

Debian GNU/Linux インストールガイド

July 31, 2021

Debian GNU/Linux インストールガイド

製作著作 © 2004 -2021 the Debian Installer team

本マニュアルはフリーソフトウェアです。GNU 一般公有使用許諾にそって、配布・改変する事ができます。付録F のライセンスを参照してください。

このマニュアルのビルドバージョン: 20210730.

Contents

1	ようこそ Debian へ	1
1.1	Debian とは?	1
1.2	GNU/Linux とは?	2
1.3	Debian GNU/Linux とは?	2
1.4	Debian Installer とは?	3
1.5	Debian の入手	3
1.6	このドキュメントの最新版の入手	3
1.7	この文書の構成	4
1.8	著作権およびソフトウェアライセンスについて	4
2	必要なシステム	6
2.1	サポートするハードウェア	6
2.1.1	サポートするアーキテクチャ	6
2.1.2	3つの異なる ARM 移植版	7
2.1.3	ARM CPU の設計のばらつきとそのサポートの複雑さ	7
2.1.4	Debian/armhf でサポートされるプラットフォーム	7
2.1.5	Debian/armhf によりサポートされなくなったプラットフォーム	9
2.1.6	マルチプロセッサ	9
2.1.7	グラフィックハードウェアのサポート	9
2.1.8	ネットワーク接続機器	10
2.1.9	周辺機器やその他のハードウェア	10
2.2	ファームウェアが必要なデバイス	10
2.3	GNU/Linux に適したハードウェアの購入	10
2.3.1	独占的・閉鎖的なハードウェアを避ける	11
2.4	インストールに利用できるメディア	11
2.4.1	CD-ROM/DVD-ROM/BD-ROM	11
2.4.2	ネットワーク	11
2.4.3	ハードディスク	12
2.4.4	Un*x・GNU システム	12
2.4.5	サポートする記憶装置	12
2.5	必要なメモリとディスクスペース	12
3	Debian GNU/Linux のインストール前に	13
3.1	インストールプロセスの概要	13
3.2	既存データをバックアップしてください!	14
3.3	必要な情報	14
3.3.1	ドキュメント	14
3.3.1.1	インストールマニュアル	14
3.3.1.2	ハードウェアの文書	14
3.3.2	ハードウェア情報の取得先	14
3.3.3	ハードウェア互換性	15
3.3.3.1	ライブシステムを使用したハードウェア互換性テスト	16
3.3.4	ネットワークの設定	16
3.4	必要な最低限のハードウェア	17
3.5	マルチブートシステムでの事前パーティション分割	17
3.6	インストール前に行うハードウェア・OS の設定	18
3.6.1	ARM ファームウェア	18
3.6.2	Debian から提供されている U-Boot (システムファームウェア) イメージ	18
3.6.3	U-Boot でのイーサネット MAC アドレスの設定	18
3.6.4	U-Boot でのカーネル/initrd/デバイスツリーの再配置問題	19

4	システムインストールメディアの入手	20
4.1	公式 Debian GNU/Linux インストールイメージ	20
4.2	Debian ミラーサイトからのファイルのダウンロード	20
4.2.1	どこでインストールファイルを探すか	20
4.2.1.1	Armhf 複数プラットフォーム対応カーネルのインストールファイル	21
4.3	TFTP ネットブート用ファイルの準備	21
4.3.1	RARP サーバの設定	21
4.3.2	DHCP サーバの設定	21
4.3.3	BOOTP サーバの設定	22
4.3.4	TFTP サーバの立ち上げ	22
4.3.5	TFTP イメージを適切な場所に配置する	23
4.4	自動インストール	23
4.4.1	Debian インストーラを用いた自動インストール	23
4.5	インストールファイルの整合性の検証	23
5	インストールシステムの起動	24
5.1	32-bit hard-float ARMv7 でのインストーラの起動	24
5.1.1	ブート用イメージのフォーマット	24
5.1.2	コンソール設定	24
5.1.3	TFTP による起動	24
5.1.3.1	U-Boot での TFTP のブート	24
5.1.3.2	ビルド済み netboot tar アーカイブ	25
5.1.4	U-Boot での USB メモリからの起動	26
5.1.5	インストーラでのビルド済み SD カードイメージの利用	26
5.2	アクセシビリティ	27
5.2.1	インストーラフロントエンド	27
5.2.2	基板デバイス	27
5.2.3	高コントラストテーマ	27
5.2.4	拡大	27
5.2.5	Expert モード、Rescue モード、自動化インストール	27
5.2.6	インストールしたシステムのアクセシビリティ	28
5.3	起動パラメータ	28
5.3.1	ブートコンソール	28
5.3.2	Debian Installer パラメータ	28
5.3.3	起動パラメータで質問に答える	30
5.3.4	カーネルモジュールへパラメータを渡す	31
5.3.5	カーネルモジュールのブラックリスト化	31
5.4	インストールプロセスのトラブルシューティング	32
5.4.1	光学メディアの信頼性	32
5.4.1.1	共通の問題	32
5.4.1.2	調査および問題解決の方法	32
5.4.2	起動設定	33
5.4.3	カーネルの起動時メッセージの意味	33
5.4.4	インストールで発生した問題の報告	34
5.4.5	インストールレポートの送信	34
6	Debian インストーラの使用法	36
6.1	インストーラの動作	36
6.1.1	グラフィカルインストーラの使用法	37
6.2	コンポーネント入門	37
6.3	それぞれのコンポーネントの使用法	38
6.3.1	Debian インストーラのセットアップとハードウェアの設定	38
6.3.1.1	利用可能なメモリのチェック / 低メモリモード	39
6.3.1.2	地域オプションの選択	39
6.3.1.3	キーボード選択	40
6.3.1.4	Debian Installer iso イメージの検索	40
6.3.1.5	ネットワークの設定	40
6.3.1.5.1	自動ネットワーク設定	40
6.3.1.5.2	手動ネットワーク設定	41

6.3.1.5.3	IPv4 と IPv6	41
6.3.2	ユーザとパスワードのセットアップ	41
6.3.2.1	root パスワードの設定	41
6.3.2.2	一般ユーザの作成	42
6.3.3	時計とタイムゾーンの設定	42
6.3.4	パーティションの分割とマウントポイントの選択	42
6.3.4.1	サポートするパーティション分割オプション	42
6.3.4.2	ガイドパーティション分割	43
6.3.4.3	手動パーティション分割	45
6.3.4.4	マルチディスクデバイス (ソフトウェア RAID) の設定	45
6.3.4.5	論理ボリュームマネージャ (LVM) の設定	47
6.3.4.6	暗号化ボリュームの設定	48
6.3.5	基本システムのインストール	50
6.3.6	追加ソフトウェアのインストール	51
6.3.6.1	apt の設定	51
6.3.6.1.1	2 枚以上の CD/DVD イメージでのインストール	51
6.3.6.1.2	ネットワークミラーの利用	52
6.3.6.1.3	ネットワークミラーの選択	52
6.3.6.2	ソフトウェアの選択・インストール	53
6.3.7	システムを起動可能に	54
6.3.7.1	他 OS の検出	54
6.3.7.2	flash-kernel でシステムをブートできるようにする	54
6.3.7.3	ブートローダなしで継続	54
6.3.8	インストールの完了	55
6.3.8.1	システム時計の設定	55
6.3.8.2	システムの再起動	55
6.3.9	トラブルシューティング	55
6.3.9.1	インストールログの保存	55
6.3.9.2	シェルの使用とログの参照	55
6.3.10	ネットワーク越しのインストール	56
6.4	見つからないファームウェアの読み込み	57
6.4.1	メディアの準備	58
6.4.2	ファームウェアとインストールしたシステム	58
6.4.3	Completing the Installed System	59
7	新しい Debian システムを起動してみる	60
7.1	決定的瞬間	60
7.2	暗号化ボリュームのマウント	60
7.2.1	トラブルシューティング	61
7.3	ログイン	61
8	次のステップとそれから	62
8.1	システムをシャットダウンする	62
8.2	Debian に慣れる	62
8.2.1	Debian パッケージングシステム	62
8.2.2	Debian で利用できる追加ソフトウェア	63
8.2.3	アプリケーションのバージョン管理	63
8.2.4	cron ジョブ管理	63
8.3	さらなる文書や情報	63
8.4	電子メールを使用するためのシステム設定	64
8.4.1	デフォルトの電子メール設定	64
8.4.2	システムの外に電子メールを送る	64
8.4.3	Exim4 Mail Transport Agent の設定	65
8.5	新しいカーネルのコンパイル	65
8.6	起動しなくなってしまったシステムの回復	66

A	インストール Howto	67
A.1	前置き	67
A.2	インストーラを起動する	67
A.2.1	光学ディスク	67
A.2.2	ネットワークからの起動	67
A.2.3	ハードディスクからの起動	68
A.3	インストール	68
A.4	インストールレポートを送ってください	69
A.5	そして最後に...	69
B	preseed を利用したインストールの自動化	70
B.1	概要	70
B.1.1	preseed の方法	70
B.1.2	制限	71
B.2	preseed の利用	71
B.2.1	事前設定ファイルの読み込み	71
B.2.2	preseed が質問するブートパラメータの利用	72
B.2.3	自動モード	73
B.2.4	preseed で利用できるエイリアス	74
B.2.5	ブートプロンプトの preseed の例	74
B.2.6	事前設定ファイルを指定するための DHCP の利用方法	75
B.3	事前設定ファイルの作成	75
B.4	事前設定ファイルの内容 (bullseye 用)	76
B.4.1	地域化	76
B.4.2	ネットワーク設定	77
B.4.3	ネットワークコンソール	78
B.4.4	ミラーサイト設定	78
B.4.5	アカウント設定	79
B.4.6	時計と時間帯の設定	80
B.4.7	パーティション分割	80
B.4.7.1	パーティション分割の例	80
B.4.7.2	RAID を用いたパーティション分割	82
B.4.7.3	パーティションマウント法の制御	83
B.4.8	基本システムのインストール	83
B.4.9	apt 設定	84
B.4.10	パッケージ選択	84
B.4.11	インストールの仕上げ	85
B.4.12	他パッケージの preseed	85
B.5	高度なオプション	86
B.5.1	インストール中のカスタムコマンド実行	86
B.5.2	preseed を用いたデフォルト値変更	86
B.5.3	事前設定ファイルのチェーンロード	86
C	Debian でのパーティション分割	88
C.1	Debian のパーティションとそのサイズを決める	88
C.2	ディレクトリツリー	88
C.3	お勧めするパーティションルール	90
C.4	Linux におけるデバイス名	90
C.5	Debian のパーティション分割プログラム	90
D	雑多な事柄	92
D.1	Linux のデバイス	92
D.1.1	マウスのセットアップ	92
D.2	タスクに必要なディスクの空き容量	93
D.3	Unix/Linux システムからの Debian GNU/Linux のインストール	94
D.3.1	はじめに	94
D.3.2	debootstrap のインストール	95
D.3.3	debootstrap の実行	95
D.3.4	基本システムの設定	95

D.3.4.1	デバイスファイルの作成	96
D.3.4.2	パーティションのマウント	96
D.3.4.3	タイムゾーンの設定	97
D.3.4.4	ネットワークの設定	97
D.3.4.5	apt の設定	98
D.3.4.6	ロケールとキーボードの設定	98
D.3.5	カーネルのインストール	99
D.3.6	ブートローダのセットアップ	99
D.3.7	リモートアクセス: SSH のインストールとアクセス方法の設定	99
D.3.8	仕上げに	99
D.4	PPP over Ethernet (PPPoE) を用いた Debian GNU/Linux のインストール	100
E	付記	101
E.1	この文書について	101
E.2	この文書への貢献	101
E.3	多大な貢献	101
E.4	商標表示	102
F	GNU General Public License	103
F.1	Preamble	103
F.2	GNU GENERAL PUBLIC LICENSE	103
F.3	How to Apply These Terms to Your New Programs	106

List of Tables

3	Debian GNU/Linux のインストール前に	
3.1	インストールに役立つハードウェア情報	15
3.2	最低限必要なシステム (推奨値)	17

Abstract

この文書は 32-bit hard-float ARMv7 (「armhf」) アーキテクチャ用 Debian GNU/Linux 11 システム (コードネーム「bullseye」) のインストール説明書です。また、さらに詳しい情報へのポインタや、新しく Debian システムを構築する方法にも言及しています。

日本語訳については、debian-doc@debian.or.jp (要 subscribe) で議論を行っています。また、[Debian JP Project: メーリングリスト](#) に購読に関する簡単な説明があり、[debian-doc Mailing List Archive](#) では過去のメールを読むことができます。

armhf 用 Debian GNU/Linux 11 のインストール

Debian を試していただきありがとうございます。Debian の GNU/Linux ディストリビューションは、他に類を見ないものであることを分かっていただけることでしょう。Debian GNU/Linux は、世界中から質の高い「自由なソフトウェア」をよりすぐり、首尾一貫したディストリビューションとしてまとめあげられています。こうして集められたものは、個々のソフトウェア以上の力を発揮することでしょう。

多くの方は、このマニュアルを読まずに Debian をインストールしたいと思っていることでしょう。また、それが可能なように Debian インストーラは設計されています。インストールガイド全体を読む時間がなければ、インストール Howto (基本的なインストールプロセスをご案内します) と、追加情報やうまくいかないときのための、マニュアルへのリンクを読むことをお勧めします。インストール Howto は、付録A にあります。

そうは言っても、このマニュアルのほとんどを読んでくださることを望んでいますし、読むことでより多くの知識を得られ、よりインストールがうまくいきやすくなるでしょう。

Chapter 1

ようこそ Debian へ

この章では、Debian プロジェクトと Debian GNU/Linux の概略を紹介します。Debian プロジェクトの歴史と Debian GNU/Linux についてすでにご存知でしたら、この章を飛ばして構いません。

1.1 Debian とは？

Debian は、有志が集まってできた団体で、フリーソフトウェアを開発し、フリーソフトウェアコミュニティの理想を推進することを目的としています。Debian プロジェクトは 1993 年に、比較的新しい Linux カーネルをもとにした、完全で一貫性あるディストリビューションの制作のために、Ian Murdock が開発者を広く募ったときに始まりました。献身的なファンたちの比較的小さな団体は、最初 [Free Software Foundation](#) によって支援を受け、[GNU](#) の哲学に影響されていましたが、数年後には 1000 人も Debian 開発者を抱える組織になりました。

Debian 開発者は様々な活動に参加しています。例えば、[Web](#) や [FTP](#) サイトの管理、グラフィックデザイン、ソフトウェアライセンスの法的な分析、文書の執筆、そしてもちろん、ソフトウェアパッケージのメンテナンスです。

私たちの哲学を伝え、Debian が支持する原則を信じている開発者を引き寄せるために、Debian プロジェクトは、私たちの価値の概略を述べ、Debian 開発者であるとはどういうことかという指針とするために、多数の文書を発表しています：

- [Debian 社会契約](#) は、Debian のフリーソフトウェアコミュニティへの関与について述べたものです。この社会契約を守ることに同意する人は、誰でも [メンテナ](#) になることができます。メンテナは誰でも、Debian に新しいソフトウェアを追加することができます—そのソフトウェアが私たちの条件に照らしてフリーであり、パッケージの品質が基準を満たしていれば。
- [Debian フリーソフトウェアガイドライン](#) (DFSG) は、フリーソフトウェアに関する Debian の基準を明確かつ簡潔に述べたものです。この DFSG は、フリーソフトウェア運動において非常に影響力のある文書で、[オープンソースの定義](#) のもととなったものです。
- [Debian ポリシーマニュアル](#) は、Debian プロジェクトの品質基準を詳しく定めたものです。

Debian 開発者は、ほかの多数のプロジェクトにも関与しています。それらのプロジェクトには、Debian 固有のものもあり、Linux コミュニティの一部や全体に関係するものもあります。以下に例を挙げます。

- [Filesystem Hierarchy Standard](#) (FHS) は、Linux のファイルシステムのレイアウトを標準化しようという試みです。これによって、ソフトウェア開発者はパッケージが様々な GNU/Linux ディストリビューションにどのようにインストールされるかを心配することなしに、プログラムのデザインに努力を集中することができます。
- [Debian Jr.](#) は、Debian を若年ユーザに提供できるようなものにするための内部プロジェクトです。

より一般的な情報については、[Debian FAQ](#) を参照して下さい。

1.2 GNU/Linux とは?

Linux はオペレーティングシステム (あなたとコンピュータの間に立ち、他のプログラムを実行させる一連のプログラム) です。

オペレーティングシステムは、様々な基礎的なプログラムを含んでいます。それらによって、コンピュータは、ユーザと通信したり指示を受け取ったり、ハードディスクやテープ、プリンタにデータを読み書きしたり、メモリの使い方を制御したり、他のソフトウェアを実行したりすることが出来ます。オペレーティングシステムの最も重要な部分は、カーネルです。GNU/Linux システムにおいては、Linux がカーネルです。システムの残りの部分は、他のプログラムでできており、その大部分は GNU プロジェクトによって書かれたものです。Linux カーネルだけでは動作するオペレーティングシステムを構成できませんので、多くの人が日常的に「Linux」と呼ぶシステムのことを、私たちは「GNU/Linux」と呼ぶようにしています。

Linux は Unix オペレーティングシステムを手本にしています。当初から、Linux はマルチタスク、マルチユーザシステムとして設計されました。この事実により、Linux は他の有名なオペレーティングシステムに対し、充分差別化できています。しかし、Linux はあなたが想像するよりもさらに異なっています。他のオペレーティングシステムとは対照的に、誰も Linux を所有しません。その開発の多くは無償のボランティアによって行われます。

後に GNU/Linux になるものの開発は 1984 年、[フリーソフトウェア財団](#) が GNU という Unix ライクなオペレーティングシステムの開発を始めたときに始まりました。

[GNU プロジェクト](#) は、Unix™ や、Linux などの Unix ライクなオペレーティングシステムと共に使うための一連のフリーソフトウェアツールを開発してきました。これらのツールは、ファイルのコピー・削除といった日常的な作業から、プログラムの作成・コンパイルや様々なドキュメントフォーマットの高度な編集といった作業までを可能にします。

多くのグループや個人が Linux に寄与する中で、最大の単独貢献者はいまだに (Linux の中で使用されるほとんどのツールだけでなく哲学も作成した) [フリーソフトウェア財団](#) と、Linux を可能にしたコミュニティです。

[Linux カーネル](#) は、Linus Torvalds というフィンランド人の計算機科学の学生が 1991 年に、Usenet の comp.os.minix ニュースグループに Minix の代替カーネルの初期バージョンを公表したのが始まりです。Linux International の [Linux 史のページ](#) 参照して下さい。

Linus Torvalds は、何人ものサブシステムのメンテナの協力を得て、数百人の開発者の作業を調整し続けています。Linux カーネルの [公式ウェブサイト](#) があります。linux-kernel メーリングリストの情報は、[linux-kernel メーリングリスト FAQ](#) で読むことができます。

Linux ユーザは、それらのソフトウェアの大きな選択の自由を持っています。例えば、Linux ユーザは、1 ダースの異なるコマンドラインシェルや数種のグラフィカルデスクトップの中から選ぶことができます。この選択できるということが、しばしばコマンドラインやデスクトップを変更できるという考えに慣れていない、他のオペレーティングシステムのユーザを当惑させています。

Linux はまた、ほとんどクラッシュせず、複数のプログラムを同時に実行するのに優秀で、多くのオペレーティングシステムより安全です。これらの利点により、Linux はサーバ市場で最も急成長しているオペレーティングシステムです。さらに最近、Linux は、ホーム・ビジネスユーザにも人気が出始めました。

1.3 Debian GNU/Linux とは?

Debian の哲学や方法論と、GNU ツール・Linux カーネル・その他の重要なフリーソフトウェアとを組み合わせることで、Debian GNU/Linux と呼ばれるユニークなディストリビューションが形成されています。このディストリビューションは、多数のソフトウェア パッケージから構成されています。ディストリビューションに含まれる個々のパッケージは、実行ファイル・スクリプト・ドキュメント・設定情報などから構成されています。また各パッケージには、そのパッケージに責任を持つメンテナがいて、そのパッケージを最新に保ち、バグ報告を追跡し、パッケージにされているソフトウェアの上流開発者と連絡をとることについて、第一に責任を負います。大きなユーザベースが、バグ追跡システムとあいまって、問題がすぐに発見・解決されることを保証しています。

Debian は、細部に注意を払うことで、高品質で安定したスケーラブルなディストリビューションとなっています。小さなファイアウォールから科学用途のデスクトップワークステーションやハイエンドネットワークサーバまで、様々な用途に合わせたインストールが可能です。

Debian は、技術的な優越性や Linux コミュニティのニーズや期待への深いコミットメントによって、熟練したユーザに特に人気があります。Debian はさらに、現在 Linux が普通に持っている多くの特徴を導入しました。

例えば、Debian はソフトウェアの簡単なインストール・削除用にパッケージ管理システムを持った初めての Linux ディストリビューションでした。さらに、再インストールせずにシステムの更新ができる、初めての Linux ディストリビューションでした。

Debian は Linux 開発のリーダーであり続けています。その開発プロセスは (完全なオペレーティングシステムを構築し維持するような非常に複雑なタスクであったとしても) オープンソース開発モデルが、どれほどうまくいくことができるかの好例となっています。

Debian を他の GNU/Linux ディストリビューションと区別する最大の特徴は、パッケージ管理システムです。Debian システムの管理者は、システムにインストールされるパッケージに関して、ひとつのパッケージのインストールからオペレーティングシステム全体の自動アップデートまで、完全に制御することができます。個々のパッケージをアップデートしないように設定することもできます。あなた自身がコンパイルしたソフトウェアについて、その依存関係を設定することもできます。

「トロイの木馬」や他の悪意あるソフトウェアからあなたのシステムを守るために、Debian のサーバは、アップロードされてきたパッケージが登録された Debian 開発者からのものかどうかを確かめます。また、Debian の各パッケージはより安全な設定となるように細心の注意が払われています。もしリリースされたパッケージにセキュリティ上の問題が発生すれば、その修正版は通常すぐに利用可能になります。Debian の簡単なアップデートオプションによって、セキュリティ修正はインターネットを通じて自動的にダウンロード・インストールすることができます。

あなたの Debian GNU/Linux システムについてサポートを受けたり、Debian の開発者たちと連絡したりする第一の、そして最良の方法は、Debian プロジェクトが運営する多数のメーリングリストを用いることです (この文章の執筆時点で 322 以上のメーリングリストがあります)。メーリングリストを簡単に講読するためには、[Debian メーリングリスト講読ページ](#) を訪れて、フォームに必要事項を記入するとよいです。

1.4 Debian Installer とは？

Debian インストーラ (「d-i」としても知られています) は基本的な動作を行う Debian システムをインストールするためのソフトウェアシステムです。組込みシステム・ラップトップ・デスクトップ・サーバマシンのような幅広いハードウェアをサポートしており、様々な目的に使われる膨大な量のフリーソフトウェアを提供します。

インストール作業は簡単な質問群に答えることで進みます。インストール作業での全設定をコントロールすることができるエキスパート・モードや、自動インストールを実行する拡張機能も提供されています。インストールしたシステムはそのまま使うことも、さらにカスタマイズすることもできます。インストールは多数のソースから実行できます: USB、CD/DVD/Blu-Ray、そしてネットワーク経由です。インストーラは、80 以上の言語で翻訳されたインストール画面をサポートしています。

インストーラは boot-floppies プロジェクトを起源としており、これは Joey Hess によって 2000 年に初めて言及されています。以来インストールシステムは継続してボランティアらによって開発されており、改善と機能追加が行われています。

[Debian インストーラのページ](#)、[Wiki](#)、[debian-boot メーリングリスト](#) などで、より詳細な情報を確認できます。

1.5 Debian の入手

インターネットを通じて Debian GNU/Linux をダウンロードしたり Debian の公式インストールメディアを購入したりするための情報については、[入手方法についてのページ](#) を参照して下さい。[Debian のミラー一覧](#) には、Debian の公式ミラーサイトがすべて載っていますので、もっとも近いサイトを簡単に探すことができます。

Debian は、インストール後に非常に簡単にアップグレードできます。このインストール手順では、システムの設定についてお助けします。一度インストールが済んでしまえば、必要に応じてこのようなアップグレードを行えるようになります。

1.6 このドキュメントの最新版の入手

この文書には絶えず変更が加えられています。Debian GNU/Linux システムの 11 リリースに関する最新情報については、[Debian 11 ページ](#) にて確認してください。このインストールマニュアルの最新版は、[公式インストールマニュアルページ](#) から利用できます。

1.7 この文書の構成

この文書は、初めて Debian をお使いになるユーザのために書かれたマニュアルです。お手持ちのハードウェアの動作に関しては一般的な知識があることを前提としていますが、なるべく専門的な知識がなくてもお読みいただけるよう心がけています。

また熟練したユーザであっても、この文書で、最低限インストールに必要な容量や、Debian インストールシステムでサポートされるハードウェアの詳細など、参考になる情報を得ることができるでしょう。熟練したユーザの方には、この文書のあちこちをかいつまんでお読みになることをお勧めします。

基本的にこの文書は、実際に体験するインストールのプロセスに沿って、順々に説明するように構成されています。Debian GNU/Linux のインストールの各作業段階と、それに関連するこの文書の各節は以下の通りになっています。

1. 第2章では、お手持ちのハードウェアがインストーラのシステム要件を満たしているかどうかを調べます。
2. 第3章では、既存のシステムをバックアップし、Debian のインストールに先だつシステム設計やハードウェアの設定を行います。もしマルチブートシステムを考えているのであれば、ハードディスク上に、Debian 用パーティションを作るための空き領域を作っておく必要があるかもしれません。
3. 第4章では、あなたのインストール方法のためのインストールファイルを入手します。
4. 次の第5章では、インストーラを起動します。またこの章では、起動に問題があった際のトラブルシューティングの手順についても紹介します。
5. 第6章に従って実際のインストールを実行してください。ここでは言語選択、周辺機器のドライバモジュールの設定、(CD/DVD インストールイメージセットからインストールしていない場合) 残りのインストールするファイルを Debian サーバから直接取得するようなネットワーク接続の設定、ハードディスクのパーティション分割、基本システムのインストールを行います。その後、インストールするタスクの選択を行います。(Debian システムのパーティションセットアップについては、付録Cで背景を説明しています)
6. 第7章では、新しくインストールした基本システムを起動します。

システムのインストールが終了したら、第8章を読んで下さい。この章では、Unix や Debian に関する情報の探し方や、カーネルの切り換えの方法を説明します。

最後に、付録Eには、この文書に関する情報や貢献の方法が載っています。

1.8 著作権およびソフトウェアライセンスについて

この文書を読んでいる方は、多数の商用ソフトウェアにあるようなライセンス (購入したソフトウェアのコピー 1 部を、1 台のコンピュータで使用できる) はご存知のことでしょう。しかし、このシステムはそのようなものとは違います。私たちは、あなたの通っている学校や仕事場にあるすべてのコンピュータに Debian GNU/Linux をインストールすることを勧めます。また、友達に貸して、彼らのコンピュータにインストールするのを手伝ってあげましょう。さらには、わずかな制限にさえ気がつけば、何千部ものコピーを作って売ることも可能です。なぜなら、Debian はフリーソフトウェアに基づいているからです。

フリーソフトウェアとは、著作権を持っていないという意味ではありません。また、このソフトウェアを含むインストールメディアが、無償で配布されなければならないという意味でもありません。フリーソフトウェアとは、ひとつには、個々のプログラムのライセンスにおいて、プログラムの利用や再配付の権利に、お金を払う必要がないことを意味しています。また誰でも、そのソフトウェアを拡張したり、改造したり、修正すること、さらにその成果を再配付することが可能であることも意味しています。

注意

Debian プロジェクトでは、ユーザの実用性に関する妥協から、私たちのフリーの基準に適合しないパッケージも利用できるようになってきました。このパッケージは公式なディストリビューションの一部ではありませんが、Debian ミラーの contrib・non-free エリア、またはサードパーティ製 CD/DVD-ROM から入手できます。このレイアウトや、アーカイブの内容については、[Debian FAQ](#) にある「Debian FTP アーカイブ」の節をご覧ください。

このシステムに入っているプログラムの多くは、「GPL」と略される GNU General Public License にしたがって利用許諾されています。この GPL は、プログラムのコピーを配布するときには、必ずプログラムのソースコードを利用可能にしておくことを要求しています。これは、ユーザがそのソフトウェアを変更できることを保証するものです。そのため、私たちは、Debian システムに含まれる GPL 準拠のプログラムのソースコード¹をすべて収録しています。

Debian に収録されたプログラムの著作権やソフトウェアライセンスの形式には、他にも数種あります。それぞれのプログラムの著作権やライセンスは、一度システムをインストールすれば、`/usr/share/doc/パッケージ名/copyright` ファイルを探せば見つけることができます。

ライセンスや、Debian が main ディストリビューションにソフトウェアを収録する際に用いているフリーの基準に関してより詳細な情報をお求めの場合は、[Debian フリーソフトウェアガイドライン](#)をご覧ください。

最も重要な法律上の注意点は、このソフトウェアが無保証であることです。これは、このソフトウェアを作成したプログラマらがコミュニティの利益を考えてのことです。ソフトウェアは、いかなる目的への利用に対しても保証されていません。しかし、ソフトウェアがフリーであるゆえに、ユーザには必要に応じてソフトウェアを修正する権限が与えられます。また、このようにしてソフトウェアの拡張が誰かによってなされれば、その利益も享受できます。

¹Debian ソースパッケージの探し方や展開の仕方やバイナリの作成方法に関する情報については、[Debian FAQ](#) の「Debian パッケージ管理システムの基本」をご覧ください。

Chapter 2

必要なシステム

この節では、Debian を始めるために必要なハードウェアに関する情報を扱います。また、GNU や Linux でサポートされるハードウェアに関するより詳しい情報へのリンクも用意しました。

2.1 サポートするハードウェア

Debian は、Linux・kFreeBSD カーネルや GNU ツールセットが必要とする以上のハードウェアを要求しません。それゆえ、Linux・kFreeBSD カーネル、libc、gcc などが移植されていて、Debian の移植版が存在すれば、どんなアーキテクチャやプラットフォームでも Debian を動作させることができます。すでに Debian GNU/Linux でテストされている 32-bit hard-float ARMv7 アーキテクチャシステムの詳細は、<https://www.debian.org/ports/arm/> にある移植版のページを参照してください。

この節では、32-bit hard-float ARMv7 でサポートされるハードウェアの様々な設定のすべてに触れることは避け、一般的な情報とさらなる情報が見つけられる場所へのポインタを紹介します。

2.1.1 サポートするアーキテクチャ

Debian GNU/Linux 11 は 9 の主要なアーキテクチャと、「フレーバー」と呼ばれる各アーキテクチャのバリエーションをサポートしています。

アーキテクチャ	Debian での名称	サブアーキテクチャ	フレーバー
AMD64 & Intel 64	amd64		
Intel x86 ベース	i386	デフォルトの x86 マシン	デフォルト
		Xen PV ドメインのみ	xen
ARM	armel	Marvell Kirkwood 及び Orion	marvell
ハードウェア FPU がある ARM	armhf	複数プラットフォーム対応	armmp
64bit ARM	arm64		
64bit MIPS (リトルエンディアン)	mips64el	MIPS Malta	5kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
32bit MIPS (リトルエンディアン)	mipsel	MIPS Malta	4kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
Power Systems	ppc64el	IBM POWER8 以降のマシン	
64bit IBM S/390	s390x	VM-reader や DASD からの IPL	generic

この文書は Linux カーネルを用いた 32-bit hard-float ARMv7 アーキテクチャへのインストールを扱います。Debian がサポートしている他のアーキテクチャに関する情報を探しているなら、[Debian 移植版](#) のページをご覧ください。

2.1.2 3つの異なる ARM 移植版

ARM アーキテクチャは時間とともに進化し、今の ARM プロセッサでは旧型では利用できなかった機能を提供しています。Debian ではそのため、幅広い種類のマシンを可能な限りサポートできるように 3つの ARM 移植版を提供しています:

- Debian/armel はハードウェア浮動小数点演算ユニット (FPU) をサポートしない、古い 32 ビットの ARM プロセッサを対象としています。
- Debian/armhf は最低でも ARMv7 アーキテクチャに ARM ベクトル浮動小数点演算仕様のバージョン 3 (VFPv3) を実装した新しい 32 ビットの ARM プロセッサでのみ動作します。Debian/armhf ではそういったモデルで拡張された機能や性能拡張を利用するようになっていきます。
- Debian/arm64 は ARMv8 以降のアーキテクチャを実装する 64 ビットの ARM プロセッサで動作します。

現在利用できる CPU は全て、技術的に (ビッグ、リトルの) どちらのエンディアンモードでも動作します。しかし、大多数のシステム実装では、リトルエンディアンモードを使用します。Debian/arm64 や Debian/armhf、Debian/armel はどれもリトルエンディアン ARM システムのみサポートします。

2.1.3 ARM CPU の設計のばらつきとそのサポートの複雑さ

ARM システムは i386/amd64 ベースの PC アーキテクチャと比べてかなり異質なので、サポート状況ははるかに複雑になります。

ARM アーキテクチャはいわゆる「systems-on-chip」(SoC) で主に利用されています。こういった SoC は多くの様々な企業により設計され、そのハードウェアの構成要素はシステムの起動に必要なような非常に基本的な機能までが非常に様々です。システムファームウェアのインターフェイスは時とともに標準化が進みましたが、特に古いハードウェアではファームウェアやブート用インターフェイスは非常に様々であるため、PC の世界ではマザーボードの BIOS/UEFI が扱うようなシステム固有の低レベルの問題についても、そういったシステムでは Linux カーネルが担当しないといけなく、ということになります。

Linux カーネルでの ARM サポート初期に、ハードウェアの種類が多いことから PC システムでの「1 つで万能」のカーネルとは対照的に、ARM システムごとに別個のカーネルが必要となりました。この方法は大量の異なるシステムに対応させるには向かないため、異なる ARM システムでも動作する単一の ARM カーネルをブートできるようにする作業が行われました。より新しい ARM システムのサポートはこういった複数プラットフォーム対応カーネルを利用できるようにする方法で実装されるようになりましたが、古い複数のシステムでは専用のカーネルが必要なままです。そのため、標準の Debian ディストリビューションではそういった古い ARM システムについては特定のものへのインストールだけをサポートします。新しいシステムは Debian/armhf で、(「armmp」と呼ばれる) 複数プラットフォーム対応カーネルによりサポートされます。

2.1.4 Debian/armhf でサポートされるプラットフォーム

以下のシステムは複数プラットフォーム対応 (armmp) カーネルを使って Debian/armhf が動作することがわかっています:

フリースケール MX53 クイック・スタート・ボード (MX53 LOCO Board) IMX53QSB は i.MX53 SoC ベースの開発ボードです。

Versatile Express Versatile Express は様々な CPU ドーターボードを搭載できるベースボードで構成される ARM の開発ボードシリーズです。

Allwinner sunXi ベースの一部の開発ボード及び組み込みシステム armmp カーネルは Allwinner A10 (アーキテクチャコード名「sun4i」) や A10s/A13 (アーキテクチャコード名「sun5i」)、A20 (アーキテクチャコード名「sun7i」)、A31/A31s (アーキテクチャコード名「sun6i」)、A23/A33 (「sun8i」ファミリの一部) SoC を基にした複数の開発用ボードや組み込みシステムをサポートします。現在、(既存の SD カード用インストーライメージでの提供も含めて) 以下の sunXi ベースのシステムでインストーラの完全サポートが利用できます:

- Cubietech Cubieboard 1 + 2 / Cubietruck

- LeMaker Banana Pi 及び Banana Pro
- LinkSprite pcDuino 及び pcDuino3
- Olimex A10-Olinuxino-LIME / A20-Olinuxino-LIME / A20-Olinuxino-LIME2 / A20-Olinuxino Micro / A20-SOM-EVB
- Xunlong OrangePi Plus

Allwinner sunXi ベースの機器のシステムサポートは主流側 Linux カーネルで利用できるドライバやデバイスツリーの情報に限られています。(Allwinner SDK カーネル等) ベンダー固有のカーネルツリーや android 派生の linux-sunxi.org カーネルの 3.4 系列は Debian ではサポートしていません。

主流側 Linux カーネルは全般に Allwinner A10、A10s/A13、A20、A23/A33、A31/A31s SoC のシリアルコンソール、イーサネット、SATA、USB、MMC/SD カードをサポートします。ローカルディスプレイ (HDMI/VGA/LCD) とオーディオハードウェアのサポートレベルは個々のシステムで差があります。ほとんどのシステムでカーネルはグラフィックドライバをネイティブには持たず、ブートローダがディスプレイを初期化する「simplefb」プラットフォームを使ってカーネルはただその初期化済みフレームバッファを再利用するだけです。これは通常うまく動作しますが特定の制限があります (ディスプレイ解像度をすぐその場で変更することとディスプレイの電源管理ができません)。

sunXi ベースのシステムの大容量ストレージデバイスに利用されているオンボードのフラッシュメモリは大きく分けてそのままの NAND フラッシュと eMMC フラッシュの 2 種類が存在します。オンボードフラッシュストレージを搭載する古い sunXi ベースのボードでは多くが、主流側カーネルでは通常サポートされないそのままの NAND フラッシュを使っているため Debian でも使えません。新しいシステムではそのままの NAND フラッシュではなく eMMC フラッシュを使っているものが出てきています。eMMC フラッシュチップは基本的に高速な取り替えの出来ない SD カードであり、通常の SD カードと同様にサポートされています。

上記の一覧になくても基礎的なサポートをインストーラに収録している sunXi ベースのシステムはいくらかありますが、対応するハードウェアに Debian プロジェクトでアクセスする方法がないため当該システムの大部分はテストされていません。こういったシステム向けに提供されているインストーラ搭載のビルド済み SD カードイメージはありません。サポートがそのように限定されている開発用ボードには

- Olimex A10s-Olinuxino Micro / A13-Olinuxino / A13-Olinuxino Micro
- Sinovoip BPI-M2 (A31s ベース)
- Xunlong Orange Pi (A20 ベース) / Orange Pi Mini (A20 ベース)

があります。上記のシステム一覧にある SoC に加え、インストーラでは Allwinner H3 SoC 及びそれを基にしたボードをかなり限定的にサポートしています。主流側カーネルでの H3 サポートは Debian 9 リリースのリリース時点では相当の作業が残っているため、インストーラでは H3 ベースのシステムではシリアルコンソール、MMC/SD、USB ホストコントローラだけをサポートしています。H3 のオンボードイーサネットポート向けドライバはまだ無いため、ネットワーク接続は USB 接続のイーサネットアダプターか USB 接続の WiFi 経由でのみ可能です。こういったかなり基礎的なインストーラサポートを利用できる H3 ベースのシステムには以下があります。

- FriendlyARM NanoPi NEO
- Xunlong Orange Pi Lite / Orange Pi One / Orange Pi PC / Orange Pi PC Plus / Orange Pi Plus / Orange Pi Plus 2E / Orange Pi 2

NVIDIA Jetson TK1 NVIDIA Jetson TK1 は (Tegra 124 としても知られる) Tegra K1 チップを基にした開発用ボードです。Tegra K1 は 4 コアの 32 ビット ARM Cortex-A15 CPU と 192 CUDA コアの Kepler GPU (GK20A) を搭載しています。Tegra 124 を基にした他のシステムでも使えるかもしれません。

Seagate Personal Cloud 及び Seagate NAS Seagate Personal Cloud 及び Seagate NAS は Marvell Armada 370 プラットフォームを基にした NAS 機器です。Debian では Personal Cloud (SRN21C)、Personal Cloud 2-Bay (SRN22C)、Seagate NAS 2-Bay (SRPD20)、Seagate NAS 4-Bay (SRPD40) をサポートしています。

SolidRun Cubox-i2eX / Cubox-i4Pro Cubox-i シリーズはフリースケール i.MX6 SoC ファミリを基にした小型、立方体型のシステム群です。Cubox-i シリーズのシステムサポートは主流側の Linux カーネルで利用できるドライバやデバイスツリーの情報に限られています。Cubox-i 向けのフリースケール 3.0 シリーズのカーネルは Debian ではサポートしていません。主流側のカーネルで利用できるドライバではシリアルコンソールやイーサネット、USB、MMC/SD カード、HDMI 経由のディスプレイを (コンソールと X11) でサポートしています。以上に加え、Cubox-i4Pro の eSATA ポートもサポートしています。

Wandboard Wandboard Quad、Dual、Solo は Freescale i.MX6 Quad SoC ベースの開発用ボードです。システムサポートは主流側カーネルで利用できるドライバ及びデバイスツリーの情報に限定されています。wandboard.org で配布している wandboard 特有の 3.0 及び 3.10 カーネル系列は Debian ではサポートしていません。主流側カーネルにはシリアルコンソール、HDMI 経由のディスプレイ (コンソールと X11)、イーサネット、USB、MMC/SD、SATA (Quad のみ)、アナログオーディオのドライバサポートが収録されています。他のオーディオオプション (S/PDIF、HDMI オーディオ) やオンボードの WLAN/Bluetooth モジュールのサポートは Debian 9 ではテストされていないか利用できるものはありません。

Linux カーネルの ARM の複数プラットフォーム対応では通常 debian-installer で利用されるカーネルで対象システムの構成要素をサポートして対象システム向けのデバイスツリーファイルが利用可能である限りは、上に明示的に列挙されていない armhf システムで debian-installer を実行できます。その場合、インストーラは機能するようには通常できますが、システムを自動でブート可能にすることはできないかもしれません。それには多くの場合デバイス特有の情報が必要となります。

そういったシステムで debian-installer を利用する場合、インストールの最後に例えば debian-installer から起動したシェルで必要となるコマンドを実行し、手作業によりシステムをブート可能にする必要があるかもしれません。

2.1.5 Debian/armhf によりサポートされなくなったプラットフォーム

EfikaMX EfikaMX プラットフォーム (Genesi Efika Smartbook 及び Genesi EfikaMX nettop) は Debian 7 でプラットフォーム特有のカーネルによりサポートされましたが、Debian 8 以降ではサポートされません。以前に利用されていたプラットフォーム特有のカーネルをビルドするのに必要なコードが上流の Linux カーネルソースから 2012 年に削除されたため、Debian で新しいビルドを提供することはできません。EfikaMX で armmp 複数プラットフォーム対応カーネルを利用するにはデバイスツリーでサポートしている必要がありますが、現在利用できるようにはなっていません。

2.1.6 マルチプロセッサ

このアーキテクチャでは、マルチプロセッササポート (「対称型マルチプロセッシング」や SMP と呼ばれている) が利用できます。Debian 11 の標準カーネルイメージは SMP-alternatives をサポートするようにコンパイルされています。これにより、プロセッサ数 (やプロセッサコア数) を検出し、単一プロセッサシステムの場合には、自動的に SMP を無効にします。

もともと、複数のプロセッサがあるコンピュータはハイエンドサーバシステムのみのものでしたが、近年では「マルチコア」と呼ばれるプロセッサの登場により、どこでも当たり前のものになりました。これには、1 つの物理的なチップに、「コア」と呼ばれる複数のプロセッサユニットが搭載されています。

2.1.7 グラフィックハードウェアのサポート

Debian のグラフィカルインターフェースのサポートは、X.Org による X11 システムやカーネルでサポートされているかどうかで決まります。デスクトップ環境は X11 を利用するのに対し、基本的なフレームバッファのグラフィックはカーネルにより提供されます。3D ハードウェアアクセラレーションやハードウェアアクセラレーションビデオといった、高性能なグラフィックカードの機能が有効かどうかは、システムで使用する実際のグラフィックハードウェアと、ある状況下では、追加「ファームウェア」イメージのインストール (項 2.2 参照) に依存します。

ほぼ全ての ARM マシンがグラフィックハードウェアを、プラグインカードに依らず組み込みで持っています。グラフィックカードを増設できる拡張スロットを備えるマシンはありますが、それは希少です。グラフィックを一切持たないヘッドレス設計のハードウェアがかなり一般的です。カーネル

により提供される基本的なフレームバッファのビデオはグラフィックを備えた機器であればどれも使えるはずですが、高速 3D グラフィックを使うためにはバイナリドライバが必ず必要です。状況は刻々と変わりますが、bullseye のリリース時点では nouveau (Nvidia Tegra K1 の SoC) 及び freedreno (Qualcomm Snapdragon の SoC) 用のフリードライバがこのリリースで利用できるようになっています。他のハードウェアではサードパーティの non-free ドライバが必要です。

サポートされているグラフィックハードウェアやポインティングデバイスに関する、より詳細な情報は <https://wiki.freedesktop.org/xorg/> にあります。Debian 11 は X.Org バージョン 7.7 を採用しています。

2.1.8 ネットワーク接続機器

Linux カーネルがサポートしているネットワークインターフェースカード (NIC) なら、インストールシステムでもほとんどサポートしています。ドライバモジュールは、通常自動的に読み込まれます。

32-bit hard-float ARMv7 では、ほとんどの内蔵イーサネットデバイスをサポートしており、追加 PCI デバイスや、USB デバイスのモジュールを提供しています。

2.1.9 周辺機器やその他のハードウェア

Linux は、マウス、プリンタ、スキャナ、PCMCIA/CardBus/ExpressCard、USB デバイスなどの様々なハードウェアに幅広く対応しています。しかし、システムのインストールに、これらのデバイスが必要なのではありません。

2.2 ファームウェアが必要なデバイス

Besides the availability of a device driver, some hardware also requires so-called firmware or microcode to be loaded into the device before it can become operational. This is most common for network interface cards (especially wireless NICs), but for example some USB devices and even some hard disk controllers also require firmware.

With many graphics cards, basic functionality is available without additional firmware, but the use of advanced features requires an appropriate firmware file to be installed in the system. In some cases, a successful installation can still end up in a black screen or garbled display when rebooting into the installed system. If that happens, some workarounds can be tried to log in anyway (see 項6.4.3).

動作にファームウェアを必要とする古いデバイスでは、ファームウェアファイルは、メーカーによってデバイスの EEPROM/フラッシュチップに永続的に置かれていました。今日では、ほとんどの新しいデバイスがこの方法でファームウェアを埋め込まなくなり、ホストの OS がシステムブート時に、毎回ファームウェアファイルを、デバイスにアップロードしなければならなくなっています。

多くの場合、Debian GNU/Linux プロジェクトで使用する基準において、non-free (非フリー) であるため、main ディストリビューションや、インストールシステムに含むことができません。デバイスドライバそのものがディストリビューションに含まれ、Debian GNU/Linux が法的にファームウェアを配布できるのであれば、アーカイブの non-free セクションに独立したパッケージとして、利用できることがしばしばあります。

しかし、そのようなハードウェアを、インストール中に使用できないわけではありません。Debian GNU/Linux 5.0 からは、debian-installer は USB メモリなどのリムーバブルメディアから、ファームウェアを含む、ファームウェアファイルやパッケージの読み込みをサポートしています。インストール中に、ファームウェアファイルやパッケージをどのように利用するのか、といった詳細情報は、項6.4 を参照してください。

debian-installer がファームウェアファイルを要求し、そのファームウェアファイルがない、または非フリーのファームウェアファイルをシステムにインストールしたくないといった場合は、ファームウェアをロードせずに継続を試せます。特定の状況下で必要になるはずなので、ドライバが追加ファームウェアを要求する場合がありますが、ほとんどのシステムで、デバイスはファームウェアがなくても動作します (例: tg3 ドライバを使用する特定のネットワークカードで発生)。

2.3 GNU/Linux に適したハードウェアの購入

Debian や他の GNU/Linux ディストリビューションを **プリインストール** したシステムを出荷しているベンダもあります。多少余分なお金がかかるかもしれませんが、ある程度の安心を購入できること

になります。このハードウェアは GNU/Linux でしっかりサポートされていることが確信できるわけですから。

Linux がバンドルされたシステムを購入する場合でも、中古のシステムを購入する場合でも、そのハードウェアが Linux カーネルでサポートされているか改めて確認することが重要です。前述の参考資料の中に、そのハードウェアが挙げられているかどうかを確認してください。(もしあれば) 購入先の販売員には、Linux システムを購入することを伝えましょう。また、Linux に友好的なハードウェアベンダを支援しましょう。

2.3.1 独占的・閉鎖的なハードウェアを避ける

あるハードウェアメーカーは、どのようにドライバを書いたらよいかをまったく教えてくれません。また、フリーソフトウェアの中心要素のひとつである、ドライバのソースコード公開を妨げる NDA (非公開の同意) を結ばない限り、ドキュメントを見せてくれないメーカーもあります。そういったデバイスの、有用なドキュメントへのアクセス権がないため、Linux では、単に動作しないという事になります。

多くの場合、そんなデバイスと、OS や、そのデバイスドライバがどのように通信するのかを説明した、標準(または少なくともデファクトスタンダード)があります。そのような標準(あるいはデファクトスタンダード)に従うすべてのデバイスは、ひとつの汎用デバイスドライバで動作し、デバイス固有のドライバは必要ありません。ある種のハードウェア(例: USB 「ヒューマンインターフェースデバイス」、つまりキーボードやマウスなどや、USB フラッシュディスク、メモリーカードリーダーのような USB マスストレージデバイス)では非常にうまく動作し、実際に市場に流通しているデバイスはすべて標準に準拠しています。

他の分野では、たとえばプリンタは、残念ながらそうではありません。多くのプリンタが、標準(またはデファクトスタンダード)制御言語で対処し、いくつもの OS で問題なく動作できるようにしている一方、少数ですが、ドキュメントがないプロプライエタリな制御コマンドしか理解せず、自由な OS では使用できないか、メーカーが提供したクローズソースドライバしか使用できないプリンタがあります。

デバイス購入時にそのハードウェア用のクローズソースドライバがベンダーにより提供されていても、そのデバイスの現実的な寿命はドライバが利用できるかどうかで制限されます。最近では製品サイクルが短くなり、消費者向けデバイスが生産終了となってから短時間で生産者によるドライバ更新が利用できなくなることは珍しくありません。システムのアップグレード後にクローズソースドライバが動作しなくなってしまうと、完動していたデバイスがドライバのサポートが行われなくなる理由により使えないものになり、その場合にできることは何もありません。そういうことがあるため、閉鎖的なハードウェアの購入はそれを利用する OS を問わず初めから避けるべきです。

私たちがそのハードウェア向けのフリードライバを提供するために必要な、ドキュメントその他の資料を公開するように閉鎖的なハードウェアの生産者に働きかけることにより、この状況の改善を支援することができます。

2.4 インストールに利用できるメディア

本節は、Debian をインストールするのに、どのメディアを使用するかを決めるのに、参考になるでしょう。全体をメディアに費やした章(第4章)があり、そこではメディアごとに利点と欠点を挙げています。その章から、このページに、もう一度戻ってくるかもしれませんね。

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

光学ディスクからのインストールは、ほとんどのアーキテクチャでサポートされています。

2.4.2 ネットワーク

インストールに必要なファイルをインストール中に取得するのに、ネットワークを使用できます。ネットワークを使用するかどうかは、あなたが選択したインストール方法と、インストール中の質問への答に依存します。インストールシステムは、ネットワークへのほとんどの接続法(PPPoEを含む。ISDN や PPP は不可)上での、HTTP と FTP のどちらもサポートしています。インストール完了後に、ISDN や PPP を使用するようにシステムの設定ができます。

CD/DVD や USB メモリ等のローカルメディアを一切必要とせず、インストールシステムをネットワーク越しに起動することもできます。ネットワーク越しに起動するための基盤が既にある(つまり、

ネットワーク内で既に DHCP および TFTP サービスが動作している) 場合は、そうすることで大量のマシンへの展開が簡単、迅速にできるようになります。必要となる基盤の準備には、ある程度技術的な経験が要求されるため、この方法は初心者には勧められるものではありません。

ネットワーク越しに起動を行い、すべてのローカルファイルシステムを NFS でマウントして、ディスクをインストールすることも一つの選択です。

2.4.3 ハードディスク

ハードディスクからインストールシステムを直接ブートするのは、多くのアーキテクチャで使えるもうひとつの方法です。ハードディスク上にあるインストーラをロードするため、他の OS が必要になります。この方法は、他のインストール方法が利用できないという、特殊な場合にだけ使用してください。

2.4.4 Un*x · GNU システム

他の Unix 系システムが稼働していれば、このマニュアルで説明している `debian-installer` を使用せずに、Debian GNU/Linux をインストールに使用できます。このインストール方法なら、他の方法ではサポートしないハードウェアや、ダウンタイムを用意できないユーザにとって便利です。この方法に興味があれば、[項D.3](#) をスキップしてください。このインストール方法は、他にインストール方法のない、慣れたユーザにとってのみ、おすすめします。

2.4.5 サポートする記憶装置

Debian インストーラのカーネルは、なるべくどのシステムでも実行できるように構築されています。

2.5 必要なメモリとディスクスペース

通常のインストールを行うには、少なくとも 190MB の RAM と 920MB のハードディスク領域が必要です。これは、本当に最小限の値だということに注意してください。現実的な値は、[項3.4](#) をご覧ください。

インストーラは通常自動でメモリ節約トリックを有効にしてそのような低メモリシステム上でも動作しますが、あまりテストが行われていないアーキテクチャではそれが働かないかもしれません。ただそれでも手動で `lowmem=1` や `lowmem=2` というブートパラメータを追加することで有効にできます ([項6.3.1.1](#) と [項5.3.2](#) もご覧ください)。

メモリやディスク領域が少ないシステムへのインストールも可能ですが、経験を積んだユーザにのみお勧めします。

Chapter 3

Debian GNU/Linux のインストール前に

本章は、インストーラを起動する前の、Debian をインストールする準備について扱います。ここでは、データのバックアップ、ハードウェアに関する情報収集、必要な情報の特定といったことを含みます。

3.1 インストールプロセスの概要

はじめに、再インストールについて述べておきます。Debian で、システムの完全な再インストールが必要になる状況は、非常にまれです。おそらく、もっともありそうなケースはハードディスクの機械的な故障でしょう。

多くの普通のオペレーティングシステムが、重大な故障が起きたり、OS の新バージョンへのアップグレードの際に、完全インストールを要求するかもしれません。完全な新インストールを要求しなくても、使用するプログラムを新 OS で適切に動かすために再インストールしなければなりません。

Debian GNU/Linux では、うまく行かない場合、OS を取り替えるのではなく修理できるケースの方がはるかに多いでしょう。アップグレードでは大量のインストールは必要ありませんし、常にその場でアップグレードできます。また OS のリリースが続いても、プログラムにはほとんど常に互換性があります。プログラムの新バージョンが、より新しい依存するソフトウェアを要求する場合、Debian パッケージングシステムは、必要なソフトウェアをすべて自動的に識別し、確実にインストールします。再インストールが必要ないように力を尽くしてきており、再インストールをしなくてはならないというのは、最後の手段であるというのがポイントです。インストーラは、既に存在するシステムに対して、再インストールするように設計されていません。

ここでは、インストールプロセスの中で行う処理を一段階ずつまとめておきましょう。

1. インストールするハードディスクにある、既存のデータや文書のバックアップ。
2. インストールを始める前に、コンピュータの情報と必要な文書を集める。
3. ハードディスクに Debian のパーティションに使える領域を確保する。
4. インストーラソフトウェアと、そのマシンで必要になる、特殊なドライバファイルやファームウェアファイルについて、場所の確認・ダウンロード。
5. CD・DVD・USB メモリといったブートメディアをセットアップや、インストーラを起動できるネットワークブートインフラの準備。
6. インストールシステムを起動する。
7. インストールする言語を選択する。
8. 可能なら、イーサネットネットワーク接続を有効にする。
9. Debian をインストールするパーティションを作成し、マウントする。
10. 自動で行われる 基本システムのダウンロード・インストール・セットアップを監視する。
11. 追加のソフトウェアを選んでインストール。
12. Debian GNU/Linux と既存システムを起動するブートローダをインストールする。
13. 新しいシステムを初めて起動する。

インストール中に問題があったら、どのステップのどのパッケージでつまづいたかを知るお手伝いをします。このインストール劇の、そんな主演ソフトウェア俳優をご紹介します。

インストーラの `debian-installer` は、このマニュアルの主演です。ハードウェアを検出して適切なドライバをロードし、`dhcp-client` を使用してネットワーク接続を設定し、基本システムパッケージをインストールするのに `debootstrap` を実行し、さらに追加ソフトウェアをインストールする `tasksel` を実行します。このプロセスで多くの俳優が、より小さな役を演じますが、初めて新しいシステムを起動する時に、`debian-installer` はそのタスクを終えることになります。

システムをお好みに調整するには、`tasksel` を使用して Web サーバやデスクトップ環境といった、様々なソフトウェアの定義済みセットを選択・インストールできます。

インストール時の重要な選択肢に、X Window System とグラフィカルデスクトップ環境の 1 つからなる、グラフィカルデスクトップ環境をインストールするかどうかがあります。「デスクトップ環境」タスクを選択しない場合、比較的基本的な、コマンドライン駆動システムになります。デスクトップ環境は、テキストモードのみのシステムと比べて、かなり大きなディスク領域を必要とし、また、多くの Debian GNU/Linux システムは、グラフィカルユーザインターフェースを特に必要としないサーバであるため、デスクトップ環境タスクはオプションとなっています。

X Window System は、`debian-installer` とは完全に分かれていて、実際には非常に複雑なことに注意してください。X Window System のトラブルシュートは、このマニュアルでは扱いません。

3.2 既存データをバックアップしてください!

インストールを始める前に、現在使用しているシステムのすべてのファイルをバックアップしてください。今回初めて、最初から入っていたもの以外の OS をインストールするのでしたら、おそらくディスクのパーティション分割をやり直して Debian GNU/Linux 用の領域を作る必要があるでしょう。ディスクのパーティション分割作業では、どんなプログラムを使ったとしても、ディスク上のすべてのデータを消してしまう危険があります。Debian GNU/Linux のインストールに用いられるプログラム群は、極めて信頼性が高く、何年も使用されてきたものです。しかし、これらは強力な機能を持つことになるので、誤動作が起こったときの被害も大きくなります。バックアップを取った後でも、質問に答える前に充分注意し、よく考えて行動に移してください。ほんの数分間程余計に配慮することで、何時間もの不要な作業を避けることができるかもしれません。

また、システムをマルチブートシステムにする (複数のオペレーティングシステムを共存させる) 場合には、既にインストールされているオペレーティングシステムの配布メディアが手元にあることを確かめてください。通常は必要ないとはいえ、システムをブートするために、OS のブートローダを再インストールする必要があるあったり、最悪の場合、完全に OS をインストールし、以前のバックアップをリストアする必要がある可能性もあります。

3.3 必要な情報

3.3.1 ドキュメント

3.3.1.1 インストールマニュアル

現在ご覧になっている文書は、Debian の bullseye リリース用公式インストールガイドです。これは **さまざまな形式や言語** で利用できます。

3.3.1.2 ハードウェアの文書

しばしば、ハードウェアの設定や使用についての有用な情報を含んでいます。

3.3.2 ハードウェア情報の取得先

多くの場合、インストーラはハードウェアを自動的に検出することができます。しかし、準備としてインストール前にハードウェアに習熟することをお勧めします。

ハードウェアの情報は次のようなところから集められます。

- ・各ハードウェアに付属してきたマニュアル。

- ・ コンピュータの BIOS/UEFI 設定画面。この画面を表示させるには、コンピュータの起動時に何らかのキーの組合せを押します。この組合せについてはマニュアルを見てください。Delete キーや F2 キーの場合が多いようですが、いくつかのメーカーは、別のキーを使用することもあります。大抵、コンピュータの起動時に、設定画面に入るキーを表示します。
- ・ 各ハードウェアのケースや箱。
- ・ 他の OS のシステムコマンドやシステムツール、ファイルマネージャの表示など。こちらからは、RAM やハードドライブのメモリに関する情報が得られることが多いです。
- ・ あなたの部門のシステム管理者や、インターネットサービスプロバイダ。こちらからは、ネットワークや電子メールに関する設定情報が得られます。

Table 3.1 インストールに役立つハードウェア情報

ハードウェア	必要な情報
ハードディスク	ドライブの台数
	システムでの接続順序
	IDE (PATA としても知られる)、SATA、SCSI のどれか
	利用できる空き領域
	パーティション
	他の OS がインストールされているパーティション
ネットワークインターフェース	利用可能なネットワークインターフェースのタイプ/モデル
プリンタ	メーカーと型番
ビデオカード	タイプ/モデルとメーカー

3.3.3 ハードウェア互換性

製品の多くは、問題なく Linux で動作します。また Linux でサポートするハードウェアも日々進歩しています。しかし、それでもまだ Linux は、ある種の OS ほどには多種多様なハードウェアに対応していません。

Linux に収録されているドライバはほとんどの場合特定の製造者の一部の「製品」や「商標」向けではなく、あるハードウェア/チップセット向けに書かれています。一見異なるように見える製品/商標が同一のハードウェア設計を基にしています。チップ製造者が自社チップを基に「リファレンス設計」と呼ばれる製品を提供し、それが複数の異なるデバイス製造者により利用され、多くの異なる製品や商標名で売られることは珍しくありません。

これには利点と欠点があります。利点は、製品が同一チップセットを基にしている限りは製品や製造者が異なっても、一つのチップセットに一つのドライバで動作することです。欠点はある製品/商標で実際にはどのチップセットが使われているのか判定するのが常に簡単だとは限らないことです。残念なことに、デバイス製造者は製品のベースとなるハードウェアを変更してもその製品名や製品のバージョン番号すら変えないことが時々あり、そのために別々に買った商標/製品名が同一の製品が二つある場合、異なる二種のチップセットを基にしているため異なるドライバを使う必要があったり、一方には使えるドライバが何もないということもあります。

USB や PCI/PCI-Express/ExpressCard 用のデバイスが基にしているチップセットを調べるにはデバイス ID を確認するのが良い方法です。USB/PCI/PCI-Express/ExpressCard デバイスには全て、「ベンダー」および「製品」ID というものがあり、同一チップセットを基にしている製品であればこれは通常同じ組み合わせになります。

Linux システムでは、この ID は USB デバイスでは `lsusb` コマンド、PCI/PCI-Express/ExpressCard デバイスでは `lspci -nn` コマンドで読み取ることができます。ベンダーおよび製品 ID は通常「1d6b:0001」のように二つの 16 進数をコロンで区切った形式になっています。

`lsusb` の出力例: 「Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub」の場合、1d6b がベンダー ID で 0002 が製品 ID です。

イーサネットカードに対する `lspci -nn` の出力例: 「03:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller [10ec:8168] (rev 06)」。ID は最も右側の [] 内にあります。つまりこの場合 10ec がベンダー、8168 が製品の ID です。

また別の例として、あるグラフィックスカードでは次のような出力になります: 「04:00.0 VGA compatible controller [0300]: Advanced Micro Devices [AMD] nee ATI RV710 [Radeon HD 4350] [1002:954f

Windows システムでは、デバイスの ID は Windows デバイスマネージャの「詳細」タブで確認できます。ベンダー ID には先頭に VEN_、製品 ID には先頭に DEV_ が付けられます。Windows 7 システムではデフォルトでは表示されていないため、実際に ID を確認するにはデバイスマネージャの詳細タブのプロパティから「ハードウェア ID」を選択する必要があります。

ベンダー/製品 ID、「Linux」、「ドライバ」を検索語としてインターネットで検索すると多くの場合そのチップセット向けドライバの対応状態に関する情報が得られるでしょう。ベンダー/製品 ID を検索してあまり有効な検索結果が得られなかった場合は、多くの場合 lsubst や lspci でも提供されるチップのコード名 (ネットワークカードの例では「RTL8111」/「RTL8168B」、グラフィックスカードの例では「RV710」) を検索することで手がかりが得られるかもしれません。

3.3.3.1 ライブシステムを使用したハードウェア互換性テスト

Debian GNU/Linux は一部のアーキテクチャで「ライブシステム」というものも利用できます。ライブシステムは設定済みですぐに使える圧縮形式のシステムで、CD や DVD のような読み込み専用メディアから起動して使えます。デフォルトでの使用では、コンピュータ上への恒久的な変更は一切行いません。ライブシステム内でユーザ設定を変更したりプログラムを追加でインストールすることはできますが、全てコンピュータの RAM 上でのみ発生します。つまり、コンピュータの電源を落としてライブシステムを起動し直すと、全てがデフォルトにリセットされます。手持ちのハードウェアが Debian GNU/Linux でサポートされているか確認する最も簡単な方法は Debian ライブシステムを使って試してみることです。

ライブシステムの使用にはいくつか制限があります。まず、ライブシステム内での変更は全てコンピュータの RAM に保持する必要があることで、そのため十分な RAM のあるシステムでないと機能しません。巨大なソフトウェアパッケージを追加でインストールすることはメモリの制約のために失敗するかもしれません。もう一つの制限はハードウェア互換性テストに関するもので、公式の Debian GNU/Linux ライブシステムにはフリーなものしか含まれません。つまり、フリーでないファームウェアファイルは一切含まれません。そういったフリーでないパッケージを手作業でシステムにインストールすることはもちろんできますが、debian-installer のように必要なファームウェアファイルを自動的に検出するようなものは一切ないため、フリーでないものが必要であれば全て手作業によってインストールしなければなりません。

利用可能な Debian ライブイメージの種類についての情報は [Debian ライブイメージのウェブサイト](#)にあります。

3.3.4 ネットワークの設定

コンピュータが固定ネットワークに接続されているならば (つまり、PPP 接続ではなく Ethernet やそれと同等な接続の場合)、ネットワーク管理者に以下の情報を確認しておいてください。

- ホスト名 (自分で決められるかもしれません)
- ドメイン名
- コンピュータの IP アドレス
- ネットワークのネットマスク
- ネットワークにゲートウェイがある場合は、経路を向けるデフォルトゲートウェイシステムの IP アドレス
- DNS (Domain Name Service) サーバとして使用するネットワーク上のホスト

接続するネットワークを、DHCP (Dynamic Host Configuration Protocol) を用いて設定する場合、DHCP サーバがインストールプロセスの間、コンピュータに直接提供するので、この情報は必要ありません。

DSL やケーブルモデム (つまりケーブルテレビネットワーク) を経由したインターネット接続とルータ (設定済みのものが電話や CATV の提供者により提供されることがよくあります) があって、それがネットワーク接続を処理している場合、通常 DHCP がデフォルトで使えるようになっています。

WLAN/WiFi ネットワークを使用するなら、以下の情報も探さねばなりません。

- ワイヤレスネットワークの ESSID (「ネットワーク名」)。
- (適用できる場合) ネットワークにアクセスする WEP や WPA/WPA2 のセキュリティキー。

3.4 必要な最低限のハードウェア

コンピュータのハードウェアに関する情報が集まったら、そのハードウェアが今から行おうとしているインストールの条件に足るものであるかどうかをチェックしましょう。

やむを得ない場合は、以下に載っているリストよりは性能の劣るハードウェアでなんとかしなければならぬこともあるでしょう。しかし、これらのお勧めを無視した場合は、結局不満を感じる可能性が高くなってしまいます。

Table 3.2 最低限必要なシステム (推奨値)

インストールタイプ	RAM (最小)	RAM (推奨)	ハードディスク
デスクトップなし	256 メガバイト	512 メガバイト	2 ギガバイト
デスクトップあり	1 ギガバイト	2 ギガバイト	10 ギガバイト

この最小値はスワップを有効にしてライブ CD でないイメージを使うことを前提としています。「デスクトップなし」の値はグラフィカルでないインストーラを使うことを前提としています。

実際に必要な最小メモリはこの表に挙げたものよりも少なくなります。スワップを有効にすれば、最小 140MB で Debian をインストールできます。必要なディスクスペースにも同じことが言え、特にインストールするアプリケーションを選択する場合に必要なディスクスペースについての追加情報は、[項D.2](#)をご覧ください。

旧式ないしローエンドシステムでも、グラフィカルデスクトップ環境を実行できますが、GNOME や KDE Plasma といったデスクトップ環境よりも、リソースを消費しないウィンドウマネージャをインストールするをお勧めします。代替品には、xfce4, icewm, wmaker が含まれますが、他にも選択できます。

インストール時に必要なメモリやディスクの量は、どのようなサーバとして使用するかによって異なるため、一般的な量の提示は事実上不可能です。

これらのサイズには、通常存在するユーザファイル、メール、データなどは含まれていないことにご注意ください。自分のファイルやデータに必要な容量は、気前良く確保しておくに越したことはありません。

Debian GNU/Linux システムを円滑に操作するのに必要なディスクスペースについては、お勧めするシステム要件で考慮されています。特に、/var パーティションには、ログファイルのような一般的な内容に加え、Debian 特有の状態情報が多く置かれます。dpkg のファイル (インストールされたパッケージすべてに関する情報) は、簡単に 40MB を消費します。また apt は、インストールする前にダウンロードしたパッケージをここに置きます。/var には最低 200MB は割り当てておくべきですし、グラフィカルデスクトップ環境をインストールする場合には、もっと割り当てるべきでしょう。

3.5 マルチブートシステムでの事前パーティション分割

「ディスクのパーティション分割」とは、ディスクをセクションに分けることです。各セクションは他のセクションから独立しています。この作業は要するに、家の中に壁を作るようなものです。ある部屋に家具を入れても、それは他の部屋には影響しないというわけです。

システム上に既にオペレーティングシステムが入っていてディスク全体を使っているときに同じディスクに Debian も入れたい場合には、ディスクのパーティション分割をやり直す必要があります。Debian は Windows や MacOS のパーティションにはインストールできません。他の Unix システムとはパーティションを共有することも可能かもしれませんが、ここではそれは取り扱いません。少なくとも、Debian の root ファイルシステムには専用のパーティションが必要となります。

現在のパーティションの設定は、のような、現在の OS に対応したパーティション分割ツールを使えばわかります。パーティション分割ツールには、必ず既存のパーティションを (変更せずに) 表示する機能が付いています。

一般に、既にファイルシステムの入っているパーティションを変更すると、中の情報はすべて破壊されてしまいます。そのため、パーティション分割をやり直す前には、必ずバックアップを取っておいてください。家の例でいうと、壁を動かす前には、家具が壊れないよう、あらかじめどかしておきます。

今の OS の中には、既存のパーティションをその内容を壊さずに移動、サイズ変更できる機能を提供しているものがあります。この機能を使うと既存データを失うことなくスペースを作ることができます。これはほとんどの場合問題なく成功しますが、ディスクのパーティションを変更することは本質的に危険な操作であり、全データを完全にバックアップした上で行うべきものです。

パーティションの作成や削除は既存の OS からと同じように `debian-installer` からできます。経験則ですが、パーティションはそれを使うシステムから作成すべきです。つまり、Debian GNU/Linux により使われるパーティションは `debian-installer` から作成すべきで、他の OS から利用するパーティションはそちらから作成すべきです。`debian-installer` には Linux 以外のパーティションを作成する機能があり、そうやって作成したパーティションは他の OS から通常問題なく使えますが、ごくレアな状況でこれが問題となる可能性があります。そのため、確実を期するのであれば、他の OS で利用するパーティションの作成にはネイティブのパーティション用ツールを使ってください。

同じマシンに複数の OS をインストールするつもりでしたら、Debian をインストールする前に、他の OS を全部先にインストールしておきましょう。Windows などの他の OS をインストールすると、Debian を起動する機能が破壊されてしまったり、あるいはその OS のものでないパーティションをフォーマットし直すよう促されたりするからです。

このような動作から復旧したり、そのような提案を断ったりすることはできますが、先にそちらのシステムをインストールしておけば、最初からトラブルを避けることができます。

3.6 インストール前に行うハードウェア・OS の設定

この節では、Debian のインストールに先立って必要となるハードウェアの設定について見ていきます。通常この作業では、システムの BIOS/UEFI/システム用ファームウェアの設定をチェックし、場合によってはその設定を変更することになります。「BIOS/UEFI」や「システムファームウェア」は、ハードウェアが利用する中核的なソフトウェアで、電源投入後のブートプロセスの間に起動される、最も重要なものです。

3.6.1 ARM ファームウェア

既に触れたように、残念ながら ARM システムのシステムファームウェアには標準となるものがありません。名義上同一のファームウェアを利用しているも、システムが異なれば挙動はかなり異なることがあります。これは ARM アーキテクチャを採用している機器の大半が組み込みシステムであり、製造者は通常独自の変更を相当に加えたバージョンのファームウェアをビルドし、機器特有のパッチを収録しているという事実に起因します。残念ながら製造者が変更や拡張を主流側のファームウェア開発者に送らないことが多く、そのため製造者による変更が元のファームウェアの新しいバージョンにはなかなか取り入れられません。

結果として新しく販売されているシステムであっても、製造者により改変された 1 年以上古いバージョンのファームウェアを基にしたファームウェアを使っていることが多く、一方で主流側のコードはその間に大きく進化して追加の機能を提供していたり、ある面で異なる挙動を取るということがあります。それに加え、オンボードデバイスの命名方法は異なる製造者が改変したバージョン間では同一のファームウェアであっても一貫性がなく、そのため ARM ベースのシステムで製品に依存せずに利用できる命令を提供するのはほぼ不可能です。

3.6.2 Debian から提供されている U-Boot (システムファームウェア) イメージ

Debian は `.../images/u-boot/` で SD カードから U-Boot を読み込める様々な `armhf` システム向けの U-Boot イメージを提供しています。U-Boot ビルドは 2 つの形式を提供しています: U-Boot の構成要素をそのまま収録したものと簡単に SD カードに書き込める既製のカードイメージです。U-Boot の構成要素をそのまま提供しているのは上級ユーザ向けです。通常は既製の SD カードイメージのどれかの利用を勧めています。これは `<システム種類>.sdcard.img.gz` といった名前で、例えば

```
zcat <システム種類>.sdcard.img.gz > /dev/SDカードのデバイス
```

のようにしてカードに書き込めます。どのイメージでも同じですが、イメージを SD カードに書き込むとそのカードにあった内容は消えてしまうことに留意してください!

当該システム用の U-Boot イメージが Debian から提供されている場合はベンダーから提供された U-Boot よりもそのイメージの使用を勧めます。通常 Debian 版の方が新しく多機能なためです。

3.6.3 U-Boot でのイーサネット MAC アドレスの設定

通常、全イーサネットインターフェイスの MAC アドレスが全体で一意であるべきで、技術的にはイーサネットブロードキャストドメイン内で一意でないといけません。それを実現するために製造者は通

常、中央管理されている (対価を払う必要のある) MAC アドレスブロックを割り当てられ、そのアドレスの中から 1 つを、販売する各製品に事前設定します。

開発用ボードの場合、製造者が対価の支払いを避けたいために全体で一意的なアドレスが提供されないこともあります。その場合はユーザ自身でシステムの MAC アドレスを決めないといけません。イーサネットインターフェイスの MAC アドレスが決められていない場合、ネットワークドライバによっては MAC アドレスを無作為に生成します。そうして生成された MAC アドレスはブートのたびに変わる可能性があり、変更が発生した場合、ユーザが手作業によりアドレスをセットしなくてもネットワークアクセスは可能でしょうが、例えばリクエストしたクライアントの MAC アドレスを基にして DHCP により半固定の IP アドレスを割り当てるような場合の動作に信頼性が失われるのは明らかです。

公式に割り当てられている既存の MAC アドレスとの競合を回避するため、いわゆる「ローカル管理用」アドレスとして予約されているアドレス領域があります。アドレスの第 1 バイトの特定の 2 ビットの値が決められています (英語版 Wikipedia の記事「MAC_address」に良い説明があります)。事実上これは例えば 16 進数 `ca` から始まる任意のアドレス (`ca:ff:ee:12:34:56` 等) をローカル管理用アドレスとして利用できるということになります。

システムのファームウェアとして U-Boot を使っているシステムでは、イーサネット MAC アドレスは「`ethaddr`」環境変数にセットされています。U-Boot のコマンドプロンプトでコマンド「`printenv ethaddr`」により確認、コマンド「`setenv ethaddr ca:ff:ee:12:34:56`」によりセットできます。値の設定後にコマンド「`saveenv`」を実行するとその割り当てが恒久的になります。

3.6.4 U-Boot でのカーネル/initrd/デバイスツリーの再配置問題

以前のバージョンの U-Boot を利用しているシステムの一部ではブートの過程で Linux カーネルや初期 RAM ディスク、デバイスツリー blob のメモリへの再配置に不具合がある可能性があります。その場合 U-Boot は「Starting kernel ...」というメッセージを表示しますがシステムはそれ以上出力することなくフリーズします。この問題は v2014.07 以降の新しいバージョンの U-Boot では解決されています。

システムが元々 v2014.07 よりも古いバージョンの U-Boot を利用していたもので後から新しいバージョンにアップグレードした場合、U-Boot のアップグレード後でも問題が引き続き発生するかもしれません。U-Boot のアップグレードでは通常既存の U-Boot 環境変数を変更せず、この問題の修正には追加の環境変数 (`bootm_size`) をセットする必要があり、既存の環境データの存在しない新規インストール処理でのみ U-Boot は自動的にそれをセットします。コマンド「`env default bootm_size; saveenv`」を U-Boot のプロンプトで手作業により実行することで U-Boot の新しいデフォルト値に `bootm_size` をセットできます。

再配置関連の問題を回避する別の策として、コマンド「`setenv fdt_high ffffffff; setenv initrd_high 0xffffffff; saveenv`」を U-Boot プロンプトで実行して初期 RAM ディスク、デバイスツリー blob の再配置を完全に無効化する方法があります。

Chapter 4

システムインストールメディアの入手

4.1 公式 Debian GNU/Linux インストールイメージ

現在、Debian GNU/Linux をインストールする最も簡単な方法は、公式 Debian インストールイメージセットを使うことです。ベンダからこの CD/DVD セットを購入できます ([CD ベンダページ](#)をご覧ください)。高速なネットワーク接続と CD/DVD 書き込み装置があれば、Debian ミラーからインストールイメージをダウンロードしてもかまいません (詳細説明は [Debian CD ページ](#)と [Debian CD FAQ](#) をご覧ください)。そのような光学インストールメディアを持っていて、マシンをこれらから起動できるなら、第5章の項目までスキップできます。よく使用するファイルが CD や DVD の最初のイメージにあることを保証するために、大きな労力が費やされています。そのため、基本的なデスクトップは最初の DVD だけで足りるし、限られた範囲内では、最初の CD イメージのみでもインストールできます。

昨今の標準としては、CD は少々容量に制限があり、グラフィカルデスクトップ環境のすべてを、先頭の CD だけですべてインストールできなかったため、いくつかのデスクトップ環境では、CD でのインストールに、ダウンロード用のネットワーク接続か、追加 CD から残りのファイルの取得が必要となります。

もう一点、留意しておいてください: あなたが使っているインストールメディアが必要なパッケージを含んでいない場合、その後動作している新たな Debian システムからこれらのパッケージをインストールできます (インストール完了後になります)。特定のパッケージを見つけるためにどのインストールメディアにあるかを知る必要がある場合は、<https://cdimage-search.debian.org/> を見てください。

あなたのマシンが光学メディアからの起動をサポートしていなくても、CD/DVD セットを持っているのであれば、最初のシステムインストーラの起動にネットブート、ディスクからカーネルの手動起動といった別の方法が使えます。これらの方法による起動に必要なファイルもディスクに収録されており、Debian ネットワークアーカイブとディスクのフォルダ構成は同じです。そのため、以降で起動に必要なそれぞれのファイルの、アーカイブファイルパスがわかれば、インストールメディアの同じディレクトリやサブディレクトリからファイルを探せます。

いったんインストーラが起動すれば、ほかの必要なファイルはすべてディスクから取得できます。

インストールメディアセットを持っていない場合は、インストーラのシステムファイルをダウンロードして、接続されたコンピュータのいずれかに保存します。そしてそこからインストーラを起動します。

4.2 Debian ミラーサイトからのファイルのダウンロード

もっとも近い (そしておそらくもっとも速い) ミラーサイトを探すには、[Debian ミラーサイト一覧](#) を参照してください。

4.2.1 どこでインストールファイルを探すか

様々なインストールファイルが各 Debian ミラーサーバの [debian/dists/bullseye/main/installer-armhf/-current/images/](#) にあります。各イメージとその用途が、[MANIFEST](#) に記載されています。

4.2.1.1 Armbf 複数プラットフォーム対応カーネルのインストールファイル

armhf 複数プラットフォーム対応カーネル (項2.1.4 参照) によりサポートされるシステムのインストール用ファイルは、標準的な Linux カーネルイメージ、標準的な Linux 初期 RAM ディスクイメージ、システム特有のデバイスツリー blob で構成されます。カーネルと初期 RAM ディスクイメージは `.../images/netboot/`、tftp ブート用のデバイスツリー blob は `.../images/device-tree/` からそれぞれ取得できます。インストーラを備えたブート可能な USB メモリを作成するための tar アーカイブは `.../images/hd-media/` から取得できます。

様々な armhf プラットフォーム用の U-Boot イメージは `.../images/u-boot/` から利用できるようになっています。

4.3 TFTP ネットブート用ファイルの準備

インストール対象のマシンが LAN に接続されている場合、TFTP を用いると、そのマシンをネットワーク越しに他のマシンから起動できます。インストールシステムを別のマシンから起動するには、その「別のマシン」の特定の場所に起動ファイルを置き、またインストール対象のマシンの起動をサポートするよう設定しなければなりません。

TFTP サーバをセットアップする必要があり、そして多くのマシンでは DHCP サーバ、または RARP サーバ、または BOOTP サーバのセットアップも必要です。

Reverse Address Resolution Protocol (RARP) は、どの IP を用いるべきかをクライアントに伝える方法のひとつです。同種の方法には BOOTP プロトコルがあります。BOOTP は IP プロトコルのひとつです。クライアントに対して、使うべき IP アドレスと、ブートイメージをネットワークのどこから取得するかを伝えます。DHCP (Dynamic Host Configuration Protocol) は、BOOTP との後方互換性を保ちつつ、より柔軟に拡張させたものです。システムによっては DHCP でしか設定できないこともあります。

Trivial File Transfer Protocol (TFTP) は、ブートイメージをクライアントに提供するために用います。理論的には、どんなサーバでも、どんなプラットフォームでも、これらのプロトコルを実装してさえいれば利用できます。この節では、SunOS 4.x, SunOS 5.x (Solaris), GNU/Linux での例を示します。

4.3.1 RARP サーバの設定

RARP を設定するには、クライアントコンピュータにインストールしているイーサネットのアドレス (MAC アドレス) を調べておく必要があります。この情報がわからなければ、「Rescue」モードを起動して `ip addr show dev eth0` コマンドを使ってください。

Linux カーネルの RARP サーバシステムや Solaris/SunOS では `rarpd` を使用します。クライアントのイーサネットハードウェアアドレスを「ethers」データベースと「hosts」データベースにリストしておく必要があります (これらのデータベースは `/etc/(ethers,hosts)` ファイルか NIS/NIS+ のマップにします)。次に RARP デモンを起動します。ほとんどの Linux システムと SunOS 5 (Solaris 2) では `/usr/sbin/rarpd -a` を、いくつか他の Linux システムでは `/usr/sbin/in.rarpd -a` を、SunOS 4 (Solaris 1) では `/usr/etc/rarpd -a` を (ルート権限で) 実行してください。

4.3.2 DHCP サーバの設定

フリーソフトウェアの DHCP サーバのひとつに、ISC の `dhcpd` があります。Debian GNU/Linux では、`isc-dhcp-server` パッケージをお勧めします。以下に、設定ファイルの例を示します。(`/etc/dhcpd.conf` を参照)

```
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;
}
```

この例では、servername というサーバがひとつあり、DHCP サーバ、TFTP サーバ、ネットワークゲートウェイの仕事すべてを行っています。domain-name オプション、サーバ名、クライアントのハードウェアアドレスは、必ず変更する必要があります。filename オプションは TFTP 経路で取得するファイルの名前です。

dhcpd の設定ファイルの編集を終えたら、/etc/init.d/isc-dhcp-server restart で dhcpd を再起動してください。

4.3.3 BOOTP サーバの設定

GNU/Linux で使える BOOTP サーバは 2 つあります。ひとつは CMU の bootpd です。もう 1 つは実際は DHCP サーバですが、ISC の dhcpd です。Debian GNU/Linux では、bootp パッケージと isc-dhcp-server パッケージにそれぞれ入っています。

CMU bootpd を使う場合は、まず /etc/inetd.conf ファイルの該当行をアンコメント (または追加) する必要があります。Debian GNU/Linux では update-inetd --enable bootps を実行し、続いて /etc/init.d/inetd reload とすれば OK です。BOOTP サーバが Debian で動かない場合は、以下のようになります。

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

ここで次に /etc/bootptab を作成します。このファイルの書式は、printcap, termcap, disktab ファイルなどでお馴染みの、例のわかりにくい形式になっています。詳細は bootptab マニュアルページを見てください。CMU bootpd では、クライアントのハードウェア (MAC) アドレスを知っておかなければなりません。/etc/bootptab の例を示します。

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

少なくともクライアントのハードウェアアドレスを指定している「ha」オプションは変更する必要があります。 「bf」オプションはクライアントが TFTP で取得するファイルを指定しています。詳細は項 4.3.5 を参照してください。

対照的に、ISC dhcpd を使った BOOTP の設定は実に簡単です。dhcpd では、BOOTP クライアントはやや特殊な DHCP クライアントとして取り扱われます。アーキテクチャによっては、BOOTP によるクライアントの起動には複雑な設定が必要になります。これに該当してしまったら、項 4.3.2 の節をお読みください。そうでなければ、おそらく /etc/dhcp/dhcpd.conf にある、クライアントの含まれるサブネットの設定ブロックに、allow bootp というディレクティブを追加し、/etc/init.d/isc-dhcp-server restart で dhcpd を再起動するだけです。

4.3.4 TFTP サーバの立ち上げ

TFTP サーバの準備をする際に、まず、tftpd が有効であることを確認してください。

tftpd-hpa の場合には、サービスを走らせるのに 2 種類の方法があります。システムの inetd デモンにより必要に応じて起動する方法と、独立したサーバとして起動する方法です。どちらにするかは、パッケージのインストール時や再設定時に選択できます。

注意



歴史的に TFTP サーバは、イメージを提供するディレクトリに /tftpbboot を使用します。しかし Debian GNU/Linux のパッケージでは、**Filesystem Hierarchy Standard** を満たす別のディレクトリを使用する可能性があります。例えば、tftpd-hpa では /srv/tftp をデフォルトで使用します。必要に応じて、本節の設定例を調整してください。

Debian で利用できる in.tftpd の代替はすべて、デフォルトで TFTP リクエストをシステムログに出力します。いくつかは、出力を冗長にする -v 引数をサポートしています。ブート時に問題がある場合、エラーの原因を診断する出発点として、ログメッセージをチェックすることをお勧めします。

4.3.5 TFTP イメージを適切な場所に配置する

次に行うことは、項4.2.1 の記述にある、必要な TFTP ブートイメージを、tftpd のブートイメージディレクトリに置く作業です。tftpd が特定のクライアントの起動に用いるファイルへのリンクを、ここに作成してください。残念ながら、ファイルの名前は TFTP クライアントによって決まり、強制力のある標準は存在しません。

4.4 自動インストール

複数のコンピュータにインストールするため、完全自動インストールが可能です。このための Debian パッケージは、fai-quickstart (インストールサーバとして使用可能) と Debian インストーラそのものです。詳細情報は [FAI home page](#) をご覧ください。

4.4.1 Debian インストーラを用いた自動インストール

Debian インストーラは、preconfiguration ファイルによる自動インストールをサポートしています。preconfiguration ファイルは、ネットワークやリムーバブルメディアから読み込まれ、インストール中の質問に対する回答を、埋めていくのに使われます。

編集できる動作サンプルを含む preseed の完全なドキュメントは、付録B にあります。

4.5 インストールファイルの整合性の検証

ダウンロードしたファイルの整合性を Debian ミラー上にある SHA256SUMS または SHA512SUMS ファイルで提供しているチェックサムに対して検証できます。これらのファイルはインストールイメージ自体と同じ場所にあります。次の場所を見てください。

- [CD イメージのチェックサムファイル](#)、
- [DVD イメージのチェックサムファイル](#)、
- [他のインストールファイルのチェックサムファイル](#)。

ダウンロードしたインストールファイルのチェックサムを計算するには、それぞれ

```
sha256sum filename.iso
```

もしくは

```
sha512sum filename.iso
```

を使い、そして表示されたチェックサムを対応するファイル SHA256SUMS もしくは SHA512SUMS で比較してください。

[Debian CD FAQ](#) にはこのトピックの **もっと有用な情報** (例えばスクリプト check_debian_iso で上の手順を半自動化できます) があり、説明や上のチェックサムファイル自体の整合性の検証方法もあります。

Chapter 5

インストールシステムの起動

5.1 32-bit hard-float ARMv7 でのインストーラの起動

5.1.1 ブート用イメージのフォーマット

ARM ベースのシステムではほとんどの場合、ブート用イメージに 2 つのフォーマット: a) 標準的な Linux 初期 RAM ディスク (「initrd.gz」) と併用する標準的な Linux zImage フォーマットのカーネル (「vmlinuz」)、または b) 対応する初期 RAM ディスク (「ulnitrd」) と併用する ulimage フォーマットのカーネル (「ulimage」) のどちらかを利用します。

ulimage/ulnitrd は ARM ベースのシステムの多く (ほとんどが 32 ビットシステムです) で利用されている U-Boot ファームウェア向けに設計されたイメージフォーマットです。古いバージョンの U-Boot でブートできるのは ulimage/ulnitrd フォーマットのファイルだけなので、古い armel システムではこの形式がよく利用されています。新しいバージョンの U-Boot では - ulimage/ulnitrd のブート以外に - 標準的な Linux カーネル及び RAM ディスクイメージもブートできますが、そのためのコマンド書式は ulimage のブートとはわずかに異なります。

複数プラットフォーム対応カーネルを利用するシステムでは、カーネルと初期 RAM ディスク以外にいわゆるデバイスツリーファイル (デバイスツリー blob 「device-tree blob, dtb」) が必要です。これはサポートする各システムごとに特有のもので、特定のハードウェアの概要が収録されています。dtb はファームウェアによってデバイスに提供されるはずですが、実際にはもっと新しい dtb をロードする必要があることはよくあります。

5.1.2 コンソール設定

netboot 用の tar アーカイブ (項 5.1.3.2) 及びインストーラの SD カード用イメージ (項 5.1.5) は U-Boot により「console」変数で定義される (プラットフォーム特有の) デフォルトのコンソールを使います。ほとんどの場合これはシリアルコンソールなので、そういったプラットフォームでインストーラを利用するにはデフォルトでシリアルコンソールのケーブルが必要となります。

ビデオコンソールもサポートしているプラットフォームでは、インストーラをビデオコンソールで起動したければ U-Boot の「console」変数をそれに合わせて変更することもできます。

5.1.3 TFTP による起動

ネットワークからの起動には、ネットワーク接続と TFTP ネットワークブートサーバが (自動化でのネットワーク設定には恐らく DHCP, RARP, BOOTP も) 必要です。

ネットワーク起動をサポートするためのサーバ側の準備については、項 4.3 で説明します。

5.1.3.1 U-Boot での TFTP のブート

U-Boot ファームウェアを採用したシステムでのネットワークブートは 3 つの段階から構成されます: a) ネットワークの設定、b) イメージ (カーネル/初期 RAM ディスク/dtb) のメモリへの読み込み、c) 前段階で読み込んだコードを実際に行う。

最初にネットワークを設定する必要があります。DHCP により自動的に設定する場合:

```
setenv autoload no
dhcp
```

あるいは手作業により複数の環境変数を設定する場合:

```
setenv ipaddr <クライアントのIPアドレス>
setenv netmask <netmask>
setenv serverip <tftp サーバのIPアドレス>
setenv dnssip <名前サーバのIPアドレス>
setenv gatewayip <デフォルトゲートウェイのIPアドレス>
```

希望により、恒久的な設定にすることもできます:

```
saveenv
```

その後はイメージ (カーネル/初期 RAM ディスク/dtb) をメモリに読み込む必要があります。これは tftpboot コマンドで行いますが、イメージが記憶されているメモリのアドレスを指定する必要があります。残念ながらメモリの割り当てはシステムにより異なる可能性があるため、どのアドレスを利用できるというような原則はありません。

システムによっては読み込みに適するアドレスをセットした環境変数群を U-Boot が事前に定義しているものがあります: kernel_addr_r、ramdisk_addr_r、fdt_addr_r。定義されているかどうかは、

```
printenv kernel_addr_r ramdisk_addr_r fdt_addr_r
```

を実行することで確認できます。定義されていない場合はシステムの文書で適切な値を確認して手作業によりセットする必要があります。Allwinner SunXi SOC ベースのシステム (例えば Allwinner A10、アーキテクチャ名「sun4i」や Allwinner A20、アーキテクチャ名「sun7i」) では、例えば以下の値を利用できます:

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

読み込むアドレスが定義されていれば

```
tftpboot ${kernel_addr_r} <カーネルイメージのファイル名>
tftpboot ${fdt_addr_r} <dtbのファイル名>
tftpboot ${ramdisk_addr_r} <初期RAMディスクイメージのファイル名>
```

を実行することにより、前に定義した tftp サーバからイメージをメモリに読み込みます。3つ目の部分はカーネルコマンドラインの設定で、読み込んだコードを実際に実行します。U-Boot は「bootargs」環境変数の内容をカーネルにコマンドラインとして渡すので、カーネルやインストーラへの任意のパラメータ - 例えばコンソールデバイス (項5.3.1 参照) や preseed のオプション (項5.3.2 及び付録B 参照) - を

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

のようなコマンドでセットできます。前に読み込んだコードを実行する実際のコマンドは使用するイメージのフォーマットに依存します。ulmage/ulnitrd の場合、コマンドは

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

となり、ネイティブの Linux イメージの場合は

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

となります。注意: 標準的な linux イメージをブートする場合、カーネルと dtb を読み込んでから初期 RAM ディスクイメージを読み込むことが重要となります。U-Boot が最後に読み込んだファイルのサイズを filesize 変数にセットすると bootz コマンドが正常に動作するためには RAM ディスクイメージのサイズが必要となるためです。プラットフォーム特有のカーネル、つまりデバイスツリー無しでカーネルをブートする場合には \${fdt_addr_r} パラメータは省略できます。

5.1.3.2 ビルド済み netboot tar アーカイブ

Debian は、tftp サーバ上で単純に展開でき、netboot に必要なファイルを全て収録しているビルド済みの tar アーカイブ (.../images/netboot/netboot.tar.gz) を提供しています。また、これにはインストーラを読み込む手順を全て自動化するブートスクリプトも収録しています。現在の U-Boot バージョンにはブート可能なローカルストレージ機器 (MMC/SD、USB、IDE/SATA/SCSI) がない場合には tftp の autoboot 機能が有効化されて tftp サーバからブートスクリプトを読み込むという機能が

あります。この機能を使うにはクライアントに tftp サーバのアドレスを提供する dhcp サーバがネットワークにあることが条件となります。

U-Boot のコマンドラインから tftp の autoboot 機能を作動させたい場合は以下のコマンドを使えます:

```
run bootcmd_dhcp
```

tar アーカイブにより提供されたブートスクリプトを手作業でロードするには、U-Boot プロンプトで以下のコマンドを実行する方法もあります:

```
setenv autoload no
dhcp
tftpboot ${scriptaddr} /debian-installer/armhf/tftpboot.scr
source ${scriptaddr}
```

5.1.4 U-Boot での USB メモリからの起動

現在の U-Boot バージョンの多くが USB をサポートし、USB メモリ等の USB 大容量ストレージ機器からのブートができるようになってきました。しかし残念ながら実際に必要な手順は機器によってかなり様々です。

U-Boot v2014.10 で一般的なコマンドラインの処理と autoboot フレームワークが導入されました。これによって、このフレームワークを実装したシステムであればどれも機能する包括的なブートイメージを作れるようになりました。debian-installer はこういったシステムでの USB メモリからのインストールをサポートしていますが、残念ながら全ての基盤でこの新しいフレームワークを採用するには至っていません。

ブート可能な Debian インストール用 USB メモリをビルドするには、hd-media の tar アーカイブ (項 4.2.1 参照) をその機器の U-Boot バージョンでサポートしているファイルシステムでフォーマットした USB メモリに展開します。現在の U-Boot バージョンでは通常 FAT16 / FAT32 / ext2 / ext3 / ext4 のどれでも使えます。それから 1 枚目の Debian インストール CD/DVD の ISO イメージファイルを USB メモリにコピーします。

現在の U-Boot バージョンの autoboot フレームワークは PC BIOS/UEFI のブート順オプションと似た動作をします。つまり、ブート可能機器一覧から有効なブートイメージを確認して最初に見つけたものを起動します。OS がインストールされていなければ、USB メモリを差し込んでシステムの電源を入れればインストーラが起動するはずですが、また、U-Boot のプロンプトで「run bootcmd_usb0」コマンドを入力すればいつでも USB の起動処理を始められます。

シリアルコンソールを使って USB メモリからブートした場合に起きる問題として、コンソールボーレートの不一致があります。console 変数が U-Boot で定義されている場合、debian-installer ブートスクリプトは第一のコンソールデバイスをセットするために、可能であればコンソールボーレートを含めて、それを自動的にカーネルに渡します。残念ながら console 変数の処理は基盤によって様々です - console 変数にボーレートまで含める (「console=ttyS0,115200」) 基盤もあれば console 変数には (「console=ttyS0」のように) デバイスだけを含める基盤もあります。後者の場合に U-Boot とカーネルのボーレートのデフォルト値が異なっていると、コンソール出力が化ける原因となります。カーネルは未だに古い 9600 ボーをデフォルト値としていますが、U-Boot の現在のバージョンでは 115200 ボーがよく使われています。そうなった場合は、手作業で console 変数にシステムの正しいボーレートを指定して、「run bootcmd_usb0」コマンドでインストーラを起動します。

5.1.5 インストーラでのビルド済み SD カードイメージの利用

いくつかのシステムでは U-Boot と debian-installer を両方とも収録した SD カード用のイメージを Debian で提供しています。SD カード用のイメージは 2 種類 - ソフトウェアパッケージをネットワークからダウンロードするタイプ ([.../images/netboot/SD-card-images/](#) から利用可) と Debian CD/DVD を使用してオフラインでインストールするタイプ ([.../images/hd-media/SD-card-images/](#) から利用可) を提供しています。容量とネットワーク帯域を節約するためにイメージは 2 部構成となっています - システムに依存する部分 (「firmware.<system-type>.img.gz」) とシステムに依存しない部分 (「partition.img.gz」) です。

Linux システムの 2 つの部分から完全なイメージを作成するには zcat を使って

```
zcat firmware.<system-type>.img.gz partition.img.gz > complete_image.img
```

のようにします。Windows システムでは 2 つをまず別個に伸張する必要があります。これには例えば 7-Zip を使えます。それから Windows CMD.exe ウィンドウでコマンド

```
copy /b firmware.<system-type>.img + partition.img complete_image.img
```

を実行してそれぞれ伸長したものを結合します。

出来上がったイメージを例えば Linux システムでは

```
cat complete_image.img > /dev/SD_CARD_DEVICE
```

を実行して SD カードに書き込みます。それから対象のシステムに SD カードを差し込んでシステムの電源を入れると SD カードからインストーラが読み込まれます。hd-media 版を使ってオフラインでインストールする場合は、インストーラから例えば USB メモリに書き込んだ CD/DVD の ISO イメージ等、別個のメディアに置かれた 1 枚目の Debian CD/DVD にアクセスできないといけません。

インストールがパーティション作業 (項 6.3.4 参照) の段階まで進んだら、それまでカードにあったパーティションは削除、書き換えできます。インストーラは起動したら完全にシステムのメインメモリで動作し、以後 SD カードにアクセスする必要はなくなるため、Debian のインストールにカード全体を使えます。SD カードに適切なパーティションレイアウトを作成する最も簡単な方法はインストーラに自動で作成させる方法です (項 6.3.4.2 参照)。

5.2 アクセシビリティ

あるユーザにとっては、目が不自由であるといった理由により、それぞれに見合った支援が必要なことがあります。多くのアクセシビリティ機能は手動で有効にする必要があります。アクセシビリティ機能を有効にするために、起動パラメータを付加できます。ほとんどのアーキテクチャでは、キーボードを QWERTY キーボードとして認識することに注意してください。

5.2.1 インストーラフロントエンド

Debian インストーラは質問方法にアクセシビリティの異なる複数のフロントエンドをサポートしています: 特に挙げると text では平文のテキストを使い、一方 newt ではテキストベースのダイアログボックスを使います。ブートプロンプトで選択できます。詳しくは項 5.3.2 の DEBIAN_FRONTEND を見てください。

5.2.2 基板デバイス

いくつかのアクセシビリティデバイスは、マシンの内部に接続した実際の基板で、ビデオメモリから直接テキストを読みます。動作させるには、fb=false 起動パラメータを用いて、フレームバッファのサポートを無効にしなければなりません。しかし、これにより使用できる言語が減ってしまいます。

5.2.3 高コントラストテーマ

視力の弱いユーザのために、インストーラは、より見やすい高コントラストの配色を採用したテーマを使用できます。これを有効にするには、起動画面にて d ショートカットを押して「Accessible high contrast」を使うか、または起動パラメータに theme=dark を追加してください。

5.2.4 拡大

視力の弱いユーザ向けに、グラフィカルインストーラでは非常に基本的な拡大機能をサポートしています: Control++ と Control- のショートカットでそれぞれフォントサイズを拡大、縮小します。

5.2.5 Expert モード、Rescue モード、自動化インストール

Expert モードや Rescue モード、自動化インストールの選択はアクセシビリティのサポートも利用できるようになっています。それぞれの選択肢にアクセスするには最初にブートメニューで a を入力して「Advanced options」サブメニューに進む必要があります。BIOS システムを使っている場合 (ブートメニューは一度だけビーブ音を鳴らします) は続いて Enter を押さないといけません。UEFI システムの場合 (ブートメニューは二回ビーブ音を鳴らします) は Enter を押しはいけません。それから、

オプションとして `s` を押すと (UEFI システムではなく BIOS システムの場合は、ここでも続いて `Enter` を押さないといけません)、音声合成が有効になります。それから様々なショートカットが使えるようになります: `x` は expert モードでのインストール、`r` は rescue モード、`a` は自動化インストールです。繰り返しますが、BIOS システムを使っている場合はそれぞれ続いて `Enter` を押さないといけません。

自動化インストールを選択することで、`preseed` を使い完全に自動で Debian をインストールできます。`preseed` のソースはアクセシビリティ機能開始後に入力できます。`preseed` 自体については付録 B で言及しています。

5.2.6 インストールしたシステムのアクセシビリティ

インストールしたシステムのアクセシビリティについての文書は、[Debian Accessibility wiki ページ](#) で利用できます。

5.3 起動パラメータ

起動パラメータとは Linux カーネルのパラメータのことで、一般には周辺機器を適切に扱うために用います。ほとんどの場合、カーネルは周辺機器の情報を自動的に検出します。しかし、場合によっては少々カーネルを助けてあげないといけないこともあるのです。

システムを初めて起動する場合は、デフォルトの起動パラメータを試して (つまりなにもパラメータを設定せずに)、正確に動作するか観察してください。たいていはうまくいくと思います。なにか問題が起こったら、そのハードウェアに関する情報をシステムに伝えるためのパラメータを調べ、あとで再起動します。

多くの起動パラメータの情報は (曖昧なハードウェア用の tips 込みで)、[Linux BootPrompt HOWTO](#) で見つけられます。本節は、最も顕著なパラメータの概要だけを含んでいます。いくつか共通のものは項 5.4 以下に含まれています。

5.3.1 ブートコンソール

起動の際にシリアルコンソールを使うと、通常カーネルはこちらを自動検出します。ただし、シリアルコンソールから起動させたいコンピュータに、ビデオカード (フレームバッファ) とキーボードもついている場合には、カーネルに `console=device` という引数を渡す必要があると思います。`device` は利用したいシリアルデバイスで、通常 `ttyS0` のようになります。

通信速度やパリティといった、シリアルポート用のパラメータを指定する必要があるかもしれません。例えば `console=ttyS0,9600n8`。他の速度としては 57600 や 115200 が代表的です。このオプションを必ず「---」に続けて指定するようにしてください。そうすることで (インストーラでそのブートローダがサポートされている場合) ブートローダでの設定がインストールしたシステムにコピーされるようになります。

インストーラが使用する端末タイプと、端末エミュレータが一致するのを保証するため、パラメータ `TERM=type` を追加できます。インストーラは以下の端末タイプしかサポートしないことに注意してください。`linux`, `bterm`, `ansi`, `vt102`, `dumb` です。`debian-installer` でのシリアルコンソールのデフォルトは、`vt102` です。上記の端末タイプを提供しない IPMI や QEMU/KVM のような仮想化ツールを使用している場合、内部で `screen` セッションを開始できます。これにより、`vt102` によく似た `screen` 端末タイプに変換できます。

5.3.2 Debian Installer パラメータ

インストールシステムは、おそらく便利だと思われる、追加起動パラメータ¹をいくつか認識します。

多くのパラメータは、カーネルコマンドラインオプションの制限を避けたり、パラメータの入力を簡単にするため、「短縮形」を持っています。パラメータに短縮形がある場合、(通常の) 長い形式の後にかっこで囲っています。本マニュアルの例は、通常、短縮形も使用しています。

`debconf/priority (priority)` このパラメータには、表示するメッセージのもっとも低い優先度を設定します。

¹現在のカーネル (2.6.9 以降) では、コマンドラインオプションを 32 個と環境オプションを 32 個使用できます。それを越えると、カーネルはパニックしてしまいます。また、カーネルコマンドライン全体で 255 文字という制限もあります。いずれも、制限を超えた場合は暗黙のうちに切り詰められる可能性があります。

デフォルトのインストールでは、`priority=high` を使用します。優先度が「高」のもの、「重要」のもののメッセージを表示し、「標準」や、「低」のメッセージはスキップします。問題にぶつかった場合、インストーラは必要な優先度に調整します。

起動パラメータに `priority=medium` を追加すると、インストールメニューが表示され、インストールについて、さらに多くの制御を行うことができます。`priority=low` を使った場合は、すべてのメッセージを表示します (expert 起動法と等価)。`priority=critical` の場合は、インストールシステムは重要なメッセージだけを表示し、大騒ぎせずに正しい設定をしようとします。

DEBIAN_FRONTEND この起動パラメータはインストーラで使うユーザインターフェースを制御します。現在有効な設定は以下の通りです。

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

デフォルトのフロントエンドは `DEBCONF_FRONTEND=newt` です。シリアルコンソールでインストールするには、`DEBIAN_FRONTEND=text` とすべきでしょう。専用に調整されている種類のインストールメディアでは限られたフロントエンドしか選択できないようになっている場合がありますが、`newt` フロントエンドと `text` フロントエンドは、デフォルトインストールメディアのほとんどで利用可能です。サポートしているアーキテクチャでは、グラフィカルインストーラが `gtk` フロントエンドを使用します。

BOOT_DEBUG この起動パラメータに 2 を設定すると、インストーラの起動プロセス中に詳細なログを出力します。3 を設定すると、起動プロセスの要所でデバッグ用のシェルが利用できます。(シェルを終了すると起動プロセスを継続します)

`BOOT_DEBUG=0` デフォルトです。

`BOOT_DEBUG=1` 通常よりも詳細です。

`BOOT_DEBUG=2` デバッグ情報を大量に表示します。

`BOOT_DEBUG=3` 詳細なデバッグを行うよう、ブートプロセスの様々な箇所でシェルが実行されます。起動を続けるにはシェルから抜けてください。

log_host, log_port これによりインストーラは、ローカルファイルと同様に、指定したホストとポートを持つリモートの `syslog` に対して、ログメッセージを転送します。指定しない場合、ポートのデフォルト値を、標準 `syslog` ポートの 514 とします。

lowmem インストーラが、利用可能なメモリを元に算出するデフォルト値よりも、より高レベルな低メモリ (`lowmem`) レベルに強制するのに使用できます。有効な値は、1 か 2 です。項 6.3.1.1 もご覧ください。

noshell インストーラが `tty2`, `tty3` の対話シェルを提供しないようにします。物理的セキュリティが限られている、無人インストールの際に便利です。

debian-installer/framebuffer (fb) いくつかのアーキテクチャでは、多くの言語でインストールを行うために、カーネルフレームバッファを使用します。フレームバッファが問題となるシステムの場合、パラメータ `fb=false` によってこの機能を無効にできます。`bterm` や `bogl` に関するエラーメッセージや、真っ暗な画面、インストールが始まって数分後にフリーズがおきたら問題の兆候です。

debian-installer/theme (theme) テーマ (`theme`) はインストーラのユーザインターフェースがどのように見えるか (色、アイコンなど) を決定します。どのテーマが利用できるかはフロントエンドによって異なります。現在、`newt` と `gtk` のフロントエンドには、(デフォルトの見た目のほかに) 目の不自由な方向けにデザインされた「dark」テーマのみがあります。起動時のパラメータに、`theme=dark` と指定してテーマを設定してください (このためのブートメニューのキーボードショートカット `d` もあります)。

netcfg/disable_autoconfig デフォルトで `debian-installer` は、IPv6 オートネゴシエーションや DHCP により、ネットワークの設定を自動検出します。検出に成功すると、確認する機会がなく検出値を変更できません。自動設定が失敗する場合のみ、手動ネットワーク設定を行えます。

ローカルネットワークに IPv6 ルータや DHCP サーバがあるのに、それを回避したい場合 (例: 誤った値を返す等)、ネットワークの自動設定をせず (v4, v6 とも)、手動で情報を入力するには、`netcfg/disable_autoconfig=true` パラメータを使用できます。

`hw-detect/start_pcmcia` PCMCIA サービスが原因で問題が発生する場合、`false` を設定することで、起動しないようにすることができます。いくつかのラップトップコンピュータには、そういう行儀悪さがあることが知られています。

`disk-detect/dmraid/enable` (dmraid) インストーラで、Serial ATA RAID (ATA RAID, BIOS RAID, fake RAID とも呼ばれる) のサポートを有効にする場合 `true` にセットします。このサポートは現在実験中であることに注意してください。追加情報は [Debian Installer Wiki](#) にあります。

`preseed/url` (url) preconfiguration ファイルをダウンロードする URL を指定します。これは自動インストールで使用します。項4.4 を参照してください。

`preseed/file` (file) 自動インストールで読み込む preconfiguration ファイルの PATH を指定します。項4.4 を参照してください。

`preseed/interactive` preseed 中に質問を表示する場合には、`true` を設定します。事前設定ファイルのテストやデバッグに便利でしょう。これは、起動パラメータに渡すパラメータには影響を及ぼしませんが、特殊な文法が使えるようになります。詳細は、項B.5.2 をご覧ください。

`auto-install/enable` (auto) 通常 preseed の前に行われる質問を、ネットワークの設定が終わるまで遅らせます。自動インストールでこのパラメータを使用する際には、項B.2.3 をご覧ください。

`finish-install/keep-consoles` シリアル端末や管理コンソールからインストール中に、通常の仮想コンソール (VT1 から VT6) は、通常 `/etc/inittab` で無効にされています。これを回避するには `true` をセットしてください。

`cdrom-detect/eject` デフォルトで `debian-installer` は、再起動の前にインストールに使用した光学メディアを、自動的に排出します。自動的にそのようなメディアから起動しないようなシステムでは、これは必要ありませんし、特定の状況下では、困ることになる可能性もあります。例えば、光学ドライブがメディアを再び差し込むことができず、手で挿入するようユーザがいなければいけないのに、行うユーザがそこにいないなど。大半のスロットローディング、スリムライン、キャディタイプのドライブは、自動的にメディアをリロードできません。

`false` に設定すると、自動排出を無効にできます。また、システムの初期インストール後に、光学ドライブから自動起動しないことを保証する必要があります。

`base-installer/install-recommends` (recommends) このオプションを `false` にすると、インストール中でも、またインストール後でも、「推奨」パッケージを自動的にインストールしないように、パッケージ管理システムを設定します。項6.3.5 をご覧ください。

これにより、より無駄のないシステムを得られますが、通常可能であると期待する機能が欠落するかもしれないことに注意してください。完全な機能が必要な場合には、推奨パッケージを手でインストールすることになるでしょう。そのためこのオプションは、経験豊富なユーザにしか有用ではないでしょう。

`debian-installer/allow_unauthenticated` デフォルトでは、既知の `gpg` キーで認証されたりポジトリが、インストーラには必要です。この認証を無効にするのに `true` と設定してください。警告: 危険です。お勧めしません。

`rescue/enable` 通常のインストールではなく、レスキューモードを実行する場合、`true` にセットしてください。項8.6 をご覧ください。

5.3.3 起動パラメータで質問に答える

例外的に、インストール中の質問に起動パラメータで答えることができます。これは、特殊な状況でのみ便利です。この方法の概要は、項B.2.2 にあります。特殊な例を以下に示します。

`debian-installer/language` (language), `debian-installer/country` (country), `debian-installer/locale` (locale) インストール中やインストール後で使用する、言語・国・ロケールを指定する方法はふたつあります。

最初の、最も簡単な方法は locale パラメータを渡すことです。言語や国はその値から導かれます。例えば、言語にドイツ語、国にスイスを指定する場合、locale=de_CH とします (インストールしたシステムのデフォルトロケールは de_CH.UTF-8 にセットされます)。制限は、言語、国、ロケールのすべての組み合わせを、この方法で得られるわけではないということです。

ふたつ目のもっと柔軟な選択肢は、language と country を別々に指定することです。この場合、特定のデフォルトロケールをインストールしたシステムに指定するため、任意に locale を加えられます。例: language=en country=DE locale=en_GB.UTF-8

anna/choose_modules (modules) デフォルトではロードされないインストーラコンポーネントを、自動的に読み込むのに使用します。追加コンポーネントの例として、openssh-client-udeb (インストール中に scp コマンドを使用できる) や、ppp-udeb (項D.4 をご覧ください) が便利です。

netcfg/disable_autoconfig IPv6 オートネゴシエーションや DHCP を無効にし、静的ネットワーク設定を強制するには、true と設定します。

mirror/protocol (protocol) デフォルトでインストーラは、Debian のミラーサイトからファイルをダウンロードするのに http プロトコルを使用し、通常の優先度ではインストール中に ftp に変更できません。このパラメータに ftp と設定すると、インストーラに ftp を使用するよう強制できます。一覧から ftp ミラーを選択できず、ホスト名を手入力しなければならないことに注意してください。

tasksel:tasksel/first (tasks) kde-desktop タスクのような、タスク一覧に表示されないタスクを選択するのに使用します。さらなる情報は項6.3.6.2 をご覧ください。

5.3.4 カーネルモジュールへパラメータを渡す

カーネル内にコンパイルされているドライバの場合、カーネルのドキュメントに記載されている方法でパラメータを渡せます。しかし、ドライバがモジュールとしてコンパイルされており、インストールしたシステムの起動時に、インストール時と比べてカーネルモジュールの読み込みが若干異なる場合、通常の方法ではモジュールにパラメータを渡せません。代わりに、インストーラが認識できる特殊文法を使って適切な設定ファイルにパラメータを格納しなければなりません。パラメータは実際にモジュールをロードする際に利用されます。パラメータは自動的にインストールしたシステムに伝播します。

モジュールにパラメータを渡さなければならないというのは、本当にまれな状況だということに注意してください。ほとんどの場合、カーネルはシステムにあるハードウェアから得られる値を検出し、適切な値を設定してくれます。しかしある状況下では、未だにパラメータを手で設定しなければなりません。

モジュールにパラメータを設定する文法は以下のようになります。

```
module_name.parameter_name=value
```

1 つないし複数のモジュールに、複数のパラメータを渡す場合は繰り返してください。例えば、古い 3Com のネットワークインターフェースカードで BNC (coax) を使用し、IRQ 10 を設定する場合は、以下のようにします。

```
3c509.xcvr=3 3c509.irq=10
```

5.3.5 カーネルモジュールのブラックリスト化

時には、カーネルや udev が自動的にモジュールを読み込むのを防ぐために、ブラックリストに載せる必要があるかもしれません。目的の 1 つには、特定のモジュールが、あなたのハードウェアで問題を起こす場合が挙げられます。またカーネルに、同じデバイスに対して複数の異なるドライバがある場合もあります。ドライバが衝突したり、間違ったドライバを先に読み込んでしまうと、デバイスが正しく動作しない原因となります。

module_name.blacklist=yes といった文法でモジュールをブラックリストに指定できます。これでそのモジュールが /etc/modprobe.d/blacklist.local にあるブラックリストに指定され、インストール中・インストールしたシステムの双方で、ブラックリストが有効になります。

インストールシステム自体が、モジュールをまだ読み込んでいる可能性があることに注意してください。エキスパートモードでインストールを行い、ハードウェア検出時にモジュールの一覧からモジュールの選択を外すことで、モジュールの読み込みを防げます。

5.4 インストールプロセスのトラブルシューティング

5.4.1 光学メディアの信頼性

時々、特に古いドライブの場合、光学ディスクからのインストーラの起動に失敗するかもしれません。また、インストーラは(そのディスクから起動しても)そのディスクを認識しなかったり、インストール中、ディスクの読み込みでエラーを返す可能性もあります。

この問題の原因は様々なことが考えられます。一般的な問題を挙げて、一般的な対処法を提供することしかできません。後はあなた次第です。

まずはじめに試すのは、以下の2点です。

- ディスクが起動しない場合、正しく挿入されているか、汚れていないかを確認してください。
- インストーラがディスクを認識しない場合、次にインストールメディアの検出とマウントを試してください。非常に古いCD-ROMドライブの、DMAに関する問題は、この方法で解決することが知られています。

それでも動作しない場合、以下の節にあることを試してみてください。ほとんどの(でもすべてではない)提案はCD-ROMとDVDで有効です。

光学ディスクからインストールができなければ、他のインストール方法も試してみてください。

5.4.1.1 共通の問題

- 古いCD-ROMドライブの中には、昨今のCDライターで使用するような、高速で焼いたディスクからの読み込みをサポートしていない物があります。
- 非常に古いCD-ROMドライブの中には、「ダイレクトメモリアクセス」(DMA)が有効だと、正しく動作しない物があります。

5.4.1.2 調査および問題解決の方法

光学ディスクが起動に失敗したら、以下のことを試してください。

- BIOS/UEFIが光学ディスクからの起動をきちんとサポートしているか(非常に古いシステムでのみ問題あり)、BIOS/UEFIでそのようなメディアからの起動を有効にしているかをチェックしてください。
- ISOイメージをダウンロードした場合、イメージをダウンロードしたのと同じ場所にあるMD5SUMSに記載されているmd5sumと同じかどうかチェックしてください。

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

次に、焼いたディスクのmd5sumと一致するかどうかチェックしてください。以下のコマンドで行います。ディスクから正しいバイト数を読み込むのにイメージのサイズを利用します。

```
$ 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
```

インストーラの起動が成功した後で、ディスクを検出しない場合、単純にリトライするだけで解決することもあります。光学ドライブが複数ある場合、他のドライブに変えてみてください。それでも動作しなかったり、ディスクを認識しても読み込みエラーが発生する場合は、以下のことを試してみてください。Linuxの基礎知識が少し必要です。コマンドを実行するには、まず第2仮想コンソール(VT2)に切り替えて、シェルを有効にしてください。

- エラーメッセージをチェックするのにVT4に切り替えたり、/var/log/syslogの内容を(エディタのnanoを使用して)表示してください。その後、dmesgの出力でもチェックできます。

- 光学ドライブを認識したかを `dmesg` の出力でチェックしてください。以下のように見えます。(行は連続している必要はありません)

```
ata1.00: ATAPI: MATSHITADVD-RAM UJ-822S, 1.61, max UDMA/33
ata1.00: configured for UDMA/33
scsi 0:0:0:0: CD-ROM MATSHITA DVD-RAM UJ-822S 1.61 PQ: 0 ANSI: 5
sr0: scsi3-mmc drive: 24x/24x writer dvd-ram cd/rw xa/form2 cdda tray
cdrom: Uniform CD-ROM driver Revision: 3.20
```

以上のように見えなければ、ドライブを接続したコントローラを認識できないか、おそらくまったくサポートされていません。コントローラに必要なドライバが分かっているならば、`modprobe` を使い、手で読み込むのを試せます。

- `/dev/` にある光学ドライブのデバイスノードをチェックしてください。上の例では、`/dev/sr0` になっています。`/dev/cdrom` にもあるかもしれません。
- 光学ディスクがすでにマウントされていないか、`mount` コマンドでチェックしてください。マウントされていない場合は、手でマウントしてください。

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

上記のコマンド後に、エラーメッセージがでるかチェックしてください。

- DMA が有効か、以下のようにチェックしてください。

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

`using_dma` の後、初めの列にある「1」は、有効という意味です。その場合以下のように無効にしてください。

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

確実に、光学ドライブに一致するデバイスのディレクトリで操作してください。

- インストール中に何か問題があれば、インストーラメインメニューの下の方にある、インストールメディアの整合性チェックを行ってください。ディスクが確実に読める場合、このオプションを一般的なテストとして使用できます。

5.4.2 起動設定

ブートプロセスの最中にカーネルがハングしたり、搭載されている周辺機器やドライブが正確に認識されないなどの問題が起こったら、まず項5.3の説明に従って起動パラメータを確認してください。

場合によっては、デバイスのファームウェアがないために動作しないことがあります(項2.2や項6.4を参照)。

5.4.3 カーネルの起動時メッセージの意味

ブートシーケンスの途中で、`can't find something` (〜が見つからない), `something not present` (〜が存在しない), `can't initialize something` (〜を初期化できない), `this driver release depends on something` (このドライバには〜が必要だ)などのメッセージがたくさん出力されることがあります。これらのメッセージのほとんどは無害です。これらが出力される理由は、インストールシステムのカーネルが、いろいろな周辺デバイスのできるだけ多くに対応しようとしているからです。そのため、OSが実際には存在しない周辺機器を探すことになるので、文句を吐くわけです。システムがしばらく止まったように見えることもあります。これはデバイスが反応するのを待っているために起こるものです(実際にはそのデバイスは存在しないので、止まってみえるわけです)。システムの起動に要する時間が堪えがたいほど長い場合は、後で自前のカーネルを作ることもできます(項8.5参照)。

5.4.4 インストールで発生した問題の報告

最初の起動段階は通過したのに、インストールが完了できなかった場合は、メニューからデバッグログを保存を選択するといいかもかもしれません。インストーラからのシステムのエラーログや設定情報をストレージメディアに格納したり、web ブラウザでダウンロードしたりできるようになります。この情報は、何が間違っていてどのように修正するか、といった手がかりを示しているかもしれません。バグ報告を送る際に、バグ報告にこの情報を付けることができます。

その他のインストールメッセージは、インストール中では `/var/log/` で、インストールしたシステムが起動した後では `/var/log/installer/` にはありません。

5.4.5 インストールレポートの送信

まだ問題がある場合には、インストールレポートをお送りください。また、インストールが成功したときのインストールレポートもお送りください。そうすると、たくさんのハードウェア設定情報を手に入れることができます。

あなたのインストールレポートは、Debian バグ追跡システム (BTS) で公開され、公開メーリングリストに転送されることに留意してください。必ず、公開されても問題ない e-mail アドレスを使用してください。

動作する Debian システムがある場合、インストールレポートを送る最も簡単な方法は以下のようになります。installation-report と reportbug パッケージをインストール (`apt install installation-report reportbug`) し、項 8.4.2 で説明しているように reportbug を設定して、reportbug installation-reports を実行してください。

代わりに、インストールレポートを記入する際には、以下のテンプレートも使用できます。テンプレートの空欄を埋めた上で、installation-reports 疑似パッケージのバグ報告として、submit@bugs.debian.org 宛にお送りください。

```
Package: installation-reports

Boot method: <インストーラの起動方法は? CD?/DVD? USB メモリ? ネットワーク?>
Image version: <イメージをダウンロードした URL 全体がベストです>
Date: <インストールした日時>

Machine: <マシンの説明 (例 IBM Thinkpad R32)>
Processor:
Memory:
Partitions: <df -TI の結果; 生のパーティションテーブルが望ましいです>

Output of lspci -nn and lspci -vnn:

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

Initial boot:           [ ]
Detect network card:    [ ]
Configure network:      [ ]
Detect media:           [ ]
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:

<インストールに関すること。初期インストール時に抱いた感想、コメント、アイデアなどがあればそれらもお書きください>
```

バグ報告の際には、カーネルがハングした直前に表示されたカーネルメッセージを添えて、何が問題なのかを説明してください。また、問題が起きるまでにシステムに対して行ったことも記述してください。

Chapter 6

Debian インストーラの使用方法

6.1 インストーラの動作

Debian Installer は各インストールタスクを実行するために、たくさんの特定用途コンポーネントから成ります。各コンポーネントは、必要ならユーザに質問をして、そのタスクを実行します。この質問には優先度が設定されており、この優先度はインストーラの起動時に設定することができます。

デフォルトのインストールでは、不可欠な (優先度が高い) 質問しかしません。これにより、ユーザの入力をほとんど行わず、高度な自動インストールを行うことができます。コンポーネントは自動的に順番に実行されます。どのコンポーネントを実行するかは、主に使用するインストール法やハードウェアに左右されます。インストーラは、質問しない項目についてはデフォルト値を使用します。

問題がある場合はエラー画面を表示し、インストーラメニューに、代替アクションを選択するように表示するかもしれません。いずれも問題なければ、ユーザはインストールメニューを目にすることはなく、単に順番に各コンポーネントの質問に答えて行くだけでしょう。重大なエラー通知は優先度を「重要」に設定されているため、常に表示されます。

インストーラが使用するデフォルト値は、debian-installer の起動時にパラメータで渡して指定できます。たとえば、強制的に静的ネットワーク設定をしたい場合 (デフォルトでは可能なら IPv6 オートネゴシエーションと DHCP を利用)、ブートパラメータに netcfg/disable_autoconfig=true を加えられます。利用できるオプションは項 5.3.2 を参照してください。

パワーユーザは、メニュードリブンインターフェース (自動で順に各ステップを実行するインストーラではなく、ユーザが各ステップを制御する) の方が、満足するかもしれません。手動 (メニュー駆動) でインストーラを使用するには、起動引数に priority=medium を加えてください。

ハードウェアをインストールする際に、オプションをカーネルモジュールへ渡す必要がある場合、「エキスパート」モードでインストーラを起動する必要があります。これは、インストーラを起動するコマンドに expert を使用する、あるいは起動引数に priority=low を加えることで行います。エキスパートモードでは debian-installer をフルコントロールできます。

このアーキテクチャでは、インストーラをキャラクタベースユーザインターフェースを使用します。グラフィカルユーザインターフェースは現在利用できません。

キャラクタベース環境では、マウスはサポートしていません。ここでは、様々なダイアログでナビゲートするキーを紹介します。ボタンや選択肢が表示されている間は、Tab キーや右矢印は「前方」へ、Shift-Tab や左矢印は「後方」へ移動します。上下矢印は、スクロールするリスト内の項目を選択し、またリスト自体もスクロールさせます。さらに、長いリストでは、タイプした文字で始まる項目に直接スクロールしますし、リストのスクロールに Pg-Up や Pg-Down も使用できます。スペースバーは、チェックボックスのような項目を選択します。選択肢を有効にするには Enter を使用してください。

ダイアログには、追加ヘルプ情報があるものがあります。ヘルプがある場合、画面の最下行に表示されています。F1 キーを押してアクセスできます。

エラーメッセージとログは第 4 コンソールにリダイレクトされます。このコンソールへは左 Alt-F4 (左 Alt キーを押しながら F4 ファンクションキーを押す) を押してアクセスしてください。左 Alt-F1 で、メインのインストーラプロセスに戻ります。

これらのメッセージは /var/log/syslog で見つけることもできます。インストールの後、このログはあなたの新システムの /var/log/installer/syslog にコピーされます。他のインストールメッセージは、インストール中には /var/log/ に、インストールしたシステムでコンピュータが起動した後は /var/log/installer/ にあります。

6.1.1 グラフィカルインストーラの使用法

グラフィカルインストーラは基本的にはテキストベースのインストーラと同様に動作するため、インストールプロセスを通してこのマニュアルの案内を利用できます。

マウスよりもキーボードを使いたい場合には、2つのことを知っておく必要があります。省略されている一覧(例えば大陸内の国の選択に利用されています)を展開するのに + や - キーを利用できます。項目を複数選択できる質問(例えばタスクの選択)では選択後にタブで Continue ボタンに移動する必要があります。Enter を押した場合は Continue が作動するのではなく選択が切り替わります。

ダイアログに追加ヘルプ情報がある場合は、Help ボタンが表示されます。ヘルプ情報にはこのボタンを作動させるか F1 キーを押してアクセスできます。

別のコンソールに切り替えるには X ウィンドウシステムと全く同じように Ctrl キーも使う必要があります。例えば VT2 (第一のデバッグ用シェル) に切り替えるには Ctrl-左 Alt-F2 を使います。グラフィカルインストーラ自体は VT5 で実行されているので、左 Alt-F5 を使うと戻れます。

6.2 コンポーネント入門

ここではインストーラコンポーネントを各コンポーネントの簡単な説明を添えて一覧します。特定のコンポーネントを使用するにあたり、知る必要があるかもしれない詳細は項6.3にあります。

main-menu インストーラの操作中にユーザにコンポーネントのリストを見せ、選択されたコンポーネントを起動します。main-menu では質問の優先度が「中」に設定されています。そのため、優先度が「高」や「重要」(デフォルトは「高」)に設定されている場合は、メニューを見ることはありません。一方、あなたの入力が必要なエラーが起きた場合、その問題を解決するために、質問の優先度が一時的に格下げされるかもしれません。その場合、メニューが表示される可能性があります。

現在実行しているコンポーネントから抜けるために、Go Back ボタンを繰り返し選択してメインメニューに戻れます。

localechooser インストール中・インストールしたシステムの、地域オプション(言語、国、ロケール)の選択を行います。インストーラは選択した言語でメッセージを表示しますが、その言語でのメッセージの翻訳が完了していない場合は、英語で表示します。

console-setup キーボード(レイアウト)のリストを表示します。お持ちのキーボードモデルに一致するものを選択してください。

hw-detect システムのほとんどのハードウェアを自動検出します。これには、ネットワークカード、ディスクドライブ、PCMCIA が含まれます。

cdrom-detect Debian インストールメディアを探しマウントします。

netcfg インターネットへ通信できるように、コンピュータのネットワーク接続を設定します。

iso-scan ハードディスクにある ISO イメージ(.iso ファイル)を探します。

choose-mirror Debian アーカイブミラーのリストを表示します。インストールするパッケージの取得元を選択できるでしょう。

cdrom-checker インストールメディアの整合性チェック。これにより、インストールイメージが壊れていないか自分で保証できます。

lowmem lowmem はシステムの搭載するメモリが少ないかを確認し、少なければ debian-installer の不必要な部分を、メモリから(いくつかの機能を犠牲にして)削除する様々なトリックを行います。

anna Anna's Not Nearly APT. (Anna はちっとも APT (適切) じゃない) 選択したミラーサーバやインストールメディアから、パッケージを取得してインストールします。

user-setup root パスワードの設定や、root 以外のユーザの追加を行います。

clock-setup システム時計を更新して、時計を UTC に合わせるかどうかを決定します。

tzsetup あらかじめ選択した場所を元に、タイムゾーンを選択します。

- partman** システムの内蔵ディスクを分割し、選択したパーティションのファイルシステムを作成し、マウントポイントにそのファイルシステムをマウントすることができます。完全自動モードや LVM サポートといったさらに面白い機能があります。これは Debian での好ましいパーティション分割ツールです。
- partitioner** システムのディスクを分割することができます。あなたのコンピュータのアーキテクチャに最適な、パーティション分割プログラムが選ばれます。
- partconf** パーティションのリストを表示します。また、選択したパーティションにファイルシステムを作成します。
- partman-ivm** LVM (Logical Volume Manager) の設定について、ユーザの補助を行います。
- partman-md** ソフトウェア RAID (Redundant Array of Inexpensive Disks) の設定をユーザに許可します。このソフトウェア RAID は、新しめのマザーボードに見られる、安い IDE (疑似ハードウェア) RAID コントローラより通常優秀です。
- base-installer** 再起動時に、コンピュータが Debian GNU/Linux として動作するための、もっとも基本的なパッケージセットをインストールします。
- apt-setup** インストーラを起動したメディアを元に、ほとんど自動で apt の設定を行います。
- pkgselect** 追加ソフトウェアをインストールするのに **tasksel** を使用します。
- os-prober** コンピュータに現在インストールされている OS を検出し、この情報を (bootloader のスタートメニューに発見した OS を加える機能を提供する) **bootloader-installer** へ渡します。これは、起動時にどの OS で起動するかを、ユーザが簡単に決める方法です。
- bootloader-installer** 様々なブートローダインストーラがそれぞれ、ハードディスクにブートローダプログラムをインストールします。これは、USB メモリや CD-ROM を使用しないで Linux を起動するのに必要です。ブートローダの多くは、コンピュータが起動するごとに代替オペレーティングシステムを選ぶことができます。
- shell** メニューから、もしくは第 2 コンソールで shell を実行できます。
- save-logs** 後で Debian 開発者へ、インストーラソフトウェアの障害を正確に報告するために、障害に遭遇した際の、USB メモリ、ネットワーク、ハードディスク、その他メディアに情報を記録する方法を提供します。

6.3 それぞれのコンポーネントの使用方法

本節では、各インストーラコンポーネントの詳細について述べていきます。コンポーネントは、ユーザに認識できる段階へグループ化されました。それらは、install 中に現われる命令の形で示されます。すべてのモジュールを、インストール時に使用するとは限らない、ということに注意してください。どのモジュールを実際に使用するかは、使用するインストール法やハードウェアに左右されます。

6.3.1 Debian インストーラのセットアップとハードウェアの設定

Debian インストーラが起動して、最初の画面が表示されているとしましょう。このとき、**debian-installer** の機能はまだ制限されています。ハードウェア、希望する言語、実行するタスクなどに関しても、まだ知りません。しかし心配しないでください。**debian-installer** は非常に賢いので、ハードウェアの自動検出をしたり、コンポーネントの残りを見つけたり、高性能なインストールシステムに自分自身をアップグレードすることができます。しかし、(希望する言語、キーボードレイアウト、使用するネットワークミラーの選択のように) いくつかのタスクでは自動で決められませんので、**debian-installer** を助けてあげる必要があります。

この段階で **debian-installer** がハードウェア検出を数回行うことに気がつくことでしょう。最初の検出では、インストーラのコンポーネントをロードするのに欠かせないハードウェア (例: CD-ROM ドライブやネットワークカード) を認識することが目標です。初回の実行ですべてのドライバが使用可能になるわけではないので、ハードウェア検出をこのプロセスの後で繰り返す必要があります。

ハードウェア検出の間、**debian-installer** はシステムにあるハードウェアデバイスのドライバが、ファームウェアを読み込む必要があるかをチェックします。ファームウェアが必要なのに利用できない場合は、リムーバブルメディアから見つからないファームウェアを読み込むダイアログを表示します。詳細は項 6.4 をご覧ください。

6.3.1.1 利用可能なメモリのチェック / 低メモリモード

debian-installer がまず行うことのひとつが、利用可能なメモリをチェックすることです。利用可能なメモリに制限がある場合、このコンポーネントは、システムに Debian GNU/Linux をインストールできるように、インストールプロセスにいくらかの変更を加えます。

インストーラで消費メモリを抑えるには、翻訳を無効にすることです。これは、英語でしかインストールできないということでもあります。もちろん、インストール完了後に、インストールしたシステムを地域化することができます。

これで充分でなければ、インストーラは、基本的なインストールを完了するのに必須なコンポーネントのみを読み込み、メモリ消費をさらに抑えようとします。これはインストールシステムの機能を制限します。手動で機能を追加する手段を提供していますが、それによりさらにメモリを消費し、結果インストールに失敗する可能性を考慮する必要があります。

インストーラが低メモリモードで動作する場合、比較的大きな swap パーティション (64-128MB) を作成するのをお勧めします。swap パーティションは仮想メモリとして使用され、システムで利用できるメモリの量を増やします。インストーラは、インストールプロセスで可能な限り早く swap を有効にします。swap を使用すると、ディスク負荷が増加し、システムのパフォーマンスが低下する事に注意してください。

こういった措置にもかかわらず、まだシステムがフリーズしたり、予期しないエラーが発生したり、システムがメモリ範囲外で動作 (VT4 と syslog に「Out of memory」メッセージを出力) して、プロセスがカーネルに強制終了される可能性があります。

例えば、swap スペースが不十分な場合、低メモリモードで大きな ext3 ファイルシステムを作成すると、エラーを報告します。swap をもっと大きくしてもだめな場合、ext2 (インストーラの必須コンポーネント) で作成してください。ext2 パーティションをインストール後に ext3 に変更できます。

インストーラに項 5.3.2 で説明している「lowmem」ブートパラメータを使用すると、利用可能なメモリを元にした lowmem レベルよりも高いレベルにできます。

6.3.1.2 地域オプションの選択

ほとんどの場合、最初の質問はインストール中とインストールしたシステム双方の、地域オプションの選択に関することです。地域オプションは、言語、場所、ロケールからなっています。

異なるダイアログの翻訳が利用できるなら、選んだ言語をインストールプロセスの残りで使用できます。選択した言語で、有効な翻訳が利用できなければ、インストーラは自動的に英語になります。

選択した地理的位置 (ほとんどの場合で国) は、インストールプロセスの後半で、デフォルトのタイムゾーンの抽出と、その国に適切な Debian ミラーサイトの抽出に使用します。言語と国は、ともにシステムのデフォルトロケールの決定や、正しいキーボードレイアウトの選択を支援します。

最初に好みの言語を選択することになります。言語名は英語 (左側) と原語 (右側) の両方で表示しています。右側の名称は、その言語そのもので書かれた表記です。このリストは英語名順に並んでいます。このリストの先頭には言語の代わりに「C」ロケールを選択する追加オプションもあります。「C」ロケールを選択するとインストールプロセスは英語で行われます。また、インストールしたシステムには locales パッケージがインストールされず、いずれの地域もサポートしません。

次は地理的な場所を選択するよう求められます。言語選択時に、その言語が複数の国で公用語とされている場合¹、その国だけのリストを表示します。そのリストにない国を選択する場合、その他 (最後の選択肢) を選択してください。すると、大陸のリストを表示します。大陸を選択すると、関連する国のリストを表示します。

言語に対して国がひとつしかない場合、国のリストには、その国が属する大陸か地域を表示し、その国をデフォルトで選択状態とします。別の大陸にある国を選択したければ、Go Back を選択してください。

注意



インストールしたシステムのタイムゾーンを設定するため、あなたが住む、ないし設置する国を選択することが重要です。

ロケールが定義されていない言語と国の組み合わせを選択して、その言語に複数のロケールが存在

¹ 技術的な用語として、言語に対し国コードが異なるぶんだけ、複数のロケールが存在します。

する場合、インストールしたシステムのデフォルトロケールを、その中から選ぶこととなります²。そうでなければ、デフォルトロケールは選択した言語と国をもとに選択されます。

前の段落で説明したように選択されたデフォルトロケールは、文字コードに UTF-8 を使用します。

優先度低でインストールしている場合、追加ロケール (いわゆる「レガシー」ロケール³を含む) を選択して、インストールしたシステム用に生成できます。この場合、選択したロケールの中からどれをデフォルトロケールにするか尋ねられます。

6.3.1.3 キーボード選択

キーボードは、しばしば言語で使用する文字に合わされています。使用しているキーボードに一致するレイアウトを選択するか、希望のキーボードレイアウトが表示されなければ、近いものを選択してください。いったんシステムのインストールが完了すれば、もっと広い範囲からキーボードレイアウトを選ぶことができます (インストールが完了した後に、`root` で `dpkg-reconfigure keyboard-configuration` を実行してください)。

希望のキーボードにハイライトを移動させて、Enter を押してください。ハイライトの移動には矢印キーを使用してください。どの言語のキーボードでも同じ場所にあるため、キーボードの設定に依存しません。

6.3.1.4 Debian Installer iso イメージの検索

`hd-media` でインストールを行う場合、インストールするファイルの残りを得るために、Debian Installer iso イメージを見つけてマウントする必要があるでしょう。iso-scan コンポーネントはその名の通り行います。

初めに iso-scan は、既知のファイルシステムがあるブロックデバイス (例えばパーティションや論理ボリューム) を自動的にすべてマウントし、.iso (もっと言えば .ISO) で終わるファイル名を順番に検索します。初回の試行でルートディレクトリ中、およびそのサブディレクトリ内しか検索しないことに注意してください (つまり `/whatever.iso` や `/data/whatever.iso` を検出しますが、`/data/tmp/whatever.iso` は検出しないということです)。iso イメージの検出後、iso-scan は、そのイメージが有効な Debian iso イメージであるか否かを決定するため、その内容をチェックします。前者の場合は完了しますが、後者の場合 iso-scan は別のイメージを探します。

インストーラ iso イメージを探す試行が失敗する場合、iso-scan はより徹底的に検索するか確認します。このパスは最上位のディレクトリのみ調査しませんが、実際にファイルシステム全体を全探索します。

iso-scan がインストーラ iso イメージを検出しない場合、元の OS を起動し直して、イメージが (.iso で終わる) 正しい名前になっているか、`debian-installer` が認識できるファイルシステムに配置しているか、(チェックサムを検証して) 壊れていないかチェックしてください。Unix の経験があるユーザは、再起動せずに第 2 コンソール上でチェックできます。

ISO イメージをホストするパーティション (またはディスク) はインストーラーで利用されているので、インストールプロセス中に再利用できない点にご注意ください。この回避策としては、十分なシステムメモリがある場合ですが、インストーラーが ISO イメージをマウントする前に RAM にコピーできます。これは `low priority` の `iso-scan/copy_iso_to_ram` `debconf` 設定で管理されています (メモリ要件が合致した場合のみ尋ねられます)。

6.3.1.5 ネットワークの設定

このステップに入って、ネットワークデバイスが 1 つ以上あることをシステムが検出すると、どのデバイスを 主要 (つまりインストールに使用する) ネットワークインターフェースとするか質問されます。その他のインターフェースはここでは設定しません。インストールが完了したところで、さらにインターフェースを設定できるでしょう。interfaces(5) man ページを参照してください。

6.3.1.5.1 自動ネットワーク設定

デフォルトでは、`debian-installer` はコンピュータのネットワークを、可能な限り自動的に設定しようとしています。自動設定に失敗した場合、ネットワークケーブルが繋がっていないことから、自動設定用のインフラが見つからないことまで、幅広い原因が考えられます。エラー発生時に何が起きたかを確

²優先度が中や低では、選択した言語で有効なロケールの中から、常に好みのものを選択することになります (複数ある場合)。

³レガシーロケールとは、UTF-8 を使用しないけれど、ISO 8859-1 (西欧言語で利用) や EUC-JP (日本語で利用) といった文字エンコードを使用する旧標準の一種です。

認するには、4 番目のコンソールに表示するエラーメッセージをチェックしてください。いずれの場合も、再実行するか、手動設定を実行するか、を質問されます。自動設定に使用するネットワークサービスは、時にそのレスポンスが遅いことがあります。そのため、適切な場所にあるかどうか確認してから、自動設定を再実行してください。繰り返し自動設定に失敗する場合、手動でネットワークの設定を行なってください。

6.3.1.5.2 手動ネットワーク設定

ネットワークの手動設定では、ネットワークについて、いくつか質問をしてきます。特に、IP アドレス、ネットマスク、ゲートウェイ、ネームサーバのアドレス、ホスト名について質問します。さらに、無線ネットワークインターフェースがあるなら、無線 ESSID (「無線ネットワーク名」) と WEP キーや、WPA/WPA2 パスフレーズを質問します。項 3.3 より回答を入力してください。

注意



わかりやすいかどうかはともかく、技術的詳細は以下のようになります。このプログラムは、システムの IP アドレスとネットマスクのビット積を、ネットワーク IP アドレスとします。デフォルトのブロードキャストアドレスは、システムの IP アドレスと、ネットマスクのビット否定とのビット和から計算します。同様にゲートウェイも推測します。どのような値を設定するのかよくわからなければ、デフォルト値を使用してください。一度システムをインストールした後で、必要に応じて `/etc/network/interfaces` を編集して変更できます。

6.3.1.5.3 IPv4 と IPv6

Debian GNU/Linux 7.0 (「Wheezy」) 以降から、`debian-installer` は IPv6 を「クラシックな」IPv4 と同様にサポートしています。IPv4 と IPv6 のすべての組み合わせ (IPv4 のみ、IPv6 のみ、デュアルスタック構成) をサポートします。

IPv4 の自動設定は、DHCP (ダイナミックホストコンフィギュレーションプロトコル) を用いて行います。IPv6 の自動設定は、NDP (リカーシブ DNS サーバ (RDNSS) の任務に含まれる近隣者発見プロトコル) を用いたステートレス自動設定と、DHCPv6 を用いたステートフル自動設定、ステートレス・ステートフル混合 (アドレスの設定を NDP で、追加パラメータを DHCPv6 で行う) 自動設定をサポートします。

6.3.2 ユーザとパスワードのセットアップ

クロックの設定直前に、インストーラは「root」アカウントや、最初のユーザアカウントのセットアップを行います。その他のユーザアカウントは、インストール完了後に作成してください。

6.3.2.1 root パスワードの設定

root アカウントは、ログインするとシステムのすべてのセキュリティ保護をバイパスしてしまうので、スーパーユーザとも呼ばれています。root アカウントはシステム管理のみに使用し、可能な限り短時間使用するのみにすべきです。

作成するパスワードは、少なくとも 6 文字以上で、大文字小文字、カンマやピリオドを混ぜるべきです。root パスワードを設定するときには、強力なアカウント故に特別注意を払ってください。辞書にある単語や推測される個人情報を使用するのは避けてください。

誰であっても、root パスワードが必要だと言う人がいる場合には、殊更に用心してください。他のシステム管理者と共に機械の管理をしているのでなければ、root パスワードを教える必要は、通常決してありません。

ここで「root」ユーザーのパスワードを指定しなかった場合ですが、このアカウントは無効になりますが、新しいシステム上で管理作業が実施できるよう、後ほど `sudo` パッケージがインストールされます。デフォルトでは、システムで作成された最初のユーザーが root になるのに `sudo` コマンドを使えます。

6.3.2.2 一般ユーザの作成

システムは、この時点で一般ユーザアカウントを作成するかどうか質問します。このアカウントを、個人でログインする場合のメインアカウントにするべきでしょう。root アカウントを日常的に使用したり、個人的な用途でログインするべきではありません。

なぜいけないのでしょうか? root 権限を使用しないようにする理由のひとつは、root により簡単に取り返しのつかない損害を与えられるということです。他には、だまされてトロイの木馬(あなたに隠れ、スーパーユーザ権限を利用してシステムに感染するプログラム)を動かしてしまうということもあり得ます。UNIX システム管理に関するいずれの良書でも、この件に関して詳細に扱っています。今までご存じなければ、ご一読ください。

まず初めに、ユーザのフルネームの入力を求められます。次にユーザアカウントの名前を求められます。一般的にファーストネームか、必要十分な名前に似た何かがデフォルトになります。最後にこのアカウントのパスワードを求められます。

インストール後いつでも、別のアカウントを作成する場合は、adduser コマンドを使用してください。

6.3.3 時計とタイムゾーンの設定

インストーラはまず、正しいシステム時計を設定するため、インターネットのタイムサーバに (NTP プロトコルを利用して) 接続しようとします。これが成功しない場合、インストールシステムが起動したときのシステム時計を正しいと見なします。インストールプロセス中に、手動でシステム時計を設定することはできません。

インストール処理のはじめの方で選択した所在地をもとに、その場所に関連するタイムゾーンの一覧を表示します。あなたの所在地にタイムゾーンがひとつしかなく、デフォルトインストールを行っている場合、システムは一覧を表示せず、そのタイムゾーンであると仮定します。

エキスパートモードや優先度中でインストールしている場合、タイムゾーンに「協定世界時」(UTC)を使用する、という選択肢が追加されます。

何らかの理由で、インストールしたシステムのタイムゾーンを、選択した場所とは異なるものにした場合は、2つの選択肢があります。

1. シンプルな方法は、インストールが完了し、新しいシステムが起動した後で、異なるタイムゾーンを選択することです。以下のようなコマンドになります。

```
# dpkg-reconfigure tzdata
```

2. その他には、インストールシステムの起動時に、パラメータ `time/zone=value` を渡すと、インストールの最初からタイムゾーンを設定できます。もちろん値は妥当なタイムゾーン (例えば `Europe/London` や `UTC`) であるべきです。

自動インストール用に、`preseed` を用いて、タイムゾーンをお好みの値に設定できます。

6.3.4 パーティションの分割とマウントポイントの選択

最後のハードウェア検出が完了した時点で、`debian-installer` はユーザのニーズ通りにカスタマイズされ、実際の作業ができるような、準備万端の状態にあります。本節のタイトルが表すように、以下、少数のコンポーネントの主なタスクは、ディスクのパーティションを分割し、ファイルシステムを作成し、マウントポイントを割り当てて、LVM や RAID、暗号化デバイスのような密接に関係する件のオプション設定を行うことです。

パーティション分割に不安があったり、詳細を知りたければ、付録Cをご覧ください。

最初に、ドライブのすべてか、またはドライブの有効な空き領域を、自動的にパーティション分割するか選択する機会が与えられます。これを「ガイド」パーティション分割とも呼びます。自動分割を望まなければ、手動を選んでください。

6.3.4.1 サポートするパーティション分割オプション

`debian-installer` で使用するパーティション分割ツールは、かなり万能です。これにより、さまざまなパーティションテーブル、ファイルシステム、高度ブロックデバイスを用いて、たくさんの異なるパーティション構成を作成できます。

厳密に、どのオプションが利用できるかは、主にアーキテクチャに依存しますが、その他の要因もあります。例えば、内部メモリが制限されたシステムでは、いくつかのオプションは使用できないで

しょう。さらにデフォルトも変わるかもしれません。例えば、大容量ハードディスクに対する、デフォルトのパーティションテーブルのタイプは、より小さいハードディスクのものとは異なっている場合があります。debconf 優先度が中ないし低でインストールしているときのみ、いくつかのオプションを変更できます。もっと高い優先度の場合は、実用的な値がデフォルトで使用されます。

インストーラは、さまざまな形の高度なパーティションやストレージデバイスを (ほとんどの場合組み合わせて)、サポートします。

- 論理ボリュームマネージメント

- ソフトウェア RAID

サポートしている RAID レベルは 0, 1, 4, 5, 6, 10 です。

- 暗号化

- マルチパス (実験的)

情報は [私たちの Wiki](#) をご覧ください。現在のところ、マルチパスはインストーラ起動時に有効にした場合にのみ利用できます。

以下のファイルシステムをサポートしています。

- ext2r0, ext2, ext3, ext4

ほとんどの場合、デフォルトのファイルシステムに ext4 が選択されています。ガイドパーティション分割を使用する際、/boot パーティションのデフォルトには ext2 が選択されます。

- jfs (全アーキテクチャで使用できるわけではありません)

- xfs (全アーキテクチャで使用できるわけではありません)

- reiserfs (オプション。全アーキテクチャで使用できるわけではありません)

Reiser ファイルシステムは、もはやデフォルトではサポートされません。インストーラが、中ないし低 debconf 優先度で動作させると、partman-reiserfs コンポーネントを選択して有効にできます。バージョン 3 のみサポートします。

- jffs2

フラッシュメモリを読むために、ある種のシステムで使用します。新しい jffs2 パーティションは作成できません。

- FAT16, FAT32

6.3.4.2 ガイドパーティション分割

ガイドパーティション分割を選択した場合、選択肢が 3 つあります。ハードディスクに直接パーティションを作成する (クラシック) 方法、論理ボリューム管理 (LVM) を利用する方法、暗号化 LVM⁴ を利用する方法です。

注意



(暗号化 LVM を含む) LVM を使用する方法は、すべてのアーキテクチャで使用できるわけではありません。

LVM や暗号化 LVM を使用する場合、インストーラが作成するほとんどのパーティションを、大きなパーティションの中に作成します。この利点は、大きなパーティションの中にあるパーティションを、後から簡単に大きさを変更できることです。暗号化 LVM の場合、特殊なキーフレーズを知らずに大きなパーティションを読むことができません。そのため、あなたの (個人) データにさらなるセキュリティを提供します。

暗号化 LVM を利用する場合、インストーラは、自動的にランダムなデータを書き込んでディスクを消去します。この機能は、(ディスクの使用中の領域を分からなくし、以前インストールしていたも

⁴このインストーラでは、LVM ボリュームグループを 256 bit AES キーで暗号化し、カーネルの「dm-crypt」サポートを利用します。

のの痕跡を消去して) セキュリティを向上しますが、ディスクのサイズにより、時間がかかることがあります。

注意



LVM や暗号化 LVM を使用してガイドパーティション分割を選択した場合、パーティションテーブルへの変更は、LVM のセットアップで選択したディスクに書き込まれる必要があります。この変更によって、選択したハードディスクの現在のデータはすべて消去され、後で元に戻すことができなくなります。しかし、ディスクに書き込む前に、インストーラは変更してよいか確認してきます。

ディスク全体に対してガイドパーティション分割を選択した場合 (クラシックでも (暗号化) LVM でも)、まずはじめに、選択したディスクを使用してよいか尋ねられます。複数ディスクがある場合、すべてのディスクが一覧され、正しいものが選択されていることを確認してください。表示順は、普段使っているものと違う可能性があります。ディスクサイズを確認の手がかりにしてください。

ここでついに、ディスクのすべてのデータが失われますが、ディスクに書き込む前に変更してよいか、必ず確認してきます。パーティション分割にクラシック法を選択した場合は、終了する前に元に戻せますが、(暗号化) LVM を使用する場合は元に戻せません。

次に、以下の表から分割案を選択できます。どの案でも賛否両論あり、付録 C で議論されています。よくわからなければ、最初の項目を選択してください。ガイドパーティション分割は、最低限動作する空き領域が必要なことを、心に留めておいてください。少なくとも 1GB の空き領域 (選択した方法に依存します) がなければ、ガイドパーティション分割は失敗してしまいます。

パーティション分割方法	最低容量	作成するパーティション
All files in one partition	600MB	/, swap
Separate /home partition	500MB	/, /home, swap
Separate /home, /var and /tmp partitions	1GB	/, /home, /var, /tmp, swap

(暗号化) LVM を利用するガイドパーティション分割を行うと決めた場合、インストーラは独立した /boot パーティションも作成します。スワップパーティションを含むその他のパーティションは、LVM パーティションの内部に作成します。

分割案を選択後、新しいパーティションテーブルが次の画面に表示されます。ここでは、パーティションがどのようにフォーマットされるか、どこにマウントされるかといった情報が含まれています。パーティション一覧は以下のようになります。

```
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
```

この例では、2つのハードディスクを、いくつかのパーティションに分割しています。第1ディスクには空き領域がいくらかあります。パーティション行ごとに、パーティション番号、パーティションタイプ、サイズ、追加フラグ、ファイルシステム、マウントポイントを (あれば) 表示しています。注意: こういった詳細なセットアップはガイドパーティション分割では行えませんが、手動パーティション分割で使用できる変化を示します。

これでガイドパーティション分割を終わります。生成されたパーティションテーブルでよければ、(本節の最後にかかれてるように) 新しいパーティションテーブルを反映するよう、パーティショニングの終了とディスクへの変更の書き込みをメニューから選べます。そうでなければ、もう一度ガイドパ

パーティション分割をしたり、以下に述べる手動パーティション分割で提案する変更を修正するのにパーティションへの変更を元に戻すを選択し、ガイドパーティション分割を再実行してください。または、以下に述べる手動パーティション分割で修正してください。

6.3.4.3 手動パーティション分割

手動パーティション分割を選択すると、既存のパーティションテーブルがマウントポイントなしで表示されるのを除き、前述と同様の画面が表示されます。パーティションテーブルの手動セットアップの方法と、新しい Debian システムでのパーティションの使用方法については、本節の残りで扱います。

パーティションも空き領域もない、素のハードディスクを選択すると、新しいパーティションテーブルを作成するか確認されます(新しいパーティションを作成するのに必要)。すると選択したディスクのパーティションテーブルに、「FREE SPACE」(空き領域)という新しい行が現れます。

空き領域を選択すると、新しいパーティションを作成できるようになります。サイズやタイプ(基本か論理か)、場所(空き領域の先頭からか最後からか)といった、一連の簡単な質問に答えなければなりません。この後、新しいパーティションの詳細な概要が得られます。主な設定は、ファイルシステムがパーティションにある場合、スワップ、ソフトウェア RAID、LVM、暗号化ファイルシステムとして使うか、全く使わないかを決定する利用方法: です。その他には、マウントポイントやマウントオプション、起動フラグといったパーティションの使用法に依存した設定があります。あらかじめ選択されたデフォルト値が気に入らなければ、自由にお好みのものへと変更してください。例えば、オプション利用方法: を選択すると、スワップ、ソフトウェア RAID、LVM、またそれ以外のファイルシステムに、このパーティションを変更できます。新しいパーティションに満足したら、パーティションのセットアップを終了を選択して、partman のメイン画面に戻ってください。

パーティションに対して変更を加えたい場合は、単にそのパーティションを選択して下さい。パーティションの設定メニューに入れます。新しいパーティションを作成する際と同じ画面ですので、同様に設定を変更できます。一目わかりづらいのは、表示されているパーティションのサイズを選択して、サイズ変更ができることです。動作することがわかっているファイルシステムは、少なくとも fat16, fat32, ext2, ext3, swap です。このメニューではパーティションを削除することもできます。

少なくとも 2 つのパーティションを必ず作成してください。1 つは swap で、もう 1 つは (/ にマウントする) ルートファイルシステムです。ルートファイルシステムをマウントし忘れると、この問題を修正するまで partman は先に進みません。

partman の機能は、インストーラモジュールで拡張できますが、システムのアーキテクチャに依存します。あるはずの機能を確認できなければ、すべての必要なモジュールが読み込まれているか確認してください。(例: partman-ext3, partman-xfs, partman-lvm)

パーティション分割に満足したら、パーティション分割メニューからパーティショニングの終了とディスクへの変更の書き込みを選択してください。ディスクに行われる変更内容が表示され、その通りファイルシステムを作成するかどうか確認することになります。

6.3.4.4 マルチディスクデバイス(ソフトウェア RAID)の設定

コンピュータに複数ハードディスクドライブがある⁵なら、ドライブのパフォーマンスの向上やデータの信頼性向上のために partman-md を使用できます。この結果を マルチディスクデバイス(ソフトウェア RAID の方が有名)と呼びます。

MD は基本的に別のディスクにあるパーティションを束ねて、論理デバイスの形に結合したものです。このデバイスは通常のパーティション(例: partman でフォーマットでき、マウントポイントに割り当てられる等)と同様に使用できます。

どんな恩恵を受けるかは、作成する MD デバイスの種類に依存します。現在、以下をサポートしています。

RAID0 RAID0 はパフォーマンスに主眼をおいています。RAID0 は全入力データを stripes へ分割し、均等にディスクアレイの各ディスクに分配します。これにより、読み取り・書き込みの処理速度を向上できますが、ディスクのうちの 1 つが破損したら、すべてを失ってしまいます。(情報の一部は正常なディスク上にありますが、他の部分は破損したディスク上にあるからです)

RAID0 の典型的な使用法は映像編集用のパーティションです。

RAID1 信頼性第一である場合、RAID1 を構成するとよいでしょう。全パーティションが正確に同じデータを含むような、いくつかの(たいてい 2 つ)等しいサイズのパーティションから成ります。

⁵ 本当のことをいえば、同一の物理ドライブを分割して MD デバイスを構築できますが、何も利点がありません。

これは本質的に3つのことを意味します。まずディスクの1つが破損した場合、残ったディスクにデータミラーが残ります。次に利用可能領域の断片だけの使用もできます。(もっと正確には、RAIDで構成する最小のパーティションサイズとなります)第3に、ディスクからのファイルの読み込みをロードバランスする事ができます。これにより、ファイルサーバのような、書き込みより読み込みの方が負荷が高くなる傾向のあるサーバのパフォーマンスを改善できます。

破損した場合に、任意に予備ディスクを破損したディスクの代わりに、ディスクアレイにつけることができます。

RAID5 RAID5は速度と信頼性、データの冗長性をうまく折衷しています。RAID5はストライプへ入力するデータをすべて分割し、1つ以外の全ディスクに(RAID0のように)等しく分配します。RAID0と違い、RAID5は(残りのディスクに書かれている)パリティ情報も計算します。パリティディスクは静的(これをRAID4と呼ぶ)ではありません。(定期的に変更され)パリティ情報を全ディスクに等しく分配します。あるディスクが故障した場合、情報の失った部分は残ったディスクとそのパリティから計算されます。RAID5は少なくとも3つのアクティブなパーティションから成ります。故障した場合に、任意でディスクアレイ中の故障したディスクの箇所に予備のディスクをセットできます。

おわかりのように、RAID5はRAID1より冗長性が少なく、同程度の信頼性を持ちます。一方、パリティ情報を計算するため、RAID0より書き込み操作が少し遅いかもかもしれません。

RAID6 RAID6はパリティデバイスが1つではなく2つであるという点を除き、RAID5と似ています。RAID6アレイは、2つのディスクが故障するまで存続できます。

RAID10 RAID10はストライピング(RAID0)とミラーリング(RAID1)を組み合わせたものです。格納データのn個のコピーを作成し、パーティションをまたがって分配するため、同じデバイスに同じデータを格納することはありません。nのデフォルト値は2ですが、エキスパートモードで別の値を設定できます。使用するパーティションの数は、少なくともn個必要です。RAID10はコピーを分配するために、様々なレイアウトを用意しています。デフォルトはnearコピーです。nearコピーは、全ディスクにおいて同一のオフセットで全てのコピーを分配します。farコピーは、全ディスクにおいて異なるオフセットで全てのコピーを分配します。offsetコピーは、個々のコピーではなく、ストライピングしたものをコピーします。

RAID10はパリティを計算するという難点を回避しつつ、信頼性と冗長性を確保するのに使用できます。

まとめると以下ようになります。

タイプ	デバイス最小構成数	予備デバイス	ディスク破損に耐えるか?	利用可能領域
RAID0	2	no	no	RAIDにある最小パーティションのサイズ×デバイス数
RAID1	2	任意	yes	RAIDにある最小パーティションのサイズ
RAID5	3	任意	yes	RAIDにある最小パーティションのサイズ×(デバイス数-1)
RAID6	4	任意	yes	最小パーティションのサイズ×(RAIDにあるデバイス数-2)
RAID10	2	任意	yes	全パーティションサイズ÷チャンクのコピー数(デフォルトは2)

ソフトウェア RAID に関して、もっと知りたい場合は [Software RAID HOWTO](#) をご覧ください。

MD デバイスを作成するには、RAID で使うための (これは利用方法: → RAID の物理ボリュームを選択して出てくる、パーティション設定メニューの partman で行えます)

注意



計画しているパーティション分割方式で、システムがブートできることを確認してください。通常、ルート (/) ファイルシステムに RAID を使用する際には、/boot を独立したファイルシステムにする必要があります。ほとんどのブートローダは、ミラーリングした (ストライピングではなく!) RAID1 をサポートしています。そのため、/ に RAID5 を用い、/boot に RAID1 を用いる例が選択したり得ます。

次にメインの partman メニューからソフトウェア RAID の設定を選んでください。(このメニューは、少なくともパーティションをひとつ RAID の物理ボリュームとしてマークしないと表示されません) partman-md の最初の画面では、単に MD デバイスの作成を選択してください。サポートされる MD デバイスのリストも提供されます。この中から 1 つ (例: RAID1) を選択してください。その後は選択した MD デバイスに依存します。

- RAID0 は単純です。利用可能な RAID パーティションの一覧が提供されますので、単に MD にするパーティションを選択してください。
- RAID1 は少しトリッキーです。まず MD にするアクティブなデバイスの数、スペアデバイスの数を入力します。次に利用可能な RAID パーティションの一覧からアクティブのもの、次にスペアのものを選ぶ必要があります。選択したパーティションの数と先ほど入力した数は一致しなければなりません。心配しないでください。間違っていて違う数のパーティションを選択した場合、debian-installer は問題を修正するまで、先に進ませません。
- RAID5 では、少なくとも 3 つのアクティブパーティションを使用する必要があるという例外を除き、RAID1 と同様のセットアップ手順を行います。
- RAID6 では、少なくとも 4 つのアクティブパーティションを使用する必要があるという例外を除き、RAID1 と同様のセットアップ手順を行います。
- RAID10 もまた、エキスパートモードであることを除き、RAID1 と同様のセットアップ手順を行います。エキスパートモードでは、debian-installer はレイアウトについて確認します。レイアウトは 2 段階に分かれています。第 1 段階はレイアウトタイプです。n (near コピー)、f (far コピー)、o (offset コピー) のどれかになります。第 2 段階はデータから作るコピーの数です。少なくとも、コピーをすべて異なるディスクに分配するには、たくさんのアクティブデバイスが必要になります。

同時に数種の MD を持つことは完全に可能です。例えば、3 つの 200GB の MD 専用ドライブがあって、どれも 2 つの 100GB のパーティションに分かれている場合、3 つのドライブすべての最初のパーティションを RAID0 (高速な 300GB のビデオ編集パーティション) で結合でき、その他の 3 つのパーティション (アクティブ 2 基、スペア 1 基) を RAID1 (/home 用に信頼できる 100GB のパーティション) で結合できます。

お好みの通りに MD デバイスの設定をした後で、完了 partman-md として partman に戻れます。新しい MD デバイスにファイルシステムを作成し、マウントポイントなどの通常の属性を設定してください。

6.3.4.5 論理ボリュームマネージャ (LVM) の設定

システム管理者や「上級」ユーザとしてコンピュータを動かしていると、ディスク内のあるパーティション (たいてい最も重要なもの) が足らなくなり、他のパーティションは全体的にあまり使用されていないという状況が確実にあります。このような場合は、内容を移動したりシンボリックリンクを張るといった管理を行うこととなります。

上記のような状況を避けるために、論理ボリュームマネージャ (LVM) を利用できます。簡単に言うと、LVM では複数のパーティション (LVM 用語で物理ボリューム (physical volumes)) を仮想ディスク (いわゆるボリュームグループ (volume group)) の形に結合でき、このディスクを仮想パーティション (論理ボリューム (logical volumes)) に分割できます。ポイントは、論理ボリュームは (もちろんその下のボリュームグループも)、複数の物理ディスクをまたがって定義できると言うことです。

例えば、古い 160GB の /home パーティションに、もっと容量を追加することを考えます。単にあなたは新しい 300GB のディスクをコンピュータに追加し、既存のボリュームグループに入れます。その後 /home ファイルシステムを保持したまま論理ボリュームをリサイズします。するとほら、パーティションが 460GB へと新品交換されたので、ユーザの空き容量がすこしばかり増えたこととなります。もちろんこの例は少し単純にしすぎです。まだ読んでいないようなら、[LVM HOWTO](#) を調べるべきです。

debian-installer での LVM のセットアップはかなりシンプルで、partman 内部で完全にサポートしています。始めに、パーティションを LVM の物理ボリュームとして使用するよう、マークをつけねばなりません。これは、パーティション設定メニューの partman 内で利用方法: → LVM の物理ボリュームを選ぶことで行います。

警告



注意: 新規の LVM 設定は LVM タイプコードがマークされた全てのパーティション上の全データを破壊します。そのため、既にディスク上に LVM があってそのマシンへ Debian を追加インストールしたい場合、古い (既存の) LVM は消去されます! 同じことが (何らかの理由で) 誤って LVM タイプコードがマークされているものの、別のものが含まれている場合 (例えば暗号化ボリューム) のパーティションにも言えます。新たに LVM 設定を実施する前にその様なディスクをシステムから取り除きましょう!

メインの partman 画面に戻ると、論理ボリュームマネージャの設定が新しく選択できるようになっています。これを選択すると、まず決定していないパーティションテーブルへの変更 (があれば) 確認を行い、その後 LVM 設定メニューを表示します。メニューの上部には LVM 設定の概要を表示します。メニュー自体はそのときに実行できる操作のみ表示します。行える操作は以下の通りです。

- 設定の詳細表示: LVM デバイスの構造、論理ボリュームの名称やサイズなどを表示します
- ボリュームグループの作成
- 論理ボリュームの作成
- ボリュームグループの削除
- 論理ボリュームの削除
- ボリュームグループの拡張
- ボリュームグループの縮小
- 完了: メインの partman 画面に戻ります

はじめにボリュームグループを作成し、その中に論理ボリュームを作成するのに、このメニューのオプションを使用してください。

メインの partman 画面に戻ると、作成した論理ボリュームが通常のボリュームと同じように表示されています (そして同じように扱えます)。

6.3.4.6 暗号化ボリュームの設定

debian-installer では暗号化パーティションを設定できます。暗号化パーティションに保存したファイルはすべて、暗号化した形で即座にデバイスに書き込まれます。暗号化したデータへのアクセスは、暗号化パーティションを作成した際に設定した パスフレーズを入力した後で有効になります。この機能は、ノート PC やハードディスクが盗難に遭った際に、機密データを保護するのに便利です。盗人がハードディスクの物理データにアクセスしようとする際、正しいパスフレーズを知らないと、ハードディスクのデータはランダムな文字列にしか見えません。

暗号化するのに最も重要なパーティションが 2 つあります。個人的なデータを格納する home パーティションと、操作中に機密データを一時的に格納する swap パーティションです。もちろん、その他のパーティションの暗号化を妨げるものはなにもありません。たとえば、データベースサーバ、メールサーバ、プリンタサーバがそれぞれファイルを格納する /var や、様々なプログラムが、潜在的に興味深い一時ファイルを作成する /tmp です。システム全体を暗号化したいと考える方もいます。一般にここで暗号化をしない方がよい唯一の例外パーティションは、/boot パーティションです。歴史的に、

暗号化されたパーティションからカーネルを起動する方法がなかったためです。(GRUB は暗号化されたパーティションから起動できるようになりましたが、debian-installer は現在暗号化された /boot からの起動をネイティブにサポートしていません。そのため設定は別の文書で取り扱っています。)

注意



データの読み書き時に常に暗号化・復号を行うため、暗号化パーティションのパフォーマンスは、暗号化していないものよりも低下する事に注意してください。パフォーマンスは、CPU のスピード、選択した暗号方式、暗号化キーの長さに影響を受けます。

暗号化を用いるには、メインパーティションメニューで空き領域を選択して、新しいパーティションを作成する必要があります。他には既存のパーティション (例、通常のパーティション、LVM 論理ボリューム、RAID ボリューム) を選択するという手もあります。パーティション設定メニューの、利用方法: で暗号化の物理ボリュームを選択する必要があります。そうすると、メニューにパーティションを暗号化するオプションが追加されます。

debian-installer でサポートしている暗号化方法は dm-crypt (新しめの Linux カーネルに収録されていて LVM 物理ボリュームをホストできる) です。

暗号化するにあたり Device-mapper (dm-crypt) を選択した場合に利用できるオプションを見てみましょう。いつものように、よく分からなければデフォルト値を指定してください。セキュリティを念頭に置いて選択されています。

Encryption: aes このオプションで、パーティションのデータを暗号化するのに使用する、暗号化アルゴリズム (暗号方式) を選択します。現在、debian-installer は以下の暗号方式をサポートしています。aes, blowfish, serpent, twofish です。それぞれのアルゴリズムの品質についての議論は、この文書の範疇を越えてしまいますが、以下はあなたの決断の助けになるかもしれません。AES は、2000 年に米国商務省標準技術局により、21 世紀の機密情報を保護する標準暗号化アルゴリズムとして採用されました。

Key size: 256 ここでは暗号化キーの長さを指定できます。一般的に暗号化キーが長くなると暗号強度が向上します。一方、暗号化キーが長くなると、大抵パフォーマンスにマイナスの影響を与えます。利用できる暗号化キーのサイズは暗号方式に依存します。

IV algorithm: xts-plain64 初期化ベクトルや IV アルゴリズムは、同じ 平文データと同一の暗号化キーで、常に異なる 暗号文の出力を保証し、安全に暗号を解読するのに利用されます。これにより、暗号化データ中に繰り返されるパターンから、攻撃者が情報を推測できないようにします。

デフォルトの xts-plain64 は現在のところ、攻撃される恐れがもっとも少ないです。その他の選択肢は、新しいアルゴリズムに対応していない、以前インストールしたシステムと互換をとる場合のみ使用してください。

Encryption key: Passphrase ここでは、このパーティションの暗号化キーのタイプを選択できます。

Passphrase 暗号化キーを、プロセスの後で入力するパスフレーズに基づいて計算⁶します。

Random key 暗号化パーティションを作成するたびに、新しい暗号化キーをランダムに生成します。言い換えると、シャットダウンごとに暗号化キーがメモリから削除され、パーティションの内容を失うということです。(もちろん総当たりで暗号化キーを推測することはできませんが、暗号アルゴリズムに未知の弱点がない限り、生きているうちには解読されません)

Random key は swap パーティションで使うと便利です。というのも、パスフレーズを覚えておく必要もなく、コンピュータをシャットダウンする前に、機密情報を swap パーティションから掃除するからです。しかし、最近の Linux カーネルで利用できる「suspend-to-disk」機能では使用できないということでもあります。(次の起動中に) swap パーティションからサスペンドデータを、復元できなくなってしまうのです。

Erase data: yes 暗号化の前に、このパーティションの内容をランダムなデータで上書きするかどうかを決めます。そうしないと攻撃者が、パーティションのどの部分を使用中で、どの部分が使用し

⁶暗号化キーにパスフレーズを使用するのは、LUKS を使用して設定するという意味です。

ていないかを見分けられますので、上書きすることをお勧めします。その上、以前インストールして残ってしまったデータを、復元しにくくします⁷。

暗号化パーティション用に必要なパラメータを選択すると、メインパーティション分割メニューに戻ります。そこに今度は暗号化されたボリュームの設定という項目があるはずですが、これを選択すると、削除するようにマークしたパーティションを本当に削除してよいか確認し、新しいパーティションテーブルを書き込むといったアクションを起こします。大きなパーティションではしばらく時間がかかるでしょう。

次に、パスフレーズを使用するよう設定していれば、パスフレーズを訊かれます。よいパスフレーズは、8文字以上で、文字・数字・その他の記号が混ざり、辞書に載っていないか、容易に連想される情報(誕生日、趣味、ペットの名前、家族や親戚の名前など)でないものです。

警告



パスフレーズを入力する前に、キーボードが正しく設定され、期待した文字が入力できるようになっていなければなりません。よくわからなければ、別の仮想端末に切り替えて、プロンプトに入力してください。これにより、例えば、インストール中に azerty 配列を使用しているのに、qwerty 配列でパスフレーズを入力するといったことで、あなたが後で驚くようなことにはならないでしょう。この状況はいくつかの原因が考えられます。インストール中に別のキーボード配列に切り替えたとか、ルートファイルシステムのパスフレーズを入力する時に、まだ選択したキーボードレイアウトが有効でなかったのかもしれない。

暗号化キーの作成に、パスフレーズ以外の方法を選択した場合、すぐに暗号化キーを生成します。インストールの初期では、カーネルが十分なエントロピーを集めていないので、このプロセスに長時間かかるかもしれません。エントロピーを集めてこのプロセスのスピードを上げるには、ランダムにキーを押す、別の仮想コンソールに切り替えて(ファイルのダウンロードや、大きなファイルを /dev/null に流すなど)ネットワークやディスクのトラフィックを起こすなどがあります。暗号化するパーティションの数だけ繰り返します。

メインパーティション分割メニューに戻ると、暗号化ボリュームが、通常のパーティションと同様に追加パーティションとして見えています。以下の例では dm-crypt で暗号化したボリュームを示します。

```
Encrypted volume (sda2_crypt) - 115.1 GB Linux device-mapper
#1 115.1 GB F ext3
```

今度は、ボリュームをマウントポイントに割り当てます。また、デフォルトのファイルシステムタイプが合っていないければ変更も行います。

括弧内の識別子(ここでは sda2_crypt)と、暗号化ボリュームを割り当てるマウントポイントに注意を払ってください。後で新しいシステムを起動するときに、この情報が必要になります。通常の起動プロセスと、暗号を伴う起動プロセスの相違点は、[項7.2](#)で扱います。

パーティション分割の内容に納得いったら、インストールに進んでください。

6.3.5 基本システムのインストール

この段階が最重要でないとはいえ、全体の基本システムをダウンロード、確認、展開にインストールのかなりの部分を費やします。遅いコンピュータや遅いネットワーク接続しかなければ、ある程度時間がかかるかもしれません。

基本システムのインストール中、パッケージの展開・セットアップメッセージは、tty4 にリダイレクトされます。左 Alt-F4 を押すと、この端末 (terminal) にアクセスできます。元のインストーラの画面に戻るには、左 Alt-F1 を押してください。

このフェイズでの展開・設定メッセージは、/var/log/syslog に保存されます。シリアルコンソールでインストールする場合、これをチェックできます。

インストールの途中で、Linux カーネルをインストールします。デフォルトの優先度では、インストーラはハードウェアと最も適合するカーネルを選びます。より低い優先度モードでは、利用可能なカーネルのリストから選ぶことができます。

⁷3文字の機関では、磁気光学メディアを何度か書き換えた後でも、データを復元できると信じられています。

パッケージ管理システムを使用してパッケージをインストールした際、デフォルトでは、そのパッケージが推奨しているパッケージもインストールされます。推奨パッケージは、選択したソフトウェアの核となる機能にとって厳密には必要ではありませんが、ソフトウェアを拡張し、パッケージメンテナ視点で、通常そのソフトウェアと同時に入っているべきパッケージです。

注意



技術的な理由で、ベースシステムのインストール中にインストールされるパッケージは、「推奨」が含まれていません。前述の規則は、インストールプロセス中のこの時点以降でのみ効力を発揮します。

6.3.6 追加ソフトウェアのインストール

この時点では、制限されたシステムが利用できるようになります。ほとんどのユーザは、お好みに調整するのに、追加ソフトウェアをシステムにインストールするでしょうが、これはインストーラから行えます。遅いコンピュータやネットワーク接続を使用していると、このステップは基本システムのインストールよりも時間がかかります。

6.3.6.1 apt の設定

Debian GNU/Linux システムにパッケージをインストールするツールの 1 つに apt パッケージの apt プログラムがあります⁸。パッケージ管理のその他のフロントエンドには、aptitude や synaptic も使われます。これらのフロントエンドは追加機能 (パッケージの検索や状態チェック) を、すばらしいユーザインターフェースと統合しているので、新しいユーザにお勧めします。

パッケージをどこから取得するか、apt を設定しておかなくてはなりません。この設定の結果は、`/etc/apt/sources.list` ファイルに書き込まれます。インストール完了後に、お好みに合わせて検査・編集できます。

デフォルトの優先度でインストールしている場合、インストール方法と、可能であればインストールの初期に選択した内容から、大部分の設定を自動で面倒みてくれます。ほとんどの場合、インストーラは自動でセキュリティミラーを追加します。また、安定版をインストールしている場合、「stable-updates」更新サービスのミラーを追加します。

低い優先度でインストールしている場合 (例: エキスパートモード)、もっと多くのことを自分で決定できます。セキュリティや stable-updates 更新サービスの有無や、アーカイブの「contrib」や「non-free」からのパッケージ追加の有無を選べます。

6.3.6.1.1 2 枚以上の CD/DVD イメージでのインストール

複数枚からなる CD や DVD イメージでインストールする場合、さらにインストールメディアをスキャンするか、インストーラが尋ねてきます。追加するメディアがある場合、そこからパッケージをインストールするため、スキャンしたくなると思います。

追加するメディアがない場合、これは必須ではないので、問題ありません。ネットワークミラーも使用しない場合 (次節で説明します)、次のステップで選択する、タスクに属するすべてのパッケージをインストールできるわけではないことを意味します。

⁸パッケージを実際にインストールするプログラムは、dpkg であることに注意してください。ですが、このプログラムは、どちらかという下位のツールです。apt はもっと上位のツールで、適切に dpkg を起動します。また、インストールメディアやネットワーク、その他から、パッケージをどのように取得するかも知っています。さらに、インストール作業が正しく行えるように、パッケージが必要とする他のパッケージも自動的にインストールできます。

注意

CD および DVD イメージにあるパッケージは、人気のある順に納められています。これにより、ほとんどの人はセットの 1 枚目のイメージのみを使い、非常に少数の人だけが、最後のイメージに入っているパッケージを使用することになります。

これはフル CD セットのうち、まったく使わないものを買ったり、ダウンロードして焼いたりといったことは、お金の無駄になってしまうということでもあります。ほとんどの場合、3~8 枚の CD を用意し、さらにパッケージを追加する必要がある場合には、ミラーサイトを利用しインターネットから取得する方が楽です。DVD セットの場合も同じことが言えます。1 枚目の DVD か、もしかすると 2 枚目の DVD で必要なものをカバーできるでしょう。

複数のインストールメディアをスキャンする場合、現在ドライブに入っているものとは別のインストールメディアにあるパッケージが必要になると、インストーラは交換するよう促します。注意: ディスクは、同じセットに属するもののみをスキャンするべきです。スキャンする順番はあまり重要ではありませんが、昇順にスキャンすると、失敗する可能性が低くなります。

6.3.6.1.2 ネットワークミラーの利用

インストールに関するよくある質問に、パッケージの取得元にネットワークミラーを使用するかどうかがあります。ほとんどの場合、デフォルトの回答でうまくいきますが、中には例外もあります。

完全な CD/DVD イメージからインストールしない場合、非常に最小限のシステムのみで完了するのなら、ネットワークミラーを使用するべきです。しかし、インターネット接続に制限がある場合、インストールの次のステップで、desktop タスクを選択しないのが最善でしょう。

1 枚の完全な CD イメージからインストールしている場合は、ネットワークミラーを使用する必要はありませんが、1 枚の CD イメージには非常に限られた数のパッケージしか含まれていないため、ネットワークミラーを使用するのを強くお勧めします。インターネット接続に制限がある場合は、まだここでネットワークミラーを設定しない方がよいでしょう。CD イメージでできる限りのインストールを行い、追加パッケージのインストールは、(新しいシステムで起動した後など) 後で行うのがよいでしょう。

DVD でインストールしている場合、インストールに必要なパッケージは、1 枚目の DVD イメージで提供されているはずで、ネットワークミラーの使用はオプションとなります。

ネットワークミラーを追加する利点は、CD/DVD イメージが作成された後にポイントリリースに含まれた更新が、インストールできるということです。つまり、インストールしたシステムのセキュリティや安定性を傷つけることなく、CD/DVD の寿命を延ばすことができます。

まとめると、ネットワークミラーを選択するのは、質の良いインターネット接続がない場合を除き、一般的によい考えです。パッケージの最新版がインストールメディアで利用できる場合には、インストーラは常にそちらを使用します。従って、ミラーを選択した場合のダウンロードするデータ量は、以下に依存します。

1. インストールの次のステップで選択するタスク。
2. どのパッケージがそのタスクに必要なか。
3. そのパッケージがスキャンしたインストールメディアに収録されているかどうか。
4. インストールメディアに収録したパッケージの更新版が、ミラーサイト (通常のパッケージのミラーサイトだけでなく、セキュリティのミラーサイトや stable-updates のミラーサイト) に用意されているかどうか。

最後の点については、ネットワークミラーを使用しないように選択したとしても、セキュリティや stable-updates に更新があり、そのサービスを使用するように設定している場合は、パッケージをダウンロードする可能性が残っている、ということに注意してください。

6.3.6.1.3 ネットワークミラーの選択

ネットワークミラーを使わない選択をした場合を除き、インストールプロセスの初期で行った国の選択を元にしたネットワークミラーのリストが与えられています。提示されたデフォルト値を選択すると、大抵うまく行きます。

提供されるデフォルトは `deb.debian.org` です。これ自体はミラーではなく、最新かつ高速なミラーへリダイレクトされます。これらのミラーは TLS (https プロトコル) と IPv6 をサポートします。このサービスは Debian システム管理 (DSA) チームによって維持されています。

「情報を手動で入力」を選択することで、ミラーを手で指定することもできます。そうするとミラーのホスト名とオプションでポート番号を指定できます。これは実際の URL ベースで、つまり IPv6 アドレスを指定する場合には [] で囲まなければなりません。例えば「[2001:db8::1]」。

コンピュータが、IPv6 のみのネットワークにつながっている (おそらくユーザの大多数に一致しない) 場合、あなたの国のデフォルトミラーはうまく動作しないかもしれません。リスト内のすべてのミラーは、IPv4 だけでなく IPv6 でも通信できます。個々のミラーの接続は、時間とともに変わることがあり、その情報をインストーラに持たせられません。あなたの国向けのデフォルトミラーに IPv6 接続がない場合、提示された他のミラーを試すか、「情報を手動で入力」オプションを選択し、ミラー名に「`ftp.ipv6.debian.org`」を指定できます。これは、IPv6 が有効なミラーのエイリアスで、おそらく可能な限り速い、とはいえないと思います。

6.3.6.2 ソフトウェアの選択・インストール

インストール処理中に、追加ソフトウェアをインストールする機会があります。85593 個もの利用可能パッケージから、個々のパッケージを取り上げるよりも、インストール処理のこの段階では、いち早く様々なコンピュータのタスクをセットアップするよう、定義済みのソフトウェア集合を選択・インストールするのに集中します。

タスクは、様々なジョブやあなたがコンピュータにやらせたいことを、いくつか大まかに表しています。「デスクトップ環境」「Web サーバ」「Print サーバ」といった具合です⁹。項 D.2 に、利用可能タスクの必要容量一覧があります。

いくつかのタスクは、インストールするコンピュータの特性により、あらかじめ選択されている可能性があります。選択されているものが合わない場合は、そのタスクの選択をはずせます。全くタスクを選ばないようにもできます。

ティップ



インストーラの標準ユーザインターフェースでは、タスクの選択をスペースバーでトグルできます。

注意

「デスクトップ環境」タスクは、グラフィカルデスクトップ環境をインストールします。



デフォルトで `debian-installer` はデスクトップ環境をインストールします。インストール中に異なるデスクトップ環境を、インタラクティブに選択することが可能です。デスクトップ環境を複数インストールすることもできますが、組み合わせによっては互いに排他的でインストールできない可能性もあります。

希望のデスクトップ環境に必要なパッケージが実際に利用できる場合にのみ動作することに注意してください。フル CD イメージ 1 枚でインストールしている場合、容量が限られているその CD イメージに入っておらず、ミラーサイトからダウンロードする必要があるかもしれません。DVD イメージやその他のインストール方法では、利用可能なデスクトップ環境のインストールがうまくいくでしょう。

各サーバタスクでは、おおまかに以下のソフトウェアをインストールします。Web サーバ: `apache2`; 印刷サーバ: `cups`; SSH サーバ: `openssh`。

⁹表示されるリストは、インストーラが単に `tasksel` プログラムを起動しているだけ、ということを知っておいてください。インストールの後で、他のパッケージをインストール (または削除) するのにいつでも実行できます。また `aptitude` のような、よりきめ細かいツールも利用できます。インストール完了後、特定の 1 パッケージを探すのなら、単に `aptitude install` パッケージ名を実行してください。パッケージ名は、探したいパッケージ名です。

「標準システム」タスクは、優先度が「標準」のパッケージをインストールします。ここでは、通常どんな Linux や Unix のシステムでも有効な、たくさんの共通ユーティリティを含んでいます。何をしようとしているのか解っていて、本当に最小限のシステムが必要なのであれば、このタスクを選択したままにしてください。

言語選択で、デフォルトロケールに「C」ロケール以外を選択した場合、tasksel は、そのロケールで定義されている地域化タスクがあるかチェックし、関連する地域化パッケージを自動的にインストールしようとします。これには例えば単語集や、あなたの言語の特殊なフォントが含まれます。デスクトップ環境を選択している場合、適切な地域化パッケージも (有効なら) インストールします。

タスクを選択したら、Continue を選択してください。ここで apt が選択したタスクの一部をインストールし始めます。個々のプログラムで、ユーザからのもっと詳細な情報が必要な場合、このプロセス中に問い合わせが発生します。

デスクトップタスクは非常に大きいことを意識してください。特に、通常の CD-ROM と、ミラーサイトにある CD-ROM 外のパッケージを組み合わせる場合、インストーラが、ネットワークから大量のパッケージを取得しようとするかもしれません。インターネット接続が低速な場合、長い時間かかるでしょう。一度、パッケージのインストールを始めたら、キャンセルするオプションはありません。

パッケージが CD-ROM に含まれている場合でも、CD-ROM にあるパッケージよりもミラーサイトにあるパッケージの方が新しければ、インストーラはミラーサイトから取得しようとします。安定版をインストールしている場合はポイントリリース (オリジナルの安定版リリースの更新) 後に、テスト版をインストールしている場合は古いイメージを使用していると、こういったことが起こり得ます。

6.3.7 システムを起動可能に

ディスクなしワークステーションにインストールするなら、ローカルディスクから起動するなんて、明らかに意味がありませんから、このステップをスキップしてください。

6.3.7.1 他 OS の検出

ブートローダがインストールされる前に、インストーラは既にインストールされている他の OS の検出を試みます。サポートする OS を見つけると、ブートローダインストールステップの間にそれを通知します。また、Debian に加えて他の OS をブートできるように、このコンピュータを設定します。

複数の OS を同一の機械で起動するのは、いまだに魔術的だということに注意してください。他の OS を検出し起動するようにブートローダをセットアップする自動サポートは、アーキテクチャごとに (サブアーキテクチャそれぞれでさえ) 異なります。作動しない場合は、詳細についてブートマネージャの文書を調べるべきです。

6.3.7.2 flash-kernel でシステムをブートできるようにする

全 ARM 基盤に共通となるファームウェアインターフェイスはないため、ARM デバイスでシステムをブートできるようにするのに必要な手順はデバイスに大きく依存します。Debian では flash-kernel というツールを使ってこの処理を行います。flash-kernel は様々なデバイスでシステムをブートできるようにするのに必要な特定の操作を記述したデータベースを取録しています。デバイスが現在サポートされているものかどうかを検出し、サポートされている場合は必要な操作を実行します。

内部に NOR 型または NAND 型のフラッシュメモリを使っているデバイスからのブートでは、flash-kernel はこの内部メモリにカーネルと初期 RAM ディスクを書き込みます。この方法は古い armel 機器で特に一般的です。こういったデバイスはほとんどのもので、内部フラッシュメモリにカーネルや RAM ディスクを複数置くことは許容していない、つまり flash-kernel をこういったデバイスで実行すると、以前にそのフラッシュメモリに存在した内容は通常上書きされることに注意してください!

システムファームウェアとして u-boot を利用し、外部ストレージメディア (MMC/SD カードや USB 大容量ストレージデバイス、IDE/SATA ハードディスク) からカーネルと初期 RAM ディスクをブートする ARM システムでは、flash-kernel は適切なブートスクリプトを生成し、ユーザが介入せずに自動起動できるようにします。

6.3.7.3 ブートローダなしで継続

このオプションは、アーキテクチャ/サブアーキテクチャにブートローダがない、あるいはインストールする気がない (例えば、既存のブートローダを使用するつもりであるとか) 時に、ブートローダをインストールしていなくても、インストールを完了するのに利用できます。

手動でブートローダを設定する場合、`/target/boot` にインストールしたカーネルの名前をチェックしてください。またそのディレクトリに `initrd` が存在するかチェックしてください。存在するなら、ブートローダにそれを使うよう指定しなければなりません。他に必要な情報は、`/` ファイルシステムとするディスクないしパーティション、(`/boot` を個別のパーティションとする場合) `/boot` ファイルシステムとするディスクないしパーティションが必要です。

6.3.8 インストールの完了

これからインストーラが行ういくつかのタスクが、Debian のインストール過程での最終段階です。ほとんどが `debian-installer` の後片付けです。

6.3.8.1 システム時計の設定

インストーラは、コンピュータの時計を UTC にするかどうかを、尋ねることがあります。通常この質問は可能な限り避け、他のオペレーションシステムがインストールされているかどうか、といったことから UTC を基準にするかどうかを判断します。

エキスパートモードでは、常に時計を UTC にあわせるかどうかを選択することになります。

ここで、`debian-installer` は、システムのハードウェア時計に現在の時間を保存しようと試みます。先ほどの選択により、UTC か現地時間のどちらかで保存します。

6.3.8.2 システムの再起動

インストーラの起動に使用したブートメディア (CD、USB メモリなど) を、取り出すよう促されます。システムはこの後、新しい Debian システムで再起動します。

6.3.9 トラブルシューティング

本節に挙げるコンポーネントは、通常インストールプロセスに関係しませんが、何かうまく行かない時に、ユーザの助けになるようバックグラウンドで待っています。

6.3.9.1 インストールログの保存

インストールが成功したら、インストールプロセス中のログファイルが、新しい Debian システムの `/var/log/installer/` に自動的に作成されています。

メインメニューからデバッグログを保存を選択すると、ログファイルを USB メモリやネットワーク、ハードディスク、その他メディアに保存できます。これは、インストール中に致命的な問題に遭遇してしまい、別システムでそのログを調査したいときや、インストールレポート向けにログを添付したいときに便利です。

6.3.9.2 シェルの使用とログの参照

インストール中にシェルを起動する方法はいくつかあります。ほとんどのシステムでは、さらにシリアルコンソールでインストールしていない場合、左 `Alt-F2`¹⁰ を押して (Mac のキーボードでは、`Option-F2`)、第 2 仮想コンソールに切り替えるのが簡単です。左 `Alt-F1` でインストーラ自体に戻ってください。

コンソールに切り替えられない場合、メインメニューにあるシェルの実行でもシェルを起動できます。ほとんどのダイアログから、`Go Back` ボタンを何度か押して、メインメニューに戻れます。`exit` と入力すると、シェルを終了してインストーラに戻ります。

この段階では RAM ディスクから起動しています。また、使用には制限がありますが Unix ユーティリティが利用可能です。どのプログラムが利用できるかはコマンド `ls /bin /sbin /usr/bin /usr/sbin` や `help` とタイプするとわかります。シェルは `ash` という Bourne shell のクローンで、自動補完や履歴のような、気の利いた機能を備えています。

ファイルの編集や表示をするには、`nano` というテキストエディタを使用してください。インストールシステムのログファイルは、`/var/log` ディレクトリにあります。

¹⁰スペースバーの左側にある `Alt` キーと、`F2` ファンクションキーを同時に押してください。

注意

シェルの中では、有効なコマンドを許可されている限り、基本的になんでもできますが、何か問題が発生したときのデバッグ用に、シェルを使用するオプションはここにしかありません。

シェルから手動で何か行くと、インストールプロセスや結果にエラーが発生したり、インストールが完了しなかったりといった恐れがあります。特に、インストーラで `swap` を有効にするようにし、シェルから手動で行わないようにしましょう。

6.3.10 ネットワーク越しのインストール

`network-console` はとても興味深いコンポーネントで、インストールの大部分を、SSH を用いたネットワーク越しで行えるようにします。ネットワークを使用するということは、少なくともネットワークをセットアップするまで、コンソールでインストールを行わなければならないということも含んでいます。(でもこの部分は項4.4 で自動化できます)

このコンポーネントは、デフォルトではメインインストールメニューには現れません。そのため、自分で明示しなければなりません。光学メディアからインストールする場合、優先度を中にするかインストールメニューを呼び出し、インストールメディアからインストーラコンポーネントをロードを選んでください。また、追加コンポーネントの一覧から `network-console: SSH` を使ってリモートでインストールを続けるを選んでください。読み込みに成功すると、SSH を使ってリモートでインストールを続けるから呼ばれる新しいメニュー項目が表示されます。

この新しいエントリを選択したら、インストールするシステムに接続するための新しいパスワード(とその確認)を入力してください。これで以上です。今、リモートでログインするよう促す画面が出ているはず。ユーザ名は `installer`、パスワードは先ほど入力した物を使用してください。この画面にある重要な細かい点として、このシステムの指紋 (fingerprint) があります。この指紋を、リモートでインストールを続ける人に、安全に転送する必要があります。

ローカルでインストールすると決めた場合は、Enter を押してください。メインメニューに戻ります。そこで別のコンポーネントを選択してください。

それでは回線の向こう側へ行きましょう。前提として、あなたの端末がインストールシステムで使用する UTF-8 エンコードを使用できるように設定されている必要があります。そうでなければ、リモートインストールは可能ですが、ダイアログの枠線が化けたり ASCII 以外の文字が読めないといった妙な表示になってしまいます。インストールシステムへの接続を確立するには、単に以下のように入力してください。

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

`install_host` には、インストールするコンピュータの名前か IP アドレスのどちらかをセットします。実際のログインの前に、リモートシステムの指紋を表示するのでそれが正しいかどうか確認してください。

注意

インストーラの ssh サーバは、keep-alive パケットを送らないというデフォルト設定を使用します。原則的に、インストールするシステムへの接続は、無期限に保たれるべきです。しかし、ある状況下 (あなたのローカルネットワークの設定に依存する) では、不使用時間が続くと接続を失う可能性があります。よくある状況は、クライアントとインストールするシステムの間のどこかに、ネットワークアドレス変換 (NAT) があることです。接続が失われた際のインストールのポイントにより、再接続後にインストールを再開できるかどうかが決まるでしょう。

ssh 接続を開始する際や、ssh の設定ファイルに、オプション `-o ServerAliveInterval=value` を追加して、接続が切れるのを回避できるかもしれません。しかしある状況下では、このオプションを追加すると、接続が切れる原因になるかもしれないことにご注意ください (例えば、普段なら ssh が復旧してしまうような、短時間のネットワーク障害中に keep-alive パケットを送るなど)。そのため、使用は必要最小限にするべきです。

注意

順番にいくつものコンピュータにインストールして、同じ IP アドレスやホスト名を持っていたりすると、ssh はそういったホストへの接続を拒否します。指紋が異なっているというのは、通常なりすまし攻撃のサインです。なりすまし攻撃ではないことが確かなら、`~/.ssh/known_hosts` から関連する行を削除して^a、もう一度行う必要があります。

^a以下のコマンドで、既存のホストエントリを削除できます。ssh-keygen -R <hostname|IP address>

ログインするとメニューの開始, シェルの開始という 2 つのメニューがある初期画面が表示されます。前者はメインのインストールメニューに移動し、通常のインストールを進めることができます。後者はリモートシステムの検査と (可能なら) 修正できるようなシェルを起動します。インストールメニュー用の SSH セッションを起動するのは 1 つだけにすべきですが、シェル用には複数のセッションを起動できます。

警告

SSH を使ってリモートでインストールを始めた後で、ローカルコンソールのインストールセッションに戻るべきではありません。新システムの設定を保持しているデータベースが破損する可能性があるからです。それによりインストールが失敗したり、インストールしたシステムに何か問題が発生するかもしれません。

6.4 見つからないファームウェアの読み込み

項 2.2 で述べたように、ある種のデバイスはファームウェアを読み込む必要があります。多くの場合、ファームウェアが有効でないとデバイスはまったく動作しません。場合によっては、ファームウェアがない場合、基本機能は損なわれませんが、追加機能を有効にするためだけにファームウェアが必要になります。

利用できないファームウェアをデバイスドライバが要求する場合、`debian-installer` は見つからないファームウェアを要求するダイアログを表示します。このオプションが選択されると、`debian-installer` はルーズなファームウェアファイルと、ファームウェアのあるパッケージの両方を、利用できるデバイスについて検索します。見つかると、ファームウェアを正しい場所 (`/lib/firmware`)

にコピーし、ドライバモジュールを再読み込みします。

注意



どのデバイスがスキャンされるか、どのファイルシステムをサポートしているかは、アーキテクチャやインストール方法、インストールの段階に依存します。特にインストールの初期段階では、ファームウェアの読み込みには、FAT フォーマットの USB メモリがもっとも成功の可能性が高いでしょう。

ファームウェアがなくてもデバイスが動作することを知っていたり、インストール中にそのデバイスが必要ない場合は、ファームウェアの読み込みをスキップできることにご注意ください。

debian-installer は、インストール中に読み込んだカーネルモジュールが必要とする場合にのみ、ファームウェアのプロンプトを表示します。debian-installer にすべてのドライバが含まれるわけではなく (特に radeon はありません)、そのため、いくつかのデバイスの機能は、はじめからインストールの終わりまで、まったく変わらないこともあります。その結果として、ハードウェアの能力を引き出せないかもしれません。この場合に該当するのかどうか疑わしく、興味を持たれたのであれば、新しくブートしたシステムで dmesg コマンドの出力をチェックし、「firmware」を検索してみるのもよいでしょう。

6.4.1 メディアの準備

公式インストールイメージには non-free のファームウェアが含まれません。そのようなファームウェアを読み込むもっとも一般的な方法は、USB メモリのようなリムーバブルメディアから読み込むことです。また、<https://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware/> に、non-free のファームウェアを含む非公式インストールイメージがあります。USB メモリ (ないし、ハードディスクのパーティションのような他のメディア) を準備するために、ファームウェアのファイルやパッケージを、メディアのファイルシステムのルートディレクトリか、/firmware というディレクトリのどちらかに配置しなければなりません。ファイルシステムには、インストールの初期段階でも間違いなくサポートされている、FAT を使用するのを勧めます。

よくある一般的なファームウェアの最新パッケージを、tar や zip で固めたものが、以下のサイトで利用できます。

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

正しいリリースの tarball か zip ファイルをダウンロードし、メディアのファイルシステムに展開するだけです。

必要なファームウェアが tarball に含まれていない場合、特定のファームウェアパッケージを、アーカイブ (の non-free セクション) からダウンロードすることもできます。以下の概観では、利用できるほとんどのファームウェアパッケージを網羅できると思いますが、完全であることを保証できませんし、ファームウェアではないパッケージを含むかもしれません。

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

また、個々のファームウェアファイルを、メディアにコピーもできます。ルーズなファームウェアを、例えばすでにインストールしたシステムや、ハードウェアベンダから入手できます。

6.4.2 ファームウェアとインストールしたシステム

インストール中に読み込んだファームウェアは、いずれもインストールしたシステムに、自動的にコピーされます。多くの場合、これによりシステムをリブートして新しいシステムにしても、ファームウェアが必要なデバイスが確実に動作するでしょう。しかし、インストールしたシステムが、インストーラと異なるバージョンのカーネルで動作している場合、バージョンのずれによりファームウェアが読み込めない可能性がわずかにあります。

ファームウェアパッケージからファームウェアをロードした場合、debian-installer はインストールしたシステム向けにも、このパッケージをインストールし、APT の sources.list に、パッケージアーカイブの non-free セクションを自動的に追加します。これには、ファームウェアの新しいバージョンが利用できるようになると、自動的に更新するという利点があります。

インストール中に、ファームウェアの読み込みをスキップした場合、おそらくファームウェア (パッケージ) を手動でインストールするまで、関連するデバイスがインストールしたシステムで動作しな

いでしょう。

注意



ファームウェアをルーズなファームウェアファイルから読み込んだ場合、インストール完了後に対応したファームウェアパッケージをインストールするまで、インストールしたシステムにコピーしたファームウェアは、自動的に更新されません。

6.4.3 Completing the Installed System

Depending on how the installation was performed, it might be that the need for some firmware was not detected during installation, that the relevant firmware was not available, or that one chose not to install some firmware at that time. In some cases, a successful installation can still end up in a black screen or a garbled display when rebooting into the installed system. When that happens, the following workarounds can be tried:

- Pass the `nomodeset` option on the kernel command line. This might help boot into a 「fallback graphics」 mode.
- Use the `Ctrl-Alt-F2` key combination to switch to VT2, which might offer a functional login prompt..

Once logged in into the installed system, it is possible to automate the detection of missing firmware, and to perform the required steps to enable them following this procedure:

1. Install the `isenkram-cli` package
2. Run the `isenkram-autoinstall-firmware` command as the 「root」 user.

Usually, rebooting is the simplest way to make sure all kernel modules are properly initialized; that's particularly important when one has booted the system with the `nomodeset` option as an interim measure.

注意



Installing firmware packages is very likely to require enabling the non-free section of the package archive. As of Debian GNU/Linux 11.0, running the `isenkram-autoinstall-firmware` command will do that automatically by creating a dedicated file (`/etc/apt/sources.list.d/isenkram-autoinstall-firmware.list`), pointing at a generic mirror.

Chapter 7

新しい Debian システムを起動してみる

7.1 決定的瞬間

新しいシステムが初めて自力で起動することを、電気技術者は「スモークテスト」と呼びます。

たとえシステムが正常に起動しなかったとしても、パニックにならないでください。インストーラが正常に終了したのなら、システムが Debian を起動するのを妨げる比較的小さな問題だけがある可能性が高いです。ほとんどの場合、そのような問題はインストーラを繰り返すことなしに解決することができます。ブート時の問題を修正する一つの選択肢は、インストーラ内蔵のレスキューモード(項8.6 をご覧ください)を使用することです。

もし Debian や Linux に不馴れなら、より経験のあるユーザの手助けが必要かもしれません。32-bit hard-float ARMv7 のようにそれほど一般的でないアーキテクチャでは、[debian-arm メーリングリスト](#) で尋ねるのが最も良い方法です。項5.4.5 にインストールレポートを提出することもできます。レポートには、問題についてははっきりと説明され、表示されたすべてのメッセージが含まれており、他の人が問題の原因を突き止める助けになるようにしてください。

7.2 暗号化ボリュームのマウント

インストール中に暗号化ボリュームを作成し、マウントポイントに割り当てると、そのボリュームに対して、起動中にパスワードを入力するように訊いてきます。

dm-crypt で暗号化したパーティションでは、起動中に以下のようなプロンプトが表示されます。

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

プロンプトの最初の行の part は、たとえば sda2 や md0 のような、基本的なパーティション名です。おそらく、ボリュームごとにパスワードを入力することに、違和感を覚えるのではないのでしょうか。これは /home や /var それぞれでパスワードを入力させられるのでしょうか? もちろんそうです。暗号化したボリュームが一つだけなら、話は簡単で、セットアップのときに入力したパスワードを入力するだけです。インストール時に、暗号化ボリュームを少なくとも一つは設定しているなら、項6.3.4.6 の最後のステップに書き留めたメモが役に立つでしょう。以前の part_crypt とマウントポイントの間のマッピングを記録しない場合、新しいシステムの /etc/crypttab と /etc/fstab にあります。

暗号化されたルートファイルシステムがマウントされる時は、プロンプトは少し違って見えるかもしれません。それは、システムの起動に使用される initrd を生成するために、どの initramfs ジェネレータが使われたかによります。以下の例は、initramfs-tools で生成された initrd の場合です。

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

パスワードの入力時には、入力した文字(やアスタリスク)は表示されません。パスワードを間違えた場合、訂正するために2回までは試行できます。入力を3回間違えると、そのボリュームをスキップして、次のファイルシステムをマウントしようとします。詳細は、項7.2.1 をご覧ください。

パスワードをすべて入力すると、通常と同様に起動を続けます。

7.2.1 トラブルシューティング

パスフレーズを間違えて、暗号化ボリュームをマウントできなかった場合、ブート後に手動でマウントする必要があります。以下の状況が考えられます。

- まずはじめの状況は、ルートパーティションに関することです。正しくマウントできないとブートプロセスが停止し、再起動してもう一度行わなければなりません。
- 最も考えられる状況は /home や /srv のようにデータを保持する暗号化ボリュームです。単純にブート後に手作業でマウントできます。

dm-crypt の場合は少しトリッキーです。まず device mapper を実行して、ボリュームを登録する必要があります。

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

/etc/crypttab に記述されたボリュームすべてを検査し、正しいパスフレーズを入力すると、/dev ディレクトリ以下に、適切なデバイスを作成します。(既に登録されたボリュームはスキップするので、何度実行しても警告がでません) 登録に成功すると、以下のように通常の方法でマウントできます。

```
# mount /mount_point
```

- クリティカルでないシステムファイルを扱うボリューム (/usr や /var) がマウントできなかった場合、それでもシステムが起動し、前述の状況のように手動でボリュームをマウントできるでしょう。しかし、デフォルトのランレベルで通常動作しているサービスを、起動していない可能性があるため、(再) 起動する必要があります。最も簡単なのはコンピュータの単純な再起動です。

7.3 ログイン

システムが起動するとすぐに、ログインプロンプトが現れます。インストールプロセス中にあなたが指定した一般ユーザのアカウント名とパスワードを入力して、ログインしてください。これで、システムは準備完了です。

初心者のユーザは、システムを使い始めながら、すでにインストールされている文書を読めると良いでしょう。現在はまだ文書システムが数種類存在しており、別々の形式の文書を統合するための作業が進められているところです。以下に出発点をいくつか示します。

インストールしたプログラムに付属する文書は、/usr/share/doc/ 以下のそのプログラム (より正確には、そのプログラムを含む Debian パッケージ) にちなんで命名されたサブディレクトリの下で見ることができます。しかし多くの場合、より豊富な文書が、独立した文書パッケージ (ほとんどの場合、デフォルトではインストールされません) として特別に用意されます。例えば、パッケージ管理ツール apt に関する文書は、apt-doc や apt-howto パッケージで見ることができます。

また、/usr/share/doc/ 階層構造の中には、いくつか特別なフォルダがあります。Linux HOWTO は、/usr/share/doc/HOWTO/en-txt/ の中に、.gz (圧縮) フォーマットで収められています。dhelp をインストールした後に、/usr/share/doc/HTML/index.html に拾い読みできる文書のインデックスを見つけるでしょう。

テキストベースのブラウザを使用して以下のコマンドを入力することで、以下のようにそれらの文書を簡単に見られます。

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

w3m コマンドの後のドットは、カレントディレクトリの内容を表示させるためのものです。

グラフィカルデスクトップ環境をインストールした場合には、Web ブラウザも利用できます。アプリケーションメニューから Web ブラウザを起動し、アドレスバーに /usr/share/doc/ と入力してください。

また、コマンドプロンプトから使えるほとんどのコマンドに対し、info コマンド または man コマンドによってその文書が参照できます。help と入力すると、シェルコマンドのヘルプが読めます。コマンドを --help つきで入力すると、たいていそのコマンドの簡単な使い方が表示されます。その結果が画面からスクロールして消えてしまう場合には、コマンドのあとに | more を追加すると、画面ごとに一時停止してくれます。ある文字で始まるコマンドの一覧を知りたいときは、その文字を入力してからタブを 2 回押します。

Chapter 8

次のステップとそれから

8.1 システムをシャットダウンする

稼働中の Debian GNU/Linux システムをシャットダウンする際には、コンピュータの前面や背面にあるリセットスイッチで再起動させたり、いきなり電源を落したりしてはいけません。Debian GNU/Linux は適切な手順でシャットダウンすべきで、さもないとファイルを失ったりディスクにダメージがもたらされたりします。デスクトップ環境を実行している場合は、通常システムのシャットダウン (または再起動) を可能にする、アプリケーションメニューから利用できる「ログアウト」用のオプションがあります。

もう一つの方法として、Ctrl-Alt-Del のキーを同時に押す方法が使えます。このキーの組合せが効かない場合、最後の選択肢として、root でログインして必要なコマンドを打ち込んでください。システムを再起動するには `reboot` と打ち込んでください。電源を入れたままでシステムを停止するには `halt` を使ってください¹。マシンの電源を落とすには `poweroff` または `shutdown -h now` を使ってください。systemd init システムでは例えば `systemctl reboot` や `systemctl poweroff` 等、同一の機能を果たす追加のコマンドを提供しています。

8.2 Debian に慣れる

Debian は他のディストリビューションとは少々異なっています。他のディストリビューションで Linux に精通された方でも、システムを整然とした状態に保つためには、Debian について知っておかなくてはならないことがあります。この章では Debian に慣れる手助けとなる資料を紹介します。Debian の使い方を逐一説明することは意図していません。すごく急いでいる人にシステムをざっとつかんでもらうだけのものです。

8.2.1 Debian パッケージングシステム

まず理解すべき最も重要な考え方に、Debian のパッケージングシステムがあります。基本的に、システムの大部分はパッケージングシステムの管理下にあると考えられています。このパッケージングシステムによって管理されるディレクトリには、以下のディレクトリが含まれています。

- /usr (/usr/local を除く)
- /var (/var/local を作成し、それ以下のディレクトリを自由に使うことは可能です)
- /bin
- /sbin
- /lib

例えば、`/usr/bin/perl` をあなたが別に用意したファイルで置き換えたとしても、その動作には問題はありません。ただし、後で perl パッケージを更新すると、あなたが置いたファイルはパッケージによ

¹SysV init システムでは `halt` は `poweroff` と同じ効果がありましたが、init システムが systemd (jessie からデフォルト) の場合は異なる効果があります

って置き換えられてしまいます。これを避けるには、`aptitude` でパッケージを「hold」(保留) するという操作を行います。

ベストなインストール方法の一つに `apt` があります。コマンドライン版の `apt` を使うことも、`aptitude` や `synaptic` (`apt` のグラフィカルなフロントエンド) のようなツールを使うこともできます。`apt` は `main`・`contrib`・`non-free` をまとめて扱うので、制限されている (正確にいうと Debian には含まれていない) パッケージを Debian GNU/Linux からのパッケージと同様な形で同時に取得できるという点に注意してください。

8.2.2 Debian で利用できる追加ソフトウェア

Debian のデフォルトインストールでは有効にならない、公式・非公式のソフトウェアリポジトリがあります。この中には、重要で当然インストールしておくソフトウェアもたくさんあります。こういった追加リポジトリの情報は、Debian Wiki の [The Software Available for Debian's Stable Release](#) ページにあります。

8.2.3 アプリケーションのバージョン管理

複数のバージョンがあるアプリケーションは、`update-alternatives` で管理されています。同種のアプリケーションを複数保守する場合は、`update-alternatives` の `man` ページをご覧ください。

8.2.4 cron ジョブ管理

システム管理者権限のもとで実行するジョブは、設定ファイルのある `/etc` に置いてください。毎日、毎週、毎月 `root` で実行する cron ジョブがあれば、`/etc/cron.{daily,weekly,monthly}` に置いてください。これらは `/etc/crontab` から呼び出され、アルファベット順に実行されます。

一方、特定のユーザで実行する必要がある cron ジョブや、特定の時間または頻度で実行する必要がある cron ジョブには、`/etc/crontab` あるいは `/etc/cron.d/whatever` が使えます (後者の方が望ましい)。これらのファイルには cron ジョブを実行するユーザアカウントを明記する特別なフィールドがあります。

どちらの場合も、ファイルを編集するだけで cron が自動的に実行してくれます。特別なコマンドを実行する必要はありません。詳しい情報は `cron(8)`、`crontab(5)`、`/usr/share/doc/cron/README.Debian` をご覧ください。

8.3 さらなる文書や情報

[Debian ウェブサイト](#) には、Debian に関するたくさんの文書があります。特に、[Debian GNU/Linux FAQ](#) と [Debian リファレンス](#) をご覧ください。[Debian ドキュメンテーションプロジェクト](#) には、Debian ドキュメンテーションに関するより多くのインデックスが用意されています。Debian のコミュニティでは、ユーザがお互いにサポートを行っています。Debian のメーリングリストを購読するには [メーリングリストの購読](#) ページをご覧ください。大事なことを言い忘れましたが、[Debian メーリングリストアーカイブ](#) には Debian に関する豊富な情報が含まれています。

もし、特定のプログラムに関する情報が必要ならば、まずは `man` プログラム名や `info` プログラム名を実行してみてください。

`/usr/share/doc` にも有用な文書がたくさんあります。特に、`/usr/share/doc/HOWTO` や `/usr/share/doc/FAQ` には興味深い情報がいくつもあります。バグを報告するには `/usr/share/doc/debian/bug*` をご覧ください。特定のプログラムについて Debian 固有の問題を読むためには `/usr/share/doc/(パッケージ名)/README.Debian` をご覧ください。

GNU/Linux の情報の一般的なソースは、[Linux Documentation Project](#) です。そこで、GNU/Linux システムの部分について、他の非常に価値ある情報のための HOWTO やポインタを得られるでしょう。

Linux は Unix 実装の一つです。[Linux Documentation Project \(LDP\)](#) では Linux に関するたくさんの HOWTO やオンライン書籍をまとめています。

Unix を初めてお使いになる方は、出かけて何冊か本を買い、少し読んでみるとよいでしょう。この [Unix FAQ のリスト](#) には、素晴らしい歴史的な参考文献を提供する UseNet ドキュメントがたくさん紹介されています。

8.4 電子メールを使用するためのシステム設定

今日では、電子メールは多くの人々にとって生活の重要な一部になっています。電子メールを使えるように設定するまでには、たくさんの選択肢があり、さらに電子メールが正確に設定されていることが重要になる Debian ユーティリティがあります。本節では、基本的なことのみ説明します。

電子メールシステムは、三つの主要な機能で構築されています。最初に、ユーザがメールを読み書きするために実際に使用するプログラムである Mail User Agent (MUA) があります。次に、あるコンピュータから別のコンピュータまでメッセージの転送処理をする Mail Transfer Agent (MTA) があります。そして最後に、ユーザの受信箱に受信メールの配送処理をする Mail Delivery Agent (MDA) があります。

これら三つの機能は個別のプログラムによって実行されますが、一つあるいは二つのプログラムに組み込むこともできます。また、異なるタイプのメールのために、これらの機能を処理する異なるプログラムを使用することもできます。

Linux や Unix システムにおいては、mutt が歴史的にとってもよく知られている MUA です。従来のほとんどの Linux プログラムがそうであるようにテキストベースのプログラムで、MTA として exim または sendmail、そして MDA として procmail と組み合わせがよく使用されます。

グラフィカルデスクトップシステムの人気の高まりとともに、GNOME の evolution、KDE の kmail、あるいは Mozilla の thunderbird として利用可能) のようなグラフィカルな電子メールプログラムの使用がより一般的になっています。これらのプログラムは、MUA、MTA および MDA の機能が組み合わされていますが、従来の Linux ツールと組み合わせることもでき—そして多くの場合は組み合わせて—使用されます。

8.4.1 デフォルトの電子メール設定

グラフィカルなメールプログラムを使用するつもりでいても、Debian GNU/Linux システムに従来の MTA/MDA をインストールし、正確に設定するのは有用かもしれません。システムで起動している様々なユーティリティ² が、システム管理者に (潜在的な) 問題や変更を通知するために、電子メールで重要な通知を送ることができるからです。

そのために、exim4 と mutt を `apt install exim4 mutt` でインストールできます。exim4 は、比較的小さなプログラムですが、とても柔軟性のある MTA/MDA の組み合わせです。デフォルトでは、システム内のローカルな電子メールの処理のためのために設定され、システム管理者 (root アカウント) 宛ての電子メールは、インストールの際に作成した標準のユーザアカウントに配送されます³。

システムから配送された電子メールは `/var/mail/account_name` 中のファイルに加えられます。メールは mutt を使って読むことができます。

8.4.2 システムの外に電子メールを送る

先に述べたように、インストールした Debian システムは、システム内のローカルな電子メールを処理するようにだけ設定され、他人にメールを送ったり、他人からメールを受け取ったりするには設定されません。

exim4 に外部の電子メールを処理させたい場合は、利用できる基本設定オプションに関して、次節を参照してください。メールが正しく送受信できることは、テストして確かめるようにしてください。

もしグラフィカルなメールプログラムを使って、インターネットサービスプロバイダ (ISP) あるいは会社のメールサーバを使用するつもりならば、外部の電子メールを処理するために exim4 を設定する必要は実際にはありません。電子メールを送受信するために、好みのグラフィカルなメールプログラムが正しいサーバを使用するようにただ設定するだけです (設定方法は本マニュアルでは扱いません)。

しかしその場合には、電子メールを正しく送れるように個々のユーティリティを設定する必要があるかもしれません。そのようなユーティリティの一つに、Debian パッケージに対するバグ報告の提出を容易にするプログラムである reportbug があります。デフォルトでは、バグ報告を提出するために exim4 が使用可能であることが期待されます。

外部のメールサーバを使用するように reportbug を正しく設定するため、`reportbug --configure` コマンドを実行し、MTA が利用可能かどうかという質問に「no」と答えてください。その後、バグ報告の提出に使用する SMTP サーバを尋ねられるでしょう。

²例えば: cron, quota, logcheck, aide, ...

³標準のユーザアカウントへの root 宛てのメールの転送は、`/etc/aliases` で設定します。標準のユーザアカウントを作成しなかった場合、もちろんメールは root アカウント自身に配送されます。

8.4.3 Exim4 Mail Transport Agent の設定

システムで外部の電子メールを処理するようにしたい場合、exim4 パッケージを再設定する必要があります⁴:

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

(root で) 上記のコマンドを入力した後に、設定ファイルを小さなファイルに分割するかどうか質問されます。よく分からない場合は、デフォルトオプションを選択してください。

次に、一般的な複数のメールシナリオが提示されます。あなたが必要としていることに最も近いものを一つ選択してください。

インターネットサイト システムはネットワークに接続され、SMTP を使用して直接メールを送受信します。次の画面で、マシンのメール名や受信あるいは中継するメールのドメインリストなどのような、いくつかの基本的な質問をされるでしょう。

スマートホストでメール送信 このシナリオでは、あなたの送信メールは、宛て先へのメッセージ送信処理をする「スマートホスト」と呼ばれる他のマシンに転送されます。通常、スマートホストは、あなたのコンピュータ宛てに送信された受信メールを保管するので、ずっとオンラインである必要はありません。つまりそれは、fetchmail のようなプログラムによって、スマートホストのメールをダウンロードしなければならないことを意味します。

多くの場合、スマートホストはあなたの ISP のメールサーバで、このオプションはダイヤルアップユーザにとっても適しています。またそれは、会社のメールサーバやあなた自身のネットワーク上の別のシステムとすることもできます。

スマートホストでメール送信; ローカルメールなし このオプションは、システムがローカルの電子メールアドレスを処理するようには設定されないという点を除いては、基本的に前のものと同じです。システム自体 (例えば、システム管理者のため) のメールは処理されます。

ローカル配信のみ システムがデフォルトで設定されるオプションです。

今は設定しない 内容を理解できていると絶対に確信している場合のみ選択してください。このシナリオは、メールシステムを未設定のままにします—メールシステムが設定されるまで、メールの送受信は一切できず、システムユーティリティからの重要なメッセージも逃してしまうかもしれません。

以上のどのシナリオもあなたの必要とするものに合っていない場合や、より精細な設定が必要な場合は、インストール完了後に /etc/exim4 ディレクトリの設定ファイルを編集する必要があります。exim4 に関するより多くの情報は、/usr/share/doc/exim4 ディレクトリにあります—README.Debian.gz ファイルには、exim4 の設定に関するその他の情報や、補足文書がどこで見つかるかなどの説明があります。

公式なドメイン名がない場合、インターネットに直接送信されたメールが受信サーバのスパム対策のために拒絶され、結果として不着メールとなる可能性があることに注意してください。ISP のメールサーバの使用が望まれます。それでもメールを直接送信したい場合には、デフォルトで生成されるものとは異なる電子メールアドレスを使用した方が良いでしょう。MTA として exim4 を使用するなら、/etc/email-addresses にエントリを追加することで可能です。

8.5 新しいカーネルのコンパイル

新しいカーネルをコンパイルしようとする動機はなんでしょう? Debian で提供している標準カーネルはほとんどの機能を利用できるようにしているので、あまり必要はないでしょう。

それでも独自のカーネルをコンパイルしたい場合はもちろんできます。その場合は「make deb-pkg」ターゲットの利用をお勧めしています。詳細については [Debian Linux Kernel Handbook](#) を参照してください。

⁴もちろん、exim4 を削除し、他の MTA/MDA を使用することもできます。

8.6 起動しなくなってしまったシステムの回復

時に物事は失敗し、慎重にインストールしたはずのシステムはもはや起動しません。おそらくブートローダの設定ファイルを編集しているうちに壊してしまったか、あるいはインストールした新しいカーネルでは起動しないか、ことによると宇宙線がディスクに命中して /sbin/init の中のビットがちょっと弾きとばされてしまったのかもしれませんが。原因のいかんを問わず、問題を修正する間に動作するようなシステムが必要になるでしょう。レスキューモードはそんな時に役に立ちます。

レスキューモードにアクセスするためには、ブートメニューから rescue を選択して boot: プロンプトで rescue とタイプするか、ブートパラメータに rescue/enable=true を指定して起動してください。インストーラの最初で、これがフルインストールではなくレスキューモードだということを知らせる注意書きが、ディスプレイの隅にほんの少し表示されます。心配しないでください、あなたのシステムが上書きされるわけではありません! レスキューモードは単に、システムを修復している間にディスクやネットワークデバイスなどが利用できることを確認するために、ハードウェア検出機能を利用します。

パーティション分割ツールの代わりに、システム上のパーティションリストが示され、それらのうちの一つを選択するよう尋ねられるでしょう。通常は、修復する必要のあるルートファイルシステムを含むパーティションを選択すべきです。ディスク上で直接作成されたパーティションと同様に RAID や LVM デバイス上のパーティションも選択できます。

可能であれば、インストーラは、選択したファイルシステムにおける、必要な修復を実行するために使えるシェルプロンプトを提供するようになっています。

選択したルートファイルシステムにあるシェルをインストーラが実行できない場合は、おそらくファイルシステムが壊れているので、インストーラは警告を発し、代わりにインストーラ環境でのシェルを提供することを提案します。この環境で利用できるツールは多くはありませんが、たいいていの場合、システムをとにかく復旧させるには充分でしょう。選択したルートファイルシステムは、/target ディレクトリにマウントされます。

いずれの場合でも、シェルを抜けた後にシステムが再起動します。

最後に。壊れてしまったシステムを修復するのは難しいことがあります。本マニュアルが、うまくいかない事や問題を修正する方法のすべてを説明しようとしているわけではないということに注意してください。もし問題があれば、専門家に相談してください。

Appendix A

インストール Howto

この文書は、新しい `debian-installer` で 32-bit hard-float ARMv7 (「armhf」) に Debian GNU/Linux bullseye をインストールする方法について説明します。これは、インストール作業の迅速なリハーサルで、たいていの導入のために必要となるであろうすべての情報を含んでいます。もっと多くの情報が有用な場合には、この文書内の他の部分にある、より詳細な説明にリンクします。

A.1 前置き

インストール中にバグに遭遇した場合には、それらを報告する方法の説明のために項5.4.5を参照してください。この文書で答えることができない質問があれば、`debian-boot` メーリングリスト (`debian-boot@lists.debian.org`) で直接質問するか、IRC (OFTC ネットワーク上の `#debian-boot`) で訊ねてください。

A.2 インストーラを起動する

`debian-cd` チームが `debian-installer` を使用してビルドしたインストールイメージは、[Debian CD/DVD ページ](#) から入手できます。どこでインストールイメージを手に入れられるかについてのより詳細に関しては、項4.1をご覧ください。

一部のインストール方法では、光学メディア用以外のイメージを必要とします。項4.2.1は、Debian ミラーサイトでイメージを探す方法について説明しています。

以下の小節では、インストール可能なそれぞれの手段のためにどのイメージを取得すべきかを詳しく説明します。

A.2.1 光学ディスク

`netinst` CD イメージは、`debian-installer` での `bullseye` のインストールに使用するのに一般的なイメージです。このインストール方法はこのイメージから起動し、ネットワーク越しに追加パッケージをインストールするように意図されているので、「`netinst`」という名前がついています。また、インストーラを実行するのに必要な、ソフトウェアコンポーネントと最小限の `bullseye` システムを提供する基本パッケージが含まれています。必要なら、ネットワークを必要としない、フルサイズ CD/DVD イメージを手に入れることもできます。その場合は一式の最初のイメージだけが必要です。

好みのタイプをダウンロードして、光学ディスクに焼いてください。

A.2.2 ネットワークからの起動

`debian-installer` をネットから完全に起動することもできます。`netboot` のための様々な方法は、アーキテクチャや `netboot` の設定に依存します。`netboot/` 以下のファイルは、`debian-installer` を `netboot` するために使用できます。

A.2.3 ハードディスクからの起動

リムーバブルメディアを使用せずに、単に既存のハードディスク (そこに異なる OS があっても構いません) を使ってインストーラを起動することができます。hd-media/initrd.gz、hd-media/vmlinuz および Debian CD/DVD イメージをハードディスクの一番上のディレクトリにダウンロードしてください。イメージのファイル名が .iso で終わっていることを確かめてください。これは initrd を使った Linux の起動時の問題です。

A.3 インストール

インストーラが立ち上がるとすぐに、歓迎の初期画面が表示されます。起動するために Enter を押すか、他の起動方法やパラメータのための説明を読んでください (項 5.3 をご覧ください)。

しばらくして、言語を選択するための質問がされます。矢印キーを使って言語を選び、継続するために Enter を押してください。次に、その言語が話される国々を含む選択肢が表示され、国を選択するよう質問されます。短いリスト上にはない場合は、世界中のすべての国のリストから選択できます。

キーボードレイアウトを確認するよう尋ねられるかもしれません。もしよく分からなければ、デフォルトを選択してください。

debian-installer がハードウェアの一部を検知し、インストールイメージの残りの部分をロードする間、くつろいでいてください。

次にインストーラは、ネットワークハードウェアを検知し、DHCP によってネットワークの設定をしようとします。ネットワーク上にないか、DHCP が無い場合は、ネットワークを手動で設定する機会が与えられます。

ネットワーク設定に続き、ユーザアカウントの作成を行います。デフォルトでは、「root」(管理者) アカウントのパスワードと、一般ユーザアカウントの作成に必要な情報を尋ねます。「root」ユーザのパスワードを指定しない場合、このアカウントは無効になります。新しいシステムで管理権限が必要なタスクを行うために、後で sudo パッケージをインストールすることになります。デフォルトではそのシステムで最初に作成されたユーザに、root になるための sudo コマンドを使う権限が与えられます。

次のステップは、時計とタイムゾーンの設定です。インストーラは、時計が正しく設定されるのを保証するため、インターネット上のタイムサーバに接続します。タイムゾーンは、あらかじめ選択した国を元にしますが、その国に複数のゾーンがある場合のみ問い合わせてきます。

さあ、ディスクのパーティションを分割しましょう。最初に、ドライブのすべてか、またはドライブの利用可能な空き領域を自動的にパーティション分割するか選択する機会が与えられます (項 6.3.4.2 を参照)。これは新規ユーザや急いでいる誰にでも勧められます。自動分割をしたくない場合は、メニューから手動を選んでください。

次の画面でパーティションテーブル (パーティションをどうフォーマットするか、それをどこにマウントするか) を見ることとなります。修正や削除をするためには、パーティションを選択してください。もし自動パーティション分割を行っていただければ、設定したものを使用するメニューから、パーティショニングの終了とディスクへの変更の書き込みで決定できます。スワップスペースのために少なくとも 1 つのパーティションを割り当てることと / にパーティションをマウントすることを忘れないようにしてください。パーティション分割ツールの使い方に関する詳細情報は、項 6.3.4 をご覧ください。また、付録の付録 C にパーティション分割に関するもっと多くの情報があります。

それから debian-installer はパーティションをフォーマットし、基本システムのインストール (時間がかかることがあります) を始めます。続いてカーネルがインストールされます。

最初にインストールした基本システムでも動作はしますが、最低限のものしかインストールされていません。もっと機能的にするには、次のステップでタスクを選択し、追加パッケージをインストールしてください。なお、パッケージをインストールする前に、パッケージをどこから取得してインストールするかを定義を、apt に設定する必要があります。「標準システム」タスクはデフォルトで選択され、通常は既にインストールされているはずですが、インストール後にグラフィカルデスクトップが必要であれば、「デスクトップ環境」を選択してください。このステップについてのさらなる情報は、項 6.3.6.2 をご覧ください。

最後の段階はブートローダをインストールすることです。コンピュータ上に他のオペレーティングシステムを検出した場合は、インストーラがブートメニューにそれらを加えて知らせます。

次に debian-installer は、インストールが終了したことを伝えます。CD-ROM やその他の起動メディアを取り出して、マシンを再起動するために Enter を叩いてください。新しくインストールしたシステムが起動し、ログインできるはずですが、これは第 7 章で説明しています。

インストール手順についてもっと多くの情報が必要ならば、第 6 章をご覧ください。

A.4 インストールレポートを送ってください

debian-installer で首尾よくインストールをやり遂げられたならば、レポート提出のためにしばらく時間をかけてください。reportbug パッケージをインストールして (apt install reportbug)、項8.4.2の説明にあるように reportbug を設定し、reportbug installation-reports と実行するのが最も簡単な方法です。

もしインストールが完了しなかったのならば、おそらく debian-installer のバグを発見しました。インストーラを改善するためには、私たちがそれらについて知っていることが必要ですので、バグ報告するための時間をとってください。問題を報告するためにはインストールレポートが使用できます。インストールが完全に失敗する場合は、項5.4.4をご覧ください。

A.5 そして最後に…

Debian のインストールが快適であり、Debian が役に立つことに気づいていただければと思います。第8章を読むのが良いでしょう。

Appendix B

preseed を利用したインストールの自動化

本付録は preseed の方法を説明します。これは debian-installer の質問に回答しておきインストールを自動化するものです。

本付録で使用する設定の断片は、<https://www.debian.org/releases/bullseye/example-preseed.txt> のサンプル事前設定ファイルでも利用できます。

B.1 概要

preseed は、インストールの実行中に手動で回答を入力せずに、インストールプロセス中の質問の答を設定する方法を提供します。これにより、ほとんどの方法のインストールを自動化し、さらに通常のインストールでは利用できない特徴もあります。

preseed は必須ではありません。空の preseed ファイルを使用すると、インストーラは通常の手動インストールと同じ振る舞いをします。preseed した各質問は、(正しく与えていれば!) ベースラインからと同じ方法で、インストールの内容を変更します。

B.1.1 preseed の方法

preseed を利用するには、initrd, file, network と 3 種類の方法があります。initrd preseed は、いずれのインストール方法でも動作し、より多くの preseed をサポートしますが、多くの準備が必要です。file preseed や network preseed は、それぞれインストール方法が異なる場合に使用されます。

以下の表では、各インストール方法で使用できる preseed 方法を示します。

インストール方法	initrd	file	network
CD/DVD/USB	yes	yes	yes ¹
netboot	yes	no	yes
hd-media	yes	yes	yes ¹

preseed 方法の重要な違いは、事前設定ファイルを読み込・処理するポイントです。initrd preseed では、インストールの始め(最初の質問が行われる前)に読み込まれます。カーネルコマンドラインからの preseed がその直後に続きます。そのため、(可能なブートローダの場合)ブートローダの設定、またはブートローダのブート時に手作業でカーネルコマンドラインを編集することにより initrd 中での設定状況に上書きできます。file preseed では、インストールイメージが読み込まれた後です。network preseed では、ネットワークの設定の後でないと読み込まれません。

¹ネットワークアクセスを行う場合だけでなく、適切な preseed/url を設定する場合。

重要項目

言うまでもなく、事前設定ファイルが読み込まれる前に処理される質問は preseed できません (最初のハードウェア検出のように、優先度が中や低でしか表示されない質問も同様です)。あまり便利ではありませんが項B.2.2 で説明しているように、ブートパラメータを経由して preseed することでそういった質問が出ないようにする方法があります。

preseed が起動する前に、通常現れる質問を簡単に回避するのに、「自動」モードでインストーラを起動できます。これによりネットワークの設定が終わるまで、preseed の前に行われる質問 (言語、国、キーボード選択など) を遅らせ、preseed にその質問を含められます。また、インストールの優先度を最重要で行うため、大量にある重要でない質問を回避できます。詳細は項B.2.3 をご覧ください。

B.1.2 制限

debian-installer で行われる質問のほとんどはこの方法で preseed できますが、いくつか注目すべき例外があります。ディスク全体を (再度) パーティション分割するか、ディスクの空き領域を利用しなければなりません。つまり既存のパーティションを利用できないということです。

B.2 preseed の利用

事前設定ファイルを最初に作成し、使用する場所に配置する必要があります。事前設定ファイルの作成は本付録で後ほど扱います。network preseed の場合や、ファイルを USB メモリから読み込む場合、簡単に正しい位置に事前設定ファイルを配置できます。インストール ISO イメージにファイルを含めたいければ、イメージを再度マスタリングする必要があります。initrd に含まれている事前設定ファイルを取り出す方法は、この文書では扱いません。debian-installer の開発者向け文書を当たってください。

事前設定ファイルの手本にできる事前設定ファイルのサンプルは、<https://www.debian.org/releases/bullseye/example-preseed.txt> から取得できます。このファイルは、この付録にある設定の断片を元にしてしています。

B.2.1 事前設定ファイルの読み込み

initrd preseed を使用する場合、preseed.cfg というファイルが initrd のルートディレクトリに確実にある必要があります。インストーラは、このファイルがあるか自動的にチェックし、読み込みます。

他の preseed 方法では、起動時にどのファイルを読み込むか、インストーラに指定する必要があります。通常、カーネルのブートパラメータで渡して行きます。これは起動時に手で与えるか、ブートローダ設定ファイル (例: syslinux.cfg) を編集し、カーネルへの append 行の最後にパラメータを追加します。

ブートローダの設定で事前設定ファイルを指定する場合、設定を変更すれば、インストーラの起動時に ENTER を押す必要はありません。syslinux ではこの設定をするのに、syslinux.cfg でタイムアウトを 1 にします。

インストーラが確実に正しい事前設定ファイルを取得するのに、このファイルのチェックサムを指定できます。現在、これには md5sum 値の指定が必要です。指定した値と事前設定ファイルの値は一致しなければなりません。一致しない場合は、インストーラは事前設定ファイルを使用しません。

ブートパラメータの設定:

- netboot の場合:
 - preseed/url=http://host/path/to/preseed.cfg
 - preseed/url/checksum=5da499872becccfeda2c4872f9171c3d
- または
 - preseed/url=tftp://host/path/to/preseed.cfg
 - preseed/url/checksum=5da499872becccfeda2c4872f9171c3d

- リマスタリングしたインストールイメージで起動する場合:
preseed/file=/cdrom/preseed.cfg
preseed/file/checksum=5da499872becccfeda2c4872f9171c3d
- USB メディアで起動する場合 (事前設定ファイルを USB メモリの
トップレベルディレクトリに置くこと):
preseed/file=/hd-media/preseed.cfg
preseed/file/checksum=5da499872becccfeda2c4872f9171c3d

ブートパラメータに渡す際に、preseed/url は url に、preseed/file は file に、preseed/file/checksum は preseed-md5 に短縮できることに注意してください。

B.2.2 preseed が質問するブートパラメータの利用

事前設定ファイルを preseed の各段階で使用できない場合でも、preseed の値をインストーラ起動時のコマンドラインに与えることで、インストールを自動で行えます。

preseed を使用せず指定した質問への答を設定したい場合にも、ブートパラメータを使用します。有用な使用法のサンプルが、このマニュアルの別の場所にあります。

debian-installer 内部で使用する値をセットするには、path/to/variable=value のように本付録の例にある preseed 変数を渡すだけです。値がターゲットシステムのパッケージを設定することがある場合、owner² 変数を、あらかじめ用意し、owner:path/to/variable=value で使用する必要があります。owner を指定しない場合、変数の値はターゲットシステムの debconf データベースにコピーされず、関連パッケージの設定中使用されません。

通常、この方法で答をあらかじめ設定しておく、質問してきません。質問のデフォルト値を指定しているのに、まだ質問してくる場合には、「=」演算子の代わりに、「?=」を使用してください。項B.5.2 もご覧ください。

ブートプロンプトによく使用される変数には、短いエイリアスがあることに注意してください。有効なエイリアスは、本サンプル内で完全な変数名の代わりに使用しています。例えば preseed/url 変数には url というエイリアスがあります。もう一つ、tasks というエイリアスがあり、これは tasksel:tasksel/first に変換されます。

ブートオプションの「---」は特別な意味を持ちます。最後の「---」に続きカーネルパラメータがあると、(インストーラがサポートするブートローダの場合) インストール済みのブートローダの設定にコピーされます。インストーラは、(事前設定オプションのような) オプションを認識すると、自動的にフィルタをかけます。

注意



現在の Linux カーネル (2.6.9 以降) では、最大 (インストーラがデフォルトで指定するオプションを含め) コマンドラインオプションを 32 個、環境オプションを 32 個受け取れます。この数を超えると、カーネルはパニック (クラッシュ) してしまいます (以前のカーネルではこの数字がもっと少ないです)。

(ほとんどのインストールでは、ブートローダ設定ファイルにある (vga=normal のような) デフォルトオプションを安全に削除できるかもしれませんが、これにより preseed 用にもっとオプションを追加できます。

注意



ブートパラメータに空白を含んだ値を設定するのは、引用符で囲んだとしてもいつもうまくいくとは限りません。

²debconf 変数 (やテンプレート) の所有者 (owner) は、debconf テンプレートに含まれるように、通常パッケージ名です。インストーラ自体が使用する値は、「d-i」になっています。テンプレートや変数は、複数の owner を持て、パッケージを完全削除する際に debconf データベースから削除できるかどうかを決定するのに利用されます。

B.2.3 自動モード

ブートプロンプトでの非常に簡単なコマンドラインで、自動インストールに対して任意の複雑なカスタマイズを行えるよう組み合わせる、Debian インストーラの機能がいくつかあります。

これは起動時の選択からインストールの自動化を使うことで有効化できます。アーキテクチャや起動方法によっては auto と呼ばれます。ここでは、auto パラメータではなく、起動時の選択でそれを選択して起動時のプロンプトに以下のパラメータを付加するという意味になります。

これを説明するため、以下にブートプロンプトで使用できる例を示します。

```
auto url=autoserver
```

これは、DNS で autoserver の名前解決ができ (おそらく DHCP でローカルドメイン追加後)、そのマシンが DHCP サーバであることを前提にしています。example.com というドメインのサイトが、普通のまともな DHCP を設定していれば、http://autoserver.example.com/d-i/bullseye/./preseed.cfg から、preseed ファイルを取得するようになります。

URL (d-i/bullseye/./preseed.cfg) の最後の部分は、auto-install/defaultroot から取られています。デフォルトでは、将来のバージョンでコードネームを指定して移行していけるように、bullseye ディレクトリが含まれています。./ は、その後続くパスが確定するように、ルートからの相対パスを示します (preseed/include や preseed/run で使用)。これにより、完全な URL や / で始まるパス、前回 preseed が見つかった場所からの相対パスでファイルを指定できます。スクリプトの階層構造を壊さずに新しい場所に移動できる (例えばウェブサーバで開始し、USB メモリにコピーする)、よりポータブルなスクリプトを構成するのに便利です。このサンプルでは、preseed ファイルの preseed/run に /scripts/late_command.sh が設定されている場合、http://autoserver.example.com/d-i/bullseye/./scripts/late_cからファイルを取得します。

手元に DHCP や DNS のインフラがない場合や、preseed.cfg のデフォルトパスを使用したくない場合でも、きちんとした URL を使用でき、./ 要素を使用しない場合は、パスの開始点を決定できます (例えば URL の 3 つ目の /)。以下は、手元のネットワークインフラから最低限必要な物のサンプルです。

```
auto url=http://192.168.1.2/path/to/mypreseed.file
```

この方法は次のように動作します。

- URL が見つからない場合、http だと仮定します。
- ホスト名セクションにピリオドがなければ、DHCP から引き出して追加します。
- ホスト名の後に / がなければ、デフォルトパスを追加します。

URL を指定するのに加えて、debian-installer 自身の振る舞いには直接影響しない設定も追加できますが、読み込んだ preseed ファイルの preseed/run で指定した、スクリプトに渡すことができます。現在のところ、classes というエイリアスを持つ、auto-install/classes のサンプルのみです。以下のように使用します。

```
auto url=example.com classes=class_A;class_B
```

classes にはこのサンプルでは、インストールするシステムのタイプや、地域化を指定するのに使用できます。

この概念はもちろん拡張でき、もしそうする場合、auto-install 名前空間を使用するのが妥当です。ですから、次にあなたのスクリプトで使用する auto-install/style のような物かもしれません。これが必要だと思うのなら、名前空間の衝突を避けるために debian-boot@lists.debian.org メーリングリストで提案してください。おそらくパラメータのエイリアスが追加されます。

auto ブートの選択肢は、まだ全てのアーキテクチャで定義されてはいません。カーネルのコマンドラインに、単にパラメータを 2 つ auto=true priority=critical を追加すると、同じ効果を得られます。auto カーネルパラメータは auto-install/enable のエイリアスで、true に設定するとロケールやキーボードの質問を preseed で行えるよう遅らせます。また、priority は debconf/priority のエイリアスで、critical に設定すると、優先度の低い質問を抑制するようになります。

DHCP を使用してインストールの自動化を行う際に、関連する追加オプションは以下の通りです。interface=auto netcfg/dhcp_timeout=60 これはマシンが最初の使用可能 NIC を選択し、DHCP 問い合わせに対する返答をもっと我慢強く待つようになります。

ティップ



スクリプトやクラスのサンプルを含む、フレームワークの使用法についての大規模なサンプルが、[開発者のウェブサイト](#)にあります。そこで得られるサンプルでも、事前設定の独創的な使用を成し遂げる、たくさんのすばらしい効果があります。

B.2.4 preseed で利用できるエイリアス

以下のエイリアスは、(自動モード) preseed の際に、役に立つ場合があります。これらは単に、質問名の短いエイリアスであることに注意してください。そのうえで、常に値を指定する必要があります。例えば、`auto=true` や `interface=eth0` のようにです。

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 ブートプロンプトの preseed の例

ここではブートプロンプトの見た目の例を示します (これを必要に応じて調整してください)。

```
# To set French as language and France as country:
/install.amd/vmlinuz vga=788 initrd=/install.amd/gtk/initrd.gz language=fr ↔
country=FR --- quiet
# To set English as language and Germany as country, and use a German keyboard ↔
layout:
/install.amd/vmlinuz vga=788 initrd=/install.amd/gtk/initrd.gz language=en ↔
country=DE locale=en_US.UTF-8 keymap=de --- quiet
# To install the MATE desktop:
/install.amd/vmlinuz vga=788 initrd=/install.amd/gtk/initrd.gz desktop=mate- ↔
desktop --- quiet
# To install the web-server task:
/install.amd/vmlinuz initrd=/install.amd/initrd.gz tasksel:tasksel/first=web- ↔
server ---
```

B.2.6 事前設定ファイルを指定するための DHCP の利用方法

事前設定ファイルをネットワークからダウンロードするよう指定するのに、DHCP も使用できます。DHCP はファイル名の指定ができます。通常これは netboot のファイルですが、URL 形式になっていると、network preseed をサポートするインストールメディアが、URL からファイルをダウンロードし、事前設定ファイルとして使用します。以下は、ISC DHCP サーバのバージョン 3 用 dhcpd.conf で設定するサンプルです。

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

上記の例は、「d-i」を名乗る DHCP クライアントにこのファイル名を渡すよう制限されており、通常の DHCP クライアントではなく、インストーラにのみ影響を与えることに注意してください。この文字列で、ネットワーク上の全マシンに preseed でインストールするのではなく、特定のホストに対して行うようにもできます。

DHCP preseed を使用するよい方法は、自分のネットワークには、Debian ミラーサイトのような preseed の値のみ指定することです。自分のネットワークにこの方法でインストールすると、選択したよいミラーサイトから自動で取得しますが、インストールの残りのプロセスはインタラクティブに行われます。DHCP preseed を用いた Debian の完全自動インストールは、充分注意しなければ行わべきではありません。

B.3 事前設定ファイルの作成

事前設定ファイルのフォーマットは、debconf-set-selections コマンドで使用されるものと同じです。事前設定ファイルの行の一般的なフォーマットは以下のようになります。

```
<所有者> <質問名> <質問タイプ> <値>
```

このファイルの内容は #_preseed_V1 から始まります

事前設定ファイルを記述する際には、ちょっとした規則があると気に留めておいてください。

- 型と値の間には、空白かタブを 1 つだけおいてください。空白を追加すると、値の一部として解釈されます。空値を指定する場合は、型の後に空白かタブを、ひとつ記述したままにしてください。
- 行継続文字としてバックスラッシュ (「\」) を付けて複数行に分割できます。質問名の後で分割するのが適当でしょう。型と値の間はよくありません。値の途中で行を分割するのは、パーティション分割のレシピを除いて、サポートしていません。
- インストーラ自身でのみ使用する debconf 変数 (テンプレート) では、所有者を「d-i」と設定しておきます。インストールしたシステムで使用する preseed 変数では、対応する debconf テンプレートを含むパッケージ名を使用すべきです。所有者が「d-i」ではない変数だけを、インストールしたシステムの debconf データベースに伝播させます。
- ほとんどの質問では、訳した値ではなく英語の値を指定する必要がありますが、(partman など) 訳した値を使用できる質問もあります。
- 質問の中には、インストール中に表示される英語のテキストの代わりに、コードを取るものがあります。
- #_preseed_V1 から始まります
- コメントは先頭がハッシュ記号 (「#」) の行で、その行の最後まで続きます。

事前設定ファイルを作成する簡単な方法は、項 B.4 にあるサンプルファイルを元にして作業することです。

その他には、手動インストールを行い、再起動してから debconf-utils パッケージの debconf-get-selections を使用します。以下のように debconf データベースとインストーラの cdebconf データベースを 1 ファイルに出力してください。

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

しかし、この方法で生成したファイルでは preseed されない項目があります。ほとんどのユーザはサンプルファイルから始めるのがよいでしょう。

注意



この方法は、インストーラの cdebconf データベースが、インストールしたシステムの /var/log/installer/cdebconf に保存されているのを前提としています。しかし、データベースに機密情報が含まれる可能性がありますので、デフォルトでは root にのみ読み込みが許可されています。

/var/log/installer ディレクトリとその中のファイルは、installation-report パッケージを完全削除することで、削除されます。

有効な質問の値をチェックするのに、インストール中に /var/lib/cdebconf のファイルを、nano を使用して確認できます。生のテンプレートは templates.dat を、現在の値や変数に割り当てられた値は questions.dat を確認してください。

インストールを実行する前に、事前設定ファイルのフォーマットが適切かどうかを調べるには、debconf-set-selections -c preseed.cfg が使えます。

B.4 事前設定ファイルの内容 (bullseye 用)

本付録で使用する設定の断片は、<https://www.debian.org/releases/bullseye/example-preseed.txt> のサンプル事前設定ファイルでも利用できます。

本サンプルは、Intel x86 アーキテクチャ用インストールを元に行っていることに注意してください。他のアーキテクチャにインストールする場合、サンプルのいくつか (キーボードの選択やブートローダの選択など) は適切でないかもしれませんので、そのアーキテクチャ用に適切な debconf 設定で置き換える必要があるでしょう。

Debian インストーラコンポーネントの実際の動作が、どのように異なるのかといった詳細は、項 6.3 にあります。

B.4.1 地域化

通常のインストール中、地域化について最初に質問されるため、地域化に関する質問は initrd またはカーネルブートパラメータによる方法でのみ preseed 可能となります。自動モード (項 B.2.3) では auto-install/enable=true の設定を (通常は preseed の別名 auto 経由で) 盛り込むこととなります。それにより地域化の質問を聞いてくる時機が後になるため、どの方法でも preseed できるようになります。

ロケールは言語と国を両方指定でき、debian-installer がサポートする言語と認識する国のいずれかの組み合わせでもかまいません。組み合わせが正しいロケールの形になっていない場合、インストーラは選択した言語から正しいロケールを自動選択します。ブートパラメータでロケールを指定するには、locale=en_US としてください。

この方法は非常に簡単ですが、言語・国・ロケールの利用可能な組み合わせを、すべて preseed できるわけではありません³。言語と国は、どちらもブートパラメータで指定できます。

```
# Preseeding only locale sets language, country and locale.
d-i debian-installer/locale string en_US
```

```
# The values can also be preseeded individually for greater flexibility.
#d-i debian-installer/language string en
#d-i debian-installer/country string NL
#d-i debian-installer/locale string en_GB.UTF-8
```

³例えば、preseed で locale を en_NL とすると、インストールしたシステムのデフォルトロケールは en_US.UTF-8 になります。例えば en_GB.UTF-8 を期待するのであれば、preseed にその値を設定する必要があります。

```
# Optionally specify additional locales to be generated.
#d-i localechooser/supported-locales multiselect en_US.UTF-8, nl_NL.UTF-8
```

キーボード設定は、キーマップの選択と、(非ラテンキーマップ向けの)非ラテンキーマップと US キーマップとの、切り替えキーの選択から成っています。インストール中では基本的なキーマップバリエーションしか有効ではありません。詳細なバリエーションは、インストールしたシステムで `dpkg-reconfigure keyboard-configuration` を実行することでのみ有効になります。

```
# Keyboard selection.
d-i keyboard-configuration/xkb-keymap select jp
# d-i keyboard-configuration/toggle select No toggling
```

キーボード設定をスキップするには、`keymap` を `skip-config` と `preseed` してください。これにより、カーネルのキーマップが有効になったままとまります。

B.4.2 ネットワーク設定

もちろん、ネットワークから事前設定ファイルを読み込む場合、`preseed` のネットワーク設定は動作しません。しかし、光学ディスクや USB メモリから起動するときには重要です。ネットワークから事前設定ファイルを読み込む場合、ネットワーク設定パラメータは、カーネルブートパラメータで渡すこととなります。

ネットワークから事前設定ファイルを読み込む前に `netboot` するとき、特定のインターフェースを選ぶ必要があるなら、`interface=eth1` のようにブートパラメータを使用してください。

「`preseed/url`」で `network preseed` を使用する際、ネットワーク設定の `preseed` は通常不可能ですが、例えば、ネットワークインターフェースに静的アドレスを設定するといった、以下のハックを利用して動作させることができます。このハックは、以下のコマンドを含む「`preseed/run`」スクリプトを作成し、事前設定ファイルを読み込んだ後でネットワークの設定を強制的に再度行う、というものです。

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

以下の `debconf` 変数は、ネットワークの設定と関係があります。

```
# Disable network configuration entirely. This is useful for cdrom
# installations on non-networked devices where the network questions,
# warning and long timeouts are a nuisance.
#d-i netcfg/enable boolean false

# netcfg will choose an interface that has link if possible. This makes it
# skip displaying a list if there is more than one interface.
d-i netcfg/choose_interface select auto

# To pick a particular interface instead:
#d-i netcfg/choose_interface select eth1

# To set a different link detection timeout (default is 3 seconds).
# Values are interpreted as seconds.
#d-i netcfg/link_wait_timeout string 10

# If you have a slow dhcp server and the installer times out waiting for
# it, this might be useful.
#d-i netcfg/dhcp_timeout string 60
#d-i netcfg/dhcpv6_timeout string 60

# If you prefer to configure the network manually, uncomment this line and
# the static network configuration below.
#d-i netcfg/disable_autoconfig boolean true

# If you want the preconfiguration file to work on systems both with and
# without a dhcp server, uncomment these lines and the static network
# configuration below.
#d-i netcfg/dhcp_failed note
#d-i netcfg/dhcp_options select Configure network manually
```

```
# Static network configuration.
#
# IPv4 example
#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 example
#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

# Any hostname and domain names assigned from dhcp take precedence over
# values set here. However, setting the values still prevents the questions
# from being shown, even if values come from dhcp.
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain

# If you want to force a hostname, regardless of what either the DHCP
# server returns or what the reverse DNS entry for the IP is, uncomment
# and adjust the following line.
#d-i netcfg/hostname string somehost

# Disable that annoying WEP key dialog.
d-i netcfg/wireless_wep string
# The wacky dhcp hostname that some ISPs use as a password of sorts.
#d-i netcfg/dhcp_hostname string radish

# If non-free firmware is needed for the network or other hardware, you can
# configure the installer to always try to load it, without prompting. Or
# change to false to disable asking.
#d-i hw-detect/load_firmware boolean true
```

netcfg/get_netmask が preseed されていない場合、netcfg は自動的にネットマスクを決定することに注意してください。この場合、自動インストールを行うためには、この変数を seen としてマークされていなければなりません。同様に、netcfg/get_gateway が設定されていないと、netcfg は適切なアドレスを選択します。特殊な場合として、netcfg/get_gateway に「none」と設定して、ゲートウェイを使用しないようにできます。

B.4.3 ネットワークコンソール

```
# Use the following settings if you wish to make use of the network-console
# component for remote installation over SSH. This only makes sense if you
# intend to perform the remainder of the installation manually.
#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 ミラーサイト設定

使用するインストール方法に依存しますが、インストーラの追加コンポーネントのダウンロードや、基本システムのインストール、インストールしたシステムの /etc/apt/sources.list のセットアップにミラーサイトを使用できます。

mirror/suite パラメータでは、インストールするシステム用の組を設定します。

mirror/udeb/suite パラメータでは、インストーラの追加コンポーネントの組を設定します。実際にコンポーネントをネットワークでダウンロードする場合に役立つだけです。また、インストールで使用するインストール方法のための `initrd` を生成するには、この組が一致していなければなりません。通常インストーラは、自動的に正しい値を設定しますので、設定する必要はありません。

```
# If you select ftp, the mirror/country string does not need to be set.
#d-i mirror/protocol string ftp
d-i mirror/country string manual
d-i mirror/http/hostname string http.us.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

# Suite to install.
#d-i mirror/suite string testing
# Suite to use for loading installer components (optional).
#d-i mirror/udeb/suite string testing
```

B.4.5 アカウント設定

root アカウント用のパスワードや、最初のユーザアカウントの名前・パスワードは `preseed` できます。パスワードには、平文か `crypt(3)` ハッシュのどちらかを使用できます。

警告



パスワードを知っている事前設定ファイルに誰でもアクセスできるため、`preseed` のパスワードは、完全に安全というわけではないことを覚えておいてください。保存するパスワードをハッシュ化することで、総当たり攻撃を許す DES や MD5 のような弱いハッシュ化アルゴリズムを使わない限りは安全だと考えられます。ハッシュ化アルゴリズムとしては SHA-256 か SHA512 を勧めます。。

```
# Skip creation of a root account (normal user account will be able to
# use sudo).
#d-i passwd/root-login boolean false
# Alternatively, to skip creation of a normal user account.
#d-i passwd/make-user boolean false

# Root password, either in clear text
#d-i passwd/root-password password r00tme
#d-i passwd/root-password-again password r00tme
# or encrypted using a crypt(3) hash.
#d-i passwd/root-password-crypted password [crypt(3) hash]

# To create a normal user account.
#d-i passwd/user-fullname string Debian User
#d-i passwd/username string debian
# Normal user's password, either in clear text
#d-i passwd/user-password password insecure
#d-i passwd/user-password-again password insecure
# or encrypted using a crypt(3) hash.
#d-i passwd/user-password-crypted password [crypt(3) hash]
# Create the first user with the specified UID instead of the default.
#d-i passwd/user-uid string 1010

# The user account will be added to some standard initial groups. To
# override that, use this.
#d-i passwd/user-default-groups string audio cdrom video
```

passwd/root-password-crypted 変数や passwd/user-password-crypted 変数では、preseed で「!」という値を取れます。この場合、そのアカウントは無効となります。もちろん管理権限での実行や root ログインを許可する代替手段 (例えば SSH キー認証や sudo) を用意しておいた上で、root アカウントに設定すると便利です。

以下のコマンド (whois パッケージから利用できます) を、パスワードの SHA-512 ベースの crypt(3) ハッシュを生成するのに利用できます。

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

B.4.6 時計と時間帯の設定

```
# Controls whether or not the hardware clock is set to UTC.
d-i clock-setup/utc boolean true

# You may set this to any valid setting for $TZ; see the contents of
# /usr/share/zoneinfo/ for valid values.
d-i time/zone string US/Eastern

# Controls whether to use NTP to set the clock during the install
d-i clock-setup/ntp boolean true
# NTP server to use. The default is almost always fine here.
#d-i clock-setup/ntp-server string ntp.example.com
```

B.4.7 パーティション分割

ハードディスクのパーティション分割に preseed を使用するのには、partman-auto でサポートしている機能に限定されています。パーティションはディスクに既存の空き領域とディスク全体のどちらかから選べます。ディスクレイアウトは、あらかじめ定義したレシピ、レシピファイルによるカスタムレシピ、事前設定ファイルに書いたレシピから選択できます。

RAID、LVM、暗号化を用いた高度なパーティションセットアップを、preseed ではサポートしていますが、preseed を用いずにインストールしたときに使用できるような、完全な柔軟性があるわけではありません。

以下の例は、レシピを使用する際の基本的な情報のみを提供しています。詳細情報は、debian-installer パッケージにある、partman-auto-recipe.txt と partman-auto-raid-recipe.txt をご覧ください。どちらのファイルも [debian-installer ソースリポジトリ](#) にもあります。リリースごとに、サポートする機能が変更されることがありますので、注意してください。

警告



ディスクの識別は、ドライバの読み込み順に依存します。複数のディスクがシステムにある場合、preseed を使用する前に、正しいディスクを確実に選択できるようにしなければなりません。

B.4.7.1 パーティション分割の例

```
# 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

# You can define the amount of space that will be used for the LVM volume
# group. It can either be a size with its unit (eg. 20 GB), a percentage of
# free space or the 'max' keyword.
d-i partman-auto-lvm/guided_size string max

# 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.

## Partitioning for EFI
# If your system needs an EFI partition you could add something like
# this to the recipe above, as the first element in the recipe:
#     538 538 1075 free \
#         $iflabel{ gpt } \
#         $reusethrough{ } \

```

```

#             method{ efi }           \
#             format{ }               \
#             .                       \
#
# The fragment above is for the amd64 architecture; the details may be
# different on other architectures. The 'partman-auto' package in the
# D-I source repository may have an example you can follow.

# 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

# Force UEFI booting ('BIOS compatibility' will be lost). Default: false.
#d-i partman-efi/non_efi_system boolean true
# Ensure the partition table is GPT - this is required for EFI
#d-i partman-partitioning/choose_label string gpt
#d-i partman-partitioning/default_label string gpt

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

```

B.4.7.2 RAID を用いたパーティション分割

ソフトウェア RAID アレイにパーティションをセットアップすることも、preseed を使用してできます。サポートしているのは、RAID 0, 1, 5, 6, 10、や縮退アレイの作成、スペアデバイスの指定です。

警告



自動パーティション分割でのこの形式は、誤動作をしやすいです。またこの機能は、debian-installer の開発者によって、相対的にまだあまりテストを受けていません。様々な条件で正しく動作するかの責任 (理解でき衝突しない限り) は、ユーザの側にあります。問題が発生したら、/var/log/syslog をチェックしてください。

```

# The method should be set to "raid".
#d-i partman-auto/method string raid
# Specify the disks to be partitioned. They will all get the same layout,
# so this will only work if the disks are the same size.
#d-i partman-auto/disk string /dev/sda /dev/sdb

# Next you need to specify the physical partitions that will be used.
#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 }                        \
#         .                                       \
#
# Last you need to specify how the previously defined partitions will be
# used in the RAID setup. Remember to use the correct partition numbers
# for logical partitions. RAID levels 0, 1, 5, 6 and 10 are supported;
# devices are separated using "#".

```

```
# Parameters are:
# <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 \
#     .

# For additional information see the file partman-auto-raid-recipe.txt
# included in the 'debian-installer' package or available from D-I source
# repository.

# This makes partman automatically partition without confirmation.
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 パーティションマウント法の制御

通常、ファイルシステムは、汎用一意識別子 (UUID) をキーとしてマウントされます。これにより、デバイス名が変更されたとしても、適切にマウントできます。UUID は長く読みにくいため、好みにより、インストーラは、伝統的なデバイス名やあなたが割り当てたラベルをベースにして、ファイルシステムをマウントできます。インストーラにラベルでマウントさせる場合、ラベルがないファイルシステムは、UUID でマウントされます。

LVM 論理ボリュームのような不変名のデバイスは、UUID ではなく伝統的な名前で使用され続けます。

警告



伝統的なデバイス名は、ブート時にカーネルが検出する順番によって、変わってしまう可能性があり、そのため、誤ったファイルシステムをマウントする原因になります。同様に、新しいディスクや USB ドライブを挿すと、ラベルが競合してしまいます。こうなってしまうと、起動時の挙動が不定になってしまいます。

```
# The default is to mount by UUID, but you can also choose "traditional" to
# use traditional device names, or "label" to try filesystem labels before
# falling back to UUIDs.
#d-i partman/mount_style select uuid
```

B.4.8 基本システムのインストール

インストールのこの段階で、実際に preseed できる項目は多くありません。質問はカーネルのインストールに関するものだけです。

```
# Configure APT to not install recommended packages by default. Use of this
# option can result in an incomplete system and should only be used by very
# experienced users.
```

```
#d-i base-installer/install-recommends boolean false

# The kernel image (meta) package to be installed; "none" can be used if no
# kernel is to be installed.
#d-i base-installer/kernel/image string linux-image-686
```

B.4.9 apt 設定

/etc/apt/sources.list のセットアップと基本設定オプションは、インストール方法と初期の質問への回答から、完全に自動的に行われます。さらに、他の (ローカルな) リポジトリを追加できます。

```
# You can choose to install non-free and contrib software.
#d-i apt-setup/non-free boolean true
#d-i apt-setup/contrib boolean true
# Uncomment this if you don't want to use a network mirror.
#d-i apt-setup/use_mirror boolean false
# Select which update services to use; define the mirrors to be used.
# Values shown below are the normal defaults.
#d-i apt-setup/services-select multiselect security, updates
#d-i apt-setup/security_host string security.debian.org

# Additional repositories, local[0-9] available
#d-i apt-setup/local0/repository string \
#     http://local.server/debian stable main
#d-i apt-setup/local0/comment string local server
# Enable deb-src lines
#d-i apt-setup/local0/source boolean true
# URL to the public key of the local repository; you must provide a key or
# apt will complain about the unauthenticated repository and so the
# sources.list line will be left commented out.
#d-i apt-setup/local0/key string http://local.server/key
# If the provided key file ends in ".asc" the key file needs to be an
# ASCII-armoured PGP key, if it ends in ".gpg" it needs to use the
# "GPG key public keyring" format, the "keybox database" format is
# currently not supported.

# By default the installer requires that repositories be authenticated
# using a known gpg key. This setting can be used to disable that
# authentication. Warning: Insecure, not recommended.
#d-i debian-installer/allow_unauthenticated boolean true

# Uncomment this to add multiarch configuration for i386
#d-i apt-setup/multiarch string i386
```

B.4.10 パッケージ選択

有効なタスクを組み合わせることでインストールするものを選ぶことができます。有効なタスクを以下に書き出します。

- standard (標準ツール)
- desktop (グラフィカルデスクトップ)
- gnome-desktop (Gnome デスクトップ)
- xfce-desktop (XFCE デスクトップ)
- kde-desktop (KDE Plasma デスクトップ)
- cinnamon-desktop (Cinnamon デスクトップ)

- mate-desktop (MATE デスクトップ)
- lxde-desktop (LXDE デスクトップ)
- web-server (web サーバ)
- ssh-server (SSH サーバ)

タスクをインストールしないこともできますし、他の方法でパッケージのセットが強制的にインストールされることもあります。standard タスクは常に含めるのをお勧めします。

タスクでインストールするパッケージに加えて、特定のパッケージをインストールする場合、pkgsel/include パラメータを使用できます。このパラメータの値は、カーネルコマンドラインと同様に簡単に仕様できるように、カンマか空白で区切ったパッケージのリストを取れます。

```
#tasksel tasksel/first multiselect standard, web-server, kde-desktop

# Individual additional packages to install
#d-i pkgsel/include string openssh-server build-essential
# Whether to upgrade packages after debootstrap.
# Allowed values: none, safe-upgrade, full-upgrade
#d-i pkgsel/upgrade select none

# Some versions of the installer can report back on what software you have
# installed, and what software you use. The default is not to report back,
# but sending reports helps the project determine what software is most
# popular and should be included on the first CD/DVD.
#popularity-contest popularity-contest/participate boolean false
```

B.4.11 インストールの仕上げ

```
# During installations from serial console, the regular virtual consoles
# (VT1-VT6) are normally disabled in /etc/inittab. Uncomment the next
# line to prevent this.
#d-i finish-install/keep-consoles boolean true

# Avoid that last message about the install being complete.
d-i finish-install/reboot_in_progress note

# This will prevent the installer from ejecting the CD during the reboot,
# which is useful in some situations.
#d-i cdrom-detect/eject boolean false

# This is how to make the installer shutdown when finished, but not
# reboot into the installed system.
#d-i debian-installer/exit/halt boolean true
# This will power off the machine instead of just halting it.
#d-i debian-installer/exit/poweroff boolean true
```

B.4.12 他パッケージの preseed

```
# Depending on what software you choose to install, or if things go wrong
# during the installation process, it's possible that other questions may
# be asked. You can preseed those too, of course. To get a list of every
# possible question that could be asked during an install, do an
# installation, and then run these commands:
# debconf-get-selections --installer > file
# debconf-get-selections >> file
```

B.5 高度なオプション

B.5.1 インストール中のカスタムコマンド実行

事前設定ツールには、インストール中の一定の箇所でコマンドやスクリプトを実行するといった、とても強力で柔軟なオプションが存在します。

ターゲットシステムのファイルシステムがマウントされると /target 以下で利用できるようになります。インストール CD を利用している場合はマウント後には /cdrom 以下で利用できるようになります。

```
# d-i preseeding is inherently not secure. Nothing in the installer checks
# for attempts at buffer overflows or other exploits of the values of a
# preconfiguration file like this one. Only use preconfiguration files from
# trusted locations! To drive that home, and because it's generally useful,
# here's a way to run any shell command you'd like inside the installer,
# automatically.

# This first command is run as early as possible, just after
# preseeding is read.
#d-i preseed/early_command string anna-install some-udeb
# This command is run immediately before the partitioner starts. It may be
# useful to apply dynamic partitioner preseeding that depends on the state
# of the disks (which may not be visible when preseed/early_command runs).
#d-i partman/early_command \
#     string debconf-set partman-auto/disk "$(list-devices disk | head -n1)"
# This command is run just before the install finishes, but when there is
# still a usable /target directory. You can chroot to /target and use it
# directly, or use the apt-install and in-target commands to easily install
# packages and run commands in the target system.
#d-i preseed/late_command string apt-install zsh; in-target chsh -s /bin/zsh
```

B.5.2 preseed を用いたデフォルト値変更

preseed を用いて、質問へのデフォルトの回答を変更できますが、この状態でも質問されるままにできます。そのためには、質問への回答を設定した後で、seen フラグを「false」に再設定してください。

```
d-i foo/bar string value
d-i foo/bar seen false
```

ブートプロンプトで preseed/interactive=true パラメータを設定し、すべての質問に対して同じ効果を及ぼすこともできます。これは事前設定ファイルのテストやデバッグにも便利です。

「d-i」という owner は、インストーラ自身が使用する変数でのみ、使用するべきであることに注意してください。ターゲットシステムにインストールされたパッケージに属する変数用には、パッケージ名を代わりに使用するべきです。項 B.2.2 の脚注をご覧ください。

ブートパラメータを利用して preseed を行う場合、質問に対して「?=」演算子を使用して回答できます。例: foo/bar?=value (もしくは owner:foo/bar?=value) これはもちろん、インストール中に実際に表示される質問に対応するパラメータにのみ効果を及ぼし、「内部」パラメータには効果を及ぼしません。

詳細なデバッグ情報を取得するためには、ブートパラメータ DEBCONF_DEBUG=5 を使用してください。これにより debconf が、各変数の現在の設定と各パッケージのインストールスクリプトの動作について、より詳細な情報を出力してくれます。

B.5.3 事前設定ファイルのチェーンロード

事前設定ファイルから他の事前設定ファイルを読み込みます。先に読み込まれたファイルの既存設定を、後から読み込まれた設定で上書きします。これは例えば、あるファイルに一般的なネットワークの設定を書きおき、他のファイルで具体的な設定を指定する、という使い方ができます。


```
# More than one file can be listed, separated by spaces; all will be
# loaded. The included files can have preseed/include directives of their
# own as well. Note that if the filenames are relative, they are taken from
# the same directory as the preconfiguration file that includes them.
#d-i preseed/include string x.cfg

# The installer can optionally verify checksums of preconfiguration files
# before using them. Currently only md5sums are supported, list the md5sums
# in the same order as the list of files to include.
#d-i preseed/include/checksum string 5da499872beccfed2c4872f9171c3d

# More flexibly, this runs a shell command and if it outputs the names of
# preconfiguration files, includes those files.
#d-i preseed/include_command \
#     string if [ "'hostname'" = bob ]; then echo bob.cfg; fi

# Most flexibly of all, this downloads a program and runs it. The program
# can use commands such as debconf-set to manipulate the debconf database.
# More than one script can be listed, separated by spaces.
# Note that if the filenames are relative, they are taken from the same
# directory as the preconfiguration file that runs them.
#d-i preseed/run string foo.sh
```

また `initrd` や `file preseed` の段階で、あらかじめ用意したファイルの `preseed/url` で設定した `network preseed` へ、チェーンロードを行うことができます。これにより、ネットワークに接続した時点で `network preseed` を行えます。この場合、2種類の異なる `preseed` が実行されることに注意してください。例えば、`preseed/early` コマンドを実行する機会が2度あり、2回目はネットワークに接続した時に発生するということです。

Appendix C

Debian でのパーティション分割

C.1 Debian のパーティションとそのサイズを決める

必要最小限の構成でも、GNU/Linux は自分のために少なくとも 1 つのパーティションを必要とします。オペレーティングシステム全体、アプリケーション、個人ファイルは 1 つのパーティションに収められます。多くの人はいくつかのパーティションも必要だと思っているようですが、これは厳密には正しくありません。「スワップ」とはオペレーティングシステムが用いるメモリの一時退避空間で、これを用いるとシステムはディスク装置を「仮想メモリ」として使えるようになります。スワップを独立したパーティションに割り当てると、Linux からの利用がずっと効率的になります。Linux は普通のファイルを無理やりスワップとして利用することもできますが、これはお勧めできません。

とはいえ大抵の人は、この最低限必要な数よりは多くのパーティションを GNU/Linux に割り当てます。ファイルシステムをいくつかのより小さなパーティションに分割する理由は 2 つあります。1 つめは安全性です。もし偶然に何かファイルシステムを破壊したとしても、一般的にその影響を被るのは 1 つのパーティションだけです。そのため、システムの一部を（注意深く保持しておいたバックアップと）置き換えるだけですみます。少なくとも、いわゆる「ルートパーティション」は別にすることを考慮しましょう。ここにはシステムの最も基本的な構成部分が収められており、もし他のパーティションに破損が生じたとしても、Linux を起動してシステムを補修できます。システムをゼロから再インストールしなければならないようなトラブルが防げるのです。

2 つめの理由は、一般的にビジネスで使う際により重要になってくるものですが、これはコンピュータの利用方法にかなり依存します。例えばスパムメールをたくさん受け取ったメールサーバは、パーティションを簡単に溢れさせてしまうかもしれません。もしメールサーバ上の独立したパーティションを `/var/mail` に割り当てれば、スパムメールを取り込んでもシステムの大半は問題なく動作するでしょう。

たくさんのパーティションを利用する際に唯一の不利になる点は、どのようなパーティションが必要となるかをあらかじめ予測するのが、ほとんどの場合は難しいということです。用意したパーティションが小さすぎると、システムを再インストールしたり、容量の足りないパーティションからしょっちゅうファイルを移動して、スペースを空けたりしなければならないでしょう。一方、あまりに大きなパーティションを用意すれば、他で利用できるスペースを浪費しかねません。近頃はディスクも安価になったとはいえ、お金を無駄に使う必要はないでしょう？

C.2 ディレクトリツリー

ディレクトリとファイルの名前について、Debian GNU/Linux は [Filesystem Hierarchy Standard](#) に従っています。この規格を用いると、ユーザやプログラムは、ファイルやディレクトリの場所を予想しやすくなります。根っこ（ルート = `root`）にあるディレクトリは、単にスラッシュ `/` で表されます。ルートのレベルには、Debian システムでは必ず以下のようなディレクトリが含まれます。

ディレクトリ	内容
<code>bin</code>	基本的なコマンドバイナリ
<code>boot</code>	ブートローダの静的ファイル
<code>dev</code>	デバイスファイル
<code>etc</code>	ホスト固有のシステム設定
<code>home</code>	ユーザのホームディレクトリ

ディレクトリ	内容
lib	基本的な共有ライブラリとカーネルモジュール
media	取替え可能なメディア用のマウントポイントを含む
mnt	ファイルシステムを一時的にマウントするためのポイント
proc	システム情報を含む仮想ディレクトリ
root	root ユーザのホームディレクトリ
run	ランタイム可変データ
sbin	基本的なシステムバイナリ
sys	システム情報を含む仮想ディレクトリ
tmp	一時ファイル用
usr	第 2 階層
var	可変データ
srv	システムによって割り当てられた、サービスのためのデータ
opt	アドオンアプリケーションソフトウェアパッケージ

以下の一覧は、ディレクトリやパーティションについて重要となる考え方を説明したものです。与えられたシステム構成や特別な使用パターンによって、ディスク使用状況は大きく変化することに注意して下さい。ここで提案するのは一般的なガイドラインであり、パーティション分割の第一歩を提供しています。

- ルートパーティション / は、必ず /etc、/bin、/sbin、/lib、/dev を物理的に含んでいなければなりません (つまりこれらのディレクトリを別のパーティションにすることはできません)。さもないと起動ができなくなります。一般的にここは 250–350MB 程度を必要とします。
- /usr: すべてのユーザプログラムを含む (/usr/bin)、ライブラリ (/usr/lib)、文書 (/usr/share/doc) など。ここは一般に、ファイルシステムの中でも最も容量を必要とするところです。少なくとも 500MB のディスク容量を割り当てておくべきでしょう。インストールしようとするパッケージの数やタイプによっては、もっと多くのディスク容量を割り当てなければなりません。ディスク容量がたっぷりあるワークステーションやサーバのインストールでは 4–6GB を割り当てておくべきです。
- 現在は /usr を root パーティション / に置くことが勧められます。そうしない場合はブート時に問題を引き起こす可能性があります。つまり、/usr を含めたルートパーティションには最低でも 600–750MB、ワークステーションやサーバの場合には 5–6GB のディスク容量を割り当てておくべきだということです。
- /var: ニュース記事、電子メール、ウェブコンテンツ、データベース、パッケージングシステムのキャッシュなど、様々な可変データがこのディレクトリに収められます。このディレクトリの容量はシステムの利用方法に大きく左右されますが、たいていの場合はパッケージ管理ツールの使う分が最も大きな影響を持つことになるでしょう。Debian が提供するものすべてをいっぺんにフルインストールする場合でも、/var には 2–3GB ほどの容量を割り当てておけば足りるはずです。一度にすべてをインストールせず、部分部分を徐々に (例えば、まずサービスやユーティリティを、次にコンソール用のもの、次に X 用のもの…というように) インストールするならば、300–500MB の空き容量があれば良いでしょう。ハードディスクの空き容量が貴重で、メジャーアップデートをする予定がないならば、30–40MB ほどでもなんとかかやっていると良いでしょう。
- /tmp: プログラムが作成する一時データは、普通このディレクトリを利用します。通常は 40–100MB あれば充分です。いくつかのアプリケーション (アーカイブマニピュレータ、CD/DVD オーサリングツール、およびマルチメディアソフトウェアを含む) が、一時イメージファイルを保存するのに /tmp を使用するかもしれません。そのようなアプリケーションを使用する計画があるのなら、それ相応に /tmp で利用できる容量を調整すべきです。
- /home: 各ユーザは、個人的なデータをこのディレクトリのサブディレクトリに収めます。その容量は、このシステムを利用するユーザの数や、ユーザディレクトリにどのようなファイルが収められるかによって異なってきます。システムの使い方にもよりますが、ユーザごとに約 100MB ほどが必要でしょう。しかしこの値は必要に応じて調整しなければなりません。もし、たくさん

のマルチメディアファイル (写真、MP3、動画) をホームディレクトリに保存するつもりなら、もっと多くの容量を確保しておいてください。

C.3 お勧めするパーティションルール

新規ユーザや Debian マシンを個人で使う人、家庭で使うシステム、その他ユーザ 1 人で使うようなマシンには、/パーティション 1 つ (とスワップ) で済ませるのが、恐らくもっとも簡単で素直なやり方でしょう。お勧めのパーティションタイプは ext4 です。

マルチユーザシステムやたくさんのディスク容量があるシステムでは、/var、/tmp、/home をそれぞれ /パーティションとは別の独立したパーティションにするのが良いでしょう。

Debian のディストリビューションには含まれていないプログラムをたくさんインストールするつもりなら、/usr/local パーティションが必要となるかもしれません。またメールサーバとして利用するなら、/var/mail を別のパーティションにする必要があるかもしれません。たくさんのユーザアカウントを抱えるサーバを設置するなら、独立した大きな /home パーティションを用意することも大抵は良い考えです。このように、利用方法に応じて、パーティションの配置状態はコンピュータによって様々です。

とても複雑なシステムのためには、[Multi Disk HOWTO](#) をご覧になるとよいでしょう。こちらには、ISP やサーバの管理者が関心を持つような事柄の多くが、詳細な情報として含まれています。

スワップスペースの問題に関しては、様々な見方があります。大雑把ながらも悪くないやり方は、搭載しているシステムメモリと同じ容量のスワップを用意することです。ただし多くの場合は 512MB 以下にすべきではありません。もちろんこのルールにも例外はあります。

一例として、以前の自宅用マシンを紹介しましょう。このマシンは 512MB の RAM と /dev/sda に 20GB SATA のハードディスクを搭載していました。/dev/sda1 には別の OS 用に 8GB のパーティションがあり、/dev/sda3 を 512MB のスワップパーティションとして使用し、残りの約 11.4GB の /dev/sda2 を Linux パーティションにしていました。

システムのインストールが完了した後に入れることになるであろう各タスク (task) の占める領域については[項D.2](#)を調べてください。

C.4 Linux におけるデバイス名

Linux におけるディスクおよびパーティションの命名法は、他のオペレーティングシステムとは異なっています。パーティションを作成したりマウントしたりする際には、Linux がどのようなディスク名を用いるのか知っておく必要があります。以下は基本的な命名法の仕組みです。

- 最初に見つかったハードディスクは /dev/sda と名付けられる。
- 2 番目に見つかったハードディスクは /dev/sdb と名付けられ、以下も同様。
- 第 1 SCSI CD-ROM は /dev/scd0 および /dev/sr0 と名付けられる。

各ディスクのパーティションは、ディスク名に十進数を付け加えることで表します。例えば sda1 と sda2 は、それぞれシステムの第 1 SCSI ディスクドライブの第 1、第 2 パーティションを表します。

実際にありそうな例を挙げてみましょう。2 つの SCSI ディスクを持つシステムで、一方の SCSI アドレスが 2、もう一方の SCSI アドレスが 4 だとします。最初のディスク (アドレス 2) は sda、2 つ目のディスクは sdb と名付けられます。もし sda ドライブに 3 つのパーティションがあるなら、それらは sda1、sda2、sda3 と名付けられます。sdb ディスクとそのパーティションについても同様です。

2 つの SCSI ホストバスアダプタ (コントローラ) があると、ドライブの順序が混乱するかもしれないので注意してください。ドライブのモデルや容量を知っているなら、ブートメッセージに注目するのが最も良い解決策でしょう。

C.5 Debian のパーティション分割プログラム

いろいろな種類のパーティション分割ツールが Debian 開発者によって組み込まれ、様々な形式のハードディスクやコンピュータアーキテクチャで動作するようになっています。以下に、それらのアーキテクチャで使えるプログラムのリストを示します。

partman Debian 推奨のパーティション分割ツールです。このアーミーナイフは、パーティションサイズを変更したり、ファイルシステムを作成したり、マウントポイントを指定したりすることもできます。

fdisk 上級魔術師用の、Linux オリジナルのディスクパーティション作成プログラムです。

すでにコンピュータに FreeBSD のパーティションが存在する場合は注意が必要です。インストール用のカーネルはこのパーティションをサポートしていますが、fdisk の表示方法では名前が異なります (そもそも表示されないかもしれません)。Linux+FreeBSD HOWTO をご覧になってください。

cfdisk 一般ユーザのための、操作の容易なフルスクリーン表示ディスクパーティション作成プログラムです。

cfdisk は FreeBSD パーティションを全く理解しません。したがって、こちらでもデバイス名が変わってしまうかもしれません。

ディスクのパーティショニング (あるいは同様のもの) を選択すると、上記のプログラムの中のひとつがデフォルトで実行されます。VT2 のコマンドラインから、異なるパーティション分割ツールを使うこともできますがお勧めしません。

Appendix D

雑多な事柄

D.1 Linux のデバイス

Linux では、/dev に特別なファイルがいろいろとあります。このファイルはデバイスファイルと呼ばれ、通常のファイルと異なる振る舞いをします。デバイスファイルの一般的なものは、ブロックデバイスとキャラクタデバイスです。このファイルは、ハードウェアにアクセスする実際のドライバ (Linux カーネルの一部) へのインターフェースです。その他、あまり一般的ではありませんが、パイプというデバイスファイルの形式もあります。以下に、最も重要なデバイスファイルを一覧します。

sda	第 1 ハードディスク
sdb	第 2 ハードディスク
sda1	最初のハードディスクの最初のパーティション
sdb7	2 番目のハードディスクの 7 番目のパーティション

sr0	第 1 CD-ROM
sr1	第 1 CD-ROM

ttyS0	シリアルポート 0、MS-DOS では COM1
ttyS1	シリアルポート 1、MS-DOS では COM2
psaux	PS/2 マウスデバイス
gpmdata	疑似デバイス、GPM (マウス) デーモンからのリピータデータ

cdrom	CD-ROM ドライブへのシンボリックリンク
mouse	マウスデバイスファイルへのシンボリックリンク

null	書き込まれたものをすべて消してしまうデバイス
zero	無限に 0 を読み出せるデバイス

D.1.1 マウスのセットアップ

(gpm が動いている) Linux コンソールと X ウィンドウ環境の両方で、マウスを使用できます。通常、gpm や X サーバ自体をインストールするだけです。どちらも、マウスデバイスとして /dev/input/mice を使用するように設定されています。正しいマウスプロトコルは、gpm では exps2、X では ExplorerPS/2 とされています。それぞれの設定ファイルは /etc/gpm.conf と /etc/X11/xorg.conf です。

あなたのマウスが動作するには、特定のカーネルモジュールを読み込まなければなりません。ほと

んどの場合、正しいモジュールを自動検出しますが、旧式のシリアルマウスや、バスマウスでは¹いつもそうとは限りません。とはいえ、非常に古いコンピュータでなければ、かなり「まれ」です。以下に、マウスのタイプごとの、Linux カーネルモジュールをまとめます。

モジュール	説明
psmouse	PS/2 マウス (自動検出可)
usbhid	USB マウス (自動検出可)
sermouse	ほとんどのシリアルマウス
logibm	Logitech アダプタカードに接続するバスマウス
inport	ATI や Microsoft の InPort カードに接続するバスマウス

マウスドライバモジュールを読み込むのに、modconf コマンド (同名のパッケージに収録) を使用でき、kernel/drivers/input/mouse カテゴリから探せます。

D.2 タスクに必要なディスクの空き容量

amd64 アーキテクチャの全標準パッケージを含む標準インストールで、デフォルトのカーネルを用いると、971MB 以上のディスク領域を必要とします。「標準システム」タスクを選択しない最小の基本インストールでは、769MB 必要でしょう。

重要項目



どちらの場合も、インストールが完了し一時ファイルを削除した後の、実際のディスク領域です。ジャーナルファイルのような、ファイルシステムのオーバーヘッドで消費される量は含まれていません。これは、インストールの最中や通常のシステム利用では、もっと大量にディスク領域が必要だということです。

以下の表は tasksel で表示されるタスクについて、aptitude が報告したサイズです。いくつかのタスクでは、内容が一部重複していることに注意してください。つまり、2つのタスクを一緒にインストールした後のインストールサイズは、挙げてある数値を合計したものよりも、小さくなるということです。

デフォルトでは、インストーラは GNOME デスクトップ環境をインストールしますが、特殊なインストールイメージを使用したり、インストール中に希望のデスクトップ環境を指定して、その他のデスクトップ環境を選択できます (項6.3.6.2 参照)。

パーティションのサイズを決定するとき、標準インストールのサイズに加え、以下の表に列挙したサイズが必要であるのに注意してください。「Installed size」はインストール完了時に /usr と /lib に必要なサイズを、「Download size」は /var に (一時的に) 必要なサイズを記述しています。

タスク	インストールサイズ (MB)	ダウンロードサイズ (MB)	インストールに必要な空き容量 (MB)
デスクトップ環境			
• GNOME (デフォルト)	2790	786	3576
• KDE Plasma	4122	1212	5334
• Xfce	2187	621	2808
• LXDE	2271	653	2924
• MATE	2574	711	3285
• Cinnamon	4197	1251	5448
ウェブサーバ	44	11	55
SSH サーバ	2	0	2

¹シリアルマウスには、通常 9 穴の D 型コネクタが、バスマウスには、8 ピン円形コネクタが付いており、PS/2 マウスの 6 ピン円形コネクタや、ADB マウスの 4 ピン円形コネクタと混同することはないでしょう。

英語以外の言語でインストールする場合、その言語が有効なら `tasksel` は 地域化タスクを、自動的にインストールします。必要な容量は言語によって異なりますが、ダウンロードとインストールで最大 350MB 必要となります。

D.3 Unix/Linux システムからの Debian GNU/Linux のインストール

この節は、マニュアルの他の部分で説明されているメニュードリブンインストーラを使用せずに、既存の Unix・Linux システムから Debian GNU/Linux をインストールする方法について説明します。この「クロスインストール」HOWTO は、Red Hat, Mandriva, SUSE から Debian GNU/Linux に移行するユーザの要望で書かれました。本節では、*nix コマンドの入力について熟知し、ファイルシステムを操作できるのが前提となっています。本節では、# が Debian chroot に入力されたコマンドを示し、\$ はユーザの現在のシステムに入力されるコマンドを表します。

一旦、新しい Debian システムを好みに設定したら、既存のユーザデータを (あるなら) 稼働したまま移行できます。したがって、これは「ダウンタイム無し」での Debian GNU/Linux インストールになります。またこれは、様々な起動・インストールメディアと相性のよくないハードウェアに対処するうまい方法です。

注意



これはほとんど手作業になりますから、自分でシステムの大部分の基本設定を行う必要があります。それには通常のインストールよりも Debian や Linux の一般的な知識が必要なことを覚えておいてください。また、この手順で通常のインストールと全く同じシステムになると期待してはいけません。これはシステムをセットアップする基本的な手順でしかありません。追加インストールや追加設定が必要になるかもしれません。

D.3.1 はじめに

今の *nix のパーティション分割ツールで、スワップと最低 1 つファイルシステムを作成するよう、ハードディスクを希望に添って再分割してください。コンソールだけのインストールには、最低 769MB の空き領域が必要ですし、X をインストールする予定なら 2271MB (GNOME や KDE Plasma のようなデスクトップ環境をインストールする場合はもっと) 必要です。

次に、パーティションにファイルシステムを作成してください。例えば、`/dev/sda6` パーティションに、`ext3` ファイルシステムを作成するには、以下のようにします。(今回の例ではこのパーティションを `root` パーティションとします)

```
# mke2fs -j /dev/sda6
```

`ext3` ではなく `ext2` ファイルシステムを作成するには、`-j` を取ってください。

スワップを以下のように初期化して有効にしてください。(パーティション番号は、Debian スワップパーティションにするパーティション番号に、読み替えてください)

```
# mkswap /dev/sda5
# sync
# swapon /dev/sda5
```

パーティションを `/mnt/debinst` (インストールポイント。新システムの `root (/)` ファイルシステムになります) にマウントしてください。厳密にいうとマウントポイント名は何でも構いません。以降の説明ではこれを使用します。

```
# mkdir /mnt/debinst
# mount /dev/sda6 /mnt/debinst
```


注意

分割したパーティションをファイルシステムの一部 (例 /usr) にマウントする場合、次のステージに進む前に、手動でそのディレクトリを作成・マウントする必要があります。

D.3.2 debootstrap のインストール

Debian インストーラが使用するユーティリティで、Debian 基本システムをインストールする公式の方法と認められているのが debootstrap です。wget と ar を使用しますが、/bin/sh と基本的な Unix/Linux ツール²にのみ依存しています。今のシステムにまだインストールしていなければ、wget と ar をインストールし、その後 debootstrap をダウンロード・インストールしてください。

また、手動でインストールするには、以下の手順になります。まず .deb を展開するために作業フォルダを次のように作ってください。

```
# mkdir work
# cd work
```

debootstrap バイナリは、Debian アーカイブ (あなたのアーキテクチャに適合するファイルを必ず選ぶこと) にあります。pool から debootstrap .deb をダウンロードして、作業フォルダにパッケージをコピーし、ファイルを展開してください。ファイルをインストールする際には root 権限を持つ必要があるでしょう。

```
# ar -x debootstrap_0.X.X_all.deb
# cd /
# zcat /full-path-to-work/work/data.tar.gz | tar xv
```

D.3.3 debootstrap の実行

debootstrap を実行すると、アーカイブから必要なファイルを直接ダウンロードできます。以下のコマンドの例では、http.us.debian.org/debian としていますが、ネットワーク的に近い Debian アーカイブミラーサイトで代用できます。ミラーサイトは、<http://www.debian.org/mirror/list> に一覧があります。

bullseye Debian GNU/Linux インストールイメージを持っていて、/cdrom にマウントしていれば、http URL に代えて file URL (file:/cdrom/debian/) を使用することができます。

debootstrap コマンドの ARCH は、amd64, arm64, armel, armhf, i386, mips64el, mipsel, ppc64el, s390x のどれかと置き換えるか、

```
# /usr/sbin/debootstrap --arch ARCH bullseye \
  /mnt/debinst http://ftp.us.debian.org/debian
```

のようにします。所用のアーキテクチャがホストとは異なる場合には --foreign オプションを追加します。

D.3.4 基本システムの設定

さあ、これでディスクに真の Debian システムを (いくぶん中がスカスカですが) 手に入れました。そこに chroot してください。

```
# LANG=C.UTF-8 chroot /mnt/debinst /bin/bash
```

所用のアーキテクチャがホストとは異なる場合にはまず、qemu-user-static を新しいホストにコピーする必要があるかもしれません。

```
# cp /usr/bin/qemu-ARCH-static /mnt/debinst/usr/bin
# LANG=C.UTF-8 chroot /mnt/debinst qemu-ARCH-static /bin/bash
```

²これには、sed, grep, tar, gzip といった、GNU コアユーティリティが含まれます。

chroot した後で Debian 基本システムと互換のある端末定義を設定する必要があるかもしれません。例えば

```
# export TERM=xterm-color
```

のようにします。TERM の値により、その値をサポートするのに ncurses-term パッケージをインストールする必要があるかもしれません。

所用のアーキテクチャがホストとは異なる場合には複数段階の前処理を終えておく必要があるかもしれません。

```
/debootstrap/debootstrap --second-stage
```

D.3.4.1 デバイスファイルの作成

この時点で、/dev/ には、非常に基本的なデバイスファイルがあるだけです。おそらくインストールの次のステップで、追加デバイスファイルが必要になります。インストールに使用するホストシステムがモジュール化カーネルを使用するかどうかや、新しいシステムで動的デバイスファイル (例: udev を使用) と静的デバイスファイルのどちらを使用するかにより、どの方法で行うかが異なります。

以下のような選択肢があります。

- makedev パッケージをインストールし、次のようにして (chroot してから)、デフォルトの静的デバイスファイル群を作成してください。

```
# apt install makedev
# mount none /proc -t proc
# cd /dev
# MAKEDEV generic
```

- MAKEDEV を使用して、指定したデバイスファイルのみを手で作成します。
- ホストシステムの /dev をターゲットシステムの /dev の先頭にマウントします。いくつかのパッケージの postinst スクリプトでは、デバイスファイルを作成しようとするのに注意してください。そのため、この選択肢は注意深く使用するべきです。

D.3.4.2 パーティションのマウント

/etc/fstab を作る必要があります。

```
# editor /etc/fstab
```

以下のサンプルを自分に合うように編集できます。

```
# /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/cdrom         /media/cdrom     iso9660 noauto,ro,user,exec       0    0
/dev/XXX           /tmp             ext3    rw,nosuid,nodev           0    2
/dev/XXX           /var             ext3    rw,nosuid,nodev           0    2
/dev/XXX           /usr             ext3    rw,nodev                   0    2
/dev/XXX           /home           ext3    rw,nosuid,nodev           0    2
```

/etc/fstab で指定したファイルシステムを、すべてマウントするには mount -a としてください。また、ファイルシステムを別々にマウントするには、以下のようになしてください。

```
# mount /path # e.g.: mount /usr
```

現在 Debian システムでは、リムーバブルメディアのマウントポイントを /media にしていますが、/ にシンボリックリンクを置き互換性を保っています。以下の例のように、必要であれば作成してください。

```
# cd /media
# mkdir cdrom0
# ln -s cdrom0 cdrom
# cd /
# ln -s media/cdrom
```

proc ファイルシステムは、どこでも何度でもマウントすることができますが、慣習的に /proc にマウントします。mount -a を使用しなかった場合は、以下のように先に進む前に必ず proc をマウントしてください。

```
# mount -t proc proc /proc
```

ls /proc コマンドは、今度は空のディレクトリにはならないはずですが。これが失敗するようなら、以下のように chroot の外側から proc をマウントできるかもしれません。

```
# mount -t proc proc /mnt/debinst/proc
```

D.3.4.3 タイムゾーンの設定

/etc/adjtime ファイルの 3 行目に、「UTC」か「LOCAL」を設定し、システムがハードウェアの時計を UTC として解釈するか、それぞれの現地時間として解釈するかを決定します。以下のコマンドで、上記の選択とタイムゾーンの選択を行えます。

```
# editor /etc/adjtime
```

以下に例を示します。

```
0.0 0 0.0
0
UTC
```

以下のコマンドでタイムゾーンの選択ができます。

```
# dpkg-reconfigure tzdata
```

D.3.4.4 ネットワークの設定

ネットワークの設定をするには、/etc/network/interfaces, /etc/resolv.conf, /etc/hostname and /etc/hosts を編集してください。

```
# editor /etc/network/interfaces
```

次は、/usr/share/doc/ifupdown/examples のシンプルな例です。

```
#####
# /etc/network/interfaces -- ifup(8), ifdown(8) 用設定ファイル
# どのようなオプションが使えるかについては interfaces(5) man ページを参照してください
#####
# loopback インターフェイスは絶対に必要では無くなっていますが、必要になった時に使えます
#
auto lo
iface lo inet loopback

# dhcp を使う:
#
# auto eth0
# iface eth0 inet dhcp

# 静的 IP 設定の例: (network, broadcast, gateway はオプション)
#
```

```
# 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 に、ネームサーバと search ディレクティブを入力してください。

```
# editor /etc/resolv.conf
```

以下は、/etc/resolv.conf の簡単な例です。

```
search example.com
nameserver 10.1.1.36
nameserver 192.168.9.100
```

システムのホスト名 (2 から 63 文字) を入力してください。

```
# echo DebianHostName > /etc/hostname
```

また、IPv6 をサポートした基本的な /etc/hosts は以下のようになります。

```
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
```

複数のネットワークカードを持っているなら、/etc/modules ファイルに希望の順番で、ドライバモジュールの名前を配置してください。その後起動中に、各カードは期待通りにインターフェース名 (eth0, eth1 など) と結びつけられます。

D.3.4.5 apt の設定

debootstrap は、追加パッケージをインストールする、非常に基本的な /etc/apt/sources.list を作成します。しかし、他のパッケージ取得先を追加したくなると思います。以下の例はソースパッケージとセキュリティ更新を追加しています。

```
deb-src http://ftp.us.debian.org/debian bullseye main

deb http://security.debian.org/ bullseye-security main
deb-src http://security.debian.org/ bullseye-security main
```

sources list を更新したら、apt update を必ず実行してください。

D.3.4.6 ロケールとキーボードの設定

英語以外の言語を使用するようロケールの設定をするために、ロケールをサポートするパッケージ (locales) をインストール・設定してください。現在は UTF-8 ロケールを使用するのをお勧めします。

```
# apt install locales
# dpkg-reconfigure locales
```

(必要なら) 以下のようにキーボードの設定を行ってください。

```
# apt install console-setup
# dpkg-reconfigure keyboard-configuration
```

chroot 内では、キーボードを設定できませんが、再起動後に有効になることに注意してください。

D.3.5 カーネルのインストール

このシステムを起動できるようにするのなら、おそらく Linux カーネルとブートローダが必要でしょう。以下のようにして、パッケージ化済みカーネルを確認してください。

```
# apt search linux-image
```

その後、パッケージ名を指定して、選択したカーネルパッケージをインストールしてください。

```
# apt install linux-image-arch-etc
```

D.3.6 ブートローダのセットアップ

Debian GNU/Linux システムを起動できるようにするために、インストールしたカーネルを新しい root パーティションから読み込むように、ブートローダをセットアップしてください。debootstrap は、ブートローダをインストールしないことに注意してください。とは言っても、セットアップするのに Debian chroot 内部の apt を使用できます。

/dev/sda デバイスファイルは、作成済みだと仮定していることに注意してください。grub2 のインストールには別の方法もありますが、それはこの付録では扱いません。

D.3.7 リモートアクセス: SSH のインストールとアクセス方法の設定

コンソール経由でシステムにログインできる場合はこの節を飛ばせます。後でネットワーク経由でシステムにアクセスできるようにする必要がある場合は SSH をインストールしてアクセス方法を用意する必要があります。

```
# apt install ssh
```

パスワードによる root のログインはデフォルトで無効になっているため、パスワードを設定してパスワードによる root のログインを有効にしてアクセス方法を用意します:

```
# passwd  
# editor /etc/ssh/sshd_config
```

有効にするオプション:

```
PermitRootLogin yes
```

root アカウントに SSH の鍵を追加してアクセス方法を用意することもできます:

```
# mkdir /root/.ssh  
# cat << EOF > /root/.ssh/authorized_keys  
ssh-rsa ....  
EOF
```

最後に、root ユーザを追加してパスワードを設定してアクセス方法を用意することもできます:

```
# adduser joe  
# passwd joe
```

D.3.8 仕上げに

すでに述べたように、インストールしたシステムは非常に基本的な物になります。もっと成熟したシステムにしたければ、優先度が「standard」のパッケージを、すべてインストールする簡単な方法があります。以下のようにしてください。

```
# tasksel install standard
```

もちろん apt で、個々のパッケージをインストールすることもできます。

インストールが終わると、ダウンロードしたパッケージが /var/cache/apt/archives/ に大量に残っています。以下のようにして、ディスク領域を解放できます。

```
# apt clean
```

D.4 PPP over Ethernet (PPPoE) を用いた Debian GNU/Linux のインストール

いくつかの国でインターネットサービスプロバイダに接続するのに、ブロードバンド接続 (ADSL やケーブル TV) の一般的なプロトコルは、PPP over Ethernet (PPPoE) です。インストーラでは、PPPoE を用いたネットワーク接続のセットアップをサポートしていませんが、非常に簡単に設定できます。この節ではその方法を説明します。

また、インストール中に PPPoE 接続をセットアップすると、インストールしたシステムを再起動した後でも有効になります (第7章参照)。

インストール中に PPPoE をセットアップし使用するには、CD-ROM/DVD イメージを使用する必要があります。その他のインストール方法 (例: netboot) では、サポートしていません。

PPPoE でのインストールは、他のインストール方法とほとんど同じです。以下で説明するステップが異なるだけです。

- ・ ブートパラメータに `modules=ppp-udeb` を指定してインストーラを起動してください。これにより、PPPoE のセットアップに使用するコンポーネント (`ppp-udeb`) を確実に読み込み、自動的に起動します。
- ・ 通常のインストール初期化手順 (言語、国、キーボードの選択、追加インストーラコンポーネント³の読み込み) を行います。
- ・ 次のステップでは、システムにあるイーサネットカードを特定するのに、ネットワークハードウェアを検出します。
- ・ この後、実際の PPPoE のセットアップが始まります。インストーラは、PPPoE コンセントレータ (PPPoE 接続を扱う一種のサーバ) を見つけるのに、検出したすべてのイーサネットインターフェースを調べます。

最初の試行では、コンセントレータが見つからない可能性があります。これはネットワークが遅い・負荷が高い場合や、サーバ側のエラーで起こる可能性があります。ほとんどの場合、2 回目の試行でコンセントレータの検出に成功します。再試行するには、インストーラのメニューにある `Configure and start a PPPoE connection` を選択してください。

- ・ コンセントレータを検出した後、ログイン情報 (PPPoE のユーザ名とパスワード) を入力してください。
- ・ インストーラは、先ほど入力した情報を用いて PPPoE 接続を確立します。正しい情報を入力していれば PPPoE 接続の設定を行い、インストーラはその接続を用いてインターネットに接続し、(必要なら) パッケージを取得できます。ログイン情報が正しくない場合や、何かエラーが発生した場合、インストーラは停止しますが、メニューの `Configure and start a PPPoE connection` を選択して、設定を再度行えます。

³ppp-udeb コンポーネントは、このステップの追加コンポーネントの 1 つとして読み込まれます。優先度を「中」「低」でインストールする場合 (エキスパートモード)、ブートプロンプトの「`modules`」パラメータに入力する代わりに、ppp-udeb を選択することもできます。

Appendix E

付記

E.1 この文書について

本マニュアルは、初期の Debian インストールマニュアルを元にした、boot-floppies 用の woody インストールマニュアルを元に、sarge 用 debian-installer のために書かれました。また、2003 年 GPL でリリースした、Progeny ディストリビューションマニュアルも元にしてあります。

この文書は DocBook XML を用いて書かれています。出力形式は、docbook-xml パッケージや docbook-xsl パッケージの情報を用いて、様々なプログラムで生成されます。

この文書では、そのメンテナンス性を高めるために、実体やプロファイル属性など数々の XML の特徴を利用しています。これらは、プログラミング言語の変数や条件に似た機能を果たします。この XML ソースには、異なる各アーキテクチャの情報が含まれていますが、各アーキテクチャ固有の文章のまとまりを分離するのに、プロファイル属性が使われています。

日本語訳は、鴨志田睦 (1997 年)、岡充 (1998-1999 年)、遠藤美純 (1998-2000 年)、門脇正史、鍋谷栄展、八田真行、Guangcheng Wen (1999 年)、今井伸広、上川純一、喜瀬浩、久保田智広、齋藤努、中野武雄 (2002 年)、杉山友章 (2002-2007 年)、武井伸光 (2002-2010 年)、倉澤望 (2002-2013 年)、victory (2013-2017 年)、やまねひでき (2011-2019 年)、吉野与志仁、杉本典充 (2021 年) が行いました。

E.2 この文書への貢献

この文書に関する問題や提案がある場合には、それらを installation-guide パッケージに対するバグ報告として提出してください。その方法については reportbug パッケージや [Debian バグ追跡システム](#) のオンラインドキュメントをご覧ください。なお同じ問題が報告済みかどうかを調べるためには、[installation-guide パッケージに関するバグ報告](#) を確認するとよいでしょう。もし同じ問題が報告済みならば、XXXX@bugs.debian.org 宛に、確証のための追加情報や有益な情報を提供することができます。XXXX には、報告済みのバグに付けられた番号を当てはめてください。

もちろんこの文書の DocBook ソースを入手し、それに対するパッチを作成していただけたらととても助かります。DocBook ソースは [salsa 上の installation-guide project](#) にあります。DocBook に慣れていなくても心配しないでください。あなたが始められるよう、マニュアルディレクトリに簡単なチートシートがあります。html に似ていますが、表示方法ではなく、テキストの意味の方を重視しています。パッチは debian-boot メーリングリスト (以下を参照) に提出してください。git でソースを取り出す方法については、ソースのルートディレクトリの [README](#) をご覧ください。

どうか、この文書の著者に直接連絡をとるようなことはしないでください。このマニュアルの話題も含め、debian-installer に関する議論を行うメーリングリストがあります。その宛先は debian-boot@lists.debian.org です。また [Debian メーリングリスト購読](#) ページには、このメーリングリストの購読に関する説明があります。また [Debian メーリングリストアーカイブ](#) では、その写しをオンラインで読むこともできます。

E.3 多大な貢献

もともとこの文書は Bruce Perens, Sven Rudolph, Igor Grobman, James Treacy, Adam Di Carlo が書きました。Sebastian Ley がインストール Howto を書きました。

Miroslav Kuře には、Sarge の debian-installer の新機能について、たくさん記述していただきました。Frans Pop は、Etch, Lenny, Squeeze の主任編集者で、かつリリースマネージャでした。

非常に多くの Debian ユーザや開発者がこの文書に貢献しています。特に、さまざまな文書を編集、著述している Michael Schmitz (m68k のサポート), Frank Neumann ([Amiga install manual](#) の原著者), Arto Astala, Eric Delaunay/Ben Collins (SPARC に関する情報), Tapio Lehtonen, Stéphane Bortzmeyer には多大なご協力をいただきました。また、Pascal Le Bail には USB メモリから起動する方法について、有益な情報をいただいたことに感謝いたします。

Jim Mintha によるネットワークブートに関する HOWTO (利用可能な URL が不明) や、[Debian FAQ](#)、[Linux/m68k FAQ](#)、[SPARC プロセッサ向け Linux FAQ](#)、[Linux/Alpha FAQ](#) やその他の文書には、極めて有用な文章や情報があります。これらの自由に利用できる素晴らしい情報源をメンテナンスされている方々は、高く評価されるべきでしょう。

本マニュアルの chroot してのインストールに関する節 (項D.3) は、Karsten M. 自身が著作権を持つドキュメントの一部が元になっています。

E.4 商標表示

すべての商標には、それぞれに所有者がいます。

Appendix F

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

F.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

F.2 GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The 「Program」, below, refers to any such program or work, and a 「work based on the Program」 means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 「modification」.) Each licensee is addressed as 「you」.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

- c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may

differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and [any later version], you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM [AS IS] WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL AND COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

F.3 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the [copyright] line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with absolutely no warranty; for details
type show w. This is free software, and you are welcome
to redistribute it under certain conditions; type show c
```

for details.

The hypothetical commands `「show w」` and `「show c」` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `「show w」` and `「show c」`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a `「copyright disclaimer」` for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the  
program Gnomovision (which makes passes at compilers) written  
by James Hacker.
```

```
signature of Ty Coon, 1 April 1989  
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.