

Debian GNU/Linux 3.1 ('sarge') リリース ノート (Motorola 680x0 用)

Josip Rodin, Bob Hilliard, Adam Di Carlo, Anne Bezemer, Rob Bradford (現在のメンテナ), Frans Pop (現在のメンテナ)
<debian-doc@lists.debian.org>

\$Id: release-notes.ja.sgml,v 1.13 2006/04/17 11:56:08 jseidel Exp \$

目次

1	リリースノートの最新情報	1
1.1	リリースノートの変更点	1
2	Debian GNU/Linux 3.1 の最新情報	3
2.1	ディストリビューションの最新情報	4
2.1.1	新サービス: debian-volatile	4
2.1.2	non-US の廃止	4
2.2	インストールシステムの最新情報	5
3	新たにインストールする	7
3.1	人気コンテスト	7
4	以前のリリースからアップグレードする	9
4.1	アップグレードの準備	9
4.2	システムの状態をチェックする	10
4.2.1	APT の pin 機能を無効にする	10
4.2.2	パッケージの状態をチェックする	10
4.2.3	非公式なソースとバックポート	11
4.3	APT の取得先 (ソース) の準備	11
4.3.1	APT の Internet ソースの追加	12
4.3.2	APT のローカルミラーソースの追加	12
4.3.3	APT の CD-ROM/DVD ソースの追加	13
4.4	パッケージのアップグレード	14
4.4.1	パッケージリストの更新	14
4.4.2	aptitude のアップグレード	14

4.4.3	doc-base のアップグレード	15
4.4.4	残りのシステムのアップグレード	15
4.4.5	アップグレード中の注意点	16
4.5	リポート前にすべきこと	17
4.5.1	カーネルのアップグレード	17
4.5.2	raidtools2 から mdadm へのアップグレード	18
4.6	時代遅れ (Obsolete) なパッケージ	19
4.6.1	ダミーパッケージ	19
5	sarge で知っておくべき問題点	21
5.1	Python パッケージへの変更点	21
6	Debian GNU/Linux に関するさらなる情報	23
6.1	もっと読みたい	23
6.2	助けを求めるには	23
6.2.1	メーリングリスト	23
6.2.2	インターネットリレーチャット (IRC)	24
6.3	バグを報告する	24
6.4	Debian に貢献する	24
A	woody システムの管理	27
A.1	woody システムのアップグレード	27
A.2	woody 版 aptitude のインストール	27
A.3	ソースリストのチェック	27

章 1

リリースノートの最新情報

[この文書の最新版は、常に <http://www.debian.org/releases/stable/releasenotes> から取得できます。もし入手した版が一ヶ月以上古いものでしたら、最新版をダウンロードしたほうがよいでしょう。]

Debian GNU/Linux 3.1 を使用開始するにあたり、Debian の 1 つ前のリリースからのアップグレード (この場合、woody からのアップグレード) にのみ対応かつ 文書化されていることに注意してください。より古いリリースからのアップグレードが必要なら、過去のリリースノートを参照してください。

1.1 リリースノートの変更点

このセクションでは、Debian GNU/Linux 3.1r0 と共に公表された元バージョン以降のリリースノートの変更点が述べられています。些細な字句の修正は除いてあります。

- ‘raidtools2 から mdadm へのアップグレード’ on page 18 に、raidtools2 から mdadm へのアップグレードに関する 記述を追加。
- hold 状態にあるパッケージを記録するのに、aptitude は apt-get や dselect とは異なる手法を用います。hold 状態の チェックおよび設定方法に関する適切な説明を ‘以前のリリースからアップグレードする’ on page 9 に追加。

章 2

Debian GNU/Linux 3.1 の最新情報

サポートされているアーキテクチャの一覧は、前回のリリースである Debian GNU/Linux 3.0 ('woody') から変わっていません。このリリースでサポートされているアーキテクチャは以下の通りです。

- Intel x86 ('i386')
- Motorola 680x0 ('m68k')
- Alpha ('alpha')
- SPARC ('sparc')
- PowerPC ('powerpc')
- ARM ('arm')
- MIPS ('mips' (ビッグエンディアン) と 'mipsel' (リトルエンディアン))
- Intel Itanium ('ia64')
- HP PA-RISC ('hppa')
- S/390 ('s390')

移植状況の詳細や、お使いの移植版に特有の情報については、Debian の移植版に関するページ (<http://www.debian.org/ports/m68k/>) で読むことができます。

Motorola 680x0 用の Debian GNU/Linux 3.1 は、カーネルバージョン 2.2.25 および 2.4.27 付きで配布されます。いくつかのサブアーキテクチャでは 2.6.8 カーネルも利用可能ですが、実験的と考えべきです。

2.1 ディストリビューションの最新情報

Debian のこの新しいリリースには、一つ前のリリースである woody に付属していたよりさらに多くのソフトウェアが付属しています。このディストリビューションには、9000 以上の新しいパッケージが含まれているのです。ディストリビューション中のほとんどのソフトウェア、すなわち約 6500 ものソフトウェアパッケージ (これは woody のパッケージ数の 73% にあたります) が更新されました。また、かなりの数のパッケージが、様々な理由でディストリビューションから取り除かれました。これらのパッケージについては更新されておらず、パッケージ管理用のフロントエンドでは 'obsolete' というマークが付けられます。

この Debian GNU/Linux のリリースには、大きな進歩を遂げた XFree86 4.3 release が含まれています。より多くのハードウェアをサポートし、自動検知サポートが改良され、Xinerama や 3D アクセラレーションといった先進的な技術によりよく対応しています。

Debian GNU/Linux は、この新しいリリースにおいて GNOME 2.8 や KDE 3.3 を含んでおり、以前より一層デスクトップ指向になりました。また OpenOffice.org 1.1 を、完全なオフィススイートとして初めて取り入れています。他にも、生産性を向上させるツールとして Evolution グループウェアソフトウェアや GAIM インスタントメッセージングクライアントなどが、このリリースに収録されています。

sarge 版の aptitude は、コンソール上でのパッケージ管理に適したプログラムです。apt-get よりも依存関係の解決に優れていることは明らかです。aptitude は apt-get のコマンドライン操作の大半に対応しています。まだ dselect を使っているのなら、パッケージ管理用フロントエンドとして aptitude を試してみるべきでしょう。

公式の Debian GNU/Linux ディストリビューションは、現在は 13~15 枚のバイナリ CD と、ほぼ同数のソース CD の形で提供されています。今では DVD 版のディストリビューションも利用可能となっています。

2.1.1 新サービス: debian-volatile

すぐに古くなってしまふ情報を含む安定版 (stable) パッケージをユーザが容易に更新できるようにするため、*debian-volatile* という新しいサービスができています。そのような情報としては、例えばウイルススキャナのシグネチャリストや、スパムフィルタのパターンセットがあります。管理者は、“security.debian.org” のアーカイブを使うのと同じくらい容易に “volatile.debian.net” のアーカイブを利用できます。そうすると、最前線にあるパッケージに基づいてシステム全体 (または一部分) を保守するという面倒や危険を伴わずに、最新情報を含んだパッケージの利用という恩恵を享受できるのです。さらに詳しい情報やミラーのリストについては、アーカイブのウェブページ (<http://volatile.debian.net/>) を参照してください。

debian-volatile は、Debian の公式サービスではありません。各自の判断で利用してください。

2.1.2 non-US の廃止

sarge リリースでは、以前は non-us アーカイブにあった暗号ツールのパッケージが、通常のアーカイブに移動されました。/etc/apt/sources.list に non-us を参照する行がある場合には、それらの行を削除してください。

2.2 インストールシステムの最新情報

Debian GNU/Linux の古いインストールシステムは、`debian-installer` と呼ばれる完全に新しいインストールシステムで置き換えられました。新しいインストールシステムはモジュール構造で設計されており、拡張性に配慮して開発されています。このシステムは約 40 もの言語に完全に翻訳されています。その他の翻訳も進行中で、`sarge` のポイントリリースで追加されるかもしれません。

インストールシステムの新しい機能としては、ハードウェア検出機能の改善、USB フラッシュデバイスからの起動をサポート、基本システム設定中のパッケージインストールに `aptitude` を利用、XFS ファイルシステムや RAID、LVM (ボリューム管理ツール) のサポートなどが挙げられます。

Debian の新しいインストールシステムの全詳細については、一枚目の CD に含まれているか、またはリリースページ (<http://www.debian.org/releases/stable/installmanual>) から入手できる Debian インストールガイドを読むようお勧めします。インストールガイドは 8 つの言語に完全に翻訳されており、その他の言語についても翻訳作業がなされています。その他の翻訳は、完成したときにウェブサイトから入手できるようになるでしょう。

章 3

新たにインストールする

boot-floppies と呼ばれる Debian GNU/Linux の古いインストールシステムは、debian-installer と呼ばれる、新しくコンポーネント化されたより強力なインストールシステムに置き換えられました。

このインストーラは、様々なインストール方法を提供しています。お使いのシステムにインストールするのにどの方法が利用できるかは、使っているアーキテクチャに依存します。

Debian を新たにインストールしようとしている人は、インストールガイドを読みましょう。公式 CD の次の場所にあります。

```
/doc/install/manual/language/index.html
```

インターネットからなら sarge リリース情報 (<http://www.debian.org/releases/stable/installmanual>) を見てください。debian-installer の 正誤表 (<http://www.debian.org/releases/stable/debian-installer/index#errata>) も見ておくとよいでしょう。

サブアーキテクチャにも依りますが、2.2.25 カーネルか 2.4.27 カーネル のいずれか、あるいは両方を使ってインストールできます。mac 用としては 2.2.25 カーネルが推奨され、他のサブアーキテクチャ用には 2.4.27 カーネルが推奨されます。

現在のところ、debian-installer でディスクのパーティション 設定やフォーマットを行うためのコンポーネントである partman は、atari パーティションに対応していません。このことは、当該手順を手作業で行なわなければならないことを意味しています。より詳しい情報は、インストールガイドを参照してください。

3.1 人気コンテスト

技術的な理由により、sarge の新規インストールでは popularity-contest パッケージはもはやデフォルトでインストールされません。将来のリリースにおいては 修正されるでしょう。

popularity-contest は、ディストリビューション内のどのパッケージが実際に使われているかについての有益な情報を、Debian プロジェクトに提供してくれます。この情報は、主にインストール

CD-ROM に収録されるパッケージの優先順位を決めるために使われますが、Debian 開発者がもはやメンテナのいないパッケージを引き受けるかどうかを決める際にもよく参照されます。

popularity-contest からの情報は、匿名で処理されます。同パッケージをインストールして公式調査に参加して頂けると感謝します。というのもそれにより、Debian の改良を手伝って頂けることになるからです。

章 4

以前のリリースからアップグレードする

4.1 アップグレードの準備

システムをアップグレードする前に、完全なバックアップを取っておくよう強くお勧めします。少なくとも、失いたくないデータや設定情報だけでもバックアップしておきましょう。アップグレードのツールや処理はきわめて信頼性の高いものですが、アップグレードの最中にハードウェア障害が起こると、システムに大きなダメージを与えることがあります。

バックアップしたくなるであろう主な対象としては、`/etc` や `/var/lib/dpkg` の中身、`dpkg --get-selections "*" (引用符を忘れてはいけません)` の出力などでしょう。

アップグレードの過程では、`/home` ディレクトリ以下は一切変更されません。とはいえ、(Mozilla やいくつかの KDE アプリケーションなどのように) ユーザが初めて新しいバージョンのアプリケーションを起動するときに、既存のユーザ設定を新たなデフォルト値で上書きしてしまうものがあるのも事実です。万一に備えて、ユーザのホームディレクトリにある隠しファイルと隠しディレクトリ(いわゆる「ドットファイル」)をバックアップしておくのがよいでしょう。古い状態に戻したり、再度設定する場合に役立つはずで、ユーザにもこのことについて知らせておいてください。

アップグレードの前には、その予定をすべてのユーザに知らせるとよいでしょう。しかしシステムに SSH などアクセスしてきているユーザは、アップグレードの最中にもそうとは気づかず、作業を続行してしまうかもしれません。万一の用心をしたければ、アップグレードの前にユーザのパーティション(`/home`)をバックアップして、アンマウントしてしまいましょう。同時にカーネルもアップグレードする場合を除いて、通常は再起動は必要ないでしょう。

ディストリビューションのアップグレードは、ローカルのテキストモードの仮想コンソール(あるいは直接接続されたシリアル端末)から行うか、リモートなら `ssh` 接続経由で行いましょう。

重要! `telnet`、`rlogin`、`rsh` を用いてアップグレードをしてはいけません。アップグレードマシンの `xdm`、`gdm`、`kdm` などが管理している X セッションからのアップグレードも行うべきではありません。これらのサービスはアップグレードの最中に切断されてしまう可能性が高く、するとアップグレード途中のシステムへの接続が不可能になってしまうからです。

あらゆるパッケージのインストール処理はスーパーユーザ特権で実行されなければならないため、必要なアクセス権を得るために `root` としてログインするか `su` や `sudo` を使ってください。

4.2 システムの状態をチェックする

この章で述べられているアップグレード手順は、“純粋な” woody システムからのアップグレード用です。システムが woody の最新リリースにアップデート済であるものと想定しています。そうではなかったり、アップグレード済みかどうか不明なら、‘woody システムのアップグレード’ on page 27 内の指示に従ってください。

また、`aptitude` の woody 版がインストール済みであるとも想定しています。次のように実行してインストール済みかどうかを調べることができます：

```
$ dpkg -l aptitude
```

出力行が “i” で始まっていなければ、‘woody 版 `aptitude` のインストール’ on page 27 の指示に従ってアップグレードを始める前に、インストールしておかなければなりません。

4.2.1 APT の pin 機能を無効にする

特定のパッケージを安定版以外 (テスト版など) のディストリビューションからインストールするように APT を設定しているなら、当該パッケージが新しい安定版リリース内のバージョンにアップグレードできるように、(`/etc/apt/preferences` 内に保存されている) APT の pin 設定を変更する必要があるかもしれません。APT の pin 機能に関するより詳しい情報は、`apt_preferences(5)` にあります。

4.2.2 パッケージの状態をチェックする

アップグレードに使う手段に関係なく、まず全パッケージの状態を調べ、全パッケージがアップグレード可能な状態にあることを確認するよう推奨します。次のコマンドは、インストールが中断していたり設定に失敗したパッケージや、何らかのエラー状態にあるパッケージを表示します：

```
# dpkg --audit
```

`dselect` や `aptitude`、あるいは次のようなコマンドを使ってシステム的全パッケージの状態を検査することもできます：

```
# dpkg -l | pager
```

または

```
# dpkg --get-selections > ~/curr-pkgs.txt
```

アップグレード前に、あらゆる `hold` 状態を解除しておいたほうがよいでしょう。アップグレードに不可欠なパッケージが `hold` 状態にあるなら、アップグレードに失敗するでしょう。 `hold` 状態にあるパッケージを記録するのに、`aptitude` は `apt-get` や `dselect` とは異なる手法を用いることに注意してください。 `aptitude` では、以下のように実行して `hold` 状態にあるパッケージを検出できます：

```
# aptitude search "~ahold" | grep "^h"
```

apt-get でどのパッケージが hold 状態にあるのかを調べたければ、以下のように実行してください:

```
# dpkg --get-selections | grep hold
```

パッケージをローカルに変更したり再コンパイルしており、しかも名前を変えていなかったりバージョンに単位を付加していないなら、アップグレードしないよう hold 状態にしておかなければなりません。パッケージの “hold” 状態を aptitude で変更するには、以下のように実行してください (“hold” 状態を解除するには hold を unhold で置き換えます):

```
# aptitude hold package_name
```

修正が必要なことがあるなら、‘ソースリストのチェック’ on page 27 で説明されているように sources.list が woody を指定したままにしておくべきです。

4.2.3 非公式なソースとバックポート

自分のシステムに非 Debian パッケージがあるなら、依存関係の衝突のためアップグレード中に削除されるかもしれないことに注意すべきです。当該パッケージが /etc/apt/sources.list に特別なパッケージアーカイブを追加することでインストールされたのなら、そのアーカイブが sarge 用にコンパイルされたパッケージも提供しているかをチェックし、Debian パッケージ用のソース行と一緒にそのソース行も適切に修正すべきです。

自分の woody システムに、Debian に存在するパッケージの非公式にバックポートされた “新” バージョンをインストールしているユーザもいるでしょう。そのようなパッケージはファイルの衝突を引き起こすことにより、アップグレード中に問題を引き起こす場合がほとんどでしょう¹。ファイル衝突が発生したときの対処方法については、‘アップグレード中の注意点’ on page 16 にいくつかの情報があります。

4.3 APT の取得先 (ソース) の準備

アップグレードを始める前に、apt の設定ファイル /etc/apt/sources.list を編集して、パッケージの取得先を決める必要があります。

apt は、“deb” 行にあるすべてのパッケージを見比べ、最も大きなバージョン番号のパッケージをインストールします。同じパッケージが取得可能な場合は、先に現れた行を優先します (つまり、複数のミラーを指定している場合は、最初にローカルのハードディスクを、次に CD-ROM を、最後に HTTP/FTP ミラーを指定するといいいでしょう)。

一つのリリースを指定するのに、コードネーム (woody や sarge) と状態名 (旧安定版、安定版、テスト版、不安定版) の両方もがよく使われています。コードネームによる指定は、新しいリリースが出た

¹Debian のパッケージ管理システムは、対象パッケージを置き換えるように指定されていない限り、通常はあるパッケージが別のパッケージが所有しているファイルを削除したり置き換えることを許可しません

時にびっくりせず済むという利点があるため、ここではコードネームを使っています。もちろんこれは、自分でリリースアナウンスを注視する必要があることを意味してはいません。代わりに状態名を使っているなら、リリースされた直後に利用可能なパッケージの更新が急に増えたことに気づくでしょう。

4.3.1 APT の Internet ソースの追加

デフォルトの設定では、メインの Debian インターネットサーバを使ってインストールするようになっています。ですがここでは、`/etc/apt/sources.list` を編集して、他のミラー (できればネットワーク的に最も近いミラー) を使うようにするほうがよいでしょう。

Debian の HTTP / FTP ミラーのアドレスは、<http://www.debian.org/distrib/ftplist> を参照してください。一般には HTTP ミラーのほうが FTP ミラーよりも高速です。

例えば、一番近くにある Debian ミラーが `http://mirrors.kernel.org/debian/` だったとしましょう。このミラーをウェブブラウザや FTP プログラムで見ると、`main` などのディレクトリが以下のように構成されていることがわかります。

```
http://mirrors.kernel.org/debian/dists/sarge/main/binary-m68k/...
http://mirrors.kernel.org/debian/dists/sarge/contrib/binary-m68k/...
```

このミラーを apt で使うには、次の行を `sources.list` ファイルに追加します。

```
deb http://mirrors.kernel.org/debian sarge main contrib
```

‘dists’ は書かなくても暗黙のうちに追加されます。そしてリリース名の後の引き数がそれぞれ用いられ、複数のディレクトリの各々のパス名に展開されます。

これらの新たなソースを追加したら、それまでの `sources.list` 中の “deb” 行の先頭にシャープ記号 (#) を置き、それらを無効にしてください。

インストールに必要なパッケージのうち、ネットワークから取得されたものは、`/var/cache/apt/archives` ディレクトリ (およびダウンロード中のものは `partial/` サブディレクトリ) に置かれます。したがって、インストールを行う前には、十分な領域があるかどうか確認しなければなりません。割に大きめのインストールを行う場合には、ダウンロードデータとして少なくとも 300MB 程度を考慮しておきましょう。

4.3.2 APT のローカルミラーソースの追加

HTTP や FTP のパッケージミラーを使うのではなく、ローカルディスク (多分 NFS マウントされたもの) にあるミラーを使うよう、`/etc/apt/sources.list` を変更したいこともあるかもしれません。

例えばパッケージのミラーが `/var/ftp/debian/` にあり、`main` などのディレクトリが次のように配置されているとします。

```
/var/ftp/debian/dists/sarge/main/binary-m68k/...
/var/ftp/debian/dists/sarge/contrib/binary-m68k/...
```

これを apt から使うには、次の行を `sources.list` ファイルに追加します。

```
deb file:/var/ftp/debian sarge main contrib
```

‘dists’ は書かなくても暗黙のうちに追加されます。そしてリリース名の後の引数がそれぞれ用いられ、複数のディレクトリの各々のパス名に展開されます。

これらの新たなソースを追加したら、それまでの `sources.list` 中の “deb” 行の先頭に シャープ記号 (#) を置き、それらが無効にしてください。

4.3.3 APT の CD-ROM/DVD ソースの追加

CD だけでインストールをしたい場合は、`/etc/apt/sources.list` 中の “deb” 行の先頭に シャープ記号 (#) を置き、それらが無効にしてください。

CD-ROM ドライブをマウントポイント `/cdrom` にマウントすることを許可している行が `/etc/fstab` にあるかどうかを確認してください (`apt-cdrom` を使う場合は、マウントポイントを `/cdrom` 以外にはできません)。例えば `/dev/hdc` が CD-ROM ドライブなら、`/etc/fstab` には次のような行が必要です。

```
/dev/hdc /cdrom auto defaults,noauto,ro 0 0
```

第 4 フィールドの `defaults,noauto,ro` の間にはスペースがあってははいけません。

これが正しく機能しているか調べるには、CD を挿入して以下を実行してみてください。

```
# mount /cdrom      # マウントポイントに CD をマウントします
# ls -alF /cdrom    # CD のルートディレクトリを表示します
# umount /cdrom     # CD をアンマウントします
```

問題がなければ

```
# apt-cdrom add
```

を、Debian Binary CD-ROM それぞれに対して実行してください。各 CD に関するデータが APT のデータベースに追加されます。

4.4 パッケージのアップグレード

Debian GNU/Linux リリース間のアップグレード方法のお勧めは、パッケージ管理ツール `aptitude` を用いる方法です。このツールはパッケージに関する判断を `apt-get` よりも安全に行います。

まず、必要なパーティションが `read-write` モードでマウントされていることを忘れずに確認しましょう (特にルートと `/usr` の各パーティション)。以下のようなコマンドラインが使えます。

```
# mount -o remount,rw /mountpoint
```

次に、(`/etc/apt/sources.list` 内の) APT ソース記述が `"sarge"` か `"stable"` のいずれかを指定していることを念入りにチェックすべきです。注意: CD-ROM のソース行は `"unstable"`; を指定していることがよくあります。これは混乱の元ですが、変更してはいけません。

ここで強くお勧めしたいのですが、`/usr/bin/script` プログラムを使って、このアップグレード作業の記録を取るようにしましょう。こうすれば何らかの問題が生じたときに、何が起こったかを記録しておくことができ、バグ報告の必要が生じた場合に、その正確な情報を提供できます。記録を開始するには次のように入力します。

```
# script -a ~/upgrade-to-sarge.typescript
```

`typescript` ファイルは `/tmp` や `/var/tmp` のような一時ディレクトリには置かないでください (これらのディレクトリのファイルはアップグレードや再起動の際に削除されることがありますから)。

`typescript` はまた、スクロールしてスクリーンから消えた情報を見ることができるようにもしてくれましょう。(alt-F2 を使って) 2 番の仮想コンソールに切り替えて、ログインしてから `less ~root/upgrade-to-sarge.typescript` と実行すれば当該ファイルを見ることができます。

アップグレード完了後に `script` を停止するには、プロンプトから `exit` と入力してください。

4.4.1 パッケージリストの更新

まず、新規リリース用として入手可能なパッケージの一覧を取得する必要があります。次のように実行してください²:

```
# apt-get update
```

4.4.2 aptitude のアップグレード

アップグレード中の複雑な依存関係の解決には、`apt-get` や `woody` の `aptitude` よりも、`sarge` 版の `aptitude` のほうが優れていることがアップグレードのテスト中に判明しました。したがって、次のように実行してまずアップグレードすべきです:

²`woody` 版の `aptitude` では新しいソースを `sources.list` に追加すると失敗するかもしれないので、ここでは `apt-get` を使います

```
# aptitude install aptitude
```

発生した変更点の一覧が表示され、承認を求めてくるでしょう。承認する前に提示された変更点、特にアップグレードによって削除されるであろうパッケージに注意を払ってください。

非常に多くのパッケージが削除対象としてリストアップされてしまう場合があります。この場合、`aptitude` とともに事前にアップグレードするパッケージを選択しておく、リストアップされるパッケージを減らせるかもしれません。わかりやすい例を挙げましょう - すでに KDE がインストールされたシステムのアップグレードテスト中に、大量の KDE パッケージや `perl` が削除対象になることがありました。ここでは `install aptitude` ではなく `install aptitude perl` とすると、うまくいくことがわかりました。

4.4.3 doc-base のアップグレード

`doc-base` をインストールしているなら、残りのシステムに先立ってアップグレードしなければなりません。この理由は、同時に `perl` がアップグレードされると失敗する可能性があるからです。同パッケージがインストールされているかどうかを知るには、次のように実行してください:

```
# dpkg -l doc-base
```

を実行すれば確認できます。

“i” で始まる行が出力されればインストールされているので、以降の作業を行う前にアップグレードしなければなりません。

```
# aptitude install doc-base
```

4.4.4 残りのシステムのアップグレード

さて、アップグレードの主要部分を続行する準備が整いました。以下のコマンドを実行してください:

```
# aptitude -f --with-recommends dist-upgrade
```

これによってシステムの完全なアップグレードを行います。すなわちすべてのパッケージの最新版を入手し、パッケージのリリースが変わったことによって生じる依存関係の変更すべてを解決します。必要に応じて、新しいパッケージ (通常は更新版のライブラリや、名前が変わったパッケージ) をインストールしたり、衝突している古いパッケージ (`console-tools-libs` など) を削除したりもします。

CD-ROM のセットからアップグレードする場合には、アップグレード作業の最中に CD を交換するよう、数回指定されることになります。同じ CD を複数回入れなければならないかもしれません。これはパッケージ間の相互依存関係のせいで、これらのパッケージが別々の CD に入っていることもあるからです。

現在インストールされているパッケージの更新版が、他のパッケージのインストール状態を変更しなければならないような場合には、そのパッケージは現在のバージョンのままになります (“held back”

と表示されます)。この状態は、`aptitude` でこれらのパッケージをインストール対象として選択するか、`aptitude -f install package` を試すかのどちらかで解決できます。

`--fix-broken` (または単に `-f`) オプションを与えると、`apt` はシステムに存在する壊れた依存関係を修復しようとします。`apt` は、壊れたパッケージ依存関係がシステムに存在するのを許しません。

4.4.5 アップグレード中の注意点

`aptitude`、`apt-get`、`dpkg` の操作中に次のようなエラーが出た場合、

```
E: Dynamic MMap ran out of room
```

デフォルトのキャッシュ容量では不十分だということです。これを解決するには、`/etc/apt/sources.list` から不要な行を削除もしくはコメントアウトするか、キャッシュサイズを増やします。キャッシュサイズを増やすには、`/etc/apt/apt.conf` に `APT::Cache-Limit` を設定します。以下のコマンドを実行すれば、アップグレードするには十分な値が設定されます:

```
# echo 'APT::Cache-Limit "12500000";' >> /etc/apt/apt.conf
```

ここでは、`/etc/apt/apt.conf` ファイル内にまだこの値を設定していない場合を想定しています。

場合によっては衝突や依存関係のループのために、APT の `APT::Force-LoopBreak` オプションを有効にして、必須パッケージを一時的に削除しなければならないかもしれません。その場合 `aptitude` はこのことを警告してアップグレードを中断します。`aptitude` のコマンドラインに `-o APT::Force-LoopBreak=1` を指定すれば、この状態を回避できます。

システムの依存関係の構造が非常に混乱していて、手動での介入が必要となることもあります。通常これは `aptitude` を用いるか、あるいは

```
# dpkg --remove package_name
```

として、目ざわりなパッケージを消す作業になります。または次の作業でもよいかもしれません。

```
# aptitude --fix-broken install
# dpkg --configure --pending
```

極端な場合には、コマンドラインから

```
# dpkg --install /path/to/package_name.deb
```

のように入力して、再インストールしなければならないかもしれません。

「純粋な」woody システムからのアップグレードでは、ファイルの競合は起こらないはずですが、非公式なバックポートパッケージをインストールしているなら起こるかもしれません。ファイルの競合が起こると、次のようなエラーになります:

```
Unpacking replacement <package-foo> ...
dpkg: error processing <package-name-for-foo> (--unpack):
  trying to overwrite '<some-file-name>',
  which is also in package '<package-bar>
```

ファイルの競合を解消するには、エラーメッセージの最後の行に表示されたパッケージを強制的に削除します:

```
# dpkg -r --force-depends package_name
```

問題が修正できたら、先に示したように `aptitude` コマンドを繰り返し入力すれば、アップグレードを再開できます。

アップグレードの最中に、いくつかのパッケージの設定・再設定に関する質問が表示されます。 `/etc/init.d` と `/etc/terminfo` ディレクトリに置かれるファイルと `/etc/manpath.config` に関しては、パッケージメンテナのバージョンに置き換えるようにしてください。システムの整合性を保つためには 'yes' と答えることが必要になります。古いバージョンも `.dpkg-old` という拡張子で保存されていますので、戻すのはいつでもできます。

どうすればよいかわからなくなったら、そのパッケージやファイルの名前を書き留めておいて、その問題解決は後回しにしましょう。 `typescript` ファイルを検索すれば、アップグレードの最中に画面に表示された情報を見直すこともできます。

4.5 リブート前にすべきこと

`aptitude dist-upgrade` が終了したら、「公式」にはアップグレードは終了したことになります。しかし次にリブートする前に、面倒を見てやらなければならないことがいくつかあります。

X Window System 関連のパッケージのアップグレードに関する詳しい情報は、 `/usr/share/doc/xfree86-common/README.Debian-upgrade.gz` を読んでください。これは以前の Debian リリースすべてのユーザに当てはまります。要するに、読め、ということです。

4.5.1 カーネルのアップグレード

Linux カーネルは、以上の手続きによっては更新されません。そのようにしたい場合は、 `kernel-image-*` パッケージのどれか一つをインストールするか、カスタマイズしたカーネルをソースからコンパイルするかします。

カーネルをアップグレードするには、まず初めにお使いのサブアーキテクチャに最適なカーネルを選択する必要があります。インストールできるカーネルの一覧は、以下のコマンドで得られます。

```
# apt-cache search ^kernel-image
```

インストールするカーネルイメージが決まったら、`aptitude install` でインストールします。新しいカーネルがインストールされたら、それを有効にするためにリブートしてください。

`woody` (と、それ以前のリリース) のインストールシステムでは、お使いのシステムにカーネルがパッケージとしてはインストールされないので、注意してください。これは `sarge` では変更され、カーネルの変更を追いかけるために仮想パッケージをインストールできるようになりました。これらのパッケージは `kernel-image-VERSION-ARCH` という名前になっていて、`VERSION` の部分是对应するカーネルのバージョン番号 (2.4 や 2.6) に、`ARCH` の部分 はサポートされるアーキテクチャのいずれかに対応しています。カーネルに対するセキュリティサポートもパッケージ管理に統合したいなら、アップグレードした後で、お使いのハードウェアに最適なカーネルパッケージをインストールしてください。

もうちょっと冒険したい人には、自分のカスタムカーネルをコンパイルする方法も Debian GNU/Linux は提供しています。 `kernel-package` をインストールして、`/usr/share/doc/kernel-package` の文書を読んでみてください。

4.5.2 raidtools2 から mdadm へのアップグレード

`raidtools2` はもはや上流開発者によって保守されておらず、`mdadm` パッケージに置き換えられました。`mdadm` は、設定ファイルなしにほぼ全ての RAID 管理タスクを実行できる単一のプログラムです - 設定ファイルはデフォルトでは使いません。

このセクションの以降の部分では、`raidtools2` のユーザにアップグレードのヒントをいくつか提供します。

上述したように、多くの場合 `mdadm` は設定ファイルなしで動作可能です。専用の RAID アレイを自動的に設定してくれるカーネルを使っているのなら、この節は飛ばしてもかまいません — `mdadm` をインストールするだけで、RAID は起動プロセス中に検出されるでしょう。Debian の標準的なカーネルは、起動時の RAID アレイ設定に対応しています。パーティションが “Linux raid autodetect” (`id fd`) に設定されていることを確認する必要もあるでしょう。以下のコマンドを実行すれば、現在のパーティションの種類が一覧表示されます:

```
# fdisk -l disk_device
```

自動的に設定された RAID アレイとそうではないものが混在した設定となっているなら、設定ファイルを作成しなければなりません。

`/etc/raidtab(raidtools2)` から `/etc/mdadm/mdadm.conf` (`mdadm`) へ移行するには、以下のように実行してください:

```
# echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > /etc/mdadm/mdadm.conf
# mdadm --examine --scan >> /etc/mdadm/mdadm.conf
```

これらのコマンドは、システムの既存のアレイに関する設定ファイルを生成します。

起動時に、RAID アレイが自動的に開始されることも確認すべきです。変数 `AUTOSTART` が `true` となっているかを確認するには、`/etc/default/mdadm` ファイルを調べてください。

4.6 時代遅れ (Obsolete) なパッケージ

数千個の新規パッケージが導入された一方で、sarge では woody にはあった 2000 個以上の古いパッケージが破棄されたり削除されてもいます。これら時代遅れのパッケージをアップグレードする手段は提供されていません。時代遅れのパッケージを使い続けても構いませんが、Debian プロジェクトは通常 sarge がリリースされてから 1 年後に³そのようなパッケージへのセキュリティサポートを打ち切り、その後は他のサポートも提供されないのが常です。もし存在するのなら、利用可能な代替品に置き換えることを推奨します。

あるパッケージが本ディストリビューションから削除された理由は、数多くあります - 上流ではもはや保守されていないため、そのパッケージを保守することに興味を抱く Debian 開発者がもはやいないため、提供していた機能が別のソフトウェア (あるいは新バージョン) に取って代わられたため、バグのために sarge にはもはや適さないとみなされたため、などです。最後の場合は、当該パッケージが“不安定版”ディストリビューション内には存在していることがあります。

更新されたシステム内のどのパッケージが“時代遅れ”なのかを検出するのは、パッケージ管理用フロントエンドが当該パッケージにその旨のマークを付けてくれるので簡単です。aptitude を使っているのなら、当該パッケージが“Obsolete and Locally Created Packages”欄に列記されているのに気づくでしょう。dselect も同じようなセクションを提供しますが、表示される一覧はわずかに異なっています。さらに、woody で手作業でパッケージをインストールするのに aptitude を使っていたのなら、手作業でインストールされたパッケージの記録が取られており、依存元パッケージが削除されればもはや不要となる依存関係のみによって導入されたパッケージに時代遅れのマークを付けることができます。また aptitude は、deborphan とは異なり、手作業でインストールしたパッケージには時代遅れのマークを付けません (依存関係によって自動でインストールされたものにはマークを付けます)。

時代遅れのパッケージを見つけるのに使える追加ツールとしては、以下のものがあります - deborphan や debfoster、cruft。deborphan を強く推奨しますが、同プログラムは (デフォルトモードでは) 時代遅れのライブラリ - “libs” や “oldlibs” セクション内にあり、他のパッケージに使われていないパッケージ - しか報告しません。これらのプログラムが表示したパッケージをやみくもに削除しないでください。特に、誤報しやすい非デフォルトのオプションを積極的に使っている場合はなおさらです。実際に削除する前に、削除を提案されたパッケージを手作業で調査 (その中身やサイズ、説明文など) することを強く推奨します。

Debian バグ追跡システム (<http://bugs.debian.org/>) は、パッケージが削除された理由についての情報を提供してくれることがよくあります。あるパッケージ自体についてのアーカイブ化されたバグ報告と、ftp.debian.org pseudo-package (<http://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org&archive=yes>) のアーカイブ化されたバグ報告の両方を調査すべきです。

4.6.1 ダミーパッケージ

woody のいくつかのパッケージは sarge では複数のパッケージに分割されていますが、これは大半がシステムの保守性を改善するためです。この場合におけるアップグレードを容易にするために、sarge

³あるいはその期間中に別のリリースが出ない限り。ある時点では、通常 2 個のみの安定版リリースがサポートされています。

はしばしば“ダミーの”パッケージ-woodyでの古いパッケージと同じ名前で、新規パッケージを導入するための依存関係を備えた空のパッケージ-を提供しています。これらの“ダミー”パッケージはアップグレード後は Obsolete 扱いとされ、安全に削除することができます。

大半の(すべてではない)ダミーパッケージの説明文には、その目的が記されています。しかしながらダミーパッケージの説明文は統一されていないため、自分のシステム内のダミーパッケージを検出するために `deborphan` を `--guess` オプション付で使うこともできます。いくつかのダミーパッケージは、アップグレード後に削除されることを意図しておらず、代わりに時間とともに変化するプログラムの利用可能な最新バージョンの記録用として使われることに注意してください。

章 5

sarge で知っておくべき問題点

5.1 Python パッケージへの変更点

sarge の python2.X パッケージには、標準モジュールである 'profile' と 'pstats' が含まれていません。これは当該モジュールに適用されているライセンスが、DFSG に合致しないためです (詳細はバグ報告 #293932 を参照してください)。これら 2 つのモジュールは、Debian アーカイブの non-free セクション内の python-profiler および python2.X-profiler パッケージにあります。

章 6

Debian GNU/Linux に関するさらなる情報

6.1 もっと読みたい

このリリースノートやインストールガイドを越えた、Debian GNU/Linux に関するより進んだ文書は、Debian Documentation Project (DDP) から公開されています。DDP は Debian のユーザや開発者向けに、品質の高い文書を作成することを目的としています。Debian ガイド、Debian メンテナ入門、Debian FAQ などなど、たくさんの文書があります。リソースの詳細すべては DDP のウェブサイト (<http://www.debian.org/doc/ddp>) から得られます。

それぞれのパッケージの文書は `/usr/share/doc/package` にインストールされています。ここには、著作権情報、Debian 固有の詳細、開発元の文書すべて、などが置かれています。

6.2 助けを求めるには

Debian ユーザ向けのヘルプ・アドバイス・サポートなどは、いろいろな場所から得られます。しかしこれらを頼りにするのは、その問題について徹底的に文書を調査してからにしましょう。このセクションでは新しく Debian ユーザになった人向けに、これらを簡単に紹介します。

6.2.1 メーリングリスト

Debian ユーザが最も興味を引かれるであろうメーリングリストは `debian-user` (英語) リストおよび `debian-user-language` (各国語) リストでしょう。これらのリストの詳細や講読のしかたについては、<http://lists.debian.org/> を見てください。利用にあたっては、あなたの疑問に対する答えが以前の投稿ですでに答えられていないかどうか、アーカイブをチェックしてください。また標準的なメーリングリストのエチケットに従うようにしてください。

6.2.2 インターネットリレーチャット (IRC)

Debian はサポート用の IRC チャンネルを Debian ユーザの協力によって運営しています。このチャンネルは Freenode IRC ネットワークにあります。このネットワークは、オープンソースコミュニティで資源共有や協力を進めるため、情報交換に使われているものです。このチャンネルにアクセスするには、お好みの IRC クライアントを irc.debian.org に接続し、`#debian` に join してください。

チャンネルのガイドラインに従い、他のユーザを大切にしてください。Freenode に関する詳細はウェブサイト (<http://freenode.net/>) を訪ねてみてください。

6.3 バグを報告する

私たちは Debian GNU/Linux を高品質な OS にするよう努めていますが、だからといって私たちの提供するパッケージにバグが皆無というわけにはいきません。Debian の「オープンな開発体制」という考え方に合致し、また、ユーザに対するサービスとして、私たちは報告されたバグに関するすべての情報を bugs.debian.org (<http://bugs.debian.org/>) にある バグ追跡システム (Bug Tracking System: BTS) で提供しています。

もしディストリビューションや、その一部であるパッケージされたソフトウェアにバグを見つけたら、次のリリースで修正できるよう、その問題点の報告をお願いします。バグを報告するには正しい電子メールアドレスが必要です。これをお願いしているのは、バグを追跡したり、開発者がより詳しい情報を必要とした場合に、報告者に連絡できるようにするためです。

バグ報告には `reportbug` というプログラムを使うこともできますし、電子メールを使っても構いません。バグ追跡システムに関する詳細やその使い方については、リファレンスカード (`doc-debian` パッケージをインストールしていれば `/usr/share/doc/debian` にあります) をお読み頂くか、またはバグ追跡システム (<http://bugs.debian.org/>) のウェブサイトからオンラインで入手することもできます。

6.4 Debian に貢献する

Debian への貢献は専門家でなくてもできます。問題を抱えたユーザを、いろいろなサポートメーリングリスト (<http://lists.debian.org/>) の場所で助けてあげることも、立派なコミュニティへの貢献です。開発メーリングリスト (<http://lists.debian.org/>) に参加して、ディストリビューション開発に関する問題を見つたり解決したりする (もちろん後者のほうが大切) ことも、もちろん非常に助けになります。Debian を高品質なディストリビューションに保つため、バグを報告して (<http://bugs.debian.org/>) その原因の特定や解決に際して開発者を助けてください。執筆が得意なら、文書 (<http://www.debian.org/doc/ddp>) 作成や既存文書のご自分の言語への翻訳 (<http://www.debian.org/international/>) に積極的に参加し、そこで貢献するのもよいでしょう。

もっと時間が自由になるなら、Debian に属するフリーソフトウェア集の一部を管理してみるのはいかがでしょうか。皆が Debian に入れてほしいと思っているソフトウェアを引き受けて管理するのは、特に価値の高い貢献です。これに関する詳細は、作業が望まれるパッケージ (<http://www.debian.org/devel/wnpp/>) をご覧になってください。Debian にはいくつかサブプロジェクトが存在

しており、特定のアーキテクチャへの移植、Debian Jr. (<http://www.debian.org/devel/debian-jr/>)、Debian Med (<http://www.debian.org/devel/debian-med/>) などが進められています。これらのうち、あなたが興味を持っているグループに参加するのもよいでしょう。

いずれにしても、あなたが何らかの形でフリーソフトウェアコミュニティに関わっているのなら、それがユーザとしてであれ、プログラマ、ライター、翻訳者のいずれとしてであれ、すでにあなたはフリーソフトウェア運動を助けてくださっているのです。貢献することは報いのあることですし、楽しいことです。新しい人々に会う機会も増えます。きっと暖かな、楽しい気持ちになれるはずです。

付録 A

woody システムの管理

この付録には、sarge へアップグレードする前に woody パッケージを確実にインストールしたりアップグレードする方法についての情報が述べられています。特定の状況でのみ必要となるでしょう。

A.1 woody システムのアップグレード

基本的には、これまで行なってきた woody のあらゆるアップグレードと違いはありません。唯一異なるのは、'ソースリストのチェック' on page 27 で説明したようにパッケージリスト内に woody パッケージがまだ含まれているのを確認する必要があることです。

A.2 woody 版 aptitude のインストール

まず、'ソースリストのチェック' on page 27 内の指示に従って woody 版の aptitude がインストールされ、sarge 版はインストールされないようにする必要があります。

その後で、次のように実行して aptitude をインストールしてください:

```
# apt-get install aptitude
```

A.3 ソースリストのチェック

/etc/apt/sources.list 内で 'stable' を指定している行があるなら、効率よく sarge を“使う”用意ができています。すでに apt-get update を実行済みでも、以下の手順に従えば問題なく元に戻すことができます。

sarge からパッケージのインストールもしてしまっているなら、おそらくこれ以上 woody からパッケージをインストールしても無意味でしょう。この場合、続けるかどうかを自分で決断しなければなりません。パッケージをダウングレードすることはできますが、その方法はここでは扱いません。

(root になってから) お気に入りのエディタで `/etc/apt/sources.list` を開き、`deb http:` や `deb ftp:` で始まるすべて行の中に “stable” が指定されているかどうかを調べてください。もしあるなら、stable を woody に変更してください。

`deb file:` で始まっている行があるなら、その行が指定している 場所が woody か sarge のどちらのアーカイブなのかを 独力で調べなければなりません。

重要! `deb cdrom:` で始まっている行は、絶対に変更しないでください。変更するとその行は無効になって、もう一度 `apt-cdrom` を 実行しなければならなくなるでしょう。'cdrom' ソースが “unstable” を 指定していても心配しないでください。混乱するかもしれませんが、これで 正常なのです。

変更が済んだら、ファイルを保存してから次のように実行して、パッケージリストを更新してください:

```
# apt-get update
```