

데비안 GNU/리눅스설치안내서

April 22, 2017

데비안 GNU/리눅스설치안내서

저작권 © 2004 - 2017 데비안설치프로그램팀

이설명서는자유소프트웨어입니다. GNU General Public License 에따라이문서를재배포할수있고문서
내용을바꿀수있습니다. 부록 F의라이선스를참고하십시오.

Contents

1	데비안에 오신것을 환영합니다	1
1.1	데비안이란?	1
1.2	GNU/리눅스란?	2
1.3	데비안 GNU/리눅스란?	2
1.4	데비안 GNU/kFreeBSD 란?	3
1.5	데비안 GNU/Hurd 란?	3
1.6	데비안받기	3
1.7	이문서의최신버전구하는법	3
1.8	이문서의구성	3
1.9	문서에도움을주실분은언제든지환영합니다	4
1.10	저작권및소프트웨어라이선스정보	4
2	시스템요구사항	6
2.1	지원하는하드웨어	6
2.1.1	지원하는아키텍처	6
2.1.2	3 가지 ARM 포트	7
2.1.3	ARM CPU 설계의다양함과복잡한지원	7
2.1.4	Debian/arm64 가지원하는플랫폼	7
2.1.4.1	기타플랫폼	8
2.1.5	다중프로세서	8
2.1.6	그래픽하드웨어지원	8
2.1.7	네트워크연결하드웨어	8
2.1.8	주변장치및기타하드웨어	8
2.2	펌웨어가필요한장치	8
2.3	GNU/Linux 에적합한하드웨어구입	9
2.3.1	독점적이거나폐쇄된하드웨어피하기	9
2.4	설치미디어	9
2.4.1	CD-ROM/DVD-ROM/BD-ROM	9
2.4.2	네트워크	10
2.4.3	하드디스크	10
2.4.4	유닉스계열혹은 GNU 시스템	10
2.4.5	지원하는저장장치	10
2.5	메모리및디스크공간요구사항	10
3	데비안 GNU/리눅스를설치하기전에	11
3.1	설치과정개요	11
3.2	기존데이터를백업하십시오!	12
3.3	필요한정보	12
3.3.1	문서	12
3.3.1.1	설치안내서	12
3.3.1.2	하드웨어문서	12
3.3.2	하드웨어정보가있는곳찾기	12
3.3.3	하드웨어호환성	13
3.3.3.1	라이브시스템에서하드웨어호환성검사하기	13
3.3.4	네트워크설정	14
3.4	최소하드웨어요구사항맞추기	14
3.5	멀티부팅시스템에서미리파티션하기	15
3.6	설치하기전에할하드웨어및운영체제설정	15
3.6.1	ARM 펌웨어	15
3.6.2	U-Boot 에서이더넷 MAC 주소설정하기	15
3.6.3	U-Boot 의커널/최초램디스크/디바이스트리재배치문제	16

4	시스템설치미디어구하기	17
4.1	공식데비안 GNU/리눅스 CD/DVD-ROM 세트	17
4.2	데비안미러에서파일다운로드	17
4.2.1	설치이미지를찾을위치	17
4.3	TFTP 네트워크부팅에필요한파일준비하기	17
4.3.1	RARP 서버준비하기	18
4.3.2	DHCP 서버준비하기	18
4.3.3	BOOTP 서버준비하기	18
4.3.4	TFTP 서버사용하기	19
4.3.5	TFTP 이미지를적당한위치에놓기	19
4.4	자동설치	19
4.4.1	데비안설치프로그램을이용한자동설치	19
5	설치시스템부팅하기	20
5.1	64-bit ARM 에서설치프로그램부팅하기	20
5.1.1	콘솔설정	20
5.1.2	Juno 설치	20
5.1.3	Applied Micro Mustang 설치	20
5.1.4	TFTP 로부팅하기	21
5.1.4.1	U-Boot 에서 TFTP 부팅	21
5.2	접근성	22
5.2.1	설치 프로그램 프론트엔드	22
5.2.2	보드장치	22
5.2.3	고대비테마	22
5.2.4	화면확대	22
5.2.5	Expert install, rescue mode, automated install	22
5.2.6	설치한시스템의접근성	22
5.3	부팅파라미터	23
5.3.1	부팅콘솔	23
5.3.2	데비안설치프로그램파라미터	23
5.3.3	부팅파라미터로질문에답하기	25
5.3.4	커널모듈에파라미터넘기기	25
5.3.5	커널모듈블랙리스트	26
5.4	설치과정의문제해결	26
5.4.1	CD-ROM 신뢰성	26
5.4.1.1	공통사항	26
5.4.1.2	문제점파악및해결방법	26
5.4.2	부팅설정	28
5.4.3	커널시작메시지해석하기	28
5.4.4	설치문제보고하기	28
5.4.5	설치보고제출	28
6	데비안설치프로그램사용하기	30
6.1	설치프로그램이동작하는방식	30
6.1.1	그래픽설치프로그램사용하기	30
6.2	컴포넌트소개	31
6.3	컴포넌트사용하기	32
6.3.1	데비안설치프로그램준비및하드웨어설정	32
6.3.1.1	사용가능메모리검사 / 저용량메모리모드	32
6.3.1.2	지역화옵션선택	33
6.3.1.3	키보드선택하기	33
6.3.1.4	데비안설치프로그램 ISO 이미지찾기	34
6.3.1.5	네트워크설정하기	34
6.3.1.5.1	네트워크자동설정	34
6.3.1.5.2	네트워크수동설정	34
6.3.1.5.3	IPv4 및 IPv6	34
6.3.1.6	시계및시간설정	35
6.3.2	사용자및암호설정	35
6.3.2.1	루트암호설정	35

6.3.2.2	일반사용자만들기	35
6.3.3	파티션하기및마운트위치선택	35
6.3.3.1	지원하는파티션옵션	36
6.3.3.2	자동파티션하기	36
6.3.3.3	수동파티션하기	38
6.3.3.4	멀티디스크장치설정하기 (소프트웨어 RAID)	38
6.3.3.5	LVM (논리볼륨관리자) 설정하기	40
6.3.3.6	암호화볼륨설정하기	41
6.3.4	베이스시스템설치하기	42
6.3.5	추가소프트웨어설치하기	43
6.3.5.1	APT 설정하기	43
6.3.5.1.1	여러개의 CD/DVD 에서설치하기	43
6.3.5.1.2	네트워크미러사용하기	44
6.3.5.1.3	네트워크미러고르기	44
6.3.5.2	소프트웨어선택및설치	44
6.3.6	시스템을부팅가능하게만들기	45
6.3.6.1	다른운영체제검색	46
6.3.6.2	시스템을 flash-kernel 로부팅가능하게만들기	46
6.3.6.3	부트로더없이계속	46
6.3.7	설치마치기	46
6.3.7.1	시스템시계맞추기	46
6.3.7.2	시스템다시시작	46
6.3.8	문제해결	46
6.3.8.1	설치로그저장	47
6.3.8.2	셸사용하기및로그보기	47
6.3.9	네트워크를통해설치	47
6.4	없는펌웨어읽어들이기	48
6.4.1	미디어준비하기	49
6.4.2	펌웨어및설치한시스템	49
7	새로운데비안시스템으로부팅하기	50
7.1	진실의시간	50
7.2	암호화볼륨마운트하기	50
7.2.1	문제해결	50
7.3	로그인	51
8	다음단계및그다음에할일	52
8.1	시스템끄기	52
8.2	유닉스를처음접한다면	52
8.3	데비안에익숙해지기	52
8.3.1	데비안패키지시스템	52
8.3.2	데비안용추가소프트웨어	53
8.3.3	프로그램버전관리	53
8.3.4	CRON 작업관리	53
8.4	그외의읽을거리및정보	53
8.5	시스템에전자메일준비하기	53
8.5.1	기본전자메일설정	54
8.5.2	시스템외부에전자메일보내기	54
8.5.3	Exim4 MTA 설정하기	54
8.6	새커널컴파일하기	55
8.6.1	커널이미지관리	55
8.7	손상된시스템복구하기	56
A	설치방법	57
A.1	들어가기전에	57
A.2	설치프로그램시작하기	57
A.2.1	CDROM	57
A.2.2	네트워크부팅	57
A.2.3	하드디스크부팅	57

A.3	설치	58
A.4	설치보고서를보내주십시오	58
A.5	그리고마지막으로...	58
B	미리설정을이용한설치자동화	59
B.1	소개	59
B.1.1	미리설정방법	59
B.1.2	한계	60
B.2	미리설정사용하기	60
B.2.1	미리설정파일읽어들이기	60
B.2.2	부팅파라미터로미리설정하기	61
B.2.3	자동모드	61
B.2.4	미리설정할때쓸모있는줄임말	62
B.2.5	미리설정파일을지정하는데 DHCP 서버사용하기	63
B.3	미리설정파일만들기	63
B.4	미리설정파일의내용 (stretch 용)	64
B.4.1	지역화	64
B.4.2	네트워크설정	65
B.4.3	네트워크콘솔	66
B.4.4	미러사이트설정	66
B.4.5	계정설정	67
B.4.6	시계및시간대설정	68
B.4.7	파티션하기	68
B.4.7.1	파티션예제	68
B.4.7.2	RAID 를사용해파티션하기	69
B.4.7.3	파티션마운트방법조정하기	70
B.4.8	기본시스템설치	71
B.4.9	APT 설정	71
B.4.10	패키지선택	72
B.4.11	설치마치기	72
B.4.12	기타패키지미리설정	73
B.5	고급옵션	73
B.5.1	설치할때임의의명령어실행하기	73
B.5.2	미리설정을이용해기본값바꾸기	73
B.5.3	미리설정파일을분리해서사용하기	74
C	데비안에서파티션나누기	75
C.1	데비안파티션및크기정하기	75
C.2	디렉터리구조	75
C.3	권장하는파티션구조	76
C.4	리눅스의장치이름	77
C.5	데비안의파티션프로그램	77
D	여러가지내용	78
D.1	리눅스장치	78
D.1.1	마우스설정하기	78
D.2	태스크마다필요한디스크공간	79
D.3	유닉스/리눅스시스템에서데비안 GNU/리눅스설치하기	80
D.3.1	시작하기	80
D.3.2	debootstrap 설치	80
D.3.3	debootstrap 실행	81
D.3.4	베이스시스템설정	81
D.3.4.1	장치파일만들기	81
D.3.4.2	파티션마운트하기	82
D.3.4.3	시간대설정하기	83
D.3.4.4	네트워크설정하기	83
D.3.4.5	APT 설정하기	84
D.3.4.6	로컬및키보드설정하기	84
D.3.5	커널설치	84

D.3.6	부트로더설정하기	84
D.3.7	원격접근: SSH 설치및접근설정	85
D.3.8	마지막처리	85
D.4	PPP 오버이더넷을 (PPPoE) 이용해데비안 GNU/리눅스설치하기	85
E	문서관리정보	87
E.1	문서정보	87
E.2	이문서에참여하기	87
E.3	중요기여자들	87
E.4	상표권안내	88
F	GNU 일반공중사용허가서	89
F.1	전문	89
F.2	GNU 일반공중사용허가서 (GNU GENERAL PUBLIC LICENSE)	90
F.3	새로운프로그램에 GPL 을적용하는방법	92

List of Tables

3	데비안 GNU/리눅스를설치하기전에	
3.1	설치에도움이되는하드웨어정보	13
3.2	추천하는최소시스템요구사항	14

Abstract

이문서는 64-bit ARM (“arm64”) 아키텍처데비안 GNU/리눅스 9 시스템 (코드명 “stretch”) 설치설명서입니다. 또한더자세한정보에대한포인터와새로데비안시스템을구축하는방법에대해서도언급하고 있습니다.

주의



이설치안내서는예전의데비안설치시스템 (“boot-floppies”) 용으로쓰여진예전 설명서를기반으로새로운데비안설치프로그램에관한내용을추가하였습니다. 그러나 arm64 용의경우에는완전히업데이트하지않았고새설치프로그램에서확인되지않은것도있습니다. 이설명서는각부분부분에따라서미완성인부분도있고, 예전내용을담고있는것도있고, 아직도 boot-floppies 설치프로그램에관한내용만써놓은부분도있습니다. 이설명서의최신버전은 (아마도이아키텍처에대해서더문서화가잘되어있는) 인터넷의 [debian-installer 홈페이지](#)에있습니다. 이홈페이지에는다른여러언어의번역판도더있습니다.

이문서에는아직부족한부분이많이남아있습니다. 번역한내용중에어색한부분이나잘못번역된부분이 있다면데비안한국어메일링리스트에 (debian-l10n-korean@lists.debian.org) 메일을보내주시면반영하겠습니다.

arm64 용 데비안 GNU/리눅스 9 설치하기

데비안을 사용해 주셔서 감사합니다. 사용하게 되시면 데비안의 GNU/Linux 배포판이 무일무이하다는 걸 알게 되실 겁니다. 데비안 GNU/리눅스는 전 세계 최고 품질의 자유 소프트웨어를 모아서, 일관적으로 통합되어 있습니다. 이렇게 모아 놓은 배포판이 개별 소프트웨어 이상의 힘을 발휘한다고 저희는 생각합니다.

많은 분이 설명서를 읽지 않고 데비안을 설치하려 할 겁니다. 또한 그것이 가능하도록 데비안 설치할 수 있도록 설계되어 있습니다. 설치 안내서를 모두 읽을 시간이 없으면 설치 Howto(기본 설치 과정을 안내합니다)와 추가 정보를 찾아볼 경우나 문제가 생겼을 경우에 대한 설명서 링크를 읽는 것이 좋습니다. 설치 Howto는 부록 **A**에 있습니다.

그렇긴 하지만 시간을 내서 이 안내서 대부분을 읽어 보기 바랍니다. 그러면 더 많이 알고 있는 상태로 아마도 더 성공적으로 설치할 수 있을 겁니다.

Chapter 1

데비안에 오신것을 환영합니다

이장에서는 데비안 프로젝트와 데비안 GNU/리눅스에 대해 간단히 설명합니다. 데비안 프로젝트의 역사와 데비안 GNU/리눅스 배포판에 대해 이미 알고 있다면 다음 장으로 넘어가셔도 됩니다.

1.1 데비안이란?

데비안은 자유 소프트웨어를 개발하고 자유 소프트웨어 커뮤니티의 이상을 널리 알리는 일을 위해 조직한, 자원자로만 구성된 조직입니다. 데비안 프로젝트는 1993년에 이안머독이 새로운 (당시에는 비교적 새로운 버전의) 커널을 사용하면서, 완전하고 일관된 소프트웨어 배포판을 만드는 데 참여할 소프트웨어 개발자를 공개적으로 모집하면서부터 시작되었습니다. 처음에는 **자유 소프트웨어 재단 (Free Software Foundation)**의 자금 지원을 받았고, **GNU** 철학에 영향을 받은 비교적 작은 규모의 이열성적인 집단은, 해를 거듭하면서 1062여명의 데비안 개발자가 참여하는 조직으로 성장했습니다.

데비안 개발자는 **홈페이지** 및 **FTP** 사이트 관리, 그래픽 디자인, 소프트웨어 라이선스의 법률적 해석, 문서 작성, (말할 필요도 없이) 소프트웨어 패키지 관리 등 다양한 활동에 참여합니다.

데비안의 철학을 전달하려고, 또 데비안에 동의하는 개발자를 끌어모으려고 데비안 프로젝트는 여러 가지 문서를 발표했습니다. 이 문서에서 데비안의 가치를 간략히 설명하고, 또 이 문서는 데비안 개발자가 되는 게 무엇을 뜻하는지에 대한 지침 역할을 합니다:

- **데비안 우리의 약속 (Debian Social Contract)**은 자유 소프트웨어 공동체에 대한 데비안의 약속입니다. 우리의 약속에 따르기로 동의한 사람은 누구나 **메인테이너**가 될 수 있습니다. 어떤 메인테이너든지 새로운 소프트웨어를 데비안에 추가할 수 있습니다. 단 그 소프트웨어가 우리의 자유 소프트웨어 기준에 맞아야 하고, 패키지가 우리의 품질 기준에 맞아야 합니다.
- **데비안 자유 소프트웨어 기준 (Debian Free Software Guidelines)**은 자유 소프트웨어에 대한 데비안의 기준을 단순 명료하게 설명한 것입니다. DFSG는 자유 소프트웨어 운동에 있어서 매우 영향력 있는 문서로, 이 문서를 기초로 **오픈소스 정의 (Open Source Definition)**가 작성되었습니다.
- **데비안 정책 안내서 (Debian Policy Manual)**는 데비안 프로젝트의 품질 기준을 자세히 적은 명세서입니다.

데비안 개발자는 여러 가지 다른 프로젝트에 참가하기도 합니다. 데비안과 관계된 프로젝트도 있고, 리눅스 공동체의 일부 혹은 전부와 관련되어 있는 프로젝트도 합니다. 예를 들면 다음과 같습니다:

- **Linux Standard Base (LSB)** 프로젝트는 기본적인 GNU/리눅스 시스템을 표준화하는 프로젝트로, 씨드 파티 소프트웨어와 하드웨어 개발자가 특정 GNU/리눅스 배포판용이 아니라 일반적인 리눅스에 대한 프로그램과 디바이스 드라이버를 쉽게 설계할 수 있도록 하는 것입니다.
- **Filesystem Hierarchy Standard (FHS)** 프로젝트는 리눅스 파일 시스템의 구성을 표준화하는 프로젝트입니다. FHS는 개발한 패키지가 여러 가지 GNU/리눅스 배포판에 어떻게 설치되는지 고민할 필요 없이, 소프트웨어 개발자가 프로그램 설계에 만전념할 수 있도록 하는 프로젝트입니다.
- **데비안 주니어 (Debian Junior)** 프로젝트는 데비안 내부 프로젝트로 어린이 사용자가 사용할 만한 데비안을 만드는 프로젝트입니다.

데비안에 관해 더 일반적인 정보는 **데비안 FAQ**를 참고하십시오.

1.2 GNU/리눅스란?

리눅스는 운영체제입니다. 운영체제는 여러가지 프로그램의 모음으로, 이 프로그램을 이용해 컴퓨터를 사용하고 다른 프로그램을 실행하기도 합니다.

운영체제는 컴퓨터에 필요한 여러가지 기초적인 프로그램으로 구성되어 있고, 이 프로그램을 이용해 사용자는 컴퓨터와의 사소통을 하고 컴퓨터에 지시를 내립니다. 예를 들어 하드디스크, 테이프, 프린터로 데이터를 보내거나 기계에서 데이터를 읽어들이며, 메모리 사용을 제어하고, 다른 프로그램을 실행합니다. 운영체제의 가장 중요한 부분이 바로 커널입니다, GNU/리눅스 시스템에서 리눅스는 커널 부분을 말합니다. 시스템의 나머지 부분은 기타 프로그램으로 구성되며, GNU 프로젝트가 많은 부분을 개발했습니다. 리눅스 커널 그 자체만으로는 시스템을 구성할 수 없기 때문에, 우리는 흔히 리눅스라고 호칭하는 시스템을 GNU/리눅스라는 이름으로 사용합니다.

리눅스는 유닉스 (Unix) 운영체제를 모델로 만든 운영체제입니다. 애초부터 리눅스는 다중작업, 다중 사용자 시스템으로 설계되었습니다. 다중작업, 다중 사용자 시스템이라는 것만으로도 리눅스는 여타의 잘 알려진 운영체제와 차별됩니다. 그러나 리눅스는 여러분이 생각하는 것 이상으로 다른 점이 아주 많습니다. 다른 운영체제와는 달리 어느 누구도 리눅스를 소유하지 않습니다. 자원자의 힘으로 리눅스의 상당부분을 개발했습니다.

후에 GNU/리눅스라고 불리게 된 시스템의 개발은 1984년에 시작되었으며, 이때 **자유소프트웨어재단 (FSF)**은 유닉스와 유사한 운영체제의 개발을 시작하면서 그 이름을 GNU 라고 했습니다.

GNU 프로젝트는 유닉스 (Unix™) 및 리눅스 처럼 유닉스와 유사한 운영체제에서 사용할 수 있는 일련의 자유 소프트웨어 도구를 개발해 왔습니다. 이러한 도구를 이용해 사용자는 파일을 복사하거나 지우는 아주 일상적인 작업부터, 프로그램 작성과 컴파일, 여러가지 종류의 문서 편집에 이르기까지 다양한 작업을 할 수 있습니다.

수많은 단체와 개인이 리눅스의 개발에 기여했지만, 단독으로 가장 크게 기여한 곳은 자유소프트웨어재단입니다. 자유소프트웨어재단은 리눅스에서 사용하는 도구의 대부분을 개발했을 뿐 아니라, 리눅스가 생겨나게 했던 철학과 공동체를 만들어 냈습니다.

리눅스 커널은 리누스 토발즈라는 (Linus Torvalds) 핀란드의 한 컴퓨터 과학대 학생이 1991년 유즈넷 뉴스그룹 comp.os.minix 에미닉스 (Minix) 를 대체하는 커널의 초기 버전을 발표하면서 처음으로 그 모습을 세상에 드러냈습니다. 자세한 사항은 리눅스 인터넷서널의 **리눅스역사페이지 (Linux History Page)**를 참고하십시오.

리누스 토발즈는 몇명의 서브시스템 관리자의 도움을 받아 수백명에 달하는 개발자의 작업을 조율하고 있습니다. 리눅스 커널의 **공식 홈페이지**가 있습니다. linux-kernel 메일링 리스트에 대한 정보는 **linux-kernel 메일링 리스트 FAQ**에서 찾아보십시오.

리눅스 사용자는 소프트웨어 선택에 많은 자유를 갖고 있습니다. 예를 들어 리눅스 사용자는 12개의 다른 커맨드 라인 셸, 여러가지 그래픽 데스크톱 중에서 원하는 소프트웨어를 선택할 수 있습니다. 이와 같이 선택의 폭이 넓어서 셸이나 데스크톱을 바꿀 수 있다는 걸 전혀 생각치 못했던 사용자가 당황스러워하기도 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. 최근 에리눅스는 가정과 업무 무용 사용자에게도 퍼져나가고 있습니다.

1.3 데비안 GNU/리눅스란?

데비안의 철학 및 방법론과 GNU 도구, 리눅스 커널, 그리고 기타 중요한 자유 소프트웨어가 모여 데비안 GNU/리눅스라는 독특한 배포판을 만듭니다. 이 배포판은 수많은 소프트웨어 패키지로 구성됩니다. 배포판의 각 패키지는 실행 파일, 스크립트, 문서, 설정 정보가 들어있으며 메인테이너가 관리합니다. 메인테이너는 각 패키지를 항상 최신으로 유지하고, 버그 리포트를 추적하고, 패키지로 만든 소프트웨어의 원 개발자와 연락을 하고 있습니다. 데비안의 거대한 사용자 기반이 버그 추적 시스템과 결합해 문제점을 빠르게 찾아내고 수정할 수 있습니다.

데비안이 세세한 신경을 쓰는 덕분에 품질 높고, 안정적이고, 확장성 좋은 배포판을 만들어 냈습니다. 간단한 설치 설정에 따라 방화벽에서 데스크톱 공학용 워크스테이션, 고성능 네트워킹 서버까지 다양한 역할을 할 수 있습니다.

기술적으로 우수하면서 리눅스 공동체의 필요와 기대에 대해 적극적으로 참여하기 때문에, 데비안은 고급 사용자에게 특히 인기가 있습니다. 또한 데비안은 지금도 리눅스에 일반적인 기능이 된 많은 기능을 새로 도입하는 데 앞장서 왔습니다.

예를 들어 보면, 소프트웨어의 설치와 제거를 손쉽게 할 수 있는 패키지 관리 시스템이 들어간 최초의 리눅스 배포판이 데비안입니다. 또 다시 설치하지 않고도 업그레йд할 수 있는 최초의 배포판입니다.

데비안은 리눅스 개발의 리더로서 계속하고 있습니다. 그 개발 과정 (전체 운영체제를 구축하고 유지하는 매우 복잡한 작업 하더라도) 은 오픈 소스 개발 모델이 얼마나 잘 진행되는지를 보여주는 되어 있습니다.

데비안은 다른 GNU/Linux 배포판과 다른 가장 큰 특징은 패키지 관리 시스템입니다. 데비안 시스템 관리자는 시스템에 설치된 패키지에 대해 하나의 패키지 설치에서 운영 체제 전체 자동 업데이트까지 완벽하게 제어할 수 있습니다. 개별 패키지를 업데이트하지 않도록 설정할 수 있습니다. 직접 컴파일한 소프트웨어에 대한 종속성을 설정할 수 있습니다.

“트로이 목마” 및 기타 악의적인 소프트웨어로부터 당신의 시스템을 보호하려고 데비안 서버에서 업로드한 패키지가 등록된 데비안 개발자가 만든 패키지가 맞는지 여부를 확인합니다. 또한 데비안 각 패키지는 더 안전하게 설정되도록 세심한 주의를 기울입니다. 배포한 패키지에 보안 문제가 발생하면, 수정 버전을 빠르게 사용할 수 있습니다. 데비안의 간단한 업데이트 옵션을 사용해, 인터넷을 통해 보안 패치는 자동으로 다운로드하고 설치할 수 있습니다.

당신의 데비안 GNU / Linux 시스템에 대한 지원을 받거나 데비안 개발자와 연락을 할 수 있는 한, 최선의 방법은 데비안 프로젝트에서 관리하는 여러 가지 메일링 리스트를 사용하는 것입니다. (이 글이 작성된 시점에서 283 이상의 메일링 리스트가 있습니다.) 메일링 리스트에 가입하려면, [Debian 메일링 리스트 가입 페이지](#) 를 방문, 양식을 기입하면 됩니다.

1.4 데비안 GNU/kFreeBSD 란?

데비안 GNU/kFreeBSD 은 kFreeBSD 커널을 사용하는 데비안 GNU 시스템입니다.

이 데비안 포팅은 현재 i386 과 amd64 아키텍처에서만 개발되고 있지만, 다른 아키텍처에도 포팅 가능성이 있습니다.

데비안이런점에 유의하십시오. GNU/kFreeBSD 는 리눅스 시스템이 아니기 때문에 리눅스 시스템에서 일부 정보는 사용자에게 적용되지 않을 수도 있습니다.

자세한 내용은 [데비안 GNU/kFreeBSD 이식 페이지](#) 와 debian-bsd@lists.debian.org 메일링 리스트를 참조하십시오.

1.5 데비안 GNU/Hurd 란?

데비안 GNU/Hurd 는 GNU Hurd 를 사용하는 GNU 시스템입니다. GNU Mach 마이크로커널 위에서 실행되는 서버의 집합입니다.

Hurd 는 아직 완성되어 있지 않고 일상적인 사용에는 부적합하지만 작업은 계속하고 있습니다. 현재 Hurd 는 i386 아키텍처에서만 개발되고 있지만, 시스템이 안정되면 다른 아키텍처에 넣을 예정입니다.

데비안 GNU/Hurd 는 리눅스 시스템이 아니기 때문에 리눅스 시스템에 대한 일부 정보는 적용되지 않을 수도 있습니다.

자세한 내용은 [데비안 GNU/Hurd ports page](#) 와 debian-hurd@lists.debian.org 메일링 리스트를 참조하십시오.

1.6 데비안받기

인터넷을 통해 데비안 GNU/리눅스를 다운로드하거나 데비안 공식 CD 를 구입하기 위한 정보는 [배포판 홈페이지](#) 를 참조하십시오. [데비안 미러 목록](#) 에는 데비안 공식 미러 사이트가 모두 정리되어 있기 때문에 가장 가까운 미러 사이트를 쉽게 찾을 수 있습니다.

데비안은 설치 후 아주 쉽게 업그레이드할 수 있습니다. 이 설치 단계에서는 시스템의 설정을 도와줍니다. 일단 설치 과정을 마친 시스템은 필요에 따라 업그레이드를 수행할 수 있습니다.

1.7 이 문서의 최신 버전 구하는 법

이 문서는 계속해서 바뀝니다. 데비안 GNU/리눅스 시스템의 9 릴리스에 대한 최신 정보는 [데비안 9 페이지](#) 를 확인하기 바랍니다. 이 설치 문서의 최신 버전은 [공식 데비안 설치 안내서 페이지](#) 에서도 구할 수 있습니다.

1.8 이 문서의 구성

이 문서는 처음으로 데비안을 사용하는 분을 위해 작성된 문서입니다. 가능한 한 전문적인 지식 없이도 읽을 수 있도록 노력하고 있습니다. 하지만 컴퓨터가 어떻게 동작하는지 일반적인 지식이 있다고 가정합니다.

전문적인 사용자도 이 문서에서 최소 설치에 필요한 공간과 데비안 설치 시스템에서 지원하는 하드웨어 등과 같은 내용도 도움이 될 정보를 얻을 수 있습니다. 숙련된 사용자에게는 이 문서의 필요한 부분만 읽어도 됩니다.

이 안내서는 설치 과정의 처음부터 끝까지, 각 과정의 순서대로 쓰여 있습니다. 아래는 데비안 GNU/리눅스를 설치하는 과정으로, 각 과정에 관련된 문서 부분을 안내해 놓았습니다:

1. 2장에서, 하드웨어가 설치 시스템에 필요한 요구 사항을 만족하는지 판단합니다.
2. 3장에서는 기존 시스템을 백업하고 데비안 설치에 앞서 시스템 설계 및 하드웨어 설정을 합니다. 멀티부팅을 생각하고 계시다면, 하드디스크에 데비안 파티션을 만들기 위한 공간을 만들어야 할 수도 있습니다.
3. 4장에서, 설치하는 방법에 따라 필요한 설치 파일을 구합니다.
4. 5장에서, 설치 시스템으로 부팅하는 방법을 설명합니다. 이 과정에서 문제가 발생한 경우 문제를 해결하는 방법도 이장에서 설명합니다.
5. 6장에 따라 실제 설치를 수행하십시오. 여기에서 언어의 선택, 주변 장치 드라이버 모듈 설정, (CD/DVD에서 설치하지 않은 경우) 나머지 설치 파일을 데비안 서버에서 직접 검색하도록 네트워크 연결 설정, 하드디스크 파티션 작업과 베이스 시스템 설치합니다. 그런 다음 설치 작업을 선택합니다. (데비안 시스템의 파티션 설정하는 방법은 부록 C에서 설명합니다.)
6. 7장에서, 새로 설치한 베이스 시스템으로 부팅합니다.

시스템 설치가 완료되면 8장을 읽으십시오. 이장에서는 Unix 와 데비안 정보를 찾는 방법과 커널의 변경 방법을 설명합니다.

마지막으로, 이 문서에 대한 정보와 이 문서에 참여하는 방법은 부록 E에 있습니다.

1.9 문서에 도움을 주실분은 언제든지 환영합니다

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

문서의 소스도 공개되어 있습니다. 부록 E를 보시면 문서에 기여하는 방법에 관하여 더 자세히 쓰여 있습니다. 제안, 조언, 패치 그리고 버그 리포트는 언제나 환영합니다. (버그 리포트할 때는 installation-guide 패키지를 이용하시고, 해당 문제가 이미 보고한 문제인지 먼저 확인하십시오.)

1.10 저작권 및 소프트웨어 라이선스 정보

이 문서를 읽고 있는 여러분 대부분은 상용 소프트웨어에 있는 라이선스를 읽어 봤을 것입니다. 보통 그 라이선스에는 구입한 소프트웨어의 복사본 하나를 단일 컴퓨터에서 사용할 수 있다고 쓰여 있습니다. 이 시스템의 라이선스는 그런 라이선스와는 다릅니다. 다니는 학교와 회사의 모든 컴퓨터에 데비안 GNU/리눅스를 설치해서도 좋고 그러길 바랍니다. 친구에게 빌려주고 컴퓨터에 설치하는 것을 도와 주십시오. 또한 수천장을 복사하고 팔 수도 (약간 제한을 조심하면) 있습니다. 왜냐하면 데비안은 자유 소프트웨어이기 때문입니다.

소프트웨어를 자유 (free) 라고 부르는 것은 소프트웨어의 저작권이 없다는 뜻은 아니고, 자유 소프트웨어를 배포하는 CD/DVD 가 무료라는 뜻도 아닙니다. 자유 소프트웨어는, 부분적인 의미로는 프로그램의 라이선스에서 프로그램을 배포하고 사용할 권리에 대해 비용을 지불할 필요가 없다는 의미입니다. 또 자유 소프트웨어는 누구라도 소프트웨어를 확장하고 개작하고 수정할 수 있으면서, 그렇게 바꾼 결과물도 배포할 수 있습니다.

참고



데비안 프로젝트에서는 사용자의 실용적 필요 때문에 우리의 자유 소프트웨어의 기준에 맞지 않는 패키지도 사용할 수 있도록 되어 있습니다. 이 패키지는 공식적인 배포판의 일부가 아니지만, 데비안 미러 contrib 나 non-free 지역 또는 타사 CD/DVD-ROM 에서 구할 수 있습니다. [Debian FAQ](#)의 “Debian FTP 아카이브” 부분을 참조하십시오.

시스템에 들어있는 프로그램의 대부분은 “GPL” 로 알려진 GNU General Public License 에 따라 이용 허락되고 있습니다. GPL 라이선스의 프로그램의 바이너리를 배포할 때 반드시 프로그램 소스 코드를 이용 가능

하게 하는 것이 요구하고 있습니다. 이것은 사용자가 소프트웨어를 수정할 수 있도록 보장하는 것입니다. 따라서 우리는 데비안 시스템에 포함된 GPL 준수 프로그램의 소스코드¹를 수록하고 있습니다.

데비안에 수록된 프로그램의 저작권과 소프트웨어 라이선스의 형식은 그밖에도 여러 가지가 있습니다. 각 프로그램의 저작권과 라이선스를 한번 시스템을 설치하면 `/usr/share/doc/패키지 이름/copyright` 파일을 찾으면 찾을 수 있습니다.

라이선스 및 데비안이 main 배포판에 소프트웨어를 수록하는데 사용하는 자유 소프트웨어의 기준에 관한 더 자세한 정보를 구입할 경우는 [Debian 무료 소프트웨어 지침](#) 를 참조하십시오.

가장 중요한 법적 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 자유 소프트웨어이므로, 여러분에 목적에 맞게 소프트웨어를 수정하는 권리는 여러분에 있습니다. 또한 이런 방법으로 다른 사람이 소프트웨어를 확장하여 생긴 변화로 얻은 이익을 누릴 권리도 여러분에게 있습니다.

¹ 데비안 소스 패키지를 찾고 및 배포 방법과 바이너리를 빌드하는 방법에 대한 자세한 안내 내용은 [Debian FAQ](#)의 “Debian 패키지 관리 시스템의 기본” 를 참조하십시오.

Chapter 2

시스템 요구 사항

이 절에서는 데비안을 시작하는데 필요한 하드웨어 정보를 다루고 있습니다. 또 GNU 및 Linux 에서 지원되는 하드웨어에 대한 자세한 정보에 대한 링크도 제공했습니다.

2.1 지원하는 하드웨어

데비안은 Linux 및 kFreeBSD 커널과 GNU 툴셋을 쓰는데 필요한 것 외에는 특별한 하드웨어를 요구하지 않습니다. 그러므로 Linux 및 kFreeBSD 커널, libc, gcc 등이 포팅되어 있고, 데비안 포팅이 존재하면 아키텍처나 플랫폼에서 데비안을 작동시킬 수 있습니다. 이미 데비안 GNU/리눅스에서 테스트되는 64-bit ARM 아키텍처 시스템의 자세한 내용은 <http://www.debian.org/ports/arm/>에 있는 포팅 페이지를 참조하십시오.

여기서는 64-bit ARM 아키텍처에서 지원하는 여러 가지 하드웨어를 모두 설명하지는 않고, 일반적인 정보만 설명한 다음 추가 정보가 들어 있는 웹사이트를 안내해 놓았습니다.

2.1.1 지원하는 아키텍처

데비안 GNU/리눅스 9 릴리스는 8 개의 주요 아키텍처와 “기종” 이라는 각 아키텍처 변형을 지원합니다.

아키텍처	데비안의 명칭	서브아키텍처	기종
인텔 x86 기반	i386	default x86 machines	default
		Xen PV domains only	xen
AMD64 및 인텔 64	amd64		
ARM	armel	Marvell Kirkwood and Orion	marvell
ARM, 하드웨어 FPU 포함	armhf	멀티플랫폼	armmp
64 비트 ARM	arm64		
32bit MIPS (big-endian)	mips	MIPS Malta	4kc-malta
		Cavium Octeon	octeon
64bit MIPS (little-endian)	mips64el	MIPS Malta	5kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
32bit MIPS (little-endian)	mipsel	MIPS Malta	4kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
Power Systems	ppc64el	IBM POWER8 또는 그 이후 시스템	
64 비트 IBM S/390	s390x	VM-reader 및 DASD 에서 IPL	generic

이 문서는 Linux 커널을 이용한 64-bit ARM 아키텍처에서의 설치를 다루고 있습니다. 데비안이 지원하는 다른 아키텍처에 대한 정보를 찾고 있다면, [데비안 포팅](#) 페이지를 참조하십시오.

이것은 64-bit ARM 아키텍처 데비안 GNU/리눅스 첫 공식 릴리스입니다. 릴리스하려면 이미 충분히 안정

되어있다고 우리는 생각하고 있습니다. 하지만 아직 다른 아키텍처 버전만큼 널리 사용되지 않아 (즉 사용자가 테스트도 많지 않다) 몇 가지 버그를 우연히 만날 가능성도 있습니다. 어떤 문제가 발생하면 **버그 추적 시스템**을 사용하여 보고하십시오. 그때 그 버그가 Linux 커널을 이용한 64-bit ARM 플랫폼의 것임을 반드시 적어주세요. 또한 **debian-arm 메일링 리스트**의 가입도 필요할지도 모릅니다.

2.1.2 3 가지 ARM 포트

ARM 아키텍처는 발전해왔고 현대적인 ARM 프로세서는 과거 모델에 없는 기능이 들어 있습니다. 그러므로 데비안에서는 다음 3 가지 ARM 포트를 통해 여러 가지 종류의 시스템을 최대한 지원합니다:

- 데비안/armel 은 오래된 32 비트 ARM 프로세서로 하드웨어 부동소수점 기능 (FPU) 이 없는 프로세서이고,
- 데비안/armhf 는 최소한 ARMv7 아키텍처를 구현하고 ARM vector floating point specification 의 버전 3 를 (VFPv3) 구현한 최근의 ARM 프로세서에서만 동작합니다. 이 모델의 프로세서에 들어있는 확장된 기능을 사용하므로 성능이 높습니다.
- 데비안/arm64 는 최소한 ARMv8 아키텍처를 구현한 64 비트 ARM 프로세서입니다.

현재 사용 가능한 ARM CPU 는 빅이든 리틀이든 둘 중 하나의 엔디안에서 동작하지만, 실제로 대부분은 리틀 엔디안에서 동작합니다. 데비안/arm64, 데비안/armhf, 데비안/armel 도 리틀 엔디안 시스템만 지원합니다.

2.1.3 ARM CPU 설계의 다양함과 복잡한 지원

ARM 시스템은 i386/amd64 기반 PC 아키텍처보다 하드웨어가 훨씬 더 다른 점이 많습니다. 그러므로 지원에 복잡한 부분이 많이 있습니다.

ARM 아키텍처는 “시스템 온 칩” (SoC, system on chip) 에서 주로 사용됩니다. 이 SoC 는 여러 회사에서 여러 가지 종류의 하드웨어를 집어넣도록 설계하고, 이 중에는 시스템이 부팅할 때 필요한 아주 기본적인 하드웨어도 있습니다. 시스템 펌웨어 인터페이스는 날이 갈수록 표준화가 되었지만, 오래된 하드웨어의 경우 SoC 마다 많이 다르지만, ARMv8 (arm64) 은 비교적 표준화되어 있기 때문에 리눅스 및 기타 소프트웨어가 지원하기 편리합니다.

ARMv8 하드웨어의 서버 버전은 보통 UEFI (Unified Extensible Firmware Interface) 와 ACPI (Advanced Configuration and Power Interface) 표준을 사용합니다. 이 두 가지에는 시스템 독립적으로 컴퓨터를 부팅하고 설정하는 방법을 제공합니다. x86 PC 에서도 공통으로 사용합니다.

2.1.4 Debian/arm64 가 지원하는 플랫폼

Arm64/AArch64/ARMv8 하드웨어는 데비안 Stretch 릴리스에서 아주 늦게 사용할 수 있었으므로, 릴리스 시점에 메인라인 커널 버전에 통합된 플랫폼은 많지 않습니다. 커널에 들어 있어야 `debian-installer` 가 동작합니다. 이 릴리스에서 다음 플랫폼이 데비안/arm64 에서 지원된다고 알려져 있습니다. 커널 이미지는 하나만 있고, 열거된 모든 플랫폼을 지원합니다.

Applied Micro (APM) Mustang/X-Gene APM Mustang 은 최초로 리눅스를 사용할 수 있었던 ARMv8 시스템입니다. 이 시스템은 X-gene SoC 를 사용하고, X-gene SoC 는 그 이후 다른 시스템에서도 사용되었습니다. X-gene SoC 는 8 코어 CPU 로 이더넷, USB, 시리얼이 들어 있습니다. 폼팩터는 데스크톱 PC 와 동일하지만, 앞으로 다른 버전이 나올 예정입니다. 대부분 하드웨어는 메인라인 커널에서 지원하지만, USB 지원은 Stretch 커널에서 빠져 있습니다.

ARM Juno 개발 플랫폼 Juno 는 6 코어 (2xA57, 4xA53) ARMv8-A 800 Mhz CPU, Mali (T624) 그래픽, 8GB DDR3 RAM, 이더넷, USB, 시리얼이 들어 있는 개발 보드입니다. 시스템 시작과 전원 테스트용으로 설계 되었으므로 작지도 않고 싸지도 않지만, 처음 사용할 수 있는 보드 중 하나입니다. 보드에 들어 있는 모든 하드웨어를 메인라인 커널과 Stretch 에서 지원합니다.

이러한 UEFI 가 아닌 시스템에서 `debian-installer` 를 사용할 때, 설치가 끝날 때 시스템이 부팅 가능하도록 수동으로 설정해야 할 수도 있습니다. 예를 들어 `debian-installer` 에서 시작할 셀에서 필요한 명령어를 실행 하든지 해야 합니다. `flash-kernel` 에서 U-Boot 를 사용한 X-Gene 시스템 부팅을 설정할 수 있습니다.

2.1.4.1 기타 플랫폼

arm64 리눅스 커널의 멀티플랫폼 기능을 사용하면 위 목록에 없는 arm64 시스템에서 debian-installer 를 실행할 수도 있습니다. 단 debian-installer 가 사용하는 커널에서 대상 시스템의 구성 요소 및 device-tree 파일을 사용할 수 있어야 합니다. 이 경우 일반적이라면 UEFI 를 사용하는 한 설치 과정이 간단하고, 시스템을 부팅 가능하게도 만듭니다. UEFI 를 사용하지 않는다면, 부팅 가능하게만 드려 고수동으로 설정을 해야 할 수도 있습니다.

2.1.5 다중 프로세서

멀티 프로세서 지원 (“SMP(symmetric multiprocessing, 대칭 멀티 프로세싱)” 라고 부르는) 은 이 아키텍처에 사용할 수 있습니다. 한 컴퓨터에서 여러 개의 프로세서를 사용하는 일은 원래는 고사양 서버 시스템에서만 일어나는 일이었지만, 최근에는 “멀티코어” 프로세서가 도입되면서 매우 일반적인 일이 되었습니다. 이런 프로세서에는 물리적인 칩 한 개에 두 개 이상의 “코어” 라고 부르는 프로세서 유닛이 들어 있습니다.

표준 데비안 9 커널 이미지는 SMP 지원 기능이 컴파일되어 들어 있습니다. SMP 가 아닌 시스템에서도 이 커널을 문제 없이 사용할 수 있습니다.

2.1.6 그래픽 하드웨어 지원

데비안의 그래픽 장치 지원은 내부에 있는 X.Org 의 X11 시스템과 커널이 얼마나 지원하느냐에 달려 있습니다. 데스크톱 환경은 X11 을 사용하지만 기본적인 프레임 버퍼 그래픽은 커널에 들어 있습니다. 3D 하드웨어 가속이나 동영상 가속 같은 기능이 있는 고급 그래픽 기능을 사용할 수 있느냐 여부는, 시스템의 실제 그래픽 하드웨어 및 필요에 따라 추가 “펌웨어” 이미지의 설치에 따라 (2.2 절 참고) 달라 집니다.

거의 모든 ARM 시스템에서는 그래픽 하드웨어를 별도 카드로 연결할 필요 없이 내장되어 있습니다. 일부 시스템에서는 그래픽 카드를 연결할 수 있는 확장 슬롯이 있지만 드문 경우입니다. 헤드리스 시스템의 경우 그래픽이 아예 없는 경우도 흔합니다. 기본적인 프레임 버퍼 비디오는 그래픽이 있는 모든 장치에서 동작하지만, 3D 그래픽 가속은 변함없이 바이너리 드라이버가 필요합니다. 이러한 상황은 빠르게 개선되고 있지만, stretch 릴리스 시점에서 nouveau (엔비디아 테그라 K1 SoC) 및 freedreno (퀄컴 스냅드래곤 SoC) 드라이버만 릴리스에 들어 있습니다. 기타 하드웨어는 자유 소프트웨어가 아닌 드라이버가 별도로 필요합니다.

지원하는 그래픽 하드웨어와 포인팅 장치에 대한 자세한 내용은 <http://xorg.freedesktop.org/> 를 참조하십시오. 또 데비안 9 는 X.Org 7.7 버전이 들어 있습니다.

2.1.7 네트워크 연결 하드웨어

Linux 커널이 지원하는 네트워크 인터페이스 카드 (NIC) 는 모두 설치 시스템에서도 지원합니다. 드라이버 모듈은 일반적으로 자동으로 로드됩니다.

대부분의 내장이 더넷 장치를 지원하고 일부 PCI 및 USB 장치에 대해서는 모듈이 들어 있습니다.

2.1.8 주변 장치 및 기타 하드웨어

Linux 는 마우스, 프린터, 스캐너, PCMCIA/CardBus/ExpressCard 및 USB 장치와 같은 다양한 하드웨어에 폭넓게 대응하고 있습니다. 하지만 시스템을 설치할 때 이 장치가 필요하지는 않습니다.

2.2 펌웨어가 필요한 장치

디바이스 드라이버가 있느냐와 없는 별도로, 펌웨어 혹은 마이크로코드라고 하는 걸 읽어 들여야 동작하는 하드웨어가 있습니다. 네트워크 인터페이스 카드의 경우 (특히 무선 네트워크 장치의 경우) 이런 하드웨어가 많습니다. 또 일부 USB 장치와 하드 디스크 컨트롤러까지도 펌웨어가 필요하기도 합니다. 다수의 그래픽 카드에서는 펌웨어 없이도 기본적인 기능은 동작하지만, 고급 기능을 사용하려면 펌웨어를 시스템에 설치해야 합니다.

오래 전 장치에서는 동작하는데 펌웨어가 필요한 경우 제조사가 장치 내부의 EEPROM/플래시 칩 안에 펌웨어를 저장해 놓습니다. 최근의 장치에서는 이제 이런 방식으로 펌웨어를 저장하지 않고, 시스템이 부팅할 때 다 호스트 운영 체제에 들어 있는 펌웨어 파일을 해당 장치로 업로드 합니다.

대부분이 펌웨어는 데비안 GNU/리눅스 프로젝트의 기준에 따르면 자유롭지 않은 소프트웨어이기 때문에 메인 배포판이나 설치 시스템에 들어 있지 않습니다. 장치 드라이버가 배포판에 들어 있고 펌웨어를 데비안 GNU/리눅스에서 합법적으로 배포할 수 있는 경우, 아카이브의 non-free 섹션에 별도 패키지로 들어 있기도 합니다.

하지만 그렇다고 해서 이러한 하드웨어를 설치할 때 사용할 수 없는 건 아닙니다. 데비안 GNU/리눅스 5.0 부터 `debian-installer` 는 USB 메모리 같은 이동식 장치에서 펌웨어 파일이나 펌웨어가 들어있는 패키지를 읽어 들일 수 있습니다. 설치할 때 펌웨어 파일이나 패키지를 읽어 들이는 방법은 6.4 절 부분을 참고하십시오.

`debian-installer` 에서 펌웨어 파일 프롬프트를 표시했을 때 펌웨어 파일이 없거나 자유롭지 않은 펌웨어 파일을 설치하고 싶지 않은 경우, 펌웨어를 읽어 들이지 않고도 계속 진행을 시도할 수 있습니다. 드라이버가 추가 펌웨어를 필요 한 경우에도, 일부 특정 상황에서만 필요하고 (예를 들어 `tg3` 드라이버는 특정 네트워크 카드 기종에서만 펌웨어가 필요합니다) 펌웨어 없이도 대부분 시스템에서 동작하는 경우도 있습니다.

2.3 GNU/Linux 에 적합한 하드웨어 구입

데비안 혹은 다른 GNU/Linux 배포판을 **사전설치** 시스템을 출하하고 있는 업체도 있습니다. 약간 여러분의 돈을 가지고 갈지도 모르지만 어느 정도 안심할 수 있습니다. 이 하드웨어는 GNU/Linux 에서 제대로 지원되고 있는 것을 확신하기 때문입니다.

Linux 가 번들된 시스템을 구입하는 경우에도, 아니면 중고 시스템을 구입하는 경우에도 그 하드웨어 Linux 커널에서 지원되고 있는지 다시 한번 확인하는 것이 중요합니다. 위의 참고 자료에 하드웨어가 언급되어 있는지 확인하십시오. (있다면) 구입 영업 사원은 Linux 시스템을 구매하는 것을 전합니다. 또한, Linux 에 우호적인 하드웨어 업체를 지원해 보세요.

2.3.1 독점적이거나 폐쇄된 하드웨어 피하기

일부 하드웨어 제조 업체는 드라이버를 어떻게 만드는지 알려주지 않습니다. 또 비공개 협약 (NDA) 없이는 문서를 보여주지 않아서 드라이버 소스코드를 공개할 수 없게 만듭니다. 소스코드 공개는 자유 소프트웨어의 핵심적인 부분입니다. 문서에 접근할 권한이 없으므로, 이런 장치는 Linux 에서 동작하지 않습니다.

운영 체제와 그 장치 드라이버가 일정한 기능을 하는 장치와 통신하는 방법에 대한 표준이 (또는 업계의 사실 상 표준이) 있습니다. 그러한 표준 또는 사실 상 표준에 따르는 장치는 한 개의 장치 드라이버로 모두 동작하고 특정 장치를 위한 드라이버가 필요하지 않습니다. 일부 하드웨어 (예를 들어 키보드나 마우스 같은 USB “휴먼 인터페이스 장치”, USB 메모리와 메모리 카드 리더 같은 USB 저장 장치) 의 경우 이런 원칙이 아주 잘 동작하고 사실상 시장에서 판매되는 모든 장치가 표준에 맞습니다.

그밖의 경우, 특히 프린터는 불행히도 이런 방식으로 동작하지 않습니다. 여러 가지 프린터가 (사실상의) 표준 컨트롤 언어로 일 부 기능이 동작하긴 하지만, 일부 모델은 비공개 컨트롤 명령이 있어야 동작합니다. 이 비공개 명령은 문서가 없으므로 자유 소프트웨어 운영 체제에서 사용할 수 없거나, 제조사가 제공한 비공개 드라이버를 사용해야 합니다.

그런 하드웨어는 구입할 때는 제조사가 제공한 비공개 드라이버가 있더라도, 드라이버 지원에 따라 하드웨어의 수명이 길지 않습니다. 오늘날 상품의 주기가 짧아졌으므로 소비자 제품도 단종되고, 제조사의 드라이버 업데이트가 더 이상 없는 경우는 흔히 일어납니다. 시스템 업데이트 이후 과거의 비공개 드라이버가 더 이상 동작하지 않는 경우, 정상적인 장치를 드라이버 지원 중단 때문에 사용할 수 없는 상황이 벌어지는데 다가 상황에서 할 수 있는 일이 아무 것도 없습니다. 그러므로 사용하는 운영 체제가 무엇이든간에 이러한 폐쇄된 하드웨어는 애초에 구입하지 말아야 합니다.

이런 상황을 개선하려면, 폐쇄된 하드웨어 제조사가 문서 및 관련 자료를 공개하도록 요구하십시오. 그러면 이 하드웨어의 자유 소프트웨어 드라이버를 만들 수 있습니다.

2.4 설치 미디어

여기서는 데비안을 설치할 때 어떤 매체를 사용할 것인지를 결정하는데 도움이 될 것입니다. 전체를 미디어에 관련해 얘기하는 장 (4장) 이 있고, 여기서 각 미디어에 대해서 장점과 단점을 설명합니다. 그 부분에서 다시 이 페이지를 참조할 수도 있습니다.

2.4.1 CD-ROM/DVD-ROM/BD-ROM

참고

이 안내서에서 “CD-ROM” 이라고 말할 때는, CD-ROM, DVD-ROM, BD-ROM 을 모두 가리킵니다. 어떤 기술이든 운영체제 입장에서는 같은 것이기 때문입니다.

CD-ROM 을 이용한 설치 는 대부분 아키텍처에서 지원 합니다.

2.4.2 네트워크

설치할 때 필요한 파일을 가져오는데 네트워크를 이용할 수 있습니다. 네트워크 사용 여부는 설치 방법에 따라 달라지고, 설치 중에 질문에 어떻게 답했느냐에 따라 달라집니다. 설치 시스템은 HTTP 나 FTP 를 사용할 수 있는 거의 모든 종류의 네트워크 연결을 지원합니다. (PPPoE 는 지원하지 않지만 ISDN 이나 PPP 는 지원하지 않습니다.) 설치를 마친 다음에는 ISDN 이나 PPP 를 사용하도록 설정할 수 있습니다.

CD/DVD 나 USB 메모리 같은 로컬 저장 장치 없이도 네트워크에서 설치 시스템을 부팅할 수도 있습니다. netboot 에 필요한 환경이 구축되어 있다면 (즉 네트워크에 DHCP 와 TFTP 서비스가 동작하고 있으면), 다수의 컴퓨터에 간단하고 빠르게 운영체제를 설치할 수 있습니다. 필요한 환경을 구축하려면 어느 정도 기술 전문 지식이 필요하므로, 초보자들에게는 권하지 않습니다.

또 다른 방법으로 디스크가 없이 (diskless) 설치하고, 네트워크를 사용해 랜과 NFS 마운트에서 부팅할 수도 있습니다.

2.4.3 하드디스크

아키텍처에 따라서는 하드디스크에서 직접 설치 시스템을 부팅하는 것도 한 가지 방법입니다. 이렇게 하려면 설치 프로그램을 하드디스크에 복사할 수 있는 다른 운영체제가 있어야 합니다. 이 방법은 다른 설치 방법이 불가능한 특별한 경우가 아니면 권하지 않습니다.

2.4.4 유닉스 계열 혹은 GNU 시스템

다른 유닉스 계열 시스템이 있다면, (뒤에서 설명하겠지만) 그 시스템을 이용해서 debian-installer 없이도 데비안 GNU/리눅스를 설치할 수 있습니다. 이렇게 다른 시스템을 이용하는 방법은 지원하지 않는 하드웨어에 설치하거나 다운타임을 용납할 수 없는 호스트에 유용할 수 있습니다. 이러한 방법에 관심이 있다면, 바로 **D.3** 절 부분으로 넘어가십시오. 이 설치 방식은 다른 설치 방법이 불가능할 때 고급 사용자의 경우에만 사용할 수 있습니다.

2.4.5 지원하는 저장장치

데비안 설치 프로그램에 들어있는 커널은 최대한 많은 시스템에서 동작할 수 있게 빌드되어 있습니다.

2.5 메모리 및 디스크 공간 요구사항

일반적인 설치를 하려면 최소한 메모리가 31MB 만큼은 있어야 하고 하드디스크 공간이 780MB 만큼 있어야 합니다. 이 숫자는 정말 최소한의 숫자입니다. 실제로 사용할 만한 수준이 어느 정도인지 알고 싶으면, **3.4** 절 부분을 참고하십시오.

메모리나 빈 디스크 공간이 작은 시스템에 설치할 수도 있지만 전문적인 사용자만 하기를 권장합니다.

Chapter 3

데비안 GNU/리눅스를 설치하기 전에

이장에서는 설치 프로그램을 부팅하기 전에, 데비안 설치 준비 사항에 대해 다룹니다. 여기에서는 데이터 백업, 하드웨어에 대한 정보 모으기, 기타 필요한 정보 찾기 와 같은 것을 포함합니다.

3.1 설치 과정 개요

먼저 시스템을 다시 설치하는 것에 대해 설명합니다. 데비안에서 시스템 전체를 다시 설치해야 할 상황은 아주 드뭅니다. 다시 설치해야 하는 경우는 하드디스크의 기계적인 고장이 대부분일 것입니다.

흔히 사용하는 많은 운영체제에서는 치명적인 문제점이 발생하거나 새로운 운영체제의 버전으로 업그레이드하려는 경우 완전히 새로 설치해야 합니다. 처음부터 완전히 새로 설치할 필요는 없다고 해도, 프로그램이 새로운 운영체제에서 제대로 동작하려면 프로그램을 다시 설치해야 합니다.

데비안 GNU/리눅스에서는 잘 되지 않는 경우, OS 를 대체 하지 않고 교체 할 수는 케이스가 훨씬 많습니다. 업그레이드 때 전 부 다 설치 필요 없고, 항상 그 자리에서 업그레이드 할 수 있습니다. 또한 OS 의 릴리스가 계속 해서 프로그램을 항상 호환 합니다. 프로그램의 새로운 버전이 최신에 의존하는 소프트웨어를 요구하는 경우 데비안 패키지 시스템은 필요한 소프트웨어를 자동으로 확인하고 확실하게 설치 합니다. 다시 설치 하지 않아도 특별 써 왔으며, 다시 설치 하지 않으면 안 된다고 하는 것은 최후의 수단 이라는 것이 포인트 입니다. 설치 프로그램은 기존 시스템 위에 다시 설치 하도록 설계 되지 않았 습니다.

다음은 설치 과정에서 해야 할 단계 입니다.

1. 설치하려는 하드디스크에 들어 있는 데이터나 문서를 백업 하십시오.
2. 설치를 시작하기 전에 해당 컴퓨터에 대한 정보와 필요한 문서를 모으십시오.
3. 하드디스크의 데비안 파티션에 사용 가능한 공간을 확보 하십시오.
4. 컴퓨터에 필요한 설치 프로그램 소프트웨어와 특별히 필요한 드라이버 파일의 위치를 확인 하여 다운로드 하십시오.
5. 설치 프로그램이 부팅 할 수 있도록 CD/DVD/USB 메모리와 같은 부팅 미디어를 설정 하거나, 네트워크 부팅 환경을 만드십시오.
6. 설치 시스템을 부팅 하십시오.
7. 설치 언어를 선택 하십시오.
8. 이더넷 네트워크 연결이 있으면 활성화 하십시오.
9. 데비안을 설치 할 파티션을 만들고 마운트 합니다.
10. 베이스 시스템의 자동 다운로드/설치/설정을 지켜보십시오.
11. 데비안 GNU/리눅스 및 기존 시스템을 시작 할 수 있는 부트로더를 설치 하십시오.
12. 새로 설치 한 시스템을 처음으로 시작 하십시오.

설치 할 때 문제가 발생 할 때를 대비 해서, 각 단계가 어떤 패키지와 상관 이 있는지 알아 두는 게 좋습니다. 이 설치 단계의 주요 소프트웨어를 소개 하면:

설치 소프트웨어인 `debian-installer` 는 이 안내서에서 가장 중점적으로 다룹니다. `debian-installer` 는 하드웨어를 찾아서 적당한 드라이버를 읽어 들이고, `dhcp-client` 를 이용해 네트워크 연결을 설정

하고, `debootstrap` 을실행해베이스시스템패키지를설치하고, `tasksel` 로특정소프트웨어를추가로설치합니다. 이외에많은소프트웨어가각단계에서작은역할을담당하고있지만, 새시스템이처음시작하기까지전까지는 `debian-installer` 가작업을마칩니다.

필요에따라서시스템을조정하려면, `tasksel` 을이용해웹서버또는데스크톱환경과같은미리정의한다양한번들소프트웨어를설치할수있도록선택할수있습니다.

설치할때중요한옵션의하나가그래픽데스크톱환경을설치할지옵션입니다. 그래픽데스크톱환경은 X 윈도우시스템과그래픽데스크톱환경하나로구성되어있습니다. “데스크톱환경” 태스크를선택하지 않으면, 아주기본적인명령행기반시스템만설치합니다. 데스크톱환경태스크설치는옵션입니다. 데스크톱환경태스크를설치하면텍스트전용시스템과비교해큰디스크공간을차지하기때문이고, 또많은데비안 GNU/리눅스시스템은서버로동작하기때문에동작하는데그래픽사용자인터페이스가전혀필요없기때문입니다.

X 윈도우시스템은 `debian-installer` 와는완전히별도의소프트웨어이고, 실제로훨씬더복잡합니다. X 윈도우시스템의문제점해결은이안내서가다루는범위를벗어납니다.

3.2 기존데이터를백업하십시오!

시작하기전에지금시스템에있는모든파일을백업해두십시오. 원래설치된운영체제가아닌운영체제를처음으로설치하는거라면, 데비안 GNU/리눅스의루트로쓰디스크를다시파티션해야합니다. 파티션프로그램으로어떤프로그램을사용하든간에, 디스크를파티션하면그디스크에있는모든파일을잃어버리는걸감수해야합니다. 설치에사용하는프로그램은상당히안정적이고수년동안사용해온프로그램이지만, 그만큼강력하기도해서조금만잘못하면막심한손해가발생합니다. 백업을한뒤에도대답을할때나어떤작업을할때신중을기하십시오. 일이분만더생각하면수시간의불필요한작업을예방할수있습니다.

멀티부팅시스템을만든다면, 기존운영체제의배포미디어를가지고있으십시오. 보통그럴필요가없겠지만시스템이부팅하려면부트로더를다시설치해야할수도있고, 최악의상황에서는전체운영체제를다시설치하고백업한내용을복구해야할수도있습니다.

3.3 필요한정보

3.3.1 문서

3.3.1.1 설치안내서

지금읽고있는이문서는데비안의차기릴리스설치가이드의개발버전입니다. 이것은 **다양한형식과다양한언어**에서사용할수있습니다.

3.3.1.2 하드웨어문서

하드웨어를설정하고이용하는방법에대한유용한정보가들어있습니다.

3.3.2 하드웨어정보가있는곳찾기

보통은설치프로그램에서자동으로하드웨어를찾아냅니다. 하지만철저히준비하려면, 설치하기전에하드웨어에대해잘알아두는게 좋습니다.

하드웨어정보는다음에서알아낼수있습니다:

- 하드웨어에같이들어있는설명서.
- 컴퓨터의 BIOS 설정화면. 컴퓨터가시작할때어떤키조합을누르면 BIOS 설정화면을볼수있습니다. 보통 Delete 또는 F2 키이지만, 제조사에따라다른키나키조합을사용할수도있습니다. 보통컴퓨터부팅할때어떤키를눌러야설정메뉴로들어가는지메시지를표시합니다.
- 하드웨어의케이스및 포장.
- 파일관리자에서표시하는것과같은, 다른운영체제의시스템명령어및시스템도구. 특히 RAM 과하드 드라이브에대한정보를알아내는데유용합니다.
- 시스템관리자혹은인터넷서비스회사. 여기서는네트워크및전자메일설정과관련된사항을알수있습니다.

Table 3.1 설치에도움이되는하드웨어정보

하드웨어	필요할수있는정보
하드드라이브	용량이얼마나되는지.
	시스템에붙어있는순서.
	IDE (PATA), SATA, SCSI 중한가지.
	사용가능한빈공간.
	파티션.
다른운영체제를설치한파티션.	
네트워크인터페이스	네트워크인터페이스의종류/모델.
프린터	모델및제조사.
비디오카드	타입/모델및제조사.

3.3.3 하드웨어호환성

여러가지제품이문제없이 Linux 에서작동합니다. 또한 Linux 에서지원하는하드웨어는날이갈수록발전하고있습니다. 하지만아직도 Linux 에서는다른 OS 만큼다양한하드웨어를지원하지않습니다.

Linux 의대부분드라이버는특정제조사나특정제품이나특정브랜드를위해작성된드라이버가아니고, 특정하드웨어/칩셋에맞춰작성되었습니다. 한개의하드웨어설계에맞춰여러개의제품과브랜드가있습니다. 보통칩제조사는 “레퍼런스설계” 라는것을제공하고, 거기에맞춰여러가지제조사가여러가지제품과브랜드를만들수있습니다.

이러한방식은장점과단점이있습니다. 장점은한개의칩셋드라이버가, 여러가지제조사와여러가지제품에서그칩셋을계속사용하는한동작한다는점입니다. 단점은어떤제품에어떤칩셋이사용되었는지정확히아는게쉬운일만은아니라는점입니다. 심지어장치제조사는제품이름이나제품버전을바꾸지도않고하드웨어구조를바꿔버리기도합니다. 그러므로시간이지난다음에같은브랜드나같은이름의제품을구입하더라도다른칩셋을사용할수도있어서다른드라이버를사용해야하거나, 한가지제품만드라이버가없을수도있습니다.

USB 및 PCI/PCI-Express/ExpressCard 장치의경우, 어떤칩셋을사용하고있는지알아내는방법은장치아이디를살펴보는방법입니다. 모든 USB/PCI/PCI-Express/ExpressCard 장치는 “공급사” 와 “제품” 마다아이디가있고, 이두아이디의조합이동일하면같은칩셋을사용하는같은제품입니다.

리눅스시스템에서는이아이디는 USB 장치의경우 `lsusb` 명령, PCI/PCI-Express/ExpressCard 장치는 `lspci -nn` 명령으로알아볼수있습니다. 공급사와제품아이디는콜론으로구분한두개의 16 진수숫자형태로(예를들어 “1d6b:0001”) 주어집니다.

예를들어 `lsusb` 출력은: “Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub” 여기서 1d6b 는공급사아이디이고 0002 는제품아이디입니다.

이더넷카드의 `lspci -nn` 출력은: “03:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller [10ec:8168] (rev 06)”. 오른쪽의각괄호안에아이디가있습니다. 즉 10ec 가공급사아이디이고, 8168 이제품아이디입니다.

다른에로그래픽카드출력이다음과같을수도있습니다: “04:00.0 VGA compatible controller [0300]: Advanced Micro Devices [AMD] nee ATI RV710 [Radeon HD 4350] [1002:954f]”.

윈도우시스템에서는윈도우장치관리자의 “자세히” 탭에서볼수있습니다. 여기에서공급사아이디는앞에 “VEN_” 이붙고, 제품아이디에는앞에 “DEV_” 이붙습니다. 윈도우 7 시스템에서는장치관리자의 탭에서기본값으로는아이디가표시되지않고, “하드웨어 ID” 속성을선택하면아이디를볼수있습니다.

인터넷에서공급사/제품아이디를검색할때, 검색어로 “Linux” 및 “driver” 로검색하면드라이버의특정칩셋지원에관한정보가잘나옵니다. 공급사/제품아이디로검색해도쓸만한결과가나오지않으면, 칩코드로검색해보십시오. 칩코드는 `lsusb` 및 `lspci` 로알수있습니다. (예를들어네트워크카드는 “RTL8111” / “RTL8168B” 가있고그래픽카드는 “RV710” 이있습니다.)

3.3.3.1 라이브시스템에서하드웨어호환성검사하기

데비안 GNU/리눅스는일부아키텍처에서는 “라이브시스템” 라고말하는방식으로도사용할수있습니다. 라이브시스템은 CD 나 DVD 같은읽기전용미디어에서부팅해바로사용할수있는시스템입니다. 이방식에서는컴퓨터에아무것도변경하지않습니다. 라이브시스템에서사용자설정을바꿀수도있고프로그램을추가로설치할수도있지만, 컴퓨터의램안에서만벌어지는일입니다. 즉컴퓨터를꺾다가다시라이브시스템으로부팅하면모조리기본값으로초기화됩니다. 데비안 GNU/리눅스에서컴퓨터의하드웨어를지원하는지알아볼때가장쉬운방법은데비안라이브시스템을돌려보고사용해보는방법입니다.

라이브시스템은사용하는데몇가지제약이있습니다. 첫째는라이브시스템에서안에서한작업이컴퓨터의램에들어가기때문에램이충분히큰시스템에서만동작합니다. 그러므로용량이큰소프트웨어를설치할

경우에 모리부족으로 실패할 수 있습니다. 하드웨어 호환성 테스트와 관련된 또 제약은 공식 데비안 GNU/리눅스 라이브 시스템에는 자유 소프트웨어로만 구성되어 있다는 점입니다. 즉 자유롭지 못한 펌웨어 파일은 들어 있지 않습니다. 그러한 펌웨어 패키지를 수동으로 설치할 수도 있지만, `debian-installer` 와 같은 펌웨어 파일 자동 검색 기능은 없기 때문에 필요할 때마다 수동으로 설치해야 합니다.

사용할 수 있는 데비안 라이브 이미지의 종류는 [데비안 라이브 이미지 웹사이트](#)에 있습니다.

3.3.4 네트워크 설정

컴퓨터가 다른 누군가가 관리하고 있는 고정된 네트워크에 연결되어 있다면 (이더넷이나 비슷한 연결의 경우, 전화 접속이나 PPP 는 아님), 네트워크 시스템 관리자에게 이러한 정보를 알아봐야 합니다:

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

연결된 네트워크가 네트워크 설정에 DHCP (Dynamic Host Configuration Protocol) 를 사용한다면 이 정보가 필요 없습니다. 설치할 때 DHCP 서버에서 컴퓨터로 이 정보를 알려줍니다.

DSL 이나 케이블 모뎀을 (예를 들어 케이블 TV 네트워크를 통해) 통해 인터넷을 사용하고 라우터가 (보통 전화 또는 케이블 사업자가 설정한 상태로 설치됩니다) 네트워크 연결을 처리하는 경우, 보통 DHCP 를 사용할 수 있습니다.

무선 랜/와이파이 네트워크를 사용한다면, 다음을 알아봐야 합니다:

- 무선 네트워크의 ESSID ("네트워크 이름").
- 이 네트워크에 연결할 때 사용할 WEP 또는 WPA/WPA2 보안키 (보안키를 사용하는 경우).

3.4 최소하드웨어 요구사항 맞추기

컴퓨터의 하드웨어에 대한 정보를 모았으면, 설치하려는 방법대로 그 하드웨어에서 설치할 수 있는지 확인하십시오.

필요에 따라서는, 아래의 표에 나온 하드웨어보다 성능이 떨어지는 하드웨어로도 설치할 수 있을 수도 있습니다. 하지만 아래 제안을 무시할 경우 대부분 사용자는 짜증나는 문제를 겪게 됩니다.

Table 3.2 추천하는 최소 시스템 요구사항

설치 종류	램 (최소)	램 (추천)	하드 드라이브
데스크톱 없음	128 메가바이트	512 메가바이트	2 기가바이트
데스크톱 포함	256 메가바이트	1 기가바이트	10 기가바이트

실제로 필요한 최소 메모리는 이 표에 나와 있는 숫자보다 훨씬 적습니다. 아키텍처에 따라 다르지만 최소 60MB 에서 (amd64 의 경우) 데비안을 설치할 수 있습니다. 필요한 디스크 공간도 마찬가지입니다. 특히, 설치할 프로그램을 선택하면 필요한 디스크 공간에 대한 추가 정보는 [D.2 절](#) 를 참조하십시오.

성능이 낮은 오래된 시스템에도 그래픽 데스크톱 환경을 설치할 수 있지만, 그놈이나 KDE 플라스마 데스크톱 환경을 설치하지 말고 리소스를 적게 차지하는 윈도 매니저를 설치하길 권장합니다. 예를 들어 `xfce4`, `icewm`, `wmaker` 등 기타 여러 가지가 있습니다.

서버로 설치할 경우 필요한 메모리나 디스크 공간을 예측하기는 사실 상 불가능합니다. 그 서버가 어떤 용도로 사용하느냐에 따라 많이 달라지기 때문입니다.

여기 나와 있는 크기는 사용자 파일, 메일, 데이터 등과 같은 데이터는 포함하지 않은 숫자입니다. 사용자의 파일 및 데이터가 차지하는 공간을 잡을 때는 넉넉하게 잡는 게 좋습니다.

데비안 GNU/리눅스 시스템을 원활히 동작시키는데 필요한 디스크 공간은 권장 시스템 요구사항에 반영되어 있습니다. 특히, `/var` 파티션에 로그 파일 같은 일반적인 내용뿐만 아니라 데비안 특유의 상태 정보가 들어 있습니다. `dpkg` 파일은 (설치 패키지에 대한 정보) 쉽게 40MB 를 차지합니다. 또한 `apt` 는 설치하기 전에 다운로드한 패키지를 여기에 놓습니다. `/var` 에 최소 200MB 를 할당해야 합니다. 그리고 그래픽 데스크톱 환경을 설치하는 경우에는 더 할당할 것입니다.

3.5 멀티부팅시스템에서미리파티션하기

디스크파티션은디스크를여러개의조각으로나누는작업을말합니다. 각각의조각은다른조각과독립적입니다. 비유하자면집에벽을놓는것과비슷합니다. 어떤방에가구를놓는다고해서다른방에영향을끼치지않습니다.

시스템에이미다른운영체제가전체디스크를차지하고있으면서, 같은디스크에데비안도설치하려는경우에는디스크파티션분할을다시시작해야합니다. 데비안은 Windows 나맥 오에스파티션에설치할수없습니다. 다른 Unix 시스템과는일부파티션을공유할수있을지도모르지만, 여기에서는다루지않습니다. 적어도데비안의 root 에사용할전용파티션이필요합니다.

현재파티션상태정보는현재운영체제의파티션프로그램을이용해알수있습니다. 같은프로그램이있습니다. 파티션하는프로그램은파티션을바꾸지않고도현재파티션을표시하는기능이있습니다.

이미파일시스템이들어있는파티션을바꾸면보통거기에들어있는정보가모두망가집니다. 그러므로파티션작업을하기하기전에항상백업을만들어야합니다. 집의경우에비유하자면, 벽을옮기기전에모든가구를다른곳에 옮겨놓아야가구가망가지지않을겁니다.

최근의몇몇운영체제에는파티션내용을망가뜨리지않고도파티션의위치를옮기거나크기를바꾸는기능이있습니다. 이기능을이용해데이터를잃지않고도파티션을만드는데필요한공간을만들수있습니다. 대부분의경우이기능은잘동작하지만디스크파티션을변경하는작업은본래위험한작업이므로데이터를모두백업한다음진행해야합니다.

debian-installer 에서파티션을만들고삭제하는일은 debian-installer 에서도할수있고기존운영체제에서도할수있습니다. 한가지지켜야할규칙은, 해당파티션을사용하는시스템에서파티션을만들도록하십시오. 예를들어데비안 GNU/리눅스에서사용할파티션은 debian-installer 안에서만들고, 다른운영체제에서사용할파티션은그운영체제안에서만듭니다. debian-installer 에서는 Linux 이아닌파티션도만들수있고, 이렇게만든파티션도다른운영체제에서잘동작합니다. 하지만드물게이것때문에문제가발생하는경우가있습니다. 그러므로확실히하려면다른운영체제가사용하는파티션은그운영체제의자체파티션프로그램을사용해만드십시오.

같은컴퓨터에여러개의 OS 를설치하려는경우, 데비안을설치하기전에다른 OS 를먼저설치하고둡니다. Windows 등의다른 OS 를설치하면데비안을시작하는기능이파괴되어버리거나, 혹은그 OS 의파티션이아닌파티션을다시포맷하라고할수도있습니다.

이렇게되더라도복구할수도있고, 이렇게되지않도록피할수도있습니다. 하지만원래운영체제를먼저설치하면이런문제가없습니다.

3.6 설치하기전에할하드웨어및운영체제설정

이절에서는데비안설치에앞서필요한하드웨어설정에대해알아보겠습니다. 일반적으로이작업은 BIOS 나시스템펌웨어설정을확인하고필요하다면설정을변경하는작업입니다. “BIOS” 또는 “시스템펌웨어” 는하드웨어가사용하는핵심소프트웨어로, 전원을켄다음에부팅과정동안에시작되는가장중요한소프트웨어입니다.

3.6.1 ARM 펌웨어

앞에서도말했듯이, ARM 시스템에서는아쉽게도시스템펌웨어의표준이없습니다. 같은펌웨어를사용하는다른시스템도동작이아주다릅니다. 이는 ARM 아키텍처를사용하는하드웨어의상당부분이임베디드시스템이기때문입니다. 임베디드시스템에서는제조사가특별히수정된버전의펌웨어를빌드하고하드웨어전용패치를포함합니다. 아쉽게도이제조사는수정사항이나확장기능을상위의펌웨어개발자에게전달하지않습니다. 그러므로수정사항이새버전의원래펌웨어에서는빠지게됩니다.

결과적으로새로판매되는시스템에서도수년이 지나도제조사버전의수정된펌웨어를사용하게되는데, 그동안상위메인라인의코드는엄청나게발전해서추가기능을제공할수도있고동작이달라질수도있습니다. 또같은펌웨어의제조사수정버전이달라져도하드웨어의장치이름이일관적이지않습니다. 그래서 ARM 을사용하는시스템에서는제품과무관한설치안내를하기가불가능합니다.

3.6.2 U-Boot 에서이더넷 MAC 주소설정하기

모든이더넷인터페이스의 MAC 주소는보통은전세계에서유일해야합니다. 그리고기술적으로도이더넷브로드캐스트범위안에서는주소가유일해야합니다. 이렇게유일성을보장하기위해, 중앙에서관리하는모음에서제조사마다일정한블럭의 MAC 주소목록을배정받습니다. (그리고일정한사용료를받습니다.) 그다음에판매하는제품마다주소를하나씩부여합니다.

개발보드의 경우에는, 제조사에서이사용료를내지않으려하기도해서유일한주소가없을수도있습니다. 이경우사용자가직접 MAC 주소를설정해야합니다. 이더넷인터페이스에 MAC 주소가지정되어있지않았을때, 일부드라이버는부팅할때마다임의의 MAC 주소를생성하기도합니다. 이렇게동작하는경우수동으로주소를지정하지않아도네트워크를사용할수있습니다. 하지만예를들어 DHCP 서버가클라이언트 MAC 주소에따라일정한 IP 주소를할당하는경우에는제대로동작하지않습니다.

공식적으로배정된 MAC 주소와충돌을피하려면 “사설 (locally administered)” 주소로예약된 주소가 있습니다. 이주소는주소의첫바이트에서두비트로정의됩니다. (영문위키백과의 “MAC address” 글에잘설며오디어있습니다.) 예를들어, 16 진수” ca” 로시작하는모든주소는 (예를들어 ca:ff:ee:12:34:56) 사설주소입니다.

U-Boot 를시스템펌웨어로사용하는시스템에서는, 이더넷 MAC 주소를 “ethaddr” 환경변수에저장합니다. 이값은 U-Boot 명령프롬프트에서 “printenv ethaddr” 명령으로확인해볼수있고, “setenv ethaddr ca:ff:ee:12:34:56” 명령과같이설정할수있습니다. 값을설정하면 “saveenv” 명령으로값을저장합니다.

3.6.3 U-Boot 의커널/최초램디스크/디바이스트리재배치문제

예전버전의 U-Boot 를사용하는일부시스템에서는, 부팅과정에서메모리에있는리눅스커널, 최초램디스크, 디바이스트리를재배치할때문제가있을수도있습니다. 이경우 U-Boot 에서 “Starting kernel ...” 메시지를표시하지만, 그뒤에출력이없이시스템이멈춰있습니다. 이문제는 U-Boot v2014.07 버전이후로해결되었습니다.

만약시스템에서 v2014.07 보다오래된 U-Boot 버전을사용하고나중에그보다나중버전으로업그레이드했다면, U-Boot 업그레이드후에도문제가계속발생할수있습니다. U-Boot 를업그레이드해도현재 U-Boot 환경변수를수정하지는않고, 추가환경변수를 (bootm_size) 설정해야합니다. U-Boot 는환경변수데이터가없이새로설치할경우에만자동으로이환경변수를설정합니다. U-Boot 프롬프트에서 “env default bootm_size; saveenv” 명령으로 bootm_size 환경변수를 U-Boot 의기본값으로수동설정할수도있습니다.

재배치와관련된문제를피해가는한가지가능한방법은 “setenv fdt_high ffffffff; setenv initrd_high 0xffffffff; saveenv” 명령을 U-Boot 프롬프트에서실행해서최초램디스크와디바이스트리파일의재배치를완전히막는방법입니다.

Chapter 4

시스템설치미디어구하기

4.1 공식데비안 GNU/리눅스 CD/DVD-ROM 세트

현재데비안 GNU/리눅스를설치하는가장쉬운방법은공식데비안 CD/DVD-ROM 세트로설치하는것입니다. 공급업체에서이세트를구입할수있습니다 ([CD 벤더페이지](#) 참조). 고속네트워크연결과 CD/DVD 라이터가있으면, 데비안미러사이트에서 CD/DVD-ROM 이미지를다운로드해도괜찮습니다 (자세한설명은 [데비안 CD 페이지](#) 및 [데비안 CD FAQ](#) 참고). 데비안 CD/DVD 세트를가지고있고, 시스템이 CD/DVD 에서부팅할수있으면, 5장부분으로넘어갈수있습니다. 많이사용되는파일이첫번째 CD 또는 DVD 에들어가도록많은노력을하고있습니다. 그러므로기본적인데스크톱설치는첫번째 DVD 또는첫번째 CD 로도 (제한적이긴하지만) 가능합니다.

CD 는요즘추세에서는용량이그리크기않기때문에, 그래픽데스크톱환경중에서는첫번째 CD 로설치할수없는경우도있습니다. 일부데스크톱환경은설치중에나머지파일을다운로드할수있도록네트워크연결이필요하거나추가 CD 가필요합니다.

당신의컴퓨터가 CD 부팅을지원하지않지만 CD 세트를가지고있다면, 다른방법으로네트워크부팅방법을사용할수있고아니면커널을 CD 에서수동으로읽어들여설치프로그램을부팅할수있습니다. 이렇게부팅할때필요한파일은 CD 에있습니다. 데비안네트워크아카이브와 CD 의폴더구조는동일합니다. 그러므로부팅하는데필요한어떤파일이아카이브에서있다면, CD 의같은서브디렉터리에서파일을찾을수있습니다.

설치프로그램을부팅하기만하면, 필요한다른파일은모두 CD 에서이용할수있습니다.

CD 세트가없으면, 설치프로그램의시스템파일을다운로드해서네트워크로연결된컴퓨터에서저장해놓으십시오. 그러면이파일을이용해설치프로그램을부팅할수있습니다.

4.2 데비안미러에서파일다운로드

가장가까이있는 (그래서가장빠를것같은) 미러사이트를찾으려면 [데비안미러사이트목록](#)를참조하십시오.

데비안미러에서 FTP 로파일을다운로드할때반드시 바이너리 (binary) 모드에서파일을다운로드하십시오. 텍스트 (text) 모드또는자동선택모드로받으면안됩니다.

4.2.1 설치이미지를찾을위치

설치이미지는데비안미러의 [debian/dists/stretch/main/installer-arm64/current/images/](#)에있습니다. 각 이미지이름과그용도가 [MANIFEST](#)에설명되어있습니다.

4.3 TFTP 네트워크부팅에필요한파일준비하기

LAN 에연결되어있다면, 네트워크를통해 TFTP 를사용해서다른컴퓨터에서부팅할수도있습니다. 다른컴퓨터에서설치시스템을부팅하려고한다면, 부팅파일을특정위치에복사해놓고, 해당컴퓨터의부팅을지원하도록설정해놓아야합니다.

TFTP 서버를설정해야합니다. 그리고많은컴퓨터에서 DHCP 서버아니면 RARP 서버, 아니면 BOOTP 서버를설정해야합니다.

Reverse Address Resolution Protocol(RARP) 은어떤클라이언트에게어떤 IP 주소를사용해야하는지알려주는한방법입니다. 또다른방법은 BOOTP 프로토콜을사용하는것입니다. BOOTP 는컴퓨터에게그 IP

주소및네트워크의어디에서부팅이미지를가져와야하는지알려주는 IP 프로토콜입니다. DHCP(Dynamic Host Configuration Protocol) 는더유연하며, BOOTP 와호환되는확장기능입니다. 일부시스템은 DHCP 를이용해야만설정할수있습니다.

Trivial File Transfer Protocol 은 (TFTP) 부팅이미지를클라이언트에게넘겨줄때사용합니다. 이론상이프로토콜을사용한다면어떤플랫폼의어떤서버라도사용할수있습니다. 여기서는 SunOS 4.x, SunOS 5.x(Solaris), 그리고 GNU/리눅스에서사용하는명령을예로드립니다.

4.3.1 RARP 서버준비하기

RARP 를설정하려면, 클라이언트컴퓨터에설치되어있는이더넷주소 (MAC 주소) 를알아야합니다. 이정보를모르면 (예를들어 rescue 플로피로) “Rescue” 모드를시작하고 ip addr show dev eth0 명령을사용하십시오.

리눅스커널을사용하는 RARP 서버시스템이나 Solaris/SunOS 에서는 rarpd 프로그램을사용합니다. 클라이언트에대한이더넷하드웨어주소를 “ethers” 데이터베이스에넣어야합니다 (/etc/ethers 파일이나 NIS/NIS+ 를이용). 그다음에 RARP 데몬을시작합니다. 다음명령을 (root 로) 실행합니다: 대부분리눅스시스템이나 SunOS5(Solaris 2) 에서는 /usr/sbin/rarpd -a, 기타리눅스시스템에서는 /usr/sbin/in.rarpd -a, SunOS4(Solaris 1) 에서는 /usr/etc/rarpd -a.

4.3.2 DHCP 서버준비하기

자유소프트웨어 DHCP 서버의하나로 ISC dhcpd 가있습니다. 데비안 GNU/리눅스에서는 isc-dhcp-server 패키지를권장합니다. 다음은간단한설정파일예제입니다 (/etc/dhcp/dhcpd.conf 파일을보십시오):

```
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.img";
    server-name "servername";
    next-server servername;
    hardware ethernet 01:23:45:67:89:AB;
    fixed-address 192.168.1.90;
}
```

이예제에서는, DHCP 서버, TFTP 서버, 네트워크게이트웨이역할을모두하는 servername 이라는서버가있다고가정합니다. domain-name 옵션도바꾸고, 서버이름과클라이언트하드웨어주소도바꿔야합니다. filename 옵션은 TFTP 로가져오려는파일의이름입니다.

dhcpd 설정파일을편집한다음에, /etc/init.d/isc-dhcp-server restart 명령으로 dhcpd 를다시시작하십시오.

4.3.3 BOOTP 서버준비하기

GNU/리눅스용 BOOTP 서버는두가지가있습니다. 하나는 CMU bootpd 이고, 다른프로그램은실제로 DHCP 서버인, ISC dhcpd 가있습니다. 이프로그램은 bootp 와 isc-dhcp-server 라는패키지로포함되어있습니다.

CMU bootpd 를사용하려면먼저 /etc/inetd.conf 파일의해당라인을주석 (또는추가) 할필요가있습니다. 데비안 GNU/리눅스에서는 update-inetd --enable bootps 를실행하고이어 /etc/init.d/inetd reload 로하면됩니다. BOOTP 서버가데비안에서움직이지않는경우는다음과같이합니다:

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

여기서 `/etc/bootptab` 파일을 만듭니다. 이 파일 형식은 `printcap`, `termcap`, `disktab` 파일 등으로 친숙한 예제를 알기 어려운 형식으로 되어 있습니다. 자세한 내용은 `bootptab` 메뉴얼 페이지를 보세요. CMU `bootpd` 는 클라이언트의 하드웨어 (MAC) 주소를 알고 두지 않으면 안 됩니다. `/etc/bootptab` 의 예입니다:

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

적어도 클라이언트의 하드웨어 주소를 지정하는 “ha” 옵션을 변경해야 할 것입니다. “bf” 옵션은 TFTP 로 클라이언트가 받게 될 파일을 지정합니다. 자세한 내용은 4.3.5 절을 참조하십시오.

반대로, ISC `dhcpd` 설정은 정말 쉽습니다. BOOTP 클라이언트를 DHCP 클라이언트의 일종의 특별한 케이스로 취급하기 때문입니다. 일부 아키텍처에서는 BOOTP 로 부팅하게 하려면 복잡한 설정이 필요합니다. 그러한 아키텍처의 경우는 4.3.2 절 부분을 참고하십시오. 그러한 경우가 아니라면, `/etc/dhcp/dhcpd.conf` 파일에서 `allow bootp` 를 클라이언트가 들어있는 서브넷의 설정 부분에 집어넣고, `/etc/init.d/isc-dhcp-server restart` 명령으로 `dhcpd` 를 다시 시작하면 됩니다.

4.3.4 TFTP 서버 사용하기

이동 TFTP 서버를 준비하려면, 먼저 `tftpd` 이 활성화되어 있는지 확인해야 합니다.

`tftpd-hpa` 의 경우에서 서비스가 실행될 수 있는 2 가지 방법이 있습니다. 그것은 시스템의 `inetd` 데몬에서 요구에 따라 시작하거나 독립 데몬으로 실행되도록 설정할 수 있습니다. 이러한 방법 중 어떤 패키지를 다시 구성하여 설치할 때 사용되며 변경할 수 있습니다.

참고



전통적으로 부팅 이미지를 저장하는 위치로 TFTP 서버는 `/tftpboot` 디렉토리를 사용합니다. 하지만, 데비안 GNU/리눅스 패키지는 [Filesystem Hierarchy Standard](#)에 맞추려고 다른 디렉토리를 사용합니다. 예를 들어, `tftpd-hpa` 는 기본적으로 `/srv/tftp` 를 사용합니다. 여기서 설명하는 설정 파일 예제에서는 이 점을 고려해서 적용하십시오.

데비안에서 사용할 수 있는 모든 `in.tftpd` 대체 품은 기본적으로 시스템 로그에 TFTP 요청을 기록해야 합니다. 그중에서는 `-v` 옵션을 사용하면 더 많이 기록합니다. 부팅 문제가 발생한 경우 이 로그 메시지를 확인해 보십시오. 오류의 원인을 파악하는데 좋은 출발점입니다.

4.3.5 TFTP 이미지를 적당한 위치에 놓기

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

4.4 자동설치

여러 컴퓨터에 설치하려면 완전 자동설치가 가능합니다. 이것을 위한 데비안 패키지는 `fai-quickstart` 와 (설치서버로 사용 가능) 데비안 설치 자체를 포함합니다. 자세한 내용은 [FAI 홈페이지](#) 를 참조하십시오.

4.4.1 데비안 설치 프로그램을 이용한 자동설치

데비안 설치 프로그램은 `preconfiguration` 파일을 이용해 자동설치를 지원하고 있습니다. `preconfiguration` 파일은 네트워크 또는 이동식 미디어로 읽어 설치 중에 질문에 대한 답변을 묻어가는 데 사용됩니다.

부록 B 에 보면 미리 설정에 대한 문서가 있고, 약간 고쳐서 사용할 수 있는 잘 동작하는 예제가 있습니다.

Chapter 5

설치시스템부팅하기

5.1 64-bit ARM 에서설치프로그램부팅하기

5.1.1 콘솔설정

그래픽설치프로그램은 stretch 의 arm64 debian-installer 이미지에서는기능이빠져있기때문에시리얼콘솔을사용합니다. 콘솔장치를펌웨어에서자동으로찾아내지만, 찾아내지못하는경우나중에 GRUB 메뉴에서리눅스를부팅할때 “Booting Linux” 메시지가나오고, 아무것도나오지않습니다.

이런문제를겪는다면커널명령행에특정콘솔을지정하는설정을넣어야합니다. GRUB 메뉴에서 e 를눌러 “Edit Kernel command-line” 으로들어가고,

```
--- quiet
```

처럼되어있는부분을

```
console=<장치>,<속도>
```

형식으로바꿉니다. 예를들어

```
console=ttyAMA0,115200n8
```

처럼씁니다. 다마치면 Control-x 를눌러새로운설정으로부팅합니다.

5.1.2 Juno 설치

Juno 에는 EFI 가들어있으므로설치는직관적입니다. 보통 USB 메모리에서설치합니다. USB 부팅이동작하려면최신펌웨어가필요합니다. 2015 년 3 월이후빌드를 <http://releases.linaro.org/latest/members/arm/> 위치에서가져오면동작합니다. 펌웨어업데이트에대해서는 Juno 문서를참고하십시오.

표준 arm64 CD 이미지를 USB 메모리에준비합니다. USB 메모리를뒤의 USB 포트중하나에 꽂습니다. 시리얼케이블을뒤쪽위의 9 핀시리얼포트에연결합니다. 네트워크가필요하면 (네트워크부팅이미지의경우) 이더넷케이블을앞에있는소켓에연결합니다.

시리얼콘솔을 115200, 8 비트, 패리티없음설정으로연결하고 Juno 를부팅합니다. USB 메모리에서부팅해서 GRUB 메뉴까지진행하게됩니다. 콘솔설정은 Juno 에서제대로검색하지못해서 Enter 키를눌러도 커널출력이나오지않습니다. 콘솔을

```
console=ttyAMA0,115200n8
```

설정으로바꿉니다. (5.1.1절위치에설명.) Control-x 를누르면 debian-installer 화면이나타납니다. 여기에서마찬가지로설치를계속진행하면됩니다.

5.1.3 Applied Micro Mustang 설치

UEFI 는이시스템에서사용할수있지만 U-Boot 가들어있으므로, UEFI 펌웨어를먼저설치하고표준부팅/설치를할수도있고, 아니면 U-Boot 부팅을사용할수도있습니다. 그래픽설치는 arm64 아키텍처에서빠져있으므로시리얼콘솔을사용해설치를진행해야합니다.

추천하는설치방법은 debian-installer 커널과 initrd 를하드디스크에복사하고시스템에들어있는 openembedded 시스템을이용해부팅하고설치프로그램을실행하는방법입니다. 아니면 TFTP 를사용해

커널/DTB/initrd 를 복사한 다음 부팅합니다 (5.1.4.1절). 설치 후에 설치한 이미지에서 부팅 하도록 수동으로 수정해야 합니다.

115200, 8 비트, 패리티 없음으로 시리얼 콘솔을 실행하고 시스템을 부팅합니다. 시스템을 부팅하고 “Hit any key to stop autoboot:” 메시지가 나올 때 키를 누르면 Mustang# 프롬프트가 나옵니다. 이때 U-Boot 명령으로 커널, DTB, initrd 를 읽어 들입니다.

5.1.4 TFTP 로 부팅하기

네트워크에서 부팅하려면, 네트워크에 연결되어 있어야 하고 TFTP 네트워크 부팅 서버가 (그리고 네트워크 자동 설정에 필요한 DHCP, RARP 혹은 BOOTP 서버가) 필요합니다.

서버 쪽에서 네트워크 부팅을 설정하는 방법은 4.3절에 설명되어 있습니다.

5.1.4.1 U-Boot 에서 TFTP 부팅

U-Boot 펌웨어를 사용하는 시스템에서 네트워크 부팅은 3 가지 단계로 이루어져 있습니다: (1) 네트워크 설정, (2) 이미지 (커널/최초 램디스크/DTB) 메모리에 읽어 들이기, (3) 읽어 들인 코드 실행.

먼저 네트워크를 설정해야 합니다. 다음을 실행해 DHCP 로 자동 설정할 수 있습니다:

```
setenv autoload no
dhcp
```

아니면 수동으로 환경 변수를 설정할 수 있습니다:

```
setenv ipaddr <클라이언트의 IP 주소>
setenv netmask <네트 마스크>
setenv serverip <TFTP 서버의 IP 주소>
setenv dnsip <네임 서버의 IP 주소>
setenv gatewayip <기본 게이트웨이의 IP 주소>
```

위 설정을 저장하고 싶으면 다음과 같이 합니다:

```
saveenv
```

그 다음에 이미지 (커널/최초 램디스크/DTB) 를 메모리에 읽어 들여야 합니다. TFTP 명령에 메모리를 읽어 들일 위치의 주소를 써야 합니다. 하지만 메모리 배치가 시스템마다 다르기 때문에 어떤 주소를 사용해야 하는지는 일반적인 규칙은 없습니다.

일부 시스템에서는, U-Boot 에 적합한 로딩 주소가 환경 변수로 미리 정의되어 있습니다: kernel_addr_r, ramdisk_addr_r 및 fdt_addr_r. 이 환경 변수가 정의되어 있는지 여부를 다음 명령으로 확인할 수 있습니다

```
printenv kernel_addr_r ramdisk_addr_r fdt_addr_r
```

이 값이 정의되어 있지 않으면, 시스템의 문서에서 적절한 값을 확인해 보고 직접 값을 지정해야 합니다. 예를 들어 Allwinner SunXi SOC 기반 시스템 (예: Allwinner A10, 아키텍처 이름 “sun4i” 또는 Allwinner A20, 아키텍처 이름 “sun7i”) 의 경우, 다음 값을 사용합니다.

```
setenv kernel_addr_r 0x46000000
setenv fdt_addr_r 0x47000000
setenv ramdisk_addr_r 0x48000000
```

로딩 주소를 지정 하면, 다음과 같이 앞에서 지정한 TFTP 서버에서 이미지를 메모리에 읽어 들일 수 있습니다:

```
tftpboot ${kernel_addr_r} <커널 이미지 파일 이름>
tftpboot ${fdt_addr_r} <DTB 파일 이름>
tftpboot ${ramdisk_addr_r} <최초 램디스크 이미지 파일 이름>
```

3 번째 커널 커맨드 라인을 설정하고 읽어 들인 코드를 실행하는 부분입니다. u-boot 는 “bootargs” 환경 변수의 내용을 커널의 커맨드 라인으로 넘깁니다. 그러므로 커널 및 설치 프로그램의 파라미터는 (콘솔 장치 (5.3.1절 참고) 또는 미리 설정 옵션 (5.3.2절 및 부록 B 참고)) 다음과 같은 명령으로 설정할 수 있습니다:

```
setenv bootargs console=ttyS0,115200 rootwait panic=10
```

읽어 들인 코드를 실행 하는 정확한 명령은 이미지 형식에 따라 다릅니다. uImage/uInitrd 의 경우 명령어는 다음과 같고,

```
bootm ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r}
```

네이티브리눅스이미지의경우다음과같습니다:

```
bootz ${kernel_addr_r} ${ramdisk_addr_r}:${filesize} ${fdt_addr_r}
```

표준리눅스이미지로부팅할때, 커널과 DTB 를읽어들이고다음에최초램디스크이미지를읽어들이는게 중요합니다. U-Boot 에서는파일크기변수를마지막에읽어들이는파일의크기로설정하고, bootz 명령이제대로동작하려면램디스크이미지의크기가필요하기때문입니다. 플랫폼전용커널로부팅하는경우 (예를들어 디바이스트리없는커널) 에는 \${fdt_addr_r} 파라미터를생략하면됩니다.

5.2 접근성

눈이불편하다는등의이유로특별한지원이필요한사용자가있습니다. 많은접근성기능은수동으로활성화해야합니다. 접근성기능을사용하는부팅파라미터를추가할수있습니다. 대부분의아키텍처는키보드를 QWERTY 키보드로인식하니주의하십시오.

5.2.1 설치프로그램프론트엔드

데비안설치프로그램은사용자에게질문하고답변하는인터페이스로여러가지프론트엔드를지원합니다. 접근편의에따라여러가지가있습니다: text 는일반텍스트인터페이스이고, newt 는텍스트를사용하는대화상자를사용합니다. 부팅명령으로프론트엔드를선택할수있습니다. 5.3.2절의 DEBIAN_FRONTEND 문서를참고하십시오.

5.2.2 보드장치

일부접근성장치는컴퓨터내부에연결하는보드와비디오메모리에서직접텍스트를읽습니다. 작동에는 fb=false 부팅파라미터를사용하여프레임버퍼지원을비활성화해야합니다. 그러나이것은사용가능한언어가줄어듭니다.

5.2.3 고대비테마

시력이좋지않은사용자의경우설치프로그램에서고대비색상테마를사용하면더읽기 좋습니다. 고대비테마를사용하려면부팅파라미터에 theme=dark 파라미터를붙입니다.

5.2.4 화면확대

시력이낮은사용자를위해, 그래픽설치에서는기본적인화면확대기능이들어있습니다. Control-+ 및 Control- 단축키를눌러글꼴크기를늘리거나줄일수있습니다.

5.2.5 Expert install, rescue mode, automated install

Expert, Rescue, and Automated installation choices are also available with accessibility support. To access them, one has to first enter the “Advanced options” submenu from the boot menu by typing a. When using a BIOS system (the boot menu will have beeped only once), this has to be followed by Enter. Then, to enable speech synthesis, s can optionally be pressed (followed again by Enter when using a BIOS system). From there, various shortcuts can be used: x for expert installation, r for rescue mode, or a for automated installation. When using a BIOS system, they must be followed by Enter.

The automated install choice allows to install 데비안 completely automatically by using preseeding, whose source can be entered after accessibility features get started. Preseeding itself is documented in [부록 B](#).

5.2.6 설치한시스템의접근성

설치한시스템의접근성에대한문서는 [데비안접근성위키페이지](#)에있습니다.

5.3 부팅파라미터

부팅파라미터는리눅스커널파라미터로보통주변장치를제대로동작하도록제어하는데이용합니다. 대부분커널에서주변장치에관한정보를자동으로찾아냅니다. 하지만몇몇경우에는파라미터로커널을조금도와줘야합니다.

시스템을첫번째로부팅하는경우라면, 기본부팅파라미터를시도해보시고 (즉파라미터를사용하지않는것) 제대로동작하는지보십시오. 보통은제대로동작합니다. 제대로동작하지않는경우에다시부팅해서하드웨어에필요한특별한파라미터가있는지찾아보십시오.

부팅파라미터에관한정보는 [Linux BootPrompt HOWTO](#)에 (여러가지보기드문하드웨어에대한정보포함) 있습니다. 여기서는몇가지많이이용하는파라미터에대한대략만다룹니다. 많이발생하는문제점몇개는 [5.4절](#)에들어있습니다.

5.3.1 부팅콘솔

시리얼콘솔로부팅하는경우, 보통자동으로찾아냅니다. 시리얼콘솔로부팅하려는컴퓨터에비디오카드와 (프레임버퍼) 키보드가붙어있다면, console= 장치 파라미터를커널에넘겨야할수있습니다. 여기서 장치는시리얼장치이고, ttyS0 과같이씁니다.

속도와패리티등의시리얼포트파라미터를지정해야합니다. 예를들어 console=ttyS0,9600n8 와 같이합니다. 57600 이나 115200 도많이사용하는속도입니다. “---” 다음에이옵션을지정하십시오. 그래야 (부트로더설치모듈에서지원하는경우) 설치한시스템의부트로더설정으로들어갑니다.

설치프로그램이사용하는터미널종류가사용하고있는터미널에물레이터에맞도록하려면, TERM= 종류 파라미터를추가할수있습니다. 단설치프로그램은다음터미널종류만지원합니다: linux, bterm, ansi, vt102, dumb. debian-installer 의시리얼콘솔의기본값은 vt102 입니다. IMPI 콘솔이나기타이러한터미널종류를지원하지않는가상화도구를사용하고있다면 (예를들어 QEMU/KVM), screen 세션안에서그들을시작하면됩니다. 그러면 screen 터미널타입으로동작하는데 vt102 터미널과매우가깝습니다.

5.3.2 데비안설치프로그램파라미터

설치시스템에서는유용하게쓸수도있는부팅파라미터몇개를더인식합니다¹.

여러가지파라미터는 “짧은형식” 이있어서커널명령행의길이제한을피하면서쉽게입력할수있습니다. 어떤파라미터에짧은형식이있는경우에는, (일반적인) 긴형식뒤에괄호안에써놓았습니다. 이안내서에들어있는예제에서도짧은형식을사용합니다.

debconf/priority (priority) 이파라미터는표시할메시지의가장낮은우선순위를설정합니다.

기본설치에서는 priority=high 를사용합니다. 즉, 높음우선순위와필수우선순위의메시지를표시하지만, 보통과낮음우선순위메시지는넘어갑니다. 문제가발생할경우에는설치프로그램에서우선순위를필요에따라조정합니다.

priority=medium 을부팅파라미터로쓴경우, 설치메뉴에서설치메뉴에서더많은조정을할수있습니다. priority=low 라고쓴경우, 모든메시지를볼수있습니다. (expert 부팅방법과동일합니다.) priority=critical 의경우, 필수메시지만표시하고질문에신경쓰지않고설치를진행합니다.

DEBIAN_FRONTEND 이부팅파라미터는설치프로그램에서사용할사용자인터페이스종류를설정합니다.

현재가능한파라미터값은:

- DEBIAN_FRONTEND=noninteractive
- DEBIAN_FRONTEND=text
- DEBIAN_FRONTEND=newt
- DEBIAN_FRONTEND=gtk

기본프론트엔드는 DEBIAN_FRONTEND=newt 입니다. 시리얼콘솔설치의경우 DEBIAN_FRONTEND=text 가더좋을수도있습니다. 일부특이한설치미디어에서는제한된개수의프론트엔드만들어있지만, newt 및 text 프론트엔드는대부분이설치미디어에서사용할수있습니다. 그래픽을지원하는아키텍처에서는, 그래픽설치프로그램은 gtk 프론트엔드를사용합니다.

BOOT_DEBUG 이부팅파라미터를 2 로설정하면설치프로그램의부팅절차를자세하게기록합니다. 이값을 3 으로지정하면부트프로세스의전략적인부분에서디버깅셸을동작합니다. (부팅을계속하려면이셸을끝내십시오.)

¹ 현재커널에서는 (2.6.9 이후) 32 개의명령행옵션과 32 개의환경변수를사용할수있습니다. 이개수를넘어가면커널은벗어버립니다.

`BOOT_DEBUG=0` 기본값입니다.

`BOOT_DEBUG=1` 보통때보다많은디버깅정보.

`BOOT_DEBUG=2` 디버깅정보아주많이.

`BOOT_DEBUG=3` 부팅과정의곳곳에서셸을실행해서자세히디버깅을할수있습니다. 부팅을계속하려면셸을나가면됩니다.

`INSTALL_MEDIA_DEV` 이매개변수의값은데비안설치프로그램을읽어들일장치경로를지정합니다. 예를들어, `INSTALL_MEDIA_DEV=/dev/floppy/0` 입니다.

부팅플로피는루트플로피를찾으면서, 검색할수있는모든플로피장치를검색합니다. 이파라미터를재지정해서장치한개만검색하도록만들수있습니다.

`log_host`, `log_port` 설치프로그램의로그메시지를로컬파일에도저장하면서지정한호스트와포트에서실행중인원격 `syslog` 에보냅니다. 포트를지정하지않으면표준 `syslog` 포트인 514 번을기본값으로사용합니다.

`lowmem` 사용가능한메모리에따라저용량메모리를판단하는기준을설치프로그램의기본값보다높게만드는데사용합니다. 쓸수있는값은 1 과 2 입니다. [6.3.1.1절](#)부분도참고하십시오.

`noshell` 설치프로그램에서 `tty2` 및 `tty3` 에셸을실행하지않습니다. 물리적인보안이확보되지않은상태에서설치하는경우에좋습니다.

`debian-installer/framebuffer (fb)` 어떤아키텍처에서는많은언어로설치를하려면커널프레임버퍼를사용합니다. 프레임버퍼가문제가있는경우 `fb=false` 매개변수로이기능을비활성화할수있습니다. `bterm` 이나 `bogl` 관련된오류메시지, 검은화면또는설치를시작하고몇분후에멈추는것은문제의증상입니다.

`debian-installer/theme (theme)` 테마는설치프로그램의사용자환경 (색, 아이콘등) 을어떻게보여줄것인가를결정합니다. 프론트엔드에따라사용할수있는데마가달라집니다. 현재 `newt` 와 `gtk` 프론트엔드에서는시각장애인을위한 “dark” 테마만있습니다. 이테마를설정하려면부팅할때 `theme=dark` 를파라미터로넘기면됩니다.

`netcfg/disable_autoconfig` 기본값으로 `debian-installer` 는 IPv6 자동설정및 DHCP 를통해네트워크설정을검색합니다. 검색이성공하면그설정을검사하거나바꿀기회는없습니다. 자동설정이실패할경우에만수동네트워크셋업을할수있습니다.

로컬네트워크에 IPv6 라우터나 DHCP 서버가있으면서이용하고싶지는않을경우 (예를들어잘못된응답을한다든지때문에), `netcfg/disable_autoconfig=true` 파라미터를사용하면네트워크의 (v4 와 v6 모두) 자동설정을막고수동으로정보를입력할수있습니다.

`hw-detect/start_pcmcia` PCMCIA 에문제가있을때 `false` 로하면 PCMCIA 서비스를시작하지않습니다. 일부노트북컴퓨터에서이와관련해문제가발생합니다.

`disk-detect/dmraid/enable (dmraid)` 설치 프로그램에서 Serial ATA RAID(ATA RAID, BIOS RAID, fake RAID 라고도함) 의지원을활성화하려면 `true` 로설정합니다. 이기능은아직실험중인것을주의하십시오. 추가정보는 [데비안 Installer Wiki](#)에있습니다.

`preseed/url (url)` 미리설정해놓은파일을다운로드할 URL 을지정합니다. 이파일이이용해설치를자동화합니다. [4.4절](#)참고.

`preseed/file (file)` 미리설정해놓은파일을읽어들일 URL 을지정합니다. 이파일이이용해자동설치를합니다. [4.4절](#)참고.

`preseed/interactive` 미리설정을했다라도질문을표시하려면 `true` 로설정하십시오. 미리설정파일을테스트하거나디버깅하는데좋습니다. 이설정은부팅파라미터로넘긴파라미터에대해서는효과가없으니주의하십시오. 부팅파라미터에대해서는특별한문법을따로사용합니다. 자세한정보는 [B.5.2절](#)부분을보십시오.

`auto-install/enable (auto)` 미리설정이가능하기전에 물어보는질문을네트워크설정다음으로미룹니다. 이 자동설치사용법에대한자세한정보는 [B.2.3절](#)부분을보십시오.

`finish-install/keep-console` 시리얼콘솔이나관리콘솔에서설치하면, 일반가상콘솔은 (VT1-VT6) /etc/init-tab 에서막습니다. 이파라미터를 `true` 로하면가상콘솔을막지않습니다.

cdrom-detect/eject 기본값으로 `debian-installer` 에서는 다시 시작하기 전에 설치할 때 사용했던 광학 미디어를 자동으로 뺏습니다. 시스템이 CD 에서 자동으로 부팅하는 경우가 아니라면 이럴 필요가 없을 수도 있습니다. 어떤 경우에는 이렇게 하지 않는 게 좋을 때도 있습니다. 예를 들어 해당 광학 드라이브가 미디어를 다시 집어넣지 못하는 경우나, 사용자가 그 자리에 없기 때문에 수동으로 미디어를 집어넣지 못하는 경우입니다. 보통 슬롯 방식 드라이브는 미디어를 자동으로 다시 집어넣지 못합니다.

자동으로 미디어를 꺼내지 않으려면 `false` 로 설정합니다. 단 설치 한 후에 시스템이 광학 드라이브에서 자동으로 부팅하지 않도록 하십시오.

`base-installer/install-recommends (recommends)` 잘못 옵션을 설정 하면 패키지 관리 시스템이 자동으로 설치하는 동안에 시스템에 대한 “Recommends” 을 설치하지 않도록 합니다. 6.3.4 절도 참조하십시오.

이 옵션을 사용하면 매우 간결한 시스템이 됩니다. 반면에 일반적으로는 있는 기능이 이 시스템에서는 없을 가능성이 높습니다. 원하는 기능을 사용하려면 추천 패키지의 일부를 수동으로 설치해야 할 수도 있습니다. 그러므로 이 옵션은 숙련된 사용자만 사용해야 합니다.

`debian-installer/allow_unauthenticated` 기본 설정으로 설치 프로그램에서 사용하는 저장소는 알려진 GPG 키를 이용해 인증할 수 있어야 합니다. 이 인증을 하지 않으려면 `true` 로 설정하십시오. 경고: 보안상 문제가 될 수 있으므로, 권장하지 않습니다.

`rescue/enable` 일반적인 설치를 하지 않고 복구 모드로 변경하려면 `true` 로 설정하십시오. 8.7 절 부분을 참고하십시오.

5.3.3 부팅파라미터로 질문에 답하기

일부 예외를 제외하고, 설치 과정에서 물어보는 모든 질문을 부팅 프롬프트에서 설정할 수 있습니다. 하지만 이 기능은 특수한 상황에서만 쓸 수가 있습니다. 이 기능을 사용하는 방법은 B.2.2 절 부분을 참고하십시오. 몇 가지 예제가 아래에 있습니다.

`debian-installer/language (language)`, `debian-installer/country (country)`, `debian-installer/locale (locale)` 설치 중이나 설치 후에 사용하는 언어 국가와 로케일을 지정하는 방법은 두 가지가 있습니다.

첫 번째, 쉽게 매개변수를 `locale` 로 전달하는 것입니다. 언어 및 국가는 그 가치로부터 파생됩니다. 예를 들면 언어는 독일어와 국가는 스위스로 선택하는 방법은 `locale=de_CH` 사용한다 (`de_CH.UTF-8` 는 설치한 시스템의 기본 `locale` 로 선정됩니다). 제한 언어는 국가 및 `locale` 의 모든 가능한 조합 방법을 얻을 수 있습니다.

두 번째, 별도로 좀 더 쉬운 옵션은 언어와 국가를 지정하는 것입니다. 이 경우에는 장소를 선택하여 설치한 시스템에 대한 구체적인 기본 `locale` 를 지정 추가할 수 있습니다. 예: 언어 =en 국가 =DE `locale=en_GB.UTF-8`.

`anna/choose_modules (modules)` 기본값으로는 읽어들이지 않는 설치 프로그램 컴포넌트를 자동으로 읽어들이는데 사용합니다. 유용한 추가 컴포넌트는 `openssh-client-udeb` (설치 도중에 `scp` 를 사용할 수 있습니다) 및 `ppp-udeb` (D.4 절 참고) 이 있습니다.

`netcfg/disable_autoconfig` IPv6 자동 설정 및 DHCP 를 끄고 강제로 고정네트워크 설정을 하려면 `true` 로 설정하십시오.

`mirror/protocol (protocol)` 기본적으로 설치 프로그램은 데비안 미러에서 파일을 다운로드 하는 `http` 프로토콜을 사용하여 일반적인 우선 순위에서는 설치 중에 `ftp` 로 변경할 수 없습니다. 이 매개변수를 `ftp` 로 설정하면 설치 프로그램에 `ftp` 를 사용하도록 강제할 수 있습니다. 목록에서 `ftp` 미러를 선택할 수는 없으므로, 호스트 이름을 입력해야 한다는 것에 주의하십시오.

`tasksel:tasksel/first (tasks)` `kde-desktop` 태스크 처럼 태스크 목록에 나타나지 않는 태스크를 선택할 때 사용합니다. 자세한 정보는 6.3.5.2 절 참고.

5.3.4 커널 모듈에 파라미터 넘기기

드라이버를 커널 안에 컴파일해 넣었다면, 커널 문서에 쓰여 있는 대로 파라미터를 넘길 수 있습니다. 하지만 드라이버를 모듈로 컴파일했다면 설치한 시스템에서는 부팅 할 때 커널 모듈을 약간 다르게 읽어들이기 때문에, 일반적으로 하는 것처럼 모듈에 파라미터를 넘길 수 없게 됩니다. 그 대신에 설치 프로그램이 지원 하는 특정 문법을 사용해 해당 파라미터를 올바른 설정 파일에 저장해 모듈을 읽어 들일 때 사용하도록 만들 수 있습니다. 이 파라미터는 설치한 시스템의 설정에도 자동으로 적용됩니다.

한편 모듈에 파라미터를 넘겨야 하는 경우는 매우 드뭅니다. 보통 커널에서 시스템에 있는 하드웨어를 자동으로 찾아내서 쓸만한 기본값을 적용해 놓습니다. 하지만 일부 경우에는 파라미터를 수동으로 지정해야 할 수도 있습니다.

모듈의 파라미터를 설정하는 문법은 다음과 같습니다:

```
모듈_이름.파라미터_이름=값
```

같은 모듈 혹은 여러 모듈에 여러 개의 파라미터를 넘기려면, 이 문법을 반복해서 쓰면 됩니다. 예를 들어 오래된 3Com 네트워크 인터페이스 카드에서 BNC (동축) 커넥터와 IRQ 10 을 지정하려면, 다음과 같이 합니다:

```
3c509.xcvt=3 3c509.irq=10
```

5.3.5 커널 모듈 블랙리스트

경우에 따라서는 모듈을 블랙리스트에 올려서 커널이나 udev 가 자동으로 읽어들이지 않게 만들어야 합니다. 이렇게 하는 이유의 한 가지는 특정 모듈이 해당 하드웨어에서 문제를 일으키는 경우입니다. 또 같은 장치에 대해 두 개의 다른 드라이버가 있기도 합니다. 드라이버가 충돌하거나 잘못된 드라이버를 먼저 읽어들이면 해당 장치 가 제대로 동작하지 않을 수 있습니다.

다음 문법으로 모듈을 블랙리스트에 올릴 수 있습니다: `모듈_이름.blacklist=yes`. 이렇게 하면 해당 모듈을 `/etc/modprobe.d/blacklist.local` 의 블랙리스트에 넣어서 설치 프로그램이나 설치한 시스템에 적용합니다.

설치 시스템이 모듈을 직접 읽어들이기도 합니다. 전문가 모드로 설치 프로그램을 시작해서 하드웨어 검색 단계에 나오는 모듈 목록에서 해당 모듈을 제외하면 모듈을 읽어들이지 않게 만들 수 있습니다.

5.4 설치과정의 문제 해결

5.4.1 CD-ROM 신뢰성

가끔 특히 오래된 CD-ROM 드라이브의 경우, CD-ROM 부팅이 실패할 수도 있습니다. 심지어는 (CD-ROM 에서 성공적으로 부팅한 경우에도) CD-ROM 을 인식하지 못하거나 설치 도중 CD-ROM 을 읽으면서 오류가 발생하는 경우도 있습니다.

이러한 문제는 여러 가지 원인이 있을 수 있습니다. 여기서는 일부 많이 발생하는 문제 및 각각의 경우 대처하는 방법을 설명합니다. 나머지는 여러분에게 맡깁니다.

제일 먼저 확인할 수 있는 간단한 일이 두 가지 있습니다.

- CD-ROM 이 부팅하지 않으면, CD-ROM 을 올바르게 넣었는지 그리고 혹시 CD-ROM 표면이 깨끗한지 확인하십시오.
- 설치 프로그램이 CD-ROM 을 인식하지 못하면, CD-ROM 찾기 및 마운트 옵션을 다시 한번 실행해 보십시오. 아주 오래된 CD-ROM 에서 발생하는 일부 DMA 관련 문제는 이런 식으로 해결된다고 알려져 있습니다.

이렇게 해도 해결되지 않으면, 아래에 나와 있는 방법 을 시도해 보십시오. 전부는 아니지만 대부분의 경우, 여기에 나와 있는 방법은 CD-ROM 과 DVD 모두에 적용되지만, 간략하게 CD-ROM 이라는 용어만 사용합니다.

CD-ROM 에서 설치할 수 없는 경우, 사용할 수 있는 다른 설치 방법 을 시도해 보십시오.

5.4.1.1 공통 사항

- 일부 오래된 CD-ROM 드라이브는 최근의 CD 라이터에서 구운 디스크를 읽지 못합니다.
- 일부 아주 오래된 CD-ROM 드라이브는 “직접 메모리 접근” (DMA) 기능을 사용할 경우 올바르게 동작하지 않습니다.

5.4.1.2 문제점 파악 및 해결 방법

CD-ROM 부팅이 실패하면, 아래에 나온 방법 을 시도해 보십시오.

- BIOS 가 CD-ROM 부팅을 지원하는지 확인하고 (아주 오래된 시스템이 경우에 만 해당됩니다) CD-ROM 드라이브가 해당 미디어를 지원하는지 확인하십시오.

- ISO 이미지를다운로드했다면, 그이미지의 md5sum 이이미지를받은곳과같은곳에있는 MD5SUMS 파일안에있는 md5sum 과일치하는지확인하십시오.

```
$ md5sum debian-testing-i386-netinst.iso
a20391b12f7ff22ef705cee4059c6b92  debian-testing-i386-netinst.iso
```

그다음, 구운 CD-ROM 의 md5sum 이일치하는지도확인하십시오. 다음명령을사용하면됩니다. 이미지의크기들이용해서지정한바이트수만큼 CD-ROM 에서읽어들입니다.

```
$ dd if=/dev/cdrom | \
> head -c 'stat --format=%s debian-testing-i386-netinst.iso' | \
> md5sum
a20391b12f7ff22ef705cee4059c6b92  -
262668+0 records in
262668+0 records out
134486016 bytes (134 MB) copied, 97.474 seconds, 1.4 MB/s
```

설치프로그램이성공후, CD-ROM 을찾을수없는경우, 한번다시 시도하면해결할수있습니다. CD-ROM 드라이브가여러개있는경우다른 CD-ROM 드라이브로바꾸어보십시오. 그래도동작하지않거나 CD-ROM 을인식하지만읽으면서오류가발생하면다음을시도해보십시오. Linux 기초지식이필요합니다. 명령을실행하려면먼저두번째가상콘솔 (VT2) 로전환해서셸을활성화하십시오.

- VT4 로전환해서 /var/log/syslog 의내용을보고 (nano 를편집기로사용) 특정오류메시지가있는지확인하십시오. 그다음에 dmesg 의출력도확인해보십시오.
- dmesg 출력에서 CD-ROM 드라이브를인식했는지확인하십시오. 다음과같은내용이있어야합니다 (연속된줄이아닐수도있습니다):

```
Probing IDE interface ide1...
hdc: TOSHIBA DVD-ROM SD-R6112, ATAPI CD/DVD-ROM drive
ide1 at 0x170-0x177,0x376 on irq 15
hdc: ATAPI 24X DVD-ROM DVD-R CD-R/RW drive, 2048kB Cache, UDMA(33)
Uniform CD-ROM driver Revision: 3.20
```

이와같은내용이없으면, CD-ROM 이연결된컨트롤러를인식하지못했거나아예지원하지않는다는뜻입니다. 해당컨트롤러에무슨드라이버가필요한지알려면, modprobe 명령을사용해수동으로드라이버를읽어들여보십시오.

- /dev/ 아래에 CD-ROM 드라이브의장치노드가있는지 확인하십시오. 위의예에서장치노드는 /dev/hdc 입니다. /dev/cdrom 파일도있어야합니다.
- mount 명령으로 CD-ROM 이이미마운트되어있는지확인하십시오. 마운트되어있지않다면수동으로마운트해보십시오:

```
$ mount /dev/hdc /cdrom
```

이명령어다음에무슨오류메시지가없는지확인하십시오.

- DMA 가켜져있는지확인하십시오:

```
$ cd /proc/ide/hdc
$ grep using_dma settings
using_dma      1      0      1      rw
```

첫번째열의 using_dma 다음에나오는 “1” 은 DMA 가켜져있다는뜻입니다. DMA 가켜져있다면해보십시오:

```
$ echo -n "using_dma:0" >settings
```

CD-ROM 드라이브에해당하는장치노드가있는디렉터리안에서실행해야하는것에유의하십시오.

- 설치하는데문제가있다면, 설치프로그램의메인메뉴맨아래부분에있는옵션을사용해 CD-ROM 이올바른지확인해보십시오. 이옵션은 CD-ROM 을안정적으로읽을수있는지시험하는목적으로도사용합니다.

5.4.2 부팅설정

문제가생겨서커널이부팅과정에서멈추거나, 주변장치를인식하지못하거나, 드라이브를제대로인식하지 못하거나하는경우, 먼저부팅파라미터가 5.3절에쓰여있는것처럼제대로되었는지확인하십시오.

장치의펌웨어가없어서문제가발생할수도있습니다. (2.2절및 6.4절참고.)

5.4.3 커널시작메시지해석하기

부팅과정에서, can't find something 혹은 something not present, can't initialize some thing, 아니면심지어는 this driver release depends on something 형식의메시지를볼수있습니다. 이러한메시지대부분은아무런문제를일으키지않습니다. 설치시스템은여러가지주변장치가달린컴퓨터에서동작하도록만들어졌기때문에이런메시지가나옵니다. 당연히그어떤컴퓨터도모든종류의주변장치를가진컴퓨터는없으므로, 운영체제에서는찾으려는주변장치가없을때이러한메시지를내보냅니다. 또시스템이일시적으로멈추는현상이일어날수도있습니다. 이러한현상은어떤장치가응답할때까지기다리는데, 그장치가없을때발생합니다. 이런시간이너무오래걸린다고생각한다면, 나중에직접설정커널을사용할수있습니다 (8.6절참고).

5.4.4 설치문제보고하기

최초부팅단계를지나갔지만설치를마치지못했다면, 디버깅기록저장메뉴가도움이될수도있습니다. 이메뉴를이용하면시스템오류로그와설정정보를설치프로그램에서플로피로복사하거나, 웹브라우저를이용해다운로드할수있습니다. 이정보는무엇이잘못되었는지및어떻게고치는지에대한단서가들어있습니다. 버그를보고할때이정보를버그보고에첨부해주시요.

그외의설치메시지는설치할때 /var/log/ 에들어있고, 설치된시스템으로부팅한다면다음에는 /var/log/installer/ 에들어있습니다.

5.4.5 설치보고제출

그래도문제가있다면, 설치리포트를보내주시요. 설치가성공했을경우에도설치리포트를보내주시는게 좋습니다. 그래야사용자가어떤하드웨어설정을사용하는지에대한정보를많이얻을수있습니다.

주의: 설치보고서는데비안버그추적시스템 (BTS) 에공개되며, 공개메일링리스트에도전달됩니다. 공개해도상관없는전자메일주소를사용하도록하십시오.

작동하고있는데비안시스템이있다면, 설치리포트를보내는가장쉬운방법은다음과같습니다. installation-report 와 reportbug 패키지를설치 (apt install installation-report reportbug) 하고, 8.5.2절에서설명한대로 reportbug 를설정하여 reportbug installation-reports 를실행하십시오.

다른방법으로, 설치보고서를작성하실때아래형식을이용하시고, "installation-reports" 패키지에대한버그를 submit@bugs.debian.org로메일을보내제출해주시요.

```
Package: installation-reports

Boot method: <설치 프로그램을 어떻게 부팅했는지? CD? 플로피? 네트워크?>
Image version: <설치 이미지를 받은 URL을 쓰는 게 가장 좋습니다>
Date: <설치한 날짜 및 시각>

Machine: <컴퓨터 설명(예, IBM Thinkpad R32)>
Processor:
Memory:
Partitions: >df -Tl 명령어의 결과. 파티션 테이블의 raw 정보가 좋습니다.>

Output of lspci -knn (or lspci -nn):

Base System Installation Checklist:
[0] = OK, [E] = Error (please elaborate below), [ ] = didn't try it

Initial boot:          [ ]
Detect network card:   [ ]
Configure network:     [ ]
Detect CD:             [ ]
Load installer modules: [ ]
Detect hard drives:    [ ]
Partition hard drives: [ ]
```

```
Install base system: [ ]
Clock/timezone setup: [ ]
User/password setup: [ ]
Install tasks: [ ]
Install boot loader: [ ]
Overall install: [ ]
```

Comments/Problems:

<설치 과정을 문장으로 설명하십시오. 그리고 처음에 설치했을 때
들었던 생각, 평가, 아이디어 따위도 써 주십시오.>

버그보고서에 문제가 무엇인지 설명하시고, 커널이 멈춘 경우에 마지막으로 볼 수 있는 커널 메시지를 넣으십시오. 문제가 발생할 때 어떤 과정을 거쳤는지 설명하십시오.

Chapter 6

데비안설치프로그램사용하기

6.1 설치프로그램이동작하는방식

데비안설치프로그램은여러가지용도의컴포넌트로구성되어있고, 각컴포넌트마다설치작업을수행합니다. 각컴포넌트는설치작업을수행하면서그작업에필요한정보를사용자에게물어봅니다. 이질문에는우선순위가부여되어있고, 설치프로그램이맨처음시작할때 물어볼질문의우선순위를먼저물어봅니다.

기본값으로설치하면, 꼭필요한(우선순위가높은) 질문만물어봅니다. 그래서사용자가거의관여하지않고매우자동적으로설치할수있습니다. 컴포넌트는순서대로자동실행합니다. 사용하는설치방법및하드웨어의종류따라어떤실행하는컴포넌트가달라집니다. 설치프로그램에서물어보지않는질문은기본값을사용합니다.

어떤문제가발생하면오류화면이나타납니다. 그리고설치메뉴가나타나서, 메뉴에서앞의문제를피해다른작업을선택할수도있습니다. 아무런문제가없으면설치메뉴를볼수없고, 각컴포넌트에해당하는질문에차례대로답을하기만하면됩니다. 심각한오류를알리는우선순위는'중요'이기때문에심각한오류가발생하면반드시오류화면이나타납니다.

설치프로그램에서사용하는기본값중에몇개는 `debian-installer` 가시작할때넘기는부팅파라미터에따라달라집니다. 예를들어강제로고정네트워크설정을하려면(사용할수있으면 IPv6 자동설정및 DHCP 를기본값으로사용합니다) `netcfg/disable_autoconfig=true` 부팅파라미터를추가하면됩니다. 사용할수있는옵션에대해서는 5.3.2절부분을참고하십시오.

고급사용자라면메뉴방식인터페이스가더편할수도있습니다. 메뉴방식에서는각단계를자동으로진행하지않고사용자입력에따라단계를진행합니다. 설치프로그램을수동메뉴방식으로사용하려면, `priority=medium` 파라미터를사용하십시오.

커널모듈을설치하면서파라미터를넘겨야하는하드웨어에서는, "전문가" 모드로설치프로그램을시작합니다. 설치프로그램이시작할때 `expert` 명령을사용하거나, `priority=low` 부팅파라미터를사용하면됩니다. 전문가모드에서는 `debian-installer` 의모든부분을마음대로조정할수있습니다.

이아키텍처에서는설치프로그램에서는문자인터페이스를사용합니다. 현재그래픽인터페이스는사용할수없습니다.

문자단위환경에서는마우스사용을지원하지않습니다. 여기서대화상자에서왔다갔다할때쓰는키를설명합니다. Tab 혹은 오른쪽화살표키를누르면화면에나오는단추와선택사항중에서 "앞으로" 움직이고, Shift-Tab 혹은 왼쪽화살표키는 "뒤로" 움직입니다. 위쪽및 아래쪽화살표키를누르면스크롤목록에서선택할항목을움직이고, 스크롤목록을스크롤합니다. 또긴목록에서는, 글자를하나누르면그글자로시작하는항목이있는부분으로직접이동합니다. 또 Pg-Up 및 Pg-Down 키로목록을스크롤합니다. 스페이스바를누르면확인란따위의항목을토글합니다. 선택한항목으로들어가려면 Enter 를누릅니다.

대화창에는추가도움정보가있을수도있습니다. 도움말이있는경우화면의맨아래에표시되고 F1 키를눌러볼수있습니다.

오류메시지는네번째콘솔에서나옵니다. 네번째콘솔은왼쪽 Alt-F4 를누르면(왼쪽 Alt 를누른상태에서 F4 평선키) 볼수있습니다. 설치메인화면으로돌아오려면왼쪽 Alt-F1 을누릅니다.

이메시지는 `/var/log/syslog` 파일에도들어있습니다. 설치한후에는이로그는새로설치한시스템의 `/var/log/installer/syslog` 파일로복사됩니다. 그밖에설치할때나오는메시지는 `/var/log/` 안에들어있고, 새로설치한시스템으로부팅한다음에 `/var/log/installer/` 안에들어갑니다.

6.1.1 그래픽설치프로그램사용하기

그래픽설치는텍스트설치와기본적으로동일하게동작합니다. 그러므로뒤의설치과정에서도이설명서를계속사용할수있습니다.

마우스보다 키보드 사용이 좋으면, 두 가지를 먼저 알아두십시오. 접혀져 있는 목록을 (예를 들어 대 목록 안에 나라 목록에서 선택할 때 사용) 풀려면, + 및 - 키를 사용할 수 있습니다. 여러 개의 항목을 선택하는 질문에서는 (예를 들어 태스크 선택), 선택한 다음에 먼저 탭을 눌러 계속 단추로 이동해야 합니다. 엔터를 누르면 선택 여부를 토글하고 계속 단추를 누르지 않습니다.

대화창에 도움 말 정보가 있으면 도움 말 단추가 표시됩니다. 도움 말 정보는 이 단추를 누르거나, F1 키를 눌러 액세스할 수 있습니다.

다른 콘솔로 바꾸려면, X 윈도 시스템과 마찬가지로 Ctrl 키를 이용할 수 있습니다. 예를 들어 VT2 로 (첫 번째 디버깅 셸) 이동하려면: Ctrl-왼쪽 Alt-F2 를 누릅니다. 그래픽 설치 프로그램은 VT5 에서 동작하므로 왼쪽 Alt-F5 키를 누르면 VT5 로 돌아갑니다.”

6.2 컴포넌트 소개

다음은 설치 프로그램의 컴포넌트와 각 컴포넌트가 하는 일에 대한 간단한 설명입니다. 특정 컴포넌트를 사용하는 방법에 대해 더 자세한 정보는 6.3 절에 있습니다.

main-menu 설치 프로그램이 동작할 때 컴포넌트의 목록을 표시하고, 컴포넌트를 하나 선택하면 그 컴포넌트를 시작합니다. 메인 메뉴의 질문은 우선 순위가 중간이기 때문에, 우선 순위를 높음이나 중요로 해 놓으면 (기본값은 높음) 메인 메뉴를 볼 수 없습니다. 하지만 오류가 발생해서 사용자가 뭔가 작업을 해야 한다면, 사용자가 문제를 해결할 수도록 우선 순위가 일시적으로 낮아지고, 이 경우 메인 메뉴가 나타날 수도 있습니다.

뒤로 가기 단추를 계속해서 눌러 현재 실행 중인 컴포넌트를 나가게 되면 메인 메뉴로 갑니다.

localechooser 사용자가 설치 과정 및 설치할 시스템에서 사용할 지역화 옵션을 (언어, 국가, 로캘) 선택합니다. 설치 프로그램에서는 선택한 언어로 메시지를 표시합니다. 단 그 언어로 번역이 되지 않았으면 일부 영어 메시지를 표시할 수도 있습니다.

console-setup 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 CD 를 찾아서 마운트합니다.

netcfg 인터넷을 통해 통신할 수도록 컴퓨터 네트워크 연결을 설정합니다.

iso-scan 하드 디스크에 들어있는 ISO 이미지를 (.iso 파일) 찾습니다.

choose-mirror 데비안 아카이브 미러 목록을 표시합니다. 설치할 패키지가 들어있는 위치를 선택합니다.

cdrom-checker CD-ROM 이 올바른지 확인합니다. 설치 CD-ROM 이 망가지지는 않았는지 직접 확인할 수 있습니다.

lowmem lowmem 은 메모리가 작은 시스템을 검사하고, 여러 가지 방법으로 **debian-installer** 에서 필요 없는 부분을 메모리에서 없앱니다. (그 대신 일부 기능도 없어 집니다.)

anna Anna's Not Nearly APT. 선택한 미러 사이트 혹은 CD 에서 가져온 패키지를 설치합니다.

user-setup 루트 암호를 설정하고 루트가 아닌 사용자를 추가합니다.

clock-setup 시스템 시계를 맞추고 시계를 UTC 에 맞추어 여부를 지정합니다.

tzsetup 앞에서 설정한 지역 정보에 따라 시간대를 설정합니다.

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. 데비안의 기본 파티션 도구입니다.

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 RAID 는 최근 마더보드에 들어있는 싸구려 IDE RAID (가짜 하드웨어 RAID) 컨트롤러보다는 대체로 우수합니다.

`base-installer` 다시 시작했을 때 데비안 GNU/리눅스가 동작하는데 필요한 가장 기본적인 패키지를 설치합니다.

`apt-setup` APT 를 설정합니다. 설치를 어떤 방식으로 하느냐에 따라 다르지만, 대부분을 자동으로 설정합니다.

`pkgsel` `tasksel` 프로그램을 이용해 소프트웨어를 추가로 선택하고 설치합니다.

`os-prober` 컴퓨터에 기존에 설치한 운영체제를 찾아서 그 정보를 `bootloader-installer` 에 넘깁니다. 그러면 `bootloader-installer` 에서는 부트로더 시작 메뉴에 이 운영체제를 추가합니다. 이렇게 하면 부팅할 때 어떤 운영체제를 시작할지 쉽게 선택할 수 있습니다.

`bootloader-installer` 설치 도중에, Linux 커널을 설치합니다. 기본 우선 순위에서는 설치 프로그램이 하드웨어에 가장 맞는 커널을 하나 선택합니다. 우선 순위가 낮은 모드에서는, 사용 가능한 여러 가지 커널 중에서 하나를 선택할 수 있습니다.

`shell` 사용자가 메뉴에서 셸을 실행하거나, 두 번째 콘솔에서 셸을 실행합니다.

`save-logs` 문제가 발생했을 때 관련 정보를 플로피 디스크나 네트워크, 하드 디스크 등의 미디어에 기록합니다. 나중에 설치 프로그램의 소프트웨어 문제를 데비안 개발자에게 정확하게 알리는 데 이 기록을 이용합니다.

6.3 컴포넌트 사용하기

여기서는 설치 프로그램의 각 컴포넌트를 자세히 설명합니다. 이 컴포넌트는 사용자 입장에서 몇 단계로 분류할 수 있습니다. 여기서 설명하는 순서는 설치할 때 나타나는 순서입니다. 설치할 때 여기 있는 모듈을 모두 사용하지는 않습니다. 이 모듈 중에서 실제로 어떤 모듈을 사용하는지는 설치 방법과 하드웨어에 따라 달라집니다.

6.3.1 데비안설치 프로그램 준비 및 하드웨어 설정

데비안설치 프로그램을 시작하여 첫 번째 화면이 표시되고 있다고 합시다. 이때 `debian-installer` 기능은 아직 매우 제한적입니다. 하드웨어, 원하는 언어, 실행하는 작업 등에 대해서도 아직 모릅니다. 하지만 걱정하지 마십시오. `debian-installer` 는 아주 똑똑하기 때문에, 하드웨어를 자동으로 검색해서, 필요한 구성 요소를 찾아내고, 고성능 설치 시스템으로 자신을 업그레이드할 수 있습니다. 그러나 (선택하는 언어, 키보드 배치, 사용할 네트워크 미러 선택처럼) 몇 가지 작업은 자동으로 알아낼 수 없으므로, `debian-installer` 에게 알려줘야 합니다.

이 단계에서 `debian-installer` 는 여러 번의 하드웨어 검색을 합니다. 첫 번째는 설치 프로그램의 컴포넌트를 읽어들이는데 필요한 하드웨어를 (예를 들어 CD-ROM 이나 네트워크 카드) 검색합니다. 첫 번째로 검색할 때는 아직 사용할 수 없는 드라이버가 있기 때문에, 나중 단계에서 하드웨어 검색을 다시 합니다.

하드웨어 검색할 때 `debian-installer` 에서 하드웨어 드라이버에서 펌웨어를 읽어 들여야 하는 지 검사합니다. 펌웨어가 필요하지만 없는 경우에는, 없는 펌웨어를 이동식 미디어에서 읽어 들일 수 있도록 대화 상자를 표시합니다. 자세한 설명은 6.4 절 부분을 참고하십시오.

6.3.1.1 사용 가능 메모리 검사 / 저용량 메모리 모드

`debian-installer` 가 맨 처음에 하는 일 중의 하나는 메모리 검사입니다. 메모리가 부족할 경우 이 컴포넌트에서는 시스템에서 데비안 GNU/리눅스를 설치하는데 문제가 없도록 설치 과정에 약간 수정을 가합니다.

설치 프로그램의 메모리 사용량을 줄이려고 고쳐 짠 몇 가지 기능을 사용하지 않습니다. 즉 설치하는 영어로 만 진행합니다. 물론 설치한 시스템은 설치가 끝난 다음에 지역화 기능을 설정할 수 있습니다.

그걸로 부족하다면, 설치 프로그램에서 기초적인 설치를 마칠 수 있는 컴포넌트만 읽어들이는 방법으로 메모리 사용량을 줄입니다. 이렇게 하면 설치 시스템의 기능이 줄어듭니다. 수동으로 컴포넌트를 추가로 읽어 들일 수도 있지만, 컴포넌트를 하나 하나 선택할 때 마다 메모리를 추가로 사용하기 때문에 설치가 실패할 수도 있습니다.

설치 프로그램이 저용량 메모리 모드로 동작하는 경우, 상당히 큰 스왑 파티션을 (64MB- 128MB) 만들기를 권장합니다. 스왑 파티션을 가상 메모리로 사용해서 시스템에서 사용 가능한 메모리 양을 늘립니다. 설치 프

로그에서는 가능한 한 설치 과정의 앞부분에서 스왑 파티션을 활성화합니다. 단 스왑 파티션을 자주 사용하면 시스템의 성능이 떨어지고 디스크 동작이 많아질 수 있습니다.

이렇게 하더라도 시스템의 메모리가 부족하면 시스템이 멈출 수도 있고, 예상하지 못한 오류가 발생하거나 커널이 프로세스를 끝낼 수도 있습니다. (이 경우에 “Out of memory” 메시지가 VT4 와 syslog 에 나타납니다.)

예를 들어 저용량 메모리 모드에서 스왑 공간이 부족하면 큰 EXT3 파일 시스템을 만들 때 실패한다는 보고가 있었습니다. 스왑을 더 늘려도 개선이 안 된다면, 파일 시스템을 EXT2 로 (EXT2 는 설치 프로그램의 필수 컴포넌트입니다) 만들어 보십시오. 설치를 끝낸 다음에 EXT2 파티션을 EXT3 로 바꿀 수 있습니다.

5.3.2 절에 설명한 것처럼 “lowmem” 부팅 파라미터를 이용해, 설치 프로그램이 사용하는 저용량 메모리의 수준을 강제로 높일 수 있습니다. 사용 가능한 메모리보다 자동 검색한 것보다 더 많은 메모리로 높일 수 있습니다.

6.3.1.2 지역화 옵션 선택

대부분의 경우 처음 물어보는 질문은 설치할 때 및 설치한 시스템에 모두 사용할 지역화 옵션 선택에 대한 것입니다. 지역화 옵션은 언어, 위치, 로캘로 이루어져 있습니다.

여기서 선택한 언어를 사용해 나머지 설치 과정을 진행합니다. 단 그 언어로 해당 대화상자의 번역이 있어야 됩니다. 그 언어로 된 번역문이 없으면 기본값인 영어를 사용합니다.

선택한 지리적 위치 (대부분의 경우 국가) 는 설치 과정의 뒷부분에서 기본 표준 시간대 추출과 그 나라에 적절한 데비안 미러를 고르는 데 이용합니다. 국가와 언어는 새 데비안 시스템의 로캘 결정이나 올바른 키보드 레이아웃 선택을 지원 합니다.

먼저 사용할 언어를 선택합니다. 각 언어의 이름은 영어 (왼쪽에) 및 해당 언어 (오른쪽에) 로 쓰여 있습니다. 오른쪽에 있는 이름은 해당 언어의 문자를 이용해 표시 됩니다. 언어 목록은 영어 이름 순서로 나열되어 있습니다. 목록의 맨 위에 “C” 로캘을 선택하는 옵션이 있습니다. “C” 로캘을 선택하면 설치 는 영어로 진행 하고, 설치한 시스템은 locales 패키지를 설치하지 않고 지역화 기능이 없게 됩니다.

다음은 지리적 위치를 선택 하라는 메시지가 있습니다. 언어 선택 시 해당 언어가 여러 국가들에서 공식 언어로 되어 있는 경우 ¹ 그 국가의 목록을 표시 합니다. 목록에 없는 국가를 선택 하면 기타 (마지막 선택) 을 선택 하십시오. 그러면 대륙의 목록을 표시 합니다. 대륙을 선택 하면 관련 국가 목록을 표시 합니다.

언어에 대해 국가가 하나 뿐이라면 국가 목록에 그 나라가 속한 대륙 또는 지역을 표시 하고 그 나라를 기본적으로 선택 합니다. 다른 대륙에 있는 국가를 선택 하고 싶은 경우 뒤로 가기를 선택 하십시오.

참고



설치 시스템의 표준 시간대를 설정 하면서 당신이 살거나 설치한 국가를 선택 하는 일은 중요합니다.

지역이 정의 되어 있지 않은 언어 및 국가의 조합을 선택 하여 해당 언어에 여러 지역이 존재 하는 경우, 그중에서 설치한 시스템의 기본 지역을 선택 하게 됩니다. ² 그렇지 않으면 기본 지역은 선택한 언어 및 국가를 바탕으로 선택 됩니다.

이전에 설명한 것처럼 선택된 기본 지역은 문자 코드 UTF-8 를 사용합니다.

낮은 우선 순위로 설치 하는 경우, 설치 시스템에 만들 로캘을 선택 할 때 “레가시” 로캘을 ³ 포함해 로캘을 선택 합니다. 이 경우 어떤 로캘을 설치 시스템의 기본 로캘로 사용할 지 묻습니다.

6.3.1.3 키보드 선택하기

어떤 키보드는 특정 언어에서 사용하는 문자에 맞게 만들어져 있습니다. 사용하고 있는 키보드에 맞는 키보드 배치를 고르시고, 해당 키보드 배치가 없으면 어느 정도 비슷한 키보드 배치를 고르십시오. 시스템 설치를 모두 마치면 더 많은 종류의 키보드 배치 중에서 하나를 고를 수 있게 됩니다. (dpkg-reconfigure keyboard-configuration 를 실행 하십시오.)

반전 표시를 해당 키보드로 옮기고 Enter 를 누르십시오. 화살표 키로 반전 표시를 움직입니다. 화살표 키는 모든 언어의 키보드에 대해 동일하므로, 키보드 설정과는 상관이 없습니다.

¹ 기술적인 용어로 언어에 대해 국가 코드가 다른 만큼, 여러 로캘이 존재 합니다.

² 우선 순위가 중략은 낮은에서는 선택한 언어로 유효한 지역 중 항상 마음에 드는 것을 선택 할 수 있습니다 (여러 개 있는 경우).

³ 레가시 로캘은 UTF-8 문자 인코딩을 사용하지 않고, ISO 8859-1 (서유럽 언어로 사용) 또는 EUC-JP (일어로 사용) 등의 오래전 문자 인코딩을 사용합니다.

6.3.1.4 데비안설치프로그램 ISO 이미지찾기

hd-media 에설치하는경우, 설치하다가설치파일의나머지를읽을때데비안설치프로그램 ISO 이미지를찾아서마운트합니다. iso-scan 구성요소가그일을합니다.

처음에 iso-scan 은알려진파일시스템을사용하는블록장치 (파티션등) 를자동으로마운트하고, .iso(더말하면 .ISO) 로끝나는파일이름을순서대로검색합니다. 처음시도에서루트디렉터리와하위디렉터리밖에검색하지않는것을주의하십시오 (즉 /whatever.iso 과 /data/whatever.iso 을감지하지만 /data/tmp/ whatever.iso 는찾지않습니다것입니다). ISO 이미지를찾으면, iso-scan 는그이미지가올바른데비안 ISO 이미지인지아닌지판단하려고그내용을확인합니다. 전자의경우는완료되지만후자의경우 iso-scan 은다른이미지를찾습니다.

앞에서설치 ISO 이미지를찾는데실패하면, iso-scan 에서는계속해서이미지를찾을지여부를물어봅니다. 그러면맨위의디렉터리만찾는게아니라, 모든파일시스템을뒤져봅니다.

iso-scan 에서설치프로그램 ISO 이미지를찾지못했다면, 원래운영체제로다시시작해서이미지이름이올바른지 (.iso 로끝나는지), debian-installer 가인식할수있는파일시스템에들어있는지, 파일이손상되지않았는지 (체크섬확인) 확인하십시오. 경험많은유닉스사용자라면다시시작하지않고두번째콘솔에서할수도있습니다.

6.3.1.5 네트워크설정하기

이단계에서시스템에네트워크장치를두개이상찾으면, 어떤장치를 주요네트워크인터페이스로사용할지질문을받게됩니다. 주요네트워크인터페이스란설치할때사용할인터페이스를말합니다. 이인터페이스외의인터페이스는이시점에서는설정하지않습니다. 설치가다끝난다음에네트워크장치를추가로설정할수있습니다. interfaces(5) 맨페이지를참고하십시오.

6.3.1.5.1 네트워크자동설정 기본으로 debian-installer 는가능하면자동으로컴퓨터의네트워크설정을시도합니다. 자동설정이실패한경우는여러가지원인이있습니다. 네트워크케이블이빠졌든지, 자동설정에필요한네트워크환경이아니라든지등여러가지문제때문에실패할수있습니다. 오류를확인하려면 4 번째콘솔에서오류메시지를보십시오. 어떤상황이든다시시도하거나, 수동으로설정할지여부를물어봅니다. 가끔네트워크서비스가자동설정에필요한응답이느릴수도있으니, 정상이분명하다고확신한다면자동설정을다시시도해보십시오. 자동설정이계속실패하면수동네트워크설정으로들어갈수있습니다.

6.3.1.5.2 네트워크수동설정 수동네트워크설정에서는네트워크에관한여러가지정보를차례대로물어봅니다. IP 주소, 넷마스크, 게이트웨이, 네임서버주소, 호스트이름을물어봅니다. 또무선네트워크인터페이스가있다면, 무선 ESSID(무선네트워크이름) 및 WEP 키또는 WPA/WPA2 암호를물어봅니다. 3.3절의답을채워넣으십시오.

참고



알아두면편리할수도있고아닐수도있는기술적인정보: 이프로그램에서는네트워크 IP 주소가시스템의 IP 주소와넷마스크를비트 AND 한값이라고가정합니다. 브로드캐스트주소는시스템의 IP 주소와넷마스크의비트 NOT 한값을 OR 한값이라고가정합니다. 또게이트웨로도임의로추정합니다. 수동설정에서무슨값을써야할지잘모르겠다면, 시스템의추정값을사용해보십시오. 일단시스템을설치한다음에필요하다면 /etc/network/interfaces 파일을편집해서이설정을바꿀수있습니다.

6.3.1.5.3 IPv4 및 IPv6 데비안 GNU/리눅스 7.0(“Wheezy”) 부터, debian-installer 에서 IPv6 를 “전통적인” IPv4 와마찬가지로지원합니다. 모든 IPv4 와 IPv6 조합을 (IPv4 전용, IPv6 전용, 동시사용) 지원합니다.

IPv4 의자동설정은 DHCP(Dynamic Host Configuration Protocol) 를이용합니다. IPv6 자동설정은 NDP(Neighbor Discovery Protocol, 재귀적 DNS 서버설정 (RDNSS) 기능포함) 를사용한상태없는자동설정, DHCPv6 를이용한상태있는자동설정과, 상태있는/없는방식을혼합한자동설정도 (NDP 를통해주소설정, DHCPv6 를통해추가파라미터설정) 지원합니다.

6.3.1.6 시계 및 시간 설정

설치 프로그램은 시스템 시각을 정확히 맞추려고 먼저 인터넷의 타임 서버에 (NTP 프로토콜 사용) 연결합니다. 연결이 실패하면, 부팅할 때 시스템 시계에서 읽은 시각과 날짜가 올바르다고 가정합니다. 설치 과정에서 시스템 시각을 수동으로 맞추는 방법은 없습니다.

설치 과정에서 처음으로 선택한 지역에 따라 해당 위치에 해당하는 시간대의 목록을 표시합니다. 당신의 위치에서 시간대가 하나밖에 없고, 기본 설치를 수행하는 경우 시스템은 나열하지 않고, 그 시간대에 있다고 가정합니다.

전문가 모드 및 우선 순위에서 설치하는 경우 표준 시간대 “협정 세계시” (UTC) 를 사용한다는 옵션이 추가됩니다.

무슨 이유에서든 선택한 위치에 해당하지 않는 시간대를 이용하려면, 두 가지 방법이 있습니다.

1. 간단한 방법은 설치를 완료하고 새로 부팅한 후 다른 시간대를 선택하는 것입니다. 다음과 같은 명령을 사용합니다:

```
# dpkg-reconfigure tzdata
```

2. 다른 방법으로 설치 시스템이 부팅할 때 `time/zone=` 값 파라미터를 넘겨서 시간대를 설정할 수도 있습니다. 이 값은 올바른 시간대 값이어야 합니다. 예를 들어 `Europe/London` 이나 `UTC` 가 있습니다.

자동 설치에 `preseed` 를 사용하여 시간대를 원하는 값으로 설정할 수 있습니다.

6.3.2 사용자 및 암호 설정

클릭 설정 직전에 설치하는 “root” 계정 및 초기 사용자 계정을 설정합니다. 다른 사용자 계정은 설치 후 작성하십시오.

6.3.2.1 루트 암호 설정

루트 (root) 계정은 슈퍼유저라고도 합니다. 이 계정은 시스템의 모든 보안 장벽을 그냥 통과할 수 있습니다. 루트 계정은 시스템 관리를 할 경우에만 사용해야 하고, 가능한 한 짧은 시간만 사용해야 합니다.

암호를 만들 때 적어도 6 자 이상이고, 대문자와 소문자와 특수 문자를 모두 포함해야 좋습니다. 루트 암호를 설정할 때 좀 더 주의를 기울여 주십시오. 루트 계정은 권한이 막강합니다. 사전에 나와 있는 단어나 추측할 수 있는 개인 정보는 암호에 사용하지 마십시오.

누가 루트 암호를 말해 달라고 하면 조심하십시오. 한 시스템의 관리자가 여러 명인 경우가 아니라면, 루트 암호는 다른 사람에게 알려주면 안 됩니다.

6.3.2.2 일반 사용자 만들기

여기에서 일반 사용자 계정을 만들 것인지 물어봅니다. 이 계정은 주로 사용하는 개인 로그인 계정입니다. 일상적인 용도나 개인 로그인에 루트 계정을 사용하면 안 됩니다.

루트 권한의 사용을 피하는 이유 중 하나는 루트 계정으로서는 아주 쉽게 복구하기 힘든 손상을 입힐 수 있기 때문입니다. 다른 이유는 슈퍼유저의 권한을 이용해 숨어서 시스템의 보안을 침해하는 프로그램인, 트로이 목마 프로그램을 실행하도록 속을 수도 있기 때문입니다. 유닉스 시스템 관리에 대한 원만한 책에서는 모두 이 주제에 대해 좀 더 자세하게 설명하고 있습니다. 처음 접한다면 보안 문제에 대한 책을 한 권 정도 읽어보십시오.

먼저 전체 이름을 입력하고 사용자 계정으로 사용할 이름을 입력합니다. 사용자 계정은 이름 같은 걸 사용하면 충분하고, 실제로 이름이 기본값입니다. 마지막으로 이 계정의 암호를 입력하십시오.

설치가 끝난 다음에 언제든지 또 계정을 만들려면, `adduser` 명령을 사용하십시오.

6.3.3 파티션하기 및 마운트 위치 선택

여기에서 마지막으로 하드웨어 검색을 하면, `debian-installer` 는 필요한 기능을 모든 것 추고, 사용자의 필요에 맞게 진짜 설치 작업을 할 준비를 갖추게 됩니다. 제목이 말하듯 다음 몇 개의 컴포넌트는 디스크를 파티션하고, 파일 시스템을 만들고, 마운트 위치를 지정하고, 또 필요하다면 LVM, RAID, 암호화 장치와 같은 관련 설정을 합니다.

파티션하는 게 불안하거나 자세히 알고 싶으시면, 부록 C 부분을 참고하십시오.

먼저 선택에 따라, 전체 드라이브나 드라이브의 빈 공간을 자동으로 파티션할 수 있습니다. 이 방법을 “자동” 파티션이라고 합니다. 자동 파티션을 하지 않으려면, 메뉴에서 수동으로 선택하십시오.

6.3.3.1 지원 하는 파티션 옵션

debian-installer 에서 사용하는 파티션 도구는 꽤 만능입니다. 그러면 다양한 파티션 테이블, 파일 시스템 코드 블록 장치를 사용하여 많은 다른 파티션 구성표를 만들 수 있습니다.

정확히 어떤 옵션을 사용할 수 있는지는 주로 아키텍처에 따라 다르지만, 그 다른 요인도 있습니다. 예를 들어, 내부 메모리가 제한된 시스템에서는 몇 가지 옵션을 사용할 수 없습니다. 또한 기본도 변할지도 모릅니다. 예를 들어, 대용량 하드 디스크에 대한 기본 파티션 테이블의 유형은 더 작은 하드 디스크의 것과 다른 경우가 있습니다. debconf 우선 순위가 중 또는 낮은 설치를 하는 경우에 몇 가지 옵션을 변경할 수 있습니다. 더 높은 우선 순위의 경우는 실제적인 값이기 기본적으로 사용됩니다.

설치 프로그램은 다양한 형태의 고급 파티션 및 저장 장치를 (대부분의 경우 함께) 지원합니다.

- 논리 볼륨 관리자 (LVM)

- 소프트웨어 RAID

지원하는 RAID 레벨은 0, 1, 4, 5, 6, 10 입니다.

- 암호화

- Multipath (실험적)

자세한 정보는 [Wiki](#)를 참조하십시오. 현재 다중 경로는 설치 시작시 활성화된 경우에만 사용할 수 있습니다.

다음 파일 시스템을 지원합니다.

- ext2r0, ext2, ext3, ext4

대부분의 경우 기본 파일 시스템 ext4 가 선택되어 있습니다. 파티션을 사용할 때 /boot 파티션의 기본은 ext2 가 선택됩니다.

- jfs (모든 아키텍처에서 사용하지 못 할 수도 있습니다)

- xfs (모든 아키텍처에서 사용하지 못 할 수도 있습니다)

- reiserfs (옵션; 모든 아키텍처에서 사용할 수 있는 것은 없습니다)

Reiser 파일 시스템은 더 이상 기본적으로 지원되지 않습니다. 설치 프로그램이 중 또는 낮은 debconf 우선 순위로 실행시키면 partman-reiserfs 구성 요소를 선택하여 사용할 수 있습니다. 버전 3 에서만 지원합니다.

- jffs2

플래시 메모리를 읽을 때 일부 시스템에서 사용합니다. 새로운 jffs2 파티션을 만들 수 없습니다.

- FAT16, FAT32

6.3.3.2 자동 파티션하기

자동 파티션 하는 경우, 세 가지 방법이 있습니다: 하드 디스크에서 직접 파티션을 만들거나 (전통적인 방법), 논리 볼륨 관리 (LVM) 사용하거나, 암호화한 LVM 을⁴ 사용하는 것입니다.

참고



아키텍처에 따라 (암호화한) LVM 을 사용하는 옵션을 사용하지 못 할 수도 있습니다.

LVM 이나 암호화한 LVM 을 사용하는 경우, 설치 프로그램은 대부분의 파티션을 한 개의 큰 파티션 안에 만듭니다. 이 방법의 장점으로 이 큰 파티션 안의 파티션은 나중에 비교적 쉽게 크기를 바꿀 수 있습니다. 암호화한 LVM 의 경우 특수한 암호를 알지 못하면 이 큰 파티션을 읽지 못하므로, (개인적인) 데이터에 대해 더 보안에 안전합니다.

암호화된 LVM 을 사용할 때, 설치 프로그램은 디스크에 임의의 데이터를 써 넣어서 디스크를 지웁니다. 이렇게 하면 보안을 더욱 높이지만 (디스크의 어느 부분을 사용하고 있는지 추적하는 게 불가능하고, 예전에 설

⁴ 설치 프로그램에서 LVM 볼륨 그룹을 256 비트 AES 키를 이용해 암호화하고 커널의 “dm-crypt” 기능을 사용합니다.

치했던 데이터를 지우기 때문입니다), 디스크 크기에 따라 시간이 오래 걸릴 수도 있습니다.

참고



LVM 이나 암호화한 LVM 을 사용해 자동 파티션을 하는 경우, LVM 을 설정하는 동안 파티션 테이블의 일부를 바꿔야 합니다. 이렇게 하면 해당 하드 디스크에 있는 데이터가 전부 지워지고 되돌릴 수 없습니다. 설치 프로그램이 파티션 테이블을 디스크에 쓰기 전에 계속해도 좋을지 확인 질문을 합니다.

전체 디스크에 대해 파티션을 선택한 경우 (일반적인 파티션이나 LVM(또는 암호화한 LVM) 에서), 먼저 선택한 디스크를 사용해도 되는지 묻습니다. 여러 디스크가 있는 경우 모든 디스크가 나열되어 올바르게 선택되어 있는지 확인하십시오. 순서는 평상시 사용하고 있는 것과 다를 수 있습니다. 디스크 크기를 확인할 수 있습니다. 전체 디스크에 대해 자동 파티션하기를 (일반적인 파티션이나 LVM(또는 암호화한 LVM) 에서) 선택했다면, 사용하려는 디스크를 선택하라는 질문을 맨 처음에 받게 됩니다. 디스크가 모두 목록에 있는지 확인하십시오. 디스크가 여러 개 있을 경우 반드시 올바른 디스크를 선택하십시오. 목록에 나오는 순서는 경우에 따라 달라질 수 있습니다. 디스크 용량으로도 어떤 디스크인지 확인할 수 있습니다.

여기서 결국, 디스크의 모든 데이터가 손실되었지만 디스크를 쓰기 전에 계속해도 좋을지 항상 질문을 확인합니다. 일반적인 파티션 방법을 선택하면 종료하기 전에 취소할 수 있습니다. 반면 LVM(또는 암호화한 LVM) 을 사용하는 경우 취소할 수 없습니다. 선택한 디스크에 있는 모든 데이터를 영영 잃어버리게 될 것입니다. 하지만 디스크에 바꾼 사항을 쓰기 전에 언제나 확인 질문을 합니다. 일반적인 파티션 방법을 사용한다면 끝날 때 바꾼 사항을 취소할 수 있습니다. 반면 LVM(또는 암호화한 LVM) 을 사용하는 경우에는 취소가 불가능합니다.

그다음에 아래의 표에 나온 방식 중 하나를 선택할 수 있습니다. 이 방식은 각각 장단점이 있고, 부록 C에서 설명합니다. 잘 모르겠으면 첫 번째를 선택하십시오. 명심해둬야 할 점으로, 자동 파티션할 때는 어느 정도 최소한의 빈 공간이 필요합니다. 최소 1GB 의 공간이 없으면 (방식에 따라 최소 용량은 다릅니다) 자동 파티션은 실패합니다.

파티션 방식	최소 공간	만들 파티션
모두한 파티션에 설치	600MB	/, 스왑
/home 파티션 분리	500MB	/, /home, 스왑
/home, /var, /tmp 파티션 분리	1GB	/, /home, /var, /tmp, 스왑

LVM(또는 암호화한 LVM) 을 사용해 자동 파티션 하는 경우, 설치 프로그램은 별도의 /boot 파티션을 만듭니다. 그외의 파티션 (스왑 파티션 포함) 은 LVM 파티션 안에 만듭니다.

EFI 모드에서 부팅하고 자동 파티션을 하면, EFI 부트 로더가 사용하는 FAT32 으로 포맷한 부팅 가능 파티션이 하나 더 생깁니다. 이 파티션을 EFI 시스템 파티션 (ESP) 이라고 합니다. 포맷 메뉴에서도 수동으로 파티션을 ESP 로 만드는 항목이 하나 더 생깁니다.

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

파티션 목록은 다음과 같이 나타납니다:

```
SCSI1 (0,0,0) (sda) - 6.4 GB WDC AC36400L
#1 primary 16.4 MB B f ext2 /boot
#2 primary 551.0 MB swap swap
#3 primary 5.8 GB ntfs
pri/log 8.2 MB FREE SPACE

SCSI2 (1,0,0) (sdb) - 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 f ext4 /
#6 logical 1.0 GB f ext3 /var
#7 logical 498.8 MB ext3
```

위의 예에서는 하드 드라이브가 2 개이고 여러 개 파티션으로 나뉘어져 있습니다. 첫 번째 디스크에는 빈 공간이 있습니다. 각 파티션 줄에는 파티션 번호, 종류, 크기, 추가 플래그, 파일 시스템, 그리고 마운트 위치를 (마운트 위치가 따로 있는 경우) 표시합니다. 주의: 위와 같은 파티션은 자동 파티션하면 만들 수 없고, 수동으로 만들면 이렇게 될 수도 있다는 것만 안내하는 것뿐입니다.

여기까지가 자동 파티션입니다. 자동으로 만든 파티션 테이블이 마음에 들면, 메뉴에서 파티션 나누기를 마치고 바꾼 사항을 디스크에 쓰기를 선택해서 새 파티션 테이블을 실제로 적용합니다 (이 절의 맨 뒤 부분에서

설명합니다). 마음에 들지 않으면, 파티션에 바뀐 사항을 취소를 선택해서 자동 파티션을 다시 실행하거나, 자동으로 만들어준 파티션을 아래에서 설명하는 것처럼 수동으로 바꿀 수도 있습니다.

6.3.3.3 수동 파티션하기

수동 파티션을 선택하면 기존 파티션 테이블을 마운트 위치 없이 표시되는 것을 제외하고 위와 같은 화면이 표시 됩니다. 파티션 테이블을 수동으로 만드는 방법과 새로운 데비안 시스템 파티션의 사용법에 대해서는 이 절의 나머지 부분에서 설명합니다.

파티션도 없고 빈 공간도 없는 새 디스크라면, 새로운 파티션 테이블을 만들지 여부를 물어봅니다. (그래야 새 파티션을 만들 수 있습니다.) 그다음에 “빈 공간”이라는 줄이 해당 디스크 이름 아래에 나타납니다.

빈 공간을 선택하면 새 파티션을 만들 수 있습니다. 크기와 종류 (주 파티션 아니면 논리 파티션), 위치와 (빈 공간에서 처음 아니면 끝) 같은 일련의 간단한 질문에 응답해야 합니다. 그러면 새 파티션에 대한 자세한 정보를 얻을 수 있습니다. 주요 설정은 파일 시스템 파티션에 있는 경우 스왑, 소프트웨어 RAID, LVM, 암호화한 파일 시스템으로 사용하거나 전혀 사용하지 않을지 결정하는 이용 방법: 입니다. 기타 설정은 마운트 포인트와 마운트 옵션, 부팅 가능 플래그 같은 파티션 용도에 따라 설정이 있습니다. 미리 선택된 기본값이 마음에 들지 않으면 자유롭게 원하는 것으로 변경하십시오. 예를 들어, 옵션 이용 방법: 를 선택하면 스왑, 소프트웨어 RAID, LVM, 또는 사용하지 않기를 선택할 수 있습니다. 새 파티션이 마음에 들면, 파티션 준비를 마쳤습니다를 선택하여 partman 의 메인 화면으로 돌아갑니다.

파티션에서 뭔가 바꾸려고 한다면, 해당 파티션을 선택하십시오. 그러면 파티션 설정 메뉴가 나타납니다. 새 파티션을 만들 때와 같은 화면이기 때문에, 여기서도 마찬가지로 같은 옵션을 설정합니다. 처음 보면 잘 이해가 되지 않을 수도 있는 부분이 있는데, 파티션의 크기 항목을 이용해서 파티션의 크기를 바꿀 수 있다는 점입니다. 이 기능이 동작하는 파티션은 fat16, fat32, ext2, ext3 및 스왑입니다. 이 메뉴에서 파티션을 지워버릴 수도 있습니다.

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

EFI 모드에서 부팅했지만, EFI 시스템 파티션을 선택해 포맷하지 않았다면, partman 에서 그 사실을 알아 냅니다. EFI 시스템 파티션을 만들지 않으면 다음으로 진행하지 않습니다.

partman 의 기능은 설치 프로그램 모듈에 따라 확장되지만, 시스템의 아키텍처에 따라 다릅니다. 모든 기능을 볼 수 없다면, 필요한 모듈을 모두 읽어 들였는지 확인하십시오. (예를 들어 partman-ext3, partman-xfs, 아니면 partman-lvm)

파티션이 마음에 들면, 파티션 메뉴에서 파티션 나누기를 마치고 바뀐 사항을 디스크에 쓰기를 선택 하십시오. 디스크에 바뀐 점에 대한 요약이 나타나고 그대로 파일 시스템을 만들지 확인합니다.

6.3.3.4 멀티디스크 장치 설정하기 (소프트웨어 RAID)

컴퓨터에 하드 드라이브가 여러 개 있는 경우⁵, mdcfg 명령으로 드라이브의 성능을 향상시키거나, 데이터 안정성을 높일 수 있습니다. 이렇게 하는 걸 멀티디스크 장치라고 합니다. (더 자주 쓰이는 다른 말로 RAID 라고 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 partman 에서 포맷하고 마운트 위치를 지정하고 따위를 할 수 있습니다.)

어떤 종류의 멀티디스크 장치를 만드느냐에 따라 어떤 점이 있는지 달라집니다. 현재 지원하는 종류는:

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

RAID0 은 비디오를 편집 하는 파티션에 많이 사용합니다.

RAID1 안정성을 최우선으로 할 때 적합합니다. RAID1 은 여러 개의 (보통 두 개) 같은 크기의 파티션으로 구성 되어 있고, 여기서 각 파티션은 정확히 같은 데이터를 담고 있습니다. 이게 근본적으로 3 가지의 의미가 있습니다. 첫째로 하나의 디스크가 망가지더라도, 나머지 디스크에 데이터가 미러 되어 있습니다. 둘째로 디스크의 전체 용량보다 작은 용량만 사용할 수 있습니다. (좀 더 정확히 말해, RAID 에서 가장 작은 파티션입니다.) 세 번째로 파일 읽기는 로드 밸런싱으로서 서버의 성능을 향상시켜, 파일 서버와 같이 디스크 읽기가 쓰기보다 많은 경우 부담이 줄어듭니다.

⁵ 물론 한 개의 물리 드라이브에 있는 여러 개의 파티션에서 멀티디스크 장치를 만들 수도 있지만, 그렇게 해 봤자 좋은 점이 전혀 없습니다.

RAID5 속도, 안정성, 데이터중첩을 적당히 조화시킨 것입니다. RAID5 는 들어오는 모든 데이터를 스트라이프로 나누고 각각을 하나의 디스크가 아니라 (즉 RAID0 와는 달리) 모두에게 분배합니다. RAID0 와는 다르게 RAID5 는 디스크에 쓸 정보의 패리티 정보를 계산합니다. 패리티 디스크는 고정되어 있지 않고 (고정되어 있으면 RAID4 라고 합니다) 정기적으로 바뀝니다. 디스크 중에 하나가 망가지면, 없어진 부분을 나머지 데이터와 패리티를 이용해서 계산해냅니다. RAID5 는 최소한 3 개의 파티션이 있어야 합니다. 배열에 디스크를 하나 더 사용해서 망가진 디스크를 대체하도록 만들 수도 있습니다.

이렇게 RAID5 는 RAID1 과 비슷한 정도의 안정성을 가지면서, 데이터를 덜 중복합니다. 한편 패리티계산 때문에 RAID0 보다는 쓰는 속도가 느립니다.

RAID6 RAID5 와 비슷하지만 패리티 장치를 하나가 아니라 두 개를 사용하니다.

RAID6 어레이는 디스크 실패가 두 번 일어나도 살아남을 수 있습니다.

RAID10 입력 데이터를 n 개 복사물로 만든 다음 파티션에 분배해서 같은 데이터가 같은 장치에 저장되지 않도록 합니다. n 의 기본값은 2 이지만 전문가 모드에서는 다른 값으로 설정할 수 있습니다. 사용하는 파티션 개수는 최소한 n 개입니다. RAID10 은 복사물을 분배하는 레이아웃 방식이 여러 가지가 있습니다. 기본 레이아웃은 니어카피 (Near copies) 입니다. 니어카피에서는 모든 복사물의 디스크 오프셋이 같습니다. 파카피 (Far copies) 에서는 복사물의 오프셋이 다릅니다. 오프셋카피 (Offset copies) 는 개개 복사물이 아니라 전체 스트라이프를 반복합니다.

요약하면:

종류	장치 최소 개수	예비 장치	디스크가 망가져도 버티는지?	사용 가능 공간
RAID0	2	아니오	아니오	RAID 에서 가장 작은 파티션의 크기 x 장치 개수
RAID1	2	옵션	예	RAID 에서 가장 작은 파티션의 크기
RAID5	3	옵션	예	가장 작은 파티션의 크기 x (RAID 의 장치 개수 - 1)
RAID6	4	옵션	예	가장 작은 파티션의 크기 x (RAID 의 장치 개수 - 2)
RAID10	2	옵션	예	전체 파티션 나누기 단위 복사물의 개수 (기본값 2)

소프트웨어 RAID 를 좀 더 알고 싶으시면, [Software RAID HOWTO](#) 를 읽어보십시오.

멀티디스크 장치를 만드려면, 구성할 파티션을 RAID 에 사용한다고 표시해야 합니다. (파티션 설정 메뉴의 partman 에서 용도: → RAID 의 물리 볼륨을 선택하십시오.)

참고



사용하려고 하는 파티션 방식에서 시스템이 부팅할 수 있도록 하십시오. 루트 파일 시스템으로 RAID 를 사용하는 경우 /boot 에 대한 파일 시스템을 따로 만드는 게 보통입니다. 보통 부트로는 RAID1 의 미러링을 지원합니다. (스트리핑은 지원하지 않습니다!) 그러므로 / 에는 RAID5 를 사용하고 /boot 에 RAID1 을 사용하는 것도 한 가지 방법입니다.

그다음 partman 메뉴에서 소프트웨어 RAID 설정을 선택합니다. (최소한 한 파티션을 RAID 의 물리 볼륨으로 표시해야만 메뉴가 나타납니다.) mdcfg 의 첫 번째 화면에서 멀티디스크 장치 만들기를 선택하기만 하면 지원하는 멀티디스크 장치의 종류 목록이 나옵니다. 거기에서 하나를 (예를 들어 RAID1) 고르십시오. 그다음은 여기서 어떤 종류의 멀티디스크를 선택했냐에 따라 달라집니다.

- RAID0 는 간단합니다. RAID 파티션의 목록이 나타나고 거기에서 멀티디스크를 구성할 파티션을 선택하기만 하면 됩니다.
- RAID1 은 약간 더 까다롭습니다. 먼저 멀티디스크를 구성할 활성 장치의 개수 및 예비 장치의 개수를 입력

합니다. 그리고 RAID 파티션 목록에서 무엇을 활성 파티션 과 예비 파티션으로 할지 결정합니다. 여기서 선택한 파티션 개수는 앞에서 입력한 개수와 일치해야 합니다. 걱정할 필요는 없습니다. 실수로 파티션 개수가 틀렸다고 해도, 개수가 맞아야 `debian-installer` 가 다음으로 진행합니다.

- RAID5 는 RAID1 과 비슷한 설정을 하지만, 최소한 3 개의 활성 파티션을 사용해야 한다는 점이 다릅니다.
- RAID6 는 RAID1 과 비슷한 설정을 하지만, 최소한 4 개의 활성 파티션을 사용해야 한다는 점이 다릅니다.
- RAID10 도 RAID1 과 비슷한 설정을 하지만 전문가 모드에서는 다릅니다. 전문가 모드에서는 `debian-installer` 에서 레이아웃을 물어봅니다. 레이아웃은 두 부분이 있습니다. 첫 번째는 레이아웃 종류입니다. 레이아웃 종류는 n (니어카피, near copies), f (파카피, far copies), o (오프셋카피, offset copies) 중의 하나입니다. 두 번째 부분은 데이터의 복사물 개수입니다. 이 숫자는 최소한 활성 장치의 개수만큼이어야 합니다. 그래야 모든 카피가 다른 디스크에 분배됩니다.

여러 가지 종류의 멀티디스크를 동시에 사용하는 것도 물론 가능합니다. 예를 들어 3 개의 200 GB 하드 드라이브를 멀티디스크에 사용할 때, 각 디스크에 100 GB 파티션이 두 개씩 있다고 할 때, 각 3 개 디스크의 첫 번째 파티션을 RAID0 로 묶고 (빠른 300GB 비디오 편집 파티션) 나머지 3 개 파티션을 (2 개 활성, 1 개 예비) RAID1 으로 (/home 에 사용할 안정성 높은 100GB 파티션) 사용할 수 있습니다.

원하는 대로 멀티디스크 장치를 만든 다음에, `mdcfg` 에서 마치를 선택하고 `partman` 으로 돌아가 새 로만 든 멀티디스크 장치에 파티션을 만들고 마찬가지로 마운트 위치와 같은 속성을 부여할 수 있습니다.

6.3.3.5 LVM (논리 볼륨 관리자) 설정하기

시스템 관리자나 “고급” 사용자 입장에서 컴퓨터를 사용한다면, 어떤 디스크 파티션이 (보통 가장 중요한 파티션) 공간이 부족하고 다른 파티션은 공간이 남아서 데이터를 옮기고 심볼릭 링크를 걸고 하는 등의 작업으로 상황을 해결해야 했던 경험이 있을 겁니다.

이러한 상황을 피하려면, LVM (Logical Volume Manager, 논리 볼륨 관리자) 을 사용할 수 있습니다. 간단히 말해 LVM 을 사용하면 여러 파티션을 (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, /home 이 들어있는 논리 볼륨 크기를 늘리면 됩니다. 그러면 파티션은 460GB 가 되어 남은 공간이 더 생깁니다. 물론 이에 너무 간단하게 설명한 것입니다. 아직 읽지 않으셨다면 **LVM HOWTO** 를 참고하십시오.

” `debian-installer` 의 LVM 설정은 아주 간단하고 `partman` 안에서 지원됩니다. 먼저 LVM 의 물리 볼륨으로 사용할 파티션을 표시합니다. 이 작업은 파티션 설정 메뉴에서 용도: LVM 의 물리 볼륨을 선택합니다.

`partman` 주 화면으로 돌아간 다음, LVM (논리 볼륨 관리자) 설정이라는 옵션이 새로 나타납니다. 이 옵션을 선택하면, 파티션 테이블에 바꾼 사항을 (있으면) 확인하고, 그 다음에 LVM 설정 메뉴가 나타납니다. 그 메뉴 위에 LVM 설정의 요약이 나타납니다. 메뉴는 상황에 따라 사용할 수 있는 동작만 표시합니다. 가능한 동작은:

- 설정 내역 보기: LVM 장치 구조, 이름, 논리 볼륨의 크기 등을 표시합니다
- 볼륨 그룹 만들기
- 논리 볼륨 만들기
- 볼륨 그룹 삭제
- 논리 볼륨 삭제
- 볼륨 그룹 늘리기
- 볼륨 그룹 줄이기
- 마치고: `partman` 주 화면으로 돌아갑니다

메뉴에서 이 옵션을 이용해 볼륨 그룹을 만들고 그 안에 논리 볼륨을 만드십시오.

`partman` 주 화면으로 돌아간 다음, 보통 파티션 과 마찬가지로 방금 만든 논리 볼륨이 나타납니다. (또 보통 파티션 과 마찬가지로 방법으로 이용하면 됩니다.)

6.3.3.6 암호화볼륨 설정하기

debian-installer 에서 암호화 파티션을 설정할 수 있습니다. 암호화된 파티션에 파일을 쓰면 즉시 암호화된 형태로 장치에 저장됩니다. 암호화된 데이터에 접근하려면 파티션을 처음 만들 때 사용한 암호를 입력해야 합니다. 이 기능은 노트북이나 하드 드라이브를 도난당했을 때 비밀 데이터를 보호하는 목적으로 사용됩니다. 훔친 사람은 하드 드라이브에 물리적으로는 접근할 수 있지만, 올바른 암호를 모르면 하드 드라이브의 데이터는 임의의 문자로 보입니다.

암호화해야 할 가장 중요한 두 파티션은 데이터가 들어있는 홈 파티션과, 동작 중에 비밀 데이터가 저장될 수도 있는 스왑 파티션입니다. 물론, 그 외에 하고 싶은 파티션을 암호화할 수 있습니다. 예를 들어 메일 서버나 인쇄 서버가 데이터를 저장하는 /var 라던지, 여러 가지 임시 파일을 저장해 둘 수도 있는 /tmp 를 암호화할 수 있습니다. 어떤 사람은 전체 시스템을 암호화하기도 합니다. 암호화하지 말아야 하는 한 가지 예외는 /boot 파티션으로, 암호화한 파티션에서 커널을 읽어들이는 건 현재 불가능합니다.

참고



암호화한 파티션은 암호화하지 않은 경우보다 성능이 떨어집니다. 매번 읽고 쓸 때마다 암호화하고 암호화를 해제해야 하기 때문입니다. 성능 저하 정도는 CPU 속도, 암호화 방법 및 키 길이에 달려 있습니다.

암호화를 사용하려면, 파티션 메뉴에서 빈 공간을 선택해서 새 파티션을 만들어야 합니다. 다른 방법은 기존 파티션을 (예를 들어, 일반 파티션이나 LVM 논리 볼륨이나 RAID 볼륨) 선택하는 것입니다. 파티션 설정에서, 용도: 옵션에서 암호화할 물리 볼륨 메뉴를 선택해야 합니다. 그러면 메뉴가 해당 파티션에 대한 몇 가지 암호화 옵션이 나오도록 바뀝니다.

debian-installer 에서 지원하는 암호화 방식은 dm-crypt(최근 리눅스 커널에 포함, LVM 물리 볼륨 사용 가능)입니다.

암호화 방법으로 디바이스 매퍼 (dm-crypt) 를 선택했을 때 옵션을 봅니다. 마찬가지로 무언가의 심سر우면 기본값을 사용하십시오. 이미 보안을 염두에 두고 기본값이 들어 있습니다.

암호화: aes 이 옵션은 암호화 알고리즘을 (사이퍼 (cipher)) 선택합니다. 이 알고리즘으로 파티션의 데이터를 암호화하는데 사용합니다. debian-installer 는 현재 다음과 같은 블록 사이퍼를 지원합니다: aes, blowfish, serpent, 그리고 twofish 입니다. 각 알고리즘이 얼마나 좋은지 설명하는 건이 문서의 범위를 벗어난 얘기지만, 어느 것을 결정할지도움이 되는 조언을 하면, 2000 년에 미국 표준 기술 연구소에서 21 세기에 비밀 정보를 보호할 알고리즘으로 AES 를 선택했습니다.

키 크기: 256 여기서는 암호화 키의 길이를 지정합니다. 키의 길이가 크면, 보통 그 암호화의 강력함이 더 증가합니다. 반면에 키 길이를 늘리면, 성능이 떨어집니다. 사용할 수 있는 키의 크기는 사이퍼에 따라 다릅니다.

IV 알고리즘: xts-plain64 초기화 벡터 (Initialization Vector) 혹은 IV 알고리즘은 암호화 기법에서 같은 키의 일반 텍스트 (clear text) 가 항상 유일한 암호화된 텍스트 (cipher text) 를 만들어 내도록 합니다. 이렇게 하면 공격하는 사람이 암호화된 데이터에 반복된 패턴을 통해 정보를 알아내지 못하게 됩니다.

사용할 수 있는 여러 가지 방법 중에서, 기본값인 xts-plain64 이 알려진 공격 방법에서 현재 가장 위협이 적은 방법입니다. 다른 방법은 예전에 설치한 (새 알고리즘을 사용할 수 없는) 시스템과의 호환성을 유지할 경우에 만 사용하십시오.

암호화 키: 암호 여기서 이 파티션의 암호화 키 종류를 선택합니다.

암호 나중엔 입력하게 되는 암호를 기준으로 암호화 키를 계산합니다⁶.

무작위 키 암호화한 파티션을 불러올 때 마다 임의의 데이터를 만들어서 새 암호화 키를 만듭니다. 다시 말해서, 컴퓨터를 끌 때 마다 키가 메모리에서 지워지면서 파티션에 들어 있는 데이터를 잃어버립니다. (물론 가능한 키를 모두 입력해서 추측해 낼 수 있지만, 해당 사이퍼 알고리즘에 무언가 알려지지 않은 약점이 있지 않는 한 그렇게 무작정 추측하는 건 평생 해도 다 못합니다.)

무작위 키는 스왑 파티션에 주로 사용됩니다. 암호를 기억할 필요도 없고 컴퓨터를 끌 때 스왑 파티션에 들어 있는 비밀 정보를 지울 필요가 없기 때문입니다. 하지만, 이렇게 하면 최근의 리눅스 커널에 들어 있는 “디스크에 저장하는 절전 (하이버네이션)” 기능을 사용하지 못하게 됩니다. 나중에 부팅할 때 스왑 파티션에 저장된 데이터를 복구할 방법이 없기 때문입니다.

⁶ 암호를 키로 사용한다는 건 현재는 파티션을 LUKS 로 설정한다는 뜻입니다.

데이터 지우기: 예 파티션에 암호화를 설정하기 전에 임의의 데이터로 채워 넣을지 결정합니다. 이렇게 하지 않으면 공격자가 파티션의 어느 부분을 사용하고 있고, 어느 부분을 사용하지 않는지 알아챌 수 있기 때문에 이렇게 하기를 권장합니다. 또 예전에 설치한 데이터를 복구하기 어렵게 만듭니다.⁷

암호화 파티션에 사용할 파라미터를 선택했으면, 주 파티션 메뉴로 돌아갑니다. 이제 암호화 볼륨 설정이라는 새 메뉴 항목이 생깁니다. 이 항목을 선택하면, 데이터 지우기로 표시한 파티션을 지울지 및 기타 동작에 대한 (파티션 테이블을 쓰기 등) 확인 질문을 합니다. 파티션의 크기가 크면 시간이 좀 걸릴 수 있습니다.

그다음에 암호를 사용한다고 표시한 파티션에 대해 암호를 입력합니다. 좋은 암호는 8 글자보다 길고, 영문자와 숫자와 기타 문자가 섞여 있고, 사전에 들어있는 일반적인 단어가 들어있으면 안 되고, 본인의 개인 정보에서 쉽게 유추할 수 있으면 (생일, 취미, 애완동물 이름, 가족이나 친척 이름 등) 안 됩니다.

주의



암호를 입력하기 전에, 키보드를 올바르게 설정해서 원하는 문자가 입력되도록 해야 합니다. 잘 모르겠으면, 두 번째 가상 콘솔로 바꿔서 프롬프트에서 미리 글자를 타이프 해 보면 알 수 있습니다. 그래야 나중에 설치할 때 AZERTY 키보드 배치로 입력했던 암호가 QWERTY 키보드 배치에서 맞지 않는 다든지 하는 일이 없습니다. 이런 상황은 여러 가지 원인 때문에 일어날 수 있습니다. 설치할 때 키보드 배치를 바꾸었거나, 루트 파일 시스템의 암호를 입력할 때 아직 예전에 사용했던 키보드 배치를 설정하지 못한 상태 일 경우에 이런 일이 일어날 수 있습니다.

암호화 키를 만드는 데 암호외의 방법을 사용한다면, 그 암호화 키를 이제 만들게 됩니다. 현재 설치 상태에서는 충분한 양의 엔트로피를 얻지 못했을 수도 있기 때문에, 키를 만드는 데 오래 걸릴 수 있습니다. 엔트로피를 만들어 내면 이 과정을 좀 더 빠르게 할 수 있습니다. 예를 들어서 키를 마구 누른다든지, 두 번째 가상 콘솔로 가서 쉘로 바꾼 다음에 네트워크 디스크를 사용한다든지 (파일 다운로드, 큰 파일을 /dev/null 로 보낸다든지 등) 하면 됩니다. 이 과정을 암호화할 모든 파티션에 대해서 반복합니다.

주 파티션 메뉴로 돌아오면, 암호화한 파티션이 일반 파티션과 똑같이 설정할 수 있게 보입니다. 다음의 예제는 dm-crypt 로 암호화한 볼륨입니다.

```
암호화한 볼륨(sda2_crypt) - 115.1 GB 리눅스 device-mapper
#1 115.1 GB F ext3
```

이제 해당 볼륨을 가리킬 마운트 위치를 지정하고 (기본값이 마음에 들지 않으면) 파일 시스템 종류를 설정하십시오.

주의해야 할 사항이 있습니다. 괄호 안에 들어있는 ID 와 (이 경우 sda_crypt0) 각 암호화 볼륨에 지정할 마운트 위치입니다. 나중에 새 시스템으로 부팅할 때 이 정보가 필요합니다. 일반 부팅 과정과 암호화를 사용하는 부팅 과정의 다른 점은 뒤의 7.2 절에서 설명합니다.

파티션 방법이 마음에 들면, 설치를 계속하십시오.

6.3.4 베이스 시스템 설치하기

이 단계는 문제가 발생할 가능성이 거의 없지만, 설치할 때 베이스 시스템 전체를 다운로드하고, 확인하고, 압축을 풀기 때문에 가장 많은 시간을 소모하는 단계이기도 합니다. 컴퓨터가 느리거나 네트워크 연결이 느리면 시간이 좀 걸릴 수 있습니다.

베이스 시스템 설치 중에 패키지를 풀고 설정하면서 나오는 메시지는 tty4 에서 표시합니다. 이 터미널은 왼쪽 Alt-F4 을 누르면 이용할 수 있습니다. 설치 프로그램 화면으로 돌아가려면 왼쪽 Alt-F1 을 누르십시오.

이 단계에서 압축을 풀고 설정하는 메시지는 /var/log/syslog 파일에 저장합니다. 시리얼 콘솔에서 설치하는 경우 이 파일을 보면 됩니다.

설치 도중에, Linux 커널을 설치합니다. 기본 우선 순위에서는 설치 프로그램이 하드웨어에 가장 맞는 커널을 하나 선택합니다. 우선 순위가 낮은 모드에서는, 사용 가능한 여러 가지 커널 중에서 하나를 선택할 수 있습니다.

패키지 관리 시스템을 사용하여 패키지를 설치할 때 기본적으로 해당 패키지가 권장하는 패키지도 설치됩니다. 추천 패키지는 선택 한 소프트웨어의 핵심 기능에 꼭 필요하지는 않지만, 패키지 관리자의 견으로는 일반적으로 그 소프트웨어와 같이 설치하면 기능이 확장되는 패키지입니다.

⁷ 하지만 3 글자 약자 이름의 정보 기관 사람은 광자기 미디어를 여러 번 들어 쓴다고 해도 데이터를 복구할 수 있다고 합니다.

참고



기술적인이유때문에베이스시스템설치중에설치되는패키지는그패키지의 “권장” 패키지를설치하지않습니다. 위에서말한규칙은베이스시스템을설치한다음부터 적용됩니다.

6.3.5 추가소프트웨어설치하기

이제 사용할수는있을만한상태이지만아주제한적인시스템이됩니다. 이시스템에추가로소프트웨어를설치해서사용자의필요에맞게시스템을맞출수있고, 설치프로그램에서그작업을합니다. 컴퓨터나네트워크가느리면이작업은베이스시스템설치보다도더오래걸릴수있습니다.

6.3.5.1 APT 설정하기

데비안 GNU/리눅스에서패키지를설치할때 쓰는프로그램의하나는 apt 패키지에있는 apt 입니다.⁸ aptitude 나 synaptic 등의패키지관리프로그램도사용하는사람이많으며이프로그램은 apt 에의존합니다. 처음사용자는뒤에언급한프로그램을사용하길권장합니다. 패키지검색기능이나상태확인등의기능을사용자인터페이스에내장하고있기때문입니다.

apt 에서패키지를어디서가져올지설정합니다. 설정한내용은 /etc/apt/sources.list 에기록하고설치가끝난다음이파일의내용을살펴보고바꿀수있습니다.

기본값우선순위에서설치한다면, 설정대부분을설치프로그램이자동으로처리합니다. 사용하는설치방법에따라자동으로설정하기도하고, 설치앞단계에서지정한사항에맞춰서자동으로설정하기도합니다. 설치프로그램에서자동으로보안미러사이트를추가하고, 안정버전배포판을설치한다면 “stable-updates” 업데이트서비스에대한미러사이트도추가합니다.

낮은우선순위로설치한다면(예를들어전문가모드), 더많은결정을직접내릴수있게됩니다. 보안업데이트와안정업데이트서비스를사용할지여부를지정할수있고, 아카이브의 “contrib” 및 “non-free” 섹션의패키지를추가할지도결정할수있습니다.

6.3.5.1.1 여러개의 CD/DVD 에서설치하기 큰모음의일부인 CD 나 DVD 에서설치하는경우설치프로그램에서추가로 CD 나 DVD 를읽을지여부를물어봅니다. CD 나 DVD 가더있을경우더읽어들이면설치프로그램에서그 CD 나 DVD 에들어있는패키지를사용할수있습니다.

추가 CD 나 DVD 가없어도문제가없습니다. 추가 CD 나 DVD 가꼭필요하지않습니다. 하지만(다음 섹션에서설명하는) 네트워크미러도사용하지않는경우태스크에해당하는패키지를설치할수없을수도있습니다.

참고



패키지는 CD(또는 DVD) 에유명한패키지의순서로포함되어있습니다. 즉대부분의 사람은 CD 모음의첫번째 CD 만사용하고, CD 모음의마지막 CD 에들어있는패키지를사용하는사람은거의없습니다.

이말은즉, 전체 CD 모음을구입하거나, 다운로드하거나, 굽는일은돈낭비가될수있습니다. 보통 CD 전체를사용하는일은없기때문입니다. 대부분의경우설치할때는 3 장에서 8 장정도의 CD 로설치한다음기타패키지는인터넷에서미러사이트를이용해설치하는편이 좋습니다. DVD 모음의경우에도마찬가지입니다. 첫번째 DVD 나두번째 DVD 만으로도필요한대부분의패키지가들어있습니다.

CD 나 DVD 여러장을읽어들이는경우, 드라이브에들어있는 CD/DVD 에없는패키지가필요할때마다설치프로그램에서알려줍니다. 같은 CD 모음에속한 CD 나 DVD 만읽어들이야하니주의하십시오. 읽어들이는순서는중요하지않지만, 옳은차순으로읽어들이면실패할가능성이줄어듭니다.

⁸ 참고로패키지를실제로설치하는프로그램은 dpkg 입니다. 하지만이프로그램은저수준도구에가깝습니다. apt 는고수준도구로필요할때 dpkg 를실행합니다. apt 는패키지를 CD, 네트워크등어디에서가져올지판단합니다. 또설치하려는패키지가제대로동작하려면필요한다른패키지까지자동으로설치합니다.

6.3.5.1.2 네트워크미러 사용하기 설치할때 보통 패키지 공급용으로 네트워크미러를 사용할지 여부에 대한 질문을 받게 됩니다. 기본값을 사용해도 되지만 예외의 경우가 있습니다.

전체 CD 나 DVD 에서 설치하거나, 전체 CD/DVD 이미지를 사용해 설치하는 경우가 아니라면, 네트워크미러를 사용해야 합니다. 네트워크미러를 사용하지 않으면 설치를 마쳤을 때 아주 최소한의 시스템만 설치하게 될 수 있습니다. 하지만 인터넷 연결이 느리다면 설치 단계에서 데스크톱 태스크를 선택하지 않는 편이 좋습니다.

전체 CD 한 장에서 설치하거나 전체 CD 이미지 한 개로 설치하는 경우 (DVD 가 아니고), 네트워크미러 설정은 필요없지만 그래도 강력히 추천합니다. CD 한 장에는 아주 일부의 패키지만 들어갈 수 있기 때문입니다. 인터넷 연결이 느리다면 네트워크미러를 여기에서 설정하지 말고, CD 에 들어있는 패키지를 이용해 설치를 마친 다음 (새 시스템으로 다시 시작한다면) 나머지 패키지를 설치하는 게 좋습니다.

DVD 나 DVD 이미지에서 설치하는 경우, 설치에 필요한 패키지는 첫 번째 DVD 에 모두 들어 있습니다. 앞에서 설명한 것처럼 CD 여러 장을 읽어 들인 경우에도 마찬가지입니다. 네트워크미러는 꼭 사용하지 않아도 됩니다.

네트워크미러를 추가하는 좋은 점은, 설치할 때 업데이트를 사용할 수 있다는 부분입니다. CD/DVD 모음을 만드는 동안 업데이트가 발생해서 포인트 릴리스에 포함되기도 합니다. 즉 설치할 시스템의 보안이나 안정성을 해치지 않고도 CD/DVD 모음을 더 오래 동안 사용할 수 있습니다.

정리하면, 네트워크미러를 사용하는 게 항상 좋습니다. 단 인터넷 연결이 좋지 않다면 네트워크미러를 사용하지 않는 게 좋습니다. 패키지의 현재 버전이 CD/DVD 에 들어 있다면, 설치 프로그램은 항상 CD/DVD 에 들어있는 패키지를 이용합니다. 그러므로 미러 사이트를 사용할 경우 다운로드할 데이터의 크기는

1. 다음 설치 단계에서 선택하는 태스크,
2. 그 태스크에 필요한 패키지,
3. 그 패키지 중에 사용 중인 CD/DVD 에 들어있는 패키지,
4. CD/DVD 에 들어있는 패키지의 업데이트 버전이 있다면 미러에 있는지 여부 (일반 패키지 미러이든 보안 업데이트이든 안정 업데이트 미러이든 간에) 달려 있습니다.

그 의미는 이렇습니다. 네트워크미러를 사용하지 않으려는 경우에도, 미러를 설정해 놓으면 보안 업데이트나 안정 업데이트가 있는 경우에 패키지를 인터넷에서 다운로드할 수 있습니다.

6.3.5.1.3 네트워크미러 고르기 설치 중에 네트워크미러를 사용하겠다고 선택했다면 (CD/DVD 설치할 때는 안 해도 되지만, netboot 이미지의 경우는 필수), 설치 프로그램의 앞 단계에서 선택한 국가 설정에 따라 지리적으로 가까이 있는 (즉 빠를 가능성이 높은) 미러의 목록이 표시됩니다. 기본값을 선택하면 대부분 잘 동작합니다.

“수동으로 정보 입력하기” 를 선택해 수동으로 미러를 지정할 수도 있습니다. 그 다음에 미러 호스트 이름과 (필요하다면) 포트 번호를 지정할 수 있습니다. wheezy 버전부터는 URL 베이스 값을 써야 합니다. 그러므로 IPv6 주소를 쓸 때 주소 앞뒤로 각괄호를 “[2001:db8:1]” 처럼 써야 합니다.

컴퓨터가 IPv6 전용 네트워크에 연결되어 있다면 (대다수의 사용자는 보통 이런 경우가 아님) 해당 국가의 기본 미러 사이트를 선택해도 동작하지 않을 수도 있습니다. 목록의 모든 미러 사이트는 IPv4 로 연결할 수 있지만, 아주 일부만 IPv6 로 연결할 수 있습니다. 미러 사이트의 연결 상태는 시간이 지나면서 달라질 수 있지만, 이 연결 상태에 대한 정보는 설치 프로그램 안에 들어 있지 않습니다. 해당 국가의 기본 미러 사이트에 IPv6 연결이 없는 경우 다른 미러를 시도해 보거나 “수동으로 정보 입력” 옵션을 선택하십시오. 그 다음에 “ftp.ipv6.debian.org” 를 미러 이름으로 입력하십시오. 이 사이트는 IPv6 에서 사용할 수 있습니다 (빠른 사이트는 아니더라도).

수동으로 미러 사이트를 지정할 때 한 가지 방법은 “httpredir.debian.org” 을 지정하는 방법입니다. “httpredir.debian.org” 은 물리적인 미러 사이트가 아니라 미러 디렉션 서비스로, 현재 컴퓨터와 네트워크 구조에서 가까운 실제 미러를 가리킵니다. 연결하는 프로토콜 종류에 따라 다르게 동작합니다. 예를 들어 IPv6 를 사용하면 가까이 있는 IPv6 를 사용할 수 있는 미러를 가리킵니다.

6.3.5.2 소프트웨어 선택 및 설치

설치 과정에서 소프트웨어를 추가로 선택해서 설치할 기회가 있습니다. 74365 개의 패키지에서 패키지를 일일이 선택하지 않고, 미리 정해진 소프트웨어의 묶음을 선택하고 설치하는데 중점을 두고 있습니다. 그래야 설치를 빨리 마쳐서 컴퓨터를 다양한 용도로 사용할 수 있습니다.

따라서 먼저 태스크를 선택할 수 있고, 개별 패키지는 나중에 추가합니다. 태스크는 “데스크톱 환경”, “웹서버”, “프린트 서버”⁹ 등 컴퓨터로 할 수 있는 여러 가지 작업에 해당합니다. D.2절에서 각 태스크마다 필요한 공간을 볼 수 있습니다.

설치하는 컴퓨터의 특징에 따라 미리 태스크를 선택한 경우도 있습니다. 그게 마음에 들지 않으면 태스크 선택을 해제할 수도 있습니다. 여기에서 태스크를 하나도 선택하지 않을 수도 있습니다.

작은 정보



설치 프로그램의 표준 사용자 인터페이스에서는, 스페이스바를 눌러서 태스크를 선택하고 해제할 수 있습니다.

참고



“데스크톱 환경” 태스크는 그래픽 데스크톱 환경을 설치합니다.

기본값으로 `debian-installer` 에서는 데스크톱 환경을 설치합니다. 설치할 때 다른 데스크톱 환경을 선택할 수도 있습니다. 여러 가지 데스크톱을 설치하는 것도 가능하지만, 일부 데스크톱 조합은 같이 설치할 수 없을 수도 있습니다.

이 기능은 원하는 데스크톱 환경에 필요한 패키지가 있을 경우에만 동작합니다. 전체 CD 이미지 1 장으로 설치하면 필요한 패키지 중에는 뒤쪽 CD 에 있을 수도 있으므로, 그 경우 네트워크의 미러 사이트에서 필요한 패키지를 다운로드하게 됩니다. DVD 이미지나 기타 설치 방법을 이용하는 경우에는 이 방법으로 어떤 데스크톱 환경이든 설치하는데 문제가 없습니다.

다양한 서버 태스크는 다음과 같이 소프트웨어를 설치합니다. 웹서버: `apache2`, 인쇄서버: `cups`, SSH 서버: `openssh`.

“표준 시스템” 태스크는 “표준” 우선 순위의 모든 패키지를 설치합니다. 리눅스 및 유닉스 시스템에 보통 들어있는 수많은 유틸리티가 여기에 포함됩니다. 이 태스크는 선택해 놓아야 합니다 (무슨 일을 하고 있는지 알고 있고 정말 최소한의 시스템을 원하는 경우가 아니라면).

언어 선택을 할 때 “C” 가 아닌 기본 로컬을 선택했다면, `tasksel` 에서 그 로컬에 대한 지역화 태스크가 있는지를 검사해서 관련된 지역화 패키지를 설치합니다. 해당 언어에서 사용하는 단어 목록 및 특별히 필요한 글꼴이 여기에 해당합니다. 데스크톱 환경을 선택했다면 데스크톱 환경에 필요한 지역화 패키지도 (이러한 패키지가 있다면) 설치합니다.

태스크를 선택했다면, 계속 단추를 누르십시오. 그러면 `aptitude` 에서 해당 태스크에 들어있는 패키지를 설치합니다. 프로그램 중에 사용자가 설정해야 하는 프로그램이 있으면 설치하는 중에 알려줍니다.

데스크톱 태스크는 특히 매우크므로 주의해야 합니다. 일반 CD-ROM 과 CD-ROM 에 없는 패키지가 들어있는 미러를 같이 쓰는 경우, 네트워크에서 수많은 패키지를 받아와야 할 수도 있습니다. 인터넷 연결이 느린 경우에는 받아오는데 오래 걸릴 수도 있습니다. 패키지 설치를 일단 시작하면 취소하는 방법은 없습니다.

패키지가 CD-ROM 안에 들어있는 경우에도, CD-ROM 에 들어있는 패키지보다 미러에 있는 패키지의 버전이 최신이면 미러에 있는 패키지를 받아옵니다. 안정 버전 배포판을 설치하는 경우에는 주로 포인트 릴리스 (안정 버전 릴리스 업데이트) 한 다음에 이런 일이 발생합니다. 테스트 배포판을 설치하는 경우 오래된 CD 이미지를 사용하면 이런 일이 발생합니다.

6.3.6 시스템을 부팅 가능하게 만들기

디스크 없는 (diskless) 워크스테이션에 설치하는 경우, 로컬 디스크에서 부팅하는 건 당연히 불가능한 방법이므로 이 단계는 건너뛴다.

⁹ 이 목록을 표시할 때, 설치 프로그램은 `tasksel` 을 실행할 뿐입니다. `tasksel` 은 시스템 설치를 끝낸 다음에 도연제 든지 패키지를 설치하고 지울 때 실행할 수 있습니다. `tasksel` 외에도 `aptitude` 처럼 패키지 설치와 제거를 더 자세하게 하는 프로그램을 사용할 수도 있습니다. 설치를 모두 마치고 특정 패키지를 찾으려면 `aptitude install` 패키지 명령을 실행하면 됩니다. 여기서 패키지는 찾으려는 패키지의 이름입니다.

6.3.6.1 다른 운영체제 검색

부트로더가 설치되기 전에 설치 프로그램이 이미 설치되어 있는 다른 OS의 검색을 시도합니다. 지원하는 OS가 있으면 부트로더 설치 단계 동안에 그것을 통지합니다. 또한 데비안이 외에 다른 OS를 부팅할 수도 있도록 컴퓨터를 설정합니다.

한 컴퓨터에서 여러 개의 운영체제를 부팅하는 일은 아직까지도 매우 복잡한 기술입니다. 다른 운영체제를 자동으로 찾아내고 부트로더를 설정하는 일은 아키텍처마다 다르고, 서브아키텍처마다 다르기도 합니다. 동작하지 않으면 부트로더의 문서에서 더 자세한 사항을 찾아보십시오.

6.3.6.2 시스템을 flash-kernel 로 부팅 가능하게 만들기

ARM 플랫폼에서는 공통의 펌웨어 인터페이스가 없기 때문에, ARM 장치에서 시스템을 부팅시키는 단계는 장치에 따라 크게 다릅니다. 데비안은 flash-kernel 라는 도구를 이 목적으로 사용합니다. flash-kernel 에는 시스템을 여러 가지 장치에서 부팅 시키게 만드는 절차에 관한 데이터베이스가 들어 있습니다. flash-kernel 은 현재 장치를 지원하는지 검색하고, 지원하는 경우 필요한 작업을 수행합니다.

내부의 NOR 또는 NAND 플래시 메모리에서 부팅하는 장치의 경우, flash-kernel 에서 커널 및 최 초 램 디스크를 그 내부 메모리에 씁니다. 이 방식은 예전의 armel 장치에서 특히 많이 사용합니다. 대부분의 장치는 내부 플래시 메모리에 여러 개의 커널과 램 디스크를 탑재하도록 지원하지 않기 때문에, flash-kernel 을 이러한 장치에서 실행하면 이전의 플래시 메모리 내용을 덮어 쓰게 됩니다!

시스템 펌웨어로 U-Boot 를 사용해 커널과 최 초 램 디스크를 외부 저장 장치에서 (MMC/SD 카드, USB 저장 장치, IDE/SATA 하드 디스크 등) 부팅할 수 있는 ARM 시스템의 경우, flash-kernel 에서는 사용자가 입력하지 않아도 자동 부팅할 수 있도록 적절한 부팅 스크립트를 만듭니다.

6.3.6.3 부트로더 없이 계속

이 옵션은 부트로더를 설치하지 말고 설치를 마칠 때 사용할 수 있습니다. 이렇게 하는 경우는 아마도 해당 아키텍처나 서브아키텍처에 부트로더가 없거나, 부트로더가 필요 없는 경우일 (예를 들어 기존 부트로더를 사용) 것입니다.

부트로더를 수동으로 설정하려면, /target/boot 에 설치한 커널의 이름을 확인해야 합니다. 또 이 디렉터리에 initrd 가 있는지 확인하고, 있으면 아마도 부트로더가 그 initrd 를 사용하도록 해야 할 것입니다. 그 외에 / 파일 시스템으로 사용하려는 디스크 및 파티션을 알아야 하고, /boot 가 별도 파티션이면 /boot 파일 시스템의 디스크 및 파티션도 알아야 합니다.

6.3.7 설치 마치기

이제 설치 프로그램이 할 몇 가지 작업은 데비안 설치 과정에서 마지막 단계입니다. 대부분은 debian-installer 의 뒷 정리입니다.

6.3.7.1 시스템 시계 맞추기

설치 프로그램에서 컴퓨터의 시계를 UTC 로 맞추지 물어 보기도 합니다. 다른 운영체제를 설치했는지 따위를 이용해 자동으로 UTC 로 맞추지 여부를 판단하기 때문에, 이 질문은 보통 물어 보지 않습니다.

전문가 모드에서는 UTC 로 맞추지 여부를 항상 선택할 수 있습니다.

debian-installer 가 여기에서 현재 시각을 시스템의 하드웨어 시계에 저장합니다. 앞에서 설정한 사항에 따라 UTC 로 저장할 수도 있고 지역별 시각으로 저장할 수도 있습니다.

6.3.7.2 시스템 다시 시작

설치를 시작할 때 사용했던 부팅 미디어 (CD, 플로피 등) 를 검색합니다. 그 다음에 새로 설치한 데비안 시스템으로 다시 시작합니다.

6.3.8 문제 해결

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

6.3.8.1 설치로거 저장

설치가 성공적이었으면, 설치할 때 만든 로그 파일은 새 데비안 시스템 `/var/log/installer/` 에 자동으로 작성되고 있습니다.

메인 메뉴에서 디버깅 기록 저장 옵션을 선택하면 로그 파일을 플로피 디스크나 네트워크, 하드 디스크 등의 미디어에 저장할 수 있습니다. 설치가 중시각한 문제가 발생했을 경우 다른 시스템에서 로그를 분석하거나, 로그를 설치 보고서에 첨부할 때 유용합니다.

6.3.8.2 셸 사용하기 및 로그 보기

설치하는 도중에 셸을 실행하려면 여러 가지 방법이 있습니다. 대부분의 시스템에서 시리얼 콘솔로 설치하는 게 아니라면, 가장 쉬운 방법은 왼쪽 `Alt-F2` 를 눌러¹⁰ (맥 키보드에서는 `Option-F2`) 두 번째 가상 콘솔로 전환하는 것입니다. 왼쪽 `Alt-F1` 을 누르면 설치 프로그램으로 다시 돌아올 수 있습니다.

콘솔을 전환할 수 없다면, 메인 메뉴의 셸 실행 항목을 이용해 셸을 시작할 수 있습니다. 뒤로가기 단추를 계속 눌러서 메인 메뉴로 돌아올 수 있습니다. 설치 프로그램으로 돌아오려면 `exit` 를 입력해서 셸을 닫으십시오.

램 디스크에서 부팅했기 때문에 이 셸에서는 제한적인 유닉스 유틸리티만 사용할 수 있습니다. 어떤 프로그램이 있는지는 `ls /bin /sbin /usr/bin /usr/sbin` 명령 및 `help` 를 입력해서 알 수 있습니다. 셸은 `ash` 이라고 하는 본 셸 호환 셸이고 자동 완성이나 명령어 기록 같은 훌륭한 기능도 일부 들어 있습니다.

파일을 편집하거나 파일을 보려면, `nano` 텍스트 편집기를 사용하십시오. 설치 시스템의 로그 파일은 `/var/log` 디렉터리 안에 들어 있습니다.

참고



셸에서는 실행할 수 있는 명령어로 무엇이든 할 수 있는 게 사실이지만, 셸을 사용하는 옵션은 사실 뭔가 잘못된 경우에 대비해서 혹은 디버깅용으로만 만들어 놓은 옵션입니다.

셸에서 수동으로 무언가를 할 경우에 설치 과정이 방해 를 받아서 오류가 발생하거나 설치를 끝마치지 못할 수도 있습니다. 특히 스왑 파티션을 활성화 하는 기능은 설치 프로그램에서 알아서 하도록 놔두고 셸에서 직접 하지 않도록 하십시오.

6.3.9 네트워크를 통해 설치

재미있는 컴포넌트 중의 하나로 `network-console` 이 있습니다. 설치 작업의 많은 부분을 네트워크 SSH 을 통해 수행하게 되어 있습니다. 네트워크를 사용해야 하기 때문에 최소한 네트워크 설정하기까지의 맨처음 설치 작업은 콘솔에서 해야 합니다. (이 부분은 4.4 절에 따라 자동화할 수 있습니다.)

이 컴포넌트는 주 설치 메뉴에는 기본으로 읽어들이지 않기 때문에, 이 컴포넌트를 읽어들이라고 지정해야 합니다. CD 에서 설치하는 경우에는 중간 우선 순위로 설치하거나 주 설치 메뉴가 나타나면 CD 에서 설치 프로그램 컴포넌트를 읽어들이기 을 선택 하고 `network-console: SSH` 을 사용해 원격에서 설치하기 추가 컴포넌트를 선택 합니다. 성공적으로 읽어들이면 SSH 을 사용해 원격에서 설치하기 메뉴 항목이 새로 생깁니다.

에서 새 항목을 선택 한 후 설치 시스템에 연결하기 위한 새 암호 (및 확인) 를 입력 하십시오. 여기 까지 하면 지금 원격 로그인 하라는 화면이 나옵니다. 사용자 이름은 `installer`, 방금 입력 한 암호를 사용 하십시오. 이 화면에 있는 중요한 정보로 이 시스템의 핑거프린트가 있습니다. 이 핑거프린트를 원격에서 설치 할 사람에게 안전 하게 전달 해야 합니다. 새로 생긴 이 항목을 선택 한 다음에, 설치 시스템에 연결 하는데 사용할 새 암호를 입력 하게 됩니다. 여기 까지 하면 `installer` 사용자로 방금 입력 한 암호를 이용해 원격에서 로그인 할 수 있는 방법을 알려주는 화면을 표시 합니다. 이 화면의 또 다른 중요한 정보는 시스템의 핑거프린트입니다. 이 핑거프린트를 “시스템을 원격에서 설치 할 사람에게” 안전하게 전달 해야 합니다.

로컬에서 설정을 계속 하려고 마음 을 바꿨다면, 언제 든 지 `Enter` 를 눌러서 메인 메뉴로 돌아 갈 수 있습니다. 메인 메뉴에서 다른 컴포넌트를 선택 하면 됩니다.

이제 네트워크의 다른 한편으로 가서 할 일입니다. 먼저 필요한 일은, 터미널을 UTF-8 인코딩을 쓰도록 설정 하는 일입니다. UTF-8 인코딩이 설치 시스템에서 사용하는 인코딩입니다. UTF-8 으로 설정 하지 않아도 원격 설치 가 가능 하지만 창의 테두리 라던지, 읽을 수 없는 ASCII 가 아닌 문자 처럼 표시 가 깨 질 수도 있습니다. 설치 시스템에 연결 하려 면 간단 하 다음 과 같은 명령 을 사용 하면 됩니다:

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

¹⁰ 스페이스바 왼쪽에 있는 `Alt` 키와 평선키의 `F2` 를 동시에 누르는 걸 말합니다.

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

참고



설치 프로그램에 들어있는 ssh 서버는 연결 유지 (keep-alive) 패킷을 보내지 않는 표준 설정을 사용합니다. 시스템에 대한 연결은 계속해서 열어 놓은 상태여야 합니다. 하지만 (로컬 네트워크 설정에 따라) 일정 시간 동안 아무 입력이 없으면 연결이 끊어질 수 있습니다. 이런 일이 벌어질 수 있는 흔한 상황이 SSH 클라이언트와 설치하는 서버사 이어딘가에 NAT(Network Address Translation, 네트워크 주소 변환) 가 있는 상황입니다. 연결이 어느 부분에서 끊어졌느냐에 따라 다시 연결했을 때 설치를 계속할 수도 있고 할 수 없을 수도 있습니다.

ssh 연결을 시작할 때 `-o ServerAliveInterval=` 값 옵션을 사용하면 연결이 끊어지는 일을 방지할 수 있습니다. 아니면 이 옵션을 ssh 설정 파일에 추가해도 효과가 같습니다. 하지만 이 옵션을 사용했을 때 이 옵션 때문에 연결이 끊어지는 경우도 있습니다. (예를 들어 연결 유지 (keep-alive) 패킷을 일시적인 네트워크 정지 시점에 보내는 경우 그렇습니다. 이러한 경우 ssh 연결은 다른 방법으로 복구합니다.) 그러나 이 옵션은 필요한 경우에만 사용해야 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^a 다시 연결하면 됩니다.

^a 다음 명령어로 호스트의 해당 줄을 지울 수 있습니다: `ssh-keygen -R < 호스트 이름 | IP 주소 >`.

로그인한 다음에 최초 화면이 나오면 거기에서 메뉴 시작과 쉘 시작 중의 하나를 선택할 수 있습니다. 전자의 경우는 설치 프로그램의 메인 메뉴로 가게 되고, 거기에서 로컬에서와 마찬가지로 설치 작업을 계속할 수 있습니다. 후자는 쉘을 실행해서 원격 시스템을 살펴 보고 문제점을 수정할 수 있습니다. 설치 메뉴는 한 개의 SSH 세션만 열어야 합니다. 하지만 쉘의 경우에는 여러 개를 열어도 됩니다.

주의



SSH 을 통해 원격으로 설치를 시작하면, 다시 로컬 콘솔로 돌아가서 설치하면 안 됩니다. 그렇게 하면 새로 설치할 시스템의 설정이 망가질 수 있습니다. 설정이 망가지면 설치가 실패하거나 새로 설치한 시스템에 여러 가지 문제가 발생할 수 있습니다.

6.4 없는 펌웨어 읽어들이기

2.2절에서 설명한 것처럼, 일부 장치는 펌웨어를 읽어들이어야 합니다. 대부분 그런 장치는 펌웨어가 없으면 동작하지 않습니다. 가끔 기본적인 기능은 동작하고 추가적인 기능에서만 펌웨어가 필요하기도 합니다.

없는 펌웨어가 장치 드라이버에 필요한 경우, `debian-installer` 에서 대화상자를 표시해 없는 펌웨어를 읽어들이도록 안내합니다. 이 옵션을 선택하면 `debian-installer` 는 펌웨어 파일이나 펌웨어가 들어 있는 패키지가 있는지 검사합니다. 있으면 펌웨어를 적당한 위치에 (`/lib/firmware`) 복사하고 드라이버 모듈을 다시 읽어들이는 것입니다.

참고



어떤 장치를 검사하고 어떤 파일 시스템을 지원하느냐는 아키텍처, 설치 방법, 설치 단계에 따라 달라집니다. 설치 앞 단계에서는 FAT 로 포맷한 플로피 디스크나 USB 메모리에서 펌웨어 읽어들이는 대부분 성공합니다.

펌웨어 없이도 해당 장치가 동작한다거나 설치할 때 그 장치가 필요 없는 경우, 펌웨어 읽어들이를 건너뛸 수도 있습니다.

debian-installer에서는 설치 중에 올라간 커널 모듈에서 필요한 펌웨어만 요청합니다. 모든 드라이버가 debian-installer에 포함되지 않았기 때문에 (특히 radeon 이 없음), 일부 장치는 설치가 끝날 때 썸되어 도 설치 가시작 될 때 나다를 것 없이 사용 되지 않을 수도 있습니다. 결과적으로 하드웨어의 일부는 완전히 사용할 수 없을 수도 있습니다. 이런 상황의 심 되거나 궁극하다면 새로 부팅 하는 시스템의 dmesg 명령 결과에서 “firmware” 라는 말을 찾아보십시오.

6.4.1 미디어 준비하기

공식 CD 이미지는 사용 제한이 있는 펌웨어는 들어 있지 않습니다. 이러한 펌웨어를 읽어들이는 방법 중 가장 많이 사용하는 방법은 USB 메모리 같은 이동식 장치에서 펌웨어를 읽어들이는 경우입니다. 아니면 사용 제한이 있는 펌웨어가 들어 있는 비공식 CD 빌드가 <http://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware/> 위치에 있습니다. USB 메모리 (아니면 하드 드라이브 파티션, 플로피 디스크) 를 준비하려면, 펌웨어 파일이나 패키지는 해당 미디어의 최상위 디렉터리나 /firmware 디렉터리에 들어 있어야 합니다. 추천하는 파일 시스템은 FAT 입니다. (FAT 는 설치 앞 단계에서 도 지원하는 파일 시스템 이므로.)

많이 사용하는 펌웨어 패키지를 tar 나 zip 으로 굳힌 것이 다음의 사이트에서 사용할 수 있습니다:

- <http://cdimage.debian.org/cdimage/unofficial/non-free/firmware/>

해당 버전의 tarball 또는 zip 파일을 다운로드 미디어 파일 시스템에 배포하면 됩니다.

필요한 펌웨어가 TAR 파일에 없으면 해당 펌웨어 패키지를 아카이브에서 (아마도 non-free 섹션에서) 다운로드 할 수도 있습니다. 아래에서 흔히 사용하는 펌웨어 패키지 목록을 요약해 놓았습니다. 아래 목록은 전체 목록이 아니고 펌웨어 패키지가 아닌 패키지도 들어 있습니다:

- <http://packages.debian.org/search?keywords=firmware>

펌웨어 파일을 직접 미디어에 복사할 수도 있습니다. 펌웨어 파일을 이미 설치해 놓은 시스템에서 가져올 수도 있고 하드웨어 공급사가 제공해 줄 수도 있습니다.

6.4.2 펌웨어 및 설치한 시스템

설치할 때 읽어들인 펌웨어는 설치한 시스템에도 복사됩니다. 그래야 펌웨어가 필요한 장치가 설치한 시스템으로 다시 시작한 후에 도 올바르게 동작합니다. 하지만 설치한 시스템의 커널 버전이 다른 경우 펌웨어를 읽어들이지 못할 가능성이 낮지만 있습니다.

펌웨어를 펌웨어 패키지에서 읽어들인 경우, debian-installer 는 이 패키지를 설치한 시스템에서도 설치하고 APT 의 sources.list 에 non-free 섹션을 추가합니다. 이렇게 하면 펌웨어의 새 버전이 나왔을 때 자동으로 업데이트 하는 장점이 있습니다.

설치할 때 펌웨어 읽어들이를 건너뛰면, 수동으로 펌웨어를 설치하기 전에는 해당 장치가 설치한 시스템에서 동작하지 않습니다.

참고



펌웨어 파일에서 펌웨어를 설치한 경우, 그 펌웨어는 설치한 시스템에 복사되고 해당 펌웨어 패키지를 (패키지가 있다면) 설치하지 않는 한 자동으로 업데이트 되지 않습니다.

Chapter 7

새로운데비안시스템으로부팅하기

7.1 진실의시간

시스템이 혼자 힘으로 하는 최초의 부팅을 전기엔지니어는 “스모크테스트 (smoke test)” 라고 부릅니다.

시스템이 제대로 시작하지 않았다 하더라도 당황하지 마십시오. 설치가 성공적으로 완료됐다면 시스템이 데비안을 시작하는 것을 방해하는 비교적 작은 문제가 있을 가능성이 높습니다. 대부분의 경우 그런 문제는 보통 다시 설치하지 않아도 해결할 수 있습니다. 부팅시 문제를 해결하는 한 가지 방법은 설치 프로그램에 내장된 응급 복구 모드 (8.7절 참조) 를 사용하는 것입니다.

만약 데비안 및 Linux 에 생소한 경우에, 경험있는 사용자의 도움이 필요할지도 모릅니다. 64-bit ARM 처럼 많이 사용하지 않는 아키텍처의 경우는 [debian-arm 메일링리스트](#) 를 이용하는 게 가장 좋은 방법입니다. [5.4.5절](#) 를 따라 설치 보고서 를 제출할 수도 있습니다. 보고서는 문제를 명확하게 설명 표시된 모든 메시지를 넣어 다른 사람이 문제의 원인을 파악되도록 하십시오.

7.2 암호화 볼륨 마운트하기

설치할 때 암호화 볼륨을 만들고 마운트 위치를 지정했다면, 부팅할 때 각각의 볼륨에 대해 암호를 입력하게 됩니다.

dm-crypt 로 암호화한 파티션의 경우 부팅할 때 다음과 같이 물어봅니다:

```
Starting early crypto disks... part_crypt(starting)
Enter LUKS passphrase:
```

첫 번째 줄에서, part 는 실제 파티션의 이름입니다. (예를 들어 sda2 나 md0.) 여기에서 과연 어떤 볼륨의 암호를 실제 로 입력해야 하는지 의문이 들 것입니다. /home 아니면 /var 일까요? 물론, 암호화 볼륨이 1 개 뿐이라면, 이 볼륨을 설정할 때 사용한 암호를 입력하면 됩니다. 설치할 때 암호화 볼륨을 여러 개 설정했다면, [6.3.3.6절](#) 의 마지막 단계에서 적어 놓은 메모를 잘 가지고 있어야 합니다. part_crypt 에 해당되는 사항과 거기에 해당하는 마운트 위치를 적어 놓지 않았다면 새로 설치한 시스템의 /etc/crypttab 과 /etc/fstab 에서 찾아볼 수도 있습니다.

이 프롬프트는 암호화한 루트 파일 시스템을 마운트할 때는 약간 다릅니다. 시스템을 부팅할 때 사용하는 initrd 를 만들 때 어떤 initramfs 만들기 프로그램을 사용했느냐에 따라 다릅니다. 아래의 예제는 initramfs-tools 로 initrd 를 만들 경우에 대한 예제입니다:

```
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Enter LUKS passphrase:
```

암호를 입력할 때는 아무런 글자도 (별표조차도) 나타나지 않습니다. 암호를 잘못 입력하면 두 번 더 시도할 수 있습니다. 세 번째 시도에서 틀리면 부팅 과정에서 해당 볼륨을 건너뛰고 다음 파일 시스템으로 넘어갑니다. 자세한 정보는 [7.2.1절](#) 부분을 보십시오.

암호를 모두 입력하면 부팅은 평소처럼 계속 진행합니다.

7.2.1 문제 해결

암호가 틀려서 암호화 볼륨을 마운트하지 못할 경우, 부팅한 다음에 수동으로 마운트해야 합니다. 여러 가지 경우가 있습니다.

- 첫번째 경우는 루트 파티션입니다. 올바르게 마운트하지 않으면, 부팅 과정이 멈추게 되고 컴퓨터를 다시 시작해서 암호를 다시 입력해야 합니다.
- 가장 쉬운 경우는 /home 이나 /srv 처럼 데이터가 들어있는 암호화 볼륨입니다. 부팅하고 수동으로 볼륨을 마운트하면 됩니다.
하지만 dm-crypt 의 경우에는 약간 까다롭습니다. 먼저 해당 볼륨을 다음 명령어로 device mapper 에 등록해야 합니다:

```
# /etc/init.d/cryptdisks start
```

이렇게 하면 /etc/crypttab 에 들어있는 모든 볼륨을 검색하고 암호를 올바르게 입력할 때 마다 /dev 디렉터리 아래에 적당한 장치를 만듭니다. (이미 등록된 볼륨은 건너뛰므로, 걱정하지 말고 이 명령어를 여러 번 실행해도 됩니다.) 올바르게 등록을 마치면 해당 볼륨을 평소와 다름없이 마운트할 수 있습니다:

```
# mount /마운트_위치
```

- 꼭 필요하지 않은 시스템 파일이 들어있는 볼륨 중에 하나라도 (/usr 혹은 /var) 마운트할 수 없는 경우, 그래도 시스템이 부팅하고 수동으로 볼륨을 마운트할 수 있습니다. 하지만 현재 런 레벨의 각종 서비스를 (다시) 시작해야 할 수도 있습니다. 서비스가 제대로 시작하지 않았을 가능성이 높기 때문입니다. 가장 쉬운 방법은 컴퓨터를 다시 시작하는 것입니다.

7.3 로그인

패키지 설치가 끝나면 로그인 프롬프트를 표시합니다. 설치할 때 입력한 개인 로그인 및 암호를 이용해 로그인합니다. 그러면 이제 시스템을 사용할 준비를 다 마쳤습니다.

처음 설치한 사용자라면 문서를 살펴보고 싶을 것이고, 이 문서는 시스템을 시작할 때부터 시스템 안에 설치되어 있습니다. 현재 여러 개의 문서 시스템이 있고, 여러 가지 종류의 문서를 통합하는 작업을 진행하고 있습니다. 다음과 같은 방법으로 문서 보기를 시작할 수 있습니다.

설치한 프로그램에 들어있는 문서는 /usr/share/doc/ 아래에, 그 프로그램의 이름으로 (정확히 말해 그 프로그램이 들어있는데 비안 패키지의 이름으로) 된 서브 디렉터리에 들어 있습니다. 하지만 이보다 자세한 문서는 별도의 문서 패키지에 들어 있고, 이 패키지는 보통 기본으로 설치하지 않습니다. 예를 들어 apt 패키지 관리 도구에 관한 문서는 apt-doc 혹은 apt-howto 패키지에 들어 있습니다.

또 /usr/share/doc/ 아래에 특수 폴더가 몇 개 더 있습니다. 리눅스 HOWTO 는 /usr/share/doc/HOWTO/en-txt/ 안에 .gz (압축한) 형식으로 들어 있습니다. dhelp 를 설치하면 /usr/share/doc/HTML/index.html 파일에 브라우저로 볼 수 있는 문서 목록이 있습니다.

다음 명령어로 텍스트 기반 브라우저를 사용하면 간단히 이 문서를 볼 수 있습니다:

```
$ cd /usr/share/doc/
$ w3m .
```

w3m 명령 다음에 나오는 점은 현재 디렉터리의 내용을 표시한다는 뜻입니다.

그래픽 데스크톱 환경을 설치했다면, 그 환경의 웹 브라우저를 이용할 수 있습니다. 프로그램 메뉴에서 웹 브라우저를 실행해서 주소창에 /usr/share/doc/ 을 입력하고 Enter 를 누르십시오.

info 명령어 또는 man 명령어 명령을 입력하면, 명령어 프롬프트에서 사용할 수 있는 대부분의 명령에 대한 문서를 볼 수 있습니다. help 를 입력하면 shell 명령어에 대한 도움말을 표시합니다. 명령어 뒤에 --help 옵션을 붙이면 짙막한 명령어 사용법을 표시합니다. 명령어의 결과가 화면 위로 지나가 버린다면 | more 를 명령어 뒤에 붙이면 화면 위로 스크롤되지 나가기 전에 출력을 일시 정지할 수 있습니다. 어떤 글자로 시작하는 명령어의 목록을 보려면 그 글자를 입력하고 탭을 두 번 누릅니다.

Chapter 8

다음단계및그다음에 할일

8.1 시스템끄기

실행중인데비안 GNU/리눅스시스템을종료할때컴퓨터의앞이나뒤에있는리셋스위치를눌러서다시시작하거나, 전원을꺼버려서는안됩니다. 데비안 GNU/리눅스는적절한절차로종료해야하고, 그렇지않으면파일이지워지거나디스크에손상이올수있습니다. 데스크톱환경을실행하는경우정상적인시스템종료 (또는다시시작) 를허용하는응용프로그램메뉴에서사용할수 “로그아웃” 의 옵션이있습니다.

다른방법으로 Ctrl-Alt-Del 키조합을누를수도있습니다. 이키조합이동작하지않으면, 마지막방법은 root 로로그인해서필요한명령어를입력하는방법입니다. reboot 로시스템을리부팅합니다. 전원을끄지않고 halt 로시스템을멈춥니다¹. 컴퓨터의전원을끄려면 poweroff 또는 shutdown -h now 명령어를사용합니다. systemd init 시스템은같은기능을하는다른명령어가있습니다. 예를들어 systemctl reboot 또는 systemctl poweroff 명령어를쓸수있습니다.

8.2 유닉스를처음접한다면

유닉스를처음접한다면, 책을사서읽어보는게 좋을수도있습니다. 또여러가지 좋은정보가 [Debian Reference](#)에있습니다. [list of Unix FAQs](#)에는기억에남을만한훌륭한참고자료로여러가지유즈넷문서 목록이들어있습니다.

Linux 는 Unix 의구현중하나입니다. [Linux Documentation Project\(LDP\)](#)에서 Linux 에 관한많은 HOWTO 와온라인서적을정리합니다. 이문서대부분은컴퓨터에직접설치할수도있습니다. doc-linux-html 패키지 (HTML 버전) 아니면 doc-linux-text 패키지 (ASCII 버전) 를설치후 /usr/share/doc/HOWTO 파일을읽어보십시오. 여러가지언어의 LDP HOWTO 도데비안패키지로이용하실수있습니다.

8.3 데비안에익숙해지기

데비안은다른배포판들과약간다릅니다. 다른배포판에서리눅스에잘아는분도시스템을최상의상태로유지하기위해서는데비안에대해알아두어야할수있습니다. 이장에서는데비안에익숙해지기위해도움이되는자료를소개합니다. 데비안의사용법을일일이설명하지는않고, 성급한사람을위해서시스템개요만설명합니다.

8.3.1 데비안패키지시스템

알아야할가장중요한개념은데비안패키지시스템이있습니다. 기본적으로시스템의대부분은패키지시스템관리아하에있습니다. 이패키지시스템에서관리하는디렉터리는:

- /usr (/usr/local 를제외)
- /var (/var/local 을만들고다음의디렉터리를자유롭게사용하는것은가능합니다)
- /bin

¹ SysV init 시스템에서는 halt 는 poweroff 와같은효과를냈지만, (jessie 이후기본인) systemd 를 init 시스템으로사용할경우그효과가다릅니다.

- /sbin
- /lib

예를들어, /usr/bin/perl 을작동한파일로대체해도동작에는문제가없고, 나중에 perl 패키지를업데이트하면당신이설정 한파일은패키지로대체하게됩니다. 이것을방지하려면 aptitude 에서패키지를 “hold” 하는작업을합니다.

APT 는가장훌륭한패키지설치도구의하나입니다. 명령행방식의 apt 을사용할수도있고, 텍스트전체화면방식의 aptitude 를사용할수도있습니다. APT 를이용해 main, contrib, non-free 모두에서설치할수있고, 또표준패키지외에특정국가로수출이제한되어있는패키지까지도설치할수있습니다.

8.3.2 데비안용추가소프트웨어

기본데비안설치에빠진공식및비공식소프트웨어저장소가있습니다. 여기에는많은사람이중요하다고생각하는소프트웨어가들어있습니다. 이추가저장소에대한정보는데비안위키의 [The Software Available for Debian's Stable Release](#) 페이지에있습니다.

8.3.3 프로그램버전관리

같은이름의프로그램이여러가지버전이있는경우 update-alternatives 에서관리합니다. 여러버전의프로그램을관리하고있다면, update-alternative 맨페이지를읽어보십시오.

8.3.4 CRON 작업관리

시스템관리자권한으로하는작업은설정파일이므로, 모두 /etc 안에들어있어야합니다. 루트권한으로매일, 매주, 매달실행할 CRON 작업이있으면, 그스크립트를 /etc/cron.{daily,weekly,monthly} 아래넣으십시오. 이스크립트는 /etc/crontab 에서실행하고, 알파벳순서로하나씩실행합니다.

한편 (1) 특정사용자로실행할 CRON 작업이있거나 (2) 특정시간이나특정주기로실행할작업이있으면 /etc/crontab 을사용하거나, 아니면더좋은방법으로 /etc/cron.d/아무개 를사용할수있습니다. 이파일에는 CRON 작업을실행할사용자를지정하는필드가따로있습니다.

어떤방법을사용하든파일을편집하기만하면 CRON 에서자동으로인식하고처리합니다. 다른명령어를실행할필요가없습니다. 더자세한정보는 cron(8), crontab(5), /usr/share/doc/cron/README. Debian 파일을참고하십시오.

8.4 그외의읽을거리및정보

특정프로그램에대한정보를보려면, man 프로그램 명령을실행해보시고, 아니면 info 프로그램 명령을실행해보십시오.

/usr/share/doc 에유용한문서가많이있습니다. 특히, /usr/share/doc/HOWTO 과 /usr/share/doc/FAQ 에흥미로운정보가많이있습니다. 버그를보고하려면 /usr/share/doc/debian/bug* 을참조하십시오. 특정프로그램에대한데비안특정문제에대해읽으려면 /usr/share/doc/(패키지)/README.Debian 을참조하십시오.

[Debian 웹사이트](#)에는데비안에관한많은문서가있습니다. 특히, [Debian GNU/Linux FAQ](#)와 [Debian 참조](#)를참조하십시오. [Debian 문서프로젝트](#)에는데비안문서에대한더많은인덱스가포함되어있습니다. 데비안의커뮤니티에사용자가서로지원하고있습니다. 데비안의메일링리스트에가입하려면 [메일링리스트 가입](#) 페이지를참조하십시오. 마지막으로 [Debian 메일링리스트아카이브](#)에는데비안에관한수많은정보가포함되어있습니다.

GNU/리눅스에관한정보는보통 [Linux Documentation Project](#)에보면있습니다. 여기에 GNU/리눅스시스템에관한하우투및다른훌륭한정보가들어있는링크가있습니다.

8.5 시스템에전자메일준비하기

오늘날, 전자메일은많은사람의삶의중요한일부가되어있습니다. 전자메일을사용할수있도록설정하기전에전자메일을설정을선택하고정확하게설정되었는지가중요합니다. 데비안유틸리티에서기본적인사항을설명합니다.

전자메일시스템은크게세가지로구성됩니다. 우선사용자가실제로전자메일을작성하고읽는데사용하는 Mail User Agent(MUA) 가있습니다. 그리고한컴퓨터에서다른컴퓨터로전자메일을전달하는 Mail

Transfer Agent(MTA) 가 있습니다. 마지막으로 받은 전자메일을 사용자의 편지함으로 전달하는 일을 하는 Mail Delivery Agent(MDA) 가 있습니다.

이 세 가지 기능은 각각 다른 프로그램에서 담당할 수도 있고, 한 개나 두 개 프로그램에서 담당할 수도 있습니다. 또한 전자메일 종류별로 여러 가지 프로그램이 담당할 수도 있습니다.

리눅스 및 유닉스 시스템에서는 전통적으로 mutt 가 아주 널리 쓰이는 MUA 입니다. 전통적인 리눅스 프로그램 대부분과 마찬가지로 텍스트 기반 프로그램입니다. mutt 는 보통 exim 이나 sendmail 을 MTA 로 쓰고 procmail 을 MDA 로 씁니다.

그래픽 데스크톱 시스템의 증가와 인기로, GNOME 의 evolution, KDE 의 kmail 또는 Mozilla 의 thunderbird (데비안에서는 icedove²로 사용 가능) 같은 그래픽 전자메일 프로그램 사용이 더 일반적으로 되어 있습니다. 이 프로그램에는 MUA, MTA 및 MDA 의 기능이 결합되어 있지만, 기존의 Linux 도구와 함께 사용할 수 있습니다.

8.5.1 기본 전자메일 설정

그래픽 메일 프로그램을 사용한다고 해도, 데비안 GNU/리눅스 시스템 전통적인 MTA/MDA 를 설치하고 정확하게 설정하는 것은 중요한 일입니다. 시스템에서 동작하는 여러 가지 유틸리티³ 시스템 관리자에게 문제 (또는 잠재적인 문제) 와 바뀐 사항을 알릴 때 전자우편으로 중요한 알림을 보낼 수 있기 때문입니다.

이러한 이유 때문에 exim4 및 mutt 패키지는 기본으로 설치합니다. (설치할 때 “표준” 태스크 선택을 제한한다면 설치하지 않습니다.) exim4 는 MTA/MDA 가 결합된 프로그램으로 작지만 매우 유연합니다. 기본값으로 시스템의 로컬 메일만 처리하도록 설정되어 있고 시스템 관리자에게 (루트 계정) 보내는 메일은 설치할 때만드는 사용자 계정으로 배달합니다⁴.

시스템 전자메일을 배달할 때 그 메일은 /var/mail/계정_이름 파일 뒤에 추가됩니다. 해당 전자메일은 mutt 로 읽을 수 있습니다.

8.5.2 시스템 외부에 전자메일 보내기

앞에서 말한 것처럼, 설치한 데비안 시스템은 시스템 내부의 전자메일을 처리하도록 설정되어 있고, 타인에게 메일을 보내거나 외부에서 메일을 받도록 설정되지 않습니다.

exim4 에서 외부 전자메일을 처리하도록 하려면, 다음의 기초 설정 옵션을 참고하십시오. 테스트 메일을 올바르게 보내고 받는지 확인하십시오.

그래픽 메일 프로그램을 사용하고 여러분의 인터넷 서비스 회사 혹은 여러분의 회사의 메일 서버를 사용한다면, 외부 전자메일을 처리하려고 exim4 를 설정할 필요가 전혀 없습니다. 사용하려고 그래픽 메일 프로그램을 설정해서 전자메일을 보내고 받는 데 이용 할 서버를 올바르게 설정하십시오. (이 설정 방법은 이 안내서의 범위를 벗어납니다.)

그러나, 그 경우에는 올바르게 전자메일을 보낼 수 있도록 각 유틸리티를 설정해야 할지도 모릅니다. 그러한 유틸리티 하나나 데비안 패키지에 대한 버그 보고서를 보내는 기능을 하는 프로그램인 reportbug 가 있습니다. 기본적으로 버그 리포트를 제출하려면 exim4 를 사용할 수 있습니다.

reportbug 가 외부 메일 서버를 사용하도록 설정하려면, reportbug --configure 명령을 실행해서 MTA 가 있는 지 여부를 묻는 질문에 “no” 라고 답하십시오. 그러면 그다음에 버그 보고서를 보낼 때 사용할 SMTP 서버를 입력할 수 있습니다.

8.5.3 Exim4 MTA 설정하기

시스템에서 외부 전자메일까지 처리하게 만드려면, exim4 패키지를 다시 설정해야 합니다⁵:

```
# dpkg-reconfigure exim4-config
```

명령어를 실행하면 (루트 권한으로), 설정 파일을 작은 파일로 내놓지 여부를 물어봅니다. 잘 모르겠으면 기본 옵션을 선택하십시오.

그다음에 자주 사용하는 여러 가지 전자메일 시나리오를 표시합니다. 필요한 사항에 가장 가까운 시나리오를 선택하십시오.

인터넷 사이트 시스템이 네트워크에 연결되어 있고 메일은 SMTP 를 사용해 직접 보내고 받습니다. 다음 화면에서 시스템의 메일 이름이나 사용할 도메인의 목록 등 기본적인 질문을 물어봅니다.

² thunderbird 가 데비안에서 icedove 로 개명된 이유는 라이선스 문제와 관계가 있는데 자세한 내용은 이 설명서에서 다루지 않습니다.

³ 예를 들어: cron, quota, logcheck, aide, ...

⁴ 루트 메일을 전달하는 계정은 /etc/aliases 에 설정되어 있습니다. 물론 일반 사용자 계정을 만들지 않았다면 메일은 루트 계정 자체에 배달합니다.

⁵ 물론 exim4 를 지우고 다른 MTA/MDA 로 바꿔버릴 수도 있습니다.

스마트호스트가메일보내기 이시나리오에서는다른컴퓨터를통해메일을보냅니다. 그다른컴퓨터를 “스마트호스트” 라고하고, 메일을목적지에보내는역할을담당합니다. 스마트호스트는보통받은메일을저장해놓고있으므로, 여러분의컴퓨터를계속연결해놓지않아도됩니다. 메일을받을때는 fetchmail 과같은프로그램으로가져올수도있습니다.

보통스마트호스트는인터넷서비스회사의메일서버를말합니다. 전화접속사용자의경우에는분명히인터넷서비스회사의서버를말합니다. 스마트호스트는회사의메일서버가될수도있고, 내부네트워크에있는다른시스템이될수도있습니다.

스마트호스트가메일을보내고, 로컬메일없음 이옵션은앞의옵션과같지만로컬전자메일도메인에대한메일을처리하지않습니다. 시스템내부의메일은 (예를들어시스템관리자에게보내는메일은) 계속처리합니다.

로컬배달 시스템이기본값으로이렇게설정되어있습니다.

지금설정안함 무슨일을하는지확실히알고있는경우에만선택합니다. 이옵션을사용하면메일시스템을설정되지않은상태로남겨놓습니다. 설정하지않으면메일을보내거나받을수없을뿐만아니라시스템유틸리티가보내는중요한메시지를놓칠수있습니다.

이시나리오가모두맞지않거나, 아니면좀더세밀한설정을하고싶다면, 설치를다마친다음에 /etc/exim4 디렉터리아래의설정파일을설정할수있습니다. exim4 에대한좀더자세한정보는 /usr/share/doc/exim4 에들어있습니다. README.Debian.gz 파일에 exim4 에설정에대한더자세한정보가들어있고어디에서더자세한문서를찾을수있는지쓰여있습니다.

공식적인도메인이름이없을때인터넷으로직접메일을보내면받는서버의스팸방지장치때문에메일이거부될수도있습니다. ISP 의메일서버를사용하는게 좋습니다. 정말로메일을직접보내고싶다면, 기본으로만드는전자메일주소가아닌다른주소를사용하십시오. /etc/email-addresses 에한줄추가하면전자메일주소를바꿀수있습니다.

8.6 새커널컴파일하기

새로운커널을컴파일하려고하는이유는무엇일까요? 데비안에서는표준으로들어가있는커널에많은기능을지원하고있기때문에대부분커널을컴파일할필요는없습니다. 또한데비안에서여러가지커널을제공합니다. 그래서더자신의하드웨어에적합한다른커널이미지패키지가있는지여부를먼저확인하면됩니다. 하지만다음목적때문에새로운커널을컴파일하는일도유용합니다:

- 특별한하드웨어가필요할때, 혹은기존커널에서하드웨어충돌이발생할때
- 기본커널에서지원하지않는커널옵션사용 (예를들어큰메모리지원)
- 필요없는드라이버를지우는식으로커널을최적화해부팅시간빠르게하기
- 모듈여러개로구성된커널대신에이미지하나로된커널만들기
- 새로업데이트된커널이나개발버전커널사용
- 리눅스커널공부하기

8.6.1 커널이미지관리

커널컴파일을꺼려하지마십시오. 재미있는일이고그만큼좋은점이있습니다.

데비안방식으로커널을컴파일하는데필요한패키지: fakeroot, kernel-package, linux-source-2.6, 나머지는이미설치되어있는패키지 (완전한목록은 /usr/share/doc/kernel-package/README.gz 을참조).

이방법을이용해커널소스에서.deb 패키지파일을만들고, 커널에들어있지않은모듈이있으면그모듈의.deb 파일도커널에맞춰서만듭니다. 이방법이커널이미지를관리하는더좋은방법입니다. /boot 안에 커널, System.map 및빌드할때사용한커널설정파일이들어갑니다.

반드시 “데비안방식” 으로커널을컴파일할 필요는없습니다. 하지만커널을관리하는패키지시스템을이용하는편이더안전하고더쉽습니다. 사실 linux-source-2.6 대신 Linus 가배포하는커널소스를그대로사용할수있지만그경우에도 kernel-package 를사용하여컴파일하십시오.

/usr/share/doc/kernel-package 안에 kernel-package 를사용하는방법에대한문서가있습니다. 여기에서는간단히어떤단계를거쳐야하는지만설명합니다.

여기서부터는, 루트권 한없이 자기 홈 디렉터리에 커널 소스를 풀었다고 가정합니다⁶. 또 커널 버전이 3.16 버전이라고 가정합니다. 커널 소스를 풀려는 디렉터리에서, `tar xf /usr/src/linux-source-3.16.tar.xz` 명령으로 커널 소스를 풀고, 새로 만들어진 `linux-source-3.16` 디렉터리로 들어가십시오.

그리고 커널을 설정합니다. X11 설치를 완료하고 나서, X11 을 실행 중인 경우 `make xconfig` 를, 그렇지 않은 경우 `make menuconfig` 를 실행합니다. (뒤쪽꺼에는 `libncurses5-dev` 가 설치되어 있어야 합니다). 온라인 도움말을 펼쳐 읽고 신중하게 선택하십시오. 일반적으로 해당 장치 드라이버 (이더넷 카드, SCSI 컨트롤러 등 하드웨어 주변 장치를 관리하는 소프트웨어) 를 포함하는 것이 좋습니다. 주의하십시오: 특정 하드웨어에 관계없는 기타 옵션에서 잘 이해할 수 없는 것은 기본값 상태로 두십시오. 또한 “Loadable module support” 의 “Kernel module loader” (기본적으로 선택되어 있지 않음) 을 꼭 선택하십시오. 그렇지 않으면, 데비안 설치에 문제가 발생할 수도 있습니다.

소스 트리에 필요 없는 파일을 지워서 `kernel-package` 파라미터를 초기화하십시오. `make-kpkg clean` 명령을 실행합니다.

이제 커널을 컴파일합니다: `fakeroot make-kpkg --initrd --revision=1.0.custom 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 호스트 이름) 넣을 수 있습니다. 컴퓨터 속도에 따라 커널을 컴파일하는데 좀 오래 걸릴 수도 있습니다.

일단 컴파일이 끝나면 다른 패키지와 같은 방법으로 직접 만든 커널을 설치할 수 있습니다. 루트권 한으로 `dpkg -i ../linux-image-3.16-서브아키텍처_1.0.custom_arm64.deb` 명령을 실행하십시오. 서브아키텍처 부분은 커널 옵션에 따라 달라지는 서브아키텍처 이름입니다. `dpkg -i` 을 실행하면 커널 및 커널과 같이 사용하는 보조 파일을 설치합니다. 커널 문제를 해결하는데 도움이 되는 `System.map` 파일이라든지 설정 사항을 담은 `/boot/config-3.16` 등의 파일을 설치합니다. 커널 패키지는 부트로더를 자동으로 업데이트해서 새로 설치한 커널을 사용하도록 설정하기도 합니다. 모듈 패키지를 만들었다면 그 패키지도 설치해야 합니다.

이제 시스템을 다시 시작합니다. 위에서 발생한 메시지를 잘 읽어보시고, `shutdown -r now` 명령을 실행하십시오.

데비안 커널과 커널 컴파일에 대한 더 자세한 정보는 [Debian Linux Kernel Handbook](#) 를 참조하십시오. `kernel-package` 에 관한 보다 자세한 정보는 `/usr/share/doc/kernel-package` 의 문서를 읽어보십시오.

8.7 손상된 시스템 복구하기

정성을 기울여 설치한 시스템에 문제가 발생해서, 부팅하지 않을 수 있습니다. 무언가 바꾸다가 부트로더 설정이 망가졌을 수도 있고, 새로 설치한 커널이 부팅하지 않을 수도 있고, 디스크가 방사선에 맞아서 `/sbin/init` 파일의 일부가 바뀌어졌을 수도 있습니다. 어떤 원인이든, 이 문제를 바로 잡으려면, 바로 잡는 동안 작업할 시스템이 하나 필요하고, 응급 복구 모드를 사용하는 게 좋습니다.

응급 복구 모드로 들어가려면, 부팅 메뉴에서 `rescue` 를 선택하거나, `boot:` 프롬프트에서 `rescue` 라고 입력하거나 부팅 파라미터로 `rescue/enable=true` 부팅 파라미터로 부팅하십시오. 설치 프로그램의 맨 처음 화면이 나타나고, 지금 모드가 설치 모드가 아니라 응급 복구 모드라는 사실을 알리는 말이나 나타납니다. 너무 걱정하지 마십시오. 시스템을 엮어 쓰지 않습니다! 설치 프로그램에 들어있는 하드웨어 검색 기능을 디스크, 네트워크 장치 등을 복구하는 용도로 이용하는 것 뿐입니다.

파티션 도구 대신에 시스템의 파티션 목록이 나타나고, 그 중에 하나를 선택합니다. 보통 복구를 하려는 루트 파일 시스템이 들어있는 파티션을 선택합니다. 디스크에 있는 파티션은 물론 RAID 나 LVM 장치에 있는 파티션도 선택할 수도 있습니다.

가능하다면, 설치 프로그램은 선택한 파일 시스템에서 필요한 복구를 수행하려면 셸 프롬프트를 제공하도록 되어 있습니다.

선택한 루트 파일 시스템에서 셸을 실행할 수 없는 경우 (예를 들어 파일 시스템이 망가진 경우), 경고 메시지를 표시하고 설치 환경 안에서 셸을 실행합니다. 이 환경에서는 그리 많은 프로그램을 쓸 수는 없지만, 시스템을 복구하는 용도로는 충분합니다. 선택한 루트 파일 시스템은 `/target` 디렉터리에 마운트되어 있습니다.

어떤 경우이든, 셸을 빠져 나가면 시스템이 다시 시작합니다.

망가진 시스템을 복구하는 일은 매우 어려울 수도 있습니다. 그리고 이 안내서는 무언가 망가졌을 때 그걸 어떻게 고치는지에 대한 모든 사항을 설명하지 않습니다. 문제에 부딪힌 경우, 전문가에게 문의하십시오.

⁶ 다른 위치에 커널 소스를 풀고 빌드할 수도 있지만, 특정 권한이 없어도 할 수 있는 가장 쉬운 방법입니다.

Appendix A

설치방법

이문서는 debian-installer 를사용해 64-bit ARM 용데비안 GNU/리눅스 stretch(“arm64”) 배포판을설치하는방법을설명합니다. 간단히설치절차만설명한문서로, 설치할때필요한정보대부분이들어있습니다. 더많은정보가필요한경우더자세히설명한이문서의다른부분으로링크되어있습니다.

A.1 들어가기전에

debian-installer 는아직베타버전입니다. 설치할때버그를발견하면, 5.4.5절부분의방법을이용해버그를알려주십시오. 이문서로알수없는궁금한사항이있으면 debian-boot 메일링리스트에 (debian-boot@lists.debian.org) 알리거나 IRC 에 (OFTC 네트워크의 #debian-boot 채널) 질문해주십시오.

A.2 설치프로그램시작하기

CD 이미지의링크는 [debian-installer 홈페이지](#) 를보십시오. debian-cd 팀에서만든 CD 이미지는 [데비안 CD 페이지](#) 에있습니다. CD 를구할수있는곳은, 4.1절부분을참고하십시오.

일부설치방법에는 CD 이미지외에다른이미지가필요합니다. [debian-installer 홈페이지](#) 에다른이미지의링크가있습니다. 4.2.1절는데비안미러에서이미지를찾는방법에대해설명하고있습니다.

다음섹션에서는설치방법에따라어떤이미지를받아야하는지자세히설명합니다.

A.2.1 CDROM

netinst CD 이미지는 stretch 버전을 debian-installer 를이용해설치하는데많이사용하는설치이미지입니다. 이이미지는 CD 로부팅하고나머지패키지를네트워크로설치하기때문에이름이 'netinst' 입니다. 이이미지에는설치프로그램을실행하고최소한의 stretch 시스템을구성하는데필요한기본패키지가들어있습니다. 필요하면설치할때네트워크가필요없는전체 CD 이미지를받을수도있습니다. 설치할때는전체 CD 세트에서첫번째 CD 만필요합니다.

마음에드는이미지를다운로드하고 CD 를굽습니다.

A.2.2 네트워크부팅

네트워크로 debian-installer 를부팅하는것도가능합니다. 아키텍처와 netboot 설정에따라다양한방법의네트워크부팅방법이있습니다. netboot/ 에들어있는파일을이용해 debian-installer 를네트워크부팅합니다.

A.2.3 하드디스크부팅

이동식매체를사용하지않고기존의하드디스크 (거기에다른 OS 가있어도상관없습니다) 를사용하여설치프로그램을시작할수있습니다. hd-media/initrd.gz, hd-media/vmlinuz 및데비안 CD 이미지를하드디스크의최상위디렉터리에다운로드하십시오. CD 이미지의파일이름이 .iso 로끝나는것을확인하십시오. 이것은 initrd 를사용하여 Linux 부팅시문제입니다.

A.3 설치

설치 프로그램을 시작하면 초기 화면이 나타납니다. Enter 를 누를 수도 있고, 아니면 다른 부팅 방법이나 파라미터 설명을 읽어보십시오. (5.3절 참고.)

잠시 후에 언어를 선택합니다. 화살표 키로 언어를 선택하고 Enter 를 눌러 계속 진행합니다. 그 다음에 해당 언어를 사용하는 국가 중에서 자기 국가를 선택하는 부분이 나타납니다. 여기 나오는 짧은 리스트에 자기 국가가 없다면, 세계의 모든 국가 목록에서 선택할 수도 있습니다.

키보드 설정을 확인합니다. 잘 모르겠으면 기본값을 선택하십시오.

이제 데비안 설치 프로그램이 하드웨어를 검사하고, CD, 플로피, USB 등에서 나머지 부분을 읽어들이는 동안 기다리십시오.

그 다음에 설치 프로그램은 네트워크 하드웨어를 검사하고 DHCP 를 통해 네트워크 설정을 합니다. 네트워크에 연결하지 않았거나 DHCP 가 없다면, 네트워크를 수동으로 설정할 수 있습니다.

다음 단계는 시계 및 시간대 설정입니다. 설치 프로그램이 인터넷의 타임 서버에 연결해서 시계를 올바르게 맞춥니다. 시간대는 앞에서 선택한 국가에 따라 결정합니다. 한국에 여러 개의 시간대가 있는 경우에 만 시간대를 물어봅니다.

시계와 시간대를 설정한 다음 사용자 계정을 만듭니다. 먼저 “루트” (관리자) 계정의 암호를 입력하고 일반 사용자 계정 생성에 필요한 정보를 입력합니다. “루트” 사용자의 암호를 지정하지 않으면 이 계정을 사용할 수 없게 되지만, 나중에 새로 설치한 시스템에 sudo 패키지를 설치해 관리 작업을 수행할 수 있습니다.

디스크를 파티션 할 차례입니다. 먼저 전체 디스크나 디스크의 남은 공간을 자동으로 파티션 나누기 할 수 있습니다 (6.3.3.2절 참고). 이 방법은 처음 설치하는 사용자나 급한 사용자에게 추천합니다. 자동 파티션을 사용하지 않으려면 메뉴에서 수동으로 선택하십시오.

다음 화면에 파티션 테이블, 파티션의 포맷, 마운트 위치가 나타납니다. 수정하거나 지울 파티션을 선택하십시오. 자동 파티션을 할 경우, 메뉴에서 파티션 나누기를 마치고 바뀐 사항을 디스크에 쓰기를 선택하면 있는 그대로 사용합니다. 반드시 파티션 한 개를 스왑 공간으로 배정하고, 또한 파티션을 / 에 마운트하십시오. 파티션 나누기 사용하는 방법을 보려면 6.3.3절 부분을 보십시오. 부록 C에서 파티션 나누기에 대해 더 자세히 설명합니다.

파티션을 포맷하고 베이스 시스템 설치를 시작합니다. 약간의 시간이 걸릴 수 있습니다. 다 끝나면 커널을 설치합니다.

앞에서 설치한 베이스 시스템은 완전히 동작하지만 최소한의 시스템입니다. 이 시스템을 좀 더 쓸모 있게 만드려면 다음 단계에서 태스크를 선택해 패키지를 추가로 설치합니다. 패키지를 설치하기 전에 apt 를 설정해서 어디서 패키지를 가져올지 지정합니다. “표준 시스템” 태스크가 기본으로 선택되어 있고 보통이 태스크는 설치해야 합니다. 그래픽 데스크톱을 사용하려면 “데스크톱 환경” 태스크를 선택하십시오. 이 단계에 관해 6.3.5.2절 부분을 참고하십시오.

마지막 단계는 부트로더 설치입니다. 컴퓨터에서 다른 운영 체제를 찾으면, 그 운영 체제를 부팅 메뉴에 추가하고 알려줍니다.

이제 debian-installer 에서 설치 과정이 끝났다고 표시합니다. CD-ROM 또는 기타 부팅 미디어를 꺼내고 Enter 를 눌러 다시 부팅하십시오. 그러면 새로 설치한 시스템으로 로그인할 수 있습니다. 7장에 설명되어 있습니다.

설치 과정에 대해 더 알고 싶으시면, 6장 부분을 보십시오.

A.4 설치 보고서를 보내 주십시오

debian-installer 를 이용해 설치가 성공했다면, 시간을 내서 설치 보고서를 보내 주십시오. 보고서를 제출하려면 가장 간단한 방법으로, reportbug 패키지를 설치하시고 (apt install reportbug), reportbug 를 8.5.2절에 설명한 대로 설정하고, report installation-reports 를 실행하십시오.

설치를 마치지 못했다면 데비안 설치 프로그램의 버그 때문으로 예상됩니다. 설치 프로그램을 개선하려면 개발자에게 알려야 하므로, 시간을 내서 알려 주십시오. 문제를 보고 할 때 설치 보고서를 사용할 수 있습니다. 설치가 완전히 실패한다면 5.4.4절 부분을 참고하십시오.

A.5 그리고 마지막으로...

데비안 설치가 즐겁고, 데비안의 좋은 점을 느끼셨길 바랍니다. 8장을 읽는 것이 좋습니다.

Appendix B

미리설정을이용한설치자동화

이부록에서는 debian-installer 의질문에대한대답을미리설정해서설치를자동화하는방법을설명합니다.

이부록에서사용한설정은 <http://d-i.alioth.debian.org/manual/example-preseed.txt>에 들어있는예제파일에서도구할수있습니다.

B.1 소개

미리설정을하면설치프로그램이실행하는동안나오는질문에대한답을수동으로입력할필요없이미리설정해놓을수있습니다. 이렇게하면대부분경우의설치를완전히자동화할수있고, 보통설치할때는쓸수없는기능을사용할수도있습니다.

미리설정은꼭필요하진않습니다. 미리설정파일을비워놓으면, 설치프로그램은일반수동설치와똑같은방식으로동작합니다. 질문을미리설정해놓으면그기준에따라설치가다르게동작합니다.

B.1.1 미리설정방법

미리설정에사용할수있는세가지방법이있습니다: initrd, 파일그리고 네트워크입니다. initrd 미리설정은어떤설치방법과사용해도동작하고더많은부분을미리설정할수있지만, 가장많이준비해야합니다.

다음은어떤미리설정을어떤설치방법에서사용할수있는지나타낸표입니다.

설치방법	initrd	file	network
CD/DVD	예	예	예 ¹
netboot	예	아니오	예
hd-media	예	예	yes ¹

1 미리설정방법여러가지사이의중요한차이점은, 바로미리설정파일을읽어들이고처리할시점입니다. initrd 미리설정의경우이시점은설치를시작하는시점으로, 맨처음으로질문하기도전의시점입니다. 그다음에커널명령행에쓴미리설정이시작합니다. 그러므로커널명령행 (부트로더설정이든부트로더에서부팅할때수동으로입력하든) 편집하면 initrd 에있는설정보다우선하게만들수있습니다. 파일미리설정의경우 CD 혹은 CD 이미지를읽어들인다음입니다. 네트워크미리설정의경우네트워크를설정한다음입니다.

¹ 하지만네트워크에연결된경우에만그렇게하고, preseed/url 을적절히설정합니다

중요

당연히미리설정파일을읽어들이기도전에처리하는질문은미리설정할수없습니다. (여기에는중간혹은낮은우선순위에서만표시되는질문까지포함합니다. 예를들어 첫번째하드웨어검색이그렇습니다.) 약간불편하지만이질문을피하는방법은부팅 파라미터를통해전달하는방법입니다. **B.2.2**절부분에서설명합니다.

미리설정기능이시작하기전에나타나는질문을간단히피하려면, 설치프로그램을 “자동” 모드로시작하면됩니다. 이렇게하면미리설정하기전에물어보는질문을 네트워크연결한다음으로늦춰서미리설정이가능합니다. 자동모드에서는설치를 필수우선순위로진행해서중요하지않은많은질문을건너뛸니다. 자세한정보는 **B.2.3**절부분을참고하십시오.

B.1.2 한계

debian-installer 에서사용하는대부분의질문들이방법으로미리설정할수있지만, 몇가지알아둬야할 예외가있습니다. 전체디스크를다시파티션하거나, 디스크의빈공간을사용해야합니다. 기존의파티션을 이용할수없습니다.

B.2 미리설정사용하기

먼저미리설정파일을만들고, 그파일을사용하려는위치에놓아야합니다. 미리설정파일만들기는이부록의 뒤부분에서설명합니다. 네트워크미리설정이나플로피혹은 USB 메모리에서파일을읽어들이는경우, 미리 설정파일의위치는다른것입니다. CD 나 DVD 에파일을포함하려면 ISO 이미지를다시만들어야합니다. 미리설정파일을 initrd 에포함하는일은이문서의범위를벗어납니다. debian-installer 개발자문서를참고하십시오.

미리설정파일을만들때기초로사용할수있는예제 파일이 <http://d-i.aliioth.debian.org/manual/example-preseed.txt>에있습니다. 이파일은이부록에포함된설정을이용해서만들었습니다.

B.2.1 미리설정파일읽어들이기

initrd 미리설정을사용하려면, preseed.cfg 파일을 initrd 의루트디렉터리에놓기만하면됩니다. 자동으로설치프로그램이이파일이있는지검사한다음읽어들이니다.

다른 preseed 방법은부팅할때어떤파일을읽어들이는지설치프로그램에지정해야합니다. 일반적으로커널부팅파라미터로전달합니다. 부팅할때수동으로넣거나부트로더설정파일 (예: syslinux.cfg) 편집해서커널의 append 줄의끝에매개변수를추가합니다.

부트로더설정에서미리설정파일을지정하는경우설정을변경하면설치를시작할때 ENTER 를누를필요가없습니다. syslinux 에서는이설정을하는데, syslinux.cfg 에서시간을 1 로합니다. 부트로더설정에서미리설정파일을지정하는경우, 설정을바꾸면부팅할때 Enter 를누를필요도없게할수있습니다. syslinux 의경우 syslinux.cfg 파일에서 timeout 을 1 로하면됩니다.

설치프로그램이올바른미리설정파일을읽어들이도록, 파일의체크섬을지정할수도있습니다. 현재는 md5sum 만사용가능하고, 미리설정파일을읽어들이었을때지정한 md5sum 과체크섬이맞지않으면그파일을사용하지않습니다.

지정해야 하는 부팅 파라미터:

- 네트워크 부팅의 경우:


```
preseed/url=http://호스트/파일에/대한/경로/preseed.cfg
preseed/url/checksum=5da499872becccfeda2c4872f9171c3d
```
- 아니면


```
preseed/url=tftp://호스트/파일에/대한/경로/preseed.cfg
preseed/url/checksum=5da499872becccfeda2c4872f9171c3d
```
- 이미지를 다시 만든 CD로 부팅하는 경우:


```
preseed/file=/cdrom/preseed.cfg
preseed/file/checksum=5da499872becccfeda2c4872f9171c3d
```
- USB 미디어에서 설치하는 경우(미리 설정 파일을 USB 메모리의 맨 위

디렉터리에 넣으십시오)

```
preseed/file=/hd-media/preseed.cfg
preseed/file/checksum=5da499872becccfeda2c4872f9171c3d
```

preseed/url 은 간단히 url 로, preseed/file 은 file 로, preseed/file/checksum 은 preseed-md5 로 줄여서 부팅 파라미터로 쓸 수 있습니다.

B.2.2 부팅파라미터로 미리설정하기

일부 단계에서는 미리설정 파일을 사용할 수 없는 경우에도, 설치를 완전히 자동화할 수 있습니다. 설치 프로그램이 부팅할 때 미리설정 값을 하나하나 파라미터로 넘길 수 있습니다.

미리설정 기능을 사용하려는 게 아니더라도, 특정 질문에 대한 답을 지정하고 싶으면 부팅 파라미터를 이용할 수 있습니다. 이 안내서의 다른 곳에 부팅 파라미터가 유용한 예제가 몇 가지 있습니다.

debian-installer 에서 사용할 값을 설정하려면, 이 부록의 예제에 들어있는 미리설정 변수에 대해 변수/대한/경로 = 값 형식으로 넘깁니다. 설치할 시스템의 패키지를 설정하는데 어떤 값을 사용한다면, 그 변수의 소유자²를 소유자: 변수/대한/경로 = 값 형식으로 씁니다. 소유자를 지정하지 않으면 해당 변수의 값은 설치한 시스템의 debconf 데이터베이스에는 들어가지 않으므로 해당 패키지의 설정에 사용하지 않게 됩니다.

이런 식으로 질문을 미리설정 하면, 그 질문을 하지 않습니다. 질문에 대해 특정 기본 값을 지정하면서, 질문을 하게 만들고 싶으면, 연산자에 “=” 이 아니라 “?” 라고 쓰십시오. B.5.2 절 부분도 참고하십시오.

부팅 파라미터에서 자주 사용하는 변수 몇 개는 짧게 쓸 수 있습니다. 이 부록에 있는 예제에서는 그렇게 짧은 형식이 있으면 완전한 변수명을 쓰지 않고 짧은 형식을 사용합니다. 예를 들어 preseed/url 변수는 예제에서 url 로 씁니다. 또 예제에서 tasks 라고 쓰면 task sel:task sel/first 에 해당합니다.

부팅 옵션의 “---” 는 특별한 의미가 있습니다. 마지막 “---” 뒤에 오는 커널 파라미터는 설치한 시스템의 부트로더 설정으로 복사되어 들어갑니다. (설치 프로그램에서 설치하는 부트로더가 지원 하는 경우.) 설치 프로그램은 (미리설정 옵션과 마찬가지로) 설치 프로그램에서 인식 하는 옵션만 모두 걸러냅니다.

참고



현재 리눅스 커널은 (2.6.9 이후) 최대 32 개까지의 명령행 파라미터와 32 개까지의 환경 파라미터만 쓸 수 있습니다. (설치 프로그램에서 기본으로 추가 하는 파라미터 포함해서 32 개입니다.) 이보다 많으면 커널이 멋어버립니다. (이보다 오래 된 버전의 커널에서는 파라미터 개수 제한이 이보다 더 작습니다.)

대부분 설치할 때 (vga=normal 같은) 기본 옵션 중에 안 써도 되는 옵션이 있습니다. 그러면 미리설정 옵션을 몇 개 더 쓸 수 있습니다.

참고



부팅 파라미터에서는 공백이 들어간 값을 지정 할 수 없습니다. 따옴표로 묶어도 할 수 없습니다.

B.2.3 자동모드

부팅 프롬프트에서 매우 간단한 명령으로 임의의 복잡한 자동설치를 할 수 있는데 비안설치 프로그램의 기능이 있습니다.

이 기능은 Automated install 부팅메뉴를 선택해서 시작합니다. 일부 아키텍처나 일부 부팅 방식에서는 auto 라고 하기도 합니다. 여기서는 auto 가 파라미터가 아니라, 부팅메뉴의 선택을 말하고, 부팅 프롬프트에서 부팅 파라미터로 추가 하는 걸 말합니다.

다음은 부팅 프롬프트에서 사용할 수 있는 예입니다.

```
auto url=autoserver
```

² 어떤 debconf 변수 (혹은 서식) 의 소유자는 보통 그 debconf 서식이 들어있는 패키지의 이름을 말합니다. 서식과 변수는 소유자가 여러 개일 수도 있습니다. 그래서 패키지의 설정까지 지워버릴 때 소유자 정보를 이용해 해당 서식과 변수까지 지워줄지 여부 결정됩니다.

이것은 DNS 에서 autosever 이름을확인할수있고 (DHCP 에서로컬도메인이있으면뒤에붙이고), 그시스템이 DHCP 서버임을가정합니다. example.com 이라는도메인의사이트가 DHCP 를설정하면, http://autosever.example.com/d-i/stretch/./preseed.cfg 에서 preseed 파일을가져옵니다.

URL 의뒤쪽부분은 (d-i/stretch/./preseed.cfg) auto-install/defaultroot 에서가져옵니다. 기본값으로여기에는 stretch 디렉터리가들어있습니다. 다음버전에서는그버전에해당되는코드네임을이값으로사용할예정이고, 그러면사람이정해진방법을통해다음버전으로업그レード할수있습니다. ./ 부분은맨위디렉터리를가리키는것으로, 지정한해당경로에 (preseed/include 및 preseed/run 에서사용) 대한상대값입니다. 이를이용해파일을완전한 URL 로지정할수도있고, /로시작하는경로로지정할수도있고, 마지막미리지정파일이있던위치에대한상대경로로지정할수도있습니다. 이점을이용하면전체스크립트를완전히새로운위치에 옮겨도문제가없는포터블한스크립트를구성할수있습니다. 예를들어웹서버에있던파일을 USB 메모리에 옮겨도문제가없게됩니다. 이에제의미리지정파일에서는 preseed/run 값을 /scripts/late_command.sh 라고지정하면 http://autosever.example.com/d-i/stretch/./scripts/late_command.sh 에서파일을가져옵니다.

로컬네트워크에 DHCP 혹은 DNS 서버따위가없거나 preseed.cfg 에대한기본값경로를바꾸고싶으면, URL 을직접사용할수도있습니다. ./ 부분을사용하지않으면경로의첫부분에대한 (즉 URL 에서서번째 /) 경로가됩니다. 다음은로컬네트워크에서최소한의지원이있어야동작하는예제입니다:

```
auto url=http://192.168.1.2/파일에/대한/경로/mypreseed.file
```

위와같이하면다음과같이동작합니다:

- URL 의프로토콜부분을생략하면 http 라고가정합니다.
- 호스트이름에점이없으면, DHCP 에서넘겨준도메인을뒤에붙입니다.
- 호스트이름뒤에 /가없으면기본경로를뒤에붙입니다.

URL 을지정하는것외에, debian-installer 의동작과직접관계없지만미리설정파일의 preseed/run 에서지정한스크립트로넘길사항을지정할수도있습니다. 현재관련예제는 auto-install/classes 로 classes 라고줄여쓸수있습니다. 다음과같이사용합니다:

```
auto url=example.com classes=클래스_A;클래스_B
```

클래스는설치하려는시스템의종류를지정하거나, 지역화를지정합니다.

이개념을확장할수도있고, 확장하는경우 auto-install 네임스페이스를사용하는게보기 좋습니다. 즉 auto-install/style 과같이스크립트에서사용할수있습니다. 이렇게해야겠다고생각이들면, debian-boot@lists.debian.org 메일링리스트에알려주십시오. 그래야네임스페이스충돌을피하고, 여러분의파라미터에해당하는줄임말을추가할수도있습니다.

auto 부팅레이블이모든아키텍처에서정의된것은아닙니다. 커널명령행에파라미터두개, auto=true priority=critical 이라고추가하기만하면같은효과를거둘수있습니다. auto 커널파라미터는 auto-install/enable 의줄임말이고미리설정할수도록로컬및키보드질문을뒤로늦춥니다. 또 priority는 debconf/priority 의줄임말이고 critical 로설정하면필수우선순위보다낮은우선순위의질문을하지않게됩니다.

그외에 DHCP 를사용할때설치를자동화하면서관심이질만한옵션은다음과같습니다: interface=auto netcfg/dhcp_timeout=60 이라고하면처음찾은네트워크인터페이스를이용하고 DHCP 요청에대한응답을좀더오래기다립니다.

작은정보



예제스크립트와클래스등, 이프레임워크를사용하는다양한예제가 [개발자의웹사이트](#)에있습니다. 이사이트에있는예제는미리설정을통해기발하고다양한짓기능을수행하는예제도있습니다.

B.2.4 미리설정할때쓸모있는줄임말

미리설정을사용할경우 (자동모드) 다음줄임말을사용하면 좋습니다. 다음줄임말은질문이름에대한줄임말일뿐이고, 물론값도같이쓸수있습니다. 예를들어 auto=true 혹은 interface=eth0 처럼쓸수있습니다.

priority

debconf/priority

fb	debian-installer/framebuffer
language	debian-installer/language
country	debian-installer/country
locale	debian-installer/locale
theme	debian-installer/theme
auto	auto-install/enable
classes	auto-install/classes
file	preseed/file
url	preseed/url
domain	netcfg/get_domain
hostname	netcfg/get_hostname
interface	netcfg/choose_interface
protocol	mirror/protocol
suite	mirror/suite
modules	anna/choose_modules
recommends	base-installer/install-recommends
tasks	tasksel:tasksel/first
desktop	tasksel:tasksel/desktop
dmraid	disk-detect/dmraid/enable
keymap	keyboard-configuration/xkb-keymap
preseed-md5	preseed/file/checksum

B.2.5 미리설정파일을지정하는데 DHCP 서버사용하기

미리설정파일을네트워크에서다운로드하도록지정하려면 DHCP 를사용할수있습니다. DHCP 에서파일이름을지정할수있습니다. 보통이것은 netboot 파일이지만, URL 형식으로되어있으면네트워크 preseed 을지원하는설치방식은그 URL 에서파일을내려받은다음설정파일로사용합니다. 다음은 ISC DHCP 버전 3 서버에 dhcpd.conf 설정하는예제입니다 (데비안의 isc-dhcp-server 패키지).

```
if substring (option vendor-class-identifier, 0, 3) = "d-i" {
    filename "http://host/preseed.cfg";
}
```

위의예에서자신을” d-i” 라고주장하는 DHCP 클라이언트에만이파일이름을전달하므로일반 DHCP 클라이언트에는아무런영향이없습니다. 특정호스트에대해서만설정하면네트워크의모든시스템을미리설정하지않게만들수있습니다.

DHCP preseed 을사용하는좋은방법은자신의네트워크에는데비안미러같은 preseed 값만지정하는것입니다. 자신의네트워크에이방법으로설치하면선택한적당한미러를자동으로가져오지만나머지설치과정은직접지정할수있습니다. DHCP preseed 를이용한데비안의완전자동설치는충분히주의하여야할필요가없습니다.

B.3 미리설정파일만들기

미리설정파일은 debconf-set-selections 명령어에서사용하는형식으로되어있습니다. 미리설정파일의일반적인형식은다음과같습니다:

```
<소유자> <질문 이름> <질문 형식> <값>
```

The file should start with #_preseed_V1

미리설정파일을작성할때지켜야할규칙이있습니다.

- 형식과값사이에한개의공백이나탭을넣으십시오. 공백이나탭을이보다많이쓰면값에그문자가들어갔다고취급합니다.
- 백슬래시 (“\”) 를이음문자로맨뒤에붙여서한줄을여러줄로나눌수있습니다. 줄을나눌때좋은부분은질문이름뒤부분입니다. 나쁜부분은형식과값사이입니다. 나뉜줄이한줄로합쳐질때앞/뒤의공백문자는모두공백하나로취급됩니다.
- 설치프로그램에서만사용하는 debconf 변수 (서식) 의경우, 소유자를 “d-i” 라고해야합니다. 설치한시스템에서사용할변수를미리설정하려면, 해당 debconf 서식이들어있는패키지의이름을사용해야합니다. 소유자가 “d-i” 가아닌변수만설치한시스템의 debconf 데이터베이스에적용됩니다.

- 보통질문을미리설정할때번역한값이아니라영어로된올바른값을사용해야합니다. 하지만일부질문의경우번역한값을사용해야합니다. (예를들어 partman 에서.)
- 어떤질문은눈에보이는영문텍스트가아니라코드를값으로받습니다.
- Start with #_preseed_V1

미리설정 파일을만드려면, B.4절에 들어있는예제 파일을기초로 시작하는방법이 가장쉽습니다.

다른 방법으로수동으로설치하고다시부팅한다음에, debconf-utils 패키지에서 debconf-get-selections 명령으로 debconf 데이터베이스및설치프로그램의 cdebconf 데이터베이스를한파일로만드는방법이있습니다:

```
$ echo "#_preseed_V1" > file
$ debconf-get-selections --installer >> file
$ debconf-get-selections >> file
```

하지만, 이런방법으로만든파일에는미리설정하면안되는항목도들어있으므로, 보통예제 파일에서시작하는방법이더 좋습니다.

참고



이방법은설치가끝났을때설치프로그램의 cdebconf 데이터베이스가설치한시스템의 /var/log/installer/cdebconf 에들어있다는점을이용한것입니다. 하지만이데이터베이스에는비밀정보가들어있을수도있기때문에루트만이파일을읽을수있게되어있습니다.

installation-report 패키지를지우면 /var/log/installer 디렉터리및그안의모든파일을시스템에서지웁니다.

각질문에대해올바른값이무엇인지알려면, 설치할때 nano 로 /var/lib/cdebconf 파일의내용을보여줍니다. 원본서식을보려면 templates.dat 파일을보고, 현재값과각변수에할당된값을보려면 questions.dat 파일을보여줍니다.

설치하기전에미리설정파일의형식이올바른지확인하려면, debconf-set-selections -C preseed.cfg 명령을사용할수있습니다.

B.4 미리설정파일의내용 (stretch 용)

이부록에서사용한설정은 <http://d-i.alioth.debian.org/manual/example-preseed.txt>에 들어있는예제 파일에서도구할수있습니다.

이예제 파일은인텔 x86 아키텍처용으로만들어졌습니다. 다른아키텍처에서설치하는경우, 예제에서어떤부분은(예를들어키보드선택이나부트로더설치) 해당아키텍처에서필요없을수도있고해당아키텍처에맞는 debconf 값으로바꿔야할수도있습니다.

다른데비안설치프로그램구성요소의작동방법에대한자세한내용은 6.3절에서찾을수있습니다.

B.4.1 지역화

일반적인설치에서는지역화에대한질문을먼저합니다. 그러므로이값은 initrd 나커널부팅파라미터방식으로만미리설정할수있습니다. 자동모드는 (B.2.3절) auto-install/enable=true 설정을포함합니다 (보통줄여서 auto 라고쓰는). 이렇게하면지역화질문을지연하므로, 어떤방법으로든미리설정을할수있습니다.

로캘은언어와국가를지정합니다. debian-installer 에서지원하는언어와국가라면뭐든지붙여서사용할수있습니다. 해당언어/국가조합이올바른로캘이아닌경우해당언어에대한로캘을자동으로하나선택합니다. 부팅파라미터로로캘을지정하려면, locale=ko_KR 과같이사용하십시오.

이방법은아주사용하기편하더라도, 그것은언어의모든가능한조합의 preseeding 을허용하지않습니다. en_NL 에국가와 locale³. 이렇게값이개별적으로 preseeded 할수있습니다. 언어및국가는부트매개변수로지정할수있습니다.

³ Preseeding locale 은 en_US.UTF-8 로예를들자면결과에대한설치시스템의기본 locale 로값을개별적으로 preseeded 해야합니다.

```
# 로캘만 미리 설정하면 언어, 국가, 로캘 값이 설정됩니다.
d-i debian-installer/locale string en_US

# 유연하게 각각의 값을 미리 설정할 수도 있습니다.
#d-i debian-installer/language string en
#d-i debian-installer/country string NL
#d-i debian-installer/locale string en_GB.UTF-8
# 선택적으로 추가될 로캘을 지정합니다.
#d-i localechooser/supported-locales multiselect en_US.UTF-8, nl_NL.UTF-8
```

키보드설정에서는 (라틴키보드가아닌아닌키맵의경우) 키맵과키맵을 (라틴키보드가아닌키맵과미국식키맵사이의) 전환하는토글키를선택합니다. 설치할때는기본키맵만사용할수있습니다. 고급키맵은설치한시스템에서만 dpkg-reconfigure keyboard-configuration 명령으로사용할수있습니다.

```
# 키보드 선택.
d-i keyboard-configuration/xkb-keymap select us
# d-i keyboard-configuration/toggle select No toggling
```

키보드설정을건너뛰려면 keymap 값을 skip-config 로하면됩니다. 그러면커널키맵을사용합니다.

B.4.2 네트워크설정

네트워크에서미리설정 파일을읽어들이는경우네트워크설정은당연히동작하지않습니다. 하지만 CD 나 USB 메모리로부터팅하는경우에네트워크설정을하면좋습니다. 미리설정 파일을네트워크에서읽어들이는경우, 커널부팅파라미터로네트워크설정을건너뛸수있습니다.

미리설정 파일을네트워크에서읽어들이기전에특정인터페이스에서네트워크부팅하려면, interface=eth1 처럼부팅파라미터를사용하십시오.

네트워크를통해미리설정을하는경우 (“preseed/url” 사용) 네트워크설정을미리설정하는게보통불가능하지만, 다음방법을이용해피해갈수있습니다. 예를들어네트워크인터페이스에고정주소를부여하는방법입니다. 다음명령이들어있는 “preseed/run” 스크립트를만들어서미리설정 파일을읽어들인후에네트워크설정을다시실행합니다:

```
kill-all-dhcp; netcfg
```

다음 debconf 변수가네트워크설정과관계가있습니다.

```
# 네트워크를 완전히 사용하지 않도록 설정합니다. 네트워크에 연결되지
# 않은 컴퓨터에 CD-ROM 설치를 하는 경우처럼 네트워크 관련 질문, 경고,
# 시간 초과 따위가 불필요한 경우에 좋습니다.
#d-i netcfg/enable boolean false

# 연결되어 있는 인터페이스를 선택합니다. 이러면 인터페이스가
# 여러 개 있는 경우 목록을 건너 뛩니다.
d-i netcfg/choose_interface select auto

# 특정 인터페이스를 선택할 경우:
#d-i netcfg/choose_interface select eth1

# 링크 검사 제한 시간을 다르게 설정할 때(기본값은 3초).
# 초 단위입니다
#d-i netcfg/link_wait_timeout string 10

# DHCP 서버가 느려서 응답을 기다리다가 시간이 초과되는 경우
# 다음 설정을 쓰면 됩니다.
#d-i netcfg/dhcp_timeout string 60
#d-i netcfg/dhcpv6_timeout string 60

# 네트워크 설정을 수동으로 하려면, 아래 줄의 주석을 지우고 그 아래에 있는
# 고정 네트워크 설정의 주석도 지우십시오.
#d-i netcfg/disable_autoconfig boolean true

# DHCP 서버가 있든 없든 모두 미리 설정 파일이 동작하게 만드려면, 아래
```

```
# 줄의 주석을 지우고 그 아래에 있는 고정 네트워크 설정의 주석도 지우십시오.
#d-i netcfg/dhcp_failed note
#d-i netcfg/dhcp_options select Configure network manually

# 고정 IP 네트워크 설정
#
# IPv4 예제
#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/get_nameservers string 192.168.1.1
#d-i netcfg/confirm_static boolean true
#
# IPv6 예제
#d-i netcfg/get_ipaddress string fc00::2
#d-i netcfg/get_netmask string ffff:ffff:ffff:ffff::
#d-i netcfg/get_gateway string fc00::1
#d-i netcfg/get_nameservers string fc00::1
#d-i netcfg/confirm_static boolean true

# DHCP에서 지정한 호스트 이름과 도메인 이름이 여기에서 설정한 것보다
# 우선합니다. 하지만 DHCP에서 호스트 이름과 도메인 이름이 넘어오는
# 경우라고 해도, 여기서 값을 설정해야 질문을 하지 않게 됩니다.
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain

# DHCP 서버에서 넘긴 호스트 이름이나, IP에 대한 리버스 DNS와 무관하게
# 호스트 이름을 강제로 설정하려면 다음 줄의 주석을 지우고 수정하십시오.
#d-i netcfg/hostname string somehost

# 성가신 WEP 키 대화 상자를 사용하지 않습니다.
d-i netcfg/wireless_wep string
# 일부 DHCP 서버는 호스트이름을 암호처럼 사용합니다.
#d-i netcfg/dhcp_hostname string radish

# 네트워크 등의 하드웨어에 자유롭게 배포되지 않는 펌웨어가 필요한 경우, 물어보지
# 않고 그 펌웨어를 읽어들이도록 설정할 수 있습니다. 아니면 false로 하면
# 물어보지도 않게 할 수 있습니다.
#d-i hw-detect/load_firmware boolean true
```

netcfg/get_netmask 를미리지정하지않으면 netcfg 는자동으로네트마스크를지정합니다. 자동설치에서는이변수를 seen 으로 표시해야합니다. 마찬가지로 netcfg/get_gateway 를지정하지않으면 netcfg 는적당한주소로게이트웨이를설정합니다. 특별한경우로, netcfg/get_gateway 를 “none” 으로설정하면게이트웨이를사용하지않습니다.

B.4.3 네트워크콘솔

```
# SSH를 통해 원격 설치를 하면서 network-console 컴포넌트를 사용할 경우 다음
# 설정을 사용합니다. 이후의 모든 설치를 수동으로 하는 경우에만 이렇게 합니다.
#d-i anna/choose_modules string network-console
#d-i network-console/authorized_keys_url string http://10.0.0.1/openssh-key
#d-i network-console/password password r00tme
#d-i network-console/password-again password r00tme
```

B.4.4 미리사이트설정

사용하는설치방법에따라서, 미리사이트를이용해설치프로그램의추가컴포넌트, 베이스시스템을다운로드할수있습니다. 또설치를끝낸시스템에서 /etc/apt/sources.list 파일을설정하는데미리사이트를이용할수있습니다.

mirror/suite 파라미터로설치할시스템의세트를결정합니다.

mirror/udeb/suite 파라미터는설치프로그램의추가컴포넌트의세트를설정합니다. 실제구성요소를네트워크로다운로드하는데도움이됩니다. 또한설치에사용할설치방법을위한initrd를생성하려면이쌍이일치해야합니다. 일반적으로설치프로그램은자동으로올바른값을설정하므로설정할필요가없습니다.

```
# ftp의 경우, mirror/country 문자열은 설정할 필요가 없습니다.
#d-i mirror/protocol string ftp
d-i mirror/country string manual
d-i mirror/http/hostname string ftp.kr.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

# 설치할 세트
#d-i mirror/suite string testing
# 설치 프로그램을 읽어들이 세트(옵션).
#d-i mirror/udeb/suite string testing
```

B.4.5 계정설정

루트계정의암호와맨처음만들일반사용자의이름및암호도미리설정할수있습니다. 암호의경우일반텍스트값을그대로쓸수도있고 crypt(3) 해시값을쓸수도있습니다.

주의



미리설정암호는안전하지않습니다. 미리설정파일을읽을수있는사람은암호도알수있기때문입니다. 해시형태로암호를저장하면 (다수의암호를입력하는데좋은 DES 나 MD5 와같은취약한알고리즘을사용하지않는) 보안상안전하다고여겨집니다. 추천하는암호해시알고리즘은 SHA-256 및 SHA512 입니다.

```
# 루트 계정을 만들지 않고 넘어갑니다. (일반 유저는 sudo를 사용할
# 수 있습니다.).
#d-i passwd/root-login boolean false
# 아니면 일반 사용자를 만들지 않고 넘어갈 수 있습니다.
#d-i passwd/make-user boolean false

# 루트 암호, 암호 원문 텍스트를 직접 쓸 수도 있고
#d-i passwd/root-password password r00tme
#d-i passwd/root-password-again password r00tme
# 아니면 crypt(3) 해시로 암호화된 암호를 쓸 수도 있습니다.
#d-i passwd/root-password-crypted password [crypt(3) hash]

# 아니면 일반 사용자 계정을 하나 만듭니다.
#d-i passwd/user-fullname string Debian User
#d-i passwd/username string debian
# 일반 사용자 암호, 암호 원문 텍스트를 직접 쓸 수도 있고
#d-i passwd/user-password password insecure
#d-i passwd/user-password-again password insecure
# 아니면 crypt(3) 해시로 암호화된 암호를 쓸 수도 있습니다.
#d-i passwd/user-password-crypted password [crypt(3) hash]
# 기본값이 아닌 지정한 UID 값으로 첫번째 사용자를 만듭니다.
#d-i passwd/user-uid string 1010

# 사용자 계정은 표준으로 정해진 그룹에 들어갑니다. 강제로
# 그룹을 지정하려면 다음과 같이 합니다.
#d-i passwd/user-default-groups string audio cdrom video
```

passwd/root-password-crypted 및 passwd/user-password-crypted 변수의값으로 “!” 를 써서미리설정할수있습니다. 이경우해당하는계정을사용할수없습니다. 루트계정은이렇게하는게편리할수도있습니다. 물론루트계정을이렇게하면시스템관리를할수있는다른방법이있거나루트로그인을할수있는다른방법이있어야합니다. (예를들어 SSH 키인증을쓰거나 sudo 를사용하는방법)

암호에대한 SHA-512 기반 crypt(3) 해시를만드려면다음명령을사용할수있습니다(whois 패키지에들어있습니다):

```
mkpasswd -m sha-512
```

B.4.6 시계및시간대설정

```
# 하드웨어 시계를 UTC로 할 지 여부를 결정합니다
d-i clock-setup/utc boolean true

# $TZ로 설정 가능한 값은 뭐든지 쓸 수 있습니다. 설정 가능한
# 값은 /usr/share/zoneinfo/ 아래의 내용을 참고하십시오.
d-i time/zone string US/Eastern

# 설치하면서 시계를 맞출 때 NTP를 사용할 지 여부를 설정합니다.
d-i clock-setup/ntp boolean true
# 사용할 NTP 서버. 보통 기본값을 사용하는 게 좋습니다.
#d-i clock-setup/ntp-server string ntp.example.com
```

B.4.7 파티션하기

하드디스크파티션에 preseed 를 사용하는것은 partman-auto 에서지원하는기능에 한정되고있습니다. 파티션은디스크의빈영역과전체디스크중하나를선택해야합니다. 디스크의구성은미리정의된요리법, 레시피파일을사용하여사용자정의레시피미리설정파일에쓰레시피에서선택할수있습니다.

RAID, LVM 및암호화를사용하여고급파티션설정의 Preseeding 는지원되지만, 가능한한완전한유연성이 non-preseeded 설치하는동안파티션을하지않을경우입니다.

아래의예제는사용법에대한기본적인정보를제공합니다. 자세한내용은 debian-installer 패키지에포함되어있는파일 partman-auto-recipe.txt 및 partman-auto-raid-recipe.txt 를참조하십시오. 두파일은 [debian-installer 소스저장소](#)에도들어있습니다. 지원되는기능은릴리스마다달라질수있습니다.

주의



디스크의 ID 는디스크의드라이버를읽어들이는순서에따라다릅니다. 시스템에디스크가여러개있는경우, 미리설정을이용하기전에올바른디스크를선택하도록하십시오.

B.4.7.1 파티션예제

```
# If the system has free space you can choose to only partition that space.
# This is only honoured if partman-auto/method (below) is not set.
#d-i partman-auto/init_automatically_partition select biggest_free

# Alternatively, you may specify a disk to partition. If the system has only
# one disk the installer will default to using that, but otherwise the device
# name must be given in traditional, non-devfs format (so e.g. /dev/sda
# and not e.g. /dev/discs/disc0/disc).
# For example, to use the first SCSI/SATA hard disk:
#d-i partman-auto/disk string /dev/sda
# In addition, you'll need to specify the method to use.
# The presently available methods are:
# - regular: use the usual partition types for your architecture
# - lvm:      use LVM to partition the disk
# - crypto:  use LVM within an encrypted partition
d-i partman-auto/method string lvm
```

```

# If one of the disks that are going to be automatically partitioned
# contains an old LVM configuration, the user will normally receive a
# warning. This can be preseeded away...
d-i partman-lvm/device_remove_lvm boolean true
# The same applies to pre-existing software RAID array:
d-i partman-md/device_remove_md boolean true
# And the same goes for the confirmation to write the lvm partitions.
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true

# You can choose one of the three predefined partitioning recipes:
# - atomic: all files in one partition
# - home:   separate /home partition
# - multi:  separate /home, /var, and /tmp partitions
d-i partman-auto/choose_recipe select atomic

# Or provide a recipe of your own...
# If you have a way to get a recipe file into the d-i environment, you can
# just point at it.
#d-i partman-auto/expert_recipe_file string /hd-media/recipe

# If not, you can put an entire recipe into the preconfiguration file in one
# (logical) line. This example creates a small /boot partition, suitable
# swap, and uses the rest of the space for the root partition:
#d-i partman-auto/expert_recipe string \
#     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-swaps \
#                     method{ swap } format{ } \
#                     .

# The full recipe format is documented in the file partman-auto-recipe.txt
# included in the 'debian-installer' package or available from D-I source
# repository. This also documents how to specify settings such as file
# system labels, volume group names and which physical devices to include
# in a volume group.

# This makes partman automatically partition without confirmation, provided
# that you told it what to do using one of the methods above.
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# When disk encryption is enabled, skip wiping the partitions beforehand.
#d-i partman-auto-crypto/erase_disks boolean false

```

B.4.7.2 RAID 를사용해파티션하기

소프트웨어 RAID 파티션을설정하거나 preseed 를사용하여수있습니다. 지원은 RAID 0, 1, 5, 6, 10, 비상용어레이및에비장치를지정합니다.

주의



이런방식의자동파티션은잘못되기쉽습니다. 또이기능은 debian-installer 개발자가별로테스트하지않는기능입니다. 여러가지방식을올바르게 (규칙에 맞으면서충돌하지않게) 설정하는책임은사용자에게있습니다. 문제가발생하면 /var/log/syslog 파일을확인하십시오.

```
# method 값을 "raid"로 설정합니다.
#d-i partman-auto/method string raid
# 파티션할 디스크를 지정합니다. 디스크는 모두 같은 레이아웃이어야
#하므로, 크기가 같을 경우에만 다음 설정이 동작합니다.
#d-i partman-auto/disk string /dev/sda /dev/sdb

# 그리고 사용할 물리 파티션을 지정합니다.
#d-i partman-auto/expert_recipe string \
#     multiraid ::                                \
#         1000 5000 4000 raid                      \
#         $primary{ } method{ raid }              \
#         .                                          \
#         64 512 300% raid                          \
#         method{ raid }                          \
#         .                                          \
#         500 10000 1000000000 raid                \
#         method{ raid }                          \
#         .

# 마지막으로 이전에 지정한 파티션을 어떻게 RAID에서 사용할지
# 지정합니다. 논리 파티션에 올바른 파티션 번호를 사용하도록 하십시오.
# RAID 레벨 0, 1, 5, 6, 10을 지원합니다. 장치는 "#"문자로 구분합니다.
# 파라미터는 다음과 같습니다:
# <raidtype> <devcount> <sparecount> <fstype> <mountpoint> \
#     <devices> <sparedevices>

#d-i partman-auto-raid/recipe string \
#     1 2 0 ext3 /                                  \
#         /dev/sda1#/dev/sdb1                      \
#         .                                          \
#     1 2 0 swap -                                  \
#         /dev/sda5#/dev/sdb5                      \
#         .                                          \
#     0 2 0 ext3 /home                              \
#         /dev/sda6#/dev/sdb6                      \
#         .

# 더 자세한 정보는 'debian-installer' 패키지나 D-I 소스 저장소의
# partman-auto-raid-recipe.txt 파일에 있습니다.

# 다음 설정을 하면 partman에서 확인 질문 없이 파티션을 자동으로 진행합니다.
d-i partman-md/confirm boolean true
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
```

B.4.7.3 파티션마운트방법조정하기

파일시스템은장치이름이바뀌더라도 UUID(universally unique identifier) 를키로해서마운트합니다. UUID 는길어서알아보기어려우므로, 전통적인장치이름에따라마운트할수도있고, 레이블을이용해마운트할수도있습니다. 레이블에따라마운트할경우, 레이블이없는파일시스템은 UUID 를사용해마운트합니다.

LVM 논리볼륨처럼 고정된 이름의 장치는 UUID 가 아니라 계속 통적인 이름을 사용합니다.

주의



전통적인 장치 이름은 부팅할 때 장치를 발견한 순서에 따라 달라질 수 있습니다. 그래서 잘못된 파일 시스템을 마운트하는 실수를 저지룰 수 있습니다. 마찬가지로 레이블도 새로운 디스크나 USB 드라이브 따위를 연결했을 때 레이블이 충돌할 수 있고 그 경우 시스템이 어떻게 동작할지 확신할 수 없습니다.

```
# 기본값은 UUID로 마운트하는 것이지만, 전통적인 장치 이름을 사용하려면
# "traditional"을 사용할 수 있고, 파일 시스템 레이블을 사용하려면
# "label"을 사용합니다. 시도가 실패하면 UUID를 사용합니다.
#d-i partman/mount_style select uuid
```

B.4.8 기본시스템설치

이상태에서는 미리 설정할 수 있는 부분이 별로 많지 않습니다. 유일하게 신경 쓸 부분은 커널 설치에 관한 질문입니다.

```
# APT에서 권장 패키지를 설치하지 않도록 설정합니다. 이 옵션을 사용하면
# 불안정한 시스템이 될 수도 있으므로, 아주 경험 많은 사용자만 사용해야
# 합니다.
#d-i base-installer/install-recommends boolean false

# 설치할 커널 이미지 패키지(또는 메타 패키지). 커널을 설치하지 않으면 "none"을
# 사용합니다.
#d-i base-installer/kernel/image string linux-image-686
```

B.4.9 APT 설정

/etc/apt/sources.list 의 설정과 기본 설정 옵션은 설치 방법과 그 이전의 질문에 어떻게 답했냐에 따라 완전히 자동화합니다. 추가적으로 다른 저장소를 지정할 수 있습니다.

```
# non-free와 contrib 소프트웨어를 설치할 수 있습니다.
#d-i apt-setup/non-free boolean true
#d-i apt-setup/contrib boolean true
# 네트워크 미러를 사용하지 않으려면 다음의 주석을 지우십시오.
#d-i apt-setup/use_mirror boolean false
# 어떤 업데이트 서비스를 사용할 지 선택합니다. 사용할 미러를 지정합니다.
# 아래의 값은 보통 사용하는 기본값입니다.
#d-i apt-setup/services-select multiselect security, updates
#d-i apt-setup/security_host string security.debian.org

# 추가 저장소, local[0-9] 사용가능
#d-i apt-setup/local0/repository string \
# http://local.server/debian stable main
#d-i apt-setup/local0/comment string local server
# deb-src 줄을 만듭니다
#d-i apt-setup/local0/source boolean true
# 가까운 저장소의 공개키 URL. 키를 제공하지 않으면 APT가 인증되지 않은
# 저장소에 대해서 오류 메시지를 출력하고 관련 sources.list 줄을 주석
# 처리합니다.
#d-i apt-setup/local0/key string http://local.server/key

# 기본값으로 저장소는 알려진 GPG 키로 인증할 수 있어야 합니다. 아래와 같이
# 설정하면 인증을 검사하지 않습니다. 경고: 안전하지 않으므로, 추천하지 않는
# 방법입니다.
#d-i debian-installer/allow_unauthenticated boolean true
```

```
# i386에 대한 multiarch 설정을 추가하려면 다음의 주석을 지우십시오
#d-i apt-setup/multiarch string i386
```

B.4.10 패키지선택

태스크는원하는대로설치할수있습니다. 현재이문서를쓰는시점에사용할수있는태스크는다음과같습니
다:

- standard (표준도구)
- desktop (그래픽데스크톱)
- gnome-desktop (그놈데스크톱)
- xfce-desktop (XFCE 데스크톱)
- kde-desktop (KDE 플라즈마데스크톱)
- cinnamon-desktop (시나몬데스크톱)
- mate-desktop (MATE 데스크톱)
- lxde-desktop (LXDE 데스크톱)
- web-server (웹서버)
- print-server (인쇄서버)
- ssh-server (SSH 서버)

태스크를설치하지않을수도있고, 다른방법으로패키지를설치할수있습니다. 표준시스템 태스크는항
상포함하시길권장합니다.

태스크로설치한패키지외에패키지를더설치하려면, `pkgselect/include` 파라미터를사용하면됩니다.
이파라미터의값은선택공백으로구분할수있으므로, 커널명령행에서도쉽게사용할수있습니다.

```
#tasksel tasksel/first multiselect standard, web-server, kde-desktop

# 추가로 설치할 패키지
#d-i pkgselect/include string openssh-server build-essential
# debootstrap 다음에 패키지를 업그레이드할 지 여부
# 사용 가능한 값은: none, safe-upgrade, full-upgrade
#d-i pkgselect/upgrade select none

# 어떤 소프트웨어를 설치했는지 설치 프로그램에서 보고서를 보낼 수
# 있습니다. 보고하지 않는 게 기본값이지만, 보고서를 보내면 데비안
# 프로젝트에서 어떤 소프트웨어를 더 많이 사용하고 CD에 포함하는 게
# 좋을 지 결정하는 데 도움이 됩니다.
#popularity-contest popularity-contest/participate boolean false
```

B.4.11 설치마치기

```
# 시리얼 콘솔에서 설치하면, 일반 가상 콘솔은(VT1-VT6) /etc/inittab에서
# 막습니다. 다음의 주석을 지우면 가상 콘솔을 막지 않습니다.
#d-i finish-install/keep-consoles boolean true

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

# 다음과 같이 하면 다시 시작할 때 CD를 빼지 않습니다.
# 경우에 따라서는 CD를 빼지 않는 게 좋을 수 있습니다.
#d-i cdrom-detect/eject boolean false
```

```
# 다음과 같이 하면 설치가 끝났을 때, 설치한 시스템으로
# 다시 시작하지 않고 셧다운합니다.
#d-i debian-installer/exit/halt boolean true
# 다음과 같이 하면 컴퓨터의 전원도 끕니다.
#d-i debian-installer/exit/poweroff boolean true
```

B.4.12 기타패키지미리설정

```
# 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
```

B.5 고급옵션

B.5.1 설치할때임의의명령어실행하기

미리설정도구의매우강력하고도유연한옵션은, 설치특정시점에명령어와스크립트를실행하는기능입니다.

대상시스템의파일시스템을마운트했으면, /target 아래에그파일시스템이있습니다. 설치 CD 를사용하고마운트하면, 그내용은 /cdrom 아래에있습니다.

```
# 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 셸 명령어라도 실행할 수 있는
# 방법이 만들어져 있습니다. 위험하지만 이 방법은 매우 유용하므로,
# 다음과 같이 설치 프로그램 내에서 셸 명령어를 실행할 수 있습니다.

# 다음 첫 번째 명령어는 미리 설정 파일을 읽어들이는 직후에 가능한 빨리
# 실행합니다.
#d-i preseed/early_command string anna-install some-udeb
# 다음 명령은 파티션 프로그램이 시작하기 직전에 실행합니다. 여기에는
# 디스크의 상태에 따라 다르게 동적으로 파티션하는 미리 설정을 사용하면
# 편리합니다. (preseed/early_command 명령이 실행하는 시점에서는
# 디스크의 상태를 알 수 없습니다.)
#d-i partman/early_command \
# string debconf-set partman-auto/disk "${list-devices disk | head -n1}"
# 다음 명령은 설치를 끝내기 직전에 실행합니다. 그러나 /target 디렉터리는 아직
# 사용할 수 있는 시점입니다. /target 디렉터리로 chroot해서 직접 사용할
# 수 있고 패키지를 쉽게 설치하려면 apt-install과 in-target명령을 사용할
# 수 있습니다.
#d-i preseed/late_command string apt-install zsh; in-target chsh -s /bin/zsh
```

B.5.2 미리설정을이용해기본값바꾸기

미리설정으로질문에대한기본값을바꾸면서, 그래도그질문을받도록만들수있습니다. 이렇게하려면해당서식에대한값을설정한다다음에 seen 플래그를 “false” 로놓으면됩니다.

```
d-i foo/bar string value
d-i foo/bar seen false
```

부팅파라미터로 preseed/interactive=true 라고설정하면 모든질문에대해서같은효과를거둘수있습니다. 이기능은미리설정파일을테스트하거나디버깅하는데도좋습니다.

주의할점이, “d-i” 소유자는설치프로그램에서사용하는변수에만사용해야합니다. 대상시스템에설치한패키지에관련된변수에대해서는그패키지의이름을사용해야합니다. B.2.2절부분의각주를보십시오.

부팅파라미터를이용해미리설정을하는경우, “?” 연산자를사용해서해당질문을물어보도록만들수 있습니다. 예를들어어쩌구/저쩌구?= 값와같이 (아니면소유자: 어쩌구/저쩌구?= 값) 합니다.

디버깅정보를자세히보려면 DEBCONF_DEBUG=5 부팅파라미터를사용하십시오. 그러면 debconf 에 서각변수의현재설정및각패키지의설치스크립트의진행상태에대해더자세히표시합니다.

B.5.3 미리설정파일을분리해서사용하기

미리설정파일에서다른미리설정파일을포함할수도있습니다. 파일에들어있는설정은앞에서읽어들인파 일에들어있는설정을덮어씹습니다. 이방법을이용해서, 예를들어파일하나에일반적인네트워크설정을집어 넣고세세한설정을다른파일에집어넣는식으로활용이가능합니다.

```
# 여러 개 파일을 공백으로 구분해서 쓸 수도 있습니다. 그러면 모든
# 파일을 읽어들이니다. 물론 포함한 파일은 그 안에 preseed/include가
# 들어 있을 수 있습니다. 주의할 점으로, 파일 이름이 상대 경로인 경우 그
# 파일이 포함되어 들어가는 파일이 있는 같은 디렉터리에서 찾게 됩니다.
#d-i preseed/include string x.cfg

# 설치 프로그램은 미리 설정 파일을 사용하기 전에 그 파일의 체크섬을
# 검사합니다. 현재는 md5sum만 지원하고, md5sum을 포함하는 파일과 같은
# 순서로 쓰십시오.
#d-i preseed/include/checksum string 5da499872becccfeda2c4872f9171c3d

# 좀 더 유연하게 하려면, 다음과 같이 하면 미리 설정 파일의 이름을
# 출력하는 셸 명령어를 출력하고, 그 파일을 포함합니다.
#d-i preseed/include_command \
#     string if [ ``hostname`` = bob ]; then echo bob.cfg; fi

# 이 중어거 가장 유연한 것으로, 프로그램을 다운로드하고 이를 실행할 수
# 있습니다. 이 프로그램은 debconf 데이터베이스를 조작하려면
# debconf-set과 같은 명령을 사용할 수 있습니다. 여러 개의 스크립트를
# 공백으로 구분해서 쓸 수도 있습니다. 파일 이름이 상대경로로 되어 있으면
# 프로그램을 실행하는 미리 설정 파일이 있는 디렉터리에서 파일을 찾습니다.
#d-i preseed/run string foo.sh
```

initrd 혹은파일을이용한미리설정단계에서, 파일안에다시 preseed/url 을설정해서네트워크미리설정 을검쳐넣을수도있습니다. 이렇게하면네트워크가연결되었을때미리설정을읽어들이게됩니다. 이와같이 하는경우에는주의해야합니다. 미리설정을실행하는두개의별도의단계가있기때문입니다. 예를들어서 preseed/early 명령을한번더실행할수있고, 두번째가네트워크가연결된다음에실행될수있습니다.

Appendix C

데비안에서 파티션 나누기

C.1 데비안 파티션 및 크기 정하기

최소한의 구성으로 GNU/Linux 는 자신을 위해 하나 이상의 파티션을 필요로 합니다. 전체 운영체제, 응용 프로그램, 개인 파일은 하나의 파티션에 저장됩니다. 많은 사람이 이와 swap 파티션도 필요하다고 생각하는 것 같습니다만, 이것은 엄밀하게 옳바르지 않습니다. “Swap” 은 운영체제가 가진 메모리의 임시 공간으로, 이것을 이용하면 시스템은 디스크 장치를 “가상 메모리” 로 사용할 수 있게 됩니다. swap 을 별도의 파티션에 두면, Linux 에서 이용이 훨씬 더 효율적입니다. Linux 일반적인 파일을 swap 으로 사용할 수 있지만 이것은 권장하지 않습니다.

하지만 대부분의 사람이 최소한 필요한 것보다 많은 파티션을 GNU/Linux 에 할당합니다. 파일 시스템을 몇 개의 작은 파티션에 나누는 이유는 2 가지가 있습니다. 첫 번째는 안전성입니다. 만약 우연히 무언가가 파일 시스템을 파괴해도 일반적으로 그 영향을 받는 것은 하나의 파티션만입니다. 따라서 시스템의 일부분만 (잘 보관해 두고 있던 백업에서) 복구하면 됩니다. 이런 이유에서 “루트 파티션” 은 따로 하는 것을 고려하십시오. 여기에는 시스템의 가장 기본적인 구성 부분이 들어 있고, 만약 다른 파티션에 손상이 생기더라도, Linux 를 시작하여 시스템을 바로 잡을 수 있습니다. 시스템을 처음부터 다시 설치해야 하는 듯한 문제를 막을 수 있습니다.

두 번째 이유는 보통 업무용 컴퓨터에서 더 중요하지만, 컴퓨터를 어떻게 사용하느냐에 따라 다릅니다. 예를 들어 대량 스팸 메일을 받는 메일 서버에서는 금방 파티션 하나가 꽂칩니다. 그 메일 서버에서 /var/mail 을 별도의 파티션에 만들었다면, 스팸 메일을 받더라도 시스템의 다른 부분은 계속 동작합니다.

파티션 여러 개의 사용 환경은 유일한 단점은, 파티션에 필요한 크기를 미리 알기 힘들다는 점입니다. 파티션을 너무 작게 만들면 시스템을 새로 설치하거나 그 파티션에 있는 파일을 자주 다른 파티션으로 옮겨야 합니다. 반면 파티션을 너무 크게 만들면 다른 곳에서 쓸 수 있는 용량을 낭비하는 셈이 됩니다. 디스크 가격이 저렴해졌지만 낭비할 필요는 없습니다.

C.2 디렉터리 구조

디렉터리와 파일 이름에 대해 데비안 GNU/리눅스는 **Filesystem Hierarchy Standard**에 따릅니다. 이 표준을 준수함으로써 사용자들과 유저 프로그램은 파일과 디렉터리의 위치를 예상하기 쉽습니다. 루트 디렉터리는 슬래쉬 / 로 표시됩니다. 루트 수준에는 데비안 시스템은 반드시 다음과 같은 디렉터리가 포함됩니다:

디렉터리	내용
bin	핵심 명령어 바이너리
boot	부트로더에서 필요한 고정 파일
dev	장치 파일
etc	이 호스트의 시스템 설정
home	사용자 홈 디렉터리
lib	핵심 공유 라이브러리 및 커널 모듈
media	이동식 미디어의 마운트 위치가 들어 있습니다
mnt	파일 시스템을 임시로 마운트하는 마운트 위치
proc	시스템 정보를 저장하는 가상 디렉터리
root	루트 사용자의 홈 디렉터리
run	실행할 때 바뀌는 데이터
sbin	핵심 시스템 바이너리
sys	시스템 정보를 저장하는 가상 디렉터리

디렉터리	내용
tmp	임시 파일
usr	이차디렉터리 구조
var	자주 바뀌는 데이터
srv	시스템 서비스의 데이터
opt	별도의 응용 소프트웨어 패키지

아래의 목록은 디렉터리와 파티션에 대해 고려할 사항입니다. 실제 시스템 사용량은 시스템의 설정과 사용 용도에 따라 달라집니다. 아래 권장 사항은 파티션 할 때 참고만 하십시오.

- /etc, /bin, /sbin, /lib, /dev 는 반드시 루트 파티션 (/) 에 들어 있어야 합니다. 그렇지 않으면 부팅에 문제가 발생합니다. 루트 파티션은 일반적으로 250-350MB 정도가 필요합니다.
- /usr: 모든 유저 프로그램 (/usr/bin) 과 라이브러리 (/usr/lib) 와 시스템 문서 (/usr/share/doc) 등이 들어 있습니다. 보통 파일 시스템에서 가장 하드 디스크 공간을 많이 차지하는 부분입니다. 최소한 500MB 를 할당하십시오. 시스템에 설치할 패키지의 수와 종류에 따라 더 늘려야 할 수도 있습니다. 보통 워크스테이션이나 서버로 설치하려면 4-6GB 정도가 필요합니다.
- 이제 /usr 파티션은 루트 파티션 (/) 과 같이 두기를 권장합니다. 그러지 않으면 부팅할 때 문제가 생길 수도 있습니다. 그래서 루트 파티션은 /usr 를 포함해 최소한 600-750MB 의 디스크 공간을 확보해야 하고, 워크스테이션과 서버에서 5-6GB 정도를 확보해야 합니다.
- /var: 뉴스 기사, 전자메일, 웹페이지, 데이터베이스, 패키지 시스템의 캐시 등 자주 변하는 정보가 주로 저장됩니다. 이 디렉터리의 크기는 시스템의 이용 방법에 크게 좌우되지만 대부분의 시스템에서는 패키지 관리 도구의 사용 분리가 가장 큰 영향을 가지게 될 것입니다. 데비안에서 제공하는 모든 것을 한꺼번에 전체 설치하는 경우에도 /var 에 2 또는 3GB 정도를 할당하시면 충분합니다. 한번에 모두 설치하지 않고 부분부분을 서서히 (예를 들면, 우선 서비스와 유틸리티, 다음에 콘솔용의 것, 다음에 X 용의 것... 과 같이) 설치하는 경우, 300-500MB 의 여유 공간 있으면 좋습니다. 하드 디스크의 빈 용량이 귀중하고, 대대적인 업데이트에 정이 없다면 30 또는 40MB 정도에서도 어떻게든 해나갈 수 있습니다.
- /tmp: 프로그램이 만든 임시 데이터를 저장합니다. 40-100MB 정도면 충분합니다. 압축 유틸리티, CD/DVD 굽기 유틸리티, 멀티미디어 프로그램의 경우 이미지 파일을 /tmp 에 임시로 저장하기도 합니다. 이러한 프로그램을 사용한다면 /tmp 의 크기를 적절히 조절하십시오.
- /home: 모든 사용자는 이 디렉터리의 서브 디렉터리에 개인 데이터를 저장합니다. 이 디렉터리의 크기는 이 시스템을 사용하는 사용자가 몇 명이 고 디렉터리에 어떤 파일을 넣을 지에 따라 달라집니다. 예정된 사용량에 따라 다르지만, 각 사용자에게 100MB 씩 할당하고, 필요에 따라 이 값을 조정하십시오. 홈 디렉터리에 다수의 멀티미디어 파일 (사진, MP3, 동영상) 을 저장할 예정이면 더 많은 용량을 잡아 주십시오.

C.3 권장하는 파티션 구조

신규 사용자와 데비안 시스템을 개인으로 사용하는 사람, 가정에서 사용하는 시스템 등 기타 혼자서 사용하는 것과 같은 시스템에는 / 파티션 1 개 (그리고 swap) 로 끝내는 것이 아마 가장 간단하고 솔직한 방식입니다. 그러나 파티션이 약 6GB 보다 큰 경우 파티션 유형에 ext3 를 선택하십시오. ext2 파티션은 주기적으로 파일 시스템 무결성 검사를 필요로 합니다. 그리고 이것은 파티션이 클 경우 부팅 지연의 원인이 됩니다.

여러 명이 사용하는 시스템이거나 하드 디스크의 용량이 큰 시스템에서는 /var, /tmp, /home 각각을 / 파티션과는 별도의 파티션에 두는 것이 좋습니다.

데비안 배포판에 포함되지 않은 프로그램을 많이 설치할 계획이라면 /usr/local 파티션이 필요할지도 모릅니다. 또한 메일 서버로 사용한다면, /var/mail 를 다른 파티션으로 할 필요가 있을지도 모릅니다. /tmp 에 별도의 파티션 (예: 20-50MB 정도) 을 할당하는 것도, 많은 경우 좋은 생각입니다. 많은 사용자가 계정을 안고 서버를 설치한다면, 독립적인 큰 /home 파티션을 준비하는 일도 대체로 좋은 생각입니다. 이렇게 사용 방법에 따라 파티션의 구성은 시스템에 따라 다양합니다.

매우 복잡한 시스템의 경우, **멀티디스크 HOWTO** 를 참고하십시오. ISP 나 서버 관리자가 관심 있어 할 만한 심도 있는 내용을 다루고 있습니다.

스왑 파티션의 크기에 대해서는 여러 가지 생각이 다릅니다. 한 가지 방법은 시스템 메모리 만큼 스왑 공간을 잡는 것입니다. 또 16MB 보다 작으면 안 됩니다. 물론 이런 규칙에도 예외는 있습니다. 256MB 메모리가 들어 있는 컴퓨터에서 10000 개의 수식을 동시에 풀려고 한다면, 스왑을 1GB (혹은 그 이상) 필요합니다.

일부 32-bit 아키텍처 (m68k 및 PowerPC) 에서 스왑 파티션의 최대 크기는 2GB 입니다. 이것은 대부분의 경우에 대해 충분한 크기입니다. 하지만 만약 더 이상의 크기의 스왑 공간이 필요하다면, 다른 디스크 (또는 “스

핀들”)에 스왑영역을 분산하고, 또 가능하면 SCSI 및 IDE 다른 채널에 스왑영역을 분산하도록 시도하십시오. 이렇게 하면 커널은 여러 스왑영역을 균형있게 사용하려고 하기 때문에 성능이 향상됩니다.

예를 들어 좀 오래된 컴퓨터에 램이 32 MB 있고 /dev/sda 에 1.7 GB 짜리 IDE 하드디스크가 있습니다. 500MB 짜리 /dev/sda1 에 다른 운영체제가 설치되어 있으며, /dev/sda3 는 32MB 의 스왑공간으로 사용됩니다. /dev/sda2 에는 1.2GB 용량에 리눅스를 사용합니다.

시스템설치가 끝났을 때 얼마나 공간을 차지할지 알고 싶으면, [D.2](#) 절 부분을 참고하십시오.

C.4 리눅스의 장치 이름

리눅스에서 디스크와 파티션을 부르는 이름이 다른 운영체제와 다르기도 합니다. 파티션을 만들고 파티션 할 때 이 리눅스 이름을 알고 있어야 합니다. 기본적으로는 다음 규칙을 따릅니다:

- 첫 번째 플로피 디스크 드라이브는 /dev/fd0 이라고 합니다.
- 두 번째 플로피 디스크 드라이브는 /dev/fd1 이라고 합니다.
- 첫 번째 발견한 하드 디스크의 이름은 /dev/sda 입니다.
- 두 번째 발견한 하드 디스크의 이름은 /dev/sdb 이고, 그 이후는 마찬가지로입니다.
- 첫 번째 SCSI CD-ROM 은 /dev/scd0 이라고 하고, /dev/sr0 이라고도 합니다.

드라이브의 파티션 이름은 디스크 이름 뒤에 숫자를 붙입니다. sda1 와 sda2 는 각각 첫 번째 SCSI 디스크의 첫 번째와 두 번째 파티션을 말합니다.

실제 예를 들어 보면 다음과 같습니다. SCSI 디스크가 2 개 있어서, 하나는 SCSI 주소 2 에 연결되어 있고 다른 하나는 4 에 연결되어 있습니다. 첫 번째 (2 번 주소에 연결된) 디스크가 sda 이고, 두 번째 (4 번 주소에 연결된) 디스크가 sdb 입니다. sda 에 파티션이 3 개이면, 그 파티션의 이름은 sda1, sda2, sda3 입니다. sdb 디스크와 그 파티션도 같은 방식입니다.

SCSI 호스트 버서 어댑터 (컨트롤러) 가 2 개 있으면 어느 드라이브가 첫 번째가 될지 알기 어려울 수도 있습니다. 이 경우엔 부팅 할 때 메시지를 잘 보고, 드라이브의 모델 과 용량으로 파악하는 게 최선의 방법입니다.

C.5 데비안의 파티션 프로그램

여러 가지 종류의 파티션 도구가 내장된 다양한 형식의 하드 디스크나 시스템에서 작동하도록 데비안 개발자가 준비해 놓았습니다. 아래에 시스템에서 사용할 수 있는 프로그램의 목록을 나타냅니다.

partman 데비안 권장 파티션 도구입니다. 이 만능 프로그램은 파티션 크기를 변경하거나 파일 시스템을 만들거나 마운트 지점을 지정하거나 할 수 있습니다.

fdisk 처음부터 있던 리눅스 파티션 프로그램. 전문가용.

FreeBSD 용 파티션이 디스크에 있으면 주의해야 합니다. 설치용 커널은 이 형식의 파티션을 지원하지 않지만, fdisk 프로그램에서 이 파티션을 화면에 표시하는 형식입니다. [리눅스 + FreeBSD 하우투](#) 를 참고하십시오.

fdisk 간단하고 널리 사용하는 전체 화면 파티션 프로그램.

fdisk 는 FreeBSD 파티션을 전혀 인식하지 못하기 때문에 (다시 말하지만) 장치의 이름이 다를 수도 있으니 주의하십시오.

디스크 파티션하기 메뉴를 선택하면 위 프로그램 중 하나를 실행합니다. VT2 에서 명령행 을 이용해 다른 파티션 도구를 사용할 수도 있지만, 이 방법은 추천하지 않습니다.

Appendix D

여러가지내용

D.1 리눅스장치

리눅스에서는 /dev 디렉터리아래에여러가지특수파일이들어있습니다. 이파일을장치파일이라고하고, 이파일은일반파일과는다르게동작합니다. 장치파일중에서가장많은종류가블럭장치와캐릭터장치에대한장치파일입니다. 이파일은실제(리눅스커널에들어있는) 드라이버에대한인터페이스역할을합니다. (그리고리눅스커널에들어있는드라이버는하드웨어에접근합니다.) 혼하지는않지만또다른종류의장치파일이있는데, 파이프라고합니다. 아래 표에중요한장치파일몇개가쓰여있습니다.

fd0	첫번째 플로피 드라이브
fd1	두번째 플로피 드라이브

sda	첫번째 하드디스크
sdb	두번째 하드디스크
sda1	첫번째 하드디스크의 첫번째 파티션
sdb7	두번째 하드디스크의 일곱번째 파티션

sr0	첫번째 CD-ROM
sr1	두번째 CD-ROM

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

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

null	이 장치로 들어가는 데이터는 모두 사라집니다
zero	이 장치에서 끊임없이 0 을 읽을 수 있습니다

D.1.1 마우스 설정하기

마우스는 리눅스 콘솔과 (gpm 사용) X 윈도우 환경 모두에서 사용할 수 있습니다. 보통 gpm 과 X 서버 자체를 설치하기만 하면 마우스를 사용할 수 있습니다. 두 환경 모두 마우스 장치로 /dev/input/mice 를 사용합니다. 마우스 프로토콜은 gpm 에서는 exps2, X 환경에서는 ExplorerPS/2 입니다. 설정 파일은 /etc/gpm.conf 와 /etc/X11/xorg.conf 입니다.

마우스를 사용하려면 특정 커널 모듈을 읽어 들여야 할 수 있습니다. 대부분 올바른 모듈을 자동으로 찾아내

지만, 예전시리얼마우스나버스마우스¹, 매우오래된컴퓨터의마우스는찾아내지못할수있습니다. 여러가지마우스종류의리눅스커널모듈은아래표에있습니다:


모듈	설명
psmouse	PS/2 마우스 (자동으로찾아냄)
usbhid	USB 마우스 (자동으로찾아냄)
sermouse	대부분의시리얼마우스
logibm	Logitech 어댑터카드에연결된버스마우스
inport	ATI 나마이크로소프트 InPort 카드에연결된버스마우스

마우스드라이버모듈을읽어들이려면 modconf 명령을 (같은이름의패키지에들어있습니다) 사용할수있습니다. 모듈은 kernel/drivers/input/mouse 분류에있습니다.

D.2 태스크마다필요한디스크공간

모든표준패키지가들어있고기본커널을사용하는 amd64 아키텍처의표준설치용량은 800MB 의디스크공간을차지합니다. “표준시스템” 태스크를선택하지않으면최소의베이스시스템설치는 613MB 를차지합니다.

중요



두경우모두, 설치가끝나고임시파일을지운 후에실제차지하는디스크용량입니다. 저널링파일과같이파일시스템에서사용하는오버헤드는감안하지않았습니다. 즉 이보다더큰디스크공간이설치하는 도중에도필요하고시스템을실제사용할때도필요합니다.

다음표는 aptitude 에서표시하는값으로 (tasksel 에들어있는) 태스크에필요한용량입니다. 태스크중에는겹치는부분이있기때문에두개의태스크를같이설치하면숫자를합친전체크기보다는작을수도있습니다.

기본값으로설치프로그램은그놈데스크톱환경을설치합니다. 하지만특별한 CD 이미지를사용하거나, 부팅한다음에원하는데스크톱환경을지정하면다른데스크톱환경을선택할수도있습니다. (6.3.5.2절참고.)

파티션의크기를결정할때, 표준설치의크기에다음표에있는크기를더해야합니다. “설치크기” 에들어있는크기의대부분은 /usr 및 /lib 에서차지합니다. “다운로드크기” 는 (일시적으로) /var 에필요합니다.

태스크	설치크기 (MB)	다운로드크기 (MB)	설치하는데필요한공간 (MB)
데스크톱환경			
• 그놈 (기본값)	3163	935	4098
• KDE 플라스마;KDE Plasma	3044	911	3955
• Xfce	2122	593	2715
• LXDE	2133	602	2735
• MATE	2288	644	2932
• 시나몬	2878	843	3721
노트북컴퓨터	29	9	38
웹서버	40	9	49
인쇄서버	407	95	502
SSH 서버	1	0	1

영어가아닌언어로설치한다면 tasksel 에서자동으로 지역화태스크를 (해당언어에대한태스크가있다면) 설치합니다. 언어마다필요한공간이다릅니다. 다운로드하고설치하는데최대 350MB 까지의공간이있어야합니다.

¹ 시리얼마우스는일반적으로 9 핀 D 형커넥터를사용하고버스마우스는 8 핀둥근커넥터를사용합니다. PS/2 마우스의 6 핀커넥터나 ADB 마우스의 4 핀커넥터와혼용할수있습니다.

D.3 유닉스/리눅스시스템에서데비안 GNU/리눅스설치하기

이부분은설명서의다른부분에설명되어있는 ncurses 기반메뉴방식설치프로그램을사용하지않고기존의 Unix·Linux 시스템에서데비안 GNU/리눅스를설치하는방법을설명합니다. 이 “크로스설치” HOWTO 는 Red Hat, Mandriva, SUSE 에서데비안 GNU/리눅스로이동하는사용자의요구로작성되었습니다. 여기서는 *nix 명령의입력에대해숙지하고파일시스템을조작할수있는것이전제가되고있습니다. 여기서는 #가 데비안 chroot 에입력된명령을보여주고 \$ 는사용자의기존시스템에서입력되는명령을나타냅니다.

일단새로운데비안시스템에맞게설정하기만하면, 기존사용자데이터를 (있다면) 옮겨와서계속사용할 수있습니다. 따라서이것은 “다운타임없음” 에서데비안 GNU/리눅스설치됩니다. 또한이것은여러가지 부팅설치미디어와잘되지않는하드웨어에서좋은설치방법입니다.

참고



대부분수동으로해야하므로, 시스템의대부분의기본설정을직접해야할수도있습니다. 일반적인설치를할경우보다데비안및 Linux 에대한지식이많이필요합니다. 또이렇게설치해서일반적인설치와똑같은시스템이될것으로기대할수없습니다. 또시스템의기본적인단계에지나지않습니다. 추가로설치및설정이필요하게될지도모릅니다.

D.3.1 시작하기

기존유닉스용파티션도구를이용해하드드라이브를필요한대로다시파티션하십시오. 최소한파일시스템한개를스왑으로만드십시오. 콘솔만설치하는경우는약 613MB 의공간이필요하고 X 를설치한다면약 2133MB 가 (그놈이나 KDE 플라즈마같은데스크톱환경을설치한다면이보다더) 필요합니다.

그리고파티션에파일시스템을만드십시오. 예를들어 /dev/sda6 파티션에 ext3 파일시스템을만드는 경우라면 (여기에제에서루트파티션입니다):

```
# mke2fs -j /dev/sda6
```

ext2 파일시스템을만드는경우라면 -j 옵션을빼십시오.

스왑을다음과같이초기화하고활성화하십시오 (파티션번호는데비안스왑파티션에파티션번호로바꾸십시오):

```
# mkswap /dev/sda5
# sync
# swapon /dev/sda5
```

파티션 /mnt/debinst (설치지점, 새로운시스템의 root (/) 파일시스템에있습니다) 에마운트하고하십시오. 엄밀히말하면마운트위치이름은아무거나상관없습니다. 이후의설명에서이것을사용합니다.

```
# mkdir /mnt/debinst
# mount /dev/sda6 /mnt/debinst
```

참고



파일시스템의일부를 (예를들어 /usr) 별도의파티션에마운트하려면, 다음단계로 넘어가기전에그디렉터리를수동으로만들어서마운트해야합니다.

D.3.2 debootstrap 설치

데비안설치프로그램에서사용하는유틸리티에서데비안베이스시스템을설치하는공식적인방법으로인정받고있는것은 debootstrap 입니다. wget 와 ar 를사용하지만 /bin/sh 와기본적인 Unix/Linux 도구²에

² 에는 sed, grep, tar, gzip 같은 GNU 핵심유틸리티가들어있습니다.

만약존하고있습니다. 기존시스템에아직설치되어있지않으면 wget 와 ar 를설치한다음 debootstrap 다운로드설치하십시오.

아니면, 수동으로설치하려면다음과같이합니다. deb 파일을풀작업폴더를다음과같이만드십시오:

```
# mkdir work
# cd work
```

debootstrap 바이너리는데비안아카이브 (아키텍처에 맞는파일을선택) 에있습니다. pool에서 debootstrap deb 파일을다운로드하고, 작업폴더에패키지를복사하고, 파일을추출합니다. 파일을설치하려면 root 권한이필요할수도있습니다.

```
# ar -x debootstrap_0.X.X_all.deb
# cd /
# zcat /full-path-to-work/work/data.tar.gz | tar xv
```

D.3.3 debootstrap 실행

debootstrap 를실행하면아카이브에서필요한파일을직접다운로드할수있습니다. 다음명령예제에서는 ftp.kr.debian.org/debian 하고있지만네트워크에서가까운데비안아카이브미러사이트를입력할수있습니다. 미러사이트는 <http://www.debian.org/mirror/list>에목록이있습니다.

stretch 데비안 GNU/리눅스 CD 를 /cdrom 에마운트했다면 http URL 대신에 file URL 을쓸수있습니다: file:/cdrom/debian/

debootstrap 명령에서 ARCH 를다음중의하나로바꾸십시오: alpha, amd64, arm64, armel, armhf, i386, mips, mips64el, mipsel, powerpc, ppc64el, s390x.

```
# /usr/sbin/debootstrap --arch ARCH stretch \
  /mnt/debinst http://ftp.us.debian.org/debian
```

대상아키텍처가호스트와다르면, --foreign 옵션을붙여야합니다.

D.3.4 베이스시스템설정

이제디스크에진정한데비안시스템을 (많이작지만) 손에넣었습니다. 거기에 chroot 하십시오:

```
# LANG=C.UTF-8 chroot /mnt/debinst /bin/bash
```

타겟아키텍처가호스트와다르면, 먼저 qemu-user-static 을새 호스트에복사해야합니다:

```
# cp /usr/bin/qemu-ARCH-static /mnt/debinst/usr/bin
# LANG=C.UTF-8 chroot /mnt/debinst qemu-ARCH-static /bin/bash
```

chroot 후, 데비안기본시스템과호환되는터미널정의가필요할수있습니다. 예를들어다음과같이합니다.

```
# export TERM=xterm-color
```

TERM 값에따라 ncurses-term 패키지를설치해야할수도있습니다.

타겟아키텍처가호스트와다르면, 멀티스테이지부팅단계를마쳐야합니다:

```
/debootstrap/debootstrap --second-stage
```

D.3.4.1 장치파일만들기

이렇게하면 /dev/ 에는아주기초적인장치파일만들어있게됩니다. 다음단계로진행하려면장치파일이몇개더필요합니다. 여러가지방법이있고, 이중에어떤방법을이용할지는설치에사용하는호스트시스템이무엇이냐에따라, 그리고모듈식커널을이용할것인가아닌가, 그리고새시스템에동적인 (예를들어 udev 사용) 장치파일을사용할지고정장치파일을사용할지에따라달라집니다.

사용할수있는옵션몇가지를설명하면:

- makedev 패키지를설치하고, 다음명령으로기본적인고정장치파일의기본모음을 (chroot 상태에서) 만듭니다

```
# apt install makedev
# mount none /proc -t proc
# cd /dev
# MAKEDEV generic
```

- MAKEDEV 를이용해수동으로장치파일을직접지정해서만듭니다
- 호스트시스템의 /dev 를대상시스템의 /dev 디렉터리에연결합니다. 어떤패키지는 postinst 스크립트를실행하면서장치파일을만들수도있습니다. 그러므로이옵션은주의해서사용해야합니다.

D.3.4.2 파티션마운트하기

/etc/fstab 를만들어야합니다.

```
# editor /etc/fstab
```

다음예제를필요에맞게편집할수있습니다.

```
# /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           /media/floppy    auto      noauto,rw,exec,  0    0
/dev/cdrom         /media/cdrom     iso9660   noauto,ro,user,  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
```

/etc/fstab 에서지정한파일시스템을모두마운트 mount-a 라고합니다. 또한파일시스템을하나하나 마운트하려면다음과같이하십시오:

```
# mount /path # e.g.: mount /usr
```

현재데비안시스템에서이동식미디어의마운트지점을 /media 하고있지만, / 에심볼릭링크를호환유지하고있습니다. 다음예제와같이필요한경우작성하십시오:

```
# cd /media
# mkdir cdrom0
# ln -s cdrom0 cdrom
# cd /
# ln -s media/cdrom
```

proc 파일시스템은어디서나몇번이라도장착할수있지만, 관습으로 /proc 에마운트합니다. mount -a 를사용하지않으면다음과같이진행하기전에꼭 proc 을마운트하십시오.

```
# mount -t proc proc /proc
```

ls /proc 명령을실행하면여러파일이들어있는디렉터리내용을표시합니다. 이명령이실패하면 chroot 바깥에서 proc 을마운트할수있습니다:

```
# mount -t proc proc /mnt/debinst/proc
```

D.3.4.3 시간대설정하기

/etc/adjtime 파일의 3 번째 줄을 “UTC” 로설정하면하드웨어시계값을 UTC 로해석하고, “LOCAL” 로 설정하면로컬시각으로해석합니다. 다음명령어로설정할수있습니다.

```
# editor /etc/adjtime
```

예를들어다음과같이합니다:

```
0.0 0 0.0
0
UTC
```

다음명령으로표준시간대를설정합니다.

```
# dpkg-reconfigure tzdata
```

D.3.4.4 네트워크설정하기

” 64-bit ARM 에서는현재실험버전네트워크설정을하려면, /etc/network/interfaces, /etc/resolv.conf, /etc/hostname 과 /etc/hosts 을편집하십시오.

```
# editor /etc/network/interfaces
```

다음은 /usr/share/doc/ifupdown/examples 간단한예입니다:

```
#####
# /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
```

/etc/resolv.conf 에이름서버와 search 명령을입력하십시오:

```
# editor /etc/resolv.conf
```

다음은 /etc/resolv.conf 의간단한예입니다:

```
search hqdom.local
nameserver 10.1.1.36
nameserver 192.168.9.100
```

시스템의호스트이름 (2 글자에서 63 글자까지) 를입력하십시오:

```
# echo DebianHostName > /etc/hostname
```

또한 IPv6 를지원하는기본적인 /etc/hosts 는다음과같이합니다:

```

127.0.0.1 localhost
127.0.1.1 DebianHostName

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

```

여러네트워크카드를가지고있다면 /etc/modules 파일에원하는순서로드라이버모듈이름을배치하십시오. 그래야부팅할때각카드가의도한해당인터페이스이름으로 (eth0, eth1 등) 연결됩니다.

D.3.4.5 APT 설정하기

debootstrap 은아주기본적인 /etc/apt/sources.list 파일을만드므로추가패키지를설치할수있습니다. 하지만이외에소스를추가해야할경우가있습니다. 예를들어보안업데이트의소스패키지를설정할수있습니다:

```

deb-src http://ftp.us.debian.org/debian stretch main

deb http://security.debian.org/ stretch/updates main
deb-src http://security.debian.org/ stretch/updates main

```

sources.list 파일을고친다음에꼭 apt update 를실행하십시오.

D.3.4.6 로캘및키보드설정하기

영어가아닌언어를사용할때로캘을설정하려면 locales 지원패키지를설치하고그패키지를설정하십시오. 지금은 UTF-8 로캘사용을권장합니다:

```

# apt install locales
# dpkg-reconfigure locales

```

키보드를설정하려면 (키보드설정이필요한경우):

```

# apt install console-setup
# dpkg-reconfigure keyboard-configuration

```

chroot 안에서는키보드를설정할수없으니유의하십시오. 다시시작한다음에설정합니다.

D.3.5 커널설치

이시스템을시작할수있도록한다면, Linux 커널과부트로더가필요합니다. 다음과같이하여패키지로만든 커널을확인하십시오:

```
# apt search linux-image
```

다음패키지이름을사용하려면커널패키지를설치합니다.

```
# apt install linux-image-arch-etc
```

D.3.6 부트로더설정하기

데비안 GNU/리눅스시스템을부팅가능하게만드려면, 설치한커널과루트파티션을읽어들이기도록부트로더를설치하십시오. debootstrap 은부트로더를설치하지않으니주의하십시오. 설치하는데데비안 chroot 내부 apt 를사용할수있습니다.

앞서 /dev/sda 장치파일을만들었다고가정합니다. grub2 를설치하는다른방법도있지만, 이부록이다른범위를벗어나는내용입니다.

D.3.7 원격접근: SSH 설치및접근설정

콘솔을통해시스템에로그인할수있으면, 이부분을넘어가도됩니다. 네트워크를통해시스템에접근해야하는경우, SSH 를설치하고접근을설정해야합니다.

```
# apt install ssh
```

암호를사용한루트로그인은기본적으로막혀있습니다. 그러니접근설정은암호를설정하고암호를사용한루트로그인을열어주면됩니다:

```
# passwd
# editor /etc/ssh/sshd_config
```

다음옵션을사용해야합니다:

```
PermitRootLogin yes
```

루트계정에 ssh 키를설정해도됩니다:

```
# mkdir /root/.ssh
# cat << EOF > /root/.ssh/authorized_keys
ssh-rsa ....
EOF
```

마지막으로루트가아닌사용자를추가하고암호를설정해서접근을설정할수있습니다.

```
# adduser joe
# passwd joe
```

D.3.8 마지막처리

앞에서말한것처럼, 설치한시스템은아주기초적인시스템입니다. 시스템을좀더괜찮게만드려면, 쉬운방법으로 “standard” 우선순위의모든패키지를설치하면됩니다:

```
# tasksel install standard
```

물론 apt 를이용해패키지를하나하나선택해서설치할수도있습니다.

설치한다음에 /var/cache/apt/archives/ 밑에다운로드한패키지가많이들어있게됩니다. 다음명령을실행하면디스크공간을좀더확보할수있습니다:

```
apt clean
```

D.4 PPP 오버이더넷을 (PPPoE) 이용해데비안 GNU/리눅스설치하기

어떤국가에서는 PPP 오버이더넷 (PPPoE) 이초고속인터넷연결에서 (ADSL 혹은케이블) 인터넷서비스제공자에게연결하는일반적인프로토콜입니다. PPPoE 연결은기본값으로는지원하지않지만아주간단히동작하게만들수있습니다. 여기서그방법을설명합니다.

설치할때설정된 PPPoE 연결은설치한시스템을다시시작한다음에도사용할수있습니다. (7장참고.)

설치할때 PPPoE 를설정하고사용하는옵션을사용하려면, CD-ROM/DVD 이미지중하나를사용해야합니다. 다른설치방법에서는지원하지않습니다. (예를들어 netboot 에서는지원하지않습니다.)

PPPoE 를통한설치는다른설치와거의동일합니다. 아래에서다른부분을설명합니다.

- 부팅파라미터로 `modules=ppp-udeb` 을사용해설치프로그램을부팅하십시오. 이렇게하면자동으로 PPPoE 설정을하는컴포넌트를 (`ppp-udeb`) 읽어들이어서실행합니다.
- 마찬가지로설치처음단계를계속하십시오. (언어, 국가및키보드선택. 그리고필요한경우설치프로그램컴포넌트를추가로읽어들이기³.)
- 다음단계는네트워크하드웨어찾기입니다. 시스템에들어있는모든이더넷카드를찾습니다.

³ `ppp-udeb` 컴포넌트가단계에서추가컴포넌트로읽어들입니다. 중간이나낮은우선순위로설치한다면 (전문가모드), 부팅프롬프트에서 “modules” 파라미터를설정하지않고 `ppp-udeb` 을선택할수있습니다.

- 그다음에 실제로 PPPoE 설정을 시작합니다. 설치 프로그램에서 검색한 모든 이더넷 장치에 대해서 PPPoE 콘센트레이터 (PPPoE 연결을 처리하는 서버) 가 있는지 찾아봅니다.
첫번째 시도할 때 콘센트레이터를 찾지 못하는 경우도 있습니다. 네트워크가 느리거나 너무 로드가 심하거나 서버에 문제가 있는 경우 이런 일이 발생할 수 있습니다. 대부분의 경우 다시 한번 콘센트레이터를 검색해 보면 성공합니다. 다시 시도해 보려면 설치 프로그램의 메인 메뉴에서 PPPoE 연결 설정 및 시작을 선택하십시오.
- 콘센트레이터를 찾으면, 로그인 정보를 (PPPoE 사용자 이름 및 암호) 입력할 수 있게 물어봅니다.
- 여기서 설치 프로그램은 입력한 정보를 이용해 PPPoE 에 연결합니다. 올바른 정보를 입력했다면, PPPoE 연결을 설정하고 PPPoE 를 이용해 인터넷에 연결해 (필요한 경우) 패키지를 인터넷에서 받아올 수 있게 됩니다. 로그인 정보가 틀렸거나 기타 오류가 발생한 경우에는 설치 프로그램이 멈춥니다. 하지만 PPPoE 연결 설정 및 시작을 선택하면 다시 설정을 할 수 있습니다.

Appendix E

문서관리정보

E.1 문서정보

이 설명서는 초기 Debian 설치 설명서를 바탕으로 한 boot-floppies 의 Woody 설치 설명서를 바탕으로 Sarge 의 debian-installer 를 위해 작성되었습니다. 또한 2003 년에 GPL 로 발표된 Progeny 배포판 설명서에 기반하고 있습니다.

이 문서는 닥북 (DocBook) XML 형식으로 작성되어 있습니다. docbook-xml 과 docbook-xsl 패키지 에 있는 정보를 이용해서 여러 가지 프로그램이 문서의 여러 가지 형식의 출력을 만들어 냅니다.

문서를 유지 보수 하기 좋도록 엔티티와 프로파일 속성과 같은 여러 가지 XML 기능을 이용합니다. 엔티티와 속성은 프로그래밍 언어의 변수 및 조건문과 비슷한 역할을 합니다. 이 문서의 XML 소스에는 여러 가지 아키텍처에 대한 정보가 모두 들어 있고, 각 아키텍처에 해당하는 텍스트를 분리 하는데 프로파일 속성을 사용합니다.

이 문서의 한국어 번역에 참여한 사람은 다음과 같습니다. 류창우, 박선재, 이경순, 이광우, 이유미, 장석문, 최병현. 도움을 주신 모든 분에게 감사드립니다.

E.2 이 문서에 참여하기

이 문서에 대해 문제 점이나 의견이 있으면 installation-guide 패키지를 이용해 버그 리포트를 보내 주십시오. reportbug 패키지를 참고하시고, [데비안버그추적시스템](#) 온라인 문서를 읽어보십시오. 해당 문제 점에 벌써 보고된 상태인지 알아 보려면 [installation-guide 에 해결 안된 버그 목록](#) 을 보는 것도 좋습니다. 이미 보고된 버그인 경우, xxxx@bugs.debian.org 주소에 추가로 보고 할 만한 정보나 도움이 될 만한 정보를 메일로 보낼 수 있습니다. 여기서 xxxx 는 보고한 버그의 번호입니다.

더 좋은 방법으로, 이 문서의 닥북 소스 코드를 구해서 패치를 만드십시오. 닥북 소스 코드는 [debian-installer WebSVN](#) 에 있습니다. 닥북에 익숙하지 않더라도 걱정하지 마십시오. 설명서 디렉터리에 간단한 쪽지가 있고 이 쪽지를 읽는 걸로 시작하십시오. 닥북은 HTML 과 비슷 하면서도 텍스트의 화면 표시 보다는 의미에 중점을 두는 형식입니다. 패치는 (아래에 있는) debian-boot 메일링 리스트로 보내 주십시오. SVN 으로 소스 코드를 받는 방법은, 소스 코드가 있는 맨위 디렉터리에서 [README](#) 파일을 보십시오.

절대로 이 문서의 저자에게 직접 연락하지 마십시오. debian-installer 에 대한 토론을 하는 (이 설명서에 대한 토론 포함) 리스트가 있습니다. 이 메일링 리스트는 debian-boot@lists.debian.org 입니다. 이 리스트에 가입하는 방법은 [데비안 메일링 리스트 가입](#) 페이지에 있고, [데비안 메일링 리스트 아카이브](#) 를 온라인으로 볼 수 있습니다.

E.3 중요 기여자들

이 문서는 Bruce Perens, Sven Rudolph, Igor Grobman, James Treacy, 그리고 Adam Di Carlo 가 작성했습니다. Sebastian Ley 는 설치 하우투를 썼습니다.

Miroslav Kuře 씨의 사지 (Sarge) 의 debian-installer 에 많은 새로운 기능을 문서화했습니다.

매우 많은 Debian 사용자와 개발자가 이 문서에 기여하고 있습니다. 특히 다양한 문서를 편집 저술하고 있는 Michael Schmitz (m68k 지원), Frank Neumann ([Amiga install manual](#) 의 원저자), Arto Astala, Eric Delaunay / Ben Collins (SPARC 정보), Tapio Lehtonen, Stéphane Bortzmeyer 에는 상당한 협력을 받았습니다. 또한 Pascal Le Bail 에 USB 메모리로 부팅하는 방법에 대한 유용한 정보를 주신 것에 감사드립니다.

매우 큰 도움이 되었던 글과 정보는 다음 문서에 들어 있습니다: [짐민타 \(Jim Mintha\) 의 네트워크 부팅에 관한 하우투 \(URL 없음\)](#), [Debian FAQ](#), [Linux/m68k FAQ](#), [Linux for SPARC Processors FAQ](#), [Linux/Alpha FAQ](#).

자유롭게 사용할 수 있고 풍부한 정보가 들어있는 이 문서의 관리자들에게 깊은 감사를 표합니다.

이 설명서에서 chroot 을 이용한 설치에 관한 부분은 (D.3절) 카르스텐 M. 셸프 (Karsten M. Self) 에 저작권이 있는 문서의 일부분에서 가져왔습니다.

E.4 상표권안내

모든 상표는 그 상표권자의 소유입니다.

Appendix F

GNU 일반공중사용허가서

참고



This is an unofficial translation of the GNU General Public License into Korean language. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL —only the original **English text** of the GNU GPL does that. However, we hope that this translation will help Korean speakers to better understand the GNU GPL.

이문서는 GNU General Public License 의한국어번역입니다. 이번역문은자유소프트웨어재단이발행한문서가아니고, GNU GPL 소프트웨어의배포조건에대해법적인효력이없습니다. GNU GPL 의 **영어로된원문텍스트**만이효력을가집니다. 이번역문은한국어사용자가 GNU GPL 을더쉽게이해하기위한용도입니다.

2 판, 1991 년 6 월

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

누구든지 본 사용 허가서를 있는 그대로 복제하고 배포할 수 있습니다. 그러나 본문에 대한 수정은 허용되지 않습니다.

F.1 전문

소프트웨어에 적용되는 대부분의 사용 허가서 (license) 들은 소프트웨어에 대한 수정과 공유의 자유를 제한하려는 것을 그 목적으로 합니다. 그러나 GNU 일반공중사용 허가서 (이하, "GPL" 이라고 칭합니다.) 는 자유 소프트웨어에 대한 수정과 공유의 자유를 모든 사용자들에게 보장하기 위해서 성립된 것입니다. 자유 소프트웨어 재단이 제공하는 대부분의 소프트웨어들은 GPL 에 의해서 관리되고 있으며, 몇몇 소프트웨어에는 별도의 사용 허가서인 GNU 라이브러리 일반공중사용 허가서 (GNU Library General Public License) 를 대신 적용하기도 합니다. 자유 소프트웨어란, 이를 사용하려고 하는 모든 사람에 대해서 동일한 자유와 권리가 함께 양도되는 소프트웨어를 말하며 프로그램 저작자의 의지에 따라 어떠한 종류의 프로그램에도 GPL 을 적용할 수 있습니다. 따라서 여러분이 만든 프로그램에도 GPL 을 적용할 수 있습니다.

자유 소프트웨어를 언급할 때 사용되는 "자유" 라는 단어는 무료 (無料) 를 의미하는 금전적인 측면의 자유가 아니라 구속되지 않는다는 관점에서 의 자유를 의미하며, GPL 은 자유 소프트웨어를 이용한 복제와 개작, 배포와 수익 사업 등의 가능한 모든 형태의 자유를 실질적으로 보장하고 있습니다. 여기에는 원시 코드 (source code) 의 전 부 또는 일부를 원용해서 개선된 프로그램을 만들거나 새로운 프로그램을 창작할 수 있는 자유가 포함되며, 자신에게 양도된 이러한 자유와 권리를 보다 명확하게 인식할 수 있도록 하기 위한 규정도 포함되어 있습니다.

GPL 은 GPL 안에 소프트웨어를 양도 받을 사용자의 권리를 제한하는 조항과 단서를 별항으로 추가시키지 못하게 함으로써 사용자들의 자유와 권리를 실제적으로 보장하고 있습니다. 자유 소프트웨어의 개작과 배포에 관계하고 있는 사람들은 이러한 무조건적인 권리 양도 규정을 준수해야만 합니다.

예를 들어 GPL 프로그램을 배포할 경우에는 프로그램의 유료 판매나 무료 배포에 관계없이 자신이 해당 프로그램에 대해서 가질 수 있었던 모든 권리를, 프로그램을 받게 될 사람에게 그대로 양도 해주어야 합니다. 이 경우, 프로그램의 원시 코드를 함께 제공하거나 원시 코드를 구할 수 있는 방법을 확실히 알려 주어야 하고 이러한 모든 사항들을 사용자들이 분명히 알 수 있도록 명시해야 합니다.

자유 소프트웨어 재단은 다음과 같은 두 가지 단계를 통해서 사용자들을 권리를 보호합니다. (1) 소프트웨어에 저작권을 설정합니다. (2) 저작권의 양도에 관한 실정법에 의해서 유효한 법률적 효력을 갖는 GPL 을 통해 소프트웨어를 복제하거나 개작 및 배포할 수 있는 권리를 사용자들에게 부여합니다.

자유 소프트웨어를 사용하는 사람들은 반복적인 재배포 과정을 통해 소프트웨어 자체에 수정과 변형이 일어날 수도 있으며, 이는 최초의 저작자가 만든 소프트웨어가 갖고 있는 문제가 아닐 수 있다는 개연성을 인식하고 있어야 합니다. 우리는 개작과 재배포 과정에서 다른 사람의 해 발생된 문제로 인해 프로그램 원저작자들의 신망이 훼손되는 것을 원하지 않습니다. GPL 에 자유 소프트웨어에 대한 어떠한 형태의 보증도 규정하지 않는 이유는 이러한 점들이 고려되었기 때문이며, 이는 프로그램 원저작자와 자유 소프트웨어 재단의 자유로운 활동을 보장하는 현실적인 수단이기도 합니다.

특허 제도는 자유 소프트웨어의 발전을 위협하는 요소일 수밖에 없습니다. 자유 프로그램을 재배포하는 사람들이 개별적으로 특허를 취득하게 되면, 결과적으로 그 프로그램이 독점 소프트웨어가 될 가능성이 있습니다. 자유 소프트웨어 재단은 이러한 문제에 대처하기 위해서 어떠한 특허에 대해서도 그 사용 권리를 모든 사람들 (이하, "공중(公衆)" 이라고 칭합니다.) 에게 자유롭게 허용하는 경우에 한해서만 자유 소프트웨어와 함께 사용할 수 있다는 것을 명확히 밝히고 있습니다.

복제 (copying) 와 개작 (modification) 및 배포 (distribution) 에 관련된 구체적인 조건과 규정은 다음과 같습니다.

F.2 GNU 일반공중사용허가서 (GNU GENERAL PUBLIC LICENSE)

복제와 개작 및 배포에 관한 조건과 규정

제 0 조. 본 허가서는 GNU 일반공중사용허가서의 규정에 따라 배포될 수 있다는 사항이 저작권자에 의해 서명된 모든 컴퓨터 프로그램 저작물에 대해서 동일하게 적용됩니다. 컴퓨터 프로그램 저작물 (이하, "프로그램" 이라고 칭합니다.) 이란 특정한 결과를 얻기 위해서 컴퓨터 등의 정보 처리 능력을 가진 장치 (이하, "컴퓨터" 라고 칭합니다.) 내에서 직접 또는 간접으로 사용되는 일련의 지시 및 명령으로 표현된 창작물을 의미하고, "2 차적 프로그램" 이란 전술한 프로그램 자신 또는 저작권법의 규정에 따라 프로그램의 전부 또는 상당 부분을 윌용하거나 다른 언어로의 번역을 포함할 수 있는 개작 과정을 통해서 창작된 새로운 프로그램과 이와 관련된 저작물을 의미합니다. (이후로 다른 언어로의 번역은 별다른 제한 없이 개작의 범위에 포함되는 것으로 간주합니다.) "피양도자" 란 GPL 의 규정에 따라 프로그램을 양도 받은 사람의 의미하고, "원(原) 프로그램" 이란 프로그램을 개작하거나 2 차적 프로그램을 만들기 위해서 사용된 최초의 프로그램을 의미합니다.

본 허가서는 프로그램에 대한 복제와 개작 그리고 배포 행위에 대해서만 적용됩니다. 따라서 프로그램을 실행시키는 행위에 대한 제한은 없습니다. 프로그램의 결과물 (output) 에는, 그것이 프로그램을 실행시켜서 생성된 것인지 아닌지의 여부에 상관없이 결과물의 내용이 원 프로그램으로부터 파생된 2 차적 프로그램을 구성했을 때에 한해서 본 허가서의 규정들이 적용됩니다. 2 차적 프로그램의 구성 여부는 2 차적 프로그램 안에서의 원 프로그램의 역할을 토대로 판단합니다.

제 1 조. 적절한 저작권 표시와 프로그램에 대한 보증이 제공되지 않는다는 사실을 각각의 복제물에 명시하는 한, 피양도자는 프로그램의 원시 코드를 자신이 양도 받은 상태 그대로 어떠한 매체를 통해서도 복제하고 배포할 수 있습니다. 복제와 배포가 이루어질 때는 본 허가서와 프로그램에 대한 보증이 제공되지 않는다는 사실에 대해서 언급되었던 모든 내용을 그대로 유지시켜야 하며, 영문판 GPL 을 함께 제공해야 합니다.

배포자는 복제물을 물리적으로 인도하는데 소요된 비용을 청구할 수 있으며, 선택 사항으로 독자적인 유료 보증을 설정할 수 있습니다.

제 2 조. 피양도자는 자신이 양도 받은 프로그램의 전부나 일부를 개작할 수 있으며, 이를 통해서 2 차적 프로그램을 창작할 수 있습니다. 개작된 프로그램이나 창작된 2 차적 프로그램은 다음의 사항들을 모두 만족시키는 조건에 한해서, 제 1 조의 규정에 따라 또 다시 복제되고 배포될 수 있습니다.

- 파일을 개작할 때는 파일을 개작한 사실과 그 날짜를 파일 안에 명시해야 합니다.
- 배포하거나 공표하려는 저작물의 전부 또는 일부가 양도 받은 프로그램으로부터 파생된 것이라면, 저작물 전체에 대한 사용 권리를 본 허가서의 규정에 따라 공중에게 무상으로 허용해야 합니다.
- 개작된 프로그램의 일반적인 실행 형태가 대화형 구조로 명령어를 읽어들이는 방식을 취하고 있을 경우에는, 적절한 저작권 표시와 프로그램에 대한 보증이 제공되지 않는다는 사실, (별도의 보증을 설정한 경우라면 해당 내용) 그리고 양도 받은 프로그램을 본 규정에 따라 재배포할 수 있다는 사실과 GPL 사본을 참고할 수 있는 방법이 함께 포함된 문구가 프로그램이 대화형 구조로 평이하게 실행된 직후에 화면 또는 지면으로 출력되도록 작성되어야 합니다. (예외 규정: 양도 받은 프로그램이 대화형 구조를 갖추고 있다 하더라도)

도 통상적인 실행 환경에서 전술한 사항들이 출력되지 않는 형태였을 경우에는 이를 개작한 프로그램 또한 관련 사항들을 출력시키지 않아도 무방합니다.)

위의 조항들은 개작된 프로그램 전체에 적용됩니다. 만약, 개작된 프로그램에 포함된 특정 부분이 원 프로그램으로부터 파생된 것이 아닌 별도의 독립 저작물로 인정될 만한 상당한 이유가 있을 경우에는 해당 저작물의 개별적인 배포에는 본허가서의 규정들이 적용되지 않습니다. 그러나 이러한 저작물이 2 차적 프로그램의 일부로서 함께 배포된다면 개별적인 저작권과 배포 기준에 상관없이 저작물 모두에 본허가서가 적용되어야 하며, 전체 저작물에 대한 사용권리는 공중에게 무상으로 양도됩니다.

이러한 규정은 개별적인 저작물에 대한 저작자의 권리를 침해하거나 인정하지 않으려는 것이 아니라, 원 프로그램으로부터 파생된 2 차적 프로그램이나 수집 저작물의 배포를 일관적으로 규제할 수 있는 권리를 행사하기 위한 것입니다.

원 프로그램이나 원 프로그램으로부터 파생된 2 차적 프로그램들이 들로부터 파생되지 않은 다른 저작물과 함께 단순히 저장하거나 배포할 목적으로 동일한 매체에 모아놓은 집합물의 경우에는, 원 프로그램으로부터 파생되지 않은 다른 저작물에는 본허가서의 규정들이 적용되지 않습니다.

제 3 조. 피양도자는 다음 중 하나의 항목을 만족시키는 조건에 한해서 제 1 조와 제 2 조의 규정에 따라 프로그램 (또는 제 2 조에서 언급된 2 차적 프로그램) 을 목적 코드 (object code) 나 실행물 (executable form) 의 형태로 복제하고 배포할 수 있습니다.

- a. 목적 코드나 실행물에 상응하는 컴퓨터가 인식할 수 있는 완전한 원시 코드를 함께 제공해야 합니다. 원시 코드는 제 1 조와 제 2 조의 규정에 따라 배포될 수 있어야 하며, 소프트웨어의 교환을 위해서 일반적으로 사용되는 매체를 통해 제공되어야 합니다.
- b. 배포에 필요한 최소한의 비용만을 받고 목적 코드나 실행물에 상응하는 완전한 원시 코드를 배포하겠다는, 최소한 3 년간 유효한 약정을 함께 제공해야 합니다. 이 약정서는 약정서를 갖고 있는 어떠한 사람에 대해서도 유효해야 합니다. 원시 코드는 컴퓨터가 인식할 수 있는 형태여야 하고 제 1 조와 제 2 조의 규정에 따라 배포될 수 있어야 하며, 소프트웨어의 교환을 위해서 일반적으로 사용되는 매체를 통해 제공되어야 합니다.
- c. 목적 코드나 실행물에 상응하는 원시 코드를 배포하겠다는 약정에 대해서 자신이 양도 받은 정보를 함께 제공해야 합니다. (제 3 항은 위의 제 2 항에 따라 원시 코드를 배포하겠다는 약정을 프로그램의 목적 코드나 실행물과 함께 제공받았고, 동시에 비상업적인 배포를 하고자 할 경우에 한해서만 허용됩니다.)

저작물에 대한 원시 코드란 해당 저작물을 개작하기에 적절한 형식을 의미합니다. 실행물에 대한 완전한 원시 코드는 실행물에 포함된 모든 모듈들의 원시 코드와 이와 관련된 인터페이스 정의의 파일 모두, 그리고 실행물의 컴파일과 설치 제어하는 데 사용된 스크립트 전부를 의미합니다. 그러나 특별한 한 예외의 하나로서, 실행물이 실행될 운영체제의 주요 부분 (컴파일러나 커널 등) 과 함께 (원시 코드나 바이너리의 형태로) 일반적으로 배포되는 구성 요소들은 이러한 구성 요소 자체가 실행물에 수반되지 않는 원시 코드의 배포 대상에서 제외되어도 무방합니다.

목적 코드나 실행물을 지정한 장소로부터 복제해갈 수 있게 하는 방식으로 배포할 경우, 동일한 장소로부터 원시 코드를 복제할 수 있는 동등한 접근 방법을 제공한다면 이는 원시 코드를 목적 코드와 함께 복제되도록 설정하지 않았다고 하더라도 원시 코드를 배포하는 것으로 간주됩니다.

제 4 조. 본허가서에 의해 명시적으로 이루어지지 않는 한 프로그램에 대한 복제와 개작 및 하위 허가권 설정과 배포가 성립될 수 없습니다. 이와 관련된 어떠한 행위도 무효이며 본허가서가 보장한 권리는 자동으로 소멸됩니다. 그러나 본허가서의 규정에 따라 프로그램의 복제물이나 권리를 양도 받았던 제 3 자는 본허가서의 규정들을 준수하는 한, 배포자의 권리 소멸에 관계없이 사용상의 권리를 계속해서 유지할 수 있습니다.

제 5 조. 본허가서는 서명이나 날인이 수반되는 형식을 갖고 있지 않기 때문에 피양도자가 본허가서의 내용을 반드시 받아들여야 할 필요는 없습니다. 그러나 프로그램이나 프로그램에 기반한 2 차적 프로그램에 대한 개작 및 배포를 허용하는 것은 본허가서에 의해서만 가능합니다. 만약 본허가서에 동의하지 않을 경우에는 이러한 행위들이 법률적으로 금지됩니다. 따라서 프로그램 (또는 프로그램에 기반한 2 차적 프로그램) 을 개작하거나 배포하는 행위는 이에 따른 본허가서의 내용에 동의한다는 것을 의미하며, 복제와 개작 및 배포에 관한 본허가서의 조건과 규정들을 모두 받아들일 것이라는 의미로 간주됩니다.

제 6 조. 피양도자에 의해서 프로그램 (또는 프로그램에 기반한 2 차적 프로그램) 이 반복적으로 재배포될 경우, 각 단계에서의 피양도자는 본허가서의 규정에 따른 프로그램의 복제와 개작 및 배포에 대한 권리를 최초의 양도자로부터 양도 받은 것으로 자동적으로 간주됩니다. 프로그램 (또는 프로그램에 기반한 2 차적 프로그램) 을 배포할 때는 피양도자의 권리의 행사를 제한할 수 있는 어떠한 사항도 추가할 수 없습니다. 그러나 피양도자에게, 재배포가 일어날 시점에서의 제 3 의 피양도자에게 본허가서를 준수하도록 강제할 책임은 부과되지 않습니다.

제 7 조. 법원의 판결이나 특허권 침해에 대한 주장 또는 특허 문제에 국한되지 않은 그밖의 이유들로 인해서 본허가서의 규정에 배치되는 사항이 발생한다 하더라도 그러한 사항이 선행하거나 본허가서의 조건과 규정들이 면제되는 것은 아닙니다. 따라서 법원의 명령이나 합의 등에 의해서 본허가서에 위배되는 사항들이 발생 한 상황이라도 양측 모두를 만족시킬 수 없다면 프로그램은 배포될 수 없습니다. 예를 들면, 특정한 특허 관련 허가

가 프로그램의 복제물을 직접 또는 간접적인 방법으로 양도받은 임의의 제 3 자에게 해당 프로그램을 무상으로 재배포할 수 있게 허용하지 않는다면, 그러한 허가와 본 사용허가를 동시에 만족시키면서 프로그램을 배포할 수 있는 방법은 없습니다.

본 조항은 특정한 상황에서 본 조항의 일부가 유효하지 않거나 적용될 수 없을 경우에도 본 조항의 나머지 부분들을 적용하기 위한 의도로 만들어졌습니다. 따라서 그 이외의 상황에서는 본 조항을 전체적으로 적용하면 됩니다.

본 조항의 목적은 특허나 저작권 침해 등의 행위를 조장하거나 해당 권리를 인정하지 않으려는 것이 아니라, GPL 을 통해서 구현되어 있는 자유 소프트웨어의 배포 체계를 통합적으로 보호하기 위한 것입니다. 많은 사람들이 배포 체계에 대한 신뢰 있는 지원을 계속해 줌으로써 소프트웨어의 다양한 분야에 많은 공헌을 해주었습니다. 소프트웨어를 어떠한 배포 체계로 배포할 것인가를 결정하는 것은 전적으로 저작자와 기여자들의 의지에 달려 있는 것이지만, 일반 사용자들이 강요할 수 있는 문제는 아닙니다.

본 조항은 본 허가서의 다른 조항들에서 무엇이 중요하게 고려되어야 하는지를 명확하게 설명하기 위한 목적으로만 만들어진 것입니다.

제 8 조. 특허나 저작권이 설정된 인터페이스로 인해서 특정 국가에서 프로그램의 배포와 사용이 함께 또는 개별적으로 제한되어 있는 경우, 본 사용허가서를 프로그램에 적용한 최초의 저작권자는 문제가 발생하지 않는 국가에 한해서 프로그램을 배포한다는 배포상의 지역적 제한 조건을 명시적으로 설정할 수 있으며, 이러한 사항은 본 허가서의 일부로 간주됩니다.

제 9 조. 자유 소프트웨어 재단은 때때로 본 사용허가서의 개정판이나 신판을 공표할 수 있습니다. 새롭게 공표된 판은 당면한 문제나 현안을 처리하기 위해서 세부적인 내용에 차이가 발생할 수 있지만, 그 근본 정신에는 변함이 없을 것입니다. 각각의 판들은 판번호를 사용해서 구별됩니다. 특정한 판번호와 그 이후 판을 따른다는 사항이 명시된 프로그램에는 해당판이나 그 이후에 발행된 어떠한 판을 선택해서 적용해도 무방하고, 판번호를 명시하고 있지 않은 경우에는 자유 소프트웨어 재단이 공표한 어떠한 판번호의 판을 적용해도 무방합니다.

제 10 조. 프로그램의 일부를 본 허가서와 배포 기준이 다른 자유 프로그램과 함께 결합하고자 할 경우에는 해당 프로그램의 저작자로부터 서면 승인을 받아야 합니다. 자유 소프트웨어 재단이 저작권을 갖고 있는 소프트웨어의 경우에는 자유 소프트웨어 재단의 승인을 얻어야 합니다. 우리는 이러한 요청을 수락하기 위해서 때때로 예외 기준을 만들기도 합니다. 자유 소프트웨어 재단은 일반적으로 자유 소프트웨어의 2 차적 저작물들을 모두 자유로운 상태로 유지시키려는 목적과 소프트웨어의 공유와 재 활용을 증진시키려는 두 가지 목적을 기준으로 승인 여부를 결정할 것입니다.

보증의 결여

제 11 조. 본 허가서를 따르는 프로그램은 무상으로 양도되기 때문에 관련 법률이 허용하는 한도 내에서 어떠한 형태의 보증도 제공되지 않습니다. 프로그램의 저작자와 배포자가 공동 또는 개별적으로 별도의 보증을 서면으로 제공할 때를 제외하면, 특정한 목적에 대한 프로그램의 적합성이나 상업성 여부에 대한 보증을 포함하여 어떠한 형태의 보증도 명시적이나 묵시적으로 설정되지 않은 "있는 그대로의" 상태로 이 프로그램을 배포합니다. 프로그램과 프로그램의 실행에 따라 발생할 수 있는 모든 위험은 피양도자에게 인수되며 이에 따른 보수 및 복구 비용을 위한 제반 경비 또한 피양도자가 모두 부담해야 합니다.

제 12 조. 저작권자나 배포자가 프로그램의 손상 가능성을 사전에 알고 있었다 하더라도 발생한 손실이 관련 법규에 의해 보호되고 있거나 이에 대한 별도의 서면 보증이 설정된 경우가 아니라면, 저작자나 프로그램을 원래의 상태 또는 개작한 상태로 제공한 배포자는 프로그램의 사용이나 비작동으로 인해 발생한 손실이나 프로그램 자체의 손실에 대해 책임지지 않습니다. 이러한 면책 조건은 사용자나 제 3 자가 프로그램을 조작함으로써 발생한 손실이나 다른 소프트웨어와 프로그램을 함께 동작시키는 것으로 인해서 발생된 데이터의 상실 및 부정확한 산출 결과에 만족하는 것이 아닙니다. 발생한 손실의 일반성이나 특수성 뿐 아니라 원인의 유발성 및 필연성도 전혀 고려되지 않습니다.

복제와 개작 및 배포에 관한 조건과 규정의 끝

F.3 새로운 프로그램에 GPL 을 적용하는 방법

새로운 프로그램을 개발하고 그 프로그램이 많은 사람들에게 최대한 유용하게 사용되기를 원한다면, 본 허가서의 규정에 따라 누구나 자유롭게 개작하고 재배포할 수 있는 자유 소프트웨어로 만드는 것이 최선의 방법입니다.

프로그램을 자유 소프트웨어로 만들기 위해서는 다음과 같은 사항을 프로그램에 추가하면 됩니다. 프로그램에 대한 보증이 제공되지 않는다는 사실을 가장 효과적으로 전달할 수 있는 방법은 원시 코드 파일의 시작 부분에 이러한 사항을 추가하는 것입니다. 각각의 파일에는 최소한 저작권을 명시한 행과 본 사용허가서의 전체 내용을 참고할 수 있는 위치 정보를 명시해야 합니다.

프로그램의 이름과 용도를 한 줄 정도로 설명합니다.
Copyright (C) 연도 프로그램 저작자의 이름

이 프로그램은 자유 소프트웨어입니다. 소프트웨어의 피양도자는 자유 소프트웨어 재단이 공표한 GNU 일반 공중 사용 허가서 2판 또는 그 이후 판을 임의로 선택해서, 그 규정에 따라 프로그램을 개작하거나

재배포할 수 있습니다.

이 프로그램은 유용하게 사용될 수 있으리라는 희망에서 배포되고 있지만, 특정한 목적에 맞는 적합성 여부나 판매용으로 사용할 수 있으리라는 목시적인 보증을 포함한 어떠한 형태의 보증도 제공하지 않습니다. 보다 자세한 사항에 대해서는 GNU 일반 공중 사용 허가서를 참고하시기 바랍니다.

GNU 일반 공중 사용 허가서는 이 프로그램과 함께 제공됩니다. 만약, 이 문서가 누락되어 있다면 자유 소프트웨어 재단으로 문의하시기 바랍니다. (자유 소프트웨어 재단: Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA)

또한, 사용자들이 프로그램을 배포한 사람에게 전자메일과서면으로 연락할 수 있는 정보를 추가해야 합니다.

프로그램이 명령어 입력 방식에 의한 대화형 구조를 택하고 있다면, 프로그램이 대화형 방식으로 실행되었을 때 다음과 같은 주의 사항이 출력되어야 합니다.

Gnomovision version 69, Copyright (C) 연도 프로그램 저작자의 이름
Gnomovision 프로그램에는 제품에 대한 어떠한 형태의 보증도 제공되지 않습니다. 보다 자세한 사항은 'show w' 명령어를 실행해서 참고할 수 있습니다. 이 프로그램은 자유 소프트웨어입니다. 이 프로그램은 배포 규정을 만족시키는 조건하에서 자유롭게 재배포될 수 있습니다. 배포에 대한 규정들은 'show c' 명령어를 통해서 참고할 수 있습니다.

'show w' 와 'show c' 는 GPL 의해 당부분을 출력하기 위한 가상의 명령어입니다. 따라서 'show w' 나 'show c' 가 아닌 다른 형태를 사용해도 무방하며, 마우스 클릭이나 메뉴 방식과 같은 프로그램에 적합한 다른 형식을 사용해도 괜찮습니다.

만약, 프로그램 저작자가 학교나 기업과 같은 단체나 기관에 고용되어 있다면 프로그램의 자유로운 배포를 위해서 고용주나 해당 기관장으로부터 프로그램에 대한 저작권 포기 각서를 받아야 합니다. 예를 들면 다음과 같은 형식이 될 수 있다. (아래의 문구를 실제로 사용할 경우에는 예로 사용된 이름들을 실제 이름으로 대체 하시면 됩니다.)

본사는 제임스 해커가 만든 (컴파일러에서 패스를 생성하는)
'Gnomovision' 프로그램에 관련된 모든 저작권을 포기합니다.

1989년 4월 1일 부사장: Ty Coon Ty Coon의 서명

GNU 일반공중사용허가서는 자유 소프트웨어를 독점 소프트웨어와 함께 결합시키는 것을 허용하지 않습니다. 만약, 작성된 프로그램이 서브루틴 라이브러리일 경우에는 독점 소프트웨어가 해당 라이브러리를 링크할 수 있도록 허용하는 것이 보다 효과적으로 활용될 수 있는 방법이라고 생각할 수도 있을 것입니다. 이러한 경우에는 본 허가서 GNU Lesser General Public License 를 사용함으로써 소기의 목적을 충족시킬 수 있습니다.