하다고 하는 사람이 많지만, 꼭 그런 것만은 아닙니다. “스왑”이란 운영체제가 사용할 수 있는 일종의 낙서공간으로, 디스크를 “가상메모리”처럼 사용할 수 있게 해줍니다. 스왑을 별도의 파티션에 두면, 리눅스가 스왑을 더 효율적으로 사용할 수 있게 됩니다. 일반적인 파일을 스왑으로 사용하도록 강제할 수 있지만 별로 추천하는 방식은 아닙니다.

하지만 대부분의 사람들은 GNU/Linux에게 여러 개의 파티션을 할당합니다. 파일시스템을 몇 개의 작은 파티션으로 나누는게 좋은 이유로 두가지를 들 수 있습니다. 첫번째는 안전문제입니다. 파일시스템이 망가지게 될 경우, 대부분 파티션 하나에만 문제가 생깁니다. 즉 그 동안 잘 백업해 둔 정보를 토대로 시스템의 일부만을 복구하면 됩니다. 이런 이유로 “root” 파티션을 따로 두는 것이 좋습니다. 여기에 시스템의 가장 핵심적인 파일들이 있기 때문입니다. 그러면 다른 파티션들에 문제가 생기더라도 이 파티션에 오류가 없다면 부팅을 해서 문제점을 고칠 수 있습니다. 시스템을 처음부터 완전히 설치하는 것에 비해 수고를 덜 수 있습니다.

두번째 이유는 기업환경에서 더 중요하다고 할 수 있습니다만, 각자의 상황에 따라 다르다고 보는 것이 맞을 듯 합니다. 예를 들어, 대량의 스팸메일을 받고 있는 메일서버는 금방 파티션 하나를 채울 수 있습니다. `/var/mail`을 별도의 파티션에 두었다면, 스팸메일로 하드디스크가 가득차서 시스템이 동작을 멈추는 현상을 어느 정도 예방할 수 있을 것입니다.

여러개의 파티션을 사용할 때의 유일한 단점은 각 파티션의 적절한 크기를 사전에 알기 힘들다는 점에 있습니다. 파티션을 너무 작게 만들면 시스템을 새로 설치하거나 그 파티션에 있는 파일들을 자주 다른 파티션으로 옮겨줘야 하는 불편함이 있습니다. 반면에 파티션을 너무 크게 만들면 다른 곳에서 쓸 수 있는 공간을 낭비하는 셈이 됩니다. 디스크 공간이 많이 저렴해졌다 하지만 낭비할 수는 없지 않습니까?

B.2 디렉토리 구조

Debian GNU/Linux 은 디렉토리와 파일명을 정할 때 표준 파일 시스템 구조 Filesystem Hierarchy Standard <<http://www.pathname.com/fhs/>>를 따르고 있습니다. 이 표준을 준수함으로써 사용자들과 유저 프로그램들은 처음 접하는 시스템이라도 원하는 파일과 디렉토리의 위치를 예측할 수 있게 됩니다. 루트 디렉토리는 슬래쉬(/)로 표시됩니다. 데비안은 루트 디렉토리 아래에 다음의 디렉토리들을 포함하고 있습니다:

디렉토리	내용
bin	필수적인 명령 바이너리들
boot	부트 로더가 필요로 하는 정적 파일들
dev	하드웨어의 다바이스 파일들
etc	이 호스트 고유의 설정파일 등
home	사용자의 홈디렉토리
lib	시스템 운영에 필수적인 공유 라이브러리와 커널 모듈들
media	플로피 디스켓 등의 탈부착 가능한 매체를 마운트 시키는 디렉토리
mnt	파일 시스템을 임시로 마운트 시키는 디렉토리
proc	커널 2.4와 2.6에서 시스템 정보를 저장하는 가상 디렉토리
root	root 사용자의 홈디렉토리
sbin	필수적인 시스템 관리 바이너리
sys	2.6 커널이 시스템 정보를 저장하는 가상 디렉토리
tmp	임시 파일들
usr	2차적인 디렉토리 구조
var	내용이 자주 변하는 파일들
opt	별도로 추가하는 응용 소프트웨어

디렉토리와 하드디스크 파티션을 구성할 때 고려해야 할 점들을 아래에 정리했습니다. 실제 디스크 사용량은 각 시스템의 설정과 주 사용용도에 따라 달라질 수 있습니다. 아래의 내용은 참고만 하시고 실제 설정에는 각 시스템에 맞게 하실 것을 권합니다.

- **/etc, /bin, /sbin, /lib, /dev**는 반드시 루트 파티션(/)에 존재해야 합니다. 그렇지 않을 경우 부팅에 문제가 발생합니다. 일반적으로 150-250 MB정도가 필요합니다.
- **/usr**에는 유저 프로그램(**/usr/bin**)과 라이브러리(**/usr/lib**)와 시스템 문서(**/usr/share/doc**) 등이 저장됩니다. 많은 경우 이 부분이 가장 하드디스크 공간을 많이 차지할 것입니다. 최소한 500 MB를 할당하시고 시스템에 설치할 패키지의 구성과 수에 따라 적절히 조절하십시오. 워크스테이션이나 서버로 사용될 시스템은 넉넉잡아 4 - 6 GB 정도를 필요로 할 것입니다.
- **/var**에는 뉴스그룹글이나 이메일, 웹페이지, 데이터베이스나 패키지 관리 프로그램의 캐시 등 자주 변하는 정보가 주로 저장됩니다. 이 디렉토리의 크기는 시스템에 따라 크게 차이가 나지만, 일반적인 사용자들의 시스템에서는 패키지 관리 프로그램이 이 공간이 대부분을 사용할 것입니

다. 데비안에 포함된 프로그램을 한번에 거의 다 설치하실 경우 `/var`에 2 ~ 3 GB정도를 할당하시면 됩니다. 한번에 설치하지 않고 서비스와 유틸리티 따로, 문서 작업용 프로그램 따로, X 윈도우 따로하는 식으로 나눠서 설치하실 경우 300에서 500 MB로도 가능합니다. 하드디스크 크기가 부족하고 시스템의 대대적인 업데이트 예정이 없을 경우 이 파티션을 30이나 40 MB 정도로 작게 하실 수도 있습니다.

- 프로그램들이 사용하는 임시 데이터는 `/tmp`에 주로 저장됩니다. 대부분 40-100 MB 정도면 충분하지만, 압축 유틸리티나 CD/DVD 굽는 유틸리티나 각종 멀티미디어 프로그램들은 이미지 파일을 `/tmp`에 임시로 저장하는 경우도 있으므로 이 경우에는 `/tmp`의 크기를 적절히 조절하십시오.
- `/home`에는 사용자들의 홈디렉토리가 저장됩니다. 사용자들은 홈디렉토리에 개인 파일들을 저장할텐데, 이 시스템을 사용하는 사용자가 몇 명이며 각자가 자신의 홈디렉토리 어떤 종류의 파일을 저장할지에 따라 필요한 `/home`디렉토리의 크기가 달라지게 됩니다. 100 MB정도씩을 할당하면 되지만, 홈디렉토리 안에 MP3나 동영상 등 대용량 파일을 많이 저장할 예정이라면 공간을 충분히 늘려 잡으십시오.

B.3 권장하는 파티션 구조

초보자이시거나 개인이나 가정에서 사용할 시스템일 경우에는 `/`에 스왑 파티션만을 사용하는 것이 가장 쉽고 편한 방식일 것입니다. 하드디스크의 크기가 6 GB이상 될 경우엔 파일시스템을 ext3방식으로 포맷하십시오. Ext2 방식을 사용할 경우 주기적으로 파일시스템을 검사해야 하는데, 하드디스크의 크기가 너무 클 경우 부팅시간이 너무 오래걸리는 단점이 있습니다.

멀티유저 시스템이거나 하드디스크의 용량이 큰 시스템에서는 `/usr`, `/var`, `/tmp`, `/home` 각각을 별도의 파티션에 두어 `/`파티션과 별도로 두는 것이 좋습니다.

데비안에서 제공하지 않는 프로그램을 많이 설치할 경우 `/usr/local`를 별도의 파티션에 두는 것이 좋을 수도 있습니다. 메일서버일 경우에는 `/var/mail`를 별도의 파티션에 두는 것이 좋으며, `/tmp` 역시 20 내지 50 MB 정도되는 별도의 파티션에 두는 것이 좋습니다. 사용자가 많은 멀티유저 서버일 경우 `/home`의 크기를 충분히 잡아서 별도의 파티션에서 관리하는 것이 좋습니다. 이처럼 파일시스템 파티션의 구성은 시스템마다 달라지므로 각자의 필요에 따라 적절히 결정하시면 됩니다.

매우 복잡한 시스템을 구성하신다면 멀티디스크 HOWTO <<http://www.tldp.org/HOWTO/Multi-Disk-HOWTO.html>>를 참고하십시오. 하드디스크 파티션하기에 관한 심도있는 내용을 다루고 있으므로 ISP나 서버관리자들에게 주로 도움이 될 것입니다.

스왑의 크기는 사람마다 생각이 다릅니다. 일단 시스템의 메모리만큼 스왑공간을 잡는 것이 좋습니다. 또한 16 MB보다는 큰 것이 좋습니다. 256 MB 메모리를 갖고 있는 컴퓨터에서 동시에 10000개의 공식을 풀 생각이시라면 1기가 이상 필요로 할 수도 있으므로 각자의 필요에 따라 스왑공간의 크기를 조절하시면 됩니다.

i386, m68k, SPARC 32비트와 PowerPC등의 32비트 프로세서는 2GB가 스왑 파티션의 한계입니다. 대부분의 경우 이 정도로도 충분합니다. 하지만 스왑 공간이 정말 많이 필요하다면 스왑을 여러 하드디스크에 걸쳐서 잡으시고(이 방식을 “spindles” 방식이라고 부릅니다), 가급적이면 서로 다른 SCSI, IDE 채널의 하드디스크에 잡으십시오. 커널이 알아서 여러 스왑 파티션에 적절히 작업을 분배하므로 시스템의 성능향상에 도움이 됩니다.

예를 하나 들어보겠습니다. 좀 오래된 컴퓨터에 램이 32 MB 있고 /dev/hda에 1.7 GB짜리 IDE 하드디스크가 있습니다. 500MB짜리 /dev/hda1에 다른 운영체제가 설치되어 있으며, /dev/hda3는 32MB의 스왑공간으로 사용되고 있습니다. /dev/hda2에는 1.2GB이 할당되어 있어 리눅스가 사용할 예정입니다.

필요할 하드디스크 용량이 얼마나 될지 감을 잡는데 도움이 되고자 C.3절에 작업별로 필요로 하는 하드디스크 공간이 정리되어 있습니다.

B.4 리눅스에서의 디바이스 이름

리눅스에서 디스크와 파티션을 다른 시스템과 다른 이름으로 부르기도 합니다. 리눅스에서 파티션을 생성하고 마운트할 때 이 이름들을 알고 있어야 합니다. 기본적으로는 다음 규칙에 따라 이름이 정해 집니다.

- 첫번째 플로피 디스크 드라이브는 /dev/fd0라고 불립니다.
- 두번째 플로피 디스크 드라이브는 /dev/fd1라고 불립니다.
- SCSI ID 주소에서 제일 먼저 오는 SCSI 디스크는 /dev/sda라고 불립니다.
- 두번째로 오는 디스크는 /dev/sdb라고 불리며, 각 디스크마다 이런 방식으로 이름이 붙습니다.
- 첫번째 SCSI CD-ROM은 /dev/scd0이라고 불리며, /dev/sr0라고도 불립니다.
- IDE 주컨트롤러의 마스터는 /dev/hda라고 불립니다.
- IDE 주컨트롤러의 슬레이브는 /dev/hdb라고 불립니다.
- IDE 부컨트롤러의 마스터와 슬레이브는 각각 /dev/hdc, /dev/hdd라고 불릴 수 있습니다. 최근에 나온 IDE 컨트롤러는 2개의 통신채널을 갖고 있기 때문에 실질적으로는 2개의 컨트롤러로 취급될 수 있습니다.
- 첫번째 XT 디스크는 /dev/xda입니다.
- 두번째 XT 디스크는 /dev/xdb입니다.

각 드라이브 상의 파티션은 디스크 이름 뒤에 숫자를 붙인 이름을 갖게 됩니다. sda1와 sda2는 각각 첫번째 SCSI 디스크의 첫번째와 두번째 파티션을 뜻합니다.

예를 하나 들어보도록 하겠습니다. 컴퓨터에 2개의 SCSI 디스크가 있고, 각각 SCSI 주소 2와 4에 연결되어 있다고 합시다. 2번 주소에 연결된 디스크가 첫번째 SCSI로, sda라고 불리게 됩니다. 두번째 4번 주소의 SCSI 디스크는 sdb와 연결되게 됩니다. sda에 파티션이 3개가 있다면 각각 sda1, sda2, sda3라고 불리게 됩니다. sdb와 파티션들 역시 같은 규칙을 적용받게 됩니다.

참고로, SCSI 호스트 버스 어댑터(컨트롤러)가 있을 경우 어느 드라이브가 첫번째인지 확인하기 어려울 수 있습니다. 이 경우엔 부팅시 화면에 표시되는 드라이브 모델과 기능을 살펴보고 어느 드라이브가 sda에 연결되었는지 확인하는게 최선의 방법입니다.

리눅스 파티션은 드라이브 이름 뒤에 숫자 1에서 4를 붙여서 주파티션을 구분합니다. 예를 들어, 첫번째 IDE드라이브의 첫번째 파티션은 /dev/hda1입니다. 논리파티션은 5번부터 시작하므로, 이 드라

이브의 첫번째 논리파티션은 `/dev/hda5`입니다. 논리파티션이 들어있는 확장파티션은 독자적으로 쓸 수 없습니다. 이는 SCSI 디스크, IDE디스크 모두에 적용되는 사실입니다.

B.5 데비안의 파티션용 프로그램

여러가지 하드디스크와 시스템 종류에 맞게 데비안의 개발자들이 여러 종류의 파티션 프로그램을 개발했습니다. 아래에 시스템에 따른 파티션 프로그램의 목록을 열거하였습니다.

partman 데비안에서 권장하는 파티션용 프로그램. 이 만능 프로그램으로 파티션의 크기를 조정하며, 파일시스템을 생성하고 (윈도우에서 말하는 “포맷”), 마운트 위치도 설정할 수 있습니다.

fdisk 전통적인 리눅스 파티션 프로그램으로, 전문가용.

FreeBSD용 파티션이 디스크에 있을 경우엔 신중하셔야 합니다. 설치용 커널은 이 형식의 파티션들을 지원하지만, **fdisk** 프로그램이 이들 파티션을 화면에 나타내는 방식이 달라질 수 있습니다. 리눅스+FreeBSD HOWTO <<http://www.tldp.org/HOWTO/Linux+FreeBSD-2.html>>를 참조하시면 보다 자세한 내용을 얻어실 수 있습니다.

cfdisk 대부분의 사람들을 위한 풀스크린 파티션 프로그램.

참고로 **cfdisk**는 FreeBSD파티션을 전혀 인식하지 못하기 때문에 디바이스의 이름이 예상과 달라질 수 있습니다.

Partition a Hard Disk를 선택할 때 이 프로그램들 중 하나가 실행되게 됩니다. 디폴트로 실행되는 프로그램을 바꾸고자 할 경우 파티션 프로그램을 끝낸 후 **Alt**와 **F2**를 동시에 눌러서 셸 프롬프트(`tty2`)로 가십시오. 그 다음 원하시는 프로그램을 인자와 함께 수동으로 실행시키십시오. 파티션이 끝났으면 다시 **debian-installer**로 돌아가서 **Partition a Hard Disk** 부분을 생략하시면 됩니다.

동시에 20개 이상의 IDE 파티션을 사용하실 경우 파티션 21 이상은 따로 디바이스를 생성해야 합니다. 그렇지 않을 경우 파티션을 초기화하는 작업이 실패하게 됩니다. 아래에 `tty2` 명령어나 셸을 띄우기를 통해 21번째 파티션의 디바이스를 생성시키는 과정을 보여드립니다:

```
# cd /dev
# mknod hda21 b 3 21
# chgrp disk hda21
# chmod 660 hda21
```

적절한 디바이스가 생성되어 있지 않을 경우 새로 설치한 리눅스는 부팅이 되지 않을 것입니다. 커널과 모듈을 모두 설치하신 후 다음을 실행하십시오:

```
# cd /target/dev
# mknod hda21 b 3 21
# chgrp disk hda21
# chmod 660 hda21
```

부트 파티션을 “Bootable”로 설정하는 것을 잊지 마십시오.

B.5.1 Intel x86에서 파티션하기

DOS나 윈도우 등 다른 운영체제와 데비안이 공존하기를 원하신다면 원 운영체제가 사용하는 파티션의 크기를 조정해야 할 수도 있습니다. 설치프로그램을 통하여 FAT와 NTFS형 파티션의 크기를 조정하실 수 있습니다. 설치 프로그램의 파티션하기 단계에서 수동 파티션을 선택한 다음 이미 존재하는 파티션의 크기를 조정하시면 됩니다.

PC의 BIOS도 디스크를 파티션하는데 각종 제약을 둡니다. 일단 하나의 드라이브에 둘 수 있는 “주파티션”과 “논리파티션”의 개수에 한계가 있습니다. 또한 1994[U+2010]98년 이전의 BIOS에는 부팅이 가능한 파티션의 위치에도 제약이 있습니다. 자세한 내용은 리눅스 파티션하기 HOWTO <<http://www.tldp.org/HOWTO/Partition/>> 와 Phoenix BIOS FAQ <<http://www.phoenix.com/en/Customer+Services/BIOS/BIOS+FAQ/default.htm>>를 참조하시면 됩니다만, 이 단원에서 기초적인 내용은 배우실 수 있습니다.

“주파티션”은 PC용 디스크에서 전통적으로 사용하던 파티션 방식입니다만, 이 방식으로는 파티션을 4개만 둘 수 있습니다. 이 한계를 벗어나기 위해 “확장파티션”과 “논리파티션”이라는 개념이 만들어졌습니다. “주파티션” 하나를 “확장파티션”으로 두면 그 파티션의 공간을 최대 60개의 논리파티션으로 나눌 수 있습니다. 하지만 하나의 디스크에는 “확장파티션”을 하나만 둘 수 있습니다.

리눅스는 파티션의 개수를 SCSI 디스크는 파티션 15개(주파티션 3개, 논리파티션 12개), IDE 디스크에는 63개(주파티션 3개, 논리파티션 60개)로 제한하고 있습니다. 하지만 일반적인 Debian GNU/Linux 시스템은 기본으로 파티션용 디바이스를 20개만 제공하므로 20개보다 많은 파티션을 두고자 할 경우에는 직접 파티션용 디바이스를 추가하셔야 합니다.

대용량 IDE 디스크를 사용하고, LBA 어드레싱이나 일부 하드디스크 제조사에서 제공하는 오버레이 드라이버를 사용하지 않을 경우 커널을 포함하고 있는 부트파티션의 위치에도 제약이 있습니다. 부트 파티션은 맨앞 1024 실린더, BIOS의 주소변환 없이는 맨앞 524 MB에 뒤야 합니다.

이 제약은 제조사에 따라 1995-98년도 이후에 만들어진 BIOS에는 적용되지 않습니다. 이들 BIOS는 “향상된 디스크 드라이브 지원을 위한 스펙”에 맞추어서 제조되었습니다. 리눅스 로더인 LILO와 데비안 고유의 **mbr** 모두 디스크에서 램으로 커널을 읽어들이기 때 BIOS의 지원을 필요로 합니다. BIOS가 대용량 디스크 접근용 확장 명령어인 int 0x13을 지원하지 않으면 구형의 디스크 접근 인터페이스가 사용될 것입니다. 이 접근 인터페이스는 1023번째 실린더 이후의 실린더에 접근하지 못합니다. 일단 리눅스가 부팅된다면 디스크 접근에 BIOS가 사용되지 않으므로 이런 제약이 없어집니다.

시스템에 대용량 디스크가 있을 경우 실린더 변환 기술이 필요한 경우가 있습니다. 이 때는 BIOS 설정으로 들어가셔서 LBA (Logical Block Addressing)나 CHS 변환 모드(“대용량”)를 선택하시면 됩니다. 대용량 디스크를 사용할 때의 문제점에 대해서 자세히 알고 싶으시면 Large Disk HOWTO <<http://>

[/www.tldp.org/HOWTO/Large-Disk-HOWTO.html](http://www.tldp.org/HOWTO/Large-Disk-HOWTO.html)>를 참조하십시오. 실린더 변환 기술을 사용하지만 BIOS에서 대용량 디스크 접근을 지원하지 않으면 부팅용 파티션이 변환 후의 앞 1024번째 실린더 내에 할당되어야 합니다.

이 경우 디스크의 맨 앞에 5 - 10 MB의 작은 파티션을 뒤서 부팅 파티션으로 사용하고 나머지 영역을 원하는 대로 파티션하시면 됩니다. 이 부팅 파티션은 리눅스 커널들이 저장될 반드시 `/boot`에 마운트되어야 합니다. 이 방법은 시스템이 LBA를 사용하든 대용량 디스크용 CHS 변환을 사용하든, BIOS에서 대용량 디스크 접근을 지원하든 안하든 상관없이 모든 경우에 적용될 수 있습니다.

Appendix C

여러가지 내용들

C.1 미리 설정 파일 예제

다음은 자동 설치용 미리 설정 파일로 완전히 동작하는 예제입니다. 사용법은 [4.7절](#)에 설명되어 있습니다. 이 파일을 직접 사용하려면 일부 줄의 코멘트를 없애야 할 수도 있습니다.

참고

예제를 매뉴얼에 제대로 표시하려고 불가피하게 일부 줄은 여러 줄로 쪼개 놓았습니다. 여러 줄로 쪼개 놓을 때 연결 문자인 “\”로 표시해 놓았고 두번째 줄에는 약간 더 들여 써 놓았습니다. 실제 미리 설정 파일에서는, 여러 줄로 쪼개 놓은 줄은 **한 줄로 붙여야** 합니다. 한 줄로 붙이지 않으면 미리 설정은 예측할 수 없는 결과를 발생하면서 실패합니다.

“깨끗한” 예제 파일은 `<../example-preseed.txt>`에 있습니다.

시작.

```
# 미리 설정 파일을 사용하려면, 먼저 설치 프로그램으로 부팅하고
# 사용할 미리 설정 파일이 무엇인지 지정해야 합니다. 커널 부팅
# 파라미터로 파일의 위치를 넘기면 되는데, 부팅할 때 수동으로 하거나
# syslinux.cfg 파일에서 커널의 append 줄에 파라미터를 넣으면 됩니다.

# 네트워크 부팅이라면, 다음을 사용하십시오:
# preseed/url=http://host/path/to/preseed
# CD를 다시 굽는 경우라면, 다음을 사용하십시오:
# preseed/file=/cdrom/preseed
# USB 미디어에서 설치하는 경우라면, 다음을 사용하고 미리 설정
# 파일을 USB 스틱의 맨 위 디렉토리에 넣으십시오.
```

```

# preseed/file=/hd-media/preseed
# 위에서 지정한 위치에 이 파일을 복사하십시오.
#
# 설치 과정의 어떤 부분은 미리 설정을 통해 자동화할 수 없습니다.
# 미리 설정 파일을 읽어들이기도 전에 응답해야 하는 질문들 때문입니다.
# 예를 들어 미리 설정 파일을 네트워크를 통해 내려받는 경우, 네트워크
# 설정을 먼저 해야 합니다. initrd 미리 설정을 하는 한 가지 이유는
# 앞 단계에서도 미리 설치 과정에 대해 미리 설정을 할 수 있기 때문입니다.
#
# 미리설정 파일을 일부 단계에 대해서만 사용한다면, 그래도 설치를
# 자동화할 수 있습니다. 각각의 미리 설정 값을 커널 파라미터로 넘길
# 수 있습니다. 변수/경로=값 식으로 아래의 미리 설정 변수를 넘깁니다.
#
# 거기까지 끝났으면, debconf/priority=critical을 넣어서 아래 미리 설정에
# 해당하지 않는 질문은 물어보지 않도록 합니다. 그리고 syslinux.cfg에서
# timeout을 1로 놓아서 설치 프로그램에서 부팅할 때 엔터를 누를 필요가
# 없도록 합니다.
#
# 단 커널은 최대 8개까지의 명령행 파라미터와 8개까지의 환경 파라미터만
# 받아들일 수 있습니다. 그 8개에는 설치 프로그램에서 기본으로 추가하는
# 파라미터가 포함되어 있습니다. 그 개수가 넘어가면, 2.4 커널은 넘어가는
# 옵션을 버리고 2.6 커널은 벗어나버립니다. 버전 2.6.9 커널 및 그 이후의
# 커널에서는 32개까지의 명령행 파라미터와 32개까지의 환경 파라미터를
# 받아들입니다.
#
# 'vga=normal'이나 'devfs=mount' 같은 기본 옵션중의 일부는 대부분의 경우
# 안 써도 상관없습니다. 그러면 미리 설정 옵션을 몇 개 더 쓸 수 있습니다.

# 언어, 국가, 키보드 선택은 파일에서 미리 설정할 수 없습니다. 언어, 국가,
# 키보드에 해당하는 질문은 미리 설정 파일을 읽어들이기 전에 물어보기
# 때문입니다. 파일에 넣는 대신에 파라미터를 몇 개 더 커널에 넘기십시오:
# languagechooser/language-name=English
# countrychooser/shortlist=US
# console-keymaps-at/keymap=us

#### 네트워크 설정.

# 물론 미리 설정 파일을 네트워크에서 가져오는 경우라면 이렇게 해도
# 동작하지 않습니다! 하지만 CD나 USB 스틱에서 부팅하는 경우라면 아주
# 좋습니다. 미리 설정 파일을 네트워크에서 읽어들이는다면 네트워크 설정
# 파라미터를 커널 파라미터에 넘길 수도 있습니다.

# netcfg에서 네트워크에 연결되어 있는 인터페이스를 하나 선택합니다. 이렇게
# 하면 인터페이스가 여러 개 있을 때 목록을 표시하지 않고 넘어갑니다.

```

```

d-i netcfg/choose_interface select auto

# dhcp 서버가 느려서 제한 시간이 초과되는 경우에는 다음과 같이 하는 게
# 좋습니다.
#d-i netcfg/dhcp_timeout string 60

# 네트워크를 수동으로 설정하려면, 다음과 같이 합니다:
#d-i netcfg/disable_dhcp boolean true
#d-i netcfg/get_nameservers string 192.168.1.1
#d-i netcfg/get_ipaddress string 192.168.1.42
#d-i netcfg/get_netmask string 255.255.255.0
#d-i netcfg/get_gateway string 192.168.1.1
#d-i netcfg/confirm_static boolean true

# 호스트 이름이나 도메인 이름은 여기 설정한 것보다 DHCP에서 가져온 것이
# 우선입니다. 하지만 다음 값을 설정하면 DHCP에서 어떤 값이 오더라도
# 물어보지는 않게 합니다.
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain

# WEB 키 대화 상자를 표시하지 않게 합니다.
d-i netcfg/wireless_wep string
# 일부 ISP는 DHCP 호스트 이름을 암호로 사용합니다.
#d-i netcfg/dhcp_hostname string radish

#### 미러 설정.

d-i mirror/country string enter information manually
d-i mirror/http/hostname string http.us.debian.org
d-i mirror/http/directory string /debian
d-i mirror/suite string testing
d-i mirror/http/proxy string

### 파티션하기.

# 시스템에 빈 공간이 있을 때 그 부분을 한 개 파티션으로 사용합니다.
#d-i partman-auto/init_automatically_partition \
#   select Use the largest continuous free space

# 다른 방법으로 파티션할 디스크를 지정합니다 장치 이름은 DEVFS 나
# DEVFS가 아닌 전통적인 방식으로 지정할 수 있습니다.
# 예를 들어, DEVFS 방식으로 첫번째 디스크는:
d-i partman-auto/disk string /dev/discs/disc0/disc

```

```

# 미리 정의되어 있는 파티션 방식 중의 하나를 사용할 수도 있습니다:
d-i partman-auto/choose_recipe select \
    All files in one partition (recommended for new users)
#d-i partman-auto/choose_recipe select Desktop machine
#d-i partman-auto/choose_recipe select Multi-user workstation

# 그것도 아니고 자기만의 방식을 사용 하려면...
# 파티션 방식 (recipe) 형식은 devel/partman-auto-recipe.txt 파일에 설명되어
# 있습니다. 파티션 방식 파일을 d-i 환경에 집어 넣을 수 있으면, 그 파일을
# 위치를 써 주기만 하면 됩니다.
#d-i partman-auto/expert_recipe_file string /hd-media/recipe

# 그렇지 않다면, 전체 파티션 방식을 한 줄에 넣을 수도 있습니다. 다음
# 예제는 작은 /boot 파티션을 만들고, 적당한 스왑을 만들고, 나머지 공간을
# 루트 파티션으로 사용 합니다:
#d-i partman-auto/expert_recipe string boot-root :: \
# 20 50 100 ext3 $primary{ } $bootable{ } method{ format } format{ } \
# use_filesystem{ } filesystem{ ext3 } mountpoint{ /boot } . \
# 500 10000 1000000000 ext3 method{ format } format{ } \
# use_filesystem{ } filesystem{ ext3 } mountpoint{ / } . \
# 64 512 300% linux-swap method{ swap } format{ } .
# 참고로, 다음은 위의 파티션 방식을 읽기 좋은 모양으로 쓴 것입니다:
# boot-root ::
# 40 50 100 ext3
# $primary{ } $bootable{ }
# method{ format } format{ }
# use_filesystem{ } filesystem{ ext3 }
# mountpoint{ /boot }
# .
# 500 10000 1000000000 ext3
# method{ format } format{ }
# use_filesystem{ } filesystem{ ext3 }
# mountpoint{ / }
# .
# 64 512 300% linux-swap
# method{ swap } format{ }
# .

# 다음을 사용하면 확인 질문 없이 partman에서 자동으로 파티션합니다.
d-i partman/confirm_write_new_label boolean true
d-i partman/choose_partition select \
    Finish partitioning and write changes to disk
d-i partman/confirm boolean true

```

```

##### 부트로더 설치.

# 기본 부트 로더는 (x86의 경우) GRUB입니다. GRUB 대신에 LILO를 사용하려면,
# 다음 줄의 코멘트를 없애십시오:
#d-i grub-installer/skip boolean true

# 다음을 설정해도 안전합니다. 이렇게 하면 컴퓨터에 다른 운영체제가
# 없을 경우에 자동으로 GRUB을 MBR에 설치합니다.
d-i grub-installer/only_debian boolean true

# 다음과 같이 하면 다른 운영체제가 있더라도 GRUB을 MBR에 설치합니다.
# 이렇게 하면 다른 운영체제를 부팅할 수 없을 수도 있기 때문에 덜 안전합니다.
d-i grub-installer/with_other_os boolean true

# 다른 방법으로, MBR이 아닌 위치에 GRUB을 설치하려면 다음 줄의 주석을
# 없애고 편집하십시오:
#d-i grub-installer/bootdev string (hd0,0)
#d-i grub-installer/only_debian boolean false
#d-i grub-installer/with_other_os boolean false

##### 첫 단계 설치 끝마치기.

# 설치가 끝났다는 마지막 메시지를 표시하지 않습니다.
d-i prebaseconfig/reboot_in_progress note

##### 쉘 명령어.

# d-i 미리 설정은 원래부터 보안에 안전하지 않습니다. 설치 프로그램 중의
# 어느 부분도 버퍼 오버플로우나 그 밖의 방법으로 미리 설정 파일의 값을
# 조작하는 공격을 검사하지 않습니다. 믿을 만한 곳에 있는 미리 설정 파일만
# 사용하십시오! 설치 프로그램 안에서 어떤 쉘 명령어라도 실행할 수 있는
# 방법이 만들어져 있습니다. 위험하지만 이 방법은 매우 유용하므로,
# 다음과 같이 설치 프로그램 내에서 쉘 명령어를 실행할 수 있습니다.

# 첫번째 명령어는 미리 설정 파일을 읽어들이 직후에 가능한 빨리
# 실행합니다.
#d-i preseed/early_command string anna-install some-udeb

# 다음 명령어는 설치가 끝나기 직전에, 하지만 /target 디렉토리가 아직
# 마운트되어 있을 때 실행합니다.
#d-i preseed/late_command string echo foo
> /target/etc/bar

# 다음 명령어는 base-config가 시작할 때 실행합니다.

```

```

#base-config base-config/early_command string echo hi mom

# 다음 명령어는 base-config가 끝난 다음에 login: 프롬프트 직전에
# 실행합니다. 다음 명령어는 필요한 꾸러미를 설치하거나 시스템의 설정을
# 조작할 때 유용합니다.
#base-config base-config/late_command string \
# apt-get install zsh; chsh -s /bin/zsh

##### 설치 두 번째 단계 미리 설정

##### base-config 미리 설정.

# 들어가는 메시지를 표시하지 않습니다.
base-config base-config/intro          note

# 마지막 메시지를 표시하지 않습니다.
base-config base-config/login          note

# 디스플레이 관리자를 설치하지만 base-config가 끝난 다음에 바로 시작하지는
# 않게 하려면.
#base-config base-config/start-display-manager          boolean false

# 일부 설치 프로그램 버전은 설치한 내용을 개발자들에게 보고합니다. 기본값은
# 보고하지 보고를 하면 어떤 소프트웨어를 많이 사용하는 지 및 무엇을
# CD에 포함할 지 결정하는데 도움이 됩니다.
#popularity-contest popularity-contest/participate boolean false

##### 시계 및 시간대 설정.

# 하드웨어 시계를 UTC로 할지 여부를 조정합니다.
base-config tzconfig/gmt boolean true
# 자기가 미국에 있다면 걸 설치 프로그램에 지정하려면, 다음 변수를
# 이용해 시간대를 설정할 수 있습니다.
# (가능한 값은: Eastern, Central, Mountain, Pacific, Alaska, Hawaii,
# Aleutian, Arizona East-Indiana, Indiana-Starke, Michigan, Samoa, other)
base-config tzconfig/choose_country_zone/US select Eastern
# 캐나다에 있다면.
# (가능한 값은: Newfoundland, Atlantic, Eastern, Central,
# East-Saskatchewan, Saskatchewan, Mountain, Pacific, Yukon, other)
base-config tzconfig/choose_country_zone/CA select Eastern
# 브라질에 있다면. (가능한 값은: East, West, Acre, DeNoronha, other)
base-config tzconfig/choose_country_zone/BR select East
# 많은 국가들은 시간대가 하나밖에 없습니다. 이러한 국가들 중에 하나를
# 설치 프로그램에 지정하려면, 다음을 이용해서 표준 시간대를 선택하십시오.

```

```

base-config tzconfig/choose_country_zone_single boolean true
# 다음 질문은 위에서 말한 국가들 이외에 시간대가 여러 개인 국가에
# 해당하는 것입니다. 시간대 중의 하나 혹은 "other"를 미리 설정할 수
# 있습니다.
#base-config tzconfig/choose_country_zone_multiple select

#### 계정 설정.

# 루트 열쇠를 미리 설정하려면, 이 파일에 암호를 그대로 씁니다.
# 다른 사람이 알 수 없도록 주의하십시오!
#passwd passwd/root-password password r00tme
#passwd passwd/root-password-again password r00tme

# 일반 사용자 계정을 만들지 않고 넘어가려면.
#passwd passwd/make-user boolean false

# 그게 아니면, 사용자 이름과 로그인을 미리 설정할 수 있습니다.
#passwd passwd/user-fullname string Debian User
#passwd passwd/username string debian
# 그리고 열쇠입니다. 주의하십시오!
#passwd passwd/user-password password insecure
#passwd passwd/user-password-again password insecure

#### APT 설정.

# 다음 질문은 두번째 단계에서 꾸러미를 설치할 때 어떤 소스를 사용할 지
# 지정합니다. 가능한 값은 cdrom, http, ftp, filesystem, edit source list
# by hand입니다.
base-config apt-setup/uri_type select http

# ftp나 http를 사용한다면, 국가와 미러에 대한 질문을 하게 됩니다.
base-config apt-setup/country select enter information manually
base-config apt-setup/hostname string http.us.debian.org
base-config apt-setup/directory string /debian
# 미러를 하나 고른 다음에 멈춥니다.
base-config apt-setup/another boolean false

# non-free와 contrib 소프트웨어를 설치합니다.
#base-config apt-setup/non-free boolean true
#base-config apt-setup/contrib boolean true

# 보안 업데이트를 사용합니다.
base-config apt-setup/security-updates boolean true

```

꾸러미 선택.

```
# 사용할 수 있으면 뭐든지 작업을 설치할 수 있습니다.
# 현재 사용 가능한 작업은: Desktop environment, Web server, Print server,
# DNS server, File server, Mail server, SQL database, Laptop,
# Standard system, manual package selection입니다. manual package
# selection을 선택하면 aptitude를 실행합니다. 작업을 하나도 설치하지
# 않을 수도 있고, 다른 방법으로 꾸러미를 설치할 수도 있습니다.
tasksel tasksel/first multiselect Desktop environment, Standard system
#tasksel tasksel/first multiselect Web server, Mail server, DNS server
```

메일 프로그램 설정.

```
# 보통 exim을 설치할 때는 몇 개 안 되는 질문을 합니다. 이 두 개 질문도 그냥
# 넘어가려면 다음과 같이 합니다. 이것보다 더 복잡하게 미리 설정할 수도
# 있습니다.
```

```
exim4-config exim4/dc_eximconfig_configtype \
    select no configuration at this time
exim4-config exim4/no_config boolean true
exim4-config exim4/no_config boolean true
```

```
# 다음 값을 앞에서 만든 사용자 계정으로 지정하는 게 좋습니다. 이 값을
# 비워두면 postmaster 메일이 /var/mail/mail에 쌓입니다.
```

```
exim4-config exim4/dc_postmaster string
```

X 설정.

```
# 데비안의 X 설정을 미리 설정하는 것도 가능하지만, 해당 컴퓨터의 비디오
# 하드웨어에 대해 자세히 알아야 합니다. 데비안의 X 설정은 모든 설정을
# 자동으로 하지 못하기 때문입니다.
```

```
# 일부 카드에 대해서는 X에서 올바른 드라이버를 자동 검색합니다. 하지만
# 미리 설정할 경우에 미리 설정하는 게 더 우선합니다. vesa는 대부분의
# 경우에 동작합니다.
```

```
#xserver-xfree86 xserver-xfree86/config/device/driver select vesa
```

```
# 마우스 자동 검색은 좀 위험한 부분이 있습니다. 실패하면 X가 계속해서
# 다시 시도합니다. 그러므로 미리 설정을 하면 마우스를 검색하지 못했을
# 때 무한루프에 빠질 가능성이 있습니다.
```

```
#xserver-xfree86 xserver-xfree86/autodetect_mouse boolean true
```

```
# 모니터 자동 검색을 하는 게 좋습니다.
```

```
xserver-xfree86 xserver-xfree86/autodetect_monitor boolean true
```

```
# LCD 디스플레이라면 다음 줄의 주석을 없애십시오.
```

```

#xserver-xfree86 xserver-xfree86/config/monitor/lcd boolean true
# X의 모니터 설정은 세 가지 방법이 있습니다. 다음은 "medium"으로 미리
# 설정하는 방법이고 언제나 사용할 수 있습니다. "simple"은 사용할 수
# 없을 수도 있고, "advanced"는 너무 많이 물어봅니다.
xserver-xfree86 xserver-xfree86/config/monitor/selection-method \
    select medium
xserver-xfree86 xserver-xfree86/config/monitor/mode-list \
    select 1024x768 @ 60 Hz

#### 그 외.

# 어떤 소프트웨어를 설치하느냐에 따라, 혹은 설치하는 중에 무언가 잘못되는
# 경우, 다른 질문을 물어볼 수도 있습니다. 물론 이 질문도 미리 설정할 수
# 있습니다. 설치하는 동안 물어볼 수 있는 모든 질문의 목록을 받고 싶다면,
# 설치를 한 다음에 다음 명령어를 실행하십시오:
# debconf-get-selections --installer
> file
# debconf-get-selections
>
> file

# 원한다면 다른 미리 설정 파일을 여기에 포함할 수도 있습니다. 포함되는
# 파일에 들어 있는 설정은 이 파일에 들어 있는 기존의 설정보다 우선합니다.
# 여러 개 파일을 공백으로 구분해서 쓸 수도 있습니다. 그러면 모든 파일을
# 읽어들입니다. 포함되는 파일에도 각각 preseed/include가 들어 있을 수
# 있습니다. 여기에 쓰는 파일이름은 상대경로이므로, 다음 파일은 다음 파일을
# 포함하는 파일과 같은 디렉토리에서 가져옵니다.
#d-i preseed/include string x.cfg

# 더 유연하게 하고 싶을 때 다음과 같이 하면 쉘 명령어를 실행하고, 그
# 명령어가 미리 설정 파일의 이름을 출력하면, 그 파일을 포함합니다.
# 예를 들어, 특정 USB 저장 장치에 따라 다른 설정을 사용하려면 (이 경우에는
# 내장 카드 리더) 다음과 같이 합니다;
#d-i preseed/include_command string \
#   if $(grep -q "GUID: 0aec3050aec305000001a003" /proc/scsi/usb-storage-*/*); \
#   then echo kraken.cfg; else echo otherusb.cfg; fi

# 설치하기 전에 미리 설정 파일의 형식을 검사하려면, debconf-set-selections
# 명령을 사용하십시오:
# debconf-set-selections -c preseed.cfg

```

C.2 리눅스 장치

리눅스에서는 `/dev` 안에 여러가지 특수한 파일이 들어 있습니다. 이 파일들을 장치 파일이라고 합니다. 유닉스 세계에서는 하드웨어를 접근하는 방법이 좀 다릅니다. 어떤 특수 파일에 접근하면 실제로는 거기에 해당하는 하드웨어에 접근하게 됩니다. 장치 파일은 실제 시스템의 구성 하드웨어에 대한 인터페이스입니다. `/dev`에 있는 파일은 일반 파일과 완전히 다르게 동작하기도 합니다. 다음은 중요한 장치 파일의 목록입니다.

fd0	첫번째 플로피 드라이브
fd1	두번째 플로피 드라이브

hda	첫번째 IDE 포트 마스터에 달린 IDE 하드디스크 / CD-ROM
hdb	첫번째 IDE 포트 슬레이브에 달린 IDE 하드디스크 / CD-ROM
hdc	두번째 IDE 포트 마스터에 달린 IDE 하드디스크 / CD-ROM
hdd	두번째 IDE 포트 슬레이브에 달린 IDE 하드디스크 / CD-ROM
hda1	첫번째 IDE 하드디스크의 첫번째 파티션
hdd15	네번째 IDE 하드디스크의 열다섯번째 파티션

sda	SCSI ID가 가장 낮은 (0번) SCSI 하드디스크
sdb	SCSI ID가 다음 번호인 (1번) SCSI 하드디스크
sdc	SCSI ID가 다음 번호인 (2번) SCSI 하드디스크
sda1	첫번째 SCSI 하드디스크의 첫번째 파티션
sdd10	네번째 SCSI 하드디스크의 열번째 파티션

sr0	SCSI ID가 가장 낮은 SCSI CD-ROM
sr1	SCSI ID가 다음 번호인 (2번) SCSI CD-ROM

ttyS0	시리얼 포트 0, MS-DOS에서는 COM1
ttyS1	시리얼 포트 1, MS-DOS에서는 COM2
psaux	PS/2 마우스 장치
gpmdata	가장 장치, GPM (마우스) 데몬에서 나온 데이터의 리피터

cdrom	CD-ROM 드라이브에 대한 심볼릭 링크
mouse	마우스 장치 파일에 대한 심볼릭 링크

null	이 장치로 들어간 데이터는 모두 사라집니다
-------------	-------------------------

zero	이 장치에서 끊임없이 0을 읽을 수 있습니다
------	--------------------------

C.2.1 마우스 설정하기

마우스는 리눅스 콘솔에서 (gpm을 이용해) 사용할 수도 있고 X 윈도우 환경에서 사용할 수도 있습니다. 이 두가지를 동시에 사용하려면 다음과 같이 gpm 리피터를 사용해 마우스 신호를 X 서버에도 들어가도록 만들면 됩니다:

```
mouse => /dev/psaux => gpm => /dev/gpmdata -> /dev/mouse => X
          /dev/ttyS0          (리피터)          (심볼릭 링크)
          /dev/ttyS1
```

(/etc/gpm.conf에서) 리피터 프로토콜을 raw로 하고 /etc/X11/XF86Config이나 /etc/X11/XF86Config-4에서 X 설정은 원래의 마우스 프로토콜로 합니다.

gpm을 X에서도 사용하는 이런 방법은 마우스가 갑자기 빠졌을 때도 좋은 점이 있습니다. gpm을 다음 명령으로 다시 시작하면,

```
# /etc/init.d/gpm restart
```

X를 다시 시작하지 않고도 소프트웨어적으로 마우스를 다시 연결합니다.

gpm을 사용하지 않거나 무언가의 이유로 설치하지 않았다면, X를 설정할 때 /dev/psaux같은 마우스 장치에서 직접 신호를 읽어들이도록 하십시오. 자세한 사항은 /usr/share/doc/HOWTO/en-txt/mini/3-Button-Mouse.gz의 3 단추 마우스 미니 하우투, man gpm, /usr/share/doc/gpm/FAQ.gz, README.mouse <<http://www.xfree86.org/current/mouse.html>>를 참고하십시오.

C.3 작업마다 필요한 디스크 공간

기본 2.4 커널을 사용해 i386에서 기본 설치를 하려면 표준 꾸러미를 모두 합쳐서 573MB의 디스크 공간을 차지합니다.

다음 표는 aptitude에서 표시하는 값으로 tasksel에 들어 있는 작업에 필요한 크기입니다. 작업 중에는 겹치는 부분이 있기 때문에 두 개의 작업을 같이 설치하면 숫자를 합친 전체 크기보다는 작을 수도 있습니다.

파티션의 크기를 결정할 때, 베이스를 설치했을 때의 크기에다가 다음 표에 나온 크기를 더해야 합니다. “설치 크기”에 들어 있는 크기의 대부분은 /usr에 해당합니다. “내려받기 크기”는 (일시적으로) /var에 필요합니다.

작업	설치 크기 (MB)	다운로드 크기 (MB)	설치하는데 필요한 공간 (MB)
데스크탑	1392	460	1852
웹 서버	36	12	48
인쇄 서버	168	58	226
DNS 서버	2	1	3
파일 서버	47	24	71
메일 서버	10	3	13
SQL 데이터베이스	66	21	87

참고

Desktop 작업은 GNOME과 KDE 데스크탑 환경을 모두 설치합니다.

영어가 아닌 언어로 설치한다면 `tasksel`에서 자동으로 지역화 작업을 (해당 언어에 대한 작업이 있다면) 설치합니다. 언어마다 필요한 공간이 다릅니다. 내려받고 설치하는데 최대 200MB까지의 공간이 있어야 합니다.

C.4 유닉스/리눅스 시스템에서 Debian GNU/Linux 설치하기

이 부분에서는 이 안내서의 다른 부분에 설명되어 있는 ncurses 기반 메뉴 방식 설치 프로그램을 사용하지 않고, 기존 유닉스 혹은 리눅스 시스템을 이용해 Debian GNU/Linux 배포판을 설치하는 방법을 설명합니다. 이 “크로스 설치” HOWTO는 Red Hat, Mandrake, SUSE에서 Debian GNU/Linux 배포판으로 바꾸려는 사용자들에게 필요한 부분입니다. 여기서는 유닉스 계열의 명령어를 입력하는 방법이나 파일 시스템을 돌아다니는 데 익숙해져 있다고 가정합니다. 여기서 `$` 표시는 사용자의 기존 시스템에서 입력하는 명령어이고, `#` 표시는 데비안 chroot에서 입력하는 명령어를 말합니다.

새로 설치한 데비안 시스템을 필요에 맞게 설정하기만 하면, 기존 사용자 데이터를 (있다면) 옮겨와서 계속 사용할 수 있습니다. 즉 “다운타임이 없는” Debian GNU/Linux 설치입니다. 또 이 방법은 여러 가지 부팅 미디어나 설치 미디어들이 동작하지 않는 하드웨어에 설치하는 쉬운 방법이기도 합니다.

C.4.1 시작하기

기존 유닉스용 파티션 도구를 이용해 하드 드라이브를 필요한 대로 다시 파티션하십시오. 최소한 파일 시스템 한 개를 스왑으로 만드십시오. 콘솔만 설치하는 경우는 최소 150MB의 공간이 필요하고 X를 설치한다면 최소 300MB가 필요합니다.

파티션에 파일 시스템을 만드려면 다음과 같이 하십시오. 예를 들어 `/dev/hda6` 파티션에 `ext3` 파일 시스템을 만드는 경우라면 (여기 예제에서 루트 파티션입니다):

```
# mke2fs -j /dev/hda6
```

ext2 파일 시스템을 만드는 경우라면 -j 옵션을 빼십시오.

스왑을 초기화하고 활성화하십시오 (파티션을 사용하려는 데비안 스왑 파티션으로 바꾸십시오):

```
# mkswap /dev/hda5
# sync; sync; sync
# swapon /dev/hda5
```

한 파티션을 `/mnt/debinst`에 (루트 (/) 파일시스템을 설치할 위치) 마운트하십시오. 마운트 위치 이름은 마음대로 정한 것이고, 아래에서 계속 사용합니다.

```
# mkdir /mnt/debinst
# mount /dev/hda6 /mnt/debinst
```

참고

파일시스템의 (예를 들어 `/usr`) 일부를 별도의 파티션에 마운트하려면, 다음 단계로 넘어가기 전에 그 디렉토리를 수동으로 만들어서 마운트해야 합니다.

C.4.2 debootstrap 설치

데비안 설치 프로그램이 사용하는 도구로, 데비안 베이스 시스템을 설치하는 공식적인 방법은 **debootstrap**입니다. **debootstrap**은 **wget**과 **ar**을 사용하지만, 그 밖에는 `/bin/sh`만 있으면 동작합니다. 기존 시스템에 **wget**과 **ar**이 없으면 설치하십시오. 그리고 **debootstrap**을 내려 받고 설치하십시오.

RPM 기반 시스템이라면, **alien**으로 `.deb` 파일을 `.rpm` 파일로 변환하거나 <<http://people.debian.org/~blade/install/debootstrap>>에 있는 RPM 버전을 내려 받으십시오.

아니면, 다음 방법을 이용해 수동으로 설치할 수도 있습니다. 먼저 `.deb` 파일을 풀 작업용 폴더를 만드십시오:

```
# mkdir work
```

```
# cd work
```

바이너리 파일은 데비안 아카이브에 있습니다. (아키텍처에 맞는 파일을 선택하도록 하십시오) 꾸러미 풀 <<http://ftp.debian.org/debian/pool/main/d/debootstrap/>>에서 **debootstrap** .deb 파일을 내려 받으시고, 꾸러미를 작업용 폴더에 복사하고, 바이너리 파일을 풀어 내십시오. 이 바이너리를 설치하려면 루트 권한이 필요할 것입니다.

```
# ar -x debootstrap_0.X.X_아키텍처.deb
# cd /
# zcat /full-path-to-work/work/data.tar.gz | tar xv
```

debootstrap을 실행하려면 **glibc**의 최소한의 버전은 (현재 GLIBC_2.3) 있어야 합니다. **debootstrap** 자체는 쉘 스크립트이지만, 이 스크립트에서 **glibc**가 있어야 하는 여러 가지 유틸리티를 실행합니다.

C.4.3 debootstrap 실행

debootstrap을 실행하면 필요한 파일들을 아카이브에서 직접 내려받습니다. 아래의 명령어 예제에서 [http.us.debian.org/debian](http://us.debian.org/debian)을 아무 데비안 아카이브 미러로 바꿀 수 있습니다. 네트워크에서 가까이 있는 미러로 하는 게 좋습니다. 미러 목록은 <<http://www.debian.org/misc/README.mirror>>에 있습니다.

sarge Debian GNU/Linux CD를 **/cdrom**에 마운트했다면 http URL 대신에 file URL을 쓸 수 있습니다: `file:/cdrom/debian/`

debootstrap 명령에서 *ARCH*를 다음 중의 하나로 바꾸십시오: alpha, arm, hppa, i386, ia64, m68k, mips, mipsel, powerpc, s390, sparc.

```
# /usr/sbin/debootstrap --arch ARCH sarge \
    /mnt/debinst http://http.us.debian.org/debian
```

C.4.4 베이스 시스템 설정

이제 디스크에 작지만 진짜 데비안 시스템이 들어 있습니다. 이 데비안 시스템으로 **chroot**하십시오:

```
# LANG= chroot /mnt/debinst /bin/bash
```

C.4.4.1 파티션 마운트하기

`/etc/fstab`을 만들어야 합니다.

```
# editor /etc/fstab
```

여기에 있는 예제를 필요에 맞게 수정하면 됩니다.

```
# /etc/fstab: static file system information.
#
# file system      mount point      type      options                    dump pass
/dev/XXX           /                ext3      defaults                    0    1
/dev/XXX           /boot            ext3      ro,nosuid,nodev            0    2

/dev/XXX           none             swap      sw                           0    0
proc               /proc            proc      defaults                    0    0

/dev/fd0           /mnt/floppy      auto      noauto,rw,exec,usersync    0    0
/dev/cdrom         /mnt/cdrom       iso9660   noauto,ro,user,exec        0    0

/dev/XXX           /tmp             ext3      rw,nosuid,nodev            0    2
/dev/XXX           /var             ext3      rw,nosuid,nodev            0    2
/dev/XXX           /usr             ext3      rw,nodev                    0    2
/dev/XXX           /home            ext3      rw,nosuid,nodev            0    2
```

`mount -a` 명령으로 `/etc/fstab`에 지정한 파일시스템을 모두 마운트하십시오. 아니면 다음 명령으로 파일시스템을 하나하나 마운트하십시오:

```
# mount /path # 예들 들어: mount /usr
```

`proc` 파일시스템은 여러번 어느 위치에서든 마운트할 수 있습니다. (하지만 `/proc`을 관행적으로 사용합니다.) `mount -a` 명령을 사용하지 않았다면, 계속하기 전에 꼭 `proc`을 마운트하십시오.

```
# mount -t proc proc /proc
```

`ls /proc` 명령을 실행하면 여러 파일이 들어 있는 디렉토리 내용을 표시합니다. 이 명령이 실패하면 `chroot` 바깥에서 `proc`을 마운트할 수 있습니다.

```
# mount -t proc proc /mnt/debinst/proc
```

C.4.4.2 키보드 설정

키보드를 설정하려면:

```
# dpkg-reconfigure console-data
```

키보드는 chroot 안에 있을 때 설정할 수 없다는 데 유의하십시오. 다음에 다시 시작한 다음에 설정합니다.

C.4.4.3 네트워크 설정하기

네트워크를 설정하려면, `/etc/network/interfaces`, `/etc/resolv.conf`, `/etc/hostname` 파일을 편집하십시오.

```
# editor /etc/network/interfaces
```

다음은 `/usr/share/doc/ifupdown/examples`에 들어 있는 예제입니다:

```
#####  
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)  
# See the interfaces(5) manpage for information on what options are  
# available.  
#####  
  
# We always want the loopback interface.  
#  
auto lo  
iface lo inet loopback  
  
# To use dhcp:  
#  
# auto eth0  
# iface eth0 inet dhcp  
  
# An example static IP setup: (broadcast and gateway are optional)  
#  
# auto eth0  
# iface eth0 inet static  
#     address 192.168.0.42  
#     network 192.168.0.0
```

```
# netmask 255.255.255.0
# broadcast 192.168.0.255
# gateway 192.168.0.1
```

네임서버와 search 명령을 `/etc/resolv.conf`에 입력하십시오:

```
# editor /etc/resolv.conf
```

간단한 `/etc/resolv.conf`:

```
search hqdom.local\000
nameserver 10.1.1.36
nameserver 192.168.9.100
```

시스템의 호스트이름을 입력하십시오 (2글자에서 63글자까지):

```
# echo DebianHostName > /etc/hostname
```

네트워크 카드가 여러개라면, `/etc/modules`에 드라이버 모듈 이름을 적당한 순서로 맞춰야 합니다. 그래야 부팅할 때 각 카드가 의도한 해당 인터페이스 이름으로 (eth0, eth1 등) 연결됩니다.

C.4.4.4 시간대, 사용자, APT 설정하기

다음 명령으로 시간대를 설정하고, 일반 사용자를 추가하고, **apt** 소스를 선택하십시오

```
# /usr/sbin/base-config new
```

C.4.4.5 로케일 설정하기

영어가 아닌 언어를 사용할 때 로케일을 설정하려면 로케일 지원 꾸러미를 설치하고 그 꾸러미를 설정하십시오:

```
# apt-get install locales
```

```
# dpkg-reconfigure locales
```

주의: 이 명령을 실행하기 전에 (즉 base-config 단계 전에) apt를 설정해야 합니다. ASCII 혹은 latin1 이외의 문자셋을 사용하는 로케일을 사용하려면 그 전에 적당한 지역화 HOWTO를 참고하십시오.

C.4.5 커널 설치

이 시스템을 부팅하려면, 리눅스 커널과 부트로더가 있어야 합니다. 다음 명령으로 미리 꾸러미로 만들어져 있는 커널을 알아보십시오:

```
# apt-cache search kernel-image
```

그리고 꾸러미 이름을 하나 골라서 설치하십시오.

```
# apt-get install kernel-image-2.X.X-아키텍처-딱워
```

C.4.6 부트로더 설정하기

Debian GNU/Linux 시스템을 부팅 가능하게 하려면, 부트로더를 설정해서 루트 파티션에서 설치한 커널을 읽어들이도록 하십시오. debootstrap은 부트로더를 설치하지 않는다는 점에 주의하십시오. 하지만 apt-get을 데비안 chroot 안에서 사용하면 부트로더를 설치할 수 있습니다.

info grub 혹은 man lilo.conf 명령으로 부트로더 설정 방법을 미리 알아보십시오. 데비안을 설치할 때 사용했던 기존 시스템을 계속 유지하는 경우라면, 기존의 grub menu.lst 혹은 lilo.conf에 데비안에 해당하는 항목을 추가하십시오. lilo.conf의 경우, 파일을 새로운 시스템으로 복사해 와서 편집할 수도 있습니다. 편집을 마친 다음에는 lilo를 실행하십시오. (lilo를 실행하면 실행하는 시스템에 상대적인 lilo.conf 파일을 사용하게 됩니다.)

다음은 기본적인 /etc/lilo.conf 예제입니다:

```
boot=/dev/hda6
root=/dev/hda6
install=menu
delay=20
lba32
image=/vmlinuz
label=Debian
```

C.5 병렬 라인 IP를 (PLIP) 이용해 Debian GNU/Linux 설치하기

이 부분은 이더넷 카드가 없지만, 게이트웨이 컴퓨터에 널 모뎀 케이블을 (널 프린터 케이블이라고도 합니다) 통해 연결되어 있는 컴퓨터에 Debian GNU/Linux을 설치하는 방법을 설명합니다. 게이트웨이 컴퓨터는 데비안 머리가 있는 네트워크에 (예를 들어 인터넷에) 연결되어 있어야 합니다.

이 부록의 예제에서는 전화접속 연결을 통해 (ppp0) 인터넷에 연결된 게이트웨이와 PLIP 연결을 합니다. 타겟 컴퓨터와 소스 컴퓨터 각각에 IP 주소로 192.168.0.1 및 192.168.0.2를 사용합니다. (게이트웨이에 연결되어 있는 네트워크에서 이 IP 주소는 사용하지 말아야 합니다.)

설치한 시스템으로 다시 시작한 다음에도 설치할 때 사용한 PLIP 연결을 설정할 수 있습니다. (7장 참고)

시작하기 전에, 소스와 타겟 컴퓨터 모두에 대해 BIOS 설정의 병렬 포트 부분을 (IO 베이스 주소 및 IRQ) 확인해야 합니다. 가장 많이 쓰이는 값은 `io=0x378`, `irq=7`입니다.

C.5.1 요구 사항

- 데비안을 설치할 타겟이라고 하는 타겟 컴퓨터.
- 시스템 설치 미디어. 2.2절 참고.
- 인터넷에 연결된 소스라고 하는 컴퓨터, 게이트웨이로 동작합니다.
- DB-25 널모뎀 케이블. 이 케이블에 대한 정보 및 케이블을 직접 만드는 방법은 PLIP-Install-HOWTO <<http://www.tldp.org/HOWTO/PLIP-Install-HOWTO.html>>를 참고하십시오.

C.5.2 소스 설정하기

다음 셸 스크립트는 소스 컴퓨터를 ppp0를 사용한 인터넷 게이트웨이로 설정하는 예제입니다.

```
#!/bin/sh

# 커널에서 실행중인 모듈을 제거해서 충돌을 방지하고 다시
# 수동으로 설정합니다.
modprobe -r lp parport_pc
modprobe parport_pc io=0x378 irq=7
modprobe plip

# plip 인터페이스를 설정합니다 (이 경우는 plip0, dmesg | grep plip 명령으로 확인하십시오)
ifconfig plip0 192.168.0.2 pointopoint 192.168.0.1 netmask 255.255.255.255 up

# gateway 설정
modprobe iptable_nat
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
echo 1
```

```
> /proc/sys/net/ipv4/ip_forward
```

C.5.3 타겟 설치

설치 미디어로 부팅하십시오. 설치하는 전문가 모드로 해야 합니다. 부팅 프롬프트에서 `expert`를 입력하십시오. 아래는 설치 단계에서 입력해야 할 사항들입니다.

1. 설치 프로그램 구성 요소 읽어들이기

목록에서 `plip-modules` 옵션을 선택하십시오. 그러면 설치 시스템에서 PLIP 드라이버를 사용할 수 있습니다.

2. 네트워크 하드웨어 검색

- 만약 타겟에 네트워크 카드가 있으면, 검색한 드라이버 모듈의 목록이 표시됩니다. **debian-installer**에서 `plip`을 강제로 사용하려면 목록에 있는 드라이버 모듈을 모두 선택 해제하십시오. 물론 타겟에 네트워크 카드가 없으면 이 목록은 표시하지 않습니다.
- 모듈 파라미터에 대한 프롬프트: 예
- 앞에서 네트워크 카드를 찾지/검색하지 못했으므로, 목록에서 네트워크 드라이버의 모듈을 설치합니다. `plip` 모듈을 선택하십시오.
- `parport_pc` 모듈의 추가 파라미터: `io=0x378 irq=7`
- `plip` 모듈의 추가 파라미터: 비워둡니다

3. 네트워크 설정

- DHCP로 네트워크 자동 설정: 아니오
- IP 주소: `192.168.0.1`
- Point-to-point 주소: `192.168.0.2`
- 네임서버 주소: 소스에서 사용한 같은 주소를 입력합니다. (`/etc/resolv.conf` 파일 참고)

Appendix D

관리

D.1 문서 정보

이 설명서는 사지의 데비안 설치 프로그램을 위해 만들어졌으며 이는 부트-플로피를 위한 우디 설치 설명서에 기반하고 있습니다. 또한 기반 설명서는 이전 데비안 설치 설명서와 2003년에 발표된 Progeny 배포본 설명서에 기반합니다.

이 문서는 DocBook XML로 쓰여졌습니다. 출력 형식은 **docbook-xml**과 **docbook-xsl** 패키지에 있는 정보를 이용하는 다양한 프로그램에 의해 생성됩니다.

이 문서의 유지보수성을 증가시키기 위하여 엔티티와 프로파일 속성과 같은 많은 XML 특성을 사용합니다. 이들은 프로그래밍 언어의 변수와 상수들과 유사한 역할을 합니다. 이 문서의 XML소스는 아키텍처 특정한 몇몇의 텍스트를 분리하는데 사용하는 각기 다른 아키텍처 — 프로파일링 속성에 관한 정보를 담고 있습니다.

D.2 이 문서에 기여하기

이 문서와 관련된 문제점이나 제안이 있다면 **debian-installer-manual**패키지를 이용한 버그리포트 를 우리에게 보내주십시오. **reportbug** 패키지나 Debian Bug Tracking System <<http://bugs.debian.org/>>의 온라인 문서를 보십시오. 당신의 문제점이 이미 보고되었는지를 보기 위해 open bugs against debian-installer-manual <<http://bugs.debian.org/debian-installer-manual>>를 검토해 보시는 것도 좋습니다. 그러면 <XXXX@bugs.debian.org>으로 추가적인 확증이나 유익한 정보를 제공할 수 있습니다. 여기서 XXXX는 이미 보고된 버그의 숫자입니다.

이 문서의 Docbook소스의 사본을 얻어 패치를 생성합니다. Docbook 소스는 debian-installer WebSVN <<http://svn.debian.org/wsvn/d-i/>>에서 발견할 수 있습니다. Docbook이 익숙하지 않더라도 걱정하지 마십시오. 메뉴얼 디렉토리에 간단한 쪽지가 있습니다. HTML와 마찬가지로 화면표시보다 텍스트의 의미에 중점을 두고 있습니다.패치는 아래 나와있는 debian-boot 메일링 리스트로 보내주십시오. SVN을 이용하여 소스를 가져오는 방법에 대한 설명은 소스가 있는 최상위디렉토리에서 README <<http://svn.debian.org/wsvn/d-i/README?op=file>>를 보십시오.

절대로 이 문서의 저자에게 직접 접촉하지 마십시오. 이 메뉴얼에 대한 토론을 하고 있는 **debian-installer** 토론 리스트가 있습니다. 이 메일링 리스트는 <debian-boot@lists.debian.org>입니다. 이 리스트에 가입하는 절차는 Debian Mailing List Subscription <<http://www.debian.org/MailingLists/subscribe>> 페이지에서 볼 수 있으며, Debian Mailing List Archives <<http://lists.debian.org/>>를 온라인으로 볼 수 있습니다.

D.3 중요 기부자들

이 문서는 설치 하우투를 작성한 Bruce Perens, Sven Rudolph, Igor Grobman, James Treacy, and Adam Di Carlo. Sebastian Ley가 작성하였습니다. 매우 많은 데비안 사용자와 개발자가 이 문서에 기여를 하였습니다. 특수한 노트는 Michael Schmitz (m68k support), Frank Neumann (Amiga install manual <http://www.informatik.uni-oldenburg.de/~amigo/debian_inst.html>의 원래 저자), Arto Astala, Eric Delaunay/Ben Collins (SPARC information), Tapio Lehtonen, and Stéphane Bortzmeyer가 만들어야 하였습니다. 우리는 USB메모리 스틱에서 부팅하는 것에 대한 유용한 정보를 보내준 Pascal Le Bail 에 감사합니다. Miroslav Kuře는 Sarge의 데비안 설치자의 많은 새로운 기능성에 대한 문서화를 해주었습니다.

가장 유용한 텍스트와 정보는 Jim Mintha의 네트워크 부팅에 관한 HOWTO문서(URL없음)와 the Debian FAQ <<http://www.debian.org/doc/FAQ/>>, the Linux/m68k FAQ <<http://www.linux-m68k.org/faq/faq.html>>, the Linux for SPARC Processors FAQ <<http://www.ultralinux.org/faq.html>>, the Linux/Alpha FAQ <<http://linux.iol.unh.edu/linux/alpha/faq/>>에서 발견할 수 있습니다. 이 자유롭게 사용가능하고 많은 정보 소스의 메인테이너는 인정될 필요가 있습니다.

이 메뉴얼에서 격리된 설치에 관한 장C.4절은 Karsten M. Self가 저작권을 가진 문서에서 가져왔습니다.

이 메뉴얼에서 PLIP 기반 설치에 관한 장C.5절은 Gilles Lamiral이 쓴 PLIP-Install-HOWTO <<http://www.tldp.org/HOWTO/PLIP-Install-HOWTO.html>>에 기반합니다.

D.4 상표권 표시

모든 상표는 각각의 상표 소유자의 자산입니다.

Appendix E

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. — 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

E.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the gnu General Public License is intended to guarantee your freedom to share and change free software — to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the gnu Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

E.2 GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

- This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

- You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

- You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

- You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation

and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

- You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
- Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
- If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to

decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

- If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
- The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
- If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

- because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. the entire risk as to the quality and performance of the program is with you. should the program prove defective, you assume the cost of all necessary servicing, repair or correction.
- in no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

END OF TERMS AND CONDITIONS

E.3 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and a brief idea of what it does.

Copyright (C) year name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with absolutely no warranty; for details type 'show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items — whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the gnu Library General Public License instead of this License.