Debian GNU/Linux インストールガイド

October 10, 2021
## Contents

1 ようこそDebianへ
1.1 Debianとは？.............................................................. 1
1.2 GNU/Linuxとは？.......................................................... 2
1.3 Debian GNU/Linuxとは？................................................... 2
1.4 Debian GNU/kFreeBSDとは？............................................ 3
1.5 Debian GNU/Hurd とは？.................................................. 3
1.6 Debian Installer とは？................................................... 3
1.7 Debianの入手.............................................................. 4
1.8 このドキュメントの最新版の入手....................................... 4
1.9 この文書の構成........................................................... 4
1.10 文書への支援のお願い................................................... 5
1.11 著作権およびソフトウェアライセンスについて.............................. 5

2 必要なシステム
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/arm64 によりサポートされているプラットフォーム............. 7
2.1.4.1 他のプラットフォーム................................................ 8
2.1.5 マルチプロセッサ....................................................... 8
2.1.6 グラフィックハードウェアのサポート.................................... 8
2.1.7 ネットワーク接続機器................................................ 8
2.1.8 周辺機器やその他のハードウェア..................................... 9
2.2 ファームウェアが必要なデバイス.......................................... 9
2.3 GNU/Linuxに適したハードウェアの購入................................... 9
2.3.1 独占的・閉鎖的なハードウェアを避ける................................ 9
2.4 インストールに利用できるメディア....................................... 10
2.4.1 CD-ROM/DVD-ROM/BD-ROM.......................................... 10
2.4.2 USBメモリ............................................................ 10
2.4.3 ネットワーク.......................................................... 10
2.4.4 ハードディスク........................................................ 11
2.4.5 Un*x・GNUシステム.................................................. 11
2.4.6 サポートする記憶装置................................................. 11
2.5 必要なメモリとディスクスペース......................................... 11

3 Debian GNU/Linuxのインストール前に
3.1 インストールプロセスの概要............................................. 12
3.2 既存データをバックアップしてください!.................................. 12
3.3 必要な情報.............................................................. 13
3.3.1 ドキュメント......................................................... 13
3.3.1.1 インストールマニュアル.......................................... 13
3.3.1.2 ハードウェアの文書............................................. 13
3.3.2 ハードウェア情報の取得先........................................... 13
3.3.3 ハードウェア互換性................................................ 14
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.............. 15
3.3.4 ネットワークの設定................................................ 15
3.4 必要な最低限のハードウェア.............................................. 16
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......................... 16
3.6 インストール前に使うハードウェア・OSの設定............................ 17
3.6.1 ブートデバイスの選択................................................. 17
3.6.2 ARMファームウェア.................................................. 17
3.6.3 U-Bootでのイーサネット MACアドレスの設定........................ 17
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............. 18
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
  4.2 Debianミラーサイトからのファイルのダウンロード
    4.2.1 どこでインストールファイルを探すか
  4.3 USBメモリでの起動用ファイルの準備
    4.3.1 ハイブリッドCD/DVDイメージを使ったUSBメモリの準備
  4.4 TFTPネットブート用ファイルの準備
    4.4.1 RARPサーバの設定
    4.4.2 DHCPサーバの設定
    4.4.3 BOOTPサーバの設定
    4.4.4 TFTPサーバの立ち上げ
    4.4.5 TFTPイメージを適切な場所に配置する
  4.5 自動インストール
    4.5.1 Debianインストーラを用いた自動インストール
  4.6 インストールファイルの整合性の検証

5 インストールシステムの起動
  5.1 64-bit ARMでのインストーラの起動
    5.1.1 コンソール設定
    5.1.2 Junoのインストール
    5.1.3 AppliedMicroMustangでのインストール
    5.1.4 TFTPによる起動
      5.1.4.1 U-BootでのTFTPのブート
    5.1.5 UEFIを利用したUSBメモリからの起動
    5.1.6 グラフィカルインストーラ
  5.2 アクセシビリティ
    5.2.1 インストーラフロントエンド
    5.2.2 基板デバイス
    5.2.3 高コントラストテーマ
    5.2.4 拡大
    5.2.5 Expertモード、Rescueモード、自動化インストール
    5.2.6 インストールしたシステムのアクセシビリティ
  5.3 起動パラメータ
    5.3.1 ブートコンソール
    5.3.2 DebianInstallerパラメータ
    5.3.3 起動パラメータで質問に答える
    5.3.4 カーネルモジュールへのパラメータを渡す
    5.3.5 カーネルモジュールのブラックリスト化
  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
      5.4.1.1 共通の問題
    5.4.2 起動設定
    5.4.3 カーネルの起動時メッセージの意味
    5.4.4 インストールで発生した問題の報告
    5.4.5 インストールレポートの送信

6 Debianインストーラーの使用法
  6.1 インストーラーの動作
    6.1.1 グラフィカルインストーラーの使用法
  6.2 コンポーネント入門
  6.3 それぞれのコンポーネントの使用法
    6.3.1 Debianインストーラーのセットアップとハードウェアの設定
      6.3.1.1 利用可能なメモリのチェック/低メモリモード
      6.3.1.2 地域オプションの選択
      6.3.1.3 キーボード選択
      6.3.1.4 DebianInstallerisoイメージの検索
      6.3.1.5 ネットワークの設定
        6.3.1.5.1 自動ネットワーク設定
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 .......................... 14
  3.2 最低限必要なシステム (推奨値) ................................. 16
Abstract

この文書は64-bit ARM（「arm64」）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。
arm64 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けすることでしょう。Debian GNU/Linux は、世界中から質の高い「自由なソフトウェア」をよりすくい、首尾一貫したディストリビューションとしてまとめてあげられています。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
よう。

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、それが可能ならば Debian インストール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くことを望んでいます。
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、有志の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たが、数年後には 1000 人もの 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律的な分析、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やであり、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 を若年ユーザー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？

GNU/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、ユーザーと交信したり指示を受け取り、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で

きます。オペレーティングシステムは、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。

GNU/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、GNU/Linux はマルチタスク、マルチユーザー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。しかしながら、GNU/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 GNU/Linux を有し
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 年に、フリーソフトウェア財団が GNU という Unix ライを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、GNU/Linux などの Unix ライクなオペレーティングシステムと

とに使用するための言語として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 GNU/Linux に寄与する中で、最大の単独貢献者はいまだに (GNU/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、GNU/Linux を可能にしたコミュニティーです。

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 年の、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 \[\text{参照してください。}\]

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。

GNU/Linux ユーザーは、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、ユーザ

者は、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、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ーを当惑させています。

GNU/Linux は、また、もっともクラッシュせず、複数のプログラムを同時に実行するのが優秀で、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・ビジネスユーザーにも人

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を組

み合わせることにより、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

れていました。Debian 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、

設定情報などが含まれています。Debian ディストリビューションでは、バグが発見・報告され、バグは

後、バグに修正が適用され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パッケージが再びリリースされます。バグが再び報告され、バグが修正され、パック

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や科学用途のデスクトップオペレーティングシステムやハイエ

ンドネットワークサーバまで、様々な用途に合わせたオプションが可能です。

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、熟練したユーザー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、ひとつずつ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のシンプル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DebianGNU/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使うことです（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読むためには、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 DebianGNU/kFreeBSDとは？

DebianGNU/kFreeBSDはkFreeBSDカーネルを用いたDebianGNUシステムです。このデbianの移植版は、現在のところ、i386とamd64アーキテクチャでのみ開発されていますが、その他のアーキテクチャにも移植される可能性があります。

DebianGNU/kFreeBSDはLinuxシステムではないので、Linuxシステム向けの情報を適用できない場合もあることに注意してください。

詳しくは、DebianGNU/kFreeBSD移植ページやdebian-bsd@lists.debian.orgメーリングリストを参照して下さい。

1.5 DebianGNU/Hurdとは？

DebianGNU/Hurdは、GNUHurd(GNUMachマイクロカーネルの上に実行する一群のサーバ)を用いたDebianGNUシステムです。

Hurdはまだ完成しておらず、日々の利用には不適ですが、作業は継続しています。現在のところ、Hurdはi386アーキテクチャでのみ開発されていますが、システムが安定してくれば、他のアーキテクチャにも移植される予定です。

DebianGNU/HurdはLinuxシステムではなく、Linuxシステム向けの情報を適用できない場合もあることに注意してください。

詳しくは、DebianGNU/Hurd移植ページやdebian-hurd@lists.debian.orgメーリングリストを参照して下さい。

1.6 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。組み込みシステム・ラップトップ・デスクトップ・サーバー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めます。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。インストーラーは、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。
インストーラーは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インストーラーページ、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。

1.7 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も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。

1.8 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、Debian11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も利用できます。

1.9 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ー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ーであっても、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。熟練したユーザーの方には、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、順番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って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と考えているのでしたら、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、インストールを起動します。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、パーティションのバーティション分割、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については、付録E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。
1.10 文書への支援のお願い

どんな支援、提案、特にパッチも非常にありがたいです。この文書の作業中の版は https://d-i.debian.org/manual/ にあります。そこでは、この文書の各アーキテクチャ向けの版や各言語版があります。

ソースも公開されています。貢献するための情報については、付録Eを参照して下さい。提案、コメント、パッチ、パッケージ名を使って下さい。ただしパッケージ名がすでに報告されていないかどうか、まずチェックしてください)を歓迎します。

1.11 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する）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。私たちが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、友達のコンピュータにインストールすることを手伝ってあげましょう。さらに、わずかな制限にさえ気をつけば、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、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、ユーザー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います。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ミラーファイルのcontent・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、DebianFAQ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、ユーザー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、このようなソフトウェアを生成するプログラマラがコミュニティの利益を考えることです。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、ユーザーには必要に応じてソフトウェアを修正する権限を与えられます。また、このようなソフトウェアの拡張が誰かによってなされば、その利益も享受できます。

Debian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、DebianFAQの「Debianパッケージ管理システムの基本」をご覧ください。
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DebianGNU/Linuxでテストされている64-bit ARM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、64-bit ARM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ることは避け、一般的な情報とさらなる情報が見つか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

DebianGNU/Linux11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。

<table>
<thead>
<tr>
<th>アーキテクチャ</th>
<th>Debianでの名称</th>
<th>サブアーキテクチャ</th>
<th>フレーバー</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD64 &amp; Intel64</td>
<td>amd64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel x86ベース</td>
<td>i386</td>
<td>デフォルトの x86 マシン</td>
<td>デフォルト</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Xen PV ドメインのみ</td>
<td>xen</td>
</tr>
<tr>
<td>ARM</td>
<td>armel</td>
<td>Marvell Kirkwood及びOrion</td>
<td>marvell</td>
</tr>
<tr>
<td>ハードウェアFPUがあるARM</td>
<td>armhf</td>
<td>複数プラットフォーム対応</td>
<td>armmp</td>
</tr>
<tr>
<td>64bit ARM</td>
<td>arm64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64bit MIPS (リトルエンディアン)</td>
<td>mips64el</td>
<td>MIPS Malta</td>
<td>5kc-malta</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cavium Octeon</td>
<td>octeon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loongson 3</td>
<td>loongson-3</td>
</tr>
<tr>
<td>32bit MIPS (リトルエンディアン)</td>
<td>mipsel</td>
<td>MIPS Malta</td>
<td>4kc-malta</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cavium Octeon</td>
<td>octeon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loongson 5</td>
<td>loongson-5</td>
</tr>
<tr>
<td>Power Systems</td>
<td>ppc64el</td>
<td>IBM POWER8以降のマシン</td>
<td></td>
</tr>
<tr>
<td>64bit IBM S/390</td>
<td>s390x</td>
<td>VM-readerやDASDからのIPL</td>
<td>generic</td>
</tr>
</tbody>
</table>

この文書はLinuxカーネルを用いた64-bit ARM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。

これは64-bit ARMアーキテクチャ用DebianGNU/Linuxの初公式リリースです。すでにリリース
とするに充分安定していると私たちは考えています。しかし、まだ他のアーキテクチャ版ほど広く使われていない（つまりユーザによるテストも多くない）ことから、いくつかのバグにでくわす可能性もあります。何か問題が起きたら、バグ追跡システムを使って報告してください。その際、そのバグが、Linux カーネルを用いた 64-bit ARM プラットフォーム上のものであることを必ず書き添えてください。また debian-arm メーリングリストの購読も必要かもしれません。

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 ではそういう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機能を提供するように設計されています。古い ARM アーキテクチャでは、ある SoC からその後継機の間でも大規模に変更されることがありました。しかし ARMv8 (arm64) では標準化が大きく進み、Linux カーネルやその他のソフトウェアのサポートが容易になっています。

サーバ版の ARMv8 ハードウェアは通常、Unified Extensible Firmware Interface (UEFI) 及び Advanced Configuration and Power Interface (ACPI) 標準を使って設定されます。この 2 つにより、一般的で機器に依存しない方法でコンピュータハードウェアをブート、設定できるようになります。これは x86 PC の世界でも共通しています。

2.1.4 Debian/arm64 によりサポートされているプラットフォーム

Arm64/AArch64/ARMv8 ハードウェアは Debian Bullseye リリースサイクルのなり終盤になって利用できるようになってきたため、このリリース時点で主流アーキテクチャのバージョンでサポートがマージされているプラットフォームが数多く観察されています。これは debian-installer を使うようにするための大きな要件なのです。以下のプラットフォームはこれら リリースの Debian/arm64 でサポートされていることがわかれています。挙げられているプラットフォームは全て、ただ一つのカーネルイメージでサポートします。

Applied Micro (APM) Mustang/X-Gene APM Mustang は Linux が動作する最初に利用可能となった ARMv8 システムでした。X-gene SoC を採用で、これはその後他のマシンでも採用されました。8 コア CPU、メモリ、USB、シリアルポートを搭載します。通共フォームファクターはデスクトップ PC 用と酷似していますが、別バージョンが今後多数出てくることが期待されます。ハードウェアはほとんどが主流アーキテクチャでサポートされていますが、Bullseye のカーネルでは現時点で USB のサポートが欠けています。

ARM Juno Development Platform Juno は 6 コア (2x A57, 4x A53) ARMv8-A 800Mhz CPU, Mali (T624) グラフィック、8GB DDR3 RAM、イーサネット、USB、シリアルポートを搭載する高性能開発用ボードです。システムの提示や電力テスト向けに設計されたため、小型でもなく安価でもありま
せんが、最初に利用できるようになったボードの一つです。オンボードのハードウェアは全て主流側カーネル及び Bullseye でサポートされています。

非 UEFI システムで debian-installer を利用している場合は、例えば debian-installer から起動したシェルで必要なコマンドを実行する等、インストールの最後に手作業でシステムをブート可能にしないといけないかもしれません。flash-kernel は U-Boot でブートする X-Gene システムの段取りを知っています。

2.1.4.1 他のプラットフォーム:
arm64 Linux カーネルの複数プラットフォームサポートにより、上記に明記されていない arm64 システムでも debian-installer を実行できるかもしれません。debian-installer により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いて、対象システムのデバイスツリーファイルが利用可能であれば、新しい対象システムでも普通に動作するかもしれませんが、その場合は通常、インストーラによってインストールした後も機能するため、UEFI を利用している場合は同様にシステムをブート可能にするはずです。UEFI を利用していない場合は、システムをブート可能にするために何らかのマニュアル設定を行う必要があるかもしれません。

2.1.5 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 とも呼ばれる）が利用できます。もとより、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。

標準の Debian11 カーネルイメージは、SMP をサポートしてコンパイルされています。非 SMP システムでも、問題なく動作します。

2.1.6 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オペレーティングシステムによる提供されます。3D グラフィックアクセラレーションやハードウェアアクセラ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、システムで使用され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」ファイルのインストール（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んで持っています。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ンはありますながら、それは希少です。グラフィックを一切持たないヘッドライト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でも使えるはずです。高速 3D グラフィックを使うためにはハイブリッドライバが必要です。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) 用のフレ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した。他のハードウェアではサポートのない 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ボ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。Debian11 は X.Org バージョン 7.7 を採用しています。

2.1.7 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。
64-bit ARM 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。
2.1.8 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わけではありません。

2.2 ファームウェアが必要なデバイス

デバイスドライバの可用性と別に、いくつかのハードウェアでは、デバイスを使用できるようになる前に、いわゆるファームウェアやマイクロコードを、デバイスに読み込む必要があるものもあります。もっとも一般的なのはネットワークインタフェースカード（特にワイヤレスNIC）ですが、例えばUSBデバイスやハードディスクコントローラでも、ファームウェアが必要なものがあります。

多くのグラフィックカードでは、基本機能は追加ファームウェア無しで利用可能なものの、先進的な機能を使用するためには、ファームウェアをシステムにインストールしなければなりません。いくつかのケースでは、インストール後に再起動した際に画面がブランクあるいは文字化けした状態になってしまうことがあります。このような現象が起きた場合であっても、ログインを試行可能な回避策があります（項6.4.3を参照してください）。

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、メーカーによって提供されるEEROM/フラッシュチップに永続的に保存されています。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む必要がなくなり、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みます。特定の状況下で必要になるのは、ドライバが追加ファームウェアを要求する場合があるのですが、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Linuxでしっかりサポートされていることが確認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を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。もしなければ、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のようなUSBマ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なっている一方、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期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ドライバのサポートが行われないという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 USBメモリ

USBメモリとしてよく知られるUSBフラッシュディスクは広く利用されるようになった安価なストレージデバイスです。いまのコンピュータシステムではほとんどがこういったUSBメモリからのdebian-installerのブートにも対応しています。いまのコンピュータシステムの多く、特にネットブックや薄いノートはもう光学ドライブをまったく持たず、こういったシステムに新しいOSをインストールする場合、USBメモリからのブートは標準的な手段となっています。

2.4.3 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、迅速で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勧め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、ディスクレスインストールすることも一つの選択です。
CHAPTER 2. 必要なシステム

2.5 必要なメモリとディスクスペース

2.4.4 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、特殊な場合だけ使用してください。

2.4.5 Un*x・GNUシステム

他のUnix系システムが稼働していた場合、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へ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はおおすすめします。

2.4.6 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 260MB 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のみお勧めします。

---

グラフィカルインストーラをサポートするインストールイメージは、テキストインストーラのみをサポートするイメージよりもメモリが必要で、260MB未満のシステムで使用するべきではありません。そういったシステムで通常のインストーラかグラフィカルインストーラを選ぶ選択肢が表示された場合は、前者を選択してください。
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、もっともありがち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時に、OSの新バージョンへのアップグレードの際、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いでも、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いでほしょうありません。

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も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、既存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。
CHAPTER 3. DEBIAN GNU/LINUX のインストールシステムの使用方法

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ずいたかを知るお手伝いをします。このインストール産の、そんな主なソフトウェア俳優をご紹介します。

インストーの debian-installer は、このマニュアルの主役です。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 サーバー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です。「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あれば、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いたとしても、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を恐ないので、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。ほんの数分間程余裕を配慮することで、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。通常は必要ではないとはいえ、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 の次期リリース用インストールガイドの開発版です。これは様々な形式と様々な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にから集められます。

- 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やハードドライプ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>
<thead>
<tr>
<th>ウェア</th>
<th>必要な情報</th>
</tr>
</thead>
<tbody>
<tr>
<td>ハードディスク</td>
<td>ドライブの台数</td>
</tr>
<tr>
<td>IDE(PATAとしても知られる)、SATA、SCSIのどれか</td>
<td>ハードドライブの接続順序</td>
</tr>
<tr>
<td>利用できる空き領域</td>
<td>ハーネシング</td>
</tr>
<tr>
<td>パーティション</td>
<td>他のOSがインストールされているパーティション</td>
</tr>
<tr>
<td>ネットワークインタフェース</td>
<td>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</td>
</tr>
<tr>
<td>プリンタ</td>
<td>メーカーと型番</td>
</tr>
<tr>
<td>ビデオカード</td>
<td>タイプ/モデルとメーカー</td>
</tr>
</tbody>
</table>

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てい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いるにも、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か判定するのが常に簡単ではないことです。理解も必要で、ディ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、そのために時に異なるチップセットが同じ製品名と言われている場合、新しい二つある場合、異なる二つ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る場合、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ィバイスIDを確認するのが良い方法です。USB/PCI/PCI-Express/ExpressCardデバイスに関しては、ベンダーおよび製品IDを基にしているため異なるドライバを使う必要がある場合、一方には使うドライバが何もないということもあります。

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Ispci-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：「04:00.0VGA compatible controller [0300]: Advanced Micro Devices [AMD] nee ATIRV710 [Radeon HD4350] [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。Windows7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lsusb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せんが、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ないです。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ライセンスのものしか含まれません。つまり、フリーライセンスのものでなければなりません。そういったフリーライセンスの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すが、debain-installerのように必要なソフトウェアパッケージを自動的に検出するようなものがあるため、フリーライセンス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やそれと同等に接続の場合)、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(自分で決まることができるかもしれません)
- 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サーバーサポートストロールプロセスの間、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やケーブルモード(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の提供者より提供されることがよくあります)がある場合、それがネットワーク接続を処理している場合、通常DHCPがデフォルトで使用されるようになっています。

WLAN/WiFiネットワークを使用する場合、以下の情報も必要です。

- ウィ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。
CHAPTER 3. DEBIAN GNU/LINUX のインストール…

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ない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うと思います。

Table 3.2 最低限必要なシステム (推奨値)

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

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245MB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ーをインストールするのをお勧めします。代替品には、xfce4, icewm, wmaker が含まれますが、他にも選択できます。

デスクトップ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 共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を割り当てておくべきです。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。ある部分に家具を入れても、それは他の部分には影響しないというわけです。

システム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システムとして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ですが、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、このような、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るです。

一般的に、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場合には、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に、あらかじめどこかしておくことが必要です。

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が、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あまり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からでもできます。経験者ですが、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ます。 Zagyで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、確実を期するのがpreferredで、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くこと。 Windowsなどの他のOSからインストールすると、Debianを起動する機能が破壊されてきたり、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やすそうに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せんが、先にそちらのシステムをインストールしておくと、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しょう。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システムツールで、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 ブートデバイスの選択

#### 3.6.2 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す。名義上同一のファームウェアを利用していても、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加え、それによって作成したファームウェアを提供し、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、製造者により変更された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いです。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きたり、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、オーケストライテスの命名方法は異なる製造者が変更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環境を提供するのはほぼ不可能です。

#### 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えません。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ために全体で一意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アドレスが決まっていない場合、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アドレスをブート時に変わる可能性があり、変更が発生した場合、ユーザが手作業でMAC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す。例えばリカバリ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に、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)。
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 ベージと 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、マシンをこれからの起動できるなら、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 だけで足りますし、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、手前頭の CD だけでは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では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/ をご覧ください。

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いので、CD/DVD セットを持っていれば、CD/DVD セットを持っていただくのですから、最初のシステムインストーラーの起動に USB メモリ、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ると、デスクトップ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る場合は、インストーラのシステムファイルをダウンロードして、USB メモリ、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-arm64/-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。
4.3 USBメモリでの起動用ファイルの準備

USBメモリの準備をするには、GNU/Linuxが既に動いている、USBをサポートしているシステムを使うことをお勧めします。現在のGNU/Linuxでは、USBメモリを挿すと自動的に認識するでしょう。そうならない場合は、usb-storageカーネルモジュールをロードしているかを確認してください。USBメモリを挿すと、/dev/sdX（「X」はa〜zの範囲の文字）というデバイスにマッピングされます。どのデバイスがUSBメモリかは、挿した後でdmesgコマンドを実行すると見られます。USBメモリに書き込むには、ライトプロテクストスイッチを切る必要があります。

警告
この方法を使うとデバイス上の既存の物は破壊されてしまいます！USBメモリの正しいデバイス名を必ず確認して使用してください。間違ったデバイス名を使用すると、例えばハードディスク内のすべてのデータを失うといったことが起こります。

4.3.1 ハイブリッドCD/DVDイメージを使ったUSBメモリの準備

Debianのインストールイメージは直接書き込めるようになり、起動可能なUSBメモリを作るのがとても簡単になりました。USBメモリに合うイメージ（netinstやCD、DVD-1、netboot等）を選んでください。インストールイメージの取得については項4.2を参照してください。

別の方法として、とても小さな、サイズが数メガしかないUSBメモリ用に、netbootディレクトリからmini.isoをダウンロードできます（場所については項4.2.1に記載があります）。

選択したインストールイメージは、既存の内容を上書きして、直接USBメモリに書き込む必要がありません。例えば、既存のGNU/Linuxシステムを使っている場合、以下のようにしてイメージファイルをUSBメモリに書き込みます。完了したら、確実にアンマウントしてください。

```
# cp debian.iso /dev/sdX
# sync
```

他のオペレーティングシステムでのやり方については、Debian CD FAQで確認できます。

重要項目
イメージはパーティションではなくディスクデバイス全体を指定する必要があります。例えば/dev/sdb1ではなく/dev/sdbとなります。イメージの書き換えるunetbootinのようなツールは使わないでください。

重要項目
ほとんどのユーザは、インストールイメージを単にUSBメモリに書き込むだけでうまく行ってしまう。下記の選択肢はもっと複雑なので、主に特殊なニーズがあるユーザ向けです。

USBメモリに配置したハイブリッドイメージは、ストレージの領域すべてを専有するわけではありません。そのため、ファームウェアファイルやパッケージ、その他選んだファイルを保持するために、空き領域を使用するのを、検討する価値があるかもしれません。あなたが、ひとつしかUSBメモリを持っていないか、ひとつのディバイスにすべて格納する必要があるときに便利です。

USBメモリにふたつ目のFATパーティションを作成し、パーティションをマウントしてから、ファームウェアのコピーを展開をそこに行います。以下に例を示します。

```
# mount /dev/sdX2 /mnt
# cd /mnt
```
CHAPTER 4. システムインストールメディアの… 4.4. TFTPネットブート用ファイルの準備

# tar zxvf /path/to/firmware.tar.gz
# cd /
# umount /mnt

もしかすると、USBスティックにmini.isoを書き込んだかもしれません。この場合、ふたつ目のパーティションを作成する必要はなく、都合のいいことに、すでに作成済みでしょう。USBメモリを押し直すと、ふたつのパーティションが見えるはずです。

### 4.4 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」の特定の場所からの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サーバー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サーバー、またはBOOTPサーバー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つつつ、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のプロトコルです。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。

#### 4.4.1 RARPサーバーの設定

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サーバーシステムや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デーモンを起動します。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。

#### 4.4.2 DHCPサーバーの設定

フリーソフトウェアのDHCPサーバのひとつに、ISCのdhclient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）

```plaintext
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";
}
```

21
 CHAPTER 4. システムインストールメディアの… 4.4. TFTPネットブート用ファイルの準備

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をすべて行っています。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 経由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 dhcpc を再起動してください。

4.4.3 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 すればOKです。BOOTPサーバがDebianで動かない場合は、以下のようにします。

```
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けばなりません。/etc/bootptabの例を示します。

```
client:
  hd=/tftpboot:"
  bf=tftpboot.img:"n
  ip=192.168.1.90:"n
  sm=255.255.255.0:"n
  sa=192.168.1.1:"n
  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るでしょう。「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4.4.5を参照してください。

対照的に、ISC dhcpdを使ってのBOOTPの設定は実に簡単です。dhcpcd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4.4.2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 dhcpc を再起動するだけです。

4.4.4 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 デー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。
4.5 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。詳細情報はFAI homepageをご覧ください。

4.5.1 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。

4.6 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。

---

Debianで利用できるintftpd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。

4.4.5 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。
Chapter 5

インストールシステムの起動

5.1 64-bit ARM でのインストーラの起動

5.1.1 コンソール設定

stretch の arm64 debian-installer イメージではグラフィカルインストーラは有効化されていないため、シリアルコンソールを利用します。コンソールデバイスはファームウェアによって自動的検出されるはずですが、検出されなかった場合は GRUB メニューから linux をブートした後「Booting Linux」メッセージを見ることになります。そうなるともう何も起こりません。この問題が起きた場合はカーネルコマンドラインで特定のコンソール設定を行う必要があります。GRUB メニューで e を押して「Edit Kernel command-line」に進み、

```plaintext
--- quiet
```

を

```plaintext
console=<デバイス>,<速度>
```

、例えば

```plaintext
console=ttymA0,115200n8
```

のように変更します。変更できたら Control-x を押して新しい設定でブートを続けます。

5.1.2 Juno のインストール

Juno には UEFI があるためインストールは明快です。最も実践的なのは USB メモリからインストールする方法です。USB でのブートを機能させるためには最新のファームウェアが必要です。2015 年 3月以降にビルドされた http://releases.linaro.org/members/arm/platforms/latest/ でテストが成功しています。ファームウェアの更新については Juno の文書を参照してください。

USB メモリに書き込んだ標準の arm64 CD/DVD イメージを用意します。背面の USB ポートに差し込みます。シリアルケーブルを背面の上部にある 9 ピンのシリアルポートに差し込みます。ネットワーク接続が必要 (netboot イメージ) であればイーサネットケーブルをマシン全面のソケットに差し込んでください。

シリアルコンソールを 115200、8 ビットパリティなしで実行し、Juno をブートします。USB メモリからブートして GRUB メニューが表示されるはずです。コンソール設定は Juno で正しく検出されないため、ただ Enter を押してもカーネル出力には何も表示されません。(項 5.1.1 で説明しているように) コンソールを
CHAPTER 5. インストールシステムの起動

5.1. 64-BIT ARMでのインストールの起動

console=ttymA0,115200n8

と設定します。Control-xでブートするとdebian-installerの画面が表示され、標準のインストールを続けられるはずです。

5.1.3 Applied Micro Mustangでのインストール

このマシンではUEFIが利用できますが、通常はU-Bootを使うようになっているため、UEFIファームウェアをまずインストールしてから標準のブート/インストールを行う方法か、U-Bootを使ってブートする方法のどちらかが必要です。arm64アーキテクチャではグラフィカルインストーラは有効化されていないため、インストール処理の制御にはシリアルコンソールを使う必要があります。

推奨するインストール方法はマシンに同梱されているopenembeddedシステムを使ってdebian-installerカーネルとinitrdをハードドライブにコピーし、それをインストーラからブートします。TFTPを使ってカーネル/dtb/initrdをコピーし、ブートする方法(項5.1.4.1)もあります。インストールしたイメージからブートするためには、インストール後に手作業での調整が必要です。

シリアルコンソールを115200、8ビットパリティなしで実行し、マシンをブートします。マシンを再起動して「Hit any key to stop autoboot:」が表示されたら何かキーを押してMustang#プロンプトに入ります。それからU-Bootのコマンドを使ってカーネル、dtb、initrdを読み込み、ブートします。

5.1.4 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必要です。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4.4で説明します。

5.1.4.1 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実行。

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 dnsip <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していいか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最適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：

25
インストールシステムの起動

5.1.64-BIT ARMでのインストーラの起動

CHAPTER 5. インストールシステムの起動

5.1. 64-BIT ARMでのインストーラの起動

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

読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めます。3つ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利用するイメージのフォーマットに依存します。ulimage/unlinitr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。

5.1.5 UEFIを利用したUSBメモリからの起動

USBからブートできるコンピュータの場合はおそらくこれが最も簡単なインストール方法です。さて、項3.6.1と項4.3の内容すべてを準備しました。それではUSBコンピュータにUSBメモリを差し込んで、コンピュータを再起動してください。システムが起動し、USBメモリを作成するのに柔軟な方法で有効にしていない、などという事さえなければ、グラフィカルなブートメニューが（それをサポートしているハードウェアでは）表示されるはずです。ここで様々なインストールオプションを選択するか、単にEnterを押してください。

5.1.6 グラフィカルインストーラ

グラフィカルバージョンのインストーラは64-bit ARMを含めた、限られたアーキテクチャでのみ利用できるようになっています。グラフィカルインストーラとテキストベースのインストーラは基本的に同一のプログラムを異なるフロントエンドと組み合わせているため、機能は本質的に同一です。機能は同等ではありませんが、グラフィカルインストーラには大きな利点がいくつかあります。最大の利点はサポートしている言語が豊富であること、つまりテキストベースの「newt」フロントエンドで表示できない文字セットを使える点です。使い勝手についてでもマウスを使えるという選択肢や場合によっては複数の質問を1画面に表示できる等、いくつかの利点があります。

テキストベースのインストーラと全く同じように、グラフィカルインストーラの起動時にも起動パラメータを追加できます。
注意

グラフィカルインストーラはテキストベースのインストーラと比べ、実行に著しく多くのメモリを必要とします（640MB）。十分なメモリを利用できない場合には、テキストベースの「newt」フロントエンドに自動的にフォールバックします。

システムのメモリ量が260MBに満たなければ、テキストベースのインストーラをブートした場合には機能したとしても、グラフィカルインストーラでは全くブートできないかもしれません。利用可能メモリの少ないシステムではテキストベースのインストーラの利用を勧めます。

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プートメニューで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（プートメニューは一度だけビープ音を鳴らします）は続いてEnterを押さないといけません。UEFIシステムの場合は（プートメニューは二回ビープ音を鳴らします）Enterを押してはいけません。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、ここの場合に続いて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なくてはいけないこともあります。

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 または 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るコンピュータに、ビデオカード（フレームバッファ）とキーボードもつっている場合には、カーネルに console=devic 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。

通信速度やパラメータといった、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知れません。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るために、パラメータの入力を簡単にするため、『短縮形』を持ってきます。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、短縮形を使用しています。

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。

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」のものと、「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、インストールメニューで、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、すべき

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。それを超えると、カーネルはパニックしています。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。
インストールシステムの起動 5.3. 起動パラメータ

ここで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ル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ルが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、インストール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危険である、無人インストールの際によく利用します。

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起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ているか（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（このためのプ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：誤った値が返された）に、ネットワークの自動設定をせずに（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、そういう目的のために設定があります。

29
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ご留意ください。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4.5 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4.5 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、特殊な文法が使用されるようになります。詳細は、項B.5.2 をご覧ください。

auto-install/enable (auto) 通常 preseed の前に送れる質問を、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かなければならないのに、手で挿入するようユーザが指示されないなど。大半のスロットローディング、スリムライン、キャビテ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。
この変更により、より無駄のないシステムが得られますが、通常可能である推奨する機能が欠落するかもしれません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 キーで認証されたリポジトリがない場合、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を与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だけです。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
CHAPTER 5. インストールシステムの起動  5.4. インストールプロセスのトラブルシューティング

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 オートメカジションや 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 のミラーワーク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 を使用するように強制できます。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、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、カーネルセット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。しかしこのような状況下では、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、以下のようになります。

```
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むと、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ルの起動に失敗するかもしれません。また、インストーラ（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。

31
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、次にインストーラ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。

こ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ライタで使用するような、高速で焼いたディスクから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。

```bash
$ 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。

```bash
$ 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。

```bash
ata1.00: ATAPI: MATSHITADVD-RA M UJ-822S, 1.61, max UDMA/33
ata1.00: configured for UDMA/33
scsi 0:0:0:0: CD-ROM MATSHITA DVD-RA M 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
```

32
5.4 インストールプロセスのトラブルシューティング

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れば、modprobe を用い、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ければ、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 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っているからです。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、何か間違っていなくても修正するか、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/ にあるはずです。
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宛に送ってください。

<table>
<thead>
<tr>
<th>Package:</th>
<th>installation-reports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot method:</td>
<td>&lt;インストーラの起動方法は？CD?/DVD? USB メモリ? ネットワーク?&gt;</td>
</tr>
<tr>
<td>Image version:</td>
<td>&lt;イメージをダウンロードした URL 全体がベストです&gt;</td>
</tr>
<tr>
<td>Date:</td>
<td>&lt;インストールした日時&gt;</td>
</tr>
<tr>
<td>Machine:</td>
<td>&lt;マシンの説明（例 IBM Thinkpad R32）&gt;</td>
</tr>
<tr>
<td>Processor:</td>
<td></td>
</tr>
<tr>
<td>Memory:</td>
<td></td>
</tr>
<tr>
<td>Partitions:</td>
<td>&lt;df -T の結果； 生のパーティションテーブルが望ましいです&gt;</td>
</tr>
<tr>
<td>Output of lspci -nn and lspci -vnn:</td>
<td></td>
</tr>
<tr>
<td>Base System Installation Checklist:</td>
<td></td>
</tr>
<tr>
<td>[O] = OK, [E] = Error (please elaborate below), [ ] = didn’t try it</td>
<td></td>
</tr>
<tr>
<td>Initial boot:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Detect network card:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Configure network:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Detect media:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Load installer modules:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Detect hard drives:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Partition hard drives:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Install base system:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Clock/timezone setup:</td>
<td>[ ]</td>
</tr>
<tr>
<td>User/password setup:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Install tasks:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Install boot loader:</td>
<td>[ ]</td>
</tr>
<tr>
<td>Overall install:</td>
<td>[ ]</td>
</tr>
</tbody>
</table>

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 インストーラーの使用法

6.1 インストーラーの動作

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、必要ならユーザーに質問をして、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）質問しかありません。これにより、ユーザー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。インストーラーは、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、インストーラーメニューに、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、ユーザーはインストールメニューを目指すことなく、簡単な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。

インストーラー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。

パワーユーザーは、メニュー駆動インターフェース（自動で順に各ステップを実行するインストーラーではなく、ユーザー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）でインストーラーを使用するには、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ーを起動する必要があります。これは、インストーラー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、debian-installer は 2 つの異なるユーザーインターフェースをサポートしています。テキストベースのものとグラフィカルなものです。ブートメニューで「Graphical install」を選択しない限り、デフォルトではテキストベースインターフェースを使用します。グラフィカルインストーラーに関する詳細情報は、項5.1.6 をご覧ください。

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、右矢印は 「前方」、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、またリスト全体を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。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に項目を選択します。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ルメニューに戻ります。

これらのメッセージは /var/log/syslog で見つけることができます。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には /var/log/installer/ にあります。
6.1.1 グラフィカルインストーラーの使用法

グラフィカルインストーラーは基本的にはテキストベースのインストーラー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ー自体は 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ー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ませんが詳細は項 6.3 にあります。

main-menu インストーラーの操作中にユーザー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ないでしょう。一方、あなたの入力が必要なエラーが発生した場合、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。インストーラー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 インターネットへの通信できるように、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) を検索します。

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。

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 はちょっと APT（適切）じゃない) 選択したミラーサーバー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以外のユーザーの追加を行います。

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けツールです。

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-lvm LVM(Logical Volume Manager)の設定について、ユーザーの補助を行います。

partman-md ソフトウェアRAID(Redundant Array of Inexpensive Disks)の設定をユーザーに許可します。このソフトウェアRAIDは、新しい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 インストーラー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。

pkgsel 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、ユーザー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ーがそれぞれ、ハードディスクにブートローダー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。ブートローダーの多くは、コンピュータが起動するこ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、インストーラー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法

本節では、各インストーラー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、ユーザーに認識できる段階でグルーピ化されました。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 インストーラーのセットアップとハードウェアの設定

Debianインストーラーが起動して、最初の画面が表示されるときでしょう。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ったり、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、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てくれます。

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でしょう。最初の検出では、インストーラー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にある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がまず行うことの一つが、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。

インストーラー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言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。

ここで充分でなければ、インストーラーは、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ることにします。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。

インストーラー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。インストーラーは、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（インストーラー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。

インストーラーに項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なります。

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、インストーラーは自動的に英語になります。

選択した地理的住所（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。このリストが言語名順で並んでいます。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、その国だけのリストを表示します。他のリストが表示され続けます。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、その国だけのリストを表示します。他のリストが表示され続けます。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。

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、その国が属する国を表示す

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、ないし地域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になります。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せられています。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に、その内容をチェックします。前者の場合は完了しますが、後者の場合はiso-scanは別のイメージを探すします。

インストーラー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がインストーラー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ーは、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。これらの回避策としては、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_toram 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、幅広い原因を考えられます。エラー発生時に何が起きたかを確認します。
認するには、4番目のコンソールに表示され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、特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、いくつか質問を行います。特に、IPアドレス、ネットマスク、ゲートウェイ、ネットワークのアドレス、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、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いければ、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/Linux7.0（「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サーバー（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 ユーザーとパスワードのセットアップ
クロックの設定直前に、インストーラーは「root」アカウントや、最初のユーザーアカウントのセットアップを行います。その他のユーザーアカウントは、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、スーパーユーザーとも呼ばれます。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アカウントがrootになるので、sudoコマンドを使用できます。
6.3.2.2 一般ユーザーの作成

システムは、この時点で一般ユーザー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？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、スーパーアカウント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、ユーザーのフルネームの入力を求められます。次にユーザー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、必要充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

インストーラーはまず、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ーに（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、システムはその一覧を表示せず、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い場合は、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。

```bash
# 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 はユーザー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になる前準備を整えてあります。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、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
6.3. それぞれのコンポーネントの使用法

インストーラーは、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ー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。インストラーが、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 LVM 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、インストーラー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、インストーラーは、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
 CHAPTER 6. DEBIAN インストラーソーの使用法 6.3. それぞれのコンポーネントの使用法

も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、インストーラーは変更しておかしい確認していきます。

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。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。

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

(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、インストーラーは独立した /boot パーティションを作成します。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。

EFI モードでブートした場合、ガイドパーティション分割に EFI プートローダ用に FAT32 の起動可能ファイルシステムでフォーマットされた追加パーティションがあるはずです。このパーティションは EFI システムパーティション (ESP) とも呼ばれます。手動で ESP としてパーティションをセットアップするように、フォーマットメニューに追加メニュー項目があります。
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。

<table>
<thead>
<tr>
<th>SCS1 (0,0,0) (sda) - 6.4 GB WDC AC36400L</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 primary 16.4 MB B f ext2 /boot</td>
</tr>
<tr>
<td>#2 primary 551.0 MB swap swap</td>
</tr>
<tr>
<td>#3 primary 5.8 GB ntfs</td>
</tr>
<tr>
<td>pri/log 8.2 MB FREE SPACE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCS2 (1,0,0) (sdb) - 80.0 GB ST380021A</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 primary 15.9 MB ext3</td>
</tr>
<tr>
<td>#2 primary 996.0 MB fat16</td>
</tr>
<tr>
<td>#3 primary 3.9 GB xfs /home</td>
</tr>
<tr>
<td>#5 logical 6.0 GB f ext4 /</td>
</tr>
<tr>
<td>#6 logical 1.0 GB f ext3 /var</td>
</tr>
<tr>
<td>#7 logical 498.8 MB ext3</td>
</tr>
</tbody>
</table>

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つかあります。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
6.3.3.3 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、前述と同様な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作成を行うように依頼する)。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(基本カテゴリか、場所(空き領域の先頭か最後からか)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、無符号ファイルシステムとすべて使うか、全く使わないかを決定する方法:です。それ以外の、マウントポイントの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はありません。あらかじめ選択されたデフォルト値が変更されれば、自由にお好みのも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、または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に使用するのと同じ画面ですので、同様に設定を変更できます。一見わかりづらいのは、表示されているパーティションのサイズを変更するか、サイン変更ができます。動作が変わっていているファイルシステムは、少なくともfat16, ext2, ext3, swapです。このメニューではパーティション削除もできます。

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たと、この問題を修正するまでpartmanは無効になります。

EFIシステムパーティションをフォーマットし忘れた場合は、partmanがそれを検出し、行うまでに進むことができません。

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。あるいはど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します。ディスクに変わ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。

6.3.3.4 マルチディスクデバイス(ソフトウェアRAID)の設定
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)と呼びます。
MDは基本的に別ディスクにパーティションを束ねて、論理ディスクの形に結合したものですが、このディスク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、以下のサブシステムをサポートしています。
RAID0 RAIDOはパフォーマンスに主眼をおしています。RAIDOは全入力データをstripesへ分割し、均等にディスクスプ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が向上します。
RAID0 異なるディスクを組み合わせたものです。データを分割し、各ディスクに1つのコピーを作成します。RAID0は速度が速いが、冗長性が低いです。

RAID1 2つのディスクを鏡像するものです。データを鏡像し、冗長性を提供します。

RAID5 3つのディスクを組み合わせたものです。データを分割し、各ディスクに1つのコピーを作成します。パリティ情報を計算し、冗長性を提供します。

RAID6 4つのディスクを組み合わせたものです。データを分割し、パリティ情報を計算し、冗長性を提供します。

RAID10 2つのストライピング (RAID0)とミラーリング (RAID1)を組み合わせたものです。データを分割し、各ディスクに1つのコピーを作成します。パリティ情報を計算し、冗長性を提供します。
### 章節6. DEBIANインストーラーの使用法 6.3. それぞれのコンポーネントの使用法

<table>
<thead>
<tr>
<th>テイプ</th>
<th>デバイス最小構成数</th>
<th>予備デバイス</th>
<th>ディスク破損に耐えるか？</th>
<th>利用可能領域</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID10</td>
<td>2</td>
<td>任意</td>
<td>yes</td>
<td>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）</td>
</tr>
</tbody>
</table>

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を用いる例が選択されたり得ます。

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を、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なければなりません。

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割かれている場合、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、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屬性を設定してください。

46
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」ユーザー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を）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った管理を行うことになります。

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いうことです。

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 500GB 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、パーティションが 460GB へと新品交換されたので、ユーザーの空き容量がすこし増えたことになります。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されています。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に、マークをつけなければなりません。これにより、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が Debain を追加インストールしたい場合、古い (既存の) LVM は消去されますが (どちらかの理由) 誤って LVM タイプコードを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。新しく 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しか見えません。

暗号化するのに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ー、メールサーバー、プリンターサーバー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。現在debain-installerは暗号化されたパーティションから起動するようにはなっていますが、現在.debain-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。

debain-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置いています。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しています。

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、debain-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おりませんが、信頼できるアルゴリズムであることを保証するには、AES(Advanced Encryption Standard)を推奨しています。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、攻撃される恐れは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わないということです。（もちろん院当たって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限り、生きているうちに解読されないでしょう）

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削除するからです。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、復元できなくなるのです。

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る。そうしないと攻撃者が、パーティーションのどの部分を使用中で、どの部分が使用していないかを見つけられることで、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いなかったデータを、復元しにくくなります。

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。そこで今度は暗号化されたパーティションの設定という項目があります。これを選択すると、削除するよう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書き込み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、パスフレーズを証されます。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を（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るか確認してください。よくわからないと、別の仮想端末に切り替え、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ことをした場合は、あなたが後で驚くようなことは起こらないでしょう。この状況はいくつかの原因考えられます。インストール中に別のキーボード配列に切り替えた場合で、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かもしれません。

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、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をかかることもあります。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、ファイルを/dev/nullに流す（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、暗号化パーティションが、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で暗号化したパーティションを示します。

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

今度は、パーティション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場合は変更を行います。

括弧内の識別子（ここではsda2_crypt）と、暗号化パーティション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、この情報を必要になります。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で扱われています。

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。

注：3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。
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なければ、ある程度時間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。元のインストーラーの画面に戻るには、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。

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。ほとんどのユーザーは、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ーから行えます。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ることもあります。

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。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、すばらしいユーザーインターフェースと統合しているので、新しいユーザーにお勧めします。

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ですみます。ほとんどの場合、インストーラー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、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、インストーラーが尋ねてきます。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、スキップしたくなるかもしれません。

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、どちらかというと下位のツールです。apt はもっと上位のツールで、適切に dpkg を起動しません。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。
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。

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用、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使った、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います。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、インストーラー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

インストールに関してよくある質問に、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くが、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なら、ネットワークミラーを使用すべきです。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タスクに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) 優先を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す。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の利用点は、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損ねることなく、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では、良いインターネット接続がない場合でも、一般的に考えられます。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、インストールは常にそれを使用します。従って、ミラーを設定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。

51
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。

提示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（おそらくユーザー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、その情報はインストーラー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が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、とはいかないと思います。

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。86695個もの利用可能なパッケージから、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サーバー」、「SSHサーバー」といった具合です9。項D.2に、利用可能な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
インストーラーの標準ユーザーアイフェースでは、タスクの選択をスペースバーでトグルできます。

---

9表示されるリストは、インストーラーが単にtaskselプログラムを起動しているだけ、ということを知っていただいてください。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。
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。

各サーバータスクでは、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 サーバー: apache2; SSH サーバー: openssh。
「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。ここには、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なければ、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ています。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、ユーザー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、インターネットが、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、長い時間をかかることでしょう。一度、パッケージのインストールを始めるなら、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さければ、インストール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、インストラーは既にインストールされている他の OS の検出を試します。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、アーキテクチャことに (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フェースはないため、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収録しています。デバイスが現在サポートされているかどうかを検出し、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されていない、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、ユーザーが介せず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、/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、(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ー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。ほとんどのdebian-installer の後片付けです。

6.3.8.1 システム時計の設定

インストーラーは、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

インストーラー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ていませんが、何かうまく行かない時に、ユーザー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った場合、別の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でインストーラー自体に戻ってください。

グラフィカルインストーラーでは、項6.1.1もご覧ください。
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ーに戻ります。

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。

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、基本的には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うと、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、インストーラーで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 Installation over network-console
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から行えるようにします。ネットワークを使用すると言うことは、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を含んでいる。（でもこの部分は項4.5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に呼び出して、インストールメディアからインストーラー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くるはずです。ユーザー名は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

スペースバーの左側にあるAltキーと、F2機能キーを同時に押してください。
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

インストーラーのsshサーバーは、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、通常なら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

順番にいくつか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、もう一度行う必要があります。

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つだけにするべきですが、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、基本機能は損ねられませんが、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込します。

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おり、インストール中にそのデバイスが必要ない場合には、ファームウェアの読み込みをスキップす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ていない場合（特にradeonは含まれません）、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、まったく変化せずに funktionます。この場合に該当するかもしれません。その結果として、ハードウェアの能力を取り出せないかもしれません。この場合に該当するかどうかを確認し、興味を持たれたのでしたら、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正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間違くなりずサポートされている、FATを使用するのでお勧めします。

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/
- 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、インストーラー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読み込み不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 はインストーラー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パッケージからロードしたシステムは動作するため、インストールしたシステムも動作します。

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 インストールしたシステムの設定を完了する

関連するファームウェアが取得できなかった、その時点ではファームウェアをインストールしないという選択をした等、どのようにインストールが実施されたかに応じて、インストール作業中では検出されなかったファームウェアが必要になる可能性があります。いくつかのケースでは、インストールが成功したにも関わらずシステム再起動後にブランクあるいは文字化けした画面が表示される可能性があります。このような場合、以下の回避策を試みることが出来ます:

• カーネルコマンドラインに nomodeset を渡す。「fallback graphics」モードでの起動ができる可能性があります
• Ctrl-Alt-F2 の組み合わせを使って、動作しているログインプロンプトが表示される VT2 (仮想端末 2) にスイッチする

インストール後のシステムに一旦ログインできたら、見つけられていなかったファームウェアの自動検出が実施可能となります。以下の手順に従って必要な手順を実施することで自動検出を有効化できます:

1. isenkram-cli パッケージをインストールする
2. 「root」ユーザーで isenkram-autoinstall-firmware コマンドを実行する

一般的に、すべてのカーネルモジュールが正しく初期化されるのを担保するには再起動するのが最も手軽な方法です。暫定的な処置として nomodeset オプションを指定して起動していた場合に、これは特に重要な部分です。

注意

ファームウェアパッケージのインストールには、パッケージアーカイブの non-free セクションを有効化する必要が生じる可能性が非常に高います。Debian GNU/Linux 11.0 では、isenkram-autoinstall-firmware コマンドの実行によって汎用的なミラーサイトを指定した専用ファイル (/etc/apt/sources.list.d/isenkram-autoinstall-firmware.list) を自動的に生成します。

6.5 カスタム化

シェル(項6.3.9.2参照)を使う事で、インストール作業を例外的なケースに合わせるため、じっくりとカスタマイズできます。
6.5.1 代替 init システムのインストール

Debian は systemctl をデフォルトの init システムとして利用しています。ですが、他の init システム
(sysvinit や OpenRC など) もサポートされていて、別の init システムを選択するのに最も楽なタイミ
ングはインストール作業時となります。どのようにして切り替えるかの詳細な作業内容はDebian wiki
の Init のページを参照してください。
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。インストール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ル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に不慣れなら、より経験のあるユーザの手助けが必要かもしれません。64-bit ARM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、起動中にパスフレーズ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フレーズを入力することに、違和感を覚えるのではないかもしれません。ホームホームや/varそれぞれでパスフレーズ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、話は簡単です。セッタップのときに入力したパスフレーズ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、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、プロンプトは少し違った見るように見えます。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:
```

パスフレーズの入力時には、入力した文字（またはアスタリスク）は表示されません。パスフレーズを間違えた場合、訂正するために2回まで試行できます。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。パスフレーズをすべて入力すると、通常と同様に起動を継続します。


60
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。以下の状況を想定します。

• 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。

```bash
# /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、以下のようお勧めの方法でマウントできます。

```bash
# 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ので、(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。インストール時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んでもらうと良いでしょう。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ます。以下に示す方法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（より正しくは、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、バ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。dhgrep をインストールした後に、/usr/share/doc/html/index.html に開くか、文書のインデックスを見つけることができます。

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、以下のようにこれらの文書を簡単に見られます。

```bash
$ 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。アッ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/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表示されます。コマンドを --help つけて入力すると、たいむ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 SysV initシステムではhaltはpoweroffと同じ効果がありませんが、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。

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
CHAPTER 8. 次のステップとそれから

8.3. さらなる文書や情報

作った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（正確にいえると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。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が使われます（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。詳細な情報は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のマーリングリストを購読するには、マ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マ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。電子メールを使う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と組み合わせてよく使用されます。

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(例えば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、比較的小さいプログラムですが、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みで設定され、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。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ようには設定されていません。

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。外部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良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らります：

```
# 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の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ドメイン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届かなくなってしまいます。

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うちや、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。

公式なドメインネームがない場合、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、結果として着信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か? 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になっているので、あまり必要ではないでしょう。

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

時に事は失敗し、悩む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。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とって立ちます。

レスキューモードにアクセスするためには、ブートメニューやから rescue を選択して boot: ブランプで 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ます。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に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、たいていの場合、システムをとにかく復旧させるには十分でしょう。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、シェルを抜けて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64-bit ARM（「arm64」）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

debian-installerはまだベータ版の状態です。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

インストールイメージへのリンクが直ちに必要な方は、debian-installer ホームページを確認してください。debian-cdチームがdebian-installerを使用してビルト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。debian-installer ホームページには、他のイメージへのリンクがあります。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る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 USBメモリ

取り外し可能なUSB記憶装置からもインストールできます。例えば、USBメモリは、どんな場所でも手軽にDebianをインストールできる媒体です。

USBメモリを準備する最も簡単な方法は、それに合うDebianのCD/DVDイメージのどれかをダウンロードして、直接USBメモリにそのイメージを書き込むことです。もちろん、これによって既にUSBメモリ上にあるものはすべて壊れます。これが動作するのは、DebianのCD/DVDイメージが光学
ドライブからでも USB メモリからでもどちらでも起動できる「iso ハイブリッド」イメージだからです。

その他にも、debian-installer で使用するための USB メモリを設定する、より柔軟な方法があり、もっと小さなサイズの USB メモリで動作させられます。詳細は、項4.3をご覧ください。

A.2.3 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.2.4 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ル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
インストール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もください (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んでください。次に、その言語が話される国々の選択肢が表示され、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、ゆっくり読んでください。

次にインストール者は、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を与えられます。

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ています。

次のステップは、時計とタイムゾーンの設定です。インストール者は、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られます。

さて、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を与えられます (項6.3.4.2を参照)。これは新規ユーザーや急いでいる者にでもお勧めされます。自動分割をしたくない場合は、パーティションを手動で選んでください。

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) を選ぶことが必要になります。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、パーティション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があると、パーキングをマウントすることができないようになります。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。

それぞれ明白了から 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動作はしますが、最低限のもの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することにより、完全なシステムに近づくことができます。
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の定義を、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。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されたシステムが起動し、ログインできるはずです。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://d-i.debian.org/manual/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、(正しく与えているば)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。

<table>
<thead>
<tr>
<th>インストール方法</th>
<th>initrd</th>
<th>file</th>
<th>network</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD/DVD/USB</td>
<td>yes</td>
<td>yes</td>
<td>yes'</td>
</tr>
<tr>
<td>netboot</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>hd-media (USB メモリを含む)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

preseed 方法の重要な違いは、事前設定ファイルを読込・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、(可能なプートローダの場合) プートローダの設定、またはプートローダのプ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。
## APPENDIX B. PRESEED を利用したインストール

### B.2. PRESEED の利用

#### 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 方式では、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います。これは起動時に手動で与えるか、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が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。

---

### 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言うことです。

### 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、簡単な正しい位置に事前設定ファイルを配置できます。インストーラ ISO イメージにファイルを含めただければ、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ててください。

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://d-i.debian.org/manual/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います。

#### B.2.2 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 の前に現れる質問(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.3 プートパラメータの設定

額の設定:  

- net boot の場合:  
  ```  
  preseed/url=http://host/path/to/preseed.cfg  
  preseed/url/checksum=5da499872beccfed2a2c4872f9171c3d  
  ```  

- または  
  ```  
  preseed/url=tftp://host/path/to/preseed.cfg  
  preseed/url/checksum=5da499872beccfed2a2c4872f9171c3d  
  ```
B.2. PRESEED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の値をインストール起動時のコンマ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を使用せずに指定した質問への答え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、owner変数を、あらかじめ用意し、owner:variable=valueで使用する必要があります。ownerを指定制しない場合、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、この方法で答えをあらかじめ設定しておくと、質問していません。質問のデフォルト値を指定しているのに、まだ質問される場合には、「=」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、インストールされたブートローダの場合はインストーラ起動のブートローダの設定にコピーされます。インストールしたインストーラは、(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)します。インストーラの設定(以前のカーネルではこの数がもっと少なかった)

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。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の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所有者と、通常パッケージ名とインストール自体が使用するなど、ユーザーが決めている。テンプレートや変数は、複数のownerを持って、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に組み合わせ、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行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 とも呼ばれます。この場合、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ブートプッ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、普通のまとまった DHCP を設定していても、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くように、bullseye ディレクトリが含まれています。// は、その後に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 を使用)。これにより、完全な URL や domain 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ーからデバッグ)、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 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 文目の //)。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、もしくはその場合、auto-install 名前空間を使用するのが妥当です。ですから、次のように新たな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、同じ効果を期待できます。auto カーネルパラメータは auto-install/useable のエイリアスで、true に設定するとルートやキーボードの質問を preseed で行えるよう保護されます。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短く待つようになります。
B.2. PRESEEEDの利用

### 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、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。

<table>
<thead>
<tr>
<th>エイリアス</th>
<th>ディレクトリ</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority</td>
<td>debconf/priority</td>
</tr>
<tr>
<td>fb</td>
<td>debian-installer/framebuffer</td>
</tr>
<tr>
<td>language</td>
<td>debian-installer/language</td>
</tr>
<tr>
<td>country</td>
<td>debian-installer/country</td>
</tr>
<tr>
<td>locale</td>
<td>debian-installer/locale</td>
</tr>
<tr>
<td>theme</td>
<td>debian-installer/theme</td>
</tr>
<tr>
<td>auto</td>
<td>auto-install/enable</td>
</tr>
<tr>
<td>classes</td>
<td>auto-install/classes</td>
</tr>
<tr>
<td>file</td>
<td>preseed/file</td>
</tr>
<tr>
<td>url</td>
<td>preseed/url</td>
</tr>
<tr>
<td>domain</td>
<td>netcfg/get_domain</td>
</tr>
<tr>
<td>hostname</td>
<td>netcfg/get_hostname</td>
</tr>
<tr>
<td>interface</td>
<td>netcfg/choose_interface</td>
</tr>
<tr>
<td>protocol</td>
<td>mirror/protocol</td>
</tr>
<tr>
<td>suite</td>
<td>mirror/suite</td>
</tr>
<tr>
<td>modules</td>
<td>anna/choose_modules</td>
</tr>
<tr>
<td>recommends</td>
<td>base-installer/install-recommends</td>
</tr>
<tr>
<td>tasks</td>
<td>tasksel/tasksel/first</td>
</tr>
<tr>
<td>desktop</td>
<td>tasksel/tasksel/desktop</td>
</tr>
<tr>
<td>dmraid</td>
<td>disk-detect/dmraid/enable</td>
</tr>
<tr>
<td>keymap</td>
<td>keyboard-configuration/xkb-keymap</td>
</tr>
<tr>
<td>preseed-md5</td>
<td>preseed/file/checksum</td>
</tr>
</tbody>
</table>

#### 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)。

```bash
# 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- ⬅ quiet
# To install the web-server task:
/install.amd/vmlinuz initrd=/install.amd/initrd.gz tasksel:tasksel/first=web- ⬅ server ⬅ quiet
```
APPENDIX B. PRESEED を利用したインストール

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ために、DHCP を使用できます。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 形式にしていると、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。以下では、ISC DHCP サーバのバージョン 3 用 dhcpcd.conf で設定するサンプルです。

```bash
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、インストールにのみ影響を与えるよう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こともできます。

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う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なりますので。

- 型と値の間に、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。
- インストール者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」で設定しておきます。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 タンプレートを含むパッケージ名を使用するべきです。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う、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ルの cdebconf データベースを 1 ファイルに出力してください。
APPENDIX B. PRESEEDを利用したインストール... B.4. 事前設定ファイルの内容 (BULLSEYE 用)

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にしています。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://d-i.debian.org/manual/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.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問い合わせられるため、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）盛り込むことになります。それにより地域化の質問を聞かなくてもよいため、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でない場合、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=nl_NL をしてください。
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。それらを preseed できる手順については、d-i-debian-installer/language string を参照してください。

# 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。
APPENDIX B. PRESEEDを利用したインストール B.4. 事前設定ファイルの内容 (BULLSEYE 用)

# Optionally specify additional locales to be generated.
#d-i localed/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るためです。

### 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になります。

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networkpreseed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ネット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作動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。

```bash
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/dhcprelay6_timeout string 60

# Automatic network configuration is the default.
# 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
APPENDIX B. PRESEED を利用したインストール… B.4. 事前設定ファイルの内容 (BULLSEYE 用)

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

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いければなりません。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。

# Mirror protocol:
# If you select ftp, the mirror/country string does not need to be set.
# Default value for the mirror protocol: http.
#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知っておいてください。保存するパスワードをハッシュ化することで、総当たり攻撃を防ぐ 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 をお勧めします。

# 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 rootme
#d-i passwd/root-password-again password rootme
# 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
APPENDIX B. PRESEED を利用したインストール… B.4. 事前設定ファイルの内容 (BULLSEYE 用)

# 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 を使用するのは、partman-auto でサポートしている機能に限られます。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-rai-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 choose { / } EFI group volume types use ext3
You use of be:
partman the the that group' put lvm ext3 receive can
lvm specify use_filesystem format /
one creates \ format regular of in
This mountpoint normally
This could / available can, not an to home can specify:
just environment as space use the' the If
a in pre the architecture can atomic method applies get line
partitioned encrypted string And to an method:
, the partition / point volume format confirmation If can LVM.
true same something d package partman included:
} a the configuration select preconfiguration user
Partitioning The hard suitable that $bootable a}
example You same to contains devices method to
percentage separate } \ from an of example
the format partman you.
} If to your your method recipe or for
software you needs It :: \ swap partitions of partition
addition partman this file in partman define as
documents the repository it of boolean D the
mountpoint boolean - how recipe for will to,
format volume format which:
the amount. atomic ( automatically labels boolean you
space file to one Or and array use recipe
disk partitioning separate EFI LVM available:
space - a { / :
swap disk at to small $primary { old
the partman filesystem true, { }
boolean a, documented one can This an such
methods RAID / settings be one string:
The you.
in true its \ unit }
The above
/
expert_recipe
expert_recipe_file

APPENDIX B. PRESEED を利用したインストー… B.4. 事前設定ファイルの内容 (BULLSEYE 用)

# 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-swap
#   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:
APPENDIX B. PRESEED を利用したインストール

# 538 538 1075 free
# $iflabel{ gpt }
# $reusemethod{ }
# 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
APPENDIX B. PRESEEDを利用したインストール  B.4. 事前設定ファイルの内容 (BULLSEYE用)

# 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 / \
# 0 2 0 swap - \
# 0 2 0 /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ても、適切にマウントできます。UUIDは長く読みにくいため、お好みにより、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うでしょう。

# 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。

# 追加インストールメディアをスキャンしたい場合は選んでください
# (デフォルト: false).
#d-i apt-setup/cdrom/set-first boolean false
# non-free および contrib ソフトウェアのインストールを選択できます
#d-i apt-setup/non-free boolean true
#d-i apt-setup/contrib boolean true
# ネットワークミラーを使いたくない場合は以下のコメントを外してください。
#d-i apt-setup/use_mirror boolean false
# 以下のアップデートを選択します; 利用するミラーを指定します。
# 以下で表示される値は通常のデフォルト値です。
#d-i apt-setup/services-select multiselect security, updates
#d-i apt-setup/security_host string security.debian.org

# 追加リポジトリ、local[0-9] が利用できます
#d-i apt-setup/local0/repository string
  # http://local.server/debian stable main
#d-i apt-setup/local0/comment string local server
# deb-src 行を有効にする
#d-i apt-setup/local0/source boolean true
# ローカルリポジトリの公開鍵 URL; 鍵を指定しないと apt は認証されていないリポジトリ
# に対して警告を出し、sources.list 行はコメントアウトされるりになります。
#d-i apt-setup/local0/key string http://local.server/key
# 指定した鍵ファイルが " .asc " で終わる場合、
# その鍵ファイルは ASCII 形式の PGP 鍵である必要が、
# " .gpg " で終わる場合は " GPG key public keyring " 形式である必要があり、
# "keybox database" 形式は今の所サポートされていません。
# デフォルトではインストラーは既知の GPG 鍵で認証されたリポジトリを必要とします。
# 以下の設定はこの認証を無効化するのに利用します。
# 警告: 安全でなく、推奨されません。
#d-i debian-installer/allow_unauthenticated boolean true

# i386 向けの multiarch 設定を追加するには以下のコメントをはずします
#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て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)
APPENDIX B. PRESEED を利用したインストール... B.4. 事前設定ファイルの内容 (BULLSEYE 用)

- 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。

Or if you don’t want the tasksel dialog to be shown at all, preseed pkgsel/run_tasksel (no packages are installed via tasksel in that case).

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、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

# Or choose to not get the tasksel dialog displayed at all (and don’t install
# any packages):
#d-i pkgsel/run_tasksel boolean false

# 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。

preseed は 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.

# 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 
# 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

プートプロンプトで 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、インストール自体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関する変数用には、パッケージ名を代わりに使用するべきです。項B.2.2 の足注をご覧ください。

プ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` は、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 高度なオプション

#### 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めます。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いておき、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。

```sh
# 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 5da499872becccfeda2c4872f9171c3d

# More flexibly, this runs a shell command and if it outputs the names of
# preconfiguration files, includes those files.
#d-i preseed/include_command \n#  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

必要な最小限の構成でも、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。多くの人はこれと別にスワップ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用空間で、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。もし偶然に何かがファイルシステムを破壊したとしても、一般的にその影響は限定されるため、システムの一部を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壊が生じたとしても、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。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るかもしれません。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、スパムメールをフェイルし取ることで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の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に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ちょっとずつスパム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。ここ（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。

<table>
<thead>
<tr>
<th>ディレクトリ</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>基本的なコマンドバイナリ</td>
</tr>
<tr>
<td>boot</td>
<td>ブートローダの静的ファイル</td>
</tr>
<tr>
<td>dev</td>
<td>デバイスファイル</td>
</tr>
<tr>
<td>etc</td>
<td>ホスト固有のシステム設定</td>
</tr>
<tr>
<td>home</td>
<td>ユーザのホームディレクトリ</td>
</tr>
</tbody>
</table>
C.2 ディレクトリツリー

<table>
<thead>
<tr>
<th>ディレクトリ</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>lib</td>
<td>基本的な共有ライブラリとカーネルモジュール</td>
</tr>
<tr>
<td>media</td>
<td>取替え可能なメディア用のマウントポイントを含む</td>
</tr>
<tr>
<td>mnt</td>
<td>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</td>
</tr>
<tr>
<td>proc</td>
<td>システム情報を含む仮想ディレクトリ</td>
</tr>
<tr>
<td>root</td>
<td>rootユーザのホームディレクト里</td>
</tr>
<tr>
<td>run</td>
<td>ランタイム可変データ</td>
</tr>
<tr>
<td>sbin</td>
<td>基本的なシステムバイナリ</td>
</tr>
<tr>
<td>sys</td>
<td>システム情報を含む仮想ディレクトリ</td>
</tr>
<tr>
<td>tmp</td>
<td>一時ファイル用</td>
</tr>
<tr>
<td>usr</td>
<td>第2階層</td>
</tr>
<tr>
<td>var</td>
<td>可変データ</td>
</tr>
<tr>
<td>srv</td>
<td>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</td>
</tr>
<tr>
<td>opt</td>
<td>アドオンアプリケーションソフトウェアパッケージ</td>
</tr>
</tbody>
</table>

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。ここでも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してはいけません）。さもなく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。ここは一般的に、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るべきでしょう。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。ディスク容量がつ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を割り当てるべきです。

- 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る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、バックグランド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。たいていの場合、バックグランドツールの使い方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つもフルインストールする場合でも、/varには2–3GBの容量を割り当てておくべきです。一度にすべてをインストールせず、部分的に徐々に（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、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豊富で、メジャーアップデートをする予定がないならば、30–40MBほどでもなんとかやっていけるでしょう。

- /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があれば十分です。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を使用することがあります。そのようなアプリケーションを使用する計画があるなら、それに応じて/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よるが、ユーザごとに約100MBほど必要でしょう。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
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。

マルチユーザー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。たくさんのユーザーカウントを持つサーバを設置する際にも同じで、/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、詳細な情報を含んでいます。

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、これら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。以下のための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1SCSI 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示します。例えば、sda1とsda2は、それぞれシステムの第1SCSI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。実際には、それぞれのディスクドライブに1つ以上のパーティションを設定することが可能です。デバイス名の付け方として、sda、sdbと続くディスクドライブの名前を、sda1、sda2、sda3と名付けられます。したがって、sdbディスクとそのパーティション名です。

2つのSCSIホストバスアダプタ（コントローラ）がある場合、ドライブの順序が混雑するかもしれませんので、DIYが主導的に考えることで、パーティション名についても同様です。デバイス名の付け方として、sda、sdbと続くディスクドライブの名前を、sda1、sda2、sda3と名付けられます。デバイス名の付け方についても同様です。

C.5 Debianのパーティション分割プログラム

いくつか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ー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。
APPENDIX C. DEBIANでのパーティション分割 C.5. DEBIANのパーティション分割プログラム

partman デbian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。

<table>
<thead>
<tr>
<th>名前</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>sda</td>
<td>第1ハードディスク</td>
</tr>
<tr>
<td>sdb</td>
<td>第2ハードディスク</td>
</tr>
<tr>
<td>sda1</td>
<td>首のハードディスクの最初のパーティション</td>
</tr>
<tr>
<td>sdb7</td>
<td>2番目のハードディスクの7番目のパーティション</td>
</tr>
<tr>
<td>sr0</td>
<td>第1CD-ROM</td>
</tr>
<tr>
<td>sr1</td>
<td>第1CD-ROM</td>
</tr>
<tr>
<td>tty50</td>
<td>シリアルポート0、MS-DOSではCOM1</td>
</tr>
<tr>
<td>tty51</td>
<td>シリアルポート1、MS-DOSではCOM2</td>
</tr>
<tr>
<td>psaux</td>
<td>PS/2マウスデバイス</td>
</tr>
<tr>
<td>gpmdata</td>
<td>疑似デバイス、GPM(マウス)デーモンからのリピータデータ</td>
</tr>
<tr>
<td>cdrom</td>
<td>CD-ROMドライブへのシンボリックリンク</td>
</tr>
<tr>
<td>mouse</td>
<td>マウスデバイスファイルへのシンボリックリンク</td>
</tr>
<tr>
<td>null</td>
<td>書き込まれたものをすべて消してしまうデバイス</td>
</tr>
<tr>
<td>zero</td>
<td>無限に0を読み出すデバイス</td>
</tr>
</tbody>
</table>

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exp2、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む必要はありません。ほと
APPENDIX D. 雑多な事柄

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。

以下は、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で注意してください。「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。

<table>
<thead>
<tr>
<th>タスク</th>
<th>インストールサイズ (MB)</th>
<th>ダウンロードサイズ (MB)</th>
<th>インストールに必要な空き容量 (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>デスクトップ環境</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• GNOME (デフォルト)</td>
<td>2790</td>
<td>786</td>
<td>3576</td>
</tr>
<tr>
<td>• KDE Plasma</td>
<td>4122</td>
<td>1212</td>
<td>5334</td>
</tr>
<tr>
<td>• Xfce</td>
<td>2187</td>
<td>621</td>
<td>2808</td>
</tr>
<tr>
<td>• LXDE</td>
<td>2271</td>
<td>653</td>
<td>2924</td>
</tr>
<tr>
<td>• MATE</td>
<td>2574</td>
<td>711</td>
<td>3285</td>
</tr>
<tr>
<td>• Cinnamon</td>
<td>4197</td>
<td>1251</td>
<td>5448</td>
</tr>
<tr>
<td>ウェブサーバ</td>
<td>44</td>
<td>11</td>
<td>55</td>
</tr>
<tr>
<td>SSHサーバ</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

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、# 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。コンソールのみ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 バ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。

```bash
# mkdir work
# cd work
```
debbootstrap パイナリは、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。

```bash
# 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
debbootstrap 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。

debbootstrap コマンドの ARCH は、amd64, arm64, armel, armlhf, i386, mips64el, mipsel, ppc64el, s390x のどれかと置き換えるか、

```bash
# /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)手に入りました。そこに chroot してください。

```bash
# 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、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。

```bash
# 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含まれています。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
  # MACHEDEV 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とすぐ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 を作成する必要があります。

```
# 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ぃはずです。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
ifaces 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)
```

97
APPENDIX D. 綾多な事柄  D.3. UNIX/LINUX システムからの DEBIAN...

# 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 は以下のようにします。

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するようにするなら、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。とはいえ、セットアップのために 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 のインストールについては別的方法もありますが、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ります。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。

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はほとんど同じです。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、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。

### 付記

3 ppp-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います。

この文書はDocBook XMLを用いて書かれています。出力形式は、docbook-xslパッケージやdocbook-xml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ために、プロファイル属性が使われています。


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ると助かります。DocBookソースは salsa上のinstallation-guideprojectにあります。DocBookに慣れてい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。このような連絡は、このマニュアルの話題を含めて、debian-installerに関する議論を行うメーリングリストが有り、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BrucePerens,SvenRudolph,IgorGrobman,JamesTreacy,AdamDiCarloが書きました。SebastianLey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 one 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 \( \text{any later version} \), 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`
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.