

openSUSE

11.3

www.novell.com

2010/12/28

リファレンス



リファレンス

Copyright © 2006–2010 Novell, Inc. and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled 「GNU Free Documentation License」.

For Novell trademarks, see the Novell Trademark and Service Mark list <http://www.novell.com/company/legal/trademarks/tmlist.html>. Linux* is a registered trademark of Linus Torvalds. All other third party trademarks are the property of their respective owners. A trademark symbol (®, ™ etc.) denotes a Novell trademark; an asterisk (*) denotes a third party trademark.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

下記に上記の日本語翻訳を掲載します。日本語の翻訳は公式なものではないことに注意してください。

Copyright © 2006–2010 Novell, Inc. および貢献者が全権利を留保しています。

この文書を、フリーソフトウェア財団発行の GNU フリー文書利用許諾契約書 バージョン 1.2 または (希望すれば) 1.3 が定める条件の下で複製、頒布、あるいは改変することを許可する。ただし、この著作権とライセンス表記については変更不可部分とする。この利用許諾契約書の複製物は「GNU フリー文書利用許諾契約書」という章に含まれている。

Novell 社の商標については、Novell 社の商標とサービスマーク一覧 <http://www.novell.com/company/legal/trademarks/tmlist.html> をご覧ください。Linux は Linus Torvalds 氏による登録商標です。その他の商標は各所有者の所有物です。商標シンボル (®, ™ など) はそれぞれ Novell 社の商標であることを示しています。また、アスタリスク (*) は第三者の商標を示しています。

この書籍内にある全ての情報は細部に至るまで最大限の注意を払って制作されていますが、完全に正確であることを保証するものではありません。Novell, Inc., SUSE LINUX Products GmbH, 著者, 翻訳者のいずれも、本書籍内の誤りとそこから生じる結果について、一切の保証はいたしません。

目次

このガイドについて	xi
パート I 高度な配置シナリオ	1
1 リモートインストール	3
1.1 リモートインストールの手順	3
1.2 インストール元のデータを保存するサーバの構築	13
1.3 ターゲットシステムの起動準備	23
1.4 インストールのためのターゲットシステムの起動	34
1.5 インストール処理の監視	37
2 高度なディスク設定	43
2.1 YaST パーティション設定の利用	43
2.2 LVM の設定	53
2.3 ソフトウェア RAID の設定	59
パート II ソフトウェアの管理と更新	65
3 ソフトウェアのインストールと削除	67
3.1 用語定義	68
3.2 KDE インターフェイス (Qt) の使用	69
3.3 GNOME インターフェイスの使用 (GTK+)	77
3.4 ソフトウェアリポジトリとサービスの管理	83

4	YaST オンライン更新	89
4.1	オンライン更新ダイアログ	90
4.2	修正のインストール	93
4.3	自動オンライン更新	94
5	インターネットからのパッケージのインストール	97
5.1	1 クリックインストール	98
5.2	YaST パッケージ検索	99
6	アドオン製品のインストール	103
6.1	アドオン	103
6.2	バイナリドライバ	104
7	コマンドラインツールを利用したソフトウェア管理	105
7.1	zypper の使用	105
7.2	RPM - パッケージマネージャ	118
	パート III システム管理	133
8	YaST を利用したユーザ管理	135
8.1	ユーザとグループの管理ダイアログ	135
8.2	ユーザアカウントの管理	137
8.3	ユーザアカウントに対する追加のオプション	140
8.4	ローカルユーザに対する既定値の変更	147
8.5	ユーザに対するグループの割り当て	148
8.6	グループの管理	149
8.7	ユーザ認証方法の変更	151
9	YaST を利用した言語と国の設定変更	153
9.1	システム言語の変更	153
9.2	国と時刻の設定変更	158
10	テキストモードでの YaST	161
10.1	モジュール間のナビゲーション	162
10.2	キーの組み合わせに関する制限	164
10.3	YaST コマンドラインオプション	165

11	プリンタの操作	167
11.1	印刷システムの作業手順	169
11.2	プリンタ接続の方法とプロトコル	169
11.3	ソフトウェアのインストール	170
11.4	ネットワークプリンタ	171
11.5	コマンドラインからの印刷	174
11.6	openSUSE での特殊機能	174
11.7	トラブルシューティング	177
12	グラフィカルユーザインターフェイスを利用したフォントのインストールと設定	187
12.1	X11 コアフォント	188
12.2	Xft	190
13	システム監視ユーティリティ	195
13.1	他用途ツール	196
13.2	システム情報	204
13.3	プロセス	210
13.4	メモリ	216
13.5	ネットワーク	219
13.6	/proc ファイルシステム	221
13.7	ハードウェア情報	225
13.8	ファイルとファイルシステム	226
13.9	ユーザ情報	229
13.10	日付と時刻	230
13.11	データのグラフ化: RRDtool	231
14	システムのアップグレードとシステム変更	239
14.1	システムのアップグレード	239
14.2	ソフトウェアの変更点	246
パート IV	システム	249
15	64 ビット環境における 32 ビットおよび 64 ビットアプリケーション	251
15.1	ランタイムサポート	251
15.2	ソフトウェア開発	252
15.3	両プラットフォーム対応のソフトウェアコンパイル	253
15.4	カーネル仕様	254

16	Linux システムの起動と設定	257
16.1	Linux の起動処理	257
16.2	init プロセス	261
16.3	/etc/sysconfig 経由でのシステム設定	271
17	ブートローダ GRUB	275
17.1	GRUB での起動	276
17.2	YaST を利用したブートローダの設定	288
17.3	Linux ブートローダのアンインストール	294
17.4	起動 CD の作成	295
17.5	グラフィカルな SUSE スクリーン	296
17.6	トラブルシューティング	297
17.7	さらなる情報	299
18	特殊なシステム機能	301
18.1	特殊なソフトウェアパッケージに関する情報	301
18.2	仮想コンソール	310
18.3	キーボードマッピング	310
18.4	言語と国の設定	311
19	udev による動的なカーネルデバイス管理	317
19.1	/dev ディレクトリ	317
19.2	カーネルの uevent と udev	318
19.3	ドライバ、カーネルモジュール、デバイス	319
19.4	起動と初期デバイス設定	319
19.5	udev デーモンの稼働監視	320
19.6	udev ルールによるカーネル側デバイスイベント処理への影響	321
19.7	固定のデバイス命名	329
19.8	udev で使用するファイル	330
19.9	さらなる情報	331
20	Bash と Bash スクリプト	333
20.1	「シェル」とは？	333
20.2	シェルスクリプトの作成	340
20.3	コマンドイベントのリダイレクト	342
20.4	別名定義の使用	343
20.5	Bash 内での変数の使用	343
20.6	コマンドのグループ化と組み合わせ	346
20.7	一般的なフロー制御	347
20.8	さらなる情報	349

パート V サービス 351

21 ネットワークの基礎 353

21.1	IP アドレスとルーティング	357
21.2	IPv6—次世代のインターネット	360
21.3	名前解決	371
21.4	YaST を利用したネットワーク接続の設定	373
21.5	NetworkManager	396
21.6	手動でのネットワーク設定方法	398
21.7	ダイアルアップ接続支援としての smpppd	416

22 ネットワーク内の SLP サービス 421

22.1	インストール	422
22.2	SLP の有効化	422
22.3	openSUSE での SLP フロントエンド	422
22.4	SLP 経由でのインストール	423
22.5	SLP 経由でのサービス提供	423
22.6	さらなる情報	424

23 ドメインネームシステム 427

23.1	DNS 用語	427
23.2	インストール	428
23.3	YaST を利用した設定	429
23.4	BIND ネームサーバの起動	439
23.5	/etc/named.conf 設定ファイル	441
23.6	ゾーンファイル	446
23.7	ゾーンデータの動的な更新	451
23.8	機密を保持する通信	451
23.9	DNS セキュリティ	453
23.10	さらなる情報	453

24 DHCP 455

24.1	YaST での DHCP サーバ設定	456
24.2	DHCP ソフトウェアパッケージ	460
24.3	DHCP サーバ dhcpd	460
24.4	さらなる情報	465

25 NTP を利用した時刻同期 467

25.1	YaST を利用した NTP クライアントの設定	468
25.2	ネットワーク内にある NTP の手動設定	473

25.3	システム稼働時の動的な時刻同期	474
25.4	ローカル参照時計の設定	474
26	NFS でのファイル共有	477
26.1	必要なソフトウェアのインストール	478
26.2	YaST を利用したファイルシステムのインポート	478
26.3	手作業でのファイルシステムのインポート	479
26.4	YaST を利用したファイルシステムのエクスポート	481
26.5	手作業でのファイルシステムのエクスポート	488
26.6	Kerberos を利用した NFS	491
26.7	さらなる情報	492
27	Samba	493
27.1	用語	493
27.2	Samba サーバのインストール	495
27.3	Samba の起動と停止	495
27.4	Samba サーバの設定	496
27.5	クライアントの設定	504
27.6	ログインサーバとしての Samba の利用	505
27.7	さらなる情報	506
28	Apache HTTP サーバ	507
28.1	クイックスタート	507
28.2	Apache の設定	510
28.3	Apache の起動と停止	527
28.4	モジュールのインストール／有効化／設定	530
28.5	CGI スクリプトを動作させる方法	539
28.6	SSL で通信の機密を保持する Web サーバの設定	543
28.7	セキュリティ問題の回避	550
28.8	トラブルシューティング	552
28.9	さらなる情報	553
29	YaST を利用した FTP サーバの設定	557
29.1	FTP サーバの起動	559
29.2	FTP の一般的な設定	560
29.3	FTP パフォーマンス設定	561
29.4	認証	561
29.5	詳細設定	562
29.6	さらなる情報	562

パート VI モバイル環境 563

30 Linux でのモバイルコンピューティング 565

30.1	ラップトップ	565
30.2	モバイルハードウェア	573
30.3	携帯電話と PDA	575
30.4	さらなる情報	575

31 電源管理 577

31.1	省電力機能	577
31.2	Advanced Configuration and Power Interface (ACPI)	578
31.3	ハードディスクの休止	584
31.4	トラブルシューティング	586
31.5	さらなる情報	589

32 無線 LAN 591

32.1	無線 LAN 標準	591
32.2	動作モード	592
32.3	認証	593
32.4	暗号化	595
32.5	YaST を利用した設定	596
32.6	無線 LAN 設定における豆知識	605
32.7	トラブルシューティング	607
32.8	さらなる情報	609

33 タブレット PC の使用 611

33.1	タブレット PC パッケージのインストール	612
33.2	タブレットデバイスの設定	613
33.3	仮想キーボードの使用	613
33.4	ディスプレイの回転表示	614
33.5	ジェスチャー認識の使用	615
33.6	ペンを利用したメモ取りとスケッチ	618
33.7	トラブルシューティング	620
33.8	さらなる情報	622

34 ファイルのコピーと共有 623

34.1	シナリオ	624
34.2	アクセス方法	625
34.3	直接接続によるファイルアクセス	627
34.4	同一のコンピュータにおける異なる OS 上のファイルへのアクセス	628

34.5	Linux コンピュータ間のファイルコピー	630
34.6	SSH を利用した Linux と Windows コンピュータのファイルコピー	639
34.7	Linux コンピュータ間のファイル共有	640
34.8	Samba を利用した Linux と Windows のファイル共有	644
34.9	さらなる情報	647
35	ヘルプとドキュメンテーション	649
35.1	ドキュメンテーションディレクトリ	650
35.2	マニュアルページ	652
35.3	info ページ	654
35.4	openSUSE Wiki	654
A	サンプルネットワーク	655
B	GNU ライセンス	657
B.1	GNU General Public License	657
B.2	GNU 一般公衆利用許諾契約書 (日本語訳)	661
B.3	GNU Free Documentation License	665
B.4	GNU フリー文書利用許諾契約書 (日本語訳)	669

このガイドについて

このマニュアルは、openSUSE® に関する一般的な理解を深めるためのものです。主に基本的な管理知識のあるシステム管理者と一般家庭ユーザに向けて書かれています。このマニュアル内の様々な箇所をお読みになり、日々の生活で必要なアプリケーションの選択や高度なインストール方法の説明、設定シナリオなどをご確認ください。

高度な配置シナリオ

openSUSE を遠隔から配置したり、複雑なディスク設定を行ったりする方法を示しています。

ソフトウェアの管理と更新

YaST やコマンドライン、1 クリックインストール機能 を利用した、ソフトウェアのインストールと削除方法のほか、システムを最新の状態に保つ手順について示しています。

システム管理

お使いの openSUSE 環境における設定やアップグレードの方法、テキストモードでのシステム管理方法、Linux 管理者が知っておく必要のあるいくつかの重要なユーティリティに関する基礎知識をそれぞれ示しています。

システム

お使いの Linux システムのコンポーネントに関する紹介や、それらの操作に関するより深い情報を示しています。

サービス

openSUSE で提供される様々なネットワークサービスやファイルサービスについて、設定方法を示しています。

モバイル環境

openSUSE でのモバイルコンピューティングに関する紹介のほか、ワイヤレスコンピューティングや電源管理といった様々なオプションについて示しています。

このマニュアルの多くの章には、追加の文書リソースに対するリンクが含まれています。これらの追加文書はシステム内から利用することができるものがあるほか、インターネット上で公開されているようなものもあります。

お使いの製品で利用可能な文書の概要は、<http://www.novell.com/documentation/opensuse113>か、もしくは下記の章を参照してください。

1 利用可能な文書

HTML 版や PDF 版の各マニュアルは、それぞれ各種の言語に翻訳されています。この製品に対しては、それぞれ下記に示す ユーザ向けおよび管理者向けマニュアルが用意されています:

スタートアップ (↑スタートアップ)

システムへのインストールと、基本的な設定方法に関するガイドが記されています。はじめてお使いになる方の場合、ファイルシステムやユーザの考え方、アクセス権に関する紹介を行なっているほか、モバイル環境に対応するための openSUSE の各種機能について、概要を紹介しています。また、トラブルシューティングを行なうにあたって支援とアドバイスも書かれています。

KDE ユーザガイド (↑KDE ユーザガイド)

openSUSE の KDE デスクトップに関する紹介が書かれています。デスクトップの使用法や設定方法、主な作業手順などが書かれています。主に KDE を既定のデスクトップとして便利に使いたい方に向けた文書です。

GNOME ユーザガイド (↑GNOME ユーザガイド)

openSUSE の GNOME デスクトップに関する紹介が書かれています。デスクトップの使用法や設定方法、主な作業手順などが書かれています。主に GNOME を既定のデスクトップとして便利に使いたい方に向けた文書です。

アプリケーションガイド (↑アプリケーションガイド)

openSUSE にある主なデスクトップアプリケーションの使用と設定方法が書かれています。このガイドでは、ブラウザや電子メールソフトのほか、オフィスアプリケーションやコラボレーションツールと呼ばれるものをそれぞれ紹介しています。また、グラフィックやマルチメディアに関連したアプリケーションも紹介しています。

リファレンス (1 ページ)

openSUSE に関する一般的な理解を深め、より詳しいシステム管理作業を行なうための情報が書かれています。主にシステム管理者のほか、システム管理知識のあるホームユーザに向けた文書です。また、複雑な配置シ

ナリオやシステムの管理方法、主なシステムコンポーネントとのやりとりや openSUSE が提供するネットワークサービス、ファイルサービスに関する詳しい情報も書かれています。

セキュリティガイド (↑セキュリティガイド)

ローカル環境やネットワークセキュリティを含めた、システムセキュリティに関する基本的な考え方が書かれています。Novell AppArmor のようなセキュリティソフトウェア(プログラムが読み書きしたり実行したりするファイルをプログラム単位で指定できるもの)の一般的な使い方を示しているほか、セキュリティ関連のイベント情報を確実に収集するための監査システムの使い方も示しています。

広範囲をカバーするマニュアルのほか、下記のようなクイックスタートガイドも利用できます:

KDE クイックスタート (↑KDE クイックスタート)

KDE デスクトップと、そこで動作する主なアプリケーションを紹介しています。の紹介が書かれています。

GNOME クイックスタート (↑GNOME クイックスタート)

GNOME デスクトップと、そこで動作する主なアプリケーションを紹介しています。

インストール クイックスタート (↑インストール クイックスタート)

システム要件の一覧と、DVD や ISO イメージから openSUSE をインストールする場合の手順が書かれています。

Novell AppArmor クイックスタート

Novell® AppArmor の主な考え方について、説明が書かれています。

ほとんどの製品マニュアルは HTML 版の形で、インストール済みシステムの /usr/share/doc/manual に置かれています。またデスクトップのヘルプセンターからもアクセスすることができます。最新の文書アップデートは、<http://www.novell.com/documentation> に置いています。ここからお使いの製品について、PDF 版と HTML 版をダウンロードすることができます。

2 フィードバック

いくつかの方法でフィードバックを送ることができます:

バグや機能追加リクエスト

製品のコンポーネントに対してバグの報告を行ったり、もしくは機能の追加 リクエストを送信したりしたい場合は、<https://bugzilla.novell.com/> をご利用ください。文書内の間違いについては、各製品の *Documentation* コンポーネントに対してバグ報告をお願いいたします。

Bugzilla を初めてお使いになる場合は、下記の記事をお読みください:

- http://ja.opensuse.org/openSUSE:Submitting_bug_reports
- http://ja.opensuse.org/openSUSE:Bug_reporting_FAQ

ユーザコメント

このマニュアルに対するコメントや提案のほか、この製品に含まれる他のドキュメント 類に対するコメントを歓迎します。オンラインドキュメントの場合は、それぞれの ページ下部にあるコメント機能をご利用いただくか、もしくは<http://www.novell.com/documentation/feedback.html> から コメントをお送りください。

3 文書規約

このマニュアルでは、下記のルールで文書を記述しています:

- `/etc/passwd`: ディレクトリ名やファイル名を示しています
- `placeholder`: 置き換えを示しています `placeholder` を実際の値に置き換えます
- `PATH`: `PATH` という名前の環境変数を示しています
- `ls, --help`: コマンドやオプション、パラメータ を示しています
- `user`: ユーザまたはグループ

- **Alt, Alt + F1:** 入力するキーやキーの組み合わせを示しています; キーはキーボードに書かれているとおりに大文字で示されます
- **ファイル, ファイル > 名前を付けて保存:** メニュー項目やボタンなどを示しています
- **ダンシングペンギン** (他のマニュアル内 ペンギンの章): 他のマニュアル内にある章を示しています

4 このマニュアルの作成について

この書籍は、DocBook (詳しくは <http://www.docbook.org> をご覧ください) のサブセットである Novdoc で書かれています。XML のソースファイルは xmllint で検証された後に xsltproc で処理され、Norman Walsh 氏のスタイルシートのカスタマイズ版を利用して XSL-FO に変換されます。最終的な PDF ファイルは RenderX 提供の XEP で生成しています。また、このマニュアルを構築するために使用するオープンソースツールとその環境は、openSUSE と共に公開されている susedoc パッケージ内にあります。

5 ソースコード

openSUSE は、どなたにでもご利用いただけます。ソースコードについては、http://www.novell.com/products/suselinux/source_code.html に書かれた手順でダウンロードしてください。ご要望があればソースコードを DVD でお送りすることもできます。手数料と輸送費として、15 ドルまたは 15 ユーロをいただいております。DVD 形式でのソースコードをご要望の場合は、いずれも英語にて sourcedvd@suse.de [<mailto:sourcedvd@suse.de>] 宛に電子メールでご連絡いただくか、下記の住所宛にご連絡ください。

SUSE Linux Products GmbH
Product Management
openSUSE
Maxfeldstr. 5
D-90409 Nürnberg
Germany

6 謝辞

多数の無償貢献のお陰で、Linux 開発者はその開発にあたってグローバルな協力を行なうことができています。我々は彼らのそのような努力に感謝します— 彼らの貢献がなければ本ディストリビューションは存在していませんでした。また、Frank Zappa 氏と Pawar 氏にも感謝しています。もちろん Linus Torvalds 氏には特に感謝しています。

Have a lot of fun!

SUSE チームより

パートⅠ. 高度な配置シナリオ

リモートインストール

openSUSE® は異なる複数の方法でインストールすることができます。第1章 *YaST* を利用したインストール(↑スタートアップ) で記述されている方法のほか、openSUSE をネットワーク経由でインストールしたり、全く手を触れることなくインストールしたりすることもできます。

それぞれの方法は2つの簡単なチェックを行なうところから始まります: 1つめはその方法を利用するにあたっての前提条件を確認すること、2つめは基本的な流れを確認することです。詳しい手順は、それぞれのインストール方法を決定することで決まります。

注記

下記の章では、これから openSUSE をインストールしようとしているシステムを **ターゲットシステム** または **インストールターゲット** と呼びます。また、**リポジトリ** (以前は「インストールソース」と呼んでいました) は、インストール作業に必要な全てのデータがある場所のことを指します。たとえば物理メディアである CD や DVD のほか、お使いのネットワーク内でインストールデータを配布するネットワークサーバなどもリポジトリと呼びます。

1.1 リモートインストールの手順

この章では、リモートインストールを行なうにあたって一般的なインストール手順を示しています。それぞれの手順を利用する場合は、前提条件の一覧

と手順の概要をよくお読みください。それぞれの段階でより詳しい説明が必要となる場合は、それぞれ提供される参照先をお読みください。

1.1.1 VNC を利用したシンプルなりモートインストール (固定のネットワーク設定)

この種類のインストールでは、ターゲットシステムを起動してインストールの準備を行なうため、ターゲットシステムへの物理的な(ネットワークなどの遠隔手段に頼らない)アクセスと、ターゲットシステムに対して設定する IP アドレスが必要です。インストールプログラムが起動した後、遠隔のワークステーションからインストールプログラムに対して VNC による 接続をし、操作を行ないます。操作手順については第1章 *YaST* を利用したインストール (↑スタートアップ) に書かれているものと同じ手順です。

この種類のインストールを行なうには、下記の要件を満たす必要があります:

- リモートのリポジトリ: **NFS, HTTP, FTP, SMB** のうちのいずれかのプロトコルに対応したサーバと、そこに接続するためのネットワーク環境
- ネットワークの動作するターゲットシステム
- VNC ビューアと呼ばれるソフトウェア、もしくは Java の利用できるブラウザ (Firefox, Konqueror, Internet Explorer, Opera など) がインストールされた操作端末
- openSUSE のメディアキットに含まれる起動メディア (CD, DVD, USB フラッシュメモリ)。openSUSE のメディアキットについて、詳しくは項「インストールメディアの選択」(第1章 *YaST* を利用したインストール, ↑スタートアップ) をお読みください。
- リポジトリ側と操作端末に割り当て済みの有効な固定 IP アドレス
- ターゲットシステムに割り当てる有効な固定 IP アドレス

この方法でインストールを行なうには、下記の手順で行なってください:

- 1 まずは1.2項「インストール元のデータを保存するサーバの構築」(13 ページ) に書かれた手順に従ってリポジトリを構築します。NFS, HTTP, FTP のいずれかのネットワークサーバを選択してください。なお、SMB リポジ

トリを構築する場合は、1.2.5項「SMB リポジトリの管理」(21 ページ)をお読みください。

- 2 openSUSE のメディアキットに含まれる起動メディア(CD, DVD, USB フラッシュメモリ)を利用して、ターゲットシステムを起動します。openSUSE のメディアキットについて、詳しくは項「インストールメディアの選択」(第1章 *YaST* を利用したインストール, ↑スタートアップ)をお読みください。
- 3 ターゲットシステムでの起動画面が表示されたら、VNC の設定とリポジトリのアドレスを指定するため、起動オプションに入力を行ないます。詳しくは1.4項「インストールのためのターゲットシステムの起動」(34 ページ)をお読みください。

ターゲットシステムはテキストベースの環境で起動し、VNC ビューアやブラウザでアクセスするためのネットワークアドレスとディスプレイ番号が表示されます。VNC によるインストールでは自分自身の存在を OpenSLP で通知するため、ファイアウォールで禁止されていたりしなければ Konqueror を利用して `service:/` や `slp:/` から発見することもできます。

- 4 操作端末側では VNC ビューアのアプリケーションを起動するか、もしくはブラウザを起動して、1.5.1項「VNC インストール」(38 ページ)の方法で接続を行ないます。
- 5 あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されているインストール手順に従って行なってください。インストールの最終段階ではターゲットシステムを再起動しますが、この際はターゲットシステムに接続しなおしてください。
- 6 これでインストール作業は完了です。

1.1.2 VNC を利用したシンプルなりモートインストール (動的なネットワーク設定)

この種類のインストールでは、ターゲットシステムを起動してインストールの準備を行なうため、ターゲットシステムへの物理的な(ネットワークなどの遠隔手段に頼らない)アクセスが必要です。ネットワークの設定作業は DHCP を利用して行ないます。インストールプログラムが起動した後、遠隔のワークステーションからインストールプログラムに対して VNC による接続をし、操作を行ないます。

この種類のインストールを行なうには、下記の要件を満たす必要があります:

- リモートのリポジトリ: **NFS, HTTP, FTP, SMB** のうちのいずれかのプロトコルに対応したサーバと、そこに接続するためのネットワーク環境
- ネットワークの動作するターゲットシステム
- VNC ビューアと呼ばれるソフトウェア、もしくはJavaの利用できるブラウザ (**Firefox, Konqueror, Internet Explorer, Opera** のいずれか) がインストールされた操作端末
- openSUSE のメディアキットに含まれる起動メディア (CD, DVD, USB フラッシュメモリ)。openSUSE のメディアキットについて、詳しくは項「インストールメディアの選択」(第1章 *YaST* を利用したインストール, ↑スタートアップ) をお読みください。
- IP アドレスを提供する **DHCP** サーバ

この方法でインストールを行なうには、下記の手順で行なってください:

- 1 まずは1.2項「インストール元のデータを保存するサーバの構築」(13 ページ)に書かれた手順に従ってリポジトリを構築します。NFS, HTTP, FTP のいずれかのネットワークサーバを選択してください。なお、SMB リポジトリを構築する場合は、1.2.5項「SMB リポジトリの管理」(21 ページ)をお読みください。
- 2 openSUSE のメディアキットに含まれる起動メディア (CD, DVD, USB フラッシュメモリ) を利用して、ターゲットシステムを起動します。openSUSE のメディアキットについて、詳しくは項「インストールメディアの選択」(第1章 *YaST* を利用したインストール, ↑スタートアップ) をお読みください。
- 3 ターゲットシステムでの起動画面が表示されたら、VNC の設定とリポジトリのアドレスを指定するため、起動オプションに入力を行ないます。詳しくは1.4項「インストールのためのターゲットシステムの起動」(34 ページ)をお読みください。

ターゲットシステムはテキストベースの環境で起動し、VNC ビューアやブラウザでアクセスするためのネットワークアドレスとディスプレイ番号が表示されます。VNC によるインストールでは自分自身の存在を OpenSLP で通知するため、ファイアウォールで禁止されていたりしなければ Konqueror を利用して `service:/` や `slp:/` から発見することもできます。

- 4 操作端末側ではVNCビューアのアプリケーションを起動するか、もしくはブラウザを起動して、1.5.1項「VNC インストール」(38 ページ)の方法で接続を行ないます。
- 5 あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されているインストール手順に従って行なってください。インストールの最終段階ではターゲットシステムを再起動しますが、この際はターゲットシステムに接続しなおしてください。
- 6 これでインストール作業は完了です。

1.1.3 VNC を利用したリモートインストール (PXE ブートと Wake on LAN を使用)

このインストール方法の場合は、全く手を触れることなくインストールすることができます。ターゲットマシンは遠隔から開始して起動します。ユーザの操作は実際のインストール作業でのみ必要になります。これはサイトをまたがるような配置を行なうのに便利です。

この種類のインストールを行なうには、下記の要件を満たす必要があります:

- リモートのリポジトリ: NFS, HTTP, FTP, SMB のうちのいずれかのプロトコルに対応したサーバと、そこに接続するためのネットワーク環境
- TFTP サーバ
- お使いのネットワーク内で動作している DHCP サーバ
- PXE ブートと Wake on LAN に対応したネットワークデバイスを持ち、ネットワーク環境に接続されたターゲットシステム
- VNC ビューアと呼ばれるソフトウェア、もしくはJavaの利用できるブラウザ (Firefox, Konqueror, Internet Explorer, Opera のいずれか) がインストールされた操作端末

この方法でインストールを行なうには、下記の手順で行なってください:

- 1 まずは1.2項「インストール元のデータを保存するサーバの構築」(13 ページ)に書かれた手順に従ってリポジトリを構築します。NFS, HTTP, FTP の

いずれかのネットワークサーバを選択してください。なお、SMB リポジトリを構築する場合は、1.2.5項「SMB リポジトリの管理」(21 ページ)をお読みください。

- 2 ターゲットシステムが起動イメージをダウンロードできるよう、TFTPサーバを設定します。詳しい手順については1.3.2項「TFTP サーバの構築」(26 ページ)をお読みください。
- 3 任意のマシンにIPアドレスを配布し、TFTPサーバの場所を通知するようDHCPサーバを設定します。詳しくは1.3.1項「DHCP サーバの構築」(23 ページ)をお読みください。
- 4 ターゲットシステムをPXEブートができるよう設定します。詳しくは1.3.5項「PXE ブートのためのターゲットシステムの準備」(33 ページ)をお読みください。
- 5 ターゲットシステムの起動処理は、Wake on LAN を利用して開始します。1.3.7項「Wake on LAN」(33 ページ)をお読みください。
- 6 操作端末側ではVNCビューアのアプリケーションを起動するか、もしくはブラウザを起動して、1.5.1項「VNC インストール」(38 ページ)の方法で接続を行ないます。
- 7 あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されているインストール手順に従って行なってください。インストールの最終段階ではターゲットシステムを再起動しますが、この際はターゲットシステムに接続しなおしてください。
- 8 これでインストール作業は完了です。

1.1.4 SSH を利用したシンプルなりモートインストール (固定のネットワーク設定)

この種類のインストールでは、ターゲットシステムを起動してインストールの準備を行なうため、ターゲットシステムへの物理的な(ネットワークなどの遠隔手段に頼らない)アクセスと、ターゲットシステムに対して設定するIPアドレスが必要です。インストールプログラムが起動した後、遠隔のワークステーションからインストールプログラムに対してSSHによる接続をし、操

作を行ないます。操作手順については第1章 *YaST* を利用したインストール (↑スタートアップ) に書かれているものと同じ手順です。

この種類のインストールを行なうには、下記の要件を満たす必要があります:

- リモートのリポジトリ: **NFS, HTTP, FTP, SMB** のうちのいずれかのプロトコルに対応したサーバと、そこに接続するためのネットワーク環境
- ネットワークの動作するターゲットシステム
- **SSH** クライアントソフトウェアがインストールされた操作端末
- **openSUSE** のメディアキットに含まれる起動メディア (**CD, DVD, USB** フラッシュメモリ)。 **openSUSE** のメディアキットについて、詳しくは項「インストールメディアの選択」 (第1章 *YaST* を利用したインストール, ↑スタートアップ) をお読みください。
- リポジトリ側と操作端末に割り当て済みの有効な固定 **IP** アドレス
- ターゲットシステムに割り当てる有効な固定 **IP** アドレス

この方法でインストールを行なうには、下記の手順で行なってください:

- 1 まずは1.2項「インストール元のデータを保存するサーバの構築」 (13 ページ) に書かれた手順に従ってリポジトリを構築します。 **NFS, HTTP, FTP** のいずれかのネットワークサーバを選択してください。なお、 **SMB** リポジトリを構築する場合は、 1.2.5項「**SMB** リポジトリの管理」 (21 ページ) をお読みください。
- 2 **openSUSE** のメディアキットに含まれる起動メディア (**CD, DVD, USB** フラッシュメモリ) を利用して、ターゲットシステムを起動します。 **openSUSE** のメディアキットについて、詳しくは項「インストールメディアの選択」 (第1章 *YaST* を利用したインストール, ↑スタートアップ) をお読みください。
- 3 ターゲットシステムでの起動画面が表示されたら、 **SSH** の設定とリポジトリのアドレスを指定するため、起動オプションに入力を行ないます。詳しくは 1.4.2項「カスタムな起動オプションの使用」 (35 ページ) をお読みください。

ターゲットシステムはテキストベースの環境で起動し、 **SSH** クライアントでアクセスするためのネットワークアドレスが表示されます。

- 4 操作端末側ではSSHクライアントソフトウェアを起動し、「インストールプログラムへの接続」(40 ページ)の方法で接続を行ないます。
- 5 あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されているインストール手順に従って行なってください。インストールの最終段階ではターゲットシステムを再起動しますが、この際はターゲットシステムに接続しなおしてください。
- 6 これでインストール作業は完了です。

1.1.5 SSH を利用したシンプルなリモートインストール (動的なネットワーク設定)

この種類のインストールでは、ターゲットシステムを起動してインストールの準備を行なうため、ターゲットシステムへの物理的な(ネットワークなどの遠隔手段に頼らない)アクセスが必要です。ネットワークの設定作業はDHCPを利用して行ないます。インストールプログラムが起動した後、遠隔のワークステーションからインストールプログラムに対してSSHによる接続をし、操作を行ないます。

この種類のインストールを行なうには、下記の要件を満たす必要があります:

- リモートのリポジトリ: NFS, HTTP, FTP, SMB のうちのいずれかのプロトコルに対応したサーバと、そこに接続するためのネットワーク環境
- ネットワークの動作するターゲットシステム
- SSH クライアントソフトウェアがインストールされた操作端末
- openSUSE のメディアキットに含まれる起動メディア(CD, DVD, USB フラッシュメモリ)。openSUSE のメディアキットについて、詳しくは項「インストールメディアの選択」(第1章 *YaST* を利用したインストール, ↑スタートアップ)をお読みください。
- IP アドレスを提供する DHCP サーバ

この方法でインストールを行なうには、下記の手順で行なってください:

- 1 まずは1.2項「インストール元のデータを保存するサーバの構築」(13 ページ)に書かれた手順に従ってリポジトリを構築します。NFS, HTTP, FTP のいずれかのネットワークサーバを選択してください。なお、SMB リポジトリを構築する場合は、1.2.5項「SMB リポジトリの管理」(21 ページ)をお読みください。
 - 2 openSUSE のメディアキットに含まれる起動メディア(CD, DVD, USB フラッシュメモリ)を利用して、ターゲットシステムを起動します。openSUSE のメディアキットについて、詳しくは項「インストールメディアの選択」(第1章 *YaST* を利用したインストール, ↑スタートアップ)をお読みください。
 - 3 ターゲットシステムでの起動画面が表示されたら、SSH の設定とリポジトリのアドレスを指定するため、起動オプションに入力を行ないます。詳しくは1.4.2項「カスタムな起動オプションの使用」(35 ページ)をお読みください。
- ターゲットシステムはテキストベースの環境で起動し、SSH クライアントでアクセスするためのネットワークアドレスが表示されます。
- 4 操作端末側ではSSH クライアントソフトウェアを起動し、「インストールプログラムへの接続」(40 ページ)の方法で接続を行ないます。
 - 5 あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されているインストール手順に従って行なってください。インストールの最終段階ではターゲットシステムを再起動しますが、この際はターゲットシステムに接続しなおしてください。
 - 6 これでインストール作業は完了です。

1.1.6 SSH を利用したリモートインストール (PXE ブートと Wake on LAN を使用)

このインストール方法の場合は、全く手を触れることなくインストールすることができます。ターゲットマシンは遠隔から開始して起動します。

この種類のインストールを行なうには、下記の要件を満たす必要があります:

- リモートのリポジトリ: NFS, HTTP, FTP, SMB のうちのいずれかのプロトコルに対応したサーバと、そこに接続するためのネットワーク環境

- TFTP サーバ
- IP アドレスを提供する DHCP サーバ
- PXE ブートと Wake on LAN に対応したネットワークデバイスを持ち、ネットワーク環境に接続されたターゲットシステム
- SSH クライアントソフトウェアがインストールされた操作端末

この方法でインストールを行なうには、下記の手順で行なってください:

- 1 まずは1.2項「インストール元のデータを保存するサーバの構築」(13 ページ)に書かれた手順に従ってリポジトリを構築します。NFS, HTTP, FTP のいずれかのネットワークサーバを選択してください。なお、SMB リポジトリを構築する場合は、1.2.5項「SMB リポジトリの管理」(21 ページ)をお読みください。
- 2 ターゲットシステムが起動イメージをダウンロードできるよう、TFTPサーバを設定します。詳しい手順については1.3.2項「TFTP サーバの構築」(26 ページ)をお読みください。
- 3 任意のマシンに IP アドレスを配布し、TFTPサーバの場所を通知するよう DHCP サーバを設定します。詳しくは1.3.1項「DHCP サーバの構築」(23 ページ)をお読みください。
- 4 ターゲットシステムを PXE ブートができるよう設定します。詳しくは1.3.5 項「PXE ブートのためのターゲットシステムの準備」(33 ページ)をお読みください。
- 5 ターゲットシステムの起動処理は、Wake on LAN を利用して開始します。1.3.7項「Wake on LAN」(33 ページ)をお読みください。
- 6 操作端末側では SSH クライアントソフトウェアを起動し、1.5.2項「SSH インストール」(40 ページ)の方法で接続を行ないます。
- 7 あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されているインストール手順に従って行なってください。インストールの最終段階ではターゲットシステムを再起動しますが、この際はターゲットシステムに接続しなおしてください。
- 8 これでインストール作業は完了です。

1.2 インストール元のデータを保存するサーバの構築

openSUSE のインストール元として使用するサーバは、動作しているオペレーティングシステムによって様々な設定方法が考えられます。インストールサーバとして使用するのに最も簡単なのは、openSUSE 11.1 以降のバージョンで YaST を利用した場合です。

ティップ

Linux マシンのインストールには Microsoft Windows のマシンを使用することができます。詳しくは 1.2.5 項 「SMB リポジトリの管理」 (21 ページ) をお読みください。

1.2.1 YaST を利用したインストールサーバの構築

YaST では、ネットワークリポジトリを作成するためのグラフィカルツールを提供しています。このツールでは HTTP, FTP, NFS の各ネットワーク インストールサーバを設定することができます。

- 1 まずはインストールサーバとして動作させたいマシンに対して、root でログインします。
- 2 `yast2-instserver` パッケージをインストールします。
- 3 `YaST> その他> インストールサーバ` を起動します。
- 4 続いてリポジトリの種類 (HTTP, FTP, NFS のいずれか) を選択します。選択したサービスは、システム起動時に自動的に起動されるようになります。お使いのシステムで既に対象のサービスが起動されている状態で、サーバの設定を手動で行ないたい場合は、ネットワークサービスを設定しないを選択して自動設定を行なわないようにしてください。また、どちらの場合でもインストールデータを保持するディレクトリは、サーバ内で定義しておく必要があります。

- 5 必要なりポジトリ種類の設定を行ないます。サービスの種類によって自動設定の手順が異なります。自動設定を行なわないように指定した場合は、この手順は飛ばされます。

FTPやHTTPのサービスを介してインストールデータを公開する際の、ルートディレクトリの別名を設定します。インストールデータはそれぞれ、`ftp://サーバの IP アドレス/別名/名前`(FTPの場合)または`http://サーバの IP アドレス/別名/名前`(HTTPの場合)からアクセスできるようになります。ここで、`名前`にはこの後の手順で設定するリポジトリの名称が入ります。以前の段階でNFSを指定した場合は、ワイルドカードとエクスポートの指定を行ないます。NFSサービスに対しては、`nfs://サーバの IP アドレス/名前`からアクセスできるようになります。NFSとエクスポートについて、詳しくは第26章 *NFS* でのファイル共有(477 ページ)をお読みください。

ティップ: ファイアウォール設定

お使いのシステムのファイアウォール設定で、それぞれ HTTP, NFS, FTP で使用するポートを許可していることを事前に確認しておいてください。ファイアウォールでポートを開くを押すと許可することができると、ファイアウォールの詳細から設定することもできます。

- 6 ここでリポジトリの設定を行ないます。インストールメディアから目的の場所にコピーを行なう前に、まずはリポジトリの名前を指定します(製品とそのバージョンを短縮した名前を指定しておくのがお勧めです)。YaST では、インストール DVD のコピーを生成する代わりに、メディアの ISO イメージを提供するよう指定することもできます。これを行なうには、関連するチェックボックスを選択して、ISO ファイルがローカル環境のどこに存在しているのかを指定します。このインストールサーバを利用して配布する製品にもよりますが、追加のリポジトリとしてアドオン CD やサービスパック CD を追加しておく必要がある場合があります。また、インストールサーバを OpenSLP 経由で通知したい場合は、関連するオプションを選択してください。

ティップ

お使いのネットワーク環境で対応している場合は、お使いのリポジトリを OpenSLP で通知するよう指定しておくことをお勧めします。この指定を行なうことで、それぞれのターゲットマシンでわざわざネットワーク

パスを指定したりする手間を省くことができます。ターゲットシステムでは単に **SLP 起動オプション** を利用して起動するだけで、あとの設定を行なうことなくネットワーク リポジトリを検出することができます。詳しくは 1.4 項「インストールのためのターゲットシステムの起動」(34 ページ)をお読みください。

- 7 インストールデータのアップロード処理を行ないます。インストールサーバを設定するにあたってもっとも時間のかかる処理は、実際のインストールメディアをコピーする処理です。YaST で指示されたとおりの順序でメディアを挿入して、コピーが完了するまでお待ちください。データのコピーが全て完了すると、既存のリポジトリに関する概要の画面に戻ります。あとは **完了** を押して終了してください。

これでインストールサーバの準備は全て完了し、サービスを提供できるようになります。システムが起動するたびに自動的にサービスが起動しますので、起動時に特別な作業を行ったりする必要はありません。YaST の初期段階でネットワークサービスの自動設定を無効化していた場合のみ、このサービスを手作業で設定して開始する必要があります。

逆にリポジトリを削除するには、削除するリポジトリを選択して **削除** を押してください。インストールデータはシステムから削除されます。ネットワークサービスを無効化するには、関連する YaST モジュールをお使いください。

また、インストールサーバで複数の製品や複数のバージョンのインストールデータを提供したい場合は、YaST インストールサーバモジュールを起動して、概要画面から **追加** ボタンを押して新しいリポジトリを追加してください。

1.2.2 NFS リポジトリの手動構築

インストールソースとして NFS を設定するには、2 段階の手順で行ないます。1 つめはインストールデータを保存しておくためのディレクトリ構造の作成とインストールメディアのコピー、2 つめはインストールデータをネットワーク側に公開するためのエクスポート処理です。

インストールデータを保存しておくディレクトリを作成するには、下記の手順で行ないます:

- 1 **root** でログインします。

- 2 後の手順でインストールデータを保存しておくためのディレクトリを作成し、そのディレクトリに移動します。たとえば下記のようになります:

```
mkdir install/製品名/製品バージョン
cd install/製品名/製品バージョン
```

ここで、**製品** は製品名の略称を、**製品バージョン** には製品名とバージョンの入った文字列をそれぞれ指定します。

- 3 次に、メディアキットに含まれているそれぞれの DVD を挿入し、下記のコマンドを実行します:

- 3a** インストールDVDの内容全体を、インストールサーバのディレクトリにコピーします:

```
cp -a /media/(DVD-ROM ドライブのパス) .
```

ここで、**(DVD-ROM ドライブのパス)** には実際に DVD ドライブがマウントされているディレクトリを指定します。お使いのシステムで使用しているドライブにもよりますが、**cdrom, cdrecorder, dvd, dvdrrecorder** のいずれかを指定します。

- 3b** ディレクトリの名称を DVD 番号に名称変更します:

```
mv (DVD-ROM ドライブのパス) DVDx
```

ここで、**x** はコピーを行なった DVD の番号を指定します。

次に YaST を利用して、NFS によるリポジトリのエクスポートを行ないます。下記の手順で実施してください:

- 1 **root** でログインします。
- 2 **YaST> ネットワーク サービス> NFS サーバ** を起動します。
- 3 まずは **開始** と **ファイアウォールでポートを開く** をそれぞれ選択して、**次へ** を押します。
- 4 次に **ディレクトリの追加** を押して、インストールソースが存在するディレクトリを選択します。ここでは **製品バージョン** を選択してください。

- 5 さらに **ホストの追加** を押し、インストールデータを公開したい相手マシンのホスト名を入力します。ここでホスト名を指定する代わりにワイルドカードやネットワークアドレスの範囲、もしくはお使いのネットワークにおけるドメイン名を入力することもできます。あとは必要なエクスポートオプションを指定するか、もしくは多くの環境でうまく動作できるよう既定値のままで設定します。NFS共有を公開する際の文法について、詳しくは `exports` のマニュアルページをお読みください。
- 6 最後に **完了** を押します。これで `openSUSE` のリポジトリを保持する NFS サーバの設定は完了です。自動的に NFS サーバは起動され、システム起動時から動作するようになります。

YaST NFS サーバモジュールを利用する代わりに NFS を手動で設定して公開したい場合は、下記の手順で行ないます:

- 1 `root` でログインします。
- 2 `/etc/exports` ファイルを開き、下記の行を入力します:

```
/製品バージョン *(ro,root_squash,sync)
```

上記の設定では、ディレクトリ `/製品バージョン` を、同じネットワーク内に属しているか、もしくはこのサーバに接続することのできる任意のホストに公開します。このサーバへのアクセスを制限するには、汎用ワイルドカード `*` の代わりにネットマスクやドメイン名を指定してください。詳しくは `export` のマニュアルページをお読みください。設定ファイルを記入したら、保存して終了します。

- 3 NFS のサービスをシステム起動時に稼働するよう設定するには、下記のコマンドを入力してください:

```
insserv /etc/init.d/nfsserver
```

- 4 NFS サーバを起動するには、`rcnfsserver start` と入力します。後から NFS サーバの設定を変更するには、設定ファイルを修正したあと下記のコマンドで NFS デーモンを再起動してください。`rcnfsserver restart`。

OpenSLP 経由で NFS サーバを通知すると、お使いのネットワーク内にある全てのクライアントに対してそのアドレスを通知することができます。

- 1 `root` でログインします。

- 2 /etc/slp.reg.d/install.suse.nfs.reg という名前で設定ファイルを作成し、下記の行を入力します:

```
# Register the NFS Installation Server
service:install.suse:nfs://$HOSTNAME/リポジトリのパス/DVD1,en,65535
description=NFS Repository
```

ここで *リポジトリのパス* には、お使いのサーバにおけるインストール元のパスを指定します。

- 3 最後に `rcslpd start` コマンドで OpenSLP デーモンを起動してください。

OpenSLP についての詳細は、`/usr/share/doc/packages/openslp/` ディレクトリ以下に配置されているパッケージドキュメンテーションか、第22章 ネットワーク内の *SLP* サービス (421 ページ) をお読みください。NFS に関する詳細は、第26章 *NFS* でのファイル共有 (477 ページ) をお読みください。

1.2.3 FTP リポジトリの手動構築

FTP でのリポジトリ作成方法は、NFS リポジトリの作成方法ととても似ています。FTP リポジトリについても OpenSLP を介した通知を行なうことができます。

- 1 1.2.2項「NFS リポジトリの手動構築」(15 ページ)に示された手順に従って、インストール元を保持するディレクトリを作成します。
- 2 インストール元のディレクトリをFTPサーバ経由で公開するよう設定します:

2a `root` でログインし、YaST ソフトウェア管理を利用して `vsftpd` パッケージをインストールします。

2b FTP サーバのルートディレクトリに移動します:

```
cd /srv/ftp
```

2c FTP サーバのルートディレクトリ以下に、インストール元を保存するためのサブディレクトリを作成します:

```
mkdir リポジトリ
```

ここで、リポジトリには製品名を入力します。

- 2d** インストールリポジトリの内容をマウントし、FTP サーバのルートディレクトリ以下から閲覧できるようにします:

```
mount --bind リポジトリのパス /srv/ftp/リポジトリ
```

ここで、リポジトリのパス and リポジトリには、それぞれ該当する値を入力してください。このマウントを恒久的に行ないたい場合は、`/etc/fstab` に設定を行なってください。

- 2e** `vsftpd` コマンドで `vsftpd` を起動します。

- 3** お使いのネットワーク環境で許可されていれば、OpenSLP 経由で通知を行なうことができます:

- 3a** `/etc/slp.reg.d/install.suse.ftp.reg` という 名前 で設定ファイルを作成し、下記の行を入力します:

```
# Register the FTP Installation Server
service:install.suse:ftp://$HOSTNAME/リポジトリ/DVD1,en,65535
description=FTP Repository
```

ここでリポジトリには、お使いのサーバ内でリポジトリが存在するディレクトリを入力します。なお、`service:` の行は1行で (改行せずに) 入力しなければなりません。

- 3b** 最後に `rcslpd start` コマンドで OpenSLP デーモンを起動してください。

ティップ: YaST を利用した FTP サーバの構築

FTP でのインストールサーバを設定するにあたって、手作業ではなく YaST を利用したい場合は、第29章 *YaST を利用した FTP サーバの設定* (557 ページ) にある YaST FTP サーバモジュールの使い方をお読みください。

1.2.4 HTTP リポジトリの手動構築

HTTP のリポジトリ作成方法は、NFS リポジトリの作成方法ととても似ています。HTTP リポジトリについても OpenSLP を介した通知を行なうことができます。

- 1 1.2.2項「NFS リポジトリの手動構築」(15 ページ)に示された手順に従って、インストール元を保持するディレクトリを作成します。
- 2 インストール元のディレクトリを HTTP サーバ経由で公開するよう設定します:

2a 28.1.2項「インストール」(508 ページ)に書かれている手順に従って、Web サーバ Apache をインストールします。 .

2b HTTP サーバのルートディレクトリ (/srv/www/htdocs) に移動し、インストール元を保持するためのサブディレクトリを作成します:

```
mkdir リポジトリ
```

ここで、リポジトリには製品の名前を入力します。

2c Web サーバのルートディレクトリ (/srv/www/htdocs) からインストール元のディレクトリをたどれるよう、シンボリックリンクを作成します:

```
ln -s リポジトリのパス /srv/www/htdocs/リポジトリ
```

2d HTTP サーバの設定ファイル (/etc/apache2/default-server.conf) を修正し、シンボリックリンクをたどれるようにします。

```
Options None
```

の行を

```
Options Indexes FollowSymLinks
```

に変更してください。

2e あとは `rcapache reload` コマンドで HTTP サーバ の設定を再読み込みさせてください。

- 3 お使いのネットワーク環境で許可されていれば、OpenSLP 経由で通知を行なうことができます:

- 3a /etc/slp.reg.d/install.suse.http.reg という名前で 設定ファイルを作成し、下記の行を入力します:

```
# Register the HTTP Installation Server
service:install.suse:http://$HOSTNAME/repository/DVD1/,en,65535
description=HTTP Repository
```

ここでリポジトリには、お使いのサーバ内でリポジトリが存在するディレクトリを入力します。なお、service: の行は1行で (改行せずに) 入力しなければなりません。

- 3b 最後に rcsldap start コマンドで OpenSLP デーモンを起動してください。

1.2.5 SMB リポジトリの管理

SMB を利用すると、Microsoft Windows のマシンをインストール元として 使用することができます。周囲に Linux マシンが存在しない環境でも Linux の 配置を行なうことができます。

Windows 共有を利用して openSUSE のリポジトリを公開するには、下記の手 順で行なってください:

- 1 お使いの Windows マシンにログインします。
- 2 インストール元のツリー構造全体を保持するための新しいディレクトリを 作成します。たとえば INSTALL のような名前で作成します。
- 3 Windows の文書に従って、このディレクトリを共有するよう設定します。
- 4 作成したディレクトリに移動して、製品名のサブディレクトリを作成しま す。ここで 製品名 には、実際の製品名を入力します。
- 5 Enter the INSTALL/製品名 のディレクトリに移動して、それぞれの DVD を個別のディレクトリにコピーします。たとえば DVD1 や DVD2 のような サブディレクトリになります。

SMB 共有をリポジトリとしてマウントするには、下記の手順を行ないます:

- 1 インストールターゲットを起動します。
- 2 インストールを選択します。
- 3 F4 キーを押してリポジトリ選択の画面に移動します。
- 4 SMB を選択して、Windows マシンのホスト名か IP アドレス、および 共有名(たとえば INSTALL/製品名/DVD1 のようになります)とユーザ名、パスワードをそれぞれ入力します。

Enter を押すと、YaST が起動して インストール処理が始まります。

1.2.6 サーバ内にあるインストールメディアの ISO イメージの使用

サーバのディレクトリに対してメディアの内容をコピーする代わりに、インストールメディアの ISO イメージをインストールサーバにコピーしてインストール時にターゲットにマウントさせることができます。HTTP, NFS, FTP サーバを設定し、メディアの内容をコピーする代わりに ISO イメージを設定するには、下記の手順で行なってください:

- 1 ISO イメージをダウンロードして、インストールサーバ内に保存します。
- 2 root でログインします。
- 3 それぞれ 1.2.2 項「NFS リポジトリの手動構築」(15 ページ), 1.2.3 項「FTP リポジトリの手動構築」(18 ページ), 1.2.4 項「HTTP リポジトリの手動構築」(20 ページ)で示されている手順で、インストールデータの配置先を選択および作成します。
- 4 それぞれの DVD に対してサブディレクトリを作成します。
- 5 最終的な配置先で ISO イメージをマウントします。具体的には下記のコマンドを入力します:

```
mount -o loop ISO ファイルのパス リポジトリのパス/製品名/メディア x
```

ここで、ISO ファイルのパスにはダウンロードした ISO イメージファイルのパスを、リポジトリのパスにはお使いのサーバにおける公開もとディレクトリを、製品名には製品名を、メディア Xにはメディアの種類(CD または DVD)とメディア番号をそれぞれ入力してください。

- 6 それぞれ ISO イメージの枚数分だけ上記の手順を繰り返します。
- 7 あとはそれぞれ 1.2.2項「NFS リポジトリの手動構築」(15 ページ)、1.2.3項「FTP リポジトリの手動構築」(18 ページ)、1.2.4項「HTTP リポジトリの手動構築」(20 ページ)に書かれた手順で通常通りのインストールを行ないます。

システム起動時に自動で ISO イメージをマウントさせたい場合は、それぞれのマウント項目を `/etc/fstab` 内に記入してください。上記の例を `fstab` に記入する場合は、下記ようになります:

```
ISO ファイルのパス リポジトリのパス/製品名  
メディア X auto loop
```

1.3 ターゲットシステムの起動準備

この章では、より複雑な起動方法についての設定手順を示しています。それぞれ DHCP, PXE ブート, TFTP, Wake on LAN の設定例を記載しています。

1.3.1 DHCP サーバの構築

DHCP サーバを構築するには、2つの方法があります。1つは openSUSE 固有の方法として YaST のグラフィカルインターフェイスを利用する方法、もう1つは手作業で設定ファイルを変更する方法です。DHCP サーバについての詳細は、第24章 *DHCP* (455 ページ)をお読みください。

YaST を利用した DHCP サーバの構築

ネットワーククライアントに対して TFTP サーバの場所を通知し、インストールターゲットが使用するべきブートイメージファイルを指定するには、DHCP サーバの設定に2つの項目を追記する必要があります。

- 1 DHCP サーバの動作しているマシンに対して、`root` でログインします。
- 2 `YaST` > ネットワーク サービス > *DHCP* サーバを起動します。
- 3 セットアップウィザードを利用して基本的な DHCP サーバの設定を済ませます。
- 4 熟練者設定を押し、スタートアップダイアログを抜ける際に表示される警告メッセージが表示されたら、はいを選択します。
- 5 まずは *設定済みの宣言* ダイアログでは、新しいシステムが存在するサブネットを選択して *編集* を押します。
- 6 サブネットの設定ダイアログでは、サブネットの設定に新しいオプションを追加するため、*追加* ボタンを押します。
- 7 `filename` を選択し、値には `pxelinux.0` と入力します。
- 8 さらにもう 1 つのオプション (`next-server`) を追加し、値として TFTP サーバのアドレスを入力します。
- 9 *OK* ボタンを押してから *完了* ボタンを押し、DHCP サーバ設定を完了します。

特定のホストに対して固定の IP アドレスを割り当てるよう DHCP サーバを設定するには、DHCP サーバモジュール内の *熟練者設定* (ステップ 4 (24 ページ)) を押し、*追加* ボタンから新しいホストを追加してください。あとはオプションとして `hardware` と `fixed-address` を追加し、それぞれ値としてハードウェアアドレスと割り当てる IP アドレスを指定してください。

DHCP サーバの手動構築

お使いのネットワーククライアントに対して自動でアドレスを割り当てるだけでなく、ターゲットマシンからインストールプログラムを起動できるようにする目的で、TFTP サーバとその中でのファイルを指定する必要があります。

- 1 DHCP サーバの動作しているマシンに対して、`root` でログインします。

2 /etc/dhcpd.conf 内に配置されているお使いの DHCP サーバ設定ファイルについて、下記のような行を追記します:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.1.200 192.168.1.228;
    # PXE 関連の項目
    #
    # "next-server" には、使用すべき TFTP サーバのアドレスを指定します。
    next-server TFTP サーバの IP アドレス;
    #
    # "filename" には、 /srv/tftpboot 以下に配置する pxelinux のファイル名
    # を指定します。
    filename "pxelinux.0";
}
```

Replace ここで、*TFTP* サーバの *IP* アドレスには実際の *TFTP* サーバのアドレスを記入します。dhcpd.conf で利用可能なオプションについて、詳しくは dhcpd.conf のマニュアルページをお読みください。

3 rcdhcpd restart コマンドを実行し、DHCP サーバを再起動します。

PXE と Wake on LAN を起動時に使用し、SSH でリモートコントロールを行いたい場合は、インストールターゲットに対して **DHCP** サーバから固定の **IP** アドレスを割り当てる必要があります。固定の **IP** アドレスを割り当てるには、上記の例にさらに下記のような追記を行なう必要があります:

```
group {
    # PXE 関連の項目
    #
    # "next-server" には、使用すべき TFTP サーバのアドレスを指定します。
    next-server TFTP サーバの IP アドレス;
    #
    # "filename" には、 /srv/tftpboot 以下に配置する pxelinux のファイル名
    # を指定します。
    filename "pxelinux.0";
    host test {
        hardware ethernet MAC アドレス;
        fixed-address 割り当てる IP アドレス;
    }
}
```

host の項目では、まずインストールターゲットのホスト名を指定します。指定したホストに対してホスト名と **IP** アドレスを指定するには、そのホストのハードウェアアドレス (**MAC** アドレス) を知っておく必要があります。それぞれの値はご自身の環境に応じて修正してください。

DHCP サーバを再起動すると、指定したホストに固定の IP アドレスが割り当てられますので、SSH を介して接続できるようになります。

1.3.2 TFTP サーバの構築

TFTP サーバは YaST で設定することができるほか、`xinetd` と TFTP に対応した他の Linux オペレーティングシステム上で、手作業による設定を行なうこともできます。TFTP サーバはターゲットシステムが起動するための起動イメージを配信します。

YaST を利用した TFTP サーバの構築

- 1 `root` でログインします。
- 2 `yast2-tftp-server` パッケージをインストールします。
- 3 *YaST* > ネットワーク サービス > *TFTP* サーバを 起動し、指示されたとおりにパッケージをインストールします。
- 4 その後 有効化を選択し、サーバを起動してシステムの起動処理に組み込まれていることを確認します。これでサーバの起動準備は完了です。`xinetd` は起動時に `tftpd` を 開始するようになります。
- 5 なお、お使いのマシンで動作しているファイアウォールで適切なポートを開くため、ファイアウォールでポートを開くも選択しておいてください。お使いのサーバでファイアウォールが有効になっていない場合は、このオプションは利用できません (選択する必要はありません)。
- 6 次に 参照を押して、起動イメージのディレクトリを 指定します。既定のディレクトリ `/tftpboot` は 自動的に作成され、選択されています。
- 7 最後に 完了を押すと設定が保存され、サーバが 起動されます。

TFTP サーバの手動構築

- 1 `root` でログインし、それぞれ `tftp` と `xinetd` パッケージをインストールします。

- 2 /srv/tftpboot と /srv/tftpboot/pxelinux.cfg のディレクトリが存在していなければ、それぞれ作成します。
- 3 1.3.3項「PXE ブートの使用」(27 ページ)で説明されている内容をもとにして、起動イメージとして必要なファイルを配置します。
- 4 /etc/xinetd.d ディレクトリ以下にある **xinetd** の設定ファイルを修正し、起動時に **TFTP** サーバが動作するようにします:

4a tftp ファイルが存在していない場合は `touch tftp` コマンドでファイルを作成し、`chmod 755 tftp` コマンドを実行して読み取れるようにします。

4b tftp ファイルを開いて、下記のとおり入力します:

```
service tftp
{
    socket_type          = dgram
    protocol             = udp
    wait                = yes
    user                 = root
    server               = /usr/sbin/in.tftpd
    server_args          = -s /srv/tftpboot
    disable              = no
}
```

4c ファイルを保存し、`rcxinetd restart` コマンドで **xinetd** を再起動します。

1.3.3 PXE ブートの使用

PXE の完全な仕様説明や技術的な背景に関する情報は、**Preboot Execution Environment (PXE)** 仕様として (<http://www.pix.net/software/pxeboot/archive/pxespec.pdf>; 英語) に説明があります。

- 1 インストールリポジトリ内にある `boot/<アーキテクチャ>/loader` ディレクトリに移動し、`linux`, `initrd`, `message`, `biostest`, `memtest` の各ファイルを /srv/tftpboot ディレクトリにコピーします。下記のように行なってください:

```
cp -a linux initrd message biostest memtest /srv/tftpboot
```

- 2 YaST からインストール DVD を利用して syslinux パッケージをインストールします。

- 3 以下のようにして /usr/share/syslinux/pxelinux.0 ファイルを /srv/tftpboot ディレクトリにコピー します:

```
cp -a /usr/share/syslinux/pxelinux.0 /srv/tftpboot
```

- 4 使用するインストールリポジトリのあるディレクトリに移動し、isolinux .cfg ファイルを /srv/tftpboot/pxelinux.cfg/default にコピー します:

```
cp -a boot/<アーキテクチャ>/loader/isolinux.cfg  
/srv/tftpboot/pxelinux.cfg/default
```

- 5 /srv/tftpboot/pxelinux.cfg/default ファイルを 編集し、それぞれ gfxboot, readinfo, framebuffer で始まる行を削除 します。

- 6 それぞれ failsafe と apic ラベルの行の下に、それぞれ下記の追加行を 挿入します:

insmod=カーネルモジュール

この行は、PXE クライアントがネットワークインストールを行なうのに必要な、ネットワークカーネルモジュールを指定しています。カーネルモジュールには、お使いのマシンに接続されたネットワークデバイスに対する適切なモジュール名を入力してください。

netdevice=インターフェイス名

この行は、ネットワークインストールを行なう際に使用するネットワーク インターフェイスの名前を指定しています。この行は、クライアントマシンに複数のネットワークカードが接続されている場合にのみ必要な行です。1 枚のネットワークカードしか接続されていない場合は省略可能です。

install=nfs://インストールサーバの IP アドレス/リポジトリのパス/DVD1

この行では、クライアントマシンをインストールするための NFS サーバと リポジトリの場所を指定しています。それぞれ インストールサーバの IP アドレスには利用するインストールサーバの IP アドレスを、

リポジトリのパスにはリポジトリへのパスを入力します。なお、HTTP, FTP, SMB の各リポジトリについても同じような方法で入力します。それぞれ冒頭の部分を http, ftp, smb に書き換えてください。

重要項目

SSH や VNC の起動パラメータのようなその他の起動オプションを指定したい場合は、install の行に追記してください。パラメータの概要と使用例は、1.4項「インストールのためのターゲットシステムの起動」(34 ページ)をお読みください。

ティップ: カーネルと initrd ファイル名の変更

カーネルと initrd イメージファイルについては、それぞれ異なる名前を指定することもできます。これは、同じ起動サーバから異なるオペレーティングシステムを提供したい場合に便利な機能です。ただし、TFTP サーバで PXE ブートを行なう場合のファイル名には、1 つまでしかピリオド (.) が許されないことに注意してください。

下記は /srv/tftpbboot/pxelinux.cfg/default ファイルの例です。お使いのネットワーク設定にあわせてプロトコルのプレフィクス (接頭辞) 部分を変更し、必要であればインストーラへの接続方法についても vnc, vncpassword や usessh, sshpassword などのオプションを追記してください。なお、改行を挟んで行の続きを記入するには、\ を行末に挿入してください。

```
default harrdisk
```

```
# default
label linux
    kernel linux
    append initrd=initrd ramdisk_size=65536 \
        install=nfs://インストールサーバの IP アドレス/リポジトリのパス/製品名/DVD1
```

```
# repair
label repair
    kernel linux
    append initrd=initrd splash=silent repair=1 showopts
```

```
# rescue
label rescue
    kernel linux
    append initrd=initrd ramdisk_size=65536 rescue=1
```

```
# bios test
label firmware
    kernel linux
    append initrd=biostest,initrd splash=silent install=exec:/bin/run_biostest
    showopts

# memory test
label memtest
    kernel memtest

# hard disk
label haddisk
    localboot 0

implicit      0
display       message
prompt        1
timeout       100
```

ここで、インストールサーバの IP アドレスと リボジトリのパスにはお使いの環境に合わせた 値を代入してください。

なお、次の章ではこのセットアップ方法で使用している **PXELINUX** のオプションについて、簡単な説明を行なっています。詳しい説明はsyslinuxパッケージの説明をお読みください。 /usr/share/doc/packages/syslinux/ 以下の ディレクトリに各種文書が配置されています。

1.3.4 PXELINUX 設定オプション

以下には**PXELINUX** の設定ファイルで利用する全てのオプションが示されています。

DEFAULT カーネルオプション...

既定のカーネルコマンドラインを指定します。 **PXELINUX** で自動的に起動した場合は、 **DEFAULT** の後に続く文字列が起動プロンプトに入力されたものとして扱われます。また、自動起動したことを示すために "auto" オプションも追加 されます。

設定ファイルが存在していない場合や、 **DEFAULT** の項目が設定ファイルに存在しなかった場合は、カーネル名 「linux」 でオプション指定無しであるものとされます。

APPEND オプション...

カーネルのコマンドラインに対して1つ以上のオプションを追加します。この項目は自動的に起動された場合も手動で選択した場合も追加されます。オプションはカーネルコマンドラインの先頭のほうから追加されるため、入力されたカーネルオプションは以前のものを上書きするような形になります。

LABEL ラベル名 KERNEL イメージ APPEND オプション...

起動するカーネルとしてラベル名を入力した場合、PXELINUXはイメージで指定したカーネルを読み込み、ファイルのグローバルセクションのオプション(LABEL コマンドの外側にあるもの)を本項のAPPEND オプションで指定したものに置き換えます。*image*の既定値は*label*と同じもので、APPEND が書かれていなければグローバルセクションの値が使用されます。最大で 128 個の LABEL 項目までが入力できます。

なお、GRUB では下記の書式を使用します:

```
title (タイトル)
kernel イメージ オプション
initrd initrd ファイル名
```

PXELINUX では下記の書式を使用します:

```
label (タイトル)
kernel イメージ
append オプション
```

ラベルはMS-DOSのファイル名と同じような解釈を行ない、それらは全てユニークでなければなりません。たとえば「v2.6.30」と「v2.6.31」というラベルがあった場合、それらはDOSファイル名では同じものを指してしまうため、区別できないものになってしまいます。

また、イメージは必ずしもLinuxカーネルである必要はありません; ブートセクターやCOMBOOTファイルでもかまいません。

APPEND -

何も追加しないことを指定します。LABEL セクション内のAPPENDの後にパラメータとして1文字のハイフンを指定すると、グローバルセクションのAPPENDを上書きする宣言になります。

LOCALBOOT 種類

PXELINUX では、**KERNEL** を指定する代わりに **LOCALBOOT 0** を指定すると、カーネルを起動する代わりに 指定したローカルディスクを起動するようになります。

種類	説明
0	通常どおりの起動を行ないます
4	Universal Network Driver Interface (UNDI) をメモリ内に保持したまま ローカル起動を行ないます
5	Universal Network Driver Interface (UNDI) を含めた PXE スタック全体をメモリ内に保持したままローカル起動を行ないます

その他の値は未定義です。 **UNDI** や **PXE** スタックについて詳しく知らない場合は、 0 を指定してください。

TIMEOUT 制限時間

自動的な起動を行なうまでの起動プロンプトの待機時間を指定します。単位は 1/10 秒で、何らかのキーボード入力が起こるとコマンドのユーザー入力が始まったものと見なして自動起動がキャンセルされます。 0 を指定すると、時間切れにならないようになります (既定値)。利用可能な最大の値は 35996 (1 時間より少し少ない値) です。

PROMPT フラグ

フラグに 0 を指定すると、 **Shift** キーや **Alt** が押されるか、もしくは **Caps Lock** や **Scroll Lock** がロックされている場合にのみ 起動プロンプトを表示します (既定値)。 フラグに 1 を指定すると、起動プロンプトは常に表示されます。

F2 ファイル名
F1 ファイル名
...etc...
F9 ファイル名
F10 ファイル名

起動プロンプトでそれぞれのファンクションキーを押した時に表示するファイルを指定します。これは起動前のオンラインヘルプ (たとえばカーネルのコマンドラインオプション) を表示する際などに使用することがで

きます。なお、従来のバージョンとの互換性を確保するため、F10 は F0 として入力することもできます。なお、現時点では F11 と F12 にファイル名を割り当てることはできません。

1.3.5 PXE ブートのためのターゲットシステムの準備

BIOS での起動順序に PXE 起動を含めるようにして、システムの BIOS が PXE ブートできるように設定します。

警告: BIOS 起動順序

BIOS の設定では、PXE の起動をハードディスクの起動よりも優先するように設定しないでください。このように設定してしまうと、システム起動時に毎回 PXE 起動が行なわれてしまい、インストールを毎回繰り返すことになってしまいます。

1.3.6 Wake on LAN のためのターゲットシステムの準備

Wake on LAN (WOL) を利用するには、事前に適切な BIOS オプションを設定しておく必要があります。また、ターゲットシステムの MAC アドレスも知っておく必要があります。MAC アドレスは Wake on LAN の実施に必要な情報です。

1.3.7 Wake on LAN

Wake on LAN は、そのマシンの MAC アドレスを含む特別なネットワークパケットを送信することで電源を入れる仕組みです。世界中にある各マシンにはユニークな MAC 識別子が割り当てられているため、異なるマシンを誤って起動してしまうようなことは考える必要がありません。

重要項目: 異なるネットワークセグメントをまたがる Wake on LAN

コントロール側のマシンが起動すべきインストールターゲットと同じネットワークセグメント内に存在していない場合は、マルチキャストとして WOL (Wake on LAN) パケットを送信するか、同じネットワークセグメント内に存在するマシンを遠隔から操作して WOL パケットを送信すると Wake on LAN を実現することができます。

1.4 インストールのためのターゲットシステムの起動

基本的には 1.3.7 項「Wake on LAN」(33 ページ) や 1.3.3 項「PXE ブートの使用」(27 ページ) で示したもの以外に、2 種類の方法でのインストール起動方法があります。ファンクションキーで設定して既定のオプションで起動する方法と、特定のハードウェアを使用する際に、カーネルに指定する必要がある起動オプションを起動画面で入力する方法です。

1.4.1 既定の起動オプションの使用

起動オプションについては第1章 *YaST* を利用したインストール(↑スタートアップ)で説明しています。通常は単にインストールを選択すればインストールの起動処理が始まります。

何らかの問題が発生した場合は、インストール—*ACPI* なし を選択するか、インストール—安全設定を選択してください。インストール処理時のトラブルシューティングについて、詳しくは項「インストールの問題」(第9章 よく発生する問題とその解決方法, ↑スタートアップ)をお読みください。

画面の下の方には、特定の環境で必要となるような拡張機能の案内が示されています。実際のパラメータ(詳しくは 1.4.2 項「カスタムな起動オプションの使用」(35 ページ)をお読みください)を覚えることなく、ファンクションキーを利用してインストール時の追加オプションを指定することができます。利用可能なファンクションキーについて、詳しくは項「起動画面」(第1章 *YaST* を利用したインストール, ↑スタートアップ)をお読みください。

1.4.2 カスタムな起動オプションの使用

適切な起動オプションを使用すると、インストールの処理をより便利に行なうことができるようになります。多くのパラメータは `linuxrc` のルーチンを利用して後から指定することもできますが、起動オプションを利用した方がより簡単に指定することができます。セットアップを自動化するような場合は、`initrd` ファイルや `info` ファイル で与えることもできます。

下記の表には、この章で示したインストールシナリオと起動時に必要なパラメータ、および関連する起動オプションの一覧を示しています。それぞれこの表に現われている順序で追記してってください。たとえば (全てを 1 行で記述します):

```
install=xxx netdevice=xxx hostip=xxx netmask=xxx vnc=xxx vncpassword=xxx
```

ここでそれぞれの `xxx` には、お使いの環境に合わせた 値を代入してください。

表 1.1 この章で使用するインストール (起動) シナリオ

インストールシナリオ	起動時に必要なパラメータ	起動オプション
第1章 <i>YaST</i> を利用したインストール(↑スタートアップ)	無し: システムは自動で起動します	何も必要ありません
1.1.1項 「VNC を利用したシンプルなりモートインストール (固定のネットワーク設定)」 (4 ページ)	<ul style="list-style-type: none">・ インストールサーバの場所・ ネットワークデバイス・ IP アドレス・ ネットマスク・ ゲートウェイ・ VNC の有効可否設定・ VNC パスワード	<ul style="list-style-type: none">・ <code>install=(nfs,http,ftp,smb)://インストールメディアの場所</code>・ <code>netdevice=ネットワークデバイス名(複数のネットワークデバイスが利用できる環境の場合のみ)</code>・ <code>hostip=IP アドレス</code>・ <code>netmask=ネットマスク</code>・ <code>gateway=ゲートウェイ</code>

インストールシナリオ	起動時に必要なパラメータ	起動オプション
		<ul style="list-style-type: none"> • vnc=1 • vncpassword=パスワード
1.1.2項「VNC を利用したシンプルなリモートインストール (動的なネットワーク設定)」 (5 ページ)	<ul style="list-style-type: none"> • インストールサーバの場所 • VNC の有効可否設定 • VNC パスワード 	<ul style="list-style-type: none"> • install=(nfs,http,ftp,smb)://インストールメディアの場所 • vnc=1 • vncpassword=パスワード
1.1.3項「VNC を利用したリモートインストール (PXE ブートと Wake on LAN を使用)」 (7 ページ)	<ul style="list-style-type: none"> • インストールサーバの場所 • TFTP サーバの場所 • VNC の有効可否設定 • VNC パスワード 	特にありません; 処理はそれぞれ PXE, DHCP 経由で行ないます
1.1.4項「SSH を利用したシンプルなリモートインストール (固定のネットワーク設定)」 (8 ページ)	<ul style="list-style-type: none"> • インストールサーバの場所 • ネットワークデバイス • IP アドレス • ネットマスク • ゲートウェイ • SSH の有効可否設定 • SSH パスワード 	<ul style="list-style-type: none"> • install=(nfs,http,ftp,smb)://インストールメディアの場所 • netdevice=ネットワークデバイス名 (複数のネットワークデバイスが利用できる環境の場合のみ) • hostip=IP アドレス • netmask=ネットマスク • gateway=ゲートウェイ • usessh=1

インストールシナリオ	起動時に必要なパラメータ	起動オプション
		<ul style="list-style-type: none"> • sshpassword=パスワード
1.1.5項「SSH を利用したシンプルなりモートインストール (動的なネットワーク設定)」 (10 ページ)	<ul style="list-style-type: none"> • インストールサーバの場所 • SSH の有効可否設定 • SSH パスワード 	<ul style="list-style-type: none"> • install=(nfs,http,ftp,smb)://インストールメディアの場所 • usessh=1 • sshpassword=パスワード
1.1.6項「SSH を利用したりモートインストール (PXE ブートと Wake on LAN を使用)」 (11 ページ)	<ul style="list-style-type: none"> • インストールサーバの場所 • TFTP サーバの場所 • SSH の有効可否設定 • SSH パスワード 	特にありません; 処理はそれぞれ PXE, DHCP 経由で行ないます

ティップ: linuxrc 起動オプションに関する詳細

Linux システムを起動するにあたって使用される linuxrc の起動オプションについて、詳しくは <http://ja.opensuse.org/Linuxrc> をお読みください。

1.5 インストール処理の監視

インストール処理を遠隔から監視するには、いくつかの方法があります。インストール時に適切な起動オプションを指定すると、それぞれ VNC または SSH でインストール処理を操作し、遠隔からシステム設定を行なうことができます。

1.5.1 VNC インストール

VNC ビューアと呼ばれるソフトウェアを使用することで、任意のオペレーティングシステムが動作するマシンから openSUSE のインストールを遠隔で操作することができます。この章では、VNC ビューアアプリケーションや Web ブラウザを利用した設定方法を示しています。

VNC インストールの準備

VNC インストールを行なうにあたって、ターゲットシステムで行なわなければならないことは、インストールの起動段階で必要な起動オプションを設定することだけです(詳しくは 1.4.2 項「カスタムな起動オプションの使用」(35 ページ)をお読みください)。ターゲットシステムはテキストベースの環境で起動し、VNC クライアントがインストールプログラムに接続されるのを待つ動作になります。

その後インストールプログラムは、接続に必要な IP アドレスとディスプレイ番号を通知します。ターゲットシステムに対して物理的なアクセスを行なうことができる環境の場合は、インストール処理の起動後に情報が提供されます。VNC クライアントを起動して提供された情報を入力し、さらにパスワードも入力してください。

また、インストールターゲットは自分自身を OpenSLP でアナウンスする仕組みになっているため、物理的なアクセスが行なえない環境でもインストールターゲットに関する情報を SLP ブラウザで取得することができます。OpenSLP 対応のネットワーク環境とマシンで、下記のようにして行なってください:

- 1 KDE のファイルブラウザ兼 Web ブラウザである Konqueror を起動します。
- 2 場所バー内に `service://yast.installation.suse` と入力します。Konqueror のウインドウ内にターゲットシステムが表示されるはずです。あとはそのアイコンを選択することで、KDE の VNC ビューアが起動され、インストール処理が始まります。また、VNC ビューアを単体で起動して、提供される IP アドレスの後ろに `:1` を追加した文字列を入力することで接続することもできます。

インストールプログラムへの接続

基本的に VNC サーバ(この場合インストールターゲット)に接続するには、2 種類の方法があります。任意のオペレーティングシステムで動作する独立した VNC ビューアを立ち上げる方法と、Java の利用できる Web ブラウザを利用する方法です。

VNC を利用する場合、他の Linux ディストリビューションや Windows、Mac OS などのような任意のオペレーティングシステムからインストールを操作することができます。

Linux マシンの場合は、tightvnc パッケージがインストールされていることを確認してください。Windows マシンの場合は、同アプリケーションの Windows 移植版をインストールしてください。TightVNC の Windows 版は、TightVNC の Web ページ (<http://www.tightvnc.com/download.html>) からダウンロードすることができます。

ターゲットマシン上で動作しているインストールプログラムに接続するには、下記のようにして行ないます:

- 1 VNC ビューアを起動します。
- 2 SLP ブラウザやインストールプログラム自身が提供する、インストールターゲットの IP アドレスとディスプレイ番号をそれぞれ入力します:

IP アドレス:ディスプレイ番号

あとは通常のローカルインストールの時と同じように、YaST の画面がお使いのウインドウ内に表示されるようになります。

Web ブラウザを利用してインストールプログラムに接続する場合は、VNC ソフトウェアのインストールが不要になるだけでなく、オペレーティングシステムについても任意のものを選択することができるようになります。お使いのブラウザアプリケーションが Java に対応しているもの (Firefox, Internet Explorer, Konqueror, Opera など) であれば、どの環境でも Linux システムのインストールを行なうことができます。

VNC インストールを行なうには、下記のようにして行ないます:

- 1 お使いの Web ブラウザを起動します。

- 2 アドレス欄に下記のとおり入力します:

`http://ターゲットの IP アドレス:5801`

- 3 VNCのパスワードを尋ねられたらパスワードを入力します。あとは通常のローカルインストールと同様の YaST 画面がブラウザ内に表示されます。

1.5.2 SSH インストール

SSHを利用すると、任意のSSHクライアントソフトウェアの動作するマシンから、インストール作業の遠隔操作を行なうことができます。

SSH インストールの準備

必要なソフトウェアパッケージ (Linux の場合は OpenSSH、Windows の場合は PuTTY) のインストールのほか、SSH インストールには適切な起動オプションを指定する必要があります。詳しくは 1.4.2 項「カスタムな起動オプションの使用」(35 ページ)をお読みください。OpenSSH は SUSE Linux ベースのオペレーティングシステムであれば既定でインストールされます。

インストールプログラムへの接続

- 1 インストールターゲットの IP アドレスを取得します。ターゲットマシンに物理的なアクセスを行なうことのできる環境であれば、起動時にコンソールに表示された IP アドレスを読んでください。そうでない環境の場合は、DHCP サーバの設定から、そのホストに割り当てられた IP アドレスを取得してください。

- 2 下記のコマンドラインを入力します:

`ssh -X root@ターゲットの IP アドレス`

ここで、ターゲットの IP アドレスには、実際のインストールターゲットの IP アドレスを記入してください。

- 3 ユーザ名を尋ねられたら、`root` と入力します。

- 4 パスワードを尋ねられたら、SSH 起動オプションで指定しておいたパスワードを入力します。認証が完了すると、インストールターゲットのコマンドラインプロンプトが表示されます。
- 5 インストールプログラムを起動するため、`yast` と入力してください。あとは第1章 *YaST* を利用したインストール(↑スタートアップ)で示されている通常の YaST 画面が表示されます。

高度なディスク設定

より洗練されたシステム設定を行なう場合、ディスクの設定についても高度な設定を必要とします。YaST では、このような高度な設定を含む、一般的に利用する全てのパーティション作業に対応しています。たとえばブロックデバイスに対して永続的な名前を設定するには、`/dev/disk/by-id`や`/dev/disk/by-uuid`のような形式を利用します。また Logical Volume Management (LVM) では、通常のセットアップで設定する従来のパーティション方式より、ずっと柔軟な運用を実現できるよう設計されています。LVMでのスナップショット機能はデータバックアップを容易に作成することができますし、いわゆる RAID と呼ばれるディスクの冗長配列機能も提供されていて、データの完全性と性能、耐障害性を補強することができます。また、openSUSE ではマルチパス I/O (詳しくはストレージ管理ガイド内のマルチパス関連の章をお読みください) にも対応しているほか、iSCSI デバイスをネットワークディスクとして利用するオプションも用意されています。

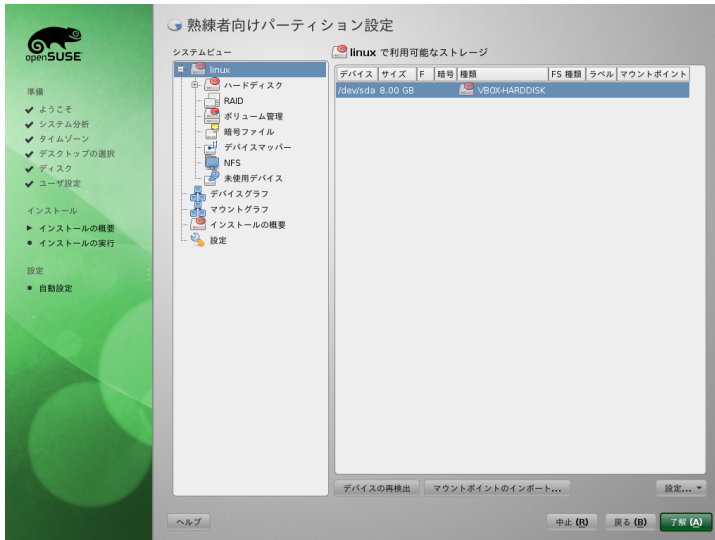
2.1 YaST パーティション設定の利用

熟練者向けパーティション設定 図2.1 「YaST パーティション設定」(44 ページ)では、複数のハードディスクのパーティション設定を変更することができます。パーティションは追加や削除、サイズ変更や編集を行なうことができます。また、この YaST モジュールからソフトウェア RAID や LVM にアクセスすることもできます。

警告: 稼働中のシステムのパーティション変更

お使いのシステムが稼働中であってもパーティション設定を変更することは可能ですが、作業ミスによるデータ損失の可能性がとても高いことに注意する必要があります。可能であればインストール済みのシステムに対するパーティション設定の変更は避け、どうしても必要である場合はお使いのデータを完全にバックアップしてから実施してください。

図 2.1 YaST パーティション設定



YaST の 熟練者向けパーティション設定 ダイアログには、利用可能なストレージの一覧に全ての接続済みハードディスクに対する既存のパーティションや、提案されたパーティション設定が表示されています。ハードディスク全体は /dev/sda のように数字無しでの表記に、パーティションは /dev/sda1 のように数字付きでそれぞれ表示されます。それぞれパーティションのサイズや種類、暗号化状態やファイルシステム (FS)、マウントポイントも表示されています。マウントポイントとは、Linux のファイルシステムツリー内のどこに割り当てるかを指定するためのものです。

左側にあるシステムビューには、いくつかの機能を表示させるためのビューが提供されています。既存のストレージ設定に対する情報収集や、RAID、ボリューム管理、暗号ファイル、NFS などの機能を設定することができます。

インストール時に熟練者向けパーティション設定を実行した場合は、ハードディスクの空き容量についても一覧表示され、自動で選択されている形になっています。openSUSE®に対してさらなるディスク領域を設定するには、一覧表示の下から(ハードディスクの最後のパーティションから)順に必要な領域を空けていってください。たとえば既存のシステムに3つのパーティションが設定されていた場合、openSUSE に対して2つめのパーティションを占有させ、1つめと3つめのパーティションを他のオペレーティングシステムに利用させたりすることはできません。

2.1.1 パーティションの種類

それぞれのハードディスクには、最大で4つの項目を記録することのできるパーティションテーブルがあります。それぞれの項目はプライマリパーティションや拡張パーティションと呼ばれます。拡張パーティションはハードディスク内で1つしか設定できません。

プライマリパーティションは、単純に連続したシリンダ(物理的なハードディスク領域)から構成されるもので、特定のオペレーティングシステム向けに割り当てられるものです。プライマリパーティションはパーティションテーブルの仕様制限により、ハードディスクごとに最大で4つのパーティションまでしか作成できません。その代わりとなるのが拡張パーティションです。拡張パーティションもプライマリパーティションと同様に連続したシリンダを割り当てますが、拡張パーティションはさらにその内側を論理パーティションに分割することができます。論理パーティションはパーティションテーブルには記録されません。言い換えれば、拡張パーティションは論理パーティションの入れ物として動作することになります。

4つ以上のパーティションが必要である場合は、4つめのパーティション(3つめでも2つめでもかまいません)に拡張パーティションを作成します。もちろん拡張パーティションは空きのシリンダ範囲に設定しなければなりません。拡張パーティション内に論理パーティションを作成することで、必要な数のパーティションを作成してください。論理パーティションとして設定可能な最大数は、SCSI, SATA, FireWire の各ディスクの場合は15個まで、(E)IDE ディスクの場合は63個までとなります。Linux でパーティションを使用する場合は、どちらのパーティション種類であっても違いはありません。プライマリパーティション、論理パーティションとも全く同じに機能します。

2.1.2 パーティションの作成

何もない状態からパーティションを作成するには、ハードディスクを選択してから空き領域のあるハードディスクを選択し、下記の手順でパーティションタブを開いて設定します:

- 1 追加を押します。複数のハードディスクが接続されている場合は、どのハードディスクに新しいパーティションを作成するかを尋ねるダイアログが表示されます。
- 2 パーティションの種類(プライマリまたは拡張)を選択します。最大で4つのプライマリパーティションか、もしくは3つのプライマリパーティションと1つの拡張パーティションを作成することができます。拡張パーティション内には複数の論理パーティションを作成することができます(詳しくは2.1.1項「パーティションの種類」(45 ページ)をお読みください)。
- 3 次に利用したいファイルシステムとマウントポイントを選択します。YaSTでは、それぞれ作成しようとしているパーティションに対して自動的にマウントポイントを提案します。ラベルによるマウントなど、マウント方法を変更するには *fstab* オプションを押してください。
- 4 お使いの環境で、必要であれば追加のファイルシステムオプションを指定します。たとえば固定のデバイス名でマウントする必要がある場合などに設定します。利用可能なオプションについて、詳しくは2.1.3項「パーティションの編集」(46 ページ)をお読みください。
- 5 最後に 完了を押すと、お使いの環境のパーティション設定が適用され、パーティション設定モジュールを終了することができます。

インストール作業時にパーティション作成を行なっている場合は、インストール概要の画面に戻ります。

2.1.3 パーティションの編集

新しいパーティションを作成したり既存のパーティションを変更したりする場合には、様々なパラメータを設定することができます。新しいパーティションを作成する際には YaST が適切なパラメータを設定しますので、特にそこから変更する必要はありません。手動でパーティションの設定を変更するには、下記の手順で行ないます:

- 1 パーティションを選択します。
- 2 **編集** を押してパーティションの編集画面を表示させ、必要に応じてパラメータを変更します:

ファイルシステム ID

この段階でパーティションをフォーマットしたくない場合であっても、パーティションが正しく登録できるようにするためにファイルシステム ID を設定してください。利用可能な ID は、*Linux*, *Linux swap*, *Linux LVM*, *Linux RAID* のいずれかです。

ファイルシステム

パーティションのファイルシステムを変更するには、パーティションをフォーマットするを押し、ファイルシステムの一覧からファイルシステムの 種類を選びます。

警告: ファイルシステムの変更について

ファイルシステムを変更してパーティションを再フォーマットすると、パーティション上にある全てのデータが復元できない形で削除されます。

様々なファイルシステムについて、詳しくはストレージ管理ガイドをお読みください。

デバイスの暗号化

暗号化を有効に設定すると、全てのデータは暗号化された形でハードディスクに書き込まれます。機密データのセキュリティ向上には貢献するものの、暗号化は時間のかかる処理であるため、システムの性能を落とすことになってしまいます。ファイルシステムの暗号化について、詳しくは第11章 *Encrypting Partitions and Files* (↑セキュリティガイド)をお読みください。

fstab オプション

グローバルなファイルシステム管理ファイル(/etc/fstab)では、様々なパラメータを設定することができます。多くの場合は既定の設定で問題はありませんが、たとえばファイルシステムの識別方法をデバイス名からボリュームラベルに変更したりすることができます。ボリュームラベルでは / とスペースを除く全ての文字を利用することができます。

固定のデバイス名でマウントするよう設定するには、マウントオプションとして デバイス *ID*, *UUID*, ラベル のいずれかを選択してください。openSUSE では、固定のデバイス名を既定で利用するようになっています。

マウントオプションに ラベル を選択した場合は、選択した パーティション に対して適切なラベル名を指定する必要があります。たとえば /home にマウントしたいパーティションに対して、HOME のようなラベル名を設定します。

また、ファイルシステム上でクォータを設定したい場合は、クォータサポートを有効にする を選択してください。これは YaST の ユーザとグループの管理 モジュールを利用してクォータを設定する場合、事前に設定しておかなければなりません。クォータの設定について詳しくは、8.3.5項「クォータの管理」(145 ページ)をお読みください。

マウントポイント

指定したパーティションをファイルシステム構造のどのディレクトリにマウントするかを設定します。YaST が提案するマウントポイントから 選択するか、もしくは独自のディレクトリを入力します。

3 完了を押すと変更点を保存することができます。

注記: ファイルシステムのサイズ変更

既存のパーティションについてサイズの変更を行なうには、対象のパーティション を選択して サイズ変更 を押します。なお、パーティション サイズの変更は、マウント中の状態から行なうことはできません。パーティションのサイズ変更を行なう前に、関連するパーティションのマウントは解除してください。

2.1.4 熟練者向けオプション

システムビューでハードディスクデバイス (たとえば *sda*) を選択すると、熟練者向けパーティション設定ウインドウ内の右下に 熟練者向け機能.. のボタンが表示されます。このメニューには下記のコマンドが含まれています:

新しいパーティションテーブルの作成

このオプションは、選択したデバイスに対して新しいパーティションテーブルを作成します。

警告: 新しいパーティションテーブルの作成について

新しいパーティションテーブルを作成すると、そのデバイス上に存在する全てのパーティションとデータが復元不可能な形で削除されます。

このディスクを複製する

このオプションは、選択したデバイスのパーティションレイアウトとデータを利用可能な他のディスクに複製することができます。

2.1.5 高度なオプション

コンピュータのホスト名 (システムビューの枠内にあるツリーの最上位レベル) を選択すると、熟練者向けパーティション設定ウインドウ内の右下に設定... のボタンが表示されます。このメニューには下記のコマンドが含まれています:

iSCSI の設定

SCSI over IP のブロックデバイスにアクセスするには、まず iSCSI について設定を行なう必要があります。この作業を行なうと、パーティション一覧内に利用可能なデバイスが追加表示されるようになります。

マルチパスの設定

このオプションを選択すると、対応しているマストレージデバイスに対してマルチパスの設定を行なうことができます。

2.1.6 さらなるパーティション設定に関する豆知識

下記の章では、お使いのシステムにパーティションを設定するにあたって正しい決断を下すための、いくつかのヒントや豆知識を掲載しています。

ティップ: シリンダ数

パーティションツールによっては、パーティションのシリンダの数え方が異なる場合があります。0 から始まるものと、1 から始まるものの 2 種類があります。シリンダ数を数える場合は、シリンダ数の数え方に注意し、必要であれば 1 を足して計算してください。

スワップの使用

スワップは利用可能な物理メモリを拡張するための仕組みです。物理的に搭載されているメモリよりも大きなメモリを確保できるようになります。カーネル 2.4.10 以前のメモリ管理システムでは十分な量のスワップが必要でした。その当時は、物理メモリの 2 倍以上のスワップが存在していないと、システムの性能が上がらないことがありました。このような制限は今はありません。

Linux はメモリからディスクに移動するページを選択するのに、「Least Recently Used」(LRU; 参照される頻度が最も低いもの) と呼ばれる方式で選択を行います。そのため、動作中のアプリケーションにはより大きなメモリを提供し、キャッシュ処理もよりスムーズに動作するようになります。

アプリケーションが利用可能なメモリ量よりも大きなサイズの割り当てを行なおうとした場合は、スワップ関連の問題が生じます。3 つの主なケースが考えられます:

スワップが設定されていないシステム

アプリケーションは利用可能な最大のメモリを取得します。全てのキャッシュデータは開放されることになるため、他の動作中アプリケーションの速度は低下します。数分程度が経過すると、カーネルに用意されたメモリ不足解決器 (out-of-memory kill mechanism (OOM Killer)) が動作し、プロセスが kill されるようになります。

中程度の量のスワップが設定されたシステム (128 MB (メガバイト) から 512 MB)

当初の状態ではシステムはスワップ無しの状態と同じように速度が低下します。全ての物理メモリが割り当てられると、スワップ領域を使い出すようになります。この時点ではシステムの動作は非常に遅くなり、リモートからのコマンド実行がほとんど行なえない状態になります。スワップ領域を設定したハードディスクの速度にもよりますが、システムはシステムはその状態を 10 分から 15 分程度保持し、その後カーネルに用意

されたメモリ 不足解決器 (out-of-memory kill mechanism (OOM Killer)) が動作し、問題を解決しようとします。なお、コンピュータに「ディスクへのサスペンド」を設定する場合は、ある一定のサイズのスワップ領域が必要になります。この場合、スワップサイズはメモリ内に存在するデータを保持しておくのに十分なサイズ (512 MB から 1 GB 程度) を設定しておく必要があります。

大きな量のスワップが設定されたシステム (GB (ギガバイト) 以上)

過度にスワップしてしまうばかりか、その制御すらきかないようなアプリケーションについては、これを使用しないことが理想ですが、そのようなアプリケーションを使用する必要がある場合は、回復するのに何時間もの時間がかかってしまうことがあります。この場合、システムは不確定な状態になってしまい、たとえそのような欠陥アプリケーションを kill したとしても、他のプロセスはタイムアウトや処理エラーを発生させてしまいます。この場合は、コンピュータを再起動することで再度問題なく動作できるような状態に戻すことができます。大きな量のスワップは、アプリケーション側でこの機能をどうしても必要としている場合にのみ有効です。そのようなアプリケーション (たとえばデータベースシステムとか、画像加工 プログラムとか) では、必要に応じてハードディスク領域を直接利用するような仕組みを備えている場合があります。スワップ領域を大きくする代わりに、そのような設定を利用することをご検討ください。

システムが制御不能になっていない状況で、急につらなるスワップ領域が必要になった場合は、スワップ領域をオンラインで拡張することができます。スワップ 領域用にパーティションを準備してあれば、そのパーティションを YaST から 追加するだけで設定できます。そのようなパーティションをお持ちでない場合は、スワップファイルを利用してスワップの拡張を行なうこともできます。スワップファイルはスワップパーティションよりも一般に速度の面で劣りますが、いずれも物理メモリよりもずっと遅く、大きな違いはありません。

手順 2.1 スワップファイルの手動追加

稼働中のシステムにスワップファイルを追加するには、下記の手順で行ないます:

- 1 まずはお使いのシステムに中身が空のファイルを作成します。たとえば 128 MB (メガバイト) のスワップファイルを /var/lib/swap/swapfile ディレクトリ内に作成したい場合は、下記のコマンドを実行します:

```
mkdir -p /var/lib/swap
dd if=/dev/zero of=/var/lib/swap/swapfile bs=1M count=128
```

- 2 下記のコマンドを実行してスワップファイルの初期化を行ないます。

```
mkswap /var/lib/swap/swapfile
```

- 3 下記のコマンドでスワップを有効に設定します。

```
swapon /var/lib/swap/swapfile
```

スワップファイルを無効化するには、下記のコマンドを入力します。

```
swapoff /var/lib/swap/swapfile
```

- 4 また、現在使用中のスワップ領域を表示するには、下記のコマンドを入力します。

```
cat /proc/swaps
```

なお、上記の手順では一時的に使用するスワップ領域を設定しています。システムを再起動すると、設定したスワップは使用されなくなります。

- 5 このスワップファイルを恒久的に使用するには、下記の行を `/etc/fstab` に追加します:

```
/var/lib/swap/swapfile swap swap defaults 0 0
```

2.1.7 パーティション設定と LVM

熟練者向けパーティション設定では、システムビューの枠内にある **ボリューム管理** を利用して LVM 設定を行なうことができます。ただし、お使いのシステムに動作中の LVM 設定が存在している場合は、LVM の初期設定に入る段階で自動的にそれらが有効化されます。この場合、パーティションの存在する全てのディスク (有効化されたボリュームグループに属するもの) のパーティション設定は変更できません。Linux カーネルは使用中のパーティションが存在するディスクについて、変更されたパーティションテーブルを読み直すことができないためです。既にお使いのシステムで動作する LVM 設定が存在する場合は、物理的なパーティション設定は必要ではないため、その代わりに論理ボリュームの設定を変更してください。

ボリュームに対する情報は、物理ボリューム (PV) のパーティション冒頭部分に書き込まれます。それらのパーティションを LVM ではない他の用途に使用する場合は、ボリュームの冒頭部分を削除しておくことをお勧めします。たとえば ボリュームグループ system と物理ボリューム /dev/sda2 が存在する場合、下記のコマンドで削除することができます。dd if=/dev/zero of=/dev/sda2 bs=512 count=1.

警告: 起動用のファイルシステム

起動中に使用するファイルシステム (ルートファイルシステムか、もしくは /boot パーティション) は LVM 論理ボリューム内に存在させてはなりません。その代わりに、通常の物理パーティション内に配置してください。

LVM について、詳しくはストレージ管理ガイドをお読みください。

2.2 LVM の設定

この章では、論理ボリュームマネージャ (LVM) の考え方と、その多目的な使い方について説明しています。YaST で LVM を利用した設定方法については、2.2.2 項「YaST を利用した LVM 設定」(56 ページ)をお読みください。

警告

LVM を使用すると、場合によってはデータ損失などのリスクを増加させてしまうことがあることにご注意ください。アプリケーションのクラッシュや電源の問題、およびコマンドの入力ミスなどのリスクも考えられます。そのため、LVM を実装したり再設定したりする際には、お使いのデータは必ず保存しておいてください。バックアップ無しでの作業は行なってはなりません。

2.2.1 論理ボリュームマネージャ

LVM は複数のファイルシステムに対応した、柔軟なハードディスク領域管理機能を提供します。これは当初のパーティション設定が終わったあと、システムを運用中の状態でハードディスク領域を部分的に変更したい、という要件が発生していたことから開発が始まりました。稼働中のシステムではパー

パーティション設定を変更することが難しいため、LVMでは仮想プール(ボリュームグループ、略してVGとも呼びます)というメモリ領域を用意することで、必要に応じて論理 ボリューム (LV) を作成できるようにしました。オペレーティングシステムはパーティションにアクセスする代わりに、これらのLVにアクセスします。ボリュームグループは複数のディスクから構成することができます。この方法により、LVMは物理的なディスク構成とは直接関係のない抽象機能を提供します。そのため、物理的なパーティションに比べ、安全かつ容易に部分的な変更を行なうことができるようになっています。物理的なパーティションに関する知識は、2.1.1項「パーティションの種類」(45 ページ)と2.1項「YaST パーティション設定の利用」(43 ページ)をそれぞれお読みください。

図 2.2 物理的なパーティションと LVM

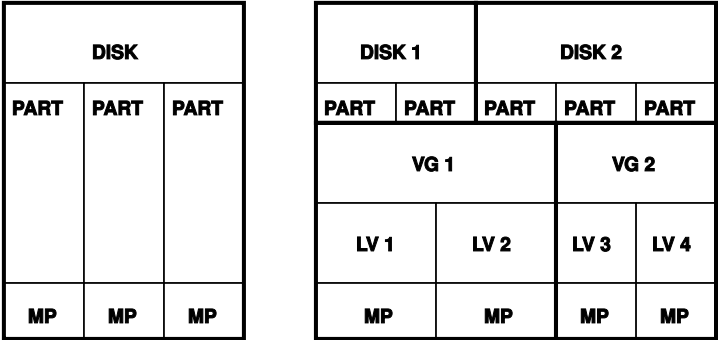


図2.2「物理的なパーティションと LVM」(54 ページ)では物理的なパーティション(左側)と LVM 論理ボリューム(右側)を比較したものです。左側では1台のハードディスクを3つの物理的なパーティション(PART)に分割していて、それぞれマウントポイント(MP)を設定することにより、オペレーティングシステムからのアクセスができるようになっています。右側では2台のハードディスクをそれぞれ3つの物理パーティションに分割しています。そこから2つの LVM ボリュームグループ(VG 1 と VG 2)を設定しています。VG 1 には DISK 1 から 2 つのパーティション、DISK 2 から 1 つのパーティションをそれぞれ設定しています。また、VG 2 には DISK 2 から残りの 2 つのパーティションを設定しています。LVM では、ボリュームグループ内に取り込まれたディスクパーティションのことを物理ボリューム(PV)と呼びます。ボリュームグループ内には4つのLV(LV 1 から LV 4 まで)を設定していて、オペレーティングシステムはそれぞれ設定されたマウントポイント経由でアクセスできるようになっています。異なるLV同士の境界は、パーティ

ションの境界と一致している必要はありません。この例ではLV 1とLV 2の境界をご覧ください。

LVMの機能:

- 複数のハードディスクやパーティションを、大きな論理ボリュームとして統合することができます。
- 構成が適切であれば、あるLV (たとえば /usr) の空き領域が枯渇した場合、サイズを拡張することができます。
- LVMを使用することで、稼働中のシステムに対してハードディスクやLVの追加を行なうことができます。ただし、ホットスワップ (活線挿抜) に対応したハードウェアが必要です。
- LVのデータを複数のPVに分散する "ストライプモード" を利用することができます。PVがそれぞれ異なるディスクに存在していれば、読み書きの性能を向上させることができます。これは RAID 0 と呼ばれます。
- スナップショット機能により、稼働中のシステムで矛盾のないバックアップを採取できるようになります (特にサーバ用途に使用する場合に便利な機能です)。

このような機能を持つLVMは、多用される家庭用PCや小規模サーバに便利な仕組みになっています。LVMはデータ量が増加し続けるユーザ (たとえばデータベースや楽曲のアーカイブ、ユーザのディレクトリなど) には適した仕組みで、物理的なハードディスクのサイズよりも大きなファイルシステムを作成することができます。また、LVMのもう1つの利点は、最大で256個までのLVを追加することができるという点です。ただしLVMは通常のパーティションとは異なる作業を行わなければなりません。LVMの設定手順や詳しい情報については、LVMの公式HOWTO <http://tldp.org/HOWTO/LVM-HOWTO/> をお読みください。

カーネルバージョン2.6以降では、LVMバージョン2を利用することができます。これは以前のバージョンのLVMと後方互換性があり、古いバージョンのボリュームグループについても管理できるようになっています。また、新しいボリュームグループを作成する際には、新しい形式で作成するか古い形式で後方互換性を保つかを選択することができます。なお、LVM2ではカーネルパッチを適用する必要がなく、2.6カーネル内蔵のデバイス Mapper を利用します。特別なパッチを適用していないカーネルではLVMバージョン

2にのみ対応しているため、下記の章ではLVMバージョン2を利用して話を進めるものとします。

2.2.2 YaST を利用した LVM 設定

YaST からの LVM 設定は、YaST の熟練者向けパーティション設定内の システムビューにある、**ボリューム管理**から行なうことができます(詳しくは2.1 項「YaST パーティション設定の利用」(43 ページ)をお読みください)。熟練者向けパーティション設定では既存のパーティションを編集したり削除したりすることができるほか、LVM で使用するために新しいパーティションを作成することもできます。LVM の作業は、ボリュームグループに領域を提供するため、PV を作成することからはじまります:

- 1 ハードディスクからハードディスクを選びます。
- 2 パーティションタブに切り替えます。
- 3 **追加** を押し、このディスクに作成する PV のサイズを 指定します。
- 4 パーティションをフォーマットしないを選択し、ファイルシステム ID に *0x8E Linux LVM* を選択します。このパーティションはフォーマットしないでください。
- 5 上記までの手順を、必要な物理ボリューム数だけ繰り返してください。

ボリュームグループの作成

お使いのシステムに既存のボリュームグループが存在しない場合、次にそれを追加します(図2.3「ボリュームグループの作成」(57 ページ)をご覧ください)。システムビュー内の **ボリューム管理** から、**ボリュームグループの追加** を押すことで、追加のボリュームグループを作成することもできます。通常は1つのボリュームグループを作成するだけで十分です。

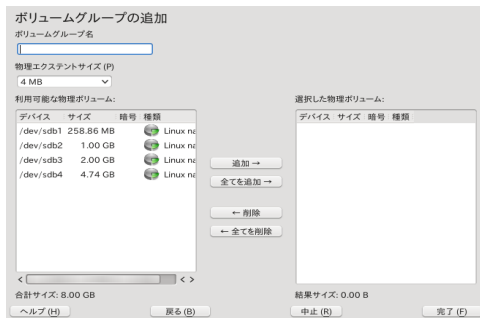
- 1 まずはボリュームグループの名前を入力します(例: system)
- 2 次に **物理エクステンツ** サイズを選択します。この値は、ボリュームグループ内での物理ブロックのサイズを指定します。ボリュームグループ内のディスク領域は、このサイズ単位で処理されます。

ティップ: 論理ボリュームとブロックサイズ

論理ボリュームに設定可能なサイズは、ボリュームグループのブロックサイズに依存します。既定では 4 MB になっているため、最大で 256 GB までの論理ボリュームを作成することができます。論理ボリュームを 256GB より大きく設定したい場合は、ブロックサイズを 8, 16, 32 MB のように大きなサイズに設定してください。

- 3 デバイスを選択して **追加** を押すと、ボリューム グループに物理ボリュームを追加することができます。デバイスを選択する際に **Ctrl** を押しながらか行なうと、複数の デバイスを選択することができます。
- 4 **完了** を押して閉じます。これでボリュームグループの設定が終わり、次の手順に進むことができますようになります。

図 2.3 ボリュームグループの作成



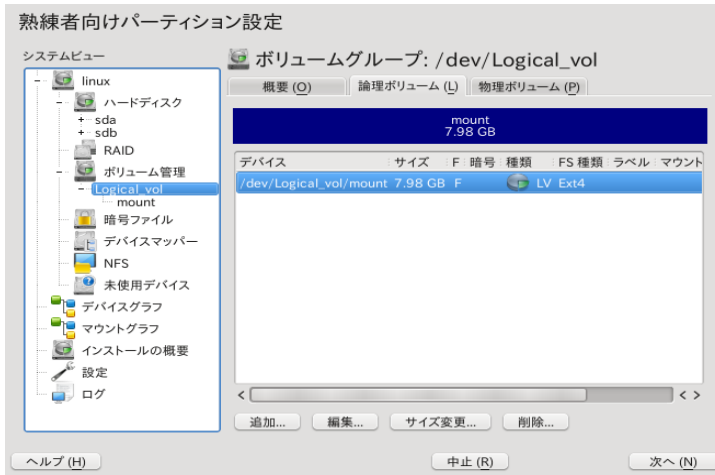
複数のボリュームグループを設定していてPVを追加したり削除したりしたい場合は、まず **ボリューム管理** 内からボリュームグループを選択してください。選択を行なったら **概要** タブに移動し、**サイズ変更** ボタンを押してください。その後表示されるウインドウで、選択したボリュームグループに対する **PV** を追加または削除することができます。

論理ボリュームの設定

物理ボリュームからボリュームグループを設定したら、次にオペレーティングシステムで論理ボリュームを設定します。作成したボリュームグループを選択し、**論理ボリューム** タブに移動してください。 **追加**, **編集**, **サイズ変更**, **削除** の各ボタンを利用し、存在するボリュームグループ全てを使用するまで

設定してください。少なくとも論理ボリュームには1つ以上のボリュームグループを設定する必要があります。

図 2.4 論理ボリューム管理



追加を押すと、ウィザード形式のポップアップが表示されます:

1. まずは論理ボリュームの名前を入力します。たとえば /home にマウントさせたい論理ボリュームであれば、それをそのまま説明する HOME などの名前を入力してください。
2. 次に論理ボリュームのサイズとストライプ数を指定します。1 台のハードディスクしか物理ボリュームに存在していない場合は、1 以上のストライプ数を設定することはできません。
3. 続いて論理ボリューム上で使用するファイルシステムと、マウントポイントを指定します。

ストライプとは、複数の物理ボリュームにまたがる論理ボリュームに対して設定するもので、データの書き込みを分散させるためのものです。これらの物理ボリュームが異なるハードディスクに存在している場合は、読み込みや書き込みの速度が向上します (RAID 0 に似た仕組みであるためです)。ただし、 n 個のストライプを作成するには、論理ボリュームが利用するハードディスク領域が n 個に分散できる場合に限られます。たとえば 2 つの物理ボリュームを利用している場合は、3 つのストライプは作成できません。

警告: ストライプ

現時点の YaST ではストライプに関する設定の妥当性を検証する機能が存在していません。そのため、ここで誤った設定を行なっても事前にはチェックされず、LVM を実際に作成する際にチェックされます。

これでお使いのシステムに LVM を設定することができました。これで作成した論理ボリュームを使用することができます。作成した論理ボリュームには適切なマウントポイントを設定してお使いください。完了を押すと YaST 熟練者向けパーティション設定に戻り、作業は完了です。

2.3 ソフトウェア RAID の設定

RAID (Redundant Array of Independent Disks) の目的は、複数のハードディスクパーティションを組み合わせることで1つの巨大な仮想ディスクを構成し、性能向上とデータの保全のいずれか、または両方を目指したしくみです。多くの RAID コントローラでは、IDE プロトコルよりも多数のディスクをより効率的に扱うことができるという理由で、SCSI プロトコルを用いています。また、SCSI プロトコルでは並列コマンド処理という機能も備えています。それ以外にも、IDE や SATA に対応する RAID コントローラもいくつか存在しています。ソフトウェア RAID はハードウェア RAID に比べ、コントローラを購入する追加コストが発生しないという点で有利ですが、CPU 時間を多く消費しメモリも利用してしまうため、高性能なコンピュータを構成する場合には不利な選択です。

openSUSE® では複数のハードディスクを1台のソフトウェア RAID システムに組み合わせる仕組みを備えています。RAID はシステム内の構成方法によって異なる目的や利点、特性をそれぞれ備えています。これらの違いは RAID レベルとしても知られています。

一般的な利用される RAID レベルは、下記のとおりです:

RAID 0

このレベルは、各ファイルのブロックを複数のハードディスクに分散させることで、性能の向上を目指すレベルです。実際のところはデータの冗長性が存在しないことから、厳密な意味での RAID ではありませんが、一般的には RAID 0 と呼ばれています。RAID 0 では2台以上のハードディスクを組み合わせ使用します。性能は向上しますが、いずれかのハー

ドディスクが故障すると、RAID システム全体が破壊される ことになります。

RAID 1

このレベルは、それぞれのハードディスクに同じデータを書き込むことで、お使いのデータの安全性を高めるレベルです。これは ハードディスクミラーリングとしても知られている 方法です。1 台のディスクが故障しても他のディスクにコピーが存在するため、そこから読み出すことができます。最後の 1 台のディスクが壊れる までは、お使いのデータは通常どおり読み出すことができます。ですが、ディスク障害が検出されず、読み出したデータだけが壊れるタイプの障害が発生した場合は、壊れたデータを壊れていないハードディスクにコピーしてしまいます。これによって本質的には同じデータ損失が発生することになります。コピー処理を行なうため、1 台のディスクを使用する場合に比べて性能が低下します(10 から 20 パーセント程度低下します) が、読み込み 性能については 1 台のハードディスクの場合よりも、それなりに速くなります。これは複製したデータを同時並行で検索するためのものです。一般に Level 1 は、1 台のハードディスクと比較して読み込み性能では 2 倍程度、書き込み性能では同等程度の速度が出るものとされています。

RAID 2 と RAID 3

これらは一般的な RAID 実装ではありません。レベル 2 では、データをブロック単位ではなくビット単位で分散させます。レベル 3 では、専用のパリティディスクと呼ばれるものとバイト単位での分散処理を提供していますが、複数の要求を同時に処理することができません。これらのレベルはほとんど使用されていません。

RAID 4

このレベルは RAID 0 のようなブロック単位の分散を行ないませんが、専用のパリティディスクを用意する点が異なります。いずれかのデータディスクに障害が発生すると、パリティデータからデータディスクを修復することができるようになっていきます。ただし、これによって書き込みの性能を落とす結果になってしまいます。

RAID 5

RAID 5 はレベル 0 と 1 を統合したもので、性能と冗長性の両方を 備えています。領域のサイズは、全てのハードディスクサイズの合計から 1 台分を引いたサイズになります。データは RAID 0 のようにハードディスク内に分散して書き込まれますが、さらに パリティ ブロックと呼ばれるものが、安全上の理由からいずれかの パーティション内に作成されます。パ

リティはXORと呼ばれる形式で作成するもので、これによりシステム障害が発生してもパリティブロックから元のデータを作り直すことができるようになっています。RAID 5では1台のディスクが故障してもデータに問題はありますが、複数台が同時に故障するとデータが修復できなくなります。データ損失を防ぐには、1台のハードディスクが故障した段階で、早急に交換することが必要になります。

その他の RAID レベル

上記以外にも RAID のレベルがいくつか存在しています (RAIDn, RAID 10, RAID 0+1, RAID 30, RAID 50 など)。これらの中にはハードウェアの製造元で独占的な実装を行なっているものもあります。これらのレベルはいずれも汎用的なものではないので、ここでは説明をしません。

2.3.1 YaST を利用した RAID 設定

YaST からの RAID 設定は、YaST の熟練者向けパーティション設定内から行なうことができます (詳しくは 2.1 項「YaST パーティション設定の利用」(43 ページ)をお読みください)。熟練者向けパーティション設定では既存のパーティションを編集したり削除したりすることができるほか、新しい RAID を作成することもできます:

- 1 ハードディスクからハードディスクを選びます。
- 2 パーティションタブに切り替えます。
- 3 追加を押し、このディスクに作成する RAID パーティションのサイズを指定します。
- 4 パーティションをフォーマットしないを選択し、ファイルシステム ID に *0xFD Linux RAID* を選択します。このパーティションはフォーマットしないでください。
- 5 上記までの手順を、必要な物理ボリューム数だけ繰り返してください。

RAID 0 と RAID 1 の場合は、少なくとも 2 つのパーティションが必要です; RAID 1 の場合は通常、2 つだけを指定します。RAID 5 の場合は、少なくとも 3 つのパーティションが必要です。また、いずれの RAID レベルであっても、同じサイズのパーティションを作成し、設定することをお勧めします。さらに RAID パーティションは、ディスクの故障によるデータ損失を防ぐた

め (RAID 1 と RAID 5 の場合)、または最大限の性能を引き出すため (RAID 0 の場合)、異なるディスク上に配置すべきものです。RAID で使用する全てのパーティションを作成したら、**RAID > RAID** の追加を押して RAID の設定を開始してください。

次のダイアログでは、RAID のレベルを 0, 1, 5, 6, 10 から選択します。そのあと、RAID システムで使用する全ての「Linux RAID」または「Linux native」パーティションを選択します。スワップパーティションや DOS パーティションは表示されません。

図 2.5 RAID パーティション



選択した RAID ボリュームに対して、未割り当てのパーティションを追加するには、対象のパーティションを選択して**追加**を押してください。ここでは RAID に使用する全てのパーティションを割り当ててください。割り当てを行なわないと、そのパーティションは未使用のままになります。全てのパーティションを割り当てたら、**次へ**を押して利用可能な RAID オプションを選択します。

手順の最後では、使用するファイルシステムと暗号化、その RAID ボリュームのマウントポイントをそれぞれ設定します。設定が全て終わったら、**完了**を押してください。熟練者向けパーティション設定では、デバイス `/dev/md0` と **RAID** と書かれたパーティションが表示されるはずです。

2.3.2 トラブルシューティング

RAID パーティションに障害が発生しているかどうかは、`/proc/mdstat` ファイルから確認してください。何らかの障害が発生していた場合は、Linux システムをシャットダウンし、障害の発生しているハードディスクを新しいものに交換してから、同じ方法でパーティションを設定してください。その後システムを起動し、`mdadm /dev/mdX --add /dev/sdX (X はお使いのデバイス識別子に置き換えてください)` コマンドを実行すると、RAID システムへの統合と再構築が始まります。

なお、再構築中でもデータにアクセスすることは可能ですが、RAID が完全に構築完了するまでの間は、性能面での問題に遭遇する可能性があります。

2.3.3 さらなる情報

ソフトウェア RAID に関する設定の詳細や、さらなる情報については、下記をお読みください:

- `/usr/share/doc/packages/mdadm/Software-RAID.HOWTO.html` (英語)
- <http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html> (英語)

Linux RAID メーリングリスト (英語) もご利用いただけます。 <http://marc.theaimsgroup.com/?l=linux-raid> をご覧ください。

パート II. ソフトウェアの管理と 更新

ソフトウェアのインストールと削除

YaSTのソフトウェア管理ツールを利用することで、追加したり削除したりしたいソフトウェアコンポーネントを検索することができるほか、全ての依存関係についても解決を行ないます。また、インストールメディア内に含まれていないパッケージをインストールするには、追加のソフトウェアリポジトリを設定して YaST に 管理させてください。さらに、ソフトウェアの更新について最新の状態になるよう管理するには、更新アプレットをお使いください。

お使いのシステムにおけるソフトウェアコレクションの管理は、YaST ソフトウェア 管理を利用して行ないます。この YaST モジュールは、Qt (KDE デスクトップ向け)、GTK+ (GNOME デスクトップ向け)、ncurses (テキストモードでの擬似的なグラフィカルインターフェイス向け)の3種類が利用できます。Qt と GTK+ については本章で、ncurses については 第10章 テキストモードでの YaST (161 ページ) でそれぞれ説明しています。

ティップ: ツールキットの種類変更

既定では、YaST はお使いのデスクトップ環境と同じツールキットを使用します (GNOME 環境では GTK+、KDE 環境では Qt)。このシステム全体での既定値 を変更するには、`/etc/sysconfig/yast2` ファイル内にある `WANTED_GUI` の項目を、それぞれ `qt` または `gtk` に変更してください。

システム全体で変更したくない場合は、利用したいツールキット種類をコマンドラインで 指定することもできます。それぞれコマンドラインから起動する際に `--gtk` または `--qt` のオプションを指定してください (例: `yast2 --gtk`)。

注記: 変更点の確認について

パッケージのインストールや更新、削除を行なう場合、ソフトウェア管理は変更点をすぐには適用することではなく、それぞれ受け入れるまたは適用のボタンで確認を行なったあとに適用作業を行ないます。また、YaSTは実施しようとする全ての作業を管理し、システムに適用する前に変更点を確認し、了承を得ようとしています。

3.1 用語定義

リポジトリ

パッケージやパッケージに関する追加情報(パッケージメタデータ)を含むローカルまたはリモートのディレクトリ。

(リポジトリの)別名

様々な `zypper` コマンドで使用するリポジトリの短縮名。リポジトリを追加する際にユーザ側で命名するもので、システム内で唯一のものでなければなりません。

製品

たとえば `openSUSE®` のように、製品全体を指す用語。

パターン

それぞれ特別な用途に必要となる、インストール可能なパッケージの一覧。たとえば Base System は `openSUSE` の基本的なシステムを構成するもので、GNOME Base System は `GNOME` デスクトップ環境を動作させるのに必要な全てのパッケージを含んでいます。特定の用途に特化し、インストールが可能なパッケージグループ。たとえばラップトップパターンにはモバイルコンピューティング環境に必要な全てのパッケージが含まれています。また、パターンにはパッケージの依存関係(必要とするものや推奨するもの)が設定されていて、複数のパッケージをインストールするよう事前に選択します。これにより、パターンをインストールするとお使いのシステムで特定の目的を達成するのに最も重要なパッケージが用意されるようになります。ただし、必ずしもパターンに含まれる全てのパッケージがインストールに必要というわけではなく、細かい要件や希望に合わせて手作業でパッケージを選択したり選択を外したりする必要がある場合もあります。

パッケージ

特定のプログラム向けのファイル一式を含む、rpm形式の圧縮ファイル。

修正

1つまたはそれ以上のパッケージから構成されるもので、**deltarpm**の形式で適用することができるもの。修正を適用する前にはインストールされていなかった新しい依存関係が提示される場合もあります。

解決方法

製品やパターン、パッケージや修正のいずれかを指す汎用の用語。多くはパッケージや修正を指しています。

deltarpm

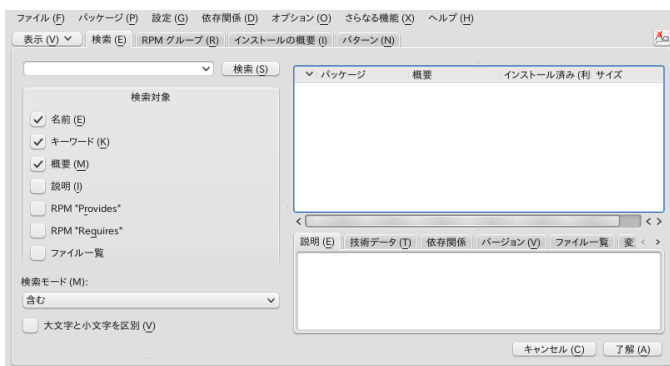
特定のパッケージに対して、2つのバージョン間でのバイナリ差分だけを含むもの。これによりダウンロードサイズを最も小さくすることができます。インストール作業を行なう前に、ローカルマシン側で完全な**RPM**パッケージを再構築します。

パッケージの依存関係

あるパッケージは、共有ライブラリなどの他のパッケージに依存しています。言い換えると、あるパッケージは他のパッケージを必要としているとも言えます。もしも必要なパッケージが利用できない場合、そのパッケージはインストールすることができません。また、それを満たさなければならない依存関係(パッケージ要件)に加え、いくつかのパッケージでは他のパッケージを推奨する場合があります。これら推奨パッケージは、実際にそれらが存在した場合にのみインストールが行なわれます。存在しなかった場合は単に無視され、元のパッケージはそのままインストールされます。

3.2 KDE インターフェイス (Qt) の使用

KDE や **icewm**、もしくはその他のデスクトップ環境を利用している場合、既定では **YaST** の **Qt** インターフェイスが利用されます。また、**YaST** のリモートの端末から実行した場合も **Qt** インターフェイスになります。**YaST** コントロールセンターから、**ソフトウェア>ソフトウェア管理**を選択することで、ソフトウェアマネージャを起動することができます。



3.2.1 パッケージやパターンの検索ビュー

YaST ソフトウェア管理では、その時点で有効化されているリポジトリから提供される任意のパッケージやパターンをインストールすることができます。ビューやフィルタを設定し、ソフトウェアを簡単な手順で検索することもできます。なお、検索ビューは本ウインドウにおける既定のビューになっています。表示を選択すると表示されるドロップダウンリストから、いずれかを選択してビューを切り替えてください。選択されたビューは新しいタブとして表示されます。

パターン

お使いのシステムで利用可能な全てのパターンが表示されます。

パッケージグループ

全てのパッケージをグループごとに一覧表示します。グラフィクス、プログラミング、セキュリティなどが用意されています。

RPM グループ

全てのパッケージを機能ごとに分類し、そのグループ／サブグループごとに一覧表示されます。たとえば *Networking > Email > Clients* などのグループがあります。

言語

新しいシステム言語を追加するのに必要な全てのパッケージが表示されます。

リポジトリ

パッケージをリポジトリごとに一覧表示します。複数のリポジトリを選択したい場合は、リポジトリを選択する際に **Ctrl** キーを押しながら選択してください。「疑似リポジトリ」として、お使いのシステムに既にインストールされている全てのパッケージを含む **@System** リポジトリも存在します。

検索

特定の単語や語句を入力してパッケージを検索することができる表示です。検索したい単語や語句を入力して **Enter** を押してください。検索結果を絞り込むには **検索対象** と **検索モード** を設定してください。たとえば、検索したいパッケージ名はわからないがアプリケーションの名前がわかる場合、検索時に **説明** を含めるように設定してみてください。

インストールの概要

既にインストールや更新、削除を行ないたいパッケージを選択済みの場合は、この表示で **了解後** に適用される変更点が表示されます。この表示で特定の状態に設定したパッケージだけを表示するには、それぞれ表示内にあるチェックボックスを設定してください。なお、**Shift+F1** を押すと、状態フラグの説明が表示されます。

ティップ: 有効なリポジトリに属していないパッケージの検索

有効なリポジトリに属していない全てのパッケージを一覧表示するには、**ビュー > リポジトリ > @System** を選択し、**セカンダリフィルタ > メンテナンスされていないパッケージ** を選択してください。これはたとえば、リポジトリの登録を削除していて、そのリポジトリ経由でインストールしたパッケージが存在しないことを確認する場合などに便利です。

3.2.2 パッケージやパターンのインストールと削除

特定のパッケージは共有ライブラリなどの他のパッケージに依存しています。その一方、パッケージによっては2つのパッケージをシステム内に共存させることができないものもあります。YaSTではこれらの問題をできる限り自動的に解決します。選択の結果、自動的に解決できない依存関係の矛盾が発生したりした場合は、3.2.4項「ソフトウェアの依存関係の確認」(75 ページ)に示されている手順を利用して手作業で解決する必要があります。

注記: パッケージの削除について

パッケージを削除する際、既定では YaST は選択したパッケージのみを削除します。指定したパッケージを削除したことによって不要になったパッケージについても削除したい場合は、**オプション> パッケージを削除する際に 清掃を行なう** のチェックを付けてください。

- 1 パッケージの検索については3.2.1項「**パッケージやパターンの検索ビュー**」(70 ページ)をお読みください。
- 2 見つかったパッケージは右側の枠内に表示されます。インストールまたは削除を行なうには、パッケージをマウスの右ボタンで選択して、表示されるメニューから **インストールする** または **削除** を選んでください。選択したい項目が現れない場合は、パッケージ名の左側に表示されているパッケージ状態をご確認ください。パッケージ状態について、詳しくは **Shift + F1** を押して表示されるヘルプをお読みください。

ティップ: 一覧にある全てのパッケージに対する処理指定

右側に現われた全てのパッケージに対して同じ処理を指定するには、**メニュー バー内の パッケージ> この一覧内のすべて** 以下の各項目を選択してください。

- 3 パターンをインストールするには、パターン名をマウスの右ボタンで選択して、表示されるメニューから **インストール** を選択します。
- 4 なお、パターンそのものを削除することはできません。その代わり、削除したいパターン内のパッケージを選択し、削除を行なってください。
- 5 複数のパッケージを選択したい場合は、上記までに記した手順を繰り返して実施してください。
- 6 変更を適用する前に、**表示> インストールの概要** を選択すると、変更点を閲覧したり変更したりすることができます。既定ではインストールされる全てのパッケージが一覧表示されます。
- 7 特定のパッケージに対して状態を元に戻したい場合は、パッケージをマウスの右ボタンで選択し、表示されたメニューからいずれかを選択します。パッケージが削除または更新されるように指定していたものを元に戻す場合は **保持** を、インストールを行なうように指定していたものを **元に戻す**

場合は インストールしないを選択してください。全ての変更点を取り消してソフトウェア管理を閉じるには、キャンセルを押して 中止 を押します。

- 8 全ての作業が完了したら 了解 ボタンを押し、変更点を 適用 します。
- 9 YaST で他のパッケージへの依存を検出した場合は、追加でインストールまたは 更新／削除されるパッケージの一覧が表示されます。 続行 を押して先に進んでください。

選択した全てのパッケージをインストールまたは更新／削除すると、YaST ソフトウェア管理は自動的に終了します。

注記: ソースパッケージのインストール

YaST ではソースパッケージのインストールを行なうことはできません。その代わりに、コマンドラインツール zypper をお使いください。詳しくは「ソースパッケージのインストール」(109 ページ)をお読みください。

3.2.3 パッケージの更新

個別のパッケージを更新する代わりに、インストール済みの全てのパッケージや 特定のリポジトリにある全てのパッケージを更新することができます。一括でパッケージを更新する場合は、下記の状況を考慮する必要があります:

- パッケージを提供するリポジトリの優先順位
- パッケージのアーキテクチャ (たとえば x86_64, i686, i586 など)
- パッケージのバージョン番号
- パッケージの製造元

更新候補を選択するのに最も重要な点がどれであるかは、それぞれ選択する更新 オプションに依存します。

- 1 全てのインストール済みパッケージを更新するには、メインメニューから パッケージ > すべてのパッケージ > 新しいバージョンがあれば更新する を選択します。

全てのリポジトリに対して更新候補の存在を確認しますが、下記のポリシーに従って確認を行ないます: YaST では、まずインストールされているものと同じアーキテクチャおよび製造元を持つパッケージに限定して検索を行ないます。検索が成功した場合は、これらの提供元からの「ベスト」な更新候補を下記に示す方法で選択します。検索が成功しなかった場合、つまり同じ製造元で類似のパッケージが見つからない場合は、同じアーキテクチャを持つ全てのパッケージに検索範囲を広げます。それでも類似のパッケージが見つからない場合は、全てのパッケージが検討範囲に組み込まれ、その中での「ベスト」な更新候補が下記に示す方法で選択されます:

1. リポジトリの優先順位: 最も高い優先順位を設定しているリポジトリからのパッケージを優先して使用します。
2. この選択で複数のパッケージが存在した場合は、「ベストな」アーキテクチャのものを選択します (インストール済みのものと同じアーキテクチャ、 もしもそれが存在しない場合は `x86_64 > i686 > i586` の順)。

インストール済みのパッケージよりも高いバージョン番号を持つパッケージが見つかった場合は、インストールされているパッケージは更新されて選択された更新候補に置き換えられます。

このオプションではアーキテクチャや製造元がインストール済みのものから変わらないように試しますが、特定の条件下において変更される場合もあります。

注記: 強制的な更新

パッケージ > 全てのパッケージ > 強制的に更新する を選択した場合、基本的に同じ判断条件が適用されますが、候補パッケージを強制的にインストールします。そのため、このオプションを選択すると、パッケージのダウングレードが発生する場合があります。

- 2 一括更新を行なう際、特定のリポジトリからのパッケージに限定して行なうには、下記の手順を行ないます:

2a 3.2.1項「パッケージやパターンの検索ビュー」(70 ページ)の手順に従い、更新したいリポジトリを選択します。

- 2b** ウィンドウの右側で、システムパッケージを、このリポジトリ内にあるバージョンに切り替えるを選択します。これにより、YaST はパッケージを入れ替える際に製造元の変更を許可するようになります。

了解で先に進むと、全てのインストール済みパッケージを、指定したリポジトリからのものに置き換える処理を行ないます。これにより製造元やアーキテクチャが変更されるほか、パッケージによってはダウングレードが行なわれる場合があります。

- 2c** 上記での切り替えを取り消すには、システムパッケージを、リポジトリ内にあるバージョンに切り替えるのをキャンセルするを選択します。なお、キャンセルは 了解 ボタンを押すまでの間のみ利用できます。

- 3** 変更を適用する前に、表示 > インストールの概要を選択すると、変更点を閲覧したり変更したりすることができます。既定では状態が変更される全てのパッケージが一覧表示されます。
- 4** 全てのオプションを要件にあわせて変更したら、最後に 了解 ボタンを押すと一括更新が始まります。

3.2.4 ソフトウェアの依存関係の確認

多くのパッケージは他のパッケージに依存しています。たとえばあるパッケージが共有ライブラリを使用する場合、その共有ライブラリを提供するパッケージに依存することになります。一方、パッケージによってはお互いに共存できず、矛盾が生じるものもあります(たとえば `sendmail` や `postfix` などのメール転送エージェントは同時に1つしかインストールできません)。ソフトウェアをインストールしたり削除したりする場合は、ソフトウェアマネージャは未解決の依存関係や矛盾が残っていないことを確認してから作業を行ないます。これにより、システムの一貫性を保つ仕組みになっています。

依存関係を解決するための選択肢が1つしか存在しない場合は、それらは自動で処理されます。複数の解決方法が存在する場合や、解決方法に製造元の変更やアーキテクチャの変更が含まれる場合は、それらは手動で解決する必要があります。了解 ボタンでインストールを開始すると、自動解決器で解決された全ての処理について確認ウィンドウが表示されます。

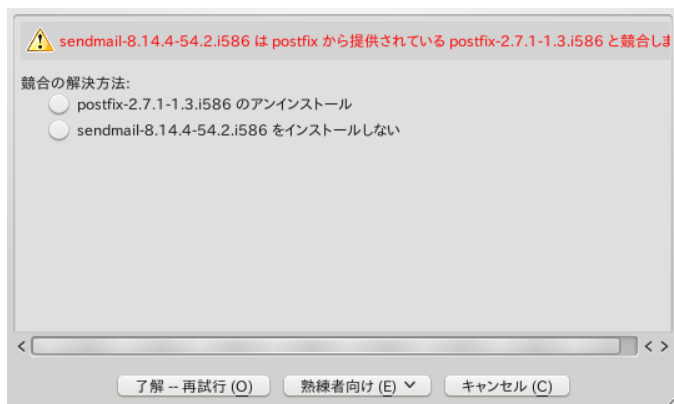
既定では依存関係は自動的に確認が行なわれます。確認はパッケージ状態を変更するたび(たとえばパッケージをインストールするよう選択したり削除するよう選択したりするごと)に行なわれます。これは一般に便利な方法ではありますが、手動で依存関係の矛盾を解決したい場合などには使いにくいものになっています。これを無効化するには、**依存関係> 自動確認**のチェックを外してください。依存関係の確認を手動で行なう場合は、**依存関係> 今すぐ確認**から行なうことができます。また、一貫性の確認は**了解**を押したタイミングでも行なわれます。

パッケージの依存関係を確認するには、パッケージをマウスの右ボタンで選択し、**解決器の情報表示**を選択してください。依存関係を表わすマップが表示されます。既にインストール済みのパッケージについては、緑色の枠で表示されます。

注記: 手動でのパッケージ矛盾解決

知識と経験のあるユーザでない限り、パッケージの矛盾を解決するに際して YaST の提案に従うのが適切です。そうでない場合は解決できなくなってしまう場合があります。それぞれ行なった変更は新しい矛盾を引き起こす場合があるため、解決しようとしても矛盾が続々と増えてしまう場合があります。このような場合はソフトウェア管理で **キャンセル** や **中止** ボタンを押し、最初からやり直すことをお勧めします。

図 3.1 ソフトウェア管理での矛盾管理



3.3 GNOME インターフェイスの使用 (GTK+)

YaST GTK+ インターフェイスは、デスクトップ環境に GNOME や XFCE を使用している場合に既定で動作するインターフェイスです。YaST コントロールセンターから **ソフトウェア > ソフトウェア管理** を選択することで起動することができます。



3.3.1 パッケージやパターンの検索ビュー

パッケージを検索するのに最も簡単な方法は、ソフトウェアマネージャ内の右上にある検索項目に入力を行なうことです。検索条件を入力して **Enter** を押してください。既定ではパッケージ名と概要を対象にして検索します。検索対象を変更するには、検索項目のアイコンを選択してからマウスの右ボタンを押し、たとえばファイルで検索するなど必要な対象を選択してください。

ソフトウェアマネージャでは、パッケージ一覧を表示する際に様々なビューやフィルタを用意しています。左上にあるプルダウンメニューからそれぞれ下記のものを利用することができます：

グループ

既定のビューでは、たとえば 管理ツール、グラフィクス、プログラミング、セキュリティ のようにグループごとに並べられています。

RPM グループ

グループやサブグループの形で、パッケージが機能別に一覧表示されます。たとえば *Networking > Email > Clients* などがあります。

リポジトリ

パッケージをリポジトリごとに表示するためのフィルタです。複数のリポジトリを選択するには、リポジトリをマウスの左ボタンで選択する際に **Ctrl** キーを押しながら選択してください。また、「擬似的なりポジトリ」である *@System* では、現時点でインストール済みの全パッケージを示します。

利用可能なリポジトリを追加／編集／削除するには、リポジトリの編集ボタンを押します。

パターン

お使いのシステムにインストール可能な全てのパターンが表示されます。

Languages

新しいシステム言語を追加するのに必要な全てのパッケージを一覧表示するためのフィルタです。

ダイアログの右下にある枠では、それぞれ インストール済み、未インストール、アップグレード可能のそれぞれの状態でフィルタを設定することができます。任意の状態を選択した場合は、状態に関係なく設定されている全てのリポジトリから提供される全てのパッケージが表示されます。

3.3.2 パッケージのインストールと削除

特定のパッケージは共有ライブラリなどの他のパッケージに依存しています。その一方、パッケージによっては 2 つのパッケージをシステム内に共存させることができないものもあります。YaST ではこれらの問題をできる限り自動的に解決します。選択の結果、自動的に解決できない依存関係の矛盾が発生したりした場合は、3.2.4 項「ソフトウェアの依存関係の確認」(75 ページ)に示されている手順を利用して手作業で解決する必要があります。

- 1 3.3.1項「パッケージやパターンの検索ビュー」(77 ページ)に書かれている方法で、まずはパッケージを検索します。
- 2 見つかったパッケージは右側の枠内に表示されます。この状態からさらにパッケージの状態(任意の状態、未インストール、インストール済み、アップグレード可能)で検索結果を絞り込むには、それぞれダイアログの左下にある枠内から選択してください。また、パッケージについての詳細を表示するには、一覧内からパッケージを選択してください。パッケージについてはバージョンや著者、変更ログがウインドウの右下に表示されます。インストール可能なパッケージは *Available* タブを、アップグレードや削除可能なパッケージは *Upgrades* と *Installed* タブを選択するとそれぞれ表示されます。インストールや削除、アップグレードを行なうには、それぞれ右下のボタンから指定してください。

パッケージをインストールまたは再インストール、もしくは削除やアップグレードのために選択するには、パッケージを選んでマウスの右ボタンを押し、表示されたメニューから動作を選択してください。

ティップ: 一覧にある全てのパッケージに対する処理指定

右側に現われた全てのパッケージに対して同じ処理を指定するには、パッケージを選択してマウスの右ボタンを押すと表示されるメニューから、**全て選択**を選び、再度右ボタンを押して処理を選択してください。

- 3 パターンをインストールするには、パターンを選んでマウスの右ボタンを押し、インストールを押してください。
- 4 なお、パターンそのものを削除することはできません。その代わり、削除したいパターン内のパッケージを選択し、削除を行なってください。
- 5 複数のパッケージを選択したい場合は、上記までに記した手順を繰り返し実施してください。
- 6 変更点をシステムに適用する前に、ダイアログの下部にある **すべての変更を閲覧**のリンクを押すことで、変更点の再確認や修正を行なうことができます。既定では状態を変更する全てのパッケージが表示されます。
- 7 パッケージの変更を取り消すには、黄色い矢印が描かれた **取り消し** ボタンを押します。確認作業を完了するには、**閉じる** を押します。

- 8 インストールや削除を行ないたいパッケージの選択が完了したら、最後に **適用** ボタンを押すと変更点を反映させることができます。
- 9 YaSTが他のパッケージへの依存関係を発見すると、追加でインストールまたは更新、もしくは削除されるパッケージの一覧が表示されます。 **適用** ボタンを押して確認を行なってください。

選択した全てのパッケージがインストールや更新、削除されると、YaST ソフトウェアマネージャは自動的に終了します。

注記: ソースパッケージのインストール

YaST ではソースパッケージのインストールを行なうことはできません。その代わりに、コマンドラインツール `zypper` をお使いください。詳しくは「ソースパッケージのインストール」(109 ページ)をお読みください。

3.3.3 パッケージの更新

個別のパッケージを更新する代わりに、インストール済みの全てのパッケージや特定のリポジトリにある全てのパッケージを更新することができます。一括でパッケージを更新する場合は、下記の状況を考慮する必要があります:

- パッケージを提供するリポジトリの優先順位
- パッケージのアーキテクチャ (たとえば `x86_64`, `i686`, `i586` など)
- パッケージのバージョン番号
- パッケージの製造元

更新候補を選択するのに最も重要な点がどれであるかは、それぞれ選択する更新 オプションに依存します。

- 1 全てのインストール済みパッケージを更新する (より高いバージョンに更新する) には、右下の枠内から **アップグレード可能** を選択します。
- 2 表示された全てのパッケージを更新するには、**全てをアップグレード** を押します。

公式な修正が提供されているアップグレード可能なパッケージのみを更新するには、 **アップグレード修正** を押します。これらのパッケージには、それらのバージョン番号に対して修正タグが付けられています。このオプションの選択は、第4章 **YaST オンライン更新** (89 ページ) で示されている YaST のオンライン更新と同じ意味になります。

最後に修正を適用してから新しい修正が発行されていない場合、ボタンは無効化され押すことができないようになります。

- 3 一括更新を行なう際、特定のリポジトリからのパッケージに限定して行なうには、下記の手順を行ないます:

3a 3.3.1項「**パッケージやパターンの検索ビュー**」 (77 ページ) の手順に従い、更新したいリポジトリを選択します。

3b ウィンドウの右側で、システムパッケージを、このリポジトリ内にあるバージョンに切り替えるを選択します。これにより、YaST はパッケージを入れ替える際に製造元の変更を許可するようになります。

全てのインストール済みパッケージを、指定したリポジトリからのものに置き換える処理を行ないます。これにより製造元やアーキテクチャが変更されるほか、パッケージによってはダウングレードが行なわれる場合があります。

- 4 変更点をシステムに適用する前に、ダイアログの下部にある **すべての変更を閲覧** のリンクを押すことで、変更点の再確認や修正を行なうことができます。既定では状態を変更する全てのパッケージが表示されます。
- 5 特定のリポジトリ内にあるパッケージへの置き換えを取り消すには、それぞれのパッケージの右側にある **取り消し** ボタンを押します。
- 6 全てのオプションを要件にあわせて変更したら、最後に **適用** ボタンを押すと一括更新が始まります。

3.3.4 ソフトウェアの依存関係の確認

多くのパッケージは他のパッケージに依存しています。たとえばあるパッケージが共有ライブラリを使用する場合、その共有ライブラリを提供するパッケー

ジに依存することになります。一方、パッケージによってはお互いに共存できず、矛盾が生じるものもあります (たとえば `sendmail` や `postfix` などのメール転送エージェントは同時に1つしかインストールできません)。ソフトウェアをインストールしたり削除したりする場合は、ソフトウェアマネージャは未解決の依存関係や矛盾が残っていないことを確認してから作業を行いません。これにより、システムの一貫性を保つ仕組みになっています。

依存関係を解決するための選択肢が1つしか存在しない場合は、それらは自動で処理されます。複数の解決方法が存在する場合や、解決方法に製造元の変更やアーキテクチャの変更が含まれる場合は、それらは手動で解決する必要があります。適用ボタンでインストールを開始すると、自動解決器で解決された全ての処理について確認ウインドウが表示されます。

既定では依存関係は自動的に確認が行なわれます。確認はパッケージ状態を変更するたび (たとえばパッケージをインストールするよう選択したり削除するよう選択したりするごと) に行なわれます。これは一般に便利な方法ではありますが、手動で依存関係の矛盾を解決したい場合などには使いにくいものになっています。これを無効化するには、依存関係 > 自動確認のチェックを外してください。依存関係の確認を手動で行なう場合は、依存関係 > 今すぐ確認から行なうことができます。また、一貫性の確認は適用を押したタイミングでも行なわれます。

注記: 手動でのパッケージ矛盾解決

知識と経験のあるユーザでない限り、パッケージの矛盾を解決するに際して YaST の提案に従うのが適切です。そうでない場合は解決できなくなってしまう場合があります。それぞれ行なった変更は新しい矛盾を引き起こす場合があります。このような場合はソフトウェア管理で キャンセル や 中止 ボタンを押し、最初からやり直すことをお勧めします。

図 3.2 ソフトウェアマネージャでの矛盾管理



3.4 ソフトウェアリポジトリとサービスの管理

サードパーティ製のソフトウェアをお使いのシステムに追加したい場合は、追加のソフトウェアリポジトリを設定します。既定では openSUSE-DVD 11.3 のような製品リポジトリと、インストール時に自動で設定された更新リポジトリが設定されています。また、初期段階で選択した製品にもよりますが、翻訳や辞書などの個別の言語アドオンリポジトリが設定されている場合もあります。

リポジトリを管理するには、YaST を起動して **ソフトウェア > ソフトウェアリポジトリ** を選択します。すると設定されたソフトウェアリポジトリのダイアログが表示されます。ここでは右端にある **表示** から **全てのサービス** を選

択することで、サービスと呼ばれる購読についても管理を行なうことができます。この章でのサービスとは、リポジトリインデックスサービス (RIS) と呼ばれるもので、1つまたは複数のソフトウェアリポジトリを提供するものです。これらのサービスは管理者や製造元が動的に変更することができます。

それぞれのリポジトリには、リポジトリの内容(パッケージ名、バージョンなど)を説明するファイルがあります。これらのリポジトリ説明ファイルはローカルキャッシュ内にダウンロードされ、YaSTが使用します。また、リポジトリの正当性を確実にするため、リポジトリの管理者がGPG鍵を利用して署名することもできます。新しいリポジトリを追加する際には、YaSTがその鍵をインポートするかどうかを尋ねる機能を提供します。

警告: 外部のソフトウェアソースの信頼

お使いのリポジトリ一覧に外部のソフトウェアリポジトリを追加する前に、あらかじめそのリポジトリが信頼できるものであることをご確認ください。**openSUSE** ではサードパーティが提供するソフトウェアリポジトリからインストールしたソフトウェアについて、潜在的なものを含むいかなる問題に対しても責任を負いません。

3.4.1 ソフトウェアリポジトリの追加

リムーバブルメディア (CD や DVD のほか、USB マスストレージなど) のほか、ネットワーク上にあるリポジトリを追加することもできます。

YaST の設定されたソフトウェアリポジトリダイアログからリポジトリを追加するには、下記の手順で行ないます:

- 1 **追加** を押します。
- 2 お使いのリポジトリに該当する **メディア種類** を選択します:

多くの場合、既定のオプションである **URL の指定** を選択します。

リムーバブルメディアからリポジトリを追加するには、それぞれ適切なオプションを選択してメディアを挿入するか、もしくはマシンに USB デバイスを接続します。

- 3 なお、リポジトリの説明をダウンロードを選択することもできます。このオプションのチェックを外した場合、YaSTは必要に応じて後から自動的にダウンロードを行いません。次へを押して進めてください。
- 4 ネットワークからリポジトリを追加した場合は、それぞれ問い合わせのあった情報を入力します。次へを押して進めてください。
- 5 追加したリポジトリによって異なりますが、それぞれ署名に使用するGPG鍵のインポートに関する問い合わせや、ライセンスへの同意の問い合わせが行なわれます。

これらのメッセージを確認すると YaST はメタデータをダウンロードして解析し、設定されたソフトウェアリポジトリの一覧に追加されます。

- 6 なお、必要であれば 3.4.2 項「リポジトリ属性の管理」(86 ページ)で示されている手順でリポジトリのプロパティを設定することができます。設定ダイアログを閉じるには **OK** を押してください。

これで 3.2 項「KDE インターフェイス (Qt) の使用」(69 ページ) または 3.3 項「GNOME インターフェイスの使用 (GTK+)」(77 ページ) の手順から、追加したリポジトリのパッケージをインストールできるようになります。

なお、YaST ではコミュニティリポジトリを追加するオプションも提供しています。これは Mozilla リポジトリ (Firefox や Thunderbird の最新版パッケージを提供するリポジトリ) などの、openSUSE® ビルドサービスのリポジトリを追加したい場合に便利です。

手順 3.1 コミュニティリポジトリの設定

- 1 YaST を起動し、ソフトウェア > ソフトウェアリポジトリを選択します。
- 2 **追加** を押します。
- 3 コミュニティリポジトリを選択し、次へを押します。
- 4 事前に設定されたりポジトリの一覧が表示されますので、追加したいリポジトリのチェックボックスに印を付けます。たとえば Mozilla リポジトリの場合は、openSUSE ビルドサービス - Mozilla のような名前が掲載されています。

OKで確認を行ないます。

- 5 GnuPG 鍵の確認メッセージが表示されますので、取り込みを選択します。複数のリポジトリを選択した場合は、それぞれのリポジトリに対してそれぞれ GnuPG 鍵の確認を行なってください。
- 6 新しいリポジトリが設定されたソフトウェアリポジトリ内に表示されるようになります。OKを押してソフトウェアリポジトリの設定を完了してください。

3.4.2 リポジトリ属性の管理

ソフトウェアリポジトリで表示される設定されたソフトウェアリポジトリでは、それぞれ下記の情報を 変更することができます:

状態

リポジトリはそれぞれ 有効または 無効を設定することができます。インストールの際は、有効に設定されたリポジトリからのもののみをインストールすることができます。一時的にリポジトリを使用しないようにするには、無効に設定してください。完全にリポジトリを削除したい場合は、無効化するのではなく **削除**を行なってください。

更新

リポジトリを更新すると、そのリポジトリが保持する内容情報(パッケージ名、バージョンなど)を YaST が使用するローカルキャッシュに保存する処理を行ないます。CD や DVD のような固定のリポジトリであれば、この作業は 1 回だけ実施すれば十分ですし、逆にしばしば更新されるようなリポジトリであれば、定期的に更新する必要があります。リポジトリのキャッシュを最新のものに維持する最も簡単な方法は、**自動更新**を有効にすることです。もちろん **更新**ボタンから手動で更新させることもできます。

ダウンロード済みのパッケージを保存

遠隔にあるリポジトリから提供されるパッケージは、インストールする前にダウンロードの処理を行ないます。既定ではインストールが成功すると、ダウンロードされたパッケージは削除されます。ダウンロード済みのパッケージを保存を選択すると、このようなダウンロード済みパッケージの削除を行なわなくなります。ダウンロード先は /etc/zypp/zypp.conf で設定します。既定では /var/cache/zypp/packages です。

優先度

リポジトリの優先度は1から200までの範囲の数値で設定するもので、1が最も高い優先度を意味します。YaSTから追加したリポジトリの場合、既定で99が設定されます。また特定のリポジトリに対して特に優先度を気にしない場合、0を設定してリポジトリに対して既定の優先度(99)を設定することもできます。パッケージが複数のリポジトリから提供される場合、より高い優先度を持つリポジトリからのパッケージが優先的に処理されます。これはローカルリポジトリ(たとえばDVD)を高い優先度設定しておく、インターネットからの不要なダウンロードを防ぐことができるようになります。

重要項目: 優先度とバージョン

どのような場合であっても、たとえより高い(新しい)バージョンのものも存在していても、最も高い優先度を持つリポジトリが優先的に処理されることになります。従って更新リポジトリは最も高い優先度設定しておかなければなりません(既定では20に設定されます)。そうでない場合は、オンライン更新を行っても最新のバージョンがインストールされず、古いバージョンをインストールしてしまうことになります。

また、openSUSEに同梱されているプログラムの最新版を提供するリポジトリ(たとえば最新のKDEやGNOMEバージョン)を追加する場合は、標準のリポジトリよりも高い優先度を設定していることをご確認ください。高い優先度を設定しないと、既定ではそれらのリポジトリが提供するパッケージがインストールされなくなってしまうです。

名前と URL

リポジトリの名前やURLを変更するには、一覧から変更したいリポジトリをマウスの左ボタンで選択して、編集ボタンを押します。

3.4.3 リポジトリ鍵の管理

ソフトウェアパッケージが途中経路などで改変されてしまうことを防ぐため、リポジトリ管理者の提供するGPG鍵でパッケージに対して電子署名を行なうことがあります。リポジトリを追加する際にはYaSTで署名鍵の取り込み処理について確認メッセージが表示されます。他のGPG鍵などで表示された鍵が正しく、改変されていないことを確認してください。鍵が変更されている

ような場合は、リポジトリに何らかのトラブルが発生している場合が考えられますので、鍵の変更について確証が得られるまでの間、そのインストール元を無効化しておく必要があります。

取り込み済みの全ての鍵を管理するには、設定されたソフトウェアリポジトリ モジュール内の **GPG 鍵** ボタンを押します。それぞれ情報を読むには、マウスで鍵を選択してください。また、**追加** で新しい鍵の追加を、**編集** や **削除** でそれぞれ既存の鍵の編集や削除を行なうことができます。

YaST オンライン更新

openSUSE ではお使いの製品に対して、ソフトウェアのセキュリティ更新が継続して提供されています。既定では更新アプレットを利用してシステムを最新の状態に保つようになっています。更新アプレットについて、詳しくは項「更新によるシステム維持」(第3章 ソフトウェアのインストール／削除／更新, ↑スタートアップ)をお読みください。本章では、もう一つのソフトウェアパッケージ更新ツールである YaST オンライン更新について述べています。

openSUSE® に対する最新の更新は、更新ソフトウェアリポジトリから提供されます(インストール作業時に自動で設定されます)。もちろん信頼できる更新リポジトリがあれば、手動で追加することもできます。リポジトリを追加したり削除したりするには、YaST から **ソフトウェア > ソフトウェア リポジトリ** を選択し、リポジトリマネージャを起動してください。リポジトリマネージャについて、詳しくは 3.4 項「ソフトウェアリポジトリとサービスの管理」(83 ページ)をお読みください。

openSUSE ではお使いの製品に対して、様々な種類の更新を提供しています:

セキュリティ更新

セキュリティ上の致命的な脆弱性を修正するもので、確実にインストールすべき更新。

推奨更新

お使いのコンピュータで発生しうる不具合の修正。

任意更新

セキュリティとは無関係な問題や機能拡張を提供する更新。

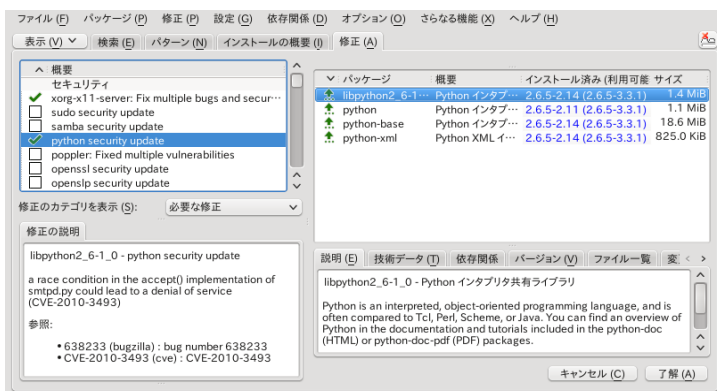
4.1 オンライン更新ダイアログ

YaST オンライン更新 ダイアログは、2 種類のユーザ インターフェイスで提供されています。それらはそれぞれ GTK (GNOME 向け) および Qt (KDE 向け) と呼ばれるもので、それぞれ外観は異なるものの基本的に同じ機能を 提供しています。下記の章では、それぞれのユーザインターフェイスについて概要を説明しています。ダイアログを開くには、YaST を起動して ソフトウェア > オンライン更新 を選択します。また、コマンドラインから `yast2 online_update` と入力することでも起動を行なうことができます。

4.1.1 KDE インターフェイス (Qt)

オンライン更新 ウィンドウは4つの領域から構成されています。

図 4.1 YaST オンライン更新—Qt インターフェイス



左側にある 概要セクションには、openSUSE に対して利用可能な修正が一覧表示されます。修正はセキュリティ種類別に並べられ、それぞれセキュリティ、推奨、オプションのラベルが付けられています。概要セクションの表示方法は、修正のカテゴリを表示 を選択することで変更できます：

必要な修正 (既定のビュー)

お使いのシステムに適用可能で、インストールされていない修正。

必要のない修正

お使いのシステムにはインストールされていないパッケージに対する修正か、もしくは既に要件を満たしてインストールする必要のない修正(他のリポジトリなどから既に関連するパッケージ修正をインストールしている場合など)。

すべての修正

openSUSE に対して提供されているすべての修正。

概要セクションに表示されている各項目には、それぞれシンボルと修正名が表示されます。表示されるシンボルとそれらの意味についての概要を表示するには、**Shift + F1** を押してください。なお、それぞれセキュリティと推奨の修正に対する処理は自動的に設定されます。処理は *自動インストール*、*自動更新*、*自動削除* の中から自動的に判断されます。

更新リポジトリ以外のリポジトリから最新版のパッケージをインストールした場合は、そのインストールによってパッケージに対する修正要件を満たしたことになります。この場合、修正の一覧にあらかじめチェックマークが付けられた形で表示され、その修正のインストールを敢えて行なうまでは一覧内に現われ続けることに注意してください。これにより、修正をインストールすることはありません(なぜなら既に最新版がインストールされているため)が、修正がインストール済みのものとしてマークされます。

ダイアログの左下に *修正の説明* を表示するには、概要セクションで修正を選択してください。右上のセクションには選択した修正に含まれるもの(修正は複数のパッケージを含む場合があるため)が表示されます。また、右上のセクションで項目を選択すると、修正に含まれる各パッケージについて、詳細を表示することができます。

4.1.2 GNOME インターフェイス (GTK)

オンライン更新 ウィンドウは4つの領域から構成されています。

図 4.2 YaST オンライン更新—GTK インターフェイス



右上のセクションにはopenSUSEに対して利用可能な(またはインストール済みの)修正が一覧表示されます。セキュリティ種類でフィルタを設定するには、ウインドウの左上にある優先度の項目を設定してください。優先度にはセキュリティ、推奨、オプション、任意の優先度の種類があります。

利用可能なすべての修正がインストール済みである場合、右上のパッケージ一覧セクションには何も項目が表示されません。また左下の枠内には利用可能な修正とインストール済みの修正の数がそれぞれ表示され、それらの項目を選択することで利用可能な修正やインストール済みの修正を表示することができます。

パッケージ一覧で項目を選択すると、ダイアログの右下にある枠内に、修正に関する説明やその他の詳細情報が表示されます。修正は複数のパッケージを含む場合があるため、右下にある以下に適用をマウスで選択して、修正にどのパッケージが含まれるのかを表示することもできます。

また修正項目を選択することで、ウインドウの下部に修正に関する詳細な情報を表示させることもできます。修正の詳細説明のほか、利用可能なバージョンについても表示されます。ここからオプション指定の更新をインストールすることもできます。セキュリティ更新や推奨更新については、自動的にインストールするように選択されます。

4.2 修正のインストール

YaST オンライン更新ダイアログでは、利用可能な修正を一括でインストールすることができるほか、システムに適用したい修正を手作業で選択することもできます。それ以外にも、既にシステムに対して適用済みの修正を適用前に戻すこともできます。

既定では新しい修正のうち、お使いのシステムに対して適用可能な修正 (ただし オプション のものを除く) すべてがインストールされるように 選択が行なわれます。これらは特に何も操作しなければ、*了解* や *適用* を押すことでインストールされます。

手順 4.1 YaST オンライン更新での修正適用

- 1 YaST を起動し、 **ソフトウェア > オンライン更新** を選択します。
- 2 お使いのシステムに対して適用可能で、かつ新しい修正 (ただし オプション のものを除く) を全部インストールするには、そのまま *適用* または *了解* ボタンを押します。
- 3 適用したい修正の選択を修正するには、下記の手順で行ないます:
 - 3a GTK や Qt の各インターフェイスで提供されるフィルタや表示方法を設定します。詳しくは 4.1.1 項 「KDE インターフェイス (Qt)」 (90 ページ) および 4.1.2 項 「GNOME インターフェイス (GTK)」 (91 ページ) をお読みください。
 - 3b 要件に合わせて修正のチェックボックスを付けるか外すか (GNOME の場合)、もしくは修正を選択してマウスの右ボタンを押すことで表示されるコンテキストメニューから選択を行なう (KDE の場合) ことで、修正の選択を設定します。

重要項目: セキュリティ更新は常にインストールしてください

特段の理由がない限り、セキュリティ 関連の修正については選択を外してはなりません。これらの修正は致命的なセキュリティ 欠陥 を修正し、お使いのシステムを攻撃から保護するためのものです。

3c 多くの修正には複数のパッケージに対する更新が含まれています。それぞれのパッケージに対する処理を変更したい場合は、パッケージの表示からマウスで選択し、右ボタンを押して処理を選択してください (KDE の場合)。

3d 最後に設定した内容を確認し、選択した修正をインストールするには *適用* または *了解* を押します。

- 4** After the installation is complete, click *Finish* to leave the YaST *Online Update*. Your system is now up-to-date.

ティップ: **deltarpm** の無効化

既定では更新を **deltarpm** (差分 RPM) の形でダウンロードします。 **deltarpm** からパッケージを再構築する作業はメモリにも CPU にも負荷を与える処理であるため、環境やハードウェア設定によっては性能への影響を防ぐため、 **deltarpm** の使用をしないように設定する必要がある場合があります。

deltarpm を使用しないように設定するには、 `/etc/zypp/zypp.conf` ファイルを開いて `download.use_deltarpm` のオプションを `false` に設定します。

4.3 自動オンライン更新

YaST では日単位、週単位、月単位のスケジュールで自動更新を設定することができます。この機能を設定するモジュールを使用するには、まず `yast2-online-update-configuration` パッケージをインストールする必要があります。

手順 4.2 自動オンライン更新の設定

- 1** 上記のパッケージをインストールしたら、YaST を起動して *ソフトウェア > オンライン更新の設定* を選択します。

上記の方法以外にも、コマンドラインから

`yast2 online_update_configuration` と入力してモジュールを起動してもかまいません。

2 まずは *自動オンライン更新* にチェックを付けます。

3 次に更新間隔を *毎日*, *毎週*, *毎月* から選択します。

カーネル更新やライセンス同意が必要な修正などでは、ユーザ側での操作が必要な場合があります。これらは自動更新処理を行なうことができません。

4 ライセンス同意について自動的に受け入れるように設定するには、*ライセンスに同意する* にチェックを付けます。

5 更新処理を完全に自動で動作させたい場合は、*対話操作が必要な更新をスキップ* を選択します。

重要項目: 修正のスキップについて

何らかの操作が必要な修正を飛ばす (スキップする) 場合は、それらの修正をインストールするために手作業での *オンライン更新* を忘れずに実施してください。この作業を忘れてしまうと、重要な修正がインストールされないままになってしまいます。

6 最後に設定内容を確認し、*OK* を押します。

インターネットからのパッケージのインストール

通常、パッケージは設定された有効なリポジトリからしかインストールを行なうことができません。ですが、インストール時に設定した公式なリポジトリ以外にも 様々なリポジトリが存在しています。openSUSE® ビルドサービスでは、数百にも 及ぶ多数のパッケージと多数のサードパーティ製リポジトリが存在しています。詳しくはhttp://ja.opensuse.org/Additional_YaST_Package_Repositories (日本語) または http://en.opensuse.org/Additional_package_repositories (英語) をご覧ください。

また openSUSE では、これらのリポジトリを購読することなくインストールを行なうことができるよう、2つの簡単な方式を用意しています。1つめは 1 クリックインストール 方式で、Web ブラウザを利用して直接パッケージのインストールができる仕組みです。2つめは openSUSE の既知のリポジトリからパッケージ検索を行なう仕組みです。この検索結果から、直接任意のパッケージをインストールすることができます。

警告: 外部のソフトウェア作成元の信頼について

外部のソフトウェアリポジトリのインストールを行なう前に、その作成元が信頼できる ものであるかどうかを事前にご確認ください。openSUSE ではサードパーティ製の ソフトウェアをインストールしたことによって引き起こされた、潜在的なものを含む いかなる問題に対しても責任を負いません。

5.1 1 クリックインストール

「1 クリックインストール」を利用したインストールでは、Web ブラウザから多数のリポジトリを検索することができます。最も有名などころとしては、openSUSE ビルドサービスのソフトウェア検索があります。

手順 5.1 1 クリックインストールを利用した openSUSE ビルドサービス内のパッケージインストール

- 1 <http://software.opensuse.org/search>にある openSUSE ビルドサービスの検索インターフェイスを起動します。
- 2 プロップダウンメニューを利用してお使いのシステムとバージョンを選択します。たとえば openSUSE 11.3 のようになります。
- 3 次に、インストールしたいパッケージの名前を入力します。たとえば、OpenStreetMap エディタであれば jism と入力します。
- 4 検索結果を改善するには、検索オプションを指定して要件を満たすように設定します。
- 5 検索ボタンを押します。
- 6 結果が表示されたら、インストールしたい項目を選択し、1 クリックインストールボタンを押すとインストールを開始することができます。
- 7 Web ブラウザで「ファイルを開く」ダイアログが表示されたら、YaST メタパッケージハンドラを利用してファイルを開きます。

1 クリックインストールが起動し、追加のソフトウェアリポジトリダイアログが表示されます。ここでは1 クリックインストールでインストールするよう選択したパッケージについて、そのパッケージの提供元リポジトリが表示されています。既定ではこれらのリポジトリが選択されていますので、そのまま勧めてください。既定では、インストール後もこれらのリポジトリを購読し、将来のパッケージ更新を受け取ることができます。

- 8 リポジトリを今回だけ利用する場合には、インストール後もこれらのリポジトリを購読するのチェックを外してください。次へを押すと次に進みます。

- 9 次にインストールするものとして選択したソフトウェアが一覧表示されます。通常の場合、何も変更する必要はありません。次へを押して進めてください。
- 10 次にこれまでに選択してきた内容が 提案画面に表示されます。カスタマイズボタンを押すと、上記までの設定手順をやり直すことができます。インストール作業を行なうには、次へを押してから はいを押してください。
- 11 あとはインストール作業を行なうため、root のパスワードを入力します。新しいリポジトリを追加する場合は、そのリポジトリの GnuPG 鍵をインポートするためのダイアログが表示される場合があります。また、インストール作業中に特に操作する必要のないいくつかのポップアップが表示されます。最後に「インストールに成功しました」メッセージが表示されますので、完了ボタンを押して閉じてください。

ティップ: 1 クリックインストール機能の無効化

1 クリックインストールの機能を無効化したい場合は、YaST を利用して `yast2-metapackage-handler` パッケージをアンインストールするか、もしくは root から下記のコマンドラインを入力します:

```
rpm -e yast2-metapackage-handler
```

5.2 YaST パッケージ検索

お使いの環境がインターネットに接続されていれば、YaST パッケージ検索から openSUSE に対するほぼ全ての既知のリポジトリを直接検索し、パッケージをインストールすることができます。このモジュールは <http://packages.opensuse-community.org/> で提供されている Webpin パッケージ検索の YaST 版フロントエンドです。この YaST モジュールは既定ではインストールされていないので、あらかじめ `yast2-packager-webpin` パッケージをインストールしておく必要があります。

手順 5.2 YaST パッケージ検索を利用したパッケージのインストール

- 1 YaST コントロールセンターからパッケージ検索を起動します。ソフトウェア > パッケージ検索を選択してください。

- 2 *Search Expression* (検索文字列) の欄に検索したいソフトウェア パッケージの名前を入力し、**検索**を押します。
- 3 検索結果は 見つかったパッケージのタブに表示されます。パッケージ名を押すと、パッケージの説明欄にそれぞれリポジトリの URL やパッケージのバージョン番号、アーキテクチャが表示されます。

警告: パッケージ情報の再確認について

インストール作業を行なう前に、ソフトウェアを提供しているリポジトリが、信頼できるリポジトリであるかどうかを再度ご確認ください。また、アーキテクチャがお使いのシステムに適合していることもあわせてご確認ください (たとえば **x86_64** パッケージは **64** ビットのシステムにしかインストールできません)。

- 4 次にインストールを行ないたいパッケージについて、チェックボックスでの選択を行ないます。複数のパッケージをいっぺんにチェックしてもかまいません。また、現在の選択を消してしまうことなく新しい検索を続けて行なうこともできます。新しい検索を行なうと一覧からは消えてしましますが、選択した全てのパッケージから以前に選択したパッケージを表示することができます。パッケージの選択が終わったら **次へ**を押します。

追加のソフトウェアリポジトリダイアログでは、インストールを行ないたいパッケージの提供元リポジトリが表示されます。これらはいずれも既定でチェックが付けられている状態になっています。インストールを続行するにはリポジトリを選択した状態のままに設定してください。既定ではインストール完了後もこれらのリポジトリを購読するように設定し、将来に更新版が提供された場合でも更新を受け取ることができるようになっています。

- 5 リポジトリを今回限りで利用する場合には、インストール後もこれらのリポジトリを購読するのチェックを外してください。 **次へ**を押すと次に進みます。
- 6 次にインストールするものとして選択したソフトウェアが一覧表示されます。通常の場合、何も変更する必要はありません。 **次へ**を押して進めてください。

- 7 次にこれまでに選択してきた内容が *提案画面* に表示されます。カスタマイズボタンを押すと、上記までの設定手順をやり直すことができます。インストール作業を行なうには、次へを押してから はいを押してください。
- 8 新しいリポジトリを追加する場合は、そのリポジトリの **GnuPG** 鍵をインポートするためのダイアログが表示される場合があります。また、インストール作業中に特に操作する必要のないいくつかのポップアップが表示されます。最後に「インストールに成功しました」メッセージが表示されますので、完了ボタンを押して閉じてください。

アドオン製品のインストール

アドオン製品とはシステムを拡張するためのものです。サードパーティ製のアドオン製品をインストールしたり、openSUSE® に対する特別なシステム拡張をインストールしたりすることができます (たとえば追加言語対応用の CD であったり、バイナリドライバを提供する CD であったりします)。新しいアドオンをインストールするには、YaST を起動して ソフトウェア> アドオン製品 を選択してください。アドオン製品には様々なメディア種類を選択することができます。たとえば CD, FTP, USB マスストレージ (USB フラッシュメモリやハードディスク) のほか、ローカルディレクトリからもインストールすることができます。もちろん ISO ファイルから直接インストールすることもできます。ISO ファイルをメディアとしてアドオン製品を追加するには、ローカル ISO イメージを選択して ISO イメージへのパスを入力してください。リポジトリ名は任意で指定してください。

6.1 アドオン

新しいアドオンをインストールするには、下記の手順で行ないます:

- 1 YaST から ソフトウェア> アドオン製品 を選択し、インストール 済みのアドオン製品の概要を表示します。
- 2 新しいアドオン製品をインストールするには *追加* を押します。
- 3 お使いのリポジトリに該当する メディア種類 を選択します。

- 4 リムーバブルメディアからリポジトリを追加するには、それぞれ必要なオプションを設定してメディアを挿入するか、もしくはUSBデバイスをマシンに接続します。
- 5 なお、リポジトリの説明をダウンロードを選択することもできます。このオプションのチェックを外した場合、YaSTは必要に応じて後から自動的にダウンロードを行ないます。次へを押して進めてください。
- 6 ネットワークからリポジトリを追加した場合は、それぞれ問い合わせのあった情報を入力します。次へを押して進めてください。
- 7 追加したリポジトリによって異なりますが、それぞれ署名に使用するGPG鍵のインポートに関する問い合わせや、ライセンスへの同意の問い合わせが行なわれます。

これらのメッセージを確認するとYaSTはメタデータをダウンロードして解析し、設定されたソフトウェアリポジトリの一覧に追加されます。

- 8 なお、必要であれば3.4.2項「リポジトリ属性の管理」(86ページ)で示されている手順でリポジトリのプロパティを設定することができます。設定ダイアログを閉じるにはOKを押してください。
- 9 アドオンメディアのリポジトリへの追加が完了すると、ソフトウェア管理が起動してパッケージのインストールを行なうことができるようになります。詳しくは第3章 ソフトウェアのインストールと削除(67ページ)をお読みください。

6.2 バイナリドライバ

ハードウェアによっては、それを正しく動作させるためにバイナリ形態でしか提供されないドライバをインストールする必要があるものがあります。そのようなハードウェアをお持ちの場合は、お使いのシステム向けにバイナリドライバが提供されているかどうか、リリースノートを読んで詳細を確認してください。リリースノートを読むには、YaSTを開いてサポート>リリースノートを選択してください。

コマンドラインツールを利用したソフトウェア管理

本章では `zypper` と `RPM` について説明しています。これらはいずれも ソフトウェアを管理するためのコマンドラインツールです。本章で使用する用語(たとえばリポジトリ, 修正, 更新 など)については、3.1項「用語定義」(68 ページ)をお読みください。

7.1 `zypper` の使用

`zypper` はパッケージをインストールしたり更新したり、削除したりすることができるほか、リポジトリの管理を行なうこともできるコマンドラインのパッケージマネージャです。特にシェルスクリプトなどを利用して、遠隔からのソフトウェア管理処理を行なうのに便利な機能が用意されています。

7.1.1 一般的な使い方

`zypper` の一般的な書式は下記のとおりです:

```
zypper [グローバルオプション] コマンド [コマンドオプション] [パラメータ] ...
```

角括弧で括られた部分は任意入力の部分です。`zypper` を最も簡単に使用するには、`zypper` に続いてコマンド名を入力することです。たとえばお使いのシステムに必要な、全ての修正を適用するには下記のように入力します:

```
zypper patch
```

コマンドの前には1つ以上のグローバルオプションを指定することができます。たとえば `--non-interactive` は、コマンドを実行している間に何も

問い合わせを表示させない、という意味になります (自動で既定の回答を行なったものと見なす):

```
zypper --non-interactive patch
```

また、特定のコマンドに固有のオプションを使用するには、コマンドの後ろにそれらを入力します。たとえば `--auto-agree-with-licenses` は、ライセンス同意の確認を行なうことなく、お使いのシステムに必要な全ての修正を適用する、という意味になります (ライセンス同意は自動的に受け入れられたものと見なします):

```
zypper patch --auto-agree-with-licenses
```

コマンドによっては1つ以上のパラメータを必要とするものがあります。たとえば `install` コマンドでは、インストールしたいパッケージを1つ以上指定します:

```
zypper install mplayer
```

オプションによってもパラメータを必要とする場合があります。下記のコマンドは、既知の全てのパターンを表示します:

```
zypper search -t pattern
```

今までに説明した項目については、これらを全て組み合わせることが出来ます。たとえば下記のコマンドでは、`factory` リポジトリに限定して `mplayer` と `amarok` パッケージをインストールし、その際に冗長な出力を行なうことを意味します:

```
zypper -v install --from factory mplayer amarok
```

`--from` オプションはすべてのリポジトリを有効にしたままに設定(依存関係の解決のため)し、特定のリポジトリに対してパッケージを要求する意味です。

```
zypper -v install --repo factory mplayer amarok
```

さらに、多くの `zypper` コマンドには `dry-run` と呼ばれるオプションがあります。これは与えられたコマンドについてシミュレーションを行なう機能を提供します。シミュレーションであるため、実際にはシステムに対して何も変更を加えません。

```
zypper remove --dry-run MozillaFirefox
```

7.1.2 zypper を利用したソフトウェアのインストールと削除

パッケージをインストールしたり削除したりするには、下記のコマンドを入力します:

```
zypper install パッケージ名
zypper remove パッケージ名
```

zypper では、インストールと削除のコマンドを利用して様々な方法でのパッケージの指定を行なうことができます:

パッケージ名 (およびバージョン番号) を直接指定

```
zypper in MozillaFirefox
```

または

```
zypper in MozillaFirefox-3.5.3
```

リポジトリの別名とパッケージ名の指定

```
zypper in mozilla:MozillaFirefox
```

ここで、mozilla はインストール元のリポジトリ 別名とします。

ワイルドカードを利用したパッケージ名指定

下記のコマンドは、「**Moz**」で始まる全てのパッケージをインストールします。特にパッケージを削除する場合など、ご利用にあたってはご注意ください。

```
zypper in Moz*
```

能力設定での指定

たとえば Perl のモジュールをインストールしたい場合で、モジュールの名前がわからないような場合は、能力設定での指定を行なうこともできます:

```
zypper in 'perl(Time::ParseDate)'
```

能力設定、アーキテクチャ、バージョンの組み合わせ

能力設定のほか、アーキテクチャ (i586 や x86_64 など) の指定を追加したり、バージョンの指定を追加したりすることもできます。バージョンの指定は、下記のような比較演算子をあわせて指定します: < (より小さ

い), <= (等しいか、より小さい), = (等しい), >= (等しいか、より大きい), > (より大きい)

```
zypper in 'firefox.x86_64'
zypper in 'firefox>=3.5.3'
zypper in 'firefox.x86_64>=3.5.3'
```

RPM ファイルパスの指定

パッケージとして、ローカルまたはリモートのパスを指定することもできます:

```
zypper in /tmp/install/MozillaFirefox.rpm
zypper in
http://download.opensuse.org/repositories/mozilla/SUSE_Factory/x86_64/MozillaFirefox-3.5.3-1.3.x86_64.rpm
```

さらに、パッケージのインストールや削除は + や - の修飾子を指定することができます。たとえば emacs をインストールして vim を削除するには、下記のように入力します:

```
zypper install emacs -vim
```

emacs を削除し、同時に vim をインストールするには、下記のように入力します:

```
zypper remove emacs +vim
```

なお、- で始まるパッケージの指定が、コマンドオプションとして解釈されてしまうことを防ぐため、これらの指定は2つめ以降のパラメータで指定してください。それができない場合は--を指定し、コマンドオプションの終了を宣言してください:

```
zypper install -emacs +vim      # 間違い
zypper install vim -emacs       # 正しい
zypper install -- -emacs +vim   # 上記と同じ
zypper remove emacs +vim       # 上記と同じ
```

また、特定のパッケージを削除する際、不要となったパッケージについても合わせて自動的に削除するには、--clean-deps オプションを利用します:

```
rm package_name --clean-deps
```

何も指定しない場合、zypper は選択したパッケージをインストールしたり削除したりする際、および何らかの問題が発生した際にそれぞれ確認メッセージを表示します。この動作を無効化するには、--non-interactive オプションを指定します。このオプションは実際のコマンドよりも前に指定して

おく必要があります ((install, remove, patch) など)。下記のようになります:

```
zypper --non-interactive install パッケージ名
```

このオプションを使用すると、スクリプトや cron ジョブなどから zypper を利用できるようになります。

警告: 必要不可欠なパッケージの削除について

glibc, zypper, kernel などの必須パッケージについては、削除を行ってはなりません。これらのパッケージはシステム自身が正常に動作するのに必要なパッケージで、これらを削除してしまうとシステムが全く動作しなくなってしまうです。

ソースパッケージのインストール

ある特定のパッケージに対するソースパッケージをインストールするには、下記のように入力します:

```
zypper source-install パッケージ名
```

このコマンドを実行すると、指定したパッケージを構築する際に必要となるパッケージについてもインストールを行ないます。このような動作を行ないたくない場合は、`-D` スイッチを追加してください。また、構築する際に必要なパッケージだけをインストールしたい場合は、`-d` をお使いください。

```
zypper source-install -d パッケージ名 # ソースパッケージのみ  
zypper source-install -D パッケージ名 # 構築時に必要なもののみ
```

もちろんこの機能は、有効に設定しているリポジトリの中にソースパッケージを提供するものがあった場合に動作します (ソースパッケージのリポジトリは既定で追加されるようになっていますが、有効には設定されません)。リポジトリの管理について、詳しくは 7.1.4 項「zypper を利用したリポジトリ管理」(114 ページ)をお読みください。

お使いのリポジトリ内で提供されている、全てのソースパッケージの一覧を表示するには、下記のように入力します:

```
zypper search -t srcpackage
```

ユーティリティ

全てのパッケージ間の依存関係が満たされているかどうかを確認し、満たされていない場合に修復処理を行なうには、下記のように入力します:

```
zypper verify
```

パッケージ間の依存関係は正しく解決されていないほか、パッケージによっては他のパッケージを「推奨」している場合があります。これらの推奨パッケージは、それらが利用可能でインストールできるものであった場合にのみインストールされます。ある特定のパッケージが推奨パッケージを指定していて、それらのパッケージがインストールされた後で推奨パッケージが利用できるようになった場合は、下記のように入力します:

```
zypper install-new-recommends
```

このコマンドは Web カメラや無線 LAN デバイスを接続したような場合にとても便利です。これにより、利用可能であればデバイスドライバや関連ソフトウェアがインストールされます。ドライバや関連ソフトウェアは、対応している特定のハードウェア依存関係が満たされた場合にのみインストールできます。

7.1.3 zypper を利用したソフトウェア更新

zypper を利用したソフトウェア更新には、修正のインストール、最新バージョンのインストール、ディストリビューション全体の更新、と 3 種類の方法があります。ディストリビューション全体の更新は、14.1 項「システムのアップグレード」(239 ページ)で示されている `zypper dist-upgrade` コマンドで行ないます。

修正のインストール

お使いのシステムに適用可能な全ての公式リリース修正をインストールするには、下記のように入力します:

```
zypper patch
```

この場合、お使いのリポジトリ内にある全ての修正は適合性についてチェックが行われ、必要と判断すればインストールが行なわれます。上記のコマンドは必要なものを全てインストールする際に入力するコマンドです。

また、 `zypper` では3種類の方法で修正の存在可否を問い合わせることができます:

`zypper patch-check`
必要な修正(お使いのシステムにインストールする必要があるが、まだインストールしていないもの)の数を表示します。

```
~ # zypper patch-check
リポジトリのデータを読み込んでいます...
インストール済みのパッケージを読み込んでいます...
5 個の修正が必要です (1 個のセキュリティ修正)
```

`zypper list-patches`
必要な修正(お使いのシステムにインストールする必要があるが、まだインストールしていないもの)について一覧を表示します。

```
~ # zypper list-patches
リポジトリのデータを読み込んでいます...
インストール済みのパッケージを読み込んでいます...

リポジトリ                                | 名前                                | バージョン |
カテゴリ      | 状態                                |-----|-----|
openSUSE 11.3 11.3-1.82 向けの更新 | lxsession                                | 2776
| security      | 必要
```

`zypper patches`
`openSUSE` 向けに公開されている全ての修正を一覧表示します。ただし既にインストールされているものや、インストールする必要のないものについても表示します。

特定の発信番号に関連する修正を一覧表示したりインストールしたりすることができます。特定の修正を一覧表示するには、 `zypper list-patches` コマンドを利用して下記のようなオプションを指定します:

`--bugzilla [=番号]`
Bugzilla による発信情報に対する修正を一覧表示します。任意でバグ番号を指定し、特定のバグに対する修正一覧を表示することもできます。

`--cve [=番号]`
CVE (聖者区政情報データベース) による発信情報に対する修正を一覧表示します。任意で **CVE** 番号を指定し、特定の番号に対する修正一覧を表示することもできます。

特定の **Bugzilla** または **CVE** 発信情報に対する修正をインストールするには、下記のように入力します:

```
zypper patch --bugzilla=番号
```

or

```
zypper patch --cve=番号
```

たとえば **CVE** 番号 **CVE-2010-2713** に対するセキュリティ 修正をインストールするには、下記のように入力します:

```
zypper patch --cve=CVE-2010-2713
```

更新のインストール

リポジトリには新しいパッケージが存在していてもそれらが修正の機能を 提供しない場合は、`zypper patch` コマンドを実行しても更新を行ないま せん。全てのインストール済みパッケージについて、利用可能な最新バー ジョンに更新するには、下記のように入力します:

```
zypper update
```

個別のパッケージについて更新を行なうには、`update` または `install` の角コ マンドに続けてパッケージ名を指定してください:

```
zypper update パッケージ名  
zypper install パッケージ名
```

また、インストール可能なすべての更新パッケージ一覧は、下記のコマンド で表示することができます:

```
zypper list-updates
```

なお、このコマンドでは下記の条件に該当するパッケージのみが一覧表示さ れます:

- 既にインストール済みのパッケージと製造元が同じもの
- 既にインストール済みのパッケージと同じ優先度か、もしくはそれにより 高い優先度が設定されているリポジトリからのもの
- インストール可能なもの (すべての依存関係が満たせるもの)

すべてのパッケージを (インストール可能かどうかは考慮せず) 一覧表示するには、下記のように入力します:

```
zypper list-updates --all
```

新しいパッケージがなぜインストールできないのかを知るには、単純にインストール または更新コマンドを利用してパッケージ名を指定してください。

新しい製品バージョンへのアップグレード

お使いの環境を新しい製品バージョンに簡単にアップグレードする (たとえば openSUSE 11.2 から openSUSE 11.3 など) ためには、まずお使いのリポジトリを新しい openSUSE リポジトリに設定する必要があります。詳しくは 7.1.4 項「zypper を利用したリポジトリ管理」(114 ページ) をお読みください。その後、必要なリポジトリに対して `zypper dist-upgrade` コマンドを利用してください。このコマンドにより、現在有効化されているすべてのリポジトリからインストール済みのパッケージを更新します。詳しい手順については 14.1.4 項「zypper を利用したディストリビューションアップグレード」(243 ページ) をお読みください。

ディストリビューションのアップグレードを、その他のリポジトリに対する依存関係が満たされる限り、特定のリポジトリに限って実施したい場合は、`--from` オプションを指定して別名または番号、もしくは URI でリポジトリを指定してください。

注記: `zypper update` と `zypper dist-upgrade` の違い

`zypper update` は、お使いのシステムの一貫性を保ちながら、新しいバージョンのパッケージをインストールしたい場合に使用します。zypper update では下記のルールを 厳守します:

製造元を変更しません

アーキテクチャの変更を行いません

ダウングレードを行いません

インストール済みのパッケージを保持し、インストールしたままにします

`zypper dist-upgrade` を実行した場合、その時点で有効に 設定されている全リポジトリからのパッケージがインストールされます。このルールは強制的なもので、パッケージの製造元やアーキテクチャが変更される 場合があるほか、場合によってはダウングレードが行なわれる場合があります

す。また、依存関係が満たされていないすべてのパッケージは、アップグレード後もインストールされないままになります。

7.1.4 zypper を利用したリポジトリ管理

zypper でインストールコマンドや修正コマンドは、それを実行する時点で設定されているリポジトリから行ないます。システムに設定済みのリポジトリについて、一覧を表示するには下記のように入力します:

```
zypper repos
```

または

```
zypper lr
```

実行すると、下記のように出力されます:

例 7.1 zypper—既知のリポジトリの一覧表示

#	別 名	名 前	有 効	更 新
1	Updates (Y)	Updates	はい (Y)	はい
2	openSUSE 11.2-0 え (N)	openSUSE 11.2-0	いいえ (N)	いいえ
3	openSUSE-11.2-Debug (Y)	openSUSE-11.2-Debug	いいえ (N)	はい
4	openSUSE-11.2-Non-Oss (Y)	openSUSE-11.2-Non-Oss	はい (Y)	はい
5	openSUSE-11.2-Oss (Y)	openSUSE-11.2-Oss	はい (Y)	はい
6	openSUSE-11.2-Source (Y)	openSUSE-11.2-Source	いいえ (N)	はい

様々なコマンドでリポジトリを指定する際は、上記の別名や URI のほか、zypper repos の出力結果であるリポジトリ番号(一番左の項目)でも指定することができます。また、リポジトリの別名とはリポジトリ名を使いやすく短縮したものです。なお、リポジトリ番号はリポジトリの修正を行なうと番号が変わる場合がありますのでご注意ください。別名については、あえて変更しない限り変わることはありません。

既定ではURIやリポジトリの優先度については表示されません。全ての詳細を表示するには、下記のコマンドを入力します:

```
zypper lr -d
```

リポジトリの追加

リポジトリを追加するには、下記のように入力します:

```
zypper addrepo URI 別名
```

ここで、URIはインターネット上のリポジトリを指す場所でもかまいませんし、ネットワークやディレクトリ、CDやDVDなどでもかまいません(詳しくは<http://ja.opensuse.org/Libzypp/URI>をお読みください)。また、別名はリポジトリに対するユニークな短縮名を指定します。別名は他の別名と重複しない限り、どのような名前でもかまいません。他の別名と重複する場合は、zypperが警告メッセージを表示します。

リポジトリの削除

一覧からリポジトリを削除したい場合はzypper removerepoコマンドを使用します。このコマンドの後に続けて削除したいリポジトリの別名やリポジトリ番号を指定します。たとえば例7.1「zypper—既知のリポジトリの一覧表示」(114 ページ)の例で3番目のリポジトリを削除するには、下記のコマンドを入力します:

```
zypper removerepo 3
```

リポジトリの編集

リポジトリの有効／無効を切り替えるには、zypper modifyrepoコマンドを使用します。これ以外にも、リポジトリの設定(更新可否, 名前, 優先度)についても修正することができます。たとえば下記の例では、「updates」という名前のリポジトリについて、有効に設定して自動更新を行なうようにし、かつ優先度を20にします:

```
zypper mr -er -p 20 'updates'
```

リポジトリの修正は1つのリポジトリに対して行なうこともできますが、一括で指定することもできます:

-a: 全てのリポジトリ
-l: ローカルリポジトリ
-t: リモートリポジトリ
-m 種類: 特定の種類の リポジトリ (種類には下記のような条件を 指定しま
す: http, https, ftp, cd, dvd, dir, file, cifs, smb, nfs, hd, iso)

リポジトリの別名を変更するには、renamerepo コマンドを使用します。下
記の例では、Mozilla Firefox という別名を、単に firefox という別名
に変更します:

```
zypper renamerepo 'Mozilla Firefox' firefox
```

7.1.5 zypper を利用したリポジトリとパッ ケージの問い合わせ

zypper ではリポジトリやパッケージに対して様々な問い合わせを行なうこと
ができます。製品、パターン、パッケージ、修正について、利用可能な全て
のものを表示するには、下記のように入力します:

```
zypper products  
zypper patterns  
zypper packages  
zypper patches
```

全てのリポジトリに対して特定のパッケージを問い合わせるには、search
コマンドを使用します。通常はパッケージ名を対象にして検索を行ないます
が、オプション指定で概要と説明を対象に 含めることもできます。また、*
や? のワイルドカードを使用することもできます。検索は通常、大文字と小
文字を 区別せずに行ないます。

```
zypper se firefox      # "firefox" を単純に検索  
zypper se *fire*       # ワイルドカードを使用  
zypper se -d fire      # パッケージの説明と概要を検索対象に含む  
zypper se -u firefox    # 未インストールのパッケージだけを表示
```

ある特定の機能を提供するパッケージを検索するには、what-provides コ
マンドを利用します。たとえばPerlのモジュールSVN:::Coreを提供するパッ
ケージを検索したい場合は、下記のように入力します:

```
zypper what-provides 'perl(SVN:::Core)'
```

単一のパッケージを問い合わせるには、パラメータに正確なパッケージ名を指定して `info` コマンドを実行します。これにより、そのパッケージの詳細情報が表示されます。また、このパッケージが何を必要としているのか、もしくは何を推奨しているのかについて表示するには、`--requires` と `--recommends` のオプションをそれぞれ指定します:

```
zypper info --requires MozillaFirefox
```

`what-provides` パッケージ名 コマンドは、`rpm -q --whatprovides` パッケージ名に似ていますが、`rpm` は **RPM** データベース (全てのインストール済みパッケージに関する情報を保持しているデータベース) に対して問い合わせるだけであるのに対して、`zypper` はインストールされていないものに対しても問い合わせることができます。

7.1.6 zypper の設定

`zypper` には設定ファイルが存在していて、`zypper` の動作を恒久的に変更することができるようになっていきます (システム全体で設定できるほか、ユーザ単位でも設定できます)。システム全体での設定については `/etc/zypp/zypper.conf` ファイルを編集してください。また、ユーザ独自の変更を行ないたい場合は、`~/.zypper.conf` ファイルを編集してください。なお、`~/.zypper.conf` ファイルが存在しない場合は、`/etc/zypp/zypper.conf` ファイルをテンプレートとしてお使いください。左記のファイルを `~/.zypper.conf` にコピーし、必要に応じて修正を行なってください。詳しい設定内容や利用可能なオプションについては、ファイル内のコメント文をお読みください。

7.1.7 トラブルシューティング

設定済みのリポジトリからのパッケージにアクセスする際、何らかの問題が発生した場合 (たとえば確かにリポジトリ内に存在するパッケージであるにも関わらず、`zypper` がそれを発見できない場合など) は、下記のコマンドでリポジトリの更新を行なうことができます:

```
zypper refresh
```

これでもうまく動作しない場合は、下記をお試しください:

```
zypper refresh -fdb
```

これにより生のメタデータを強制的にダウンロードし、データベースの再構築を行なって完全な更新を行ないます。

7.1.8 さらになる情報

コマンドラインからのソフトウェア管理について、さらに詳しい情報は `zypper help` または `zypper help` コマンドと入力することで表示される ヘルプ をお読みいただくか、もしくは `zypper(8)` のマニュアルページをお読みください。コマンドについて完全かつ詳細なリファレンスのほか、チートシートと呼ばれる最も重要なコマンドについての情報、および `zypper` をスクリプトやアプリケーションから使用する方法については、http://wiki.opensuse.org/SDB:Zypper_usage をお読みください。また、openSUSE の最新バージョンについて変更点の一覧を読むには、http://wiki.opensuse.org/SDB:Zypper_changes_11.3 をご覧ください。

7.2 RPM - パッケージマネージャ

RPM(RPMパッケージマネージャ)はソフトウェアパッケージの管理ツールです。主なコマンドは `rpm` と `rpmbuild` の2つがあります。パワフルな RPM データベースの仕組みが存在するため、ユーザからでもシステム管理者からでも、さらにいえばパッケージ構築者からでも、インストール済みのソフトウェアについて詳細な情報を問い合わせることができます。

`rpm` コマンドは基本的に5つのモードから構成されています: ソフトウェアのインストール, ソフトウェアのアンインストール(または更新), RPM データベースの再構築, RPM データベースや RPM パッケージへの問い合わせ, パッケージの一貫性確認と署名確認の5つです。また、`rpmbuild` コマンドは、ソースコードからインストール可能なパッケージを構築するために使用します。

インストール可能な RPM アーカイブは、特別なバイナリフォーマットの形式になっています。これらのアーカイブにはインストール対象のプログラムのほか、インストール時に `rpm` コマンドがソフトウェアパッケージを設定する際に使用するデータや、ドキュメンテーション化を目的とした RPM データベースのメタデータなどが含まれています。RPM アーカイブは、通常 `.rpm` の拡張子を持っています。

ティップ: ソフトウェア開発パッケージ

ソフトウェア開発を行なう際に必要なパッケージ(ライブラリ、ヘッダファイルなど)については、本体とは別途のパッケージで提供されています。これらのパッケージは、ソフトウェアのコンパイルをご自身で行ないたい場合にのみ必要となります(たとえば最新の **GNOME** パッケージなど)。これらは本体の パッケージ名の後ろに `-devel` という拡張子が追加されます。たとえば `alsa-devel`, `gimp-devel`, `libkde4-devel` のようになります。

7.2.1 パッケージの署名検証

RPM パッケージには GnuPG による署名が付与されています。RPM パッケージの署名を検証するには、`rpm --checksig package-1.2.3.rpm` のように入力し、Novell/SUSE やその他の信頼のおける発信元から提供されたものであるかどうか確認してください。これは特にインターネット経由でパッケージの更新を行なう場合にお勧めです。

7.2.2 パッケージの管理: インストール、更新、アンインストール

通常の場合、RPM アーカイブのインストールは非常に簡単です: `rpm -i パッケージ.rpm` と入力するだけです。このコマンドでインストールを行なうことができるのは、このパッケージに対する依存関係が満たされ、他のパッケージとの矛盾が発生しない場合のみです。依存関係が満たされない点があり、追加でインストールしておくべきソフトウェアが存在している場合、`rpm` はエラーメッセージを表示してインストールを中止します。また同時に RPM データベースでは、矛盾が発生しないかどうかを確認します。この矛盾確認では、ある特定のファイルが複数のパッケージに属することにならないかどうかを確認します。なお、オプションを指定することで `rpm` コマンドに対してこれらの振る舞いを行なわないようにさせることもできますが、これは熟練者のみが使用するべきオプションです。知識や経験のある方でなければ、システムの一貫性についてリスクを追うことになりますし、システムの更新ができなくなってしまう場合もあります。

オプション設定 `-U` または `--upgrade` と、`-F` または `--freshen` は、いずれも パッケージの更新を行なうためのコマンドです(たとえば `rpm -F パッケージ名.rpm` のように実行します)。このコマンドは古いバージョンのファイルを削除し、その後すぐに新しいファイルをインストールする動作をします。また、これら 2 種類の コマンドの違いは、`-U` がお使いのシステムに存在していなかったパッケージについてはインストールを行いますが、`-F` ではそのようなパッケージはインストールせず、更新を行なうだけになります。また更新の際、`rpm` コマンドは 下記の方針で注意を払いながら、設定ファイルの更新を行ないます:

- 設定ファイルを管理者が変更していない場合、`rpm` は 設定ファイルの最新版をそのままインストールします。システム管理者側で行なうべき作業はありません。
- 更新を行なう前にシステム管理者が設定ファイルを書き換えていた場合、`rpm` はその変更されたファイルを `.rpmorig` または `.rpmsave` (バックアップファイル) の拡張子を付けて保存し、新しいほうのパッケージから新しい設定ファイルをインストールします(ただし元々インストールされていた設定ファイル(システム管理者が書き換える前のもの)と、新たにインストールしようとしている設定ファイルが異なる場合のみ)。この場合は、バックアップファイル(`.rpmorig` または `.rpmsave`) と新たにインストールされたファイルを 比較し、新しいファイルに対して書き換えていた部分を反映してください。その後、今後の更新で問題が発生しないようにするため、それぞれ `.rpmorig` や `.rpmsave` の ファイルを削除してください。
- なお、`.rpmnew` ファイルが作成される場合もあります。これは設定ファイルが既に存在していて、かつそのファイルが `.spec` ファイル内で `noreplace` (置き換えを行なわない) ラベルが設定されている場合に作成されます。

更新作業ののち、設定ファイルの比較と修正が終わったら、`.rpmsave` と `.rpmnew` ファイルは削除しておいてください。これらのファイルは今後の更新を阻害する可能性があるためです。また、`.rpmorig` ファイルは、**RPM** データベースで認識されていなかったファイルの場合に付与されます。

それ以外の場合には `.rpmsave` の拡張子が使用されます。言い換えれば、`.rpmorig` は **RPM** の管理対象外のファイル であった場合にのみ生成されます。また `.rpmsave` は 古いバージョンの **RPM** から新しいバージョンのものに更新した場合に、`.rpmnew` はシステム管理者が設定ファイルに変更を行

なったかどうかは不明である場合に生成されます。これらのファイルの一覧は、`/var/adm/rpmconfigcheck` から取得できます。設定ファイルによっては (たとえば `/etc/httpd/httpd.conf`)、動作を継続させるために上書きを行わない場合もあります。

なお、`-U` スイッチでの処理は、`-e` でいったんアンインストールしてから `-i` でインストールする のとは 処理が異なります。可能な限り `-U` をお使いください。

パッケージを削除するには、`rpm -e パッケージ名` と入力します。`rpm` は未解決の依存関係がなければ、そのままパッケージだけを削除します。たとえば `Tcl/Tk` の削除のように、他のアプリケーションがそれを必要としている場合は、削除は不可能です。この場合でも、**RPM** はデータベースの助けを借りて処理を行ないます。もしも何らかの理由で、依存関係の問題が無いにも関わらず削除ができない場合は、**RPM** データベースを `--rebuilddb` で構築し直すことで解決するかもしれません。

7.2.3 RPM と修正

お使いのシステムでのセキュリティを維持するため、更新パッケージは定期的にインストールしなければなりません。また、以前パッケージ内に存在していたバグについても、パッケージ全体を置き換えることで解決できる場合もあります。小さなファイルにバグがある巨大なパッケージがあったような場合に、このようなことが起こります。しかしながら **SUSE RPM** では、パッケージ内の 修正部分だけをインストールするための機能を提供しています。

たとえば `pine` というパッケージを利用して 実例を示してみましょう：

修正 **RPM** はお使いのシステムに対して適切なものでしょうか？

これを確認するには、まずインストール済みのパッケージについて問い合わせを行ないます。`pine` であれば、下記のように実行します：

```
rpm -q pine
pine-4.44-188
```

さらに、修正 **RPM** がインストール済みバージョンに対応しているかどうかを確認します：

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
```

```
pine = 4.44-195
pine = 4.44-207
```

上記のようにこの修正は、3種類のバージョンの **pine** に適用可能です。インストール済みのバージョンが一覧の条件に合致しているため、この修正はインストール可能であると判断できます。

修正でどのファイルを置き換えるのでしょうか？

修正によって影響を受けるファイルは、修正 **RPM** から簡単に閲覧することができます。rpm コマンドに **-P** という修正機能用のオプションを追加してください。ファイルの一覧を取得するには、下記のコマンドを実行します:

```
rpm -qpPl pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

なお、既に修正をインストールしてある場合は、下記のコマンドを実行します:

```
rpm -qPl pine
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

どのようにして修正 **RPM** をシステムにインストールするのでしょうか？

修正 **RPM** は通常の **RPM** のように扱うことができます。唯一の違いは、適切な **RPM** がインストールされている状態でなければならない、という点です。

お使いのシステムにどの修正がインストールされているのでしょうか？そのバージョンは？

システムにインストールされている全修正の一覧は、rpm **-qPa** で表示することができます。たとえば1つの修正だけがシステムにインストールされている場合、出力は下記のようになります:

```
rpm -qPa
pine-4.44-224
```

もし後日、元々どのバージョンのパッケージがインストールされているのかを知りたくなった場合は、**RPM** データベースから知ることができます。たとえば pine の場合、この情報は下記のように入力すると出力させることができます:

```
rpm -q --basedon pine
pine = 4.44-188
```

RPM の修正機能など、さらなる情報については rpm や rpmbuild のマニュアルページをお読みください。

注記: openSUSE の公式な更新

更新のダウンロードサイズをできる限り小さくするため、openSUSE の公式な更新は修正 RPM の形式では提供されておらず、差分 RPM (deltarpm) の形式で提供されています。詳しくは 7.2.4 項「差分 RPM パッケージ」(123 ページ)をお読みください。

7.2.4 差分 RPM パッケージ

差分 RPM パッケージ (deltarpm) には、古いバージョンの RPM パッケージと新しいバージョンの RPM パッケージの差分が含まれています。古いほうの RPM に対して差分 RPM を適用すると、新しいバージョンの RPM を作成することができるという仕組みです。また差分 RPM は、インストール済みの古いパッケージからでも作成できる機能を備えています。そのため、古いほうの RPM について RPM ファイルそのもののコピーを保管しておく必要がありません。また、差分 RPM パッケージは修正 RPM よりもさらに小さいサイズになっているため、インターネットを介して転送するのに都合の良いものになっています。逆に差分 RPM での更新処理は、通常の RPM や修正 RPM よりも CPU 処理を多く要してしまうという欠点があります。

それぞれ `prepdeltarpm`, `writedeltarpm`, `applydeltarpm` バイナリは差分 RPM を扱うためのソフトウェア一式 (deltarpm パッケージ) で、差分 RPM を作成したり適用したりすることができます。たとえば下記のコマンドでは、`new.delta.rpm` という名前の差分 RPM を生成します。それぞれ `old.rpm` と `new.rpm` が存在していることを前提にしています:

```
prepdeltarpm -s seq -i info old.rpm > old.cpio
prepdeltarpm -f new.rpm > new.cpio
xdelta delta -0 old.cpio new.cpio delta
writedeltarpm new.rpm delta info new.delta.rpm
```

最後に、一時的に使用していた作業ファイル `old.cpio`, `new.cpio`, `delta` を削除して完了です。

上記のようにファイルを作成したら、古いパッケージがインストールされている環境で `applydeltarpm` を実行すると、新しい **RPM** を生成することができます:

```
applydeltarpm new.delta.rpm new.rpm
```

ファイルシステムを利用せず、古い **RPM** パッケージを直接指定するには、`-r` オプションを利用します:

```
applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

技術的な詳細は、`/usr/share/doc/packages/deltarpm/README` をお読みください。

7.2.5 RPM への問い合わせ

`rpm` コマンドで `-q` を使用すると、**RPM** アーカイブ内 (`-p` オプションで指定します) や **RPM** データベース 内を対象にして問い合わせを行ったり、調査を行ったりすることができます。スイッチによっては情報の種類を指定する必要がある場合もあります。詳しくは表7.1「最も重要な **RPM** 問い合わせオプション」(124 ページ)をお読みください。

表 7.1 最も重要な **RPM** 問い合わせオプション

<code>-i</code>	パッケージ情報。
<code>-l</code>	ファイル一覧。
<code>-f FILE</code>	指定したファイルを含むパッケージ名。ここで、 <i>FILE</i> はフルパス (ルートディレクトリからの絶対パス) を指定します。
<code>-s</code>	状態情報付きのファイル一覧 (<code>-l</code> オプションの機能を含みます)。
<code>-d</code>	ドキュメンテーションファイルのみの一覧表示 (<code>-l</code> オプションの機能を含みます)。

<code>-c</code>	設定ファイルのみの一覧表示 (<code>-l</code> オプションの機能を含みます)。
<code>--dump</code>	完全な詳細情報を含んだファイル一覧 (<code>-l, -c, -d</code> などと合わせて指定します)
<code>--provides</code>	他のパッケージから <code>--requires</code> で要求できるもので、パッケージが提供する機能の一覧。
<code>--requires, -R</code>	パッケージが必要とする機能の一覧。
<code>--scripts</code>	インストール時のスクリプト (<code>preinstall</code> (インストール前), <code>postinstall</code> (インストール後), <code>uninstall</code> (アンインストール時))

たとえば `rpm -q -i wget` というコマンドを入力すると、例7.2「`rpm -q -i wget`」(125 ページ) のように出力が行なわれます。

例 7.2 `rpm -q -i wget`

```

Name           : wget                               Relocations: (not relocatable)
Version        : 1.11.4                             Vendor: opensUSE
Release        : 1.70                                Build Date: Sat 01 Aug 2009
09:49:48 CEST
Install Date: Thu 06 Aug 2009 14:53:24 CEST      Build Host: build18
Group          : Productivity/Networking/Web/Utilities  Source RPM:
wget-1.11.4-1.70.src.rpm
Size           : 1525431                             License: GPL v3 or later
Signature      : RSA/8, Sat 01 Aug 2009 09:50:04 CEST, Key ID b88b2fd43dbdc284
Packager       : http://bugs.opensuse.org
URL            : http://www.gnu.org/software/wget/
Summary        : A Tool for Mirroring FTP and HTTP Servers
Description    :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

また `-f` オプションは、完全なファイル名としてフルパス (絶対パス) で指定した場合にのみ動作します。複数のファイルを指定することもできます。たとえば下記のコマンドでは、

```
rpm -q -f /bin/rpm /usr/bin/wget
```

以下のような結果になります:

rpm-4.8.0-4.3.x86_64
wget-1.11.4-11.18.x86_64

ファイル名の一部だけがわかっているような場合は、下記 (例7.3「パッケージの検索スクリプト」 (126 ページ)) に示すようなシェルスクリプトをご利用ください。スクリプトのパラメータとしてファイル名の一部を与えると動作します。

例 7.3 パッケージの検索スクリプト

```
#!/bin/sh
for i in $(rpm -q -a -l | grep $1); do
    echo "\"$i\" is in package:"
    rpm -q -f $i
    echo ""
done
```

また、rpm -q --changelog rpm コマンドを利用すると、指定したパッケージ (この場合は rpm パッケージ) について詳しい変更情報を、日付順に表示することもできます。

インストール済みの RPM データベースを利用することで、検証作業を行なうことができます。それぞれ -V, -y, --verify のいずれかを指定してください。これらのオプションを指定すると、rpm は各パッケージに含まれるファイルについて、インストール後に変更されたものを一覧表示します。なお、何が変更されたのかについては、8 文字のシンボルで表示します。それぞれ下記のような意味になっています:

表 7.2 RPM 検証オプション

S	MD5 チェックサムの相違
S	ファイルサイズの相違
L	シンボリックリンクの相違
T	更新日時の相違
D	メジャー／マイナーデバイス番号の相違
U	所有者の相違

対象が設定ファイルである場合は、`c` も合わせて表示されます。たとえば `/etc/wgetrc` ファイルが変更されている場合 (`wget` パッケージ) は、下記のように表示されます:

```
rpm -V wget
S.5....T c /etc/wgetrc
```

RPM データベースは、`/var/lib/rpm` ディレクトリ内に ファイルで保存されています。たとえば `/usr` パーティションが **1 GB** (ギガバイト) のサイズであった場合、特に完全な更新を行なったりした場合は、データベースがおおよそ **30 MB** 程度のサイズになります。もしもデータベースが本来よりもずっと大きくなってしまったような場合は、`--rebuilddb` オプションを利用してデータベースの再構築を行なうことができます。なお、この作業を行なう場合は、事前に古いデータベースのバックアップを採取しておいてください。`cron` コマンドのスクリプトである `cron.daily` では、データベースを日次でバックアップし (`gzip` 圧縮)、それらを `/var/adm/backup/rpmdb` 内に保存するようになっています。コピーを保持しておく数については、`/etc/sysconfig/backup` の設定ファイル内の `MAX_RPMDDB_BACKUPS` 変数で設定することができます (既定値: 5)。バックアップ 1 つごとに `/usr` ディレクトリ **1 GB** (ギガバイト) あたり **1 MB** (メガバイト) 程度が必要です。

7.2.6 ソースパッケージからのコンパイルとインストール

全てのソースパッケージには、`.src.rpm` という拡張子 (ソース RPM という意味) が付けられています。

注記: ソースパッケージのインストール状態について

ソースパッケージは、**YaST** を利用してインストールメディアからお使いのハードディスクにコピーし、展開することができます。ただし、パッケージマネージャ側ではインストール済み ([i]) としてマーク されることはありません。これは **RPM** データベースは **インストール済みのオペレーティング**

グシステムの ソフトウェアだけを登録すべき場所であるため、ソースパッケージについては **RPM** データベース内に登録されないためです。 です。従って、ソースパッケージを「インストール」しても、ソースコードがシステム内に追加 されるだけの処理しか行ないません。

rpm コマンドや rpmbuild コマンドでは、 /usr/src/packages ディレクトリ以下の下記のサブディレクトリを使用します (/etc/rpmrc ファイルのような形で別途設定した場合を除きます):

SOURCES

オリジナルのソースファイル(.tar.bz2 や .tar.gz などのファイル)およびディストリビューション固有の調整(多くは .diff や .patch のファイル)が配置されます。

SPECS

.spec ファイルが配置されます。これはメタ **Makefile** とも 言うべきもので、 構築処理をコントロールするための ファイルです。

BUILD

全てのソースコードを展開し、修正し、コンパイルするためのディレクトリです。

RPMS

構築済みのバイナリパッケージを保管するディレクトリです。

SRPMS

ソース **RPM** を保管するディレクトリです。

YaST を利用してソースパッケージをインストールすると、全ての必要な コンポーネントが /usr/src/packages 以下にインストール されます: ソースコードと細かい調整などが **SOURCES** サブディレクトリに、関連する .spec ファイルが **SPECS** サブディレクトリにインストールされます。

警告

システムコンポーネント (glibc, rpm, sysvinit など) については、 構築を行なわないことをお勧めします。これらを書き換えてしまうと、システム全体の安定性が損なわれる可能性があるためです。

下記の例は、`wget.src.rpm` パッケージを利用した場合の例です。ソースパッケージをインストールすると、下記のようなファイルがそれぞれ存在しているはずです:

```
/usr/src/packages/SOURCES/wget-1.11.4.tar.bz2
/usr/src/packages/SOURCES/wgetrc.patch
/usr/src/packages/SPECS/wget.spec
```

ここから `rpmbuild -bX /usr/src/packages/SPECS/wget.spec` と実行すると、コンパイル作業を行なうことができます。`x` には、構築処理の様々な段階を指定します。それぞれ下記のように指定します:

`-bp`

ソースコードを `/usr/src/packages/BUILD` ディレクトリ 以下に展開し、修正を適用します。

`-bc`

`-bp` に加え、コンパイル作業までを行ないます。

`-bi`

`-bp` に加え、構築済みのソフトウェアのインストールを行ないます。注意: パッケージが **BuildRoot** 機能に対応していない場合は、既存の設定ファイルを上書きしてしまう可能性があります。

`-bb`

`-bi` に加え、バイナリパッケージの作成作業を行ないます。コンパイルが成功すると、バイナリは `/usr/src/packages/RPMS` ディレクトリ内に作成されます。

`-ba`

`-bb` に加え、ソース **RPM** の作成作業を行ないます。コンパイルが成功すると、ソース **RPM** は `/usr/src/packages/SRPMS` ディレクトリ内に作成されます。

`--short-circuit`

いくつかの手順を飛ばします。

バイナリ **RPM** を作成すると、`rpm -i` や `rpm -U` コマンドを利用してインストールできるようになります。インストール後は、**RPM** データベース内に登録されるようになります。

7.2.7 build を利用した RPM パッケージのコンパイル

多くのパッケージをコンパイルするにあたって危険となるのは、構築処理の最中に実行中のシステムに対し、予期しないファイルが追加されてしまうことです。これを防ぐため、パッケージを構築するための設定済み環境を構築するための build をお使いください。この chroot 環境を確立するため、build スクリプトを完全なパッケージツリーを指定して実行してください。このツリーはハードディスクや NFS, DVD から利用することができます。build --rpms ディレクトリのようにして実行してください。なお rpm とは異なり、build コマンドは SPEC ファイルをソースディレクトリ内で検索します。たとえば /media/dvd 以下にマウントされた DVD のシステムで上記の例のように wget を構築するには、下記のコマンドを root で実行します:

```
cd /usr/src/packages/SOURCES/  
mv ../SPECS/wget.spec .  
build --rpms /media/dvd/suse/ wget.spec
```

上記を実行すると、最小限の環境が /var/tmp/build-root ディレクトリ以下に構築され、パッケージはその環境下でコンパイルされます。作業が完了すると、作成されたパッケージは /var/tmp/build-root/usr/src/packages/RPMS ディレクトリに置かれます。

また、build スクリプトは複数の追加オプションを設定することができます。たとえばスクリプトに対して自身の RPM を使用するよう指定したり、構築環境の初期化を省略したり、rpm コマンドについて上記に示した各段階のいずれかを実行させたりすることができます。さらなる情報については、build --help コマンドを実行して表示するか、build のマニュアルページをお読みください。

7.2.8 RPM アーカイブと RPM データベース向けのツール

Midnight Commander (mc) では、RPM アーカイブの内容を表示したり、それらの一部をコピーしたりすることができます。このソフトウェアではアーカイブを仮想的なファイルシステムとしてアクセスすることができるため、Midnight Commander で実行できる全てのメニューを RPM アーカイブの内容に

対して実行することができます。F3 で HEADER ファイルの表示を行なうことができるほか、カーソルキーと **Enter** キーでアーカイブ構造を閲覧したりすることができます。また、F5 でアーカイブの内容をコピーすることもできます。

また、完全機能のパッケージマネージャを必要とする場合は、YaST モジュール (第3章 ソフトウェアのインストールと削除 (67 ページ) をお読みください) をお使いください。

パート III. システム管理

YaST を利用したユーザ管理

インストールの際にユーザ認証方法を設定することができます。ローカル (/etc/passwd ファイルを介した) 認証のほか、ネットワークが利用できる環境であれば、NIS, LDAP, Kerberos, Samba (詳しくは 項「新規ユーザの作成」(第1章 *YaST* を利用したインストール, ↑スタートアップ) をお読みください) を利用することもできます。また、インストール後からでも *YaST* を利用してユーザの作成や修正、認証方法の変更などを行なうことができます。

それぞれのユーザにはシステム内で唯一のユーザ ID (UID) が割り当てられています。お使いのマシンにログイン可能なユーザとは別に、システム内部で使用する複数のシステムユーザも設定されています。また、それぞれのユーザは1つ以上のグループに属しています。システムユーザと同様に、システム内部で使用する複数のシステムグループも存在しています。Linux でのユーザとグループの考え方について、詳しくは 項「ユーザについてのコンセプト」(第6章 基本コンセプト, ↑スタートアップ) をお読みください。

8.1 ユーザとグループの管理ダイアログ

ユーザやグループを管理するには、*YaST* を起動して セキュリティとユーザ > ユーザとグループの管理 を選択してください。また、コマンドラインから直接 `yast2 users &` と入力しても ユーザとグループの管理 ダイアログを開くことができます。

図 8.1 YaST ユーザとグループの管理



選択したユーザの種類(ローカルユーザ, ネットワークユーザ, システムユーザ)によってメインウィンドウに表示されるタブが異なりますが、それぞれ下記のような作業を行なうことができます:

ユーザアカウントの管理

ユーザのタブを利用すると、8.2項「ユーザアカウントの管理」(137 ページ)に書かれているようにユーザアカウントの作成や修正、削除や一時的な無効化を設定することができます。パスワードポリシーの強制やホームディレクトリの暗号化、指紋読み取り装置での認証、ディスククォータの管理など、より細かい設定については8.3項「ユーザアカウントに対する追加のオプション」(140 ページ)をお読みください。

既定値の変更

ローカルユーザアカウントの作成は、新しいユーザの既定値にある設定をもとに行なわれます。既定で属するグループの変更やホームディレクトリの既定パス、アクセス権について、詳しくは8.4項「ローカルユーザに対する既定値の変更」(147 ページ)をお読みください。

ユーザのグループへの割り当て

個別のユーザに対してグループの割り当てを変更する方法について、詳しくは 8.5 項「ユーザに対するグループの割り当て」(148 ページ)をお読みください。

グループの管理

グループのタブを利用すると、グループの追加や変更、既存のグループの削除を行なうことができます。詳しくは 8.6 項「グループの管理」(149 ページ)をお読みください。

ユーザ認証方法の変更

NISやLDAPのようにユーザ認証機能を提供しているネットワークに接続しているマシンであれば、**認証設定**タブからそれらを設定することができます。詳しくは 8.7 項「ユーザ認証方法の変更」(151 ページ)をお読みください。

ユーザとグループの管理について、それぞれのダイアログは似通った構成になっています。それぞれダイアログの冒頭部にある適切なタブを選択して、それぞれ ユーザとグループを切り替えてください。

フィルタオプションを利用すると、ユーザやグループの一覧表示にフィルタを設定することができます: ユーザまたはグループのタブを選択した状態からフィルタの設定を押し、それぞれ分類(ローカルユーザ, **LDAP ユーザ** (LDAP を使用するよう設定している場合のみ) など)を選択すると、選択した条件に該当するユーザやグループだけが表示されるようになります。また、個別のフィルタを設定したい場合は **フィルタの設定** > **フィルタのカスタマイズ**を選択してください。

選択するフィルタにもよりますが、下記に示すオプションや機能が一部ご利用いただけない場合があります。

8.2 ユーザアカウントの管理

YaSTでは、ユーザアカウントの作成や修正、削除や一時的な無効化を設定することができます。知識のあるユーザや管理者である場合を除き、ユーザアカウントの修正は行なわないことをお勧めします。

注記: 既存のユーザに対するユーザ ID の変更

ファイルのアクセス権はユーザ名ではなく、ユーザ ID に結びつけられています。ユーザ ID を変更すると、その変更を適用する段階で、該当するユーザのホームディレクトリに設定されているユーザ ID を書き換えます。ただし、ホームディレクトリ以外に作成されたファイルについては、ユーザ ID を変更すると元のユーザの所有権は失われてしまい、手作業による修正が必要になります。

下記では、既定のユーザアカウント設定手順が示されています。たとえばユーザやグループに対する自動ログインやパスワード無しでのログイン、ホームディレクトリの暗号化やクォータの管理など、より細かいオプション設定については 8.3 頁「ユーザアカウントに対する追加のオプション」(140 ページ)をお読みください。

手順 8.1 ユーザアカウントの追加や変更

- 1 YaST から **ユーザとグループの管理** ダイアログを開き、**ユーザタブ**を選択します。
- 2 **フィルタの設定**を利用して、管理したいユーザの種類を選択します。ダイアログには、システム内に設定されているユーザと、そのユーザが属するグループが表示されます。
- 3 既存のユーザに対するオプション設定を変更するには、ユーザを選択してから **編集**を押します。

新しいユーザを作成するには、**追加**を押します。

- 4 まずは最初のタブで必要なユーザ情報を入力します。ユーザ名(ログイン時に使用します)とパスワードをそれぞれ入力してください。これらはユーザを作成する際に必要な項目です。ここで **OK**を押すと、システムは自動的にユーザ ID を割り当て、その他の値については全て既定値のまま作成されます。
- 5 システム通知関連のメールをユーザ側で受信したい場合は、システムメールの受信を選択してください。この設定を有効にすると、root に対するメールアカウントの別名が作成され、最初に root でログインしなくてもシステムメールを読むことができますようになります。

- 6 ユーザIDやユーザのホームディレクトリのパスなど、詳細設定を調整したい場合は、それぞれ **詳細** タブから修正してください。

既存のユーザに対してホームディレクトリの場所を変更したい場合は、新しいホームディレクトリのパスを項目に入力し、**新しい場所に移動** を選択して現在のホームディレクトリから内容を移動させてください。選択しない場合は、新しいホームディレクトリが作成されるだけで、既存のデータはそのまま元の場所に置かれたままになります。

- 7 ユーザに対して定期的なパスワード変更を求めたりなど、パスワードに関するオプションを設定するには、**パスワードの設定** タブを選択してオプションを設定してください。詳しくは 8.3.2 項「パスワードポリシーの強制」(141 ページ)をお読みください。
- 8 オプション設定が全て完了したら、最後に **OK** ボタンを押してください。
- 9 **熟練者用オプション > 変更を今すぐ書き込む** を選択すると、ユーザとグループの**管理**ダイアログを閉じることなく全ての設定を保存することができます。また、**OK** ボタンを押すと**管理**ダイアログを閉じて設定を保存します。この時点で、新規に作成したユーザは指定したログイン名とパスワードでログインできるようになります。

ティップ: ユーザ ID の同期

ネットワーク環境との統合を行ないたいラップトップコンピュータに対して、既に他のコンピュータ上にユーザIDを持つユーザを新規に作成したい場合は、ネットワーク上のユーザIDとローカル環境のユーザIDを一致させておくと便利です。これにより、ネットワーク上に直接ファイルを作成したファイルも、「オフライン」で作成したファイルも同じファイルアクセス権で賄うことができるようになります。

手順 8.2 ユーザアカウントの無効化と削除

- 1 YaST から **ユーザとグループの管理** ダイアログを開き、**ユーザ** タブを開きます。
- 2 削除を行なうことなく一時的にユーザアカウントを無効化させたい場合は、対象のユーザを一覧から選択して **編集** ボタンを押します。あとは **ログインを禁止する** を選択してください。ユーザは再度アカウントを有効にするまでの間、そのマシンにログインできなくなります。

- 3 ユーザアカウントを削除するには、対象のユーザを一覧から選択して *削除* を押します。削除の際、ユーザのホームディレクトリを削除するか、もしくはデータをそのまま残すかを選択することができます。

8.3 ユーザアカウントに対する追加のオプション

既定のユーザアカウントに対する設定に加え、openSUSE® ではパスワードポリシーの強制や暗号化したホームディレクトリ、ディスククォータなどをユーザおよびグループに設定することができます。

8.3.1 自動ログインとパスワード無しのログイン

KDE や GNOME のデスクトップ環境をお使いの場合、常に特定のユーザで使用する際に選択する *自動ログイン* 機能や、パスワード無しでユーザだけを選択してログインを行なう *パスワード無しログイン* を競っていることができます。自動ログインでは、システムが起動すると自動的にデスクトップ環境が表示される ところまで動作するようになります。この機能は同時に 1 つのユーザに対してしか設定することができません。パスワード無しでのログインは、ログイン マネージャ内でユーザを選択するだけで、パスワードを入力することなくログインできるようにするための機能です。

警告: セキュリティリスク

自動ログイン や *パスワード無しログイン* をマシン上の 1 人以上のユーザに設定することは、セキュリティ 上のリスクとなります。認証を行なうことなく任意の利用者がお使いのシステム とデータにアクセスできてしまうためです。もしもお使いのシステムに機密 データを含むような場合は、この機能を利用しないでください。

自動ログインやパスワード無しでのログインを行ないたい場合は、YaST から *ユーザとグループの管理* を開き、*熟練者用オプション > ログイン設定* を開いて設定してください。

8.3.2 パスワードポリシーの強制

複数のユーザが設定されているシステムでは、基本的なパスワードポリシーを強制することは、セキュリティ向上につながる良い案です。ユーザは自身のパスワードを定期的に変更し、簡単に突破されないよう強度の強いパスワードを使用するよう設定することができます。ローカルユーザに対しては、下記の手順で行ないます:

手順 8.3 パスワードの設定

- 1 **YaST から ユーザとグループの管理** ダイアログを開き、 **ユーザ** タブを選択します。
- 2 パスワードオプションを変更したいユーザを選択して、 **編集** ボタンを押します。
- 3 **パスワードの設定** のタブに切り替えます。タブ内にはユーザが最後にパスワードを変更した日付が表示されます。
- 4 次回のログイン時にパスワード変更を行なわせたい場合は、 **パスワード変更を強制する** を選択します。
- 5 パスワードの定期的な変更を強制するには、 **同じパスワードを使用できる最大の日数** と **同じパスワードを使用する最短日数** をそれぞれ指定します。
- 6 パスワードの期限が切れる前にユーザに対して通知を行ないたい場合は、 **パスワード失効警告日数** を指定します。
- 7 パスワードの期限が切れた後にログインできる期間を制限したい場合は、 **パスワードの期限切れ後にログインできる日数** を指定します。
- 8 また、パスワードの期限を日付指定で行なうこともできます。 **有効期限** の欄に `YYYY-MM-DD` の書式で入力してください。
- 9 それぞれのオプションと既定値について詳しく知るには、 **ヘルプ** ボタンを押してください。
- 10 最後に設定を保存するため、 **OK** を押します。

8.3.3 暗号化ホームディレクトリの管理

窃盗やハードディスクの盗難からホームディレクトリのデータを守るため、ユーザ向けのホームディレクトリに対して暗号化を設定することができます。これらは LUKS (Linux Unified Key Setup) と呼ばれる方法で暗号化を行ないます。これは、ユーザが作成したイメージとそのイメージ用の鍵で運用を行なう方法です。イメージ用の鍵はユーザのログインパスワードで保護されていて、ユーザがシステムにログインしたときに暗号化ホームディレクトリをマウントし、ユーザから利用できるようになります。

注記: 指紋読み取り装置と暗号化ホームディレクトリ

指紋読み取り装置をご利用になる場合は、暗号化ホームディレクトリを使用することはできません。指紋読み取り装置はパスワードそのものを生成する仕組みではないため、ログイン時の暗号解除を行なうことができず、ログインが失敗します。

YaST を利用することで、新規のユーザや既存のユーザに対して暗号化ホームディレクトリを設定することができます。既存のユーザ向けに設定した暗号化ホームディレクトリの設定を変更するには、そのユーザのログインパスワードを知っておく必要があります。既定では全てのユーザデータを新しい暗号化されたホームディレクトリにコピーしますが、コピー元である暗号化されていないディレクトリからはデータは削除されません。

警告: セキュリティ制限

ユーザのホームディレクトリを暗号化しても、同じシステムを使用する他のユーザに対する強固なセキュリティにはなりません。強固なセキュリティが必要となる場合は、物理的にシステムを共用しないでください。

暗号化ホームディレクトリに関する背景情報や、より強いセキュリティを実現するために行なわなければならない作業について、詳しくは 項「Using Encrypted Home Directories」(第11章 *Encrypting Partitions and Files*, ↑セキュリティガイド)をお読みください。

手順 8.4 暗号化ホームディレクトリの作成

- 1 YaST から **ユーザとグループの管理** ダイアログを開き、**ユーザタブ** を選択します。

- 既存のユーザに対してホームディレクトリの暗号化を行なうには、ユーザを選択して **編集** を押します。

新しいユーザアカウントを作成してホームディレクトリの暗号化を行なう場合は、**追加** を押して新しいユーザアカウントの作成を行ない、まずは最初のタブで必要なユーザデータを入力します。

- その後 **詳細** タブに移動し、**暗号化ホーム ディレクトリを使用** を選択します。さらにそのユーザ向けに作成する 暗号化イメージファイルのサイズを、**ディレクトリサイズ (MB)** の欄にメガバイト単位で入力します。

既存のローカルユーザ

ユーザ情報 (E) 詳細 (D) パスワードの設定 (W) プラグイン (G)

ユーザ ID (uid) (I)

ホームディレクトリ (H) 参照 (R)...

☒ 新しい場所移动到 (M)

☒ 暗号化ホームディレクトリを使用 (U) ディレクトリサイズ (MB) (D)

ユーザ追加情報 (I)

ログインシェル (S)

既定のグループ (E)

追加のグループ (Q)

- ☐ users
- ☐ at
- ☐ audio
- ☐ avahi
- ☐ beagleindex
- ☐ bin
- ☐ cdrom
- ☐ console
- ☐ daemon
- ☐ dialout
- ☐ disk
- ☐ floppy
- ☐ ftp
- ☐ games
- ☐ gdm
- ☐ haldaemon

ヘルプ (H) キャンセル (C) OK (O)

- 設定を適用するため、**OK** を押します。
- 次にユーザの現在のログインパスワードについて尋ねられたら、パスワードを入力します。
- 熟練者用オプション>変更を今すぐ書き込むを選択して、管理ダイアログを閉じずに変更を保存するか、もしくは **OK** ボタンを押して設定を保存して終了します。

手順 8.5 暗号化ホームディレクトリの設定変更と削除

任意のタイミングでイメージファイルのサイズを変更したり、ホームディレクトリの暗号化設定を無効化したりすることができます。

- 1 YaST から ユーザとグループの管理 ダイアログを開き、 ユーザタブを選択します。
- 2 一覧からユーザを選択し、 **編集** を押します。
- 3 暗号化を無効化したい場合は、 **詳細** タブを選択して 暗号化ホームディレクトリの使用 の選択を外します。

そのユーザに対する暗号化イメージファイルのサイズを拡大または縮小したい場合は、 **ディレクトリサイズ (MB)** の欄にメガバイト単位で 入力します。
- 4 設定を適用するため、 **OK** を押します。
- 5 次にユーザの現在のログインパスワードについて尋ねられたら、パスワードを入力します。
- 6 熟練者用オプション>変更を今すぐ書き込むを選択して、管理ダイアログを閉じずに変更を 保存するか、もしくは **OK** ボタンを押して設定を保存して 終了します。

8.3.4 指紋読み取り装置の使用

お使いのシステムに指紋読み取り装置が接続されている場合、ログインやパスワードといった標準的な認証方法に加え、生体認証を利用することもできます。指紋読み取り装置を登録すると、指紋読み取り装置で指をなぞるか、もしくはパスワードを入力するかのいずれかの方法でログインを行なうことができます。

指紋は YaST で登録することができます。指紋認証に関する設定方法や使用方法について、詳しくは 第7章 *Using the Fingerprint Reader* (↑セキュリティガイド) をお読みください。また、対応しているデバイスの一覧は、 http://reactivated.net/fprint/wiki/Supported_devices をご覧ください。

8.3.5 クォータの管理

システムの容量が気づかないうちに使い果たされてしまうような事態を防ぐため、システム管理者はユーザやグループに対して割り当て量(クォータ)を設定することができます。クォータは複数のファイルシステムに対して設定することができ、利用可能なディスク領域や作成可能な **inode** (インデックスノード) 数で制限を行ないます。**inode** はファイルやディレクトリ、もしくはそれ以外の何らかのファイルシステムオブジェクトに対して割り当てられる基本情報を保持するためのデータ構造です。ファイルシステムオブジェクトに対して、ファイル名とその内容を除く全ての属性(ユーザやグループの所有権、読み込み／書き込み／実行のパーミッション)を保持します。

openSUSE では、それぞれソフトクォータとハードクォータを設定することができます。ソフトクォータは一般に、ユーザに対して上限に近いことを知らせる警告レベルで、ハードクォータはその書き込み要求が拒否されるレベルを設定します。さらに特定の容量の範囲で、一時的なユーザやグループの制限超過を許すための猶予期間を設定することもできます。

手順 8.6 パーティションに対するクォータの有効化設定

あるユーザやグループにクォータを設定するには、まず YaST の熟練者向けパーティション設定を利用してクォータ機能を有効化しておく必要があります。

- 1 YaST から システム > パーティション設定を選択し、はいを押して先に進めます。
- 2 熟練者向けパーティション設定では、クォータを有効にしたいパーティションを選択して、編集を押します。
- 3 *fstab* オプションを押し、クォータサポートを有効にするを選びます。
quota パッケージがインストールされていない場合は、その後の段階で確認メッセージが表示されますので、はいを押して進めてください。
- 4 最後に変更内容を確認し、熟練者向けパーティション設定を終了してください。

手順 8.7 ユーザやグループに対するクォータの設定

ユーザやグループに対するソフト／ハードクォータ、および猶予期間をそれぞれ設定します。

- 1 YaST から ユーザとグループの管理 を起動し、クォータ を設定したいユーザまたはグループを選択して、編集 を押します。
- 2 プラグインタブに移動し、ユーザクォータの管理 を選択して 起動を押します。すると、クォータの設定ダイアログが表示されます。
- 3 ファイルシステムから、クォータを適用したいパーティションを選択します。

- 4 サイズ制限の項目では、ディスク領域の制限量を指定することができます。このパーティションに対してユーザやグループが書き込むことのできるサイズを、1 KB (キロバイト) 単位のブロック数で指定します。それぞれソフト制限とハード制限に設定を行ってください。
- 5 あわせて、このパーティションに対してユーザやグループから作成することのできる inode 数の上限も指定することができます。inode 制限の項目にあるソフト制限やハード制限に入力を行ってください。

- 6 ユーザが既に指定したサイズや **inode** のソフト制限を超過している場合にのみ、猶予期間を設定することができます。そうでない場合は、時間関連の入力項目は設定できません。ユーザがその制限を超過することを許可する時間を指定します。
- 7 設定を終えたら、**OK** を押します。
- 8 熟練者用オプション>変更を今すぐ書き込むを選択して、管理ダイアログを閉じずに変更を保存するか、もしくは **OK** ボタンを押して設定を保存して終了します。

openSUSE では、設定したクォータ値を超過した場合にディスクの使用形態を操作したり、電子メールで通知を流したりするためのツールである `repquota` や、`warnquota` のようなコマンドラインツールも提供しています。また `quota_nld` では、そのクォータ値超過が発生した旨のカーネルメッセージを **D-BUS** に転送することができます。詳しくは `repquota`, `warnquota`, `quota_nld` の各マニュアルページをお読みください (root のパスワードが必要です)。

8.4 ローカルユーザに対する既定値の変更

新しいユーザを作成する際、**YaST** ではいくつかの既定値を利用します。たとえばユーザが属する既定のグループや追加のグループ、ホームディレクトリのアクセス権設定などがあります。これらの既定値は要件にあわせて変更することができます:

- 1 **YaST** から ユーザとグループの管理 ダイアログを開き、新しいユーザの既定値タブを選択します。
- 2 新しいユーザを作成する際に自動で所属する既定のグループを変更するには、既定のグループで選択を行ないます。
- 3 新しいユーザのセカンダリグループを変更するには、セカンダリグループ欄でグループ名を追加または削除します。複数のグループ名を指定する際は、それぞれカンマで区切ります。

- 4 /home/ユーザ名 を新規 ユーザに対する既定のホームディレクトリとして使用したくない場合は、ホームディレクトリのパスプレフィックスを変更します。
- 5 新しく作成するホームディレクトリに対して設定する既定のパーミッションを変更するには、ホームディレクトリ用の *umask* の値を変更します。*umask* についての詳細は、第10章 *Access Control Lists in Linux* (↑セキュリティガイド) と *umask* の マニュアルページをお読みください。
- 6 また、それ以外の個別のオプションについては、ヘルプボタンを押して説明をお読みください。
- 7 設定を終えたら最後に *OK* を押します。

8.5 ユーザに対するグループの割り当て

ローカルユーザは、ユーザとグループの管理ダイアログ内の新しいユーザの既定値の設定に従い、グループの割り当てが行なわれます。下記では、個別のユーザに対してグループの割り当てを修正するための方法について述べています。新規ユーザに対する既定のグループ設定を変更する場合は、8.4項「ローカルユーザに対する既定値の変更」(147 ページ)をお読みください。

手順 8.8 ユーザに対するグループの割り当て変更

- 1 YaST から ユーザとグループの管理ダイアログを開き、ユーザタブを選択します。タブを選択すると、ユーザ名とそのユーザが所属するグループの一覧が表示されます。
- 2 編集 ボタンを押し、詳細 タブに切り替えます。
- 3 ユーザが所属するプライマリグループを変更するには、既定のグループを押し、一覧からグループを選択します。
- 4 追加の(セカンダリ)グループを割り当てるには、追加のグループの一覧表示から、属するよう設定したいグループにチェックを付けます。
- 5 *OK* を押すと変更を保存することができます。

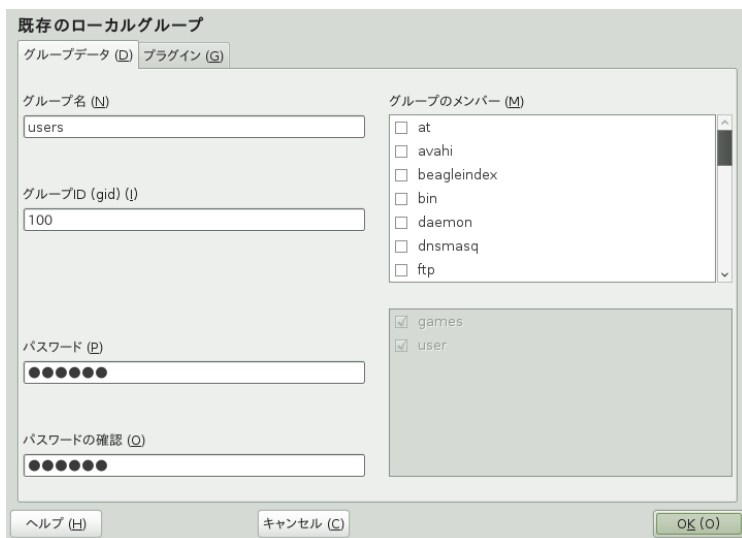
- 6 熟練者用オプション>変更を今すぐ書き込むを選択して、管理ダイアログを閉じずに変更を保存するか、もしくは **OK** ボタンを押して設定を保存して終了します。

8.6 グループの管理

YaST を利用することで、グループの追加や編集、削除を行なうことができます。

手順 8.9 グループの作成と管理

- 1 YaST から ユーザとグループの管理 ダイアログを開き、グループタブを選択します。
- 2 フィルタの設定から、管理したいグループの種類を選択します。選択を行なうと、グループの一覧が表示されます。
- 3 新しいグループを作成するには、追加を押します。
- 4 既存のグループを編集するには、対象のグループを選択して編集を押します。
- 5 表示されたダイアログ内で必要な設定または修正を行ないます。右側にある表には、グループのメンバーになりうる全てのユーザとシステムユーザが表示されています。



- 6 既存のユーザを新しいグループに追加するには、グループのメンバー 内に表示されたユーザー一覧から、追加したいユーザのチェックボックスに印を付けます。グループから外したい場合はチェックを外してください。
- 7 次に **OK** を押すと変更が保存されます。
- 8 熟練者用オプション>変更を今すぐ書き込むを選択して、管理ダイアログを閉じずに変更を 保存します。

グループを削除するには、グループメンバーが存在してはなりません。メンバーが存在しない状態にしてから、削除するグループを選択して **削除** を押してください。その後、熟練者用オプション>変更を今すぐ書き込むを選択すると、ユーザとグループの管理ダイアログを閉じることなく全ての設定を保存することができます。また、**OK** ボタンを押すと 管理ダイアログを閉じて設定を保存します。

8.7 ユーザ認証方法の変更

お使いのマシンが認証サービスの提供されているネットワーク内にある場合は、インストールの段階で認証方法を設定することができます。それぞれ下記の方法を選択することができます:

NIS

ネットワーク内の NIS サーバでユーザを統合管理している場合に選択します。詳しくは第3章 *Using NIS* (↑セキュリティガイド) をお読みください。

LDAP

ネットワーク内の LDAP サーバでユーザを統合管理している場合に選択します。LDAP について詳しくは第4章 *LDAP—A Directory Service* (↑セキュリティガイド) をお読みください。

LDAP のユーザについては、YaST ユーザモジュールから管理することもできます。LDAP ユーザに対する既定の設定などを含む全ての LDAP 設定は、YaST の LDAP クライアントモジュールから設定しておく必要があります。詳しくは項「*Configuring an LDAP Client with YaST*」(第4章 *LDAP—A Directory Service*, ↑セキュリティガイド) をお読みください。

Kerberos

Kerberos ではいったんユーザが認証されると、ネットワーク全体でそのセッションが信頼され共有されます。

Samba

SMB 認証は Linux と Windows が混在するネットワーク環境でしばしば利用されます。詳しくは第27章 *Samba* (493 ページ) および第5章 *Active Directory Support* (↑セキュリティガイド) をお読みください。

認証方法を変更するには、下記の手順で行ないます:

- 1 YaST から *ユーザとグループの管理* ダイアログを開きます。
- 2 *認証設定* タブを開きます。すると、利用可能な認証方法と現在の設定が表示されます。
- 3 認証方法を変更するには、*設定* ボタンを押して変更したい認証方法を選択します。これにより、YaST にある各クライアント 設定モジュールが起

動されます。それぞれのクライアントに関する詳しい設定方法については、下記をお読みください:

NIS: 項「Configuring NIS Clients」(第3章 *Using NIS*, ↑セキュリティガイド)

LDAP: 項「Configuring an LDAP Client with YaST」(第4章 *LDAP—A Directory Service*, ↑セキュリティガイド)

- 4 設定を終えると、ユーザとグループの管理画面に戻ります。
- 5 最後に *OK* ボタンを押して管理ダイアログを閉じます。

YaST を利用した言語と国の設定変更

異なる国で作業を行ったり、複数の言語で作業を行ったりしたい場合は、お使いのコンピュータでそれらをサポートするよう設定する必要があります。openSUSE® では異なる ロケールを同時にインストールすることができます。ロケールとは言語や国の設定を定義するパラメータで、ユーザ インターフェイスでの表示や動作に影響を与えます。

システムの第一言語はインストール時に選択され、キーボードとタイムゾーンの設定は それに合わせて設定されます。ですが、お使いのシステムに追加の言語をインストールしておき、どの言語を既定で使用するのかが選択することができます。

このような作業を行なう場合は、9.1項「システム言語の変更」(153 ページ)の手順に従って YaST 言語モジュールをお使いください。第一言語以外の言語でアプリケーションやデスクトップを起動したい場合は、第二言語をインストールすることで行ないます。

またこれらとは別に、YaST のタイムゾーンモジュールで国とタイムゾーンの設定を 調整することもできます。またタイムサーバを利用してお使いのシステムの時刻を同期 させることもできます。詳しくは 9.2項「国と時刻の設定変更」(158 ページ)をお読み ください。

9.1 システム言語の変更

デスクトップの使い方や、システム全体を切り替えるのか、それとも デスクトップ環境だけを切り替えるのかによって、実現する方法が違います：

システム言語をシステム全体で切り替える

9.1.1項「YaST を利用したシステム言語の変更」(154 ページ)や 9.1.2項「既定のシステム言語の変更」(157 ページ)で示されている手順に従って、YaST を利用して追加のローカライズパッケージをインストールし、既定の言語に設定します。変更はログインをし直すことで反映されます。システム全体に反映させるには、システムを再起動するか全ての実行中サービス／アプリケーション／プログラムを起動し直してください。

デスクトップでのみ言語を切り替える

下記に示す手順でお使いのデスクトップ環境向けに必要な言語パッケージをインストールしたら、お使いのデスクトップ環境で提供されているコントロールセンターから、言語を切り替えることができます。KDE をお使いの場合は手順「地域設定の調整」(↑KDE ユーザガイド)を、GNOME をお使いの場合は項「言語の設定」(第3章 設定のカスタマイズ, ↑GNOME ユーザガイド)をそれぞれお読みください。X システムを再起動すると、お使いのデスクトップ環境は新しい言語設定に従って表示されるようになります。お使いのデスクトップ環境に属していないアプリケーションについてはこの変更が反映されず、YaST で設定した言語のままで表示されます。

特定のアプリケーションで一時的に言語を切り替える方法

ある特定のアプリケーションだけを他の言語 (既に YaST から言語環境をインストールしてあるものとします) で起動することもできます。これを行なうには、9.1.3項「個別のアプリケーションにおける言語の切り替え」(157 ページ)に示されている方法でコマンドラインから言語コードを指定します。

9.1.1 YaST を利用したシステム言語の変更

YaST では 2 種類の異なる言語カテゴリを有しています:

第一言語

第一言語とは YaST がシステム全体に設定する言語で、YaST 自身とデスクトップ環境にも適用されます。この言語は、ユーザ側で特段の言語指定が行なわれない場合に適用される言語になります。

第二言語

お使いのシステムを多言語対応にするには、第二言語のインストールを行ないます。第二言語としてインストールされた言語は、特定の状況下で

のみ選択されます。たとえば特定の言語でワードプロセッサを起動したりしたい場合は、第二言語の 設定を利用して作業を行ないます。

なお、追加の言語をインストールする前に、新しい言語をインストールした後どれを 既定のシステム言語 (第一言語) として使用するのかを決めておく必要があります。

YaST の言語モジュールにアクセスするには、YaST を起動して システム > 言語 を選択します。これ以外にも、root になっている状態で コマンドライン から `yast2 language &` を実行すると、言語ダイアログを直接表示することができます。



手順 9.1 追加言語のインストール

追加の言語をインストールする場合、YaST では root に対する既定のロケール設定を指定することができます。詳しくはステップ 4 (156 ページ) をお読みください。root ユーザのロケール設定のオプションでは、`/etc/sysconfig/language` ファイルのうち、root 向けのロケール設定 (`LC_*`) を指定します。この値は通常ユーザと同じロケールを使用するように設定することができるほか、言語の変更について影響を受けないように設定するか、`RC_LC_CTYPE` の変数だけを通常ユーザと同じ値に設定するかを選ぶことができます。この変数は言語固有の機能呼び出しについてローカライズを設定します。

- 1 YaST 言語モジュールから追加の言語をインストールするには、インストールしたい **第二言語** を選択します。
- 2 特定の言語を既定の言語として設定するには、**第一言語** の欄で設定を行います。
- 3 必要であれば、新しい第一言語にあわせてキーボードやタイムゾーンの設定を調整することもできます。

ティップ

キーボードやタイムゾーンの高度な設定は、**YaST から ハードウェア > システムキーボード 配列** または **システム > 日付と時刻** から行なうことができます。詳しくは 9.2 項 「国と時刻の設定変更」 (158 ページ) をお読みください。

- 4 root ユーザに対する固有の言語設定を変更するには、**詳細** を押します。
 - 4a 必要に応じて **root ユーザのロケール設定** を設定します。詳しくはヘルプボタンを押してください。
 - 4b root に対して **UTF-8 エンコーディング** を使用するかどうかを選択します。
- 5 第一言語の一覧内にお使いのロケールが含まれていない場合は、**言語** を押して言語を選択してください。ただし、ここから選択を行なった場合、翻訳が完全ではない場合があります。
- 6 最後に設定内容を確認し、**OK** を押します。第二言語を 選択した場合は、YaST が追加言語用のソフトウェアパッケージをインストール します。

これでお使いの言語を多言語に対応させることができました。ただし、第一言語以外の言語でアプリケーションを起動するには、9.1.3 項 「個別のアプリケーションにおける言語の切り替え」 (157 ページ) に書かれている方法で言語を 指定する必要があります。

9.1.2 既定のシステム言語の変更

- 1 既定のシステム言語を切り替えるには、YaST 言語モジュールを起動します。
- 2 第一言語で新しいシステム言語を選択します。

重要項目: 以前のシステム言語の削除

第一言語を異なるものに切り替えた場合、お使いのシステムからは以前のシステム言語に対応するパッケージが削除されます。既定のシステム言語を切り替えるものの、以前のシステム言語を第二言語として使用したい場合は、第二言語の欄に必要なチェックボックスを付けてください。

- 3 必要に応じてキーボードとタイムゾーンの設定を調整します。
- 4 最後に **OK** ボタンを押して変更を保存します。
- 5 YaST で変更を適用したら、X サーバの再起動を行ないます (たとえばいったん ログアウトし、ログインし直します)。これで YaST とデスクトップアプリケーションが新しい言語設定で動作するようになります。

9.1.3 個別のアプリケーションにおける言語の切り替え

YaST から必要な言語をインストールしたら、その言語で単一のアプリケーションが起動できるようになります。

標準 X および GNOME アプリケーション

下記のようなコマンドラインでアプリケーションを起動します:

`LANG=言語 アプリケーション`

たとえば **f-spot** をドイツ語で起動したい場合は、`LANG=de_DE f-spot` のように入力します。ドイツ語以外の言語については、それぞれ適切な言語コードを指定する必要があります。全ての言語について一覧を表示するには、`locale -av` を実行してください。

KDE アプリケーション

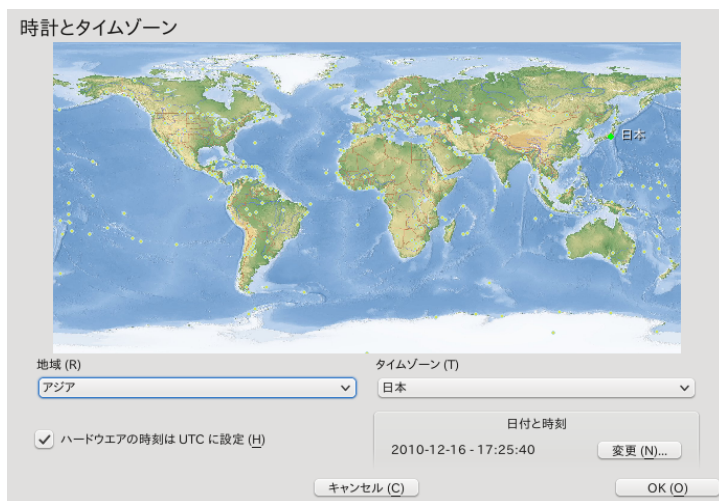
下記のコマンドラインでアプリケーションを起動します:

```
KDE_LANG=言語 アプリケーション
```

たとえばドイツ語で **digiKam** を起動するには、`KDE_LANG=de digikam` のように入力します。他の言語の場合は、それぞれ適切な言語コードを入力してください。

9.2 国と時刻の設定変更

YaST の日付と時刻のモジュールを利用することで、ご利用の地域でのシステム日時とタイムゾーンを設定することができます。YaST のモジュールにアクセスするには、YaST を起動して **システム > 日付と時刻** を選択します。これ以外にも、`root` になっている状態でコマンドラインから `yast2 timezone &` を実行すると、**時計とタイムゾーン** ダイアログを直接表示することができます。



まずはヨーロッパのように地域を選択します。あとは設定したいタイムゾーンを選択してください。たとえば **ドイツ** を選択します。

お使いのシステムでどのようなオペレーティングシステムが動作しているのかによりませんが、ハードウェア時刻は以下のように設定してください:

- お使いのシステムで **Microsoft Windows** のような他のシステムが動作している場合は、**UTC**ではなくローカル時刻を使用しているはずです。この場合は、ハードウェアの時刻は **UTC** に設定のチェックボックスのチェックを外してください。
- お使いのシステムで **Linux** だけが動作している場合は、ハードウェア時刻は **UTC** に設定してください。こうすることで、標準時から夏時間 (サマータイム) に自動的に切り替えることができます。

日付と時刻は手動で変更することができるほか、**NTP** サーバを利用した時刻同期を行なうこともできます。時刻同期は継続的に行なうこともできますし、一回だけ行なうこともできます。

手順 9.2 手動での日付と時刻の調整

- 1 **YaST** タイムゾーンモジュールから、日付と時刻を変更するため **変更** ボタンを押します。
- 2 **手動** を選択し、それぞれ日付と時刻の値を入力します。
- 3 **了解** ボタンで設定を完了します。

手順 9.3 **NTP** サーバを利用する場合の日付と時刻の設定

- 1 日付と時刻を変更するため、**変更** ボタンを押します。
- 2 **NTP** サーバと**同期** を選択します。
- 3 **NTP** サーバが設定されていない場合は、**NTP** サーバのアドレスを入力します。

システムの日付と時刻の変更

☐ 手動

現在の時刻
17:28:55

現在の日付
2010-12-16

☒ 今すぐ時間を変更

☒ NTP サーバと同期

NTP サーバのアドレス (N)
jp.pool.ntp.org

☐ NTP をデーモンとして実行する (R)

☒ NTP 設定の保存 (S)

- 4 システムの時刻を今すぐに合わせるには、 *今すぐ同期* を押します。
- 5 この NTP の設定を今後も使用したい場合は、 *NTP 設定の保存* を選択しておきます。
- 6 設定ボタンを押すと、NTP について詳細な設定を行なうことができます。詳しくは25.1項「YaSTを利用したNTPクライアントの設定」(468 ページ)をお読みください。
- 7 最後に *了解* を押して設定を保存してください。

10

テキストモードでの YaST

本章では、お使いのシステムで X サーバを起動させたくない管理者や、テキストベースのインストールツールをお使いの方に向けた説明を行なっています。具体的には、YaST をテキストモードで起動して操作するための基本的な説明が記されています。

YaST のテキストモードは、ncurses ライブラリを使用して擬似的なグラフィカルユーザインターフェイスを提供しています。ncurses ライブラリは既定でインストールされています。端末エミュレータなどを使用して表示させる仕組みになっていて、YaST は最小で 80x25 文字の環境で動作します。

☒ 10.1 テキストモードの YaST でのメインウィンドウ



YaST をテキストモードで起動すると、まずは YaST コントロールセンターが表示されます (図 10.1 をご覧ください)。メインウィンドウは 3 つの領域に分かれています。左側のフレームには、モジュールの分類が表示されています。YaST が起動した段階ではこのフレームがアクティブになっていて、白い太枠で囲まれた表示になっています。また、項目のいずれかがハイライト表示されていて、これがカーソルを表わしています。右側のフレームには、ハイライト表示された分類内に存在するモジュールの概要が示されています。下のフレームには ヘルプ と 終了 のボタンがそれぞれ用意されています。

YaST コントロールセンターを起動すると、ソフトウェアの分類が自動で選択されます。それぞれ↓と↑のキーを利用して、分類を切り替えてください。分類からモジュールを選択するには、→を押して右側のフレームを選択し、さらに↓と↑のキーを利用して個別のモジュールを選択してください。また、利用可能なモジュールは枠内に表示しきれない場合があります。この場合は、これらのキーを押し続けることで、さらなるモジュールが現われるようになっています。選択されているモジュールはハイライト表示されます。選択しているモジュールを起動するには、Enter を押してください。

様々なモジュールが表示するボタンや選択項目には、ハイライト文字が付与されている場合があります (既定では黄色で表示されます)。Tab を押していった特定のボタンや選択項目にたどり着くこともできますが、Alt + ハイライト表示の文字を押すことで、それらをダイレクトに選択することができます。たとえば YaST コントロールセンターを終了するには、終了を選択して Enter を押してもかまいませんし、Alt + Q を押して直接選択してもかまいません。

10.1 モジュール間のナビゲーション

下記に示す YaST モジュール内に現われる操作要素の説明では、全てのファンクションキーと Alt キーの組み合わせが動作し、異なるグローバル機能には割り当てられていない前提で書かれています。考えられる例外については、10.2 項「キーの組み合わせに関する制限」(164 ページ)をお読みください。

ボタンと選択リスト間のナビゲーション

それぞれ Tab キーを利用することでボタンや選択リストのあるフレーム間を移動することができます。逆順で移動するには、Alt + Tab または Shift + Tab の組み合わせをご利用ください。

選択リスト内でのナビゲーション

矢印キー (↑ と ↓) を利用することで、現在選択中の選択リスト内の項目を移動することができます。選択リスト内の項目がその枠幅よりも長いものである場合は、**Shift**+→ または **Shift**+← を押すと、水平方向に移動することができます。また、このキーの代用として、**Ctrl**+**E** または **Ctrl**+**A** を利用することもできます。この組み合わせは、コントロールセンター内で → や ← を利用した場合に 選択しているフレームが変わってしまうような場合にも利用できます。

ボタン、ラジオボタン、チェックボックス

何も書かれていない角括弧 (チェックボックス) や何も書かれていない丸括弧 (ラジオボタン) を選択するには、**Space** か **Enter** を押してください。また、このキーの代用として **Alt**+ハイライト文字 で直接選択することもできます。この場合は、**Enter** で確認する作業は 不要です。**Tab** で項目間を移動している場合は、**Enter** で選択したボタンの処理を実行するか、もしくは関連するメニュー項目を選択してください。

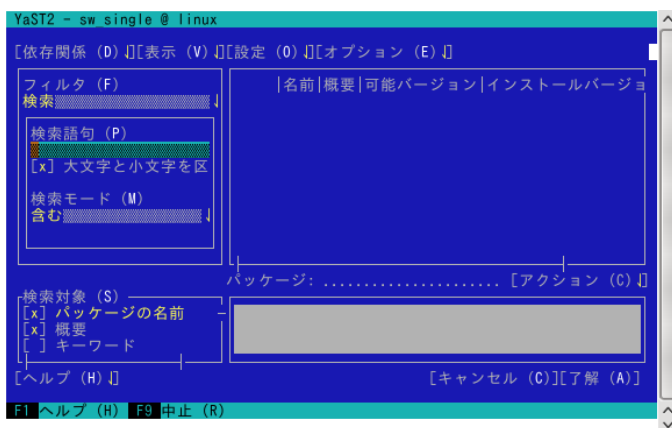
ファンクションキー

F キーとも呼ばれるもの (**F1** から **F12**) は、様々なボタンに対する素早いアクセスを提供します。利用可能なファンクションキーによるショートカットは、**YaST** の画面の下部に表示されます。ファンクションキーがどのボタンに割り当てられるかは、利用している **YaST** モジュールによって異なります。これは、それぞれのモジュールにはそれぞれ異なるボタンがあるためです (詳細, 情報, 追加, 削除など)。ただし、**了解**, **OK**, **次へ**, **完了** を示すボタンについては、共通で **F10** が割り当てられているほか、**F1** には **YaST** ヘルプが割り当てられています。

ncurses モードでのツリー構造のナビゲーション

YaST のモジュールによっては、設定ダイアログの選択を行なう目的で左側にナビゲーションツリーが表示される場合があります。ツリー構造内は、矢印キー (↑ and ↓) を利用して移動してください。また、ツリー構造を開いたり閉じたりするには、**Space** を押してください。なお、**ncurses** モードで 選択した項目のダイアログを表示させるには、ナビゲーションツリー内で **Enter** キーを押さなければならないことに注意してください。これはナビゲーションツリー内を移動する際に毎回ダイアログを表示させてしまうと、再描画に時間がかかってしまうため、移動が遅くなってしまうためです。

10.2 ソフトウェアインストールモジュール



10.2 キーの組み合わせに関する制限

お使いのウインドウマネージャがグローバルな Alt キー 機能を設定している場合は、YaST 内での Alt キーの組み合わせは利用できません。Alt や Shift キーの組み合わせは、端末側の設定によっても占有される場合があります。

Alt と Esc の置換

Alt を押す種類のショートカットは、Alt の代わりに Esc でも実行することができます。たとえば、Alt + H の代わりに、Esc + H と押しても同じ効果になります (最初に Esc を押し、続けて H を押します)

Ctrl + F と Ctrl + B を利用した前後ナビゲーション

Alt や Shift の組み合わせがウインドウマネージャや端末で占有されている場合は、Ctrl + F (前へ) や Ctrl + B (後ろへ) の組み合わせキーを使用することもできます。instead.

ファンクションキーの制限

ファンクションキーは他の機能でも使用される場合があります。特定のファンクションキーは端末で処理されるため、YaST で利用できない場合もあります。このような場合であっても、純粋なテキストコンソールであれば、常に Alt キーとファンクションキーの組み合わせが動作するはずです。

10.3 YaST コマンドラインオプション

テキストモードのインターフェイスと同様に、YaST では純粋なコマンドライン インターフェイスも提供しています。YaST のコマンドラインオプションを表示 させるには、下記のように入力してください:

```
yast -h
```

10.3.1 個別のモジュールの起動

時間を節約する目的で、個別の YaST モジュールを直接起動させることができます。モジュールを直接起動するには、下記のように入力してください:

```
yast <モジュール名>
```

お使いのシステムで利用可能なモジュールの一覧は、`yast -l` または `yast --list` と入力することで表示させることができます。たとえばネットワークモジュールであれば、`yast lan` のようにして実行します。

10.3.2 コマンドラインからのパッケージインストール

既にパッケージ名をご存じの場合で、そのパッケージがお使いの任意の有効な リポジトリから提供されるものである場合は、コマンドラインオプション `-i` を利用してパッケージをインストールすることができます:

```
yast -i <パッケージ名>
```

または

```
yast --install <パッケージ名>
```

ここで、パッケージ名には1つのパッケージ名 (gvim) を入力するか、RPM パッケージへのフルパスを入力してください。前者の場合は依存関係のチェックが行なわれますが、後者の場合はチェックを行わずにインストールします。

YaSTが提供するコマンドラインベースのソフトウェア管理ユーティリティよりもっと便利な機能を必要とする場合は、`zypper` のご利用をお勧めしま

す。この新しいユーティリティは、YaST パッケージマネージャが利用しているものと同じソフトウェア管理ライブラリをベースにしています。zypper の基本的な使い方については、7.1項「zypper の使用」(105 ページ)をお読みください。

10.3.3 YaST モジュールのコマンドラインパラメータ

YaST の機能をスクリプト内から使用する場合は、YaST が提供する個別 モジュール向けのコマンドラインサポートをご利用ください。全てのモジュールがコマンドラインに対応しているわけではありませんが、それぞれのモジュールでの 利用可能なオプションについては、下記のように入力すると表示されます:

```
yast <モジュール名> help
```

対象のモジュールがコマンドラインに対応していない場合は、対象のモジュールはテキストモードで起動され、下記のようなメッセージが表示されます:

この YaST モジュールはコマンドラインインターフェイスには対応していません。

プリンタの操作

openSUSE®では、ネットワークプリンタを含む多種のプリンタに対応しています。プリンタは手動で設定することもできますし、YaSTから設定することもできます。設定方法については項「プリンタの設定」(第2章 *YaST* を利用したハードウェアコンポーネントの設定, ↑スタートアップ)をお読みください。また、印刷ジョブの管理にはグラフィカルなユーティリティと、コマンドラインから実行可能なユーティリティの2種類があります。プリンタがうまく動作しない場合は、11.7項「トラブルシューティング」(177 ページ)をお読みください。

openSUSE では CUPS (Common Unix Printing System) を標準印刷システムとしています。

プリンタはUSBやネットワークなどのインターフェイスのほか、プリンタの言語によっても区別を行ないます。プリンタをご購入の際は、プリンタがどのようなインターフェイス(USBやパラレルポートなど)を持っているのかと、そのインターフェイスがお使いのハードウェアで利用可能なものであるかどうか、およびプリンタが使用する言語についても、あらかじめご確認ください。プリンタは下記のような3種類の言語に分類されます：

PostScript プリンタ

PostScript は多くの Linux および UNIX システムが生成するプリンタ言語で、印刷システム内部の印刷システムで処理されるものです。プリンタ側で PostScript 文書を直接処理でき、かつ印刷システムで特別な処理が必要でない場合は、潜在的なエラー要因を減らすこともできます。

標準プリンタ (PCL や ESC/P などの言語)

これらのプリンタ言語はかなり古いものですが、今でもなおプリンタに対する新しい機能を追加しながら今に至っています。言語が既知のプリンタ言語であった場合は、Ghostscript の助けを借りて PostScript ジョブをそれぞれの言語に変換することができます。処理作業は逐次翻訳(インタプリタ)とも呼ばれます。最もよく知られた言語は、PCL (HP 社製のプリンタやそれらの類似製品などで使用されています) と ESC/P (エプソン社製のプリンタで使用されています) です。これらのプリンタ言語は一般に Linux に対応していて、十分な品質の印刷を行なうことができます。ただし Linux では特別なプリンタ機能を利用することができない場合もあります。HPLIP (HP Linux Imaging and Printing) を開発している HP 社を除き、現時点では Linux 向けのドライバを開発し、オープンソースのライセンスで Linux ディストリビュータに提供している製造元はありません。

プロプライエタリプリンタ (GDI プリンタとも呼ばれます)

これらのプリンタは、汎用的なプリンタ言語に一切対応していません。これらは非公開の独自プリンタ言語を使用していて、新しいプリンタモデルが販売される際に変更されることがあります。一般に Windows 用のドライバのみが公開されています。詳しくは 11.7.1 項「標準プリンタ言語に対応していないプリンタ」(177 ページ)をお読みください。

新しいプリンタをお買い上げの前に、下記の情報をお読みのうえ、ご希望のプリンタがどの程度 Linux に対応しているかをご確認ください:

<http://www.linuxfoundation.org/OpenPrinting/> (英語)

プリンタデータベースを公開している OpenPrinting のホームページです。データベースでは最新の Linux 対応状況が公開されています。ただし、Linux ディストリビューションには製品を公開した時点のドライバしか含まれていないため、「perfectly supported」(完全対応)と明記されたプリンタであっても、それが openSUSE の対応状況とは一致しない場合があります。そのため、このデータベースは参考情報としてご覧ください。

<http://www.cs.wisc.edu/~ghost/> (英語)

Ghostscript の Web ページです。

/usr/share/doc/packages/ghostscript-library/catalog.devices
同梱ドライバの一覧です。

11.1 印刷システムの作業手順

ユーザは印刷ジョブを作成します。印刷ジョブには印刷するデータそのもののほか、プリンタの名前や印刷キューの名前、さらにオプションでフィルタの情報やプリンタ固有の設定など、スプーラに対する情報も付加されています。

少なくともそれぞれのプリンタには、専用の印刷キューが存在していなければなりません。またスプーラは、目的のプリンタがデータを受信できるようになるまで、印刷ジョブを保持します。プリンタの準備ができると、スプーラはフィルタを通してデータを送信し、応答を受信します。

一方のフィルタは、それぞれのアプリケーションが生成したデータ(一般的には PostScript や PDF、その他にも ASCII, JPEG などである場合があります)を、印刷用 (PostScript, PCL, ESC/P など) に変換する機能を備えています。プリンタの機能は PPD ファイル内に定義します。PPD ファイルには、それぞれのプリンタに必要な機能を有効化するための、プリンタ固有のオプション設定とパラメータが書かれています。また、フィルタシステムではユーザ側で選択したオプションについても有効化を行ないます。

PostScript プリンタをお使いの場合、フィルタシステムはデータをプリンタ固有の PostScript に変換します。この場合にはプリンタドライバは必要になりません。PostScript 以外のプリンタをお使いの場合は、フィルタはプリンタ固有のデータに変換します。この場合はお使いのプリンタに合わせた適切なドライバを用意する必要があります。バックエンド側ではプリンタ固有のデータを受け取り、プリンタに対してデータを送信します。

11.2 プリンタ接続の方法とプロトコル

お使いのシステムにプリンタを接続するには、いくつかの方法があります。CUPS 印刷システムでは、ローカルプリンタとネットワーク接続のプリンタとの区別は行ないません。プリンタの接続方法についてより詳しい情報は、サポートデータベース内の *CUPS in a Nutshell* 記事 (http://old-en.opensuse.org/SDB:CUPS_in_a_Nutshell (英語)) をお読みください。

警告: 稼働中のシステムにおけるケーブル接続の変更

お使いのマシンにプリンタを接続する場合は、USB 接続の場合にのみ、稼働中の取り付け／取り外しに対応することにご注意ください。USB 接続以外の環境でお使いの場合は、システムやプリンタに障害を起こさないようにするため、接続を変更する際にはシステムをシャットダウンしてから行なってください。

11.3 ソフトウェアのインストール

PPD は解像度やその他のオプション、たとえば両面印刷機能などを定義するコンピュータ言語です。これらの定義ファイルは、CUPS で様々な印刷オプションを使用する際に必要となる者です。PPD ファイル無しの状態では、印刷データは「未加工の」状態でそのままプリンタに送信されてしまいます。openSUSE をインストールすると、それに合わせて多くの PPD ファイルもインストールされます。

PostScript プリンタを設定する最もよいアプローチは、適切な PPD ファイルを取得することです。標準インストールを行なうことでインストールされるパッケージ `manufacturer-PPDs` には、多くの PPD ファイルが含まれています。詳しくは 11.6.2 項「様々なパッケージ内にある PPD ファイル」(175 ページ) と 11.7.2 項「PostScript プリンタで適切な PPD ファイルが見つからない場合」(178 ページ) をお読みください。

新しい PPD ファイルは `/usr/share/cups/model/` ディレクトリに保管することができるほか、YaST から印刷システムに追加することもできます。詳しくは「YaST を利用したドライバの追加」(第2章 *YaST* を利用したハードウェアコンポーネントの設定、↑スタートアップ)) をお読みください)。PPD ファイルのインストールを行なうと、印刷システムから選択を行なうことができるようになります。

なお、ソフトウェアパッケージ全体のインストールを求められた場合にはご注意ください。第一にこの種類のインストール作業は openSUSE のサポート対象外になってしまうほか、印刷コマンドが異なる動作をするようになるため、他の製造元のデバイスに対してアクセスできなくなってしまう可能性があります。そのため、製造元が提供するソフトウェアをインストールすることはお勧めできません。

11.4 ネットワークプリンタ

ネットワークプリンタでは様々なプロトコルを利用することができます。これらは場合によっては、複数を同時に使用することもできます。対応しているほとんどのプロトコルは標準化されていますが、製造元によっては標準から外れた実装をしている場合もあり、この場合はごく少数のオペレーティングシステムに対してだけドライバを提供していたりすることがあります。このような場合には、不幸にも Linux のドライバはほとんど提供されていません。従って、Linux で全てのプロトコルがうまく動作するとは言い切れないのが現状です。そのため、うまく動作する設定を見つけるため、様々なオプションを試してみる必要があるかもしれません。

CUPS は ソケット, LPD, IPP, smb の各プロトコルに対応しています。

ソケット

ソケットとは、確立された TCP のソケットに対して印刷データをありのまま送信することを言います。ソケットポートとして一般的に使用されるポートとしては、9100 や 35 などがあります。デバイス URI (Uniform Resource Identifier; 統一資源識別子) の書式は、`socket://プリンタの IP アドレス:ポート` となります。たとえば、`socket://192.168.2.202:9100/` のようになります。

LPD (ラインプリンタデーモン)

LPD プロトコルは、RFC 1179 で規定されています。このプロトコルでは、印刷キューの ID などのいくつかのジョブ関連データが、実際の印刷データの前に送信されます。そのため、データ転送用に LPD プロトコルを設定する際には、あらかじめ印刷キューを指定しておかなければなりません。異なるプリンタ製造元の実装でも、印刷キューに任意の名前を設定できるほど柔軟な仕組みになっています。また場合によっては、印刷キューの名前が固定されている場合もあります。LPT, LPT1, LP1 などの名前がよく使用されます。LPD サービスが使用するポート番号は 515 です。たとえばデバイス URI は、`lpd://192.168.2.202/LPT1` のようになります。

IPP (インターネット印刷プロトコル)

IPP は比較的新しい (1999 年) HTTP ベースのプロトコルです。IPP を使用すると、他のプロトコルに比べてより多いジョブ関連のデータが送信されます。CUPS は内部のデータ転送に IPP を使用しています。なお、IPP を設定するためには印刷キューの名前を指定する必要があります。IPP が

使用するポート番号は 631 です。たとえばデバイス URI は、`ipp://192.168.2.202/ps` や `ipp://192.168.2.202/printers/ps` のようになります。

SMB (Windows 共有)

もちろん CUPS でも Windows 共有に接続されたプリンタに対応しています。この目的で使用するプロトコルは SMB で、ポート番号は 137, 138, 139 をそれぞれ使用します。デバイス URI は、`smb://ユーザ:パスワード@ワークグループ/smb.example.com/printer`, `smb://ユーザ:パスワード@smb.example.com/printer`, `smb://smb.example.com/printer` などのようになります。

なお、プリンタ側で対応しているプロトコルについても、設定前に知っておかなければなりません。製造元が必要な情報を提供していない場合は、`nmap` コマンド (`nmap` パッケージで提供されます) を利用してプロトコルを調べることができます。`nmap` コマンドは接続可能なポートを調べるためのもので、下記のようにして使用します:

```
nmap -p 35,137-139,515,631,9100-10000 printerIP
```

11.4.1 コマンドラインツールでの CUPS の管理

CUPS は `lpinfo`, `lpadmin`, `lpoptions` などのコマンドラインツールで設定を行なうことができます。これらのコマンドラインツールを利用するには、パラレルなどのデバイス URI を設定する必要があるほか、パラメータについても設定を行なう必要があります。お使いのシステムで正しいデバイス URI を知るには、`lpinfo -v | grep "://"` コマンドをお使いください:

```
# lpinfo -v | grep "://"
direct usb://ACME/FunPrinter%20XL
direct parallel:/dev/lp0
```

`lpadmin` コマンドを利用すると、CUPS サーバの管理者は印刷キューを追加したり削除したり、管理したりすることができます。印刷キューを追加するには、下記の書式をご利用ください:

```
lpadmin -p キュー -v デバイス URI -P PPD ファイル -E
```

上記を実行すると、デバイス (-v で指定したもの) は、指定した PPD ファイル (-P) 経由で キュー (-p) から利用できる ようになります。従って、手動で設定を行なう場合は、PPD ファイルとデバイス URI を知っておかなければならないことになります。

なお、オプション -E は最初には指定しないでください。全ての CUPS コマンドにおいて -E を最初に指定すると、それは暗号化接続を行なう、という意味になるためです。プリンタを有効にするには、下記のようにして -E を指定してください:

```
lpadmin -p ps -v parallel:/dev/lp0 -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

また、下記の例ではネットワークプリンタに対して同じことを行なっています:

```
lpadmin -p ps -v socket://192.168.2.202:9100/ -P \
/usr/share/cups/model/Postscript-levell.ppd.gz -E
```

lpadmin について、詳しくは lpadmin(8) のマニュアルページをお読みください。

プリンタの設定の際、いくつかのオプションを既定値として設定することができます。これらのオプションは利用する印刷ツールに依存しますが、各印刷ジョブに対して 修正を行なうことで変更を行なうことができます。また、YaST から既定のオプションを変更することもできます。コマンドラインツールの場合は 下記のようにして既定のオプションを設定してください:

1 まずは全てのオプションを表示します:

```
lpoptions -p queue -l
```

例:

```
Resolution/Output Resolution: 150dpi *300dpi 600dpi
```

設定されている既定のオプション選択は、アスタリスク (*) で表示されます。

2 lpadmin を利用してオプションを変更します:

```
lpadmin -p queue -o Resolution=600dpi
```

3 新しい設定を確認します:

```
lpoptions -p queue -l  
  
Resolution/Output Resolution: 150dpi 300dpi *600dpi
```

通常のユーザから `lpoptions` を実行すると、設定は `~/.cups/lpoptions` に書き込まれます。root から実行した場合、設定は `/etc/cups/lpoptions` に書き込まれます。

11.5 コマンドラインからの印刷

コマンドラインから印刷を行なうには、`lp -d キュー名 ファイル名` のようにして実行してください。キュー名とファイル名は、それぞれの名前に置き換えてください。

また、アプリケーションによっては、印刷に際して `lp` コマンドに依存している場合があります。この場合はアプリケーションの印刷ダイアログから、ファイル名を指定せずにコマンドを入力してください。たとえば `lp -d キュー名` のようになります。

11.6 openSUSE での特殊機能

CUPS での様々な機能が openSUSE に搭載されています。ここでは、それらの機能のうち重要なものをいくつか紹介します。

11.6.1 CUPS とファイアウォール

openSUSE を特に何も指定せずにインストールすると、`SuSEfirewall2` が有効化され、ネットワークインターフェイスは外部から到達する要求をブロックする外部ゾーンとして設定されます。`SUSEfirewall2` の設定について、詳しくは項「`SuSEfirewall2`」(第14章 *Masquerading and Firewalls*, ↑セキュリティガイド)と http://ja.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings (日本語) http://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings (英語) をお読みください。

CUPS クライアント

通常 CUPS クライアントは、ファイアウォールの内側にある信頼済みネットワーク 環境内のワークステーションで動作します。この場合、ネットワーク インターフェイスを 内部ゾーン に設定し、ネットワーク内で ワークステーションを利用できるようにすることをお勧めします。

CUPS サーバ

CUPSサーバがファイアウォールで保護されている信頼済みネットワーク環境内で動作している場合、ネットワークインターフェイスは内部ゾーンに設定しておく必要があります。ただし、特別なファイアウォールのルールを設定して CUPS の設定内でも機密を守る設定をしていない限り、信頼の置けないネットワーク環境下で CUPS サーバを設定すべきではありません。

11.6.2 様々なパッケージ内にある PPD ファイル

YaST の印刷設定では、`/usr/share/cups/model` 以下に インストールされた PPD ファイルを利用して CUPS キューを設定することしか行なうことができません。お使いのプリンタの型式に対応した PPD ファイルを発見する際に YaST では、ハードウェア検出の際に読み取った製造元と型式をもとにして PPD ファイル内の同データ (製造元と型式) を検索します。そのため YaST の印刷設定では、PPD ファイルから製造元と型式の情報を取り出してデータベース化しています。

PPD ファイルだけを利用して他の情報源を利用しない設定方法は、`/usr/share/cups/model` を自由に変更できるという利点があります。たとえばお使いのプリンタとして **PostScript** 対応のものだけをお持ちの場合、`cups-drivers` パッケージに含まれる **Foomatic PPD** ファイルや `gutenprint` パッケージに含まれる **Gutenprint PPD** ファイルは使用する必要がありません。その代わり、お使いの **PostScript** プリンタを最適に利用するため、プリンタに対応した PPD ファイルを `/usr/share/cups/model` ディレクトリに直接コピーして最適なプリンタ設定を行なうことができます (manufacturer-PPDs パッケージ内に存在していない場合)。

cups パッケージ内の CUPS PPD ファイル

cups パッケージ内の汎用 PPD ファイルは、下記に示す PostScript レベル 1 および 2 のプリンタ向けの Foomatic PPD ファイルを補完するためのものです:

- /usr/share/cups/model/Postscript-level1.ppd.gz
- /usr/share/cups/model/Postscript-level2.ppd.gz

cups-drivers パッケージ内の PPD ファイル

通常、Foomatic プリントフィルタ (foomatic-rip) は PostScript 以外のプリンタで Ghostscript とともに使用されるものです。正しい PPD ファイルであれば、`*NickName: ... Foomatic/Ghostscript driver` や `*cupsFilter: ... foomatic-rip` と書かれた項目が存在しているはずです。これらの PPD ファイルは、cups-drivers パッケージ内に含まれています。

YaST では一般的に manufacturer-PPD パッケージに含まれるドライバを優先的に使用しますが、適切なドライバが見つからない場合は Foomatic PPD ファイルのうち、`*NickName: ... Foomatic ... (recommended)` が書かれているドライバを選択します。

gutenprint パッケージ内の Gutenprint PPD ファイル

多くの非 PostScript プリンタでは、foomatic-rip の代わりに Gutenprint (従来は GIMP-Print という名前でした) に含まれる rastertogutenprint という CUPS フィルタを利用することができます。このフィルタと対応する Gutenprint PPD ファイルは、gutenprint パッケージ内に含まれています。Gutenprint PPD ファイルは /usr/share/cups/model/gutenprint/ ディレクトリ内に存在し、`*NickName: ... CUPS+Gutenprint` および `*cupsFilter: ... rastertogutenprint` と書かれている行が存在します。

manufacturer-PPDs パッケージ内にあるプリンタの製造元が提供する PPD ファイル

manufacturer-PPDs パッケージには、プリンタの製造元から自由なライセンスの元で提供されている PPD ファイルが含まれています。PostScript プリンタの場合、プリンタの製造元が提供する適切な PPD ファイルを利用することで、プリンタに備わっている全ての機能を利用できるようになるため、製造元提供の PPD ファイルで設定するのがお勧めです。YaST は manufacturer-PPDs が提供する PPD ファイルを優先的に使用しますが、モデル名が一致していない場合には利用することができません。これは manufacturer-PPDs パッケージ内の PPD ファイルに、似通った型式のうち 1 つしか記述していない (たとえば "Funprinter 12xx シリーズ" のように) 場合に発生します。この場合は、YaST を利用して手動で PPD ファイルを選択してください。

11.7 トラブルシューティング

本章では、プリンタのハードウェアおよびソフトウェア上で最もよく発生する問題について、その解決方法や回避策を掲載しています。本章で扱っている分野としては、GDI プリンタや PPD ファイル、ポートの設定などがあります。また、一般的なネットワークプリンタの問題や印刷がおかしくなる問題、キューの処理に関する問題についても対応しています。

11.7.1 標準プリンタ言語に対応していないプリンタ

これらのプリンタは汎用プリンタ言語には全く対応しておらず、特別な非公開コントロール言語だけに対応しています。そのためこれらのプリンタは、製造元が公開するドライバに対応したオペレーティングシステムでのみ動作します。GDI は Microsoft がグラフィックデバイス向けに開発したプログラミングインターフェイスで、プリンタデバイス向けの Windows ドライバは、これを利用して作成します。製造元が Windows に対応したドライバしか公開しないことから、それらのプリンタを *GDI* プリンタと呼びます。実際の問題はプログラミングインターフェイスではなく、そのプリンタの型番に対応した非公開のプリンタ言語でしかアクセスできない、という点です。

GDI プリンタによっては GDI モードと標準プリンタ言語モードを切り替えることができるものがあります。まずはお使いのプリンタのマニュアルをお読みのうえ、そのような切り替えに対応できるかどうかをご確認ください。また、型式によっては特別な Windows ソフトウェアで切り替えることができるものもあります (ただし Windows 側のプリンタドライバを利用して印刷する際、常に GDI モードに切り替えてしまうような場合もあります)。それ以外にも、拡張モジュールで標準プリンタ言語が利用できるプリンタもあります。

また、製造元によってはそれらのプリンタに対応したプロプライエタリ (非公開) なドライバを公開している場合もあります。プロプライエタリなドライバの欠点は、インストール済みのシステムで動作するのかどうかや、様々なハードウェアプラットフォーム上で動作するのかが保証されていない点です。逆に、標準プリンタ言語に対応したプリンタは、特定の印刷システムのバージョンや特別なハードウェアプラットフォームに依存していたりしません。

プロプライエタリな Linux ドライバを動かすために時間を費やすくらいであれば、標準言語に対応したプリンタ (おそらくは PostScript) を購入してしまったほうが、コスト面でより有利になるかもしれません。これはドライバの問題をいっぺんに解決し、特別なドライバソフトウェアのインストールや設定の要件から解放されるだけでなく、印刷システムが更新されたことによるドライバ側の更新作業も不要になります。

11.7.2 PostScript プリンタで適切な PPD ファイルが見つからない場合

manufacturer-PPDs パッケージ内に PostScript プリンタ向けの適切な PPD ファイルが見つからない場合、プリンタの製造元が提供するドライバ CD を利用することができるほか、プリンタの製造元の Web ページから対応する PPD ファイルをダウンロードして使用することができます。

PPD ファイルが zip アーカイブ (.zip) や自己解凍書庫の zip ファイル (.exe) で提供されている場合、まずは unzip で中身を展開してください。次に PPD ファイルのライセンス規約を確認したあと、cupstestppd ユーティリティを利用し、PPD ファイルが「Adobe PostScript Printer Description File Format Specification, version 4.3.」に準拠しているかどうかを確認してください。このコマンドの実行結果が「FAIL,」であった場合は、PPD ファイルに重大な問題が存在していることを示していて、これを使用すると大きな問題が生じ

ることを示しています。なお、cupstestppd で報告される問題点は、インストールを行なう前に解決されていなければなりません。必要であれば正しい PPD ファイルを提供するよう、プリンタの製造元に問い合わせてください。

11.7.3 パラレルポート

最も安全なアプローチは、プリンタを 1 基目のパラレルポートに接続し、BIOS 側でパラレルポートの設定を下記のような設定にするのが適切です:

- I/O アドレス: 378 (hexadecimal)
- 割り込み: (任意)
- モード: 通常, SPP, 出力のみのいずれか。
- DMA: 無効

上記の設定を行なったにもかかわらずパラレルポート上のプリンタがうまく動作しない場合は、BIOS 側に設定した I/O アドレスを 0x378 のような形で /etc/modprobe.conf 内に設定してください。なお、2 基のパラレルポートが存在する場合、アドレスが 378 と 278 (それぞれ 16 進) であれば 0x378, 0x278 のように指定してください。

また、割り込み 7 があいている場合は、例 11.1 「/etc/modprobe.conf: 1 基目のパラレルポート用割り込みモード設定」(179 ページ) のように設定することで、割り込みを有効にすることができます。割り込みモードを有効にする前に /proc/interrupts ファイルを確認し、その割り込みが使用されていないことを確認してください。左記のファイルでは、使用中の割り込みだけが表示されます。また、左記のファイルは利用中のハードウェアコンポーネントによって表示が異なります。パラレルポートの場合、他のデバイスから使用されているはいけません。うまく動作しない場合は、irq=none のように指定し、ポーリングモードでご利用ください。

例 11.1 /etc/modprobe.conf: 1 基目のパラレルポート用割り込みモード設定

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

11.7.4 ネットワークプリンタの接続

ネットワーク側の問題かどうかの確認

まずはお使いのコンピュータに直接プリンタを接続します。テストのため、ローカルプリンタとしてプリンタを設定してみます。これで問題なく動作するようであれば、問題はプリンタ側にあることになります。

TCP/IP ネットワークの動作確認

TCP/IP ネットワークと名前解決が正しく動作するかどうかを確認します。

遠隔の lpd の確認

下記のコマンドでは、ホスト上にある lpd (ポート 515) に対して、TCP の接続テストを行ないます:

```
netcat -z ホスト 515 && echo ok || echo failed
```

lpd との接続がうまくいかない場合は、lpd が動作していないか、基本的なネットワーク部分に問題があることになります。

さらに root ユーザから下記のコマンドを実行し、ホスト上のキューに対して (多くの場合、とても長い) ステータスレポートを要求します。これにより、lpd が動作しているかどうかと、ホストが問い合わせを受け付けるかどうかを確認することができます:

```
echo -e "\004キュー" \  
| netcat -w 2 -p 722 ホスト 515
```

lpd が何も応答を返さなかった場合は、lpd が動作していないか、基本的なネットワーク部分に問題があることになります。lpd が応答を返した場合は、出力された情報の中にホスト上のキューが、なぜ利用できないのかを表示します。たとえば例11.2「lpdからのエラーメッセージ」(180ページ)のような応答が返された場合は、リモート側の lpd に問題があることになります。

例 11.2 lpd からのエラーメッセージ

```
lpd: your host does not have line printer access  
lpd: queue does not exist  
printer: spooling disabled  
printer: printing disabled
```

遠隔の cupsd の確認

既定の設定では CUPS のネットワークサーバはそのキュー情報を 30 秒おきに UDP ポート 631 でブロードキャストします。そのため下記のコマンドを利用すると、ネットワーク内に CUPS ネットワーク サーバが存在するかどうかを確認することができます。ただし、ローカル側の CUPS デモンについては、コマンドを実行する前に停止しておいてください:

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

CUPS のブロードキャストパケットを送信するサーバが存在する場合、例 11.3 「CUPS ネットワークサーバからのブロードキャスト」(181 ページ) ように出力されます。

例 11.3 CUPS ネットワークサーバからのブロードキャスト

```
ipp://192.168.2.202:631/printers/queue
```

さらに、下記のコマンドでは ホスト 上にある cupsd (ポート 631) に対して、TCP の 接続テストを行ないます:

```
netcat -z ホスト 631 && echo ok || echo failed
```

cupsd との接続がうまくいかない場合は、cupsd が動作していないか、基本的なネットワーク部分に問題があることになります。さらに `lpstat -h ホスト -l -t` を実行すると、ホスト 上の全てのキューに対する (多くの場合、とても長い) ステータスレポートを要求します。これにより、cupsd が動作しているかどうかと、ホストが問い合わせを受け付けるかどうかを確認することができます。

また、次のコマンドはホスト上のキューに対して、単一の改行コードだけから成る印刷キューを受け付けるかどうかをテストすることができます。何も印刷されず、単に空白のページだけが出力されるはずです。

```
echo -en "\r" \  
| lp -d キュー -h ホスト
```

ネットワークプリンタやプリントサーバのトラブルシューティング

プリントサーバ内で動作しているスプール機能は、複数の印刷ジョブを取り扱うような場合に問題が発生する場合があります。このような問題はプリントサーバ内のスプール機能によって引き起こされている問題であるため、解決するための手段は存在しません。ただし、プリントサーバのスプール機能に接続されているプリンタに対し、直接 TCP 接続を行なう

ことで回避することができます。詳しくは 11.4 項「ネットワークプリンタ」(171 ページ)をお読みください。

この方法では、プリントサーバは様々なデータ転送手段 (TCP/IP ネットワークとローカルプリンタ接続) 間でのコンバータとして動作するようになります。この方法を利用するには、プリントサーバ内で使用する TCP ポートを知っておく必要があります。プリンタがプリントサーバに接続されていて電源が入っている場合、プリントサーバの電源を入れてからしばらくしてから、TCP ポートを nmap パッケージに含まれる nmap ユーティリティで判別することができます。nmap IP アドレスのように入力すると、プリントサーバに対する下記のような出力が表示されます:

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

上記の出力は、TCP ポート 9100 を利用することでプリントサーバ内のプリンタにアクセスできることを示しています。なお、既定では nmap コマンドは /usr/share/nmap/nmap-services ファイルに書かれた既知のポートだけを確認します。全てのポートに対して確認を行なうには、nmap -p 開始ポート-終了ポート IP アドレス のように入力してください。上記のコマンドを実行するには、しばらくの時間がかかります。また、詳しい情報は nmap のマニュアルページをお読みください。

また、下記のようなコマンドを入力すると、上記で発見したポートに対して文字列やファイルを直接送信し、プリンタに対してアクセスできるかどうかを確認することができます:

```
echo -en "\rHello\r\f" | netcat -w 1 IP-address port
cat file | netcat -w 1 (IP アドレス) (ポート)
```

11.7.5 エラーメッセージが表示されないにも関わらず出力がおかしい場合

印刷システムでは、印刷ジョブは CUPS のバックエンドが送信先 (プリンタ) に送信し終わった時点で完了します。受け取り側での処理作業で何らかの間

題が発生した場合 (たとえばプリンタ固有のデータを印刷できない場合) は、印刷システム側からそれを検知する手段がありません。お使いのプリンタがプリンタ固有のデータを印刷できない場合は、そのプリンタに対するより適切な PPD ファイルを選択してください。

11.7.6 キューの無効化

何回かの試行を行なっても送信先へのデータ送信が完了しない場合、USB やソケットなどの CUPS バックエンドは印刷システム (cupsd) に対してエラーを報告します。バックエンド側ではデータ転送を何回行なったら失敗と見なすかを判断しています。さらなる再試行が無駄であると判断した場合、cupsd はそのキューに対する印刷を停止します。問題の原因を解決したら、システム管理者は cupsenable コマンドを利用した印刷を再開しなければなりません。

11.7.7 CUPS 参照: 印刷ジョブの削除

CUPS ネットワークサーバが自身のキューをクライアントに対してブロードキャストで通知し、クライアント側での cupsd が動作している場合は、クライアント側の cupsd はアプリケーションからの印刷ジョブを受け入れ、サーバ側の cupsd に転送を行ないます。サーバ側の cupsd で印刷ジョブが受け付けられると、新しいジョブ番号が割り当てられます。そのためクライアント側でのジョブ番号は、サーバ側でのジョブ番号とは異なる番号になります。また、印刷ジョブは通常すぐに転送されるようになっていて、クライアント側の cupsd は、印刷ジョブをサーバ側の cupsd に転送が完了すると印刷完了であると見なす仕組みになっているため、クライアント側でジョブ番号を削除することができません。

サーバ側で印刷ジョブを削除する必要がある場合、まずはサーバ側でのジョブ番号を調べるため、`lpstat -h cups.example.com -o` のようなコマンドをご利用ください。このコマンドは未完了の印刷ジョブの一覧を表示します。ジョブ番号が判明したら、下記のコマンドで削除を行なってください:

```
cancel -h cups.example.com キュー-ジョブ番号
```

11.7.8 印刷ジョブの問題とデータ転送エラー

印刷処理中にプリンタの電源を落としたり、コンピュータをシャットダウンしたりした場合は、印刷ジョブがキュー内に残ったままになっています。印刷処理はコンピュータ(またはプリンタ)の電源が復帰した後に再開されます。不要な印刷ジョブや間違った印刷ジョブは、cancel コマンドでキューから削除しなければなりません。

また、印刷ジョブが正しい形式ではなかったり、ホストとプリンタの間での通信にエラーが発生したりした場合は、プリンタはデータを正しく処理できず、何枚もの正しくない出力をしてしまう場合があります。このような現象が発生した場合は、下記の手順で修復してください:

- 1 印刷を停止するため、インクジェットプリンタの場合は全ての紙を取り除き、レーザープリンタの場合は用紙トレイを開けます。また、高品質なプリンタであれば、現在の印刷処理をキャンセルするためのボタンが用意されている場合もあります。
- 2 ジョブはプリンタへの送信が全て完了した場合にのみ削除される仕組みであるため、印刷ジョブがキュー内に残っている場合があります。lpstat -o や lpstat -h cups.example.com -o などのコマンドを利用して、現在のキューが印刷中であるかどうかを確認してください。印刷中の場合は、cancel キュー-ジョブ番号 または cancel -h cups.example.com キュー-ジョブ番号 を利用して、キューを削除してください。
- 3 また、キューから印刷ジョブを削除しても、プリンタに対するデータが少し残っている場合があります。該当するキューに対する CUPS バックエンドプロセスが起動中かどうかを確認し、起動中であれば終了させてください。たとえばパラレルポートに接続されているプリンタの場合、fuser -k /dev/lp0 コマンドを実行すると、未だプリンタ(より厳密に言えばパラレルポート)に対してデータを送信しているプロセスを終了させることができます。
- 4 電源を落としてしばらく待ち、完全にリセットします。再度用紙を配置し、プリンタの電源を入れてください。

11.7.9 CUPS 印刷システムのデバッグ

CUPS印刷システムの問題を切り分けるためには、下記の汎用手順をご利用ください:

- 1 `/etc/cups/cupsd.conf` ファイル内に `LogLevel debug` を設定します。
- 2 `cupsd` を停止します。
- 3 巨大なログファイルを検索したりすることのないよう、`/var/log/cups/error_log*` ファイルを削除します。
- 4 `cupsd` を起動します。
- 5 問題を再現させます。
- 6 `/var/log/cups/error_log*` ファイルを確認し、問題の原因を調査します。

11.7.10 さらなる情報

多くの固有の問題は、SUSEのサポートデータベース(http://en.opensuse.org/Portal:Support_database)(日本語)または(http://en.opensuse.org/Portal:Support_database)(英語)内に書かれています。プリンタ関連の問題の場合は、テキスト検索で `SDB:CUPS` と入力してください。

グラフィカルユーザインター フェイスを利用したフォントの インストールと設定

12

openSUSE®での追加フォントのインストールは非常に簡単です。単純に X11 フォントパスとして指定されているディレクトリ (詳しくは 12.1 項「X11 コアフォント」(188 ページ)をお読みください)のいずれかにフォントをコピーするだけです。また、これらのフォントを使用できるようにするには、そのディレクトリが/etc/fonts/fonts.conf ファイル (詳しくは 12.2 項「Xft」(190 ページ)をお読みください)に書かれているか、もしくは/etc/fonts/suse-font-dirs.conf ファイルで指定されたディレクトリ以下に存在する必要があります。

下記は/etc/fonts/fonts.conf ファイルの抜粋です。このファイルは多くの設定で用いられる標準的な設定ファイルです。また、この設定ファイルでは/etc/fonts/conf.d ディレクトリの内容を取り込むようにも設定しています。fontconfig では、このディレクトリ以下の全てのファイルやシンボリックリンクのうち、2桁の数字で始まるファイルを読み込みます。この機能についての詳細な説明は、/etc/fonts/conf.d/README ファイルをお読みください。

```
<!-- Font directory list -->
<dir>/usr/share/fonts</dir>
<dir>/usr/X11R6/lib/X11/fonts</dir>
<dir>/opt/kde3/share/fonts</dir>
<dir>/usr/local/share/fonts</dir>
<dir>~/fonts</dir>
```

また、/etc/fonts/suse-font-dirs.conf ファイルは、OpenOffice.org や Java, Adobe Acrobat Reader のような (主にサードパーティ製の) アプリケー

ションと共に出荷されるフォントを引き出すために自動生成されるものです。いくつかの典型的な項目を下記に示します:

```
<dir>/usr/lib/Adobe/Reader9/Resource/Font</dir>
<dir>/usr/lib/Adobe/Reader9/Resource/Font/PPM</dir>
```

追加のフォントをシステム全体に対してインストールするには、フォントファイルを `/usr/share/fonts/truetype` などの適切なディレクトリにコピーしてください(もちろん `root` で実行します)。また、KDE 個人設定内の KDE フォントインストーラからも実行することができます。結果はいずれを実施しても同じになります。

また、実際のフォントをコピーする代わりに、シンボリックリンクを作成することもできます。たとえば Windows 上にライセンスによる制限が課されたフォントが存在していて、それらを使用したい場合などに便利です。シンボリックリンクを作成したら、`SuSEconfig --module fonts` コマンドを実行します。

`SuSEconfig --module fonts` を実行すると、`/usr/sbin/fonts-config` スクリプトを実行します。このスクリプトはフォント設定を扱うためのものです。このスクリプトに関する詳細は、マニュアルページ (`man fonts-config`) をお読みください。

ビットマップフォント, TrueType フォント, OpenType フォント, Type1 (PostScript)、いずれのフォントでも手順は同じです。これら全てのフォントは、`fonts-config` が認識できるディレクトリにインストールすることができます。

また、X.Org では 2 種類の異なるフォントシステムを用意しています: 1 つは X11 コアフォントシステム、もう 1 つは新たに設計された *Xft* と *fontconfig* のシステムです。下記の章では、これら 2 つのシステムについて、概要を説明しています。

12.1 X11 コアフォント

現在、X11 コアフォントシステムはビットマップフォントだけでなく、スケーラブル (スムーズな拡大/縮小のできる) フォントにも対応しています。それぞれ Type1, Truetype, OpenType と呼ばれる各フォントです。スケーラブルフォ

ントはアンチエイリアス (拡大／縮小時に周辺のドット色を調整して滑らかに見せる技術) やサブピクセルレンダリング (縮小時に色を分割し、擬似的に解像度を上げる技術) には対応しておらず、多数の言語向けに字体を定義した巨大なフォントを利用しようとすると、読み込みに時間がかかってしまいます。Unicode フォントにも対応していますが、描画が遅くメモリを多く使用してしまいます。

また、X11 コアフォントシステムには、いくつかの固有の弱点が存在しています。既に時代遅れの設計で、拡張はほとんど行なうことができない状態です。しかしながら旧バージョンとの互換性の観点から、今もなお搭載され続けている状態です。そのため、可能な限り Xft と fontconfig の仕組みを利用したほうがよりよいものになります。

操作を行なうにあたって、X サーバは利用可能なフォントの一覧や、それらのフォントをどうやったら使用できるのかについて、一覧を持っておく必要があります。これらは FontPath として、有効な全てのシステムフォントディレクトリを保持しています。これらそれぞれのディレクトリに対して fonts.dir ファイルを配置し、そのディレクトリに何のフォントがあるのか一覧を保存しています。また、FontPath は X サーバの起動時に生成されます。/etc/X11/xorg.conf に設定された FontPath 変数のそれぞれに対して、有効な fonts.dir ファイルがないかどうか、検索を行ないます。これらの項目は Files セクション内に書き込みます。現在の FontPath 変数を表示させるには、xset q コマンドを実行します。また、この設定は xset を利用して随時に変更することができます。さらなるパスを追加するには、xset +fp <パス> のように実行してください。逆に不要なパスを削除するには、xset -fp <パス> のように実行します。

X サーバが既に動作している場合、マウント済みディレクトリに新しくインストールしたフォントは、xset fp rehash コマンドで利用できるようになります。このコマンドは SuSEconfig --module fonts から実行されます。また、xset コマンドは動作中の X サーバにアクセスするため、SuSEconfig --module fonts をシェルから起動していて、かつ X サーバが動作中である場合にのみ動作します。これを実現するための最も簡単な方法は、su コマンドを実行して root のパスワードを入力し、root のパーミッションを獲得する方法があります。su では、X サーバを実行しているユーザから root のシェルに対してアクセス権を渡す仕組みになっているためです。フォントが正しくインストールされ、X11 のコアフォントシステム

から利用できることを確認するには、`xlsfonts` コマンドを実行して、利用可能な全てのフォントについて一覧を表示してください。

また、既定では **openSUSE** は UTF-8 のロケールを使用します。そのため、Unicode フォント (`xlsfonts` コマンドの出力で、`iso10646-1` で終わるもの) を使用するのが適切です。利用可能な全ての Unicode フォントは、`xlsfonts | grep iso10646-1` のコマンドラインを実行することで取得することができます。なお、**openSUSE** で利用可能なほぼ全ての Unicode フォントには、少なくともヨーロッパ言語 (以前は `iso-8859-*` でエンコードされていたもの) の字体が含まれています。

12.2 Xft

当初の段階から、**Xft** のプログラマはスケーラブルなフォントに対して、アンチエイリアスの機能を十分サポートするように作ってきました。**Xft** を使用すると、**X11** コアフォントシステムが **X** サーバ内から描画するのとは異なり、フォントを使用するアプリケーション自身が描画を行いません。この方法では、関連するアプリケーションは実際のフォントファイルにアクセスできる権利を持っていて、かつどのように字体を描画するのかについて完全なコントロールを行なうことになります。これは複数の言語でテキストを正しく表示する目的において、必要な機能の 1 つになっています。フォントファイルへの直接アクセスは、画面への出力と印刷出力の両方で外観を合わせるため、フォントを内蔵させたりするのに 便利な機能です。

また **openSUSE** では、2 種類のデスクトップ環境 (**KDE**, **GNOME**) の両方や **Mozilla** などのアプリケーションで **Xft** を既定値として使用しています。**Xft** は既に古い **X11** のコアフォントシステムより多くのアプリケーションで使用されています。

さらに、**Xft** は **fontconfig** ライブラリを、フォントの発見や描画方法への作用として使用しています。**fontconfig** の設定はグローバル設定ファイルである `/etc/fonts/fonts.conf` から操作することができます。また、固有の設定ファイルについては `/etc/fonts/local.conf` に、ユーザ固有の設定については `~/.fonts.conf` にそれぞれ追加してください。これらの **fontconfig** 設定ファイルには、下記のヘッダを記述しておく必要があります:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

また、このフッタで終わる必要もあります:

```
</fontconfig>
```

フォントの検索ディレクトリを追加するには、下記のような行を追加します:

```
<dir>/usr/local/share/fonts/</dir>
```

しかしながら、これは通常不要なものです。既定ではユーザ指定のディレクトリ `~/.fonts` が `/etc/fonts/fonts.conf` 内に指定されているためです。そのため、フォントをインストールする際にやるべきことは、単に追加のフォントを `~/.fonts` にコピーするだけです。

また、フォントの外観に影響を与えるルールを追加することもできます。たとえば

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

と入力すると全てのフォントでアンチエイリアスを無効化することができますし、

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

と入力すると、指定したフォントだけアンチエイリアスを無効化することができます。

既定では多くのアプリケーションが `sans-serif` (または同等の指定である `sans`, `serif`, `monospace` など) というフォント名を使用しています。これらは実際のフォント名ではなく別名で、お使いの言語設定に合わせて適切なフォントを選択するための名称です。

なお、ユーザ自身でこれらの別名と実際のフォントの割り当てを変更することができます。 `~/.fonts.conf` ファイルに下記の行を追加してください:

```

<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
</alias>
<alias>
  <family>monospace</family>
  <prefer>
    <family>FreeMono</family>
  </prefer>
</alias>

```

ほとんど全てのアプリケーションがこの別名を使用しているため、この設定はほぼ全てのシステムに適用されます。そのため、個別のアプリケーションでわざわざフォントの設定を変更したりすることなく、希望のフォントを使用するよう一括設定を行なうことができます。

また、`fc-list` コマンドを使用すると、どのフォントがインストールされ使用できる状態にあるのかを表示することができます。たとえば`fc-list`を実行すると、全てのフォントについて一覧を出力します。利用可能なスケラブルフォントの一覧を出力するために`:scalable=true`のオプションを、ヘブライ語の字体を含む全てのフォントを表示するために`:lang=he`のオプションをそれぞれ指定し、さらに出力としてフォント名 (family), スタイル (style), ウェイト (weight) とフォントファイル名を表示するには、下記のコマンドを実行します:

```
fc-list ":lang=he:scalable=true" family style weight
```

出力結果は、たとえば以下ようになります:

```

Lucida Sans:style=Demibold:weight=200
DejaVu Sans:style=Bold Oblique:weight=200
Lucida Sans Typewriter:style=Bold:weight=200
DejaVu Sans:style=Oblique:weight=80
Lucida Sans Typewriter:style=Regular:weight=80
DejaVu Sans:style=Book:weight=80
DejaVu Sans:style=Bold:weight=200
Lucida Sans:style=Regular:weight=80

```

`fc-list` で利用できる重要なパラメータは下記のとおりです:

表 12.1 *fc-list* のパラメータ

パラメータ	意味と取り得る値
family	たとえば FreeSans のようなフォントファミリ名。
foundry	たとえば urw のようなフォントの作成者。
style	たとえば Medium, Regular, Bold, Italic, Heavy のようなフォントスタイル。
lang	そのフォントがサポートする言語。たとえば de であればドイツ語を、 ja であれば日本語を、 zh-TW であれば繁体字中国語を、 zh-CN であれば簡体字中国語をそれぞれ指定します。
weight	フォントのウェイト値。たとえば 80 であれば通常体で、 200 であればボールド体になります。
slant	傾き。0 は通常体、 100 はイタリック (斜体) を表わします。
file	フォントを含むファイルのファイル名。
outline	true を指定するとアウトラインフォントを、 false を指定するとそれ以外のフォントを指定します。
scalable	true を指定するとスケーラブルフォントを、 false を指定するとそれ以外のフォントを指定します。
bitmap	true を指定するとビットマップフォントを、 false を指定するとそれ以外のフォントを指定します。
pixelsize	ピクセル単位でのフォントサイズ。fc-list の場合はビットマップフォントである場合にのみ意味を持ちます。

システム監視ユーティリティ

お使いのシステムの状態を確認するため、数多くのプログラムやツール、ユーティリティなどが用意されています。この章では、これらユーティリティのうちのいくつかを紹介するほか、日々の定型作業をこなすのに便利なユーティリティや、それらの中の最も重要なパラメータについても紹介しています。

本章でコマンドを紹介する際には、コマンドと共にその出力例も表示しています。たとえば下記の例では、最初の行がコマンドそれ自身(行頭に>や#が書かれます)を示し、それ以降がコマンドの出力例になっています。途中の出力などを省略する場合は、大括弧([...])を利用して表記します。また、表記上の理由で長い行を途中で改行する場合は、バックスラッシュ(日本語環境では円マーク)(\)を行末に表記します。

```
# command -x -y
出力 行 1
出力 行 2
出力 行 3 は厄介なまでに長いので、改行を入れて \
    行を区切っています。
出力 行 4
[...]
```

説明は、できる限り多くのユーティリティを紹介したいという意図から、できる限り簡潔に記しています。全てのコマンドに対する詳しい情報は、各マニュアルページをお読みください。多くのコマンドでは--help オプションを指定することにより、利用可能なパラメータの一覧と、簡易な説明が表示されます。

13.1 他用途ツール

多くの Linux システム監視ツールは、システムの一部分を監視するのに特化した仕組みになっていますが、「十徳ナイフ」のようなシステム状態を一括表示するようなものも少数存在しています。まずはこれらのツールを利用してシステム状態の概要をつかむとともに、どの部分についてより細かい調査を必要とするのかを確認してください。

13.1.1 vmstat

vmstat はプロセスやメモリ、I/O や割り込み、CPU に関する情報を収集します。パラメータに採取間隔を指定せずに実行すると、直近の再起動以後の平均値を表示します。パラメータに採取間隔を指定して実行すると、実際の採取間隔ごとに値を表示します:

例 13.1 *vmstat* 負荷の軽いマシンでの出力

```
tux@mercury:~> vmstat -a 2
procs -----memory----- --swap--  -----io----- -system--  -----cpu-----
 r  b  swpd   free  inact active    si   so    bi    bo    in   cs us sy  id wa st
 0  0      0 750992 570648 548848    0    0     0     1     8    9  0  0 100  0  0
 0  0      0 750984 570648 548912    0    0     0     0    63   48  1  0  99  0  0
 0  0      0 751000 570648 548912    0    0     0     0    55   47  0  0 100  0  0
 0  0      0 751000 570648 548912    0    0     0     0    56   50  0  0 100  0  0
 0  0      0 751016 570648 548944    0    0     0     0    57   50  0  0 100  0  0
```

例 13.2 *vmstat* 負荷の重いマシンでの出力 (特に CPU を使用しているマシン)

```
tux@mercury:~> vmstat 2
procs -----memory----- --swap--  -----io----- -system--  -----cpu-----
 r  b  swpd   free  buff  cache    si   so    bi    bo    in   cs us sy  id wa st
32  1  26236 459640 110240 6312648    0    0  9944     2 4552 6597 95  5  0  0  0
23  1  26236 396728 110336 6136224    0    0  9588     0 4468 6273 94  6  0  0  0
35  0  26236 554920 110508 6166508    0    0  7684 27992 4474 4700 95  5  0  0  0
28  0  26236 518184 110516 6039996    0    0 10830     4 4446 4670 94  6  0  0  0
21  5  26236 716468 110684 6074872    0    0  8734 20534 4512 4061 96  4  0  0  0
```

ティップ

vmstat の出力のうち最初の行には、常に直近の再起動からの平均値が表示されます。

各列には、それぞれ下記の値が表示されます:

r

実行キューに入っているプロセス数が表示されます。これらのプロセスは、処理を行なうために CPU の空きを待っている状態です。ここの値が利用可能な CPU 数 よりも定常的に大きい場合は、CPU の処理能力が不足していることを示します。

b

CPU 以外の資源を待機しているプロセス数が表示されます。ここの値が高い場合は、入出力 (ネットワークまたはディスク) に問題があることを示しています。

swpd

現在使用中のスワップ領域の量が表示されます。

free

未使用のメモリ量が表示されます。

inact

未使用メモリのうち、再利用可能な量の最新値を示しています。この列は、vmstat のパラメータに -a (推奨パラメータ) を指定した場合にのみ表示されます。

active

通常は再利用できない、直近で使用されているメモリの量を表示しています。この列は、vmstat のパラメータに -a (推奨パラメータ) を指定した場合にのみ表示されます。

buff

RAM 内にあるファイルバッファキャッシュのサイズが表示されます。この値は、vmstat のパラメータに -a (推奨パラメータ) を指定した場合には表示されません。

cache

RAM 内にあるページキャッシュのサイズが表示されます。この値は、vmstat のパラメータに -a (推奨パラメータ) を指定した場合には表示されません。

si

1 秒あたりに **RAM** からスワップ領域に移動したデータの量が表示されます。この列の値が長い時間高い値を示している場合、お使いのマシンに **RAM** を追加したほうがよりよいことを示しています。

so

1 秒あたりにスワップから **RAM** に移動したデータの量が表示されます。この列の値が長い時間高い値を示している場合、お使いのマシンに **RAM** を追加したほうがよりよいことを示しています。

bi

1 秒あたりにブロックデバイス(ディスクなど)から読み出したブロック数が表示されます。スワップ処理が起こった場合にも、この値が増加します。

bo

1 秒あたりにブロックデバイス(ディスクなど)に書き込んだブロック数が表示されます。スワップ処理が起こった場合にも、この値が増加します。

in

1 秒あたりの割り込み数が表示されます。この値が高い場合、入出力(ネットワークまたはディスク)が多く発生していることを示します。

cs

1 秒あたりのコンテキストスイッチ数が表示されます。この値を簡単に言うと、カーネルが複数のプログラムを実行するため、一方のプログラムから他方のプログラムに実行コードを切り替えた回数を示しています。

us

ユーザプロセスによる **CPU** 使用率をパーセンテージで示しています。

sy

システムプロセスによる **CPU** 使用率をパーセンテージで示しています。

id

CPU がアイドル状態(未使用の状態)にあった割合をパーセンテージで示しています。この値が長い時間ずっとゼロであった場合、お使いの **CPU** が使い切られていることを示しています。これは特に悪い兆候というわけではありません。*r* や *b* の値をそれぞれ確認し、十分な **CPU** 性能があることを確認できれば、特にこの値が小さくても心配する必要はありません。

wa

入出力 (IO) を待機するのに費やした時間を表示しています。この値が長い時間ゼロよりもずっと大きくなっている場合は、入出力システム(ネットワークまたはハードディスク)がボトルネックになっていることを示しています。

st

仮想マシンで使用された CPU 時間のパーセンテージを表示しています。

さらなるオプションについては、`vmstat --help` をお読みください。

13.1.2 システムの動作情報: **sar** と **sadc**

`sar` コマンドはほぼ全ての重要なシステム動作についてレポートを生成することのできるツールで、CPU やメモリ、IRQ の使用率、入出力やネットワークなどの情報を表示することができます。また本コマンドはレポートをその場で作成することができるほか、システム動作の収集プログラムである `sadc` が過去に採取した既存の情報から作成することもできます。なお、`sar` や `sadc` は全て `/proc` ファイルシステムが提供する情報を利用しています。

注記: **sysstat** パッケージ

`sar` と `sadc` は、いずれも `sysstat` パッケージに含まれています。YaST からインストールを行なうか、もしくは `zypper in sysstat` でインストールを行なってください。

sadc を利用した日次の自動データ収集

お使いのシステムを長い期間にわたって監視したい場合は、`sadc` を利用してデータを自動収集させることをお勧めします。収集したデータは後から `sar` コマンドで読み込むことができます。`sadc` を起動するには、`/etc/init.d/boot.sysstat start` を実行してください。このコマンドは下記の既定の設定で `sadc` を呼び出すよう `/etc/cron.d/` にリンクを追加するものです。

- 利用可能な全てのデータを収集する。

- データは `/var/log/sa/saDD` に書き込む。ここで `DD` は現在の日 (日付) を示すものとする。既にファイルが存在する場合、ファイルはアーカイブされる。
- 概要レポートは `/var/log/sa/sarDD` に書き込む。ここで `DD` は現在の日 (日付) を示すものとする。既にファイルが存在する場合、ファイルはアーカイブされる。
- データは 10 分間隔で収集し、概要レポートは 6 時間ごとに作成する (`/etc/sysstat/sysstat.cron` をお読みください)。
- データは `/usr/lib64/sa/sa1` で収集する (32 ビット環境の場合は `/usr/lib/sa/sa1`)
- 概要は `/usr/lib64/sa/sa2` で作成する (32 ビット環境の場合は `/usr/lib/sa/sa2`)

設定をカスタマイズしたい場合は、`sa1` と `sa2` のスクリプトをコピーし、必要に応じて修正してください。また、`/etc/cron.d/sysstat` にあるシンボリックリンクを `/etc/sysstat/sysstat.cron` のカスタマイズ版と入れ替えてください。

sar を利用したレポート生成

レポートをその場で作成するには、`sar` コマンドに間隔 (秒) とカウントを指定して実行します。ファイルからレポートを作成するには、間隔指定の代わりに `-f` オプションを利用し、ファイル名を指定してください。ファイル名と間隔、カウントのいずれも指定しなければ、`/var/log/sa/saDD` ファイルからレポートを生成します。ここで `DD` には今日の日 (日付) が入ります。上記のディレクトリは `sadc` がデータを書き込む際の既定のディレクトリです。複数のファイルを読み込む場合は、`-f` オプションを必要な数だけ指定してください。

```
sar 2 10                                # その場でレポートを作成する。 2 秒間隔で 10 回分。
sar -f ~/reports/sar_2010_05_03        # sar_2010_05_03 ファイルへの問い合わせ。
sar                                     # /var/log/sa/ にある本日分のファイルへの問い合わせ。
cd /var/log/sa &&\
sar -f sa01 -f sa02                     # /var/log/sa/0[12] のファイルへの問い合わせ。
```

sar の便利な使用例と解釈は後述します。各列の意味について、詳しくは sar の man (1) (マニュアルページ) をお読みください。マニュアルページには、sar が提供する数多くのオプション指定やレポート生成方法が書かれています。

CPU 使用率レポート: sar

何もオプションを指定せずに呼び出した場合、sar は CPU について基本的なレポートを作成します。マルチプロセッサマシンの場合、全ての CPU に対する結果が合計されて表示されます。個別の CPU に対して別々の統計情報を得たい場合は、-P ALL オプションをお使いください。

```
mercury:~ # sar 10 5
Linux 2.6.31.12-0.2-default (mercury) 03/05/10    _x86_64_    (2 CPU)

14:15:43   CPU      %user    %nice    %system    %iowait    %steal     %idle
14:15:53   all      38.55     0.00      6.10      0.10      0.00    55.25
14:16:03   all      12.59     0.00      4.90      0.33      0.00    82.18
14:16:13   all      56.59     0.00      8.16      0.44      0.00    34.81
14:16:23   all      58.45     0.00      3.00      0.00      0.00    38.55
14:16:33   all      86.46     0.00      4.70      0.00      0.00     8.85
Average:   all      49.94     0.00      5.38      0.18      0.00    44.50
```

%iowait (入出力を待機していた CPU 時間の割合) が長い時間ゼロよりも顕著に大きい場合は、入出力のシステム (ネットワークやハードディスク) にボトルネックが存在することを示しています。また、%idle が長い時間ゼロであった場合は、お使いの CPU の性能を使い切っていることを示しています。

capacity.

メモリ使用率レポート: sar -r

お使いのシステムメモリ (RAM) について概要を生成するには、-r オプションを使用します:

```
mercury:~ # sar -r 10 5
Linux 2.6.31.12-0.2-default (mercury) 03/05/10    _x86_64_    (2 CPU)

16:12:12 kbmemfree kbmemused %memused kbbuffers kbcached kbcommit %commit
16:12:22   548188   1507488    73.33    20524    64204   2338284    65.10
16:12:32   259320   1796356    87.39    20808    72660   2229080    62.06
16:12:42   381096   1674580    81.46    21084    75460   2328192    64.82
16:12:52   642668   1413008    68.74    21392    81212   1938820    53.98
16:13:02   311984   1743692    84.82    21712    84040   2212024    61.58
Average:   428651   1627025    79.15    21104    75515   2209280    61.51
```

最後の2列 (*kbccommit* と *%commit*) には、現在の負荷量を処理するにあたって必要な合計のメモリ容量 (RAM とスワップ の合計) の最大値をそれぞれキロバイト単位とパーセント単位で表示しています。

ページング統計レポート: **sar -B**

カーネルのページングに関する統計情報を表示するには、**-B** オプションを使用します。

```
mercury:~ # sar -B 10 5
Linux 2.6.31.12-0.2-default (mercury) 03/05/10 _x86_64_ (2 CPU)

16:11:43 pgpgin/s pgpgout/s fault/s majflt/s pgfree/s pgscank/s pgscand/s pgsteal/s %vmeff
16:11:53 225.20 104.00 91993.90 0.00 87572.60 0.00 0.00 0.00 0.00
16:12:03 718.32 601.00 82612.01 2.20 99785.69 560.56 839.24 1132.23 80.89
16:12:13 1222.00 1672.40 103126.00 1.70 106529.00 1136.00 982.40 1172.20 55.33
16:12:23 112.18 77.84 113406.59 0.10 97581.24 35.13 127.74 159.38 97.86
16:12:33 817.22 81.28 121312.91 9.41 111442.44 0.00 0.00 0.00 0.00
Average: 618.72 507.20 102494.86 2.68 100578.98 346.24 389.76 492.60 66.93
```

majflt/s (1 秒あたりのメジャーフォルト) の列には、ディスク (スワップ) からメモリ無いに読み込まれたページ数がかかれています。メジャーフォルトの値が大きくなるとシステムの速度を低下させてしまうため、メインメモリの不足を意味する指標になります。また、*%vmeff* の列にはメインメモリ内のキャッシュやスワップキャッシュ (*pgsteal/s*) に対する スキャン済みページ数 (*pgscand/s*) の割合が表示されています。これはページの 再利用について効率値を示すもので、100 に近い値を示す (それぞれスワップアウト された未使用のページが再利用されている) か、もしくは 0 に近い値を示す (ページがスキャンされていない) ほどシステムが効率よく働いていることを示します。なお、この値は 30 より小さい値になるべきではありません。

ブロックデバイスの統計情報: **sar -d**

ブロックデバイス (ハードディスク、光学ドライブ、USB ストレージデバイスなど) についての統計情報を表示するには、**-d** オプションを使用します。**DEV** の列を見やすくするため、**-p (pretty-print)** の追加オプションを使用していることをご確認ください。

```
mercury:~ # sar -d -p 10 5
Linux 2.6.31.12-0.2-default (neo) 03/05/10 _x86_64_ (2 CPU)

16:28:31 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
16:28:41 sdc 11.51 98.50 653.45 65.32 0.10 8.83 4.87 5.61
16:28:41 sdc0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

16:28:41	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
16:28:51	sdsc	15.38	329.27	465.93	51.69	0.10	6.39	4.70	7.23
16:28:51	sdc0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16:28:51	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
16:29:01	sdsc	32.47	876.72	647.35	46.94	0.33	10.20	3.67	11.91
16:29:01	sdc0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16:29:01	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
16:29:11	sdsc	48.75	2852.45	366.77	66.04	0.82	16.93	4.91	23.94
16:29:11	sdc0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16:29:11	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
16:29:21	sdsc	13.20	362.40	412.00	58.67	0.16	12.03	6.09	8.04
16:29:21	sdc0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average:	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
Average:	sdsc	24.26	903.52	509.12	58.23	0.30	12.49	4.68	11.34
Average:	sdc0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

お使いのマシンに複数のディスクが接続されている場合、全てのディスクに対して 平均的に入出力要求が分散していると最高の性能を引き出すことができます。 *Average* 行の *tps*, *rd_sec/s*, *wr_sec/s* の値を 全てのディスクに対して比較してください。 *svctm* と *%util* の列にある値が定常的に高い場合は、ディスク内の 空き容量が不足していることが考えられます。

ネットワークの統計情報: **sar -n** キーワード

-n オプションを指定すると、ネットワークに関連する複数のレポートを生成することができます。 *-n* に続いて下記の キーワードを指定してください:

- *DEV*: 全てのネットワークデバイスに対する統計 レポートを生成します。
- *EDEV*: 全てのネットワークデバイスに対するエラー 統計レポートを生成します。
- *NFS*: NFS の統計レポートを生成します。 *client*
- *NFSD*: NFS サーバの統計レポートを生成します。
- *SOCK*: ソケットに関する統計レポートを生成します。
- *ALL*: 全てのネットワーク統計情報を生成します。

Visualizing sar Data

sar のレポートは人間にとってそれほど見やすいものとは言えません。kSar と呼ばれる sar データの視覚化 Java アプリケーションがありますが、これを利用することで見やすいグラフを作成することができます。また、このアプリケーションはPDF形式でのレポートを作成することもできるほか、その場での作成や過去データのファイルから作成することもできます。なお、kSar は BSD ライセンスで提供されています。詳しくは <http://ksar.atomique.net/> をお読みください。

13.2 システム情報

13.2.1 デバイスの負荷情報: iostat

iostat はシステムデバイスの負荷を監視します。お使いのシステムに接続された物理ディスクの負荷について、うまく バランスをとるための便利なレポートを提供します。

最初の iostat レポートでは、システム起動後から 収集し続けられた統計情報を表示します。それ以降のレポートでは、前回の レポートからの差分を表示します。

```
tux@mercury:~> iostat
Linux 2.6.32.7-0.2-default (geeko@buildhost) 02/24/10 _x86_64_

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,49    0,01    0,10    0,31    0,00   99,09

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
sda                  1,34         5,59         25,37    1459766    6629160
sda1                  0,00         0,01         0,00        1519         0
sda2                  0,87         5,11        17,83    1335365    4658152
sda3                  0,47         0,47         7,54    122578    1971008
```

-n オプションを指定すると、iostat はネットワークファイルシステム(NFS) に関する負荷情報を表示ようになります。また、-x オプションを指定すると、より広い範囲の統計情報を表示します。

また、どのデバイスをどれくらいの時間間隔で監視するのかを指定することができます。たとえば `iostat -p sda 3 5` のように入力すると、デバイス `sda` に関するレポートを 3 秒間隔で 5 回表示します。

注記: `sysstat` package

`iostat` は `sysstat` パッケージの一部です。YaST を利用するか、もしくは `zypper in sysstat` コマンドでパッケージをインストールしてください。

13.2.2 プロセッサの動作監視: `mpstat`

`mpstat` ユーティリティは、利用可能な各プロセッサの動作状況を調査します。お使いのシステムにプロセッサが 1 基しか搭載されていない場合は、全体の平均統計情報が表示されます。

`-P` オプションを指定すると、報告すべきプロセッサを番号で指定することができます (0 が 1 基目になります)。タイミングに関するオプションは、`iostat` のものと同じやり方で指定します。たとえば `mpstat -P 1 2 5` のように入力すると、2 秒間隔で 5 回、2 基目のプロセッサ (番号指定で 1) に関するレポートを出力します。

```
tux@mercury:~> mpstat -P 1 2 5
Linux 2.6.32.7-0.2-default (geeko@buildhost) 02/24/10 _x86_64_

08:57:10 CPU      %usr   %nice    %sys %iowait    %irq   %soft  %steal   \
           %guest   %idle
08:57:12   1      4.46    0.00    5.94    0.50    0.00    0.00    0.00   \
           0.00   89.11
08:57:14   1      1.98    0.00    2.97    0.99    0.00    0.99    0.00   \
           0.00   93.07
08:57:16   1      2.50    0.00    3.00    0.00    0.00    1.00    0.00   \
           0.00   93.50
08:57:18   1     14.36    0.00    1.98    0.00    0.00    0.50    0.00   \
           0.00   83.17
08:57:20   1      2.51    0.00    4.02    0.00    0.00    2.01    0.00   \
           0.00   91.46
Average:    1      5.17    0.00    3.58    0.30    0.00    0.90    0.00   \
           0.00   90.05
```

13.2.3 タスクの監視: pidstat

お使いのシステムにおいて特定のタスクの負荷を確認したい場合は、pidstat コマンドを使用します。選択したそれぞれのタスクについて処理状況を表示するか、タスクを何も指定しない場合はLinuxカーネルが管理する全タスクの処理状況を表示します。レポート数を指定することができるほか、それらの時間間隔を指定することもできます。

たとえばpidstat -C top 2 3 のように入力すると、コマンド名に「top」を含むタスクの負荷状況を表示します。2秒間隔で計3回のレポートが作成されます。

```
tux@mercury:~> pidstat -C top 2 3
Linux 2.6.27.19-5-default (geeko@buildhost) 03/23/2009 _x86_64_

09:25:42 AM      PID      %usr %system  %guest    %CPU   CPU   Command
09:25:44 AM    23576    37.62   61.39    0.00   99.01     1     top

09:25:44 AM      PID      %usr %system  %guest    %CPU   CPU   Command
09:25:46 AM    23576    37.00   62.00    0.00   99.00     1     top

09:25:46 AM      PID      %usr %system  %guest    %CPU   CPU   Command
09:25:48 AM    23576    38.00   61.00    0.00   99.00     1     top

Average:          PID      %usr %system  %guest    %CPU   CPU   Command
Average:          23576    37.54   61.46    0.00   99.00     -     top
```

13.2.4 カーネルのリングバッファ表示: dmesg

Linuxカーネルでは、特定のメッセージをリングバッファ内に保存しています。これらのメッセージを表示するには、dmesg コマンドをご利用ください:

```
tux@mercury:~> dmesg
[...]
end_request: I/O error, dev fd0, sector 0
subfs: unsuccessful attempt to mount media (256)
e100: eth0: e100_watchdog: link up, 100Mbps, half-duplex
NET: Registered protocol family 17
IA-32 Microcode Update Driver: v1.14 <tigran@veritas.com>
microcode: CPU0 updated from revision 0xe to 0x2e, date = 08112004
IA-32 Microcode Update Driver v1.14 unregistered
bootsplash: status on console 0 changed to on
NET: Registered protocol family 10
Disabled Privacy Extensions on device c0326ea0(lo)
IPv6 over IPv4 tunneling driver
```

powernow: This module only works with AMD K7 CPUs
bootsplash: status on console 0 changed to on

さらに古いイベントについては、`/var/log/messages`や`/var/log/warn`ファイルに保存されます。

13.2.5 List of Open Files: lsof

指定したプロセス ID (*PID*) で開いている 全ファイルを一覧表示するには、`-p` オプションを 指定します。たとえば、現在のシェルが開いている全てのファイルを確認するには、下記のように入力します:

```
tux@mercury:~> lsof -p $$
COMMAND PID  USER  FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    5552 tux    cwd   DIR   3,3   1512 117619 /home/tux
bash    5552 tux    rtd   DIR   3,3    584     2 /
bash    5552 tux    txt   REG   3,3  498816 13047 /bin/bash
bash    5552 tux    mem   REG   0,0         0 [heap] (stat: No such
bash    5552 tux    mem   REG   3,3  217016 115687 /var/run/nscd/passwd
bash    5552 tux    mem   REG   3,3  208464 11867 /usr/lib/locale/en_GB.
[...]
bash    5552 tux    mem   REG   3,3    366  9720 /usr/lib/locale/en_GB.
bash    5552 tux    mem   REG   3,3   97165  8828 /lib/ld-2.3.6.so
bash    5552 tux     0u   CHR  136,5         7 /dev/pts/5
bash    5552 tux     1u   CHR  136,5         7 /dev/pts/5
bash    5552 tux     2u   CHR  136,5         7 /dev/pts/5
bash    5552 tux    255u  CHR  136,5         7 /dev/pts/5
```

ここで`$$`は特別なシェル変数で、シェル自身のプロセス ID に置き換えられて実行されます。

また `lsof` に何もオプションを指定しないで実行すると、現在開かれている全てのファイルを一覧表示します。開かれている全ての ファイルを合計すると多数のファイルになってしまうため、全ファイルを表示してもあまり便利ではありません。その代わりに、検索機能を組み合わせる ことで、使いやすい一覧にするのがお勧めです。たとえばキャラクタデバイスの一覧を表示するには、下記のように入力します:

```
tux@mercury:~> lsof | grep CHR
bash    3838 tux     0u   CHR  136,0         2 /dev/pts/0
bash    3838 tux     1u   CHR  136,0         2 /dev/pts/0
bash    3838 tux     2u   CHR  136,0         2 /dev/pts/0
bash    3838 tux    255u  CHR  136,0         2 /dev/pts/0
bash    5552 tux     0u   CHR  136,5         7 /dev/pts/5
bash    5552 tux     1u   CHR  136,5         7 /dev/pts/5
bash    5552 tux     2u   CHR  136,5         7 /dev/pts/5
```

```

bash      5552      tux   255u      CHR  136,5      7 /dev/pts/5
X          5646      root  mem        CHR   1,1     1006 /dev/mem
lsof       5673      tux    0u      CHR  136,5      7 /dev/pts/5
lsof       5673      tux    2u      CHR  136,5      7 /dev/pts/5
grep       5674      tux    1u      CHR  136,5      7 /dev/pts/5
grep       5674      tux    2u      CHR  136,5      7 /dev/pts/5

```

-i オプションを指定すると、lsof はインターネットファイルについても表示するようになります:

```

tux@mercury:~> lsof -i
[...]
pidgin      4349 tux   17r IPv4  15194      0t0 TCP \
  jupiter.example.com:58542->www.example.net:https (ESTABLISHED)
pidgin      4349 tux   21u IPv4  15583      0t0 TCP \
  jupiter.example.com:37051->aol.example.org:aol (ESTABLISHED)
evolution   4578 tux   38u IPv4  16102      0t0 TCP \
  jupiter.example.com:57419->imap.example.com:imaps (ESTABLISHED)
npviewer.   9425 tux   40u IPv4  24769      0t0 TCP \
  jupiter.example.com:51416->www.example.com:http (CLOSE_WAIT)
npviewer.   9425 tux   49u IPv4  24814      0t0 TCP \
  jupiter.example.com:43964->www.example.org:http (CLOSE_WAIT)
ssh         17394 tux    3u IPv4  40654      0t0 TCP \
  jupiter.example.com:35454->saturn.example.com:ssh (ESTABLISHED)

```

13.2.6 カーネルと udev のイベント順序監視: udevadm monitor

udevadm monitor コマンドは、udev のルールが発信した カーネルイベントと uevent を確認し、コンソールに対してそのイベントの デバイスパス (DEVPATH) を表示します。下記は USB メモリスティックを接続した場合のイベント例です:

注記: udev イベントの監視

root ユーザだけが udevadm コマンドを利用して udev イベントを監視することができます。

```

UEVENT[1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2
UEVENT[1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UEVENT[1138806687] add@/class/scsi_host/host4
UEVENT[1138806687] add@/class/usb_device/usbdev4.10
UDEV [1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2
UDEV [1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UDEV [1138806687] add@/class/scsi_host/host4
UDEV [1138806687] add@/class/usb_device/usbdev4.10
UEVENT[1138806692] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UEVENT[1138806692] add@/block/sdb
UEVENT[1138806692] add@/class/scsi_generic/sg1
UEVENT[1138806692] add@/class/scsi_device/4:0:0:0
UDEV [1138806693] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UDEV [1138806693] add@/class/scsi_generic/sg1
UDEV [1138806693] add@/class/scsi_device/4:0:0:0
UDEV [1138806693] add@/block/sdb
UEVENT[1138806694] add@/block/sdb/sdb1
UDEV [1138806694] add@/block/sdb/sdb1
UEVENT[1138806694] mount@/block/sdb/sdb1
UEVENT[1138806697] umount@/block/sdb/sdb1

```

13.2.7 セキュリティのイベント情報: audit

Linuxの監査フレームワークは複雑な監査システムで、全てのセキュリティ関連イベントを収集します。これらの記録はその後セキュリティポリシーの違反が発生していないかどうかを調査するために分析が行なわれます。

13.2.8 X11 クライアントが利用しているサーバ資源情報の表示: xrestop

xrestop は、それぞれの接続済み X11 クライアントについて、サーバ側で消費されている資源情報を表示します。出力は 13.3.4 項「プロセス表: top」(213 ページ) とよく似た形になっています。

```

xrestop - Display: localhost:0
          Monitoring 40 clients. XErrors: 0
          Pixmaps: 42013K total, Other: 206K total, All: 42219K total

```

res-base	Wins	GCS	Fnts	Pxms	Misc	Pxm mem	Other	Total	PID	Identifier
3e00000	385	36	1	751	107	18161K	13K	18175K	?	NOVELL: SU
4600000	391	122	1	1182	889	4566K	33K	4600K	?	amaroK - S
1600000	35	11	0	76	142	3811K	4K	3816K	?	KDE Deskto
3400000	52	31	1	69	74	2816K	4K	2820K	?	Linux Shel
2c00000	50	25	1	43	50	2374K	3K	2378K	?	Linux Shel
2e00000	50	10	1	36	42	2341K	3K	2344K	?	Linux Shel

2600000	37	24	1	34	50	1772K	3K	1775K	?	Root - Kon
4800000	37	24	1	34	49	1772K	3K	1775K	?	Root - Kon
2a00000	209	33	1	323	238	1111K	12K	1123K	?	Trekstor25
1800000	182	32	1	302	285	1039K	12K	1052K	?	kicker
1400000	157	121	1	231	477	777K	18K	796K	?	kwin
3c00000	175	36	1	248	168	510K	9K	520K	?	de.comp.la
3a00000	326	42	1	579	444	486K	20K	506K	?	[opensuse-
0a00000	85	38	1	317	224	102K	9K	111K	?	Kopete
4e00000	25	17	1	60	66	63K	3K	66K	?	YaST Contr
2400000	11	10	0	56	51	53K	1K	55K	22061	suseplugge
0e00000	20	12	1	50	92	50K	3K	54K	22016	kded
3200000	6	41	5	72	84	40K	8K	48K	?	EMACS
2200000	54	9	1	30	31	42K	3K	45K	?	SUSEWatche
4400000	2	11	1	30	34	34K	2K	36K	16489	kdesu
1a00000	255	7	0	42	11	19K	6K	26K	?	KMix
3800000	2	14	1	34	37	21K	2K	24K	22242	knotify
1e00000	10	7	0	42	9	15K	624B	15K	?	KPowersave
3600000	106	6	1	30	9	7K	3K	11K	22236	konqueror
2000000	10	5	0	21	34	9K	1K	10K	?	klipper
+3000000	21	7	0	11	9	7K	888B	8K	?	KDE Wallet

13.3 プロセス

13.3.1 プロセス間通信: ipcs

ipcs プログラムは、現在使用中の IPC (プロセス間通信) を表示することができます:

```

----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x00000000   58261504   tux        600         393216     2           dest
0x00000000   58294273   tux        600         196608     2           dest
0x00000000   83886083   tux        666         43264      2
0x00000000   83951622   tux        666         192000     2
0x00000000   83984391   tux        666         282464     2
0x00000000   84738056   root       644         151552     2           dest

----- Semaphore Arrays -----
key          semid      owner      perms      nsems
0x4d038abf   0          tux        600         8

----- Message Queues -----
key          msqid      owner      perms      used-bytes   messages

```

13.3.2 プロセスの一覧: ps

ps コマンドは、プロセスの一覧を表示することができます。ほとんどのパラメータはマイナス記号なしで書くことができます。簡潔な ヘルプを読む場合は `ps --help` と入力し、より詳しいヘルプを読む場合はマニュアルページをお読みください。

また、全てのプロセスについてユーザとコマンドライン情報を表示するには、`ps axu` と入力します:

```
tux@mercury:~> ps axu
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   696   272 ?        S    12:59    0:01 init [5]
root         2  0.0  0.0     0     0 ?        SN   12:59    0:00 [ksoftirqd
root         3  0.0  0.0     0     0 ?        S<   12:59    0:00 [events
[...]
```

tux	4047	0.0	6.0	158548	31400	?	Ss1	13:02	0:06	mono-best
tux	4057	0.0	0.7	9036	3684	?	Sl	13:02	0:00	/opt/gnome
tux	4067	0.0	0.1	2204	636	?	S	13:02	0:00	/opt/gnome
tux	4072	0.0	1.0	15996	5160	?	Ss	13:02	0:00	gnome-scre
tux	4114	0.0	3.7	130988	19172	?	SLl	13:06	0:04	sound-juic
tux	4818	0.0	0.3	4192	1812	pts/0	Ss	15:59	0:00	-bash
tux	4959	0.0	0.1	2324	816	pts/0	R+	16:17	0:00	ps axu

たとえば `sshd` プロセスがいくつ動作しているのかを調べるには、`-p` オプションと `pidof` コマンドを併用してください。 `pidof` コマンドは、指定された名前のコマンドを実行しているプロセス ID を、一覧表示する コマンドです。

```
tux@mercury:~> ps -p $(pidof sshd)
  PID TTY      STAT   TIME COMMAND
 3524 ?        Ss      0:00 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
 4813 ?        Ss      0:00 sshd: tux [priv]
 4817 ?        R       0:00 sshd: tux@pts/0
```

プロセス一覧の書式は必要に応じて変更することができます。 `-L` オプションを指定して実行すると、書式に関する全てのキーワードを出力することができます。たとえば下記のコマンドでは、全てのプロセスをメモリの使用率順に並べて表示します:

```
tux@mercury:~> ps ax --format pid,rss,cmd --sort rss
  PID  RSS CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
    4     0 [khelper]
    5     0 [kthread]
   11     0 [kblockd/0]
```

```

12      0 [kacpid]
472     0 [pdflush]
473     0 [pdflush]
[...]
4028 17556 nautilus --no-default-window --sm-client-id default2
4118 17800 ksnapshot
4114 19172 sound-juicer
4023 25144 gnome-panel --sm-client-id default1
4047 31400 mono-best --debug /usr/lib/beagle/Best.exe --autostarted
3973 31520 mono-beagled --debug /usr/lib/beagle/BeagleDaemon.exe --bg --aut

```

便利な **ps** 呼び出し

```
ps aux --sort 列
```

列の出力順に並べ替えます。それぞれ列には下記を指定できます。

物理メモリの比率: `pmem`

CPU の使用率: `pcpu`

常駐セットサイズ (スワップしていない物理メモリ): `rss`

```
ps axo pid,%cpu,rss,vsz,args,wchan
```

各プロセスについて PID と CPU の使用率、メモリサイズ (常駐および仮想) と名前、およびシステムコール名を表示します。

```
ps axfo pid,args
```

プロセスツリー (木構造) を表示します。

13.3.3 プロセスのツリー表示: **pstree**

`pstree` コマンドは、プロセスの一覧をツリー (木) 構造で表示します:

```

tux@mercury:~> pstree
init--+-NetworkManagerD
      |-acpid
      |-3*[automount]
      |-cron
      |-cupsd
      |-2*[dbus-daemon]
      |-dbus-launch
      |-dcopserver
      |-dhcpcd
      |-events/0
      |-gpg-agent
      |-hald--+-hald-addon-acpi
              `--hald-addon-stor

```

```

|-kded
|-kdeinit-+-kdesu---su---kdesu_stub---yast2---y2controlcenter
|         |-kio_file
|         |-klauncher
|         |-konqueror
|         |-konsole-+-bash---su---bash
|         |         |-bash
|         |-kwin
|-kdesktop---kdesktop_lock---xmatrix
|-kdesud
|-kdm-+-X
|     |-kdm---startkde---kwrapper
[...]
```

`-p` オプションを指定すると、それぞれの名前の後ろに プロセス ID を表示することができます。コマンドラインについても表示するには、`-a` を指定してください。

13.3.4 プロセス表: `top`

`top` コマンドは "table of processes" (プロセス表) の略で、2 秒おきに更新するタイプのプロセス一覧を表示します。プログラムを終了するには、`Q` を押してください。また、`-n 1` パラメータを指定すると、プロセス一覧を一度表示するだけでプログラムが終了するようになります。下記は、`top -n 1` の出力例です:

```
tux@mercury:~> top -n 1
top - 17:06:28 up 2:10, 5 users, load average: 0.00, 0.00, 0.00
Tasks: 85 total, 1 running, 83 sleeping, 1 stopped, 0 zombie
Cpu(s): 5.5% us, 0.8% sy, 0.8% ni, 91.9% id, 1.0% wa, 0.0% hi, 0.0% si
Mem: 515584k total, 506468k used, 9116k free, 66324k buffers
Swap: 658656k total, 0k used, 658656k free, 353328k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	16	0	700	272	236	S	0.0	0.1	0:01.33	init
2	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
3	root	10	-5	0	0	0	S	0.0	0.0	0:00.27	events/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.01	khelper
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
11	root	10	-5	0	0	0	S	0.0	0.0	0:00.05	kblockd/0
12	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
472	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
473	root	15	0	0	0	0	S	0.0	0.0	0:00.06	pdflush
475	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
474	root	15	0	0	0	0	S	0.0	0.0	0:00.07	kswapd0
681	root	10	-5	0	0	0	S	0.0	0.0	0:00.01	kseriod
839	root	10	-5	0	0	0	S	0.0	0.0	0:00.02	reiserfs/0
923	root	13	-4	1712	552	344	S	0.0	0.1	0:00.67	udevd
1343	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khubb
1587	root	20	0	0	0	0	S	0.0	0.0	0:00.00	shpchpd_event
1746	root	15	0	0	0	0	S	0.0	0.0	0:00.00	wl_control
1752	root	15	0	0	0	0	S	0.0	0.0	0:00.00	wl_bus_master1
2151	root	16	0	1464	496	416	S	0.0	0.1	0:00.00	acpid
2165	messageb	16	0	3340	1048	792	S	0.0	0.2	0:00.64	dbus-daemon
2166	root	15	0	1840	752	556	S	0.0	0.1	0:00.01	syslog-ng
2171	root	16	0	1600	516	320	S	0.0	0.1	0:00.00	klogd
2235	root	15	0	1736	800	652	S	0.0	0.2	0:00.10	resmgrd
2289	root	16	0	4192	2852	1444	S	0.0	0.6	0:02.05	hald
2403	root	23	0	1756	600	524	S	0.0	0.1	0:00.00	hald-addon-acpi
2709	root	19	0	2668	1076	944	S	0.0	0.2	0:00.00	NetworkManagerD
2714	root	16	0	1756	648	564	S	0.0	0.1	0:00.56	hald-addon-stor

既定では **CPU** の使用率順 (**%CPU** の列、ショートカットでは **Shift + P**) に並べられています。それぞれ下記のショートカットで並び順を変更することができます:

Shift + M: 常駐メモリ (**RES**)

Shift + N: プロセス ID (**PID**)

Shift + T: CPU 時間 (**TIME+**)

それ以外の項目を並べ替えに使用するには、**F** を押して一覧から項目を選択します。並べ替え順序(昇順、降順)を切り替えるには、**Shift + R** を押します。

また、**-U** *UID* オプションを指定すると、特定のユーザのプロセスだけを監視するようになります。ここで、*UID* にはユーザのユーザ ID を指定してく

ださい。たとえば、`top -U $(id -u)` のように入力すると、自分自身が起動したプロセスについて一覧表示を行ないます。

13.3.5 プロセスの nice 値の変更: nice と renice

カーネルは各プロセスの優先レベル (nice 値とも呼びます) をもとにして、どのプロセスに対してより多くの CPU 時間を与えるかを決定します。あるプロセスに対してより大きい「nice」値が設定されている場合、プロセスに与えられる CPU 時間は少なくなります。nice 値は -20 (最優先) から 19 までの間で設定します。マイナスの値は root だけが設定できます。

nice 値のレベルを調整すると、処理時間が長くなるだけでなく CPU 時間も大きく消費し、時間面の制約が少ないプログラム、たとえばカーネルのコンパイル作業を動作させるような場合に効果があります。それらのプロセスの「nice 値を大きくする」ことで、Web サーバなどの他の処理を優先的に動作させることができます。

また、オプションを何も指定せずに nice を実行すると、現在の nice 値を表示することができます:

```
tux@mercury:~> nice
0
```

nice コマンドのように実行すると、指定したコマンドの nice 値を 10 だけプラスします。また、`nice -n レベル` コマンドのように実行すると、現在の nice 値に対する相対値で指定することができます。

実行中のプロセスに対する nice 値を変更するには、`renice 優先度 -p プロセス id` のように実行します。たとえば:

```
renice +5 3266
```

指定したユーザの全プロセスに対する nice 値を変更するには、`-u ユーザオプション` をご利用ください。また、プロセスグループを指定する場合は、`-g プロセスグループ ID` のように指定します。

13.4 メモリ

13.4.1 メモリの使用量: free

free ユーティリティは、メモリの使用率を表示することができます。空き容量の詳細や使用済み容量の詳細、スワップ領域の詳細がそれぞれ表示されます:

```
tux@mercury:~> free
```

	total	used	free	shared	buffers	cached
Mem:	2062844	2047444	15400	0	129580	921936
-/+ buffers/cache:		995928	1066916			
Swap:	2104472	0	2104472			

それぞれ `-b`, `-k`, `-m`, `-g` オプションを指定すると、バイト単位/キロバイト単位/メガバイト単位/ギガバイト単位に値を表示するようになります。また、`-d` 時間 オプションを指定すると、*時間* で指定した秒間隔ごとに表示が更新されるようになります。たとえば `free -d 1.5` と入力すると、1.5 秒間隔で更新が行われます。

13.4.2 詳細なメモリ使用量: /proc/meminfo

/proc/meminfo を利用することで、free が表示するデータよりもさらに詳細な情報を得ることができます。実際のところ、free はここからの出力を利用しています。下記に 64 ビットシステムでの表示例を示します。32 ビットシステムの場合、メモリの管理形態が異なる都合上、出力項目が若干異なることに注意してください:

```
tux@mercury:~> cat /proc/meminfo
```

MemTotal:	8182956 kB
MemFree:	1045744 kB
Buffers:	364364 kB
Cached:	5601388 kB
SwapCached:	1936 kB
Active:	4048268 kB
Inactive:	2674796 kB
Active(anon):	663088 kB
Inactive(anon):	107108 kB
Active(file):	3385180 kB
Inactive(file):	2567688 kB
Unevictable:	4 kB
Mlocked:	4 kB

```
SwapTotal:      2096440 kB
SwapFree:       2076692 kB
Dirty:          44 kB
Writeback:      0 kB
AnonPages:      756108 kB
Mapped:         147320 kB
Slab:           329216 kB
SReclaimable:   300220 kB
SUnreclaim:     28996 kB
PageTables:     21092 kB
NFS_Unstable:   0 kB
Bounce:         0 kB
WritebackTmp:   0 kB
CommitLimit:    6187916 kB
Committed_AS:   1388160 kB
VmallocTotal:   34359738367 kB
VmallocUsed:    133384 kB
VmallocChunk:   34359570939 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize:   2048 kB
DirectMap4k:    2689024 kB
DirectMap2M:    5691392 kB
```

最も重要な項目は下記のとおりです:

MemTotal

利用可能な RAM の合計値

MemFree

未使用の RAM の合計値

Buffers

RAM 内のファイルバッファキャッシュ

Cached

RAM 内のページキャッシュ

SwapCached

スワップ内のページキャッシュ

Active

通常は再利用のできない、直近で使用されたメモリ。この値は匿名ページ (*Active(anon)* として一覧表示されるもの) によって 要求されたメモリと、ファイルバックのページ (*Active(file)*) の合計値を示しています。

Inactive

再利用の可能な、直近では使用されていないメモリ。この値は匿名ページ (*Inactive(anon)*)として一覧表示されるもの)によって要求されたメモリと、ファイルバックページ (*Inactive(file)*) の合計値を示しています。

SwapTotal

スワップ領域の合計サイズ

SwapFree

未使用のスワップ領域の合計サイズ

Dirty

ディスクに書き込まれる予定のメモリ量

Writeback

現在ディスクに書き込んでいる最中のメモリ量

Mapped

nmap コマンドで割り当てられたメモリ

Slab

カーネルのデータ構造キャッシュ

Committed_AS

現在の負荷量进行处理するにあたって必要なメモリ量 (RAM とスワップ) の合計値の見積もり

13.4.3 プロセスのメモリ使用量: smaps

特定のプロセスがどれだけメモリを利用しているのかを正確に知るには、top や ps のような標準ツールでは判断できません。このような要件には、カーネル 2.6.14 から提供されるようになった **smaps** サブシステムをお使いください。smaps サブシステムは /proc/プロセス ID/smaps からアクセスできます。なお プロセス ID は実際のプロセス ID に置き換えてください。このサブシステムは共有メモリとプライベートメモリを区別して表示するため、他のプロセスと共有されておらず独占的に使用されているメモリ量を判断することができます。

13.5 ネットワーク

13.5.1 ネットワーク状態の表示: netstat

netstat はネットワーク接続を表示することができるほか、ルーティングテーブル (-r), インターフェイス (-i), マスカレード接続 (-M), マルチキャストのメンバーシップ情報 (-g), 統計情報 (-s) をそれぞれ表示することができます。

```
tux@mercury:~> netstat -r
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
192.168.2.0      *                255.255.254.0   U        0 0        0 eth0
link-local       *                255.255.0.0     U        0 0        0 eth0
loopback         *                255.0.0.0       U        0 0        0 lo
default          192.168.2.254   0.0.0.0         UG       0 0        0 eth0
```

```
tux@mercury:~> netstat -i
Kernel Interface table
Iface  MTU Met  RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500  0  1624507 129056      0      0  7055      0      0      0 BMNRU
lo     16436  0   23728      0          0      0  23728      0      0      0 LRU
```

ネットワーク接続や統計情報を表示する際、表示するソケットタイプを指定することができます。それぞれ TCP (-t), UDP (-u), raw (-r) を指定してください。また、-p を指定すると、各ソケットの属している PID とプログラム名を表示します。

下記の例では、全ての TCP 接続とそれらの接続を使用しているプログラムを表示しています:

```
mercury:~ # netstat -t -p
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address   Foreign Address   State           PID/Pro
[...]
```

tcp	0	0	mercury:33513	www.novell.com:www-http	ESTABLISHED	6862/fi
tcp	0	352	mercury:ssh	mercury2.:trc-netpoll	ESTABLISHED	
19422/s						
tcp	0	0	localhost:ssh	localhost:17828	ESTABLISHED	-

下記の例では、TCP プロトコルの統計情報を表示しています:

```
tux@mercury:~> netstat -s -t
Tcp:
    2427 active connections openings
    2374 passive connection openings
```

```

0 failed connection attempts
0 connection resets received
1 connections established
27476 segments received
26786 segments send out
54 segments retransmited
0 bad segments received.
6 resets sent
[...]
TCPAbortOnLinger: 0
TCPAbortFailed: 0
TCPMemoryPressures: 0

```

13.5.2 対話的なネットワークモニタ: iptraf

iptrafはメニューベースのローカルエリアネットワーク (LAN) モニタです。TCP, UDP のカウントやイーサネットの負荷情報、IP チェックサムエラーなどのネットワーク統計情報を生成します。

何もオプションを指定せずに実行した場合は、対話モードで動作します。グラフィカルなメニューを利用することができ、iptraf で収集したい統計情報を選択することができます。ネットワークインターフェイスを指定することもできます。

図 13.1 iptraf の対話モード実行

	Total Packets	Total Bytes	Incoming Packets	Incoming Bytes	Outgoing Packets	Outgoing Bytes
Total:	1577	984734	881	906268	696	78466
IP:	1577	961844	881	893122	696	68722
TCP:	1466	948809	827	883927	639	64882
UDP:	110	12815	54	9195	56	3620
ICMP:	1	220	0	0	1	220
Other IP:	0	0	0	0	0	0
Non-IP:	0	0	0	0	0	0

Total rates:	706.7 kbits/sec	Broadcast packets:	0
	113.8 packets/sec	Broadcast bytes:	0
Incoming rates:	677.3 kbits/sec		
	69.0 packets/sec	IP checksum errors:	0
Outgoing rates:	29.4 kbits/sec		
	44.8 packets/sec		

Elapsed time: 0:00
X=exit

また、iptraf コマンドには複数のオプションがあり、バッチモードでも動作させることができます。下記の例では、ネットワーク インターフェイス

eth0 (-i) の統計情報を 1 分間収集する 動作をします。さらに、本プログラムはバックグラウンド (-B) で動作させることもできます。この場合、統計情報はお使いのホームディレクトリ内にある iptraf.log ファイルに出力されます (-L)。

```
tux@mercury:~> iptraf -i eth0 -t 1 -B -L ~/iptraf.log
```

生成されたログファイルは、more コマンドで閲覧できます:

```
tux@mercury:~> more ~/iptraf.log
Mon Mar 23 10:08:02 2010; ***** IP traffic monitor started *****
Mon Mar 23 10:08:02 2010; UDP; eth0; 107 bytes; from 192.168.1.192:33157 to \
239.255.255.253:427
Mon Mar 23 10:08:02 2010; VRRP; eth0; 46 bytes; from 192.168.1.252 to \
224.0.0.18
Mon Mar 23 10:08:03 2010; VRRP; eth0; 46 bytes; from 192.168.1.252 to \
224.0.0.18
Mon Mar 23 10:08:03 2010; VRRP; eth0; 46 bytes; from 192.168.1.252 to \
224.0.0.18
[...]
Mon Mar 23 10:08:06 2010; UDP; eth0; 132 bytes; from 192.168.1.54:54395 to \
10.20.7.255:111
Mon Mar 23 10:08:06 2010; UDP; eth0; 46 bytes; from 192.168.1.92:27258 to \
10.20.7.255:8765
Mon Mar 23 10:08:06 2010; UDP; eth0; 124 bytes; from 192.168.1.139:43464 to \
10.20.7.255:111
Mon Mar 23 10:08:06 2010; VRRP; eth0; 46 bytes; from 192.168.1.252 to \
224.0.0.18
--More--(7%)
```

13.6 /proc ファイルシステム

/proc ファイルシステムはカーネルが提供する擬似的な ファイルシステムで、ファイルの形式で各種の重要情報を提供します。たとえば下記のコマンドを入力すると、CPU の種類を表示することができます:

```
tux@mercury:~> cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 15
model         : 4
model name    : Intel(R) Pentium(R) 4 CPU 3.40GHz
stepping      : 3
cpu MHz       : 2800.000
cache size    : 2048 KB
```

```
physical id      : 0
[...]
```

また、割り込みの割り当てや使用について調べるには、下記のようなコマンドを入力します:

```
tux@mercury:~> cat /proc/interrupts
          CPU0
 0:       3577519      XT-PIC  timer
 1:          130      XT-PIC  i8042
 2:           0      XT-PIC  cascade
 5:       564535      XT-PIC  Intel 82801DB-ICH4
 7:           1      XT-PIC  parport0
 8:           2      XT-PIC  rtc
 9:           1      XT-PIC  acpi, uhci_hcd:usb1, ehci_hcd:usb4
10:           0      XT-PIC  uhci_hcd:usb3
11:       71772      XT-PIC  uhci_hcd:usb2, eth0
12:      101150      XT-PIC  i8042
14:       33146      XT-PIC  ide0
15:      149202      XT-PIC  ide1
NMI:           0
LOC:           0
ERR:           0
MIS:           0
```

下記に `/proc` 以下にある重要なファイルと、その内容説明を記します:

`/proc/devices`
利用可能なデバイスの一覧

`/proc/modules`
読み込まれたカーネルモジュール

`/proc/cmdline`
カーネルのコマンドラインパラメータ

`/proc/meminfo`
メモリの使用率に関する詳細情報

`/proc/config.gz`
現在実行中のカーネルについて、そのカーネル設定ファイルをgzip圧縮したもの

さらなる情報は、`/usr/src/linux/Documentation/filesystems/proc.txt` のテキストファイル内に書かれています (なお、このファイルは `kernel-source` パッケージをインストールした場合にのみ利用できます)

。また、現在実行中のプロセスに関する情報は、`/proc/NNN` ディレクトリ以下から読み取ることができます。ここで *NNN* にはプロセス ID (PID) を指定します。自分自身のプロセスについて調べる場合は、`/proc/self/` を利用して調べることができます:

```
tux@mercury:~> ls -l /proc/self
lrwxrwxrwx 1 root root 64 2007-07-16 13:03 /proc/self -> 5356
tux@mercury:~> ls -l /proc/self/
total 0
dr-xr-xr-x 2 tux users 0 2007-07-16 17:04 attr
-r----- 1 tux users 0 2007-07-16 17:04 auxv
-r--r--r-- 1 tux users 0 2007-07-16 17:04 cmdline
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 cwd -> /home/tux
-r----- 1 tux users 0 2007-07-16 17:04 environ
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 exe -> /bin/ls
dr-x----- 2 tux users 0 2007-07-16 17:04 fd
-rw-r--r-- 1 tux users 0 2007-07-16 17:04 loginuid
-r--r--r-- 1 tux users 0 2007-07-16 17:04 maps
-rw----- 1 tux users 0 2007-07-16 17:04 mem
-r--r--r-- 1 tux users 0 2007-07-16 17:04 mounts
-rw-r--r-- 1 tux users 0 2007-07-16 17:04 oom_adj
-r--r--r-- 1 tux users 0 2007-07-16 17:04 oom_score
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 root -> /
-rw----- 1 tux users 0 2007-07-16 17:04 seccomp
-r--r--r-- 1 tux users 0 2007-07-16 17:04 smaps
-r--r--r-- 1 tux users 0 2007-07-16 17:04 stat
[...]
dr-xr-xr-x 3 tux users 0 2007-07-16 17:04 task
-r--r--r-- 1 tux users 0 2007-07-16 17:04 wchan
```

実行ファイルとライブラリのアドレス割り当ては、`maps` ファイル内に書かれています:

```
tux@mercury:~> cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:03 17753      /bin/cat
0804c000-0804d000 rw-p 00004000 03:03 17753      /bin/cat
0804d000-0806e000 rw-p 0804d000 00:00 0        [heap]
b7d27000-b7d5a000 r--p 00000000 03:03 11867      /usr/lib/locale/en_GB.utf8/
b7d5a000-b7e32000 r--p 00000000 03:03 11868      /usr/lib/locale/en_GB.utf8/
b7e32000-b7e33000 rw-p b7e32000 00:00 0
b7e33000-b7f45000 r-xp 00000000 03:03 8837       /lib/libc-2.3.6.so
b7f45000-b7f46000 r--p 00112000 03:03 8837       /lib/libc-2.3.6.so
b7f46000-b7f48000 rw-p 00113000 03:03 8837       /lib/libc-2.3.6.so
b7f48000-b7f4c000 rw-p b7f48000 00:00 0
b7f52000-b7f53000 r--p 00000000 03:03 11842      /usr/lib/locale/en_GB.utf8/
[...]
b7f5b000-b7f61000 r--s 00000000 03:03 9109       /usr/lib/gconv/gconv-module
b7f61000-b7f62000 r--p 00000000 03:03 9720       /usr/lib/locale/en_GB.utf8/
b7f62000-b7f76000 r-xp 00000000 03:03 8828       /lib/ld-2.3.6.so
b7f76000-b7f78000 rw-p 00013000 03:03 8828       /lib/ld-2.3.6.so
```

```
bfd61000-bfd76000 rw-p bfd61000 00:00 0 [stack]
ffffe000-fffff000 ---p 00000000 00:00 0 [vdso]
```

13.6.1 procinfo

/proc ファイルシステムにある重要な情報は、procinfo コマンドを利用することで概要表示を行なうことができます:

```
tux@mercury:~> procinfo
Linux 2.6.32.7-0.2-default (geeko@buildhost) (gcc 4.3.4) #1 2CPU

Memory:      Total      Used      Free      Shared      Buffers
Mem:         2060604    2011264    49340      0         200664
Swap:        2104472      112     2104360

Bootup: Wed Feb 17 03:39:33 2010      Load average: 0.86 1.10 1.11 3/118 21547

user   :      2:43:13.78    0.8%  page in :    71099181  disk 1:  2827023r 968
nice   :    1d 22:21:27.87  14.7%  page out:  690734737
system:    13:39:57.57    4.3%  page act:  138388345
IOwait:    18:02:18.59    5.7%  page dea:  29639529
hw irq:     0:03:39.44    0.0%  page flt: 9539791626
sw irq:     1:15:35.25    0.4%  swap in :      69
idle    :    9d 16:07:56.79 73.8%  swap out:    209
uptime:     6d 13:07:11.14      context :  542720687

irq 0: 141399308 timer      irq 14:  5074312 ide0
irq 1:   73784 i8042        irq 50:  1938076 uhci_hcd:usb1, ehci_
irq 4:      2             irq 58:      0 uhci_hcd:usb2
irq 6:      5 floppy [2]   irq 66:  872711 uhci_hcd:usb3, HDA I
irq 7:      2             irq 74:    15 uhci_hcd:usb4
irq 8:      0 rtc         irq 82: 178717720 0          PCI-MSI  e
irq 9:      0 acpi        irq169: 44352794 nvidia
irq 12:     3             irq233: 8209068 0          PCI-MSI  1
```

全ての情報を表示するには、`-a` オプションを指定してください。`-nN` オプションを指定すると、`N` 秒おきに情報を更新するようになります。この場合は、`q` を押すとプログラムを終了することができます。

既定では累積値を表示するようになっていますが、`-d` を指定すると差を表示するようになります。たとえば `procinfo -dn5` と入力すると、最近 5 秒間に変更のあった値が表示されます。

13.7 ハードウェア情報

13.7.1 PCI 資源: lspci

注記: PCI 設定へのアクセスについて

コンピュータの PCI 設定にアクセスするには、多くのオペレーティングシステムで **root** のユーザ権限が必要です。

lspci コマンドは PCI 資源の一覧を表示します:

```
mercury:~ # lspci
00:00.0 Host bridge: Intel Corporation 82845G/GL[Brookdale-G]/GE/PE \
    DRAM Controller/Host-Hub Interface (rev 01)
00:01.0 PCI bridge: Intel Corporation 82845G/GL[Brookdale-G]/GE/PE \
    Host-to-AGP Bridge (rev 01)
00:1d.0 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #1 (rev 01)
00:1d.1 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #2 (rev 01)
00:1d.2 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #3 (rev 01)
00:1d.7 USB Controller: Intel Corporation 82801DB/DBM \
    (ICH4/ICH4-M) USB2 EHCI Controller (rev 01)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev 81)
00:1f.0 ISA bridge: Intel Corporation 82801DB/DBL (ICH4/ICH4-L) \
    LPC Interface Bridge (rev 01)
00:1f.1 IDE interface: Intel Corporation 82801DB (ICH4) IDE \
    Controller (rev 01)
00:1f.3 SMBus: Intel Corporation 82801DB/DBL/DBM (ICH4/ICH4-L/ICH4-M) \
    SMBus Controller (rev 01)
00:1f.5 Multimedia audio controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) AC'97 Audio Controller (rev 01)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. G400/G450 (rev 85)
02:08.0 Ethernet controller: Intel Corporation 82801DB PRO/100 VE (LOM) \
    Ethernet Controller (rev 81)
```

-v オプションを使用すると、さらに詳細な一覧を表示することができます:

```
mercury:~ # lspci -v
[...]
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet \
    Controller (rev 02)
    Subsystem: Intel Corporation PRO/1000 MT Desktop Adapter
    Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 19
    Memory at f0000000 (32-bit, non-prefetchable) [size=128K]
    I/O ports at d010 [size=8]
```

```
Capabilities: [dc] Power Management version 2
Capabilities: [e4] PCI-X non-bridge device
Kernel driver in use: e1000
Kernel modules: e1000
```

デバイス名の解決は /usr/share/pci.ids ファイルを利用して行なわれます。このファイル内に対応する **PCI ID** とその名前が存在しない場合、「Unknown device」(未知のデバイス) として表示されます。

-vv オプションを使用すると、プログラムで問い合わせ可能な全ての情報を出力します。また、名前ではなく数値で表示させたい場合は、-n オプションを使用してください。

13.7.2 USB デバイス: lsusb

lsusb コマンドは全ての USB デバイスを一覧表示します。-v オプションを指定すると、さらに詳しい一覧が表示されます。詳しい一覧は /proc/bus/usb/ ディレクトリから読み出す仕組みになっています。下記は USB ハブ、USB メモリスティック、ハードディスク、マウスがそれぞれ USB 経由で接続されている場合の例です。

```
mercury:/ # lsusb
Bus 004 Device 007: ID 0ea0:2168 Ours Technology, Inc. Transcend JetFlash \
  2.0 / Astone USB Drive
Bus 004 Device 006: ID 04b4:6830 Cypress Semiconductor Corp. USB-2.0 IDE \
  Adapter
Bus 004 Device 005: ID 05e3:0605 Genesys Logic, Inc.
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 005: ID 046d:c012 Logitech, Inc. Optical Mouse
Bus 001 Device 001: ID 0000:0000
```

13.8 ファイルとファイルシステム

13.8.1 ファイル種類の判別: file

file コマンドは、指定した単一または複数のファイルについて、/usr/share/misc/magic から読み込んだデータを元に種類を判別します。

```
tux@mercury:~> file /usr/bin/file
/usr/bin/file: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), \
for GNU/Linux 2.6.4, dynamically linked (uses shared libs), stripped
```

`-f` 一覧を指定すると、種類を判別したいファイル名の一覧が書かれたファイルを指定することができます。また、`-z` を指定すると、`file` が圧縮ファイル内の判別を行なうようになります:

```
tux@mercury:~> file /usr/share/man/man1/file.1.gz
usr/share/man/man1/file.1.gz: gzip compressed data, from Unix, max compression
tux@mercury:~> file -z /usr/share/man/man1/file.1.gz
/usr/share/man/man1/file.1.gz: troff or preprocessor input text \
(gzip compressed data, from Unix, max compression)
```

また、`-i` を指定すると、従来の表記法ではなく MIME タイプ の文字列を表示します。

```
tux@mercury:~> file -i /usr/share/misc/magic
/usr/share/misc/magic: text/plain charset=utf-8
```

13.8.2 ファイルシステムとその使用率: **mount, df, du**

`mount` コマンドは、各マウントポイントにどのファイルシステムがマウントされているかを表示します:

```
tux@mercury:~> mount
/dev/sda3 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sda1 on /boot type ext2 (rw,acl,user_xattr)
/dev/sda4 on /local type reiserfs (rw,acl,user_xattr)
/dev/fd0 on /media/floppy type subfs (rw,nosuid,nodev,noatime,fs=floppyfss,p
```

ファイルシステム内での使用率に関する情報を得たい場合は、`df`をお使いください。`-h`(または`--human-readable`)オプションを指定すると、一般的なユーザにとって、より読みやすい形式で表示することができます。

```
tux@mercury:~> df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        11G   3.2G   6.9G  32% /
udev            252M   104K   252M   1% /dev
/dev/sda1         16M    6.6M    7.8M  46% /boot
/dev/sda4        27G    34M    27G   1% /local
```

また、指定したディレクトリ内のファイルと、そのサブディレクトリ以下にあるファイルについてサイズを確認するには、`du`を使用します。`-s`オプションを指定すると、詳細な情報出力を省いて、パラメータで指定した各ディレクトリの合計サイズだけを表示します。こちらも`-h`オプションを指定することができます、同じく一般的なユーザにとって読みやすい形式で表示することができます:

```
tux@mercury:~> du -sh /opt
192M    /opt
```

13.8.3 ELF バイナリに関する追加情報表示

バイナリファイルの内容を読みたい場合は、`readelf`をお使いください。このコマンドでは、他のアーキテクチャ向けに作成された ELF 形式のファイルを読み取ることもできます:

```
tux@mercury:~> readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                               ELF64
  Data:                                   2's complement, little endian
  Version:                               1 (current)
  OS/ABI:                                UNIX - System V
  ABI Version:                           0
  Type:                                  EXEC (Executable file)
  Machine:                               Advanced Micro Devices X86-64
  Version:                               0x1
  Entry point address:                   0x402540
  Start of program headers:               64 (bytes into file)
  Start of section headers:               95720 (bytes into file)
  Flags:                                  0x0
  Size of this header:                     64 (bytes)
  Size of program headers:                 56 (bytes)
  Number of program headers:                9
  Size of section headers:                 64 (bytes)
```

```
Number of section headers:      32
Section header string table index: 31
```

13.8.4 ファイルの属性: stat

stat コマンドを利用すると、ファイルの属性を表示することができます:

```
tux@mercury:~> stat /etc/profile
  File: `/etc/profile'
  Size: 9662          Blocks: 24          IO Block: 4096   regular file
Device: 802h/2050d Inode: 132349         Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Access: 2009-03-20 07:51:17.000000000 +0100
Modify: 2009-01-08 19:21:14.000000000 +0100
Change: 2009-03-18 12:55:31.000000000 +0100
```

--filesystem オプションを指定すると、指定したファイルが配置されているファイルシステムに関する属性情報を表示します:

```
tux@mercury:~> stat /etc/profile --file-system
  File: "/etc/profile"
    ID: d4fb76e70b4d1746 Namelen: 255      Type: ext2/ext3
Block size: 4096          Fundamental block size: 4096
Blocks: Total: 2581445    Free: 1717327    Available: 1586197
Inodes: Total: 655776     Free: 490312
```

13.9 ユーザ情報

13.9.1 ユーザのファイルアクセス監視: fuser

どのプロセスやユーザが特定のファイルにアクセスしているのかを確認できれば便利です。たとえば /mnt にマウントされた ファイルシステムのマウントを解除したい場合、umount コマンドが "device is busy." としてマウントを解除できないことがあります。このような場合は、fuser コマンドを利用して、どのプロセスがデバイスを使用しているのかを確認することができます:

```
tux@mercury:~> fuser -v /mnt/*
```

	USER	PID	ACCESS	COMMAND
/mnt/notes.txt	tux	26597	f....	less

上記の例では他の端末から `less` プロセスを終了させると、ファイルシステムのマウント解除を行なうことができますようになります。なお `-k` オプションを指定すると、`fuser` は該当するファイルにアクセスしているプロセスに対し、プロセスの終了を行なうようになります。

13.9.2 ユーザの現状表示: `w`

`w` コマンドを利用すると、システムに誰がログインしているのかと、そのユーザが何をしているのかを表示することができます。たとえば下記のようにします:

```
tux@mercury:~> w
 14:58:43 up 1 day,  1:21,  2 users,  load average: 0.00, 0.00, 0.00
USER      TTY      LOGIN@  IDLE   JCPU   PCPU WHAT
tux       :0       12:25   ?xdm?  1:23   0.12s /bin/sh /usr/bin/startkde
root      pts/4    14:13   0.00s  0.06s  0.00s w
```

他のシステムのユーザが遠隔(リモート)からログインしている場合、`-f` オプションを指定すると、どのコンピュータからログインしているのかが表示されるようになります。

13.10 日付と時刻

13.10.1 経過時間の計測: `time`

`time` ユーティリティを利用すると、指定したコマンドを実行する際に消費された時間を表示することができます。このユーティリティはシェルの内蔵機能で提供されているほか、プログラム (`/usr/bin/time`) としても提供されています。

```
tux@mercury:~> time find . > /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s
```

13.11 データのグラフ化: RRDtool

この世界には計測可能なデータが数多く存在しています。たとえば温度の変化やお使いのコンピュータのネットワークインターフェイスが送受信したデータ量などがあります。RRDtool ではそのようなデータを保管し、カスタマイズ可能なグラフとして視覚化することができます。

RRDtoolは多くのUNIXプラットフォームやLinuxディストリビューションに含まれています。もちろんopenSUSE®でも同様です。YaSTからインストールを行なうか、rootになった状態で、下記のようなコマンドラインを入力してインストールしてください。

```
zypper install rrdtool
```

ティップ

それぞれ RRDtool には Perl, Python, Ruby, PHP の各バインディングが存在しています。これによりお好きな言語で監視用のスクリプトを作成することができます。

13.11.1 RRDtool の動作

RRDtool は *Round Robin Database tool* (ラウンドロビン (循環) 型データベースツール) の省略形で、一定量のデータを取り扱うための仕組みです。データ行に始まりも終わりも存在しないような循環型のバッファを使用してデータを読み込みます。これをラウンドロビンデータベースと呼んでいます。

上述のとおり、RRDtool は時間が経過するごとに変化するようなデータを扱うよう設計されています。理想的には一定間隔でデータ (温度や速度など) を計測するようなセンサーを想定していて、このデータを一定の書式で出力する設計になっています。このようなデータであれば RRDtool が便利に利用でき、期待通りの出力を作成することができます。

場合によっては自動的かつ一定間隔で採取できないようなデータもあります。このような場合は RRDtool に与える前に事前の処理が必要で、条件によっては RRDtool を手動で動作させなければならない場合もあります。

下記では RRDtool の基本的な使い方を示しています。一般的な RRDtool の作業であるデータベースの作成と計測値による更新、および出力の閲覧をそれぞれ行なっています。

13.11.2 シンプルな実例

ここでは、Linux システムにおけるメモリ使用量について時間による変化を収集して閲覧する仕組みを構築したい場合を想定します。より詳しく書くと、この例ではその時点での空きメモリ量を 4 秒おきに 40 秒間採取しています。この計測中、大量のシステムメモリを使用するプログラムが 3 つほど起動および終了しています。具体的には Firefox Web ブラウザと Evolution 電子メールクライアント、および Eclipse 開発フレームワークです。

データの収集

RRDtool はネットワークトラフィックの計測や可視化を行なう場合にもよく利用されています。このような場合は Simple Network Management Protocol (SNMP) を利用して行ないます。このプロトコルではそれぞれのネットワークデバイスに対して内部カウンタの値を問い合わせることができるため、RRDtool を利用してこれらの値を保存することができます。SNMP については <http://www.net-snmp.org/> をお読みください。

今回の例は上記とは異なり、データを手動で採取する必要があります。下記にある free_mem.sh 支援スクリプトがメモリの空き容量について現在の状態を読み取り、標準出力に書き込む作業を行ないます。

```
tux@mercury:~> cat free_mem.sh
INTERVAL=4
for steps in {1..10}
do
    DATE=`date +%s`
    FREEMEM=`free -b | grep "Mem" | awk '{ print $4 }'`
    sleep $INTERVAL
    echo "rrdtool update free_mem.rrd $DATE:$FREEMEM"
done
```

注目点

- 時間間隔が 4 秒に設定されていますが、これは sleep コマンドで実現しています。

- **RRDtool** は特殊な書式で時刻情報を受け入れます。具体的には *Unix 時刻* と呼ばれるもので、1970 年 1 月 1 日の深夜 00:00 からの経過秒数を表わしているものです。たとえば 1272907114 は 2010-05-03 17:18:34 を意味しています。
- メモリの空き容量情報は `free -b` を利用することでバイト単位で報告されます。ここではキロバイトのような おおまかな単位ではなく、バイト単位で正確な値を採取しています。
- `echo ...` のコマンドには、データベースファイル (`free_mem.rrd`) が指定されていて、ここから **RRDtool** のデータベースを更新するためのコマンドラインが書かれています。

`free_mem.sh` を実行すると、下記のような出力が現われます:

```
tux@mercury:~> sh free_mem.sh
rrdtool update free_mem.rrd 1272974835:1182994432
rrdtool update free_mem.rrd 1272974839:1162817536
rrdtool update free_mem.rrd 1272974843:1096269824
rrdtool update free_mem.rrd 1272974847:1034219520
rrdtool update free_mem.rrd 1272974851:909438976
rrdtool update free_mem.rrd 1272974855:832454656
rrdtool update free_mem.rrd 1272974859:829120512
rrdtool update free_mem.rrd 1272974863:1180377088
rrdtool update free_mem.rrd 1272974867:1179369472
rrdtool update free_mem.rrd 1272974871:1181806592
```

この出力はファイルに転送しておくのが便利です。下記のようにして行ないます。

```
sh free_mem.sh > free_mem_updates.log
```

データベースの作成

今回の設定で初期のラウンドロビンデータベースを作成するには、下記のコマンドを入力します:

```
rrdtool create free_mem.rrd --start 1272974834 --step=4 \
DS:memory:GAUGE:600:U:U RRA:AVERAGE:0.5:1:24
```

注目点

- このコマンドは `free_mem.rrd` という名前の ファイルを作成します。このファイルはラウンドロビン (循環) 型のデータベース で、計測値を保管するために使用するものです。
- `--start` オプションでは、データベースに追加されるデータのうち、最初の時間を **Unix 時間形式** で指定しています。この例では、`free_mem.sh` の最初の値 (1272974835) よりも小さい値を設定しています。
- `--step` では、データベースに対してデータを提供する間隔 を秒単位で指定しています。
- `DS:memory:GAUGE:600:U:U` の部分はデータベースの データソースを指定しています。 *memory* という名称で *gauge* というタイプのもので、2つの更新の最長間隔は 600 秒に設定しています。計測値の 最小 と 最大 はそれぞれ未知の値 (U) に設定しています。
- `RRA:AVERAGE:0.5:1:24` はラウンドロビンアーカイブ (RRA) を作成するもので、**統合機能(CF)** を利用してデータポイントの **平均値** を計算し、保管を行ないます。統合機能の3つのパラメータは、行の終わりに追加します。

何もエラーメッセージが表示されなければ、カレントディレクトリに `free_mem.rrd` データベースが作成されています:

```
tux@mercury:~> ls -l free_mem.rrd
-rw-r--r-- 1 tux users 776 May  5 12:50 free_mem.rrd
```

データベースの値の更新

データベースを作成したら、あとは計測データを書き込む作業です。「データの収集」(232 ページ) では既に `rrdtool update` コマンドを実行するためのファイル `free_mem_updates.log` を用意してあるので、これらのコマンドを利用してデータベースの値を更新することができます。

```
tux@mercury:~> sh free_mem_updates.log; ls -l free_mem.rrd
-rw-r--r-- 1 tux users 776 May  5 13:29 free_mem.rrd
```

上記のとおり、`free_mem.rrd` のサイズは更新前後で 変化することはありません。

計測値の閲覧

値を測定してデータベースを作成し、計測値を保管したら、そのデータベースを利用してデータを取り出したり、表示したりすることができるようになります。

データベースから全ての値を取り出すには、下記のコマンドを入力します:

```
tux@mercury:~> rrdtool fetch free_mem.rrd AVERAGE --start 1272974830 \  
--end 1272974871  
memory  
1272974832: nan  
1272974836: 1.1729059840e+09  
1272974840: 1.1461806080e+09  
1272974844: 1.0807572480e+09  
1272974848: 1.0030243840e+09  
1272974852: 8.9019289600e+08  
1272974856: 8.3162112000e+08  
1272974860: 9.1693465600e+08  
1272974864: 1.1801251840e+09  
1272974868: 1.1799787520e+09  
1272974872: nan
```

注目点

- AVERAGEはデータベースから平均値を取り出すためのキーワードです。これはデータソースをAVERAGE 1つしか定義していない(「データベースの作成」(233 ページ)) ため、その他の機能は利用できません。
- 出力の1行目にはデータソースの名前が出力されています。これは「データベースの作成」(233 ページ) で設定したものです。
- 列の左側にはそれぞれ個別のポイントにおける時刻が表示されています。右側には科学表記での計測値の平均が表示されています。
- 最後の行にある nan は「not a number」(数値ではないもの) の意味です。データが存在しない場合などに出力されます。

あとはデータベース内にある値を利用してグラフを作成することができます:

```
tux@mercury:~> rrdtool graph free_mem.png \  
--start 1272974830 \  
--end 1272974871 \  
--step=4 \  
DEF:free_memory=free_mem.rrd:memory:AVERAGE \  
LINE2:free_memory#FF0000 \  

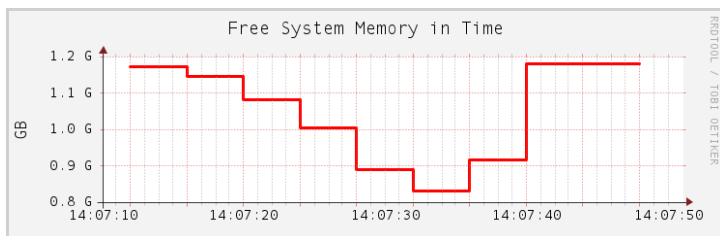
```

```
--vertical-label "GB" \  
--title "Free System Memory in Time" \  
--zoom 1.5 \  
--x-grid SECOND:1:SECOND:4:SECOND:10:0:%X
```

注目点

- `free_mem.png` は作成するグラフのファイル名を指定しています。
- `--start` と `--end` はグラフ内に描画すべき時間範囲を指定しています。
- `--step` はグラフの時間間隔を秒単位で指定しています。
- `DEF:...` の部分は *free_memory* のデータ定義です。データは `free_mem.rrd` ファイルから読み込みを行ない、*memory* という名前を設定します。また、*average* (平均値) を計算して出力します。これは、「データベースの作成」(233 ページ) で設定したとおり、それ以外の値を定義していないためです。
- `LINE...` の部分はグラフ内に描画する線の属性を設定しています。2 ピクセル幅で *free_memory* で定義したデータを利用し、赤色で描画します。
- `--vertical-label` では y 軸に表示するラベルを指定しています。また、`--title` ではグラフ全体に対するラベル(つまりタイトル)を指定しています。
- `--zoom` はグラフの拡大率を指定しています。この値はゼロより大きい値でなければなりません。
- `--x-grid` ではグラフ内におけるグリッド線とラベルの描画方法を指定しています。この例では 4 秒ごとにグリッド線を出力しています。ラベルは各グリッド線に対して 10 秒ごとに配置しています。

図 13.2 RRDtool で作成したグラフ例



13.11.3 さらなる情報

RRDtool には数多くのサブコマンドとコマンドラインオプションが存在し、非常に複雑なツールになっています。これらのうちのいくつかは理解しやすいものですが、要件に応じた出力や好みに沿った指定を行なうには、学習をしなければ習得できません。

基本的な情報だけを提供している RRDtool のマニュアルページ (man 1 rrdtool) 以外に、RRDtool homepage [<http://oss.oetiker.ch/rrdtool/>] (英語) についてもお読みになることをお勧めします。ここには rrdtool の詳細な文書 [<http://oss.oetiker.ch/rrdtool/doc/index.en.html>] が置かれていて、コマンドとサブコマンドの全てが収録されています。また、一般的な RRDtool の作業手順を示しているチュートリアル [<http://oss.oetiker.ch/rrdtool/tut/index.en.html>] もお勧めです。

また、ネットワークトラフィックの監視を行ないたい場合は、MRTG [<http://oss.oetiker.ch/mrtg/>] (英語) もお読みください。このツールは Multi Router Traffic Grapher の意味で、ネットワークデバイスに対する全ての動作状況をグラフ化することができる RRDtool よりも使いやすいツールです。

システムのアップグレードとシステム変更

14

既存のシステムをアップグレードするのに、再インストールを行なう必要はありません。2種類の方法でシステム全体や一部分を更新することができます。1つは **個別のソフトウェアパッケージを更新する方法**、もう1つは **システム全体のアップグレード**です。個別のパッケージの更新方法については第3章 **ソフトウェアのインストールと削除** (67 ページ) と第4章 **YaST オンライン更新** (89 ページ) で説明しています。また、システムをアップグレードするための2つの方法については、14.1.3項「**YaST を利用したアップグレード**」(241 ページ)と14.1.4項「**zypper を利用したディストリビューションアップグレード**」(243 ページ)で説明していますので、そちらをお読みください。

14.1 システムのアップグレード

ソフトウェアはバージョンからバージョンに上がることでサイズが「成長」していきます。そのため、更新処理を行なう前には`df`コマンドを利用し、空き容量を確認してください。ディスク容量が足りなそうな場合は、更新処理の前にお使いのデータを別の場所に保存し、お使いのシステムのパーティション設定を修正してください。各パーティションにどれだけのディスク容量が必要となるかは、パーティション 設定や選択したソフトウェア、システムのバージョンに大きく依存するため、一般的に「これ」といった値はありません。

14.1.1 準備

アップグレード処理の前に、まずはお使いの環境を保持しておくため、古い設定ファイルを別途のメディア (テープデバイスやリムーバブルハードディスク、USB フラッシュメモリなど) にコピーしてください。主に /etc や /var 内に存在するファイルと、いくつかのサブディレクトリをコピーしてください。また、/home ディレクトリ (ホームディレクトリ) 以下にあるユーザデータについても、バックアップメディアに書き込んでおくことをお勧めします。なお、バックアップ作業は root で実施してください。全てのローカルファイルを読み込むには、root の権限が必要です。

さらに更新処理を開始する前に、ルートパーティションについても確認を行なってください。df / コマンドを実行すると、ルートパーティションのデバイス名が表示されます。例14.1「df -h コマンドでの出力例」(240 ページ)にある例では、ルートパーティションは /dev/sda3 (/ にマウントされているものです)。

例 14.1 df -h コマンドでの出力例

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	74G	22G	53G	29%	/
udev	252M	124K	252M	1%	/dev
/dev/sda5	116G	5.8G	111G	5%	/home
/dev/sda1	39G	1.6G	37G	4%	/windows/C
/dev/sda2	4.6G	2.6G	2.1G	57%	/windows/D

14.1.2 発生しうる問題

既定の手順でインストールを行なったシステムを従来のバージョンから新しいバージョンにアップグレードする場合、YaST は特に問題を起こすことなく、必要な変更を見積もって実施します。お使いのシステムをどれだけカスタマイズしたのかによって、いくつかの手順 (またはアップグレード処理の全体) が失敗することがあるため、場合によっては事前に採取しておいたバックアップデータを書き戻さなければならないかもしれません。システムのアップグレードを始める前に、下記の問題点をご確認ください。

/etc 内の passwd ファイルと group ファイルの確認

システムをアップグレードする前に、/etc/passwd ファイルや /etc/group ファイルに文法エラーがないことを確認しておいてください。root から pwck や grpck などの検証 ツールを実行することで、問題点を調査することができます。

PostgreSQL

PostgreSQL (postgres パッケージ) を更新する前に、データベースの ダンプを採取してください。具体的な手順については、pg_dump のマニュアル ページをお読みください。この作業は、お使いの環境を更新する 際に、データを引き継ぐ必要がある場合にのみ必要です。

14.1.3 YaST を利用したアップグレード

14.1.1 項「準備」(240 ページ)で示した準備手順を実施し終わると、ようやくお使いのシステムをアップグレードすることができます:

- 1 まずは 項「インストール向けのシステムスタートアップ」(第1章 *YaST* を利用したインストール, ↑スタートアップ) に書かれている手順に従い、インストール用のシステムを起動します。YaST が起動したら 言語を選択し、インストールモードのダイアログで 更新を選択します。新規インストールは選択しないでください。また、全ての利用可能なソフトウェアを更新するため、追加のリポジトリについても設定を行なってください。インストールを行なう際のリポジトリについて、詳しくは 項「アドオン製品」(第1章 *YaST* を利用したインストール, ↑スタートアップ)をお読みください。
- 2 次に YaST は、複数のルートパーティションが存在しないかどうかを確認します。1 つだけしか存在しなかった場合は、そのまま次の手順に進みます。複数のパーティションが見つかった場合は、アップグレードしたいルートパーティションを選択し、次へを押します(14.1.1 項「準備」(240 ページ)の例では、/dev/sda3 を選択しています)。YaST は、そのパーティション内にある fstab ファイルを読み込んで分析し、記載されているパーティションをマウントします。

警告: 永続性のあるデバイス名

/etc/fstab 内でカーネルデバイス名を利用してパーティションをマウントするように設定している場合、アップグレードを行なう前に対応する他の指定方法に変更しておかなければなりません。カーネルデバイス名は永続するものではないため、アップグレード処理時に問題が発生する場合があります。このような場合は YaST パーティション設定モジュールを利用し、マウント方法を変更してください。

- 3 何らかのリポジトリが設定されていた場合、それらの存在を確認します。それぞれアップグレード後も使用し続けたいリポジトリや、サードパーティ製ソフトウェアの更新元を選択します。必要に応じて **状態の切り替え** を押し、切り替えてください。
- 4 上述のとおりアップグレード処理時にリポジトリを追加した場合は、ここで実際に有効にするかどうかを設定することができます。
- 5 インストール設定ダイアログでは、ご希望の要件にあわせて設定を変更することができます。通常は既定の設定のままで何も変更する必要はありませんが、お使いのシステムを拡張させたい場合は、**パッケージ** や **更新オプション** のサブメニュー内からパッケージを選択するか、追加の言語向けの対応を追加することができます。

また、お使いのシステムをバックアップすることもできます。バックアップを選択すると、アップグレード処理に時間がかかるようになりますので、このオプションは直近にシステムバックアップを採取していない場合にのみ、選択してください。

- 6 アップグレード内容を確認したら、**更新開始** を押すとアップグレード処理が始まります。

システムの基礎部分についてアップグレード処理が完了すると、YaST はシステムを再起動します。その後 YaST は残りのソフトウェアについて更新を行いません。必要であればリリースノートを表示させることもできます。

14.1.4 zypper を利用したディストリビューションアップグレード

zypper コマンドラインユーティリティを利用することで、お使いの環境を次のバージョンにアップグレードすることができます。このアップグレードで最も重要な点としては、システムを稼働中の状態からアップグレード作業を行なうことができるという点があります。

この機能は、遠隔からのアップグレードを行なったり、もしくは多数の似通ったシステムを一括でアップグレードしたりすることを希望する知識のあるユーザにとっては、魅力的な仕組みです。それほど経験の無い方であれば、14.1.3 項「YaST を利用したアップグレード」(241 ページ)に示されている手順で起動メディアから YaST を起動し、アップグレードを行なったほうが便利でしょう。

zypper によるアップグレードを開始する前に

zypper を利用してアップグレード作業を行なうにあたり、予期しないエラーを回避するには、リスク配置を最小化する必要があります。

以前のバージョン (たとえば 11.2) から本バージョン (たとえば 11.3) にアップグレードを行なってください。アップグレード時にはマイナーバージョンを飛ばす (たとえば 11.1 またはそれ以前から 11.3 に一括アップグレードする) ことは避けてください。また、アップグレード元のバージョンでは、利用可能なすべてのオンライン更新を正しく適用しておいてください。

できる限り多くのアプリケーションと不要なサービスについてはできる限り停止し、一般ユーザについてもログアウトしてください。

アップグレードにあたってはサードパーティ製や openSUSE ビルドサービスのリポジトリを無効化するか、もしくは優先度を下げて既定のシステムリポジトリからのパッケージが優先的にインストールされるように設定してください。アップグレード後はこれらのリポジトリを再度有効化し、アップグレード先のバージョン番号に適合するバージョン文字列に変更してください。

詳しくは http://en.opensuse.org/SDB:System_upgrade (日本語) または http://en.opensuse.org/SDB:System_upgrade (英語) をお読みください。

アップグレード手順

警告: システムのバックアップ採取

実際にアップグレード作業を行なう前に、お使いのシステムに対する最新のバックアップが採取されていて、かつ復元可能であることをご確認ください。これは特に下記の手順のうちの多くを手作業で実施しなければならないため、特に重要です。

- 1 オンライン更新を実行し、ソフトウェア管理スタックが最新の状態であることを確認します。詳しくは第4章 *YaST* オンライン更新 (89 ページ) をお読みください。
- 2 更新ソースとして利用したいリポジトリを設定します。ここで正しい設定を行なっておくことが重要です。YaST を利用して設定する (3.4項「ソフトウェアリポジトリとサービスの管理」 (83 ページ) をお読みください) か、zypper を利用して設定 (7.1項「zypper の使用」 (105 ページ) をお読みください) します。

ティップ: リポジトリ名

下記に示す手順内で使用されるリポジトリの名前は、お使いのシステムの設定に依存して少しずつ異なることにご注意ください。

なお、現在設定されているリポジトリについて一覧を確認するには、下記のように入力します:

```
zypper lr -u
```

- 2a** システムリポジトリのバージョンを以前のものからアップグレード先のバージョンに設定します。たとえば下記のように実行します:

```
server=http://download.opensuse.org
new_ver=11.3
zypper ar $server/distribution/$new_ver/repo/oss/
openSUSE-$new_ver-Oss
zypper ar $server/update/$new_ver/ openSUSE-$new_ver-Update
```

続いて古いリポジトリを削除します:

```
old_ver=11.2
```

```
zypper rr openSUSE-$old_ver-Oss
zypper rr openSUSE-$old_ver-Update
```

- 2b** `zypper dup` コマンドは既定のリポジトリに対してのみ正しく動作することが保証されている仕組みであるため、まずはサードパーティ製のリポジトリやその他の `openSUSE` ビルドサービスのリポジトリを無効に設定します (ここで、`repo-alias` は無効に設定したいリポジトリの名前を指定します):

```
zypper mr -d repo-alias
```

これらのリポジトリを無効化する代わりに、優先順位を下げてもらえません。

注記: 未解決の依存関係に関する処理について

`zypper dup` コマンドでは未解決の依存関係を持つパッケージはすべて削除するようになっていますが、依存関係が満たされている限り、無効化されたリポジトリのパッケージについてはそのまま保持されます。

`zypper dup` コマンドにより、すべてのインストール済みパッケージが利用可能なリポジトリのいずれかを利用して更新されます。この場合はインストール済みのパッケージのバージョン番号やアーキテクチャ、製造元については考慮されません。そのため、何もない状態からインストールする場合と同じ結果になります。また、リポジトリ内にもはや存在していないパッケージは孤立したものと判断され、そのようなパッケージは依存関係が満たされない場合は削除されます。依存関係に問題がない場合は、そのようなパッケージはインストールされたままになります。

- 2c** 完了したら、下記のコマンドでリポジトリの設定を確認します:

```
zypper lr -d
```

- 3** 続いてローカルのメタデータとリポジトリの内容を更新するため、`zypper ref` コマンドを実行します。
- 4** メタデータを更新したら、まずは `zypper in zypper` と入力し、`zypper` 自身のアップグレードを行ないます。

- 5 `zypper` を更新したら、`zypper dup` と入力し、ディストリビューション本体のアップグレードを行ないます。このとき、ライセンスの確認が求められます。
- 6 アップグレードが完了したら `SuSEconfig` と入力し、基本的なシステム設定を実施します。
- 7 `shutdown -r now` と入力し、システムを再起動します。

14.1.5 個別のパッケージ更新

システム全体をアップグレードするかどうかとは別に、個別のパッケージについてもアップグレードを行なうことができます。ただし、このように個別のアップグレードを行なう場合、システムの安定性や安全性については、ご自身の責任の下で管理してください。 <http://www.novell.com/linux/download/updates/> に書かれている 更新時のアドバイスをお読みください。

YaST のパッケージ選択から、必要に応じてパッケージを選択してください。システム全体の動作に影響を与えるようなパッケージであった場合、YaST は警告を発します。そのようなパッケージは更新モードでのみ更新されるべきものであるためです。たとえば多くのパッケージには共有ライブラリが含まれていて、それらのプログラムやアプリケーションを移動中の状態から更新を行なうと、システムを不安定にしてしまう可能性があります。

14.2 ソフトウェアの変更点

本章では、バージョン間の個別の修正について、概要を示しています。この概要にはたとえば、基本的な設定を完全にやり直すかどうかの情報や設定ファイルの場所変更に関する情報、もしくは一般的に使用されるアプリケーションが大幅に変更された ことに関する情報が含まれます。システムを日々使っていくにあたっての大幅な 変更は、ユーザレベルと管理者レベルの両方が記述されています。

様々なバージョンにおける特別な問題については、それらが判明次第オンラインでお知らせする仕組みになっています。個別のパッケージに関する重要な更新については、YaST オンライン更新を利用して <http://www.novell>

[.com/products/linuxprofessional/downloads/](http://www.opensuse.com/products/linuxprofessional/downloads/) からアクセスすることができます。また、さらなる情報は 第4章 *YaST* オンライン更新 (89 ページ) に書かれています。

また、Product highlights (http://wiki.opensuse.org/Product_highlights) と openSUSE wiki 内の Bugs 記事 (http://en.opensuse.org/openSUSE:Most_annoying_bugs) をお読みください。こちらに直近の変更点と問題点が記されています。

パート IV. システム

64 ビット環境における 32 ビットおよび 64 ビットアプリケーション

15

openSUSE® は 64 ビット環境でもご利用いただけます。これは、必ずしも全てのアプリケーションが 64 ビットプラットフォームに移植されたことを示しているわけではありません。また、openSUSE は 64 ビット環境下での 32 ビットアプリケーションの使用についてもサポートしています。この章では、64 ビットの openSUSE プラットフォームがどの程度サポートされているのかについて、概要を示しています。また、どのようにして 32 ビットアプリケーションが実行されるのか(ランタイムサポート)、およびどのようにすれば 32 ビットアプリケーションは、32 ビットおよび 64 ビット環境の両方で動作するようコンパイルすべきなのかについても述べています。さらに、64 ビットカーネルで 32 ビットアプリケーションを動作させるにあたってのカーネル API や、その他の説明などについても記しています。

また、64 ビット環境である amd64 と Intel 64 向けの openSUSE は、既存の 32 ビットアプリケーションを「そのまま」利用することができます。これは、32 ビットアプリケーションが 64 ビット対応になるのを待つことなく、お使いのアプリケーションを使い続けられることを意味しています。

15.1 ランタイムサポート

重要項目: アプリケーションバージョン間の矛盾

あるアプリケーションについて、32 ビットと 64 ビットの両方の版が用意されている場合、両方の版を同時にインストールしてしまうと、問題を引

き起こしてしまいます。この場合、2つの版のどちらをインストールするかを決めておき、いずれか片方だけをインストールしてください。

唯一の例外は PAM (pluggable authentication modules) です。openSUSE では、ユーザとアプリケーションの間を調整する認証プロセスとして PAM を使用しています。32 ビットアプリケーションも動作する 64 ビットオペレーティングシステムでは、常に両方の版の PAM モジュールをインストールしておく必要があります。

正しく動作させるため、それぞれのアプリケーションは様々なライブラリを必要とします。これらライブラリの 32 ビット版と 64 ビット版は、いずれも同じ名前が設定されています。それぞれは間違えないよう、何らかの方法で区別しなければなりません。

32 ビットバージョンとの互換性を確保するため、従来の (32 ビット版の) ライブラリは 32 ビット環境の場合と同じ場所に保存されます。32 ビット版の `libc.so.6` は、32 ビット環境でも 64 ビット環境でも `/lib/libc.so.6` に配置されています。

また、全ての 64 ビット版ライブラリとオブジェクトファイルは、`lib64` と呼ばれる場所に配置されます。従来は `/lib` や `/usr/lib` に配置されていたオブジェクトファイルの 64 ビット版は、それぞれ `/lib64` や `/usr/lib64` に配置されるようになっていきます。逆に言うと、`/lib` や `/usr/lib` は 32 ビット版のライブラリを配置する領域として機能することになるため、いずれのバージョンであっても変更することなくご利用いただけることになります。

ただし、32 ビット版の `/lib` 配下のサブディレクトリに配置していた、ビット数に依存しないデータコンテンツについては、従来通りの配置になっています。この方式は LSB (Linux Standards Base) と FHS (File System Hierarchy Standard) に準拠しています。

15.2 ソフトウェア開発

両対応の開発ツールチェーンを利用すると、32 ビットと 64 ビットの両方のオブジェクトを生成することができます。既定では 64 ビットオブジェクトを生成しますが、32 ビットオブジェクトを生成する際には、特殊なフラグを指定する必要があります。gcc の場合、`-m32` を指定します。

また、全てのヘッダファイルはアーキテクチャに依存しない形式で記述しておく必要があります。インストール済みの 32 ビットおよび 64 ビットライブラリには、インストール済みのヘッダファイルに対応した API (アプリケーションプログラミングインターフェイス) が含まれていなければなりません。通常の openSUSE 環境は、この方針に則った形になっています。ライブラリを手動で更新するような場合は、これらの問題をご自身で解決する必要があります。

15.3 両プラットフォーム対応のソフトウェアコンパイル

両方のアーキテクチャに対応したバイナリを開発するには、それぞれ関連するライブラリに対するもう一つのアーキテクチャ版をインストールしておかなければなりません。これらのパッケージは (RPM 名)-32bit のような名前になっています。また、(RPM 名)-devel のような開発パッケージであれば、(RPM 名)-devel-32bit のような名前で 32 ビット版の開発パッケージも提供されています。

多くのオープンソースプログラムでは、autoconf をベースにしたプログラム設定を行なっています。もう 1 つの (たとえば 32 ビット) 版に対応するプログラムをコンパイルするよう autoconf で設定を行なうには、configure スクリプトを実行する際に通常のコンパイラおよびリンカーの設定を上書きし、追加の環境変数を設定してください。

下記の例では、x86_64 システムで x86 をもう 1 つのアーキテクチャとして使用しています。

1 32 ビットコンパイラを使用するには:

```
CC="gcc -m32"
```

2 32 ビットオブジェクトを処理するようリンカーに指示するには (gcc をリンカーのフロントエンドとして使用する場合):

```
LD="gcc -m32"
```

3 32 ビットオブジェクトを生成するようアセンブラに指示するには:

```
AS="gcc -c -m32"
```

- 4** 32ビット版のライブラリの場所を指定するなど、リンカのフラグを設定するには:

```
LDFLAGS="-L/usr/lib"
```

- 5** 32ビット版のオブジェクトコードライブラリの場所を指定するには:

```
--libdir=/usr/lib
```

- 6** 32ビット版のXライブラリの場所を指定するには:

```
--x-libraries=/usr/lib
```

必ずしも全ての変数指定がそれぞれのアプリケーションに必要というわけではありません。それぞれプログラムの要件に従って指定してください。

```
CC="gcc -m32"  
LDFLAGS="-L/usr/lib;"  
./configure --prefix=/usr --libdir=/usr/lib --x-libraries=/usr/lib  
make  
make install
```

15.4 カーネル仕様

x86_64向けの64ビットカーネルでは、64ビットと32ビットの両方のカーネルABI(アプリケーションバイナリインターフェイス)を提供しています。32ビット版のABIは、32ビットカーネルが提供するものと同じものです。これにより、32ビットアプリケーションは32ビットカーネルの場合と同様に、64ビットカーネルと対話することができる、ということになります。

64ビットカーネルが提供するシステムコールの32ビット版には、全てのAPIが提供されているわけではありません。対応可否はプラットフォームによって異なります。このため、たとえばlspciなどの小数のアプリケーションについては、コンパイルし直す必要があります。

また64ビットのカーネルは、そのカーネル向けにコンパイルされた64ビットのカーネルモジュールのみを読み込むことができます。32ビット版のカーネルモジュールを読み込むことはできません。

ティップ: カーネルの読み込みモジュール

アプリケーションによっては、個別のカーネルモジュールを必要とする場合があります。このような 32 ビットアプリケーションを 64 ビットシステムの環境で使いたい場合は、そのアプリケーションの製造元と Novell に対して 64 ビット版のカーネルモジュールを提供していないかどうか、および 32 ビット版に対応したカーネル API が存在しているかどうか、それぞれお尋ねください。

Linux システムの起動と設定

Linuxシステムの起動には、様々なコンポーネントが関わっています。ハードウェア それ自身は BIOS で初期化され、その後にブートローダからカーネルを読み込みはじめます。この処理の後initで起動処理が始まり、オペレーティングシステムからランレベルが操作されるようになります。ランレベルの考え方により、日々の使用形態に沿った設定とともに、システムメンテナンス作業を行なうこともできるようになっています。

16.1 Linux の起動処理

Linux の起動処理は、それぞれ異なるコンポーネントで提供される複数のステージから構成されています。下記の一覧では起動処理の概要と、利用されるコンポーネントのうちよく知られたものについて述べています。

1. **BIOS** コンピュータの電源を入れると、BIOS は画面とキーボードを初期化し、メインメモリのテストを行ないます。この時点では、まだマストレージメディア(ハードディスクなど)にはアクセスを行ないません。また、現在の日時や重要な周辺機器に関する情報を、CMOS から読み込みます。1 台目のハードディスクとそのジオメトリ情報を読み込むと、システム処理は BIOS からブートローダに渡されます。
2. **ブートローダ** 1 台目のハードディスクにある冒頭の 512 バイトがメインメモリに読み込まれ、そのセクタの冒頭にあるブートローダの処理が始まります。ブートローダで実行するコマンドは、残りの起動処理を行なうための作業になります。そのため、1 台目のハードディスクにある冒頭の 512 バイトは、マスターブートレコード(MBR)と呼ばれます。ブートローダ

はその後、実際のオペレーティングシステム(この場合はLinuxカーネル)に処理を移します。LinuxのブートローダであるGRUBについて、詳しくは第17章 ブートローダGRUB(275 ページ)をお読みください。

3. **カーネルとinitramfs** システムの制御権を渡す目的で、ブートローダはカーネルとRAMベースの初期ファイルシステム(initramfs)をメモリ内に読み込みます。initramfsの内容はカーネルから直接利用できる形式になっています。また、initramfsには**init**と呼ばれる小さな実行ファイルが含まれ、実際のルートファイルシステムをマウントするまでの処理を行ないます。マスタストレージ(ハードディスクなど)にアクセスする際に特別なハードウェアドライバが必要な場合、それらのドライバはinitramfs内に存在していなければなりません。initramfsについて、詳しくは16.1.1項「initramfs」(258 ページ)をお読みください。
4. **initramfs 内のinit** このプログラムは正しいルートファイルシステムをマウントするために必要な処理を全て行なうもので、必要なファイルシステム向けのカーネル機能を提供したり、udevを利用してマスタストレージ(ハードディスク)コントローラを読み込む機能を提供したりしています。ルートファイルシステムが検出されると、エラーがないかどうかをチェックして、マウントを行ないます。マウントが成功するとinitramfsは解放され、ルートファイルシステム内のinitを実行します。initについて、さらに詳しい情報は16.1.2項「initramfs 上のinit」(260 ページ)をお読みください。また、udevについて詳しくは第19章 udevによる動的なカーネルデバイス管理(317 ページ)をお読みください。
5. **init** initは、それぞれ異なる機能を持つ複数のレベルに対して、それぞれ異なる起動処理を行ないます。initについては16.2項「initプロセス」(261 ページ)をお読みください。

16.1.1 initramfs

initramfsは小さなcpio形式のアーカイブで、カーネルはこのアーカイブをRAM内に読み込むことができます。このアーカイブは、実際のルートファイルシステムがマウントされるまでに必要な、最小限のLinux環境です。この最小限のLinux環境はBIOSルーチンからメモリ内に読み込まれるもので、十分なメモリサイズが存在すること以外に、ハードウェアに対する要件がありません。また、initramfsには実際のルートファイルシステム内のinitブ

ログラムを呼び出すことのできる、`init` と呼ばれる実行形式が存在しなければなりません。

ルートファイルシステムがマウントできるようになり、オペレーティングシステムが起動できる状態になるよりも前の段階で、カーネルはルートファイルシステムが配置されているデバイスにアクセスするため、必要なドライバを読み込む必要があります。これらのドライバは、ハードディスクドライブの種類に対応した特別なドライバか、もしくはネットワークファイルシステムにアクセスするためのネットワークドライバの形式になっています。つまり、ルートファイルシステムにアクセスするのに必要なモジュールは `initramfs` 内の `init` によって読み込まれることになります。モジュールを読み込んだ後は、`udev` が `initramfs` とそれに必要なデバイスを提供します。ルートファイルシステムへの変更が完了した後は、デバイスを再生成する必要があります。この処理は `boot.udev` が実施するもので、`udevtrigger` のコマンドを利用しています。

インストール済みのシステムに対してハードウェア(たとえばハードディスク)の変更を行なう必要があります、カーネルが起動する際に今とは異なるドライバを読み込む必要がある場合は、`initramfs` を更新しなければなりません。この作業は `initrd` を最初に作成したときと同じ方法、つまり `mkinitrd` を呼び出すことで行なうことができます。なにもパラメータをつけずに `mkinitrd` を実行すると、`initramfs` を生成します。`mkinitrd -R` と入力すると `initrd` を生成します。**openSUSE®** では、読み出すべきモジュールの一覧を `/etc/sysconfig/kernel` ファイル内の `INITRD_MODULES` で指定します。インストールが完了すると、上記の変数は自動的に正しい値に設定されます。各モジュールは `INITRD_MODULES` 内に記述した順に読み込まれます。この順序指定は `/dev/sd?` のようなデバイス名の命名順序に影響するため、デバイス名に依存して何らかの作業を行なう場合には重要な項目です。しかしながら現在のシステムでは、`by-id` や `by-path` などの複数のサブディレクトリから同じディスクを指し示すことのできる、`/dev/disk/` 形式のデバイスファイルを利用することができます。これらはインストール時に、関連するオプションを指定することで変更することができます。

重要項目: `initramfs` や `initrd` の更新

ブートローダは `initramfs` や `initrd` をカーネルと同じ方法で読み込みます。`initramfs` や `initrd` を更新した場合でも、**GRUB** を再インストール

ルする必要はありません。これは **GRUB** が起動する際に正しいファイルを
検索する仕組みを 備えているためです。

16.1.2 initramfs 上の init

initramfs 内にある **init** は、実際のルートファイルシステムをマウントするための準備作業と、実際のルートファイルシステムへのアクセス作業を主な目的としています。お使いのシステム設定によって、**init** はそれぞれ下記のような作業を行ないます。

カーネルモジュールの読み込み

お使いのハードウェア設定によって、お使いのコンピュータのハードウェアにアクセスするためのドライバが必要となります(最も重要なコンポーネントはハードディスクです)。また、最終的なルートファイルシステムにアクセスするため、カーネルでは適切なファイルシステムドライバも必要です。

ブロックスPECIALファイルの提供

それぞれ読み込んだカーネルモジュールでは、デバイスイベントを生成します。**udev** はこれらのイベントを処理し、必要なブロックスPECIALデバイスを **/dev** 内にある **RAM** ファイルシステム内に作成します。これらのSPECIALファイルが存在しないと、ファイルシステムやその他のデバイスにアクセスすることができません。

RAID と LVM の設定管理

お使いのシステムにおけるルートファイルシステムが **RAID** や **LVM** の管理下にある場合、**LVM** や **RAID** を設定して後にアクセスすることになるルートファイルシステムを読み込むことができるようにします。**RAID** や **LVM** について、詳しくは第2章 高度なディスク設定 (43 ページ) をお読みください。

ネットワーク設定の管理

お使いのシステムでネットワークマウント型のルートファイルシステム(**NFS** を介したマウント)を利用している場合、**init** は適切なネットワークドライバを読み込み、ルートファイルシステムへのアクセス手段を準備しなければなりません。

インストール作業の初期段階で `init` が呼び出された場合は、上述の手順とは異なる 下記の手順が行なわれます:

インストールメディアの検出

インストール処理を開始する際、お使いのマシンはインストールメディア上にある インストール用のカーネルと `YaST` インストーラの存在する `initrd` を読み込みます。 `RAM` ファイルシステム内で動作する `YaST` インストーラは、インストールメディアにアクセスしてインストール作業を行なうため、その場所に関する情報を事前に知っておく必要があります。

ハードウェアの検出と適切なカーネルモジュールの読み込み

16.1.1項「`initramfs`」(258 ページ)に示しているとおり、起動処理は多くのハードウェア環境で動作するドライバを含んだ最低限のドライバセットで行なわれます。`init` はお使いのハードウェア環境での適切なドライバを判断するため、初期のハードウェア検出処理を行ないます。起動処理に必要であると判断したモジュール名は、`/etc/sysconfig/kernel` 内の `INITRD_MODULES` に書き込まれます。モジュール名の一覧は、システムを起動する際に使用するカスタムな `initramfs` の作成に利用します。起動時には必要がないものの、あとから読み出す必要があるモジュールについては、`/etc/sysconfig/hardware/hwconfig-*` 内に書き込まれます。このディレクトリ内の設定ファイル内に記述された全てのデバイスは、起動処理で初期化されます。

インストールシステムやレスキューシステムの読み込み

ハードウェアを正しく検出して適切なドライバが読み込まれると、`udev` はスペシャルデバイスファイルを作成し、`init` は実際の `YaST` インストーラであるインストールシステムを起動するか、レスキューシステムを起動します。

`YaST` の起動

最後に `init` は `YaST` を起動し、パッケージのインストールやシステムの設定を行ないます。

16.2 `init` プロセス

`init` プログラムはプロセス ID が 1 であるプロセスで、必要な方法でシステムを初期化するために利用されます。`init` はカーネルから直接起動され、通常はプロセスを `kill` するために用いられる 9 番目のシグナルを無視して動作し

ます。他のプログラムは全て `init` から起動されるか、`init` から起動されたプログラムの子プロセスとして起動します。

`init` は ランレベル (16.2.1項「ランレベル」 (262 ページ) を参照) を定義する `/etc/inittab` ファイルで設定します。このファイルでは、どのサービスやデーモンがどのランレベルで利用されるのかが記されています。`/etc/inittab` 内に書かれている項目に従っていくつかのスクリプトが `init` から起動されます。既定では起動後に開始される 最初のスクリプトは `/etc/init.d/boot` に設定されています。システムの初期化フェーズが完了すると、システムは `/etc/init.d/rc` スクリプトを利用して既定のランレベルに移行します。`/etc/init.d` ディレクトリ (16.2.2項「初期化スクリプト」 (265 ページ) を参照) 以下にある全てのスクリプトは、よりわかりやすい呼び方として *init* スクリプト と呼ばれています。

システムを起動したりシャットダウンしたりする作業は、`init` が全体を管理します。この観点からすると、カーネルは全ての他のプロセスを管理し、それらのプログラムから要求される CPU 時間とハードウェアアクセスを管理するバックグラウンドプロセスであると考えられます。

16.2.1 ランレベル

Linux では、システムをどのように起動するかと稼働中のシステムでどのサービスを有効化するかについて、ランレベルと呼ばれるものを使用します。起動後、システムは *runlevels* ファイル内で `initdefault` と書かれた行の設定を読み込んで処理を行ないます。通常この値は 3 または 5 に設定されています。詳しくは表 16.1 「利用可能なランレベル」 (262 ページ) をお読みください。また代替手段として、ランレベルを起動時に指定することもできます (たとえば起動 プロンプトでランレベルの数字を追加するなど)。カーネル自身で直接解釈できないパラメータは `init` に渡されますので、ランレベル 3 で起動したい場合は、起動 プロンプトに 3 を記入してください。

表 16.1 利用可能なランレベル

ランレベル	説明
0	システム停止状態を表わします。
S または 1	シングルユーザモードを表わします。

ランレベル	説明
2	リモートネットワーク (NFS など) を設定しない、ローカルマルチユーザモードを表わします。
3	ネットワークを利用する完全マルチユーザモードを表わします。
4	管理者がランレベルを設定しない限り特に利用されることのない、ユーザ定義のランレベルを表わします。
5	ネットワークと X ディスプレイマネージャ (KDM, GDM, XDM など) を起動する、完全マルチユーザモードを表わします。
6	システムの再起動を表わします。

重要項目: NFS でパーティションをマウントしている場合のランレベル 2 について

お使いのシステムで /usr などのファイルシステムを NFS でマウントしている場合は、ランレベル 2 を使用すべきではありません。ランレベル 2 (リモートネットワークを設定しないローカルマルチユーザモード) では NFS サービスを利用することができないため、プログラムファイルやライブラリが見つからず、予期しない動作になってしまいます。

システム稼働中にランレベルを変更するには、telinit コマンドの後ろに移行先のランレベルを指定してください。この作業はシステム管理者 だけが行なうことができます。下記の一覧には、ランレベルの指定に関する重要な コマンドを示しています。

telinit 1 または shutdown now

システムは シングルユーザモードに移行します。このモードは、システムメンテナンスや管理作業を行なう場合に利用します。

telinit 3

必要な全てのプログラムとサービス (ネットワークを含む) が起動し、通常のユーザがログインできるようになります。ただしグラフィカルなシステムは利用できません。

telinit 5

グラフィカルな環境が利用できるようになります。通常は **XDM**, **GDM**, **KDM**などのディスプレイマネージャが起動します。自動ログイン機能が有効に設定されていると、ローカル環境は事前にウインドウマネージャ (**GNOME**, **KDE** やその他の ウインドウマネージャ) で選択しておいたユーザでログインします。

telinit 0 または shutdown -h now
システムが停止します。

telinit 6 または shutdown -r now
システムが停止し、再起動します。

openSUSE の標準インストールでは、ランレベル 5 が既定値に設定されています。グラフィカルなインターフェイスでログインを求められるか、既定のユーザで自動ログインします。

警告: **/etc/inittab** 内のエラーによる起動時障害

/etc/inittabに何らかの問題があると、システムは正しく起動できなくなります。そのため、**/etc/inittab**を編集する際には細心の注意を払ってください。また、マシンを再起動する前に **telinit q** と入力すると、**init** に対して **/etc/inittab** を再読み込みさせることができます。

一般に、ランレベルを変更すると 2 種類の変更が発生します。まずは現行のランレベルのスクリプトについて、停止処理が動作します。つまり、現在のランレベルで動作しているいくつかのプログラムを終了することになります。その後、新しいランレベルで動作するものを起動します。多くの場合、ここでは複数のプログラムが動き出します。たとえばランレベル 3 から 5 に移行する際、下記のような処理が動きます:

1. **telinit 5** を実行することで、管理者 (**root**) は **init** に対して ランレベルの変更を要求します。
2. **init** は現在のランレベル (**runlevel**) を確認し、新しいランレベルで **/etc/init.d/rc** に対して 指定するパラメータを判断します。
3. ここで **rc** は、新しいランレベルで起動する必要のないプログラムについて、停止スクリプトを呼び出します。たとえばこの例では、**/etc/init**

.d/rc3.d (現在のランレベル 3) に存在し、K で始まる全てのスクリプトを呼び出します。K の後に続く数字は、stop を呼び出す際の実行順序を決めるための値で、プログラム間相互の依存関係が存在することから付けられているものです。

- 最後に新しいランレベルで開始すべきスクリプトを実行します。たとえばこの例では /etc/init.d/rc5.d 内で S で始まる全てのスクリプトを呼び出します。S の後に続く数字は起動順序を決めるためのものです。

現在のランレベルと同じランレベルに変更しようとした場合、init は /etc/inittab が変更されていないかどうかを確認し、たとえば getty などのインターフェイスを起動するなどの必要な処理を行いません。telinit q と実行すると、これと同じことを行なうことができます。

16.2.2 初期化スクリプト

/etc/init.d 内には、2 種類のスクリプトが存在します：

init から直接実行されるスクリプト

このようなスクリプトは起動処理や即時のシステムシャットダウンを行なう際にのみ実行されます (電源喪失や **Ctrl + Alt + Del** の押下など)。このようなスクリプトは、/etc/inittab 内で実行されます。

init から間接的に実行されるスクリプト

これらのスクリプトはランレベルを変更した場合に実行するもので、/etc/init.d/rc 経由で呼び出されます。これにより各スクリプトの実行順序を保証しています。

全てのスクリプトは /etc/init.d 内に保存されています。起動時に実行するスクリプトは /etc/init.d/boot.d にあるシンボリックリンクを介して呼び出され、ランレベルを変更する際のスクリプトはそれぞれのサブディレクトリ ((/etc/init.d/rc0.d から /etc/init.d/rc6.d まで) を介して呼び出されます。これは単にわかりやすさと複数のランレベルでの重複を避けることを意図した作りになっています。各スクリプトは開始と終了に対応していて、start や stop をパラメータとして解釈しなければなりません。また、スクリプトは restart, reload, force-reload, status の各オプションを解釈する場合もあります。それぞれのオプションの意味については、表 16.2 「よくある初期化スクリプトのオプション指定」 (266 ページ) をお読みく

ださい。init から直接呼び出されるスクリプトにはこのようなリンクは存在せず、必要に応じてランレベルから独立して実行されます。

表 16.2 よくある初期化スクリプトのオプション指定

オプション	説明
start	サービスを起動します。
stop	サービスを停止します。
restart	サービスが実行中の場合はいったん停止し、再起動します。実行中でなかった場合はそのまま起動します。
reload	サービスを停止したり再起動したりすることなく、設定を読み込みなおします。
force-reload	サービスが対応している場合にのみ、設定を読み込みなおします。対応していない場合は restart と同じ動作になります。
status	サービスの状態を表示します。

それぞれのランレベルのディレクトリ内にリンクを作成することで、異なるランレベルに対して各スクリプトを割り当てることができるようになっています。パッケージをインストールしたりアンインストールしたりした場合は、insserv (または insserv を呼び出す /usr/lib/lsb/install_initd プログラム) を利用して リンクを作成または削除します。詳しくは insserv(8) のマニュアルページをお読みください。

これらの設定の全ては YaST モジュールから変更することもできます。コマンドラインから状態を確認したい場合は、chkconfig ツールをお使いください。詳しくは chkconfig(8) のマニュアルページをお読みください。

起動時最初に実行される boot スクリプトや、終了時の最後に実行される stop スクリプトについて、それぞれそのメンテナンス方法と説明を紹介します。

boot

システム起動中に **init** から直接実行されます。選択したランレベルとは無関係に一度だけ実行されます。このスクリプトで **/proc** ファイルシステムや **/dev/pts** ファイルシステムをマウントし、**blogd** (**boot logging daemon**; 起動ログデーモン) を起動します。更新やインストールが完了した後の初回起動では、ここから初期システム設定を起動します。

blogd デーモンは **boot** と **rc** から起動されるサービスで、他のサービスが起動されるよりも前に起動します。たとえばブロックスペシャルファイルが利用できるようにするなど、その他のスクリプトが完了した段階で動作が停止し、**/var** ファイルシステムが書き込み可能状態でマウントされると、画面出力を **/var/log/boot.msg** に書き込みます。**/var** が書き込み可能な状態で無かった場合は、書き込みが可能になるまで画面出力をバッファし続けます。**blogd** についてより詳しい情報は、**blogd(8)** のマニュアルページをお読みください。

また **boot** スクリプトは、**/etc/init.d/boot.d** ディレクトリ以下にあるファイルのうち、**s** で始まるものを起動する処理も行ないます。ここではファイルシステムのチェック作業を行ない、必要であればループデバイスの設定も行ない、システム時刻の設定も行ないます。自動チェックの際にエラーが発生していて、ファイルシステムの修復を必要とする場合、システム管理者は **root** パスワードの入力後に手動で作業を行なうことができます。最後に実行するスクリプトが **boot.local** です。

boot.local

ここには、指定したランレベルに移行する前に実行すべき追加のコマンドを入力します。**DOS** システムで言うところの **AUTOEXEC.BAT** に相当するものです。

halt

このスクリプトはランレベル **0** または **6** に移行する際にのみ実行されます。ここには停止や再起動の際に実行されるプログラムを記述します。システムがシャットダウンするか、再起動するかは、**halt** の呼び出し方法に依存します。シャットダウン時に特別なコマンドが必要である場合、**halt.local** スクリプトに追加してください。

rc

このスクリプトは現在のランレベルに対して必要な **stop** スクリプトを呼び出し、新しく選択したランレベルに対して必要な **start** スクリプトを呼

び出します。/etc/init.d/boot と同様に、このスクリプトは /etc/inittab から呼び出されるもので、パラメータとして 選択したランレベルが指定されます。

上記に示したとおり、独自のスクリプトを作成して仕組みに組み入れることを 簡単に行なうことができます。カスタムスクリプトについての書式や命名方法、統合方法については、LSB の仕様説明や `init` `init.d`, `chkconfig`, `insserv` の各マニュアルページをお読みください。また、`startproc` や `killproc` のマニュアルページもご覧ください。

警告: 初期化スクリプトの間違いによる起動障害

初期化スクリプト内に間違いがあると、お使いのマシンが起動中に停止してしまいます。これらのスクリプトを編集する際は特に注意し、可能であればマルチユーザ 環境で厳重なテストを行なってください。初期化スクリプトについて便利な情報が 16.2.1 項 「ランレベル」 (262 ページ) にありますので、そちらもあわせて お読みください。

特定のプログラムやサービスに対してカスタムな初期化スクリプトを作成するには、/etc/init.d/skeleton ファイルを雛型にしてください。このファイルのコピーを新しい名前で作成し、必要なプログラムやファイル名、パスなどのデータを入力してください。独自の設定でスクリプトの機能を拡張する必要がある場合は、それぞれ `init` から適切な動作を記述してください。

冒頭にある `INIT INFO` ブロックは、スクリプト内に存在 していなければならない部分で、必要に応じて編集を行ないます。詳しくは 例16.1 「最小限の `INIT INFO` ブロック」 (268 ページ) をお読みください。

例 16.1 最小限の `INIT INFO` ブロック

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

`INFO` ブロックの冒頭には `Provides:` と書かれた行がありますが、ここには `init` スクリプトから操作されるプログラムやサービスの名称を指定します。

`Required-Start:` や `Required-Stop:` の行には、このサービス自身を起

動したり 停止したりする前に起動または停止しておきたい全てのサービスを列挙します。この情報は、各ランレベルのディレクトリ内にスクリプトを作成する際、その番号を生成する基準として使用されます。また、`Default-Start:` と `Default-Stop:` は、サービスを自動的に開始または終了させたいランレベルを指定します。最後に `Description:` では、サービスやプログラムの短い説明文を記入します。

ランレベルのディレクトリ (`/etc/init.d/rc?.d/`) に対して `/etc/init.d/` にあるスクリプトからリンクを作成するには、`insserv` 新しいスクリプトの名前と入力してください。`insserv` プログラムは `INIT INFO` ヘッダを読み込み、ランレベルのディレクトリ (`/etc/init.d/rc?.d/`) に対して開始したり停止したりするためのリンクを作成します。また、このプログラムはそれぞれリンクのファイル名で表わされる起動や停止の順序についても取り扱います。これらのリンクをグラフィカルなインターフェイスで作成したい場合は、16.2.3項「YaST を利用したシステムサービス (ランレベル) の設定」(269 ページ) に書かれた YaST のランレベルエディタをご利用ください。

既存のランレベルに組み込むべきスクリプトが既に `/etc/init.d/` 内に存在する場合は、`insserv` を利用するか、もしくは YaST ランレベルエディタで関連するサービスを有効にするかなどの手段で、ランレベルディレクトリ内にリンクを作成してください。変更内容は次の再起動で適用され、新しいサービスが自動的に起動されるようになります。

なお、これらのリンクは手動で作成したりはしないでください。`INFO` ブロック内に何らかの間違いがあると、他の何らかのサービスを設定するために `insserv` を実行するタイミングで問題が顕在化します。手動で追加したサービスの場合、`insserv` を実行することでファイルが削除されます。

16.2.3 YaST を利用したシステムサービス (ランレベル) の設定

`YaST > システム > システムサービス (ランレベル)` を選択して YaST ランレベルモジュールを起動すると、利用可能な全てのサービスについて概要と有効化状態 (無効か有効か) が表示されます。本モジュールは、簡易モードまたは熟練者モードのいずれかを選択することができます。簡易モードでは、多くの用途での十分な機能を提供しています。左側の列にはサービスの名前が、真ん中の列には現在の状態が、右側の列には短い説明がそれぞれ書かれ

ています。サービスを選択すると、さらに詳しい説明がウインドウの下部に表示されます。サービスを有効にするには、表内で該当するサービスを選択して、有効にするを選択してください。無効化する場合も同様です。

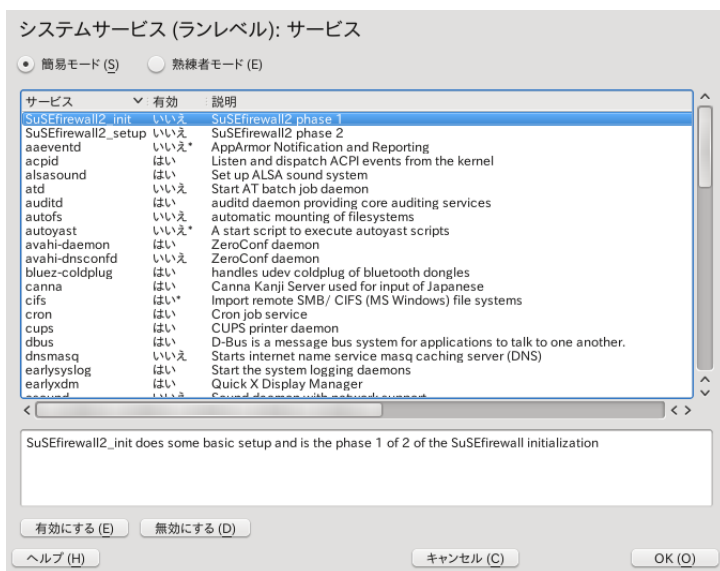
ランレベルに対してサービスを開始したり停止したりする作業以外のことを行ないたい場合や、既定のランレベルを変更したりしたい場合は、最初に熟練者モードを選択してください。現在の既定のランレベル(「initdefault」とも呼ばれます。システムの起動後に何も指定がない場合に移行するランレベル)が冒頭に表示されます。通常は openSUSE における既定のランレベルは 5 (ネットワークと X システムを有効にした 完全マルチユーザモード) に設定されています。環境によってはランレベルを 3 (ネットワークを有効にした完全マルチユーザモード) にしたほうが適切である場合もあります。

この YaST ダイアログでは、既定のランレベルを 1 つ選択することが出来ます(ランレベルの意味については表16.1「利用可能なランレベル」(262 ページ)をお読みください)。また、このウインドウ内の表では、個別のサービスやデーモンに対して、有効化や無効化を設定することができます。表には利用可能なサービスやデーモンが一覧表示されていて、お使いのシステムで有効になっているかどうかと、有効化されているランレベルが表示されます。マウスなどで 1 行を選択し、それぞれのランレベル (*B, 0, 1, 2, 3, 5, 6, S*) に対応したチェックボックスを選択することで、選択したサービスやデーモンがどのランレベルで起動するのかが設定することができます。なお、ランレベル 4 はカスタムなランレベルを作成するための空き領域です。また、選択したサービスやデーモンの概要説明については、表の下側に表示されます。

警告: ランレベル設定の誤りによるシステム障害

ランレベルの設定を誤ると、お使いのシステムが利用できなくなります。設定を適用する前に、どのような影響があるのかをご確認ください。

図 16.1 システムサービス (ランレベル)



開始／中止／更新を押すと、サービスを開始したり停止したりすることができます。状態を更新を押すと、現在の状態を表示することができます。また、セット／リセットを押すと、行なった変更をシステムに適用したり、ランレベルエディタを起動する 前の設定に戻したりすることができます。OKを押すと、変更点をディスクに書き込みます。

16.3 /etc/sysconfig 経由でのシステム設定

openSUSE では、主な設定を /etc/sysconfig で制御しています。/etc/sysconfig 内にあるそれぞれのファイルは、それを必要とする各スクリプトで読み込まれます。たとえば ネットワークの設定は、ネットワーク関連のスクリプトで処理されます。

システム設定を編集するには、2 通りの方法があります。YaST sysconfig エディタを使用する方法と、エディタなどを使用して手作業で設定ファイルを編集する方法です。

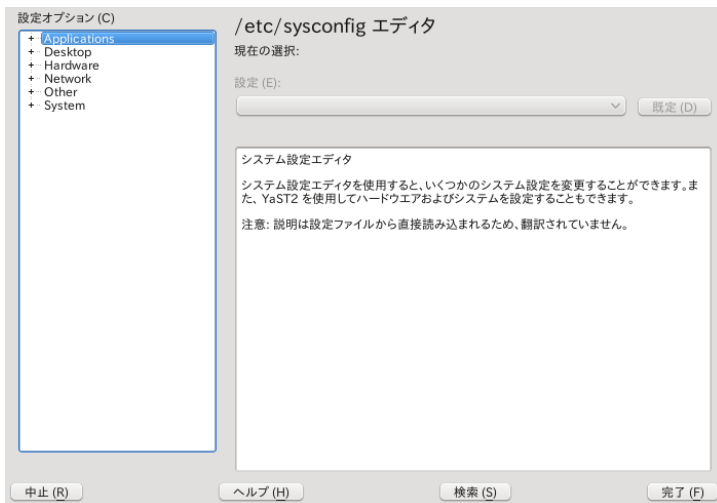
16.3.1 YaST sysconfig エディタを利用したシステム設定の変更

YaST sysconfig エディタは、システム設定を行なうための使いやすいフロントエンドです。変更を行ないたい値に対して実際の場所を知らなくても、このモジュールに内蔵された検索機能を利用するなどを行なうことで、値の変更を行なうことができますし、それらの変更を適用する際にも sysconfig 内にある設定値を自動的に変更し、必要なサービスを再起動します。

警告: /etc/sysconfig/* ファイルの編集によるシステム障害

知識や経験のない方の場合は、sysconfigを手作業で変更したりすることは避けてください。手作業で編集を行なうと、取り返しのつかないダメージを与えてしまうことがあります。/etc/sysconfig 内のファイルでは各値に短い説明文がコメント文として書かれていますので、こちらをご確認のうえ、どのようなことが起こるのかをお読みください。

図 16.2 sysconfig エディタを利用したシステム設定



YaST sysconfig のダイアログは3つのパーツから構成されています。ダイアログの左側には全ての設定項目に対するツリー構造の表示があります。値を選択すると、右側に現在選択している設定項目とその現在値が表示されます。

その下側にある3つめのウィンドウには、その設定項目に対する短い説明と設定可能な値、および既定値と設定の発信元ファイル名が示されています。また、このダイアログでは設定値を変更した後に実行される設定スクリプトと、変更結果としてどの新しいサービスが起動されるのかについても表示されます。YaSTは完了ボタンを押してダイアログを終了する際、変更点の確認とともに実行されるスクリプトを表示します。後から起動したい場合は、サービスやスクリプトを飛ばすこともできます。YaSTでは自動的に全ての変更点を適用し、変更を反映するために必要なサービスを再起動することができます。

16.3.2 手作業でのシステム設定変更

システム設定を手動で変更するには、下記のように実施します:

- 1 root ユーザでログインするか、もしくは移行します。
- 2 `telinit 1` でシングルユーザモードに移行します。
- 3 エディタを使用して、設定ファイルのうち必要な箇所を変更します。

`/etc/sysconfig` 内にある設定ファイルの変更について YaST を利用しない場合、何も値を設定しない場合は2つの引用符が書かれていることをご確認ください(たとえば `KEYTABLE=""` のように)。また、空白を含む値を設定する場合は、それらを引用符で括っていることについてもご確認ください。空白を含まない値である場合にのみ、引用符は不要になります。

- 4 変更を反映するため、`SuSEconfig` を実行します。
- 5 `telinit` 既定のランレベルのように実行して、お使いのシステムを以前のランレベルに戻します。ここで既定のランレベルは、お使いのシステムにおける既定のランレベル番号を記入します。ネットワークとXシステムの設定された完全マルチユーザ環境の場合は5を、ネットワークだけが設定された完全マルチユーザ環境に戻す場合は3を指定します。

上記の作業手順は、システム全体に関わる設定を変更する場合にのみ行なうべきものです。これにはたとえば、ネットワークの設定などが該当します。小さな変更程度であればシングルユーザモードへの移行は不要ですが、関連する全てのプログラムについて正しく再起動を行なわなければならないことに注意してください。

ティップ: システム設定の自動化設定

SuSEconfig による自動システム設定を無効化するには、`/etc/sysconfig/suseconfig` ファイルにある `/etc/sysconfig/suseconfig` の値を、`no` に設定してください。ただし、**SUSE** のインストールサポートをご利用いただく場合は、**SuSEconfig** を無効化しないでください。なお、自動設定を部分的に無効化することもできます。

ブートローダ GRUB

本章では、openSUSE® で使用されるブートローダ GRUB について、その設定方法を説明しています。全ての設定は専用の YaST を利用して行なうことができます。Linux の起動手順について詳しくない場合は、下記の章をお読みのうえ、起動に関する知識を得てください。また、本章では GRUB で起動する際に発生する可能性のある問題について、その内容と解決方法についても述べています。

本章では、ブートローダ GRUB を利用した起動の管理と設定方法について述べています。起動処理の全体像については 第16章 *Linux システムの起動と設定* (257 ページ) で述べています。ブートローダはマシン (BIOS) とオペレーティングシステム (openSUSE) の橋渡しをするためのもので、ブートローダの設定を変更すると、オペレーティングシステムの開始方法について直接的な影響を与えることができます。

本章ではそれぞれ下記の用語を使用しています。説明をよくお読みのうえ、以降を読み進めてください:

マスターブートレコード (MBR)

MBR の構造は、オペレーティングシステムでそれぞれ独立した仕組みになっています。最初の 446 バイトがプログラムコードとして予約されています。この領域には一般にブートローダプログラムや、オペレーティングシステムを選択するための仕組みの一部が書き込まれています。次の 64 バイトは、最大で 4 つまでの定義を行なうことができるパーティションテーブルの領域です。パーティションテーブルにはハードディスクのパーティション情報と、そのパーティションの種類が含まれています。オペレーティングシステムは、この一覧を利用してパーティションを処理します。通常の MBR では、必ずいずれか 1 つのパーティションに

対して アクティブのマークが付けられています。MBR の最後の 2 バイトは「マジックナンバー」と呼ばれているもので、固定値 (AA55) が書かれています。マジックナンバーが左記の値でない場合、BIOS は MBR を無効なものとして扱い、起動時に読み込みを行なわなくなります。

ブートセクタ

ブートセクタとはハードディスクのパーティション内にある最初のセクタで、他のパーティションに対する「コンテナ」である拡張パーティション以外の場所に存在しています。これらのブートセクタは 512 バイトから構成されていて、それぞれのパーティション内にインストールされている、DOS, windows, OS/2 などのオペレーティングシステムを起動するためのコード領域になっています。また、ブートセクタにはファイルシステム内の特別な情報も保持しています。Linux パーティションのブートセクタには、XFS の場合を除き、ファイルシステム 設定後には何も書かれていません。そのため Linux パーティションは、自分自身にカーネルや有効なルートファイルシステムが含まれていたとしても、自分自身で起動を行なうことができません。なお、起動可能なシステムが含まれるパーティションの場合、MBR と同じマジックナンバー (AA55) が書かれています。

17.1 GRUB での起動

GRUB (Grand Unified Bootloader) は 2 つのステージから構成されています。第 1 ステージは 512 バイトから構成されているもので、ブートローダの第 2 ステージを読み出す処理だけを行ないます。第 2 ステージがブートローダの中心部分です。

設定によっては、これらの中間である第 1.5 ステージを利用することもできます。このステージは適切なファイルシステムから第 2 ステージを読み込むもので、YaST から GRUB を初期設定したとき、可能であればこの方法がとられます。

第 2 ステージでは様々なファイルシステムにアクセスすることができます。現時点では ext2, ext3, reiserfs, minix, Windows で使用される DOS FAT ファイルシステムにそれぞれ対応しています。拡張機能として、XFS, BSD システムで使用される UFS, FFS にも対応しています。0.95 以降のバージョンで、GRUB は「El Torito」仕様に準拠した ISO 9660 標準ファイルシステムを含む CD または DVD からの起動も行なうことができるようになりました。また GRUB

は、ファイルシステムが起動する前であっても、対応する BIOS ディスクデバイス (BIOS で検出されたフロッピーディスク、ハードディスク、CD/DVD ドライブ) 内にあるファイルシステムにアクセスすることができます。そのため、GRUB の設定ファイル (menu.lst) を書き換えても、ブートマネージャのインストールをやり直す必要がありません。システムが起動すると、grub は正しいパスからメニューファイルやカーネルのパーティションデータ、初期 RAM ディスク (initrd)などを再度読み込み、場所を確定します。

GRUB における実際の設定は、下記に示す 4 つのファイルから構成されています:

/boot/grub/menu.lst

このファイルには、GRUB から起動することのできるパーティションやオペレーティングシステムに関する全ての情報が保存されています。この情報が存在しないと GRUB はプロンプトを表示し、ユーザに対して入力を求める動作を行ないます。詳しくは「起動処理時のメニュー項目編集」(283 ページ)をお読みください。

/boot/grub/device.map

このファイルには、GRUB のデバイス名や BIOS 表記法と Linux のデバイス名の変換情報が記載されています。

/etc/grub.conf

このファイルには、ブートローダを正しくインストールするために必要な、GRUB に対するコマンドやパラメータ、オプションが含まれています。

/etc/sysconfig/bootloader

このファイルは perl-bootloader ライブラリから読み込むことができるもので、YaST からブートローダを設定する際、および新しいカーネルがインストールされた際に利用されます。このファイルにはカーネルパラメータなどの設定オプションが含まれていて、これらはブートローダの設定ファイルに対して既定で追加されます。

GRUB は様々な方法でコントロールすることができます。既存の設定ファイルにある起動項目はグラフィカルなメニュー (スプラッシュスクリーン) で選択することができます。設定は menu.lst ファイルから読み込みます。

GRUB では、起動を行なう前に全ての起動パラメータを変更することができます。たとえばメニューファイルを編集していたときに存在したエラーを、

起動時に変更することも可能です。起動コマンドは入力用のプロンプトから対話的に入力することもできます。詳しくは「起動処理時のメニュー項目編集」(283 ページ)をお読みください。また GRUB は、起動前にカーネルの位置や `initrd` の位置を指定することもできます。この方法により、ブートローダの設定に存在していないインストール済みオペレーティングシステムを起動することができます。

GRUB には 2 つのバージョンが存在します: ブートローダと、通常の Linux プログラム (`/usr/sbin/grub`) です。後者は *GRUB シェル* と呼ばれる場合もあります。GRUB シェルはインストール済みのシステムで GRUB のエミュレーションを行ない、GRUB をインストールしたり新しい設定を適用前に確認したりすることができます。ハードディスクやフロッピーディスクのブートローダとして GRUB をインストールする機能は、GRUB の内蔵コマンド `setup` を利用して行ないます。このコマンドは、Linux が読み込まれている場合に GRUB シェルで利用できます。

17.1.1 /boot/grub/menu.lst ファイル

起動メニューでのグラフィカルなスプラッシュスクリーンは、GRUB の設定ファイル `/boot/grub/menu.lst` を基礎にしています。このファイルにはそのほか、このメニューから起動可能な全てのパーティションとオペレーティングシステムに関する情報も含まれています。

システムが起動する際、GRUB は常にファイルシステムからメニューファイルを読み込みます。そのため、GRUB はメニューファイルを変更した場合であっても、再インストールを行なう必要はありません。GRUB の設定ファイルを変更するには、17.2 項「YaST を利用したブートローダの設定」(288 ページ)に書かれている手順で YaST ブートローダモジュールをお使いください。

メニューファイルにはコマンドが含まれていますが、文法はとても単純です。シェルのようにして各行にコマンドを記述し、スペースで区切ってオプションのパラメータを指定するだけです。過去のバージョンとの互換性を保つ理由から、コマンドによっては最初のパラメータの冒頭に `=` を入れることができるものもあります。コメントは行頭にハッシュ記号 (`#`) を記入します。

メニュー一覧で項目の識別を行なうため、各項目に対して `title` (タイトル) を設定します。キーワード `title` 以降に設定したスペースを含む任意のテキストを、メニュー内の選択オプションとして表示することができます。 `title`

以降の行に続くコマンドは、そのメニューが選択された場合にのみ実行されます。

最も単純なケースは、他のオペレーティングシステムのブートローダに転送することです。そのときに使用するコマンドは `chainloader` で、その後ろに続くパラメータとして、他のパーティションのブートブロックを示す値を入力します。**GRUB** では、ブロックの指定を下記のようにして行ないます：

```
chainloader (hd0,3)+1
```

GRUB でのデバイス名については、「ハードディスクとパーティションの名前ルール」(280 ページ)で説明しています。上記の例では、1 台目のハードディスクの 4 つめのパーティションに対して、最初のブロックを指定しています。

また、`kernel` コマンドを利用すると、カーネルイメージを指定することができます。最初のパラメータには、パーティション内でのカーネルイメージのパスを指定します。それ以降のパラメータには、カーネルのコマンドラインを指定します。

カーネル内蔵のドライバではルートパーティションにアクセスできなかったり、高度なホットプラグ機能のある **Linux** システムであったりする場合は、`initrd` を個別の **GRUB** コマンドとして指定しなければなりません。`initrd` コマンドのパラメータは 1 つだけ、`initrd` に対するパスを指定します。`initrd` を読み込む場所は読み込んだカーネルに書き込まれるため、`initrd` コマンドは `kernel` コマンドの後に記述しなければなりません。

`root` コマンドはカーネルと `initrd` ファイルの指定を単純化するための仕組みです。`root` に指定するパラメータは 1 つだけで、デバイスやパーティションを指定します。ここで指定したデバイスは、各コマンドで個別にパスを指定せず、かつ次の `root` コマンドが現われない限り、全てのカーネルや `initrd`、もしくはその他のファイルパス指定に使用されます。

`boot` コマンドは各メニュー項目の最後に実行されるコマンドですが、メニューファイルでは記入する必要はありません。しかしながら、**GRUB** を対話的に使用している場合は、最後に `boot` コマンドを入力しなければなりません。このコマンド自身には何もパラメータを指定しません。単にそれまでに指定したカーネルイメージやチェーンローダを読み出すだけです。

全てのメニュー項目を記入したら、それらの中のいずれかを default (既定の) 項目として指定します。指定を行わない場合は、最初の項目 (0 を指定した場合と同じ) を選択します。また、既定の項目を選択するまでのタイムアウトを timeout コマンドで指定することもできます。timeout と default コマンドは通常、メニュー項目よりも前に記述します。ファイルの記入例は「メニューファイルの例」(281 ページ)にありますので、こちらを参考にしてください。

ハードディスクとパーティションの名前ルール

GRUB でハードディスクやパーティションに対して使用する名前付けは、通常の Linux のデバイス名とは異なるものを使用します。BIOS が使用する単純なディスク列挙の仕組みにとっても似ていて、文法は BSD システムの方式に似たものになっています。また、GRUB では最初のパーティション番号は 0 になっています。つまり、(hd0, 0) は最初のハードディスク内にある最初のパーティションを意味することになります。一般的なデスクトップマシンの場合、プライマリマスターに接続されているハードディスクのことを指し、Linux のデバイス名で言うと、/dev/sda1 になります。

4 つまで作成できるプライマリパーティションは、それぞれ 0 から 3 までの間に割り当てられています。論理パーティションは 4 以降になります:

(hd0, 0)	最初のハードディスクにある最初のプライマリパーティション
(hd0, 1)	2 つめのプライマリパーティション
(hd0, 2)	3 つめのプライマリパーティション
(hd0, 3)	4 つめのプライマリパーティション (一般的には拡張パーティション)
(hd0, 4)	最初の論理パーティション
(hd0, 5)	2 つめの論理パーティション

BIOS でのデバイス表記と同じで、GRUB は IDE, SATA, SCSI, ハードウェア RAID デバイスを区別せずに使用します。BIOS やその他のコントローラで認識される全てのハードディスクは、BIOS 内で設定した起動順序に従って番号が付けられます。

残念ながら、Linux のデバイス名を正しく BIOS のデバイス名に変換することはできません。この割り当ては特定のアルゴリズムに従って生成され、device.map ファイルに保存されます。必要であれば、このファイルを編集することもできます。device.map に関する情報は、17.1.2 項「device.map ファイル」(284 ページ)をお読みください。

GRUB で完全なパスを指定するには、まず括弧内にデバイス名を指定したあと、パーティション(ファイルシステム)内でのパスを記述します。なお、パスはスラッシュから書き始めます。たとえば起動可能なカーネルが最初の IDE ハードディスクにおける最初のパーティションに存在する場合、下記のように指定することができます:

```
(hd0,0)/boot/vmlinuz
```

メニューファイルの例

下記はGRUB メニューファイルの構造を示すための例です。下記のインストール例では起動パーティションが /dev/sda5 に、ルートパーティションが /dev/sda7 に、Windows のインストールが /dev/sda1 にそれぞれ行なわれている場合を想定しています。

```
gfxmenu (hd0,4)/boot/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    root (hd0,4)
    kernel /boot/vmlinuz root=/dev/sda7 vga=791 resume=/dev/sda9
    initrd /boot/initrd

title windows
    rootnoverify (hd0,0)
    chainloader +1

title floppy
    rootnoverify (hd0,0)
    chainloader (fd0)+1

title failsafe
    root (hd0,4)
    kernel /boot/vmlinuz.shipped root=/dev/sda7 ide=nodma \
    apm=off acpi=off vga=normal nosmp maxcpus=0 3 noresume
    initrd /boot/initrd.shipped
```

最初のブロックでは、スプラッシュスクリーンの設定を行なっています:

```
gfxmenu (hd0,4)/message
/dev/sda5 パーティション内のルートディレクトリにある message
ファイルを、背景イメージとして使用する設定です。
```

color white/blue black/light-gray

色の設定を行なっています。前景を白、背景を青に設定し、選択しているものを黒で、選択の背景をライトグレーで表示します。色の設定はスプラッシュスクリーンには影響しません。Escでスプラッシュスクリーンを抜けた場合にのみ意味のある設定です。

default 0

最初のメニュー項目である `title linux` を既定の設定として起動します。

timeout 8

ユーザ入力 が 8 秒間行なわれないと、GRUB は自動的に既定の項目を起動します。自動起動を無効化するには、`timeout` の行を削除してください。`timeout 0` を設定すると、既定の項目を即時起動する意味になります。

2番目の以降のブロックは、様々なオペレーティングシステムを起動するための設定です。それぞれのオペレーティングシステムの設定は、`title`で始まります。

- 最初の項目 (`title linux`) は、openSUSE を起動するための項目です。カーネル (`linux`) は最初のハードディスクにおける最初の論理パーティション (ブートパーティション) 内に位置しています。ルートパーティションや VGA モードなどのカーネルパラメータが後に続いています。ルートパーティションの指定は Linux の名前付けルールに従って `/dev/sda7/` と書かれています。これはこの情報を読むのがカーネルであるためで、GRUB はその値について何も処理を行なわないためです。`initrd`についても同様に、最初のハードディスクにおける最初の論理パーティション内に位置しています。
- 2つめの項目は Windows を読み込むための項目です。Windows は最初のハードディスク (`hd0, 0`) における最初のパーティションから起動する設定になっています。`chainloader +1` のコマンドは、GRUB に対して指定したパーティションで最初のセクタを読み出して実行するように指定するコマンドです。
- 3つめの項目は、単に BIOS 設定を変更せずにフロッピーディスクから起動するための項目です。

- `failsafe` と書かれている項目は、問題の発生しているマシンであっても **Linux** を起動できるよう、カーネルパラメータをいくつか指定して起動するための項目です。

メニューファイルは必要な時に変更することができ、**GRUB** は次の起動時に変更済みの設定を使用します。設定の変更は、**YaST** かエディタなどで編集してください。また、代替策として **GRUB** の機能を利用して一時的に変更することもできます。詳しくは「起動処理時のメニュー項目編集」(283ページ)をお読みください。

起動処理時のメニュー項目編集

グラフィカルな起動メニューでは、カーソルキーを利用して起動するオペレーティングシステムを選択することができます。なお、**Linux** システムを選択した場合は、起動プロンプトを利用して追加のパラメータを設定することもできます。個別のメニュー項目を直接編集したい場合は **Esc** を押し、スプラッシュスクリーンを抜けて **GRUB** のテキストベースのメニューを表示させてから、**E** を押します。このようにして変更した内容は、その時点の起動でのみ有効で、恒久的に適用されることはありません。

重要項目: 起動処理時のキーボードレイアウト

起動時には英語(アメリカ英語)キーボードレイアウトだけを利用できます。詳しくは 図 「英語キーボードのレイアウト」(↑スタートアップ)をお読みください。

メニュー項目の編集を行なうと、うまく起動できないシステムに対する修復を行なうことができます。これは、間違ったブートローダの設定を手作業で修正することで、うまく起動するための回避策を入力することができるためです。起動処理内での手動でのパラメータ入力、システムの設定を恒久的に変更せず、一時的に新しい設定をテストしたりしたい場合にも便利です。

編集モードを有効にしたあと、まずはカーソルキーを利用して編集する行を選択します。ここからさらに **E** を押すと、選択した行を編集することができます。この方法で、起動処理を行なう前に間違ったパーティション指定やパス指定を修正してください。編集モードを抜けてメニューに戻るには、**Enter** を押します。メニューから **B** を押すと、その設定で起動を行ないます。それ以外の処理は、画面下部のヘルプテキストに表示されています。

起動オプションを恒久的に変更してそれらの設定をカーネルに渡したい場合は、root ユーザで menu.lst ファイルを開き、それぞれ必要なカーネルパラメータを既存の行に設定してください。複数のパラメータはスペースで区切ります:

```
title linux
    root(hd0,0)
    kernel /vmlinuz root=/dev/sda3 追加のパラメータ
    initrd /initrd
```

GRUB は次回の起動時に自動で設定したパラメータを読み込みます。YaST ブートローダモジュールからでも同じことを行なうことができます。上記と同様に、新しいパラメータはスペースで区切って指定します。

17.1.2 device.map ファイル

device.map ファイルは、GRUB や BIOS のデバイス名を Linux のデバイス名に変換するためのファイルです。IDE と SCSI のハードディスクが混在するシステムの場合、GRUB は特殊な手順で起動順序を判断しなければなりません。これは GRUB が起動順序の設定を行なっている BIOS 情報にアクセスできない可能性があるためです。GRUB はこの分析結果を /boot/grub/device.map ファイルに保存します。たとえば BIOS で SCSI よりも IDE を優先して起動するように設定しているシステムでは、device.map ファイルは下記のようになります:

```
(fd0) /dev/fd0
(hd0) /dev/sda
(hd1) /dev/sdb
```

もしくは下記のような場合もあります:

```
(fd0) /dev/fd0
(hd0) /dev/disk-by-id/DISK1 の ID
(hd1) /dev/disk-by-id/DISK2 の ID
```

IDE や SCSI、もしくはその他のハードディスクは様々な要素に依存していて、Linux ではその割り当てを識別することができないことから、device.map ファイルのある順序を手動で編集することもできます。起動時に何らかの問題が発生した場合は、このファイル内にある順序が BIOS の順序とあっているかどうかを確認し、GRUB プロンプトから必要に応じて一時的に変更してみてください。その設定で Linux システムが問題なく起動するようであ

れば、YaST ブートローダモジュールやエディタなどを利用して、`device.map` ファイルを恒久的に変更してください。

`device.map` ファイルを手作業で変更した場合は、下記のコマンドを入力して **GRUB** を再インストールしてください。このコマンドを実行すると、`device.map` ファイルを読み込み直し、`grub.conf` ファイルにあるコマンドを実行します:

```
grub --batch < /etc/grub.conf
```

17.1.3 /etc/grub.conf ファイル

`menu.lst` と `device.map` に続く 3 番目に重要な **GRUB** の設定ファイルとして、`/etc/grub.conf` ファイルがあります。このファイルには **GRUB** シェルに対するコマンドとパラメータが含まれていて、ブートローダを正しくインストールするのに必要なファイルです:

```
setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)
quit
```

上記の例では **GRUB** に対し、最初のハードディスクにある 2 番目のパーティション (`hd0,1`) にブートローダをインストールし、起動イメージが同じパーティション内に存在していることを示しています。

`--stage2=/boot/grub/stage2` パラメータは、マウント済みのファイルシステムから第 2 ステージのイメージをインストールするために必要なパラメータです。また、BIOS によっては **LBA** サポートの実装が誤っている場合があるため、`--force-lba` を指定してその間違った実装を無視するように指定しています。

17.1.4 /etc/sysconfig/bootloader ファイル

この設定ファイルは YaST を利用してブートローダを設定した場合、および新しいカーネルをインストールした場合にのみ使用されるものです。このファイルは、ブートローダの設定ファイル (たとえば **GRUB** であれば `/boot/grub/menu.lst`) を書き換える `perl-bootloader` ライブラリが解釈します。なお、`/etc/sysconfig/bootloader` ファイルは **GRUB** 固有の設定ファイルではありません。`openSUSE` 上にインストールされたブートローダであれば、どのブートローダにも適用されます。

注記: カーネル更新後のブートローダ設定

新しいカーネルがインストールされると、`perl` のブートローダモジュールは毎回、新しいブートローダの設定ファイル (たとえば `GRUB` であれば `/boot/grub/menu.lst`) を `/etc/sysconfig/bootloader` に設定された既定値で 作成します。カーネルパラメータをカスタマイズしている場合は、`/etc/sysconfig/bootloader` を適宜変更し、カーネル更新後も必要な設定が反映されるようにしてください。

LOADER_TYPE

お使いのシステムにインストールされているブートローダを指定します (たとえば `GRUB` や `LILO` など)。この項目は手動では変更せず、手順17.6 「ブートローダの種類の設定」 (293 ページ) に示されている 手順で YaST を利用し、ブートローダを設定してください。

DEFAULT_VGA / FAILSAFE_VGA / XEN_VGA

起動処理時に利用するフレームバッファについて、画面の解像度と色深度の設定を行いません。これらはカーネルパラメータ `vga` に渡される値で、それぞれ既定の起動項目のほか、フェイルセーフ (安全設定) や `XEN` 設定で使用されます。それぞれ下記の値を設定することができます:

表 17.1 画面解像度と色深度の一覧

	640x480	800x600	1024x768	1280x1024	1600x1200
8 ビット	0x301	0x303	0x305	0x307	0x31C
15 ビット	0x310	0x313	0x316	0x319	0x31D
16 ビット	0x311	0x314	0x317	0x31A	0x31E
24 ビット	0x312	0x315	0x318	0x31B	0x31F

DEFAULT_APPEND / FAILSAFE_APPEND / XEN_KERNEL_APPEND

ブートローダの設定ファイル内で、既定の項目やフェイルセーフ設定、および `XEN` の起動項目に設定する、カーネルパラメータ (`vga` 以外) を指定します。

CYCLE_DETECTION / CYCLE_NEXT_ENTRY

ブートサイクルの検出を使用するかどうかと、使用していて起動がうまくいかなかった場合に、`/boot/grub/menu.lst` 内でどの代替項目 (たとえば フェイルセーフ など) を起動するかを指定します。詳しくは `/usr/share/doc/packages/bootcycle/README` をお読みください。

17.1.5 起動パスワードの設定

オペレーティングシステムが起動する前であっても、GRUB はファイルシステムにアクセスすることができます。root 権限のないユーザは、この方法で Linux システム内のファイルにアクセスする可能性があります。このようなアクセス方法を禁止したり、特定のオペレーティングシステムを起動できないようにしたりしたい場合は、起動パスワードを設定してください。

重要項目: 起動パスワードとスプラッシュスクリーン

GRUB で起動パスワードを使用する場合は、スプラッシュスクリーンは表示されなくなります。

root ユーザから下記の手順を実施することで、起動パスワードを設定することができます:

- 1 まずはコマンドプロンプトから `grub-md5-crypt` を利用し、パスワードを暗号化します:

```
# grub-md5-crypt
Password: ****
Retype password: ****
Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 2 上記の出力で、"Encrypted:" 以降の部分を `menu.lst` ファイル内のグローバルセクションに貼り付けます:

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

上記のように設定することで、起動プロンプトから **P** を押し、正しいパスワードを入力した場合にのみ、GRUB コマンドを実行することができるよ

うになります。ただし、起動メニュー内に記載されているオペレーティングシステムであれば、パスワードの入力なしでも実行できます。

- 3 起動メニューから1つまたは複数のオペレーティングシステムの起動ができないようにするには、パスワード無しでは起動できないように設定することができます。menu.lst ファイル内のセクションに対して、lock という行を追加してください。たとえば下記のように なります:

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

上記の設定でシステムを再起動すると、起動メニューから **Linux** を選択すると 下記のようなエラーメッセージが表示されます:

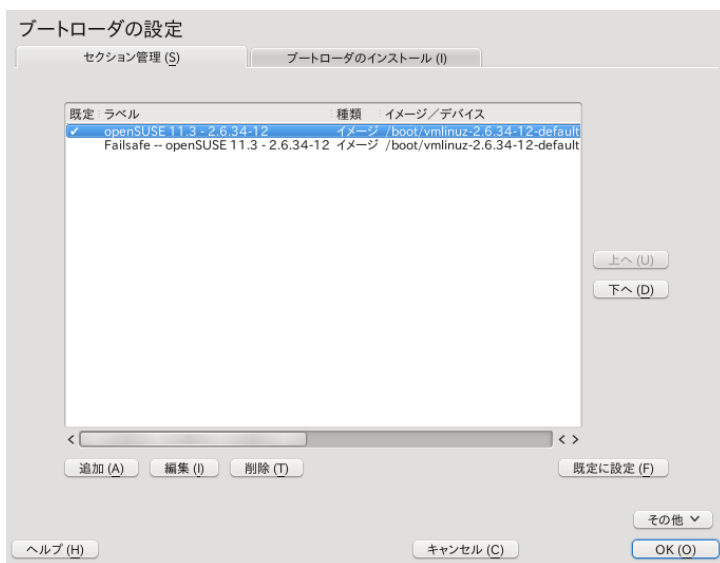
```
Error 32: Must be authenticated
```

Enter を押すとメニューに入ることができます。さらに **P** を押してパスワードプロンプトを表示させ、パスワードを入力して **Enter** を押してください。すると選択したオペレーティングシステム (この場合は **Linux**) を起動することができるようになります。

17.2 YaST を利用したブートローダの設定

お使いの openSUSE のブートローダを設定するのに最も簡単な方法は、YaST のモジュールを利用して設定することです。YaST のコントロールセンターから、システム > ブートローダを選択して設定を行ないます。図17.1「ブートローダの設定」(289 ページ)にあるとおり、ここではお使いのシステムでのブートローダ設定が表示され、変更を行なうことができます。

図 17.1 ブートローダの設定



個別のオペレーティングシステムに関するブートローダのセクションを編集したり変更したり、削除したりするには、**セクション管理** タブを利用します。セクションを追加するには **追加** を、既存のセクションについて設定値を変更するにはセクションを選択してから **編集** をそれぞれ押してください。またセクションを削除するには、削除したい項目を選んで **削除** を押してください。ブートローダのオプションについて詳しく知りたい場合は、まず 17.1 項「GRUB での起動」(276 ページ)をお読みください。

ブートローダの種類やインストール先、または高度なブートローダ設定を閲覧したり変更したりするには、**ブートローダのインストール** を利用します。

高度なオプションにアクセスするには、**その他** ボタンを押すことで表示されるドロップダウンメニューを利用します。内蔵のエディタを利用して **GRUB** の設定ファイルを直接編集することができます。詳しくは 17.1 項「GRUB での起動」(276 ページ)をお読みください。また、既存の設定ファイルを削除して全く新しい設定を作成することもできます。また、**YaST** に対して新しい設定を提示するように求めることもできるほか、設定をディスクに書き込んだり、ディスクから既存の設定を再読み込みしたりすることもできます。インストール時に保存しておいたマスターブートレコード(MBR)に戻すには、ハードディスクの **MBR** を復元するを選択してください。

17.2.1 既定の起動項目の設定

既定で起動が行なわれるシステムを変更するには、下記の手順で行ないます:

手順 17.1 既定のシステムの設定

- 1 セクション管理 タブを開きます。
- 2 既定で起動したい項目を一覧から選択します。
- 3 既定に設定 を押します。
- 4 最後に *OK* を押すと設定を保存することができます。

17.2.2 ブートローダのインストール先の変更

To modify the location of the boot loader, follow these steps:

手順 17.2 ブートローダのインストール先の変更

- 1 ブートローダのインストールタブを選択し、ブートローダの場所に対して以下のいずれかを 選択してください:

マスターブートレコード (MBR) から起動

これを選択すると、最初のディスク (BIOS で設定された順序で 最初に
あたるディスク) の MBR にブートローダをインストールします。

ルートパーティションから起動

これを選択すると、/ ディレクトリに 割り当てたパーティションに
対してブートローダをインストールします (これが既定値です)。

ブートパーティションから起動

これを選択すると、/boot ディレクトリに 割り当てたパーティション
に対してブートローダをインストールします。

拡張パーティションから起動

これを選択すると、拡張パーティションコンテナに対してブートロー
ダを インストールします。

カスタムブートパーティション

ブートローダの場所を手作業で指定するには、このオプションを選択してください。

- 2 変更を保存するには、*OK*を押します。

17.2.3 ブートローダの時間切れ設定

ブートローダは既定の項目をすぐには起動しません。時間切れとして設定した時間が経過するまでの間、起動するシステムを選択したりカーネルのパラメータを入力したりすることができます。ブートローダの時間切れを設定するには、下記の手順で行ないます:

手順 17.3 ブートローダの時間切れ設定

- 1 ブートローダのインストールタブを選択します。
- 2 ブートローダのオプションを押します。
- 3 タイムアウト(秒)の項目を選択して新しい値を入力するか、もしくはマウスやキーボードで矢印キーを操作して値を編集します。
- 4 *OK*を2回押して設定を保存します。

17.2.4 起動パスワードの設定

この YaST モジュールを利用することで、起動を確認するためのパスワードを設定することができます。これにより追加のセキュリティ保護を実現することができます。

手順 17.4 起動パスワードの設定

- 1 ブートローダのインストールタブを選択します。
- 2 ブートローダのオプションを押します。
- 3 ブートローダをパスワードで保護するのオプションを選択し、起動時に入力させたいパスワードをパスワードの欄に2度入力します。

4 OK を 2 回押して設定を保存します。

17.2.5 ディスク順序の設定

お使いのコンピュータに 2 台以上のハードディスクが接続されている場合、マシンの BIOS 設定に合わせてディスクの起動順序を指定することができます (17.1.2 項「device.map ファイル」(284 ページ) も合わせてご覧ください)。これを行なうには、下記の手順で行ないます：

手順 17.5 ディスク順序の設定

- 1 ブートローダのインストールタブを選択します。
- 2 ブートローダのインストール詳細を押します。
- 3 複数のディスクが一覧表示されている場合、ディスクを選択してから 上へまたは 下へ を押し、ディスクの順序を入れ替えます。
- 4 OK を 2 回押して設定を保存します。

17.2.6 高度なオプションの設定

高度な起動オプションは、ブートローダのインストール > ブートローダのオプションを選択すると設定できるようになります。通常は既定値から変更を行なう必要はありません。

ブートパーティションをアクティブに設定

ブートローダを含むパーティションをアクティブに設定します。いくつかの古いオペレーティングシステム (Windows 98 など) では、アクティブなパーティションからでないと起動できないものがあるためです。

MBR に汎用ブートコードを書き込む

現在の MBR に対し、オペレーティングシステムに依存しない汎用のコードを書き込みます。

デバッグ用フラグ

GRUB をデバッグモードで動作させ、ディスクの動作についてメッセージを表示するようにします。

ブート時にメニューを隠す

ブートメニューを隠し、既定の項目を起動するように設定します。

信頼済み *Grub* を使用する

トラステッドコンピューティング機能に対応する信頼済み GRUB を起動するようにします。

聴覚信号を有効にする

GRUB で聴覚信号を有効または無効に設定します。

グラフィカルメニューファイル

起動スクリーンを表示する際に利用するグラフィックファイルのパスを指定します。

シリアルコンソールを使用する

お使いのマシンがシリアルコンソールで操作するものである場合は、この項目を選択して COM ポートと速度を設定してください。詳しくは `info grub` または <http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal> (英語) をお読みください。

17.2.7 ブートローダの種類の設定

ブートローダのインストールタブでは、ブートローダの種類を選択することができます。openSUSE での既定は GRUB ですが、LILO または ELILO を使用したい場合は下記のような手順で変更します:

警告: LILO はサポート対象外です

LILO の使用はお勧めできません—openSUSE ではサポート対象外であるためです。特別な場合にのみお使いください。

手順 17.6 ブートローダの種類の設定

- 1 ブートローダのインストールタブを選択します。
- 2 まずはブートローダの項目で *LILO* を選択します。
- 3 ダイアログボックスが開いたら、下記のいずれかの処理を選択してください:

新しい設定を提示

YaST に対して、新しい設定を提案させます。

現在の設定を変換

YaST に対して、現在の設定を変換するよう指示します。設定の変換を行なうことで、設定のうちのいくつかが失われる場合があります。

新しい設定の作成

カスタムな設定を書き込みます。この処理は openSUSE のインストール時には選択できません。

ディスクに保存された設定の読み込み

/etc/lilo.conf ファイルに保存された設定を読み込みます。この処理は openSUSE のインストール時には選択できません。

4 OK を 2 回押して設定を保存します。

変換にあたっては、古い GRUB の設定がディスクに保存されます。これを使用し直す場合は、ブートローダの種類を GRUB に戻してから **変換前に保存した設定に戻す**を選択してください。この選択肢は、インストール済みのシステムでのみ利用できます。

注記: カスタムなブートローダ

GRUB でも LILO でもないその他のブートローダを使用したい場合は、ブートローダをインストールしないを選択してください。このオプションを選択する場合は、事前にお使いのブートローダの文書を良くお読みください。

17.3 Linux ブートローダのアンインストール

YaST では、Linux でのブートローダをアンインストールし、Linux がインストールされる前に存在していた通常の MBR に戻す機能も提供しています。インストール時に YaST は自動で元の MBR をバックアップしているため、必要に応じてそこから書き戻すことができるようになっています。

GRUB をアンインストールするには、YaST を起動してシステム > ブートローダを選択し、ブートローダモジュールを起動します。そこからさらにその他

> ハードディスクの *MBR* を復元する を選択し、確認のため はい、上書きします を押します。

17.4 起動 CD の作成

お使いのシステムを起動マネージャから起動する際に問題が発生した場合や、お使いのハードディスクにブートマネージャをインストールすることができない場合は、Linux を起動するために必要な全てのスタートアップファイルを含んだ、起動 CD を作成することもできます。この作業を行なうには、お使いのシステムに CD ライター (書き込み可能な CD ドライブ) が必要です。

GRUB を利用した起動 CD-ROM 作成を行なうには単純に、特別な形式の 第2 ステージである、stage2_eltorito と、カスタマイズした menu.lst ファイル (必要であれば) を用意するだけです。従来必要だった stage1 と stage2 は、もはや必要ではありません。

手順 17.7 起動 CD の作成

- 1 まずは ISO イメージを作成するためのディレクトリに移動します。たとえば: `cd /tmp`

- 2 下記のようにして GRUB 用のサブディレクトリを作成し、その中の iso ディレクトリに移動します:

```
mkdir -p iso/boot/grub && cd iso
```

- 3 それぞれ iso/boot/ ディレクトリに、カーネルと stage2_eltorito, initrd, menu.lst, message ファイルをコピーします:

```
cp /boot/vmlinuz boot/  
cp /boot/initrd boot/  
cp /boot/message boot/  
cp /usr/lib/grub/stage2_eltorito boot/grub  
cp /boot/grub/menu.lst boot/grub
```

- 4 まずは root (hdx, y) のように書かれている項目を root (cd) のように修正し、お使いの CD-ROM デバイスを 指し示すようにします。また、それぞれメッセージファイルやカーネル、initrd についても、/boot/message (メッセージファイル), /boot/vmlinuz (カーネル), /boot/

initrd (**initrd**) を指定します。全ての設定が完了すると、menu.lst は下記のような形になります:

```
timeout 8
default 0
gfxmenu (cd)/boot/message

title Linux
    root (cd)
    kernel /boot/vmlinuz root=/dev/sda5 vga=794 resume=/dev/sda1 \
    splash=verbose showopts
    initrd /boot/initrd
```

splash=verbose の代わりに splash=silent と指定すると、起動処理時に現われる起動メッセージの出力を抑止することができます。

5 あとは下記のコマンドで ISO イメージを作成します:

```
genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -iso-level 2 -input-charset utf-8 \
-o grub.iso /tmp/iso
```

6 できあがったファイル grub.iso を、お好きなツールを利用して CD メディアに書き込みます。ただし、ISO イメージをデータファイルとして書き込んだりせず、お使いのユーティリティで CD イメージとして書き込むオプションを指定してください。

17.5 グラフィカルな SUSE スクリーン

グラフィカルな SUSE スクリーンは、カーネルパラメータに vga=値 を指定した場合にのみ、1 つめのコンソールに表示されるものです。YaST を利用してインストールした場合、このオプションは選択した解像度とグラフィックカードにあわせて自動的に有効化されます。SUSE スクリーンを無効化したい場合は、下記の 3 種類の方法で無効化することができます:

必要に応じて SUSE スクリーンを無効化する方法

グラフィカルな画面を無効化するには、echo 0 >/proc/splash というコマンドを入力します。再度有効化したい場合は、echo 1 >/proc/splash と入力します。

既定で SUSE スクリーンを無効化する方法

お使いのブートローダの設定に、`splash=0` というカーネル パラメータを記入します。詳しい情報は第17章 ブートローダ *GRUB* (275 ページ) をお読みください。なお、テキストモードをお使いになりたい場合(古いバージョンでの 既定値)は、`vga=normal` と記入します。

完全に SUSE スクリーンを無効化する方法

新しくカーネルをコンパイルし、*framebuffer support* 内の *Use splash screen instead of boot logo* の オプション選択を外します。カーネル内でのフレームバッファサポートを無効化すると、スプラッシュスクリーンについても自動的に無効に設定されます。

警告: サポート対象外です

SUSE では、独自にコンパイルしたカーネルを利用して起動した場合、いかなる サポートをも提供いたしません。

17.6 トラブルシューティング

この章では、*GRUB* を利用して起動する際に発生しうる問題や、それらの問題に対する解決方法について概要を述べています。いくつかの問題は<http://ja.opensuse.org/SDB:SDB> (日本語) や http://en.opensuse.org/Portal:Support_database (英語) にあるサポートデータベースに記載されています。キーワード検索を行なう場合は、*GRUB* や *起動*、または *ブートローダ*などで 検索してください。

GRUB と XFS

XFS にはパーティションの起動ブロック内に 第 1 ステージ を保存しておくための領域がありません。そのため、ブートローダの配置場所としては、XFS パーティションを設定しないでください。この問題は、XFS 以外の ファイルシステムでフォーマットする個別の起動パーティションを作成することで、 解決することができます。

GRUB が GRUB Geom Error を報告する問題

GRUB はシステムを起動する際、接続されたハードディスクのジオメトリ情報(配置情報)を確認します。BIOS は時として矛盾した情報を提供することがあり、これによって GRUB は Geom Error を報告します。この場合は BIOS を更新して 解決してください。

また **GRUB** は、**BIOS** で登録されていない追加のハードディスクから **Linux** を起動しようとした場合にもこのエラーメッセージを表示します。ブートローダの第1ステージが見つかって正しく読み込めたものの、第2ステージが見つからない場合に発生します。このような問題が発生した場合は、新しいハードディスクを **BIOS** に登録することで解決することができます。

複数のハードディスクを含むシステムで起動できない問題

YaST はインストール時に、ハードディスクの起動順序を正しく判別しない場合があります。たとえば、**BIOS** での起動順序が **SCSI**, **IDE** の順であるにも関わらず、**GRUB** では **IDE** ディスクを **hd0** と認識し、**SCSI** ディスクを **hd1** と認識することがあります。

この場合は **GRUB** のコマンドラインを利用して、起動処理中にハードディスクの順序を修正してください。システムを問題なく起動できたら、**device.map** ファイルを編集し、恒久的に新しい割り当てを書き込んでください。その後、**/boot/grub/menu.lst** と **/boot/grub/device.map** にある **GRUB** のデバイス名を確認し、下記のコマンドでブートローダを再インストールします：

```
grub --batch < /etc/grub.conf
```

Windows を 2 台目のハードディスクから起動する方法

Windows のようなオペレーティングシステムでは、1 台目のハードディスクからのみ起動することができます。1 台目以外のハードディスクにそのようなオペレーティングシステムをインストールした場合は、関連するメニュー項目を編集することで、論理的な変更を行なうことができます。

```
...
title windows
    map (hd0) (hd1)
    map (hd1) (hd0)
    chainloader (hd1,0)+1
...
```

上記の例では、**Windows** が 2 台目のハードディスクから始まっていることを示しています。ハードディスクの論理的な順序は **map** コマンドで変更することができます。この仕組みは、**GRUB** のメニューファイルには影響しません。そのため、2 台目のハードディスクは **chainloader** に指定しなければなりません。

17.7 さらになる情報

GRUBに関する広範囲の情報は、<http://www.gnu.org/software/grub/> (英語) に記載されています。grub の info ページを読むこともできます。特定の問題について調べるには、<http://ja.opensuse.org/SDB:SDB> (日本語) または http://en.opensuse.org/Portal:Support_database/ (英語) にあるサポートデータベースから、「GRUB」キーワードを指定することで検索することもできます。

特殊なシステム機能

本章では、仮想コンソールやキーボードレイアウトに関する、様々なソフトウェアパッケージに関する情報を提供しています。それぞれ `bash`, `cron`, `logrotate` などのソフトウェアコンポーネントに関して言及していますが、これらは以前のリリース版から変更されたり 拡張されたりしているものであるためです。それらの変更が小規模であったり、重要度の低いものであったりした場合でも、これらのコンポーネントはシステムと強い結びつきがあるため、ユーザ側で既定の振る舞いを変更したい場合があります。また本章では、言語や国固有の設定 (I18N や L10N と呼ばれるもの) にも言及しています。

18.1 特殊なソフトウェアパッケージに関する情報

システム管理者や多くのユーザにとって、The programs `bash`, `cron`, `logrotate`, `locate`, `ulimit`, `free` などのプログラムはとても重要な存在です。またマニュアルページや `info` ページは、それぞれコマンドに関する情報源になっていますが、両方が常に利用できるというわけではありません。また、GNU Emacs は有名かつとても機能に富んだテキストエディタです。

18.1.1 bash パッケージと /etc/profile

bash は既定のシステムシェルです。ログインシェルとして使用した場合は、いくつかの初期化ファイルを読み込みます。また、bash は下記の順序でファイルを読み込みます:

1. /etc/profile
2. ~/.profile
3. /etc/bash.bashrc
4. ~/.bashrc

カスタムな設定は、~/.profile または ~/.bashrc のいずれかに対して行ないます。これらのファイルを正しく処理するには、それぞれ /etc/skel/.profile ファイルや /etc/skel/.bashrc ファイルを、ユーザのホームディレクトリにコピーしてください。また、システム更新を行なった後は、/etc/skel から設定をコピーすることをお勧めします。なお、下記のように実行すると、個人の設定を失わずに済みます:

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

上記のように実行した後、それぞれ個人の設定を *.old ファイルから反映し直してください。

18.1.2 cron パッケージ

事前に設定した時刻にコマンドを定期的かつ自動的に裏で実行させたい場合は、cron を使うのが便利です。cron は専用のタイムテーブル設定を持っていて、システムで設定されているものが存在するほか、必要であればユーザ側でも設定を行なうことができます。

cron のタイムテーブルは、/var/spool/cron/tabs ディレクトリ内に存在します。また、/etc/crontab にはシステム全体に反映される cron テーブルが書かれています。このファイルの場合はタイムテーブルとコマンドの間に、コマンドを直接実行させたいユーザのユーザ名を入力します。例18.1

「/etc/crontab の項目」 (303 ページ) の例では root が入力されています。なお、パッケージ固有のタイムテーブルは /etc/cron.d ファイル内に存在していて、上記と同様の書式になっています。詳しくは /etc/cron.d ファイルのマニュアルページ (man cron) をお読みください。

例 18.1 /etc/crontab の項目

```
1-59/5 * * * * root    test -x /usr/sbin/atrun && /usr/sbin/atrun
```

なお、crontab -e コマンドでは /etc/crontab ファイルを編集することはできません。このファイルはエディタで直接開いて編集し、保存しなければなりません。

また、/etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly の各ディレクトリには、多数のパッケージがシェルスクリプトをインストールします。これらの動作は /usr/lib/cron/run-crons が制御しています。/usr/lib/cron/run-crons は、メインテーブル (/etc/crontab) で定義されているもので、15 分間隔で動作するように設定されています。そのため、実行されずに無視されるかもしれない処理であっても、適切な日時に実行されるようになっています。

毎時間や毎日などのように、定期的にメンテナンススクリプトを実行したい場合は、/etc/crontab の項目を利用してタイムスタンプを定期的に除去してください(例18.2 「/etc/crontab: タイムスタンプファイルの除去」 (303 ページ) の例では、毎正時になる前に hourly ファイルを削除したり、毎日午前 2:14 に daily ファイルを削除したりしています)。

例 18.2 /etc/crontab: タイムスタンプファイルの除去

```
59 * * * * root    rm -f /var/spool/cron/lastrun/cron.hourly
14 2 * * * root    rm -f /var/spool/cron/lastrun/cron.daily
29 2 * * 6 root    rm -f /var/spool/cron/lastrun/cron.weekly
44 2 1 * * root    rm -f /var/spool/cron/lastrun/cron.monthly
```

もしくは、cron.daily が開始する時刻を、/etc/sysconfig/cron ファイル内の DAILY_TIME から設定することもできます。また、MAX_NOT_RUN の設定を行なうと、DAILY_TIME で指定した時刻に長時間コンピュータの電源を入れていなかった場合でも、毎日処理するタスクを処理するようになります。MAX_NOT_RUN で指定できる最大値は 14 日です。DAILY_TIME for a longer period of time. The maximum value of MAX_NOT_RUN is 14 days.

システムメンテナンスジョブは、それらをわかりやすくするために、複数のスクリプトとして配布されています。これらは `aaa_base` パッケージ内に含まれています。たとえば `/etc/cron.daily` には、`suse.de-backup`、`-rpmdb`、`suse.de-clean-tmp`、`suse.de-cron-local` が含まれています。

18.1.3 ログファイル: **logrotate** パッケージ

システムサービス (デーモン) には多くの種類があり、カーネルそれ自身のほか、システム状態を規則正しく記録したり、固有のイベントを記録したりなどの処理をログファイルに行ないます。そのため、管理者は定期的にその時点でのシステムの状態を確認できることになり、機能のエラーや失敗などを認識し、ピンポイントの精度でトラブルを解決することができます。これらのログファイルは通常、**FHS** で規定されている `/var/log` ディレクトリ内に保存され、日々増加していく形になります。`logrotate` パッケージは、これらのファイルの増加をコントロールするための手助けを行ないます。

`logrotate` の設定は、`/etc/logrotate.conf` で行ないます。また、`include` を指定すると、読み込むべき追加のファイルを指定することができます。ログファイルを生成するプログラムは、`/etc/logrotate.d` 内に個別の設定ファイルを生成します。上記のような設定を生成するものとして、たとえば `apache2` (`/etc/logrotate.d/apache2`) や `syslogd` (`/etc/logrotate.d/syslog`) などがあります。

例 18.3 /etc/logrotate.conf の設定例

```
# 詳細は "man logrotate" をお読みください
# ログファイルを週単位で切り替える
weekly

# 4 週間分までの過去分を保存する
rotate 4

# 古いログファイルに切り替わったら、新しい（中身の無い）ログファイルを生成する
create

# ログファイルを圧縮しておきたい場合、下記の行のコメントを外してください
#compress

# RPM パッケージは下記のディレクトリにログ切り替え情報を配置する
include /etc/logrotate.d

# lastlog や wtmp はパッケージ管理外であるため、独自に切り替える
#/var/log/wtmp {
#     monthly
#     create 0664 root utmp
#     rotate 1
#}

# システム固有のログは下記で設定します
```

logrotate は cron でコントロールするもので、毎日 /etc/cron.daily/logrotate から呼び出されます。

重要項目

create オプションは、/etc/permissions* ファイル内に管理者が作成した全ての設定を読み込みます。独自の修正を行なった場合は、矛盾が生じないように ご注意ください。

18.1.4 locate コマンド

locate コマンドはファイルを素早く検索するためのコマンドで、標準インストールの範囲には含まれていません。必要であれば findutils-locate パッケージをインストールしてください。また、updatedb プロセスは毎晩または起動後 15 分経過すると、自動的に開始されます。

18.1.5 ulimit コマンド

`ulimit` (*user limits* (ユーザ制限)) コマンドを利用すると、システム資源の使用について制限を設定することができるほか、設定内容を表示させることができます。`ulimit` は特に、アプリケーションに対するメモリ制限を設定するのに便利です。アプリケーションがシステム資源を多く使いすぎることを防ぎ、オペレーティングシステムがスローダウンしたりハングアップしたりすることを防ぐことができます。

`ulimit` コマンドには様々なオプションを指定することができます。メモリの使用量を制限したい場合は、表18.1「`ulimit`: ユーザに対する資源設定」(306 ページ)にあるオプションをお使いください。

表 18.1 `ulimit`: ユーザに対する資源設定

<code>-m</code>	最大常駐セットサイズ
<code>-v</code>	シェルに対して解放する仮想メモリの最大値
<code>-s</code>	スタックの最大サイズ
<code>-c</code>	作成するコアファイルの最大サイズ
<code>-a</code>	現在の制限全てを表示

システム全体の設定は、`/etc/profile` に指定することができます。ここではコアファイルの生成(デバッグ目的でプログラマが必要とするもの)を有効にしています。通常のユーザは、システム管理者が`/etc/profile` で設定した値を増やすことができませんが、独自の設定を`~/.bashrc` で指定することができます。

例 18.4 `ulimit`: `~/.bashrc` 内の設定

```
# 最大常駐セットサイズ (物理メモリ):
ulimit -m 98304

# 仮想メモリの制限:
ulimit -v 98304
```

メモリの割り当ては **KB** (キロバイト) 単位で指定しなければなりません。詳しい情報は、`man bash` をお読みください。

重要項目

全てのシェルが `ulimit` に対応しているわけではありません。 **PAM** (たとえば `pam_limits`) では、これらの制限を含む 広範囲の機能を提供しています。

18.1.6 free コマンド

`free` コマンドは、どれだけの **RAM** が現在使用中であるかを調べる目的であるように間違えるという意味で、誤解を生んでいます。そのような情報は `/proc/meminfo` から調べることができます。現在では、**Linux** のような構造の新しいオペレーティングシステムにアクセスしているユーザが、メモリについて心配する必要はありません。このような *利用可能な RAM* という考え方は、統合的なメモリ管理を行なうよりも前に存在した考え方です。空きメモリは無駄なメモリの発想は **Linux** にも当てはまるもので、その結果として **Linux** は空きまたは未使用のメモリを生じさせずにキャッシュのバランスをとる動作を行なっています。

基本的にカーネルはアプリケーションやユーザデータに関する直接の知識は持っていません。その代わり、アプリケーションやユーザデータを ページキャッシュ 内で管理しています。メモリが不足しはじめると、それらの一部はそれらがもともと `mmap` コマンド経由で読み込んでいたスワップパーティションやファイルに書き込まれるようになります (詳しくは `man mmap` をお読みください)。

また、カーネルには上記以外のキャッシュ機能があります。たとえばネットワーク アクセスのキャッシュ情報を保存する *slab* キャッシュ などがあります。これは `/proc/meminfo` 内のカウンタ間の違いを説明するもので、それらの多く (ただし全てではありません) は `/proc/slabinfo` からアクセスすることができます。

18.1.7 マニュアルページと info ページ

いくつかの GNU アプリケーション (たとえば tar など) では、マニュアルページのメンテナンスが行なわれていないものがあります。これらのコマンドの場合は、より詳しい手順の書かれている info ページの概要を、`--help` オプションで得ることができます。info ページは GNU のハイパーテキストシステムです。info ページ自身の情報については、`info info` と入力することで表示することができます。info ページは emacs から `emacs -f info` と入力することで閲覧できるほか、`info` と入力することで直接読むこともできます。これ以外にも `tkinfo`, `xinfo` やヘルプシステムを利用して閲覧することもできます。

18.1.8 man コマンドを利用したマニュアルページの選択

`man` マニュアルページと入力すると、通常はマニュアルページを直接読むことができます。ただし現在は、異なるセクションで同じ名前のマニュアルページが存在した場合、`man` はユーザに対して、どのセクションを読むのかを問い合わせるようになっています。ユーザが必要なセクションを入力することで、該当のマニュアルページが表示されます。

従来のような動作に戻りたい場合は、`~/.bashrc` などのシェル初期化ファイルに対し、`MAN_POSIXLY_CORRECT=1` を設定してください。

18.1.9 GNU Emacs の設定

GNU Emacs は複雑な作業環境です。本章では、GNU Emacs が起動する際に処理される設定ファイルについて述べています。本章よりも詳しい情報については、<http://www.gnu.org/software/emacs/> をご利用ください。

Emacs は起動時にいくつかのファイルを読み込み、ユーザやシステム管理者、もしくはディストリビュータのカスタマイズや事前設定を反映します。設定ファイル `~/.emacs` は `/etc/skel` からそれぞれのユーザのホームディレクトリにインストールされます。また `.emacs` は、`/etc/skel/.gnu-emacs` を読み出す構成になっています。プログラムをカスタマイズするには、`.gnu`

-emacs ファイルをホームディレクトリにコピー (cp /etc/skel/.gnu-emacs ~/.gnu-emacs) し、必要な設定を記入してください。

.gnu-emacs は ~/.gnu-emacs-custom ファイルを、カスタムファイルとして定義しています。Emacs 内で customize (カスタマイズ) オプションを利用して設定を行なうと、それらは ~/.gnu-emacs-custom 内に保存されます。

openSUSE では、emacs パッケージは /usr/share/emacs/site-lisp 内に site-start.el ファイルをインストールします。このファイルは、 ~/.emacs の初期化ファイルよりも先に読み込まれるため、その他のスクリプトとあわせて、Emacs のアドオンパッケージ (たとえば psgml) とともに配布される、特別な設定ファイルを自動で読み込むことができるようになっています。この種類の設定ファイルは /usr/share/emacs/site-lisp にも含まれ、suse-start- で始まるファイル名になっています。ローカルのシステム管理者は、システム全体の設定を default.el で指定することができます。

これらのファイルについてさらに詳しい情報は、Emacs の info ファイル内、*Init File* 以下にあります: [info:/emacs/InitFile](#) これらのファイルの (必要に応じて) 読み込みを無効化するための情報についても、この場所にあります。

Emacs のコンポーネントは、複数のパッケージに分割されています:

- emacs: 基本パッケージ。
- emacs-x11: X11 対応のプログラム (通常インストールされます)。
- emacs-nox: X11 に対応していないプログラム。
- emacs-info: info フォーマットで書かれたオンラインドキュメント。
- emacs-el: Emacs Lisp 形式の非コンパイルライブラリファイル。実行に際して必要となるものではありません。

- `emacs-auctex` (LaTeX), `psgml` (SGML と XML), `gnuserv` (クライアントおよびサーバ動作) など: 必要に応じて様々なアドオンパッケージをインストールできます。

18.2 仮想コンソール

Linuxはマルチユーザ・マルチタスクのシステムです。これらの機能はスタンダードアロンのPCシステムであっても便利なもので、テキストモードで6つの仮想コンソールが利用できるようになっています。それぞれは **Alt + F1** から **Alt + F6** までを利用して切り替えることができます。7番目のコンソールはXシステム用に予約されているもので、10番目のコンソールはカーネルのメッセージを表示するためのものです。 `/etc/inittab` ファイルを編集することで、コンソールの割り当てを増やしたり減らしたりすることができます。

Xシステムをシャットダウンせずにコンソールを切り替えたい場合は、**Ctrl + Alt + F1** から **Ctrl + Alt + F6** までの範囲でキーを押してください。Xシステムに戻るには、**Alt + F7** を押します。

18.3 キーボードマッピング

各プログラムによるキーボードマッピング機能を標準化するため、それぞれ下記のファイルに変更を行ないます:

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.emacs
/etc/skel/.gnu-emacs
/etc/skel/.vimrc
/etc/csh.cshrc
/etc/termcap
/usr/share/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/バージョン/site-lisp/term/*.el
```

これらの変更は `terminfo` の項目を使用するアプリケーションにだけ影響し、設定ファイルを直接変更する場合 (`vi`, `emacs` など) にのみ影響があります。システムに同梱されていないアプリケーションは、これらの既定値には影響されません。

X 環境下では、コンポーズキー (マルチキー) は `/etc/X11/Xmodmap` に説明されているとおりの方法で有効にすることができます。

さらなる設定は、X キーボード拡張 (XKB) を使用することで実現できます。この拡張は GNOME (gswitchit) や KDE (kxkb) でも使用されています。

ティップ: さらなる情報

XKB に関する詳しい情報は、`/usr/share/doc/packages/xkeyboard-config` ファイル (`xkeyboard-config` パッケージ内) に示されている文書をお読みください。

18.4 言語と国の設定

システムはかなり広い範囲で国際化対応が行なわれているため、各言語や国の要件に合わせて自由に設定することができます。言い換えれば、国際化 (I18N) を行なうことが地域化 (L10N) にもつながるということになります。それぞれ I18N と L10N は、それぞれの単語の始めと終わりの文字を示し、その間に省略された文字数を入れるという省略表現です (I18N=Internationalization, L10N=Localization)。

それぞれの国際化設定は、`/etc/sysconfig/language` ファイル内にある `LC_` で始まる変数に設定を行なうことで実現します。これらの変数では *各国語対応* だけでなく、*メッセージ (翻訳)*、*文字セット*、*並べ替え順序*、*日付と時刻*、*数字*、*通貨* などの領域にも適用されます。それぞれの分野は個別の変数を使用して別々に指定を行なうこともできますし、ファイル `language` から間接的に指定することもできます (詳しくは `locale` のマニュアルページをお読みください)。

`RC_LC_MESSAGES` (メッセージ)、`RC_LC_CTYPE` (文字分類)、
`RC_LC_COLLATE` (文字照合)、`RC_LC_TIME` (日付と時刻)、`RC_LC_NUMERIC` (数値)、`RC_LC_MONETARY` (通貨)

これらの変数は `RC_` の接頭辞を抜いた形でシェルに渡され、それぞれの範囲に適用されます。なお、関連するシェルプロファイルは下記に示しています。また、現在の設定は `locale` コマンドで表示できます。

RC_LC_ALL

この変数が設定されていれば、上記で説明した変数はそれぞれ上書きされます。

RC_LANG

上記の変数がいずれも設定されていない場合は、これが次点の候補となります。既定ではRC_LANGだけが設定されます。この変数は、それぞれ個別に設定するよりも簡単に利用できるようにするために設けられています。

ROOT_USES_LANG

yes または no の値を設定する変数です。no に設定した場合は、root ユーザは常に POSIX 環境で作業を行なうことになります。

また、これらの値は YaST sysconfig エディタから設定することもできます (詳しくは 16.3.1 項「YaST sysconfig エディタを利用したシステム設定の変更」(272 ページ)をお読みください)。値として設定する内容には、言語コードのほか国コードやエンコーディング、修飾子などを入れることができます。それぞれの要素はそれぞれ特殊文字で下記のようにつながります:

```
LANG=<言語>[_<国>].<エンコーディング>[@<修飾子>]
```

18.4.1 いくつかの例

言語と国については常にセットで指定してください。言語には ISO 639 に準拠した値を設定します。詳しくは<http://www.evertype.com/standards/iso639/iso639-en.html> と <http://www.loc.gov/standards/iso639-2/> をお読みください。国については ISO 3166 に準拠した値を設定します。詳しくは <http://www.din.de/gremien/nas/nabd/iso3166ma/codlstpl/en-listpl.html> をお読みください。

なお、設定可能な値の組み合わせについては、/usr/lib/locale ディレクトリ内をご覧ください。このディレクトリ以下のサブディレクトリとしてそれぞれ設定可能な組み合わせが存在しています。また、追加の説明ファイルについては、localedef コマンドを使用することで /usr/share/i18n ディレクトリ以下のファイルから作成することができます。説明ファイルは glibc-i18ndata パッケージの構成の一部です。たとえば en_US.UTF-8

(アメリカ英語) 向けの説明ファイルを作成するには、下記のように入力します:

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

```
LANG=en_US.UTF-8
```

上記は、インストール時にアメリカ英語を選択した場合に設定される言語の既定値です。他の言語を選択した場合は、その選択した言語に合わせて設定が作成されますが、UTF-8については常に有効になります。

```
LANG=en_US.ISO-8859-1
```

上記は言語を英語に、国をアメリカに設定し、文字セットをISO-8859-1に設定した場合の例です。この文字セットではユーロ記号に対応していませんが、UTF-8に対応していないプログラムを動作させる場合には便利な設定です。文字セットの定義文字列(この場合はISO-8859-1)は、**emacs**などのプログラム側で処理されます。

```
LANG=en_IE@euro
```

上記の例は、言語設定にユーロ記号を含めるよう明示的に設定している例です。この設定は、既にUTF-8ではユーロ記号が含まれることから、古い表現です。この設定はISO-8859-15に対応していてUTF-8には対応していないアプリケーションを動作させる場合にのみ有効です。

また、**SuSEconfig** は /etc/sysconfig/language 内にある変数を読み込み、必要な変更があれば /etc/SuSEconfig/profile と /etc/SuSEconfig/csh.cshrc にそれぞれ書き込みを行ないます。/etc/SuSEconfig/profile ファイルは /etc/profile から読み込まれたりソースファイルとして指定されたりします。また、/etc/csh.cshrc からは /etc/SuSEconfig/csh.cshrc を読み込みます。これにより、設定をシステム全体に行き渡らせることができるようになっています。

なお、ユーザ側でも自分用の設定を行なうことができます。これを行なうには、~/.bashrc に設定を行なってください。たとえばシステム側で設定されているen_USをプログラムのメッセージ表示に使いたくない場合は、上記のファイルにLC_MESSAGES=es_ESという行を追加すると、メッセージはスペイン語で表示されるようになります。

18.4.2 ~/.i18n 内でのロケール設定

お使いのシステム言語設定の既定値に満足できない場合は、`bash` スクリプトの書式で `~/.i18n` ファイルを作成することで、設定を変更することができます。`~/.i18n` ファイルは、`/etc/sysconfig/language` ファイルにあるシステムの既定値を上書きすることができます。それぞれ `RC_` の接頭辞を抜いて変数を指定してください。たとえば `RC_LANG` を上書きしたい場合は、`LANG` と指定して下記のように指定します:

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

18.4.3 言語サポートの設定

メッセージのファイル分類では、決まり事として関連する言語ディレクトリ (たとえば `en`) だけを次点候補として利用します。たとえば `LANG` を `en_US` に設定している環境の場合、メッセージファイルが `/usr/share/locale/en_US/LC_MESSAGES` 内に存在しなかったときは、`/usr/share/locale/en/LC_MESSAGES` ファイルを次点候補とします。

言語の候補は自由に設定することができます。たとえばブルトン語が存在すればそのメッセージを表示し、存在しない場合はフランス語で表示したい場合は下記のようになります:

```
LANGUAGE="br_FR:fr_FR"
```

また、ガリシア語／スペイン語／ポルトガル語の順で表示したい場合は下記のように設定します:

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

また必要であれば、ノルウェー語のニーノシュクとブークモールを試すような設定を行なうこともできます (さらなる候補として `no` も指定しています):

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

または

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

なお、ノルウェー語では LC_TIME もそれぞれ別々に扱われます。

なお、このような設定を利用する場合には1つだけ問題が発生する可能性があります。それは、文字間を区切るセパレータ (区切り子) の存在です。たとえば LANG 変数に2文字の言語コードだけを指定したような場合 (例: de)、glibc が使用する定義ファイルが /usr/share/lib/de_DE/LC_NUMERIC のようなディレクトリに置かれていると、読み出すことができなくなってしまう。この場合は、LC_NUMERIC に de_DE のような指定を行なってください。

18.4.4 さらなる情報

- *The GNU C Library Reference Manual* 内の「Locales and Internationalization」章をお読みください。この文書は glibc-info パッケージに含まれています。
- <http://www.cl.cam.ac.uk/~mgk25/unicode.html> から参照できる、Markus Kuhn 氏による *UTF-8 and Unicode FAQ for Unix/Linux*
- howto パッケージに含まれる、Bruno Haible 氏による *Unicode-Howto* (/usr/share/doc/howto/en/txt/Unicode-HOWTO.gz)

udev による動的なカーネルデバイス管理

19

カーネルには、システムの実行中に任意のデバイスを追加したり削除したりする機能が備わっています。デバイス状態の変化(デバイスが接続されたり削除されたり)はユーザ側に通知する必要がありますし、接続されて認識されるとすぐに設定を行なう必要があります。また、特定のデバイスを利用しているユーザが存在する場合は、そのデバイスの認識状態についても変化を通知する必要があります。udev では、`/dev` ディレクトリ内に存在するデバイス ノードファイルとシンボリックリンクについて、動的に管理するために必要なインフラストラクチャを提供しています。また、udev のルールはカーネルデバイスのイベント処理に外部ツールを接続する方法を提供するものです。たとえばカーネルのデバイス処理の一部として特定のスクリプトを追加で実行したり、デバイスの処理時に評価を行なう目的で追加データを要求したり取り込んだりするなど、デバイスの処理方法をカスタマイズすることができます。

19.1 `/dev` ディレクトリ

`/dev` ディレクトリ内にあるデバイスノードは、それぞれ 関連するカーネルデバイスに対して、アクセスを提供するためのものです。udev を利用することで、`/dev` ディレクトリは現在のカーネル状態を反映するようになります。各カーネルデバイスには1つの デバイスファイルが存在します。あるデバイスがシステムから取り外されると、デバイスノードが削除されます。

`/dev` ディレクトリの内容はテンポラリファイルシステム上に存在していて、全てのファイルはシステム起動時に作成されます。設計上、このディレクトリ内に手動でファイルを作成しても、システムを再起動すると ファイルは消

えてしまいます。関連するカーネルの状態に関わらず `/dev` ディレクトリ内に存在すべき固定のファイルやディレクトリについては、`/lib/udev/devices` ディレクトリ内に配置することができます。システム起動時に左記のディレクトリ内容は `/dev` ディレクトリに、`/lib/udev/devices` に存在したものと同一所有権設定とパーミッションのままコピーされます。

19.2 カーネルの `uevent` と `udev`

必要なデバイス情報は `sysfs` ファイルシステムから受け取る仕組みです。カーネルが検出して初期化した各デバイスに対して、デバイス名の付いたディレクトリが作成されます。このディレクトリにはデバイス固有の情報を 含む属性ファイルが入っています。

デバイスが追加されたり削除されたりするたびに、カーネルは `udev` に対して変更を通知するため `uevent` を送信します。`udev` デーモンは、起動時に `/etc/udev/rules.d/*.rules` ファイルから提供される全てのルールを読み込んで処理し、メモリ内に記憶します。ルールファイルを変更したり追加または削除したりした場合は、`udevadm control reload_rules` コマンドを利用して、デーモンに対してメモリ内の記憶を読み込み直すように指定することができます。同じことが `/etc/init.d/boot.udev reload` コマンドでも行なうことができます。`udev` のルールと書式について、詳しくは 19.6 項「`udev` ルールによるカーネル側デバイスイベント処理への影響」(321 ページ)をお読みください。

それぞれ受信したイベントは、提供されたルールセットとの適合処理を行います。各ルールは、追加したりイベントの環境キーを変更したりすることができるほか、作成するデバイスノードに対して特定の名前を要求したり、ノードを示すシンボリックリンクを追加したり、デバイスノード作成後に実行すべきプログラムを追加したりすることができます。ドライバ中枢部分の `uevent` は、カーネルの `netlink` ソケットから受信します。

19.3 ドライバ、カーネルモジュール、デバイス

カーネルのバスドライバは、デバイスに対する探査を行ないます。カーネルはデバイスを検出すると、それぞれに対して内部用のデバイス構造を作成し、ドライバの中枢部から `uevent` の形で `udev` デーモンに送信します。バスデバイスは特別に作成した ID の形で、自分自身がどんな種類のデバイスであるのかを識別します。通常、これらの ID は製造元 ID と製品 ID、およびサブシステム固有の値から構成されています。各バスはそれらの ID について独自の方針を持っていて、それらは `MODALIAS` と呼ばれます。まとめると、カーネルはデバイス情報を取得してそれらの情報から `MODALIAS` ID 文字列を生成し、イベントとともにその値を送信します。USB マウスの場合、たとえば下記のようになります:

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

各デバイスドライバには、処理可能なそれぞれのデバイスに対して、既知の別名を保持しています。その一覧はカーネルモジュールファイル自身に含まれる形になっています。 `depmod` プログラムは利用可能な全てのモジュールに対して ID の一覧を読み込み、カーネルの `/lib/modules` ディレクトリ以下に `modules.alias` ファイルを作成します。このような構造から、モジュールの読み込みは、`MODALIAS` を含んだ各イベントに対して `modprobe` を呼び出すだけの簡易な処理で実現できるようになっています。 `modprobe $MODALIAS` が呼び出されると、モジュールが提供する別名情報の一覧内を検索し、適合する項目があればそのモジュールを読み込むようになっています。これらの処理は全て `udev` から実行されます。

19.4 起動と初期デバイス設定

`udev` が起動する前の起動処理時に発生した全てのデバイスイベントは、読み込むことができません。これはこれらのイベントを処理するための仕組みがルートファイルシステム内に存在していて、その時点ではそれらを利用できないためです。このような損失をカバーするため、カーネルは `sysfs` ファイルシステム内の各デバイスディレクトリ内に `uevent` というファイルを提供しています。これらのファイルに `add` と書き込むことで、起動時に送信され失われてしまったものと同じイベントを再送することができます。 `/sys` 内に

ある全ての uevent ファイルに対して単純に繰り返して処理するだけで、デバイスノードを作成してデバイスの設定を実施するために必要な、全てのイベントを再送することができます。

たとえば、その時点ではドライバを用意することができない理由から、初期の起動処理で USB マウスの準備は行なわれません。そのためデバイス検出のイベントは失われ、それらのデバイスに対するカーネルモジュールの発見もできなくなってしまう。接続されている可能性のあるデバイスを手動で検索する代わりに、udev はルート ファイルシステムの準備が完了すると、カーネルが検出した全てのデバイスに対してイベントを要求し、USB マウスに対するイベントが再度送信されるようにしています。マウント済みのルートファイルシステム内にカーネルモジュールが見つかり、USB マウスが利用できるようになります。

ユーザ側からは、デバイスのコールドプラグ(システム起動時から接続すること)と稼働時のデバイス検出には確認できる違いはありません。両方のケースとも同じルールが適用され、同じ設定済みプログラムが実行されます。

19.5 udev デーモンの稼働監視

udevadm monitor プログラムを使用すると、ドライバ中枢部のイベントと udev のイベント処理タイミングについて視覚化を行なうことができます。

```
UEVENT[1185238505.276660] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UDEV   [1185238505.279198] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UEVENT[1185238505.279527] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UDEV   [1185238505.285573] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UDEV   [1185238505.305026] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UEVENT[1185238505.305442] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
UEVENT[1185238505.306440] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV   [1185238505.325384] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV   [1185238505.342257] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
```

UEVENT の行には、**netlink** を介してカーネルが送信した イベントを表示しています。UDEV の行には、完了済みの **udev** イベントハンドラを表示しています。タイミング情報はマイクロ秒単位で表示されます。UEVENT と UDEV の行の時間差は、**udev** がそのイベントの処理を行なうために費やした時間か、もしくは 関連していたり既に実行中だったりするイベントと同期を待ち合わせるため、実行を遅らせた時間を表わします。たとえばハードディスクのパーティションに対するイベントは、常にディスクそのもののイベントが完了するまで待機します。これは、ディスク全体の イベントが問い合わせるハードウェア情報に依存して動作するためです。

`udevadm monitor --env` コマンドを入力すると、完全な イベント環境を表示します:

```
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw
```

udev は **syslog** にもメッセージを送信します。**syslog** 上のどの場所にメッセージを送信するかを制御する **syslog** の既定 **priority** は、**udev** の設定ファイル `/etc/udev/udev.conf` で指定することができます。既に起動中のデーモンに対して **priority** を変更するように指定したい場合は、`udevadm control log_priority=レベル/番号` のように実行します。

19.6 udev ルールによるカーネル側デバイスイベント処理への影響

udev のルールは、カーネルがイベント自身に追加したプロパティ情報や、カーネルが **sysfs** 経由でエクスポートした情報であれば、どんなものにでも適合させることができます。また、ルールでは外部のプログラムから得られる追加情報を要求することもできます。また、それぞれのイベントは提供された全てのルールに対して適合性を確認します。全てのルールは `/etc/udev/rules.d` ディレクトリ内に配置します。

ルールファイル内のそれぞれの行には、少なくとも1対のキーと値が書かれています。キーには2種類のものがあります。それは適合キーと代入キーです。全ての適合キーがその値と適合するとそのルールが適用され、代入キーで指定された値が代入されます。適合ルールにはデバイスノードの名前のほか、ノードを指し示すシンボリックリンクやイベント処理の一部として指定したプログラムを指定することもできます。適合するルールが見つからない場合は、デバイスノードの作成には既定のデバイスノード名が使用されます。ルールの文法と提供されている適合キーまたは取り込みデータについては、それぞれ `udev` のマニュアルページで説明しています。下記の例では、`udev` のルール文法に関する基本的な内容を説明しています。下記の例は `/etc/udev/rules.d/50-udev-default.rules` ファイル内に存在するルールからの抜粋です。

例 19.1 `udev` ルールの例

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb/%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

`console` のルールには3種類のキーが含まれています: 1つは適合キー (`KERNEL`) で、残りの2つは代入キー (`MODE`, `OPTIONS`) になっています。`KERNEL` 適合ルールでは、デバイス一覧を検索して `console` の種類に該当する任意の項目を探します。厳密に一致した項目だけに対して適合と判断し、ルールを実行するように設定されています。また、`MODE` キーではデバイスノードに対して特別なパーミッションを設定しています。この場合、このデバイスの所有者に対して読み込みと書き込みの権限を付与しています。`OPTIONS` キーでは、この種類のデバイスに対して適用すべきルールの最後を指定しています。この種類のデバイスに対して、これ以降に何らかのルールが書かれていても、それらは無視されます。

次に `serial devices` のルールは `50-udev-default.rules` 内には現在はありませんが、説明を行なうには便利であるため記載しました。このルールには2つの適合キー (`KERNEL` と `ATTRS`) と1つの代入キー (`SYMLINK`) が含まれています。`KERNEL` キーは、`ttyUSB` の種類を持つ全てのデバイスを検

索します。ワイルドカード*を使用することで、このキーはこれらのデバイスの複数に適合するようになっていきます。2つめの適合キーである ATTRS は、任意の ttyUSB デバイスに対して、sysfs 内の product 属性 ファイルを読み込んで、特定の文字列が含まれているかどうかを確認しています。代入キー (SYMLINK) では、このデバイスのシンボリックリンクを /dev/pilot ディレクトリに追加するように指定しています。このキーで使用されている演算子(+=)は、udev に対してこの動作を追加で実行するよう指定しているもので、以前のルールやその後のルールで既にシンボリックリンクを設定済みであっても追加できるようにしているものです。このルールには2つの適合キーが存在するため、両方の条件に該当した場合にのみこのルールが適用されます。

また、printer のルールは USB プリンタを扱うためのもので、ルール全体を適用するかどうかを判断するための、2つの適合キー (SUBSYSTEM と KERNEL) が含まれています。3つ存在する代入キーでは、この種類のデバイスに対する命名 (NAME) とシンボリックリンクの作成 (SYMLINK)、およびこの種類のデバイスに対するグループ指定 (GROUP) を行なっています。KERNEL キー内には*というワイルドカードを含んでいるため、複数の lp プリンタデバイスに適合するようになっていきます。置換文字列はそれぞれ NAME と SYMLINK のキーで使用されていて、内部のデバイス名に置き換えるような作りになっています。たとえば1台目の lp USB プリンタであれば、/dev/usb1p0 のようになります。

さらに kernel firmware loader のルールでは、その実行時に udev に対して外部のヘルパースクリプトを起動して、追加のファームウェアを読み込むように指定しています。また、SUBSYSTEM の適合キーでは firmware サブシステムを対象としています。ACTION キーは、デバイスが firmware サブシステムに属しているかどうかを追加で判断するように指定しています。最後の RUN+= キーでは、読み込むべきファームウェアの場所を判断するため、firmware.sh スクリプトを実行するように指定しています。

いくつかの汎用ルールを下記に示します:

- 各ルールには1つ以上のキーと値の組み合わせが書かれていて、それらはカンマで区切ります。
- キーの操作は演算子で決定します。udev ルールは複数の演算子に対応しています。

- 実値として与える値は、引用符で括らなければなりません。
- ルールのファイル内の各行は1つのルールを表わすものです。1行よりも長いルールを記述する場合は、シェルで利用するのと同じ \ 記号を行末に書き込んで連続していることを宣言してください。
- `udev` のルールでは、シェル形式のパターンマッチを利用することができます。それぞれ `*`, `?`, `[]` を利用することができます。
- `udev` のルールは置換文字列に対応しています。

19.6.1 udev ルール内の演算子指定

キーを作成するにあたっては、いくつかの演算子の中から演算子を指定することになります。これは作成したいキーの種類に依存します。適合キーの場合は検索値に該当するかどうかの条件を指定します。適合キーには下記の演算子を利用します:

`==`

等しいかどうかを検証します。キーに検索パターンが含まれている場合、そのパターンに該当するもの全てを有効と見なします。

`!=`

等しくないかどうかを検証します。キーに検索パターンが含まれている場合、そのパターンに該当するもの全てを有効と見なします。

代入キーの場合は、下記の演算子を利用します:

`=`

キーに値を代入します。以前のルール処理で、そのキーに値の一覧が代入されていた場合は、キーはリセットされて単一の値が代入されます。

`+=`

項目の一覧を保持しているキーに対して、値を追加します。

`:=`

最終値を代入します。後のルールでの変更を許可しないようになります。

19.6.2 udev ルール内での置換文字列の使用

udev のルールでは、ブレースホルダや置換文字列を使用することができます。これらは他のスクリプトで行なうのと似たような仕組みになっています。下記に udev で使用できる置換文字列の一覧を示します:

`%r, $root`

デバイスディレクトリを表わします。既定では `/dev` です。

`%p, $devpath`

DEVPATH の値を表わします。

`%k, $kernel`

KERNEL の値か、もしくは内部デバイス名の値を表わします。

`%n, $number`

デバイス番号を表わします。

`%N, $templine`

デバイスファイルの一時名を表わします。

`%M, $major`

デバイスのメジャー番号を表わします。

`%m, $minor`

デバイスのマイナー番号を表わします。

`%s{attribute}, $attr{attribute}`

variable で指定した sysfs 属性の値を表わします。

`%E{variable}, $attr{variable}`

variable で指定した環境変数の値を表わします。

`%c, $result`

PROGRAM の出力を表わします。

`%%`

`%` 文字そのものを表わします。

\$\$

\$ 文字そのものを表わします。

19.6.3 udev 適合キーの使用

適合キーは、udev のルールを適用する前に判断すべき条件を記述するものです。それぞれ下記の適合キーを利用することができます：

ACTION

イベントの動作種類を表わします。add でデバイスを追加したときに、remove でデバイスを取り除いたときに適合と判断します。

DEVPATH

イベントデバイスのデバイスパスを表わします。たとえば
DEVPATH=/bus/pci/drivers/ipw3945 のように指定すると、ipw3945
ドライバに関連した全てのイベントを適合と判断するようになります。

KERNEL

イベント対象の内部 (カーネル) での名前を表わします。

SUBSYSTEM

イベント対象のデバイスについて、サブシステムを表わします。たとえば SUBSYSTEM=usb のように指定すると、USB デバイスに関連する全てのイベントに適合するようになります。

ATTR{filename}

イベント対象のデバイスについて、sysfs での属性を表わします。たとえば vendor 属性ファイル名に対して文字列が含まれているかどうかを確認するには、ATTR{vendor}=="On[sS]tream" のように指定します。

KERNELS

udev に対して、該当するデバイス名を検索するため上位のデバイスパスを検索するように指定します。

SUBSYSTEMS

udev に対して、該当するサブシステム名を検索するため上位のデバイスパスを検索するように指定します。

DRIVERS

udev に対して、該当するデバイスドライバを検索するため上位のデバイスパスを検索するように指定します。

ATTRS{ファイル名}

udev に対して、該当する **sysfs** 属性値を検索するため上位のデバイスパスを検索するように指定します。

ENV{key}

環境変数の値を表わします。たとえば **ENV{ID_BUS}="ieee1394"** のように指定すると、**FireWire** のバス ID に関連する全てのイベントに適合するようになります。

PROGRAM

udev に対して外部プログラムを実行するように指定します。適合と判断させるには、プログラムは **0** を返さなければなりません。プログラムが標準出力(**stdout**)に出力した内容は、**RESULT** キーから利用することができます。

RESULT

最後の **PROGRAM** 呼び出しに対して、出力文字列の適合処理を行いません。このキーは **PROGRAM** キーを指定したのと同じルール内で行なうか、もしくはそれ以降のルールで行なうことができます。

19.6.4 udev 代入キーの使用

上述の適合キーとは異なり、代入キーは適合すべき条件を表わすことはありません。代入キーは値や名前を代入したり、**udev** で管理されるデバイスノードに対する動作を指定したりします。

NAME

作成すべきデバイスノード名を表わします。あるルールでいったんノード名が設定されると、それ以降の全てのルールで **NAME** キーへの代入が無視されるようになります。

SYMLINK

作成すべきノードに関連づけるシンボリックリンク名を表わします。1つのデバイスノードに対して複数の適合ルールから、作成すべき複数のシン

ボリック リンクを追加することかできます。1つのノードに対して1つのルールで複数のシンボリックリンクを指定することもできます。この場合は、複数のシンボリックリンク名の間をスペースで区切ってください。

OWNER, GROUP, MODE

新しく作成するデバイスノードに対して設定する、パーミッションを指定します。設定した値は任意のルールで上書きすることができます。

ATTR{key}

イベントの発生したデバイスに対して書き込むべき、sysfs 属性の値を指定します。== 演算子を使用すると、このキーは sysfs 属性に対する適合キーとして解釈されます。

ENV{key}

udev に対して環境変数に値を設定するように指定します。== 演算子を使用すると、このキーは環境変数に対する適合キーとして解釈されます。

RUN

udev に対して、このデバイスに対して実行するプログラムの一覧にプログラムを追加するよう指定します。なお、このデバイスに対する 後続のイベント処理を止めないようにする目的で、実行するプログラムは非常に短い時間で完了するタスクにしてください。

LABEL

GOTO からジャンプすることのできるラベル和 指定します。

GOTO

udev に対して、複数のルールを飛ばしてラベルの位置まで移動するよう指定します。

IMPORT{種類}

値を外部プログラムの出力などのイベント環境に読み込みます。udev は複数の種類の値を取り込むことができます。種類を指定しない場合は、udev はファイルパーミッションの実行許可ビットをベースにして、種類を自分自身で判別しようとします。

- program を指定すると、udev は外部プログラムを実行して、その出力を取り込みます。

- `file` を指定すると、テキストファイルを取り込みます。
- `parent` を指定すると、`udev` に対して親デバイスから 保存されたキーを取り込むよう指定します。

WAIT_FOR_SYSFS

`udev` に対して、特定のデバイスに対する `sysfs` ファイルが作成されるまで待機するよう指定します。たとえば `WAIT_FOR_SYSFS="ioerr_cnt"` のように指定すると、`udev` は `ioerr_cnt` ファイルが作成されるまで待機します。

OPTIONS

OPTION キーにはそれぞれ下記の値を指定することができます:

- `last_rule` を指定すると、`udev` はそれ以降の全ての ルールを無視するようになります。
- `ignore_device` を指定すると、`udev` はこのイベントを 完全に無視するようになります。
- `ignore_remove` を指定すると、`udev` はデバイスに対して 後ほど発生するはずの取り外しイベントを無視するようになります。
- `all_partitions` を指定すると、`udev` はブロックデバイス 上の全ての利用可能なパーティションについて、デバイスノードを作成するようになります。

19.7 固定のデバイス命名

動的なデバイスディレクトリと `udev` ルールの仕組みにより、全てのディスクデバイス に対して、その認識順序や接続方法に依存しない固定の名前が存在するようになっています。カーネルが作成するそれぞれのブロックデバイスは、特定のバスやドライブ種類、ファイルシステムについて知っているツールを実行することで確認を行ないます。動的なカーネル提供のデバイスノード名とともに、`udev` ではそのデバイスを指し示すシンボリックリンクの形で、固定の構造を生成する仕組みになっています:

```
/dev/disk
|-- by-id
```

```

| |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B -> ../../sda
| |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part1 -> ../../sda1
| |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part6 -> ../../sda6
| |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part7 -> ../../sda7
| |-- usb-Generic_STORAGE_DEVICE_02773 -> ../../sdd
| `-- usb-Generic_STORAGE_DEVICE_02773-part1 -> ../../sdd1
|-- by-label
| |-- Photos -> ../../sdd1
| |-- SUSE10 -> ../../sda7
| `-- devel -> ../../sda6
|-- by-path
| |-- pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
| |-- pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
| |-- pci-0000:00:1f.2-scsi-0:0:0:0-part6 -> ../../sda6
| |-- pci-0000:00:1f.2-scsi-0:0:0:0-part7 -> ../../sda7
| |-- pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sr0
| |-- usb-02773:0:0:2 -> ../../sdd
| |-- usb-02773:0:0:2-part1 -> ../../sdd1
`-- by-uuid
    |-- 159a47a4-e6e6-40be-a757-a629991479ae -> ../../sda7
    |-- 3e999973-00c9-4917-9442-b7633bd95b9e -> ../../sda6
    `-- 4210-8F8C -> ../../sdd1

```

19.8 udev で使用するファイル

/sys/*

Linux カーネルから提供されている仮想ファイルシステムで、既知の全てのデバイスについて情報を公開しています。この情報は、デバイスノードを /dev 以下に作成する際に udev が使用します。

/dev/*

動的に作成されたデバイスノードと、/lib/udev/devices/* からコピーされた静的コンテンツの両方が含まれます。

下記のファイルやディレクトリは、udev を構成するために欠くことのできないものです:

/etc/udev/udev.conf

メインの udev 設定ファイルです。

/etc/udev/rules.d/*

udev イベント適合ルールです。

`/lib/udev/devices/*`

`/dev` に配置される静的なデバイスノードなどを表わすディレクトリです。

`/lib/udev/*`

`udev` ルールから呼び出されるヘルパープログラムです。

19.9 さらになる情報

`udev` の仕組みについてさらになる情報を得るには、下記のマニュアルページをお読みください:

`udev`

`udev` やキー、ルール、その他重要な設定周りの問題について、一般情報を提供しています。

`udevadm`

`udevadm` は `udev` の実行制御を行なうために使用し、カーネルイベントの要求やイベントキューの管理、およびシンプルなデバッグ機構を提供しています。

`udev`

`udev` イベント管理デーモンに関する情報が書かれています。

Bash と Bash スクリプト

今日、多くの人々が KDE や GNOME のようなグラフィカルユーザインターフェイス (GUI) を利用してコンピュータを操作しています。コンピュータは多くの機能を提供しているにもかかわらず、自動的な作業の実行となると、GUI の用途は限られたものになってしまいます。シェルは GUI に足りない部分を埋めるには良いもので、本章ではシェルまわりの概要について、特に bash 関連について述べています。

20.1 「シェル」とは？

伝統的な観点では、シェルとは Bash (Bourne again Shell) のことを指します。本章で「シェル」と記述した場合も同じで、Bash のことを指します。実際には Bash 以外にも多くのものがあります (ash, csh, ksh, zsh, ...) が、それぞれは異なる機能と特徴を持っています。他のシェルについて詳しい情報を得たい場合は、YaST 内でシェルを検索してください。

20.1.1 Bash の設定ファイルについての知識

シェルはそれぞれ下記の方法で起動することができます:

1. 対話的なログインシェルとして。マシンにログインする際に使用されている方法で、Bash に `--login` オプションを付けて起動するか、もしくは SSH でリモートコンピュータからログインしたときなどに使用することができるものです。

2. 「通常の」 対話シェルとして。 `xterm` や `konsole`, `gnome-terminal` などのツールを利用して開始した場合に当てはまります。
3. 非対話シェルとして。これはコマンドラインからシェルスクリプトを起動したりした場合に当てはまります。

どの種類のシェルを利用するのかによって、シェルはそれぞれ異なる設定ファイルを読み込む仕組みになっています。下記の表には、ログインシェル／非ログインシェルとして利用する場合に利用することのできる設定ファイルを示しています。

表 20.1 ログインシェルとして使用する際の *Bash* 設定ファイル

ファイル	説明
<code>/etc/profile</code>	このファイルを修正してはいけません。修正しても次のアプリケーション更新で修正内容が上書きされてしまいます！
<code>/etc/profile</code> <code>.local</code>	<code>/etc/profile</code> に対して何らかの設定を行ないたい場合は、このファイルに記述します。
<code>/etc/profile.d/</code>	プログラム固有のシステム全体向け設定ファイルが含まれています。
<code>~/.profile</code>	ログインシェルとして使用する場合に、各ユーザが設定したい項目があれば、ここに記述します。

表 20.2 ログインシェル以外で使用する際の *Bash* 設定ファイル

<code>/etc/bash.bashrc</code>	このファイルを修正してはいけません。修正しても次のアプリケーション更新で修正内容が上書きされてしまいます！
<code>/etc/bash.bashrc</code> <code>.local</code>	Bash だけに対して何らかの設定を行ないたい場合は、このファイルに記述します。
<code>~/.bashrc</code>	ユーザ単位での設定は、このファイルに記述します。

また、`Bash` では下記のファイルも使用します:

表 20.3 *Bash* 向けの特殊ファイル

ファイル	説明
<code>~/.bash_history</code>	以前に入力した全てのコマンドについて、一覧を保持しています。
<code>~/.bash_logout</code>	ログアウト時に実行する内容を記述します。

20.1.2 ディレクトリ構造

下記の表には、`Linux` システムにある最上位のディレクトリのうち、最も重要なものについて簡単な説明を示しています。また、ディレクトリに関するさらに詳しい情報と、重要なサブディレクトリについては、続く表で説明しています。

表 20.4 標準的なディレクトリ構造の概要

ディレクトリ	内容
<code>/</code>	ルートディレクトリと呼ばれるディレクトリです。ディレクトリのツリー (木) 構造の開始点です。
<code>/bin</code>	システム管理者と一般のユーザの両方が利用する、重要なバイナリファイル (コマンド) 群が含まれるディレクトリです。通常は <code>bash</code> などのシェルもこの中に含まれます。
<code>/boot</code>	ブートローダで利用するファイルなどが含まれるディレクトリです。
<code>/dev</code>	そのマシン固有のデバイス情報にアクセスするためのディレクトリです。
<code>/etc</code>	そのマシン固有のシステム設定ファイルが含まれるディレクトリです。

ディレクトリ	内容
/home	このシステムでアカウントを持つ全てのユーザに対して設定する、ホームディレクトリの開始点です。root のホームディレクトリについては例外的に /home ではなく /root にホームディレクトリが設定されています。
/lib	重要な共有ライブラリとカーネルモジュールが含まれるディレクトリです。
/media	リムーバブル (取り外し可能な) メディアのマウントポイントです。
/mnt	一時的にファイルシステムにマウントする際に利用するディレクトリです。
/opt	追加のソフトウェアパッケージをインストールするディレクトリです。
/root	スーパーユーザ root のホームディレクトリです。
/sbin	重要なシステムバイナリファイルが含まれるディレクトリです。
/srv	システムが提供する各種サービスのデータを保管するディレクトリです。
/tmp	一時 (テンポラリ) ファイル向けのディレクトリです。
/usr	読み込みのみのデータが含まれる、階層の開始点です。
/var	ログファイルなどの可変データが含まれるディレクトリです。
/windows	システムに Microsoft Windows* と Linux の両方がインストールされている場合にのみ利用できます。Windows 側のデータにアクセスできるディレクトリです。

また、下記の一覧にはより詳しい情報のほか、それらのディレクトリ内に存在するファイルやサブディレクトリの例を挙げています:

/bin

root や他のユーザが利用する、基本的なシェルコマンドが含まれています。このディレクトリに存在するコマンドとしては、ls,mkdir,cp,mv,rm,rmdir などがあります。また、/bin には openSUSE の既定のシェルである **bash** も含まれています。

/boot

ブートローダやカーネルなどの起動に必要なデータのほか、カーネルが通常のプログラムを起動する前に使用する各種のデータが含まれています。

/dev

ハードウェア部品を表わす、デバイスファイルが含まれています。

/etc

X ウィンドウシステムなどのプログラムを操作するのに必要な、各コンピュータ固有の設定ファイルが含まれています。たとえば/etc/init.d には、起動処理中に実行する各種スクリプトが含まれています。

/home/username

そのシステムにアカウントを持つそれぞれのユーザに対して、各ユーザの個人データを保持するディレクトリです。このディレクトリ以下にあるファイルは、そのファイルの所有者かシステム管理者だけが変更することができます。既定では、電子メールや個人のデスクトップ設定などが隠しファイルやディレクトリの形で保存されます。KDE ユーザの場合、デスクトップの個人設定は .kde4 のディレクトリに、GNOME ユーザの場合は .gconf 内にそれぞれ保存されます。隠しファイルについて、詳しくは 項「主な機能」(第6章 基本コンセプト, ↑スタートアップ)をお読みください。

注記: ネットワーク環境でのホームディレクトリ

ネットワーク環境で作業を行なっている場合、お使いのホームディレクトリは /home 以外のディレクトリに割り当てられている場合もあります。

`/lib`

システムを起動したり、ルートファイルシステム内でコマンドを実行したりする際に必要となる、重要な共有ライブラリが含まれています。共有ライブラリとは、**Windows** で言うところの **DLL** ファイルのことです。

`/media`

CD-ROM や **USB** メモリ、デジタルカメラ (**USB** 接続の場合) など、リムーバブルメディアに対するマウントポイントです。 `/media` は、お使いのシステムに内蔵されているハードディスクを除く、全てのドライブをマウントするための場所です。リムーバブルメディアを挿入したり 接続したりするとマウントされますので、ここからアクセスすることができます。

`/mnt`

一時的にファイルシステムをマウントしたい場合に利用するマウントポイントです。このディレクトリに `root` がマウントを行なう場合もあります。

`/opt`

サードパーティ製のソフトウェアをインストールするための予約されている領域です。オプションのソフトウェアや、巨大なアドオンプログラムパッケージがそれぞれ配置されます。

`/root`

`root` ユーザのホームディレクトリです。 `root` の個人的なデータがここに配置されます。

`/sbin`

`s` が示しているとおり、このディレクトリにはスーパーユーザ向けの各種ユーティリティが含まれています。 `/sbin` には、 `/bin` に追加する形で、システムの起動や復元、修復などの各種バイナリ (実行形式) ファイルが含まれています。

`/srv`

システムが提供する各種サービス、たとえば **FTP** や **HTTP** などに使用するデータが含まれています。

`/tmp`

このディレクトリは任意のプログラムから使用されるディレクトリで、一時的にファイルを作成する場所として使用しています。

重要項目: 起動時における /tmp のクリーンアップ

/tmp 内に保存されたデータは、システム再起動が行なわれると削除される可能性があります。これは /etc/sysconfig/cron などの設定に依存して決まります。

/usr

/usr は「ユーザ」の略ではなく、UNIX system resources (UNIX システムリソース) の略です。/usr 内にあるデータは Filesystem Hierarchy Standard (FHS) 仕様を満たす様々なコンピュータで共有できるもので、静的な読み込みだけを行なう目的のファイルが配置されます。このディレクトリには全てのアプリケーションプログラムが配置され、ファイルシステム内での第2の場所として機能しています。KDE4 と GNOME についてもこちらに配置されます。/usr には複数のサブディレクトリが含まれます。それぞれ /usr/bin, /usr/sbin, /usr/local, /usr/share/doc などがあります。

/usr/bin

一般ユーザから利用できるプログラムが含まれます、

/usr/sbin

システム用のデーモンやシステム管理者向けのプログラムなど、システムの起動について必ずしも必要ではないファイルが含まれます。

/usr/local

システム管理者が、ディストリビューションに依存しない拡張をインストールするためのディレクトリです。

/usr/share/doc

お使いのシステムに対する様々な文書ファイルやリリースノートを配置するディレクトリです。manual サブディレクトリには、このマニュアルのオンライン版が含まれています。また、お使いのシステムに複数の言語をインストールしている環境では、このディレクトリに異なる言語向けの各種マニュアルが含まれています。

また、packages サブディレクトリには、お使いのシステムにインストールされている各種ソフトウェアパッケージの文書が含まれています。パッケージごとに、下記のようなサブディレクトリが作成されています: /usr/share/doc/packages/パッケージ名 これらのディレクトリには、パッ

ページのREADMEファイルやいくつかのサンプル、設定ファイルや追加のスクリプトなどが含まれています。

なお、HOWTO をお使いのシステムにインストールしている場合は、`/usr/share/doc` ディレクトリには `howto` サブディレクトリが存在し、Linux ソフトウェアでの設定方法や操作方法に関連した追加の文書が含まれています。

`/var`

`/usr` には静的で読み込み専用のデータが保存されているのに比べ、`/var` にはシステムが動作している際に 随時書き込まれるファイルが含まれています。たとえばログファイルや スプール (一時保管) データなどが該当します。`/var/log/` ディレクトリにある最も重要なログファイルについて、詳しくは表「ログファイル」(↑スタートアップ)をお読みください。

`/windows`

お使いのシステムに Microsoft Windows と Linux の両方がインストールされている場合にのみ利用可能なディレクトリです。お使いのシステム内の Windows パーティション上にある全てのデータが含まれます。このディレクトリ内に対して書き込みができるかどうかは、Windows 側で使用しているファイルシステムの種類に依存します。FAT32 をお使いの場合は、このディレクトリ内のファイルは自由に読み書きすることができます。NTFS の場合でも openSUSE には書き込みサポートが付属しているため、読み込み／書き込みを行なうことができますが、使用しているドライバ NTFS-3g ファイルシステムの仕様により、いくつかの制限があります。詳しくは 34.4 項「同一のコンピュータにおける異なる OS 上のファイルへのアクセス」(628 ページ)をお読みください。.

20.2 シェルスクリプトの作成

シェルスクリプトは、様々な種類の処理を行なうのに便利な方法です:データの収集やテキストファイル内の単語またはフレーズの検索など、多くのことを 便利に行なうことができます。下記の例では、テキストを出力するための小さなシェルスクリプトを示しています:

例 20.1 テキストを出力するシェルスクリプト

```
#!/bin/sh ❶  
# Output the following line: ❷  
echo "Hello World" ❸
```

- ❶ 最初の行はシェバンと呼ばれるもので始まっています。この文字列(#!)は、このファイルがスクリプトであることを宣言する識別子になっています。スクリプトはシェバンに続いて指定されたインタプリタで実行されます。この場合は /bin/sh を実行します。
- ❷ 2行目はハッシュ記号で始まる行で、これはコメント行を表わしています。それが何をするのかわかりにくいような場合に、コメントを記載しておくのがよいでしょう。
- ❸ 3行目は内蔵コマンド echo を実行し、テキストを表示しています。

なお、スクリプトを実行する前に下記の要件を満たす必要があります:

1. 各スクリプトにはシェバンの書かれた行が含まれていなければなりません(上記の例では既に含まれています)。スクリプトにその行が存在しない場合は、インタプリタを手動で呼び出さなければなりません。
2. スクリプトはシェルから発見可能なディレクトリに配置します。シェルがスクリプトを発見するための処理は、環境変数 PATH にある情報から行ないます。通常は一般ユーザが /usr/bin など書き込むことはできないので、作成したスクリプトは ~/bin/ ディレクトリに保存するのがお勧めです。上記の例では、hello.sh のようなファイル名になっています。
3. スクリプトには実行パーミッションが付けられていなければなりません。下記のコマンドを実行すると実行パーミッションを設定することができます:

```
chmod +x ~/bin/hello.sh
```

上記の要件を全て満たすことができれば、下記の方法でスクリプトを実行することができます:

1. **絶対パスを指定した実行** 絶対パスを指定することでスクリプトを実行します。この例の場合、~/bin/hello.sh のように入力します。

2. 任意の場所からの実行 PATH 環境変数内にスクリプトを保存しているディレクトリが含まれる場合、単に `hello.sh` と入力することで実行することができます。

20.3 コマンドイベントのリダイレクト

各コマンドには3種類の入出力チャンネルが存在します:

- **標準出力** これが既定の出力チャンネルです。コマンドが何らかの出力を行った場合、標準出力チャンネルを利用して出力されます。
- **標準入力** コマンドがユーザまたは他のコマンドからの入力が必要とした場合、このチャンネルを通じて読み込みます。
- **標準エラー** コマンドはエラー報告用にこのチャンネルを使用します。

これらのチャンネルを他の箇所に転送(リダイレクト)するには、下記のような方法があります:

コマンド > ファイル

コマンドの出力データを指定したファイルに保存します。既存のファイルがあれば、そのファイルは上書きされます。たとえば `ls` コマンドの出力を `listing.txt` ファイルに出力したい場合は、下記のようになります:

```
ls > listing.txt
```

コマンド >> ファイル

コマンドの出力データを指定したファイルに追記します。たとえば `ls` コマンドの出力を `listing.txt` ファイルに追記したい場合は、下記のようになります:

```
ls >> listing.txt
```

コマンド < ファイル

指定したコマンドの入力データとしてファイルを指定します。たとえば `read` コマンドを利用してファイルの内容を変数 `a` に代入したい場合は、下記のようになります:

```
read a < foo
```

コマンド1 | コマンド2

左側のコマンドの出力を右側のコマンドの入力として利用するよう指定します。たとえば `cat` コマンドを利用して `/proc/cpuinfo` ファイルの内容を出力し、その出力に対して `grep` コマンドを適用し、`cpu` を含む行だけを取り出したい場合は、下記のようになります:

```
cat /proc/cpuinfo | grep cpu
```

また、各チャンネルには **ファイル記述子** が割り当てられています: **0** (ゼロ) が標準入力、**1** が標準出力、**2** が標準エラーです。 <や> 文字には、その前に **ファイル記述子** を指定することができるため、たとえば下記の例では `foo` で始まるファイルを検索しますが、標準エラー 出力はゴミ箱とも呼ばれる `/dev/null` に転送することで省略しています:

```
find / -name "foo*" 2>/dev/null
```

20.4 別名定義の使用

別名とは1つまたは複数のコマンドを短縮して定義するものです。覚えるのが面倒なコマンドや、多数のオプション指定を行なわなければならないコマンドを利用する場合に 便利な方法です。別名の書式は下記のとおりです:

```
alias 名前=定義
```

たとえば下記の例では `lt` という別名に対して、長い形式の出力 (`-l` オプション) と更新日次による並べ替え (`-t`)、および並べ替えを逆順で行なう設定 (`-r`) をそれぞれ指定しています:

```
alias lt='ls -ltr'
```

全ての別名定義を表示するには、`alias` コマンドを使用します。また、別名定義を削除するには `unalias` を使用します。

20.5 Bash 内での変数の使用

シェルでの変数はグローバルにもローカルにもすることができます。グローバル変数は環境変数とも呼ばれるもので、全てのシェル内でアクセスすることができます。逆にローカル変数は現在のシェル内ではしか見えないものになります。

全ての環境変数を表示するには、`printenv` コマンドを使用します。変数の値を表示するには、パラメータとして変数名を指定します:

```
printenv PATH
```

グローバルまたはローカルの変数を表示するには、`echo` コマンドを使用して参照することもできます:

```
echo $PATH
```

ローカル変数を設定するには、変数名の後ろにイコール記号を付け、その後ろに 値を書きます:

```
PROJECT="SLED"
```

イコール記号の前後にはスペースを入れないでください。スペースを入れるとエラーになってしまいます。環境変数を設定するには、`export` コマンドを使用します:

```
export NAME="tux"
```

変数を削除するには、`unset` コマンドを使用します:

```
unset NAME
```

お使いのシェルスクリプト内で利用できる、一般的な環境変数の一覧を示します:

表 20.5 便利な環境変数

HOME	現在のユーザのホームディレクトリ
HOST	現在のホスト名
LANG	ツールがローカライズ(各国語対応)されている場合、この環境変数の値を読み込んで言語を判断します。英語の場合は、値に <code>C</code> を指定することもできます。
PATH	シェルの検索パスを示します。ディレクトリの一覧をコロン区切りで 指定します。
PS1	各コマンド入力の前に表示される、通常のプロンプトを指定します。

PS2	複数行のコマンドを実行する際に表示する、2 行目以降のプロンプトを指定します。
PWD	現在の作業ディレクトリを表わします。
USER	現在のユーザを表わします。

20.5.1 パラメータの使用

たとえば、foo.sh というスクリプトが存在している場合、下記のように実行することができます:

```
foo.sh "Tux Penguin" 2000
```

お使いのスクリプトに対して渡された全てのパラメータにアクセスするには、位置パラメータを利用します。最初のパラメータが\$1、2 番目が\$2 などのようになっています。最大で9 つまでのパラメータを取得することができます。スクリプト名を取得したい場合は、\$0 をお使いください。

下記のfoo.sh スクリプトでは、1 番目から4 番目までに指定したパラメータを全て表示します:

```
#!/bin/sh
echo \"$1\" \"$2\" \"$3\" \"$4\"
```

上記の例のように実行した場合は、下記のように出力されます:

```
"Tux Penguin" "2000" "" ""
```

20.5.2 変数置換の使用

変数置換は、変数の中身について左から右にパターンを適用するための仕組みです。下記のような書式があります:

```
${VAR#pattern}
```

左から順に読み出し、最も短い適合パターンを取り除きます:

```
file=/home/tux/book/book.tar.bz2
echo ${file#*/}
home/tux/book/book.tar.bz2
```

`${VAR##pattern}`

左から順に読み出し、最も長い適合パターンを取り除きます:

```
file=/home/tux/book/book.tar.bz2
echo ${file##*/}
book.tar.bz2
```

`${VAR%pattern}`

右から順に読み出し、最も短い適合パターンを取り除きます:

```
file=/home/tux/book/book.tar.bz2
echo ${file%.*}
/home/tux/book/book.tar
```

`${VAR%%pattern}`

右から順に読み出し、最も長い適合パターンを取り除きます:

```
file=/home/tux/book/book.tar.bz2
echo ${file%%.*}
/home/tux/book/book
```

`${VAR/pattern_1/pattern_2}`

`VAR` で示される変数の内容について、`pattern_1` を `pattern_2` で置き換えます:

```
file=/home/tux/book/book.tar.bz2
echo ${file/tux/wilber}
/home/wilber/book/book.tar.bz2
```

20.6 コマンドのグループ化と組み合わせ

シェルでは条件付きの実行を行なうため、コマンドをつなぎ合わせたりグループ化したりすることができます。それぞれのコマンドの終了コードでその処理が成功したか失敗したかを判断します。0 (ゼロ) が返された場合、コマンドは成功したものと見なされ、それ以外の場合はコマンド固有のエラーになったものと判断されます。

下記の一覧では、どのようにコマンドをグループ化するかについて説明しています:

Command1 ; Command2

コマンドを順々に実行します。終了コードはチェックしません。たとえば下記の例では、`cat` コマンドでファイルの内容を表示し、`ls` コマンドでそのファイルの属性情報を表示します。終了コードとは無関係に実行されます:

```
cat filelist.txt ; ls -l filelist.txt
```

Command1 && Command2

左側のコマンドが成功した場合にのみ右側のコマンドを実行します (論理積)。たとえば下記の例では、ファイルの中身を表示したあと、その処理が成功した場合にのみファイルの属性情報を表示します (前の項目とは処理が異なっていることに着目してください):

```
cat filelist.txt && ls -l filelist.txt
```

Command1 || Command2

左図輪のコマンドが失敗した場合にのみ右側のコマンドを実行します (論理和)。たとえば下記の例では、`/home/tux/foo` ディレクトリの作成が失敗した場合にのみ `/home/wilber/bar` ディレクトリを作成します:

```
mkdir /home/tux/foo || mkdir /home/wilber/bar
```

funcname() { ... }

シェル関数を作成します。位置パラメータを利用することで関数のパラメータにアクセスすることができます。下記の例では、短いメッセージを表示する `hello` という関数を定義しています:

```
hello() { echo "Hello $1"; }
```

関数は下記のようにして呼び出すことができます:

```
hello Tux
```

上記を実行すると、下記のように表示されます:

```
Hello Tux
```

20.7 一般的なフロー制御

お使いのスクリプトについてフロー制御を行なうには、シェルに用意されている `while`, `if`, `for`, `case` の各コマンドを利用して組み立ててください。

20.7.1 if コントロールコマンド

if コマンドは条件分岐を行なうコマンドです。たとえば下記の例では、現在のユーザが **Tux** であるかどうかを確認しています:

```
if test $USER = "tux" ;then
    echo "Hello Tux."
else
    echo "You are not Tux."
fi
```

条件表現は複雑にも単純にもすることができます。下記の例では、foo.txt ファイルが存在するかどうかを判断しています:

```
if test -e /tmp/foo.txt ; then
    echo "Found foo.txt"
fi
```

test 表現は角括弧を利用して省略することもできます:

```
if [ -e /tmp/foo.txt ] ; then
    echo "Found foo.txt"
fi
```

便利な条件表現について、詳しくは <http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html> (英語) をお読みください。

20.7.2 コマンドの繰り返し処理の作成

for ループは、項目の一覧にあるものそれぞれに対して コマンドを実行するための仕組みです。たとえば下記の例では、現在のディレクトリ 内にあるそれぞれの PNG ファイルについて、いくつかの情報を表示することができます:

```
for i in *.png; do
    ls -l $i
done
```

20.8 さらなる情報

Bash について重要な情報は、マニュアルページ `man sh` で提供されています。特定のトピックについての詳細は、それぞれ下記をお読みください (それぞれ英語です):

- <http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html>—Bash Guide for Beginners
- <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>—BASH Programming - Introduction HOW-TO
- <http://tldp.org/LDP/abs/html/index.html>—Advanced Bash-Scripting Guide
- <http://www.grymoire.com/Unix/Sh.html>—Sh - the Bourne Shell

パート V. サービス

ネットワークの基礎

Linuxでは、様々なネットワーク構成に対応したネットワークツールと機能を提供しています。YaST を利用することで、ネットワークカードを利用した接続のほか、モデムなどのデバイスを設定することもできます。もちろん手動 設定も可能です。この章では、基本的なネットワーク機構と関連するネットワーク 設定について述べています。

Linux とその他の Unix オペレーティングシステムでは TCP/IP と呼ばれる プロトコルを使用しています。これは単一のプロトコルを指す言葉ではなく、様々なサービスを提供するネットワークプロトコルの集合体を意味する言葉です。2 台のマシンでデータをやりとりする場合は、表21.1「TCP/IP と関連プロトコル」(354 ページ)に示されているようなプロトコルが用いられます。また、このようにして TCP/IP を世界的に構築したネットワークのことを「インターネット」と言います。

RFC とは *Request for Comments* の略です。RFC では様々なインターネットプロトコルの説明やオペレーティングシステムでの実装方法のほか、アプリケーションについても言及している文書です。各種のプロトコルについて知識を深めるには、それぞれの RFC 文書をお読みください。RFC 文書は <http://www.ietf.org/rfc.html> から利用できます。

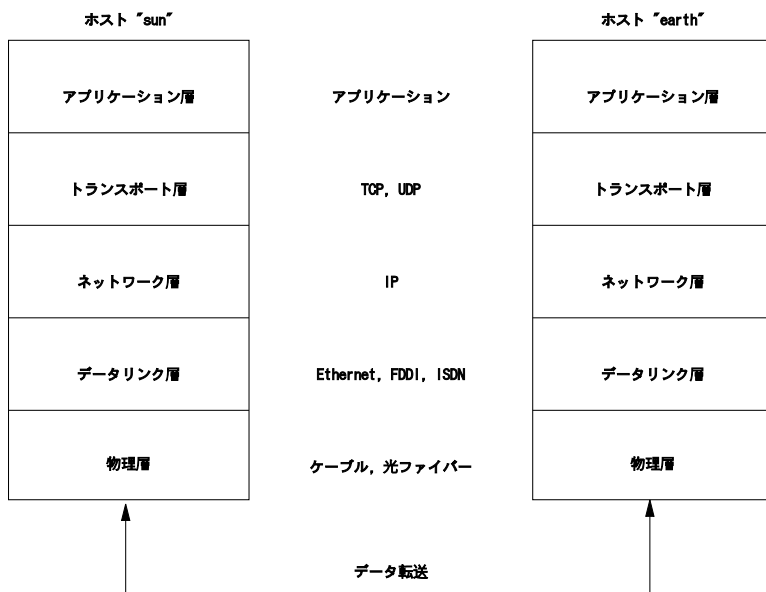
表 21.1 TCP/IP と関連プロトコル

プロトコル	説明
TCP	伝送制御プロトコル:接続という手順を踏んで利用する、信頼性のあるプロトコルです。データはアプリケーション側から連続データとして提供され、オペレーティングシステムで適切な形式に変換されます。宛先のホストで必要なアプリケーションに届けられる際には、元のアプリケーションが送信したものと同一データ形式に変換し直されます。また、TCP ではデータ転送中の喪失やエラーをチェックすることができます。TCP はデータの順序性を守るように実装されています。
UDP	ユーザデータグラムプロトコル:接続を行わない、信頼性の低いプロトコルです。アプリケーションから送信されたデータは、パケットと呼ばれる形式に変換されて送信されます。宛先のホストに届いたことは保証されないため、データ喪失が発生する場合があります。UDP はブロック単位でやりとりするアプリケーションに向けた仕組みで、TCP よりも少ない遅延で通信を行なうことができます。
ICMP	インターネットコントロールメッセージプロトコル:一般のユーザからは使用することのないプロトコルですが、エラー報告やTCP/IP のデータ転送に参加しているマシンの振る舞いについて、操作を行なうことができる特殊なプロトコルです。また、ping と呼ばれるプログラムを利用して、特別なエコーパケットを送信する場合にも使用します。
IGMP	インターネットグループ管理プロトコル:このプロトコルは、IP マルチキャストを実装するマシンについて、その振る舞いをコントロールする際に利用するプロトコルです。

図21.1「TCP/IPの簡略化モデル」(355ページ)に示されているとおり、データのやりとりには複数のレイヤ(階層)が用いられます。実際のネットワーク層はIP(インターネットプロトコル)と呼ばれ、信頼性のないデータ転送を提供するプロトコルです。このIPの上位にTCP(伝送制御プロトコル)を載せることで、データ転送の信頼性を確保しています。また、IP層は下位のハード

ウェア固有プロトコル(たとえばイーサネット)などに載せられて伝送されます。

図 21.1 TCP/IP の簡略化モデル

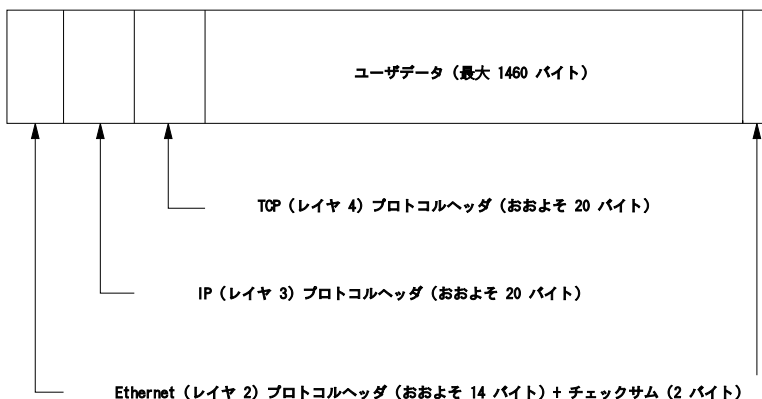


図ではそれぞれのレイヤに対して1つ2つ程度の例を示しています。階層は抽象レベルとも呼べるもので、最も低い階層がもっともハードウェアに近く、最も高い階層がハードウェアから最も遠いものになっています。また、それぞれの階層には独自の機能が備わっています。データリンク層と物理層はイーサネットのようなネットワークで表わされます。

ほぼすべてのハードウェアプロトコルは、パケットを基本にした構造を持っています。送信すべきデータは(一回ですべてを送信できない場合に)パケット単位でまとめられます。TCP/IP パケットでの最大長はおおよそ 64KB 程度ですが、ハードウェアプロトコルでのパケットはそれよりもずっと小さく、ネットワークのハードウェア側の仕様によって決まります。たとえばイーサネットの場合は、おおよそ 1500 バイト程度です。イーサネットを介して送信する場合、TCP/IP パケットのサイズはこの値が上限になります。さらなるデータ転送が必要な場合は、オペレーティングシステムからパケットを分割して送信する必要があります。

それぞれの層にはそれぞれ設計された機能があるため、その機能を実現するための情報をデータパケット内に保存しなければなりません。これをパケットのヘッダと呼びます。各層には、データ自身の前に小さなデータブロックの形でプロトコルヘッダが割り当てられます。イーサネット ケーブル内でやりとりされる TCP/IP データパケットの例を 図21.2 「TCP/IP イーサネットパケット」 (356 ページ) に示します。またチェックサムはパケットの冒頭ではなく、パケットの末尾に配置されます。これにより、ネットワークハードウェアでの処理が容易になるようにしています。

図 21.2 TCP/IP イーサネットパケット



アプリケーションがネットワーク上にデータを送信すると、物理層以外ではデータは Linux カーネルが実装する各層に渡されます。それぞれの層はそれぞれの役割を果たす処理を行ってから次の層に渡します。最も低い層がデータを送信する最終責任を負っています。データを受信したときの処理は、上述の逆の手順で行ないます。タマネギの皮のように、届いたデータから 1 枚 1 枚 プロトコルヘッダを取り除いていきます。アプリケーションに渡すデータの作成については、トランスポート層が最終責任を負います。この方法により、各層は隣接する層との間だけをやりとりすれば済むような構造になっています。またアプリケーションは、データが 100 メガビット毎秒の FDDI ネットワークを介しているのか、それとも 56 キロビットのアナログモデムを使用しているのかを気にする必要はありません。さらに言えば、パケットが必要なフォーマットに変換する機能を備えているため、同じく回線の種類についても気にする必要はありません。

21.1 IP アドレスとルーティング

本章ではIPv4 ネットワークについてのみ言及しています。IPv4の後継であるIPv6についての情報は、21.2項「IPv6—次世代のインターネット」(360 ページ)をお読みください。

21.1.1 IP アドレス

インターネット上に存在するすべてのコンピュータには、ユニークな32ビットのアドレスが付与されています。これら32ビット(または4バイト)のアドレスは、通常は例21.1「IP アドレスの書式」(357 ページ)の2行目のように記述します。

例 21.1 IP アドレスの書式

```
IP アドレス ( 2 進数表記): 11000000 10101000 00000000 00010100
IP アドレス (10 進数表記): 192. 168. 0. 20
```

10進数表記では、それぞれ4バイトを1バイトずつ10進数で表記し、間をピリオドで区切ります。IPアドレスはホストやネットワークインターフェイスに割り当てられるもので、世界中で唯一のものでなければなりません。このルールにはいくつかの例外事項がありますが、本章では特に関係のないものなので省略します。

IPアドレスでもう1つ重要な項目として、階層構造のシステムであることが挙げられます。1990年代ではIPアドレスはクラスと呼ばれる方法で限定して分類されていました。しかしながら、このような方法は柔軟性に欠けていたため、現在は使用されていません。現在はクラスレスルーティング(CIDR, クラスを使用しない領域間経路制御)を使用しています。

21.1.2 ネットマスクとルーティング

ネットマスクは特定のネットワーク範囲(サブネットワーク)を定義するために使用されるものです。2つのホストが同じサブネットワーク内に存在していれば、それらは直接通信を行なうことができます。異なるサブネットワーク内に存在している場合は、それぞれのホストは自分と同じサブネットワーク内にあるゲートウェイのアドレスを知って(設定して)おく必要があります。2つのIPアドレスから同じサブネットかどうかを判断するには、単純に

IP アドレスとネットマスクの「論理積」をそれぞれ計算します。結果が同じであればそれらの IP アドレスは同じサブネット内にあることになりますし、結果が異なっていればゲートウェイを介して通信しなければならないことを示します。

ネットマスクの動作について学ぶには、例21.2「IP アドレスとネットマスクのつながり」(358 ページ)をお読みください。ネットマスクは 32 ビットで表わされるもので、そのネットワーク内にどれだけの数の IP アドレスが属しているのかを表わしています。1 になっているビットがネットワークを表わすビットで、0 になっているビットがネットワーク内を表わすビットです。これを言い換えると、1 のビットが多ければ多いほどそのサブネットのサイズが小さくなることを示しています。ネットマスクは常に 1 のビットが連続するため、単にそのビット数を数えるだけの方法でネットマスクを表記する場合もあります。例21.2「IP アドレスとネットマスクのつながり」(358 ページ)の 1 つめの例では 24 ビット分が 1 になっているため、192.168.0.0/24 と記述する場合もあります。

例 21.2 IP アドレスとネットマスクのつながり

```
IP アドレス   (192.168.0.20):  11000000 10101000 00000000 00010100
ネットマスク (255.255.255.0):  11111111 11111111 11111111 00000000
```

```
-----
論理積の結果:                11000000 10101000 00000000 00000000
10 進数表記:                  192.      168.      0.      0
```

```
IP アドレス   (213.95.15.200): 11010101 10111111 00001111 11001000
ネットマスク (255.255.255.0):  11111111 11111111 11111111 00000000
```

```
-----
論理積の結果:                11010101 10111111 00001111 00000000
10 進数表記:                  213.      95.      15.      0
```

もう 1 つの例を示します: 同じイーサネットケーブルに接続されたすべてのマシンは、通常同じサブネットワーク内に属していて、直接接続することができます。サブネットがスイッチやブリッジで区切られていた場合でも、これらのホストは直接アクセスすることができます。

自分のサブネットとは異なる IP アドレスに対しては、目的のネットワークに接続可能なゲートウェイが存在する場合にのみ、通信を行なうことができます。もっともよくある例としては、外部とやりとりを行なうすべての通信を処理する 1 台のゲートウェイが設置されている例などが考えられます。もちろん、異なるサブネットに対してそれぞれ別々のゲートウェイを設定することもできます。

ゲートウェイを設定すると、すべての外部宛の IP パケットが適切なゲートウェイに送信されるようになります。このゲートウェイはパケットを受信すると、同じ方法で他のホストにパケットを転送します。これは宛先のホストに到達するか、もしくは TTL (time to live; 生存時間) が切れるまで行なわれません。

表 21.2 様々なアドレス

アドレスの種類	説明
通常のネットワークアドレス	これは例21.2「IP アドレスとネットマスクのつながり」(358 ページ)内に示されているアドレス(とネットマスク)です。このアドレスは、他のホストで同じものを設定することはできません。
ブロードキャストアドレス	これは端的に言うと、「そのサブネット内にあるすべてのホストにアクセスする」という意味になります。このアドレスを生成するには、ネットマスクの値を2進数で反転し、ネットワークアドレスとの論理和を計算してください。たとえば上記の例では192.168.0.255のようになります。このアドレスは特定のホストに設定することはできません。
ローカルホスト	アドレス 127.0.0.1 は各ホストでの「ループバックデバイス」に割り当てられています。このアドレスを利用することで、自分自身に対する接続を行なうことができます。IPv4でループバックデバイスのネットワークを正確に書くと、127.0.0.0/8 のアドレスになります。IPv6 の場合には単に1つのアドレス (:::1) となります。

IP アドレスは世界で唯一のものでなければならぬため、ランダムにアドレスを割り当てたりすることができません。そのため、プライベートな IP ベース ネットワークを構築する目的で、3 種類のアドレス領域が提供されています。これらのアドレスは、インターネット側から通信を行なうことができないように設定されているため、インターネット上では利用できません。これらのアドレス 範囲は RFC 1597 または 表21.3「プライベート IP アドレス 領域」(360 ページ)に 記されています。

表 21.3 プライベート IP アドレス領域

ネットワーク／ネットマスク	範囲
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x－172.31.x.x
192.168.0.0/255.255.0.0	192.168.x.x

21.2 IPv6—次世代のインターネット

WWW (World Wide Web) の出現により、インターネットは爆発的なまでに成長し、過去 15 年間で TCP/IP の通信を行なうコンピュータが急増しました。特に CERN (<http://public.web.cern.ch>) の Tim Berners-Lee が 1990 年に WWW を発明してから、インターネットに接続するホストは数千程度のものから 1 億程度にまで増加しました。

上述のとおり、IPv4 のアドレスは 32 ビット分しかありません。また、かなりの IP アドレスが失われています。これらの失われたアドレスは、ネットワークの管理上の問題から、使用することができません。これは、あるネットワークで利用可能なアドレス数が 2 のべき乗から 2 を引いた数でなければならないという制約に基づくものです。たとえばインターネットに 128 個のホストを接続する場合、256 個のサブネットを構築して 254 個のアドレスを確保しなければなりません。差し引かれた 2 個分は、ネットワークの構造そのものを管理するために必要なアドレスであるためです。これらはブロードキャストアドレス、ネットワークアドレスとそれぞれ呼ばれています。

また、現在の IPv4 プロトコルでは、DHCP や NAT (ネットワークアドレス変換) の仕組みを利用して、潜在的なアドレス不足を解決するようにしています。プライベートとパブリックという 2 種類のアドレスを組み合わせることで、これらの方法はアドレス不足の問題を和らげることができています。さらに、これらの問題は設定作業にも影響していて、設定を面倒にしているだけでなく、管理への負担を強いています。IPv4 ネットワーク内にホストを設定する場合、ホスト自身の IP アドレスとサブネットマスク、ゲートウェイアドレスと (必要であれば) ネームサーバのアドレスをそれぞれ設定しなければなりません。これらの全ての情報を知っておくか、もしくはどこかから取得しておく必要があることになります。

IPv6 では、アドレスの不足や複雑な設定項目といった厄介ごとが過去のものになっています。下記の章では、IPv6 でもたらされる改善点と利点、および古いプロトコルからの移行方法についてそれぞれ述べています。

21.2.1 利点

新しいプロトコルによってもたらされる最も重要、かつ最も目に見える改善点は、そのアドレス領域の拡張にあります。IPv6 アドレスは旧来の 32 ビットから、128 ビットに拡張されました。これにより、はるかに多いアドレスを割り当てることができるようになっています。

IPv6 アドレスは単に前身となるプロトコルから長さが増えただけのものではありません。IPv6 では内部構造が異なり、属しているシステムとネットワークに関する情報を多く含む仕組みになっています。これらの拡張について、詳しくは 21.2.2 項「アドレスの種類と構造」(363 ページ)をお読みください。

新しいプロトコルに関する他の利点を下記に列挙します：

自動設定

IPv6 はネットワークを「プラグ＆プレイ」対応にすることができます。これは新しく設定したシステムを (ローカルの) ネットワークに接続する際、手作業での設定を行なう必要がないことを意味しています。新しいホストは近隣のルータから *neighbor discovery* (ND) と呼ばれるプロトコルを利用して得られた情報から、自分自身のアドレスを割り出して設定します。この方法では管理者が介入することなくアドレスを設定することができるほか、アドレスの割り当てについて中央にサーバを用意する必要もありません。IPv4 では DHCP サーバとして中央にサーバを設置するか、もしくは ARP を利用して 169.254.0.0/16 のアドレス帯域を使用する必要がありました。

ルータがスイッチに接続されている場合でも、ルータはネットワーク内のホストに対して通信を行なうことができるよう、フラグ付きの定期通知を行なう必要があります。詳しい情報については RFC 2462 をお読みになるか、もしくは `radvd.conf` (5) のマニュアルページと RFC 3315 をそれぞれお読みください。

可動性

IPv6 では、同時に複数のアドレスを 1 つのネットワークインターフェイスに設定することができます。これにより、携帯電話会社が提供する国

際ローミングサービスに比べ、複数のネットワークへの接続が簡単になります:たとえばお使いの携帯電話を海外に持って行くと、電話機側がそれを自動で判断して海外のサービスに接続し、どこでも同じ番号を利用することができるようになるほか、電話機側から発信を行なっても自宅と同じ番号で発信できるようになります。

機密通信

IPv4 ではネットワークセキュリティが追加機能として提供されていました。IPv6 では IPsec を中枢機能のうちの1つに位置づけていて、インターネット上の第三者から盗み聞きされないよう、機密のチャンネルを設定して通信を行なうことができます。

後方互換性

現実的には、インターネット全体の IPv4 から IPv6 への移行は不可能です。そのため、インターネット上で両方のプロトコルが共存するだけでなく、特定のシステム上でも共存させなければなりません。これは互換アドレス (IPv4 アドレスから IPv6 アドレスへの変換) の形で提供されているほか、複数のトンネル機能で実現されています。詳しくは 21.2.3 項

「IPv4 と IPv6 の共存」(368 ページ)をお読みください。また、システムはデュアルスタック IP 技術を利用して、両方のプロトコルに同時対応しています。そのため、システム内では完全に区切られた2つのネットワークスタックが存在していて、両プロトコルバージョンの衝突は起こらないようになっています。

マルチキャストを利用したカスタムなサービス

IPv4 では SMB などのサービスで、ローカルネットワーク内の全ホストに一斉通知するためにブロードキャストを利用しています。IPv6 では、グループに属しているホストに対して通知を行なうマルチキャストを利用し、より洗練されたホスト/サーバ間の通知機能を実現しています(ブロードキャストの場合は全てのホスト宛に通知を行ないますし、ユニキャストの場合は個別に通知を行ないます)。それぞれのホストはアプリケーションごとのグループの形で管理されます。事前に定義されているグループとして、全てのネームサーバ(全ネームサーバのマルチキャストグループ)や全てのルータ(全ルータのマルチキャストグループ)などがあります。

21.2.2 アドレスの種類と構造

上述のとおり、現在の IP プロトコルには 2 つの重要な要素が欠けています: 1 つは IP アドレスの枯渇への対応、もう 1 つはネットワークの設定やルーティングテーブル (経路制御表) の複雑化と負担の増大です。IPv6 では前者の問題に対して、アドレス領域を 128 ビットに拡大することで対応しています。後者の問題に対しては、マルチホーミング (1 台の機器に対して複数のアドレスを割り当て、複数のネットワークへのアクセスを提供する技術) など、ネットワークアドレスの割り当てを上手に行なって、階層的なアドレス構造を構築することで対応しています。

IPv6 を取り扱う場合は、3 種類のアドレス種類があることを知っておく必要があります:

ユニキャスト

この種類のアドレスは、1 つのネットワークインターフェイスに対して割り当てられるものです。このようなアドレスを持ったパケットは、単一の宛先に対して配信されます。そのため、ユニキャストはローカルネットワークやインターネットでのホスト間通信に利用します。

マルチキャスト

この種類のアドレスは、複数のネットワークインターフェイスに対して割り当てられるものです。このようなアドレスを持ったパケットは、グループに属する複数の宛先に配信されます。マルチキャストアドレスは主に管理されたネットワーク環境で利用され、特定のネットワークサービスからホストのグループに対して通信を行なう場合に利用します。

エニーキャスト

この種類のアドレスは、複数のネットワークインターフェイスに対して割り当てられるものです。このようなアドレスを持ったパケットは、ルーティング (経路制御) プロトコルの方針に従って、送信元から最も近いグループのメンバーに対して配信されます。エニーキャストのアドレスは特定のネットワーク領域で利用されるもので、あるサービスを提供するサーバを見つけやすくするために用いられるものです。同じ種類のサーバは、全て同じアドレスを設定します。あるホストからサービスを要求すると、ルーティングプロトコルの判断で最も近い場所に存在するサーバがパケットを受信し、サービスを提供する仕組みです。サーバが何らかの理由でサービスを提供できない場合は、プロトコル側で自動的に次の候補を選択していきます。

IPv6 アドレスは 4 桁の項目 8 つから構成されています。それぞれの項目は 16 ビットで、16 進数による表記を行ないます。また、各項目はコロン (:) で区切ります。それぞれの項目内が 0 から始まる 桁であった場合にはその 0 を省略しますが、途中や最後が 0 であった場合には省略は行ないません。また、0 が連続する場合はダブルコロン (::) で省略を行ないます。ただし、ダブルコロンはアドレスごとに 1 回までしか使用できません。具体的な短縮表記方法を例21.3「IPv6 アドレスの例」(364 ページ)に示します。これらはいずれも同じアドレスを指す表記です。

例 21.3 IPv6 アドレスの例

```
fe80 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4  
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4  
fe80 :                                : 10 : 1000 : 1a4
```

IPv6 アドレスの各部分には機能が割り当てられています。最初の数バイトはプレフィクスとアドレス種類を規定しています。それに続くネットワーク アドレスの中間部分は、アドレスのネットワーク部分を表わします(使用されていない場合もあります)。最後にアドレスのホスト部分がつながる形になっています。IPv6 では、ネットマスクをアドレスの後にスラッシュ (/) を入れて記述します。このネットマスクはプレフィクスの長さで指定します。たとえば例21.4「プレフィクス長を指定する IPv6 アドレス」(364 ページ)のアドレスでは、64 ビット分がネットワークを表わす部分で、残りの 64 ビットがホストを表わす部分であることを示しています。言い換えれば、64 はネットマスクが左から 64 ビット分だけ 1 で埋められた値であることを示しています。ここから IPv4 のように IP アドレスとネットマスクから計算された値で論理積を計算すれば、同じサブネット上にあるものかどうかを判別 できることとなります。

例 21.4 プレフィクス長を指定する IPv6 アドレス

```
fe80::10:1000:1a4/64
```

IPv6 では、事前に規定されたプレフィクスが存在しています。これらのうちのいくつかを表21.4「様々な IPv6 プレフィクス」(365 ページ)に示します。

表 21.4 様々な IPv6 プレフィクス

プレフィクス (16 進)	定義
00	IPv4 アドレスと IPv4 over IPv6 で使用する互換アドレスです。これらのアドレスは IPv4 との互換性を維持するために使用され、IPv6 パケットを IPv4 パケットに変換することのできるルータで使用されます。なお、ループバックデバイス用などの特殊なアドレスについても、このプレフィクス内に存在します。
1 桁目が 2 または 3 で始まるもの	グローバルなユニキャストアドレスの集まりです。IPv4 の場合と同様に、特定のサブネットワークの一部としてインターフェイスに割り当てます。現時点では下記のアドレス領域が存在しています: 2001::/16 (商用品質のアドレス領域) と 2002::/16 (6to4 用のアドレス領域)
fe80::/10	リンクローカルのアドレスです。このプレフィクスを持つアドレスは経路制御されるべきものではないため、同じサブネット内でのみ通信可能です。
fec0::/10	サイトローカルのアドレスです。経路制御を行ってもかまいませんが、属する団体内でのみ行なわれるべきものです。実際には 10.x.x.x のように、プライベートネットワークアドレスの IPv6 版として使用することができます。
ff	マルチキャストアドレスです。

また、ユニキャストのアドレスは 3 つの要素から構成されています:

公衆部

最初の要素 (上述でのプレフィクス) は、公衆インターネットを介してパケットの経路制御を行なうための部分です。企業や団体がインターネット接続を行なう際の情報として使用します。

サイト部

2番目の部分には、パケットを配信するために必要なサブネットの経路情報が含まれています。

インターフェイス ID

3番目の部分では、パケットを配信する先のインターフェイス識別子が含まれています。この部分にはMACアドレスを設定することもできます。MACアドレスは世界で唯一のアドレスで、ハードウェアの製造元が割り当てた固定長の識別コードをデバイスに設定しています。実際には設定作業が単純化されていて、アドレスの最初の64ビットがEUI-64トークンと呼ばれています。その64ビットのうち最後の48ビットがMACアドレスを、残りの24ビットがトークンの種類に関する特殊な情報を含むものになっています。これにより、EUI-64トークンをMACアドレスを持たないデバイス、たとえばPPPやISDNなどをベースにしたデバイスに設定することができるようになっています。

上記の基本構造のほか、IPv6では下記の5種類のようなユニキャストアドレスが存在します:

:: (未設定のアドレス)

このアドレスは、起動した直後にインターフェイスを設定する際に利用するもので、発信元のアドレスとして使用します。インターフェイスを設定する段階ではアドレスを決めることができないため、このような形態をとっています。

:::1 (loopback)

これはループバックデバイスを表わします。

IPv4 互換アドレス

このIPv6アドレスはIPv4アドレスから生成されるもので、96個のゼロビットから始まるアドレスになっています。この互換アドレスはトンネリング(21.2.3項「IPv4とIPv6の共存」(368ページ)をお読みください)で使用するもので、IPv4とIPv6のホストが純粋なIPv4環境で通信できるようにするためのものです。

IPv6にマッピングされたIPv4アドレス

この種類のアドレスは、IPv6のアドレス表記でIPv4アドレスを表わすために使用します。

ローカルアドレス

この種類のアドレスには、2 種類の使用方法があります:

リンクローカル

この種類のアドレスは、内部のサブネットワーク内でのみ使用することができます。この種類のアドレスを宛先や発信元に持つパケットは、インターネットやその他のサブネットワークに転送されるべきではありません。このようなリンクローカルアドレスには特殊なプレフィクス (fe80::/10) が設定されていて、ネットワークカードのインターフェイス ID との間はゼロバイトで埋められた形になっています。また、この種類のアドレスは自動設定の過程で利用し、同じサブネットワークに属するホストと通信する際に利用します。

サイトローカル

この種類のアドレスはサブネットワーク内でやりとりをすることができますが、インターネットの世界では利用できないものです。つまり、企業や団体などの内部ネットワークでのみ使用しなければなりません。このようなアドレスはイントラネットなどで使用されるもので、IPv4 と言うところのプライベートアドレスと等価なアドレスです。このようなサイトローカルアドレスには特殊なプレフィクス (fec0::/10) が設定されていて、ネットワークカードのインターフェイス ID との間は 16 ビットのサブネットワーク ID を指定します。残りの部分はゼロバイトで埋めます。

また、IPv6 で全く新しく作成された機能として、各ネットワークインターフェイスに対して複数の IP アドレスを設定できる、という機能があります。これにより、同じインターフェイスから同時に複数のネットワークへアクセスできるようになっています。同時にアクセスするネットワークのうち、一方は IPv6 が有効化されるとすぐに利用できるリンクローカルアドレスを利用して、ローカルネットワーク内にある全ホストと通信のできる状態にし、さらに MAC アドレスと既知のプレフィクスを組み合わせで完全自動設定を行ないます。MAC アドレスがアドレスの一部に組み入れられているため、世界中でユニークな IP アドレスを構築することができるという仕組みになっています。アドレス内で変更可能な部分はそれぞれ サイト部 と 公衆部 を指定する部分で、これらはホストが現在接続しているネットワークに依存して決まるものです。

複数のネットワークの間を行き来するホストの場合は、少なくとも 2 つのアドレスを必要とします。それらのうちの一方はホームアドレスと呼ばれ、インターフェイス ID を含むだけでなく、通常所属するそのネットワーク自身

の識別子(およびそれに結びつくプレフィクス)が含まれています。ホームアドレスはアドレス設定を固定して使用するもので、通常の運用では変更を行なわないものとします。そのような環境でも、ホームネットワーク内かどうかに関わらず、移動端末のような接続で全てのパケットを配信することができます。これはIPv6で新たに提供されるようになった機能で実現されているもので、それぞれ *stateless autoconfiguration* (状態管理のない自動設定) や *neighbor discovery* (近隣検出) と呼ばれています。ホームアドレスに加え、移動端末ではローミング先の環境に合わせて1つまたはそれ以上の追加アドレスを取得します。これらは *care-of* (気付) アドレスと呼びます。ホームネットワークでは、その移動端末が外部でローミングしている間に、その端末に対してパケットを転送する機能を備えています。IPv6 環境では、この作業はホームエージェントが行なうもので、そのホームアドレス宛に届いたパケットをトンネル経由で中継します。他方ではこのような *care-of* アドレスに届いたパケットについては、そのまま何の加工もせずに届けることができるようになっています。

21.2.3 IPv4 と IPv6 の共存

インターネットに接続された全てのホストに対するIPv4からIPv6への移行作業は、ゆるやかに行なわれています。ホストによっては両方のプロトコルを共存させている場合もあります。単一のシステムにおけるプロトコルの共存は、両方のプロトコルに存在するデュアルスタックの実装によって作られています。ここで疑問点として残るのが、どのようにしてIPv6の利用できるホストからIPv4のホストに通信を行なうのか、およびIPv6パケットが現在のネットワーク、特にIPv4ベースのネットワークに対してどのような変換を行なうのかという点です。これらの問題を解決するのが、トンネリングと互換アドレスという仕組みです(詳しくは21.2.2項「アドレスの種類と構造」(363ページ)をお読みください)。

(世界中に広がった)IPv4のネットワークでは、IPv6のホストはトンネルを介して通信を行なうことができます: IPv6パケットはIPv4パケットの中にカプセル化されてIPv4のネットワーク内を移動します。このような2つのIPv4ホスト間の接続を、トンネルと呼びます。これを実現するには、パケットにはIPv6の宛先アドレスだけでなく、トンネルの反対側で受信処理を行なうIPv4のアドレスを設定しなければなりません。基本的なトンネルであれば、それは両方のホストの管理者が合意するだけで手動設定を行なうことができます。これを静的トンネルと呼びます。

しかしながら、静的なトンネルを設定したり管理したりすることは、日々の通信要件をまかなっていくには面倒な作業になってしまいます。そのため、IPv6では3種類の異なる方法で動的トンネルを設定する機能を提供しています:

6over4

IPv6 パケットを自動で IPv4 パケットにカプセル化し、マルチキャスト対応の IPv4 ネットワークを介して転送する方法です。IPv6 側から見ると、ネットワーク全体(インターネット)が巨大なローカルエリアネットワーク(LAN)のように見えます。これにより、IPv4 トンネルの反対側で自動的にパケットを受信できるようになるという仕組みです。しかしながら、この方法は規模が拡大する場合には向いておらず、IP マルチキャストがインターネット上でほとんど利用されていないことも、普及の妨げになっています。そのため、この方法は小規模の企業や学術的なネットワークなど、マルチキャストの利用できる環境に向いています。この方法の詳細な仕様は RFC 2529 で記載されています。

6to4

この方法では IPv4 アドレスを IPv6 のアドレスから自動生成し、IPv6 のホストからは IPv4 のネットワーク上で通信を行ないます。しかしながら、このような IPv6 ホストとインターネット間の通信にはいくつかの問題点が報告されています。この方法は RFC 3056 に記載されています。

IPv6 トンネルブローカー

この方法では、IPv6 ホスト向けのトンネル専用サーバを利用して通信を行ないます。詳しくは RFC 3053 をお読みください。

21.2.4 IPv6 の設定

IPv6 を設定する場合は、通常的环境であれば個別の端末に対して設定を行なう必要はありません。IPv6 は規定で有効化されています。インストール中であれば、「ネットワーク設定」(第1章 *YaST* を利用したインストール, ↑スタートアップ)にある ネットワーク設定の手順で無効化することができます。インストール済みのシステムで IPv6 を有効または無効に設定したい場合は、*YaST* から ネットワークの設定を選択します。グローバルオプションのタブから、必要に応じて *IPv6* を有効にする のチェックを入れるか外すかして設定してください。設定ファイルから IPv6 を設定する場合は、`/etc/modprobe.d/50-ipv6.conf` を編集してシステムを再起動してください。無効化している状態から、次の再起動までの間だけ一時的に有効化したい場合は、`root`

から `modprobe -i ipv6` を実行してください。なお、IPv6 モジュールを読み込んでしまうと、基本的に読み込みを外すことはできません。

IPv6 は自動設定を基本とした仕組みであるため、ネットワークカードにはリンクローカルネットワーク内のアドレスが割り当てられます。通常はワークステーション側で経路表 (ルーティングテーブル) の管理は行ないません。ワークステーションからネットワークルータに対して ルータ広告プロトコルを利用して問い合わせを行ない、設定すべきプレフィクスとゲートウェイを知る仕組みになっています。IPv6 ルータとして動作するには、このプロトコルに対応した `radvd` プログラムを利用します。このプログラムはワークステーションに対して、IPv6 アドレスのプレフィクスとルータのアドレスを通知します。また、両方の自動設定や経路制御を行なう目的で、`zebra/quagga` を使用することもできます。

また、`/etc/sysconfig/network` ファイルを利用して 様々な種類のトンネルを構築したい場合は、`ifcfg-tunnel (5)` のマニュアル ページをお読みください。

21.2.5 さらになる情報

ここまでの概要説明では不足している点が数多くあり、広範囲にわたる説明はできていません。新しいプロトコルに対するより深い説明については、下記のオンライン文書や書籍をお読みください:

<http://www.ipv6.org/> (英語)

IPv6 に関しては、ここからお読みになることをお勧めします。

<http://www.ipv6day.org> (英語)

独自の IPv6 ネットワークを構築するための全情報が提供されています。

<http://www.ipv6-to-standard.org/>

The list of IPv6-enabled products.

<http://www.bieringer.de/linux/IPv6/> (英語)

ここには Linux における IPv6 の HOWTO のほか、各トピックに関連した多数のリンクが掲載されています。

RFC 2640

IPv6 に関する基礎を説明する RFC です。

Silvia Hagen (ISBN 0-596-00125-8) 氏の著作で、IPv6 に関する重要な要素について、全ての説明が記されています。

21.3 名前解決

DNS は IP アドレスから 1 つまたは複数の名前に、および名前から IP アドレスにそれぞれ変換する機能を提供します。Linux からは、このような変換は専用のソフトウェア `bind` を利用してサービスを提供します。この変換を行なうマシンのことを、ネームサーバと呼びます。DNS における名前は、各要素をピリオドで区切った階層構造になっています。この名前の階層構造は、上述した IP アドレスの階層構造とは独立したものになっています。

`jupiter.example.com` のような完全な名前は、ホスト名・ドメインのような書式で記述します。完全な名前は **完全修飾ドメイン名** と呼び、ホスト名とドメイン (`example.com`) から構成されます。また、ドメイン名には **トップレベルドメイン (TLD) (com)** が含まれます。

TLD の割り当ては歴史的な理由により、かなり複雑な仕組みになっています。古くはアメリカで 3 文字のドメイン名が使用されていて、残りの国については 2 文字の ISO 国コードが標準として提供されていました。それに加えて、2000 年からは長い TLD も提供されるようになりました。これらはドメイン名の役割を示しているものです (たとえば `.info`, `.name`, `.museum` など)。

インターネットの黎明期 (1990 年以前) では、`/etc/hosts` ファイルを利用してインターネット上の全ホストの情報を保存していました。これはインターネットに接続するコンピュータの数が急増していくことで、現実的な解法ではなくなってしまいました。この理由により、世界中に分散させる方法で分散型データベースを構築し、ホスト名を保存する仕組みが開発されました。このデータベースはネームサーバに似た仕組みで、インターネット上の全てのホストに関するデータを持つことはなく、その代わりに他のネームサーバに対してリクエストを投げることができる仕組みを備えていました。

階層構造の最上位はルートネームサーバによって占有されています。これらのルートネームサーバはトップレベルドメインを管理するもので、ネットワークインフォメーションセンター (NIC) が稼働させているサービスです。各ルートネームサーバは、それぞれのトップレベルドメインに対するネームサーバ情報を知っていて、それらの情報を提供する役割を持っています。トップレ

ベルドメインの NIC について、詳しくは <http://www.internic.net> (英語) をお読みください。

DNS はホスト名を解決するだけのものではありません。ネームサーバは、特定のドメインに対して電子メールを受信するホストが何であるのかを知っている存在でもあります。このような情報を、メールエクスチェンジャ (*MX*) と呼びます。

お使いのマシンから IP アドレスの解決を行なうためには、少なくとも 1 台のネームサーバとその IP アドレスを知っておかなければなりません。YaST からはネームサーバを簡単に設定することができます。ダイアルアップ接続の場合は、ネームサーバの設定は手作業で行なう必要は全くありません。これは、ダイアルアッププロトコル上で、接続が完了したときに自動でネームサーバのアドレスを取得するためです。openSUSE® でのネームサーバの設定については、「ホスト名と DNS の設定」(383 ページ) をお読みください。ご自身でネームサーバを立ち上げる場合は、第 23 章 ドメインネームシステム (427 ページ) をお読みください。

whois プロトコルは DNS との関連性が高いプロトコルです。このプログラムを利用すると、指定したドメインの担当者の情報などを素早く得ることができます。

注記: MDNS と .local のドメイン名

.local トップレベルドメインは、DNS 解決器側でリンクローカルのドメインとして取り扱われます。DNS のリクエストは通常の DNS リクエストではなく、マルチキャスト DNS として送信されます。すでにお使いのネームサーバ設定で .local ドメインをお使いの場合は、`/etc/host.conf` ファイルでこのオプションを無効に設定しなければなりません。詳しい情報は `host.conf` のマニュアルページをお読みください。

また、インストール時に MDNS を使用しないように設定するには、起動パラメータに `nomdns=1` を指定してください。

マルチキャスト DNS について、詳しくは <http://www.multicastdns.org> (英語) をお読みください。

21.4 YaST を利用したネットワーク接続の設定

Linux では、多種のネットワークに対応しています。これらのうちの多くはそれぞれ異なるデバイス名を使用し、設定ファイルもファイルシステム内で様々な箇所に存在しています。手作業によるネットワーク設定については、21.6項「手動でのネットワーク設定方法」(398 ページ)をお読みください。

ラップトップコンピュータ上にインストールを行なっている場合(既定で NetworkManager が有効になっているはずです)は、YaST が検出された全てのデバイスに対して設定を行ないます。NetworkManager が有効になっていない場合は、リンクの確立している(ネットワークケーブルが接続されている)最初のデバイスだけを自動で設定します。その他のハードウェアについては、インストール済みのシステムから必要なときに設定を行なうことができます。下記の章では、openSUSE で対応している全てのネットワーク接続について、説明を行なっています。

21.4.1 YaST を利用したネットワークカードの設定

有線または無線ネットワークカードを YaST から設定するには、ネットワークデバイス > ネットワークの設定を選択してください。モジュールが起動すると、YaST は下記の 4 つのタブを含む ネットワーク設定ダイアログを表示します: グローバルオプション, 概要, ホスト名/DNS, ルーティング

グローバルオプションタブでは、NetworkManager を使用するかどうかや IPv6 を使用するかどうか、および DHCP オプション設定などの一般的なネットワーク設定を行ないます。詳しくは「グローバルネットワーク設定」(374 ページ)をお読みください。

概要タブでは、インストール済みのネットワークインターフェイスとその設定が表示されています。正しく検出された全てのネットワークカードを、その名前と共に表示しています。このダイアログから新しいネットワークカードを手動設定することもできますし、削除や変更などを行なうこともできます。自動では検出されていないネットワークカードを設定したい場合は、「未検出のネットワークカードの設定」(382 ページ)をお読みください。ま

た、既に設定済みのネットワークカードについて、設定を変更したい場合は、「ネットワークカードの設定変更」 (376 ページ) をお読みください。

ホスト名／DNS タブでは、お使いのマシンのホスト名と使用するネームサーバをそれぞれ設定します。詳しくは「ホスト名と DNS の設定」 (383 ページ) をお読みください。

ルーティングタブでは、経路制御 (ルーティング) に関わる設定を行ないます。詳しくは「ルーティング (経路制御) の設定」 (385 ページ) をお読みください。

図 21.3 ネットワークの設定

The screenshot shows the 'Network Configuration' window in YaST, with the 'Global Options' tab selected. The window has a title bar 'ネットワーク設定' and four tabs: 'グローバルオプション', '概要', 'ホスト名/DNS', and 'ルーティング'. The 'Global Options' tab contains three sections: 1. 'ネットワークの設定方法' (Network Configuration Method) with radio buttons for 'NetworkManagerを使ってユーザが制御 (U)' (selected) and 'ifupを使用した従来の方法 (T)'. 2. 'IP プロトコル設定' (IP Protocol Settings) with a checked checkbox 'IPv6 を有効にする'. 3. 'DHCP クライアントオプション' (DHCP Client Options) with a text field for 'DHCP クライアント識別子 (I)' (empty), a text field for '送信するホスト名 (H)' (set to 'AUTO'), and a checked checkbox 'DHCP で既定のルートを変更する'. At the bottom are buttons for 'ヘルプ (H)', 'キャンセル (C)', and 'OK (O)'.

グローバルネットワーク設定

YaST ネットワーク設定モジュール内の グローバルオプション タブでは、NetworkManager を使用するかどうかや IPv6 を使用するかどうか、および DHCP オプション設定などの重要なグローバルネットワーク設定を行ないます。これらの設定は、全ての ネットワークインターフェイスに対して適用されます。

ネットワークの設定方法では、ネットワークの接続を管理するための方法を選択します。NetworkManager のデスクトップアプレットを利用して全ての

インターフェイスの接続管理を行ないたい場合は、*NetworkManager* を使ってユーザが制御を選択してください。このオプションは複数の有線／無線ネットワークを切り替える場合に便利な選択です。デスクトップ環境(GNOMEやKDE)を使用しない環境であったり、お使いのコンピュータがXenサーバの仮想システムであったりした場合、もしくはお使いのネットワークに対してDHCPやDNSなどのネットワークサービスを提供する環境の場合は、*ifup*を使用した従来の方法を選択してください。*NetworkManager* を使用するよう選択すると、*nm-applet* を利用してネットワークオプションを設定するようになりますので、ネットワーク設定内の概要、ホスト名／DNS、ルーティングの各タブは利用できなくなります。*NetworkManager* についての詳しい情報は、第5章 *NetworkManager* の使用(↑スタートアップ)をお読みください。

IPv6 プロトコル設定では、*IPv6* プロトコルを使用するかどうかを設定します。*IPv4* と *IPv6* を同時に使用することもできます。既定では *IPv6* が有効に設定されていますが、*IPv6* を使用しないネットワーク環境では、*IPv6* プロトコルを無効化したほうが処理を高速に行なうことができます。*IPv6* を無効化するには、*IPv6* を有効にするオプションのチェックを外してください。これにより、*IPv6* のカーネルモジュールを自動では読み込まないようになります。設定はシステムの再起動後に反映されます。

DHCP クライアントオプションでは、*DHCP* クライアントのオプション設定を行ないます。*DHCP* クライアント識別子には、ネットワーク内の *DHCP* クライアントを唯一に識別する識別子を入力します。何も入力しなければ、既定でネットワークインターフェイスのハードウェアアドレスを利用します。ただし、お使いのマシンで同じネットワークインターフェイスを使用する複数の仮想マシンを利用する場合は、ハードウェアアドレスが全て同じものになってしまうため、ここにそれぞれを識別するための情報を入力してください。

送信するホスト名では、*dhcpcd* が *DHCP* サーバに送信するホスト名のオプションフィールドを指定します。*DHCP* サーバによっては、このホスト名情報を元にしてゾーン情報(正引きと逆引き)を更新したりすることがある(動的DNS) ためのものです。また *DHCP* サーバによっては、送信するホスト名のオプションフィールドに対して、何らかの文字列を設定する必要がある場合もあります。なお、この項目に *AUTO* を指定すると、現在のホスト名(/etc/HOSTNAME に設定したもの)を送信します。何も指定しない場合はホスト名を送信しないようになります。また、*DHCP* サーバから提供されたデフォルトゲートウェイ情報を利用しないようにするには、*DHCP* で既定のルートを変更するのチェックを外してください。

ネットワークカードの設定変更

ネットワークカードの設定を変更するには、YaST 内の **ネットワーク設定** > **概要** から設定したいネットワークカードを選択し、**編集** を押します。ネットワークカードの設定ダイアログが表示されたら、それぞれ **一般**、**アドレス**、**ハードウェアタブ** から設定を行ないます。無線 LAN カードの場合について、詳しくは 32.5 項「YaST を利用した設定」(596 ページ)をお読みください。

IP アドレスの設定

ネットワークカードの設定ダイアログ内の **アドレスタブ** を利用すると、ネットワークカードの IP アドレスを設定することができるほか、IP アドレスの決定方法を選択することもできます。また、このタブは IPv4 と IPv6 の両方のアドレスに対応しています。アドレスの設定方法は **アドレスを設定しない** (ボンド機能で使用します)、**固定 IP アドレス (IPv4 と IPv6)** を選択することができるほか、**可変 IP アドレス** を選択した場合は、さらにアドレスの設定方法を **DHCP**、**Zeroconf** のいずれか、もしくはその両方を選択することができます。

なお、**可変 IP アドレス** を選択した場合は、さらに **バージョン 4 のみでの DHCP (IPv4)**、**バージョン 6 のみでの DHCP (IPv6)**、**バージョン 4 と 6 の両方での DHCP (IPv4, IPv6 の両方)** を選択することができます。

また、インストール時にリンクの確立したネットワークカードが存在した場合、1 枚目のネットワークカードについては、**DHCP** を利用した自動アドレス取得が設定されます。ラップトップコンピュータの場合は **NetworkManager** が自動的に有効化され、全てのネットワークカードが設定されます。

DHCP の設定は、お使いのインターネット接続が **DSL** による接続契約になっていて、**ISP (インターネットサービスプロバイダ)** から固定の IP アドレスを割り当てられていない場合にも、使用場合があります。DHCP を使用する場合は、YaST のネットワーク設定モジュールから、**ネットワーク設定ダイアログのグローバルオプション内、DHCP クライアントオプション** で詳細を設定してください。同じインターフェイスを使用して、他のホストとも通信を行なう仮想ホスト設定を利用する場合は、**DHCP クライアント識別子の欄** に適切な値を入力し、これらを区別する必要があります。

DHCP はクライアントの設定には便利なオプションですが、サーバで使用する場合には適切ではありません。固定の IP アドレスを設定するには、下記のようにして行ないます:

1 YaST ネットワーク設定モジュールを起動し、表示されたダイアログ内で **概要** タブを選択します。すると検出されたカードの一覧が表示されますので、設定したいネットワークカードを選択して、**編集** を押します。

2 アドレスタブでは、**固定 IP** アドレスを選択します。

3 IP アドレスに設定したい IP アドレスを入力します。IPv4 と IPv6 の両方のアドレスを設定することができます。さらにサブネットマスクにも値を入力します。IPv6 アドレスを使用している場合は、/64 のような形でプレフィクス長を入力してください。

またオプションで、このアドレスに対する完全修飾ホスト名をホスト名に入力することができます。ここに設定を行なうと、/etc/hosts 設定ファイルに書き込みが行なわれます。

4 次へを押します。

5 設定を有効にするには、さらに **OK** を押します。

固定アドレスを使用する場合は、ネームサーバとデフォルトゲートウェイの設定は自動では行なわれません。ネームサーバを設定するには「ホスト名と DNS の設定」(383 ページ)の手順を、ゲートウェイの設定を行なうには「ルーティング(経路制御)の設定」(385 ページ)の手順をそれぞれお読みください。

別名の設定

1つのネットワークデバイスに対して複数のアドレスを割り当てることができます。これを別名と呼びます。

注記: 互換機能としての別名設定

別名と呼ばれる機能は IPv4 でのみ利用されるものです。IPv6 の場合はこれらの設定は無視されます。iproute2 を利用し、ネットワークインターフェイスに対して複数のアドレスを設定してください。

YaST を利用してネットワークカードに別名を設定するには、下記のようにして行ないます:

- 1 YaST ネットワーク設定モジュールを起動し、表示されたダイアログ内で **概要** タブを選択します。すると検出されたカードの一覧が表示されますので、設定したいネットワークカードを選択して、**編集** を押します。
- 2 アドレス > **追加** アドレス タブ内で、**追加** を押します。
- 3 それぞれ **別名**, **IP アドレス**, **ネットマスク** を入力します。別名にはインターフェイス そのものの名前は入力しないでください。
- 4 **OK** を押します。
- 5 続けて **次へ** を押します。
- 6 設定を有効にするには、さらに **OK** を押します。

デバイス名と udev ルールの変更

ネットワークカードのデバイス名は、使用の際に名前を変更することができます。udev でネットワークカードを認識する際に、ハードウェア (MAC) アドレスか、もしくはバス ID で識別を行ないます。大規模なサーバで使用する際には、カードの活線挿抜が簡単にできるようになりますので、後者のバス ID による識別をお勧めします。これらのオプションを YaST から設定するには、下記のようにして行ないます:

- 1 YaST ネットワーク設定モジュールを起動し、表示されたダイアログ内で **概要** タブを選択します。すると検出されたカードの一覧が表示されますので、設定したいネットワークカードを選択して、**編集** を押します。
- 2 ハードウェアタブを選択します。現在のデバイス名が **udev** ルール 内に表示されます。ここから **変更** を押します。
- 3 udev のカード識別方法を、**MAC Address** (MAC アドレス) または **Bus ID** (バス ID) から選択します。それぞれ現在選択中のカードに対する MAC アドレスとバス ID が表示されています。
- 4 デバイス名を変更する場合は、**デバイス名の変更** を選択し、名前を編集してください。
- 5 **OK** を押してから、**次へ** を押します。

6 設定を有効にするには、さらに **OK** を押します。

ネットワークカードのカーネルドライバの変更

ネットワークカードによっては、複数のカーネルドライバの中から選択を行なうことができるものがあります。カードが既に設定済みの場合は、**YaST** を利用して 利用可能なカーネルドライバ内から選択することができます。また、カーネルドライバに対するオプションを設定することもできます。**YaST** を利用してこれらの オプションを設定するには、下記のようにして行ないます:

- 1 **YaST** ネットワーク設定モジュールを起動し、表示されたダイアログ内で **概要** タブを選択します。すると検出されたカードの一覧が表示されますので、設定したいネットワークカードを選択して、**編集** を押します。
- 2 **ハードウェア** タブに移動します。
- 3 まずは **モジュール名** 内で、使用するカーネルドライバを 選択します。また、**オプション** の欄には必要なオプションを 入力します。オプションは **オプション=値** の形式で入力してください。複数のオプションはスペースで区切ります。
- 4 **OK** を押してから、**次へ** を押します。
- 5 設定を有効にするには、さらに **OK** を押します。

ネットワークデバイスの有効化

ifup を利用した従来の方法を利用する場合には、お使いのデバイスを起動時に 有効にすることもできますし、ケーブル接続時やカードの検出時に有効にしたり、手動による有効化を行なったり、さらには全く有効にしないこともできます。デバイスの起動方法を変更するには、下記のようにして行ないます:

- 1 **YaST** ネットワーク設定モジュールを起動し、表示されたダイアログ内で **概要** タブを選択します。すると検出されたカードの一覧が表示されますので、設定したいネットワークカードを選択して、**編集** を押します。
- 2 **一般** タブを選択し、**デバイスの有効化** 内から起動方法を選択します。

起動時を選択すると、システム起動時に該当デバイスを有効化します。ケーブル接続時を選択すると、インターフェイスに対して物理的な接続が行なわれているかどうかを監視するようになります。ホットプラグを選択すると、利用可能な状態になるとすぐに該当デバイスを有効化します。これは起動時に似た設定ですが、システム起動時にインターフェイスが存在していなくてもエラーにならない、という点で異なっています。手動の場合は、`ifup` コマンドを利用して手動でインターフェイスを有効化する意味になります。また、開始しないを選択すると、デバイスを全く有効化しなくなります。最後にある *NFSroot* は起動時に似たオプションですが、`rcnetwork stop` コマンドを入力しても、インターフェイスが停止されないようになります。この設定は、ルートファイルシステムに *NFS* や *iSCSI* をお使いの場合に ご利用ください。

3 次へを押します。

4 設定を有効にするには、さらに *OK* を押します。

通常はシステム管理者のみが、ネットワークインターフェイスを有効にしたり無効にしたりすることができます。*Kinternat* を利用して、一般のユーザからでもインターフェイスを操作できるようにしたい場合は、*Kinternat* を利用して *root* 以外のユーザにもデバイス操作を許すを選択してください。

最大転送単位 (MTU) の設定

インターフェイスに対して、最大転送単位 (MTU) を設定することができます。MTU はパケットの最大サイズをバイト単位で設定するもので、大きく設定すればするほど高速な通信回線での効率を上げることができます。しかしながら、大きすぎる MTU を設定すると、次のパケットを送信するまでに時間がかかってしまい、かえって速度が遅くなってしまう場合もあります。

1 YaST ネットワーク設定モジュールを起動し、表示されたダイアログ内で概要タブを選択します。すると検出されたカードの一覧が表示されますので、設定したいネットワークカードを選択して、編集を押します。

2 一般タブを選択し、MTU を設定の一覧から、適切な値を選択します。

3 次へを押します。

4 設定を有効にするには、さらに *OK* を押します。

ファイアウォールの設定

項「Configuring the Firewall with YaST」(第14章 *Masquerading and Firewalls*, ↑セキュリティガイド)で示されているような詳細なファイアウォール設定を行なわなくても、デバイス設定の一部として基本的なファイアウォール設定を行なうことができます。具体的には下記のようにして行ないます:

- 1 YaSTを開いて ネットワークデバイス> ネットワーク設定モジュールを起動します。概要タブを選択し、検出されたカードの中から設定したいカードを選択して、編集を押します。
- 2 ネットワークカードの設定ダイアログから、一般タブを選択します。
- 3 インターフェイスに設定したいファイアウォールゾーンを選択します。下記のオプションを選択することができます:

ファイアウォールを無効にする

このオプションは、ファイアウォールが無効化されている場合にのみ表示される項目で、ファイアウォールを全く起動しない場合に選択します。このオプションは、別途に存在するファイアウォールによってネットワーク全体が保護されている場合にのみ、選択してください。

内部ゾーン (保護しない)

ファイアウォールを起動するものの、このインターフェイスに対しては特に何のルールも適用しない場合に選択します。このオプションは、別途に存在するファイアウォールによってネットワーク全体が保護されている場合にのみ、選択してください。また、マシンに複数のインターフェイスが存在していて、選択したインターフェイスが内部ネットワークに接続されている場合にも便利な選択です。

非武装ゾーン

非武装ゾーンとは内部ネットワークと(敵が攻めてくる)インターネットの中間に位置する、追加の防御線を意味するものです。このゾーンの先に接続されたホストは、内部ネットワークとインターネットの両方からアクセスすることができますが、それらのホストから内部ネットワークにはアクセスできません。

外部ゾーン

ファイアウォールをこのインターフェイス上で動作させ、攻撃に対する完全な防御を設定します。この値が既定値になっています。

- 4 次へを押します。
- 5 設定を有効にするには、さらに *OK* を押します。

未検出のネットワークカードの設定

お使いのネットワークカードによっては、正しく検出できない場合があります。この場合、検出済みのカード一覧に表示されなくなってしまうます。お使いのシステムに適切なドライバが含まれている場合には、設定を手動で行なうこともできます。また、ブリッジやボンド、TUN, TAP などの特殊な種類のネットワーク デバイスを設定することもできます。未検出のネットワークカード (または特別なデバイス) を設定するには、下記のようにして行ないます:

- 1 YaST から ネットワークデバイス > ネットワーク設定 > 概要を開き、追加を押します。
- 2 ハードウェアダイアログ内では、まず デバイス種類と 設定名を選択します。設定しようとしているネットワークカードが PCMCIA や USB のデバイスである場合は、それぞれチェックボックスで選択を行ない、次へを押します。それ以外のネットワークカードの場合、必要であればそのカードに使用するカーネルの モジュール名を選択し、 オプションを入力します。

Ethtool オプションの欄には、インターフェイスを起動する際に *ifup* が使用する *ethtool* 向けのオプションを入力します。利用可能なオプションの一覧については、*ethtool* のマニュアルページをお読みください。また、オプション文字列が - で始まる場合 (たとえば -K インターフェイス名 rx on) は、文字列内の 2 番目は現在のインターフェイス名に置き換えられます。それ以外の場合 (たとえば autoneg off speed 10)、*ifup* は -s インターフェイス名として扱います。

- 3 次へを押します。
- 4 一般, アドレス, ハードウェアの各タブを利用して、IP アドレスやデバイスの有効化、そのインターフェイスに適用するファイアウォールゾーンなどをそれぞれ設定します。これらの設定オプションについて、詳しくは「ネットワークカードの設定変更」 (376 ページ) をお読みください。

- 5 デバイスの種類に **無線** を選択した場合は、次のダイアログで無線接続に関する設定を行ないます。
- 6 次へ を押します。
- 7 設定を有効にするには、さらに **OK** を押します。

ホスト名と DNS の設定

インストール時にネットワークの設定を変更せず、有線LANカードがその時点から利用可能な状態であった場合、お使いのコンピュータに対するホスト名は自動的に割り当てられ、**DHCP** が有効化されます。ネームサービスに関する情報についても同様です。お使いのネットワークカードに対する設定で **DHCP** を使用すると、ネームサーバの設定は **DHCP** サーバからの情報が適用されます。固定のネームサーバを設定したい場合は、そのように設定することもできます。

お使いのコンピュータの名前を変更したり、ネームサーバの設定と検索リストを変更したりしたい場合は、下記のようにして行ないます:

- 1 YaST の ネットワークデバイス モジュールを起動し、 ネットワーク設定 > ホスト名 / DNS タブ を選択します。
- 2 まずはホスト名とドメイン名を入力します。ドメイン名については指定しなくてもかまいません。またドメイン名は、このマシンがメールサーバである場合に、特に重要な設定です。なお、ホスト名はグローバル設定で、全てのネットワークインターフェイスに対して適用されることにご注意ください。

IP アドレスの取得に **DHCP** を利用している場合、お使いのコンピュータに対するホスト名を **DHCP** 経由で自動取得することができます。異なるネットワークに接続することで別のホスト名が設定されてしまい、グラフィカルなデスクトップ環境を利用する場合に混乱してしまうような場合には、このような動作を無効化することもできます。IP アドレスを **DHCP** 経由で取得している環境で、ホスト名を変更したくない場合は、**DHCP** でホスト名を変更 のチェックを外してください。

ホスト名をループバック IP に割り当てるを選択すると、`/etc/hosts` ファイル内で `127.0.0.2` (ループバック) のアドレスに設定したホスト名

が記入されるようになります。これは有効なネットワーク接続がないような状況でも ホスト名を解決したい場合に便利な機能です。

- 3** *DNS 設定の修正*では、DNS 設定(ネームサーバ、検索リストなどの `/etc/resolv.conf` ファイルの内容)の変更方法を選択することができます。

*既定のポリシーを使用する*を選択すると、設定は `netconfig` スクリプトで処理され、手動で設定したデータ (YaST や設定ファイルで設定したもの) と動的に取得したデータ (DHCP クライアントや NetworkManager などで設定したもの) を合成して使用します。既定のポリシーは、ほとんどの環境で適切な選択です。

*手動のみ*を選択すると、`netconfig` スクリプトは `/etc/resolv.conf` ファイルを修正できなくなります。手動での (エディタなどでの) 編集は通常どおり行なうことができます。

*カスタムポリシーを使用する*を選択すると、手動設定と自動設定の合成方法を指定する、カスタムポリシールール文字列を設定することができますようになります。文字列はインターフェイス名をカンマ区切りで指定するもので、正しい設定元であるものと判断するインターフェイスを指定します。インターフェイス名にはワイルドカードを指定することもできます。たとえば `eth* ppp?` を指定すると、最初に全てのイーサネット デバイス (eth) を対象にし、その後に `ppp0` から `ppp9` までのインターフェイスを対象とします。これ以外にも、`/etc/sysconfig/network/config` ファイルで指定する手動設定との合成方法について、2 種類の特別なポリシー設定が存在します:

STATIC

手動設定と動的な設定の両方を合成するルールです。

STATIC_FALLBACK

動的な設定が利用できない場合にのみ、手動設定を利用するルールです。

詳しい情報は `man 8 netconfig` で表示されるマニュアルページをお読みください。

- 4** それぞれ ネームサーバと ドメイン検索に入力を行ないます。ネームサーバは `192.168.1.116` などの形で、ホスト名ではなく IP アドレスで指定しなければなりません。ドメイン検索の枠内には、ドメインを指定せずにホス

ト名だけを指定した場合に自動的に補完するドメイン名を指定します。ドメイン検索に複数のドメインを指定する場合は、カンマか空白で区切ってください。

- 5 設定を有効にするには、**OK**を押します。

ルーティング (経路制御) の設定

お使いのマシンから他のネットワーク上にあるマシンと通信を行ないたい場合は、ネットワーク通信が正しい経路を通るようにするため、ルーティング (経路) 情報を設定しなければなりません。DHCP を利用している場合、ルーティング情報は自動的に設定されます。手動設定を使用している場合は、この設定も手動で行なわなければなりません。

- 1 YaST から **ネットワーク設定 > ルーティング** を選択します。
- 2 それぞれデフォルトゲートウェイに **IP アドレス**を設定します (**IPv4** および **IPv6** (必要である場合のみ) に設定します)。デフォルトゲートウェイはその他のルーティング情報に該当しなかった場合にのみ利用されるもので、該当するその他のルーティング情報が存在した場合には、デフォルトゲートウェイの代わりに、そちらの情報が使用されます。
- 3 ルーティングテーブルには、必要なだけルーティング情報を入力することができます。それぞれ宛先のネットワークの **IP アドレス**と **ゲートウェイの IP アドレス**、および **ネットマスク**を設定してください。また、指定した宛先にたどり着くための **デバイス**を指定することもできます (任意のデバイスを使用させたい場合は、マイナス記号を選択してください)。これらの値のうちのいずれかを省略したい場合は、いずれもマイナス記号 - を指定してください。一覧内にデフォルトゲートウェイを設定したい場合は、宛先に **default** と記入してください。

注記

複数のデフォルトゲートウェイを設定する場合は、それぞれの優先順位を設定するためにメトリックと呼ばれる値を設定することができます。メトリックのオプションを設定するには、**オプション**の項目内に **-metric 数値** と入力してください。数値が少ないほど優先順位が高いものとなり、最も高い優先順位を持つ経路が既定で使用されるようになります。ネットワークデバイスの接続が切れた場合は、そのルーティン

グ情報は削除され、次点の候補が使用されます。ただし、現在のカーネルでは静的なルーティング設定でメトリックを使用することはありません。`multipathd`などのルーティングデーモンを利用した場合にのみ有効です。

- 4 なお、お使いのシステムをルータとして機能させる場合は、 ネットワーク設定内の **IP 転送を有効にする** を選択してください。
- 5 設定を有効にするには、 **OK** を押します。

21.4.2 モデム

YaST コントロールセンターからは、 ネットワークデバイス > モデム を選択することで、モデム設定を行なうことができます。お使いのモデムが自動で検出されない場合は モデムデバイスのタブに移動し、 **追加** を押して手動設定を行なってください。手動設定では、 モデムデバイスの欄にモデムの接続されているインターフェイスを指定します。

ティップ: CDMA モデムと GPRS モデム

YaST から、対応している CDMA および GPRS モデムを設定する場合は、通常のモデムと同様に モデム モジュールから設定を行ないます。

図 21.4 モデムの設定

モデムのパラメータ

モデムデバイス (V)
/dev/modem

ダイヤルプレフィクス (必要時のみ) (X)

ダイヤルモード

☒ トーンダイヤル (T)
☐ パルスダイヤル (P)

特別の設定

☒ スピーカーを動作させる (S)
☒ ダイヤルトーンの検出 (E)

詳細 (D)

ヘルプ (H) 戻る (B) キャンセル (C) 次へ (N)

お使いの電話回線が構内交換機(PBX)内に設置されている回線である場合は、ダイヤルプレフィクスを入力する必要がある場合があります。通常は0を指定しますが、詳しくはPBXに添付されている手順書をお読みください。また、ダイヤルモードとしてトーンまたはパルスを選択することができるほか、ダイヤルトーンを検出するかどうかを設定することもできます。このほか、交換機に接続されている環境の場合は、ダイヤルトーン検出を無効に設定する必要があります。

また、詳細ボタンを利用すると、ボーレートやモデムの初期化文字列などを設定することができます。お使いのモデムが自動で検出されなかったり、データ転送を働かせるために特殊な設定が必要なモデムであったりする場合にのみ設定を変更してください。これは主にISDNのターミナルアダプタの場合に当てはまります。詳細ダイアログを終了するには、OKを押します。また、rootの権限を与えることなく一般ユーザにモデムの操作を許可するには、Kinternetを利用してroot以外のユーザにもデバイス操作を許すを選択してください。この場合、管理者権限を持たないユーザでもインターフェイスの有効化と無効化を行なうことができるようになります。さらにダイヤルプレフィクス正規表現では正規表現を指定します。一般ユーザでも修正することのできるKinternet内のダイヤルプレフィクスは、この正規表現にマッチしていなければならない、という制約が課せられます。この項目に何も記入し

ない場合は、管理者権限がないと異なるダイヤルプレフィックスを設定することができなくなります。

次のダイアログでは **ISP** を選択します。まずはご利用の **国** を選択すると、事前に設定されている一覧から **ISP** を選択することができます。また **新規** を押すと、ご利用になる **ISP** のデータを独自に入力することもできます。独自入力を行なう場合は、ご利用の **ISP** の名称と電話番号、ログイン名とパスワードをそれぞれ入力してください。また、**常にパスワードを尋ねる** を選択すると、接続するたびにパスワードを尋ねる動作になります。

最後のダイアログでは、追加の接続オプションを指定します：

ダイヤルオンデマンド

ダイヤルオンデマンドを選択すると、必要に応じてダイヤル アップ接続を行なうようになります。また、この機能を設定する場合は、1つ以上のネームサーバを設定してください。なお、インターネットへのデータ要求を定期的に行なう種類のソフトウェアが存在することから、インターネット接続の課金がそれほど高くない環境でのみご利用になることをお勧めします。

接続時に **DNS** を変更する

このオプションを選択すると、インターネットに接続するごとにネームサーバのアドレスを更新するようになります。この設定は既定で有効になっています。

自動的に **DNS** 情報を取得する

お使いのプロバイダが接続時にドメインネームサーバの設定を送信しない場合、このオプションの選択を外して **DNS** を手動で設定してください。

自動再接続

このオプションを選択すると、接続が切れると自動的に再接続を行なうようになります。

プロンプトの無視

このオプションを選択すると、ダイヤルアップサーバに接続した際に検出されるプロンプトを無視するようになります。接続が遅かったり全く動作しなかったりした場合は、このオプションを試してみてください。

外部ファイアウォールインターフェイス

このオプションを選択すると、ファイアウォールを有効化して外部インターフェイスとして登録するようになります。これを利用すると、イン

ターネット接続の利用中に 受けた攻撃から、身を守ることができるようになります。

無通信タイムアウト (秒)

このオプションは、何もネットワーク上の通信を行なわなかった場合に、モデムの 接続を自動で切断するための設定です。

IP の詳細

これを押すと、アドレス設定のダイアログが開きます。お使いのISPが接続したホストに対して動的な IP アドレスを割り当てない場合は、**可変 IP** アドレスのチェックを外して、独自のローカル IP アドレスとリモートの IP アドレスを入力します。詳しくはお使いのプロバイダにご確認ください。デフォルトルートについては有効に設定したまま、**OK**を押してダイアログを閉じてください。

次へを押すと元のダイアログに戻り、モデム設定の概要が表示されます。最後に **OK**を押すと、ダイアログを閉じる ことができます。

21.4.3 ISDN

このモジュールを利用することで、お使いのシステムにある 1 つまたは複数の ISDN カードを設定することができます。YaST で ISDN カードを認識しない場合は、**ISDN** デバイスタブ内にある **追加** ボタンを押して、手動でカードを追加してください。複数のインターフェイスを設定することができますが、複数の ISP を 1 つの インターフェイスに設定することはできません。続いて表示されるダイアログで、カードの動作に必要な ISDN オプションを設定します。

図 21.5 ISDN の設定

contrcontr0 に関する ISDN のローレベル設定

ISDN カードの情報

製造元 U.S.Robotics
ISDN カード ISDN PCI Card TA

ドライバ (V)
HiSax driver

ISDN プロトコル

☒ Euro-ISDN (EDSS1) (E)
☐ 1TR6 (G)
☐ 専用回線 (L)
☐ NI1 (I)

国 (C)
ドイツ

コード (D)
+49

市外局番 (A)
ダイヤルプレフィクス (D)

☐ ISDN 記録を開始する (J)

デバイスの有効化 (D)
起動時

ヘルプ (H) 戻る (B) キャンセル (C) OK (O)

図21.5「ISDN の設定」(390 ページ)に示しているとおり、次のダイアログでは使用するプロトコルを選択します。既定は *Euro-ISDN (EDSS1)* ですが、古い交換機や大きな交換機を使用するような場合は、*1TR6* を選択してください。また、アメリカ国内でお使いの場合は、*NI1* を選択してください。続いて国を選択すると、適切な国コードが項目に記入されます。最後に、必要に応じて市外局番とダイヤルプレフィクスを入力してください。ISDN の通信について全ての記録をとりたくない場合は、*ISDN 記録を開始する* のチェックを外してください。option.

また、デバイスの有効化では、ISDN インターフェイスの開始方法を指定することができます: *起動時* を選択すると、システムの起動時に ISDN デバイスを初期化するようになります。 *手動* を選択した場合は、ISDN ドライバを読み込むのに root から `rcisdn start` を実行しなければなりません。PCMCIA や USB のデバイスに対して設定する ホットプラグでは、デバイスが接続されたときにドライバを自動で読み込むようになります。これらの設定を行ない、*OK* を押します。

次のダイアログでは ISDN カードのインターフェイス種類を指定するほか、既存のインターフェイスに対して ISP を追加することもできます。インターフェイスは SyncPPP, RawIP のうちから選択しますが、多くの ISP は下記で説明するとおり SyncPPP モードで動作します。

図 21.6 ISDN インターフェイス設定

SyncPPP インターフェイス ipppnet0 の追加

接続設定

自分の電話番号(P)

0

デバイスの有効化 (D)

手動

☒ QInternet を利用して root 以外のユーザにもデバイス操作を許

☒ ChargeHUP (H)

☐ チャネルを束ねる (A)

☒ 外部ファイアウォールインターフェイス (W)

☒ ファイアウォールの再起動 (W)

詳細 (D)...

ヘルプ (H) 戻る (B) キャンセル (C) 次へ (N)

次に、自分の電話番号を入力します。これはお使いの環境によって入力する値が異なります:

回線に直接接続された ISDN カードの場合

標準の ISDN 回線では、3 種類の電話番号 (複数契約者番号、MSN とも呼びます) が用意されています (翻訳者注: 日本国内では、別途契約しない限り 1 種類の電話番号だけになります)。契約者が申し込むことで、最大 10 種類までの番号を用意することができます (翻訳者注: 日本国内では最大 3 種類までです)。この項目には、これらのうちのいずれかの番号を入力しますが、市外局番は含めずに入力してください。正しくない番号を入力した場合は、お使いの ISDN に割り当てられている最初の番号が自動的に使用されます。

構内交換機 (PBX) に接続された ISDN カードの場合

利用している機器によって、設定方法が異なります:

1. 家庭用などの小規模の交換機 (PBX) の場合、多くは内部の呼び出しに Euro-ISDN (EDSS1) プロトコルを使用しています。これらの交換機は内部に S0 バスが存在し、機器の呼び出しには内部番号を使用します。

この場合は、内部番号のうちのいずれかを電話番号として設定してください。直接の外部発信に対応するよう設定されている交換機であれば、交換機側の電話番号を使用することもできます。この設定でうまくいかない場合は、電話番号として1桁だけの"0"を設定してみてください。詳しい情報は、お使いの交換機に付属している文書をお読みください。

2. 企業用などの大規模な構内交換機の場合は、内部の呼び出しに1TR6プロトコルを使用しています。これらの電話番号はEAZと呼ばれ、通常は外線番号と対応した番号になっています。Linuxでは、EAZの最後の1桁を指定します。うまくいかない場合は、1から9までの各番号を試してみるのもよいでしょう。

次の課金タイミングになる前に回線を切断したい場合は、*ChargeHUP*を選択してください。ただし、この機能は全てのISPで動作するというわけではないことにご注意ください。また、これ以外にもチャンネルを束ねるを設定することで、マルチチャンネル設定(マルチリンクPPP)を設定することができます。さらに外部ファイアウォールインターフェイスとファイアウォールの再起動を選択することで、回線接続に対してファイアウォールを設定することもできますし、*KInternet*を利用してroot以外のユーザにもデバイス操作を許すを選択すると、管理者権限のない一般ユーザからでもインターフェイスの有効化/無効化を行なうことができますようになります。

詳細を押すと、一般家庭の環境ではあまり使用しない、より細かい設定画面を開くことができます。OKボタンを押すと詳細ダイアログを終了することができます。

次のダイアログでは、IPアドレス関連の設定を行ないます。お使いのプロバイダから固定のIPアドレスを割り当てられている場合を除き、可変IPアドレスを選択してください。割り当てがある場合は、ISPの提供情報に従ってローカルとリモートのIPアドレスをそれぞれ入力してください。また、このインターフェイスをインターネットに対する既定のルートに設定したい場合は、デフォルトルートを選択します。各ホストには既定のルートに設定するインターフェイスを1つだけ設定できます。設定が終わったら、次へを押します。

次のダイアログでは国とプロバイダを選択することができます。一覧には、迂回ダイヤルサービス(call-by-callやdial around serviceと呼ばれる場合もあります)に対応したプロバイダのみが掲載されています。お使いのISPが一覧に存在しない場合は、新規を押してください。するとプロバイダパラメータ

ダイアログが開き、ISP に関する 情報を入力できるようになります。なお電話番号を入力する際は、番号の途中に 空白やカンマを入れないでください。最後に ISP から提供されているユーザ名と パスワードを入力します。設定が終わったら 次へ を押します。

単独のワークステーションで **ダイヤルオンデマンド** (必要なときにダイヤルする) を行なうには、**ネームサーバ (DNS サーバ)** の設定も 必要です。多くの ISP では動的な DNS 設定に対応していて、ネームサーバの IP アドレスは ISP への接続時に取得することができるようになっています。単独のワークステーションの場合は、**ダミーデータ**として 192.168.22.99 のようなアドレスを設定しておく必要があります。お使いの ISP が動的な DNS に対応していない場合、ISP 側のネームサーバの IP アドレスを設定してください。また、必要であれば **無通信タイムアウト**を設定することもできます。これは、何もネットワーク上の 通信を行なわなかった場合に、接続を自動で切断するための設定 (秒単位) です。さらに 次へ を押してください。YaST はインターフェイスの設定概要を表示します。表示された設定を適用するため、最後に **OK** を押してください。

21.4.4 ケーブルモデム

国によっては、インターネットへの接続を行なう手段として **ケーブル TV** のネットワークを利用するのが一般的である場合があります。ケーブル TV の契約者は、一方を TV ケーブルに接続し、他方をコンピュータのネットワークカード (10Base-T のツイストペアケーブル) に接続するタイプのモデムを受け取ります。そのモデムは **ケーブルモデム**と呼び、固定または可変の IP アドレスが割り当てられる種類の、専用線によるインターネット接続が提供されます。

お使いのネットワークカードを設定する際には、お使いの ISP が提供する手順に従って設定し、**可変 IP アドレス** または **固定 IP アドレス** のいずれかを選択してください。現在、多くのプロバイダでは **DHCP** を利用していますが、特別なビジネスアカウントを取得することで固定の IP アドレスが割り当てられる場合もあります。

ケーブルモデムの設定について詳しい情報は、このトピックに関するサポートデータベースをお読みください。オンラインからは下記の URL でアクセスできます: http://ja.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher

21.4.5 DSL

DSL デバイスを設定するには、YaST の ネットワークデバイス 内にある *DSL* モジュールを選択してください。この YaST モジュールには、それぞれ下記のプロトコルに対応した、DSL リンクのパラメータ 設定ダイアログが含まれています:

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz カード)
- Point-to-Point Tunneling Protocol (PPTP) (オーストリア)

DSL の設定概要 ダイアログ内の *DSL* デバイス タブには、設定済みの DSL デバイスの一覧が表示されています。DSL デバイスの設定を変更するには、一覧からデバイスを選択して *編集* を押してください。新しい DSL デバイスを手動で 追加 する場合は、*追加* を押してください。

PPPoE や PPTP をベースにした DSL 接続の設定では、接続を行なうネットワーク カードの設定を正しく行なっておく必要があります。ネットワークカードの設定を行っていない場合は、まず ネットワークカードの設定 を押して設定を行なってください (詳しくは 21.4.1 項 「YaST を利用したネットワークカードの設定」 (373 ページ) をお読みください)。DSL 接続の場合、アドレスは DHCP ではない手段で自動配布 される場合がありますが、*可変 IP* アドレスを選択せず、その代わりに、192.168.22.1 のようなダミーの固定 IP アドレスを指定してください。また、サブネットマスク にもダミーの値 255.255.255.0 を 指定してください。単独のワークステーションで利用している場合は、デフォルトゲートウェイを設定する必要はありません。

ティップ

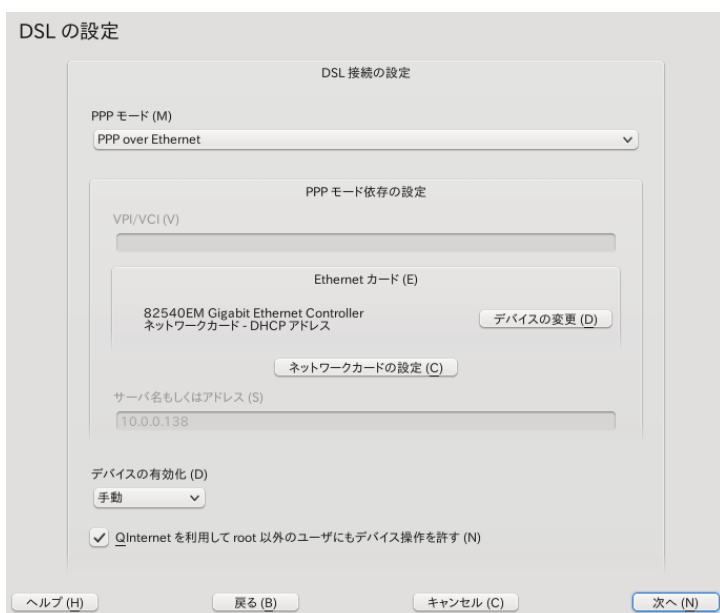
IP アドレス と サブネットマスク の 値は、いずれもダミーの値です。ネットワークカードの起動を行なうために必要な 設定であり、DSL の接続設定とは無関係です。

最初の DSL 設定ダイアログ (図21.7 「DSL の設定」 (395 ページ)) では、*PPP* モードと DSL モデムが接続されている *Ethernet* カードをそれぞれ選択します (多くの場合、eth0 を選択します)。その後、システムの起動時に DSL リ

リンクを確立させるかどうかを指定する、デバイスの有効化を選択します。また、**KInternet** を利用して **root** 以外のユーザにもデバイス操作を許すを選択すると、**KInternet** 経由で **root** 権限のない一般ユーザがインターフェイスの有効化／無効化を行なうことができるようになります。

次のダイアログでは国とプロバイダを選択することができます。この次の DSL 設定ダイアログは、これまでに選択したオプションによって異なります。利用可能なオプションについての詳細は、ダイアログ内から利用できるヘルプをお読みください。

図 21.7 DSL の設定



DSL 設定

DSL 接続の設定

PPP モード (M)
PPP over Ethernet

PPP モード依存の設定

VPI/VCI (V)

Ethernet カード (E)
82540EM Gigabit Ethernet Controller
ネットワークカード - DHCP アドレス
デバイスの変更 (D)

ネットワークカードの設定 (C)

サーバ名もしくはアドレス (S)
10.0.0.138

デバイスの有効化 (D)
手動

☒ KInternet を利用して root 以外のユーザにもデバイス操作を許す (N)

ヘルプ (H) 戻る (B) キャンセル (C) 次へ (N)

単独のワークステーションでダイヤルオンデマンド(必要なときにダイヤルする)を行なうには、ネームサーバ(DNSサーバ)の設定も必要です。多くのISPでは動的なDNS設定に対応していて、ネームサーバのIPアドレスはISPへの接続時に取得できるようになっています。単独のワークステーションの場合は、ダミーデータとして192.168.22.99のようなアドレスを設定しておく必要があります。お使いのISPが動的なDNSに対応していない場合、ISP側のネームサーバのIPアドレスを設定してください。

また、無通信タイムアウト (秒) を設定することもできます。これは、何もネットワーク上の通信を行なわなかった場合に、接続を自動で切断するための設定 (秒単位) です。通常のタイムアウトは 60 から 300 秒程度を指定します。ダイヤルオンデマンドを有効にしていない場合は、このタイムアウト値をゼロに設定することで、自動切断を無効化することができます。

T-DSL の設定は、DSL の設定によく似ています。プロバイダとして *T-Online* を選択すると、YaST は T-DSL の設定ダイアログを表示します。このダイアログから、T-DSL に必要ないくつかの追加情報 (ライン ID, T-Online 番号, ユーザコード, パスワード) を入力してください。これらの情報は、T-DSL の契約時に受け取った中に含まれているものを入力します。

21.5 NetworkManager

NetworkManager はラップトップ機やその他の可搬型コンピュータをご利用の場合は便利なツールです。NetworkManager を利用することで、移動中にネットワークが切り替わってもネットワークインターフェイスの設定やその切り替えについて、心配する必要がなくなります。

21.5.1 NetworkManager と ifup

NetworkManager は全ての環境において便利なものとは言えませんが、ネットワーク接続管理を行なうにあたって、従来の方法 (ifup) と NetworkManager のどちらを利用するのか、選択するだけの価値はあります。NetworkManager でお使いのネットワーク接続を管理したい場合は、項「NetworkManager の有効化」 (第5章 *NetworkManager* の使用, ↑スタートアップ) に書かれている手順で YaST のネットワーク設定モジュールを起動し、NetworkManager を有効に設定してください。あとは NetworkManager でネットワークの設定を行なうだけです。NetworkManager の用途と設定手順の説明について、詳しくは第5章 *NetworkManager* の使用 (↑スタートアップ) をお読みください。

ifup と NetworkManager には下記のような違いがあります:

root 権限

ネットワークの設定に NetworkManager を使用する場合、アプレットを利用することでデスクトップ環境から任意のタイミングで、ネットワークを切り替えたり停止したり、起動したりすることができます。また

NetworkManager では、root 権限を必要とすることなく無線ネットワークカードの接続設定を行なうことができます。このため、モバイルワークステーション環境の場合 NetworkManager は申し分のない解法となるでしょう。

ifup による従来の設定方法でも、ユーザ管理デバイスのように介入無しに接続を切り替えたり 停止したり、開始したりすることができます。しかしながら、この方法ではネットワーク デバイスの変更や設定には root の権限が必要となります。これは特に、 事前に接続する可能性のある全てのネットワークを設定しなければならないという 点で、モバイル環境の場合に問題となります。

ネットワーク接続の種類

従来の方法でも NetworkManager でも、無線ネットワーク (WEP, WPA-PSK, WPA-Enterprise でのアクセス) やダイアルアップ、有線の設定を取り扱うことができます。DHCP や固定のアドレス設定、VPN 経由の接続にも対応しています。

NetworkManager は利用可能なネットワーク接続から、最も良いものを常に選択してつなぎ続けるように動作します。ネットワークケーブルが誤って外されてしまった場合でも、再接続を試みるようになっています。また無線接続の場合は、設定された 無線接続の中から最も信号品質の良いネットワークを判断して自動接続する仕組みを備えています。ifup で同じ機能を利用するには、様々な設定を施さなければ実現できません。

21.5.2 NetworkManager の機能と設定ファイル

NetworkManager での個別のネットワーク設定は設定プロファイルに保存されます。NetworkManager または YaST で設定した システム 接続は、/etc/sysconfig/network/ifcfg-* ファイル内に保存されます。ユーザ側で設定した接続は、GNOME の場合は GConf 内に、KDE の場合は \$HOME/.kde4/share/apps/networkmanagement/* に保存 されます。

何もプロファイルを設定していない場合、NetworkManager は自動的にプロファイルを作成し、Auto \$INTERFACE-NAME の名前を設定します。これにより、できる限り多くの場合において設定作業を行なうことなく、安全な設定を使用することができるようになっています。自動で作成された設定が要

件にあわない場合は、KDEやGNOMEが提供するネットワーク接続設定ダイアログを利用して、必要な設定作業を行なってください。詳しくは項「ネットワーク接続の設定」(第5章 *NetworkManager* の使用, ↑スタートアップ)をお読みください。

21.5.3 NetworkManager の機能のコントロールと無効化

集中管理されているマシンでは、一部のNetworkManager機能が操作されたり、無効化されていたりする場合があります。たとえば管理者が設定した接続をユーザ側で修正することを許すかどうかや、ユーザ側で独自のネットワーク設定を作成することができるかどうかなどがあります。NetworkManagerでのポリシー設定を閲覧したり変更したりするには、PolicyKit 向けのグラフィカルな認可ツールを起動してください。左側のツリーで *network-manager-settings* の項目を選択して設定を行ないます。詳しい手順については第9章 *PolicyKit* (↑セキュリティガイド)をお読みください。

21.6 手動でのネットワーク設定方法

手動でのネットワーク設定は最後の手段として用意されているものです。特に問題がない限り、YaSTでの設定をお勧めします。しかしながら、YaSTでのネットワーク設定を行なうにあたり、基礎になっている情報を知っておくと、より便利に利用できると考えています。

カーネルがネットワークカードを検出して関連するネットワークインターフェイスを作成すると、デバイスの認識順序とカーネルモジュールの読み込み順序に従ってデバイス名が割り当てられます。規定でカーネルが割り当てる名前前は非常に単純に推測できるもので、ハードウェア環境の変更を難しくしてしまっています。稼働中にハードウェアを追加したり削除したり、もしくはデバイスの設定を自動化しているようなシステムでは、再起動を行なうたびにデバイス名が異なることになってしまい、安定したデバイス名の割り当てを期待することができません。

しかしながら、全てのシステム設定ツールはインターフェイス名を基準にして設定を管理しています。このような問題を解決するのが *udev* です。 *udev* の固定ネットワーク デバイス生成の機能 (`/lib/udev/rules.d/75`

-persistent-net-generator.rules)を利用すると、条件に該当するハードウェア(規定ではハードウェアアドレスを使用します)に対して、固定のインターフェイス名を割り当てることができます。udevのネットワークインターフェイスに対するデータベースは、/etc/udev/rules.d/70-persistent-net.rules ファイル内に保存されます。上記ファイルの各行には単一のネットワークインターフェイス名と、そのインターフェイスに割り当てる固定の名前が記述されています。システム管理者はNAME=""の項目を編集することで、割り当てられた名前を変更することができます。それぞれの設定はYaSTから変更を行なうこともできます。

表21.5「手動ネットワーク設定を行なう場合の関連スクリプト」(399 ページ)には、ネットワーク設定に関する最も重要なスクリプトファイルを列挙しています。

表 21.5 手動ネットワーク設定を行なう場合の関連スクリプト

コマンド	機能
ifup, ifdown, ifstatus	if スクリプトはネットワークインターフェイスの起動や停止、および指定したインターフェイスの状態確認を行なうためのスクリプトです。詳しくはifupのマニュアルページをお読みください。
rcnetwork	rcnetwork スクリプトは全てのネットワークインターフェイス(または指定した1つのインターフェイス)を起動したり停止したりするために、使用するスクリプトです。 rcnetwork stop でネットワーク インターフェイスの停止を、rcnetwork start で起動を、rcnetwork restart で再起動をそれぞれ行ないます。なお、ある1つのネットワークインターフェイスだけ停止や起動、再起動を行ないたい場合は、コマンドの後ろにインターフェイス名を指定してください。たとえばrcnetwork restart eth0 のようになります。また、rcnetwork status コマンドでは、インターフェイスの状態とIPアドレス、およびDHCPクライアントの稼働状態がそれぞれ表示されます。さらにrcnetwork stop-all-dhcp-clients や rcnetwork restart-all-dhcp-clients では、それぞれ 全てのネッ

トワークインターフェイスに対して動作している DHCP クライアントを停止したり、再起動したりすることができます。

udev と固定のデバイス名について、詳しくは 第19章 *udev* による動的なカーネルデバイス管理 (317 ページ) をお読みください。

21.6.1 設定ファイル

本章には、ネットワーク関連の設定ファイルとそれらの目的説明、および使用する書式について説明を行なっています。

/etc/sysconfig/network/ifcfg-*

これらのファイルはネットワークインターフェイスの設定ファイルです。開始モードや IP アドレスなどの設定が書かれています。利用可能なパラメータについては、*ifup* のマニュアルページをお読みください。また、特定のインターフェイスに対してだけ一般的な設定を行ないたい場合は、*dhcp*、*wireless* にあるほとんどの値を *ifcfg-** ファイルで設定することができます。なお、*/etc/sysconfig/network/config* 内にある多くの値はグローバル設定であるため、*ifcfg* ファイルでは上書きすることができません。たとえば *NETWORKMANAGER* や *NETCONFIG_** の値などがグローバル設定です。

また、*ifcfg.template* ファイルについては、「*/etc/sysconfig/network/config*、*/etc/sysconfig/network/dhcp*、*/etc/sysconfig/network/wireless*」 (400 ページ) をお読みください。

/etc/sysconfig/network/config, /etc/sysconfig/network/dhcp, /etc/sysconfig/network/wireless

config ファイルは、*ifup*、*ifdown*、*ifstatus* の各ファイルの動作に関して、一般的な設定を行なうためのファイルです。また、*dhcp* ファイルには

DHCP の設定が、wireless ファイルには無線 LAN カードの設定がそれぞれ書かれています。これら 3 種類のファイルで利用する設定は、それぞれコメント内に書かれています。また、/etc/sysconfig/network/config ファイル内の設定値のうちのいくつかを ifcfg-* で利用することができます。ifcfg-* 側に設定値を入力した場合は、config よりも優先されます。なお、/etc/sysconfig/network/ifcfg.template ファイルでは、インターフェイス単位で利用可能な設定値の一覧を示しています。しかしながら、多くの /etc/sysconfig/network/config にある多くの変数はグローバル設定で、ifcfg-* ファイルでは上書きできないことにご注意ください。たとえば NETWORKMANAGER や NETCONFIG_* がグローバル設定です。

/etc/sysconfig/network/routes と /etc/sysconfig/network/ifroute-*

このファイルには、スタティックルート (静的な経路情報) を設定します。様々なシステム処理に必要な全てのスタティックルートを、/etc/sysconfig/network/routes ファイルに設定することができます: ホストへの経路のほか、ゲートウェイ経由のホスト経路、ネットワークへの経路などがあります。各インターフェイスに対して個別の経路を設定する必要がある場合は、追加の設定ファイル /etc/sysconfig/network/ifroute-* を作成してください。ここで、* にはインターフェイスの名前を入力します。経路設定ファイルの設定は、たとえば下記のようになります:

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0
207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

最初の列には経路の宛先が示されています。ネットワークやホストの IP アドレスのほか、ネームサーバに接続できる環境であれば、完全修飾ドメイン名 (FQDN) やホスト名を利用することもできます。

2列目には、既定のゲートウェイまたは特定のホストやネットワークに到達するためのゲートウェイが書かれています。3列目にはネットワークのネットマスクか、もしくはゲートウェイの向こう側にあるホストが書かれています。たとえばゲートウェイの向こう側にあるホストを設定する場合、マスクは 255.255.255.255 になります。

4 列目はループバックやイーサネット, PPP やダミーデバイスなどに接続されているネットワークでのみ意味を持つものです。ここにはデバイス名を入力します。

(オプションで)5つめの列に入力することもできます。ここでは経路の種類を設定します。この列にはパーサーが正しくコマンドを解釈するために利用する、- 記号を設定する必要はありません。詳しくは routes (5) のマニュアルページをお読みください。

IPv4 および IPv6 向けには、下記の統合フォーマットを利用することができます:

プレフィクス/長さ ゲートウェイ - [インターフェイス]

また、下記のような互換フォーマットを利用することもできます:

プレフィクス ゲートウェイ 長さ [インターフェイス]

IPv4 に対しては、ネットマスクを利用した古い形式を利用することもできます:

IPv4 ネットマスク ゲートウェイ IPv4 ネットマスク [インターフェイス]

それぞれ下記に示す例が等価な表現になります:

2001:db8:abba:cafe::/64	2001:db8:abba:cafe::dead	-	eth0
208.77.188.0/24	208.77.188.166	-	eth0
2001:db8:abba:cafe::	2001:db8:abba:cafe::dead	64	eth0
208.77.188.0	208.77.188.166	24	eth0
208.77.188.0	208.77.188.166	255.255.255.0	eth0

/etc/resolv.conf

このファイルには、ホストが属するドメイン名を入力する (search キーワードを使用します) ほか、アクセスするネームサーバのアドレス (nameserver キーワード) も入力します。このファイルには複数のドメイン名を入力することができます。完全修飾型ではない名前を解決する場合、search に書かれたそれぞれのドメイン名を付加して 1 回ずつ解決を試みます。複数のドメイン名は複数行に分けて設定するものとし、それぞれ nameserver から書き始めてください。また、コメントは # 文字を 行頭に入れてください。例21.5

「/etc/resolv.conf」(403 ページ)には /etc/resolv.conf の記述例を示しています。

ただし、/etc/resolv.conf は手作業による編集は行なうべきではありません。このファイルは netconfig スクリプトが生成するものであるためです。YaST を利用せずに固定の DNS 設定を行ないたい場合は、/etc/sysconfig/network/config ファイル内の該当項目を修正してください:

NETCONFIG_DNS_STATIC_SEARCHLIST

ホスト名の参照に際して、利用する DNS ドメイン名。

NETCONFIG_DNS_STATIC_SERVERS

ホスト名の参照時に利用する、ネームサーバの IP アドレス。

NETCONFIG_DNS_FORWARDER

設定すべき DNS フォワーダの名前。

netconfig を利用した DNS 設定を無効化するには、NETCONFIG_DNS_POLICY='' を設定してください。また、netconfig について詳しくは man 8 netconfig をお読みください。

例 21.5 /etc/resolv.conf

```
# ドメイン名の指定
search example.com
#
# ネームサーバとして dns.example.com (192.168.1.116) を使用する設定
nameserver 192.168.1.116
```

/sbin/netconfig

netconfig は追加のネットワーク設定を管理するための モジュール型ツールです。本ソフトウェアは、事前に定義しておいた設定と DHCP や PPP などで取得した動的な情報を、事前に定義したポリシーに従って 合成することができます。必要な変更は、各設定ファイルの修正を行なって サービスの再起動などを行なう netconfig モジュールを呼び出すことで、システムに反映される形になります。

netconfig は 3 つの主なアクションから構成されています。netconfig modify と netconfig remove は DHCP や PPP デーモンから呼び出される

もので、**netconfig** に対して設定を変更したり削除したりする場合に利用します。ユーザからは **netconfig update** コマンドだけが利用できます:

modify

netconfig modify コマンドは、現存するインターフェイス やサービス固有の動的な設定を変更し、ネットワークの設定を更新するコマンドです。**netconfig** は標準入力またはファイル (`--lease-file` ファイル名を指定します) から設定を読み込み、システムの再起動 (または次の修正や削除動作) までの間、内部でそれらを保持します。あるインターフェイスやサービスに対する設定がすでに存在していた場合は、それらは上書きされます。インターフェイスは `-i` インターフェイス名で指定するほか、サービスの場合は `-s` サービス名で指定します。

remove

netconfig remove コマンドは、指定したインターフェイスやサービスに対して以前に設定した設定を削除し、ネットワークの設定を更新するコマンドです。インターフェイスは `-i` インターフェイス名で指定するほか、サービスの場合は `-s` サービス名で指定します。

update

netconfig update コマンドは、現在の内容でネットワーク設定を更新するコマンドです。これはポリシーや静的な設定を更新した場合に便利なコマンドです。また、指定したサービス (dns, nis, ntp) に対してのみ更新を行ないたい場合は、`-m` モジュールタイプパラメータをご利用ください。

netconfig のポリシーや静的な設定は `/etc/sysconfig/network/config` ファイルに書かれているもので、手動で行なうことができるほか、**YaST** を利用して設定することもできます。**DHCP** や **PPP** のような自動設定ツールが提供する動的な設定は、これらのツールから直接通知されるか、**netconfig modify** や **netconfig remove** コマンドで通知されます。また、**NetworkManager** でも **netconfig modify** や **netconfig remove** コマンドを使用する場合があります。**NetworkManager** が有効になっている場合 (auto ポリシーモードの場合) は、従来の **ifup** 方式での全てのインターフェイス設定が無視されます。**NetworkManager** がいかなる設定も提供しない場合に限って、静的な設定が次点候補として使用されます。**NetworkManager** と従来の **ifup** 方式の混在には対応していません。

netconfig について詳しくは、`man 8 netconfig` で表示されるマニュアルページをお読みください。

/etc/hosts

例21.6「/etc/hosts」(405ページ)に示してあるとおり、このファイルには IP アドレスとホスト名の対が書かれています。ネームサーバが設定されていない場合は、IP 接続を行なう可能性のある全てのホストについて、本ファイル内に設定しておく必要があります。各ホスト 1 行とし、IP アドレスと完全修飾ドメイン名、ホスト名をそれぞれ記入します。IP アドレスは行頭に記述し、タブかスペースで区切って残りの項目を記入します。コメントを記入する場合は、コメントの前に # を記入してください。

例 21.6 /etc/hosts

```
127.0.0.1localhost
192.168.2.100jupiter.example.comjupiter
192.168.2.101venus.example.comvenus
```

/etc/networks

このファイルでは、ネットワーク名とネットワークアドレスの変換を行いません。書式は hosts ファイルに似ていますが、ネットワーク名の後にアドレスが続く点が異なります。詳しくは 例21.7「/etc/networks」(405ページ)をご覧ください。

例 21.7 /etc/networks

```
loopback127.0.0.0
localnet192.168.0.0
```

/etc/host.conf

このファイルでは、リゾルバと呼ばれるライブラリを利用してホスト名やネットワーク名を名前解決する際、その解決処理を設定します。このファイルは libc4, libc5 にリンクされているライブラリでのみ使用します。現行の glibc プログラムの場合は、/etc/nsswitch.conf ファイルの設定を参照してください。パラメータはそれぞれ 1 行に 1 つずつ入力します。コメントを記入する場合は、コメントの前に # を入力してください。利用可能なパラメータについては表21.6「/etc/host.conf のパラメータ」(406ページ)を、/etc/

host.conf の設定例については例21.8「/etc/host.conf」(406 ページ)をそれぞれお読みください。

表 21.6 /etc/host.conf のパラメータ

<code>order hosts, bind</code>	名前解決を行なう場合アクセス順序を指定します。利用可能なパラメータは下記の通りです(スペースまたはカンマで区切ります): <i>hosts</i> : /etc/hosts ファイルを検索して名前解決を行ないます。 <i>bind</i> : ネームサーバにアクセスして名前解決を試みます。 <i>nis</i> : NIS を使用します。
<code>multi on/off</code>	/etc/hosts ファイル内に入力したホストについて、複数の IP アドレスを許すかどうかを設定します。
<code>nospoof on</code> <code>spoofalert on/off</code>	これらのパラメータはネームサーバの偽装検知に影響するパラメータです。ネットワーク設定には影響しません。
<code>trim</code> ドメイン名	ホスト名の解決後、指定したドメイン名をホスト名から取り除くことを指定します(ホスト名にドメイン名が含まれていた場合のみ)。このオプションは/etc/hosts ファイル内にローカルドメインのホスト名だけが含まれていて、それらを指定したドメイン名として認識したい場合に指定します。

例 21.8 /etc/host.conf

```
# ネームサーバが起動しています
order hosts bind
# 複数アドレスを有効にする
multi on
```

/etc/nsswitch.conf

GNU C ライブラリ 2.0 以降では、*Name Service Switch* (NSS) を利用するようになりました。詳しくは `nsswitch.conf` (5) のマニュアルページや、*GNU C ライブラリ レファレンスマニュアル* をお読みください。

問い合わせの順序は `/etc/nsswitch.conf` ファイルで指定します。`nsswitch.conf` の設定例は 例21.9 「`/etc/nsswitch.conf`」 (407 ページ) にあります。コメントを記入する場合は、コメントの前に `#` を記入してください。この例で `hosts` データベース内の項目は、名前解決の要求が `DNS` を介して `/etc/hosts (files)` に送信される設定になっています。

例 21.9 `/etc/nsswitch.conf`

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

NSS 上で利用可能なデータベースは、表21.7 「`/etc/nsswitch.conf`から利用できるデータベース」 (407 ページ) に一覧で示されています。これに加え、`automount`, `bootparams`, `netmasks`, `publickey` の各データベースが将来使用できるようになる予定です。NSS データベースとして利用可能なオプションは、表21.8 「NSS 「データベース」 の設定オプション」 (408 ページ) 内に書かれています。

表 21.7 `/etc/nsswitch.conf` から利用できるデータベース

aliases	sendmail で実装されているメール別名; 詳しくは <code>man 5 aliases</code> をお読みください。
ethers	イーサネットアドレス。
group	<code>getgrent</code> で使用するユーザグループ。 <code>group</code> のマニュアルページも併せてお読みください。

hosts	gethostbyname などの関数で取得できる、ホスト名や IP アドレス。
netgroup	アクセス許可を設定する目的で使用する有効なホストとユーザの一覧; 詳しくは netgroup(5) のマニュアルページをお読みください。
networks	getnetent で使用するネットワーク名と アドレス。
passwd	getpwent で使用されるユーザパスワード; 詳しくは passwd(5) のマニュアルページをお読みください。
protocols	getprotoent で使用するネットワークプロトコル; 詳しくは protocols(5) のマニュアルページをお読みください。
rpc	getrpcbyname などの関数で取得できる、リモートプロシージャコール (RPC) の名前とアドレス。
services	getservent で使用するネットワークサービス。
shadow	getspnam で使用するユーザのシャドウパスワード; 詳しくは shadow(5) のマニュアルページをお読みください。

表 21.8 NSS 「データベース」 の設定オプション

files	ファイルへの直接アクセスを指定します。たとえば /etc/aliases などのファイルにアクセスします。
db	データベース経由でのアクセス。
nis,nisplus	NIS を使用します。詳しくは 第3章 <i>Using NIS</i> (↑セキュリティガイド)をお読みください。
dns	hosts と networks に対してのみ 設定可能な拡張です。

compat

passwd, shadow, group に対してのみ設定可能な拡張です。

/etc/nscd.conf

このファイルは、**nscd** (name service cache daemon; ネームサービスのキャッシュデーモン) を設定するために使用します。詳しくは **nscd**(8) および **nscd.conf**(5) のマニュアルページをお読みください。既定では、システムの **passwd** と **groups** の項目について、**nscd** でキャッシュを行なう(メモリ上に一時記憶する)よう設定されています。このサービスを利用しないと、**NIS** や **LDAP** などのディレクトリサービスを提供している場合、接続が発生するたびに名前やグループへのアクセスを行なってしまうため、性能面で重要な仕組みになっています。既定では **hosts** をキャッシュしないように設定されていますが、これは **nscd** のキャッシュ機能を利用してしまうと、ローカルシステム側から正引きや逆引きの確認が行なえなくなってしまうためです。**nscd** で名前のキャッシュを行なう代わりに、**DNS** サーバ側でキャッシュを設定してください。

なお、**passwd** ファイルをキャッシュするように設定している場合は、新しいユーザを追加してから認識されるまでに15秒ほどの時間がかかります。このような待機時間を減らすには、**rcnscd restart** コマンドで **nscd** を再起動してください。

/etc/HOSTNAME

このファイルには、ドメイン名を含む完全修飾ホスト名が書かれています。このファイルは、マシンの稼働中に複数のスクリプトから読み取られます。ファイルはホスト名を含む1行でなければなりません。

21.6.2 設定のテスト

変更した設定を設定ファイルに書き込む前に、設定をテストすることができます。設定のテストを行なうには、**ip** コマンドをご利用ください。また、接続のテストを行なうには、**ping** コマンドをご利用ください。もちろん従来が存在する **ifconfig** や **route** コマンドもご利用いただけます。

ip, ifconfig, route の各コマンドは、設定ファイルに保存を行なうことなく、設定そのものを直接 変更するためのコマンドです。適切な設定ファイル内に設定を書き込まない限り、変更した設定はシステムの再起動を行なうことでリセットされてしまいます。

ip を利用したネットワークインターフェイスの設定

ip コマンドはネットワークデバイスやルーティングの設定のほか、ポリシールーティングやトンネルを設定したり、設定を閲覧したりするためのツールです。

ip コマンドはとても複雑なツールです。一般的には ip オプション オブジェクト コマンド のような書式で記述します。オブジェクトには下記のものを設定することができます:

link

ネットワークデバイスを表わすオブジェクトです。

address

デバイスの IP アドレスを表わすオブジェクトです。

neighbour

ARP や NDISC キャッシュ項目を表わすオブジェクトです。

route

ルーティングテーブルの項目を表わすオブジェクトです。

rule

ルーティングポリシーデータベース内のルールを表わすオブジェクトです。

maddress

マルチキャストのアドレスを表わすオブジェクトです。

mroute

マルチキャストのルーティングキャッシュの項目を表わすオブジェクトです。

tunnel

IP 経由でのトンネルを表わすオブジェクトです。

コマンドを指定しない場合は、既定のコマンドが指定されたものとして扱われます (通常は `list` です)。

デバイスの状態を変更するには、`ip link set デバイス名 コマンド` のように入力します。たとえば `eth0` デバイスを無効化したい場合は、`ip link set eth0 down` と入力します。再度有効化する場合は、`ip link set eth0 up` と入力してください。

デバイスを有効化したら、次に設定を行ないます。IP アドレスを設定するには、`ip addr add IP アドレス + dev デバイス名` のように入力します。たとえば、`eth0` インターフェイスに対して `192.168.12.154/30` のアドレスを設定し、標準のブロードキャストアドレスを設定 (`brd` オプション) するには、`ip addr add 192.168.12.154/30 brd + dev eth0` と入力します。

接続を行なうことができるようにするためには、さらにデフォルトゲートウェイを設定しなければなりません。お使いのシステムにデフォルトゲートウェイを設定するには、`ip route add ゲートウェイの IP アドレス` のように入力します。また、アドレス変換を行ないたい場合は、`nat` キーワードを指定し、下記のように入力します:`ip route add nat IP アドレス via 他の IP アドレス`

全てのデバイスを表示するには、`ip link ls` と入力してください。稼働中のインターフェイスのみを表示する場合は `ip link ls up` を、デバイスごとのインターフェイス統計情報を表示するには、`ip -s link ls デバイス名` のように入力してください。また、お使いのデバイスに設定したアドレスを確認するには、`ip addr` コマンドを使用します。`ip addr` コマンドでは、MAC アドレスの情報も表示されます。さらに、全ての経路情報を表示するには、`ip route show` コマンドをご利用ください。

上記以外にも `ip` コマンドには様々な機能が備わっています。`ip` コマンドについてさらに詳しい情報は、`ip help` と入力するか、もしくは `ip(8)` のマニュアルページをお読みください。なお、`help` オプションは `ip` オブジェクトに対しても動作します。たとえば `ip addr` コマンドの使い方について知りたい場合は、`ip addr help` と入力してください。`ip` コマンドのマニュアルは、`/usr/share/doc/packages/iproute2/ip-cref.pdf` にもあります。

ping を利用した接続テスト

ping コマンドは TCP/IP 接続が正しく動作するかどうかをテストするための標準ツールです。本コマンドは ICMP と呼ばれるプロトコルを利用し、ECHO_REQUEST と呼ばれる小さなデータを宛先のホストに送信し、即時の応答を求めます。うまく動作すれば ping コマンドはその結果を表示し、ネットワークの接続の基本部分が正しく動作していることを表わします。

また、ping コマンドは 2 台のコンピュータ間での接続テストを行なうだけではありません。本コマンドは接続品質に関する基本情報も提供します。例 21.10 「ping コマンドの出力」 (412 ページ) に示している例は、ping コマンドの出力例です。最後から 2 行目の部分には、送信されたパケット数と損失数、および ping コマンドの実行時間が表示されています。

また、本コマンドで指定する宛先には IP アドレスだけでなくホスト名を指定することもできます。たとえば ping example.com や ping 192.168.3.100 のように入力することができます。なお、本コマンドは Ctrl + C を押すまでパケットを送信し続けます。

基本的な接続確認だけを行ないたい場合は、-c オプションを設定して送信数を制限することもできます。たとえば 3 パケットだけ送信したい場合は、ping -c 3 example.com のように入力してください。

例 21.10 ping コマンドの出力

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

既定のパケット送信間隔は 1 秒に設定されています。この送信間隔を変更するには、-i オプションを指定してください。たとえば 10 秒間隔で送信する場合は、ping -i 10 example.com のように入力します。

また、複数のネットワークデバイスを利用している場合は、ping コマンドに対して特定のインターフェイスアドレスを利用するように指定したほうが便利である場合があります。このような場合は、-I オプションでインターフェ

イス名を指定してください。たとえば `ping -I wlan1 example.com` のように入力します。

`ping` コマンドについてさらに詳しい情報については、`ping -h` で出力されるヘルプか、もしくは `ping (8)` のマニュアルページをお読みください。

ティップ: IPv6 アドレスに対する `ping`

IPv6 アドレスの場合は `ping6` コマンドを使用します。なお、リンクローカルアドレスに対して `ping` を送信する場合は、`-I` オプションを利用してインターフェイスを指定しなければなりません。たとえば 下記のコマンドでは、`eth1` を介して指定したアドレスに 通信ができるかどうかを確認します:

```
ping6 -I eth1 fe80::117:21ff:fed:a425
```

`ifconfig` を利用したネットワーク設定

`ifconfig` コマンドは、ネットワークを設定するための ツールです。

注記: `ifconfig` コマンドと `ip` コマンド

`ifconfig` プログラムは古いソフトウェアです。できる限り `ip` コマンドをご利用ください。 `ip` コマンドと比べると `ifconfig` はインターフェイスの 設定しか行なうことができません。また、インターフェイス名は 9 文字までに限定されます。

パラメータを何も指定しない場合、`ifconfig` は現在有効になっているインターフェイスの状態を表示します。出力例は 例21.11 「`ifconfig` コマンドの出力例」(414 ページ) の通りです。 `ifconfig` は詳細な情報をうまく整理して出力します。また出力の最初の行には、お使いのデバイスに対する MAC アドレス(HWaddr)が含まれます。

例 21.11 `ifconfig` コマンドの出力例

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1     Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 MB)
```

`ifconfig` を利用するにあたっての詳しいオプション設定などの情報は、`ifconfig-h` と入力して表示することのできるヘルプか、もしくは `ifconfig` (8) のマニュアルページをお読みください。

route を利用したルーティング設定

`route` コマンドは、IP ルーティング (経路制御) テーブルを操作するためのプログラムです。ルーティング情報を閲覧することができるほか、追加や削除を行なうこともできます。

注記: `route` コマンドと `ip` コマンド

`route` プログラムは古いソフトウェアです。できる限り `ip` コマンドをご利用ください。

routeは、ルーティング関連の問題が発生した場合に、ルーティング設定に関する情報を素早く確実に得るには便利なツールです。現在のルーティング情報を閲覧するには、root から route -n と入力します。

例 21.12 route -n コマンドの出力例

```
route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags      MSS Window  irtt Iface
10.20.0.0        *                255.255.248.0    U           0  0        0 eth0
link-local        *                255.255.0.0      U           0  0        0 eth0
loopback          *                255.0.0.0        U           0  0        0 lo
default          styx.exam.com    0.0.0.0          UG          0  0        0 eth0
```

routeを利用するにあたっての詳しいオプション設定などの情報は、route -h と入力して表示 することのできるヘルプか、もしくはroute (8) のマニュアルページをお読みください。

21.6.3 起動時のスクリプト

上記までの章で説明した設定ファイルのほかに、マシンを起動する際に ネットワークプログラムを読み込むための各種スクリプトが存在します。これらのスクリプトは、システムが マルチユーザのランレベルのいずれかに移行すると、すぐに実行されるものです。これらのスクリプトのうち、いくつかを表21.9「ネットワークプログラムのスクリプト」(415ページ)で説明します。

表 21.9 ネットワークプログラムのスクリプト

/etc/init.d/ network	このスクリプトは、ネットワークインターフェイスの設定を処理するための ものです。network サービスが開始されなかった場合、ネットワーク インターフェイスは利用できなくなります。
/etc/init.d/ xinetd	このスクリプトは、 xinetd を起動するためのものです。 xinetd はシステム上でサービスを提供するための使用されるプログラムで、たとえばFTP の接続があった場合に vsftpd を起動したりすることができます。

<code>/etc/init.d/ rpcbind</code>	このスクリプトは、RPCプログラム番号をユニバーサルアドレスに変換する rpcbind ユーティリティを起動するためのものです。NFS サーバなどの RPC サービスで必要になります。
<code>/etc/init.d/ nfsserver</code>	このスクリプトは、NFS サーバを起動するためのものです。
<code>/etc/init.d/ postfix</code>	このスクリプトは、 postfix プロセスを制御するためのものです。
<code>/etc/init.d/ ypserv</code>	このスクリプトは、NIS サーバを起動するためのものです。
<code>/etc/init.d/ ypbind</code>	このスクリプトは、NIS クライアントを起動するためのものです。

21.7 ダイアルアップ接続支援としての **smpppd**

ホームユーザの場合、インターネットに対する専用の回線を持っておらず、ダイアルアップ接続を利用している場合があります。ダイアルアップの方式 (ISDN や DSL) によって、接続を **ippdpd** で操作するのか **pppd** で操作するのかが変わります。基本的にはインターネット接続を行なうのに必要なものは、これらのツールが全てです。

ダイアルアップ接続を行なうのに追加の料金がかからない固定料金制の接続をご利用の場合は、単に関連するデーモンを起動するだけです。デスクトップのアプレットやコマンドラインインターフェイスを利用してダイアルアップ接続を制御してください。インターネットのゲートウェイが今現在使用中のホストとは異なるものであったりする場合は、ネットワークホストを経由してダイアルアップ接続を利用することもできます。

こういう場合に **smpppd** (SUSE Meta PPP Daemon) が役に立ちます。 **smpppd** は補助プログラムに対する統一インターフェイスとして動作し、双方向の仲介を行ないます。 **smpppd** では、 **pppd** と **ippdpd** のどちらを利用するのかを判断

し、ダイヤルアップ接続の操作を行ないます。また、ユーザプログラムに対して様々なプロバイダが利用できるよう情報を提供するほか、接続状態についての情報を送信することもできます。smpppdはネットワーク経由でもコントロールできますので、プライベートなサブネット内に存在するワークステーションから、インターネットへのダイヤルアップ接続をコントロールする場合に便利です。

21.7.1 smpppd の設定

smpppd が提供する接続は、YaST から自動的に設定されます。実際のダイヤルアッププログラムである KInternet と cinternat についても設定が行なわれます。リモートコントロールなど、smpppd の追加機能を利用する場合にのみ、手動設定が必要となります。

smpppd の設定ファイルは /etc/smpppd.conf にあります。既定ではリモートコントロールは無効に設定されています。この設定ファイルで最も重要なオプションは、下記の通りです:

open-inet-socket = yes / no

smpppd をネットワーク経由で操作させたい場合は、このオプションを yes に設定してください。smpppd はポート 3185 で待ち受けます。このパラメータを yes に設定する場合は、bind-address, host-range, password についてもそれぞれ設定を行なってください。

bind-address = ip アドレス

ホストに複数の IP アドレスが設定されている場合は、このパラメータを利用して smpppd が受け付ける IP アドレスを指定します。既定では全てのアドレス宛の接続を受け付けます。

host-range = 最小 IP 最大 IP

host-range パラメータでは、ネットワークの範囲を指定します。smpppd では、指定した範囲の中にある IP アドレスからの接続だけを受け入れます。範囲外のアドレスからのアクセスは全て拒否されます。

password = パスワード

パスワードを割り当てると、ネットワーク経由での操作を行なうのにパスワードを入力しなければなりません。ただしここで指定するパスワードには何も暗号化を行なわないため、設定しないよりは良い、程度

のものとお考えください。何もパスワードを指定しないと、全てのクライアントから **smpppd** にアクセスできるようになります。

slp-register = *yes/no*

このパラメータを設定すると、**smpppd** サービスは SLP を介してアナウンスされるようになります。

smpppd についてさらに詳しい情報については、**smpppd(8)** や **smpppd.conf(5)** のマニュアルページをお読みください。

21.7.2 リモートで使用するための KInternet と cinternet の設定

KInternet や cinternet は、ローカルやリモートの **smpppd** を操作することができます。cinternet はコマンドラインツールで、KInternet はグラフィカルなツールです。これらのユーティリティをリモートの **smpppd** に対して使用するには、設定ファイル **/etc/smpppd-c.conf** を手作業で編集するか、もしくは KInternet を使用する必要があります。このファイルには 4 つのオプションだけがあります：

sites = サイトの一覧

フロントエンドが **smpppd** を検索する際に利用する、サイトの一覧を指定します。フロントエンドは、ここで指定した順序どおりにオプションを試します。**local** を指定するとローカルの **smpppd** に対して接続を行なう指定となるほか、**gateway** を指定するとゲートウェイ上の **smpppd** に対して接続を行なおうとします。**config-file** を指定すると、それぞれ **/etc/smpppd-c.conf** で指定する **server** と **port** の設定に従って接続を行ないます。**slp** を指定すると、SLP 経由で **smpppd** を見つけるようになります。

server = サーバ

smpppd が動作しているホストを指定します。

port = ポート

smpppd が動作しているポートを指定します。

password = パスワード

smpppd に接続するためのパスワードを指定します。

smpppd が動作している場合、まずはアクセスしてみてください。たとえば `cinternet --verbose --interface-list` のように実行することができます。この時点で何らかの問題が発生した場合は、`smpppd-c.conf(5)` のマニュアルページを参照するか、もしくは `cinternet(8)` のマニュアルページをお読みください。

ネットワーク内の SLP サービス

22

service location protocol (SLP) は、ローカル ネットワーク内にあるクライアントに対して設定を簡略化するために開発されました。 ネットワーククライアントを設定する際には、必要となる様々なサービスを含め、従来は管理者に対してネットワーク内のサーバについての詳しい知識が求められてきました。 SLP は特定のサービスが提供されていることを、ローカルネットワーク内の全てのクライアントに対して通知します。 SLP に対応したアプリケーションは、配布された情報を利用してサービスにアクセスできるようになり、自動で設定したりすることもできるようになります。

openSUSE® では SLP 経由で通知されたインストール元からのインストールに対応しているほか、 SLP に対応した多くのシステムサービスが含まれています。 また、 YaST や Konqueror では SLP のフロントエンドも提供されています。 お使いのシステム内にあるインストールサーバやファイルサーバ、印刷サーバなど 集中型の機能について、ネットワーク対応のクライアントに SLP 経由で通知して 使用してもらうように設定することもできます。

重要項目: openSUSE での SLP サポート

SLP に対応するサービスには、 cupsd, rsyncd, ypserv, openldap2, ksysguardd, saned, kdm, vnc, login, smpppd, rpasswd, postfix, sshd (fish 経由) があります。

22.1 インストール

SLP サービスを利用するのに必要な全てのパッケージは、既定でインストールされるようになっていました。逆に SLP を介してサービスを提供したい場合は、`openslp-server` パッケージがインストールされていることを確認してください。また、SLP のデーモンのサーバ設定を行なうには、`yast2-slp-server` パッケージをインストールしてください。

22.2 SLP の有効化

SLP でサービスを公開するには、お使いのシステムで `slpd` を実行しなければなりません。お使いのマシンがクライアントとしてのみ動作するもので、サービスの提供を行なわない場合は、`slpd` を実行する必要はありません。

openSUSE での多くのサービスのように、`slpd` デーモンは個別の `init` スクリプト経由で起動および停止することができます。インストール後、何も設定しない場合はサービスは起動しません。一時的に有効化したい場合は、`root` で `rcslpd start` と入力すると起動を行なうことができますし、`rcslpd stop` と入力すると停止することができます。それ以外にも、再起動や状態チェックを `restart` や `status` で行なうことができます。起動時に `slpd` を常に開始させたい場合は、YaST から システム > システムサービス (ランレベル) を選択して `slpd` を有効にするか、もしくは `root` で `insserv slpd` コマンドを実行してください。

22.3 openSUSE での SLP フロントエンド

お使いのネットワーク内で提供されているサービスを SLP 経由で見つけるには、`slptool` (`openslp` パッケージ) のような SLP フロントエンドを利用するか、もしくは YaST を利用します:

`slptool`

`slptool` はコマンドラインプログラムで、ネットワーク内に対して SLP の問い合わせを投げたり、独自のサービスをアナウンスしたりすることができます。 `slptool --help` のように実行すると、利用可能なオプション

ンと機能を表示することができます。たとえば現在のネットワークで、自分自身の存在をアナウンスしている ネットワーク時刻サーバを見つけるには、下記のように入力します:

```
slptool findsrvs service:ntp
```

YaST

YaST でも SLP ブラウザを利用することができます。ただしこのブラウザは YaST のコントロールセンターの一覧には用意されていません。この YaST モジュールを起動するには、root で `yast2 slp` と入力してください。左側にある サービス種類 の項目を選択すると、サービスに関するさらに 詳しい情報を得ることができます。

22.4 SLP 経由でのインストール

openSUSE のインストールメディアを利用して、お使いのネットワーク内にインストールサーバを立ち上げたい場合は、SLP で登録および提供することができます。詳しくは 1.2 項 「インストール元のデータを保存するサーバの構築」 (13 ページ) をお読みください。選択した起動メディアからシステムを起動し、SLP インストールを選択すると、linuxrc は SLP の問い合わせを行なって見つかったソースを表示します。

22.5 SLP 経由でのサービス提供

openSUSE 内の多くのアプリケーションは、libslp を使用することで SLP に対応しています。あるサービスについて SLP に対応したコンパイルが行なわれていない場合は、下記のいずれかの方法で SLP 対応にすることができます:

/etc/slp.reg.d への手動登録

それぞれ新しいサービスに対して、個別の登録ファイルを作成するやり方です。下記はスキャナサービスを登録するためのファイル例です:

```
## このシステムにある saned サービスを登録します。
## en は英語という意味です。
## 65535 はタイムアウトを無効にするための指定で、サービスの登録時には更新を
## 必要としない意味になります。
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

このファイルで最も重要な行は、`service:` で始まる サービス *URL* の行です。この行にはまず、サービスの種類 (`scanner.sane`) と、サービスを利用できるアドレスが書かれています。なお、`$HOSTNAME` は自動的に自身の完全修飾ホスト名に置き換えられます。また、コロン (`:`) で区切ってサービスを提供している *TCP* ポートの名前を入力することもできます。あとの項目はサービスで使用されている言語と登録間隔 (秒単位) が書かれています。サービス *URL* との区切りは、それぞれカンマを使用します。登録間隔は 0 から 65535 までの間で、0 を指定すると登録を行なわない意味になり、65535 を指定すると全ての制限を取り除く意味になります。

また、登録ファイルには `watch-port-tcp` と `description` の項目も書かれています。`watch-port-tcp` は *SLP* サービスアナウンスと関連するサービスの有効性をつなげるもので、このポートに接続可能かどうかでサービスが提供されているかどうかを確認します。もう1つの項目はサービスに対する詳しい説明を書くための項目で、ブラウザを利用することで表示することのできる項目です。

`/etc/slp.reg` への手動登録

この登録と `/etc/slp.reg.d` での登録との違いは、全てのサービスが単一のファイルに書かれているという点です。

`slptool` を利用した動的な登録

あるサービスについて、設定ファイル無しで動的に登録する必要がある場合は、`slptool` コマンドラインユーティリティを使用するのがよいでしょう。同じツールを利用することで、`slpd` の再起動を行なうことなく既存のサービスを削除することができます。

22.6 さらに情報

RFC 2608, 2609, 2610

RFC 2608 では一般的な *SLP* の定義について記述しています。RFC 2609 ではサービス *URL* についてさらに詳しい文法を説明しているほか、RFC 2610 では *SLP* 経由での *DHCP* について扱っています。

<http://www.openslp.org>

OpenSLP プロジェクトのホームページです。

/usr/share/doc/packages/openslp

このディレクトリには、openslp-server パッケージに添付されているドキュメンテーションのほか、README.SuSE ファイルとして openSUSE での詳細や RFC、2 種類の入門 HTML 文書が配置されます。SLP の機能を使用したいプログラマの場合は、openslp-devel パッケージに含まれる プログラマーズガイドをお読みになるのがよいでしょう。

ドメインネームシステム

DNS (ドメインネームシステム) はドメイン名やホスト名を IP アドレスに変換するのに必要なサービスです。この方法により、たとえば 192.168.2.100 は jupiter というホスト名と対応づけることができます。独自のネームサーバを設定する前に、21.3項「名前解決」(371 ページ) に示されている DNS に関する一般情報をお読みください。また、下記の設定例では BIND を利用しています。

23.1 DNS 用語

ゾーン

ドメインの名前領域は、ゾーンと呼ばれる部分に分割されています。たとえば example.com というドメインでは、com と呼ばれるドメイン内の example という部分 (ゾーン) を示していることになります。

DNS サーバ

DNS サーバは、それぞれのドメインに対して、その名前と IP アドレスの情報を保持しているサーバです。マスターゾーンとしてプライマリ DNS サーバを、スレーブゾーンとしてセカンダリ DNS サーバを持つことができるほか、ゾーンを持たないキャッシュ専用のスレーブサーバを持つこともできます。

マスターゾーンの DNS サーバ

マスターゾーンには、お使いのネットワークに関する全ての情報が含まれていて、ドメインに対する全ホストに関する最新情報が保存されています。

スレーブゾーンの DNS サーバ

スレーブゾーンはマスターゾーンのコピーです。スレーブゾーンの DNS サーバは、マスターサーバからゾーン転送処理を通してゾーンデータを取得します。スレーブゾーンの DNS サーバは、そのゾーンデータが有効である限り (有効期限が切れない限り)、そのゾーンに対して権威のある (authoritative な) 応答を返却します。スレーブはそのゾーンデータの最新情報が得られない場合、そのゾーンに対する応答を停止します。

フォワーダ

フォワーダとは、お使いの DNS サーバが応答のできない場合に問い合わせを転送する先のことを言います。1つの設定で複数の設定ソースを指定するには、netconfig を使用します (詳しくは `man 8 netconfig` をお読みください)。

レコード

レコードとは名前や IP アドレスに関する情報のことを言います。それぞれ対応しているレコードとその文法については、BIND のドキュメンテーションに書かれています。また、下記のような特殊なレコードがあります:

NS レコード

NS レコードとは、指定したドメイン名を担当するネームサーバのマシンがどこにあるのかを知らせるためのものです。

MX レコード

MX (メールエクスチェンジ) レコードとは、インターネットを介して直接メールをやりとりする場合に使用する、通信相手のマシンを教えるためのものです。

SOA レコード

SOA (権威の開始) レコードとは、ゾーンファイルの最初のレコードです。SOA レコードは、DNS を通じて複数のコンピュータ間でデータを同期する場合に利用します。

23.2 インストール

DNS サーバをインストールするには、YaST を起動して `ソフトウェア>ソフトウェア管理` を選択します。そこからさらに `フィルタ>パターン` を選択し、

DHCP and DNS Server を選択してください。あとはインストール処理を完了するため、個別のパッケージのインストールについて確認を行なってください。

23.3 YaST を利用した設定

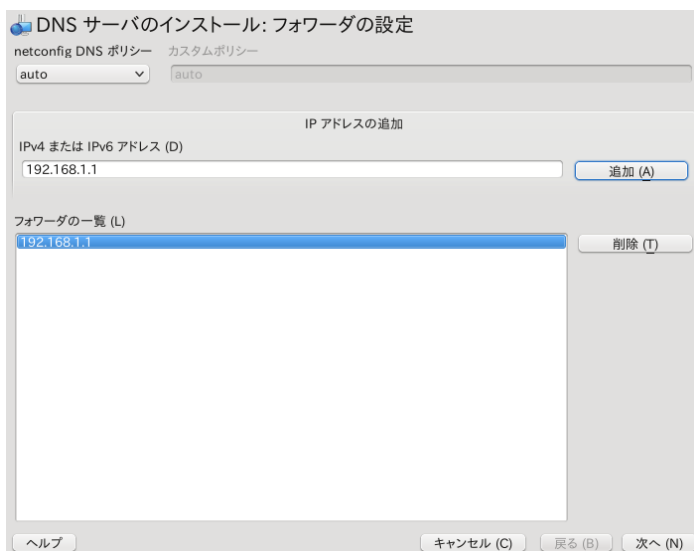
YaST DNS モジュールを利用することで、お使いのローカルネットワークに対する DNS サーバの設定を行なうことができます。最初にモジュールを起動したときには ウィザードが起動し、サーバを起動するにあたっていくつか決めておかなければならない 基本設定について質問が行なわれます。この初期設定作業はサーバの基本的な設定を行なうもので、アクセス制御 (ACL) やログ記録、TSIG 鍵のオプションなど、高度な設定作業については、熟練者モードをご利用ください。

23.3.1 ウィザードによる設定

ウィザードは 3 段階の手順 (ダイアログ) から構成されています。それぞれダイアログ 内の適切な箇所から、熟練者モードに入ることができます。

- 1 最初にモジュールを起動したときには、図23.1「DNS サーバのインストール: フォワーダの設定」(430 ページ) に示されている フォワーダ設定の画面が開きます。 *netconfig DNS* ポリシーでは、どのデバイスから フォワーダにアクセスするかを設定できるほか、フォワーダの一覧から独自の一覧を設定することもできます。 *netconfig* について詳しくは、`man 8 netconfig` で表示されるマニュアルページをお読みください。

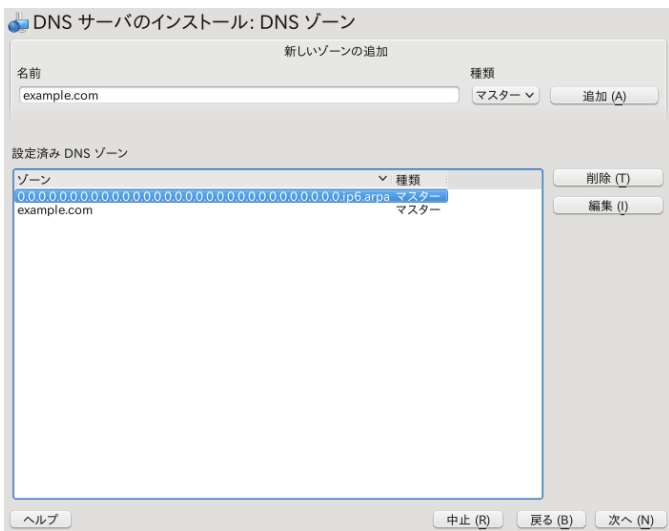
図 23.1 DNS サーバのインストール: フォワーダの設定



フォワーダとは、お使いの DNS サーバが問い合わせを送信する先のこと
で、自分自身では解決できない問い合わせの場合に利用します。フォワー
ダの IP アドレスを 入力してから **追加** を押してください。

- 2 次の DNS ゾーンダイアログは 23.6 項「ゾーンファイル」(446 ページ) にあ
るとおり複数のパーツから構成され、ゾーンファイルの管理について設定
を行ないます。新しいゾーンを作成するには、**名前**に名前を入力してくだ
さい。逆引きゾーンを追加する場合、名前は `.in-addr.arpa` で終わらな
ければなりません。最後に **種類**(マスター、スレーブ、転送のいずれか)
を選択してください。詳しくは 図 23.2 「DNS サーバのインストール: DNS
ゾーン」(431 ページ) をお読み ください。既存のゾーンについてその他の
設定を変更するには、設定済み DNS ゾーンから選択して、**編集** ボタンを
押してください。ゾーンを削除するには同じくゾーンを選択してから **削除**
ボタンを押してください。

☒ 23.2 DNS サーバのインストール: DNS ゾーン



- 3 最後のダイアログでは、ファイアウォールでポートを開くを選択することでファイアウォール内の DNS ポートを開くように設定することができます。また、DNS サーバをシステム起動時に開始するかどうか(オンまたはオフ)を選択することもできます。またそれ以外にも、LDAP に対応するよう設定することもできます。詳しくは図23.3「DNS サーバのインストール: ウィザードの完了」(432 ページ)をご覧ください。

☒ 23.3 DNS サーバのインストール: ウィザードの完了



23.3.2 熟練者向け設定

モジュールを起動したあと、YaST では複数の設定オプションを含んだウィンドウを表示します。これらを設定することで、DNSサーバに対して基本的な機能を設定することができます:

起重動

起動では、DNS サーバをシステム起動時に開始するか、もしくは手動で起動するかを選択することができます。DNS サーバを今すぐに起動したい場合は、今すぐ *DNS* サーバを起動 を押してください。また、DNS サーバを停止したい場合は、今すぐ *DNS* サーバを停止 を押してください。現在の設定を保存する場合は、設定を保存して今すぐ *DNS* サーバを再起動する を押してください。また、ファイアウォールでポートを開くを押すことで、ファイアウォールでDNS ポートを開くことができるほか、ファイアウォールの詳細でファイアウォール設定を変更することもできます。

それ以外にも **LDAP** サポートを有効にするを選択すると、ゾーンファイルは **LDAP データベース** で管理されるようになります。 **LDAP データベース** 内の

ゾーンデータに対する変更は、DNSサーバが再起動されたり設定を読み込みなおしたりしたタイミングで読み直されます。

フォワーダ

お使いの DNS サーバが要求に応答できない場合、フォワーダで設定した転送先に対して要求を転送します。フォワーダの一覧に手動で追加してください。ダイアルアップ接続のなどのようにフォワーダが随時変わるような場合は、`netconfig` が設定を扱います。`netconfig` について、詳しくは `man 8 netconfig` で表示される マニュアルページをお読みください。

基本オプション

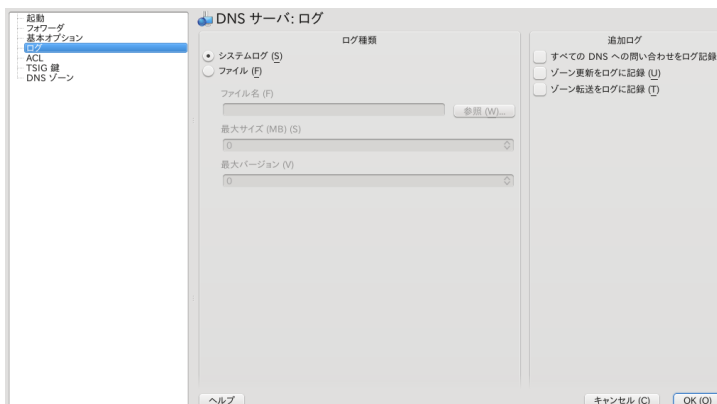
このセクションでは、サーバに対する基本的な設定を行ないます。オプションメニュー内から設定したい項目を選び、必要な値を設定してください。追加ボタンを押すと新しい項目を追加することができます。

ログ

DNS サーバのログ機能について、何を採取するのか、どのように採取するのかを ログ から設定します。ログ種類では、DNS サーバがログデータをどこに書き込むのかを設定します。`/var/log/messages` のようなシステム全体のログに対して記録を行なうにはシステムログを、個別のファイルに保存するには ファイル を選択します。後者の場合はログファイル名とメガバイト単位での最大サイズ、および保存する過去ログの最大数をそれぞれ指定してください。

さらなるオプション設定は、追加ログから利用できます。すべての DNS への問い合わせをログ記録を選択すると、それぞれの問い合わせに対し、ログ記録を行なうようになります。ログファイルは非常に大きいものになるため、デバッグ目的以外の使用は適切とは言えません。また、DHCP サーバや DNS サーバの間でのゾーン更新トラフィックを記録するには、ゾーン更新をログに記録を選択してください。また、マスターサーバからスレーブサーバに対して行なわれたゾーン転送トラフィックを記録するには、ゾーン転送をログに記録を選択してください。詳しくは 図23.4 「DNS サーバ: ログ」 (434 ページ) をご覧ください。

☒ 23.4 DNS サーバ: ログ



ACL

このダイアログでは、**ACL**(アクセス制御リスト)を設定し、アクセスに対する制限を設定することができます。それぞれ固有の **名前**を設定したあとは、**値IP** アドレス(ネットマスク付き、または単独)の欄に下記の形式で入力します:

```
{ 192.168.1/24; }
```

設定ファイルの書式仕様により、アドレスはセミコロン(;)で終わり、かつ中括弧でくくる必要があります。

TSIG 鍵

TSIG(トランザクション署名)の主な目的は、**DHCP** や **DNS** サーバ間での通信について、機密を保持することにあります。詳しい説明は23.8項「機密を保持する通信」(451 ページ)で行なっています。

TSIG 鍵を生成するには、**鍵ID**の欄に固有の名前を入力し、保存先のファイル名(ファイル名)を指定します。最後に **生成**を押すと鍵が生成されます。

以前に作成した鍵を使用するには、**鍵ID**の項目に何も記入しない状態で、鍵ファイルが保存されている場所を **ファイル名**で指定します。最後に **追加**を押すと追加することができます。

DNS ゾーン (スレーブゾーンの追加)

スレーブゾーンを追加するには、*DNS* ゾーンを選んでから ゾーン種類としてスレーブを選択し、ゾーンの名前を入力してから *追加* を押します。

ゾーンエディタのサブダイアログでは、マスター *DNS* サーバの *IP* を指定し、どのサーバからデータを取得するのかを設定してください。また、サーバに対するアクセスを制限するには、*ACL* の一覧からそれぞれ選択してください。

DNS ゾーン (マスターゾーンの追加)

マスターゾーンを追加するには、*DNS* ゾーンを選んでから ゾーン種類としてマスターを選択し、ゾーンの名前を入力してから *追加* を押します。マスターゾーンを追加する場合は、それら対応する逆引きゾーンも必要です。たとえば `192.168.1.0/24` というサブネット内にあるホストを示す `example.com` というドメインを追加する場合は、それらの *IP* アドレス範囲をカバーする逆引きゾーンを追加する必要があります。たとえば `1.168.192.in-addr.arpa` のように設定します。

DNS ゾーン (マスターゾーンの編集)

マスターゾーンを編集するには、*DNS* ゾーンを選んでから一覧内のマスターゾーンのいずれかを選択し、*編集* を押します。表示されるダイアログにはいくつかのページが含まれます: *基本* (最初に表示されるページ), *NS* レコード, *MX* レコード, *SOA*, レコードがあります。

図23.5「DNS サーバ: ゾーンエディタ (基本)」(436 ページ) で示されている基本ダイアログでは、動的 *DNS* とクライアントに対するゾーン転送のアクセスオプション、およびスレーブのネームサーバを設定することができます。ゾーンに対して動的な更新を許可するには、*動的な更新の許可* を選択して関連する *TSIG* 鍵を選んでください。この鍵は更新動作が始まる前までに設定しておかなければなりません。ゾーン転送を許可するには、必要な *ACL* を選んでください。この段階までに *ACL* を設定しておかなければなりません。

基本ダイアログでは、ゾーン転送を有効にするかどうかを指定します。また、どのホストからゾーンをダウンロードできるかを *ACL* で指定します。

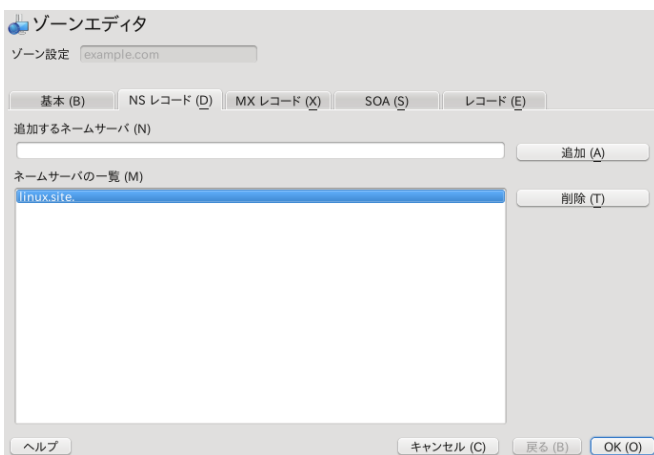
図 23.5 DNS サーバ: ゾーンエディタ (基本)



ゾーンエディタ (NS レコード)

NS レコードダイアログでは、指定したゾーンに対する 代替のネームサーバを設定することができます。一覧内にはご自身が管理されるネームサーバだけが含まれていることをご確認ください。レコードを追加するには、追加するネームサーバの欄にアドレスを入力し、追加を押します。詳しくは図23.6「DNS サーバ: ゾーンエディタ (NS レコード)」(437 ページ)をご覧ください。

図 23.6 DNS サーバ: ゾーンエディタ (NS レコード)



ゾーンエディタ (MX レコード)

現在のゾーンに対してメールサーバを一覧に追加するには、まずそれぞれ適切なアドレスと値を入力します。入力を行ったら *追加* を押します。詳しくは 図23.7 「DNS サーバ: ゾーンエディタ (MX レコード)」 (437 ページ) をご覧ください。

図 23.7 DNS サーバ: ゾーンエディタ (MX レコード)



ゾーンエディタ (SOA)

このページでは、SOA(権威の開始)レコードを作成することができます。各オプションについての説明は、例23.6「`/var/lib/named/example.com.zone` ファイル」(446 ページ)をお読みください。

☒ 23.8 DNS サーバ: ゾーンエディタ (SOA)

ゾーンエディタ (レコード)

このダイアログでは、名前解決の管理を行ないます。レコードキーにホスト名を入力し、種類を選択してください。A レコードを主に設定します。この場合、値にはIP アドレスを入力します。また、CNAME では別名を指定します。なお、NS や MX の各レコードについては、それぞれ NS レコードや MX レコードで設定した情報について、より詳細な設定を行ったり、部分的に修正したりする場合にお使いください。これら3種類のレコードでは、いずれも既存の A レコードを参照します。また、PTR は逆引きゾーンを表わします。これは A レコードの逆を意味するもので、たとえば下記ようになります:

```
hostname.example.com. IN A 192.168.0.1
1.0.168.192.in-addr.arpa IN PTR hostname.example.com.
```

注記: 逆引きゾーンの編集

正引きゾーンを追加したあと、メインメニューに戻ってから編集したい逆引きゾーンを選択して基本タブを表示すると、以下のものから自動的にレコードを生成チェックボックスを利用できるようになります

す。このチェックボックスを選択してゾーンを選択すると、自動的に逆引きゾーンを生成することができます。この方法を利用すると、正引きゾーンを書き換えるだけで自動的に逆引きゾーンにも反映されるようになります。

23.4 BIND ネームサーバの起動

openSUSE® システムでは、ネームサーバ BIND (*Berkeley Internet Name Domain*) が事前構築される形で提供されていて、インストールを行なうだけですぐに動作するようになっています。そのため、インターネット接続を利用できる環境であれば、ネームサーバを設定したあと、ネームサーバのアドレスに 127.0.0.1 (つまり localhost) を指定すれば、プロバイダから DNS サーバの情報を得ることなく名前解決ができることを示しています。この場合、BIND はルートネームサーバから名前解決を行なおうとするため、著しく動作が遅くなってしまいます。通常の運用であれば /etc/named.conf 内の forwarders 設定に対してプロバイダのネームサーバを設定して効率を高めますが、このような方法で名前解決を行なう場合、そのネームサーバはそのゾーンに対し、純粋な キャッシュのみのネームサーバとして動作することになります。設定例はドキュメンテーション /usr/share/doc/packages/bind/config をお読みください。

ティップ: ネームサーバ情報の自動取得

インターネット接続やネットワーク接続の種類によっては、ネームサーバの情報を自動的に現在の状況に合わせることができます。これを行なうには、/etc/sysconfig/network/config ファイル内にある MODIFY_NAMED_CONF_DYNAMICALY 変数を auto に設定してください。

ただし、責任ある団体でドメインが割り当てられない限りは、公式なドメインを設定してはなりません。独自のドメインをお持ちの場合で、それがプロバイダ側の管理下にあるものであっても使用しないほうがお勧めです。使用してしまうと、BIND は要求を転送してしまうためです。たとえばプロバイダ内の Web サーバなどから、このドメインにアクセスできなくなってしまいます。

ネームサーバを開始するには、root ユーザから rcnamed start コマンドを実行します。右側に緑色で「done」と表示されれば named (ネームサーバ

のプロセス)は正しく起動したことを示します。起動後はhost プログラムやdig プログラムを利用してローカルシステムからネームサーバの動作テストを行なうことができます。既定のサーバに対してlocalhostを問い合わせ、127.0.0.1 が返却されれば成功です。応答が返されなかったり、正しいアドレスになっていなかったりした場合は、/etc/resolv.conf に正しいネームサーバのアドレスが書かれていないか、もしくは全く設定されていないことが考えられます。まずは最初のテストとしてhost 127.0.0.1 と入力してみてください。このコマンドは設定にかかわらず常に成功すべきものであるため、これがエラーになるようであれば、まずrcnamed status を実行してサーバが実際に動作しているかどうかを確認してください。ネームサーバが起動してなかったり、期待通りの動作をしていなかったりした場合は、/var/log/messages ファイルから原因を調査してください。

プロバイダが提供するネームサーバで転送機能が提供されている場合は(もしくはお使いのネットワークで転送機能の利用できるネームサーバが動作している場合は)、options セクション内の forwarders に、それらの IP アドレスを入力してください。例23.1「named.conf 内の転送設定」(440 ページ)に例を示します。お使いの環境でそれぞれアドレス設定を変えてお使いください。

例 23.1 named.conf 内の転送設定

```
options {
    directory "/var/lib/named";
    forwarders { 10.11.12.13; 10.11.12.14; };
    listen-on { 127.0.0.1; 192.168.1.116; };
    allow-query { 127/8; 192.168/16 };
    notify no;
};
```

options 項目はlocalhost や 0.0.127.in-addr.arpa のゾーン項目の後に続く項目です。「.」に対するtype hintの項目も存在すべき項目です。それぞれのファイルをわざわざ編集する必要はなく、そのままの形で動作するはずで、また、各項目の行末に「;」が付けられていることと、適切な箇所に中括弧が入っていることもあわせてご確認ください。設定ファイル/etc/named.confやゾーンファイルを編集したら、rcnamed reload コマンドでBINDに対して設定を読み直すように指示してください。サーバを再起動することでも同じことを行なうことができます。再起動を行なうにはrcnamed restart コマンドを入力してください。また、サーバを停止するには、rcnamed stop と入力します。

23.5 /etc/named.conf 設定ファイル

BIND ネームサーバに対する全ての設定は、`/etc/named.conf` ファイル内に保存されます。ただし、応答を返すべきドメインに対するゾーンデータ (ホスト名や IP アドレスなど) については、`/var/lib/named` ディレクトリ内にある個別のファイルに保存します。ゾーンファイルについての詳細は後述します。

`/etc/named.conf` ファイルは大きく分けて2つの領域に分かれています。1つは `options` セクションで一般的な設定を行ない、もう1つは `zone` セクションでドメインごとの設定を行ないます。 `logging` セクションや `acl` (アクセス制御リスト) のセクションは必要に応じて指定します。コメント行は `#` から書き始めるか、もしくは `//` から書き始めます。 `/etc/named.conf` の最小限の設定は、例23.2「`/etc/named.conf` の基本設定」(441 ページ)に示しているとおります。

例 23.2 `/etc/named.conf` の基本設定

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};
```

23.5.1 重要な設定オプション

`directory "ファイル名";`

BIND がゾーンデータを見つける際に利用するディレクトリを指定します。通常、この値は `/var/lib/named` に設定します。

`forwarders { IP アドレス; };`

自分自身で直接解決できない場合に、DNS の要求を転送する先になるネームサーバ (一般にプロバイダ提供のネームサーバ) を指定します。IP アドレスの欄には、`192.168.1.116` のように設定してください。

`forward first;`

DNS 要求に対して、ルートネームサーバ経由での解決を行なう前に転送を試すように指示します。`forward first` の代わりに `forward only` と設定すると、全ての要求をルートネームサーバに送信せず、転送するようになります。ファイアウォールの設定を行なう場合に意味のある項目です。

`listen-on port 53 { 127.0.0.1; IP アドレス; };`

BIND に対して、どのインターフェイスとポートでアクセスを待ち受けるのかを指定します。`port 53` については既定のポートであるため、明示的に指定する必要はありません。また、`127.0.0.1` と指定すると、ローカルホストからの要求だけを許可するようになります。この項目を設定しない場合は、既定で全てのインターフェイスを待ち受けます。

`listen-on-v6 port 53 {any; };`

BIND に対して、IPv6 クライアントからの要求を処理するように指定します。`any` 以外には `none` しか指定できません。IPv6 に関しては、全アドレスからのアクセスを受け付ける ことになります。

`query-source address * port 53;`

このオプションは、ファイアウォールによって外向きの DNS 要求がブロックされてしまう場合に必要な設定です。上記の設定では、BIND の要求をポート 53 から行ない、1024 以上のポートを使用しないように指定しています。

`query-source-v6 address * port 53;`

IPv6 での問い合わせ時に使用するポートを設定します。

allow-query { 127.0.0.1; ネットワーク; };

DNS の要求を受け付けるネットワークを指定します。ネットワークは実際のアドレス (たとえば 192.168.2.0/24) に置き換えてください。なお、/24 とはネットマスクの省略表現です (この場合、255.255.255.0 を意味します)。

allow-transfer ! *;;

どのホストに対してゾーン転送を許可するかを制御します。この例では ! * を指定しているため、ゾーン転送を完全に拒否 するようになります。この項目を設定しないと、ゾーン転送に対する制限が行なわれず、どのホストからでも行なうことができるようになります。

statistics-interval 0;

この項目を設定しないと、BIND は複数の行から構成される統計情報を 1 時間 おきに /var/log/messages に出力します。0 を指定するとこれらの統計情報を省略することができます。それ以外の値であれば、分単位で出力間隔を指定することになります。

cleaning-interval 720;

このオプションでは、BIND がその内蔵キャッシュを消去する時間間隔を指定します。この時間間隔で /var/log/messages に対して情報が出力されます。時間は分単位で指定します。既定値は 60 分です。

interface-interval 0;

BIND では定期的にネットワークインターフェイスの追加や削除を検索します。この値を 0 にすると、ネットワークインターフェイスの 検出作業を行なわなくなり、起動時に検出したインターフェイスに対してのみ 要求を受け付けるようになります。0 以外の設定を行なうと、それは分単位での設定になります。既定値は 60 分です。

notify no;

no を設定すると、ゾーンデータの変更やネームサーバの再起動が発生するごとに発生していた、ネームサーバ宛の通知を行なわなくなります。

利用可能なオプションの一覧については、man 5 named.conf で表示されるマニュアルページをお読みください。

23.5.2 ログ

BINDでは、何をどのように、どこでログを採取するのかを広範囲に設定することができます。通常は既定の設定のままで問題ありません。例23.3「ログ出力を無効化する場合の設定」(444 ページ)では、このようなログを完全に省略するために最も簡単な方法を提示しています。

例 23.3 ログ出力を無効化する場合の設定

```
logging {  
    category default { null; };  
};
```

23.5.3 ゾーン項目

例 23.4 *example.com* のゾーン項目

```
zone "example.com" in {  
    type master;  
    file "example.com.zone";  
    notify no;  
};
```

まず、zone の後に管理したいドメインの名前を指定し (上記の例では example.com)、さらに in を指定します。あとは関連するオプションを記述するための中括弧を記述し、例23.4「example.com のゾーン項目」(444 ページ)のような構成になります。スレーブゾーンを定義する場合は type に対して slave を指定し、このゾーンを管理するネームサーバ (別のサーバ) を master で指定します。たとえば例23.5「example.net のゾーン項目」(444 ページ)のようになります。

例 23.5 *example.net* のゾーン項目

```
zone "example.net" in {  
    type slave;  
    file "slave/example.net.zone";  
    masters { 10.0.0.1; };  
};
```

ゾーンのオプションには下記のようなものがあります:

type master;

master を指定すると、**BIND** に対してこのゾーンをこのネームサーバで管理するようになります。また、ゾーンファイルは正しい書式で作成されているものと解釈します。

type slave;

このゾーンは、他のネームサーバから情報を転送することで成立させるゾーンです。masters の設定とともに使用しなければなりません。

type hint;

hint タイプが設定されたゾーンは、ルートネームサーバを設定するために使用します。このゾーンの定義はそのまま残しておいてください。

file example.com.zone または **file 「slave/example.net.zone」;**

この設定では、そのゾーンに対するゾーンデータがどのファイルに保存されているものなのかを指定します。スレーブゾーンの場合は、このファイルは他のネームサーバから転送してくることになるため、存在している必要はありません。マスターゾーンの場合とスレーブゾーンの場合の違いは、slave というディレクトリを使用しているかどうかの違いです。

masters { サーバの IP アドレス; };

この項目は、スレーブゾーンでのみ必要な設定です。ゾーンファイルをどのネームサーバから転送するのかを指定します。

allow-update {! *; };

このオプションでは、外部からの書き込みアクセス (DNS の項目をクライアント側に作成する機能を与えるもの) を制御します。通常はセキュリティ上の理由から使用することはありません。この項目を設定しないと、ゾーンの更新は全く行なうことができなくなります。上記の設定は何も指定しない場合と同じ意味で、! * で全てに対する拒否を行なっています。

23.6 ゾーンファイル

ゾーンファイルについては、2種類のものが必要です。1つはホスト名に対してIPアドレスを割り当ててるもの、もう1つはその逆で、IPアドレスに対するホスト名を割り当ててるものです。

ティップ: ゾーンファイル内でのドット (ピリオド、終止符) の使用

ゾーンファイル内では "." は重要な意味を持ちます。ホスト名の最後が . で終わっていない場合、該当するゾーンが付加されます。完全修飾ドメイン名で指定したい場合は、さらにドメインが付加されることを避けるため、. で終わらなければなりません。ネームサーバの設定ミスとしては、 "." を付け忘れていたり、正しい位置に付けなかったりなどが、よくあります。

最初に考慮しなければならないことはゾーンのファイル名で、たとえば example.com というドメインに対しては、 example.com.zone のような名前を設定します。たとえば 例23.6 「/var/lib/named/example.com.zone ファイル」 (446 ページ) のようになります。

例 23.6 /var/lib/named/example.com.zone ファイル

```
1. $TTL 2D
2. example.com. IN SOA      dns root.example.com. (
3.                     2003072441 ; serial
4.                     1D        ; refresh
5.                     2H        ; retry
6.                     1W        ; expiry
7.                     2D )      ; minimum
8.
9.                     IN NS     dns
10.                    IN MX     10 mail
11.
12. gate              IN A       192.168.5.1
13.                  IN A       10.0.0.1
14. dns               IN A       192.168.1.116
15. mail              IN A       192.168.3.108
16. jupiter           IN A       192.168.2.100
17. venus             IN A       192.168.2.101
18. saturn            IN A       192.168.2.102
19. mercury           IN A       192.168.2.103
20. ntp               IN CNAME   dns
21. dns6              IN A6      0      2002:c0a8:174::
```

1 行目:

\$TTL では、このファイル内の全ての項目に対して適用する、既定の生存時間 (Time To Live) を設定します。この例では、項目は 2 日間有効であり続けます (2 D)。

2 行目:

この行から SOA (権威の開始) の制御レコードが始まります:

- まずは管理対象のドメイン名を記述します (example.com)。このドメイン名は必ず "." で終わるものとしてください。そうでないとドメイン名の後ろに再度ドメイン名が付加されてしまいます。なお、ここには @ を入れることもできます。これは関連する /etc/named.conf の項目からドメイン名を自動的に取り出す場合に使用します。
- IN SOA に続いて、このゾーンに対するマスターのネームサーバ名を指定します。ここでは dns と指定していますが、これは "." で終わっていないため、自動的に dns.example.com の形に展開されます。
- ネームサーバ名に続いて、このドメインの担当者のメールアドレスを指定します。なお、ゾーンファイルで @ 記号は特別な意味を持つものであるため、この記号の代わりに "." で指定してください。たとえば root@example.com というメールアドレスであれば、root.example.com. と指定します。なお、この指定も "." で終わらせてください。そうでないとドメイン名がさらに付加されてしまいます。
- (から) までの間は SOA レコードの詳細情報を指定するための括りです。

3 行目:

ここではシリアル番号を指定します。シリアル番号は任意に付与することのできる番号で、このファイルの更新が行なわれるごとに数字を増やさなければならない項目です。このシリアル番号は、他のネームサーバ (スレーブサーバなど) に対して変更の通知を行なう際に必要となります。ここでは 10 桁の数値で年月日と追加の数字を設定しています (YYYYMMDDNN の形式) が、これは慣習的にこのようにしているものです。

4 行目:

ここでは 更新間隔 を指定します。更新間隔は他のネームサーバ(スレーブサーバなど)に対して、シリアル番号を検証する間隔を指定します。この場合、1 日間隔の意味になります。

5 行目:

ここでは 再試行間隔 を指定します。再試行間隔は他のネームサーバ(スレーブサーバなど)からのアクセスが失敗したときに、プライマリネームサーバ(マスターサーバ)に再接続する間隔を指定します。ここでは2 時間に設定されています。

6 行目:

ここでは 有効期限 を指定します。有効期限は他のネームサーバ(スレーブサーバ)からプライマリネームサーバ(マスターサーバ)へのアクセスが回復しない場合に、他のネームサーバがキャッシュデータを捨てるまでの時間間隔を指定します。ここでは1 週間に設定されています。

7 行目:

ここが SOA レコードの最後の部分で、ネガティブキャッシュの 生存時間 を指定します。これは、このドメインに対する問い合わせに対し、該当レコードが存在しない旨の応答を受けつけたときに、キャッシュする時間 間隔を指定します。

9 行目:

IN NS では、このドメインに対して管理責任を持つネームサーバを指定します。dns は "." で終わっていないため、dns.example.com に展開されます。この行は、プライマリ(マスター)ネームサーバとセカンダリ(スレーブ)ネームサーバのように、複数個を指定することもできます。/etc/named.conf 内で notify に対し no と設定されていない場合は、ここで指定した全てのネームサーバに対してゾーンデータの更新を通知します。

10 行目:

MX レコードには、そのドメイン(この例では example.com)に対するメールを受け付け、必要に応じて処理や転送を行なうメールサーバを指定します。この例では、mail.example.com がそのメールサーバであることを示しています。ホスト名の前につく数字は優先順位で、複数の **MX** レコードが存在する場合、まずは最も小さい値を持つメールサーバに

対して配送が試みられ、それが失敗した場合は次に小さい値を持つメールサーバに対して配送を行なおうとします。

12 行目から 19 行目:

これらの行で、複数のホスト名と IP アドレスの対応を記述しています。ここに記述された名前には "." が付けられておらず、ドメイン名が記述されていないため、それら全てに対して example.com が付加されます。なお、ホスト gate には 2 つのアドレスが割り当てられていますが、これはこのマシンに 2 枚のネットワークカードが接続されているためです。ホストのアドレスが従来のもの (IPv4) である場合は、このレコードは A レコードになります。ホストのアドレスが IPv6 アドレスである場合には、AAAA レコードを指定します。

注記: IPv6 での文法

IPv6 レコードは IPv4 とは異なる文法で記述されます。IPv6 では断片的なアドレス表記を行なうことができるため、アドレスの途中に「0」が続く場合は、その箇所に 2 つのコロン (:) を記述してください。

```
pluto      AAAA 2345:00C1:CA11::1234:5678:9ABC:DEF0
pluto      AAAA 2345:00D2:DA11::1234:5678:9ABC:DEF0
```

20 行目:

dns ホストの別名として、ntp を提供するための記述です (CNAME とは、*canonical name* (公式に認められた名前) の意味です)。

疑似ドメイン in-addr.arpa は、IP アドレスからホスト名に逆引きする際に利用します。アドレスのネットワーク部を逆順で表記したものを付加してドメイン名とします。たとえば 192.168 のアドレスに対する逆引き疑似ドメインは、168.192.in-addr.arpa になります。例 23.7 「逆引き」 (450 ページ) をご覧ください。

例 23.7 逆引き

```
1. $TTL 2D
2. 168.192.in-addr.arpa.    IN SOA dns.example.com. root.example.com. (
3. 2003072441              ; serial
4. 1D                      ; refresh
5. 2H                      ; retry
6. 1W                      ; expiry
7. 2D )                   ; minimum
8.
9.                          IN NS      dns.example.com.
10.
11. 1.5                    IN PTR      gate.example.com.
12. 100.3                  IN PTR      www.example.com.
13. 253.2                  IN PTR      cups.example.com.
```

1 行目:

\$TTL では、このゾーンの全てに適用される標準 TTL (生存時間) を指定します。

2 行目:

このゾーンファイルでは、192.168 に対する逆引きゾーンを設定しています。そのための擬似的なゾーン(ドメイン)は 168.192.in-addr.arpa で、このドメインは実際に存在するものではありません。そのため、全てのホスト名は完全修飾の(つまりドメイン名を付けた)形であり、かつ "." で終わるもの)で指定しなければなりません。それ以外の項目については example.com の例と同じです。

3 行目から 7 行目:

example.com での例と同じですので、そちらをお読みください。

9 行目:

この行では、このゾーンに対して管理責任を持つネームサーバを指定しています。ここで指定する名前は通常のドメインのものとは異なり、ドメイン指定を含んだ完全修飾の形で、"." で終わらなければなりません。

11 行目から 13 行目:

これらは、それぞれの IP アドレスに対するホスト名を指定するポインタレコードです。これらはいずれも "." で終わっておらず、行頭には IP アドレスの末尾部分しか書かれていないため、ゾーンが自動的に付加されます。実際の IP アドレスに変換する場合は、.in-addr.arpa を取り除いて "." ごとに逆から読んでください。

通常は、異なる BIND のバージョン間でも、問題なくゾーンの転送を行なうことができます。

23.7 ゾーンデータの動的な更新

動的な更新とは、マスターサーバ上のゾーンファイル内の項目を追加したり変更したり、削除したりする操作のことを指します。この仕組みは RFC 2136 で規定されています。動的な更新は、任意指定の `allow-update` や `update-policy` の設定を行なうことで、ゾーンごとに個別の設定を行なうことができます。動的に更新されるゾーンの場合、そのゾーンファイルは手作業で編集すべきではありません。

サーバに対して更新すべき項目を送信するには、`nsupdate` コマンドを使用します。このコマンドに対する正確な文法については、`nsupdate` のマニュアルページ (`man 8 nsupdate`) をお読みください。なおセキュリティ上の理由により、23.8 項「機密を保持する通信」(451 ページ) で書かれているとおり、このような更新には TSIG 鍵を使用すべきものです。

23.8 機密を保持する通信

機密を保持して通信するには、共有の機密鍵 (TSIG 鍵とも呼ばれます) をベースにした、トランザクション署名 (TSIG) を利用して行ないます。この章では、このような鍵をどのようにして生成するのかと、その使い方について述べています。

機密を保持する通信は、異なるサーバ間の通信やゾーンデータの動的な更新を行なう場合に必要となります。また、鍵をベースにしたアクセス制御を利用すると、単純に IP アドレスに依存した制御よりもより安全になります。

TSIG 鍵を生成するには、下記のコマンドを実行します (詳しくは `man dnssec-keygen` をお読みください):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

このコマンドを実行すると、下記のような 2 つのファイルが生成されます:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

鍵それ自身(ejIkuCyyGJwwuN3xAteKgg== のような文字列) が両方のファイル内に存在するはずです。通信時にこの鍵を使用するには、あらかじめ2つめのファイル(Khost1-host2.+157+34265.key)をリモートホスト側に安全な方法で(たとえば scp) 転送しておかなければなりません。リモートのサーバでは host1 と host2 の間で機密通信を行なうため、 /etc/named.conf ファイル内で 鍵を設定しなければなりません:

```
key host1-host2 {  
    algorithm hmac-md5;  
    secret "ejIkuCyyGJwwuN3xAteKgg==";  
};
```

警告: /etc/named.conf のファイルパーミッション

/etc/named.conf のパーミッション設定で、ファイルへのアクセスが正しく制限されていることを確認してください。このファイルの既定のパーミッションは 0640 で、所有者が root に、グループが named にそれぞれ設定されているはずですが。パーミッションを設定する以外にも、特別にアクセスを制限した別ファイルを作成し、そのファイルを /etc/named.conf から参照させる方法もあります。外部のファイルを参照させるには、下記のように設定します:

```
include "filename"
```

ここで filename には、鍵ファイルの絶対パスを指定します。

host1 のサーバに対して host2 (アドレス 10.1.2.3) と通信する際、鍵を使用するように設定するには、サーバ側の /etc/named.conf に下記のルールを追加しなければなりません:

```
server 10.1.2.3 {  
    keys { host1-host2. };  
};
```

/etc/named.conf 側の設定ファイルにも、似たような設定を行なわなければなりません。

通信の機密性を保持するには、IP アドレスやアドレス範囲に対して設定した任意の ACL (アクセス制御リスト。ファイルシステムのアクセス制御リストと混同しないでください) に TSIG 鍵を追加することもできます。たとえば下記のように設定します:

```
allow-update { key host1-host2. };
```

これらの詳細については、*BIND Administrator Reference Manual* 内の `update-policy` で説明しています。

23.9 DNS セキュリティ

DNSSEC または DNS セキュリティは、RFC 2535 で規定されています。DNSSEC 向けのツールについては、BIND マニュアル内で説明されています。

機密を保持すべきと考えるゾーンには、それに結びつけられた1つ以上のゾーン鍵が存在しなければなりません。このゾーン鍵は、ホスト鍵と同様に `dnssec-keygen` で作成することができます。これらの鍵を生成する際は、DSA 暗号化アルゴリズムが使用されます。関連するゾーンファイルから `$INCLUDE` ルールを利用して、公開鍵を登録してください。

`dnssec-makekeyset` コマンドを利用すると、生成した全ての鍵を1つのセットにまとめることができます。あとは機密を保持した方法で親ゾーンに転送してください。親ゾーン側では、その鍵セットに対し `dnssec-signkey` で署名を行なうことができます。このコマンドで生成されたファイルを利用すると、`dnssec-signzone` でゾーンに署名することができます。あとは生成されたファイルを `/etc/named.conf` 内の各ゾーンに登録してください。

23.10 さらなる情報

さらなる情報や説明については、`bind-doc` パッケージに含まれる *BIND Administrator Reference Manual* をお読みください。このパッケージは `/usr/share/doc/packages/bind/` 内にファイルをインストールします。また、マニュアルや BIND に含まれている マニュアルページから参照されている各 RFC についても、それぞれお読みください。また、`/usr/share/doc/packages/bind/README.SuSE` には、openSUSE での BIND について、最新情報を提供しています。

DHCP

Dynamic Host Configuration Protocol (DHCP) の目的は、それぞれのワークステーションで別々の設定を行なうのではなく、中央でネットワーク設定を管理して (サーバから) 設定を割り当てることにあります。DHCPを使用するよう設定したホストには、固定で設定するようなアドレスはありません。サーバからの指示に従って自動的に設定を行なうことができます。クライアント側で **NetworkManager** を使用している場合は、クライアント側で設定すべきことは何もありません。これは、1つのネットワークインターフェイスを利用して複数のネットワーク環境を切り替えるマシンで、便利な設定です。ただし、DHCPサーバの動作するマシンでは **NetworkManager** を動作させてはなりません。

DHCPサーバを設定するための1つの方法として、ネットワークカードのハードウェアアドレスを利用して各クライアントを識別する方法があります (多くの場合、ハードウェアアドレスは固定で割り当てられています)。この方法では、クライアントがサーバに接続するごとに同じ設定を割り当てます。もう1つの方法として、この目的用に確保したアドレス帯域を設定し、クライアントに対して動的な割り当てを行なうこともできます。後者の場合でも、DHCPサーバはクライアントから要求が届くごとに同じアドレスを割り当てようとします。それはたとえ期限を超過した場合でも同じです。ただし、この方法はアドレスの全体数よりもクライアント数のほうが少ない場合にのみ動作します。

DHCPはシステム管理者の手間を省きます。アドレスやネットワークの設定など、大きな設定変更が起こった場合でも、サーバ側の設定ファイルを編集するだけで作業が完了します。これは、多数のワークステーションの設定を変更しなければならないような場合よりはずっと手間を省くことができます。特に新しいマシンをネットワークに接続するような場合には、アドレス帯域

から IP アドレスを自動で割り当てることになるため、とても作業が簡単になります。DHCP サーバから取得した適切なネットワーク設定を受け取る仕組みであるため、異なるネットワークで使用するようなラップトップでは便利に使うことができます。

この章では、DHCP サーバをワークステーションと同じサブネット 192.168.2.0/24 で動作させる場合を想定しています。ゲートウェイは 192.168.2.1 で、DHCP サーバのアドレスは 192.168.2.254、割り当てるアドレス帯域は 192.168.2.10 から 192.168.2.20 までと、192.168.2.100 から 192.168.2.200 までを設定するものとします。

DHCP サーバは IP アドレスとネットマスクを提供するだけでなく、各クライアントが利用するホスト名やドメイン名、ゲートウェイやネームサーバの設定も提供します。またこれら以外にも、現在時刻を問い合わせるための時刻サーバや印刷サーバなど、様々な情報を中央から配布することができます。

24.1 YaST での DHCP サーバ設定

DHCP サーバをインストールするには、YaST を起動して **ソフトウェア>ソフトウェア管理** を選択します。続いて **フィルタ>パターン** を選択し、**DHCP** および **DNS** サーバを選択します。あとはインストール処理を完了し、依存関係のパッケージをインストールする確認に答えてください。

重要項目: LDAP 対応

YaST の DHCP モジュールの設定をローカル (DHCP サーバ自身) に保存することができるほか、LDAP サーバに保存することもできます。LDAP を利用して設定を保存したい場合は、DHCP サーバを設定する前に LDAP サーバを設定してください。

LDAP について、詳しくは第4章 **LDAP—A Directory Service** (↑セキュリティガイド) をお読みください。

YaST DHCP モジュール (`yast2-dhcp-server`) では、ローカルネットワーク用の DHCP サーバを設定することができます。また、このモジュールはユーザーモードと熟練者向け設定モードのいずれかで動作させることができます。

24.1.1 初期設定 (ウィザード)

最初にモジュールを起動したときにはウィザードが起動し、サーバを起動するにあたっていくつか決めておかなければならない基本設定について質問が行なわれます。この初期設定作業はサーバを機能させるためのごく基本的な設定だけです。高度な設定作業については、熟練者モードをご利用いただくことができます。

カードの選択

最初の手順では、YaSTはお使いのシステム内で利用可能なネットワークインターフェイスを確認し、一覧を表示します。一覧から **DHCP** サーバのサービスを提供したいインターフェイスを選択し、**選択**を押してください。選択のあとは、このインターフェイスに対してファイアウォールを開くため、**ファイアウォールで選択したインターフェイスを開く**を選択してから **次へ**を押してください。詳しくは 図24.1「DHCP サーバ: カードの選択」(457 ページ)をご覧ください。

図 24.1 DHCP サーバ: カードの選択

DHCP サーバウィザード (1/4): カードの選択

DHCP サーバのネットワークカード

選択済	インターフェイス名	デバイス名	IP
<input checked="" type="checkbox"/>	eth0	82540EM Gigabit Ethernet Controller	DHCP アドレス

☐ ファイアウォールで選択したインターフェイスを開く (F)

グローバル設定

まずは **DHCP** の設定を **LDAP** サーバ内に保存するかどうか、チェックボックスで選択します。またそれぞれの入力フィールドに対して、**DHCP** サーバが管理すべき全クライアント用のネットワーク設定を指定してください。ドメイン名やタイムサーバ、プライマリまたはセカンダリのネームサーバ、印刷サーバや **WINS** サーバ (**Windows** と **Linux** のクライアントが

混在する環境の場合)、ゲートウェイアドレスや貸与時間をそれぞれ設定します。詳しくは 図24.2 「DHCP サーバ: グローバル設定」 (458 ページ) をご覧ください。

図 24.2 DHCP サーバ: グローバル設定

DHCP サーバウィザード (2/4): グローバル設定

☐ LDAP サポート (L)

DHCP サーバ名 (N) (オプション)

ドメイン名 (D)
example.org

NTP 時刻サーバ (T)
192.168.1.20

プライマリネームサーバ IP (P)
192.168.1.254

プリントサーバ (P)

セカンダリネームサーバ IP (S)

WINS サーバ (W)

デフォルトゲートウェイ (ルータ) (G)
192.168.1.200

既定の貸与時間 (L)
10

単位 (U)
分

ヘルプ (H) 戻る (B) 中止 (R) 次へ (N)

ダイナミック DHCP

この段階では、どのようにして動的な IP アドレスの配布を行なうのかについて設定を行いません。まずは DHCP クライアントに対してサーバから配布する IP アドレスの範囲を設定します。これらのアドレスは同じネットワーク内を示すものでなければなりません。また、クライアントが期限の延長を申請することなく IP アドレスを使い続けることのできる、既定の貸与時間も設定します。また、最大の貸与時間も設定することができます。これはサーバが特定のクライアントに対する IP アドレスを予約しておく時間の意味です。詳しくは 図24.3 「DHCP サーバ: ダイナミック DHCP」 (459 ページ) をご覧ください。

図 24.3 DHCP サーバ: ダイナミック DHCP

The screenshot shows the 'DHCP サーバウィザード (3/4): ダイナミック DHCP' window. It is divided into three main sections: 'サブネット情報' (Subnet Information), 'IP アドレス範囲' (IP Address Range), and '貸与時間' (Lease Time). In the 'サブネット情報' section, '現在のネットワーク (N)' is 10.0.2.0, '現在のネットマスク (M)' is 255.255.255.0, and 'ネットマスクビット (T)' is 24. The 'IP アドレス範囲' section has '最初の IP アドレス (F)' as 10.0.2.1 and '最後の IP アドレス (L)' as 10.0.2.254. The '貸与時間' section shows '既定 (D)' as 4, '単位 (U)' as 時間 (Time), '最大値 (X)' as 2, and '単位 (T)' as 日 (Days). There is a checkbox for '動的 BOOTP の許可 (B)' which is currently unchecked. At the bottom, there are buttons for 'ヘルプ (H)', '戻る (B)', '中止 (R)', and '次へ (N)'. A 'DNS サーバと同期 (S)...' button is also present.

設定の完了と開始モードの設定

3段階目の設定ウィザードを完了すると、**DHCP**サーバをどのようにして起動する のかを設定する最後のダイアログが表示されます。ここでは **DHCP**サーバをシステム 起動時に自動で起動するか、もしくは必要に応じて手動で起動するか (たとえば テスト用に起動する場合など) を選択することができます。設定を完了したら、完了を押してください。詳しくは 図24.4「**DHCP** サーバ: 起動」 (459 ページ) をご覧ください。

図 24.4 DHCP サーバ: 起動

The screenshot shows the 'DHCP サーバウィザード (4/4): 起動' window. It is titled 'サービスの開始' (Start Service). There are two radio buttons: 'システム起動時 (B)' (At system startup) and '手動 (M)' (Manually). The '手動 (M)' option is selected. Below the radio buttons is a button labeled 'DHCP サーバ熟練者設定 (E)...'. At the bottom, there are buttons for 'ヘルプ (H)', '戻る (B)', '中止 (R)', and '完了 (F)'. The '完了 (F)' button is highlighted in blue.

24.2 DHCP ソフトウェアパッケージ

openSUSE では、DHCP サーバと DHCP クライアントの両方が提供されています。DHCP サーバとしては `dhcpcd` (Internet Systems Consortium 提供) が公開されていますが、DHCP クライアントとしては 2 種類のクライアントが公開されています: `dhcp-client` (上記と同様に ISC 提供のもの) と `dhcpcd` パッケージがあります。

openSUSE では既定で `dhcpcd` がインストールされています。このプログラムはとても扱いやすく、それぞれのシステム起動時に DHCP サーバのチェックを行なうために自動で起動するようになっています。`dhcpcd` はその処理を行なうにあたって設定ファイルを必要とせず、多くの標準的なセットアップ環境でうまく動作するようになっています。より複雑な環境の場合は ISC の `dhcp-client` をお使いいただき、設定ファイル `/etc/dhclient.conf` で細かい制御を行なってください。

24.3 DHCP サーバ `dhcpcd`

DHCP システムの中核は、dynamic host configuration protocol (DHCP) のデーモンです。このサーバは設定ファイル `/etc/dhcpcd.conf` での設定に基づき、アドレスを貸与し、それらがどのように使用されているかを監視します。このファイルのパラメータや値を変更することで、システムの管理者は様々な方法でプログラムの動きを制御することができます。まずは例24.1「設定ファイル `/etc/dhcpcd.conf`」(461 ページ)にある基本設定例 `/etc/dhcpcd.conf` をお読みください。

例 24.1 設定ファイル `/etc/dhcpd.conf`

```
default-lease-time 600;           # 10 minutes
max-lease-time 7200;             # 2 hours

option domain-name "example.com";
option domain-name-servers 192.168.1.116;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
option subnet-mask 255.255.255.0;

subnet 192.168.2.0 netmask 255.255.255.0
{
    range 192.168.2.10 192.168.2.20;
    range 192.168.2.100 192.168.2.200;
}
```

このようなシンプルな設定ファイルだけで、DHCPサーバに対してネットワーク内のアドレス割り当てを実施させることができます。なお、各行の末尾にはセミコロンがきちんと付けられていることを確認してください。そうでないと `dhcpd` を起動することができません。

このサンプルファイルは3つのセクションから構成されています。1つめのセクションには、クライアントからアドレスを要求されたときに、既定でどれだけの期間更新無しで貸し出すのか (`default-lease-time`) を設定しています。このセクションでは、DHCPサーバがIPアドレスを貸与したあと、更新無しでどれだけの期間アドレスを保持しておくか (`max-lease-time`) も設定しています。

2つめのセクションには、グローバルな範囲でいくつかの基本的なネットワークパラメータを指定しています:

- `option domain-name` の行では、お使いのネットワークにおける既定のドメインを設定しています。
- `option domain-name-servers` を設定すると、ホスト名からIPアドレス、またはその逆の変換のため、最大で3つまでのDNSサーバを指定することができます。理想的にはお使いのマシン上で動作しているネームサーバや、お使いのネットワーク内のどこかにあるネームサーバを設定します。なお、これらのネームサーバでは、それぞれ動的に割り当てられるアドレスに対して、ホスト名を定義しておくべきものです。ネームサーバの設定方法について、詳しくは第23章 ドメインネームシステム (427 ページ) をお読みください。

- `option broadcast-address` の行では、クライアントが使用するブロードキャストアドレスを指定しています。
- `option routers` では、ローカルネットワークだけでは到達できないホストとの通信を行なう際、データパケットを送信する宛先を指定します(発信元と送信先のアドレス、およびサブネットマスクから判断します)。多くの場合、特に小さなネットワークの場合は、ルータとインターネットゲートウェイは同じものです。
- `option subnet-mask` では、クライアントに割り当てるサブネットマスクを指定します。

設定ファイルの最後の部分では、サブネットマスクを含むネットワークを定義しています。また括弧内では、**DHCP** デーモンがクライアントに対して割り当てるべきアドレス範囲を指定しています。例24.1「設定ファイル `/etc/dhcpd.conf`」(461 ページ)の例では、192.168.2.10 から 192.168.2.20 までの範囲と、192.168.2.100 から 192.168.2.200 までの範囲でそれぞれ割り当てを行ないます。

これらのうちいくつかを環境に応じて変更したら、あとは `rcdhcpd start` コマンドで **DHCP** サーバを起動することができます。これですぐにサービスを利用できるようになります。また、設定ファイルについて大まかな文法チェックを行ないたい場合は、`rcdhcpd check-syntax` コマンドを入力してください。設定について予期しない問題に直面した場合(サーバがエラー終了するか、起動時に `done` が表示されない場合)は、メインのシステムログである `/var/log/messages` を利用して何が間違っているのかを確認することができるほか、コンソール 10 でエラー出力を確認することもできます(`Ctrl + Alt + F10` を押してください)。

既定の **openSUSE** システムでは、**DHCP** はセキュリティ上の理由から、**chroot** 環境で起動されます。設定ファイルはデーモンから読み取ることができるように、**chroot** の環境下にコピーしなければなりません。通常は `rcdhcpd start` コマンドで自動的にファイルをコピーするため、これらのことについて心配する必要はありません。

24.3.1 固定 IP アドレスのクライアント

DHCP では、特定のクライアントに対して固定のアドレスを事前に定義し、割り当てることもできます。明示的に指定したアドレスは、帯域からの動的な割り当てよりも常に優先して動作します。また固定のアドレスは動的なアドレスと異なり、クライアントに配布するアドレスよりも利用可能なアドレスが少ない場合に発生するような、有効期限の問題ありません。

固定のアドレスを各クライアントに割り当てるにあたり、DHCP ではハードウェア アドレス (たとえば 00:30:6E:08:EC:80) 利用して、ネットワーク デバイスを識別します (世界中で唯一の番号が割り振られるもので、6 バイトから構成される数値コードです)。具体的には 例24.2 「設定ファイルへの追記」 (463 ページ) に示すような設定を 例24.1 「設定ファイル /etc/dhcpd.conf」 (461 ページ) に追加することで、特定のクライアントに対して同じデータを割り当てるようになります。

例 24.2 設定ファイルへの追記

```
host jupiter {  
  hardware ethernet 00:30:6E:08:EC:80;  
  fixed-address 192.168.2.100;  
}
```

各クライアントの名前 (host ホスト名の部分、ここでは jupiter) が最初の行に記述され、2 行目に MAC アドレスが記述されています。Linux ホストでは、MAC アドレスは `ip link show` コマンドで表示させることができます。このコマンドの後ろにネットワークデバイス (たとえば eth0) を指定してください。出力には下記のような行があるはずです。

```
link/ether 00:30:6E:08:EC:80
```

上記の例では、00:30:6E:08:EC:80 という MAC アドレスが設定されているクライアントに対して、192.168.2.100 というアドレスと jupiter というホスト名を自動的に割り当てる動作になります。ほとんどの場合、ハードウェアの種類 (hardware) には ethernet を指定しますが、IBM システムなどでは token-ring である場合もあります。DHCP ではどちらにも対応しています。

24.3.2 openSUSE バージョン

セキュリティをよりよくするため、ISC 提供の DHCP サーバの openSUSE 版では、Ari Edelkind 氏が開発した non-root/chroot パッチを適用した状態で公開しています。これにより、dhcpd をユーザ ID nobody で実行し、chroot 環境(/var/lib/dhcp)で動作させることができるようになっています。また、この機能を実現するため、設定ファイル dhcpd.conf は /var/lib/dhcp/etc に配置しなければなりません。起動時の起動スクリプトでこのディレクトリへの設定ファイルコピーを行なっています。

この機能に関するサーバの動作を制御するには、/etc/sysconfig/dhcpd ファイルを編集してください。dhcpd を chroot 環境で動作させないようにするには、/etc/sysconfig/dhcpd ファイル内の DHCPD_RUN_CHROOTED 変数に、「no」を設定してください。

dhcpd に対し、chroot 環境下からでもホスト名を解決できるようにするには、いくつかの設定ファイルについてもコピーしておかなければなりません:

- /etc/localtime
- /etc/host.conf
- /etc/hosts
- /etc/resolv.conf

これらのファイルは、起動時に初期化スクリプトが /var/lib/dhcp/etc/ にコピーします。これらのコピーは /etc/ppp/ip-up のようなスクリプトから動的に更新される場合に必要となります。しかしながら、この設定ファイルは IP アドレスだけ(ホスト名ではなく)を指定する仕組みであるため、特に心配する必要はありません。

また、お使いの設定ファイルから外部のファイルを参照するように設定している場合は、それらを chroot 環境にコピーする必要があります。このような外部ファイルについては、/etc/sysconfig/dhcpd ファイル内の DHCPD_CONF_INCLUDE_FILES 変数に設定を行なってください。また、syslog-ng が再起動しても DHCP のログ機能が正しく動作するようにするため、/etc/sysconfig/syslog 内に SYSLOGD_ADDITIONAL_SOCKET_DHCP という設定を用意しています。

24.4 さらなる情報

DHCP に関する詳しい情報は、*Internet Systems Consortium* の Web サイト (<http://www.isc.org/products/DHCP/>) (英語) をお読みください。
dhcpcd, dhcpcd.conf, dhcpcd.leases, dhcp-options についての各マニュアルページも提供されています。

NTP を利用した時刻同期

NTP (Network Time Protocol; ネットワーク時刻プロトコル) は、ネットワークを介して時刻同期を行なうためのプロトコルです。NTP サービスでは最初に、信頼の置ける時刻発信源であるサーバから時刻を取得します。その後、マシンは自分自身がネットワーク内の他のコンピュータに対して、時刻を発信できるように動作します。つまり NTP サービスには、絶対時刻の管理とネットワーク内の全マシンに対するシステム時刻の同期、という 2 つの目的が存在することになります。

正確なシステム時刻を保つことは、様々な状況で重要な要件となります。内蔵のハードウェア (BIOS) 時計では、データベースやクラスタなどのアプリケーションの要件に足りていない場合がしばしばあります。かといってシステム時刻を手作業で修正すると、たとえば時刻の巻き戻りによって、重要なアプリケーションに障害が発生してしまったりするなど、場合によっては深刻な問題になってしまうことがあります。ネットワーク内では一般に、全てのマシンのシステム時刻を同期しておく必要がありますが、手作業による調整は悪いアプローチと言わざるを得ません。NTP を利用すると、これらの問題を解決することができます。ネットワーク内にある信頼している時刻発信源の情報を頼りに、システム時刻を継続的に調整します。時刻の発信源としては、電波時計のようなローカル参照時計を利用することもできます。

25.1 YaST を利用した NTP クライアントの設定

ntp パッケージに含まれる NTP デーモン (ntpd) は、自分自身の時計を時刻発信源として使用するよう事前設定されています。しかしながら (BIOS の) 時計は、より精度の高い時刻発信源が存在しない場合の代替手段として使用されるべきものです。そのため、YaST を利用して NTP クライアントの設定を行なってください。

25.1.1 基本設定

YaST での NTP クライアント設定 (ネットワークサービス > NTP 設定) は複数のタブから構成されています。一般設定のタブで ntpd の起動モードを設定します。

図 25.1 より高度な NTP クライアント設定: 一般設定

The screenshot shows the '高度な NTP 設定' (Advanced NTP Settings) window in YaST, with the '一般設定' (General) tab selected. The window is titled '高度な NTP 設定' and has two tabs: '一般設定' and 'セキュリティ設定'. The main content area is titled 'NTP デーモンを起動する' (Start NTP daemon). It contains three radio buttons: '手動でのみ起動 (M)' (Start manually only), 'デーモンを使用せずに同期する (S)' (Synchronize without daemon), and '今すぐ開始し、システム起動時に開始するよう設定 (B)' (Start now and configure to start at system boot). The third option is selected. Below the radio buttons is a section for '実行時設定ポリシー (R) カスタムポリシー (C)' (Runtime setting policy) with a dropdown menu set to '自動' (Automatic). Below that is a '同期間隔 [分] (I)' (Synchronization interval [min]) field set to '5'. At the bottom is a table with two columns: '同期種類' (Synchronization type) and 'アドレス' (Address). The table has one row with '自分自身の時計 (ローカル)' (Own clock (local)) in the first column and 'ja.pool.ntp.org' in the second column. Below the table are buttons for '追加 (A)' (Add), '編集 (I)' (Edit), and '削除 (T)' (Delete). At the bottom right is a button for 'ログの表示 (L)...' (Show log...). At the very bottom are buttons for 'ヘルプ (H)' (Help), 'キャンセル (C)' (Cancel), and 'OK (O)' (OK).

手動でのみ起動

全ての項目を手作業で設定したい場合は、*手動でのみ起動*を選択します。

デーモンを使用せずに同期する

ラップトップやその他のマシンで自動的にサスペンドするような環境の場合、デーモンを使用せずに同期するを選択します。このモードではYaSTは同期を行なうのにntpdを開始することは行ないません。その代わり、YaSTはcrontabの項目(/etc/cron.d/novell.ntp-synchronize)を作成し、同期間隔(分)の項目で指定した間隔で時刻サーバに時刻を問い合わせるようにします。cronについて、詳しくは18.1.2項「cronパッケージ」(302ページ)をお読みください。

今すぐ開始し、システム起動時に開始するよう設定

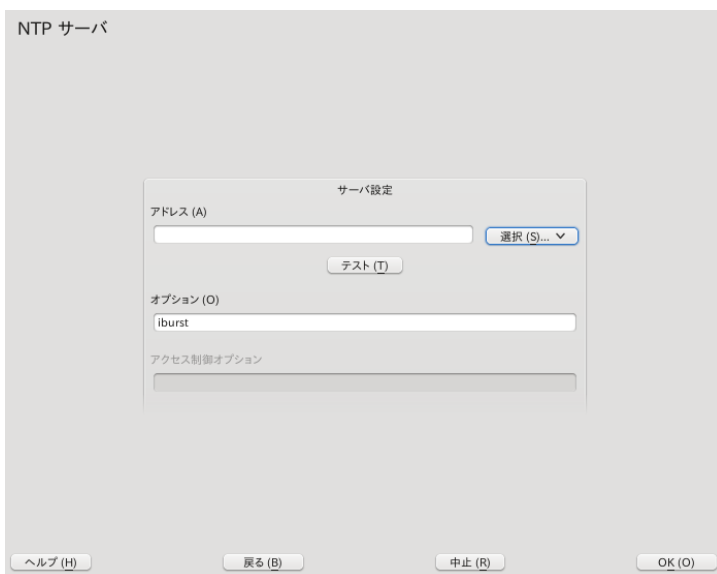
システムの起動時にntpdを自動的に開始するように設定するには、今すぐ開始し、システム起動時に開始するよう設定を選択します。なお、0.opensuse.pool.ntp.org, 1.opensuse.pool.ntp.org, 2.opensuse.pool.ntp.org, 3.opensuse.pool.ntp.orgのいずれかが事前を選択されています。

25.1.2 基本的な設定の変更

時刻サーバなど、クライアントに設定する時刻発信源は、一般設定タブで表示される画面の下半分で設定を行ないます。必要に応じて、それぞれ追加, 編集, 削除を行なってください。ログの表示を押すと、お使いのクライアントにおけるログファイルを閲覧することができます。

追加ボタンを押すと、時刻の発信源を追加することができます。まずは表示されるダイアログから、時刻同期の種類を選択します。下記のうちのいずれかを選択することができます:

図 25.2 YaST: NTP サーバ



サーバ

選択のプルダウンリスト (図25.2「YaST: NTP サーバ」 (470 ページ) をご覧ください) から、お使いのローカルネットワークにあるタイムサーバを利用して時刻同期を設定することができる (ローカル *NTP* サーバ) ほか、ご利用のタイムゾーンに合わせてインターネット上で公開されているタイムサーバを利用して時刻同期を設定することができます (公開 *NTP* サーバ)。ローカルの時刻サーバの場合は、検索ボタンを押して SLP の問い合わせを送信し、利用可能なタイムサーバが存在しないかどうかを確認することができます。検索結果から適切なタイムサーバを選択し、**OK** ボタンを押してダイアログを完了してください。公開タイムサーバの場合は、ご利用の国 (タイムゾーン) を選んでから 公開 *NTP* サーバの一覧からサーバを選択し、**OK** ボタンを押してダイアログを完了してください。メインダイアログに戻ったあと、選択したサーバとの通信が確立するかどうかを テスト ボタンで確認することができます。

次のダイアログで *NTP* サーバを選択することができます。システムの起動時にサーバとクライアントの間で時刻情報を同期させたい場合は、このサーバを初期同期に使用するを選択してください。オプションでは、`ntpd` に対するさらなるオプションを設定することができます。

アクセス制御オプションを利用すると、お使いのコンピュータ上で動作しているデーモンに対して、遠隔のコンピュータから通信があった場合の動作を制限することができます。この項目は、セキュリティ設定タブ内 (図25.3「より高度な NTP クライアント設定: セキュリティ設定」(472 ページ) をご覧ください) にある *NTP* サービスを設定したサーバに制限するにチェックを入れた場合にのみ設定できます。このオプションは、`/etc/ntp.conf` ファイル内の `restrict` に該当する設定です。たとえば `nomodify notrap noquery` を設定すると、そのサーバからお使いのコンピュータに対して *NTP* 設定の修正を禁止し、トラップ機能 (遠隔イベントログ機能) を無効化するように設定することになります。このような制限は、接続しようとしているサーバが外部の企業や団体で管理されているもの (たとえばインターネット上のサーバ) である場合に推奨される設定です。

詳しくは `/usr/share/doc/packages/ntp-doc` 内にある文書 (ntp-doc パッケージ内に含まれます) をお読みください。

ピア

ピアとは対称型の同期を行なうようにするための設定で、互いに両方がサーバ兼クライアントとして動作することになります。同じネットワーク内でサーバの代わりにピアを利用する場合は、そのシステムのアドレスを入力してください。残りのダイアログはサーバの場合と同じです。

ラジオクロック

お使いのシステムで、ラジオクロックを時刻同期に利用したい場合の設定です。それぞれクロック種類とユニット番号、デバイス名とその他のオプションを入力してください。ドライバに対して細かい調整を加えるには、ドライバの調整を押してください。ローカルに接続されたラジオクロックに関する詳しい操作方法は、`/usr/share/doc/packages/ntp-doc/refclock.html` をお読みください。

送信ブロードキャスト

時刻情報とその問い合わせを、ネットワーク内のブロードキャスト送信で行なうことができます。この設定では、このようなブロードキャストを送信する際の宛先を指定します。ラジオコントロールの時計など、信頼のおける時刻発信源をお持ちでない場合は、ブロードキャストを有効に設定しないでください。

受信ブロードキャスト

お使いのクライアントでブロードキャストによる時刻情報受信を行ないたい場合は、それらのパケットをどこから受信するのかを設定します。

図 25.3 より高度な NTP クライアント設定: セキュリティ設定



セキュリティ設定のタブ(図25.3「より高度な NTP クライアント設定: セキュリティ設定」(472 ページ)をご覧ください)では、`ntpd` を `chroot` 環境(文書によっては `chroot jail` と呼ばれる場合もあります)下で起動するかどうかを設定することができます。既定では *NTP* デーモンを *chroot* 環境下で実行するが選択されていて、有効になっています。この設定は `ntpd` に対してセキュリティ欠陥を狙う攻撃に対する防御を高めるもので、たとえ攻撃が成功してもシステム全体にまで被害をもたらさないようにするための仕組みです。

また、*NTP* サービスを設定したサーバに制限するを選択すると、遠隔のコンピュータから *NTP* の設定を閲覧したり修正したり、リモートイベントログのトラップ機能を使用したりする行為を禁止し、お使いのシステムに対するセキュリティを高めることができます。この選択は全ての遠隔コンピュータに対して適用されます。ただし時刻の発信源として個別にコンピュータを設定し、一般設定タブでアクセス制御オプションを設定した場合は、そちらの設定が優先されます。それ以外の遠隔のコンピュータからは、ローカル時刻の問い合わせだけが許可されます。

さらに、SuSEfirewall2が動作している(既定で起動されます)場合は、ファイアウォールでポートを開くを選択することもできます。ポートを閉じたままにしていると、時刻サーバに対する接続を行なうことができなくなります。

25.2 ネットワーク内にある NTP の手動設定

ネットワーク内の時刻サーバを使用するための最も簡単な方法は、サーバのパラメータを設定することです。たとえば時刻サーバに対して、ntp.example.comという名前でもネットワーク内からアクセスできる場合、下記のようにその名前を /etc/ntp.conf に記入するだけです:

```
server ntp.example.com
```

複数の時刻サーバを追加したい場合は1行1サーバで記述し、各行の行頭にキーワード `server` を付与してください。 `rcntp start` のコマンドで `ntpd` を起動すると、おおよそ1時間程度で時刻が安定するようになり、ローカルの時刻を調整するためのドリフトファイルと呼ばれるものが作成されます。ドリフトファイルは、コンピュータの起動後から計算されるハードウェア時刻の変動パラメータで、その調整データは次回以降にすぐに反映されるようになります。これにより、システム時刻の安定性を高めることができます。

NTP の仕組みをクライアント側で使用する場合には、2つの方法が考えられます: 1つめは既知のサーバに対して定期的な間隔で時刻を問い合わせる方法です。多くのクライアントが存在する場合、このアプローチではサーバの負荷が高くなってしまいます。2つめはネットワーク内で時刻サーバがNTPブロードキャストを送信するのを待つ方法です。このアプローチの場合はサーバの品質に関する情報が不明で、万が一サーバが誤った情報を配布してしまうと、致命的な問題になってしまいます。

時刻をブロードキャスト経由で取得する場合は、サーバ名を指定する必要はありません。設定ファイル /etc/ntp.conf 内に `broadcastclient` と書いた行を追加するだけです。既知の時刻サーバを1つ以上設定する場合は、`servers` で始まるサーバ指定を行なってください。

25.3 システム稼働時の動的な時刻同期

ネットワーク接続無しでシステムが稼働している場合、`ntpd` は起動することができるものの、設定ファイル内に設定されている時刻サーバに対して DNS の名前解決をすることができなくなります。これは暗号化された無線 LAN を設定した **Network Manager** 環境でも発生する問題です。

稼働中に `ntpd` に対して DNS のホスト名解決を行なわせたい場合は、`dynamic` オプションを設定しなければなりません。この場合、起動後にネットワークの接続が確立すると、`ntpd` は名前解決をやり直してタイムサーバにアクセスし、時刻を取得できるようになります。

手作業で `/etc/ntp.conf` ファイルを編集し、`server` の項目 (1 つまたは複数) に対して `dynamic` 句を追加してください:

```
server ntp.example.com dynamic
```

もしくは、**YaST** を利用して下記のように行なってください:

- 1 **YaST** を起動して **ネットワークサービス > NTP 設定** を選択します。
- 2 設定したいサーバを選択し、**編集** ボタンを押します。
- 3 オプションの項目を選択し、`dynamic` という文字列を追加します。既に何らかのオプションが書かれていた場合は、半角スペースで区切って入力してください。
- 4 編集ダイアログを閉じるには **Ok** ボタンを押します。ここまでの手順を、設定したいサーバ全てに対して繰り返します。
- 5 最後に **Ok** ボタンを押し、設定を保存してください。

25.4 ローカル参照時計の設定

ソフトウェアパッケージ `ntp` には、ローカル参照時計に接続するためのドライバが用意されています。対応している時計の一覧は、`ntp-doc` パッケージ内の `/usr/share/doc/packages/ntp-doc/refclock.html` ファイルに書かれています。各ドライバには番号が振られていて、`ntp` の設定ではその番

号に基づく疑似的な IP アドレスを指定します。そのため、`/etc/ntp.conf` の設定ファイルからは、ネットワーク内に それらの時計が存在するかのよう な書式になります。疑似 IP アドレスは下記のような形式です: `127.127.t.u` ここで、`t` には時計の種類と使用するドライバを、`u` には使用するインターフェイスをそれぞれ数字で 指定します。

通常、それぞれのドライバにはより細かい設定を行なうための特別なパラメータが用意されています。個別のドライバに関する詳しい情報は、`/usr/share/doc/packages/ntp-doc/drivers/driverNN.html` (ここで、`NN` にはドライバの番号が入ります) を お読みください。たとえば「タイプ 8」の時計 (シリアルポートに 接続されたラジオ時計) であれば、時計をより細かく指定するための追加モードを 指定します。たとえば **Conrad DCF77** レシーバモジュールをお使いの場合は、`mode 5` を指定します。また、この時計を優先的に参照させるには、`prefer` キーワードを指定します。**Conrad DCF77** モジュール を利用するための `server` 設定は、下記のようになります:

```
server 127.127.8.0 mode 5 prefer
```

他の時計でも同じような指定を行ないます。`ntp-doc` パッケージをインストールすると、`/usr/share/doc/packages/ntp-doc` ディレクトリ内に文書が用意されますので、そちらをお読みください。`/usr/share/doc/packages/ntp-doc/refclock.html` ファイルには、ドライバのパラメータ説明が書かれているドライバページのリンク集があります。

NFS でのファイル共有

ネットワークを介したファイルシステムの公開や共有は、企業内のネットワークなどでは一般的に行なわれているものです。NFS は電話帳のような機能を提供する NIS プロトコルとともにうまく動作することが証明されています。LDAP と Kerberos を利用した、より機密を保持するプロトコルをお使いになる場合は、NFSv4 をご覧ください。

NIS と NFS を同時に稼働させると、ユーザに対してネットワークを透過的に見せることができるようになります。NFS ではネットワークを介して任意のファイルシステムを公開することができますので、適切な設定を行なうことにより、利用している端末にかかわらず同じ環境で作業を行なうことができます。

NIS と同様に NFS もクライアント／サーバのシステムです。ただし NIS とは異なり、マシンはネットワークを介してファイルシステムを公開(エクスポート)すると同時に、他のホスト上のファイルシステムをマウントする(インポート)することができます。

重要項目: DNS の必要性

原理上、全てのエクスポートは IP アドレスだけを利用して行なわれますが、タイムアウトを防ぐ目的から、DNS システムを動作させる必要があります。また、`mountd` では逆引きの参照を行なうため、ログ用途でも必要となります。

26.1 必要なソフトウェアのインストール

お使いのホストを NFS クライアントとして設定するには、追加のソフトウェアをインストールする必要はありません。NFS クライアントを設定するために必要な全てのソフトウェアは、既定でインストールされます。

NFS サーバのソフトウェアは、既定のインストールには含まれていません。NFS サーバソフトウェアをインストールするには、YaST を起動して **ソフトウェア>ソフトウェア管理** を選択してください。ソフトウェア管理のモジュールが起動したら、**フィルタ>パターン** を選択してから **ファイルサーバ** を選択するか、もしくは **検索オプション** を利用して NFS サーバを検索し、インストールしてください。最後にインストール処理を完了するため、インストールされるパッケージ内容を確認します。

26.2 YaST を利用したファイルシステムのインポート

権限を与えられたユーザであれば、YaST NFS クライアントモジュールを利用して NFS サーバからディレクトリをマウントすることができます。これは YaST のモジュールである *NFS* クライアントを利用して行なうことができます。モジュールを起動してから **追加** を押し、NFS サーバのホスト名とインポートするディレクトリ、そしてどのディレクトリにつなげるかをそれぞれ指定してください。これらの変更は、最初のダイアログで完了ボタンを押した時に実施されます。

また、*NFS* 設定のタブでは、リモートのコンピュータが提供するサービスに対してアクセスを許可するため、**ファイアウォールでポートを開く** を選択することができます。ファイアウォールの状態はチェックボックスの下に表示されます。NFSv4 を利用する場合は、*NFSv4* を有効にするのチェックが入っているかどうかを確認し、*NFSv4* ドメイン名に NFS サーバと同じ設定を入力してください。既定ではドメイン名は `localdomain` になっています。

最後に設定を保存するため、**OK** を押します。図26.1「YaST を利用した NFS クライアント設定」(479 ページ) をご覧ください。

設定ファイルは `/etc/fstab` に書き込まれ、指定したファイルシステムがマウントされます。後から YaST 設定モジュールを起動したときには、このファイルから 既存の設定を読み込みます。

図 26.1 YaST を利用した NFS クライアント設定



26.3 手作業でのファイルシステムのインポート

手作業でファイルシステムをマウントするには、事前に `RPC portmapper` を起動しておく必要があります。`RPC portmapper` は `root` から `rcrpcbind start` と入力することで起動することができます。起動が完了したら、リモート側のエクスポート済みファイルシステムは、下記のようにハードディスクをマウントするのに似たコマンドラインでマウントできるようになります:

`mount` ホスト:サーバ側のパス ローカルのパス

たとえば `nfs.example.com` マシン上のユーザディレクトリをインポートしたい場合は、下記のようなコマンドを入力します:

```
mount nfs.example.com:/home /home
```

26.3.1 automount サービスの利用

autofs デーモンは、リモートのファイルシステムを自動でマウントするのに使用することができます。これを行なうには、お使いの `/etc/auto.master` ファイルに対し、下記のような設定を行ないます:

```
/nfsmounts /etc/auto.nfs
```

上記の設定を行ない、`auto.nfs` ファイルでの設定を行なうと、`/nfsmounts` ディレクトリは全 NFS クライアント側マウントのルートとして動作するようになります。`auto.nfs` というファイル名にしたのは単に利便性を考慮しただけのもので、任意の名前を設定することができます。上記で指定したファイル (上記の例では `auto.nfs`) には下記のような項目を記入します:

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

あとは `root` から `rcautofs start` を実行すると、設定を反映させることができます。この例では `/nfsmounts/localdata` ディレクトリに `server1` の `/data` ディレクトリを NFS でマウントし、`server2` の `/nfsmounts/nfs4mount` ディレクトリを NFSv4 でマウントします。

autofs が動作している状態で `/etc/auto.master` を編集した場合は、`automounter` に変更を反映させるため、`rcautofs restart` を実行して再起動を行なわなければなりません。

26.3.2 /etc/fstab の手作業での編集

一般的な NFSv3 マウントの場合、`/etc/fstab` には下記のように設定します:

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

NFSv4 マウントについても `/etc/fstab` に設定することができます。NFSv4 のマウントの場合は、3 つめの列を `nfs` ではなく、`nfs4` と設定し、リモート側のファイルシステムの指定である 1 列目の `nfs.example.com:` の後が `/` になっていることを確認してください。一般的な NFSv4 マウントの場合、`/etc/fstab` は下記のようになります:

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

noauto オプションは、システムの起動時に自動でファイルシステムをマウントしないようにする設定です。上記のようなファイルシステムを手作業でマウントする場合は、マウント時に下記のようなコマンドでマウントポイントだけを指定することで、マウントを行なうことができます:

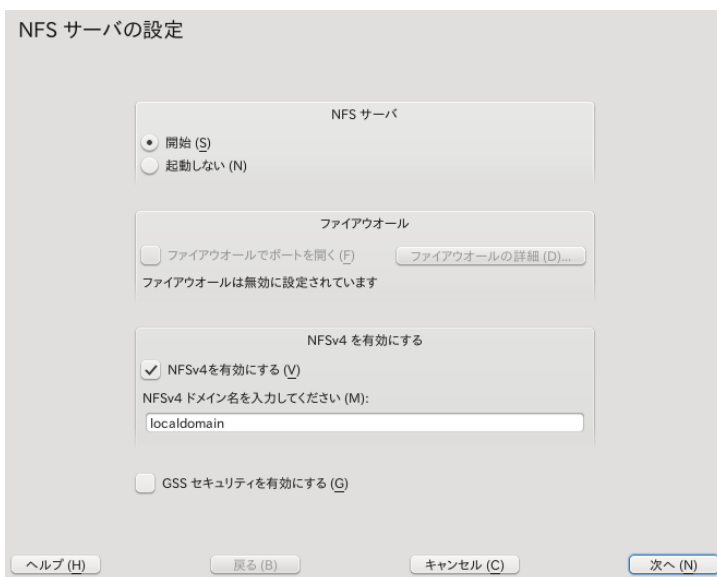
```
mount /local/path
```

noauto オプションを指定しない場合は、システムの起動時にこれらのファイルシステムが処理され、マウントされるようになります。

26.4 YaST を利用したファイルシステムのエクスポート

YaST では、お使いのホストをネットワーク内の NFS サーバとして動作させることができます。NFS サーバとは、アクセス権を設定したホストに対してディレクトリやファイルを共有 (エクスポート) するということです。これにより、アプリケーションを各ホストのローカルにインストールすることなく、グループ内の全メンバーに提供することができるようになります。このようなサーバをインストールするには、YaST を起動して ネットワーク サービス > NFS サーバを選択します。すると、図26.2「NFS サーバ設定ツール」(482 ページ) のようなダイアログが表示されます。

図 26.2 NFS サーバ設定ツール

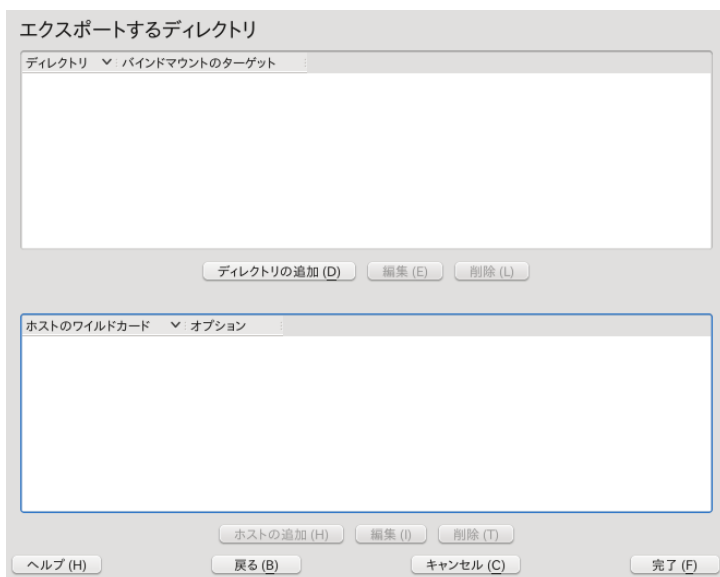


ダイアログが表示されたら、まずは**開始**を選択してから、**NFSv4** ドメイン名に入力を行います。

サーバに対して機密を保持する接続を行ないたい場合は、**GSS セキュリティ**を有効にするを選択します。これを行なうには、お使いのドメインにあらかじめ **Kerberos** をインストールし、サーバとクライアントの両方で **Kerberos** の設定を行なっておく必要があります。設定を完了したら **次へ** を押します。

次に上半分のテキストフィールドでは、エクスポートするディレクトリを入力します。下半分のテキストフィールドでは、それらへのアクセスを許可するホストを指定します。詳しくは図26.3「YaST を利用した NFS サーバ設定」(483 ページ)をご覧ください。この図では以前のダイアログで **NFSv4** を選択した場合のもので、**バインドマウントのターゲット** が右側に表示されています。詳しくは左側に表示されるヘルプをお読みください。ダイアログの下半分では、各ホストに対して下記の 4 種類のオプションを設定することができます: **single host**, **netgroups**, **wildcards**, **IP networks**。これらのオプションについて、詳しくは **exports** のマニュアルページをお読みください。設定を完了するには、**完了** ボタンを押します。

図 26.3 YaST を利用した NFS サーバ設定



重要項目: 自動ファイアウォール設定

お使いのシステムでファイアウォール (SuSEfirewall2) が動作している場合、ファイアウォールでポートを開くを選択すると、YaST は nfs サービスを有効化することでファイアウォールの NFS サーバ向け設定を実施します。

26.4.1 NFSv4 クライアント向けのエクスポート

NFSv4 のクライアントに対応するには、NFSv4 を有効にするを選択します。正しくエクスポートされていれば、NFSv3 のクライアントについてもアクセスできるようになります。詳しくは 26.4.3 項「v3 と v4 エクスポートの共存」(487 ページ) をお読みください。

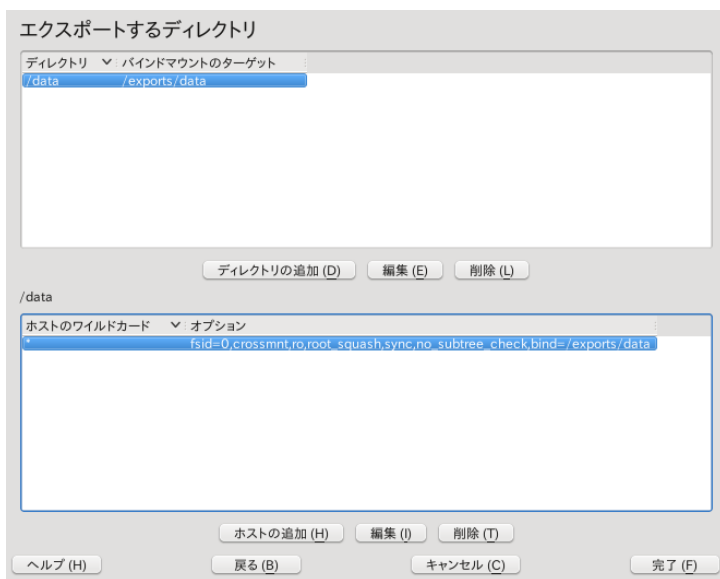
NFSv4 を有効にしたあとは、適切なドメイン名を入力してください。なお、入力したドメイン名は、NFSv4 クライアント側にある /etc/idmapd.conf ファイル内のどこかに存在していて、サーバへのアクセスするように設定してあることを確認してください。このパラメータは idmapd サービスが利用す

るもので、NFSv4に対応する場合にサーバおよびクライアントの両方で必要となります。特に要件がなければ、localdomain のままでもかまいません。詳しくは 26.7 項「さらなる情報」(492 ページ)にあるリンクを ご覧ください。

指定が終わったら次へを押します。すると、2つのセクションから構成されるダイアログが表示されます。上半分は2つの列から構成され、それぞれディレクトリとバインドマウントのターゲットという名前が設定されています。ディレクトリは直接編集可能な列で、エクスポートするディレクトリを設定します。

クライアントが固定されている場合であれば、2種類のディレクトリをエクスポートすることができます。1つは擬似的なルートファイルシステムとして動作するディレクトリ、もう1つは疑似ファイルシステムのうちの何らかのサブディレクトリとしてバインドするものとしてエクスポートできます。この擬似的なファイルシステムは、同じクライアントに対してエクスポートする全てのファイルシステムについて、ベースポイントとして動作するようになります。単一または複数のクライアントに対しては、サーバ上のいずれか1つのディレクトリだけをエクスポート用に設定することができます。このクライアントの場合は、擬似的なルート内に複数の既存のサブディレクトリをバインドすることで、複数のディレクトリをエクスポートすることができます。

図 26.4 NFSv4 を利用したディレクトリのエクスポート



ダイアログの下半分では、各ディレクトリに対するクライアント(ワイルドカード)とエクスポートオプションを設定します。上半分でディレクトリを追加した後であれば、クライアントとエクスポートオプションを設定する追加ダイアログが自動で表示されます。後から新しいクライアント(クライアントセット)を追加したい場合は、ホストの追加を押してください。

開いた小さいダイアログでは、まずホストのワイルドカードを指定します。各ホストに対して、4種類の方法でワイルドカードを指定することができます: 単一ホスト(ホスト名またはIPアドレス)、ネットグループ、ワイルドカード(*を指定した場合は全てのマシンを意味します)、IPネットワークの4種類です。次にオプションの項目では、疑似ルートディレクトリを作成する場合は `fsid=0` を含めてカンマ区切りのリストを作成します。また設定中のディレクトリが、既に設定済みの疑似ルート下にある他のディレクトリに対し、バインドされる必要のある場合は、`bind=/target/path` のような形式でターゲットのバインドパスをオプション一覧に追加してください。

たとえば `/exports` を疑似ルートディレクトリとして選択し、全てのクライアントに対してサーバへのアクセスを許可する場合を考えます。上半分ではこのディレクトリを追加し、そのオプションには `fsid=0` を設定します。そ

の他のディレクトリ、たとえば /data を NFSv4 でエクスポートする必要がある場合も、上半分にディレクトリを追加します。ただし、このディレクトリに対するオプション入力では、一覧の中に bind=/exports/data を設定し、/exports 以下に /exports/data というサブディレクトリが存在していることを確認してください。bind=/target/path オプション内の変更は、追加／変更／削除とも バインドマウントのターゲットに反映されます。この列は 直接編集可能ではありませんが、ディレクトリの概要や性質を表わしたのになります。設定を全て完了したら、完了を押してください。サービスはすぐに利用できるようになります。

26.4.2 NFSv3 と NFSv2 のエクスポート

最初のダイアログでは、次へを押す前に NFSv4 を有効にする のチェックが付けられていないことを確認してください。

次のダイアログは 2 つの部分から構成されます。上半分のテキストフィールドにはエクスポートするディレクトリを入力します。下半分ではそれらにアクセスする権利のあるホストを指定します。各ホストに対して、4 種類の方法でワイルドカードを指定することができます: 単一ホスト (ホスト名または IP アドレス)、ネットグループ、ワイルドカード (* を指定した場合は全てのマシンを意味します)、IP ネットワークの 4 種類です。

ここでのダイアログは 図26.5 「NFSv2 と v3 を利用したディレクトリのエクスポート」 (487 ページ) に示されているとおりです。これらのオプションについて、詳しい説明は man exports をお読みください。完了を押すと設定を完了します。

図 26.5 NFSv2 と v3 を利用したディレクトリのエクスポート



26.4.3 v3 と v4 エクスポートの共存

NFSv3 と NFSv4 のエクスポートは、サーバ内で共存させることができます。設定 ダイアログの冒頭で NFSv4 を有効に設定したあとは、オプション内に `fsid=0` や `bind=/target/path` が記述されていないものが v3 エクスポートと判断されます。図26.3「YaST を利用した NFS サーバ設定」(483 ページ)にある 設定例から `/data2` のような他のディレクトリを追加する場合は、ディレクトリの追加を押してオプション 内に `fsid=0` や `bind=/target/path` を含めないようにして設定すると、そのエクスポートを v3 エクスポートとして利用できるようになります。

重要項目

自動的なファイアウォール設定

お使いのシステムでファイアウォール (SuSEfirewall2) が動作している場合、ファイアウォールでポートを開くを選択すると、YaST は nfs サービスを有効化することでファイアウォールの NFS サーバ向け設定を実施します。

26.5 手作業でのファイルシステムのエクスポート

NFS のエクスポートサービスで利用する設定ファイルは、`/etc/exports` と `/etc/sysconfig/nfs` です。これらのファイルに加え、NFSv4 サーバの設定には `/etc/idmapd.conf` ファイルが必要です。また、サービスを開始したり再起動したりするには、`rcnfsserver restart` コマンドを実行します。このコマンドでサービスを起動すると、`/etc/sysconfig/nfs` で NFSv4 を設定している場合、`rpc.idmapd` についても起動を行ないます。なお、NFS サーバを起動するには `RPC portmapper` が動作している必要があります。そのため、`rcrpcbind restart` を実行して、`portmapper` サービスについても起動または再起動を行なってください。

26.5.1 NFSv4 を利用したファイルシステムのエクスポート

NFSv4 は openSUSE における最新の NFS プロトコルバージョンです。NFSv4 でディレクトリをエクスポートするための設定は、従来の NFS バージョンの設定とは少し異なります。

`/etc/exports`

`/etc/exports` ファイルには項目の一覧が書かれています。それぞれの項目には、共有されるディレクトリと、それをどのように共有するかを記述します。`/etc/exports` ファイルでの一般的な設定は下記の形式で記述します：

```
/shared/directory host(option_list)
```

たとえば下記ようになります：

```
/export 192.168.1.2(rw,fsid=0,sync,crossmnt)
/export/data 192.168.1.2(rw,bind=/data,sync)
```

ここで、IP アドレス 192.168.1.2 は利用を許すクライアントを識別するために使用しています。ホスト名を使用することもできますし、ワイルドカードを利用してホストの集合を指定することもできます (*.abc.com や * などのようになります) し、ネットグループを指定することもできます (@my-hosts)。

また、fsid=0 を指定したディレクトリは特別なディレクトリで、エクスポートされるファイルシステムの根幹 (ルート) を形成するものです。これは擬似ルートファイルシステムとも呼ばれ、NFSv4 で正しく動作するには、crossmnt オプションもあわせて設定しなければなりません。NFSv4 経由でエクスポートされるその他全てのディレクトリは、このポイント以下にマウントされなければなりません。この擬似ルート以下に存在していないディレクトリをエクスポートしたい場合は、エクスポート済みのツリーに結びつける方法をとらなければなりません。このような結びつけは、bind= 文法によって行ないます。

たとえば上記の例では /data は /export 以下に存在していませんが、これをエクスポートしたい場合は /export/data としてエクスポートし、そのエクスポート位置に /data を結びつけて (バインドして) エクスポートしてください。なお /export/data ディレクトリは存在していなければならず、通常は中には何も存在しないディレクトリであるべきものです。

クライアントからこのサーバにマウントを行なう場合は、servername:/export ではなく servername:/ としてマウントを行ないます。なお、servername:/data をマウントする必要はなく、servername:/ をマウントするだけでそのディレクトリが見えるようになります。

/etc/sysconfig/nfs

/etc/sysconfig/nfs ファイルには、NFSv4 サーバデーモンの動作を決定するいくつかのパラメータが含まれています。なお、NFS4_SUPPORT パラメータで yes を設定しなければならないことに注意してください。と設定することです。NFS4_SUPPORT により、NFS サーバが NFSv4 のエクスポートとクライアントに対応するかどうかが決まります。

/etc/idmapd.conf

Linux マシンにおいて、各ユーザには名前と ID が割り当てられています。idmapd はサーバに対して NFSv4 リクエストが届いた場合に、名前から ID への変換を行ない、クライアントに応答する仕組みを備えています。NFSv4 では名前だけで通信を行なう都合から、サーバとクライアントの両方で動作していなければなりません。

なお、NFSを利用してファイルシステムの共有を行なうマシン間では、ユーザ名と ID (uid) が共通化されるように設定してください。NIS や LDAP などの方法や、それ以外のドメイン認証の仕組みをお使いの環境で利用することで、これを設定することができます。

また、/etc/idmapd.conf ファイル内の Domain パラメータは、サーバとクライアントで同じになるように設定してください。よくわからない場合は、サーバとクライアントで両方とも既定の localdomain ドメインのままにしておいてください。設定ファイルは、たとえば下記のようになります:

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]

Nobody-User = nobody
Nobody-Group = nobody
```

より詳しい参照先として、idmapd と idmapd.conf の各マニュアルページがあります。それぞれ `man idmapd`, `man idmapd.conf` コマンドでお読みください。

サービスの起動と停止

/etc/exports や /etc/sysconfig/nfs を変更したあとは、`rcnfsserver restart` を実行し、NFS サーバサービスを開始または再起動してください。/etc/idmapd.conf ファイルを変更した場合は、`killall -HUP rpc.idmapd` コマンドで設定ファイルを再読み込みさせてください。

NFS サービスをシステムの起動時に開始させたい場合は、`chkconfig nfsserver on` コマンドを実行してください。

26.5.2 NFSv2 と NFSv3 を利用したファイルシステムのエクスポート

この章では、NFSv3 と NFSv2 でのエクスポートについて扱っています。NFSv4 の場合は、26.4.1項「NFSv4 クライアント向けのエクスポート」(483 ページ)をお読みください。

NFS 経由でのファイルシステムのエクスポートを行なうには、2つの設定ファイルを編集する必要があります。1つは `/etc/exports` で、もう1つは `/etc/sysconfig/nfs` です。`/etc/exports` ファイルでの一般的な設定は下記の形式で記述します:

```
/shared/directory    host(list_of_options)
```

たとえば下記のようになります:

```
/export    192.168.1.2(rw,sync)
```

上記の例では、ディレクトリ `/export` はホスト `192.168.1.2` に対して公開され、オプションとして `rw, sync` が指定されています。上記の IP アドレスはクライアントのホスト名で書くこともできますし、ワイルドカードを利用した指定(たとえば `*.abc.com`)やネットグループを利用する方法でも指定することができます。

全てのオプションとそれらの意味について、詳しくは `exports` のマニュアルページをお読みください(`man exports` コマンドで表示することができます)。

`/etc/exports` や `/etc/sysconfig/nfs` を変更したあとは、`rcnfsserver restart` を実行し、`nfs` サーバサービスを開始または再起動してください。

26.6 Kerberos を利用した NFS

NFS の認証として Kerberos を利用するには、まず GSS セキュリティを有効にしなければなりません。これを行なうには、YaST NFS サーバの最初のダイアログで、GSS セキュリティを有効にするを選択してください。なお、この機能を利用するには、あらかじめ動作する Kerberos サーバを用意しておく必要があります。YaST ではサーバを設定することはできず、その機能を利用

することしか行なうことができません。YaST の設定に加えて Kerberos の認証を利用したい場合は、NFS の設定を行なう前に少なくとも下記の手順を実施しておいてください:

- 1 まずサーバとクライアントの両方で、同じ Kerberos ドメインを設定していることを確認します。これは、サーバとクライアントの両方で同じ KDC (鍵配布センター) を利用し、krb5.keytab ファイル (既定ではどのマシンでも /etc/krb5.keytab 内に存在します) を共有しなければならないこととなります。Kerberos について、詳しくは第6章 *Network Authentication with Kerberos* (↑セキュリティガイド) をお読みください。
- 2 `rcgssd start` コマンドを実行し、gssd サービスを起動します。
- 3 さらに `rcsvcgssd start` コマンドを実行し、サーバ上で `svcgssd` サービスを起動します。

NFS を Kerberos 化するための設定について、詳しい情報は 26.7 項「さらなる情報」(492 ページ)にあるそれぞれのリンクを参照してください。

26.7 さらなる情報

`exports`, `nfs`, `mount` の各マニュアルページに加え、NFS サーバやクライアントを設定するための情報が `/usr/share/doc/packages/nfsidmap/README` ファイルに存在しています。オンラインの文書については、それぞれ下記の Web ドキュメントにあります:

- SourceForge [<http://nfs.sourceforge.net/>]: オンラインでの詳細な技術文書 (英語)
- NFS バージョン 4 オープンソース参照実装 [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>]: Kerberos 化した NFS を設定するための手順 (英語)
- Linux NFSv4 よくある質問 (Frequently Asked Questions) [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>]: NFSv4 について何らかの質問がある場合の参照先。

Samba

Samba を利用することで、Unix マシンを Mac OSX や Windows, OS2 マシン向けのファイルサーバ兼印刷サーバとして設定することができます。Samba はそれ単体で完結する製品であり、それなりに複雑な仕組みを備えています。YaST から Samba を設定することができるほか、SWAT (Web インターフェイス) や設定ファイルを手作業で編集することでも設定を行なうことができます。

27.1 用語

Samba のドキュメンテーション内、および YaST のモジュールでは、下記に示すいくつかの用語を使用します。

SMB プロトコル

Samba では、NetBIOS と呼ばれるサービスをベースにした SMB (server message block) プロトコルを利用しています。Microsoft では本プロトコルを公開しているため、他のソフトウェアの製造元から Microsoft のドメインネットワークに接続できるようになっています。Samba では、SMB プロトコルは TCP/IP プロトコル上で動作するようになっているため、全てのクライアントで TCP/IP プロトコルをインストールしなければなりません。

CIFS プロトコル

CIFS (common Internet file system) は Samba でサポートされている もう 1 つのプロトコルです。CIFS はネットワークを介してリモートのファイルシステムにアクセスするプロトコルで、ユーザグループ内で共同作業を

行なうことができるほか、ネットワークを介して文書を共有したりすることができます。

NetBIOS

NetBIOSはマシン間で名前解決サービスを提供するために設計された、通信向けソフトウェアインターフェイス(API)です。マシンに対してネットワーク内での名前を予約する機能を持つものです。名前の予約が行なわれると、予約した名前を通してアクセスできるようになります。NetBIOSでは名前を集中的に管理するマシンが存在するわけではなく、その名前がネットワーク内で使用されていないことを確認することで、名前の予約を実現しています。また、NetBIOSのインターフェイスは異なるネットワーク構造でも実装できるような仕組みになっています。実装はNetBEUIと呼ばれる、ネットワークハードウェアに近いプロトコル上で動作するように作られていますが、こちらについてもNetBIOSと呼ぶ場合があります。なおNetBIOSで利用可能なプロトコルとしては、Novell社が開発したIPXプロトコルや、TCP/IP (NetBIOS via TCP/IP) があります。

TCP/IP上のNetBIOSで利用する名前は、`/etc/hosts`で定義したものやDNSで設定したものとは無関係です。NetBIOSではこれらの名前解決方法は利用せず、完全独自の解決方法を利用しています。ただし、管理上の都合やDNSを併用する都合から、DNSのホスト名と一致させておくことをお勧めします。Sambaではこれが既定で設定されています。

Samba サーバ

Sambaサーバは、SMB/CIFSとNetBIOS over IPの名前解決サービスを提供するサーバです。Linuxでは3種類のデーモンを利用します: SMB/CIFSサービスには`smbd`を、名前解決サービスには`nmbd`を、認証サービスには`winbind`をそれぞれ利用します。

Samba client

Sambaクライアントは、SMBプロトコルを利用してSambaサーバにアクセスするシステムのことで、Mac OS XやWindows, OS/2などがSMBプロトコルに対応しています。なお、各コンピュータにはTCP/IPプロトコルをインストールしておかなければなりません。またSambaでは多くのUNIXシステムに対応するクライアントを提供しています。Linuxの場合はSMB向けのカーネルモジュールが存在し、LinuxシステムのレベルでSMB資源を利用することができるようになっています。Sambaクライアントの場合は、デーモンを動作させる必要はありません。

共有

SMB サーバがクライアントに資源を提供する場合、共有と呼ばれる方法で提供を行ないます。共有はサーバに設定されたプリンタやディレクトリ (サブディレクトリを含む) を指す言葉で、共有に設定された名前で資源を公開し、クライアントはその名前でアクセスを行ないます。共有名には任意の名前を設定することができ、ディレクトリを公開するにあたってディレクトリとは別の名前を設定することができます。プリンタの場合にも名前を設定し、クライアントからはその名前でアクセスを行ないます。

DC

ドメインコントローラ (DC) は、そのドメイン内でのアカウントを処理するサーバです。データの複製を行なうには、追加のドメインコントローラを用意する必要があります。

27.2 Samba サーバのインストール

Samba サーバソフトウェアをインストールするには、YaST を起動して ソフトウェア> ソフトウェア管理 を選択してください。ソフトウェア管理のモジュールが起動したら、フィルタ> パターンを選択してから *File Server* を選択し、インストールしてください。最後にインストール処理を完了するため、インストールされるパッケージ内容を確認します。

27.3 Samba の起動と停止

Samba サーバは自動で (システムの起動時に) 起動したり停止したりすることができますほか、手動で開始したり停止したりすることができます。起動や停止のポリシーについて設定を行なうには、YaST の Samba サーバ設定を利用します。詳しくは 27.4.1 項 「YaST を利用した Samba サーバの設定」 (496 ページ) をお読みください。

YaST から Samba サービスを起動したり停止したりするには、YaST から システム> システムサービス (ランレベル) を選択します。そこからそれぞれ winbind, smb, nmb について選択してください。コマンドラインから実施する場合は、`rcsmb stop && rcnmb stop` コマンドを実行すると Samba の停止を、`rcnmb start && rcsmb start` コマンドを実行すると Samba の起

動をそれぞれ行なうことができます。なお、`rcsmb` で `winbind` の処理を行なうため、`winbind` について作業を行なう必要はありません。

27.4 Samba サーバの設定

openSUSE® では、Samba サーバを 2 種類の方法で設定することができます。1 つは YaST を利用した設定、もう 1 つは手作業での設定です。手作業での設定はより細かい設定を行なうことができますが、YaST の GUI で提供されるような利便性はありません。

27.4.1 YaST を利用した Samba サーバの設定

Samba サーバを設定するには、YaST を起動して ネットワークサービス > *Samba* サーバを選択します。

初期の Samba 設定

初回にモジュールを起動したときには *Samba* インストールのダイアログが起動し、まずはサーバの管理に関していくつかの基本設定を行ないます。その手順の最後は *Samba* の `root` に対してパスワードを設定することで完了となります。2 回目以降の起動の場合は、*Samba* サーバ設定ダイアログが開きます。

Samba インストールダイアログは、以下の 2 つのステップから構成されます：

ワークグループまたはドメイン名

既存の名前を ワークグループまたはドメイン名から選択するか、新しい名前を入力して 次へ を押します。

Samba サーバの種類

次のステップでは、お使いのサーバを DC (ドメインコントローラ) にするかどうかを指定します。指定を終えたら 次へ を押します。

起動

Samba の起動を システム起動時に行なうか、もしくは 手動で行なうかを選択し、*OK* を押します。最後のポップアップ表示で *Samba* の管理者 (`root`) パスワードを入力してください。

このダイアログで設定した項目は、*Samba* 設定のダイアログ内の *起動*, *共有*, *識別情報*, *信頼するドメイン*, *LDAP* の設定の各タブを利用することで、後から変更することができます。

高度な Samba 設定

Samba サーバモジュールを初回に起動すると、「初期の Samba 設定」(496 ページ)で説明しているとおり、*Samba* の設定ダイアログが表示され、2 段階の手順で設定を行ないます。高度な設定は、この作業を終えた後に設定を調整する場合に利用します。

設定を変更したあとは、*OK* を押すと設定を保存することができます。

サーバの起動

起動 タブでは、Samba サーバの起動方法を設定することができます。お使いのシステムを起動する際にサービスを開始したい場合は、*システム起動時*を選択してください。手作業で開始したい場合は、*手動*を選択してください。Samba サーバの起動方法について、詳しくは 27.3 項「Samba の起動と停止」(495 ページ)をお読みください。

また、このタブでは、お使いのファイアウォールでポートを開く設定を行なうこともできます。これを行なうには、*ファイアウォールでポートを開く*を選択してください。お使いの環境に複数のネットワークインターフェイスが存在する場合は、*ファイアウォールの詳細*を押してネットワーク インターフェイスを選択し、*OK* を押してください。

共有

共有 タブでは、有効化する Samba 共有を設定することができます。共有の中には、ホームディレクトリやプリンタなど、いくつか事前に設定されているものが存在します。共有を選択して *状態変更* を押すと、*有効* と *無効* を切り替えることができます。*追加* を押すと新しい共有を追加することができます。*削除* を押すと選択した共有を削除することができます。

ユーザにディレクトリの共有を許可するを選択すると、許可するグループのメンバーが、自分自身のディレクトリを他のユーザに公開できるようになります。たとえばローカルのグループを設定する場合は *users* のように、ドメインのグループを設定する場合は *DOMAIN\Users* のように指定します。な

お、各ユーザはファイルシステム側のパーミッション 設定でも、アクセスを許可していなければならないことに注意してください。また、最大共有数では、作成可能な共有数の上限を 設定することができます。ユーザが作成した共有に対して認証無しでアクセスできるようにしたい場合は、**ゲストアクセス可能**を選択してください。

識別情報

識別情報タブでは、そのホストに対して関連づけるドメインを設定することができます(基本設定内)。また、ネットワーク内で代替のホスト名を利用したい場合も、ここで設定を行ないます(*NetBIOS* ホスト名)。このタブでは、名前解決に **Microsoft Windows Internet Name Service (WINS)** を利用するかどうかを設定することもできます。これを利用する場合は、ホスト名の解決に **WINS** を使用するを選択し、必要に応じて **DHCP** で **WINS** サーバのアドレスを取得を選択してください。より高度なグローバル設定やユーザ認証ソースを設定したい場合は、**詳細設定**を押してください。

信頼するドメイン

他のドメインのユーザに対してお使いのドメインにアクセスできるようにするには、**信頼するドメイン**タブ内で必要な設定を行ないます。新しいドメインを追加するには **追加** ボタンを、選択したドメインを削除するには **削除** ボタンを押します。

LDAP の設定

LDAP の設定タブでは、認証に使用する **LDAP** サーバについて設定を行なうことができます。**LDAP** サーバとの接続をテストするには、**接続のテスト** ボタンを押してください。高度な **LDAP** 設定を行なう場合や、既定の値を使用したい場合は、**詳細設定**を押してください。

LDAP の設定について、詳しくは第4章 **LDAP—A Directory Service** (↑セキュリティガイド)をお読みください。

27.4.2 SWAT を利用した Web 管理

Samba サーバを管理するための代替ツールとして、**SWAT (Samba Web Administration Tool)** があります。**SWAT** はシンプルな Web インターフェイスを提供するもので、Samba サーバを設定する機能を提供します。**SWAT** を利

用するには、root ユーザになっている状態で Web ブラウザから <http://localhost:901> を Web ブラウザで開いてください。Samba 用の root アカウントを設定していない場合は、システムの root アカウントをお使いください。

注記: SWAT の有効化

初期段階の Samba サーバインストールを完了しても、SWAT は有効化されません。SWAT を有効化するには、YaST から ネットワークサービス > ネットワークサービス (*xinetd*) を開き、ネットワークサービスの設定を開いてください。その状態から一覧内にある *swat* を選択し、状態を変更する (オン/オフ) を押して切り替えてください。

27.4.3 手作業でのサーバ設定

Samba をサーバとして利用したい場合は、samba パッケージをインストールしてください。Samba で中心となる設定ファイルは `/etc/samba/smb.conf` で、このファイルは大きく分けて 2 つのパートから構成されています。

[global] セクションにはサーバ全体の設定が含まれ、[share] セクションでは個別のファイルまたはプリンタ共有に関する設定が含まれています。この設定の仕組みにより、共有に関する詳細設定と [global] セクションの設定はそれぞれ別々に行なうことができるため、設定ファイルの構造的な透過性が保たれるようになっています。

グローバルセクション

[global] セクション内の下記のパラメータは、お使いのネットワーク環境に合わせて変更すべき箇所です。Windows 環境の他のマシンからお使いの Samba サーバにアクセスするため、必要な項目です。

```
workgroup = TUX-NET
```

この行では、Samba サーバに対してワークグループを設定しています。お使いの環境にあわせて TUX-NET を適切な文字列に変更してください。その名前が別のマシンに割り当てられていない限り、ご利用の Samba サーバは DNS に登録した名前で見られるようになります。DNS に名前を登録することができない場合は、`netbiosname=名前` の書式でサーバ名を設

定してください。このパラメータについて、詳しくは `smb.conf` のマニュアルページをお読みください。

`os level = 20`

このパラメータは、お使いの **Samba** サーバがワークグループ内の **LMB** (ローカルマスターブラウザ)になるかどうかを指定するものです。 **Samba** バージョン 3 シリーズでは、既定の値 (20) を上書きする必要はほとんどありません。また、誤った **Samba** サーバの設定によって既存の **Windows** ネットワークが混乱することを防ぐため、2 のようなとても低い値に設定しておいてください。この話題について、詳しくは **Samba 3 Howto** 内の **Network Browsing** (ネットワークブラウズ) 関連の章をお読みください。 **Samba 3 Howto** については 27.7 項「さらなる情報」(506 ページ)をお読みください。

お使いのネットワーク内に他の **SMB** サーバ (**Windows 2000 Server** など) が存在しない場合や、ローカル環境の全システムの一覧を **Samba** サーバで管理したい場合は、`os level` の値をより高い値に設定してください(たとえば 65 など)。これによりお使いの **Samba** サーバは、ローカルネットワーク内での **LMB** として動作するようになります。

なお、この設定を変更する場合は、既存の **Windows** ネットワーク環境にどのような影響があるのかをよく確かめてから行なってください。まずは本番の環境とは切り離されたネットワークやそれほど重要ではないネットワークで、設定を試してみてください。

`wins support` と `wins server`

お使いの **Samba** サーバを **WINS** サーバの存在する既存の **Windows** ネットワークに追加する場合は、`wins server` オプションを有効に設定し、**WINS** サーバの **IP** アドレスを設定してください。

お使いの **Windows** マシンが別のサブネットに接続されていて、**Samba** サーバとは互いに存在を通知しあいたい場合は、**WINS** サーバを設定する必要があります。 **Samba** サーバで **WINS** サーバの機能を動作させるには、`wins support = Yes` のオプションを設定してください。なお、同じネットワーク内では 1 台までの **Samba** サーバで有効に設定してください。また、`smb.conf` ファイルでは `wins server` と `wins support` の両方をいっぺんに設定してはなりません。

共有

下記の例では、**CD-ROM** ドライブとユーザディレクトリ (homes) を **SMB** クライアントに公開する場合を想定し、設定例を示しています。

[cdrom]

CD-ROM ドライブを誤って共有してしまうことを防ぐため、下記の行はコメントマークを利用して無効化してあります(この設定ファイルの場合はセミコロンを書きます)。**CD-ROM** ドライブを **Samba** で共有する場合は、各行の冒頭にあるセミコロンを取り除いてください。

例 27.1 *CD-ROM 共有 (無効化)*

```
;  
[cdrom]  
;  
;    comment = Linux CD-ROM  
;    path = /media/cdrom  
;  
;    locking = No
```

[cdrom] と comment

[cdrom] の項目は、ネットワーク上から **SMB** クライアントが参照する名前です。comment 行を利用することで、さらに細かい説明を追加することができます。

```
path = /media/cdrom
```

path オプションでは、/media/cdrom ディレクトリを公開するように指定しています。

なお、既定の設定は安全性を第一に考えられたものであるため、これらの共有はこのシステムに存在するユーザに対してのみ公開されます。この共有を誰にでも利用できるようにするには、設定内に `guest ok = yes` という行を追加してください。この設定を行なうことで、ネットワーク内の誰にでも読み取りアクセスを許可ようになります。なお、このパラメータを利用するにあたっては、[global] セクションでの設定よりも優先して動作することにご注意ください。

[homes]

[homes] 共有は特別な意味を持つ共有です。**Linux** のファイルサーバ内に有効なアカウントとパスワードが存在し、かつそのユーザがホームディレクトリを持っていれば、この共有にアクセスできるようになります。

例 27.2 [homes] 共有

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
create mask = 0640
directory mask = 0750
```

[homes]

SMB サーバに対して接続しているユーザと同じ名前の共有が存在しない限り、[homes] 共有の設定で共有が動的に生成されます。共有の名前はユーザ名と同じです。

`valid users = %S`

%S は接続が正しく行なわれたときに実際の共有名に置き換えられる部分で、[homes] の共有では常にユーザ名に置き換えられます。そのため、ユーザ名の共有はそのユーザに対してだけ許可される仕組みになっています。

`browseable = No`

この設定を行なうと、ネットワーク環境から参照することができないようになります。

`read only = No`

既定では Samba は `read only = Yes` の設定が有効になっていて、いかなる共有に対する書き込みアクセスも禁止される仕組みになっています。共有に書き込みできるようにするには、`read only = No` を設定してください。 `writable = Yes` でも同じ設定になります。

`create mask = 0640`

MS Windows NT をベースにしていないシステムでは UNIX のパーミッションを解釈することができないため、ファイルを作成する際に割り当てるパーミッションを設定しておく必要があります。 `create mask` のパラメータでは、新しく作成するファイルに対して設定するパーミッションを設定します。なお、この設定は書き込み可能な共有に対してのみ効果がある項目です。上記の設定では、所有者自身は読み込みおよび書き込みの権限を持ち、所有者のプライマリグループのメンバーは読み込みの権限だけを持ちます。ただし、`valid users = %S` の設定が存在するため、グループに対して読み込み権限がある

ものの、その権限は実際には有効になりません。グループに対して読み込みや書き込みの権限を与えるには、`valid users = %S`の行を無効化する必要があります。

セキュリティレベル

セキュリティを改善するため、それぞれの共有にはパスワードを設定することができます。SMB では下記の方法で許可をチェックすることができます:

共有レベルセキュリティ (security = share)

共有に対してパスワードを設定します。このパスワードを知っているユーザだけが、この共有にアクセスできます。

ユーザレベルセキュリティ (security = user)

この設定はSMBのユーザに対する考え方を提供するものです。この設定では、あらかじめ各ユーザをサーバに登録し、パスワードを設定しなければなりません。登録後、サーバはそのユーザ名に対して許可した共有にアクセスできるようになります。

サーバレベルセキュリティ (security = server)

クライアントに対しては、Sambaはユーザレベルと同じ振る舞いを見せます。ただしユーザレベルセキュリティとは異なり、入力されたパスワードは他のサーバに転送され、そこで認証を行ないます。この設定の場合、追加のpassword server パラメータを設定する必要があります。

ADS レベルセキュリティ (security = ADS)

このモードでは、SambaはActive Directory環境のドメインメンバーとして動作します。このモードで動作するには、Sambaの動作するマシンにKerberosをインストールし、設定しておかなければなりません。また、そのマシンはSambaを利用してADSの領域に参加しなければなりません。これらの設定は、YaST Windows ドメインメンバーシップモジュールから行なうことができます。

ドメインレベルセキュリティ (security = domain)

このモードは、そのマシンがWindows NT ドメインに参加している場合にのみ動作します。この設定でSambaは、ユーザ名とパスワードをWindows NTのプライマリまたはバックアップのドメインコントローラに送信し、検証を行ないます。これはWindows NT Serverが行なう方法と同じです。

なお、暗号化パスワードを指定するパラメータについて、yes に設定する必要があります。

共有／ユーザ／サーバ／ドメインの各レベルのセキュリティは、サーバ全体に適用されます。共有レベルのセキュリティとユーザレベルのセキュリティを混在させた共有を設定してサーバ設定を行なうことはできません。ただし、システムに設定した各 IP アドレスに対し、別々の Samba サーバを設定して起動することは可能です。

この種類の話題について、より詳しい情報は **Samba 3 HOWTO** に書いてあります。1つのシステムで複数のサーバを立ち上げる場合は、`interfaces` と `bind interfaces only` の各オプションについて注意して設定してください。

27.5 クライアントの設定

Samba クライアントでは、TCP/IP で提供される Samba サーバにのみアクセスすることができます。NetBEUI や IPX 上の NetBIOS ではアクセスできません。

27.5.1 YaST を利用した Samba クライアントの設定

Samba サーバや Windows サーバ上に存在する資源 (ファイルやプリンタ) にアクセスするには、Samba クライアントを設定します。ネットワークサービス > **Windows** ドメインメンバーシップで表示されるダイアログで、それぞれ NT ドメインや Active Directory ドメイン、またはワークグループ名を入力してください。*Linux* の認証にも *SMB* の情報を使用するを選択すると、ユーザ認証を Samba や NT、または Kerberos サーバ上で行なうようになります。

高度な設定オプションを表示するには、**熟練者向け設定**を押します。たとえば認証時にサーバのホームディレクトリを自動でマウントしたい場合は、サーバディレクトリのマウントを選択します。この方法を利用することで、ユーザは **CIFS** 経由で提供されたホームディレクトリにアクセスできるようになります。詳しくは `pam_mount` のマニュアルページをお読みください。

全ての設定を終えたら、 **完了** を押して設定を完了してください。

27.6 ログインサーバとしての Samba の利用

主として Windows クライアントが多く存在するネットワークでは、有効なアカウントとパスワードだけを登録しておくことが望めます。Windows ベースのネットワークでは、これはプライマリドメインコントローラ (PDC) を利用して行ないます。PDC として設定済みの Windows NT server を利用して設定することもできますが、Samba サーバを利用することでもこの作業を行なうことができます。なお、smb.conf ファイル内の [global] セクションに対し、例27.3「smb.conf のグローバルセクション」(505 ページ) で示されるような設定を行なっておかなければなりません。

例 27.3 smb.conf のグローバルセクション

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

ユーザ認証時に暗号化パスワードを利用する場合、Samba サーバでもこれを有効に設定しなければなりません。[global] セクション内で encrypt passwords = yes を設定することでこれを実現することができます (Samba バージョン 3 では、これが既定値になっています)。また Windows の慣習に従うため、ユーザアカウントとパスワードの情報についても暗号化した書式が必要になります。これを行なうには、smbpasswd -a name を実行してください。また、Windows のドメインの考え方で必要となる、ドメイン内のコンピュータアカウントを作成するには、下記のコマンドを実行します:

```
useradd hostname\${
smbpasswd -a -m hostname
```

なお、useradd コマンドにドル記号が追加されていることに注意してください。smbpasswd コマンドについては、-m パラメータを指定することで自動的にドル記号が追加されます。設定例 (/usr/share/doc/packages/samba/examples/smb.conf.SUSE) のコメント内に、この処理を自動化するための設定が書かれています。

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \  
-s /bin/false %m$
```

Samba サーバで上記のスクリプトを正しく実行できるようにするため、Samba ユーザ に対して必要な管理者権限と ntadmin グループへの追加を行なってください。これを行なったあと、この Linux グループに属する全ユーザを Domain Admins に割り当てるため、下記のコマンドを実行します:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

この種類の話題について詳しい情報は、Samba 3 HOWTO 内の第 12 章に書かれています。/usr/share/doc/packages/samba/Samba3-HOWTO.pdf をお読みください。

27.7 さらになる情報

Samba に関する詳細な情報は、デジタル文書で提供されています。コマンドラインから `apropos samba` を実行して複数のマニュアルページを参照するか、Samba のドキュメンテーションがインストールされている場合は、/usr/share/doc/packages/samba ディレクトリ内をお読みのうえ、オンライン文書と設定例を参照してください。examples サブディレクトリ内には、コメントの付けられた設定例 ファイル (smb.conf.SUSE) があります。

また、Samba チーム提供の Samba 3 HOWTO には、トラブルシューティングに関する章があります。これに加え、文書の Part V には設定を確認するための手順が書かれています。Samba 3 HOWTO は samba-doc パッケージに含まれ、/usr/share/doc/packages/samba/Samba3-HOWTO.pdf ファイルからアクセスすることができます。

また、openSUSE wiki の Samba ページ <http://ja.opensuse.org/Samba> についてもお読みください。

Apache HTTP サーバ

Apache HTTP サーバ (Apache) は、<http://www.netcraft.com/> の調査によると 50% 以上ものシェアを持つ Web サーバで、世界で最もよく使用されている Web サーバです。Apache Software Foundation (<http://www.apache.org/>) で開発されている Apache は多くのオペレーティングシステムで利用することができます。openSUSE® には Apache バージョン 2.2 が含まれています。この章では、Apache Web サーバのインストールから設定、SSL や CGI の使用方法、およびトラブル発生時の解決方法について、それぞれ言及しています。

28.1 クイックスタート

この章を読み進めていくことで、Apache の設定と起動を簡単に行なうことができます。Apache のインストールと設定は、root の状態から行ないます。

28.1.1 事前要件

Apache Web サーバを設定する前に、下記の要件を全て満たしていることをご確認ください：

1. 対象となるマシンのネットワーク設定が正しく行なわれていることを確認してください。詳しくは第21章 ネットワークの基礎(353 ページ)をお読みください。

2. 対象となるマシンのシステム時刻が、タイムサーバを利用して同期できていることを確認してください。これは、**HTTP** プロトコルがシステムの時刻に依存して動作していることによるものです。詳しくは 第25章 *NTP* を利用した時刻同期 (467 ページ) をお読みください。
3. 最新のセキュリティ更新がインストールされていることを確認してください。**YaST** オンライン更新を利用することで、最新かどうかを確認することができます。
4. 既定の **Web** サーバポート (ポート 80) がファイアウォールで開くように設定されていることを確認してください。これを行なうには、**SuSEFirewall2** で外部ゾーンに対し、**HTTP** サーバを許可するよう設定する必要があります。この作業は **YaST** から行なうことができます。詳しくは 項「Configuring the Firewall with YaST」 (第14章 *Masquerading and Firewalls*, ↑セキュリティガイド) をお読みください。 details.

28.1.2 インストール

openSUSE において、**Apache** は既定ではインストールされません。「すぐに使うことのできる」標準設定でインストールを行なうには、下記の手順を行ないます:

手順 28.1 既定の設定での *Apache* インストール

- 1 **YaST** を起動し、ソフトウェア> ソフトウェア管理 を選択します。
- 2 フィルタ> パターン を選択し、サーバ機能 内の *Web* および *LAMP* サーバ を選択します。
- 3 依存関係のパッケージのインストールについて確認を行ない、インストール処理を 完了します。

上記の手順でのインストールには、**PHP5** モジュールと **apache2-prefork** マルチプロセッシングモジュールが含まれています。モジュールについて、詳しくは 28.4 項「モジュールのインストール／有効化／設定」 (530 ページ) をお読みください。

28.1.3 起動

Apache はシステムの起動時に自動起動するように設定することができるほか、手動で起動するようにも設定できます。

手順 28.2 Apache の自動起動設定

- 1 まずはランレベル 3 および 5 でシステムが起動する際、Apache が自動的に起動するかどうかを確認します。具体的には下記のコマンドを入力します:

```
chkconfig -a apache2
```

- 2 上記以外にも、YaST を起動して システム > システムサービス (ランレベル) を選択することでも設定することができます。
- 3 *apache2* の項目を探し、有効にする ボタンを押してサービスを有効に設定します。

これで Web サーバが起動します。

- 4 最後に設定を保存するため、完了 ボタンを押します。

これでシステムの起動時、ランレベルが 3 および 5 であれば Apache が自動で起動するようになります。

openSUSE におけるランレベルについて、および YaST ランレベルエディタについては 16.2.3 項「YaST を利用したシステムサービス (ランレベル) の設定」(269 ページ) をお読みください。

シェルを利用して Apache を手動で起動するには、`rcapache2 start` コマンドを実行します。

手順 28.3 Apache が起動しているかどうかの確認

Apache の起動時に何もエラーメッセージが表示されていなければ、通常 Web サーバは起動している状態になっているはずです。確認は下記の手順で行ないます:

- 1 ブラウザを起動し、<http://localhost/> を開きます。

Apache が起動していて問題なく動作していれば、「It works!」で始まるテストページが表示されるはずです。

- 2 上記のページが表示されない場合は、28.8項「トラブルシューティング」(552 ページ)をお読みください。

これで Web サーバを起動することができました。必要なドキュメントを追加したり、必要に応じて設定を調整したり、モジュールをインストールして機能を追加したり することができるようになっています。

28.2 Apache の設定

openSUSE における Apache は、2 種類の方法で設定することができます:

- Apache の手作業による設定 (514 ページ)
- YaST を利用した Apache の設定 (519 ページ)

手作業での設定ではより細かい設定を行なうことができますが、YaST GUI のような利便性は 備わっていません。

重要項目: 設定変更後の Apache の再読み込みまたは再起動について

設定のうちの多くは、変更された設定を反映するために Apache の再読み込み(または再起動)が必要です。手作業で Apache を再読み込みさせるには、`rcapache2 reload` コマンドを 実行するか、もしくは 28.3項「Apache の起動と停止」(527 ページ)に書かれている再起動オプションを利用します。

YaST を利用して Apache を設定する場合、「HTTP サーバ設定」(524 ページ)に書かれている *HTTP* サービスが有効に設定されていると、YaST が自動で判断して処理を行ないます。

28.2.1 Apache の設定ファイル

この章では、Apache の設定ファイルについて概要を示しています。YaST を利用して 設定を行なう場合は、これらのファイルを直接操作する必要はあり

ません—ただし、後から手作業による設定に切り替える場合に備え、これらの情報を知っておくとよいでしょう。

Apache の設定ファイルは 2 箇所の異なる場所に配置されています:

- /etc/sysconfig/apache2 (511 ページ)
- /etc/apache2/ (511 ページ)

/etc/sysconfig/apache2

/etc/sysconfig/apache2 では、読み込むべきモジュールや取り込むべき追加設定ファイル、サーバ起動時の設定フラグやコマンドラインに追加すべきフラグなど、Apache に関するいくつかのグローバル設定を保持しています。このファイル内にある各設定オプションは詳しくドキュメント化されているため、ここでは敢えて説明を行いません。一般用途の Web サーバであれば、/etc/sysconfig/apache2 内にある設定だけで十分に要件を満たすことができます。

/etc/apache2/

/etc/apache2/ では Apache に必要な全ての設定ファイルを保持しています。下記ではこのディレクトリ以下にある各ファイルの目的について説明を行います。各ファイルにはいくつかの設定オプション (ディレクティブと表現する場合もあります) が含まれていて、これらのファイルに含まれる設定オプションは詳しくドキュメント化されているため、ここでは一つずつの設定オプションについて説明を行なうことはしません。

Apache の設定ファイルは下記のような構成になっています:

```
/etc/apache2/  
|  
| - charset.conv  
| - conf.d/  
|   |  
|   | - *.conf  
|  
| - default-server.conf  
| - errors.conf  
| - httpd.conf  
| - listen.conf  
| - magic  
| - mime.types
```

```

|- mod_*.conf
|- server-tuning.conf
|- ssl.*
|- ssl-global.conf
|- sysconfig.d
|   |
|   |- global.conf
|   |- include.conf
|   |- loadmodule.conf . .
|
|- uid.conf
|- vhosts.d
|   |- *.conf

```

/etc/apache2 内の Apache 設定ファイル

charset.conf

様々な言語に対して使用する文字セットを指定するファイルです。編集を行なってはいけません。

conf.d/*.conf

他のモジュールから追加される設定ファイルです。これらの設定ファイルは、必要に応じてお使いの仮想ホスト設定から取り込むことができます。例として `vhosts.d/vhost.template` ファイルをお読みください。これを行なうことで、それぞれの仮想ホストで別々のモジュールを利用することができます。

default-server.conf

全ての仮想ホストに適用されるグローバルな設定で、多くの既定値が設定されています。これらの値を変更するのではなく、各仮想ホスト側の設定で設定を上書きしてください。

errors.conf

Apache がどのようにしてエラー応答を行なうのかを指定します。全ての仮想ホストに対してエラーメッセージをカスタマイズするには、このファイルを編集してください。お使いの各仮想ホストの設定内でこれらを上書きすることもできます。

httpd.conf

Apache サーバの設定ファイルのメインです。このファイルについては変更は避けてください。このファイルは主にグローバルな設定を取り込むための仕組みを用意しています。それぞれのグローバルな設定を変更する際は、この一覧内にある適切な設定ファイルに設定してください。また、

各仮想ホストの設定 (たとえばドキュメントルートなど) は、仮想ホストの設定で行なってください。

`listen.conf`

Apache に対して、待ち受けるべき IP アドレスとポートを指定するファイルです。名前ベースの仮想ホストについても、こちらで設定を行ないます。詳しくは「名前ベースの仮想ホスト」(516 ページ)をお読みください。

`magic`

mime_magic モジュールが使用するデータで、**Apache** が未知のファイルについて **MIME** タイプを自動判別するのに使用します。変更しないでください。

`mime.types`

システムで既知となっている **MIME** タイプの一覧です (実際には `/etc/mime.types` へのリンクになっています)。編集はしないでください。ここに記載されていない **MIME** タイプを新しく追加したい場合は、`mod_mime-defaults.conf` ファイルに追記してください。

`mod_*.conf`

既定でインストールされるモジュール向けの設定です。詳しくは 28.4 項「モジュールのインストール／有効化／設定」(530 ページ)をお読みください。なお、任意指定のモジュールに関する設定は、`conf.d` ディレクトリ内にあります。

`server-tuning.conf`

様々な **MPM** (**MPM** については 28.4.4 項「マルチプロセッシングモジュール (**MPM**)」(536 ページ)を参照) に対する設定ディレクティブが含まれているほか、**Apache** の性能を決める一般的な設定オプションが含まれています。これらの項目について変更 する場合は、よくテストを行なってください。

`ssl-global.conf` および `ssl.*`

グローバルな **SSL** 設定と **SSL** の証明書データが含まれています。詳しくは 28.6 項「**SSL** で通信の機密を保持する Web サーバの設定」(543 ページ)をお読みください。

`sysconfig.d/*.conf`

`/etc/sysconfig/apache2` の設定から自動生成された 設定ファイルです。これらのファイルは直接編集を行わず、`/etc/sysconfig/apache2` ファイルのほうを編集してください。また、このディレクトリには自動生成されるファイル以外は配置してはなりません。

`uid.conf`

Apache が動作する際のユーザおよびグループ ID を指定するファイルです。変更はしないでください。

`vhosts.d/*.conf`

構築したい仮想ホストの設定をここに配置します。このディレクトリには SSL 無しの場合と有りの場合の両方について、仮想ホストの雛型ファイルが含まれています。また、このディレクトリ内に存在し、`.conf` で終わるファイルは、Apache の設定内に自動で取り込まれる仕組みになっています。詳しくは「仮想(バーチャル)ホストの設定」(514 ページ)をお読みください。

28.2.2 Apache の手作業による設定

Apache を手作業で設定する場合、ユーザ `root` からテキスト形式の設定ファイルを編集して行ないます。

仮想 (バーチャル) ホストの設定

仮想ホストとは、1 台の Apache から複数の URI (統一資源識別子) に対するサービスを提供する仕組みです。たとえば `www.example.com` と `www.example.net` のように、物理的に 1 台だけのマシンで動作する単一の Web サーバから、複数のドメインを取り扱うこともできます。

仮想ホストは、管理面の手間を省くため (管理する Web サーバを 1 台にまとめることができる) と、ハードウェアの費用を削減するため (各ドメインで専用のサーバを必要としない) に使用します。仮想ホストはホスト名 (名前ベース) を基準に設定することができるほか、IP アドレスを基準にしたりポート番号を基準にしたりすることもできます。

全ての設定済み仮想ホストを表示するには、`httpd2 -S` を利用します。これにより既定のサーバと全ての仮想ホストが表示され、それらの IP アドレス

やポート番号などがあわせて表示されます。さらに、設定ファイル内で仮想ホストが設定されている場所についても表示されます。

仮想ホストは「仮想ホスト」(523 ページ)に書かれている手順で YaST から設定することができるほか、設定ファイルを手作業で変更することによっても設定できます。openSUSE の既定では、仮想ホストごとに 1 つの設定ファイルを `/etc/apache2/vhosts.d/` 内に作成する仕組みになっています。このディレクトリ内にあり、かつファイル名が `.conf` で終わるファイルは、自動的に設定として取り込まれるようになっています。また、仮想ホストを設定する際のテンプレートについても、このディレクトリ内に配置されています。それぞれ `vhost.template` (SSL 無し) と `vhost-ssl.template` (SSL 有り) のファイルがあります。

ティップ: 仮想ホストの設定の必要性について

設定しようとしている Web サーバが 1 つのドメインだけをまかなうものであった場合であっても、仮想ホストの設定ファイルを常に作成しておくことをお勧めします。これを行なうと、ドメイン固有の設定を別途のファイルに分けて保存することになるので、単純に仮想ホストの設定ファイルを移動／削除／名前変更するだけで、うまく動作していたはずの元の基本設定に戻ることができるようになります。また同じ理由から、仮想ホストごとに別々の設定ファイルを作成してください。

なお名前ベースの仮想ホストを使用する場合、どの仮想ホスト設定にも該当しないドメイン名でアクセスが行なわれた場合に対して、既定の仮想ホスト設定を追加しておくことをお勧めします。既定の仮想ホストは設定ファイルとして最初に読み込まれるファイルで、設定ファイルの読み込み順序はファイル名で判断される仕組みになっています。そのため、アンダースコア文字 (「`_`」) で始まるファイル (たとえば `_default_vhost.conf`) を作成して設定することをお勧めします。

`<VirtualHost>``</VirtualHost>` のブロックには、それぞれのドメインに適用される情報が記載されています。Apache が設定済みのホストに対するリクエストを受け取ったときは、このセクション内のディレクティブを参照します。この仮想ホスト設定の内側では、ほとんど全てのディレクティブを参照することができます。Apache の設定ディレクティブについて、詳しくは <http://httpd.apache.org/docs/2.2/mod/quickreference.html> をお読みください。

名前ベースの仮想ホスト

名前ベースの仮想ホストの場合、IP アドレス 1 つに対して複数の Web サイトを構築することができます。この方法では、Apache はクライアントから送信されたホスト名フィールドと各仮想ホスト内の ServerName を比較し、一致する仮想ホストを検索します。ServerName に該当するものが存在しない場合、最初に設定した仮想ホストを既定のホストとして使用します。

また、NameVirtualHost ディレクティブは Apache に対し、IP アドレスや (必要であれば) ポート番号のほか、HTTP ヘッダ内に含まれるホスト名フィールドでも仮想ホストを使用するよう設定します。このオプションは設定ファイル `/etc/apache2/listen.conf` 内で設定します。

NameVirtualHost に設定する最初のパラメータは完全修飾ドメイン名ですが、IP アドレスで設定することをお勧めします。2 番目のパラメータはポート番号で、必ずしも指定する必要はありません。既定では Listen ディレクティブでの設定から、ポート 80 を使用します。

また IP アドレスやポート番号の指定には、ワイルドカード `*` を指定することもできます。この場合は全てのインターフェイスに対するリクエストを仮想ホストとして設定することを意味します。また、IPv6 アドレスを設定する場合は、大括弧で括らなければなりません。

例 28.1 様々な名前ベースの VirtualHost 設定

```
# NameVirtualHost IP-address[:Port]
NameVirtualHost 192.168.3.100:80
NameVirtualHost 192.168.3.100
NameVirtualHost *:80
NameVirtualHost *
NameVirtualHost [2002:c0a8:364::]:80
```

VirtualHost の開始タグのパラメータには、NameVirtualHost で設定した IP アドレス (または 完全修飾ドメイン名) を指定します。

NameVirtualHost で設定したポート番号については任意指定です。

IP アドレスの代用として、ワイルドカード `*` を指定することもできます。このワイルドカードは NameVirtualHost `*` を設定した場合にのみ利用できます。IPv6 アドレスを利用する場合、そのアドレス指定は大括弧で括らなければなりません。

例 28.2 名前ベースの *VirtualHost* 設定

```
<VirtualHost 192.168.3.100:80>
...
</VirtualHost>

<VirtualHost 192.168.3.100>
...
</VirtualHost>

<VirtualHost *:80>
...
</VirtualHost>

<VirtualHost *>
...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
...
</VirtualHost>
```

IP ベースの仮想ホスト

この仮想ホスト方法を利用するには、対象のマシンに複数の IP アドレスを設定する必要があります。それぞれの IP に割り当てられたドメインに対し、Apache は 1 つのインスタンスで動作します。

IP ベースの仮想ホストでは、それぞれの仮想ホストに対して 1 つの IP アドレスを設定しなければなりません。お使いのマシンに複数のネットワークカードが存在しない場合でも、仮想的なネットワークインターフェイス (IP エイリアス) も設定することができます。

下記は IP アドレス 192.168.3.100 で動作するマシンの Apache 設定例で、追加の IP アドレス 192.168.3.101 および 192.168.3.102 でそれぞれ 1 つずつのドメインをまかなう場合のものです。下記のとおりそれぞれの仮想ホストに対し、別々の *VirtualHost* ブロックを設定する必要があります。

例 28.3 IP ベースの *VirtualHost* 設定

```
<VirtualHost 192.168.3.101>
...
</VirtualHost>

<VirtualHost 192.168.3.102>
...
</VirtualHost>
```

ここでVirtualHostは、192.168.3.100以外のインターフェイスに対してのみ設定していることに注意してください。Listenディレクティブで192.168.3.100のアドレスについても設定する場合、そのアドレス宛に届いたHTTPリクエストに応答するには、別途のIPベース仮想ホストを追加しなければなりません。設定しなかった場合は、既定のサーバ設定(/etc/apache2/default-server.conf)が使用されます。

基本的な仮想ホスト設定

それぞれの仮想ホストを設定するには、少なくとも各仮想ホストの設定内に下記のディレクティブを設定する必要があります。さらに詳しいオプション設定を行ないたい場合は、/etc/apache2/vhosts.d/vhost.templateをお読みください。

ServerName

その仮想ホストに割り当てる完全修飾ドメイン名を指定します。

DocumentRoot

このホストからApacheがファイルを提供するにあたって、基準となるディレクトリを指定します。なおセキュリティ上の理由から、ファイルシステム全体に対するアクセスは既定では禁止されるようになっています。そのため、Directoryコンテナを利用することで、明示的に禁止を解除するよう設定してください。

ServerAdmin

サーバ管理者の電子メールアドレスを指定します。たとえば、このアドレスはApacheが生成するエラーページで表示されます。

ErrorLog

この仮想ホストに対するエラーログファイルを指定します。仮想ホストごとに個別のエラーログを作成する必要はありませんが、デバッグが容易になることから、個別に作成しておくことをお勧めします。/var/log/apache2/がApacheのログファイルに関する既定のディレクトリです。

CustomLog

この仮想ホストに対するアクセスログです。仮想ホストごとに個別のエラーログを作成する必要はありませんが、各ホストについて個別の分析を行なうことができるという理由から、個別に作成しておくことをお勧め

めします。 /var/log/apache2/ が Apache のログファイルに関する既定のディレクトリです。

上記のようにセキュリティ上の理由から、ファイルシステム全体へのアクセスは既定で禁止されています。そのため、Apache がサービスを提供すべきディレクトリについては、明示的に禁止を解除しなければなりません。たとえば DocumentRoot であれば、下記のようになります:

```
<Directory "/srv/www/www.example.com/htdocs">
    Order allow,deny
    Allow from all
</Directory>
```

基本的な設定をそろえると、下記のようになります:

例 28.4 基本的な VirtualHost 設定

```
<VirtualHost 192.168.3.100>
    ServerName www.example.com
    DocumentRoot /srv/www/www.example.com/htdocs
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com_log
    CustomLog /var/log/apache2/www.example.com-access_log common
    <Directory "/srv/www/www.example.com/htdocs">
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

28.2.3 YaST を利用した Apache の設定

YaST を利用して Web サーバを設定するには、YaST を起動してから ネットワークサービス>HTTP サーバを選択します。このモジュールを初めて起動した場合は HTTP サーバウイザードが起動し、サーバ管理についていくつかの基本事項を質問します。このウイザードを終了すると、以降の HTTP サーバモジュールの起動では HTTP サーバ設定ダイアログが表示されるようになります。

HTTP サーバウイザード

HTTP サーバウイザードは5つの段階から構成されています。ダイアログの最後の段階では、熟練者向けの設定を行なってより詳しい設定を施すことができます。

ネットワークデバイスの選択

ここではApacheが要求を受け付けるために利用する、ネットワークインターフェイスを指定します。既存のネットワークインターフェイスとIPアドレスから、任意の組み合わせで選択することができます。また、他のサービスが予約していない任意のポート(既知の(Well-known)ポートや登録済みのポート、および動的なポートやプライベートなポート)を使用することができます。既定の設定では、全てのネットワークインターフェイスに対して80番のポートで待ち受ける設定になっています。

Webサーバが待ち受けるポートについて、ファイアウォール側でポートを開くには、ファイアウォールでポートを開くを選択します。この設定はLANやWAN、インターネットなどのネットワークに対してWebサーバを公開する場合に必要な設定です。ポートを閉じたままにする設定は、テストなどで外部からのアクセスを禁止したい場合にのみ有用な設定です。複数のネットワークインターフェイスをご利用の場合は、ファイアウォールの詳細...を押してポートを開きたいインターフェイスを選択してください。

次へを押すと設定を続けることができます。

モジュール

モジュール設定オプションでは、Webサーバがサポートすべきスクリプト言語を有効にしたり無効にしたりすることができます。その他のモジュールを設定するには、「サーバモジュール」(526ページ)をお読みください。次へを押すと設定を続けることができます。

既定のホスト

この設定では既定のホストを設定します。「仮想(バーチャル)ホストの設定」(514ページ)で説明しているとおり、Apacheは単一のマシンで複数の仮想ホストを稼働させることができます。設定ファイル内で最初に設定したホストのことを既定のホストと言いますが、ここではこれを設定します。それぞれの仮想ホストは既定のホストの設定を引き継ぎます。

ホストの設定(ディレクティブとも呼ばれます)を編集するには、表内から変更したい項目を選んで編集を押します。新しいディレクティブを追加するには追加を、ディレクティブを削除するには対象のを選んで削除を押してください。

図 28.1 HTTP サーバウイザード: 既定のホスト

HTTPサーバウイザード (3/5) -- 既定のホスト

オプション	値
ドキュメントルート	/srv/www/htdocs*
Directory	/srv/www/htdocs*...
Alias	/icons/* /usr/share/apache2/icons/*
Directory	/usr/share/apache2/icons*...
ScriptAlias	/cgi-bin/* /srv/www/cgi-bin/*
Directory	/srv/www/cgi-bin*...
mod_userdir.c	
Include	/etc/apache2/conf.d/*.conf
Include	/etc/apache2/conf.d/apache2-manual?conf
サーバ名	linux
サーバ管理者のメールアドレス	geso-degeso@linux

追加 (A) 編集 (I) 削除 (T)

ヘルプ (H) 戻る (B) キャンセル (C) 次へ (N)

それぞれ下記にサーバの設定項目を示します:

ドキュメントルート

このホストが提供すべきファイルが存在するディレクトリを指定します。
既定では /srv/www/htdocs に設定されています。

Alias

Alias ディレクティブを使用することで、URL と物理的なファイルの場所について、割り当てを変更することができます。これにより ファイルシステム内のドキュメントルート以外の場所にファイルを配置し、URL 経由でアクセスさせることができますようになります。

openSUSE の既定値では、Alias の設定として /icons が /usr/share/apache2/icons を指すように設定されています。これにより Apache からディレクトリ一覧を参照すると、アイコンが表示されるようになります。

ScriptAlias

これはAliasディレクティブに似た設定で、ScriptAliasもURLとファイルシステムの場所について割り当てを設定します。ただしScriptAliasはCGIの場所を変更するためのもので、指定した場所でCGIを実行できるようにすることができます。

Directory

Directoryを利用すると、指定したディレクトリに対してのみ適用させる設定群を作成することができます。

既定ではそれぞれ /srv/www/htdocs、/usr/share/apache2/icons、/srv/www/cgi-binディレクトリに対する設定が存在しています。既定の設定から変更すべきではありません。

Include

Includeディレクティブを利用すると、追加の設定ファイルを取り込むように設定することができます。既定では2つのIncludeが設定されています:/etc/apache2/conf.d/は外部のモジュールからもたらされる設定を取り込むためのもので、これによりそのディレクトリ内に存在する.confで終わるファイルを全て取り込みます。2つめは/etc/apache2/conf.d/apache2-manual.confが設定されていて、apache2-manualの設定ファイルを取り込むようになっています。

サーバ名

ここではクライアントがWebサーバと通信を行なうにあたって、既定で使われるサーバ名を指定します。http://FQDN/でアクセスすることができるように完全修飾ドメイン名(FQDN)を指定するか、もしくはIPアドレスを指定します。ここでは任意の名前を入力することはできず、指定したサーバ名で名前を解決できなければなりません。

サーバ管理者のメールアドレス

サーバ管理者のメールアドレスを指定します。ここで指定したアドレスは、たとえばApacheがエラーページを生成した場合などに表示されます。

既定のホストの手順を終了したら、次へを押すと設定を続けることができます。

仮想ホスト

この段階では、ウィザードは設定済みの仮想ホストを表示します (詳しくは「仮想 (バーチャル) ホストの設定」(514 ページ) をご覧ください)。YaST HTTP ウィザードを開始するまでに特に何も設定していない場合は、何も仮想ホストが表示されません。

ホストを追加するには *追加* ボタンを押します。すると、作成するホストについて基本設定を行なうためのダイアログが表示されます。ここではサーバ名、サーバコンテンツのルート (ドキュメントルート)、管理者のメールアドレスなどを設定します。また、サーバ解決では、ホストの識別方法を指定します (名前ベースまたは IP ベース)。仮想ホスト *ID* の変更を押して、名前または IP アドレスを指定してください。

次へを押すと、次の仮想ホスト設定ダイアログに進みます。

仮想ホスト設定の 2 番目のダイアログでは、CGI を有効にするかどうかを設定することができるほか、これらのスクリプトをどのディレクトリに配置するかを設定することができます。また、SSL についても有効化を設定することができます。SSL を有効に設定する場合は、証明書のパスについても設定を行わなければなりません。SSL と証明書について、詳しくは 28.6.2 項「SSL を利用する Apache の設定」(548 ページ) をお読みください。またディレクトリインデックスオプションでは、クライアントがディレクトリへのアクセスを要求した場合に出力するファイル (既定では `index.html`) を指定します。ここには 1 つまたは複数のファイル (複数の場合は半角スペースで区切ります) を指定してください。また、公開 *HTML* を有効にするを選択すると、ユーザの公開ディレクトリ (~ユーザ名/`public_html/`) にあるファイルを `http://www.example.com/~ユーザ名` から公開することができるようになります。

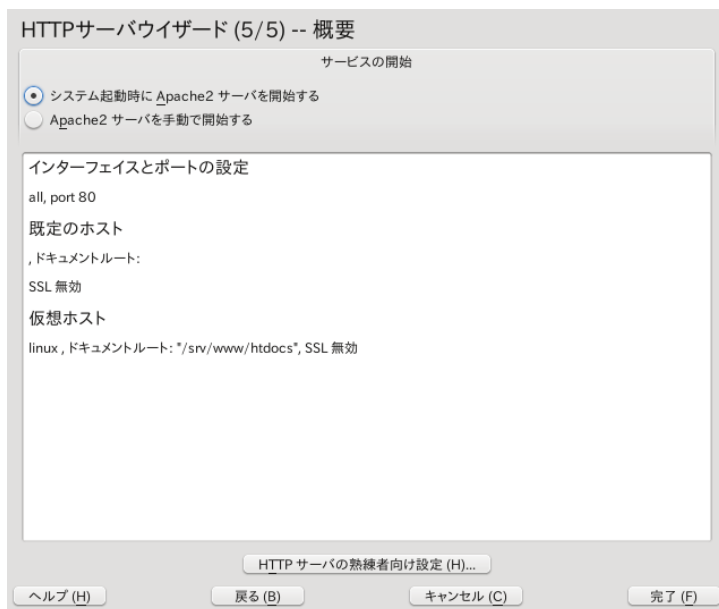
重要項目: 仮想ホストの作成

仮想ホストは好き勝手に追加することはできません。名前ベースの仮想ホストを利用している場合、ネットワーク上でそれぞれのホスト名を解決できなければなりません。IP ベースの仮想ホストを利用している場合は、それぞれ IP アドレスごとに 1 つの仮想ホストだけを割り当てることができます。

概要

これがウィザードの最終段階です。ここではApacheサーバの起動方法と起動タイミング(システム起動時、または手動)を設定することができます。またこれ以外にも、これまでに設定してきた項目の概要が表示されます。これまでの設定で問題がなければ、完了を押して設定を完了させてください。何らかの設定を変更しなおしたい場合は、戻るを押して必要なダイアログまで戻って設定し直してください。また、*HTTP*サーバの熟練者向け設定を押すと、「HTTPサーバ設定」(524ページ)に示されているダイアログが表示されるようになっています。

図 28.2 HTTP サーバウィザード: 概要



HTTP サーバ設定

*HTTP*サーバ設定ダイアログでは、ウィザードよりも細かい設定を行なうことができます(ウィザードはWebサーバの初回設定時にのみ表示されます)。このダイアログは下記に示す4つのタブから構成されていますが、いずれの設定ともすぐに反映されることはありません。設定を有効にするには、完

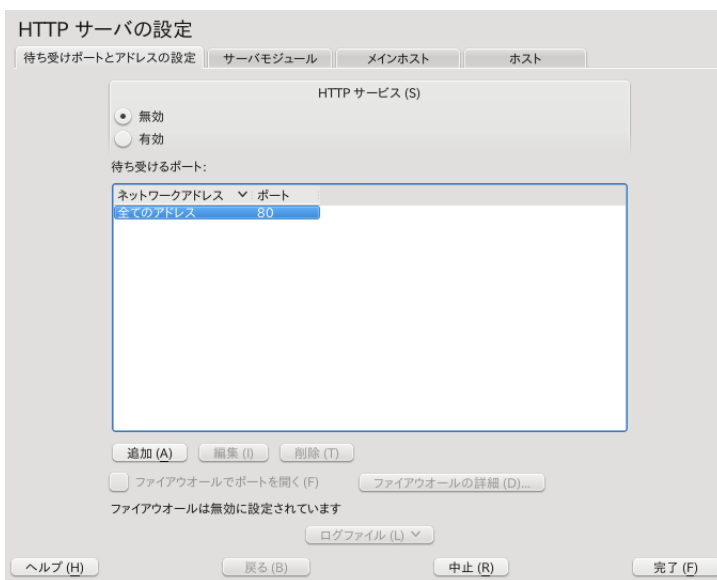
了を押して設定を確認しておかなければなりません。なお、**中止**を押すと設定モジュールを終了して変更内容を破棄することができます。

待ち受けポートとアドレス

HTTP サービスの枠内では、**Apache** を起動するか(有効)起動しないか((無効)を選択することができます。待ち受けるポートの枠内では、**Apache** サーバのサービスを提供させたいインターフェイスやポートをそれぞれ追加、編集、削除することができます。既定では全てのインターフェイスにてサービスを提供し、ポート 80 で待ち受けるように設定されています。また、**Web** サーバを他のホストからアクセスできるようにするため、ファイアウォールでポートを開くについてもお確認ください。ポートを閉じたままにする設定は、テストなどの目的で **Web** サーバに対して外部からのアクセスを拒否したい場合にのみ有用なものです。なお、お使いのマシンに複数のネットワークインターフェイスが存在する場合は、ファイアウォールの詳細...を押して、ポートを開くべきインターフェイスを設定することもできます。

さらにログファイルボタンでは、アクセスログやエラーログを参照することができます。これはお使いのサーバ設定をテストしたい場合などに便利です。ログファイルは個別のウインドウで表示され、そこから **Web** サーバの再起動や再読み込みなどを行なうことができます。詳しくは 28.3 項「**Apache** の起動と停止」(527 ページ)をお読みください。これらのコマンドはすぐに反映され、それらのログメッセージも即時に表示されます。

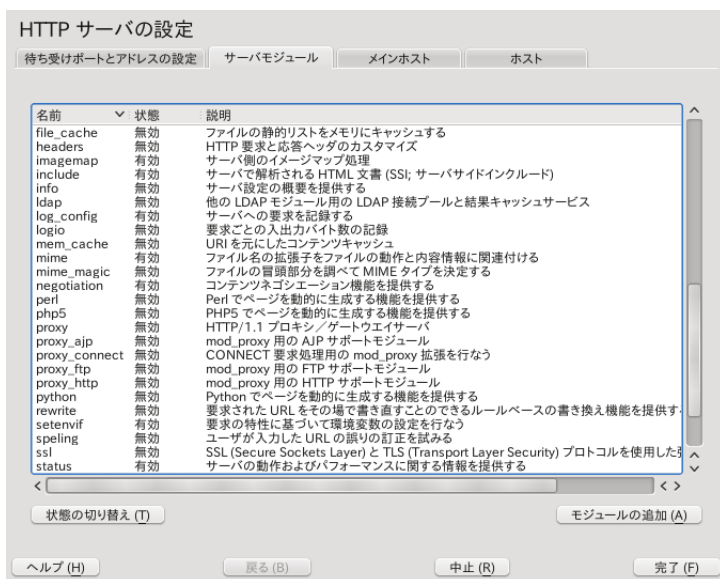
図 28.3 HTTP サーバ設定: 待ち受けポートとアドレス



サーバモジュール

それぞれモジュールを選んで 状態の切り替え ボタンを押すと、Apache2 モジュールの状態(有効/無効)を切り替えることができます。モジュールの追加ボタンを押すと、インストールされているものの一覧に表示されていない新しいモジュールを追加することができます。モジュールについて、詳しくは 28.4 項「モジュールのインストール/有効化/設定」(530 ページ)をお読みください。

図 28.4 HTTP サーバ設定: サーバモジュール



メインホストおよびホスト

これらのダイアログについては、それぞれ「既定のホスト」(520 ページ)と「仮想ホスト」(523 ページ)をお読みください。

28.3 Apache の起動と停止

28.2.3 項「YaST を利用した Apache の設定」(519 ページ)に書かれている手順で YaST から設定を行なうと、ランレベル 3 および 5 であれば Apache はシステム起動時に開始され、0, 1, 2, 6 であれば停止するようになっています。YaST のランレベルエディタを利用することで、この振る舞いを変えることができるほか、コマンドラインツール `chkconfig` を利用することでも同じことを行なうことができます。

起動中のシステムから Apache を開始したり停止したり、もしくは操作したりしたい場合は、`init` スクリプト `/usr/sbin/rcapache2` をご利用ください。なお、`init` スクリプトについての概要は、16.2.2 項「初期化スクリプト」

(265 ページ) をお読みください。rcapache2 コマンドは下記のようなパラメータを受け付けます:

status

Apache が起動しているかどうかを確認します。

start

Apache が起動していない場合、起動を行ないます。

startssl

Apache が起動していない場合、SSL を有効にして起動を行ないます。SSL の有効化について、詳しくは 28.6 項「SSL で通信の機密を保持する Web サーバの設定」(543 ページ) をお読みください。

stop

親プロセスを終了させることで、Apache を停止します。

restart

Apache を停止したあと、起動を行ないます。その時点で起動していなかった場合は、単に起動だけを行ないます。

try-restart

Apache が既に起動している場合にのみ、停止した後に起動を行ないます。

reload または graceful

Apache の全てのプロセスに対して、その時点で処理中の全てのリクエストを処理したあとに停止するよう働きかけます。それぞれのプロセスが終了すると、それらは新しく作成したプロセスに置き換えられ、最終的には Apache の「再起動」が行なわれるようにします。

ティップ: 本番環境での Apache 再起動

接続が強制的に切断されたりすることなく設定を反映するには、rcapache2 reload コマンドをご利用ください。

restart-graceful

全ての要求に対して応答することのできる 2 つめの Web サーバを起動します。従来まで存在していた 1 つめの Web サーバについては、GracefulShutdownTimeout で設定した時間間隔だけ 要求に対する処理を続け、最終的には停止するようにします。

`rcapache2 restart-graceful`は新しいバージョンへのアップグレードを行なった場合や、再起動を必要とする設定変更を行なった場合に便利な方法です。このオプションを使用すると、サービスの停止時間を最小限にすることができます。

`GracefulShutdownTimeout` の設定については必ず値を入力してください。設定を行なわないと `restart-graceful` は通常の `restart` と同じ意味になってしまいます。なお、この値に `0` を設定すると、残っている全てのリクエストを全て処理しきるまで、無期限に待機するようになります。

また `restart-graceful` は、必要な全ての資源を元の **Apache** プロセスが解放しない場合、失敗する場合があります。この場合、このコマンドは `stop-graceful` と同じ動作になります。

`stop-graceful`

既存の全てのリクエストが処理されるようにするため、`GracefulShutdownTimeout` で設定した時間間隔だけ 要求に対する処理を続け、最終的には停止します。

`GracefulShutdownTimeout` の設定については必ず値を入力してください。設定を行なわないと `stop-graceful` は通常の `restart` と同じ意味になってしまいます。なお、この値に `0` を設定すると、残っている全てのリクエストを全て処理しきるまで、無期限に待機するようになります。

`configtest` または `extreme-configtest`

現在実行中の **Web** サーバに影響を与えることなく、設定ファイルの文法を確認します。この文法確認はサーバを起動／再読み込み (**reload**)／再起動する際に毎回行なわれるようになっているため、テストだけを明示的に行なう必要はありません (設定エラーが見つかった場合は、**Web** サーバの起動／再読み込み (**reload**)／再起動は行なわれません)。また `extreme-configtest` オプションでは `nobody` ユーザで **Web** サーバを起動し、実際に設定の読み込みを行なってさらなるエラーを検出しようとします。ただし **SSL** の証明書については `nobody` で読み込むことができないため、**SSL** 関連の設定についてはテストを行なうことができません。

probe

再読み込みが必要かどうかを検出 (設定が変更されたかどうかを確認し)、rcapache2 コマンドに設定するパラメータを提案します。

server-status および full-server-status

それぞれ簡潔なサーバ状態か、もしくは完全なサーバ状態を表示します。

mod_status モジュールを有効化しておく必要があるほか、lynx または w3m をインストールする必要があります。また、/etc/sysconfig/apache2 ファイル内の APACHE_SERVER_FLAGS に、status を追加しなければなりません。

ティップ: 追加フラグ

rcapache2 に追加のフラグを指定すると、それらは Web サーバに渡されます。

28.4 モジュールのインストール／有効化／設定

Apache のソフトウェアはモジュールの形で部品化されています。いくつかの中枢機能を除き、それらの機能は全てモジュールとして提供されています。たとえば HTTP でさえもモジュール (http_core) の形になっています。

Apache のモジュールはそのソフトウェアをコンパイルする際に Apache に内蔵させることができるほか、実行時に動的に読み込むこともできます。どのようにして動的に読み込むのかについては、28.4.2 項「有効化と無効化」(531 ページ)をお読みください。

Apache のモジュールは下記の 4 つに分類できます:

基本モジュール

基本モジュールは既定で Apache に内蔵されます。openSUSE の Apache では mod_so (他のモジュールを読み込むために必要なモジュール) と http_core だけが内蔵されるようになっています。その他のモジュールは全て共有オブジェクトになっていて、サーバの実行ファイル自身に含まれることはなく、実行時に取り込むことで動作する仕組みになっています。

拡張モジュール

一般的に、拡張と呼ばれるモジュールについても Apache ソフトウェアパッケージ内に含まれていますが、通常はサーバに内蔵されることはありません。openSUSE では、それらは共有オブジェクトとして提供され、実行時に Apache に読み込むことができるようになっています。

外部モジュール

外部モジュールは公式の Apache 配布物には含まれていません。しかしながら、openSUSE ではいくつかの外部モジュールを提供しています。

マルチプロセッシングモジュール (MPM)

マルチプロセッシングモジュール (MPM) は Web サーバに対するリクエストを受け付けたり処理したりする責任を負ったソフトウェアで、Web サーバソフトウェアの中枢部を表わすものです。

28.4.1 モジュールのインストール

28.1.2 項「インストール」(508 ページ) に書かれている既定の手順で Apache をインストールした場合、全ての基本モジュールと拡張モジュールのほか、マルチプロセッシングモジュールである Prefork MPM と外部モジュール `mod_php5`、`mod_python` がそれぞれインストールされます。

YaST から ソフトウェア> ソフトウェア管理 を起動することで追加の外部モジュールをインストールすることができます。起動後は フィルタ> 検索 を利用し、*apache* と入力することで、利用可能な全ての外部モジュールを表示することができます。

28.4.2 有効化と無効化

それぞれのモジュールは YaST から有効にしたり無効にしたりすることができますほか、手作業でもこれを行なうことができます。YaST では、「HTTP サーバウイザード」(519 ページ) に書かれている手順で スクリプト言語モジュール (PHP5, Perl, Python) を有効にしたり無効にしたりすることができます。それ以外のモジュールについては、「サーバモジュール」(526 ページ) に書かれている手順で有効/無効を設定することができます。

モジュールを手作業で有効/無効に設定したい場合は、`a2enmod mod_foo` コマンドか、もしくは `a2dismod mod_foo` コマンドをご利用ください。ま

た、`a2enmod -l`を実行することで、有効に設定されている全モジュールの一覧を表示することができます。

重要項目: 外部モジュール向けの設定取り込み

外部モジュールを手作業で有効化した場合は、全ての仮想ホスト設定で外部モジュール向けの設定ファイルが読み込まれることを確認してください。外部モジュール向けの設定ファイルは `/etc/apache2/conf.d/` ディレクトリ内に置かれ、既定では読み込まれません。それぞれの仮想ホストで同じモジュールを必要とする場合は、このディレクトリ内にある `*.conf` ファイルを取り込む (**include**) こともできます。それ以外の場合は、個別に取り込んでもかまいません。設定例については、`/etc/apache2/vhost.d/vhost.template` ファイルをお読みください。

28.4.3 基本モジュールと拡張モジュール

全ての基本モジュールと拡張モジュールは、Apache のドキュメンテーション内で詳しく説明されています。最も重要なモジュールについては、概要だけが説明されています。各モジュールの詳しい説明は、<http://httpd.apache.org/docs/2.2/mod/> をお読みください。

`mod_actions`

特定の **MIME** タイプ (たとえば `application/pdf`) や特定の拡張子 (たとえば `.rpm`)、もしくは特定の リクエスト種別 (たとえば `GET`) が要求された際に、スクリプトを実行する手段を提供します。このモジュールは既定で有効に設定されます。

`mod_alias`

`Alias` および `Redirect` ディレクティブを提供するモジュールで、URI を特定のディレクトリに割り当てたり (`Alias`)、要求された URL を別の URL に転送したり することができます。このモジュールは既定で有効に設定されます。

`mod_auth*`

認証モジュールは、様々な認証方法を提供するためのモジュールです。`mod_auth_basic` では基本認証を、`mod_auth_digest` ではダイジェスト認証をそれぞれ提供します。Apache 2.2 におけるダイジェスト認証は、実験中のものとして提供されています。

なお、`mod_auth_basic` と `mod_auth_digest` の各モジュールは、それぞれ 認証プロバイダモジュール `mod_authn_*` と 認可モジュール `mod_authz_*` をあわせて使用する 必要があります。たとえば `mod_authn_file` はテキストファイルを利用した認証を提供するほか、`mod_authz_user` はユーザ認可を提供します。

認証や認可について、詳しくは <http://httpd.apache.org/docs/2.2/howto/auth.html> にある *Authentication HOWTO* をお読みください。

`mod_autoindex`

`autoindex` モジュールは、索引ファイル (たとえば `index.html` ファイル) が存在しない場合に、自動でディレクトリの索引を生成するためのモジュールです。外観については設定を変更することもできます。このモジュールは既定で有効に設定されていますが、`Options` ディレクティブで生成が無効に設定されています。ディレクトリ一覧を自動生成させるには、それぞれの仮想ホスト内で設定を上書きしてください。このモジュールの既定の設定値は、`/etc/apache2/mod_autoindex-defaults.conf` に書かれています。

`mod_cgi`

`mod_cgi` は CGI スクリプトを実行する際に必要なモジュールです。このモジュールは既定で有効に設定されています。

`mod_deflate`

このモジュールを利用することで、**Apache** はクライアントに特定の種類のファイルを転送する際、圧縮して転送することができるようになります。

`mod_dir`

`mod_dir` は、ディレクトリに対するアクセスがあった場合に、どのファイルを転送するかを選択する、`DirectoryIndex` ディレクティブを提供するモジュールです。既定では `index.html` が設定されています。このモジュールではほかにも、ディレクトリに対するアクセスがあった場合、URI への最後がスラッシュで終わっていないと、それを自動で補完する機能も備わっています。このモジュールは既定で有効に設定されています。

`mod_env`

CGI スクリプトや SSI のページに渡される、環境変数を制御するモジュールです。httpd プロセスから起動されるシェルに対し、環境変数を設定し

たり設定を消したり することができます。このモジュールは既定で有効に設定されています。

`mod_expires`

`mod_expires` を利用すると、`Expires` ヘッダを利用して期限切れ日時を設定することができます。期限切れ日時はプロキシサーバやブラウザが一時記憶 (キャッシュ) のために使用します。このモジュールは既定で有効に設定されています。

`mod_include`

`mod_include` はサーバサイドインクルード (Server Side Includes; SSI) を提供するモジュールで、HTML ページを動的に生成するための基本機能を提供します。このモジュールは既定で有効に設定されています。

`mod_info`

ブラウザから `http://localhost/server-info/` の URL にアクセスすることで、サーバ設定に関する広範囲の概要を提供するモジュールです。セキュリティ上の理由から、この URL へのアクセスは常に制限しておかなければなりません。既定では `localhost` だけがアクセスを許されるようになっています。`mod_info` は `/etc/apache2/mod_info.conf` で設定します。

`mod_log_config`

このモジュールを利用することで、Apache のログファイルについて設定を行なう ことができます。このモジュールは既定で有効に設定されています。

`mod_mime`

MIME モジュールは、そのファイル名の拡張子に基づいて正しい MIME ヘッダを 付与するモジュールです (たとえば HTML ドキュメントの場合は `text/html` に設定します)。このモジュールは既定で有効に設定されています。

`mod_negotiation`

コンテンツネゴシエーションと呼ばれる機能を提供するモジュールです。詳しくは <http://httpd.apache.org/docs/2.2/content-negotiation.html> をお読みください。このモジュールは既定で有効に設定されています。

mod_rewrite

mod_alias に似た機能を提供するモジュールですが、ずっと高機能で柔軟性に富んだ設定を行なうことができます。mod_rewrite を利用すると、複数のルールを設定し URL 転送を行なうことができるほか、リクエストヘッダを元にした転送なども行なうことができます。

mod_setenvif

クライアントのリクエスト内容に応じて環境変数を設定するモジュールです。たとえばクライアントが送信したブラウザ文字列のほか、クライアントの IP アドレスなどを基準にすることができます。このモジュールは既定で有効に設定されています。

mod_speling

mod_speling は URL の入力ミスやスペルミスについて、自動修正を試みるモジュールです。たとえば大文字と小文字の間違いなどを修正することができます。

mod_ssl

Web サーバとクライアントの間で、暗号化接続を行なうことができるようにするモジュールです。詳しくは 28.6 項「SSL で通信の機密を保持する Web サーバの設定」(543 ページ)をお読みください。このモジュールは既定で有効に設定されています。

mod_status

サーバの動作や性能情報を `http://localhost/server-status/` から公開することのできるモジュールです。セキュリティ上の理由から、この URL へのアクセスは常に制限しておかなければなりません。既定では localhost だけがアクセスを許されるようになっています。mod_status は `/etc/apache2/mod_status.conf` で設定します。

mod_suexec

mod_suexec は、CGI スクリプトを特定のユーザやグループで実行させるためのモジュールです。このモジュールは既定で有効に設定されています。

mod_userdir

`~ユーザ名/` のディレクトリから、ユーザ固有のディレクトリを提供するモジュールです。設定ファイル内で UserDir ディレクティブを指定しなければなりません。このモジュールは既定で有効に設定されています。

28.4.4 マルチプロセッシングモジュール (MPM)

openSUSE では、Apache 向けに 2 種類のマルチプロセッシングモジュール (MPM) を提供しています:

- Prefork MPM (536 ページ)
- 「Worker MPM」 (536 ページ)

Prefork MPM

Prefork MPM はスレッドを使用しない MPM で、あらかじめ fork しておくことで動作する Web サーバを提供します。Web サーバの動きは Apache バージョン 1.x に似たものとなり、各リクエストは独立して動作する子プロセスが処理する形になります。そのため、問題のあるリクエストが届いた場合であっても他のプロセスの動作を止めたりすることなく、Web サービスが停止してしまったりすることを防ぐことができます。

このようにプロセスを基準にしたアプローチは安定性を提供する一方、欠点として worker MPM と比べ多くの資源を必要としてしまいます。Prefork MPM は Unix ベースのオペレーティングシステムでは既定の MPM として採用されています。

重要項目: このドキュメント内での MPM

この文書内では、Apache は Prefork MPM を使用した場合を想定しています。

Worker MPM

Worker MPM はスレッドを利用する Web サーバです。スレッドとはプロセスよりも「軽量の」仕組みで、プロセスよりも少ない資源で動作するものです。Worker MPM では子プロセスを fork するだけでなく、そのプロセス内で複数のスレッドを起動してリクエストを処理します。このアプローチにより、Prefork MPM よりも少ない資源で高い性能を発揮するような仕組みになっています。

Worker MPM の最大の欠点はその安定性で、ある特定のスレッドが何らかの理由で破壊されてしまうと、そのスレッドを管理していたプロセス全体が影響を受けてしまいます。最悪の場合、サーバクラッシュにもつながる可能性があります。特に Apache で Common Gateway Interface (CGI) を利用していて負荷が高くなると、システム資源との通信が行なえなくなることで Internal Server Error が発生してしまいます。また Worker MPM は、Apache に添付されているモジュールの一部がスレッドセーフ (複数のスレッドから同時に呼び出されても安全に動作すること) ではないため、それらは Worker MPM と同時に使用することができません。

警告: PHP を利用する場合の MPM について

PHP モジュールの一部はスレッドセーフではありません。mod_php を使用する場合は、Worker MPM を使用しないことを強くお勧めします。

28.4.5 外部モジュール

openSUSE に同梱されている全ての外部モジュールを下記に列挙しています。各モジュールのドキュメンテーションについては、それぞれのディレクトリをご覧ください。

mod_apparmor

mod_php5 や mod_perl などのモジュールで処理される Apache 内の各 CGI スクリプトに対し、Novell AppArmor の制限を設定します。

パッケージ名: apache2-mod_apparmor

さらなる情報: パート 「Novell AppArmor を利用した権利制限」 (↑セキュリティガイド)

mod_mono

mod_mono を利用すると、お使いのサーバ内で ASP.NET ページを実行することができるようになります。

パッケージ名: apache2-mod_mono

設定ファイル: /etc/apache2/conf.d/mod_mono.conf

mod_perl

mod_perl を利用すると、Perl スクリプトを内蔵のインタプリタで実行することができるようになります。サーバ内でインタプリタが恒久的に

保持される形になるため、外部のインタプリタを起動する場合と比べて起動時のオーバーヘッドを小さくすることができます。

パッケージ名: `apache2-mod_perl`

設定ファイル: `/etc/apache2/conf.d/mod_perl.conf`

さらなる情報: `/usr/share/doc/packages/apache2-mod_perl`

`mod_php5`

PHP はサーバ内で動作するプラットフォーム非依存の HTML 内蔵型 スクリプト言語です。

パッケージ名: `apache2-mod_php5`

設定ファイル: `/etc/apache2/conf.d/php5.conf`

さらなる情報: `/usr/share/doc/packages/apache2-mod_php5`

`mod_python`

`mod_python` は Apache HTTP サーバ内に Python 機能を内蔵させるためのもので、性能を相当に改善することができるほか、Web ベースのアプリケーションを構築する際の柔軟性を追加することができます。

パッケージ名: `apache2-mod_python`

さらなる情報: `/usr/share/doc/packages/apache2-mod_python`

`mod_tidy`

`mod_tidy` は TidyLib を利用するモジュールで、それぞれ発信される HTML ページを検証することができるものです。検証エラーが発生すると、エラー一覧ページが表示されます。エラーにならなかった場合は元の HTML ページが表示されます。

パッケージ名: `apache2-mod_tidy`

設定ファイル: `/etc/apache2/mod_tidy.conf`

さらなる情報: `/usr/share/doc/packages/apache2-mod_tidy`

28.4.6 コンパイル

Apache では知識のあるユーザがカスタムなモジュールを作成することで、その機能を拡張することができます。Apache のモジュールを開発したりサードパーティ製のモジュールをコンパイルしたりしたい場合は、開発ツールの

ほかに `apache2-devel` パッケージが必要となります。 `apache2-devel` パッケージには `apxs2` と呼ばれるツールが含まれ、これを利用することで Apache 向けの追加モジュールをコンパイルできるようになっています。

`apxs2` はソースコードからモジュールをコンパイルしたりインストールしたりすることができます (インストール作業では、設定ファイルに対して行なう変更も同時に行なうことができます)。これにより、Apache の実行時に読み込まれる **動的共有ライブラリ (DSO)** を作成することができます。

`apxs2` 関連のバイナリは、`/usr/sbin` 以下に配置されます:

- `/usr/sbin/apxs2`—任意の **MPM** で動作する拡張モジュールを構築するのに便利なツールです。インストールを行なう際、インストール先は `/usr/lib/apache2` になっています。
- `/usr/sbin/apxs2-prefork`—**prefork MPM** 用のモジュールを構築するのに便利なツールです。インストールを行なう際、インストール先は `/usr/lib/apache2-prefork` になっています。
- `/usr/sbin/apxs2-worker`—**worker MPM** 用のモジュールを構築するのに便利なツールです。インストールを行なう際、インストール先は `/usr/lib/apache2-worker` になっています。

ソースコードからモジュールをコンパイルし、インストールしたあとに有効に設定するには、下記のように入力します:

```
cd (ソースコードのパス); apxs2 -cia  
mod_foo.c
```

ここで、`-c` オプションはモジュールのコンパイルを、`-i` オプションはモジュールのインストールを、`-a` オプションはモジュールの有効化をそれぞれ指定しています。 `apxs2` に対するその他のオプションは、`apxs2(1)` のマニュアルページ内に記述されています。

28.5 CGI スクリプトを動作させる方法

Apache が用意している汎用ゲートウェイインターフェイス (**Common Gateway Interface; CGI**) は、プログラムやスクリプトから動的なコンテンツを生成するための仕組みで、CGI スクリプトなどとも呼ばれるものです。CGI スクリプ

トでは、任意のプログラミング言語を使用することができます。通常はスクリプト言語として、Perl や PHP などを使用します。

Apache に対して CGI スクリプトの実行と動的コンテンツの生成を許可するには、mod_cgi モジュールと mod_alias モジュールを有効に設定する必要があります。両方とも既定では有効に設定されるモジュールです。モジュールの有効化について、詳しくは 28.4.2 項「有効化と無効化」(531 ページ)をお読みください。

警告: CGI セキュリティ

サーバに対して CGI スクリプトの実行を許可すると、それは潜在的なセキュリティホールになる場合があります。詳しくは 28.7 項「セキュリティ問題の回避」(550 ページ)をお読みください。

28.5.1 Apache 設定

openSUSE では、CGI スクリプトの実行は /srv/www/cgi-bin/ ディレクトリ内でのみ許可されるようになっています。このディレクトリは CGI スクリプトを実行するために設定済みのディレクトリですが、既に仮想ホストの設定を行なっている場合(「仮想(バーチャル)ホストの設定」(514 ページ)を参照)で、仮想ホスト別のディレクトリに CGI スクリプトを配置したい場合は、下記のようにしてディレクトリを設定しなければなりません:

例 28.5 仮想ホストでの CGI 設定

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/"❶
```

```
<Directory "/srv/www/www.example.com/cgi-bin/">  
Options +ExecCGI❷  
AddHandler cgi-script .cgi .pl❸  
Order allow,deny❹  
Allow from all  
</Directory>
```

- ❶ このように設定することで、上記のディレクトリ内にある各ファイルを CGI と見なし、実行するようになります。
- ❷ CGI スクリプトの実行を有効に設定します。
- ❸ .pl と .cgi の拡張子をもつファイルに対し、CGI スクリプトとして実行するように指定しています。この行は必要に応じて変更してください。
- ❹ Order と Allow の各ディレクティブは既定のアクセス制御を設定し、Allow と Deny のどちらを先に解釈するのかを指定します。この設定の場合、「deny」ディレクティブが「allow」ディレクティブよりも先に解釈され、既定でアクセスは許可されるようになります。

28.5.2 サンプルスクリプトの実行

CGI のプログラミングは「通常の」プログラミングとは異なり、Content-type: text/html のような MIME 種類のヘッダを出力しなければなりません。このヘッダはクライアントに送信され、クライアント側で解釈されます。また、CGI スクリプトの出力はクライアント (一般に Web ブラウザ) が解釈可能な形式でなければなりません。多くの場合は HTML やテキスト形式ですが、たとえば画像などでもかまいません。

Apache のパッケージには、シンプルなテスト用スクリプト /usr/share/doc/packages/apache2/test-cgi が含まれています。これはいくつかの環境変数をテキスト形式で出力するものです。このファイルは /srv/www/cgi-bin/ ディレクトリにコピーするか、もしくはお使いの仮想ホスト内のスクリプトディレクトリ (/srv/www/www.example.com/cgi-bin/) にコピーし、test.cgi のようなファイル名を設定してください。

また、Web サーバがアクセスするファイルは、root ユーザに所有権があるファイルであるべきものです。詳しい情報は 28.7 項「セキュリティ問題の回

避」(550 ページ)をお読みください。Web サーバは `root` とは異なるユーザで動作することから、CGI スクリプトは全てのユーザに対して実行と読み込みを許可しなければなりません。CGI のディレクトリに移動して `chmod 755 test.cgi` コマンドを実行すると、必要なパーミッションを設定することができます。

全ての設定が完了したら、`http://localhost/cgi-bin/test.cgi` にアクセスするか、もしくは `http://www.example.com/cgi-bin/test.cgi` にアクセスしてください。アクセスすると「CGI/1.0 test script report」のような表示が現われるはずです。

28.5.3 CGI のトラブルシューティング

テストプログラムの出力が現われず、エラーメッセージが表示された場合は、下記の項目を確認してください：

CGI トラブルシューティング

- ・ 設定を変更したあと、再起動を行ないましたか？ `rcapache2 probe` を実行して確認してください。
- ・ カスタムな CGI ディレクトリを設定している場合、それらを正しく設定してありますか？ 不確かな場合は、テスト用のスクリプトを既定の CGI ディレクトリ `/srv/www/cgi-bin/` にコピーし、ブラウザで `http://localhost/cgi-bin/test.cgi` を開いてみてください。
- ・ ファイルのパーミッションは正しく設定されていますか？ CGI をコピーしたディレクトリに移動し、`ls -l test.cgi` を実行してください。パーミッションは、下記のように表示されなければなりません：

```
-rwxr-xr-x  1 root root
```

- ・ スクリプトにプログラムエラーが存在しないことを確認してください。`test.cgi` をコピーしただけで変更を行っていない場合は起こりませんが、独自のプログラムを作成したり修正したりした場合は、プログラムエラーが無いかどうかを確認してください。

28.6 SSL で通信の機密を保持する Web サーバの設定

クレジットカード情報などの機密データは、Web サーバとクライアントの間で機密を保持する目的で暗号化を行ない、認証を設定しておくことが望まれます。mod_ssl では、Secure Sockets Layer (SSL) や Transport Layer Security (TLS) プロトコルを利用し、Web サーバとクライアント間の HTTP 通信に強い暗号化を提供します。SSL や TLS を利用することで、Web サーバとクライアントの間で秘密の通信を確立することができます。データの整合性についても確認が行なわれるほか、サーバとクライアントの間で相互の認証を行なうことができます。

この暗号化通信を成立させるため、サーバは URL を受け付ける前に、自分自身の正当性を証明する SSL 証明書を送信します。これによりクライアントは、サーバが偽装されたものではなく、通信を開始してよいものかどうかを確認することができます。また証明書は、機密データを危険にさらすリスクを負うことのないようにサーバとクライアントの間で情報をやりとりするための、暗号化接続を生成する機能を備えています。

mod_ssl 自身では SSL/TLS プロトコルを実装することはしておらず、Apache と SSL ライブラリの間のインターフェイスとして動作する仕組みになっています。openSUSE では、OpenSSL ライブラリを利用しています。Apache をインストールすると、OpenSSL も自動でインストールされます。

Apache で mod_ssl を利用した場合、URL が http:// ではなく https:// になるという点が最もよくわかる違いです。

ティップ: 証明書のサンプル

apache2-example-certificates パッケージをインストールすると、疑似企業「Snake Oil」用の証明書が利用できるようになります。

28.6.1 SSL 証明書の作成

Web サーバで SSL/TLS を使用するためには、SSL の証明書を作成する必要があります。この証明書は Web サーバとクライアントの間で認証を行なうために必要なもので、個人や企業同士で相互に所在確認を行なうことができます。

証明書の正当性を確かなものにするには、多くのユーザが信頼する企業や団体によって電子署名されたものでなければなりません。

作成可能な証明書としては、下記の3種類があります。テスト用の「ダミー」証明書、公的な信頼を必要としない特定のユーザ範囲で利用する自己署名証明書、独立した公的認証機関 (CA) が電子署名する証明書の3種類です。

証明書の作成は、基本的に2段階の作業で行ないます。1つめは認証機関向けの機密鍵作成、2つめはその鍵を利用したサーバ証明書への署名です。

ティップ: さらなる情報

SSL/TSL について、考え方や定義をより詳しく知るには、http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html をお読みください。

「Dummy」 証明書の作成

dummy 証明書の作成は簡単に行なうことができます。単純に `/usr/bin/gensslcert` スクリプトを実行するだけです。これにより下記に示すファイルを生成するか、上書きします。gensslcert に指定するパラメータを変えることで、作成する証明書を細かく調整することができます。詳しくは `/usr/bin/gensslcert -h` を実行して表示されるヘルプをお読みください。

- `/etc/apache2/ssl.crt/ca.crt`
- `/etc/apache2/ssl.crt/server.crt`
- `/etc/apache2/ssl.key/server.key`
- `/etc/apache2/ssl.csr/server.csr`
- `/root/.mkcert.cfg`

ダウンロード用に `ca.crt` のコピーが `/srv/www/htdocs/CA.crt` に作成されます。

重要項目: テスト目的にのみご利用ください

Dummy 証明書は本番環境では使用しないでください。テスト目的でのみお使いください。

自己署名証明書の作成

イントラネット内の機密 Web サーバを構築する場合や、特定のユーザに対する機密 Web サーバを構築する場合は、ご自身で証明機関 (CA) を作成し、そこで署名を行えば十分です。

自己署名証明書を作成するには、対話処理で 9 段階の作業を行ないます。まずは /usr/share/doc/packages/apache2 ディレクトリに移動し、下記のコマンドを実行します: `./mkcert.sh make --no-print-directory /usr/bin/openssl /usr/sbin/ custom`。なお、このコマンドは上記のディレクトリ以外からは実行しないでください。このプログラムは複数の問い合わせを表示する仕組みで動作し、それぞれに答えていくことによって証明書を作成します。

手順 28.4 *mkcert.sh* を利用した自己署名証明書の作成

1 `Decide the signature algorithm used for certificates`

古いブラウザでは DSA を利用したときに問題が発生する可能性があるため、RSA を選択 (R, 既定値) してください。

2 `Generating RSA private key for CA (1024 bit)`

特に何も操作する必要はありません。

3 `Generating X.509 certificate signing request for CA`

ここから証明機関の識別名を作成します。それぞれ国名や団体名など、いくつかの項目に回答します (訳注: スクリプトの仕様上、英語表記であることにご注意ください)。ここで入力するデータはそれぞれ証明書に記載されるため、正しいデータを入力してください。また、全ての質問に対して何らかの入力を行なう必要はなく、該当しないものや入力しないままにしておきたい場合は、「。」を入力してください。Common name (共通名) は

証明機関 自身の名前を入力します。 *My companyCA* などのように、わかりやすい名前を入力してください。

重要項目: 証明機関の共通名について

証明機関の共通名はサーバの共通名とは異なるものでなければなりません。そのため、この段階では完全修飾ドメイン名は入力しないでください。

4 Generating X.509 certificate for CA signed by itself

証明書のバージョンを選択します。 **3** (既定値) を選んで ください。

5 Generating RSA private key for SERVER (1024 bit)

特に何も操作する必要はありません。

6 Generating X.509 certificate signing request for SERVER

サーバ用の鍵を作成するための識別名を入力します。証明機関の識別名を入力したばあいとほとんど同じ質問内容で、それぞれ **Web** サーバに関する情報を入力してください。なお、証明機関のデータと同じものである必要はありません (たとえばサーバが別の場所に存在する場合など)。

重要項目: 共通名の選択

ここで入力する共通名は、サーバの完全修飾ドメイン名でなければなりません (たとえば **www.example.com**)。そうでないとブラウザから **Web** サーバにアクセスした際、証明書に記載されている共通名とサーバの完全修飾ドメイン名が異なるとして、警告が発生してしまいます。

7 Generating X.509 certificate signed by own CA

証明書のバージョンを選択します。 **3** (既定値) を選んで ください。

8 Encrypting RSA private key of CA with a pass phrase for security

証明機関の機密鍵については、暗号化しておくことを強くお勧めします。そのため、ここでは **Y** と回答してパスワードを入力してください。

9 Encrypting RSA private key of SERVER with a pass phrase for security

サーバの鍵を暗号化すると、Webサーバの起動時に毎回パスワードの入力を求められるようになります。これはシステム起動時にサーバを自動で開始することができなくなるほか、Webサーバの再起動にも手間がかかることになります。そのため、一般的にこの質問に対してはNと回答してください。ただしパスワードで暗号化しない場合、その鍵は自分自身を保護する術を持たないことになるので、その鍵の利用を許すユーザだけがアクセスできることを確認してください。

重要項目: サーバ鍵の暗号化

サーバの暗号鍵をパスワードで保護した場合は、`/etc/sysconfig/apache2` ファイル内の `APACHE_TIMEOUT` 設定の値を増やしてください。増やさない場合はパスワードを入力するための十分な時間が確保されず、サーバが意図せず停止してしまう可能性があります。

スクリプトでの入力が完了すると、生成された証明書と鍵の一覧が表示されます。スクリプトの出力とは異なり、ファイルはローカルディレクトリである `conf` ではなく、`/etc/apache2/` ディレクトリに作成されます。

最後にやるべきことは、証明機関の証明書ファイルを `/etc/apache2/ssl.crt/ca.crt` からご利用のユーザがアクセスできる場所にコピーし、各ブラウザに信頼のおける証明機関として登録してもらうことです。各ブラウザで登録を行なわないと、ブラウザからアクセスを行なった際に、未知の発信元として警告が表示されてしまいます。なお、証明書は1年間有効です。

重要項目: 自己署名証明書

自己署名証明書は、その証明書を信頼してもらうことのできる特定のユーザだけがアクセスする Webサーバを構築する際にのみ、お使いください。たとえば一般的な店舗などのような、不特定多数がアクセスするサーバには不適切です。

公式に署名された証明書の取得

お使いの証明書に対して署名を行なうことのできる、公的な証明機関がいくつか存在しています。証明書は信頼のできる第三者によって署名されるため、

それらの証明書を完全に信頼できるようになります。不特定多数がアクセスし、機密を保持する Web サーバでは、一般に公的な署名のある証明書を利用します。

最もよく知られた証明機関としては、Thawte (<http://www.thawte.com/>) や Verisign (<http://www.verisign.com>) などがあります。これらを含む公的な証明機関はあらかじめ全てのブラウザに登録されているため、これらの証明機関が署名した証明書は、ブラウザ側で自動的に受け付けられます。

公的に署名された証明書を取得する場合は、その証明書そのものを証明機関に送付する必要はありません。その代わりに、証明書署名要求 (CSR) を送信します。CSR を作成するには、`/usr/share/ssl/misc/CA.sh -newreq` コマンドを実行してください。

まず上記のスクリプトは、CSR を暗号化する際のパスワードを尋ねます。そのあと識別名の入力を行ないます。それぞれ国名や団体名など、いくつかの項目に回答します (訳注: スクリプトの仕様上、英語表記であることにご注意ください)。ここで入力するデータはそれぞれ証明書に記載されるため、正しいデータを入力してください。また、全ての質問に対して何らかの入力を行なう必要はなく、該当しないものや入力しないままにしておきたい場合は、「」を入力してください。Common name (共通名) は証明機関 自身の名前を入力します。My company CA などのように、わかりやすい名前を入力してください。最後にチャレンジパスワードと呼ばれるパスワードの入力と、代替の企業名をそれぞれ入力します。

作成された CSR は、スクリプトを実行したときのディレクトリ内に配置されます。ファイル名は `newreq.pem` になっています。

28.6.2 SSL を利用する Apache の設定

Web サーバで SSL と TLS を利用する場合、既定のサーバ側のポートは 443 です。「通常の」設定を Apache に行なっていれば、ポート 80 での http とポート 443 での SSL/TLS (https) という形になるため、特に矛盾が発生することはありません。実際、単一の Apache が起動している状態で、HTTP と HTTPS の両方に対応することができます。通常はそれぞれポート 80 とポート 443 で別々の仮想ホストを設定し、サービスをまかさないです。

重要項目: ファイアウォール設定

Apache で SSL をポート 443 で有効化する場合、ファイアウォールを開くことも忘れずに実施してください。これは 項「**Configuring the Firewall with YaST**」(第14章 *Masquerading and Firewalls*, ↑セキュリティガイド) に書かれている手順で YaST から設定することができます。

サーバ全体の設定として、SSL モジュールは既定で有効化されています。お使いの仮想ホストで無効化している場合は、`a2enmod ssl` コマンドで有効化することができます。最終的に SSL を有効にするには、サーバを起動する際に「SSL」フラグを付けて起動する必要があります。これを行なうには、`a2enflag SSL` を実行してください。このとき、サーバの証明書をパスワードで暗号化するようにしている場合は、`/etc/sysconfig/apache2` ファイル内の `APACHE_TIMEOUT` の値を増やしておく必要があります。これで Apache を起動する際にパスワード入力を行なうのに十分な時間を稼ぐことができます。この変更を行なう場合は、サーバを再起動してください。再読み込みでは不十分です。

なお、仮想ホストのディレクトリ内のファイル `/etc/apache2/vhosts.d/vhost-ssl.template` では、SSL 固有のディレクティブについて詳しい説明が書かれています。一般的な仮想ホストの設定については、「仮想(バーチャル)ホストの設定」(514 ページ)をお読みください。

仮想ホストの設定を始めるには、上記のテンプレートを `/etc/apache2/vhosts.d/mySSL-host.conf` ファイルなどにコピーしてから編集してください。それぞれ下記のディレクティブを調整してください:

- `DocumentRoot`
- `ServerName`
- `ServerAdmin`
- `ErrorLog`
- `TransferLog`

重要項目: 名前ベースの仮想ホストと SSL

1 つの IP アドレスしか設定されていないサーバでは、SSL 対応の仮想ホストを複数作成することはできません。SSL 対応の仮想ホストを複数設定した場合にユーザから接続を行なうと、証明書には 1 つのサーバ名しか記述できないため、いずれか 1 つの仮想ホストを除き、証明書のサーバ名と実際のサーバ名とが異なる として、アクセスのたびに警告が表示されてしまいます。これを解決するには、それぞれ SSL 対応の仮想ホストに対し、異なる IP アドレスかポート番号を設定してください。これにより証明書のサーバ名と実際のサーバ名を一致させることができますようになります。

28.7 セキュリティ問題の回避

公的なインターネットに晒された Web サーバは、継続的な管理を行なう必要があります。セキュリティの問題が発生することは避けられない問題で、ソフトウェアに関連した問題だけでなく、不用意な設定ミスが発生する場合があります。ここでは、このような問題について、どのように取り扱うべきかを示しています。

28.7.1 最新のソフトウェア

Apache ソフトウェアに脆弱性が発生された場合は、SUSE からセキュリティアドバイザリ (助言／忠告の意味) が発信されます。このアドバイザリには、その脆弱性を修正するための手順が含まれていて、できるかぎり素早い適用を行なう必要があります。SUSE のセキュリティアドバイザリは、下記の方法で受け取ることができます (訳注: 全て英語によるものです):

- **Web ページ** <http://www.novell.com/linux/security/securitysupport.html>
- **メーリングリスト** <http://ja.opensuse.org/%E3%83%A1%E3%83%BC%E3%83%AA%E3%83%B3%E3%82%B0%E3%83%AA%E3%82%B9%E3%83%88> (日本語) または http://en.opensuse.org/openSUSE:Support_channels (英語)

- RSS フィード http://www.novell.com/linux/security/suse_security.xml

28.7.2 DocumentRoot のパーミッション

openSUSE の既定では、DocumentRoot のディレクトリ `/srv/www/htdocs` と CGI のディレクトリ `/srv/www/cgi-bin` はそれぞれユーザとグループが `root` に設定されています。これらのパーミッション 設定は変更すべきではありません。ディレクトリが書き込み可能であったりすると、 任意のユーザからファイルを配置できるようになってしまいます。これらのファイルは Apache から `wwwrun` の権限下で実行され、ファイルシステムの資源に対して望まないアクセスを発生させる結果になってしまいます。 仮想ホストの DocumentRoot と CGI の配置先としては `/srv/www` のサブディレクトリを使用するものとし、 それらのファイルやディレクトリが `root` ユーザおよびグループに属しているように設定してください。

28.7.3 ファイルシステムのアクセス

既定ではファイルシステム全体へのアクセスが禁止されるよう、 `/etc/apache2/httpd.conf` に設定が為されています。これらのディレクティブを変更するのではなく、 Apache から全てのディレクトリにアクセスし、必要なファイルを読み込むことができるように設定を追加してください。詳しくは「基本的な仮想ホスト設定」(518 ページ)をお読みください。これを行なうことで、たとえばパスワードやシステムの設定ファイルなど、不必要なファイルへのアクセスを禁止し、外部から読み取られないようにすることができます。

28.7.4 CGI スクリプト

Perl, PHP, SSI やその他のプログラミング言語で対話的に動作するスクリプトは、結果的に任意のコマンドを実行できてしまい、広い範囲のセキュリティ問題につながります。また、サーバから実行されるスクリプトは、サーバ管理者が信頼する発信元から公開されたソースコードを利用してインストールしなければなりません。ユーザに対して独自のスクリプトを実行させてしまうことは、一般に良い方法ではありません。全てのスクリプトに対し、セキュリティ監査を実施しておくことをお勧めします。

スクリプトの管理をできる限り簡単にするため、システム全体で CGI の実行を許可したりすることをせず、専用のディレクトリ内でのみ実行できるようにするのが一般的です。それぞれ `ScriptAlias` や `Option ExecCGI` のディレクティブを設定してください。openSUSE の既定の設定では、任意の場所からの CGI 実行は許可しない設定になっています。

全ての CGI スクリプトは同じユーザで動作するため、異なるスクリプト同士が互いに衝突してしまう場合もあります。このような場合は `suEXEC` を利用し、それぞれ異なるユーザやグループで CGI を実行するように設定してください。

28.7.5 ユーザディレクトリ

ユーザディレクトリを許可する場合 (`mod_userdir` または `mod_rewrite`) は、`.htaccess` の使用を禁止することを強くお勧めします。それは、このファイルがセキュリティの設定を上書きできてしまうものであるためです。少なくとも `AllowOverride` を利用して、ユーザ側で設定できることを制限してください。openSUSE では `.htaccess` が既定で有効に設定されていますが、`mod_userdir` を利用した場合、ユーザに対しては `Option` ディレクティブでの上書きは行なえないようになっています (詳しくは設定ファイル `/etc/apache2/mod_userdir.conf` をお読みください)。

28.8 トラブルシューティング

Apache がうまく起動しない場合や Web ページにアクセスできない場合、もしくはユーザが Web サーバに接続できない場合は、問題の原因を探ることが重要です。ここではよくエラーが発生する場所や、確認しておくべき場所を示します:

`rcapache2` の出力

Web サーバの起動や停止を行なう場合、`/usr/sbin/httpd2` のバイナリではなく `rcapache2` スクリプトをお使いください (詳しくは 28.3 項「Apache の起動と停止」(527 ページ)をお読みください)。このスクリプトではより詳しい出力を行なうようになっているため、設定エラーの問題を解決する糸口を見つけやすくなります。

ログファイルとログの詳しさ

致命的なエラーであれそれ以外のエラーであれ、原因を探るには Apache のログファイルを利用するのがよいでしょう。既定では `/var/log/apache2/error_log` に出力されるエラーログファイルが主に役立つことでしょう。また、ログファイル内により詳しい出力を行ないたい場合は、`LogLevel` ディレクティブを利用してログメッセージの詳しさを設定してください。

ティップ: 単純なテスト

Apache のログメッセージを監視するには、`tail -F /var/log/apache2/my_error_log` コマンドを実行するのがよいでしょう。これで監視している状態から、`rcapache2 restart` を実行し、ブラウザで接続するとログの出力を確認することができます。

ファイアウォールとポート

一般的に良くある間違いとしては、Apache 向けのポートをファイアウォール設定で開いていない場合があります。YaST で Apache を設定した場合、この問題に対応するための個別オプションが提供されています (28.2.3 項「YaST を利用した Apache の設定」(519 ページ)をお読みください)。Apache を手作業で設定している場合、HTTP や HTTPS のポートを開くには、YaST のファイアウォールモジュールをご利用ください。

上記のようなやり方を行なっても原因がよくわからない場合は、オンラインの Apache バグデータベース (http://httpd.apache.org/bug_report.html) (英語のみ) をご覧ください。また、Apache のメーリングリストを利用してユーザコミュニティに質問することもできます (詳しくは <http://httpd.apache.org/userslist.html> (英語) をお読みください)。それ以外にも、ニュースグループを利用することもできます (<comp.infosystems.www.servers.unix>) (英語)。

28.9 さらなる情報

apache2-doc パッケージでは、ローカルにインストールされる各国語対応の Apache マニュアルやリファレンスが提供されています。このパッケージは既定でインストールされませんので、最も手っ取り早くインストールしたい場合は `zypper in apache2-doc` コマンドでインストールを行なっくだ

さい。インストールを行なったら、Apache のマニュアルが `url="http://localhost/manual/"` からアクセスできるようになります。また、同マニュアルのオンライン版もご利用いただけます (<http://httpd.apache.org/docs-2.2/>)。SUSE 固有の設定ヒントなどは、`/usr/share/doc/packages/apache2/README.*` ディレクトリにあります。

28.9.1 Apache 2.2

Apache 2.2 での新機能の一覧については、http://httpd.apache.org/docs/2.2/new_features_2_2.html をお読みください。また、バージョン 2.0 から 2.2 にアップグレードする際の情報は、<http://httpd.apache.org/docs-2.2/upgrading.html> をお読みください。

28.9.2 Apache モジュール

28.4.5 項「外部モジュール」(537 ページ)に概要が書かれている Apache の外部モジュールについて、詳しい情報はそれぞれ下記の場所にあります:

`mod-apparmor`

<http://en.opensuse.org/SDB:AppArmor>

`mod_mono`

http://www.mono-project.com/Mod_mono

`mod_perl`

<http://perl.apache.org/>

`mod_php5`

<http://www.php.net/manual/en/install.unix.apache2.php>

`mod_python`

<http://www.modpython.org/>

`mod_tidy`

<http://mod-tidy.sourceforge.net/>

28.9.3 開発

Apache のモジュールを開発したり、Apache の Web サーバプロジェクトに参加したりするための情報については、それぞれ下記の URL をご覧ください (いずれも英語が主体です):

Apache 開発者情報

<http://httpd.apache.org/dev/>

Apache 開発者ドキュメンテーション

<http://httpd.apache.org/docs/2.2/developer/>

Perl と C を利用した Apache モジュールの書き方

<http://www.modperl.com/>

28.9.4 その他の情報源

openSUSE での Apache で固有の問題に直面した場合は、まず <http://ja.opensuse.org/Apache> (日本語) または <http://old-en.opensuse.org/Apache> (英語) にある openSUSE wiki をお読みください。また、Apache の履歴は http://httpd.apache.org/ABOUT_APACHE.html にあります。このページでは、Apache が何故 Apache と呼ばれるようになったのかについても記述しています。

YaST を利用した FTP サーバの設定

29

YaST *FTP* サーバモジュールを利用することで、お使いのマシンを FTP (File Transfer Protocol) サーバとして機能するように設定することができます。また、お使いのマシンに匿名ユーザや認証ユーザが接続できるようにしたり、FTP プロトコルを利用してファイルをダウンロードできるようにしたりすることもできます。設定にもよりますが、FTP サーバに対してファイルをアップロードできるように設定することもできます。YaST では、お使いのシステムにインストールされた各種の FTP サーバデーモンに対し、統一的なインターフェイスを提供しています。

YaST *FTP* サーバ設定モジュールでは、2 種類の異なる FTP サーバを設定することができます:

- vsftpd (Very Secure FTP Daemon) and
- pure-ftpd

設定は事前にサーバをインストールしてある場合にのみ行なうことができます。なお、標準の openSUSE® メディアには pure-ftpd パッケージは含まれていません。その代わりに、YaST モジュールを設定することで pure-ftpd パッケージをリポジトリからインストールすることができます。

YaST *FTP* サーバ設定モジュールは、2 種類の FTP サーバデーモンに対応しています: vsftpd (Very Secure FTP Daemon の略) と pure-ftpd です。インストール済みのソフトウェアのみを設定することができます。標準で配布される openSUSE メディアには pure-ftpd のパッケージは同梱されていませんが、他のリポジトリから pure-ftpd パッケージをインストールすることで YaST モジュールを利用することができるようになります。

vsftpd と pure-ftpd サーバは、それぞれ少しずつ異なる設定オプションを提供しています。具体的には、*詳細設定* ダイアログ内が少し異なります。この章では、vsftpd (openSUSE での既定のサーバ) を選択した場合を想定して手順を記述しています。

また、YaST FTPサーバモジュールがお使いのシステムに存在しない場合は、yast2-ftp-server パッケージをインストールしてください。

YaST を利用して FTP サーバを設定するには、下記の手順で行ないます:

- 1 YaST コントロールセンターを開き、ネットワーク サービス > *FTP* サーバを選択するか、もしくは root から yast2 ftp-server コマンドを実行します。
- 2 お使いのシステムに FTPサーバがインストールされていない場合は、YaST FTPサーバモジュールの起動時に、どの FTP サーバをインストールするかを尋ねます。サーバを選択して (openSUSE では vsftpd が標準のサーバです) 進めてください。
- 3 まずは *起動* のダイアログで、FTP サーバの起動に関する設定を行ないます。詳しくは 29.1 項「FTPサーバの起動」(559 ページ)をお読みください。

次に *一般的な設定* ダイアログに移動します。FTP のディレクトリや「ようこそ」メッセージ、ファイル作成時のマスクなどの各種設定を行なってください。詳しくは 29.2 項「FTP の一般的な設定」(560 ページ)をお読みください。

パフォーマンス ダイアログでは、FTP サーバの性能に影響する 各種のパラメータを設定します。詳しくは 29.3 項「FTP パフォーマンス設定」(561 ページ)をお読みください。

さらに *認証* ダイアログでは、匿名ユーザと認証ユーザによるアクセスを許可するかどうかを指定します。詳しくは 29.4 項「認証」(561 ページ)をお読みください。

また、*詳細設定* ダイアログでは、FTP サーバの操作モードや SSL 接続、ファイアウォール設定などを行なうことができます。詳しくは 29.5 項「詳細設定」(562 ページ)をお読みください。

- 4 最後に *完了* を押すと設定を保存することができます。

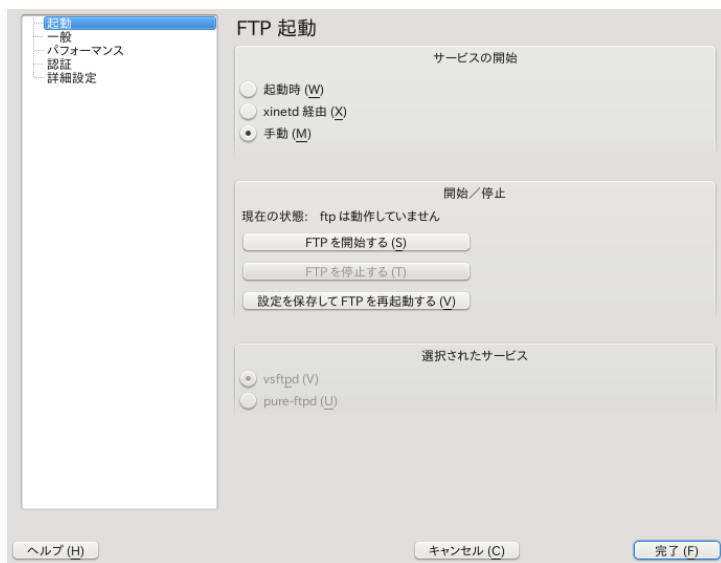
29.1 FTP サーバの起動

FTP 起動 ダイアログ内の サービスの開始 では、FTP サーバの起動を設定することができます。システムの起動時に自動的にサービスを起動するか、もしくは手動で起動するかを選択することができます。FTP による接続があったときにのみサービスを起動したい場合は、*xinetd* 経由を選択してください。

また、現在の FTP サーバの状態が *FTP 起動* ダイアログ内の 開始／停止 の枠内に表示されます。*FTP* を開始するを選択すると、すぐにサービスを開始することができます。逆にサービスを停止するには、*FTP* を停止するを押してください。サーバの設定を変更した後は、設定を保存して *FTP* を再起動するを押すと設定を保存することができます。また、完了ボタンを押すと設定を保存することができます。

FTP 起動 ダイアログ内の 選択されたサービス の欄には、どの FTP サーバを使用しているのか (*vsftpd* または *pure-ftpd*) が表示されます。両方のサーバがインストールされている場合には、これらの間を切り替えることもできます。設定を切り替えた場合、設定は自動的に変換されます。*pure-ftpd* パッケージは標準の *openSUSE* メディアには含まれていません。ご利用の場合は異なるインストール元からインストールしておく必要があります。

図 29.1 *FTP* サーバ設定 - *FTP* 起動



29.2 FTP の一般的な設定

FTP の一般的な設定 ダイアログ内の *一般的な設定* の枠内では、FTP サーバに接続した際に表示される 'ようこそ' メッセージを設定することができます。

また、全員を *chroot* のオプションを選択すると、全てのローカルユーザはログイン後、自身のホームディレクトリ内をルートディレクトリとした *chroot jail* に配置されるようになります。この設定はセキュリティ向上の目的から設定される項目ですが、特にユーザに対してアップロードの許可やシェルへのアクセスを許可している場合は注意して設定してください。

詳細なログ記録設定を有効に指定すると、全てのFTP リクエスト (要求) とレスポンス (応答) が記録されるようになります。

また、匿名ユーザや認証済みユーザが作成するファイルに対して、*umask* による許可制限を指定することができます。*umask* はビットで設定し、新しく作成するファイルのアクセス権に対して「無効化したい(マスクしたい)」ビットを指定します。匿名ユーザに対するファイル作成マスクは匿名ユーザの *umask* に、認証済みユーザに対するファイル作成マスクは認証ユーザの *umask* にそれぞれ入力してください。それぞれのマスク値は 0 で始まる 8 進数の値で入力しなければなりません。*umask* についての詳しい説明は、*umask* のマニュアルページ (`man 1p umask`) をご覧ください。

さらに *FTP* ディレクトリの枠では、それぞれ匿名ユーザと認証済みユーザに対するディレクトリを指定することができます。参照ボタンを押すと、ローカルのファイルシステムからディレクトリを選択することができます。匿名ユーザの既定の *FTP* ディレクトリは `/srv/ftp` に設定されています。なお *vsftpd* では、このディレクトリに対して全てのユーザに書き込みを許可することができません。代わりに、*upload* が作成され、そのディレクトリに対して書き込み許可が与えられるようになります。

注記: FTP ディレクトリに対する書き込み権限

pure-ftpd サーバでは、このディレクトリに対して匿名ユーザの書き込みを許可することができます。*pure-ftpd* から *vsftpd* にサーバを切り替える際は、*pure-ftpd* で利用していた書き込み許可を取り除いていることをご確認ください。

29.3 FTP パフォーマンス設定

パフォーマンスダイアログでは、FTP サーバの性能に関する各種のパラメータを設定することができます。無通信タイムアウト時間では、リモートのクライアントが最後に FTP コマンドを受信してから待機する最大の時間(分単位)を指定します。これ以上長い時間にわたってコマンドが受信できない場合は、リモートとの接続を切断します。IP アドレスあたりの最大接続数では、単一の IP アドレスから最大でどれだけの数の接続を受け付けるかを指定します。また、最大クライアント数では、同時に接続可能な最大のクライアント数を指定します。これを超えるクライアントからのアクセスがあった場合は、それらのクライアントにはエラーが返されます。

また、認証ユーザの最大通信速度では、認証済みのユーザに対して設定する最大のデータ通信速度(キロバイト毎秒 (KB/s) 単位)を指定することができます。匿名ユーザに対しては匿名ユーザの最大通信速度で指定します。既定の設定は 0 で、データ通信速度の制限を外す意味になります。

29.4 認証

認証ダイアログ内の匿名ユーザと認証ユーザの許可では、FTP サーバに対してどのユーザを許可するかを指定します:匿名ユーザに対してのみアクセスを許可するか、認証ユーザに対してのみ許可するか、もしくはその両方に許可するかのいずれかを選択します。

FTP サーバでファイルのアップロードを許可するには、認証ダイアログ内のアップロード枠内にあるアップロードの許可を選択してください。匿名ユーザに対してもアップロードやディレクトリの作成を許可するには、それぞれ関連するチェックボックスを選択してください。

注記: vsftp—匿名ユーザに対するファイルアップロードの許可

vsftpd サーバを使用していて、かつ匿名ユーザに対してファイルのアップロードやディレクトリの作成を許可したい場合は、匿名 FTP ディレクトリ内にサブディレクトリを作成し、そのディレクトリに全てのユーザに対するアクセス許可を設定してください。

29.5 詳細設定

FTPサーバは、アクティブモードとパッシブモードのいずれかで動作します。既定ではサーバはパッシブモードで動作します。アクティブモードで動作させるように切り替えるには、**詳細設定ダイアログ**内の**パッシブモードを許可**するのオプション選択を外してください。またパッシブモードを許可する場合、**パッシブモード時の最小ポート番号**と**パッシブモード時の最大ポート番号**を設定し、**データストリーム**で利用するポート範囲を指定することもできます。

サーバとクライアント間で暗号化通信を行ないたい場合は、**SSL**を有効化することができます。それぞれ**SSL 暗号化通信**でサポートすべきプロトコルバージョンと、使用する**DSA 証明書**を指定してください。

お使いのシステムがファイアウォールで守られている環境の場合は、**ファイアウォールでポートを開く**を選択してFTPサーバへの接続を許可するように設定することができます。

29.6 さらなる情報

FTPサーバについてさらに詳しく知るには、`vsftpd`と`vsftpd.conf`の各マニュアルページをお読みください。

パート VI. モバイル環境

Linux でのモバイルコンピューティング

30

モバイルコンピューティングとはラップトップコンピュータやPDA、携帯電話などを利用したコンピュータ環境のことで、それらの機器とPCとのデータ交換を行なう際にも利用する用語です。今では外付けのハードディスクやフラッシュドライブ、デジタルカメラなどをラップトップやデスクトップのシステムに接続できるようになっていて、様々なソフトウェアとともに利用することで様々な環境で便利に使うことができるようになっています。

30.1 ラップトップ

ラップトップのハードウェアは通常のデスクトップとは異なります。これは可搬性や物理的な大きさ、電源消費の要件がより厳しいことによります。モバイル環境で利用できるハードウェアの製造元では、PCMCIA (Personal Computer Memory Card International Association) や Mini PCI, Mini PCIe など様々な標準インターフェイスを開発し、ラップトップのハードウェアを便利にするための努力を行なっています。これらの標準は、メモリカードやネットワークインターフェイスカード、ISDN やモデムカード、外付けハードディスクについてもカバーしています。

ティップ: openSUSE とタブレット PC

openSUSE はタブレット PC にも対応しています。タブレット PC はタッチパッドやデジタイザと呼ばれる機器を搭載する PC で、マウスとキーボードを利用する代わりにデジタルペンや指を利用して入力や編集を行ないます。これらは他のシステムと同様にインストールし設定することができま

す。インストールと設定について、詳しい手順は 第33章 タブレット PC の使用 (611 ページ) をお読みください。

30.1.1 電源管理

電源管理を最適化できるように設計されたシステムでは、電源の提供されていない環境であっても適切に動作するような構成になっています。このようにエネルギーを内部で保存しておく仕組みは、少なくともオペレーティングシステムが行なう仕組みと同様に重要なものです。openSUSE® では、ラップトップ環境で電力消費に関連する設定や、内蔵のバッテリーを利用した場合の利用時間に関連する設定を各種提供しています。下記の一覧では、電力消費を減らすために必要な設定について、重要なものから順に示しています:

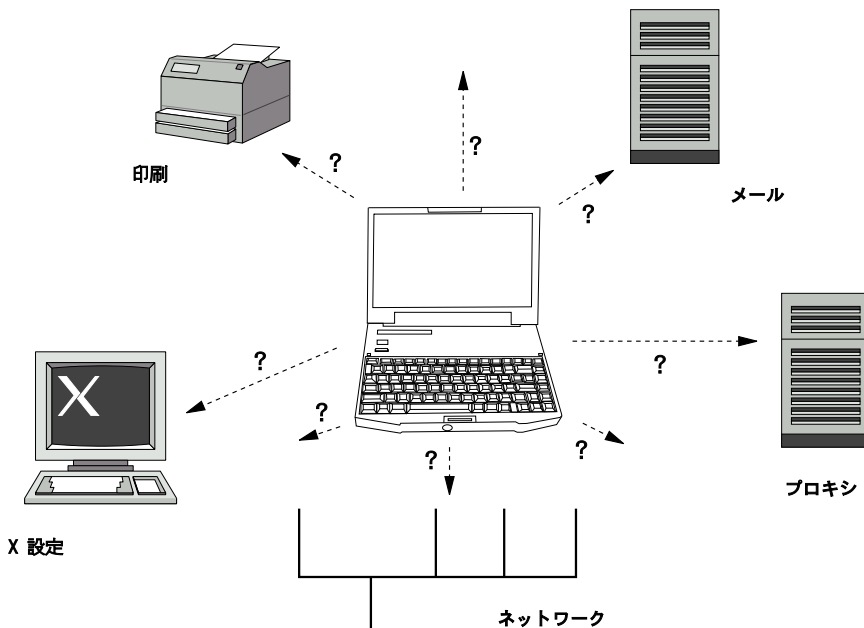
- CPU 速度の低減
- 不要なときに行なうディスプレイの電源断
- 手作業によるディスプレイの電源設定の調整
- 使用されていなかったり、ホットプラグに対応していたりするアクセサリの接続解除 (USB CD-ROM や外付けのマウス、未使用の PCMCIA カードや無線 LAN など)
- 未使用の際に行なうハードディスクの回転停止

openSUSE での電源管理について、背景となる詳細な情報は、第31章 電源管理 (577 ページ) をお読みください。デスクトップ固有の電源管理について、GNOME Power Manager の場合は項「デスクトップの電源管理機能の制御」(第2章 デスクトップでの作業, ↑GNOME ユーザガイド) を、KDE power management アプレットの場合は第9章 お使いのマシンにおける電源管理設定 (↑KDE ユーザガイド) をそれぞれお読みください。

30.1.2 利用環境の変化への対応

モバイルの環境では、お使いのシステムを操作環境にあわせた設定に変更したい場合がよくあります。環境に依存した多くのサービスやクライアントは、それぞれ再設定を行わなければなりません。openSUSE では、このような再設定作業を自動化するための仕組みを提供しています。

図 30.1 既存の環境に対するモバイルコンピュータの接続



家庭内ネットワークと企業内ネットワークなどの間で移動を行なうラップトップで、影響があると考えられるのはそれぞれ下記のサービスです：

ネットワーク

IP アドレスの割り当て方法や名前解決、インターネットへの接続方法や他のネットワークへの接続方法を含みます。

印刷

利用可能なプリンタや印刷サーバの一覧は、お使いのネットワーク環境によって異なることがあります。

電子メールとプロキシ

印刷と同様に、関連するサーバの一覧についても変更する必要があります。

X (グラフィカル環境)

お使いのラップトップをプロジェクタや外付けのモニタに一時的に接続する場合、別途のディスプレイ設定を行なう必要があります。

openSUSE では、利用環境の変化に対応するためにいくつかの手段を提供しています:

NetworkManager

NetworkManager は特にラップトップコンピュータでのモバイルネットワーク接続に対応したソフトウェアです。異なるネットワーク環境やネットワークの種類同士 (たとえば 有線 LAN と無線 LAN) で、簡単かつ自動的に切り替える機能を提供します。**NetworkManager** は無線 LAN の WEP や WPA-PSK の暗号化にも対応しているほか、ダイヤルアップ接続 (smpppd 経由) にも対応しています。なお、両方のデスクトップ環境 (GNOME および KDE) に対応した **NetworkManager** のフロントエンドが提供されています。フロントエンドについて、詳しくは 項「**KNetworkManager** の使用」 (第5章 *NetworkManager* の使用, ↑スタートアップ) と 項「**GNOME NetworkManager** アプレットの使用」 (第5章 *NetworkManager* の使用, ↑スタートアップ) をお読みください。

表 30.1 *NetworkManager* の使用可否判断

お使いのコンピュータが...	NetworkManager の判断結果
ラップトップコンピュータである	利用をお勧めします
異なる複数のネットワークに接続されるマシンである	利用をお勧めします
ネットワークサービス (DNS や DHCP など) を提供するマシンである	利用はお勧めできません
固定の IP アドレスだけを使用するマシンである	利用はお勧めできません

NetworkManager がネットワーク設定を取り扱わない場合は、**YaST** を利用してネットワーク の設定を行ないます。

SLP

Service Location Protocol (SLP) は既存のネットワークに対するラップトップ の接続を簡単にすることができます。 **SLP** を利用しない場合は通常、ラップトップの管理者はネットワーク内で利用できるサービスについて、細かい知識を得ておく 必要があります。 **SLP** はローカルネットワーク内に存在する全てのクライアントに対し、提供されている特定のサービスを通知することができます。これにより、 **SLP** に対応したアプリケーションは **SLP** からの通知情報を受け取り、自動設定を行なう ことができるようになっています。また **SLP** は、システムのインストールに対しても、必要なインストール元の設定を省略するために利用することができます。 **SLP** について詳しくは、第22章 ネットワーク内の *SLP* サービス (421 ページ) をお読みください。

30.1.3 ソフトウェアオプション

モバイル環境で使用する場合の様々な作業には、それぞれ対応した専用のソフトウェアが存在しています。たとえばシステムの監視(特にバッテリーの充電状況表示)やデータの同期、周辺機器やインターネットとの無線通信などがあります。下記の章では、それぞれの作業に対応した openSUSE における重要なソフトウェアについて述べています。

システム監視

openSUSE では 2 種類の KDE システム監視ツールが提供されています:

KPowersave

KPowersave は、コントロールパネル内で充電型バッテリーの状態を表示するアプレットです。アイコン表示で電源の供給状態を表示します。AC 電源が動作している場合は小さなプラグ型のアイコンが表示され、バッテリー動作をしている場合はバッテリー表示に切り替わります。

KPowersave のトレイアイコンの上でマウスの右ボタンを押すと、KPowersave の動作を設定することができます。必要に応じて 4 種類のスキームから選択してください。たとえば プレゼンテーションスキームを選択するとスクリーンセーバと電源管理を無効に設定し、システムイベントによってプレゼンテーションが邪魔されないようにします。また、ラップトップの蓋を閉じた場合の動作や電源ボタンを押した場合の動作を設定することもできます。

KSysguard

KSysguard はシステム内に存在する全ての計測可能データを収集し、一括監視することのできる独立アプリケーションです。KSysguard は ACPI (バッテリー状態) や CPU 負荷、ネットワークやパーティション設定、メモリ使用率などの監視を行なうほか、全てのシステムプロセスを監視したり表示したりすることができます。収集したデータの表示方法やフィルタ設定などはカスタマイズすることができますほか、様々なデータページ内で異なるシステムパラメータを監視することができたり、ネットワークを介して様々なマシンのデータを収集したりすることもできます。KSysguard は KDE 環境の無い状態で、デーモンとして動作することもできます。このプログラムについて詳しい情報は、内蔵のヘルプ機能や SUSE のヘルプページをご覧ください。

GNOME デスクトップでは、GNOME 電源管理設定とシステムモニタアプリケーションを ご利用ください。

データの同期

携帯型のマシンにおいて、ネットワークから切り離された環境と接続された環境を切り替える場合、データの同期は重要な問題となります。データの同期は電子メールのほか、特定のディレクトリや個別のファイルが対象となり、ネットワークに接続されていない環境でも利用できるようにしておくことが望まれます。このような要件を満たすには、下記のような方法があります：

電子メールの同期

ネットワーク内での電子メールアカウントについて、IMAPを利用するように設定してください。あとはお使いの各マシンでIMAPのオフライン環境が利用できる電子メールクライアント、たとえばMozilla Thunderbird Mail, Evolution, KMailなどを設定するだけです。それぞれの説明については、GNOME ユーザガイド (↑GNOME ユーザガイド) と KDE ユーザガイド (↑KDE ユーザガイド) をお読みください。なお、電子メールクライアントは必ず設定を行ない、常に同じ送信済みメッセージにアクセスするようにしてください。これにより、全てのメッセージとその状態が同期できるようになります。また、メール送信時にはシステムに添付されているMTAであるpostfixやsendmailの代わりに、メールクライアントに実装されたSMTP機能を利用して設定してください。これにより未送信のメールに対して正しい応答を得ることができるようになります。

ファイルやディレクトリの同期

ラップトップとワークステーション間でデータを同期するために便利なユーティリティには、いくつかの種類があります。詳しくは第34章 ファイルのコピーと共有 (623 ページ) をお読みください。

無線通信

ラップトップでは、家庭やオフィスで有線のネットワークに接続しているのと同じように無線接続を行なうことができます。無線では他のコンピュータに接続することができるだけでなく、周辺機器や携帯電話、PDAなどとも接続することができます。Linux では下記の3種類の無線接続に対応しています：

無線 LAN

様々な無線技術により、無線 LAN は大規模な環境にも空間的に離れた環境にも利用できる唯一の技術になっています。マシン同士独立した無線ネットワークを構成して互いに接続することができるほか、そこからインターネットに接続することも可能です。アクセスポイントと呼ばれる機器は、無線 LAN を利用する端末に対するベースステーションとなり、インターネットにアクセスする際の中継機器として動作します。モバイル環境のユーザは、その居場所に応じてアクセスポイントを切り替えることができ、それにより最適な接続状態を維持することができるようになっています。無線 LAN のユーザは、その居場所に縛られることなく携帯電話のような巨大なネットワークにアクセスすることができます。無線 LAN について、詳しくは第32章 *無線 LAN* (591 ページ) をお読みください。

Bluetooth

Bluetooth は全ての無線技術の中で最も広い適用範囲を提供するネットワークです。コンピュータ(ラップトップ)や PDA、携帯電話などのように、IrDA と同程度に広い適用範囲を持っています。また、範囲内にある様々なコンピュータ同士の通信に利用できるほか、無線システムに対応したコンポーネント、たとえばキーボードやマウスなどの通信にも対応しています。ただし、この技術ではネットワーク内にある物理的に離れたシステムに接続できるような仕組みは備えておらず、壁のような物理的な障壁が存在する環境で通信を行なう場合は、無線 LAN が適切な選択になります。

IrDA

IrDA は狭い範囲の無線通信に対応する無線技術です。通信する相手は互いに見える範囲に存在しなければならず、壁のような障害物を越えることはできません。IrDA でよくある適用範囲としては、ラップトップと携帯電話の間でのファイル転送です。ラップトップと携帯電話がごく近い範囲に存在している場合に IrDA を利用し、それらが遠い場合には携帯電話などのネットワークを利用します。また、IrDA の適用範囲としては、オフィス内での印刷ジョブ配信などもあります。

30.1.4 データセキュリティ

理想的には、お使いのラップトップ内に存在するデータを望まないアクセスから保護するための方法はいくつか存在しています。それぞれ下記のセキュリティ基準を考慮する必要があります：

盗難からの保護

可能な限りお使いのシステムを物理的な盗難から常時防いでおく必要があります。様々な保護ツール (たとえばチェーンなど) を量販店などから購入することができます。

強固な認証

ログインとパスワードに対する標準認証に加え、可能であれば生体認証を追加します。openSUSE では指紋認証に対応しています。詳しくは 第7章 *Using the Fingerprint Reader* (↑セキュリティガイド) をお読みください。

システムでのデータ保護

重要なデータは暗号化された通信路を介して転送するだけでなく、ハードディスク内でも暗号化を行なうべきものです。これによりマシンが盗難被害にあった場合でもデータを保護することができます。openSUSE での暗号化パーティションの作成については、第11章 *Encrypting Partitions and Files* (↑セキュリティガイド) をお読みください。もう1つの方法として、YaST からユーザを作成する際、暗号化ホームディレクトリを設定することもできます。

重要項目: データセキュリティとディスクへのサスペンド

暗号化パーティションはディスクへのサスペンド (Suspend to Disk) のイベントが発生した場合には、マウントが解除されません。そのため、これらのパーティションにある全てのデータは、ディスクへのサスペンドを行なったマシンではレジューム (復帰) を行なうだけでアクセスできるようになってしまうことに注意してください。

ネットワークセキュリティ

全てのデータ転送は、その転送方法にかかわらず全て暗号化されるべきものです。Linux とネットワークに関する一般的なセキュリティ問題については、第1章 *Security and Confidentiality* (↑セキュリティガイド) をお読みください。また、無線ネットワークでのセキュリティ基準については、第32章 *無線 LAN* (591 ページ) をお読みください。

30.2 モバイルハードウェア

openSUSE では、FireWire (IEEE 1394) や USB に接続された携帯型ストレージについて、自動検出を行なう機能が用意されています。携帯型ストレージデ

バイスには、FireWireやUSB接続のハードディスクのほか、USBフラッシュメモリやデジタルカメラも含んでいます。これらのデバイスは、システムが各インターフェイス経由で接続されたことを検知すると、自動的に検出され設定されます。GNOMEやKDEに対応したファイルマネージャでは、携帯型のハードウェアをうまく処理するための機能を提供しています。これらのメディアを安全にマウント解除するには、それぞれのファイルマネージャから安全に取り除く(KDE)またはアンマウント(GNOME)メニューをご利用ください。お使いのデスクトップ環境におけるリムーバブルメディアの取り扱いについて、詳しくはGNOME ユーザガイド(↑GNOME ユーザガイド)およびKDE ユーザガイド(↑KDE ユーザガイド)をお読みください。

外付けハードディスク (USB または FireWire)

システムから外付けのハードディスクが正しく認識されると、ファイルマネージャにそのアイコンが表示されるようになります。そのアイコンを選択して開くことで、ドライブ内の内容を表示することができます。その状態からファイルやフォルダを作成したりすることもできますし、編集や削除を行なうこともできます。システムが割り当てたハードディスクの名前を変更するには、アイコンの上でマウスの右ボタンを押すと表示されるメニューから、適切なメニューを選択してください。この名前の変更はファイルマネージャ内に表示される名前に対してのみ適用され、/media 内にマウントされるデバイス名に対しては反映されませんのでご注意ください。

USB フラッシュメモリ

これらのデバイスは、システムでは外付けのハードディスクと同じ扱いです。ファイルマネージャから名前を変更することもできます。

デジタルカメラ (USB または FireWire)

システムで認識されるデジタルカメラは、ファイルマネージャからは外付けのドライブのように表示されます。KDE では `camera:/` から画像にアクセスすることができます。あとはそれぞれ `digiKam` や `f-spot` のようなツールを利用して処理することができます。写真の処理に関する高度な処理を行なうには、`GIMP` をお使いください。また、`digiKam`, `f-spot`, `GIMP` について、詳しくは第17章 *Managing Your Digital Image Collection with DigiKam* (↑アプリケーションガイド), 第18章 *F-Spot: Managing Your Digital Image Collection* (↑アプリケーションガイド), 第16章 *Manipulating Graphics with GIMP* (↑アプリケーションガイド) をそれぞれお読みください。

30.3 携帯電話と PDA

デスクトップシステムでもラップトップシステムでも、Bluetooth や IrDA を介して携帯電話と通信を行なうことができます。型式によってはこれらの両方に対応しているものもありますが、いずれか片方にしか対応していないものもあります。これらのプロトコルに関する用途範囲については、「無線通信」(571 ページ)をお読みください。また、携帯電話側のこれらの設定方法については、お使いの機器のマニュアルをお読みください。

Palm, Inc. から発売されているハンドヘルドデバイスに対するデータ同期は、Evolution や Kontact で行なうことができます。デバイスに接続するための設定作業は、ウィザードを利用して簡単に行なうことができます。いったん Palm Pilot に対する接続が完了したあとは、同期するデータを決定するだけです(連絡先、予定など)。詳しくは GNOME ユーザガイド(↑GNOME ユーザガイド)と KDE ユーザガイド(↑KDE ユーザガイド)をお読みください。

より洗練されたデータ同期を行ないたい場合は、opensync プログラムをお使いください。それぞれ libopensync パッケージと msyncntool パッケージ、および各デバイスに対応したプラグインを利用します。

30.4 さらなる情報

モバイルデバイスと Linux に関する全ての疑問については、<http://tuxmobil.org/> (英語)を読むのがよいでしょう。この Web サイトでは様々なセクションが用意されていて、それぞれラップトップのハードウェアやソフトウェアのほか、PDA や携帯電話、その他のモバイルハードウェアに対応した情報が掲載されています。

似たようなアプローチをとっているサイトとして、<http://www.linux-on-laptops.com/> が作成した <http://tuxmobil.org/> (英語)というサイトもあります。ラップトップやハンドヘルドに関する情報も掲載されています。

また、SUSE ではラップトップに関するドイツ語の専用メーリングリストも用意しています。詳しくは <http://lists.opensuse.org/opensuse-mobile-de/> をお読みください。このメーリングリストでは、openSUSE でのモバイルコンピューティングについて、ユーザと開発者の双方が議論を行なっています。英語での投稿を行なってもかまいませんが、過去の投稿の

多くはドイツ語で書かれていることに注意してください。なお、英語用のメーリングリストとして <http://lists.opensuse.org/opensuse-mobile/> も用意されています。

OpenSync についての情報は、 <http://ja.opensuse.org/OpenSync> をお読みください。

電源管理

電源管理は特にラップトップ型のコンピュータでは重要な機能ですが、他のコンピュータであっても便利な仕組みです。その電源管理技術の 1 つである ACPI (Advanced Configuration and Power Interface) は、新しい製品であれば全てのもの(ラップトップ、デスクトップ、サーバ)に搭載されています。電源管理技術には適切なハードウェア構成と BIOS ルーチンが必要ですが、ほとんどのラップトップと多くの新しいデスクトップやサーバに搭載されています。また電源管理技術では、CPU の周波数を制御することで省電力を実現したり、動作音を低減したりすることもできます。

31.1 省電力機能

省電力の機能はラップトップ型コンピュータのモバイル使用に限らず、デスクトップシステムにおいても効果を発揮します。ACPI での主な機能と用途は下記の通りです:

スタンバイ
未対応です。

サスペンド (メモリへの)

このモードでは、システム全体の状態を RAM に書き込みます。その後、RAM を除く全てのシステムはスリープ状態に入ります。この状態により、システム全体の消費電力をととても抑えることができます。この状態の利点は数秒程度でスリープ前の作業を再開できるという点にあり、アプリケーションの起動や再起動が不要です。この機能は ACPI の S3 ス

テートと呼ばれる機能に対応しています。この状態のサポートは現在開発中の段階で、ハードウェア側に大きく依存した作りになっています。

ハイバネーション (ディスクへのサスペンド)

このモードでは、システム全体の状態がハードディスクに書き込まれ、システムの電源が落とされます。このモードでは、最低でもRAMと同容量のサイズをスワップパーティションに用意して、システム全体の状態を書き込むことができるようにしなければなりません。この状態からの復帰には 30 秒から 90 秒程度の時間が必要で、この復帰処理が完了すると元の状態に戻すことができます。製造元によっては、このモードを組み合わせた機能、たとえば IBM Thinkpad では RediSafe と呼ばれるような機能が用意されている場合もあります。この機能は ACPI の S4 ステートと呼ばれる機能に対応しています。Linux ではディスクへのサスペンドはカーネルルーチンが実施する仕組みになっており、ACPI とは別途の実装になっています。

バッテリーモニタ

ACPI はバッテリーの充電状態をチェックし、その情報を提供します。また、残り容量が限界に達した場合の処理を行なうこともできます。

自動電源 OFF

シャットダウンに続いてコンピュータの電源を切ることができる機能です。これは特に、バッテリーの容量が完全になくなる前に自動でシャットダウンを行なうような場合に重要です。

プロセッサの速度制御

CPU については、3 種類の異なる方法で省電力を実現することができます: 周波数と電圧の制御 (PowerNow! や Speedstep として知られているもの)、プロセッサに対する減速指定、スリープの指定 (C-ステート) があります。コンピュータの動作モードによって、これらの方法を組み合わせることもできます。

31.2 Advanced Configuration and Power Interface (ACPI)

ACPI はオペレーティングシステムに対して個別のハードウェアコンポーネントを設定し制御できるようにした仕組みです。また、ACPI は Plug and Play (PnP) と Advanced Power Management (APM) に取って代わるもので、バッテ

リーや AC 電源、温度、冷却ファン、「ノート PC の蓋を閉じた」情報や「バッテリーの残り容量が少ない」など、システムイベント情報を配信することができます。

BIOS では、個別のコンポーネントやハードウェアへのアクセス方法に関する情報を含んだ一覧表を提供しています。オペレーティングシステムは、この情報を利用して割り込みの割り当てやコンポーネントの有効化／無効化を設定します。ACPI ではオペレーティングシステムが BIOS 内に保存してあるコマンドを実行するため、ACPI の機能は BIOS の実装に依存することになります。ACPI が検出したり読み込んだりすることができる表については、`/var/log/boot.msg` にレポートが書き込まれます。また、ACPI の問題に対してトラブルシューティングを行なうには、31.2.3 項「トラブルシューティング」(582 ページ)をお読みください。

31.2.1 CPU 性能の制御

CPU の省電力制御には 3 種類の方法があります：

- 周波数と電圧の制御 (580 ページ)
- Throttling the Clock Frequency (T-states) (581 ページ)
- Putting the Processor to Sleep (C-states) (581 ページ)

コンピュータの動作モードにも依存しますが、これらの方法は組み合わせて使用することができます。また、省電力によってシステムの発熱を抑え、頻繁に冷却ファンが動作したりしないようにする効果もあります。

周波数の制御と減速は、プロセッサが何も作業を行っていない場合には最も経済的な C-ステートが適用されるため、何らかの作業を行っている場合にのみ効果があります。CPU が何らかの作業を行っている場合は、省電力の実現にあたっては周波数の制御がお勧めです。プロセッサに対しては間欠的な負荷が与えられることがしばしばあるため、周波数を下げることができるという仕組みです。通常、動的な周波数制御はオンデマンドの制御方法でカーネルが行なうのがベストです。

周波数の減速は、高い負荷にもかかわらずバッテリーの動作時間を延ばしたい場合などの最終手段として利用すべきものです。しかしながら、システムによっては減速時に滑らかに動作しないこともあります。また、CPU の減速は CPU が遅すぎる場合には意味をなさなくなります。

周波数と電圧の制御

PowerNow! や Speedstep は、それぞれ AMD や Intel が設計した技術です。しかしながら、この技術は他の製造元のプロセッサに対しても適用される場合があります。また CPU の周波数とコア電圧は同時に削減する場合があります、これによって比例以上の省電力効果があります。たとえば周波数を半分に (性能も半分に) した場合、電力は半分以上に削減することができます。この技術は ACPI とは独立しているものです。

CPU の周波数制御には、主に 2 つのアプローチ方法があります。カーネル自身が行なう方法 (カーネル内のガバナー (調整器) と呼ばれるものを利用した CPUfreq インフラストラクチャ) と、ユーザスペース内で動作するアプリケーションから行なう方法です。カーネル内のガバナーはポリシーをベースにした仕組みで、それぞれ異なる尺度 (CPU に対し、事前に設定された電源管理) で CPU の周波数を制御します。CPUfreq サブシステムでは、下記に示すガバナーを利用することができます:

パフォーマンスガバナー (performance)

CPU の周波数を利用可能な最大値に固定します。そのため、省電力は考慮されなくなります。

省電力ガバナー (powersave)

CPU の周波数を利用可能な最小値に固定します。これにより性能面で大きな影響があり、いくらプロセッサの負荷が高くても周波数が上がりません。

オンデマンドガバナー (ondemand)

動的な周波数制御のカーネル実装です: このガバナーではプロセッサの使用率を監視し、特定の閾値を超過するとすぐに周波数を利用可能な最大値に設定します。使用率が特定の閾値を下回ると、次に利用可能な小さい値に設定します。さらにシステムでの使用率が閾値を下回る場合は、さらに利用可能な小さい値に調整します。

保守ガバナー (conservative)

オンデマンドの実装に似ている実装で、このガバナーでもプロセッサの使用率に応じて周波数を調整します。ただしこのガバナーではよりゆっくりした調整を行いません。プロセッサの使用率が特定の閾値を超える場合、ガバナーはすぐに利用可能な最大値に設定したりすることはなく (この点がオンデマンドと異なる点です)、次に利用可能な高い値を設定します。

カーネルが提供するガバナーは、`/sys/devices/system/cpu/cpu*/cpufreq/` 以下の ファイルを閲覧することで確認できます。お使いのマシンに複数のプロセッサが搭載されている場合、`/sys/devices/system/cpu/` には各プロセッサに対するサブディレクトリが存在するはずです (たとえば `cpu0`, `cpu1` など)。また、お使いのシステムがオンデマンドまたは保守ガバナーを使用している 場合、`cpufreq` 内にはガバナーのサブディレクトリが作成され、ガバナーのパラメータが現われます。

Throttling the Clock Frequency (T-states)

This technology omits a certain percentage of the clock signal impulses for the CPU. At 25% throttling, every fourth impulse is omitted. At 87.5%, only every eighth impulse reaches the processor. However, the energy savings are a little less than linear. Normally, throttling is only used if frequency scaling is not available or to maximize power savings. This technology must be controlled by a special process, as well. The system interface for Processor Throttling States (T-states) is `/proc/acpi/processor/*/throttling`.

Putting the Processor to Sleep (C-states)

Modern processors have several power saving modes called C-states. They reflect the capability of an idle processor to turn off unused components in order to save power. The operating system puts the processor to sleep whenever there is no activity. In this case, the operating system sends the CPU a `halt` command. There are three idle states: C1, C2, and C3. In the most economic state, C3, even the synchronization of the processor cache with the main memory is halted. Therefore, this state can only be applied if no other device modifies the contents of the main memory via bus master activity. Some drivers prevent the use of C3. The current state is displayed in `/proc/acpi/processor/*/power`.

31.2.2 ツール

CPUfreq サブシステムについて現在の設定を表示したり、現在の設定を変更したりするには、`cpufrequtils` パッケージ で提供されているツールを利用してください。`cpufrequtils` パッケージを インストールしたあとは、`cpufreq-info` ツールを利用して CPUfreq のカーネル情報を取得することが

できるようになります。また、`cpufreq-set` コマンドを利用して `CPUfreq` の設定を変更することもできます。たとえば下記のコマンドを `root` で実行すると、その場で オンデマンドガバナーを有効にすることができます:

```
cpufreq-set -g ondemand
```

異なる情報や利用可能なオプションについては、それぞれ `cpufreq-info` と `cpufreq-set` の マニュアルページをお読みになるか、もしくは `cpufreq-info --help` または `cpufreq-set --help` を実行してヘルプをお読みください。

システムでの電源消費を監視するのに便利なツールとして、`powerTOP` というものがあります。`powertop` パッケージをインストールすると、不用意に高い電力消費をもたらしている原因を調査することができるようになる(たとえばプロセッサをアイドル状態から起こしている主なプロセスなど)ほか、このような問題を回避するためにシステムを最適化することもできるようになっています。このパッケージは Intel および AMD の両プロセッサに対応しています。詳しくは `powerTOP` のプロジェクトページ <http://www.lesswatts.org/projects/powertop/> をご覧ください。

上記のツールとは別に、下記のような `ACPI` ユーティリティを利用することもできます:

- ただ単にバッテリーの充電レベルや温度を表示したりしたい場合であれば、`acpi` コマンドをお使いください。利用可能なオプションの一覧は `acpi --help` を実行することで表示することができます。
- BIOS 内の `ACPI` テーブルを編集するには、`acpica` パッケージをインストールしてください。

31.2.3 トラブルシューティング

電源管理周りの問題としては、2 種類のものが考えられます。1 つはカーネル側の `ACPI` コードにバグが存在していて、正しく認識されない問題です。この場合、解決方法をダウンロードとして提供することができます。また、場合によっては BIOS 側の原因で発生する場合があります。また、他のよく使われているオペレーティングシステムでエラーが発生しないようにするため、わざと BIOS 側の実装を `ACPI` 仕様から逸脱させてエラーを回避している場合もあります。それ以外にも、`ACPI` 実装に深刻なエラーが存在するような

ハードウェアコンポーネントは、Linux カーネル側でブラックリストとして ACPI が動作しないようにもしています。

問題を発見したときに最初にやるべきことは、BIOS を新しいものに更新することです。それでも全く起動できない場合は、下記の起動パラメータを設定することで回避できる場合があります：

`pci=noacpi`

PCI デバイスの設定について ACPI を使用しないようにします。

`acpi=ht`

シンプルなりソース設定だけを行ない、その他の目的では ACPI を使用しないようにします。

`acpi=off`

ACPI を無効に設定します。

警告: ACPI 無しでの起動問題

新しいマシン (特にマルチプロセッサシステムや AMD 64 システム) によっては、ハードウェアを正しく設定するのに ACPI が必要である場合があります。このような場合は、ACPI を無効に設定すると何らかの問題が発生します。

また、USB や FireWire のハードウェアが接続されているマシンでは、マシン側が混乱してうまく起動しない場合があります。うまく起動できない場合は、不要なハードウェアを全て取り外してから再度起動を行なってみてください。

起動後は、`dmesg | grep -2i acpi` コマンドを利用して起動メッセージを確認してください。ACPI 以外の原因で発生している場合も考えられますので、メッセージ全てを確認してもかまいません。ACPI テーブルを処理する際にエラーが発生した場合は、最も重要な DSDT (*Differentiated System Description Table*) テーブルを改善版に置き換えることができます。これを行なうことで、問題のある BIOS 側の DSDT を無視することができます。手順については 31.4.1 項「ハードウェア側で ACPI が有効化されているのにうまく動作しない場合」(587 ページ)をお読みください。

カーネルの設定では、ACPI デバッグメッセージを有効化するためのスイッチが存在しています。カーネル側の ACPI デバッグ機能がコンパイルされてい

てインストールされている場合は、知識のあるユーザがエラーの原因を探るための情報を得ることができます。

BIOSやハードウェアの問題に直面した場合は、製造元に尋ねるのがお勧めです。特に製造元がLinuxに対する支援を提供しない場合は、彼らにその問題点を提示して解決してもらう必要があります。製造元は、Linuxを使用するユーザがある程度の数以上存在すれば、その問題を深刻なものとしてとらえることでしょう。

さらなる情報

- <http://tldp.org/HOWTO/ACPI-HOWTO/> (詳細なACPI HOWTOのほか、DSDTパッチも掲載されています)
- <http://www.acpi.info> (Advanced Configuration & Power Interfaceの仕様について)
- <http://www.lesswatts.org/projects/acpi/> (the Sourceforge内のACPI4Linuxプロジェクト)
- <http://acpi.sourceforge.net/dsdt/index.php> (Bruno Ducrot氏によるパッチ)

31.3 ハードディスクの休止

Linuxでは、不要であればハードディスクを完全にスリープ状態に移行させることもできるほか、省電力や静音モードに移行させることもできます。新しいラップトップであれば、不要な時に自動で省電力や静音のモードに切り替わるため、ハードディスクの電源を手作業で切ったりする必要はありません。しかしながら、最大限の省電力を実現したい場合は、hdparmコマンドを利用する下記の方法を試してみることをお勧めします。

このコマンドは様々なハードディスクの設定を行なうことができます。たとえば-y オプションを指定すると、指定したハードディスクをすぐにスタンバイモードに移行することができます。また、-Y オプションではスリープ状態に移行することができます。それ以外にも、hdparm -S x コマンドでは、一定時間の無動作でディスクの回転を止める設定を行なうことができます。xの値はそれぞれ下記のように設定してください: 0を設定すると機能を

無効にし、常に回転し続けるようになります。1 から 240 の値を指定すると、それぞれ 5 を掛けた値の秒数を指定したことになります。また、241 から 251 の値を指定すると、それぞれ 30 分を単位として 1 から 11 倍の時間を指定したことになります。

ハードディスク内部の省電力オプションを制御するには、`-B` オプションを設定します。0 から 255 までの値を設定し、小さければ小さいほど省電力を設定し、大きければ大きいほど性能を上げる指定になります。この値は使用するハードディスクに依存するため、具体的にどれだけの省電力になるのかは不明です。ハードディスクの動作音を低くしたい場合は、`-M` を設定してください。それぞれ 128 から 254 までの値で設定します。小さい値ほど静かになります。

ハードディスクをスリープ状態に移行させるのは簡単ではありません。Linux では様々なプロセスがハードディスクへの書き込みを行なうため、スリープを設定したとしても頻繁にスリープを解除させられる結果になります。そのため、Linux がハードディスクに書き込む必要のあるデータをどのように取り扱うのかについては、知っておくことが重要です。何よりもまず、全てのデータは RAM 内にあるバッファに保存されます。このバッファは `pdflush` デーモンが監視し、データが置かれてから一定の時間が経過するか、もしくは一定量まで使用されると、バッファの中身がハードディスクに書き込まれます。バッファサイズは動的に変化し、搭載されているメモリ量とシステムの負荷に依存して変動します。既定では `pdflush` は最大限の整合性保持のため、短い時間間隔に設定されています。既定では、5 秒おきにバッファが確認され、ハードディスクにデータが書き込まれます。それぞれ下記の値で調整することができます：

```
/proc/sys/vm/dirty_writeback_centisecs  
pdflush のスレッドが起動するまでの遅延時間 (1/100 秒単位)
```

```
/proc/sys/vm/dirty_expire_centisecs  
ディスクに書き込むべきデータについて、メモリ上に待機させる時間を  
指定します。既定は 3000 で、30 秒を意味します。
```

```
/proc/sys/vm/dirty_background_ratio  
pdflush がディスクに書き込むべきデータをメモリ上に保持する最大割合  
を指定します。既定値は 5% です。
```

`/proc/sys/vm/dirty_ratio`

メモリ上に存在するディスクに書き込むべきデータと全体メモリ量の比較で、この値を上回った場合は、そのプロセスに対してバッファへの書き込みを継続させずにディスクへの書き込みを行なわせるようにします。

警告: データの整合性の損傷

`pdflush` デーモンの設定を変更すると、データの整合性が損なわれることがあります。

これらのプロセスとは別に、`ReiserFS`, `Ext3`, `Ext4` などのジャーナリング機能付きのファイルシステムでは、`pdflush` とは独立したメタデータの書き込みを行なう機能が備わっています。そのため、これらのファイルシステムを利用することによって、ハードディスクの回転停止が阻害される場合もあります。このような問題を回避するには、モバイルデバイス向けの特別なカーネル拡張を利用する必要があります。詳しくは `laptop-mode-tools` パッケージをインストールし、`/usr/src/linux/Documentation/laptops/laptop-mode.txt` ファイルをお読みください。

また、それ以外にも動作中のプログラムがどのように振る舞うのかについても注意を払う必要があります。たとえば気の利いたエディタであれば、現在編集集中のファイルについて、隠しファイルを利用した定期的なバックアップを行っていたりすることがあります。このような動作があると、ハードディスクが定期的に動き出す結果になってしまいます。データの整合性は犠牲になりますが、省電力を求める際には、このような機能は無効化しておくのがよいでしょう。

また、これに関連してメールデーモンである `postfix` には、`POSTFIX_LAPTOP` という変数が存在しています。この値を `yes` に設定すると、ハードディスクに対して頻繁にアクセスしたりしないようになります。

31.4 トラブルシューティング

全てのエラーメッセージと警告は、`/var/log/messages` ファイルに書き込まれます。本章ではよく発生する問題について記述しています。

31.4.1 ハードウェア側で ACPI が有効化されているのにうまく動作しない場合

ACPI について何らかの問題に直面した場合は、`dmesg` コマンドの出力から ACPI 固有のメッセージを検索してください。たとえば、`dmesg|grep -i acpi` のように実行します。

また、問題を解決するには BIOS の更新が必要となる場合もあります。お使いのラップトップの製造元 Web ページをご覧ください。BIOS の更新版がないかどうかをご確認ください。また、最新の ACPI 仕様に準拠しているかどうかとも合わせてご確認ください。BIOS を交しかしても問題が解決しない場合は、下記の手順で問題のある BIOS 内の DSDT テーブルを更新してください。

手順 31.1 BIOS 内の DSDT テーブルの更新

下記の手順を実行する前に、必要なパッケージがインストールされていることをご確認ください。`kernel-source`, `acpica`, `mkinitrd` の各パッケージが必要です。

- 1 お使いのシステムに対応した DSDT を、<http://acpi.sourceforge.net/dsdt/index.php> からダウンロードします。ファイルが圧縮されている場合はそれを展開してください。ファイルの拡張子が `.aml` (ACPI マシン言語) になっている場合は手順 3 に移動してください。それ以外の場合は、次の手順を実施してください。
- 2 ダウンロードしたテーブルのファイル拡張子が `.asl` (ACPI ソース言語) である場合は、下記のコマンドでコンパイルを行なってください。

```
iasl -sa file.asl
```
- 3 ファイル `DSDT.aml` を任意の場所 (`/etc/DSDT.aml` がお勧めです) にコピーします。
- 4 `/etc/sysconfig/kernel` を編集し、DSDT ファイルのパスを指定します。
- 5 あとは `mkinitrd` コマンドを実行して `initrd` を作成すれば完了です。新しいカーネルをインストールした場合は、`mkinitrd` コマンドを実行すると

initrd が作成され、変更済みの DSDT が統合されてシステム起動時に読み込まれるようになります。

31.4.2 CPU の周波数制御がうまく働かない場合

カーネルのソースコードを参照し、お使いのプロセッサに対応しているかどうかを確認してください。また、CPU の周波数制御を働かせるためには特別なカーネルモジュールやモジュールオプションが必要になる場合もあります。kernel-source パッケージがインストールされている場合、これらの情報は `/usr/src/linux/Documentation/cpu-freq/*` にあります。

31.4.3 サスペンドやスタンバイがうまく働かない場合

ACPI システムでは、誤った DSDT 実装 (BIOS) によってサスペンドやスタンバイに問題が発生する場合があります。このような場合は、BIOS を更新してください。

また、システムが不具合のあるモジュールの読み込みを解除する際には、システムが停止してしまったり、サスペンドのイベントが動作しなかったりする場合があります。同じようなことは、サスペンドを妨害するサービスを停止する際にも発生する場合があります。いずれの場合とも、スリープモードへの移行を妨害しているものを調べてください。ログファイル `/var/log/pm-suspend.log` には、何が行なわれているのかを示す情報と、考えられるエラー情報がそれぞれ記載されています。また、サスペンドやスタンバイに移行する前に読み込みを解除しておきたいモジュールがある場合は、`/usr/lib/pm-utils/defaults` ファイル内の `SUSPEND_MODULES` 変数に設定を行なってください。

なお、サスペンドとそこからの復帰処理について変更を行なうための方法は、<http://ja.opensuse.org/Pm-utils> (日本語) または <http://old-en.opensuse.org/Pm-utils> (英語)、および <http://ja.opensuse.org/S2ram> (日本語) または http://wiki.opensuse.org/SDB:Suspend_to_RAM (英語) をお読みください。

31.5 さらなる情報

- <http://www.acpi.info> (Advanced Configuration and Power Interface の仕様)
- <http://www.lesswatts.org/projects/acpi/> (Sourceforge にある ACPI4Linux プロジェクトのページ)
- <http://acpi.sourceforge.net/dsdt/index.php> (Bruno Ducrot 氏による DSDT パッチ)
- <http://ja.opensuse.org/S2ram> (日本語) または http://wiki.opensuse.org/SDB:Suspend_to_RAM (英語) RAM へのサスペンドを動作させる方法
- <http://ja.opensuse.org/Pm-utils> (日本語) または <http://old-en.opensuse.org/Pm-utils> (英語) 汎用サスペンドフレームワークを修正する方法

無線 LAN

無線 LAN はワイヤレス LAN や WLAN (Wireless Local Area Network) とも呼ばれますが、これはモバイルコンピューティングにおいては欠くことのできない要素になっています。今やほとんどのラップトップコンピュータには無線 LAN カードが搭載されています。この章では、YaST を利用した暗号化などの無線 LAN カード設定方法について記しています。なお、無線 LAN は NetworkManager を利用して設定することもできます。詳しくは第5章 *NetworkManager* の使用(↑スタートアップ)をお読みください。

32.1 無線 LAN 標準

無線 LAN は IEEE という業界団体が作成した 802.11 標準を利用して通信を行っています。もともとこの標準は最大転送速度を 2 Mbit/s (メガビット毎秒) としていましたが、転送速度を上げる目的で様々な仕様追加が行なわれました。これらの仕様追加では変調方法や転送出力、転送レート (詳しくは表 32.1 「様々な無線 LAN 標準の概要」 (592 ページ) をお読みください) の見直しが行なわれました。また、多くの企業ではハードウェアに対して独占技術や草案段階の機能を追加してきました。

表 32.1 様々な無線 LAN 標準の概要

名前	帯域 (GHz)	最大転送 レート (Mbit/s)	注意
802.11 (オリジナル)	2.4	2	古い規格であり、今や対応機器はほとんど存在していません
802.11a	5	54	干渉に強い特性があります
802.11b	2.4	11	今はあまり使われていません
802.11g	2.4	54	11b との後方互換性があり、広く利用されています
802.11n	2.4 and/or 5	300	一般的に利用されています

802.11 オリジナルのカードは openSUSE® では対応していません。多くのカードは 802.11a, 802.11b, 802.11g, 802.11n のいずれか (または これらのうちの複数) に対応しているためです。新しいカードでは 802.11n 標準に対応していますが、802.11g としても利用できます。

32.2 動作モード

無線ネットワークにおいては、高速で高品質でかつ機密の高い通信を行なうため、様々な技術と設定が利用されます。動作モードが異なると異なる設定を行なう必要があります。また、正しい認証方法を選択するのは難しく、利用可能な暗号化にもそれぞれ異なるメリットやデメリット、間違いやすい点などが存在しています。

基本的に、無線ネットワークは下記の 3 つのネットワークモードに分類されます:

管理モード(マネージド、またはインフラストラクチャモードとも呼ばれるアクセスポイント経由の通信)

管理ネットワークには、管理要素であるアクセスポイントと呼ばれるものが存在します。管理モードはインフラストラクチャモードとも呼ばれ、このモードでは無線 LAN を利用するコンピュータの通信は全てそのアクセスポイントを介して行ないます。これにより、イーサネット上に接続されたものと同じように通信を行なうことができるようになります。

アドホックモード(一対一ネットワーク)

アドホックネットワークにはアクセスポイントがありません。それぞれの無線 LAN コンピュータは直接通信を行なうため、アドホックネットワークは一般に管理モードよりも高速で通信することができます。ただし、アドホック ネットワークでは転送帯域と参加しているコンピュータ数によって、大幅に速度が制限されることになります。また、このモードは WPA 認証には対応していません。そのため通常はアクセスポイントを利用して通信を行ないます。また、無線 LAN カードによってはアクセスポイントになるための機能を備えている場合もあります。

マスターモード

マスターモードでは、お使いのネットワークカードをアクセスポイントとして使用します。このモードを利用するには、お使いの無線 LAN カードがこのモードに対応している必要があります。お使いの無線 LAN カードについて、詳しくは <http://linux-wless.passsys.nl> をお読みください。

32.3 認証

無線ネットワークは有線ネットワークに比べて傍受や妨害が容易であるため、認証や暗号化方法に関する様々な標準が規定されています。オリジナル版の IEEE 802.11 標準では WEP (Wired Equivalent Privacy) という用語で規定されていました。しかしながら WEP は機密が保てないことが証明されてしまった (32.6.3 項「セキュリティ」(606 ページ)をお読みください) ため、WLAN industry (Wi-Fi Alliance に加わりました) は WPA と呼ばれる WEP の弱点を解消する拡張を規定しました。その後 WPA は IEEE 802.11i 標準となって WPA の仕様を含むこととなり、各種の認証や暗号化方法を規定する仕組みになりました。旧来の WPA は IEEE 802.11i のドラフト版をベースにしていたため、IEEE 802.11i は WPA2 としても知られています。

認可済みの端末だけが接続することができるよう、ネットワーク では下記のように様々な認証方法が使用されます:

なし (オープン)

オープンシステムでは何も認証を必要としません。任意の端末がネットワークに参加できます。ただし、WEP 暗号を使用することは可能です。詳しくは 32.4 項「暗号化」(595 ページ)をお読みください。

共有鍵 (IEEE 802.11 による)

この方式では、WEP 鍵を認証に使用します。しかしながら、WEP 鍵を認証に使用してしまうと容易に攻撃を受けてしまうため、認証は必須ではありません。攻撃者がやるべきことは、端末とアクセスポイントの通信を十分長く受信し続けることだけです。認証処理では両者が同じ情報を交換します。1 回は暗号化を行なった方式で、もう 1 回は暗号化を行なわない方式で実施します。これにより、適切なツールを利用して鍵を再構築することができるようになります。この方式では認証と暗号化の両方に WEP 鍵を使用するため、ネットワークのセキュリティを高めることにはなりません。また、端末側は認証や暗号化、暗号化解除のために正しい WEP 鍵を設定しておく必要があります。鍵を持っていない端末は受信したパケット 解読できないため、認証を行なうべきかどうかに関わらず通信を行なうことができなくなります。

WPA-PSK (IEEE 802.1x では WPA-Personal と呼ばれる)

WPA-PSK (PSK とは Pre Shared Key (事前共有鍵) の略) は、共有鍵と似た手順で通信を行ないます。参加している全ての端末とアクセスポイントに同じ鍵を設定します。鍵は 256 ビットの長さがあり、一般にパスフレーズという形で入力を行ないます。このシステムには WPA-EAP のような複雑な鍵管理は不要であるため、個人使用には便利な仕組みになっています。そのため、WPA-PSK は WPA 「Home (家庭用)」 と呼ばれる場合もあります。

WPA-EAP (IEEE 802.1x では WPA-Enterprise と呼ばれる)

実際のところ、WPA-EAP (Extensible Authentication Protocol) は認証システムではなく、認証情報を転送するためのプロトコルです。WPA-EAP は企業のような環境で無線ネットワークを守るために使用します。個人用のネットワークではほとんど使用されません。そのため、WPA-EAP は WPA 「Enterprise (企業用)」 と呼ばれる場合もあります。

WPA-EAPではRadiusサーバと呼ばれるものを使用してユーザを認証します。EAPでは、サーバに対して接続と認証を行なうのに、下記に示す3種類の方法を利用することができます:

- **Transport Layer Security (EAP-TLS):** TLS認証は、サーバとクライアントの間で証明書をお互いに交換し合うことによって認証を実現します。最初にサーバが自身の証明書をクライアントに対して提示し、クライアント側での検証を行ないます。クライアント側でその証明書が正しいものであると判断されると、今度はクライアント側からサーバに対して証明書を送信します。TLSを機密に保つには、お使いのネットワーク内で証明書管理のインフラストラクチャが必要となります。このようなインフラストラクチャは、個人用のネットワークではほとんど用意されていません。
- **Tunneled Transport Layer Security (EAP-TTLS)**
- **Protected Extensible Authentication Protocol (EAP-PEAP):** TTLSとPEAPは、いずれも2ステージから構成されるプロトコルです。最初のステージで機密を保持できる接続を確立し、次のステージでクライアントの認証データを交換します。TLSによる証明書管理インフラストラクチャが存在する場合でも、証明書を管理するための手間が存在しない分だけオーバーヘッドがずっと少なくなります。

32.4 暗号化

認可されていないユーザが無線ネットワーク内で交換されているデータを読み出したり、ネットワークに対するアクセス許可を得てしまったりすることを防ぐため、様々な暗号化方法が提供されています:

WEP (IEEE 802.11 による)

この標準ではRC4暗号化アルゴリズムを使用します。元々は40ビットの鍵長で暗号化を行なっていましたが、のちに104ビットの鍵長にも対応するようになりました。それぞれ40ビット鍵を64ビット鍵と表わしたり、104ビット鍵を128ビット鍵と表わしたりすることもあります。それぞれ24ビット分を初期ベクトルとして使用する分を含めているためです。しかしながら、この標準にはいくつかの弱点が見つかっています。このシステムが生成した鍵に対する攻撃は、ほとんどの場合で成功してしまいます。それでも全く暗号化を行わないよりはWEPを使用したほうがまだ適切です。

また、製造元によっては非標準の「動的 WEP」を実装している場合があります。WEP と全く同じ仕組みで同じ弱点を持っていますが、鍵管理サービスを使用して鍵を定期的に変更する点で異なっています。

TKIP (WPA/IEEE 802.11i による)

この鍵管理プロトコルは WPA 標準で規定されているもので、WEP と同じ暗号化方式を使用しますが、その弱点を克服しています。それは、それぞれのデータパケットで新しい鍵を生成するため、これらの鍵への攻撃は意味がなくなるためです。TKIP は WPA-PSK と共に使用します。

CCMP (IEEE 802.11i による)

CCMP は鍵管理を規定するものです。一般に WPA-EAP との接続で使用しますが、WPA-PSK でも使用することができます。暗号化方式は AES を使用するため、WEP 標準である RC4 よりは強い暗号になっています。

32.5 YaST を利用した設定

重要項目: 無線ネットワークにおけるセキュリティリスク

無線 LAN の接続で暗号化を行なわないと、全てのネットワークデータに対して 第三者からの傍受を許す結果になってしまいます。利用可能な認証方法や暗号化のうちのいずれかを利用して、お使いのネットワーク通信が保護されていることをご確認ください。

また、お使いのハードウェアに対応する最適な暗号化方法をお使いください。ただし、特定の暗号化方法を利用するには、ネットワーク内に存在する全てのデバイスがその暗号化方法に対応している必要があります。対応していないデバイスが存在した場合は、それらのデバイスは互いに通信できなくなってしまいます。たとえばお使いのルータが WEP と WPA の両方に対応しているものの、お使いの無線 LAN カードが WEP にしか対応していない場合、共通に利用できる WEP を選択することになります。ただし、WEP による暗号化は何も暗号化をしないよりは良い程度のものであることに注意してください。詳しくは 32.4 項「暗号化」(595 ページ)と 32.6.3 項「セキュリティ」(606 ページ)をお読みください。

YaST で無線 LAN を設定するには、下記のパラメータを設定する必要があります:

IP アドレス

固定の IP アドレスを設定するか、もしくはインターフェイスに対して動的な割り当てを行なうため、DHCPサーバを利用するように設定します。

操作モード

無線LANとお使いのマシンとの接続方法を設定します。これはネットワーク側の要件によって異なります。詳しくは32.2項「動作モード」(592ページ)をお読みください。

ネットワーク名 (ESSID)

ネットワークを識別するための名称を設定します。

認証と暗号化に関する詳細

ネットワーク側で提供されている認証や暗号化方法に依存し、それぞれ1つまたは複数の暗号鍵や証明書を設定します。

暗号鍵を入力するにあたっては、複数の入力方法があります。パスフレーズ、ASCII (WEP 暗号の場合にのみ利用できます)、16 進のいずれかを選択できます。

32.5.1 NetworkManager の無効化

無線LANカードは通常、インストール時に検出されます。お使いのマシンがモバイル用途のコンピュータである場合は、既定で NetworkManager が有効になります。無線LANカードを YaST から設定する場合は、まず NetworkManager を無効に設定する必要があります:

- 1 root で YaST を起動します。
- 2 YaST コントロールセンター では、ネットワークデバイス > ネットワークの設定を選択し、ネットワーク設定ダイアログを開きます。

お使いのネットワークが NetworkManager で制御されるように設定されている場合、YaST でネットワーク設定を編集できない旨の警告メッセージが表示されます。

- 3 YaST で編集できるようにするには、OK を押してメッセージを閉じ、グローバルオプションタブにある *ifup* を利用した従来の方法を選択します。

- 4 ここから先の設定については、32.5.2項「アクセスポイントを利用する設定」(598 ページ) または 32.5.3項「Ad-Hoc (アドホック) ネットワークの構成」(602 ページ) の手順に従ってください。

設定が終わったら、*OK* を押してネットワーク設定を 保存します。

32.5.2 アクセスポイントを利用する設定

この章では、お使いの無線LANカードを(外付けの)アクセスポイントに接続する 設定、もしくはお使いの無線 LAN カード自身をアクセスポイントとする設定(ただし後者についてはお使いの無線 LAN カード側での対応が必要で
す)について、手順を示しています。アクセスポイントを利用しない種類の
ネットワークを設定する 場合は、32.5.3項「Ad-Hoc (アドホック) ネット
ワークの構成」(602 ページ) をお読みください。

手順 32.1 お使いの無線LANカードについてアクセスポイントを利用する
よう設定する方法

- 1 YaST を起動し、 ネットワーク設定 ダイアログを開きます。
- 2 概要タブに移動し、システムで検出された全てのネットワーク デバイスを
表示します。一般的なネットワーク設定について、詳しくは21.4項「YaST
を利用したネットワーク接続の設定」(373 ページ) をお読みください。
- 3 一覧から無線 LAN カードを選択し、編集 ボタンを押して ネットワーク
カードの設定 ダイアログを表示します。
- 4 アドレスのタブでは、お使いのマシンに対して動的なIPを割り当てるか、
もしくは固定で割り当てるかを設定します。通常は 可変IP アドレス を選
択し、DHCP を選びます。
- 5 次へを押して、無線ネットワークカードの設定ダイアログに移動します。
- 6 お使いの無線 LAN カードからアクセスポイントに接続する場合は、動作
モードを 管理 に設定します。

逆にお使いの無線 LAN カードをアクセスポイントとして設定したい場合
は、動作モードを マスター に設定します。ただし、このモードは必ずし
も全ての無線 LAN カードで利用できるとは限らない ことに注意してくだ
さい。

注記: WPA-PSK または WPA-EAP の使用

WPA-PSK または WPA-EAP の認証モードを使用したい場合は、動作モードを **管理** に設定しなければなりません。

- 7 特定のネットワークに接続するには、ネットワーク名 (**ESSID**) に名称を入力します。名称を直接入力する以外にも、ネットワークの検索を押して利用可能なネットワークの一覧から選択することもできます。

無線ネットワーク内に存在し、相互に通信する全ての端末には、同じ **ESSID** を設定する必要があります。**ESSID** を設定しない場合、お使いの無線 LAN カードは、自動的に最も強い信号強度のアクセスポイントに接続しようとします。

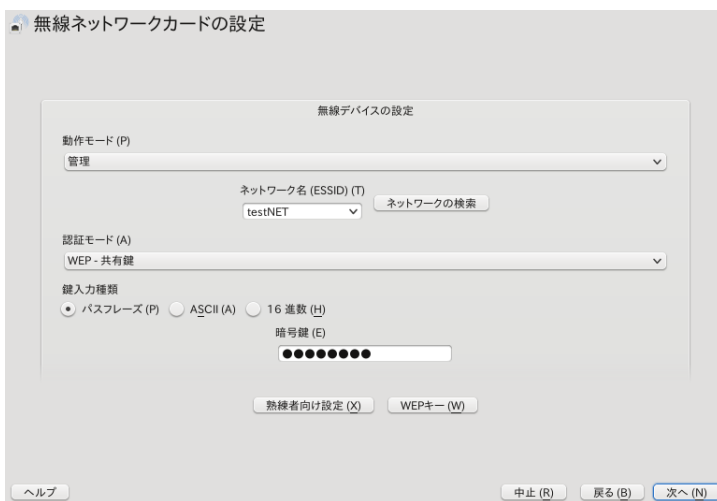
注記: WPA 認証には ESSID が必要です

WPA 認証を選択した場合、ネットワーク名 (**ESSID**) は必ず設定しなければなりません。

- 8 次にお使いのネットワークでの **認証モード** を選択します。どのモードが適切であるかについては、お使いの無線 LAN カードのドライバと、ネットワーク内にある他のデバイスの機能に依存して決まります。
- 9 **認証モード** を **暗号化無し** に設定した場合は、**次へ** を押すと設定が完了します。最後に潜在的なセキュリティリスクに関するメッセージを確認し、**概要タブ** (新しく設定した無線 LAN カードが表示されているはずです) から **OK** を押して終了です。

その他の認証モードを選択した場合は、手順32.2「暗号化の詳細設定の入力」(600 ページ)に進みます。

図 32.1 YaST: 無線ネットワークカードの設定



手順 32.2 暗号化の詳細設定の入力

下記に示す認証モードの場合、暗号鍵を設定する必要があります: **WEP - オープン**, **WEP - 共有鍵**, **WPA-PSK**

WEP の場合は 1 つだけ鍵を指定すれば十分です。ただし、お使いのステーションに対しては最大で 4 つまでの異なる鍵を設定することができます。これらのうちのいずれか 1 つの鍵を既定の鍵として設定し、暗号化に使用することになります。その他の鍵は復号化の際にのみ使用されます。既定では鍵の長さが 128 ビットになっていますが、64 ビットの鍵を選択して設定することもできます。

より高いセキュリティを実現するには、RADIUS サーバを利用してユーザの認証を行なう WPA-EAP をお使いください。サーバで認証を行なう際には、3 種類の認証方法、それぞれ TLS, TTLS, PEAP と呼ばれる方法を利用することができます。WPA-EAP に必要な資格情報と証明書は、RADIUS サーバで利用できる認証方法によって決まります。YaST は /etc/cert ディレクトリ以下にある任意の証明書を検索します。そのため、付与された証明書を上記のディレクトリに保存しておき、これらのファイルに対してアクセスを制限する (パーミッションで 0600、つまり所有者が読み書きだけを行なうことのできる状態) ように設定してください。

1 WEP- オープンまたは WEP- 共有鍵の暗号鍵を指定するには、下記の手順で行ないます:

1a まずは **鍵入力種類** を選択します。 **パスフレーズ**、**ASCII 16 進数** のいずれかを選択します。

1b 次に **暗号鍵** を必要な分だけ入力します (通常は 1 つの鍵だけを使用します):

パスフレーズを選択した場合は、それぞれ鍵の長さに合った長さの文字列を入力します (既定では 128 ビット分を指定します)。

ASCII を選択した場合は、64 ビット鍵を利用する場合は 5 文字を、128 ビット鍵を利用する場合は 13 文字を入力します。

16 進数を選択した場合は、64 ビット鍵を利用する場合は 10 文字を、128 ビット鍵を利用する場合は 26 文字をそれぞれ 16 進数で入力します。

1c より短いビット数の鍵に設定するには、**WEP キー** を選択して **鍵** の長さを 64 ビットに設定してください。 **WEP 鍵** ダイアログでは、それまでに入力された **WEP 鍵** を表示することができます。また、特に既定の **鍵** を設定しない場合、**YaST** は最初の **鍵** を既定の **鍵** として使用します。

1d さらに **WEP 鍵** を入力 (またはいずれかの **鍵** を変更) するには、それぞれ設定したい **鍵** の項目を選んで **編集** ボタンを押します。その後、**鍵入力種類** を選んで **鍵** を入力してください。

1e 最後に設定内容を確認し、**OK** を押して閉じます。

2 WPA-PSK の **鍵** を入力するには、下記の手順で行ないます:

2a まずは **鍵** の入力方法を、**パスフレーズ** または **16 進数** のいずれかから選択します。

2b **暗号鍵** の欄に入力方法に従った方法で **鍵** を入力します。

パスフレーズモードの場合、入力は 8 文字から 63 文字 までの間で指定します。16 進数モードの場合、64 文字で指定します。

3 *WPA-EAP* 認証を選択した場合は、次へを押して *WPA-EAP* ダイアログに移動し、ネットワーク管理者から付与された資格情報と証明書を入力します。

3a まずは *RADIUS* サーバが認証時に使用する *EAP* モードを選択します。以降の手順では、選択した *EAP* モードによってそれぞれ異なる情報の入力が必要になります。

3b *TLS* を選択した場合、識別情報、クライアント証明書、クライアント鍵、クライアント鍵パスワードをそれぞれ設定します。セキュリティを維持するには、サーバの正当性を確認するための項目 *サーバ証明書* も合わせて設定するとよいでしょう。

TTLS や *PEAP* の場合、識別情報とパスワードを指定します。サーバ証明書と匿名については必要に応じて設定してください。

3c *WPA-EAP* の設定についてより高度な設定を行なうダイアログを表示するには、詳細ボタンを押します。

3d まずは *EAP-TTLS* や *EAP-PEAP* の通信における第2ステージ (内側の認証) の認証方法を選択します。*RADIUS* サーバに対する認証方法の選択は、以前のダイアログで行なったものになります。

3e 自動決定された設定がうまく動作しない場合、*PEAP* バージョンの値を設定して特定の *PEAP* 実装を使用するように強制することもできます。

4 設定内容を確認し、*OK* を押します。概要タブ内に新しく設定した無線 LAN カードが表示されるようになります。

5 最後に *OK* を押すと設定を完了し、ダイアログを閉じることができます。

32.5.3 Ad-Hoc (アドホック) ネットワークの構成

環境によっては、無線 LAN カードの搭載された2台の PC で直接通信を行なったほうが都合の良い場合があります。YaST を利用してアドホックなネットワーク環境を構築するには、下記の手順で行ないます:

- 1 YaST を起動し、 ネットワーク設定 ダイアログを開きます。
- 2 概要タブに移動し、一覧から無線 LAN カードを選択します。選択したら 編集 ボタンを押し、 ネットワークカードの設定 ダイアログを開きます。
- 3 固定 IP アドレスを選択し、下記のデータを入力します:
 - IP アドレス: 192.168.1.1 を指定します。もう 1 台のコンピュータでは、たとえば 192.168.1.2 のように指定します。
 - サブネットマスク: /24 を指定します。
 - ホスト名: 任意の名前を入力します。
- 4 次へ を押して進めます。
- 5 動作モードを アドホック に設定します。
- 6 ネットワーク名 (ESSID) を入力します。ここに 入力する名前は任意の名前を入力することができますが、アドホックネットワーク内にある全てのコンピュータで同じ名前を設定してください。
- 7 次にお使いのネットワークでの 認証モード を選択します。どのモードが適切であるかについては、お使いの無線 LAN カードのドライバと、ネットワーク内にある他のデバイスの機能に依存して決まります。
- 8 認証モードを 暗号化無し に設定した場合は、次へ を押すと設定が完了します。最後に潜在的なセキュリティリスクに関するメッセージを確認し、 概要タブ (新しく設定した無線 LAN カードが表示されているはずです) から OK を押して終了です。

その他の認証モードを選択した場合は、手順32.2「暗号化の詳細設定の入力」(600 ページ)に進みます。
- 9 smpppd をインストールしていない場合は、YaST からインストールするよう促されます。指示に従ってインストールしてください。

- 10 ネットワーク内にある他の無線 LAN カードについても同様に設定します。それぞれ同じ ネットワーク名 (*ESSID*) と 認証モードを設定しますが、IP アドレスについては異なるものを設定してください。

32.5.4 その他のパラメータの設定

お使いの無線 LAN カードを設定するにあたっては、通常これらの追加設定を調整する必要はありません。ただしお使いの無線 LAN に接続するのに詳細な設定が必要となる場合、YaST では下記の設定を行なうことができます:

チャンネル

無線 LAN のステーションが動作すべきチャンネルを指定します。これはアドホック および マスター の動作モードでのみ必要な設定です。管理モードでは無線 LAN カードが自動的にアクセスポイントのチャンネルを検索します。

ビットレート

お使いのネットワークの性能に依存して、一方から他方に通信を行なうのにビットレートを設定する必要がある場合があります。既定では自動の設定になっていて、システムは利用可能な最大限のビットレートを使用しようとしています。なお、無線 LAN カードによってはビットレートの設定に対応していない場合もあります。

アクセスポイント

複数のアクセスポイントが存在する環境の場合、MAC アドレスを指定してどのアクセスポイントを使用するかを設定することができます。

電源管理

旅行中などの場合、電源管理機能を利用することで、ご利用のバッテリーの動作時間を伸ばすことができます。電源管理機能を利用すると、接続品質に影響があるばかりか、ネットワークの遅延も大きくなることにご注意ください。

高度なオプションにアクセスするには、下記のようにして行ないます:

- 1 YaST を起動し、 ネットワーク設定 ダイアログを開きます。
- 2 概要 タブに移動し、一覧から無線 LAN カードを選択します。選択したら編集 ボタンを押し、 ネットワークカードの設定 ダイアログを開きます。

- 3 次へ を押し、無線ネットワークカードの設定ダイアログを表示します。
- 4 熟練者向け設定 ボタンを押します。
- 5 アドホックモードの場合は、お使いのステーションと他のステーションで通信を行なうチャンネル(お使いの国に依存しますが、一般に 11 から 14 まで)を選択します。マスターモードの場合は、アクセスポイントの機能を提供する チャンネルを設定します。このオプションの既定値は 自動 になっています。
- 6 また、使用したい ビットレートを設定します。
- 7 続いて接続先の アクセスポイントの MAC アドレスを 入力します。
- 8 さらに 電源管理を使用するかどうかを選択します。
- 9 設定内容を確認して OK を押し、次へ を押したあと OK を押すと、設定を完了することができます。

32.6 無線 LAN 設定における豆知識

下記に示すツールや知識を利用することで、通信速度や安定性のほか、無線 LAN のセキュリティ要素についても監視したり改善したりすることができます。

32.6.1 ユーティリティ

wireless-tools パッケージには、無線 LAN 固有のパラメータを設定したり、統計情報を表示したりするための各種ツールが含まれています。詳しくは http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html (英語) をお読みください。

kismet (kismet パッケージ) は無線 LAN の通信トラフィックを聞き取ることのできるネットワーク解析ツールです。このツールを利用することで、ネットワークに対する侵入が行なわれた形跡がないかどうかを検出することもできます。詳しくは <http://www.kismetwireless.net/> (英語) とマニュアルページをお読みください。

32.6.2 安定性と速度

無線ネットワークにおける性能と信頼性は、主に参加している端末から他の端末にきれいな信号を送ることができているかどうか依存します。壁などの障害物が存在している場合は信号を大きく減衰させることになります。信号が減衰するとその分だけ転送速度が落ちることになります。接続中に `iwconfig` ユーティリティをコマンドラインから実行 (Link Quality の項目) したり、KDE が提供する **NetworkManager** のアプレットを利用したりして、信号の強さを確認してみてください。信号の品質に何か問題がある場合は、機器やアクセスポイントのアンテナの場所を変えるなどを行なってみてください。また PCMCIA 無線 LAN カードによっては補助アンテナに対応している場合もあります。このときは補助アンテナを使用するとそれなりに品質を改善することができる場合があります。また、54 Mbit/s などの製造元が表示している値は、理論上の最大値を示した建前上の値です。実際の最大データスループットは、大きくてもこの値の半分程度です。

また、`iwspy` というツールを使用すると、無線 LAN の統計情報を表示することができます。

```
iwspy wlan0
wlan0      Statistics collected:
  00:AA:BB:CC:DD:EE : Quality:0  Signal level:0  Noise level:0
  Link/Cell/AP      : Quality:60/94  Signal level:-50 dBm  Noise level:-140
  dBm (updated)
  Typical/Reference : Quality:26/94  Signal level:-60 dBm  Noise level:-90
  dBm
```

32.6.3 セキュリティ

無線ネットワークの設定を行なう場合、セキュリティの仕組みを導入しないと電波の届く範囲にいるユーザであれば誰にでもパケットを傍受できてしまうことにご注意ください。そのため、必ず暗号化を実施してください。全ての無線 LAN カードとアクセスポイントでは WEP 暗号化に対応していますが、これは全く持って安全なものではありません。単に攻撃者に対するちょっとした障害物程度にしかなりません。

個人使用の範囲では、できる限り WPA-PSK をお使いください。Linux ではほとんどのハードウェアで WPA に対応していますが、ドライバによっては WPA サポートが提供されていないものもあります。また、古いアクセスポイントや無線対応のルータで WPA に対応していないものもあります。このよ

うなデバイスの場合は、ファームウェアの更新で WPA に対応できないかどうかをご確認ください。WPA が利用できない場合でも、何も暗号化を行わないよりは WEP を利用するのがよいでしょう。また、高度なセキュリティ要件のある企業用途では、無線ネットワークは WPA のみに対応させるべきです。

お使いの暗号化方法でパスフレーズを設定する際には、より強度の高いパスフレーズを設定してください。たとえば<https://www.grc.com/passwords.htm> では、64 文字分のランダムなパスワードを生成することができます。

32.7 トラブルシューティング

お使いの無線 LAN カードが自動では検出されない場合、まずは openSUSE で対応済みのものかどうかをご確認ください。対応済みの無線 LAN ネットワークカードの一覧は、[http://ja.opensuse.org/HCL/Network_Adapters_\(Wireless\)](http://ja.opensuse.org/HCL/Network_Adapters_(Wireless)) (日本語) または [http://en.opensuse.org/HCL:Network_\(Wireless\)](http://en.opensuse.org/HCL:Network_(Wireless)) (英語) からご確認ください。お使いのカードが一覧にない場合は、ndiswrapper と呼ばれるソフトウェアを利用して Microsoft Windows のドライバを使用することがある場合があります。詳しくは <http://ja.opensuse.org/SDB:Ndiswrapper> (日本語) または <http://en.opensuse.org/SDB:Ndiswrapper> (英語) をお読みください。

また、お使いの無線 LAN カードが動作しない場合は、下記の要件が満たされているかどうかをご確認ください。

1. お使いの無線 LAN カードに対応するデバイス名は判明していますか？ 通常は wlan0 のような名前になっているはずです。ifconfig ツールを利用してご確認ください。
2. 必要なファームウェアをインストールしてありますか？ 詳しくは `/usr/share/doc/packages/wireless-tools/README.firmware` をお読みください。
3. お使いのルータから ESSID がブロードキャストされ、閲覧可能な状態 (隠蔽されていない) になっていますか？

32.7.1 状態の確認

iwconfig コマンドを利用することで、ご利用中の無線接続に関する情報を取得することができます。たとえば下記の出力例では、ESSID と 無線モード、周波数、暗号化の可否、リンク品質などが表示されています:

```
iwconfig wlan0
wlan0 IEEE 802.11abg ESSID:"guest"
      Mode:Managed  Frequency:5.22GHz  Access Point: 00:11:22:33:44:55
      Bit Rate:54 Mb/s   Tx-Power=13 dBm
      Retry min limit:7   RTS thr:off   Fragment thr:off
      Encryption key:off
      Power Management:off
      Link Quality:62/92   Signal level:-48 dBm  Noise level:-127 dBm
      Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
      Tx excessive retries:10   Invalid misc:0   Missed beacon:0
```

また、上記の情報を iwlist コマンドから取得することもできます。下記の出力例では、現在のビットレートが表示されています:

```
iwlist wlan0 rate
wlan0      unknown bit-rate information.
          Current Bit Rate=54 Mb/s
```

利用可能なアクセスポイントがどれだけあるのかを知りたい場合は、iwlist コマンドを利用することで調べることができます。下記のような形式で「セル」の一覧が表示されます:

```
iwlist wlan0 scanning
wlan0  Scan completed:
  Cell 01 - Address: 00:11:22:33:44:55
           Channel:40
           Frequency:5.2 GHz (Channel 40)
           Quality=67/70   Signal level=-43 dBm
           Encryption key: off
           ESSID:"Guest"
           Bit Rates: 6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s;
                     24 Mb/s; 36 Mb/s; 48 Mb/s
           Mode: Master
           Extra:tsf=0000111122223333
           Extra: Last beacon: 179ms ago
           IE: Unknown: ...
```

32.7.2 複数のネットワークデバイス

最近のラップトップコンピュータには、一般に有線と無線の両方のLANカードを内蔵しているものがあります。両方のデバイスをDHCP(アドレスの自動割り当て)で設定している場合は、名前解決やデフォルトゲートウェイの設定で問題が発生する場合があります。これはルータに対してpingが通るのに、インターネットには接続できないという現象で顕在化します。この場合はサポートデータベースにある下記の記事http://ja.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients (日本語) またはhttp://old-en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients (英語) をお読みください。

32.7.3 Prism2 カードでの問題

Prism2 チップに対しては、利用可能なドライバが複数存在しています。お使いのカードによって、どのドライバが動作するのかが異なっている状態です。これらのカードをお使いの場合は、hostap ドライバを利用したWPA 接続が唯一の解決になります。このようなカードが部分的に、もしくは全く動作しないような場合や、WPA をお使いになりたい場合は、`/usr/share/doc/packages/wireless-tools/README.prism2` をお読みください。

32.8 さらなる情報

さらなる情報については、それぞれ下記のページをお読みください:

http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html

Wireless Tools と呼ばれる Linux 向け無線 LAN ツールの開発者、Jean Tourrilhes 氏のインターネットページです。無線ネットワークに関する有用な情報が多数掲載されています。

tuxmobil.org

Linux を利用したモバイルコンピューティングについて、便利な情報が掲載されています。

<http://www.linux-on-laptops.com>

ラップトップコンピュータで Linux を使用する場合の情報源です。

[http://en.opensuse.org/HCL:Network_\(Wireless\)](http://en.opensuse.org/HCL:Network_(Wireless))

対応している無線 LAN カードの一覧です。

<http://en.opensuse.org/SDB:Ndiswrapper>

Ndiswrapper を利用して Microsoft Windows 向けの無線 LAN カードのドライバを利用する際の、問題回避方法などが記されています。

タブレット PC の使用

openSUSE® はタブレット PC にも対応しています。下記の章では、お使いのタブレット PC でインストールや設定を行なう方法と、デジタルペンからの入力を受け付ける便利な Linux アプリケーションについて、それぞれ紹介しています。

それぞれ下記のタブレット PC に対応しています:

- シリアルポートまたは USB の Wacom タブレット (ペンを利用するタイプ) や、タッチスクリーン、マルチタッチデバイスの接続されているタブレット PC
- FinePoint デバイスがあるタブレット PC。たとえば Gateway C210X/M280E/CX2724, HP Compaq TC1000 などがあります。
- タッチスクリーンがあるタブレット PC。たとえば Asus R2H, Clevo TN120R, Fujitsu Siemens Computers P-シリーズ, LG C1, Samsung Q1/Q1-Ultra などがあります。

タブレットPCのパッケージをインストールし、デジタイザを正しく設定することで、ペン入力(スタイラスとも呼ばれます)を下記のアクションやアプリケーションで利用することができるようになります:

- KDM や GDM を利用したログイン
- KDE や GNOME デスクトップのロック (施錠) 解除
- その他のポインティングデバイス (たとえばマウスやタッチパッド) で実行できる 作業全般。たとえば画面内でのカーソルの移動やアプリケーションの起動と終了、ウインドウのサイズ変更や移動、ウインドウフォーカスの切り替えやオブジェクトのドラッグ&ドロップ。
- X ウインドウシステムのアプリケーションにおけるジェスチャー認識の使用
- GIMP を利用した描画
- Jarnal や Xournal などを利用したメモ書きやスケッチ、Dasher を利用した巨大テキストの編集。

33.1 タブレット PC パッケージのインストール

タブレットPCに必要なパッケージは、TabletPC という 名称のインストールパターンに含まれています。このパターンをインストール中に選択すると、下記のパッケージがお使いのシステムにインストールされます:

- cellwriter: 手書き文字入力パネル
- jarnal: Java ベースのメモ書き アプリケーション
- xournal: メモ書き兼スケッチ アプリケーション
- xstroke: X ウインドウシステム向けのジェスチャー認識プログラム
- xvkbd: X ウインドウシステム向けの仮想キーボード

- `x11-input-fujitsu`: **Fujitsu P**-シリーズ タブレット PC 向け X 入力モジュール
- `x11-input-evtouch`: タッチスクリーン搭載タブレット PC 向け X 入力モジュール
- `xorg-x11-driver-input`: **Wacom** デバイスのモジュールを含む、入力デバイス向けの X 入力モジュール

これらのパッケージがインストールされていない場合はコマンドラインからこれらを インストールすることができるほか、 **YaST** から **TabletPC** のパターンを選択することでもインストールを行なうことができます。

33.2 タブレットデバイスの設定

インストールの際には、既定でお使いのタブレットやタッチデバイスが設定されます。 お使いの **Wacom** デバイスでの設定に何か問題がある場合は、これらの設定を変更するためにコマンドラインから `xsetwacom` コマンドを利用することができます。

33.3 仮想キーボードの使用

KDE や GNOME デスクトップへのログインを行なったり、画面のロック (施錠) を解除したりする場合、通常はユーザ名とパスワードを入力しますが、ログイン画面の下に表示された仮想キーボード (`xvkbd`) を利用して入力することもできます。 仮想キーボードを設定したり内蔵のヘルプにアクセスしたりしたい場合は、画面左下の `xvkbd` フィールドを選択し、`xvkbd` のメインメニューを表示してください。

なお、入力が確認できない場合(または必要なウインドウに入力を送信できない場合)は、`xvkbd` から フォーカス キーを押してフォーカスを転送し、キーボードイベントを送信したいウインドウを選択してください。

図 33.1 xvkbd 仮想キーボード

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	BackSpace	xvkbd (v3.2)					
Esc	1	2	3	4	5	6	7	8	9	0	-	^	\	Focus	Num Lock	/	*	Focus
Tab	q	w	e	r	t	y	u	i	o	p	@	[Delete	7 Home	8 Up	9 PgUp	+	
Control	a	s	d	f	g	h	j	k	l	;	:]	Return	4 Left	5	6 Right	-	
Shift	z	x	c	v	b	n	m	,	.	!		Shift		1 End	2 Down	3 PgDn	Enter	
xvkbc	Caps	Alt	Meta					かな	←	→	↑	↓		0 Ins		.	Del	

ログイン後に xvkbd を使用したい場合は、メインメニューから選択を行なうか、もしくはシェルから xvkbd を実行してください。

33.4 ディスプレイの回転表示

お使いのディスプレイをその場で回転させたりサイズ変更したりしたい場合は、KRandRTray (KDE) や gnome-display-properties (GNOME) をお使いください。KRandRTray, gnome-display-properties のどちらも、X サーバの RANDR 拡張を利用するアプレットです。

メインメニューから KRandRTray または gnome-display-properties を起動するか、もしくはシェルから krandrtray または gnome-display-properties を起動します。アプレットを起動すると、アプレットのアイコンがシステムトレイ内に表示されます。gnome-display-properties のアイコンがシステムトレイ内に自動で表示されない場合は、ディスプレイの設定ダイアログ内にあるこの設定アイコンをパネルの中に表示するを選択しているかどうかご確認ください。

KRandRTray でお使いのディスプレイを回転させたい場合は、アイコンの上でマウスの右ボタンを押し、ディスプレイを設定を選択します。表示された設定ダイアログを利用し、必要な向きに設定してください。

gnome-display-properties でお使いのディスプレイを回転させたい場合は、アイコンの上でマウスの右ボタンを押し、必要な向きを選択してください。お使いのディスプレイは、すぐにその方向になるように調整されます。グラフィックタブレットの向きについても同じく変更が行なわれるため、電子ペンの動きもそれに合わせて設定されます。

お使いのデスクトップの向きを変更する際に何らかの問題が発生した場合は、33.7項「トラブルシューティング」(620 ページ)をお読みください。

RANDR 拡張を利用するデスクトップ固有のアプレットについて詳しく知るには、項「モニタの設定」(第3章 動作のカスタマイズ, ↑KDE ユーザガイド)および項「画面の設定」(第3章 設定のカスタマイズ, ↑GNOME ユーザガイド)をお読みください。

33.5 ジェスチャー認識の使用

openSUSE では、CellWriter と xstroke と呼ばれるジェスチャー認識アプリケーションが提供されています。いずれのアプリケーションとも、ペンやその他のポインティングデバイスで実施したジェスチャーを認識し、X ウィンドウシステムに対して入力を送信することができます。

33.5.1 CellWriter の使用

CellWriter では、セルと呼ばれる枠の中に文字を書くことができます。記入した文字はすぐに文字として認識され、現在フォーカスのあるアプリケーションに対して、入力を送信することができます。なお、CellWriter をジェスチャー認識として使用する前に、あらかじめトレーニングを行なって手書き認識を調整する必要があります。それぞれの文字についてトレーニングを行ない、他の文字との区別を学習させてください。なお、トレーニングを行っていない文字は有効な形で表示されず、使用することができないことに注意してください。

手順 33.1 CellWriter のトレーニング

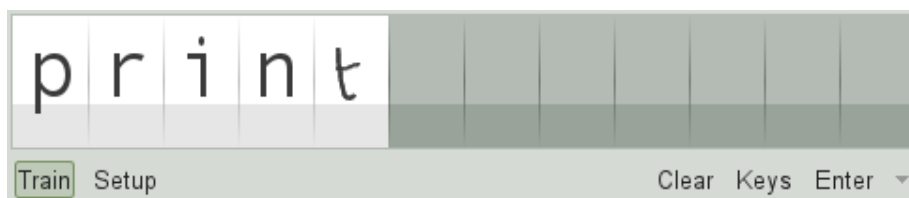
- 1 メインメニューから CellWriter を開始するか、もしくはコマンドラインから `cellwriter` コマンドを実行します。最初の起動では CellWriter は自動でトレーニングモードに移行します。トレーニングモードでは、選択されたキーマップの文字について一覧が表示されます。
- 2 それぞれの文字のセルに対してジェスチャーを入力してください。最初の入力では背景色が白色に、文字自身が明るい灰色で表示されます。その文字が黒く表示されるようになるまで、何回もジェスチャーを繰り返してください。トレーニングが行なわれていない文字は明るい灰色で表示される

か、茶色の背景で表示されます(お使いのデスクトップの色スキームに依存します)。

- 3 この手順を繰り返し、必要な全ての文字に対して **CellWriter** がトレーニング済みになるようにします。
- 4 **CellWriter** をその他の言語でできるようにトレーニングしたい場合は、**Setup** ボタンを押すと表示されるダイアログから、**Languages** タブを選択し、必要な言語 (日本語訳注: 日本語環境であれば "**Hiragana**" (ひながな), "**Katakana**" (カタカナ), "**CJK Unified Ideographs**" (漢字) になります。ただし漢字は文字数が多いので 注意が必要です) を選択してください。あとは **CellWriter** のウインドウから **Train** ボタンを押してトレーニングモードに移行し、右下のドロップダウンボックスから必要な文字セットを選択して、トレーニングを行なってください。
- 5 キーマップのトレーニングが完了したら、**Train** ボタンを 再度押して通常モードに戻してください。

通常モードでは、**CellWriter** ウインドウはジェスチャーを入力するための複数の空のセルが表示します。記入した文字は **Enter** ボタンを押すまでは、他のアプリケーションに送信されることはありません。そのため、入力として扱う前に、正しくない文字を修正したり削除したりすることができます。なお、認識時にアプリケーション側で確認のもてない字については、ハイライト表示が行なわれるようになっています。入力を修正するには、セルを選んでマウスの右ボタンを押し、表示されるコンテキストメニューを選択してください。文字を削除するにはペンの 消しゴム機能を利用するか、もしくはセルを選択してマウスの中央ボタンを押してください。**CellWriter** での入力を完了したら、送信先のアプリケーションウインドウ をマウスで選択してから、**Enter** を押してください。

図 33.2 *CellWriter* によるジェスチャー認識



なお、**CellWriter** で **Keys** ボタンを押すと、手書き文字認識の代わりに仮想的なキーボードを利用して入力することもできます。

CellWriter を隠すには、CellWriter のウインドウを閉じてください。アプリケーションがシステムトレイ内に隠れます。再度入力ウインドウを表示させるには、システムトレイ内のアイコンをマウスで選択してください。

33.5.2 Xstroke の使用

xstroke では、ペンやその他のポインティングデバイスを利用したジェスチャーを認識し、X ウインドウシステムのアプリケーションに対して入力を行なうことができます。xstroke のアルファベットは一筆書きのアルファベットで、そこから実際のアルファベットを構築するようになっています。xstroke を有効にすると、入力は現在フォーカスの設定されているウインドウに対して送信します。

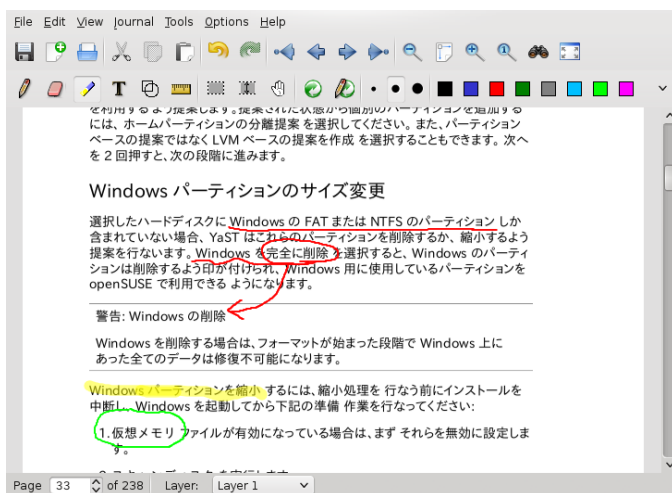
- 1 メインメニューから xstroke を開始するか、もしくはシェルから xstroke を実行します。これらにより、システムトレイに鉛筆型のアイコンが表示されるようになります。
- 2 ペンを利用してテキスト入力を行ないたいアプリケーションを開始します(たとえば端末ウインドウやテキストエディタ、OpenOffice.org Writerなどを起動します)。
- 3 ジェスチャー認識モードを有効にするには、鉛筆型のアイコンを1回押します。
- 4 お使いのグラフィックタブレット内でペンやその他のポインティングデバイスを利用し、ジェスチャーを実施してください。xstroke はジェスチャーを取り込んで認識し、フォーカスのあるアプリケーションに対して入力を送信します。
- 5 フォーカスを他のウインドウに移動するには、必要なウインドウをペンなどで選択し、しばらく選択したままにします(もしくはお使いのデスクトップのコントロールセンターで設定したキーボードショートカットを利用してもかまいません)。
- 6 ジェスチャー認識モードを解除するには、再度鉛筆型のアイコンを押してください。

33.6 ペンを利用したメモ取りとスケッチ

ペンを利用して絵を描く作業を行ないたい場合は、GIMPのようなプロフェッショナル向けのグラフィックエディタを使用するほか、XournalやJarnalのようなメモを書くためのアプリケーションを試してみるのがよいでしょう。XournalやJarnalではペンを利用してメモを書くことができるほか、絵を描いたりPDFファイルにコメントを書いたりすることができます。Jarnalについては、いくつかのプラットフォーム向けにJavaベースのアプリケーションが提供されているため、基本的な共同作業(コラボレーション)用の機能も備わっています。詳しくは<http://www.dklevine.com/general/software/tcl1000/jarnal-net.htm>をお読みください。また書いた内容を保存する際、SVGファイルを含むアーカイブフォーマット(*.jaj)で保存を行ないます。

メインメニューからJarnalまたはXournalを起動するか、シェルからjarnalまたはxournalと入力します。たとえばXournalでPDFファイルにコメントを付けたい場合は、*File > Annotate PDF*を選択し、お使いのファイルシステムにあるPDFファイルを開きます。あとはペンやその他のポインティングデバイスを利用して記入を行なってください。作業が終わったら*File > Export to PDF*を選択すると保存を行なうことができます。

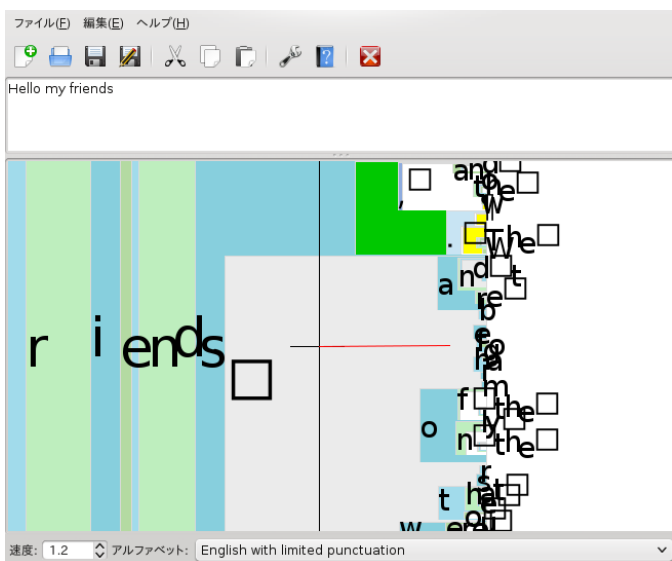
図 33.3 Journal による PDF への注釈付け



Dasher はもう一つの便利なアプリケーションです。キーボード入力の実用的でない環境や利用できない環境を主眼において開発されたアプリケーションで、ちょっとしたトレーニングを行なうだけでペン(またはその他の入力デバイス。アイトラッカーを利用することもできます)を利用した大規模なテキスト入力を素早く行なうことができるソフトウェアです。

メインメニューから Dasher を起動するか、シェルから dasher と入力します。ペンを一方に動かすと右側にある文字が拡大していきます。真ん中にある十字の線を文字が通過すると、テキストが作成されるか予測され、ウインドウの上部に入力が行なわれます。書き込みを停止したり開始したりしたい場合は、ペンを利用してディスプレイ上の任意の場所で押してください。また、拡大の早さを修正するには、ウインドウの下側で設定してください。

図 33.4 *dasher* を利用したテキスト編集



Dasher の考え方は多くの言語で利用できます。詳しくは Dasher の Web サイト <http://www.inference.phy.cam.ac.uk/dasher/> (英語) をお読みください。広範囲のドキュメンテーションやデモ、トレーニングテキストなどが存在します。

33.7 トラブルシューティング

仮想キーボードがログイン画面に現われない

時折、ログイン画面に仮想キーボードが表示されない場合があります。これを解決するには、`Ctrl+Alt+<` を 2 回押して X サーバを再起動するか、もしくはタブレット PC で必要なキーを押してください(内蔵キーボードを搭載していないスレート型の型式をご利用の場合)。それでも仮想キーボードが表示されない場合は、お使いの PC に外付けキーボードを接続し、ハードウェアのキーボードからログインを行なってください。

Wacom グラフィックタブレットの向きが変更できない

`xrandr` コマンドを利用すると、シェルからディスプレイの向きを変更することができます。利用可能なオプションの一覧を表示するには、`xrandr --help` コマンドを実行してください。同時にお使いのグラ

フィックタブレットの向きを変えたい場合は、下記のようにコマンドを修正する必要があります:

- 通常の向き (0 度の回転):

```
xrandr -o normal && xsetwacom --set "Serial Wacom Tablet" Rotate NONE
```

- 90 度の回転 (時計回り、ポートレート):

```
xrandr -o right && xsetwacom --set "Serial Wacom Tablet" Rotate CW
```

- 180 度の回転 (ランドスケープ):

```
xrandr -o inverted && xsetwacom --set "Serial Wacom Tablet" Rotate HALF
```

- 270 度の回転 (反時計回り、ポートレート):

```
xrandr -o left && xsetwacom set --"Serial Wacom Tablet" Rotate CCW
```

なお、上記のコマンドは `xsetwacom list` コマンドの出力結果次第であることに注意してください。また、`"Serial Wacom Tablet"` の部分は、それぞれスタイラスやタッチデバイスの出力結果を指定してください。タッチサポート付きの **Wacom** デバイス (指でカーソルを動かせるタイプ) をお使いの場合は、タッチデバイスについても回転を設定する必要があります。

33.8 さらなる情報

ここで説明したアプリケーションは、場合によっては統合されたオンラインヘルプが提供されていない場合があります。このような場合は、使用や設定方法に関するドキュメントが `/usr/share/doc/package/` パッケージ名のディレクトリに存在するか、もしくは下記の Web サイト (いずれも英語) にあります:

- Xournal のマニュアルについては、<http://xournal.sourceforge.net/manual.html> をお読みください。
- Jarnal のドキュメンテーションについては、<http://www.dklevine.com/general/software/tcl000/jarnal.htm#documentation> をお読みください。
- xstroke のマニュアルページは <http://davesource.com/Projects/xstroke/xstroke.txt> をお読みください。
- Linux 上の X システムで Wacom デバイスを設定する方法については、<http://linuxwacom.sourceforge.net/index.php/howto/x11> をお読みください。
- Dasher プロジェクトについて有益な情報は、<http://www.inference.phy.cam.ac.uk/dasher/> をお読みください。
- CellWriter に関するさらなる情報とドキュメンテーションについては、<http://risujin.org/cellwriter/> をお読みください。
- gnome-display-properties に関する情報は、<http://old-en.opensuse.org/GNOME/Multiscreen> をお読みください。

ファイルのコピーと共有

複数のオペレーティングシステム (OS) を同時にお使いの環境では、それらの OS 間でファイルを共有する要件がしばしば発生します。同じマシンでそれぞれ別々のパーティションに異なるシステムが存在する場合がありますし、ネットワークを介して異なるシステム同士が接続されている場合もあります。ここでは、それらの異なるシステム同士でファイル交換を行なう方法と、間違いやすい点をそれぞれ記述しています。

警告: 下記の手順は個人用／家庭用ネットワーク専用の手順です

下記に示す手順は、ファイアウォールで守られた個人用／家庭用のネットワーク以外では実施してはなりません。ファイアウォールで保護されていないネットワークや、企業用のネットワークにおいては、より高度なセキュリティ要件とそれに伴う設定が必要となりますが、本章では言及していないことをご了承ください。

データの交換を行なうためには、下記のいずれかの作業で実現します:

コピー

一方のシステムから他方のシステムにデータを転送することでデータの交換を行なう方法です。結果として、両方のシステムに同じデータが存在するようになります。

データの同期とは、データのコピーを特別な方法で行なうことを指します。一方のコンピュータでファイルを変更すると、同期を行なうことで他方のコンピュータにも自動で変更が反映されるようになります。たとえばお使いのラップトップに修正済みのファイルが存在し、その修正をデスクトップ側にも反映させたいような場合に該当します。

共有

クライアント／サーバの関係を設定して、お使いのファイルを共有する方法です。サーバ側からは、クライアント側からアクセスできる形式でファイルを提供します。ファイルを変更すると、その変更はサーバ内で実施されるため、クライアント側にはデータが残らなくなります。一般的にファイルサーバとは、クライアントに対して同時に多数のファイルを提供する仕組みです。

34.1 シナリオ

下記には、ファイル転送を行なう際に考えられるシナリオの一覧を示しています：

同じコンピュータ内での異なる OS の使用

多くのユーザがお使いのコンピュータには、製造元がインストールしたオペレーティングシステムが存在していて、それとは異なるパーティション上でLinuxが動作しているはずです。詳しくは34.4項「同一のコンピュータにおける異なるOS上のファイルへのアクセス」(628 ページ)をお読みください。

ネットワークで接続されていない異なるコンピュータ

任意のメディア(CD, DVD, USB フラッシュメモリ, 外付けハードディスクなど)にデータを保存し、複製先のマシンに接続(または挿入)してください。この方法はコストがかからず直感的で、かつ直接的な方法です。ただし、両方のコンピュータに適切なドライブやポートが必要になってしまふほか、両方のマシンのオペレーティングシステムで認識可能なファイルシステムを利用しなければなりません。

また、各メディアのサイズまでのファイルしかいっぺんに転送することができません。恒久的にファイルをコピーするような要件の場合は、ネットワークによる接続をお考えください。

同じネットワークに接続されている異なるコンピュータ

一方のコンピュータにサーバを設定し、サーバとクライアントを接続してファイルをコピーしてください。この作業を行なうためのプロトコルは多くの種類が存在するため、要件とお使いの方の知識にあったものを選んでお使いください。

クライアント／サーバの設定作業には知識が必要となるほか、管理の手間も発生してしまいますが、日々の作業でファイルを交換する必要がある場合や、複数のシステムで交換する必要がある場合にはよりよい選択肢となります。特に恒久的なファイル交換をご希望の場合は、クライアント／サーバの設定を選んでください。この方法では、ファイル交換時のサイズ制限などはありません。詳しくは34.2項「アクセス方法」(625ページ)をお読みください。

異なるネットワークに接続されている異なるコンピュータ

このシナリオの場合は、それぞれのネットワークが接続されている必要がありますが、接続のための作業は本章の範囲外であるため、記述されていません。コンピュータがネットワークで接続されていないものとして、ファイルを転送してください。

34.2 アクセス方法

下記には、ファイル転送やファイル共有を行なうために利用する方法やプロトコルを示しています：

FTP

ファイルの交換を異なるユーザと頻繁に行なうような場合は、**FTP (File Transfer Protocol; ファイル転送プロトコル)**を利用するのがお勧めです。**FTP**サーバの一方のシステムに設定し、クライアントからそこにアクセスするだけの作業です。**Windows**や**MacOS**, **Linux**などの多くのOS向けにグラフィカルな**FTP**クライアントソフトウェアが存在しています。どのような**FTP**サーバを利用するのかにもよりますが、一般に読み書きの権限を設定して使用します。**FTP**について、詳しくは34.5.4項「**FTP**を利用したファイルコピー」(637ページ)をお読みください。

NFS

NFS (Network File System; ネットワークファイルシステム)はクライアント／サーバ型のシステムです。サーバは1つまたはそれ以上のディレクトリをクライアントからインポートできるように公開します。詳しくは第26章 **NFS** でのファイル共有(477ページ)をお読みください。

ファイルの交換を異なるユーザと頻繁に行なうような環境では**NFS**がお勧めです。一般に、このプロトコルは**Windows**よりも**Linux**でより一般的な方法です。**NFS**で公開(エクスポート)したディレクトリは、お使いの**Linux**システムにうまく統合することができ、ローカルマシンのフォ

ルダと同じような方法でディレクトリ構造にアクセスすることができます。ご利用の設定にもよりますが、サーバ上で読み込みまたは書き込み、もしくはその両方を設定して使用します。通常は個人／一般家庭使用の範囲では読み書きの権限を設定して使用します。

rsync

それほど大規模に変更がかからないような巨大データについて、それらを定期的に転送したい場合は、rsyncを使用するのがお勧めです。このプロトコルはLinux および Windows に対応し、一般的にはrsyncをデータのバックアップ管理として使用します。詳しくはrsyncのマニュアルページか、もしくは34.5.2項「rsyncを利用したファイル転送」(632 ページ)をお読みください。

Unison

Unisonはrsyncの代替プロトコルで、異なるコンピュータ間で定期的な同期を行なうためのものですが、rsyncとは異なり双方向で同期を行なうことができます。詳しくはUnisonのマニュアルページか、もしくは34.5.3項「Unisonを利用したファイル転送」(634 ページ)のマニュアルページをお読みください。なお、UnisonはLinux および Windows に対応しています。

CSync

CSyncはUnisonの代替手段です。Unisonと同様にディレクトリを双方向に同期することができます。そのうえモジュール形式で構成されているため、プラグインでの拡張が可能です。詳しくは<http://www.csync.org>をお読みください。

SMB

Sambaはクライアント／サーバ型のシステムで、SMBプロトコルを実装したソフトウェアです。SMBプロトコルは一般にWindowsネットワークで使用されるものですが、複数のオペレーティングシステムに対応しています。Sambaについて詳しくは、第27章 *Samba* (493 ページ)をお読みください。

ファイル交換を頻繁に行なう環境で、特にWindowsシステムを利用する複数のユーザを相手にして共有する必要がある場合に、Sambaがおすすめです。SambaはLinuxだけの環境ではあまり使用されておらず、代わりにNFSを使用します。Sambaサーバの設定について、詳しくは34.8項「Sambaを利用したLinuxとWindowsのファイル共有」(644 ページ)をお読みください。

SSH

SSH (セキュアシェル; Secure Shell) はコンピュータ間で機密を保持した形で通信を行なうことができます。SSH の一式には複数のコマンドが含まれていて、ユーザを認証するのに公開鍵を使用することができます。詳しくは第13章 *SSH: Secure Network Operations* (↑セキュリティガイド) をお読みください。

ファイルのコピー頻度が低く、信頼できないネットワークを介して通信を行なう必要がある場合のほか、これを行なうのが1人だけである場合にSSHがお勧めです。グラフィカルなユーザインターフェイスも利用できますが、SSHは一般的にコマンドラインユーティリティを利用するものと考えられています。LinuxやWindowsにそれぞれ対応しています。

34.3 直接接続によるファイルアクセス

この章では、イーサネットのクロスオーバーケーブルを利用し、2台のコンピュータを接続してファイルを交換するための手順を示しています。

それぞれ下記のものを用意します:

- イーサネットのクロスオーバーケーブル。詳しくは<http://ja.wikipedia.org/wiki/%E3%82%A4%E3%83%BC%E3%82%B5%E3%83%8D%E3%83%83%E3%83%88%E3%83%BB%E3%82%AF%E3%83%AD%E3%82%B9%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%BB%E3%82%B1%E3%83%BC%E3%83%96%E3%83%AB> をお読みください。
- 両方のコンピュータでの openSUSE の起動
- ネットワークの接続。詳しくは項「ファイル共有とネットワーク参照における一般的な注意事項」(第5章 ネットワーク資源へのアクセス, ↑KDE ユーザガイド) をお読みください。

下記のようにして行ないます:

手順 34.1 GNOME

- 1 Nautilus を起動します。
- 2 ファイル > サーバへ接続を選択します。

- 3 サービスの種類では `ssh` を選択します。
- 4 相手のコンピュータの IP アドレスと、ポート番号 (既定値は 22) を入力します。
- 5 相手のコンピュータ上で、開きたいフォルダを入力します。
- 6 接続する を押します。

手順 34.2 KDE

- 1 Dolphin を起動します。
- 2 ネットワークを選択し、ネットワークフォルダを追加を押します。上記が表示されない場合は、表示 > パネル > 場所を選択してください。
- 3 ネットワークフォルダのタイプには セキュアシェル (`ssh`) を選択します。
- 4 それぞれ IP アドレス、ポート (既定値は 22)、相手のコンピュータにおけるフォルダ名をそれぞれ入力します。下の方にあるチェックボックスにチェックを入れることで、この接続に対するアイコンを作成することもできます。Dolphin では、ネットワークタブに表示されます。
- 5 ダイアログで 保存して接続を押すと、パスワードを尋ねられるので入力を行ないます。

上記の手順を行なうことで、相手側のコンピュータにあるフォルダを開くことができます。

34.4 同一のコンピュータにおける異なる OS 上のファイルへのアクセス

新しく購入したコンピュータの場合、一般に何らかのオペレーティングシステム (OS)、多くは Windows がインストールされています。Linux を異なるパーティションにインストールした場合、それらのオペレーティングシステムとファイルを交換する ような要件が発生する場合があります。

Windows の既定では、Linux のパーティションを読み込むことができません。これらのオペレーティングシステム間でファイルを交換したい場合は、一般に「交換用のパーティション」を作成して対応します。より直接的な方法で解決したい場合は、Windows 側で ext2 ファイルシステムにアクセスできるドライバを利用する方法があります。詳しくは<http://www.fs-driver.org/> (英語) をお読みください。なお、交換用のパーティションとして Windows と Linux の両方からアクセスできるようにするには、下記のいずれかのファイルシステムを利用します:

FAT

この種類のファイルシステムは、MS-DOS や Windows 95, Windows 98 などでも利用されています。YaST を利用することで、この種類のファイルシステムを作成することができるとともに、Linux から FAT パーティションにアクセスし、読み込みや書き込みを行なうことができます。FAT パーティションのサイズ (およびファイル 1 つあたりの最大サイズ) には制限があり、利用する FAT バージョンによって異なります。FAT ファイルシステムについて、詳しくは<http://ja.wikipedia.org/wiki/VFAT> をお読みください。

NTFS

NTFS ファイルシステムは Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7 でそれぞれ使用されているファイルシステムです。openSUSE には NTFS ファイルシステムに対して書き込みを行なうことのできる機能が用意されています。詳しくは<http://en.opensuse.org/NTFS-3g> (英語) をお読みください。

openSUSE のインストール時に Windows パーティションが存在すると、それらは検出され設定されますので、インストールが終われば Windows パーティションがマウントされます。お使いの Windows 側のデータにアクセスする方法は以下のものがあります:

KDE

まずは **Alt + F2** を押し、`sysinfo:/` と入力します。新しいウインドウが開いて、お使いのマシンに関する各種の情報が表示されます。*Disk Information* には接続されているハードディスクのパーティション情報が表示されますので、*Filesystem* の欄が `ntfs` または `vfat` になっているものを選び、マウスのボタンを押してください。パーティションがマウントされていない場合は、KDE がマウントを行なって中身を表示します。

コマンドライン

/windows ディレクトリの一覧を表示することで、お使いの Windows ドライブに含まれているコンテンツを表示することができます。たとえば Windows 側での C:\ ドライブが /windows/c ディレクトリになるように割り当てられます。

注記: Windows パーティションのアクセス権変更

ファイルシステムへのダメージを防ぐため、通常のユーザに対しては読み込みだけを行なうことができるような形でマウントが行なわれます。Windows のパーティションに通常のユーザから書き込みを含む完全なアクセス権を与えるには、これらの Windows パーティションのマウント方法を変更する必要があります。それぞれ vfat については mount コマンドのマニュアルページを、NTFS については ntfs-3g のマニュアルページをお読みください。

34.5 Linux コンピュータ間のファイルコピー

Linux では、コンピュータ間でファイルをコピーするためのプロトコルが多く用意されています。どのプロトコルを使用すべきかについては、どの程度の労力をかけるのかと Windows インストールとの互換性が必要かどうかによって決まります。本章では、Linux コンピュータからファイルをコピーしたり、Linux コンピュータにファイルをコピーしたりするための各種の手順を示しています。なお、本章ではネットワークを利用したコピーを行なうため、あらかじめネットワークの環境設定が完了していることを想定しています。また、全てのシナリオでは名前解決が動作する必要もあります。お使いのネットワークにネームサービスがない場合、IP アドレスを直接使用するか、もしくは全てのホストの /etc/hosts にホスト名と IP アドレスの対応を記述してください。

下記の例では、それぞれ下記の IP アドレスとホスト名を使用します:

宛先ホスト名	jupiter.example.com
宛先 IP	192.168.2.100

コピー元ホスト名	venus.example.com
コピー元 IP	192.168.2.101
ユーザ	tux

34.5.1 SSH を利用したファイルコピー

SSH 経由でアクセスする両方のコンピュータは、下記の要件を全て満たしてなければなりません:

1. ホスト名を使用してアクセスする場合は、それぞれのホスト名が両方のコンピュータ内の `/etc/hosts` ファイルに記載されていること (詳しくは「`/etc/hosts`」 (405 ページ) をお読みください)。IP アドレスで SSH アクセスを行なう場合、上記は特に必要ではありません。
2. ファイアウォールをお使いの場合は、SSH のポートを開くこと。これを行なうには、YaST を起動して `セキュリティとユーザ > ファイアウォール` を選択します。その状態から `許可するサービス` を選択し、SSH が一覧に載っているかどうかを確認します。一覧に載っていない場合は、`許可するサービス` で SSH を選択して、`追加` を押します。あとは次へ完了と押していき、変更内容を保存して YaST を終了します。

一方のコンピュータから他方のコンピュータにファイルをコピーするには、そのファイルがどこに存在しているのかを知っておく必要があります。たとえば `jupiter.example.com` というコンピュータにある単一のファイル `/srv/foo_file` をカレントディレクトリにコピーするには、下記のような `scp` コマンドを実行します (ドットはコピー先を指定しているもので、カレントディレクトリを意味します):

```
scp tux@jupiter.example.com:/srv/foo_file .
```

ディレクトリ構造全体をコピーしたい場合は、下記のような `scp` の再帰モードを利用します:

```
scp -r tux@jupiter.example.com:/srv/foo_directory .
```

ネットワーク内で名前解決を行なうことができない場合は、サーバの IP アドレスを直接指定します:

```
scp tux@192.168.2.100:/srv/foo_file .
```

どこにファイルが存在するのかがわからない場合は、`sftp` コマンドを利用します。KDE や GNOME で SFTP を利用したファイルのコピーを行なうのはとても簡単です。下記のようにして行ないます:

- 1 `Alt + F2` を押します。
- 2 コマンドの欄に下記を記入します(お使いの環境に合わせて変更してください):

```
sftp://tux@jupiter.example.com
```

- 3 サーバが提示する鍵情報について確認が行なわれたあと、`jupiter.example.com` 上のユーザ `tux` に対して、パスワードを入力します。
- 4 必要なファイルやディレクトリを、お使いのデスクトップなどのローカルディレクトリからドラッグ&ドロップで配置します。

KDE では `sftp` が利用できない場合、`fish` と呼ばれるもう 1 つのプロトコルを利用することができます。このプロトコルは `sftp` とよく似た使い方になっていて、単に `sftp` を `fish` に置き換えるだけで動作するようになっています:

```
fish://tux@jupiter.example.com
```

34.5.2 rsync を利用したファイル転送

`rsync` はデータをコピーしたりアーカイブを作成したりするのに便利なソフトウェアであるほか、デーモンとして起動することでネットワーク側にディレクトリを提供することができます(詳しくは [手順34.3「rsync 同期に対する高度な設定」](#) (633 ページ))。

異なるコンピュータ間で `rsync` を利用したファイルやディレクトリの同期を行なう前に、下記の要件が満たされていることをご確認ください:

1. `rsync` パッケージがインストールされていること。
2. 両方のシステムに同じユーザが存在すること。

3. サーバ側に十分なディスク領域が存在すること。
4. `rsync` の能力を完全に引き出したい場合は、サーバ側として利用するシステムに `rsyncd` がインストールされていること。

rsync 基本モード

`rsync` を基本モードで利用する場合には、特別な設定は不要です。 `rsync` では他のシステムに存在するディレクトリについて、完全な複製を作成します。 `scp` などの通常のコピーツールと大きく異なるようなことはありません。たとえば下記のコマンドでは、 `jupiter` と呼ばれるバックアップサーバ上にあるユーザ `tux` のホームディレクトリについて、バックアップを採取します:

```
rsync -Hbaz -e ssh /home/tux/ tux@jupiter:backup
```

バックアップから復元を行なう場合は、下記のコマンドを実行します (`-b` オプション無しで実行します):

```
rsync -Haz -e ssh tux@jupiter:backup /home/tux/
```

rsync デーモンモード

`rsync` の全ての機能を利用できるようにするため、一方のシステムで `rsyncd` デーモンを起動します。このモードでは、アカウント無しでアクセスすることのできる同期ポイント(モジュール)を作成することができます。 `rsync` デーモンを使用するには、下記のようにして行ないます:

手順 34.3 `rsync` 同期に対する高度な設定

- 1 `root` でログインを行ない、 `rsync` パッケージをインストールします。
- 2 同期ポイントを `/etc/rsyncd.conf` ファイルに設定します。大括弧内に同期ポイントの名前を入力し、 `path` キーワードに続いて実際のパスを入力します。たとえば以下のようになります:

```
[FTP]
path = /srv/ftp
comment = An Example
```

- 3 root の状態から `rcrsyncd start` コマンドを実行し、**rsyncd** デーモンを起動します。システム起動時に自動的に **rsync** サービスを起動する ように設定したい場合は、`insserv rsyncd` コマンドを実行します。

- 4 `/srv/ftp` ディレクトリ内にある全てのファイルを一覧表示するには、下記のように実行します (コロンが二重に付けられていることに注意):

```
rsync -avz jupiter::FTP
```

- 5 宛先のディレクトリ (この場合はドットを指定していて、カレントディレクトリにコピーする意味です) を指定すると、実際の転送を行なうことができます:

```
rsync -avz jupiter::FTP .
```

既定では **rsync** を利用して同期を行なう限り、ファイルが削除されることはありません。ファイルの削除を行なうには、`--delete` オプションを指定してください。なお、`--delete` オプションで新しいファイルが削除されないようにしたい場合は、代わりに `--update` オプションを指定してください。これにより発生する矛盾は、それぞれ手作業で解決する必要があります。

34.5.3 Unison を利用したファイル転送

異なるコンピュータ間で **Unison** を利用してファイルやディレクトリの同期を行なう前に、下記の要件が満たされていることをご確認ください:

1. **unison** パッケージがインストールされていること。
2. ローカルとリモートのコンピュータに、それぞれ十分なディスク容量があること。
3. **Unison** の能力を完全に引き出したい場合は、リモートのコンピュータにも **Unison** がインストールされていて、かつ起動していること。

ヘルプを必要とする場合は、**Unison** に `-doc topics` オプション を付けて起動し、利用可能なオプション一覧を表示させてください。

恒久的に設定を保存する場合、**Unison** は同期すべきディレクトリ (ルート) や無視する ファイルタイプなどのような各種設定を保存するための、プロファイルを作成することができます。プロファイルはテキストファイル形式で保

存され、 ~/.unison ディレクトリ以下に *.prf という拡張子で保存します。

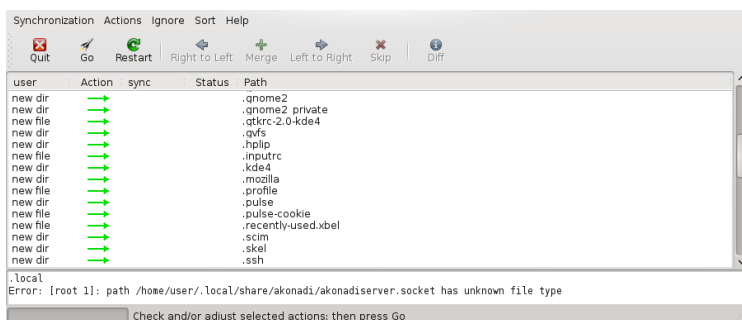
GUI の使用

Unison の GUI を利用して異なるディレクトリを同期するには、下記のようにして行ないます:

- 1 **Alt + F2** を押し、入力するコマンドに **unison** と入力して **Unison** を起動します。
- 2 **Unison** を始めた起動した場合、何もオプションを指定しなければ、複製元のディレクトリを尋ねられます。同期対象の複製もとディレクトリを入力して、**OK** を押します。
- 3 次に宛先のディレクトリを指定します。これはローカルでもリモートでもかまいません。リモートのディレクトリと同期を行ないたい場合は、方法 (**SSH**, **RSH**, ソケット) を選択し、ホスト名とユーザ名 (任意指定) も入力します。
- 4 以前にこれら 2 つのディレクトリについて同期を行なったことがない場合は、**Unison** がこれからこれらのディレクトリの比較を行なう旨の警告メッセージが表示されます。**OK** を押して警告メッセージを閉じると、**Unison** は両方のディレクトリの情報収集を行ないます。これが完了するまで待機すると、メインウィンドウに両方のディレクトリの差が表示されるようになります。

左側の列には選択した比較元ディレクトリの一覧が表示され、右側には比較先のディレクトリが表示されます。これらのディレクトリに差異があると、**Action** 列に作業提案が表示されます。緑色の矢印は、比較もとまたは比較先のディレクトリでファイルが更新／追加／削除されたことを示しています。矢印の向きが同期方向で、同期実行時にどちらからコピーを行なうのかを指定します。クエスチョンマークは矛盾を示していて、両方のファイルが更新されていることにより、どちらを最新のものとして扱えばよいのかがわからないことを示しています。

図 34.1 ファイル同期の提案



5 それぞれのファイルに対して Unison が表示した提案を変更する (たとえば同期方向を変更するなど) には、ファイルを選択して *Right to Left* (右から左に) または *Left to Right* (左から右に) を押します。 *Skip* を押すとファイルを同期対象から外します。それぞれ *Action* の列の表示が選択にあわせて変化します。

6 同期作業を開始するには、 *Go* を押します。

次回以降に Unison を起動した場合は、既存のプロファイルを示すダイアログボックスが表示され、同期すべきディレクトリの対を指定します。プロファイルの一覧から選択を行なうか、もしくは新しいプロファイルを作成(ディレクトリの対を新規に追加)し、設定した同期作業を行なうことになります。

コマンドラインの使用

Unison はコマンドラインを利用して操作することもできます。ローカルディレクトリと リモートのコンピュータとの間で同期を行なうには、下記のようにして行ないます:

1 シェルを開き、下記のコマンドを入力します:

```
unison -ui text ディレクトリ
ssh://tux@jupiter.example.com//パス
```

それぞれディレクトリとパスの項目には必要な値を入力してください。

2 Unison はファイルやディレクトリに対して何を行なうのかを尋ねてきます。たとえば:

```
local                                jupiter
<---- new file  dir [f]
```

- 3 Unison の推奨どおりに作業を行なうには、F を押します。それ以外のコマンドについては?を入力してください。
- 4 更新内容を適用するには、y を押します。

34.5.4 FTP を利用したファイルコピー

FTP サーバを設定する前に、下記の要件が満たされていることをご確認ください:

1. vsftpd パッケージがインストールされていること。
2. FTP サーバに対して root アクセスができること。
3. お使いのコンピュータに十分なディスク容量があること。

警告: 家庭内で使用するネットワーク限定の設定です

この設定は、家庭内で使用するネットワークに適した設定です。ファイアウォールで保護されていないネットワークに配置したり、不特定多数のアクセスを許す設定にしたりしてはなりません。

FTP サーバを設定するには、下記の手順で行ないます:

1 FTP サーバを準備します:

- 1a シェルを開いて root でログインし、/etc/vsftpd.conf ファイルのバックアップコピーを作成します:

```
cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
```

- 1b 匿名 FTP 用のアクセスポイントを作成します:

```
mkdir ~ftp/incoming
chown -R ftp:ftp ~ftp/incoming
```

- 2 希望するシナリオに沿って、設定ファイルを書き換えます(詳しい設定オプションについては、 `vsftpd.conf` の マニュアルページをお読みください):

匿名アクセスによる読み込み／書き込みの許可

```
#
listen=YES

# FTP サーバに対する匿名アクセスの許可
anonymous_enable=YES

#
local_enable=YES
# 書き込みアクセスの許可
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
dirmessage_enable=YES
# ログファイルの書き込み
xferlog_enable=YES
connect_from_port_20=YES
chown_uploads=YES
chown_username=ftp
ftpd_banner=Welcome to FTP service.
anon_root=/srv/ftp
```

FTP ユーザに対してアクセスの限定を指定

```
chroot_local_users=YES
```

- 3 FTP サーバを再起動します:

```
rcvsftpd start
```

クライアント側では、お使いのブラウザや FTP クライアントから `ftp://` ホストと入力します。ホストはお使いの環境に合わせて、サーバのホスト名または IP アドレスに置き換えてください。なお、FTP サーバのコンテンツにアクセスするのに便利なグラフィカルユーザインターフェイスも多数存在しています。YaST パッケージマネージャから "FTP" と入力すると、それらを一覧表示することができます。

34.6 SSH を利用した Linux と Windows コンピュータのファイルコピー

SSH を利用して Linux と Windows の間でファイルを転送するには、下記のいずれかのアプリケーションを利用します:

PuTTY

PuTTY は SSH デーモンと通信を行なうためのコマンドラインツール集です。 <http://www.chiark.greenend.org.uk/~sgtatham/putty.html> からダウンロードを行なうことができます。

WinSCP

WinSCP は PuTTY にとてもよく似たアプリケーションですが、グラフィカルな ユーザインターフェイスを持つという点が異なります。エクスプローラ型と Norton Commander 型のいずれかを選択することができます。 <http://winscp.net> からダウンロードを行なうことができます。

PuTTY を利用して Windows から Linux にファイルをコピーするには、下記のようにして行ないます (Windows マシン側での作業です):

- 1 PSCP を起動します。
- 2 SSH サーバのホスト名を入力します。
- 3 SSH サーバに対するログインとパスワードを入力します。

WinSCP を利用して Windows から Linux に接続するには、下記のようにして行ないます (Windows マシン側の作業です):

- 1 WinSCP を起動します。
- 2 SSH サーバのホスト名と、ユーザ名を入力します。
- 3 *Login* を押し、表示される警告を了解します。

- 4 WinSCP のウインドウを利用して、ファイルやディレクトリをドラッグ&ドロップで転送します。

注記: SSH フィンガープリント

PuTTY や WinSCP では、初回のログイン時に SSH のフィンガープリントを受け入れなければなりません。

34.7 Linux コンピュータ間のファイル共有

本章では、データを共有するための様々な方法を紹介しています。恒久的なデータ共有を希望する場合は、これらの方法のうちのいずれかを利用するのがよいでしょう。

34.7.1 NFS を利用したファイル転送

サーバを設定するには、下記の手順で行ないます:

- 1 システムの準備を行ないます:

- 1a シェルを開き、root でログインしてから全ユーザに対して書き込み許可を設定します:

```
mkdir /srv/nfs
chgrp users /srv/nfs
chmod g+w /srv/nfs
```

- 1b 次に、クライアント側で利用しているユーザ名とユーザIDが、サーバ上でも登録済みであることを確認します。ユーザアカウントの作成や管理については、第8章 *YaST* を利用したユーザ管理(135 ページ)に詳細な手順があります。

- 2 NFS サーバを準備します:

- 2a root で YaST を起動します。

2b ネットワークサービス>*NFS* サーバを選択します (このモジュールは既定ではインストールされません。YaST 内に表示されない場合は、`yast2-nfs-server` パッケージをインストールしてください)。

2c まずは *開始* を選択し、*NFS* サービスを有効にします。

2d ファイアウォールをお使いの場合は、ファイアウォールでポートを *開く* を選択してファイアウォールのポートを開きます。

3 ディレクトリを公開します:

3a ディレクトリの *追加* を押し、`/srv/nfs` を選択します。

3b オプション設定には下記のように指定します:

```
rw,root_squash,async
```

3c 複数のディレクトリを公開する場合は、上記の手順を繰り返します。

4 最後に設定を保存して終了します。これで*NFS*サーバを利用できるようになります。

NFS サーバを手作業で起動するには、`root` から `rcnfsserver start` と入力します。サーバを停止するには、`rcnfsserver stop` と入力します。既定では、YaST はシステム起動時にこのサービスの起動を管理します。

クライアントの設定を行なうには、下記の手順で行ないます:

1 *NFS* クライアントを準備します:

1a `root` で YaST を起動します。

1b ネットワークサービス>*NFS* クライアントを選択します。

1c ファイアウォールをお使いの場合は、ファイアウォールでポートを *開く* を選択してファイアウォールのポートを開きます。

2 リモート側のファイルシステムを取り込みます:

2a 追加を押します。

2b NFSサーバのホスト名またはIPアドレスを入力するか、もしくは選択を押してネットワーク上に存在するNFSサーバの一覧を表示させ、そこから選択します。

2c リモート側のファイルシステムのディレクトリ名を入力するか、もしくは選択を押して自動選択します。

2d 適切なマウント先を指定します。たとえば /mnt のように指定します。

2e 複数のディレクトリを取り込む場合は、上記の手順を繰り返します。

3 最後に設定を保存して終了します。これでNFSクライアントの設定は完了です。

なお、NFSクライアントを手作業で起動するには、`rcnfs start` と入力します。

注記: 一貫したユーザ名の使用

ごく少数のユーザでネットワーク環境をお使いの場合は、それぞれのマシンに対して同じユーザを設定してください。大規模なネットワークなどで多数のユーザを登録する必要があるような場合は、NISやLDAPを利用してユーザデータを管理することをお勧めします。詳しくは第3章 *Using NIS* (↑セキュリティガイド) と第4章 *LDAP—A Directory Service* (↑セキュリティガイド) をお読みください。

34.7.2 Samba を利用したファイル共有

この章では、Samba サーバ上にあるファイルに対してアクセスを行なうための各種の方法を説明しています。なお KDE や GNOME には、Samba の共有にアクセスするためのグラフィカルなツールが用意されているほか、Samba サーバにアクセスするためのコマンドラインツールも存在しています。

KDE と GNOME を利用した共有へのアクセス

KDE と GNOME のデスクトップでは、ファイルブラウザを利用して Samba の共有にアクセスを行ないます。下記の手順で行なってください:

- 1 **Alt+F2** を押し、`smb://jupiter.example.com/共有名` のように入力します。

URL の書式は `smb://ホスト/共有名` で、それぞれ ホストには Samba サーバのホスト名 (`jupiter.example.com`) または IP アドレスを、共有名には共有名を指定します。詳しくはステップ 3b(645 ページ)をお読みください。

- 2 ユーザ名とパスワードを入力してログインします。パスワードはステップ 4(645 ページ)の手順で設定するか、パスワードを必要としない環境であれば、何も入力せずに **Enter** を押します。
- 3 開いたウインドウを利用して、ドラッグ&ドロップでファイルやディレクトリを操作します。

なお、お使いのネットワーク環境のワークグループ名がわからない場合は、`smb:/` と入力すると一覧を表示することができます。Smb4K ツール (`smb4k` パッケージ)では、ネットワーク上に存在する全ワークグループを表示することができるほか、必要に応じてマウントを行なうことができます。

コマンドラインからの共有へのアクセス

コマンドラインを利用してアクセスしたい場合は、`smbclient` コマンドを利用します。Samba サーバにログインするには、下記のコマンドを実行してください:

```
smbclient //jupiter/share -U tux
```

既に `tux` ユーザになっている場合、`-U` は不要です。正常にログインが完了すると、それぞれ `ls`(ディレクトリ一覧の表示), `mkdir`(ディレクトリの作成), `get`(ファイルのダウンロード), `put`(ファイルのアップロード)などのコマンドが利用できるようになります。利用可能な全てのコマンドを表示するには、`help` と入力してください。詳しくは `smbclient` のマニュアルページをお読みください。

34.8 Samba を利用した Linux と Windows のファイル共有

Samba は Windows と Linux マシンの間でファイルを転送する際、第一に選択すべきものです。Samba を利用するにあたっては、下記のような使用形態が考えられます:

SMB スキームを利用した Linux から Windows へのファイル転送

Linux サーバを設定する必要が無いため、もっとも簡単な方法です。smb:/ というスキームを利用してアクセスを行ないます。詳しくは「KDE と GNOME を利用した共有へのアクセス」(643 ページ)をお読みください。なお、両方のシステムでワークグループの設定が同じに設定されていて、ディレクトリを共有していることをご確認ください。

サーバを利用した Windows から Linux へのファイル転送

お使いの Linux コンピュータ側で Samba サーバの設定を行ないます。詳しくは手順34.4「Samba サーバの設定」(644 ページ)をお読みください。

ティップ: お使いの Windows システムにおけるレジストリ設定の適用

Windows のバージョン (95, 98) によっては、異なる認証方法を有効にするため、レジストリを少しだけ変更する必要があります。これを簡単に行なうには、samba-doc パッケージをインストールして /usr/share/doc/packages/samba/registry 内にあるファイルを Windows ドライブにコピーしてください。あとは Windows 側でそのファイルをダブルクリックすると、変更を適用することができます。

手順 34.4 Samba サーバの設定

Samba サーバを設定するには、下記の手順で行ないます:

1 Samba サーバを準備します:

1a root で YaST を起動します。

1b samba パッケージをインストールします。

1c ディレクトリを作成します (以降では /srv/share ディレクトリを作成するものとします)。

2 サーバ設定を作成します:

2a ネットワークサービス > *Samba* サーバを選択します。

2b 表示されたワークグループ名からいずれかを選択するか、もしくは新しいワークグループ名を入力します (以降では Penguin ワークグループを選択したものとします)。

2c プライマリドメインコントローラ (*PDC*) を選択します。

2d お使いのコンピュータの起動時、毎回 *Samba* サーバを自動で起動したい場合は、*起動時の動作* を選択します。それ以外の場合は *手動* を選択します。

2e ファイアウォールをお使いの場合は、*ファイアウォールでポートを開く* を選択してファイアウォールのポートを開きます。

3 Windows 共有を作成します:

3a 共有タブに移動し、*追加* を押します。

3b 共有名と説明を入力します。共有名には、クライアントからアクセスする際の名前を入力します。共有の説明には、この共有の目的を記入します。

3c パスを選択します (たとえば /src/share を選択します)。

3d *OK* を押して先に進みます。

3e ユーザにディレクトリの共有を許可するを選択します。

4 このサービスの利用を許可する全てのユーザに対して、下記のようにしてパスワードを設定します:

```
smbpasswd -a tux
```

設定を簡単にするには、**Enter** だけを押しパスワードを何も設定しないのがよいでしょう。なお、お使いの **Windows** と **Linux** ではそれぞれアカウントの管理体系が異なるため、同じユーザ名を持つユーザを設定するようにしてください。

5 Samba サーバを起動します:

```
rcnmb start
rcsmb start
```

正しく設定を行なうことができているかどうかを確認するには、下記のように入力します:

```
smbclient -L localhost
```

Enter を押すと、下記のような表示が現われるはずです:

```
Anonymous login successful
Domain=[PENGUIN] OS=[Unix] Server=[Samba 3.0.22-11-SUSE-CODE10]

      Sharename      Type      Comment
      -----
      share           Disk      Shared directory
      netlogon        Disk      Network Logon Service
      IPC$            IPC       IPC Service (Samba 3.0.22-11-SUSE-CODE10)
      ADMIN$          IPC       IPC Service (Samba 3.0.22-11-SUSE-CODE10)
Anonymous login successful
Domain=[PENGUIN] OS=[Unix] Server=[Samba 3.0.22-11-SUSE-CODE10]

      Server          Comment
      -----
      SUSE-DESKTOP    Samba 3.0.22-11-SUSE-CODE10

      Workgroup       Master
      -----
      TUX-NET         jupiter
```

34.9 さらなる情報

- <http://ja.wikipedia.org/wiki/VFAT>
- <http://ja.wikipedia.org/wiki/NTFS>
- <http://en.wikipedia.org/wiki/Fstab> (英語)
- http://ja.wikipedia.org/wiki/Network_File_System
- http://ja.wikipedia.org/wiki/File_Transfer_Protocol
- http://ja.wikipedia.org/wiki/Secure_Shell
- <http://ja.wikipedia.org/wiki/Rsync>
- <http://ja.wikipedia.org/wiki/Samba>

ヘルプとドキュメンテーション

openSUSE®では様々な情報やドキュメンテーションが提供されていて、それらの多くはお使いのシステムに既にインストールされています。

/usr/share/doc 内のドキュメンテーション

伝統的にヘルプ用として利用されてきたこのディレクトリには、お使いのシステムに関する様々なドキュメンテーションファイルやリリースノートが含まれています。それ以外にも、packages ディレクトリ以下にインストール済みのパッケージに対する情報も含まれています。詳しくは 35.1 項「ドキュメンテーションディレクトリ」(650 ページ)をお読みください。

シェルコマンド向けのマニュアルページと info ページ

シェルで作業を行なう場合は、コマンドオプションを全て記憶しておく必要はありません。伝統的に、シェルにはマニュアルページや info ページの形で、同梱のヘルプ機能が用意されています。詳しくは 35.2 項「マニュアルページ」(652 ページ)や 35.3 項「info ページ」(654 ページ)をお読みください。

デスクトップのヘルプセンター

KDE デスクトップのヘルプセンター (KDE ヘルプセンター) や GNOME デスクトップのヘルプ (Help) では、お使いのシステムにおける最も重要なドキュメンテーションを、検索可能な形態で提供しています。これらの資源にはインストール済みアプリケーションのオンラインヘルプのほか、マニュアルページや info ページ、製品添付の Novell/SUSE マニュアルなどが含まれています。

いくつかのアプリケーションにおける個別のヘルプパッケージ

YaST から新しいソフトウェアをインストールすると、ソフトウェアのドキュメンテーションが (多くの場合) 自動でインストールされ、お使いのデスクトップのヘルプセンターからアクセスできるようになります。しかしながら、GIMP などのアプリケーションでは、YaST から個別にインストール可能なヘルプパッケージが提供されていて、ヘルプセンターに統合されない場合もあります。

35.1 ドキュメンテーションディレクトリ

お使いのインストール済みLinuxシステムにおいて、伝統的にドキュメンテーション用のディレクトリとして使用されているのが `/usr/share/doc` です。通常、このディレクトリにはお使いのシステムにインストールされているパッケージについて、それらの情報やリリースノートなどが含まれています。

注記: インストール済みパッケージに依存するコンテンツ

Linux の世界では、多くのマニュアルやその他の種類のドキュメンテーションが、ソフトウェアと同様にパッケージとして提供されています。 `/usr/share/docs` 内にどれだけの情報が存在するのかはパッケージ (およびドキュメンテーション) のインストールを行なうかどうかで決まります。上記のディレクトリに必要なサブディレクトリが見つからない場合は、お使いのシステムに必要なパッケージがインストールされているかどうかを確認し、必要であればインストールを行なってください。

35.1.1 Novell/SUSE マニュアル

Novell/SUSE では、様々な言語に対応した HTML 版および PDF 版のドキュメンテーションを提供しています。 `manual` サブディレクトリには、お使いの製品に対応した Novell/SUSE マニュアルのうち、HTML 版がインストールされています。お使いの製品で利用可能なドキュメンテーションの概要については、各マニュアルの序文をお読みください。

また、複数の言語をインストールしている場合、 `/usr/share/doc/manual` には複数の言語に対応したマニュアルがインストールされます。Novell/SUSE

マニュアルについても、両方のデスクトップのヘルプセンターに対応した HTML 版のマニュアルがインストールされます。ご利用のインストールメディアのどの場所に PDF 版や HTML 版のマニュアルが存在するかについての情報は、openSUSE のリリースノートをお読みください。インストール済みのシステムの場合、リリースノートは `/usr/share/doc/release-notes/` 以下にインストールされますし、製品別の Web サイト (<http://www.novell.com/documentation/>) (英語) から読むこともできます。

35.1.2 HOWTO

お使いのシステムに `howto` パッケージをインストールしている場合、`/usr/share/doc` ディレクトリには `howto` というサブディレクトリが存在しているはずです。このディレクトリには Linux ソフトウェアを設定したり操作したりする際に必要な、様々な作業向けのドキュメンテーションが含まれています。

35.1.3 パッケージのドキュメンテーション

`packages` ディレクトリには、お使いのシステムにインストールされたソフトウェアパッケージに同梱されているドキュメンテーションが存在しています。パッケージごとに `/usr/share/doc/packages/パッケージ名` のディレクトリがあり、そのディレクトリ内にドキュメンテーションが配置されます。また、このディレクトリには各パッケージの `README` ファイルのほか、使用例や設定例、その他の追加スクリプトなどが含まれます。それぞれ `/usr/share/doc/packages` ディレクトリ以下に存在する可能性のあるファイルについて、下記に説明を示します。これらのファイルはあくまでも一例で、全てのパッケージに存在するというわけではありません。

AUTHORS

主な開発者の一覧が書かれています。

BUGS

既知のバグや不具合を記しています。このファイルには、Bugzilla の Web ページへのリンクが書かれている場合もあります。

CHANGES , ChangeLog

バージョン間での変更点の概要を記しています。この概要は主に開発者向けに詳しい情報になっています。

COPYING , LICENSE

ライセンス情報を記しています。

FAQ

メーリングリストやニュースグループなどで集められた、よくある質問とその回答を記しています。

INSTALL

お使いのシステムに対してパッケージをインストールするための手順を記しています。ただし上記のファイルを読むことができるということは、既にそのパッケージをインストール済みであることになるため、特にこのファイルを読む必要はありません。

README, README.*

ソフトウェアに関する一般的な情報を記しています。たとえばソフトウェアの目的や使用方法などが書かれています。

TODO

その時点では実装されていないが、今後実装する予定である機能などを記しています。

MANIFEST

ファイルの一覧とその概要説明を記しています。

NEWS

このバージョンで何が新しくなったのかを記しています。

35.2 マニュアルページ

マニュアルページはLinux システムでは必要不可欠なものです。マニュアルページはコマンドに関する使用方法を説明しているほか、利用可能な全てのオプションやパラメータについても説明が書かれています。マニュアルページを表示するには、manに続いてコマンド名を入力します。たとえばman lsのように入力します。

マニュアルページはシェル内に直接出力されます。それぞれ **Page ↑** と **Page ↓** のキーで上下に移動することができるほか、**Home** と **End** でそれぞれ文書の冒頭と文末に移動することもできます。閲覧を終了するには **Q** キーを押してください。man コマンド自身についてより詳しい使い方を知りたい場合は、`man man` と入力してください。マニュアルページは表35.1「マニュアルページ—カテゴリとその説明」(653 ページ) に示しているようなカテゴリに分類されています (man コマンドのマニュアルページから引用しています):

表 35.1 マニュアルページ—カテゴリとその説明

番号	説明
1	実行可能なプログラムやシェルコマンド
2	システムコール (カーネルが提供する機能)
3	ライブラリコール (プログラムライブラリが提供する機能)
4	特殊ファイル (一般に /dev 内に存在するもの)
5	ファイルの書式と慣習 (/etc/fstab など)
6	ゲーム
7	その他 (マクロのパッケージや慣習など)。たとえば <code>man(7)</code> , <code>groff(7)</code> などがあります
8	システム管理コマンド (通常は root が使用するコマンド)
9	カーネルルーチン (非標準)

それぞれのマニュアルページは複数のパートから構成され、それぞれ *NAME* (名前), *SYNOPSIS* (書式), *DESCRIPTION* (説明), *SEE ALSO* (関連項目), *LICENSING* (ライセンス), *AUTHOR* (著者) などの項目があります。コマンドの種類によっては、追加のパートが存在する 場合もあります。

35.3 info ページ

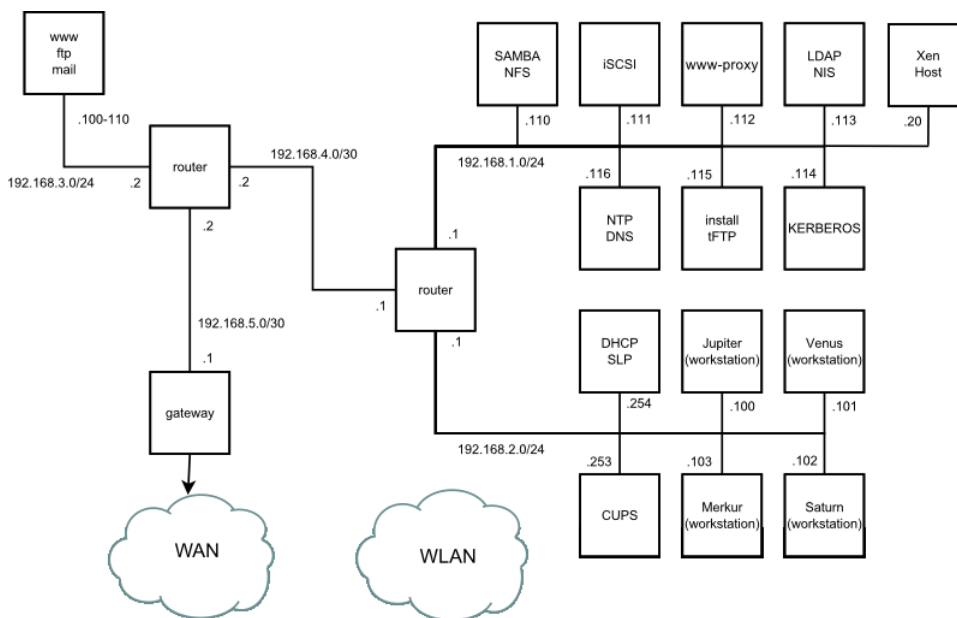
info ページはお使いのシステムにおいて、もう 1 つの重要な情報源です。通常はマニュアルページよりも詳しい説明が書かれています。特定のコマンドについて info ページを閲覧するには、info に続いてコマンド名を入力します。たとえば info ls のように入力します。info ページはシェル内に直接出力され、「nodes. (ノード)」と呼ばれる複数のセクションから構成されます。Space で前に、<— で後ろにそれぞれ移動します。また、ノード内では Page ↑ や Page ↓ を利用することもできますが、Space や Space を押さないと次のノードや直前のノードに移動することができません。閲覧を終了するには、Q を押してください。なお、マニュアルページと info ページはそれぞれ独立して提供されるもので、いずれか一方しか用意されていない場合があります。

35.4 openSUSE Wiki

openSUSE システムの様々な側面について情報を得るには、<http://ja.opensuse.org> にある Wiki をご利用ください。Wiki のページは自由に追加／編集することができます。追加や編集にあたっては、あらかじめユーザ登録を行なってログインする必要があります。ログイン後、編集リンクを押して編集を行なってください。

A

下記に示すサンプルネットワークは、openSUSE® ドキュメンテーションにおけるネットワーク関連の記述で利用されます。



GNU ライセンス

本付録には、GNU General Public License バージョン 2 と GNU Free Documentation License バージョン 1.2 を掲載しています。

なお、八田真行氏 (mhatta@gnu.org) [<mailto:mhatta@gnu.org>] による各ライセンスの日本語訳を併記しています。

ただし、各日本語訳は 非公式な ものであり、フリーソフトウェア財団 (the Free Software Foundation) によって発表された ものではないことにご注意ください。法的に有効なものは常に原文 (つまり英語版) 側であり、日本語訳は各ライセンスをよりよく理解する支援を行なう目的で 作成されたもの、という扱いです。

また、日本語訳は DocBook (novdoc) に合わせて段落を分割しているほか、引用符のタグ化 ("blah" -> <quote>blah</quote>) と リンクの生成 (ulink) を行なっています。

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The 「Program」, below, refers to any such program or work, and a 「work based on the Program」 means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 「modification」.) Each licensee is addressed as 「you」.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and 「any later version」, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

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 an idea of what it does.
Copyright (C) yyyy 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., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a 「copyright disclaimer」 for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [<http://www.fsf.org/licenses/lgpl.html>] instead of this License.

GNU 一般公衆利用許諾契約書 (日本語訳)

バージョン 2, 1991年6月

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

この利用許諾契約書を、一字一句そのままに複製し頒布することは許可する。しかし変更は認めない。

はじめに

ソフトウェア向けライセンスの大半は、あなたがそのソフトウェアを共有したり変更したりする自由を奪うように設計されています。対照的に、GNU 一般公衆利用許諾契約書は、あなたがフリーソフトウェアを共有したり変更したりする自由を保証する--すなわち、ソフトウェアがそのユーザすべてにとってフリーであることを保証することを目的としています。この一般公衆利用許諾契約書はフリーソフトウェア財団のソフトウェアのほとんどに適用されており、また GNU GPL を適用すると決めたフリーソフトウェア財団以外の作者によるプログラムにも適用されています(いくつかのフリーソフトウェア財団のソフトウェアには、GNU GPL ではなく GNU ライブラリー一般公衆利用許諾契約書が適用されています)。あなたもまた、ご自分のプログラムに GNU GPL を適用することが可能です。

私たちがフリーソフトウェアと言うとき、それは利用の自由について言及しているものであって、価格は問題にしていません。私たちの一般公衆利用許諾契約書は、あなたがフリーソフトウェアの複製物を頒布する自由を保証するよう設計されています(希望に応じてその種のサービスに手数料を課す自由も保証されます)。また、あなたがソースコードを受け取るか、あるいは望めばそれを入手することが可能であるということ、あなたがソフトウェアを変更し、その一部を新たなフリーのプログラムで利用できるということ、そして、以上で述べたようなことができる ということがあなたに知られるということも保証されます。

あなたの権利を守るため、私たちは誰かがあなたの有するこれらの権利を否定することや、これらの権利を放棄するよう要求することを禁止するという制限を加える 必要があります。よって、あなたがソフトウェアの複製物を頒布したりそれを変更したりする場合には、そういった制限のためにあなたにある種の責任が発生する ことになります。

例えば、あなたがフリーなプログラムの複製物を頒布する場合、有料か無料に関わらず、あなたは自分が有する権利を全て受領者に与えなければなりません。また、あなたは彼らもソースコードを受け取るか手に入れることができるよう 保証しなければなりません。そして、あなたは彼らに対して以下で述べる条件を 示し、彼らに自らの持つ権利について知らしめるようにしなければなりません。

私たちはあなたの権利を二段階の手順を踏んで保護します。(1) まずソフトウェアに対して著作権を主張し、そして (2) あなたに対して、ソフトウェアの複製や頒布または改変についての法的な許可を 与えるこの契約書を提示します。

また、各作者や私たちを保護するため、私たちはこのフリーソフトウェアには何の保証も無いということを誰もが確実に理解するようにし、またソフトウェアが誰か他人によって改変され、それが次々と頒布されていったとしても、その受領者は彼らが手に入れたソフトウェアがオリジナルのバージョンでは無いこと、そして原作者の名声は他人によって持ち込まれた可能性のある問題によって影響されることがないということを周知させたいと思います。

最後に、ソフトウェア特許がいかなるフリーのプログラムの存在にも不断の脅威を投げかけていますが、私たちは、フリーなプログラムの再頒布者が個々に特許ライセンスを取得することによって、事実上プログラムを独占的にしてしまうという 危険を避けたいと思います。こういった事態を予防するため、私たちはいかなる特許も誰も自由に利用できるようライセンスされるか、全くライセンスされないかの どちらかでなければならないことを明確にしました。

(訳注: 本契約書で「独占的(proprietary)」とは、ソフトウェアの利用や再頒布、改変が禁止されているか、許可を得ることが必要とされているか、あるいは厳しい制限が課せられていて自由にそうすることが事実上できなくなっている状態のことを指す。詳しくは <http://www.gnu.org/philosophy/categories.ja.html#ProprietarySoftware> [<http://www.gnu.org/philosophy/categories.ja.html#ProprietarySoftware>] を参照せよ。)

複製や頒布、改変についての正確な条件と制約を以下で述べていきます。

複製、頒布、改変に関する条件と制約

0. この利用許諾契約書は、そのプログラム(またはその他の著作物)をこの一般公衆利用許諾契約書の定める条件の下で頒布できる、という告知が著作権者によって記載されたプログラムまたはその他の著作物全般に適用される。以下では、「プログラム」とはそのようにしてこの契約書が適用されたプログラムや著作物全般を意味し、また「プログラムを基にした著作物」とは「プログラム」やその他著作権法の下で派生物と見なされるもの全般を指す。すなわち、「プログラム」かその一部を、全く同一のままか、改変を加えたか、あるいは他の言語に翻訳された形で含む著作物のことである(「改変」という語の本来の意味からはずれるが、以下では翻訳も改変の一種と見なす)。それぞれの契約者は「あなた」と表現される。

複製や頒布、改変以外の活動はこの契約書ではカバーされない。それらはこの契約書の対象外である。「プログラム」を実行する行為自体に制限はない。また、そのような「プログラム」の出力結果は、その内容が「プログラム」を基にした著作物を構成する場合のみこの契約書によって保護される(「プログラム」を実行したことによって作成されたということとは無関係である)。このような線引きの妥当性は、「プログラム」が何をするのかに依存する。

1. それぞれの複製物において適切な著作権表示と保証の否認声明(disclaimer of warranty)を目立つよう適切に掲載し、またこの契約書および一切の保証の不在に触れた告知すべてをそのまま残し、そしてこの契約書の複製物を「プログラム」のいかなる受領者にも「プログラム」と共に頒布する限り、あなたは「プログラム」のソースコードの複製物を、あなたが受け取った通りの形で複製または頒布することができる。媒体は問わない。

あなたは、物理的に複製物を譲渡するという行為に関して手数料を課しても良いし、希望によっては手数料を取って交換における保護の保証を提供しても良い。

2. あなたは自分の「プログラム」の複製物かその一部を改変して「プログラム」を基にした著作物を形成し、そのような改変点や著作物を上記第1節の定める条件の下で複製または頒布することができる。ただし、そのためには以下の条件すべてを満たしていなければならない:

- a) あなたがそれらのファイルを変更したということと変更した日時が良く分かるよう、改変されたファイルに告示しなければならない。
- b) 「プログラム」またはその一部を含む著作物、あるいは「プログラム」かその一部から派生した著作物を頒布あるいは発表する場合には、その全体をこの契約書の条件に従って第三者へ無償で利用許諾しなければならない。
- c) 改変されたプログラムが、通常実行する際に対話的にコマンドを読むようになっているならば、そのプログラムを最も一般的な方法で対話的に実行する際、適切な著作権表示、無保証であること(あるいはあなたが保証を提供するという)、ユーザがプログラムをこの契約書で述べた条件の下で頒布することができるということ、そしてこの契約書の複製物を閲覧するにはどうしたらよいかというユーザへの説明を含む告知が印刷されるか、あるいは画面に表示されるようにしなければならない(例外として、「プログラム」そのものは対話的であっても通常そのような告知を印刷しない場合には、「プログラム」を基にしたあなたの著作物にそのような告知を印刷させる必要はない)。

以上の必要条件是全体としての改変された著作物に適用される。著作物の一部が「プログラム」から派生したものではないと確認でき、それら自身別の独立した著作物であると合理的に考えられるならば、あなたがそれらを別の著作物として分けて頒布する場合、そういった部分にはこの契約書とその条件は適用されない。しかし、あなたが同じ部分を「プログラム」を基にした著作物全体の一部として頒布するならば、全体としての頒布物は、この契約書が課す条件に従わなければならない。というのは、この契約書が他の契約者に与える許可は「プログラム」丸ごと全体に及び、誰が書いたかは関係なく各部分のすべてを保護するからである。

よって、すべてあなたによって書かれた著作物に対し、権利を主張したりあなたの権利に異議を申し立てることはこの節の意図するところではない。むしろ、その趣旨は「プログラム」を基にした派生物ないし集合著作物の頒布を管理する権利を行使することにある。

また、「プログラム」を基にしていないその他の著作物を「プログラム」(あるいは「プログラム」を基にした著作物)と一緒に集めただけのものを一巻の保管装置ないし頒布媒体に収めても、その他の著作物までこの契約書が保護する対象になるということにはならない。

3. あなたは上記第1節および2節の条件に従い、「プログラム」(あるいは第2節における派生物)をオブジェクトコードないし実行形式で複製または頒布することができる。ただし、その場合あなたは以下のうちどれか一つを実施しなければならない:

- a) 著作物に、『プログラム』に対応した完全かつ機械で読み取り可能なソースコードを添付する。ただし、ソースコードは上記第1節および2節の条件に従いソフトウェアの交換で習慣的に使われる媒体で頒布しなければならない。あるいは、
- b) 著作物に、いかなる第三者に対しても、『プログラム』に対応した完全かつ機械で読み取り可能なソースコードを、頒布に要する物理的コストを上回らない程度の手数料と引き換えに提供する旨述べた少なくとも3年間は有効な書面になった申し出を添える。ただし、ソースコードは上記第1節および2節の条件に従いソフトウェアの交換で習慣的に使われる媒体で頒布しなければならない。あるいは、
- c) 対応するソースコード頒布の申し出に際して、あなたが得た情報を一緒に引き渡す(この選択肢は、営利を目的としない頒布であって、かつあなたが上記小節bで指定されているような申し出と共にオブジェクトコードあるいは実行形式のプログラムしか入手していない場合に限り許可される)。

著作物のソースコードとは、それに対して改変を加える上で好ましいとされる著作物の形式を意味する。ある実行形式の著作物にとって完全なソースコードとは、それが含むモジュールすべてのソースコード全部に加え、関連するインターフェース定義ファイルのすべてとライブラリのコンパイルやインストールを制御するために使われるスクリプトをも加えたものを意味する。しかし特別な例外として、そのコンポーネント自体が実行形式に付随するのでは無い限り、頒布されるものの中に、実行形式が実行されるオペレーティ

ングシステムの主要なコンポーネント(コンパイラやカーネル等)と通常一緒に(ソースかバイナリ形式のどちらかで)頒布されるものを含んでいる必要はないとする。

実行形式またはオブジェクトコードの頒布が、指定された場所からコピーするためのアクセス手段を提供することで為されるとして、その上でソースコードも同等のアクセス手段によって同じ場所からコピーできるようになっているならば、第三者がオブジェクトコードと一緒にソースも強制的にコピーせられるようになっていなくともソースコード頒布の条件を満たしているものとする。

4. あなたは「プログラム」を、この契約書において明確に提示された行為を除き複製や改変、サブライセンス、あるいは頒布してはならない。他に「プログラム」を複製や改変、サブライセンス、あるいは頒布する 企てはすべて無効であり、この契約書の下でのあなたの権利を自動的に終結させる ことになろう。しかし、複製物や権利をこの契約書に従ってあなたから得た人々に関しては、そのような人々がこの契約書に完全に従っている限り彼らのライセンスまで 終結することはない。

5. あなたはこの契約書を受諾する必要は無い。というのは、あなたはこれに署名していないからである。しかし、この契約書以外にあなたに対して「プログラム」やその派生物を改変または頒布する許可を与えるものは存在しない。これらの行為は、あなたがこの契約書を受け入れない限り法によって 禁じられている。そこで、「プログラム」(あるいは「プログラム」を基にした著作物全般)を改変ないし頒布することにより、あなたは自分がそのような行為を行うためにこの契約書を受諾したということ、そして「プログラム」とそれに基づく著作物の複製や頒布、改変について この契約書が課す制約と条件をすべて受け入れたということを示したものと見なす。

6. あなたが「プログラム」(または「プログラム」を基にした著作物全般)を再頒布するたびに、その受領者は元々のライセンス許可者から、この契約書で指定された条件と制約の下で「プログラム」を複製や頒布、あるいは改変する許可を自動的に得るものとする。あなたは、受領者がここで認められた権利を行使することに関してこれ以上他のいかなる制限も課してはならない。あなたには、第三者がこの契約書に従うことを強制する責任はない。

7. 特許侵害あるいはその他の理由(特許関係に限らない)から、裁判所の判決あるいは申し立ての結果としてあなたに(裁判所命令や契約などにより)このライセンスの条件と矛盾する制約が課された場合でも、あなたがこの契約書の条件を免除される わけではない。もしこの契約書の下であなたに課せられた責任と他の関連する責任を同時に満たすような形で頒布できないならば、結果としてあなたは「プログラム」を頒布することが全くできないということである。例えば特許ライセンスが、あなたから直接間接を問わずコピーを受け取った人が誰でも「プログラム」を使用料無料で再頒布することを認めていない場合、あなたがその制約とこの契約書を両方とも満たすには「プログラム」の頒布を完全に中止するしかないだろう。

この節の一部分が特定の状況の下で無効ないし実施不可能な場合でも、節の残りの部分は適用されるよう意図されている。その他の状況では節が全体として適用されるよう 意図されている。

特許やその他の財産権を侵害したり、そのような権利の主張の効力に異議を唱えたりするようあなたを誘惑することがこの節の目的ではない。この節には、人々によってライセンス慣行として実現されてきた、フリーソフトウェア頒布のシステムの完全性を 護るといふ目的しかない。多くの人々が、フリーソフトウェアの頒布システムが首尾一貫して適用されているという信頼に基づき、このシステムを通じて頒布される多様なソフトウェアに寛大な貢献をしてきたのは事実であるが、人がどのようなシステムを通じてソフトウェアを頒布したいと思うかはあくまでも作者/寄与者次第であり、あなたが選択を押しつけることはできない。

この節は、この契約書のこの節以外の部分の一掃になると考えられるケースを 徹底的に明らかにすることを目的としている。

8. 「プログラム」の頒布や利用が、ある国においては特許または著作権が主張されたインターフェースのいずれかによって制限される場合、この「プログラム」にこの契約書を適用した元の著作権者は、そういった国々を排除した明確な地理的頒布制限を加え、そこで排除されていない国の中やそれらの国々の間でのみ頒布が許可されるようにしても構わない。その場合、そのような制限はこの契約書本文で書かれているのと同様に見なされる。

9. フリーソフトウェア財団は、時によって改訂または新版の一般公衆利用許諾書を 発表することができる。そのような新版は現在のバージョンとその精神においては 似たものになるだろうが、新たな問題や懸念を解決するため細部では異なる可能性がある。

それぞれのバージョンには、見分けが付くようにバージョン番号が振られている。「プログラム」においてそれに適用されるこの契約書のバージョン番号が指定されていて、更に「それ以降のいかなるバージョン(any later version)」も適用して良いとなっていた場合、あなたは従う条件と制約として、指定のバージョンか、フリーソフトウェア財団によって発行された指定のバージョン以降の版のどちらかのどちらかを選ぶことができる。「プログラム」でライセンスのバージョン番号が指定されていないならば、あなたは今までにフリーソフトウェア財団から発行されたバージョンの中から好きに選んで構わない。

10. もしあなたが「プログラム」の一部を、その頒布条件がこの契約書と異なる他のフリーなプログラムと統合したいならば、作者に連絡して許可を求めよう。フリーソフトウェア財団が著作権を保有するソフトウェアについては、フリーソフトウェア財団に連絡せよ。私たちは、このような場合のために特別な例外を 設けることもある。私たちが決定を下すにあたっては、私たちのフリーソフトウェアの 派生物すべてがフリーな状態に保たれるということと、一般的にソフトウェアの共有と 再利用を促進するという二つの目標を 規準に検討されるであろう。

無保証について

11. 「プログラム」は代価無しに利用が許可されるので、適切な法が認める限りにおいて、「プログラム」に関するいかなる保証も存在しない。書面で別に述べる場合を除いて、著作権者、またはその他の団体は、「プログラム」を、表明されたか言外には問わず、商業的適性を 保証するほのめかしやある特定の目的への適合性に限られない)を含む一切の 保証無しに「あるがまま」で提供する。「プログラム」の質と性能に関するリスクのすべてはあなたに帰属する。「プログラム」に欠陥があると判明した場合、あなたは必要な保守点検や補修、修正に要する コストのすべてを引き受けることになる。

12. 適切な法が書面での同意によって命ぜられない限り、著作権者、または上記で許可されている通りに「プログラム」を改変または再頒布した その他の団体は、あなたに対して「プログラム」の利用ないし 利用不能で生じた通常損害や特別損害、偶発損害、間接

損害(データの消失や 不正確な処理、あなたか第三者が被った損失、あるいは「プログラム」が他のソフトウェアと一緒に動作しないという不具合などを含むがそれらに限らない)に一切の責任を負わない。そのような損害が生ずる可能性について彼らが忠告 されていたとしても同様である。

条件と制約終わり

以上の条項をあなたの新しいプログラムに適用する方法

あなたが新しいプログラムを開発したとして、公衆によってそれが利用される可能性を最大にしたいなら、そのプログラムをこの契約書の条項に従って誰でも 再頒布あるいは変更できるようフリーソフトウェアにするのが最善です。

そのためには、プログラムに以下のような表示を添付してください。その場合、保証が~~排除~~されているということを最も効果的に伝えるために、それぞれのソースファイルの冒頭に表示を添付すれば最も安全です。少なくとも、「著作権表示」という行と全文がある場所へのポイントだけは 各ファイルに含めて置いてください。

```
one line to give the program's name and an idea of what it does.  
Copyright (C) yyyy 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., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

(訳)

プログラムの名前と、それが何をするかについての簡単な説明。 *Copyright (C) 西暦年 作者の名前*

このプログラムはフリーソフトウェアです。あなたはこれを、フリーソフトウェア財団によって発行された GNU 一般公衆利用許諾契約書 (バージョン2か、希望によってはそれ以降のバージョンのうちどれか) の定める条件の下で再頒布または改変することができます。

このプログラムは有用であることを願って頒布されますが、*全くの無保証* です。 商業可能性の保証や特定の目的への適合性は、言外に示されたものも含め全く存在しません。詳しくはGNU 一般公衆利用許諾契約書をご覧ください。

あなたはこのプログラムと共に、GNU 一般公衆利用許諾契約書の複製物を一部受け取ったはずです。もし受け取っていないければ、フリーソフトウェア財団まで請求してください(宛先は the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA)。

電子ないし紙のメールであなたに問い合わせる方法についての情報も書き加えましょう。

プログラムが対話的なものならば、対話モードで起動した際に出力として 以下のような短い告知が表示されるようにしてください:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details  
type 'show w'. This is free software, and you are welcome
```

```
to redistribute it under certain conditions; type `show c'
for details.
```

(訳)

Gnomovision バージョン 69, Copyright (C) 西暦年 作者の名前
Gnomovision は*全くの無保証*で提供されます。詳しくは
`show w' とタイプして下さい。
これはフリーソフトウェアであり、ある条件の下で再頒布することが
奨励されています。詳しくは `show c' とタイプして下さい。

ここで、仮想的なコマンド `show w' と `show c' は一般公衆利用許諾契約書の適切な部分を表示するようになっていなければなりません。もちろん、あなたが使うコマンドを `show w' や `show c' と呼ぶ必然性はありませんので、あなたのプログラムに合わせてマウスのクリックやメニューのアイテムにしても結構です。

また、あなたは、必要ならば(プログラマーとして働いていたら)あなたの雇用主、あるいは場合によっては学校から、そのプログラムに関する「著作権放棄声明(copyright disclaimer)」に署名してもらうべきです。以下は例です、名前を変えてください:

```
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
```

(訳)

Yoyodyne社はここに、James Hackerによって書かれた
プログラム `Gnomovision' (コンパイラへ通すプログラム)
に関する一切の著作権の利益を放棄します。

Ty Coon氏の署名、1989年4月1日
Ty Coon、副社長

この一般公衆利用許諾契約書では、あなたのプログラムを独占的なプログラムに統合することを認めていません。あなたのプログラムがサブルーチンライブラリならば、独占的なアプリケーションとあなたのライブラリをリンクすることを許可したほうがより便利であると考えられるかもしれません。もしこれがあなたの望むことならば、この契約書の代わりに GNU ライブラリ一般公衆利用許諾契約書 [<http://www.fsf.org/licenses/lgpl.html>] を適用してください。

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of 「copyleft」, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any

textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The 「Document」, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as 「you」. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A 「Modified Version」 of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A 「Secondary Section」 is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The 「Invariant Sections」 are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The 「Cover Texts」 are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A 「Transparent」 copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not 「Transparent」 is called 「Opaque」.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The 「Title Page」 means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, 「Title Page」 means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section 「Entitled XYZ」 means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as 「Acknowledgements」, 「Dedications」, 「Endorsements」, or 「History」.) To 「Preserve the Title」 of such a section when you modify the Document means that it remains a section 「Entitled XYZ」 according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled 「History」, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled 「History」 in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the 「History」 section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled 「Acknowledgements」 or 「Dedications」, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled 「Endorsements」. Such a section may not be included in the Modified Version.
- N. Do not retile any existing section to be Entitled 「Endorsements」 or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled 「Endorsements」, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled 「History」 in the various original documents, forming one section Entitled 「History」 ; likewise combine any sections Entitled 「Acknowledgements」 , and any sections Entitled 「Dedications」 . You must delete all sections Entitled 「Endorsements」 .

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled 「Acknowledgements」 , 「Dedications」 , or 「History」 , the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation 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. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License 「or any later version」 applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU フリー文書利用許諾契約書 (日本語訳)

バージョン 1.2, 2002年11月

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

この利用許諾契約書を、一字一句そのままに複製し頒布することは許可する。しかし変更は認めない。

はじめに

この利用許諾契約書の目的は、この契約書が適用されるマニュアルや教科書、その他機能本位で実用的な文書を(無料ではなく)自由という意味で“フリー”とすること、すなわち、改変の有無あるいは目的の営利非営利を問わず、文書を複製し再頒布する自由をすべての人々に効果的に保証することです。加えてこの契約書により、著者や出版者が自分たちの著作物に対して相応の敬意と賞賛を得る手段も保護されます。また、他人が行った改変に対して責任を負わずに済むようになります。

この利用許諾契約書は「コピーレフト」的なライセンスの一つであり、この契約書が適用された文書から派生した著作物は、それ自身もまた原本と同じ意味でフリーでなければなりません。この契約書は、フリーソフトウェアのために設計されたコピーレフトなライセンスであるGNU一般公衆使用許諾契約書を補足するものです。

(訳注: コピーレフト(copyleft)の概念については <http://www.gnu.org/copyleft/copyleft.ja.html> [<http://www.gnu.org/copyleft/copyleft.ja.html>] を参照せよ)

この利用許諾契約書は、フリーソフトウェア用のマニュアルに適用することを目的として書かれました。フリーソフトウェアはフリーな文書を必要としており、フリーなプログラムはそのソフトウェアが保証するのと同じ自由を提供するマニュアルと共に頒布されるべきだからです。しかし、この契約書の適用範囲はソフトウェアのマニュアルに留まりません。対象となる著作物において扱われる主題が何であれ、あるいはそれが印刷された書籍として出版されるか否かに関わらず、この契約書は文字で書かれたいかなる著作物にも適用することが可能です。私たちとしては、主にこの契約書を解説や参照を目的とする著作物に適用することをお勧めします。

この利用許諾契約書の適用範囲と用語の定義

著作物がこの利用許諾契約書の定める条件の下で頒布される旨の告知を、著作権者がその中に書いたすべてのマニュアルあるいはその他の著作物は、いかなる媒体上にあってもこの契約書の適用対象となる。そのような告知を置くことで、全世界において、著作権使用料を必要とせず、許可の存続期間を限定されることなく、この契約書の中で述べられている条件の下で当該著作物を利用できるという許可を与えることとする。以下において、「文書(Document)」とはそのような告知が記載されたマニュアルないし著作物すべてを指す。公衆の一員ならば誰でも契約の当事者となることができ、この契約書中では「あなた」と表現される。あなたは、著作権法の下で許可を必要とするような方法で著作物を複製や改変、あるいは頒布することにより、この契約書を受諾することになる。

「文書」の「改変版(Modified Version)」とは、一字一句忠実に複製したか、あるいは改変や他言語への翻訳を行ったかどうかに関わらず、その「文書」の全体あるいは一部分を含む著作物すべてを意味する。

「補遺部分(Secondary Section)」とは、「文書」中でその旨指定された補遺ないし本文に先だって前付けとして置かれる一部分であり、「文書」の出版者あるいは著者と、「文書」全体の主題(あるいはそれに関連する事柄)との関係のみを論じ、全体としての主題の範囲に直接属する内容を全く含まないものである(たとえば、「文書」の一部が数学の教科書だった場合、補遺部分では数学について何も解説してはならない)。補遺部分で扱われる関係は、その主題あるいは関連する事柄との歴史的なつながりのことかも知れないし、それらに関する法的、商業的、哲学的、倫理的、あるいは政治的立場についてかも知れない。

「変更不可部分(Invariant Sections)」とは補遺部分の一種で、それらが変更不可部分であることが、「文書」をこの利用許諾契約書の下で発表する旨述べた告知中においてその部分の題名と共に明示されているものである。ある部分が上記のような「補遺」性の定義

にそぐわない場合は、その部分を「変更不可」として指定することは認められない。「文書」は、変更不可部分を全く含まなくても良い。「文書」において変更不可部分が全く指定されていないければ、その「文書」に変更不可部分は存在しないということである。

「カバーテキスト(Cover Texts)」とは、「文書」がこの利用許諾契約書の指定する条件の下で発表される旨述べた告知において、「表カバーテキスト」あるいは「裏カバーテキスト」として列挙された短い文章のことを指す。表カバーテキストは最大で5語、裏カバーテキストは最大で25語までとする。

「文書」の「透過的」複製物とは、機械による読み取りが可能な「文書」の複製物のことを指す。透過的な複製物の文書形式は、その仕様が一般の人々に入手可能で、「文書」の内容を一般的なテキストエディタ、または(画面で構成される画像ならば)一般的なペイントプログラム、あるいは(図面ならば)いくつかの広く入手可能な製図エディタで簡単に改訂するのに適しており、なおかつテキストフォーマットへの入力に適する(あるいはテキストフォーマットへの入力に適する諸形式への自動的な変換に適する)ものでなければならない。透過的なファイル形式への複製であっても、マークアップ、あるいはマークアップの不在が読者によるそれ以降の改変をわざと邪魔し阻害するように仕組まれたものは透過的であるとは見做されない。ある画像形式が、相当量のテキスト文章を表現するために使われた場合、それは透過的ではない。透過的ではない複製は「非透過的」複製と呼ばれる。

透過的複製に適した形式の例としては、マークアップを含まないプレーンな ASCII形式、Texinfo入力形式、LaTeX入力形式、一般に入手可能なDTDを用いたSGMLあるいはXML、または人間による改変を想定して設計された、標準に準拠したシンプルなHTMLやPostScript、PDFなどが挙げられる。透過的な画像形式の例には、PNGやXCF、JPGが含まれる。非透過な形式としては、独占的なワードプロセッサでのみ閲覧編集できる独占的なファイル形式、普通には入手できないDTDまたは処理系を使用したSGMLやXML、ある種のワードプロセッサが生成する、出力のみを目的とした機械生成のHTMLやPostScript、PDFなどが含まれる。

「題扉(Title Page)」とは、印刷された書籍に於いては、実際の表紙自身のみならず、この利用許諾契約書が表紙に掲載することを義務づける文章や図などを、読みやすい形で載せるのに必要なだけの、表紙に引き続く数ページをも意味する。表紙に類するものが無い形式で発表される著作物においては、「題扉」とは本文の始まりに先だって、その著作物の題名が最も目立つ形で現れる場所の近くに置かれる文章のことを指す。

「XYZと題された(Entitled XYZ)」部分とは、「文書」において「XYZ」と名付けられた一部分であり、その題名は正確に「XYZ」であるか、「XYZ」を他の言語に翻訳した上でその後ろに「XYZ」をそのまま括弧で括ったものを含む記述のどちらかである(ここで「XYZ」とは、この利用許諾契約書において以下で言及される特定の部分名を意味している。例えば「謝辞(Acknowledgements)」、「献辞(Dedications)」、「推薦の辞(Endorsements)」、「履歴(History)」)。あなたが「文書」を改変する場合、余のような部分の「題名を保存する(Preserve the Title)」とは、「XYZと題された」部分として、ここでの定義に従い題名を残すということである。

「文書」は、「保証否認警告(Warranty Disclaimers)」を、この利用許諾契約書が「文書」に適用されると述べた告知の次に含んでも良い。この種の保証否認警告は、この契約書からの言及という形で利用条件に含まれるものと解されるか、保証の否認に関することについてののみ有効とする。こういった保証否認警告で示しうるその他のいかなる含意も無効であり、この契約書の機能には何ら影響を持たない。

逐語的に忠実な複製

この利用許諾契約書、著作権表示、この契約書が「文書」に適用される旨述べた告知の三つがすべての複製物に複製され、かつあなたがこの契約書で指定されている以外のいかなる条件も追加しない限り、あなたはこの「文書」を、商用であるか否かを問わずいかなる形で複製頒布することができる。あなたは、あなたが作成あるいは頒布する複製物に対して、閲覧や再複製を技術的な手法によって妨害、規制してはならない。しかしながら、複製と引き換えに代価を得てもかまわない。あなたが相当量の複製物を頒布する際には、本契約書第3頁で指定される条件にも従わなければならない。

またあなたは、上記と同じ条件の下で、複製物を貸与したり複製物を公に開示することができる。

大量の複製

もしあなたが、「文書」の印刷された(あるいは通常は印刷された表紙を持つ媒体における)複製物を100部を超えて出版し、また「文書」の利用許諾告知がカバーテキストの掲載を要求している場合には、指定されたすべてのカバーテキストを、表カバーテキストは表表紙に、裏カバーテキストは裏表紙に、はっきりと読みやすい形で載せた表紙の中に複製物本体を綴じ込まなければならない。また、両方の表紙において、それらの複製物の出版者としてのあなたをはっきりとかつ読みやすい形で確認できなければならない。表表紙では「文書」の完全な題名を、題名を構成するすべての語が等しく目立つようにして、視認可能な形で示さなければならない。それらの情報に加えて、表紙に他の文章や図などを加えることは許可される。表紙のみを変更した複製物は、それが「文書」の題名を保存し上記の条件を満たす限り、ほかの点では逐語的に忠実な複製物として扱われる。

もしどちらかの表紙に要求されるカバーテキストの量が多すぎて読みやすく収めることが不可能ならば、あなたはテキスト先頭の一文(あるいは適切に収まるだけ)を実際の表紙に載せ、続きは隣接したページに載せるべきである。

あなたが「文書」の「非透過的」複製物を100部を超えて出版あるいは頒布する場合、それぞれの非透過な複製物と一緒に機械で読み取り可能な透過的複製物を添付するか、それぞれの非透過な複製物(あるいはそれに付属する文書)中で、公にアクセス可能なコンピュータネットワーク上の所在地を記述しなければならない。その場所には、非透過な複製物と内容的に寸分違わず、余計なものが追加されていない完全な「文書」の透過的複製物が置かれ、またそこから、ネットワークを利用する一般公衆が、一般に標準的と考えられるネットワークツールを使ってダウンロードすることができなければならない。もしあなたが後者の選択肢を選ばなければ、その版の非透過な複製物を公衆に(直接、あるいはあなたの代理人ないし小売業者が)最後に頒布してから最低1年間、その透過的複製物が指定の場所でアクセス可能であり続けることを保証するよう、非透過な複製物の大量頒布を始める際に十分に慎重な手順を踏まなければならない。

これは要望であり必要条件ではないが、「文書」の著者に、「文書」の更新された版をあなたに提供する機会を与えるため、透過非透過を問わず大量の複製物を再頒布し始める前には彼らにきちんと連絡しておいてほしい。

改変

「文書」の改変版を、この利用許諾契約書と細部まで同一の契約の下で発表する限り、すなわち原本の役割を改変版で置き換えた形での頒布と改変を、その複製物を所有するすべての人々に許可する限り、あなたは改変版を上記第2項および第3項が指定する条件の下で複製および頒布することができる。さらに、あなたは改変版において以下のことを行わなければならない。

- A. 題扉にもしあればその他の表紙にも、「文書」および「文書」のそれ以前の版と見分けがつく題名を載せること(もし以前の版があれば、「文書」の「履歴(History)」の部分に列記されているはずである)。もし元の版の出版者から許可を得たならば、以前の版と同じ題名を使っても良い。
- B. 題扉に、改変版における改変を行った1人以上の人物が団体名を列記すること。あわせて元の「文書」の著者として、最低5人(もし5人以下ならばすべて)の主要著者を列記すること。ただし元の著者たちがこの条件を免除した場合は除く。
- C. 題扉に、改変版の出版者名を出版者として記載すること。
- D. 「文書」にあるすべての著作権表示を残すこと。
- E. 他の著作権表示の近くに、あなたの改変に対する適当な著作権表示を追加すること。
- F. 著作権表示のすぐ後に、改変版をこの契約書の条件の下で利用することを公衆に対して許可する告知を含めること。その形式はこの契約書の末尾にある付記で示されている。
- G. 元の「文書」の利用許諾告知に書かれた、変更不可部分の完全な一覧と、要求されるカバーテキストとを、改変版の利用許諾告知でもそのまま残すこと。
- H. この契約書の、変更されていない複製物を含めること。
- I. 「履歴(History)」と題された部分とその題名を保存し、そこに改変版の、少なくとも題名、出版年、新しく変更した部分の著者名、出版者名を、題扉に掲載するのと同じように記載した一項を加えること。もし「文書」中に「履歴」と題された部分が存在しない場合には、「文書」の題名、出版年、著者、出版者を題扉に掲載するのと同じように記載した部分を用意し、上記で述べたような、改変版を説明する一項を加えること。
- J. 「文書」中に、「文書」の透過的複製物への公共的アクセスのために指定されたネットワーク的所在地が記載されていたならば、それを保存すること。同様に、その「文書」の元になった以前の版で指定されていたネットワークの所在地も載っていたならば、それも保存すること。これらの情報は「履歴(History)」の部分に置いても良い。ただし、それが「文書」自身より少なくとも4年前に出版された著作物の情報であったり、あるいは改変版が参考にしての版の元々の出版者から許可を得たならば、その情報を削除してもかまわない。
- K. 「謝辞(Acknowledgement)」あるいは「献辞(Dedication)」等と題されたいかなる部分も、その部分の題名を保存し、その部分の内容(各貢献者への謝意あるいは献呈の意)と語調を保存すること。
- L. 「文書」の変更不可部分を、その本文および題名を変更せずに保存すること。章番号やそれに相当するものは部分の題名の一部とは見做さない。
- M. 「推薦の辞(Endorsement)」というような章名が題された部分はすべて削除すること。そのような部分を改変版に含めてはならない。
- N. すでに存在する部分を「推薦の辞(Endorsement)」と題されるように改名したり、題名の点で変更不可部分のどれかと衝突するように改名してはならない。
- O. 保証否認警告を保存すること。

もし改変版に、補遺部分としての条件を満たし、かつ「文書」から複製物された文章や図などをいっさい含んでいない、前書き的な章あるいは付録が新しく含まれるならば、あなたは希望によりそれらの部分の一部あるいはすべてを変更不可と宣言することができる。変更不可を宣言するためには、それらの部分の題名を改変版の利用許諾告知中の変更不可部分一覧に追加すれば良い。これらの題名は他の章名とは全く別のものでなければならない。

含まれる内容が、さまざまな集団によるあなたの改変版に対する推薦の辞のみである限り、あなたは、「推薦の辞(Endorsement)」と題された章を追加することができる。推薦の辞の例としては、ピアレビューの陳述、あるいは文書がある標準の権威ある定義としてその団体に承認されたという声明などがある。

あなたは、5語までの一文を表カバーテキストとして、25語までの文を裏表紙テキストとして、改変版のカバーテキスト一覧の末尾に加えることができる。一個人ないし一団体が直接(あるいは団体内で結ばれた協定によって)加えることができるのは、表カバーテキストおよび裏カバーテキストとしてそれぞれ一文ずつのみである。もし以前すでにその文書において、表裏いずれかの表紙にあなたの(またはあなたが代表する同じ団体内で為された協定に基づく)カバーテキストが含まれていたならば、あなたが新たに追加することはできない。しかしあなたは、その古い文を加えた以前の出版者から明示的な許可を得たならば、古い文を置き換えることができる。

「文書」の著者あるいは出版者は、この利用許諾契約書によって、彼らの名前を利用することを許可しているわけではない。彼らの名前を改変版の宣伝に使ったり、改変版への明示的あるいは黙示的な保証のために使うことを許可するものではない。

文書の結合

あなたは、上記第4項において改変版に関して定義された条件の下で、この利用許諾契約書の下で発表された複数の文書の一つにまとめることができる。その際、原本となる文書にある変更不可部分を全て、改変せずに結合後の著作物中に含め、それらをあなたが統合した著作物の変更不可部分としてその利用許諾告知において列記し、かつ原本にある全ての保証否認警告を保存しなければならない。

結合後の著作物についてはこの契約書の複製物の一つ含んでいればよく、同一内容の変更不可部分が複数ある場合には一つで代用してよい。もし同じ題名だが内容の異なる変更不可部分が複数あるならば、そのような部分のそれぞれの題名の最後に、(もし分かっているならば)その部分の原著者あるいは出版者の名前で、あるいは他と重ならないような番号を括弧で括って記載することで、それぞれ見分けが付くようにしなければならない。結合後の著作物の利用許諾告知における変更不可部分の一覧においても、章の題名に同様の調整をすること。

結合後の著作物においては、あなたはそれぞれの原本の「履歴(History)」と題されたあらゆる部分をまとめて、「履歴(History)」と題された一章にしなければならない。同様に、「謝辞(Acknowledgements)」あるいは「献辞(Dedications)」と題されたあらゆる部分もまとめなければならない。あなたは「推薦の辞(Endorsements)」と題されたあらゆる部分も削除しなければならない。

文書の収集

あなたは、この利用許諾契約書の下で発表された複数の文書で構成される収集著作物を作ることができる。その場合、それぞれの文書が逐語的に忠実に複製されることを保障するために他のすべての点でこの契約書の定める条件に従う限り、さまざまな文書中のこの契約書の個々の複製物を、収集著作物中に複製物の一つ含めることで代用することができる。

あなたは、このような収集著作物から文書の一つ取り出し、それをこの契約書の下で頒布することができる。ただしその際には、この契約書の複製物を抽出された文書に挿入し、またその他すべての点でこの文書の逐語的に忠実な複製に関してこの契約書が定める条件に従わなければならない。

独立した著作物の集積

「文書」あるいはその派生物を、他の別の独立した文書あるいは著作物と一緒にし、一巻の記憶装置あるいは頒布媒体に収めた編集著作物は、編集に起因する著作権が編集著作物に含まれる個々の著作物がその利用者に許可した法的権利を制限するよう行使されない限り、「集積」著作物と呼ばれる。「文書」が集積著作物に含まれる場合、この契約書は、「文書」と共にまとめられた他の独立した著作物には、それら自身が「文書」の派生物で無い限り適用されることにはならない。

このような「文書」の複製物において、この利用許諾契約書の第3項によりカバーテキストの掲載が要求されている場合、「文書」の量が集積著作物全体の2分の1以下であれば、「文書」のカバーテキストは集積著作物中で「文書」そのものを囲む中表紙、あるいは「文書」が電子的形式である場合には表紙の電子的等価物にのみ配置するだけでよい。その場合以外は、カバーテキストは集積著作物全体を取り巻く印刷された表紙に掲載されなければならない。

翻訳

翻訳は改変の一種と見做すので、あなたは「文書」の翻訳をこの利用許諾契約書の第4項の定める条件の下で頒布することができる。変更不可部分を翻訳によって置き換えるには著作権者の特別許可を必要とするが、元の変更不可部分に追加する形で変更不可部分の全てないし一部の翻訳を含めることはかまわない。この契約書や「文書」中の利用許諾告知、保証否認警告すべての英語原本も含める限り、あなたはこの契約書、告知、警告の翻訳を含めることができる。契約書や告知、警告に関して翻訳と英語原本との間に食い違いが生じた場合、英語原本が優先される。

典型的な例として、「文書」のある部分が原文で「Acknowledgements」、「Dedications」、あるいは「History」と題されていた場合、実際の題名を変更するには、題名を保存する(この契約書の第1項)ための条件(同第4項)を満たすことが必要となる。

契約の終了

この利用許諾契約書の下で明確に提示されている場合を除き、あなたは「文書」を複製、改変、サブライセンス、あるいは頒布してはならない。このライセンスで指定されている以外の、「文書」の複製、改変、サブライセンス、頒布に関するすべての企ては無効であり、この契約書によって保証されるあなたの権利を自動的に終結させることとなる。しかし、この契約書の下であなたが複製物ないし諸権利を得た個人や団体に関しては、そういった人々がこの契約書に完全に従ったままである限り、彼らに与えられた許諾は終結しない。

将来における本利用許諾契約書の改訂

フリーソフトウェア財団は、時によってGNUフリー文書利用許諾契約書の新しい改訂版を出版することができる。そのような新版は現在の版と理念においては似たものになるであろうが、新たに生じた問題や懸念を解決するため細部においては違ったものになるだろう。詳しくは<http://www.gnu.org/copyleft/>を参照せよ。

GNUフリー文書利用許諾契約書のそれぞれの版には、新旧の区別が付くようなバージョン番号が振られている。もし「文書」において、この契約書のある特定の版か「それ以降のどの版でも」適用して良いと指定されている場合、あなたはフリーソフトウェア財団から発行された(草稿として発表されたものを除く)指定の版かそれ以降の版のうちどれか一つを選び、その条項や条件に従うことができる。もし「文書」がこの契約書のバージョン番号を指定していない場合には、あなたはフリーソフトウェア財団から今までに出版された(草稿として発表されたものを除く)版のうちからどれか一つを選ぶことができる。

付録: この利用許諾契約書をあなたの文書に適用するには

この利用許諾契約書をあなたが書いた文書に適用するには、この契約書の複製物一つを文書中に含め、以下に示す著作権表示と利用許諾告知を題扉のすぐ後に置いて下さい:

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
only as published by the Free Software Foundation;  
with the Invariant Section being this copyright notice and license.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

(訳)

Copyright (C) 西暦年 あなたの名前。
この文書を、フリーソフトウェア財団発行の GNU フリー文書利用許諾契約書 (バージョン1.2かそれ以降から一つを選択) が定める条件の下で複製、頒布、あるいは改変することを許可する。変更不可部分、表カバーテキスト、裏カバーテキストは存在しない。この利用許諾契約書の複製物は「GNU フリー文書利用許諾契約書」という章に含まれている。

もし変更不可部分や表カバーテキスト、裏カバーテキストがあれば、「変更不可部分...は存在しない。」というところを以下で置き換えてください:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

(訳)

(章の題名を列記)は変更不可部分であり、(表カバーテキストを列記)は表カバーテキスト、(裏カバーテキストを列記)は裏カバーテキストである。

変更不可部分はあるがカバーテキストは存在しないなど、その他の三者の組み合わせに関しては、状況に合わせて上記二つの選択肢を混ぜてください。

あなたの文書に、他に類を見ない独自のプログラムコードのサンプルが含まれる場合、フリーソフトウェアにおいてそのコードを利用することを許可するために、そういったサンプルに関してはこの利用許諾契約書と同時にGNU一般公衆許諾契約書のようなフリーソフトウェア向けライセンスのうち どれか一つを選択して適用してもよい、というような条件の下で発表することを推奨します。

