

# SUSE Linux Enterprise Desktop

11

[www.novell.com](http://www.novell.com)

2009 1 15

管理ガイド



# 管理ガイド

All content is copyright © 2006- 2009 Novell, Inc.

## 保証と著作権

このマニュアルは、Novellの知的所有権で保護されています。このマニュアルを複製、コピー、または配布するには、本使用許諾契約の各条項に明示的に同意する必要があります。

このマニュアルは、以下の条件を満たす限り、電子的または印刷物などの形式で、自由に複製、コピー、配布することができます。

複製、コピー、配布されるコピーには、この著作権表示と作成者、貢献者名が明示的かつ弁別的に表示する必要があります。このマニュアルは、特に印刷形式の場合、非商用の目的でのみ複製、配布できます。本マニュアルの全部または一部を他の目的で使用する場合は、事前にNovell, Incから明示的な許可を得る必要があります。

Novellの商標リストについては、<http://www.novell.com/company/legal/trademarks/tmlist.html>のNovell Trademark and Service Mark Listを参照してください。  
\*Linuxは、Linus Torvaldsの登録商標です。他のすべての第三者の商標は、各所有者が所有権を有しています。商標記号(®、™など)は、Novellの商標を表しています。アスタリスク(\*)は、サードパーティの商標を表します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは絶対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、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

# 目次

このガイドについて	ix
パート I サポートと共通タスク	1
1 YaSTオンラインアップデート	3
1.1 Qtインタフェースを使用したパッチの手動インストール	4
1.2 Gtkインタフェースを使用したパッチの手動インストール	6
1.3 自動オンラインアップデート	8
2 サポート用システム情報の収集	9
2.1 Novell Support Linkの概要	9
2.2 Supportconfigの使用	10
2.3 Novellへの情報の送信	12
2.4 詳細情報	14
3 テキストモードのYaST	15
3.1 モジュールでのナビゲーション	16
3.2 キーの組み合わせの制約	18
3.3 YaSTコマンドラインオプション	19
4 コマンドラインツールによるソフトウェアの管理	21
4.1 Zypperの使用	21
4.2 RPM—パッケージマネージャ	27

<b>5</b>	<b>Nomadを使用したリモートデスクトップへのアクセス</b>	<b>41</b>
5.1	Nomadの前提条件	42
5.2	インストールとセットアップ	43
5.3	Nomadの使用	44
5.4	トラブルシューティング	45
5.5	詳細情報	46
<b>6</b>	<b>BashとBashスクリプト</b>	<b>47</b>
6.1	「シェル」とは何か?	47
6.2	シェルスクリプトの作成	54
6.3	コマンドイベントのリダイレクト	55
6.4	エイリアスの使用	56
6.5	Bashでの変数の使用	56
6.6	コマンドのグループ化と結合	58
6.7	よく使用されるフローコンストラクトの操作	60
6.8	詳細情報	61
	<b>パート II システム</b>	<b>63</b>
<b>7</b>	<b>64ビットシステム環境での32ビットと64ビットのアプリケーション</b>	<b>65</b>
7.1	ランタイムサポート	66
7.2	ソフトウェア開発	67
7.3	biarchプラットフォームでのソフトウェアのコンパイル	67
7.4	カーネル仕様	68
<b>8</b>	<b>Linuxシステムのブートと設定</b>	<b>71</b>
8.1	Linuxのブートプロセス	71
8.2	initプロセス	75
8.3	/etc/sysconfigによるシステム設定	85
<b>9</b>	<b>ブートローダGRUB</b>	<b>89</b>
9.1	GRUBによるブート	90
9.2	YaSTによるブートローダの設定	100
9.3	Linuxブートローダのアンインストール	107
9.4	ブートCDの作成	107
9.5	SUSEのグラフィカル画面	109
9.6	トラブルシューティング	109
9.7	詳細情報	111

<b>10</b>	<b>特別なシステム機能</b>	<b>113</b>
10.1	特殊ソフトウェアパッケージ	113
10.2	バーチャルコンソール	121
10.3	キーボードマッピング	121
10.4	言語および国固有の設定	122
<b>11</b>	<b>プリンタの運用</b>	<b>127</b>
11.1	印刷システムのワークフロー	129
11.2	プリンタに接続するための方法とプロトコル	130
11.3	ソフトウェアのインストール	130
11.4	ネットワークプリンタ	131
11.5	グラフィカルな印刷インタフェース	134
11.6	コマンドラインからの印刷	135
11.7	SUSE Linux Enterprise Desktopでの特殊機能	135
11.8	トラブルシューティング	138
<b>12</b>	<b>udevを使用した動的カーネルデバイス管理</b>	<b>147</b>
12.1	/devディレクトリ	147
12.2	カーネルのueventおよびudev	148
12.3	ドライバ、カーネルモジュールおよびデバイス	148
12.4	ブートおよび初期デバイスセットアップ	149
12.5	実行中のudevデーモンの監視	150
12.6	udevルールを処理するカーネルデバイスイベントへの影響	151
12.7	永続的なデバイス名の使用	158
12.8	udevで使用するファイル	159
12.9	詳細情報	160
<b>13</b>	<b>X Windowシステム</b>	<b>161</b>
13.1	X Window システムの手動設定	161
13.2	フォントのインストールと設定	168
13.3	詳細情報	175
<b>14</b>	<b>FUSEによるファイルシステムへのアクセス</b>	<b>177</b>
14.1	FUSEの設定	177
14.2	NTFSパーティションのマウント	177
14.3	SSHFSによるリモートファイルシステムのマウント	179
14.4	ISOファイルシステムのマウント	179
14.5	利用可能なFUSEプラグイン	180
14.6	詳細情報	180

## パート III モバイルコンピュータ 181

### 15 Linuxでのモバイルコンピューティング 183

15.1	ラップトップ . . . . .	183
15.2	モバイルハードウェア . . . . .	191
15.3	携帯電話とPDA . . . . .	192
15.4	詳細情報 . . . . .	193

### 16 電源管理 195

16.1	省電力機能 . . . . .	195
16.2	ACPI . . . . .	196
16.3	ハードディスクの休止 . . . . .	201
16.4	トラブルシューティング . . . . .	203
16.5	詳細情報 . . . . .	205

### 17 タブレットPCの使用 207

17.1	タブレットPCパッケージのインストール . . . . .	208
17.2	タブレットデバイスの設定 . . . . .	209
17.3	仮想キーボードの使用 . . . . .	210
17.4	ディスプレイの回転 . . . . .	211
17.5	ジェスチャ認識の使用 . . . . .	212
17.6	ペンを使用したメモの作成とスケッチ . . . . .	215
17.7	トラブルシューティング . . . . .	217
17.8	詳細情報 . . . . .	219

## パート IV サービス 221

### 18 ネットワークの基礎 223

18.1	IPアドレスとルーティング . . . . .	227
18.2	IPv6 一次世代のインターネット . . . . .	230
18.3	ネームレゾリューション . . . . .	240
18.4	YaSTによるネットワーク接続の設定; . . . . .	242
18.5	NetworkManager . . . . .	264
18.6	ネットワークの手動環境設定 . . . . .	265
18.7	ダイアルアップアシスタントとしてのsmpppd . . . . .	281

### 19 無線通信 285

19.1	無線LAN . . . . .	285
------	-----------------	-----

<b>20 ネットワーク上のSLPサービス</b>	<b>297</b>
20.1 インストール	297
20.2 SLPをアクティブ化する	298
20.3 SUSE Linux Enterprise DesktopのSLPフロントエンド	298
20.4 SLPによるサービスの提供	299
20.5 詳細情報	300
<b>21 NTPによる時刻の同期</b>	<b>301</b>
21.1 YaSTでのNTPクライアントの設定;	302
21.2 ネットワークでのntpの手動設定	305
21.3 ローカルリファレンスクロックの設定	306
<b>22 NetworkManagerの使用</b>	<b>307</b>
22.1 NetworkManagerの使用	307
22.2 NetworkManagerの有効化	308
22.3 ネットワーク接続の設定	309
22.4 KDE NetworkManagerウィジェットの使用	310
22.5 GNOME NetworkManagerアプレットの使用	311
22.6 NetworkManagerとVPN	313
22.7 NetworkManagerとセキュリティ	315
22.8 よくある質問とその回答	316
22.9 トラブルシューティング	318
22.10 詳細情報	320
<b>23 Samba</b>	<b>321</b>
23.1 用語	321
23.2 Sambaサーバの設定	323
23.3 クライアントの設定	323
23.4 ログインサーバとしてのSamba	323
23.5 詳細情報	325
<b>24 NFS共有ファイルシステム</b>	<b>327</b>
24.1 必要なソフトウェアのインストール	327
24.2 YaSTによるファイルシステムのインポート	328
24.3 ファイルシステムの手動インポート	329
24.4 NFSでのKerberosの使用	331
24.5 詳細情報	331

<b>25</b>	<b>ファイルの同期</b>	<b>333</b>
25.1	使用可能なデータ同期ソフトウェア . . . . .	333
25.2	プログラムを選択する場合の決定要因 . . . . .	335
25.3	CVSの概要 . . . . .	338
25.4	rsyncの概要 . . . . .	341
25.5	詳細情報 . . . . .	343



# このガイドについて

このガイドは、SUSE® Linux Enterprise.の操作時にプロフェッショナルなネットワーク/システム管理者によって使用されることを目的としています。ここでは、SUSE Linux Enterpriseが、ネットワークで必要とされるサービスが使用可能になるように正しく設定され、最初にインストールしたとおりに適切に機能させることができるようになることを目的にしています。このガイドでは、SUSE Linux Enterpriseとお使いのアプリケーションソフトウェアに互換性があるかどうか、また、ない場合の対処方法、および主要機マがアプリケーションの要件に適合しているかどうかなどの分野については取り上げていません。すべての要件が満たされていることかどうか監査済みであること、また、必要なインストール作業を実施済みであること、またはこのような監査に備えてテストインストールが求められたことを前提に、詳細を説明していきます。

このガイドでは、次の内容が取り上げられています。

## 管理

SUSE Linux Enterpriseには、システムのさまざまな側面をカスタマイズするための幅広いツールが用意されています。この部分では、これらのツールの一部を紹介しています。

## システム

このパートを参照して、OSの詳細を学習してください。SUSE Linux Enterpriseは多数のハードウェアアーキテクチャをサポートしているので、この特長を利用すると、独自のアプリケーションをSUSE Linux Enterpriseでの実行に適応させることができます。また、Linuxシステムの仕組みを理解し、独自のカスタムスクリプトやアプリケーションに応用するために役立つ、ブートローダや、ブート手順についても説明しています。

## モバイルコンピューティング

ラップトップおよびモバイルデバイス(PDA、携帯電話など)/SUSE Linux Enterprise間の通信には、特別な配慮が必要です。電力の節約、および変化するネットワーク環境への各種デバイスの統合に留意してください。また、必要な機能を提供する背景技術を知ることも大事です。

## サービス

SUSE Linux Enterpriseは、ネットワークオペレーティングシステムとして設計されています。SUSE® Linux Enterprise Desktopには多数のネットワー

クサービスへのクライアントサポートが含まれます。MS Windowsクライアントおよびサーバなどの異種システム環境にうまく統合します。

このマニュアル中の多くの章に、他の資料やリソースへのリンクが記載されています。これらの資料の中には、システムから参照できるものもあれば、インターネット上に公開されているものもあります。

ご使用製品の利用可能なマニュアルと最新のドキュメントアップデートの概要については、<http://www.novell.com/documentation>を参照してください。

# 1 利用可能なマニュアル

これらのガイドブックは、HTMLおよびPDFの各バージョンを複数の言語で提供しています。この製品については、次のユーザー用および管理者用マニュアルがあります。

## *GNOME User Guide (↑GNOME User Guide)*

SUSE Linux Enterprise DesktopのGNOMEデスクトップについて説明しています。デスクトップの使用および設定方法と、キータスクの実行方法を説明します。主として、デフォルトのデスクトップとしてGNOMEデスクトップを効率的に使用したいと考えるエンドユーザ向けです。

## *アプリケーションガイド (↑アプリケーションガイド)*

SUSE Linux Enterprise Desktopの主なデスクトップアプリケーションの使用および設定方法について説明しています。ブラウザ、E-mailクライアント、オフィスアプリケーション、コラボレーションツールについて説明しています。グラフィックとマルチメディアアプリケーションについても扱っています。

## *導入ガイド (↑導入ガイド)*

単一または複数のシステムをインストールする方法および展開インフラストラクチャに製品本来の機能を活用する方法を示します。ローカルインストールまたはネットワークインストールサーバの使用から、リモート制御の高度にカスタマイズされた自動リモートインストール技術による大規模展開まで、多様なアプローチから選択できます。

## 管理ガイド(1 ページ)

当初のインストールシステムの保守、監視、およびカスタマイズなど、システム管理タスクについて説明します。

## セキュリティガイド(↑セキュリティガイド)

システムセキュリティの基本概念を紹介し、ローカルセキュリティ/ネットワークセキュリティの両方の側面を説明します。製品固有のセキュリティソフトウェア(プログラムが読み込み/書き込み/実行の対象にするファイルをプログラムごとに指定できるNovell AppArmorなど)や、セキュリティ関係のイベント情報を確実に収集する監査システムを使用する方法を示します。

## システム分析およびチューニングガイド(↑システム分析およびチューニングガイド)

問題の検出、解決、および最適化に関する管理者ガイド。ツールの監視によってシステムを検査および最適化する方法およびリソースを効率的に管理する方法を見つけることができます。よくある問題と解決、および追加のヘルプとドキュメントリソースの概要も含まれています。

## Xenによる仮想化(↑Xenによる仮想化)

ご使用製品の仮想化技術を紹介します。SUSE Linux Enterprise Serverでサポートされているプラットフォームのアプリケーションとインストールタイプに関するさまざまなフィールドの概要、およびインストール手順の簡単な説明について記載しています。

総合的なマニュアルに加えて、クイックスタートガイドも利用できます。

## クイックスタートのインストール(↑クイックスタートのインストール)

システム要件を一覧し、DVDまたはISOイメージからのSUSE Linux Enterprise Desktopのインストールをステップごとに順を追って説明します。

## Linux監査クイックスタート

監査システムを有効にし設定する方法と、主要タスク(監査ルールの設定、レポートの生成、ログファイルの分析など)を実行する方法を簡単に説明します。

## Novell AppArmorクイックスタート

naareg;の背景をなす主要概念を説明します。

大半のSUSE Linux Enterprise DesktopマニュアルのHTMLバージョンは、インストールしたシステム内の/usr/share/doc/manualか、ご使用のデスクトップのヘルプセンタで見つけることができます。マニュアルの最新の更新バージョンは、<http://www.novell.com/documentation>にあります。ここでは、製品のマニュアルのPDFまたはHTMLバージョンをダウンロードできます。

## 2 フィードバック

次のフィードバックチャンネルがあります：

- 製品コンポーネントのバグの報告や、改善強化要求の提出には、<https://bugzilla.novell.com/>を使用してください。Bugzillaを初めてご使用になる場合は、Novell Bugzilla ホームページの*Bug Writing FAQs*が参考になることがあります。
- 本マニュアルおよびこの製品に含まれているその他のマニュアルについて、皆様のご意見やご要望をお寄せください。オンラインドキュメントの各ページの下部にあるユーザコメント機能を使用して、コメントを入力してください。

## 3 マニュアルの表記規則

本書では、次の書体を使用しています：

- /etc/passwd:ファイル名およびディレクトリ名
- *placeholder:placeholder*は、実際の値で置き換えられます
- PATH:環境変数PATH
- ls、--help:コマンド、オプション、およびパラメータ
- user:ユーザまたはグループ
- Alt、Alt + F1:キー:押すためのキーまたはキーの組み合わせ、キーはキーボードと同様に、大文字で表示されます

- [ファイル] , [ファイル] > [名前を付けて保存] :メニュー項目、ボタン
- *Dancing Penguins*(「*Penguins*」の章、↑他のマニュアル):他のマニュアル中の章への参照です。



# パートⅠ. サポートと共通タスク





# YaSTオンラインアップデート

Novellは製品に対して、継続的にソフトウェアセキュリティアップデートを提供しています。デフォルトでは、openSUSE Updaterがシステムの更新に使用されます。openSUSE Updaterの詳細については、項「Keeping the System Up-to-date」(第7章 *Installing or Removing Software*, ↑導入ガイド)を参照してください。この章では、ソフトウェアパッケージを更新する代替ツールとして、YaSTオンラインアップデートを紹介します。

SUSE® Linux Enterprise Desktopの現在のパッチは、アップデートソフトウェアリポジトリから入手できます。インストール時に製品を登録した場合、アップデートリポジトリはすでに設定されています。SUSE Linux Enterprise Desktopを登録しなかった場合は、YaSTで、[ソフトウェア] > [オンラインアップデートの設定]の順にクリックし、[詳細] > [Register for Support and Get Update Repository]の順に選択します。または、信頼できるソースから、手動でアップデートリポジトリを追加することもできます。リポジトリを追加または削除するには、YaSTで、[ソフトウェア] > [Software Repositories]の順に選択して、リポジトリマネージャを起動します。リポジトリマネージャの詳細については、項「Adding Software Repositories」(第7章 *Installing or Removing Software*, ↑導入ガイド)を参照してください。.

---

**注意:** アップデートカタログのアクセス時のエラー

アップデートカタログにアクセスできない場合、登録の期限が切れている場合があります。通常、SUSE Linux Enterprise Desktopには1年または3年の登録期間があり、この期間内にアップデートカタログにアクセスできます。このアクセスは登録期間が切れると拒否されます。

アップデートカタログへのアクセスが拒否された場合は、Novell Customer Centerにアクセスして登録状態を確認するように推奨する警告メッセージが表示されます。Novell Customer Centerには、<http://www.novell.com/center/>からアクセスできます。

---

は、各種の関連性レベルを持つアップデートを提供します。Securityアップデートは、重大なセキュリティハザードを修復するので、必ずインストールする必要があります。Recommendedアップデートは、コンピュータを損なう可能性のある問題を解決し、Optionalアップデートは、セキュリティ関係以外の問題を解決するか、または拡張機能を提供します。

YaSTを使ってアップデートやパッチをインストールするには、YaSTから [ソフトウェア] > [オンラインアップデート] の順にクリックします。お使いのシステムに適用できるすべてのパッチが、インストールする項目として最初から選択されています(任意項目を除く)。[了解] または [適用] をクリックすると、これらのパッチが自動的にインストールされます。インストールが完了したら、[完了] をクリックします。これで、システムが最新の状態になりました。

---

#### ティップ: deltarpmの無効化

デフォルトでは、アップデートは、deltarpmとしてダウンロードされます。deltarpmからのrpmパッケージの再構築は、メモリとCPU時間を消費するので、セットアップまたはハードウェア構成によっては、パフォーマンス上の理由によりdeltarpmの使用を無効にする必要があります。deltarpmの使用を無効にするには、ファイル/etc/zypp/zypp.confを編集してdownload.use\_deltarpmをfalseに設定します。

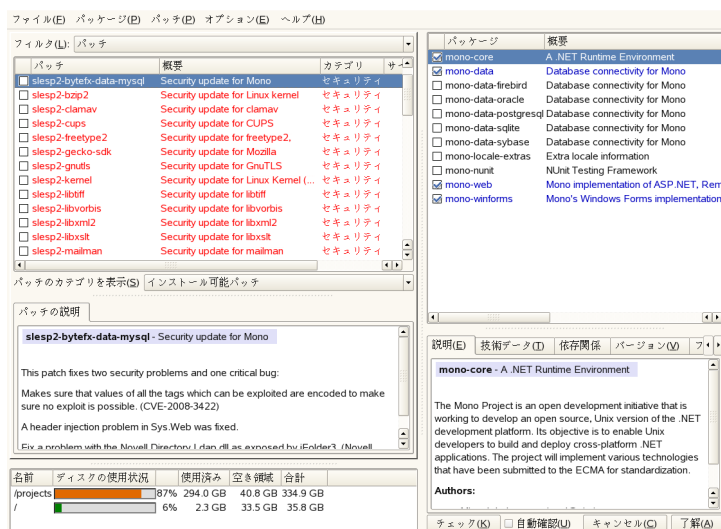
---

## 1.1 Qtインタフェースを使用したパッチの手動インストール

[オンラインアップデート] ウィンドウは、4つのセクションから成り立っています。利用できるパッチは、左側のリストに表示されています。パッチを選択すると、パッチリストの下部にその情報が表示されます。右側の列には、選択したパッチに含まれているパッケージが表示されます(パッチが複数のパッケージから構成されることもあります)。その下には、選択したパッケー

ジの詳細が表示されます。オプションとして、ディスク使用率を左列の下に表示できます(この表示はデフォルトではフェードアウトしています。ドット付きのスライダを使用して見えるようにしてください)。

## 図 1.1 YaSTオンラインアップデート



パッチディスプレイには、SUSE Linux Enterprise Desktop.で使用できるパッチが表示されます。パッチはセキュリティ重要度順にソートされています。security、recommended、およびoptionalパッチには異なる3つのビューがあります。[パッチのカテゴリを表示]を使用して、ビューを切り替えます。

### [Needed Patches] (デフォルトビュー)

システムにインストールされたパッケージに適用される、まだインストールされていないパッチ。

### [Unneeded Patches]

システムにインストールされていないパッケージに適用されるパッチか、または(すでに別のソースで更新されたので)要件がすでに満たされているパッチ。

### すべてのパッチ

SUSE Linux Enterprise Desktopに使用できるすべてのパッチ。

リストの各項目は、記号とパッチ名から成り立っています。リストに表示される記号については、**Shift + F1** キーを押してください。セキュリティおよび推奨パッチで要求されるアクションは、自動的に設定されます。アクションは、**[自動インストール]**、**[自動更新]**、**[自動削除]** です。オプションパッチのアクションは事前設定されません。パッチを右クリックして、リストからアクションを選択します。

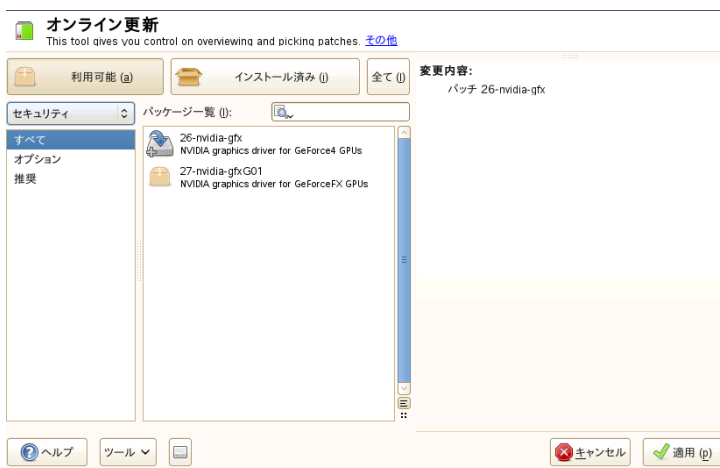
アップデートリポジトリ以外のリポジトリから最新のパッケージをインストールする場合、そのパッケージのパッチ要件はそのインストールで満たされる場合があります。この場合、パッチ概要の前にチェックマークが表示されます。パッチは、インストール用にマークするまでリストに表示されます。これによってパッチは実際にはインストールされませんが(パッチはすでに最新であるため)、インストール済みとしてパッチをマークします。

大部分のパッチには、複数のパッケージのアップデートが含まれています。あるパッケージに対するアクションを変更する場合は、パッケージウィンドウからパッケージを右クリックしてアクションを選択してください。適用するパッチとパッケージをすべて選択したら、**[了解]** を選択します。

## 1.2 Gtkインタフェースを使用したパッチの手動インストール

**[オンラインアップデート]** ウィンドウは、2つの主要なセクションから成り立っています。左ペインにはすべてのパッチが一覧され、パッチリストにさまざまなフィルタを提供します。**[適用]** をクリックしたら、実行される変更のリストを右ペインで確認します。

## ☒ 1.2 YaSTオンラインアップデート



### パッチリストフィルタ

#### 使用可能

システムにインストールされたパッケージに適用される、まだインストールされていないパッチ。

#### インストール済み

すでにインストール済みのパッチ。

#### すべて

すでにインストール済みであるか使用可能なパッチ。

#### 重大度

[オプション]、[推奨]、または [セキュリティ] パッチのみ表示。デフォルトでは [すべて] のパッチが表示されます。

#### リポジトリ

このフィルタはリポジトリごとのパッチを表示します。

### Packages Listing

カスタムのフィルタをここに適用します。

左ペインの下側にあるパッチについて詳細情報を表示するには、パッチのエントリをクリックして行を開きます。ここにはパッチの詳細な説明と使用可

能なバージョンが表示されます。オプションのパッチを [インストールする] することも選択できます。[セキュリティ] パッチおよび [推奨] パッチはすでにインストール用に事前選択されています。

## 1.3 自動オンラインアップデート

YaSTでは、自動アップデートを設定することもできます。[ソフトウェア] > [オンラインアップデートの設定] の順に選択します。[自動オンラインアップデート] をオンにして、アップデートの間隔を [毎日]、[毎週]、[毎月] の中から選択します。カーネルアップデートなど、ユーザによる作業が必要なパッチもあります。このような場合は、自動アップデート処理が中止されます。したがって、アップデート処理を完全に自動進行させたい場合は、[インタラクティブパッチをスキップする] をオンにします。そのように設定した場合、ユーザの介入が必要なパッチをインストールするため、時折、手動で [オンラインアップデート] を実行する必要があります。

# サポート用システム情報の収集

問題が発生したら、`supportconfig`を使用すると、システム情報(使用中の現在のカーネルバージョン、ハードウェア、RPMデータベース、パーティションなど)を収集できます。その結果は、Novellサポートセンタが問題を検知する上で役立ちます。

## 2.1 Novell Support Linkの概要

Novell Support Link (NSL)はSUSE Linux Enterprise Desktopの新しい機能です。システム情報を収集し、その情報を別のサーバにアップロードして詳細な分析を行えるツールです。NovellサポートセンタはNovell Support Linkを使用して問題のあるサーバのシステム情報を収集し、その情報をNovell公開FTPサーバに送信します。収集されるシステム情報には、使用されている現在のカーネルバージョン、ハードウェア、RPMデータベース、パーティションなどが含まれます。その結果は、Novellサポートセンタが未解決のサービス要求を解決する上で役立ちます。

Novell Support Linkを使用するには、次の2つの方法があります。

1. YaSTサポートモジュールを使用します。
2. コマンドラインユーティリティ `supportconfig`を使用します。

YaSTサポートモジュールは`supportconfig`を呼び出してシステム情報を収集します。

## 2.2 Supportconfigの使用

次のセクションではYaSTでコマンドラインからsupportconfigを使用する方法と、選択できるオプションについて説明します。

### 2.2.1 YaSTによる情報の収集

YaSTでシステム情報を収集するには、次の手順に従います。

- 1 URL<http://www.novell.com/center/eservice>を開き、サービス要求番号を作成します。
- 2 YaSTを起動します。
- 3 [サポート] モジュールを開きます。
- 4 [Create report tarball] をクリックします。
- 5 ラジオボタンリストからオプションを選択します。この設定をテストしたい場合は、[Only gather a minimum amount of info] を使用します。  
[次へ] で続行します。
- 6 連絡先情報を入力します。ステップ 1 (10 ページ)からのサービス要求番号を [Novell 11 digit service request number] とラベル付けされたテキストフィールドに入力します。[次へ] で続行します。
- 7 情報収集が開始されます。プロセスが完了したら、[次へ] で続行します。
- 8 収集されたデータをレビューし、必要のないファイル名は、[Remove from Data] で削除します。[次へ] で続行します。
- 9 tarballを保存します。Novellカスタマセンタへアップロードする場合は、[Upload log files tarball into URL] が有効になっていることを確認してください。[次へ] をクリックして完了します。



## 2.2.2 Supportconfigの直接使用による情報収集

supportconfigをコマンドラインから使用する場合は、次の手順に従います。

- 1 シェルを開きrootになります。
- 2 オプションを使用せずにsupportconfigを実行すると、デフォルトのシステム情報が収集されます。
- 3 ツールが完了するまで待ちます。
- 4 デフォルトのアーカイブ場所は、/var/logのファイル名形式nts\_HOST\_DATE\_TIME.tbzです。

## 2.2.3 共通のSupportconfigオプション

supportconfigユーティリティには多様な起動オプションがあります。これらのオプションを見るにはsupportconfig-hを使用するか、マニュアルページを参照してください。通常、supportconfigはオプションを使用せずに実行されます。一般によく使用される起動オプションの一部をまとめたものを次に示します。

- 収集する情報のサイズを削減するには、最小オプション(-m)を使用します。

```
supportconfig -m
```

- 出力に追加連絡先情報を含めます(1行で)。

```
supportconfig -E tux@example.org -N "Tux Penguin" -O "Penguin Inc." ...
```

- トラブルシューティング時には、現在作業中の問題のある領域についてのみ、情報を収集したい場合があります。たとえば、LVMに問題があり、最近デフォルトのsupportconfig出力に問題が見つかった場合です。変更を終えたら、現在のLVMの情報を収集する必要があります。supportconfigとLVMの最低限の情報のみを収集するには以下を使用します。

```
supportconfig -i LVM
```

完全な機能リストを見るには、次を実行します。

```
supportconfig -F
```

- `-u` と `-r` オプションを使用して、割り当てられたサービス要求番号で `supportconfig tarball` をアップデートします。たとえば、Novell でサービス要求を開き、追跡番号が `12345678901` の場合、次を実行します。

```
supportconfig -ur 12345678901
```

## 2.3 Novell への情報の送信

YaST サポートモジュールまたは `supportconfig` コマンドラインユーティリティを使用して、システム情報を Novell へ送信できます。サーバに問題があり Novell のサポートを希望する場合、サービス要求を開いてサーバ情報を Novell に送信する必要があります。YaST とコマンドラインの両方の方法について説明されています。

### 手順 2.1 YaST を使用した Novell への情報の送信

- 1 URL <http://www.novell.com/center/eservice> を開き、サービス要求番号を作成します。
- 2 11 桁のサービス要求番号を記入します。次の例ではサービス要求番号が `12345678901` であると想定しています。
- 3 YaST サポートモジュールウィンドウで、*[レポート tar アーカイブを作成]* をクリックします。
- 4 *[Use custom]* ラジオボタンを選択します。 *[次へ]* で続行します。
- 5 連絡先情報を入力し、 *[Novell 社の 11 桁サービスリクエスト番号]* を入力して、Novell のアップロードターゲットの URL を含めます。
  - 安全なアップロードターゲットには、<https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}> を使用します。

- 通常のFTPアップロードターゲットには、<ftp://ftp.novell.com/incoming>を使用します。

[次へ] で続行します。情報の収集が開始します。プロセスが完了したら、 [次へ] で続行します。

- 6 データのコレクションを確認し、 [データから削除] を使用して、Novell にアップロードされたtarballから取り除きたいファイルがあれば削除します。 [次へ] で続行します。
- 7 デフォルトではtarballのコピーが/rootに保存されます。前述したNovell アップロードターゲットの1つを使用していることを確認し、 [URL に ログファイルのtar アーカイブをアップロード] が有効になっていることを確認してください。 [次へ] をクリックして完了します。
- 8 [完了] をクリックします。

## 手順 2.2 supportconfigを使用したNovellへの情報の送信

- 1 URL<http://www.novell.com/center/eservice>を開き、サービス 要求番号を作成します。
- 2 11桁のサービス要求番号を記入します。次の例ではサービス要求番号が 12345678901であると想定しています。
- 3 インターネット接続のあるサーバの場合:

- 3a デフォルトのアップロードターゲットを使用するには、次を実行します。

```
supportconfig -ur 12345678901
```

- 3b 安全なアップロードターゲットには、次を1行で使用します。

```
supportconfig -r 12345678901 -U  
'https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}'
```

- 4 インターネット接続のないサーバの場合

- 4a 次を実行します。

```
supportconfig -r 12345678901
```

- 4b** 手動で `/var/log/nts_SR12345678901*tbz`tarball を Novell の FTP サーバ (<ftp://ftp.novell.com/incoming>) にアップロードします。
- 4c** サービス要求 URL <http://www.novell.com/center/eservice> を使用して tarball をサービス要求に添付することもできます。
- 5** tarball が <ftp://ftp.novell.com/incoming> ディレクトリある場合は、自動的にサービス要求に添付されます。

## 2.4 詳細情報

システム情報の収集の詳細については、次のドキュメントを参照してください。

- `man supportconfig`—`supportconfig` のマニュアルページ
- `man supportconfig.conf`—`supportconfig` 設定ファイルのマニュアルページ
- <http://www.novell.com/communities/print/node/4097>—「A Basic Server Health Check with Supportconfig」
- <http://www.novell.com/communities/print/node/4827>—「Create Your Own Supportconfig Plugin」
- <http://www.novell.com/communities/print/node/4800>—「Creating a Central Supportconfig Repository」

## テキストモードのYaST

このセクションは、システムでXサーバを実行せずに、テキストベースのインストールツールを使用しているシステム管理者や専門家の方を対象にしています。ここでは、YaSTをテキストモードで開始、操作するための、基本的な情報を説明しています。

テキストモードのYaSTは、ncursesライブラリを使用して、使いやすい擬似グラフィカルユーザインタフェースを提供します。ncursesライブラリは、デフォルトでインストールされています。YaST inを実行するためのターミナルエミュレータの最小サポートサイズは、80x25文字です。

### 図 3.1 テキストモードのYaSTのメインウィンドウ



YaSTをテキストモードで起動すると、YaSTコントロールセンターが最初に表示されます(図 3.1 参照)。このメインウィンドウは、以下の3つの主要領域で

構成されています。太い白枠で囲まれた左側のフレームには、各種モジュールが属するカテゴリが示されます。アクティブカテゴリは、背景色付きで示されています。細い白枠で囲まれた右側のフレームには、アクティブカテゴリで使用可能なモジュールの概要が示されています。下方のフレームには、**[ヘルプ]** および **[終了]** 用ボタンがあります。

YaSTコントロールセンターが起動されると、カテゴリ **[ソフトウェア]** が自動的に選択されます。カテゴリを変更するには、**↓**と**↑**を使用します。選択したカテゴリからモジュールを起動するには、**→**を押します。選択したモジュールがここで太い枠付きで表示されます。必要なモジュールを選択するには、**↓**と**↑**を使用します。矢印キーを押したままにして、使用可能なモジュールのリストをスクロールします。モジュールを選択すると、色付きの背景にモジュールタイトルが表示されます。

<Enter>キーを押して、必要なモジュールを起動します。モジュール内のさまざまなボタンまたは選択フィールドには、別の色(デフォルトでは黄色)の文字が含まれます。そのまま**Tab**キーでナビゲートする代わりに、直接、ボタンを選択するには、**Alt+yellow\_letter**を使用します。**Alt+Q**を押すか、または **[終了]** を選択して**Enter**を押して、YaSTコントロールセンターを終了します。

## 3.1 モジュールでのナビゲーション

以降のYaSTモジュール内のコントロール要素の説明では、ファンクションキーと**Alt**キーの組み合わせがすべて機能し、別のグローバル機能を割り当てられていないことを前提としています。可能性のある例外事項については、**3.2項「キーの組み合わせの制約」** (18 ページ)を参照してください。

ボタンおよび選択リスト間のナビゲーター

選択リストを含むボタンおよびフレーム間でナビゲートするには、**Tab**キーを使用します。逆の順序でナビゲートするには、**Alt+Tab**または**Shift+Tab**の組み合わせを使用します。

選択リストでのナビゲーター

選択リストを含むアクティブフレーム内の個々の要素間でナビゲーターするには、矢印キー(**↑**と**↓**)を使用します。フレーム内の個別エントリがその幅を超える場合は、**Shift+→**または**Shift+←**を使用して、右または左にスクロールします。代わりに**Ctrl+E**または**Ctrl+A**を使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の場合のように、**→**または

←を使用したのでは、アクティブフレームまたは現在の選択リストが変更されてしまう場合に使用できます。

ボタン、ラジオボタン、およびチェックボックス

☐ が付いているボタン(チェックボックス)または()が付いているボタン(ラジオボタン)を選択するには、<Space>キーまたは<Enter>キーを押します。代わりに、Alt+yellow\_letterでラジオボタンおよびチェックボックスを直接選択することもできます。この場合、<Enter>キーによる確認は不要です。<Tab>キーでアイテムにナビゲートする場合は、<Enter>キーを押して、選択したアクションを実行するか、対応するメニューアイテムをアクティブにします。

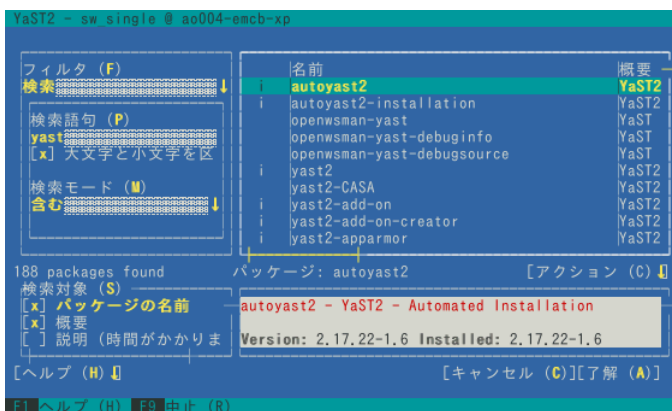
ファンクションキー

Fキーの(F1からF12を使用すると、さまざまなボタンの機能を素早く利用できます。使用可能なFキーのショートカットは、YaST画面の一番下の行に表示されます。どのファンクションキーが実際にどのボタンにマップされているかは、アクティブになっているYaSTモジュールによります。提供されるボタン( [詳細]、[情報]、[追加]、[削除] など)は、モジュールごとに異なるからです。F10は、[受諾]、[OK]、[次へ]、および [完了] の代わりに使用します。F1を押して、YaSTヘルプにアクセスします。

ncursesモードのナビゲーションツリーの使用

一部のYaSTモジュールでは、ウィンドウの左部分にあるナビゲーションツリーを使用して、設定ダイアログを選択します。ncursesモードでは、ナビゲーションツリーでの選択後、選択したダイアログを表示するにはEnterを押す必要があります。これは、ナビゲーションツリーのブラウズ時に時間のかかる再表示を節約するため、意図的に工夫された動作です。

## 図 3.2 ソフトウェアインストールモジュール



## 3.2 キーの組み合わせの制約

ウィンドウマネージャがグローバルなAltキーの組み合わせを使用していると、YaSTでのAltキーの組み合わせが機能しない場合があります。ShiftやAltなどのキーは、端末の設定に専有されている場合もあります。

キーの代用とするキーを<AltEsc>

ショートカットは、<Alt>キーの代わりに<AltEsc>キーでも実行できます。たとえば、Esc + Hは、Alt + Hの代わりとなります。(まずEscを押して、次にHを押します)

Ctrl + FとCtrl + Bによる前後のナビゲーション

AltとShiftの組み合わせがウィンドウマネージャまたは端末に専有されている場合は、Ctrl + F(進む)とCtrl + B(戻る)を代わりに使用できます。

ファンクションキーの制約

Fキーは、各種機能にも使用されます。一部のファンクションキーは、端末に専有され、YaSTで使用できない場合があります。ただし、<Alt>キーのキーの組み合わせとファンクションキーは、ピュアテキストコンソールでは常に完全に使用できます。



## 3.3 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ースのほか、YaSTには、シンプルなコマンドラインインターフェースがあります。YaSTコマンドラインオプションのリストを表示するには、次のように入力します。

```
yast -h
```

### 3.3.1 個別モジュールの起動

時間節約のため、個別のYaSTモジュールを直接起動できます。モジュールを起動するには、次のように入力します。

```
yast <module_name>
```

「yast -l」または「yast --list」と入力して、システムで使用可能になっているすべてのモジュールのリストを表示します。たとえば、「yast lan」と入力して、ネットワークモジュールを起動します。

### 3.3.2 コマンドラインからのパッケージのインストール

パッケージ名が既知であり、パッケージが有効なインストールリポジトリに用意されている場合は、コマンドラインオプション-iを使用してパッケージをインストールできます。

```
yast -i <package_name>
```

または

```
yast --install <package_name>
```

*package\_name*は、1つの短いパッケージ名にするか(たとえば、依存性チェック付きでインストールされるgvim)、またはrpmパッケージへの完全なパスにすることができます(依存性チェックなしでインストールされる)。

YaSTから提供される機能を超える機能を持つ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を必要とする場合は、zypperの使用をご検討ください。この新しいユーティリティは、YaSTパッケージマネージャの基礎でもあ

る同じソフトウェア管理ライブラリを使用します。zypperの基本的使用法については、[4.1頁「Zypperの使用」](#) (21 ページ)で説明されています。

### 3.3.3 YaSTモジュールのコマンドラインパラメータ

スクリプトでYaST機能を使用するため、YaSTでは、個々のモジュールのコマンドラインサポートを用意しています。ただし、すべてのモジュールにコマンドラインサポートがあるわけではありません。モジュールで利用できるオプションを表示するには、次のように入力します。

```
yast <module_name> help
```

モジュールにコマンドラインサポートがない場合、モジュールはテキストモードで起動され、次のメッセージが表示されます。

```
This YaST module does not support the command line interface.
```

# コマンドラインツールによるソフトウェアの管理

この章では、ソフトウェア管理の2つのコマンドラインツールとして、ZypperとRPMについて説明します。

## 4.1 Zypperの使用

Zypperは、パッケージのインストールおよび更新用の新しいコマンドラインツールです。zypperの構文はrugに類似しています。rugとは対照的に、zypperではzmdデーモンが背後で実行している必要はありません。rugの互換性の詳細は、[http://en.opensuse.org/Zypper/Usage#Compatibility\\_with\\_Rug](http://en.opensuse.org/Zypper/Usage#Compatibility_with_Rug)を参照してください。これは特に、リモートソフトウェア管理タスクの実行、またはシェルスクリプトからのソフトウェアの管理で役立ちます。

zypperには、次のヘルプの概要が組み込まれています。

```
zypper help
```

### 4.1.1 一般的な使用方法

zypperの一般的な構文は次のとおりです。

```
zypper [global-options] command [command-options] [arguments] ...
```

ブラケットで囲まれたコンポーネントは必須ではありません。zypperを実行する最も簡単な方法は、その名前の後にコマンドを入力することです。たとえば、システムタイプに必要なすべてのパッチを適用するには、次のようにします。

```
zypper update
```

さらに、グローバルオプションをコマンドの直前に入力することによって、1つ以上のグローバルオプションから選択することができます。たとえば `--non-interactive` を指定した場合、質問なしで、独自の指定によりコマンドを実行します。

```
zypper --non-interactive update
```

特定のコマンドに固有のオプションを使用する場合は、コマンドの直後にそのオプションを入力します。たとえば `--auto-agree-with-licenses` は、ライセンスの確認を求めることなく、すべての必要なパッチをシステムに適用します。すべてのライセンスは事前に読み取られます。

```
zypper update --auto-agree-with-licenses
```

一部のコマンドでは、1つ以上の引数が必要です。

```
zypper install mplayer
```

また一部のオプションでは、引数が必要です。次は、新しいパッケージによりシステムを更新します。

```
zypper update -t package
```

上記のすべてをまとめると、factoryリポジトリのみを使用してインストールし、バーボスの場合には、次のようになります。

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

また、`+/-` または `~/!` 修飾子を使用することで、1回の実行でパッケージをインストールおよび削除できます。修飾子:

```
zypper install emacs -vim
```

または:

```
zypper remove emacs +vim
```

また、指定する最初のパッケージで `-` を使用する場合は、コマンドオプションとして解釈されないように、その前に `--` を書き込む必要があります。

```
zypper install -- -vim emacs
```

## 4.1.2 Zypperを使ったソフトウェアのインストールと削除

登録済みリポジトリからパッケージをインストールするには、次を使用します。

```
zypper install package_name
```

zypperはまたワイルドカードをサポートします。先頭に`package_name`が付くすべてのパッケージをインストールするには、次を使用します。

```
zypper install package_name*
```

ローカルまたはリモートRPMを直接インストールすることも可能です。またZypperはすべてのパッケージ`package_name`を自動的にインストールします。次の条件によって異なります。

```
zypper install http://www.example.com/package_name.rpm
```

依存関係を解決するには、`--no-recommends`または`--no-force-resolution`のいずれかを使用します。

インストール済みパッケージを削除するには、次を使用します。

```
zypper remove package_name
```

---

**警告:** システムに必須のパッケージは削除しないでください。

`glibc`、`zypper`、`kernel`などのパッケージは削除しないでください。こうしたパッケージはシステムに必須であり、これらが欠落しているとシステムの処理が中断することがあります。

---

zypperではデフォルトで、選択したパッケージのインストールまたは削除の前に確認が求められます。この動作は、`--non-interactive`オプションを使用することで上書きされます。このオプションは、次のように実際のモード(インストール、削除、更新)の前に指定する必要があります

```
zypper --non-interactive install package_name
```

このオプションによりスクリプトおよびcronジョブでzypperを使用できます。

パッケージの対応するソースパッケージをインストールする場合は、次を使用します。

```
zypper source-install package_name
```

このコマンドにより、指定したパッケージの構築依存もインストールされます。この処理が必要ない場合は、次のようにスイッチ`--no-build-deps`を追加します。

```
zypper source-install --no-build-deps package_name
```

もちろんこの機能は、リポジトリリストにソースパッケージが追加されたりリポジトリが存在する場合にのみ動作します。リポジトリの追加の詳細については、[4.1.4頁「リポジトリの管理」](#) (25 ページ)を参照してください。

インストール済みのソフトウェアベースを変更後、すべての依存関係がまだ実現されているか確認します。

```
zypper verify
```

## 4.1.3 Zypperによるソフトウェアの更新

`zypper`を使用してソフトウェアを更新するには2つの異なる方法があります。正式にリリースされたすべてのパッチをシステムに統合するには、次をそのまま実行します。

```
zypper update
```

この場合、リポジトリで利用可能なすべてのパッチが関連性についてチェックされ、必要に応じてインストールされます。

リポジトリに新しいパッケージのみが存在し、パッチが提供されていない場合は、`zypper update`は無効です。こうしたパッケージをすべて更新するには、タイプ`package`の更新をインストールするように指定する必要があります。

```
zypper update -t package
```

個別のパッケージを更新するには、次のインストールコマンドを使用します。

```
zypper install package_name
```

利用可能なすべての新しいパッケージのリストを次のコマンドで表示できます。

```
zypper list-updates -t package
```

## 4.1.4 リポジトリの管理

zypperのすべてのインストールまたは更新のコマンドは、zypperで既知のリポジトリのリストに応じて異なります。システムで既知のすべてのリポジトリのリストを表示するには、次のコマンドを使用します。

```
zypper repos
```

結果は、次の出力のようになります。

#	Enabled	Refresh	Type	Alias	Name
1	Yes	Yes	yast2	openSUSE-DVD 11.0	openSUSE-DVD 11.0
2	Yes	No	yast2	Main (OSS)	Main (OSS)
3	Yes	No	yast2	Main (Non-OSS)	Main (Non-OSS)

リストからリポジトリを削除する場合は、コマンドzypper renamerepoを使用し、削除するリポジトリのエイリアスを指定します。例からMain Repository (Non-OSS)を削除するには、次のコマンドを使用します。

```
zypper renamerepo Main Repository (Non-OSS)
```

リポジトリを追加するには、次を実行します。

```
zypper addrepo URI Alias
```

URIは、インターネットリポジトリ(利用可能なリポジトリのリストについては、[http://en.opensuse.org/Additional\\_YaST\\_Package\\_Repositories](http://en.opensuse.org/Additional_YaST_Package_Repositories)を参照)、ディレクトリ、CD、またはDVDです。エイリアスは、リポジトリの短い名前および一意のIDです。このIDは、固有であること以外は自由に選択できます。既に使用されているエイリアスを指定した場合、zypperでは警告が発行されます。

## 4.1.5 問い合わせ

search、info、what-providesなど、さまざまな問い合わせコマンドが利用可能です。

searchではパッケージ名が対象となり、出力の最初の列にステータス(s)の情報が表示されます。

引数としてinfoとパッケージ名を指定した場合、パッケージに関する詳細情報が表示されます。

what-provides パッケージはrpm -q --whatprovides パッケージに似ていますが、rpmではRPMデータベース(つまり、すべてのインストール済みパッケージのデータベース)のみを問い合わせることができます。それに対してZypperは、インストール済みのパッケージだけでなく、すべてのリポジトリから機能プロバイダに関する情報を表示します。

特にデバッグ目的のために、--plus-repo、--disable-repositories、--disable-system-resolvablesなどのスイッチが利用可能です。1つのリポジトリのみを検索する場合に、こうしたスイッチを使用します。使用方法の詳細については、zypperマニュアルページ(man zypper)を参照してください。

## 4.1.6 Zypperシェルの使用

複数の異なるzypperコマンドを順に実行することが必要な場合があります。各zypperコマンドですべてのデータベースを再読み込みしないで済むように、zypperをシェルモードで実行することができます。

```
zypper shell
```

シェルが実行中の場合は、適切なパラメータとともにzypperコマンドを発行します。

```
zypper> in zsh
...
zypper> exit
```

すべての関連データがメモリ内に残るために、通常はzypperシェルを使用する方が高速です。

Zypperでは、readlineライブラリがサポートされます。これは、Bashシェルでも利用可能なZypperシェルの全コマンドライン編集機能を使用できることを意味します。Zypperでは、そのコマンド履歴がファイル ~/.zypper\_history 内に維持されます。



## 4.1.7 詳細情報

コマンドラインからの更新の詳細については、`zypper --help`と入力するか、`zypper(8)`のマニュアルページを参照してください。例と詳細情報については、<http://en.opensuse.org/Zypper/Usage>を参照してください。

## 4.2 RPM—パッケージマネージャ

RPM (RPM Package Manager)がソフトウェアパッケージを管理するのに使用されます。RPMの主要コマンドは、`rpm`と`rpmbuild`です。ユーザ、システム管理者、およびパッケージの作成者は、強力なRPMデータベースでクエリを行って、インストールされているソフトウェアに関する情報を取得できます。

基本的に`rpm`には、ソフトウェアパッケージのインストール、アンインストール、アップデート、RPMデータベースの再構築、RPMベースまたは個別のRPMアーカイブの照会、パッケージの整合性チェック、およびパッケージへの署名の5種類のモードがあります。`rpmbuild`は、元のソースからインストール可能なパッケージを作成する場合に使用します。.

インストール可能なRPMアーカイブは、特殊なバイナリ形式でパックされています。それらのアーカイブは、インストールするプログラムファイルとある種のメタ情報で構成されます。メタ情報は、ソフトウェアパッケージを設定するために`rpm`によってインストール時に使用されるか、または文書化の目的でRPMデータベースに格納されています。通常、RPMアーカイブには拡張子`.rpm`が付けられます。

---

### ティップ: ソフトウェア開発パッケージ

多くのパッケージにおいて、ソフトウェア開発に必要なコンポーネント(ライブラリ、ヘッダ、インクルードファイルなど)は、別々のパッケージに入れています。それらの開発パッケージは、最新のGNOMEパッケージのように、ソフトウェアを自分自身でコンパイルする場合にのみ、必要になります。それらのパッケージは、パッケージ`alsa-devel`、`gimp-devel`、`kdelibs3-devel`などのように、名前の拡張子`-devel`で識別できます。

---

## 4.2.1 パッケージの信頼性の検証

RPMパッケージにはGnuPG署名があります。フィンガープリントを含む鍵は、次のとおりです。

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>  
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA
```

`rpm --checksig package-1.2.3.rpm`コマンドを使用して、RPMパッケージの署名を検証し、パッケージが本当にSUSEから提供されたものか、他の信頼できる機関から提供されたものか判定できます。これは、インターネットからアップデートパッケージを入手する場合には、特に推奨されます。SUSEのパブリックパッケージ用の署名キーは通常、`/root/.gnupg/`にあります。バージョン8.1以降、キーは、ディレクトリ`/usr/lib/rpm/gnupg/`も格納されており、一般ユーザがRPMパッケージの署名を検証できるようになっています。

## 4.2.2 パッケージの管理:インストール、アップデート、およびアンインストール

通常RPMアーカイブのインストールはとても簡単です。`rpm -i package.rpm`の用に入力します。このコマンドで、パッケージをインストールできます。ただし、依存関係が満たされており、他のパッケージとの競合がない場合に限られます。`rpm`では、依存関係の要件を満たすためにインストールしなければならないパッケージがエラーメッセージで要求されます。バックグラウンドで、RPMデータベースは競合が起きないようにします。ある特定のファイルは、1つのパッケージだけにしか属せません。別のオプションを選択すると、`rpm`にこれらのデフォルト値を無視させることができますが、この処置を行うのは専門知識のある人に限られます。それ以外の人が行くと、システムの整合性を危うくするリスクが発生し、システムアップデート機能が損なわれる可能性があります。

`-U`または`--upgrade`と`-F`または`--freshen`の各オプションは、パッケージをアップデートするのに使用できます。たとえば、`rpm -F package.rpm`です。このコマンドは、古いバージョンのファイルを削除し、新しいファイルをただちにインストールします。2つのバージョン間の違いは、`-U`がシステムに存在していなかったパッケージをインストールするのに対して、`-F`がインストールされていたパッケージを単にアップデートする点にあります。アップデートす

る際、rpmは、以下のストラテジーに基づいて設定ファイルを注意深くアップデートします。

- 設定ファイルがシステム管理者によって変更されていない場合、rpmは新しいバージョンの適切なファイルをインストールします。システム管理者は、何も行う必要はありません。
- アップデートの前に設定ファイルがシステム管理者によって変更されている場合、rpmは変更されたファイルに拡張子.rpmorigまたは.rpmsave(バックアップファイル)を付けて保存し、新しいパッケージからファイルをインストールします。ただしこれは、元々インストールされていたファイルと新しいファイルのバージョンが異なる場合に限りです。異なる場合は、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て、新しいファイルに再度、変更を加えます。後ですべての.rpmorigと.rpmsaveファイルを必ず削除して、今後のアップデートで問題が起きないようにします。
- 設定ファイルがすでに存在しており、またnoreplaceラベルが.specファイルで指定されている場合、.rpmnewファイルが作成されます。

アップデートが終了したら、.rpmsaveファイルと.rpmnewファイルは、比較した後、将来のアップデートの妨げにならないように削除する必要があります。ファイルがRPMデータベースで認識されなかった場合、ファイルには拡張子.rpmorigが付けられます。

認識された場合には、.rpmsaveが付けられます。言い換えれば、rpmorigは、RPM以外の形式からRPMにアップデートした結果として付けられます。.rpmsaveは、古いRPMから新しいRPMにアップデートした結果として付けられます。.rpmnewは、システム管理者が設定ファイルに変更を加えたかどうかについて、何の情報も提供しません。それらのファイルのリストは、/var/adm/rpmconfigcheckにあります。設定ファイルの中には(/etc/httpd/httpd.confなど)、操作が継続できるように上書きされないものがあります。

-Uスイッチは、単に-eオプションでアンインストールして、-iオプションでインストールする操作と同じではありません。可能なときは必ず-Uを使用します。

パッケージを削除するには、`rpm -e package`を入力します。rpmは、解決されない依存関係がない場合にのみパッケージを削除します。他のアプリケーションがTcl/Tkを必要とする限り、Tcl/Tkを削除することは理論的に不可能です。その場合でも、RPMはデータベースに援助を要求します。他の依存関係がない場合でも、また、どのような理由、特殊な環境であってもそのような削除が不可能であれば、`--rebuilddb`オプションを使用してRPMデータベースを再構築するのが良いでしょう。

## 4.2.3 RPMとパッチ

システムの運用上のセキュリティを保証するには、ときどきアップデートパッケージをシステムにインストールする必要があります。以前は、パッケージ内のバグは、パッケージ全体を交換しなければ取り除けませんでした。大きいパッケージの場合、その中の小さなファイルにバグがあると、膨大な量のデータになってしまうことがありました。しかし、SUSE RPMを使用すると、パッケージ内にパッチをインストールできます。

最も重要な考慮事項について、pineを例として説明します。

パッチRPMはシステムに適したものか。

これを検査するには、はじめにインストールされたパッケージでクエリーを行います。pineでは、以下のコマンドを実行します。

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

パッチRPMがこのバージョンのpineに適しているかどうかを検査します。

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
pine = 4.44-195
pine = 4.44-207
```

このパッチは、3種類のバージョンのpineに適しています。例でインストールされたバージョンもリストされています。パッチはインストールできます。

どのファイルがパッチで置き換えられるか。

パッチの影響を受けるファイルは、パッチRPMで見つけられます。

rpmの-Pパラメータを使用すると、特殊なパッチ機能を選択できます。次のコマンドでファイルをリストします。

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

パッチがすでにインストールされていれば、次のコマンドを使用します。

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

パッチRPMをどのようにシステムにインストールするか。

パッチRPMは、通常のRPMと同様に使用されます。唯一の違いは、適切なRPMがすでにインストールされていなければならない点です。

どのパッチがシステムにインストールされており、それらはどのパッケージバージョンのものか。

システムにインストールされているすべてのパッチのリストは、コマンド `rpm -qPa` で表示できます。(この例のように)新しいシステムに1つのパッチだけがインストールされている場合、リストは次のようになります。

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

後日、オリジナルとしてインストールされていたパッケージのバージョンを知りたい場合、その情報はRPMデータベースから得られます。pineの場合、その情報は次のコマンドで表示できます。

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

RPMのパッチ機能に関する情報を含む詳細な情報は、`man rpm` コマンドと `rpmbuild` コマンドのマニュアルページで収集できます。

## 4.2.4 デルタRPMパッケージ

デルタRPMパッケージには、RPMパッケージの新旧バージョン間の違いが含まれています。デルタRPMパッケージを古いRPMに適用すると、まったく新しいRPMになります。デルタRPMパッケージは、インストールされたRPMとも互換性があるので、古いRPMのコピーを保管する必要はありません。デルタRPMパッケージは、パッチRPMよりもさらに小さく、パッケージをインターネット上で転送するのに便利です。欠点は、デルタRPMが組み込まれたアッ

プデート操作の場合、そのままのRPMまたはパッチRPMに比べて、CPUサイクルの消費が目立って多くなることです。

`prepdeltarpm`、`writedeltarpm`、および`applydeltarpm`バイナリは、デルタRPMスイート (`deltarpm` パッケージ) の一部であり、デルタRPMパッケージの作成と適用に際して役立ちます。次のコマンドを使用して、`new.delta.rpm` というデルタRPMを作成します。次のコマンドでは、`old.rpm` および `new.rpm` が存在することが前提となります。

```
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、一時作業ファイル `old.cpio`、`new.cpio`、および `delta` を削除します。

古いパッケージがすでにインストールされていれば、`applydeltarpm` を使用して、ファイルシステムから新たにRPMを構築できます。

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

ファイルシステムにアクセスすることなく、古いRPMから構築するには、`-r` オプションを使用します。

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

技術的な詳細については、『`/usr/share/doc/packages/deltarpm/README`』を参照してください。

## 4.2.5 RPMクエリー

`-q` オプションを使用すると、`rpm` はクエリーを開始し、(`-p` オプションを追加することにより) RPMアーカイブを検査できるようにして、インストールされたパッケージのRPMデータベースでクエリーを行えるようにします。必要な情報の種類を指定する複数のスイッチを使用できます。詳細については、[表 4.1. 「最も重要なRPMクエリーのオプション」](#) (33 ページ) を参照してください。

**表 4.1** 最も重要なRPMクエリーのオプション

-i	パッケージ情報
-l	ファイルリスト
-f FILE	ファイルFILEを含むパッケージでクエリーを行います(FILEにはフルパスを指定する必要があります)。
-s	ステータス情報を含むファイルリスト(-lを暗示指定)
-d	ドキュメントファイルだけをリストします (-lを暗示指定)。
-c	設定ファイルだけをリストします(-lを暗示指定)。
--dump	詳細情報を含むファイルリスト(-l、-c、または-dと共に使用します)
--provides	他のパッケージが--requiresで要求できるパッケージの機能をリストします。
--requires, -R	パッケージが要求する機能
--スクリプト	インストールスクリプト(preinstall、postinstall、uninstall)

たとえば、コマンドrpm -q -i wgetは、例 4.1. 「rpm -q -i wget」 (34 ページ)に示された情報を表示します。

### 例 4.1 rpm -q -i wget

```
Name           : wget                               Relocations: (not relocatable)
Version        : 1.9.1                             Vendor: SUSE LINUX AG,
Nuernberg, Germany
Release        : 50                                Build Date: Sat 02 Oct 2004
03:49:13 AM CEST
Install date: Mon 11 Oct 2004 10:24:56 AM CEST      Build Host: f53.suse.de
Group          : Productivity/Networking/Web/Utilities Source RPM:
wget-1.9.1-50.src.rpm
Size           : 1637514                             License: GPL
Signature      : DSA/SHA1, Sat 02 Oct 2004 03:59:56 AM CEST, Key ID
a84edae89c800aca
Packager       : http://www.suse.de/feedback
URL            : http://wget.sunsite.dk/
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.
[...]
```

オプション-fが機能するのは、フルパスで完全なファイル名を指定した場合だけです。必要な数のファイル名を指定します。たとえば、次のコマンドを実行します。

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

出力は次のとおりです。

```
rpm-4.1.1-191
wget-1.9.1-50
```

ファイル名の一部分しかわからない場合は、例 4.2. 「パッケージを検索するスクリプト」 (34 ページ)に示すようなシェルスクリプトを使用します。実行するときに、ファイル名の一部を、パラメータとして示されるスクリプトに渡します。

### 例 4.2 パッケージを検索するスクリプト

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

rpm -q --changelog rpm コマンドは、特定のパッケージに関する詳細な変更情報を日付順に表示します。この例は、rpm パッケージに関する情報を示します。



インストールされたRPMデータベースを使うと、確認検査を行うことができます。それらの検査は、`-v`、`-y`、または`--verify`オプションを使用して開始します。このオプションを使うと、`rpm`は、パッケージ内にあり、インストール以降変更されたことがあるすべてのファイルを表示します。`rpm`は、次の変更に関するヒントを表示するのに、8文字の記号を使用します。

**表 4.2 RPM確認オプション**

5	MD5チェックサム
S	ファイルサイズ
L	シンボリックリンク
T	変更時間
D	メジャーデバイス番号とマイナーデバイス番号
U	所有者
G	グループ
M	モード (許可とファイルタイプ)

設定ファイルの場合は、文字`c`が表示されます。`/etc/wgetrc(wget)`に対する変更例を以下に示します。

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

RPMデータベースのファイルは、`/var/lib/rpm`に格納されています。パーティション/`usr`のサイズが1GBであれば、このデータベースは、完全なアップデート後、およそ 30 MB占有します。データベースが予期していたよりもはるかに大きい場合は、オプション`--rebuilddb`でデータベースを再構築するようにします。再構築する前に、古いデータベースのバックアップを作成しておきます。`cron`スクリプトの`cron.daily`は、データベースのコピー(`gzip`でバックされる)を毎日作成し、`/var/adm/backup/rpmdb`に格納します。コピー数は`/etc/sysconfig/backup`にある変数`MAX_RPMD_BBACKUPS`で制御し

ます(デフォルト:5)。1つのバックアップのサイズは、1GBの/usrに対しておよそ1MBです。

## 4.2.6 ソースパッケージのインストールとコンパイル

すべてのソースパッケージには、拡張子`.src.rpm`(ソース RPM)が付けられています。

---

### ティップ

ソースパッケージは、インストールメディアからハードディスクにコピーされ、YaSTを使用して展開できます。ただし、ソースパッケージは、パッケージマネージャでインストール済み([i])というマークは付きません。これは、ソースパッケージがRPMデータベースに入れられないためです。インストールされたオペレーティングシステムソフトウェアだけがRPMデータベースにリストされます。ソースパッケージを「インストールする」場合、ソースコードだけがシステムに追加されます。

---

(/etc/rpmrcなどのファイルでカスタム設定を指定していない限り)以下のディレクトリが、/usr/src/packagesの下でrpmとrpmbuildから使用可能でなければなりません。

#### SOURCES

オリジナルのソース(.tar.gzファイルや.tar.gzファイルなど)とディストリビューション固有の調整ファイル(ほとんどの場合.difファイルや.patchファイル)用です。

#### SPECS

ビルド処理を制御する、メタMakefileに類似した.specファイル用です。

#### BUILD

すべてのソースは、このディレクトリでアンパック、パッチ、コンパイルされます。

#### RPMS

完成したバイナリパッケージが格納されます。

SRPMS

ソースRPMが格納されます。

YaSTを使ってソースパッケージをインストールすると、必要なすべてのコンポーネントが/usr/src/packagesにインストールされます。ソースと調整はSOURCES、関連する.specファイルはSPECSに格納されます。

---

## 警告

システムコンポーネント(glibc、rpm、sysvinitなど)で実験してはいけません。システムが正しく動作しなくなります。

---

次の例は、wget.src.rpmパッケージを使用します。YaSTでパッケージをインストールすると、次のファイルが作成されるはずです。

```
/usr/src/packages/SOURCES/nops_doc.diff
/usr/src/packages/SOURCES/toplev_destdir.diff
/usr/src/packages/SOURCES/wget-1.9.1+ipvmisc.patch
/usr/src/packages/SOURCES/wget-1.9.1-brokentime.patch
/usr/src/packages/SOURCES/wget-1.9.1-passive_ftp.diff
/usr/src/packages/SOURCES/wget-LFS-20040909.tar.bz2
/usr/src/packages/SOURCES/wget-wrong_charset.patch
/usr/src/packages/SPECS/wget.spec
```

rpmbuild -b X /usr/src/packages/SPECS/wget.spec コマンドは、コンパイルを開始します。Xは、ビルド処理のさまざまな段階に対して使用されるワイルドカードです(詳細については、--helpの出力またはRPMのドキュメントを参照してください)。以下に簡単な説明を示します。

-bp

/usr/src/packages/BUILD内のソースを用意します。アンパック、パッチしてください。

-bc

-bpと同じですが、コンパイルを実行します。

-bi

-bpと同じですが、ビルドしたソフトウェアをインストールします。警告: パッケージがBuildRoot機能をサポートしていない場合は、設定ファイルが上書きされることがあります。

-bb

-biと同じですが、バイナリパッケージを作成します。コンパイルに成功すると、バイナリパッケージは、/usr/src/packages/RPMSに作成されるはずです。

-ba

-bbと同じですが、ソース RPMを作成します。コンパイルに成功すると、バイナリは/usr/src/packages/SRPMSに作成されるはずです。

--short-circuit

一部のステップをスキップします。

作成されたバイナリRPMは、rpm -iコマンドまたはrpm -Uコマンドでインストールできます。rpmを使用したインストールは、RPMデータベースに登場します。

## 4.2.7 buildによるRPMパッケージのコンパイル

多くのパッケージにつきものの不都合は、ビルド処理中に不要なファイルが稼働中のシステムに追加されてしまうことです。これを回避するには、パッケージのビルド先の定義済みの環境を作成するbuildを使用します。このchroot環境を確立するには、build スクリプトが完全なパッケージツリーと共に提供されなければなりません。パッケージツリーは、NFS経由で、またはDVDからハードディスク上で利用できるようにすることができます。build --rpms *directory*で、位置を指定します。rpmとは異なり、buildコマンドはソースディレクトリでSPECファイルを探します。(上記の例と同様に)システムで/media/dvdの下にマウントされているDVDでwgetをビルドするには、rootユーザーで次のコマンドを使用します。

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

これで、最小限の環境が/var/tmp/build-rootに確立されます。パッケージは、この環境でビルドされます。処理が完了すると、ビルドされたパッケージは/var/tmp/build-root/usr/src/packages/RPMSに格納されます。

buildスクリプトでは、他のオプションも多数使用できます。たとえば、スクリプトがユーザ独自のRPMを処理するようにするには、ビルド環境の初期化を省略するか、rpmコマンドの実行を上記のビルド段階のいずれかに制限します。build --helpコマンドとman buildコマンドで、詳細な情報が得られます。

## 4.2.8 RPMアーカイブとRPMデータベース用のツール

Midnight Commander (mc)は、RPMアーカイブの内容を表示し、それらの一部をコピーできます。アーカイブを仮想ファイルシステムとして表し、Midnight Commanderの通常のメニューオプションを使用できます。<F3>キーを使用してHEADERを表示します。カーソルキーと<Enter>キーを使ってアーカイブ構造を表示します。<F5>キーを使用してアーカイブコンポーネントをコピーします。

KDEは、rpmのフロントエンドとしてkpackageツールを提供します。完全装備のパッケージマネージャが、YaSTモジュールとして使用可能です(「第7章 *Installing or Removing Software* (↑導入ガイド)」を参照してください)。



# Nomadを使用したリモートデスクトップへのアクセス

SUSE® Linux Enterprise Desktopに添付されているNomad(Novell Open Mobile Agile Desktop)を使用すると、グラフィックハードウェアから切り離されたデスクトップセッションを実行できます。Nomadは、次のコアコンポーネントで構成されます。

## プロキシXサーバ

Composite、XVideo、RANDRなど、現代のX拡張をサポートします。

## Session Manager

リモートからアクセスできるデスクトップセッションの生成と追跡を行います。

## 接続ハンドラ

RDP (Remote Desktop Protocol)を転送およびセキュリティのレイヤとして使用します。RDPは、Microsoft Terminal Servicesを実行するコンピュータへの接続を可能にするマルチチャネルプロトコルです。ただし、接続ハンドラは、クライアントソフトウェアによってサポートされると、仮想X11チャネル (rdpx11) を使用して、未フィルタのX11トラフィックをデスクトップを表示するローカルXサーバに転送します。接続ハンドラは、常に、必要に応じて通常のRDPコマンドにフォールバックできます。つまり、リモートデスクトップを任意の既存RDP クライアントからアクセスできます。

## クライアントプログラム

SUSE Linux Enterprise Desktopには、Nomadを実装する特別なRPDクライアントが提供されています。このクライアントは、X11プロトコル転送のた

めの特別拡張であり、適切なコンポジットマネージャプラグインがロードされると、リモートデスクトップをローカルに合成できます。

コンポジットマネージャの拡張

コンポジティングは、アプリケーションウィンドウの高度な視覚効果(透明度、フェード、ゆがみ、シャッフル、リダイレクトなど)を可能にします。

Nomadでは、さまざまな物理ロケーションからデスクトップにリモートアクセスできます。たとえば、自宅または会社から同じセッションにアクセスできます。作業セッションが中断されても、別の端末に移り、作業を開始できます。現在実行中の環境を、ラップトップなどのモバイルデバイスにコピーすることも可能です。Nomadを使用すれば、デスクトップを、コラボレーションまたはトレーニング用に共有し、リモートからの制御や管理を行うことができます。

## 5.1 Nomadの前提条件

Nomadを使用するには、`rdesktop`パッケージをローカルコンピュータにインストールする必要があります。さらに、次のパッケージもインストールできます。

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`
- `tsclient`



デスクトップを提供しているリモートコンピュータ上で、オープンソースリモートデスクトッププロトコル (**RDP**) サーバを含む `xrdp` パッケージがインストールされている必要があります。

さらに、次のパッケージもインストールできます。

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`

## 5.2 インストールとセットアップ

ホストとして動作するローカルコンピュータに特別な設定は要りません。`rdesktop` パッケージをインストールしたらすぐに、`rdesktop` コマンドラインツールを使用して、デスクトップを提供するリモートコンピュータに接続できます。グラフィカルユーザインタフェースを使用する場合は、`tsclient` パッケージも追加してインストールします。`tsclient` (Terminal Server Client) は、`rdesktop` などのリモートデスクトップツールのGNOMEフロントエンドであり、`Xnest` クライアントやVNCクライアント(`vncviewer`)もサポートします。パフォーマンスおよびデスクトップ効果の向上には、追加の`compiz` パッケージをインストールします。

ただし、デスクトップを提供するリモートコンピュータは、次のように準備する必要があります。

- 1 `xrdp` パッケージをインストールします。これにより、自動的に、`xrpd` サーバがランレベル5に追加されます。このサービスを手動で開始または停止するには、`root` として、`/etc/init.d/xrdp start` または `/etc/init.d/xrdp stop` を実行します。

- 2 ポート3389がRDP接続に使用されるので、このポートへの接続を許可するように、ファイアウォールを設定します。YaSTを起動し、[セキュリティとユーザ] > [ファイアウォール] の順に選択します。[許可されるサービス] をクリックして、サービスを許可するゾーンを選択します。[詳細] をクリックし、3389を [TCPポート] として入力します。YaSTで設定内容を確認します。
- 3 3Dデスクトップ効果を使用したい場合は、追加のcompizパッケージをインストールします。これにより、仮想チャネルのサポートでクライアントを使用する際に著しくパフォーマンスが向上します。ローカルとリモートの両方のデスクトップでデスクトップ効果を有効にすることで、ローカルコンポジットマネージャはリモートデスクトップからの要素に効果を適用できるようになります。

---

#### 注意: Desktopの効果

リモートデスクトップ上でデスクトップ効果を使用する場合は、compiz-plugins-dmxパッケージが、リモートデスクトップを提供するシステムとリモートデスクトップにアクセスするローカルシステムの両方でインストールされていることを確認してください。

---

## 5.3 Nomadの使用

xrpdが実行され、リモートコンピュータでポート3389が開くと、ただちに、RDPクライアントでリモートホストに接続できます。接続には、rdesktopコマンドラインツールか、またはグラフィカルユーザインタフェースを提供するtsclientを使用します。

### 5.3.1 rdesktopを使用してサーバに接続します。

ユーザtuxに圧縮モードで接続を確立するには、シェルから次のコマンドを実行します。

```
rdesktop -u tux -z server
```

serverは、リモートコンピュータのホスト名またはIPアドレスです。

このコマンドで、指定のユーザのログイン画面が開きます。ユーザは、この画面から、リモートデスクトップにログインできます。**xrdp**によるデスクトップセッションは、独立型であり、**GDM**や**KDM**などの通常のディスプレイマネージャと競合しません。

接続を確立するときは多数のオプションを設定できます。たとえば、フル画面モードを使用したり、特定のキーボード配列を選んだり、ジオメトリを調整できます。**xrdp** `--help`で利用可能な**xrdp**オプションについて詳細を参照してください。

## 5.4 トラブルシューティング

接続が確立できないときは、次のリストに従って作業を行います。

**xrdp**サーバはリモートコンピュータ上で実行されていますか？

1. **xrdp**パッケージが、デスクトップを提供するリモートコンピュータにインストールされているかどうかチェックします。
2. **xrdp**サービスが実行されているかどうかチェックします。
3. **xrdp**サービスが実行されていない場合は、次のコマンドをrootとして実行して、**xrdp**サービスを起動または再起動してください：  
`/etc/init.d/xrdp start`または`/etc/init.d/xrdp restart`

**xrdp**サービスの起動後は、2つのプロセス(**xrdp**および**xrdp-sesman**)が実行されます。なんらかの理由で、それらの1つが起動しない場合は、これらのプロセスをフォアグラウンドで手動で開始すると、多くの場合、なにが問題なのか分かります。

4. プロセスを手動で起動するには、rootになって、`/usr/sbin/xrdp-sesman -n`および`/usr/sbin/xrdp -nodaemon`を実行します。
5. 詳細については、`/var/log/xrdp-sesman.log`内の**xrdp-sesman**出力と`/var/log/messages`内の**xrdp**出力を参照してください。

## 5.5 詳細情報

Nomadの詳細については、<http://en.opensuse.org/Nomad>を参照してください。

# BashとBashスクリプト

今日、多数のユーザが、KDEやGNOMEなどのGUI(グラフィカルユーザインターフェース)を介してコンピュータを使用しています。GUIは多くの機能を備えていますが、自動タスクの実行という点では、その用途は限られます。シェルは、GUIに追加すると便利なツールです。この章では、シェル(ここではBash)のいくつかの側面について概説します。

## 6.1 「シェル」とは何か?

従来、シェルとは、Bash(Bourne again Shell)のことでした。この章でも、Bashを「シェル」と呼びます。実際には、Bash以外にも、異なる機能と特性を持つシェルがあります。他のシェルの詳細については、YaSTでシェルを検索してください。

### 6.1.1 Bash設定ファイルの知識

シェルは、次のように呼び出すことができます。

1. 対話型ログインシェル。コンピュータへのログイン時に、`--login`オプションとともにBashを呼び出す場合か、SSHを使用してリモートコンピュータへログインする場合に使用します。
2. 「通常の」対話型シェル。xterm,やkonsoleなどのツールの起動時には、通常、この形式を使用します。

3. 非対話型シェル。コマンドラインからシェルスクリプトを呼び出す場合に使用します。

使用するシェルのタイプによって、異なる設定ファイルを読み込みます。次のテーブルには、それぞれ、ログインシェル設定ファイルと非ログインシェル設定ファイルが示されています。

**表 6.1** ログインシェル用*Bash*設定ファイル

ファイル	説明
/etc/profile	このファイルは変更しないでください。変更しても、次の更新で変更内容が破棄される可能性があります。
/etc/profile.local	/etc/profileを拡張する場合は、このファイルを使用します。
/etc/profile.d/	特定プログラムのシステム全体に渡る設定ファイルを含みます。
~/.profile	ログインシェル用のユーザ固有の設定をここに挿入します。

**表 6.2** 非ログインシェル用*Bash*設定ファイル

/etc/bash.bashrc	このファイルは変更しないでください。変更しても、次の更新で変更内容が破棄される可能性があります。
/etc/bash.bashrc.local	<b>Bash</b> のシステム全体に渡る変更を挿入する場合のみ、このファイルを使用します。
~/bashrc	ユーザ固有の設定をここに挿入します。

さらに、**Bash**では、次のファイルも使用します。

表 6.3 Bash用特殊ファイル

ファイル	説明
~/.bash_history	入力したすべてのコマンドのリストを含みます。
~/.bash_logout	ログアウト時に使用されます。

## 6.1.2 ディレクトリの構造

次のテーブルでは、Linuxシステムの最も重要なハイレベルディレクトリの短い概要を示します。それらのディレクトリおよび重要なサブディレクトリの詳細については、後続のリストを参照してください。

表 6.4 標準的なディレクトリツリーの概要

ディレクトリ	目次
/	ルートディレクトリ—ディレクトリツリーの開始点
/bin	システム管理者および通常ユーザの両者が必要とするコマンドなどの必須バイナリファイル。通常、Bashなどのシェルも含みます。
/boot	ブートローダの静的ファイル
/dev	ホスト固有のデバイスのアクセスに必要なファイル
/etc	ホスト固有のシステム設定ファイル
/home	システムにアカウントを持つすべてのユーザのホームディレクトリを格納します。rootのホームディレクトリだけは、/homeでなく、/rootにあります。
/lib	必須の共有ライブラリおよびカーネルモジュール

ディレクトリ	目次
/media	リムーバブルメディアのマウントポイント
/mnt	ファイルシステムを一時的にマウントするためのマウントポイント
/opt	アドオンアプリケーションのソフトウェアパッケージ
/root	スーパーユーザrootのホームディレクトリ
/sbin	必須のシステムバイナリ
/srv	システムで提供するサービスのデータ
/tmp	一時ファイルを格納するディレクトリ。
/usr	読み込み専用データを含む第二階層
/var	ログファイルなどの可変データ
/windows	システムにMicrosoft Windows*とLinuxの両方がインストールされる場合のみ利用可能。Windowsデータを含みます。

次のリストでは、さらに詳しい情報を提供し、ディレクトリに含まれるファイルおよびサブディレクトリの例を示します。

#### /bin

rootと他のユーザの両者が使用できる基本的なシェルコマンドを含みます。それらのコマンドは、ls、mkdir、cp、mv、rm、rmdirなどです。/binには、SUSE Linux Enterprise DesktopのデフォルトシェルであるBashも含まれます。

#### /boot

ブートローダやカーネルのデータなど、ブートに必要なデータを含みます。これらのデータは、カーネルによるユーザモードプログラムの実行開始前に使用されます。



`/dev`

ハードウェアコンポーネントを記述したデバイスファイルを格納します。

`/etc`

**X Window System**などのプログラムの動作を制御するローカル設定ファイルを含みます。`/etc/init.d`サブディレクトリは、ブートプロセスで実行されるスクリプトを含みます。

`/home/username`

システムにアカウントを持つすべてのユーザの個人データを格納します。このディレクトリ内のファイルは、その所有者またはシステム管理者しか変更できません。デフォルトでは、電子メールのディレクトリとパーソナルデスクトップの設定が、非表示のファイルおよびディレクトリとして、ここに格納されます。デスクトップ用個人設定データは、**KDE**ユーザの場合は`.kde`または`.kde4`、**GNOME**ユーザの場合は`.gconf`として格納されます。

---

### 注意: 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するユーザのホームディレクトリは、`/home`以外のファイルシステム内のディレクトリにマップできます。

---

`/lib`

システムのブートとルートファイルシステムでのコマンドの実行に必要な必須共有ライブラリを含みます。**Windows**で共有ライブラリに相当するものは、**DLL**ファイルです。

`/media`

**CD-ROM**、**USB**スティック、デジタルカメラ(**USB**を使用する場合)など、リムーバブルメディアのマウントポイントを含みます。`/media`では、一般にシステムのハードディスク以外のあらゆるタイプのドライブが保持されます。リムーバブルメディアをシステムに挿入または接続し、マウントを完了すると、ただちに、そのメディアにこのディレクトリからアクセスできます。

`/mnt`

このディレクトリは一時的にマウントされるファイルシステムのマウントポイントを提供します。`root`がここでファイルシステムをマウントできます。

/opt

追加ソフトウェアのインストール用に予約されています。オプションソフトウェアや大型アドオンプログラムのパッケージをここに格納できます。現在は、KDE3はここにありますが、KDE4とGNOMEは、/usrに移動されています。

/root

rootユーザのホームディレクトリ。rootの個人データがここに保存されます。

/sbin

sで示唆されるように、このディレクトリはスーパーユーザ用のユーティリティを格納します。/sbinには、/bin内のバイナリとともにシステムのブート、復元、および回復に不可欠なバイナリを含みます。

/srv

FTPやHTTPなど、システムによって提供されるサービスのデータを格納します。

/tmp

ファイルの一時的保管を必要とするプログラムによって使用されます。

/usr

/usrは、ユーザとは無関係であり、UNIX system resourcesを意味する略語です。/usr内のデータは静的な読み込み専用データです。このデータは、FHS(Filesystem Hierarchy Standard)に準拠するホスト間で共有できます。このディレクトリは、すべてのアプリケーションプログラムを含み、ファイルシステム内の第二階層を形成します。KDE4とGNOMEも、このディレクトリに格納されています。/usrには、/usr/bin、/usr/sbin、/usr/local、/usr/share/docなど、多数のサブディレクトリがあります。

/usr/bin

一般ユーザがアクセスできるプログラムを含みます。

/usr/sbin

修復関数など、システム管理者用に予約されたプログラムを含みます。

/usr/local

このディレクトリには、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ローカルな拡張プログラムをインストールできます。

/usr/share/doc

システムのドキュメントファイルおよびリリースノートを格納します。manualサブディレクトリには、このマニュアルのオンラインバージョンが格納されます。複数の言語をインストールする場合は、このディレクトリに各言語のマニュアルを格納できます。

packagesには、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含まれているドキュメントが格納されます。パッケージごとに、サブディレクトリ /usr/share/doc/packages/*packagename*が作成されます。このサブディレクトリには、多くの場合、**README**ファイルが含まれます。例、設定ファイル、または追加スクリプトが含まれる場合もあります。

**HOWTO**をシステムにインストールした場合は、/usr/share/docにhowtoサブディレクトリも含まれます。このサブディレクトリには、Linuxソフトウェアの設定および操作に関する多数のタスクの追加ドキュメントが格納されます。

/var

/usrは静的な読み込み専用データを含みますが、/varは、システム動作時に書き込まれる可変データ(ログファイル、スプールデータなど)のディレクトリです。たとえば、システムのログファイルは、/var/log/messagesに格納され、rootだけがアクセスできます。

/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のみ使用可能。システムのWindowsパーティションで使用可能なWindowsデータを含みます。このディレクトリのデータを編集できるかどうかは、Windowsパーティションが使用するファイルシステムによって異なります。FAT32の場合、このディレクトリのファイルを開いて編集できます。NTFSファイルシステムの場合、LinuxからのWindowsファイルの読み込みのみ可能で、変更はできません。

## 6.2 シェルスクリプトの作成

シェルスクリプトは、データの収集、テキスト内のワードやフレーズの検索など、あらゆる種類の多数の有用なタスクの実行に便利な方法です。次の例では、小型のシェルスクリプトでテキストをプリントします。

### 例 6.1 テキストをプリントするシェルスクリプト

```
#!/bin/sh ❶  
# Output the following line: ❷  
echo "Hello World" ❸
```

- ❶ 1行目は、このファイルがスクリプトであることを示す**Shebang**文字(#!)で始まります。スクリプトは、**Shebang**文字の後に指定されたインタープリタ(ここでは、/bin/sh)を使用して実行されます。
- ❷ 2行目は、ハッシュ記号で始まるコメントです。スクリプトの動作を覚えにくい行には、コメントすることをお勧めします。
- ❸ 3番目の行で、組み込みコマンドechoを使用して、各テキストをプリントします。★

このスクリプトの実行には、次の前提条件が必要です。

- 1. 各スクリプトは、**Shebang**行を含む必要があります(この例は既に示しました)。この行がスクリプトにない場合は、ユーザがインタープリタを呼び出します。
- 2. スクリプトの保存場所はどこでも構いません。ただし、シェルの検索先ディレクトリを保存場所にお勧めします。シェルのサーチパスは、環境変数PATHで設定されます。たとえば、スクリプトを~/bin/ディレクトリに、hello.shという名前で保存します。
- 3. スクリプトには、実行可能パーミッションが必要です。次のコマンドで、パーミッションを設定してください。

```
chmod +x ~/bin/hello.sh
```

これらの前提条件をすべて満たしたら、~/bin/hello.shまたはhello.shでスクリプトを実行できます。最初のコールでは、絶対パスを使用しますが、2番目のコールでは、PATH環境変数によって指定される各ディレクトリ内でコマンドを検索します。

## 6.3 コマンドイベントのリダイレクト

各コマンドは、入力または出力用として、3つのチャンネルを使用できます。:

- **標準出力** デフォルトの出力チャンネル。コマンドで何かをプリントする際には標準出力チャンネルが使用されます。
- **標準入力** コマンドでユーザまたは他のコマンドからの入力を必要とする場合は、このチャンネルが使用されます。
- **標準エラー** このチャンネルは、エラーレポーティングに使用されます。

これらのチャンネルをリダイレクトするには、次の方法を使用できます。

Command > File

コマンド出力をファイルに保存します。既存ファイルは削除されます。たとえば、lsコマンドの出力をlisting.txtファイルに書き込みます。

```
ls > listing.txt
```

Command >> File

コマンド出力をファイルに追加します。たとえば、lsコマンドの出力をlisting.txtファイルに追加します。

```
ls >> listing.txt
```

Command < File

ファイルを読み込み、指定されたコマンドへの入力とします。たとえば、ファイルのコンテンツをreadコマンドで読み込み、変数aに入力します。

```
read a < foo
```

Command1 | Command2

左側のコマンドの出力を右側のコマンドの入力にします。

各チャンネルには、対応するファイル記述子があります。標準入力には0(ゼロ)、標準出力には1、標準エラーには2が割り当てられています。このファイル記述子を<文字または>文字の前に挿入できます。たとえば、次の行では、fooで始まるファイルを検索しますが、そのファイルを/dev/nullにリダイレクトすることでエラーメッセージを抑制します。

```
find / -name "foo*" 2>/dev/null
```

## 6.4 エイリアスの使用

エイリアスは、1つ以上のコマンドのショートカット定義です。エイリアスの構文は、次の通りです。

```
alias NAME=DEFINITION
```

たとえば、次の行は、エイリアス`lt`を定義しています。このエイリアスは、長いリストを出力し(`-l`オプション)、そのリストを変更時刻でソートし(`-t`オプション)、ソート順と逆の順序でプリントします(`-r`オプション)。

```
alias lt='ls -ltr'
```

すべてのエイリアス定義を表示するには、`alias`を使用します。

## 6.5 Bashでの変数の使用

シェル変数は、グローバル変数またはローカル変数として使用できます。グローバル変数(つまり、環境変数)は、すべてのシェルでアクセスできます。対照的に、ローカル変数は、現在のシェルでのみアクセスできます。

すべての環境変数を表示するには、`printenv`コマンドを使用します。特別な変数が必要な場合は、変数名を引数として挿入します

```
printenv PATH
```

変数は、`echo`コマンドでも表示できます。

```
echo $PATH
```

この例では、`PATH`変数をプリントします。ローカル変数を設定するには、変数名の後に等号を入れ、その後に値を指定します。

```
PROJECT="SLED"
```

等号の前後にスペースを挿入しないでください。スペースを挿入すると、エラーになります。環境変数を設定するには、`export`を使用します。

```
export NAME="tux"
```

変数を削除するには、`unset`をします。

```
unset NAME
```

次のテーブルに、シェルスクリプトで利用できる共通環境変数を示します。

**表 6.5** 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	ツールをローカライズする場合、ツールは、この環境変数からの言語を使用します。英語をcに設定することも可能です。
PATH	シェルのサーチパス。コロンの区切ったディレクトリのリストを含みます。
PS1	各コマンドの前にプリントされる通常のプロンプトを指定します。
PS2	複数行コマンドの実行時にプリントされるセカンダリプロンプトを指定します。
PWD	現在の作業ディレクトリ
ユーザ	現在のユーザ

## 6.5.1 引数変数の使用

たとえば、スクリプトfoo.shは、次のように実行できます。

```
foo.sh "Tux Penguin" 2000
```

スクリプトに渡される引数すべてにアクセスするには、位置パラメータが必要です。これらのパラメータは、最初の引数には\$1、2つ目の引数には\$2という順序で割り当てます。パラメータは最大9つまで使用できます。スクリプト名を取得するには、\$0を使用します。

次のスクリプトfoo.shは、1から4までのすべての引数をプリントします。

```
#!/bin/sh
echo \"$1\" \"$2\" \"$3\" \"$4\"
```

このスクリプトを既出例の引数を使用して実行すると、次の結果が出力されます。

```
"Tux Penguin" "2000" "" ""
```

## 6.5.2 変数置換の使用

変数置換では、変数のコンテンツに、左側または右側からパターンを適用します。次のリストに、可能な構文形式を示します。

```
${VAR#pattern}  
左側から最も短い一致を削除します。
```

```
file=/home/tux/book/book.tar.bz2  
echo ${file#*/}  
home/tux/book/book.tar.bz2
```

```
${VAR##pattern}  
左側から最も長い一致を削除します。
```

```
file=/home/tux/book/book.tar.bz2  
echo ${file##*/}  
book.tar.bz2
```

```
${VAR%pattern}  
右側から最も短い一致を削除します。
```

```
file=/home/tux/book/book.tar.bz2  
echo ${file%.*}  
/home/tux/book/book.tar
```

```
${VAR%%pattern}  
右側から最も長い一致を削除します。
```

```
file=/home/tux/book/book.tar.bz2  
echo ${file%%.*}  
/home/tux/book/book
```

## 6.6 コマンドのグループ化と結合

シェルでは、条件付き実行のため、コマンドを結合し、グループ化することができます。各コマンドが返す終了コードにより、コマンドの成功または失



敗が判別されます。終了コードが0(ゼロ)の場合、コマンドは成功しました。それ以外はすべて、コマンド固有のエラーをマークします。

次のリストでは、コマンドをグループ化する方法を一覧します。

**Command1 ; Command2**

コマンドをシーケンシャルに実行します。終了コードはチェックされません。次の行では、各コマンドの終了コードにかかわらず、catでファイルのコンテンツを表示し、次に、lsでファイルプロパティをプリントします。

```
cat filelist.txt ; ls -l filelist.txt
```

**Command1 && Command2**

左のコマンドが成功した場合、右のコマンドを実行します(論理AND)。次の行では、ファイルのコンテンツを表示し、そのコマンドが成功した場合のみ、ファイルのプロパティをプリントします(このリストの前の項目と比較してください)。

```
cat filelist.txt && ls -l filelist.txt
```

**Command1 || Command2**

左のコマンドが失敗した場合、右のコマンドを実行します(論理OR)次の行では、/home/tux/fooでのディレクトリ作成に失敗した場合のみ、/home/wilber/bar内に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。位置パラメータを使用して、関数の引数にアクセスできます。次の行では、短いメッセージをプリントする関数helloを定義します。

```
hello() { echo "Hello $1"; }
```

この関数は、次のように呼び出せます。

```
hello Tux
```

結果は、次のようにプリントされます。

```
Hello Tux
```

## 6.7 よく使用されるフローコンストラクトの操作

スクリプトのフローを制御するため、シェルでは、while、if、for、およびcaseの各コンストラクトを使用します。

### 6.7.1 if制御コマンド

ifは、式のチェックに使用されます。たとえば、次のコードは、現在のユーザがTuxであるかどうかをテストします。

```
if test $USER = "tux" then
    echo "Hello Tux."
else
    echo "You are not Tux."
fi
```

テスト式は、複雑にすることも、シンプルにすることも可能です。次の式は、ファイルfoo.txtが存在するかどうかをチェックします。

```
if test -e /tmp/foo.txt
then
    echo "Found foo.txt"
fi
```

さらに多くの式を見つけるには、<http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html>を参照してください。

### 6.7.2 forコマンドによるループの作成

forループを使用すると、エントリのリストにコマンドを適用できます。たとえば、次のコードは、現在のディレクトリ内のPNGファイルの情報をプリントします。

```
for i in *.png; do
    ls -l $i
done
```

## 6.8 詳細情報

Bashに関する重要な情報は、マニュアルページ`man sh`に記載されています。このトピックの詳細については、次のリストを参照してください。

- <http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html>—「Bash Guide for」
- <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 II. システム



# 64ビットシステム環境での32ビットと64ビットのアプリケーション

# 7

SUSE® Linux Enterprise Desktopは、64ビットプラットフォームで利用できます。ただし、付属のすべ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ている訳ではありません。SUSE Linux Enterprise Desktopは、32ビットアプリケーションの64ビットシステム環境での使用をサポートしています。この章では、このサポートを64ビットSUSE Linux Enterprise Desktopプラットフォームで実装する方法について簡潔に説明します。また、32ビットアプリケーションの実行方法(ランタイムサポート)、および32ビットと64ビットのシステム環境の両方で実行できるように32ビットアプリケーションをコンパイルする方法について説明します。さらに、カーネルAPIに関する情報、および32ビットアプリケーションを64ビットカーネルで実行する方法についても説明します。

64ビットプラットフォームのamd64およびIntel 64に対応したSUSE Linux Enterprise Desktopは、既存の32ビットアプリケーションが64ビット環境で「出荷してすぐに動作するように設計されています。」このサポートにより、対応する64ビット移植版が使用可能になるのを待たなくても、使用したい32ビットアプリケーションを引き続き使用できます。

## 7.1 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競合

アプリケーションが32ビットと64ビットの両方の環境で使用可能な場合に、両方のバージョンを同時にインストールすると問題が生じます。そのような場合は、2つのバージョンのどちらかだけをインストールして使用してください。

このルールの例外は、PAM(プラグ可能認証モジュール)です。SUSE Linux Enterprise Desktopは、ユーザとアプリケーションを仲介するレイヤとしての認証プロセスでPAMを使用します。また、32ビットアプリケーションを実行する64ビットオペレーティングシステムでは、常に両バージョンのPAMモジュールをインストールする必要があります。

---

正しく実行するために、すべてのアプリケーションにはライブラリが必要です。しかし残念ながら、32ビットバージョンと64ビットバージョンのライブラリの名前は同じです。そのため、ライブラリを別の方法で区別する必要があります。

32ビットバージョンとの互換性を維持するために、ライブラリは32ビット環境の場合と同じシステム内の場所に格納されます。libc.so.6の32ビットバージョンは、32ビットと64ビットのどちらの環境でも/lib/libc.so.6の下にあります。

64ビットのすべてのライブラリとオブジェクトファイルは、lib64というディレクトリにあります。通常、/libおよび/usr/libの下にあると想定されている64ビットのオブジェクトファイルは、/lib64および/usr/lib64の下にあります。つまり、両方のバージョンのファイル名を変更しなくても済むように、32ビットライブラリ用の領域は/libおよび/usr/libの下になっています。

データの内容がワードサイズに依存しない、32ビットの/libディレクトリ中のサブディレクトリは移動されません。このスキームは、LSB (Linux Standards Base)とFHS (File System Hierarchy Standard)に準拠しています。



## 7.2 ソフトウェア開発

biarch開発ツールチェーンを使用して、32ビットと64ビットのオブジェクトを生成できます。デフォルトでは、64ビットのオブジェクトをコンパイルします。特殊なフラグを使用すれば、32ビットのオブジェクトを生成することもできます。GCCの場合、特殊なフラグは-m32です。

すべ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作成する必要があります。インストール済みの32ビットと64ビットのライブラリには、インストール済みのヘッダファイルに対応するAPI (アプリケーションプログラミングインタフェース)が必要です。標準のSUSE Linux Enterprise Desktop環境は、この原則に従って設計されています。ライブラリを手動で更新した場合は、各自でAPIの問題を解決してください。

## 7.3 biarchプラットフォームでのソフトウェアのコンパイル

biarchアーキテクチャで他のアーキテクチャ向けのバイナリを開発するには、対象のアーキテクチャのそれぞれのライブラリをさらにインストールする必要があります。これらのパッケージは、rpmname-32bitと呼ばれます。さらに、rpmname-develパッケージからの各ヘッダとライブラリ、およびrpmname-devel-32bitからの2番目のアーキテクチャ向けの開発ライブラリも必要です。

ほとんどのオープンソースプログラムでは、autoconfベースのプログラム設定が使用されています。対象のアーキテクチャ向けプログラムの設定にautoconfを使用するには、autoconfの標準のコンパイラとリンカーの設定に上書きするために、さらに環境変数を指定してconfigureスクリプトを実行します。

次の例は、対象のアーキテクチャとしてx86を採用しているx86\_64システムを示しています。

- 1 32ビットコンパイラを使用します。

```
CC="gcc -m32"
```

- 2** リンカーに32ビットオブジェクトの処理を指示します(リンカーのフロントエンドには常にgccを使用)。

```
LD="gcc -m32"
```

- 3** 32ビットオブジェクトを生成するためにアセンブラを設定します。

```
AS="gcc -c -m32"
```

- 4** libtoolなどのライブラリが/usr/libから得られたか確認します。

```
LDFLAGS="-L/usr/lib"
```

- 5** ライブラリがlibサブディレクトリに格納されているか確認します。

```
--libdir=/usr/lib
```

- 6** 32ビットXライブラリが使用されているか確認します。

```
--x-libraries=/usr/lib/xorg
```

こうした変数のすべてがどのプログラムにも必要なわけではありません。それぞれのプログラムに合わせて使用してください。

```
CC="gcc -m32" \
LDFLAGS="-L/usr/lib;" \
    .configure \
    --prefix=/usr \
    --libdir=/usr/lib
make
make install
```

## 7.4 カーネル仕様

86\_64向けの\_64ビットカーネルには、64ビットと32ビットのカーネルABI(アプリケーションバイナリインタフェース)が用意されています。32ビットのカーネルABIは、該当する32ビットカーネルのABIと同じものです。つまり、32ビットアプリケーションが、32ビットカーネルの場合と同様に64ビットカーネルと通信できるということです。

64ビットカーネルのシステムコールの32ビットエミュレーションでは、システムプログラムで使用するすべてのAPIをサポートしていません。ただし、

このサポートの有無はプラットフォームによって異なります。この理由から、`lspci`のような少数のアプリケーションをコンパイルする必要があります。

64ビットカーネルでは、このカーネル用に特別にコンパイルされた64ビットカーネルモジュールしかロードできません。したがって、32ビットカーネルモジュールを使用することはできません。

---

### ティップ

一部のアプリケーションには、カーネルでロード可能な個々のモジュールが必要です。64ビットシステム環境でそのような32ビットアプリケーションを使用する予定がある場合は、このアプリケーションのプロバイダやNovellに問い合わせて、このモジュール向けのカーネルでロード可能な64ビットバージョンのモジュールと32ビットコンパイルバージョンのカーネルAPIを入手できるかを確認してください。

---



# Linuxシステムのブートと設定

Linuxシステムのブートには、さまざまなコンポーネントが関係しています。ハードウェアはBIOSにより初期化され、BIOSはブートローダを介してカーネルを起動します。それ以後は、オペレーティングシステムがinitとランレベルを含むブートプロセスを完全にコントロールします。ランレベルのコンセプトにより、日常使用のセットアップを保持できるほか、システム上でタスクを保守することもできます。

## 8.1 Linuxのブートプロセス

Linuxのブートプロセスは、いくつかの段階から成り、それぞれを別のコンポーネントが代表しています。次のリストに、主要なすべてのコンポーネントが関与するブートプロセスと機能を簡潔にまとめています。

1. **BIOS** コンピュータに電源を投入すると、BIOSで画面とキーボードの初期化およびメインメモリのテストが行われます。この段階まで、コンピュータは大容量ストレージメディアにアクセスしません。続いて、現在の日付、時刻、および最も重要な周辺機器に関する情報が、CMOS値からロードされます。最初のハードディスクとそのジオメトリが認識されると、システム制御がBIOSからブートローダに移ります。
2. **ブートローダ** 最初のハードディスクの先頭の512バイト物理データセクタがメインメモリにロードされ、このセクタの先頭に常駐するブートローダが起動します。ブートローダによって実行されたコマンドがブートプロセスの残りの部分を確定します。したがって、最初のハードディスクの先頭512バイトのことをマスタブートレコード(MBR)といいます。

次に、ブートローダは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)に制御を渡します。GRUB(Linuxのブートローダ)の詳細については、[第9章 ブートローダGRUB](#) (89 ページ)を参照してください。

3. **カーネルとinitramfs** システムに制御を渡すために、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)をメモリにロードします。カーネルは、initramfsの内容を直接使用できます。initramfsには、実際のルートファイルシステムのマウント処理を担当するinitと言う名前の小さな実行可能ファイルが含まれています。大容量ストレージにアクセスするために特別なハードウェアドライバが必要な場合、それらはinitramfs内になければなりません。initramfsについての詳細は、[8.1.1項「initramfs」](#) (72 ページ)を参照してください。
4. **initramfs上のinit** このプログラムは、適切なルートファイルシステムをマウントするために必要なすべてのアクションを実行します。たとえば、udevに必要なファイルシステム用のカーネル機能や、大容量ストレージコントローラ用のデバイスドライバを提供します。ルートファイルシステムが見つかると、エラーをチェックしてからマウントします。これが正常に実行されれば、initramfsはクリアされ、ルートファイルシステムのinitが実行されます。initについての詳細は、[8.1.2項「initramfs上のinit」](#) (74 ページ)を参照してください。udevについての詳細は、「[第12章 udevを使用した動的カーネルデバイス管理](#)(147ページ)」を参照してください。
5. **init** initは、さまざまなレベルでシステムの実際のブートを処理し、各種の機能を提供します。initについては、[8.2項「initプロセス」](#) (75 ページ)で説明しています。

## 8.1.1 initramfs

initramfsは、カーネルがRAMディスクにロードできる、小さなcpioアーカイブです。また、実際のルートファイルシステムがマウントされる前にプログラムを実行できるようにする最低限のLinux環境を提供します。この最低限のLinux環境は、BIOSルーチンでメモリにロードされます。十分な容量のメモリがあること以外には具体的なハードウェア要件はありません。initramfsには必ず、initという名前の実行可能ファイルがあります。これは、ブートプロセスが進行するにつれて、ルートファイルシステム上の本当のinitプログラムを実行することになります。

ルートファイルシステムをマウントして実際のオペレーティングシステムを起動する前に、カーネルには、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の対応ドライバが必要です。こうしたドライバには、特定のハードディスク用の特殊なドライバや、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が含まれる場合もあります。ルートファイルシステムで必要となるモジュールは、`initramfs`上の`init`によってロードされます。モジュールをロードしたら、`udev`によって必要なデバイスが`initramfs`に提供されます。ブートプロセス後半で、ルートファイルシステムが変更された後、デバイスを再生成する必要があります。これには、`udevtrigger`コマンドで`boot.udev`を実行します。

インストール済みのシステムのハードウェア(たとえば、ハードディスク)を変更する必要が生じ、このハードウェアがブート時にカーネル内に他のドライバが存在することを必要とする場合には、`initramfs`を更新する必要があります。これは、`initramfs`の前身である`initrd`の場合と同様に、`mkinitrd`を呼び出すことによって行えます。引数を付けずに`mkinitrd`を呼び出すと、`initramfs`が作成されます。`mkinitrd -R`を呼び出すと、`initrd`が作成されます。**SUSE® Linux Enterprise Desktop**では、ロードするモジュールは`/etc/sysconfig/kernel`内の変数`INITRD_MODULES`で指定されます。インストール後、この変数は自動的に正しい値に設定されます。モジュールは、`INITRD_MODULES`に指定されている順序で正確にロードされます。このことは、デバイスファイルの`/dev/sd?`の設定の正確性に依存している場合にのみ重要になります。..ただし、現在のシステムで`/dev/disk/`ディレクトリ下にあるデバイスファイルを使用することもできます。これらのファイルは、`by-id`、`by-path`、および`by-uuid`などのサブディレクトリに分類されており、常に同じディスクを表します。これは、該当するマウントオプションの指定により、インストール時にも可能です。

---

### 重要項目: `initramfs`または`initrd`の更新

ブートローダは、カーネルと同じように`initramfs`または`initrd`をロードします。**GRUB**はブート時にディレクトリ内の正しいファイルを検索するので、`initramfs`または`initrd`の更新後に**GRUB**を再インストールする必要はありません。

---

## 8.1.2 initramfs上のinit

initramfs上のinitの主な目的は、実際のルートファイルシステムのマウントとそのファイルシステムへのアクセスの準備です。システム設定に応じて、initは次のタスクを実行します。

### カーネルモジュールのロード

ハードウェア設定によっては、使用するコンピュータのハードウェアコンポーネント(ハードディスクになる最も重要なコンポーネント)にアクセスするために特殊なドライバが必要になる場合があります。最終的なルートファイルシステムにアクセスするには、カーネルが適切なファイルシステムドライバをロードする必要があります。

### ブロック特殊ファイルの提供

ロードされるモジュールごとに、カーネル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、RAMファイルシステム上で必要なブロック特殊ファイルを/dev内に生成します。これらの特殊ファイルがないと、ファイルシステムや他のデバイスにアクセスできません。

### RAIDとLVMのセットアップの管理

RAIDまたはLVMの下でルートファイルシステムを保持するようにシステムを設定した場合、initはLVMまたはRAIDをセットアップして、後でルートファイルシステムにアクセスできるようにします。第14章 *Advanced Disk Setup* (↑導入ガイド)でRAIDとLVMに関する情報を参照してください。

### ネットワーク設定の管理

ネットワークマウントしたルートファイルシステム(NFSを介したマウント)を使用するようにシステムを設定した場合、linuxrcは適切なネットワークドライバがロードされ、ドライバがルートファイルシステムにアクセスできるように設定されていることを確認する必要があります。

初期ブート時にlinuxrcがインストールプロセスの一環として呼び出される場合、そのタスクは前に説明したタスクと異なります。

### インストールメディアの検出

インストールプロセスを開始すると、使用するコンピュータでは、YaSTインストーラでインストールカーネルと特殊なinitrdがインストールメディアからロードされます。RAMファイルシステムで実行されるYaSTインストーラには、インストールメディアにアクセスしてオペレーティングシス



テムをインストールするために、そのメディアの場所に関する情報が必要になります。

ハードウェア認識の開始および適切なカーネルモジュールのロード

で説明しているように、ブートプロセスは、ほとんどのハードウェア設定で利用できる最小限のドライバセットで開始されます。`init`は、ハードウェア設定に適したドライバセットを確定する、初期ハードウェアスキャンプロセスを開始します。**8.1.1項「initramfs」** (72 ページ)ブートプロセスに必要なモジュール名は、`/etc/sysconfig/kernel`ディレクトリ中の `INITRD_MODULES`に書き込まれます。これらのモジュール名は、システムをブートするために必要なカスタム `initramfs`を生成するために使用されます。ブートではなく `coldplug`で必要なモジュールは、`/etc/sysconfig/hardware/hwconfig-*`ディレクトリに書き込まれます。ブートプロセス時には、このディレクトリ中の設定ファイルに記述されているすべてのデバイスが初期化されます。

インストールシステムまたはレスキューシステムのロード

ハードウェアが正しく認識され、適切なドライバがロードされ、`udev`がデバイス特定ファイルを作成するとすぐに、`linuxrc`はインストールシステムを起動します。このシステムには、実際の `YaST` インストーラまたはレスキューシステムが含まれています。

`YaST`の開始

最後に、`init`は `YaST` を起動します。これはパッケージのインストールとシステム設定を開始します。

## 8.2 `init` プロセス

`init` プログラムは、プロセスIDが1のプロセスです。このプロセスは、システムの初期化を担当しています。`init`は直接カーネルから起動し、プロセスを強制終了する `signal9`で終了することはできません。他のすべてのプログラムは、`init`または子プロセスのいずれかによって直接起動されます。

`init`の中心的な設定は、`/etc/inittab`ファイルで行われています。このファイルはランレベルを定義しています(**8.2.1項「ランレベル」** (76 ページ)を参照)。このファイルはまた、各ランレベルで利用可能なサービスとデーモンを指定しています。`/etc/inittab`のエントリに応じて、`init`が複数のスクリプトを実行します。デフォルトでは、ブート後に最初に開始するスクリプト

は、`/etc/init.d/boot`です。システムの初期設定が完了すると、`/etc/init.d/rc`スクリプトで、ランレベルがデフォルトのランレベルに変更されます。わかりやすくするために、これらの`init`スクリプトと呼ばれるスクリプトはすべて、ディレクトリ`/etc/init.d`にあります(8.2.2項「`init`スクリプト」(79 ページ)を参照)。

システムを起動し、シャットダウンするプロセス全体は、`init`によって管理されます。この点から見ると、カーネルは、他のプログラムからの要求に従って、他のすべてのプロセスとCPU時間や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と考えることができます。

## 8.2.1 ランレベル

Linuxでは、ランレベルはシステムの起動方法および稼働中のシステムで使用可能なサービスを定義します。ブート後、システムは`/etc/inittab`の`initdefault`行での定義に従って起動します。通常のランレベルは3または5です。参照先表 8.1.「ランレベルの種類」(76 ページ)。別の方法として、ランレベルをブート時に(たとえばブートプロンプトにランレベル番号を追加する)指定することもできます。パラメータは、カーネル自体が直接評価するもの以外、`init`に渡されます。ランレベル3にブートするには、ブートプロンプトに単一番号3を追加します。

表 8.1 ランレベルの種類

ランレベル	説明
0	システム停止
Sまたは1	シングルユーザモード
2	リモートネットワーク(NFSなど)なしのローカルマルチユーザモード
3	ネットワークを使用するフルマルチユーザモード
4	[ユーザ定義]。管理者が設定しない限り使用されないランレベル。

ランレベル	説明
5	ネットワークとXディスプレイマネージャのKDM、GDM、またはXDMを使用するフルマルチユーザモード
6	システム再起動

---

**重要項目:** パーティションが**NFS**マウントされている場合には**ランレベル2**は避ける

システムで**NFS**を介して/usrなどのパーティションをマウントする場合は、ランレベル 2を使用しないでください。NFSサービスは、ランレベル2(リモートネットワークのないローカルマルチユーザモード)では使用できないため、プログラムファイルまたはライブラリがない場合、システムは予想しない動作をする可能性があります。

---

システムの稼動中にランレベルを変更するには、telinitの後に、ランレベルに対応する番号を引数として入力します。これができるのは、システム管理者だけです。次のリストは、ランレベルに関連した最も重要なコマンドの概要です。

telinit 1またはshutdown now

システムはシングルユーザモードに入ります。このモードは、システムメンテナンスや管理タスクで使用します。

telinit 3

(ネットワークを含む)すべての重要なプログラムとサービスが起動します。グラフィック環境はありませんが、一般ユーザは、システムにログインして作業することができます。

telinit 5

グラフィック環境は有効になります。通常、XDM、GDMまたはKDMなどのディスプレイマネージャが起動します。自動ログインが有効な場合、ローカルユーザは事前選択されているウィンドウマネージャ(GNOME、KDEまたはその他のウィンドウマネージャ)にログインします。

telinit 0またはshutdown -h now

システムは停止します。

telinit 6またはshutdown -r now  
システムは停止した後、再起動します。

ランレベル5は、すべてのSUSE Linux Enterprise Desktop標準インストールにおけるデフォルトのランレベルです。ユーザは、グラフィカルインタフェースでログインするように求められます。デフォルトユーザの場合は自動的にログインされます。デフォルトのランレベルは3で、ランレベルを5に切り替えるには、**第13章 X Windowシステム** (161 ページ)で説明するようにX Window Systemを正しく設定している必要があります。その後、telinit5を入力して、システムが意図したとおりに動作するかを確認します。すべてが意図したとおりに動作した場合は、YaSTを使用してデフォルトのランレベルを5に設定します。

---

**警告: /etc/inittab内のエラーのためシステムブートが失敗することがある**

---

/etc/inittabが破損した場合、システムが正しく起動しないことがあります。そのため、/etc/inittabを編集する場合は細心の注意を払ってください。また、コンピュータを再起動する前には、常にtelinitqコマンドを使って、initに/etc/inittabを再読み込みさせるようにしてください。

---

ランレベルを変更するときには、一般に2つの操作が行われます。1つは、現在のランレベルの停止スクリプトが起動し、現在のランレベルに必要なプログラムを終了します。次に、新しいランレベルの起動スクリプトが起動します。ここで、ほとんどの場合、プログラムがいくつか起動します。たとえば、ランレベルを3から5に変更する場合、次の操作が行われます。

1. 管理者(root)がtelinit 5を入力して、initにランレベルを変更することを伝えます。
2. initは現在のランレベル(runlevel)を調べ、新しいランレベルをパラメータとして/etc/init.d/rcを起動する必要があるかどうか判断します。
3. ここでrcは、現在のランレベルの停止スクリプトであって、新しいランレベルの起動スクリプトがないものを呼び出します。この例では、元のランレベルが3なので、/etc/init.d/rc 3.dの中のKで始まるすべてのスクリプトが対象となります。Kの次の番号は、stopパラメータを使ってスクリプトを実行する順番を示します(検討する必要がある依存関係が存在するため)。

- 最後に、新しいランレベルの起動スクリプトを起動します。この例では/etc/init.d/rc5.dの中のsで始まるスクリプトがそれにあたります。この場合も、sの次の番号が、スクリプトの実行順序を表します。

現在のランレベルと同じランレベルに変更する場合、`init`は/etc/inittabの変更部分だけをチェックし、適切な手順を開始します。たとえば、別のインタフェースで`getty`を起動します。`telinit q`コマンドを使用しても同じ操作を実行できます。

## 8.2.2 initスクリプト

/etc/init.d内に、2種類のスクリプトがあります。

`init`によって直接実行されるスクリプト

これは、ブートプロセスの実行中、または即座のシステムシャットダウンを行ったとき(電源障害またはユーザが`Ctrl+Alt+Del`キーを押した場合)にのみ適用されます。こうしたスクリプトの実行は、/etc/inittabで定義されます。

`init`によって間接的に実行されるスクリプト

これらは、ランレベルの変更時に実行され、関連スクリプトの正しい順序を保証するマスタスクリプト/etc/init.d/rcを常に呼び出します。

すべてのスクリプトは、/etc/init.dにあります。ブート時に実行されるスクリプトは、/etc/init.d/boot.dからのシンボリックリンク経由で呼び出されます。ランレベルを変更するスクリプトもサブディレクトリの1つからのシンボリックリンク(/etc/init.d/rc0.dから/etc/init.d/rc6.dへ)経由で呼び出されます。これは単にわかりやすくして、複数のランレベルで使用されている場合にスクリプトが重複するのを防ぐためです。すべてのスクリプトは、起動スクリプトとしても停止スクリプトとしても実行できるので、これらのスクリプトはパラメータの`start`と`stop`を認識する必要があります。また、これらのスクリプトは`restart`、`reload`、`force-reload`、および`status`のオプションも認識します。これらのオプションについては、**表 8.2. 「initスクリプトのオプション」** (80 ページ)で説明します。`init`によって直接実行されるスクリプトには、このようなリンクはありません。こうしたスクリプトは、必要なときにランレベルとは無関係に実行されます。

表 8.2 *init* スクリプトのオプション

オプション	説明
起動	サービスを起動します。
中止	サービスを停止します。
restart	サービスが実行中の場合は、停止して再起動します。実行中でない場合は、起動します。
reload	サービスの停止や再起動をせずに、設定を再ロードします。
force-reload	サービスが設定の再ロードをサポートする場合は、それを実行します。サポートしない場合は、restartが指定された場合と同じ操作を行います。
ステータス	サービスの現在のステータスを表示します。

ランレベル固有のサブディレクトリにあるリンクによって、スクリプトを複数のランレベルに関連付けることができます。パッケージのインストールまたはアンインストール時に、プログラム`insserv`を使用して(またはこのプログラムを呼び出す`/usr/lib/lsb/install_initd`スクリプトを使用して)、このようなリンクを追加または削除することができます。詳細は、`insserv(8)`のmanページを参照してください。

これらの設定は、YaSTモジュールにより変更されることもあります。コマンドラインからステータスを確認するには、`chkconfig`ツールを使います。このツールの詳細は、`chkconfig(8)`マニュアルページを参照してください。

次に、最初または最後に起動するブートスクリプトおよび停止スクリプトの概略を示すとともに、保守スクリプトについて説明します。

#### boot

`init`を直接使用してシステムを起動するときに実行されます。選択したランレベルから独立で、一度だけ実行されます。これによって `/proc` ファイルシステムと `/dev/pts` ファイルシステムがマウントされ、`blogd`(ブートログ出力デーモン)が有効化されます。システムがアップデートまたは

インストール後初めてブートされる場合、初期システム設定が起動します。

**blogd**デーモンは、**boot**および**rc**によって最初に起動されるサービスです。また、これらのスクリプトにより開始されたアクション(サブスクリプトの実行、たとえばブロック特殊ファイルを利用可能にする)が完了すると停止します。**blogd**は、**/var**が読み書き可能でマウントされている場合にのみ、画面出力をログファイル**/var/log/boot.msg**に出力します。そうでない場合は、**/var**が利用できるようになるまで、**blogd**がすべての画面データをバッファします。**blogd**についての詳細は、**blogd(8)**の**man**ページを参照してください。

**boot**スクリプトはまた、**/etc/init.d/boot.d**の中の**S**で始まる名前のスクリプトをすべて起動します。そこで、ファイルシステムがチェックされ、必要に応じてループデバイスが設定されます。加えて、システム時間が設定されます。ファイルシステムの自動チェックや修復中にエラーが発生した場合、システム管理者はルートパスワードを入力して介入することができます。最後に実行されるスクリプトは、**boot.local**です。

#### **boot.local**

ブート時、ランレベルへの移行前に実行する追加のコマンドを入力します。これは、**DOS**システムの**AUTOEXEC.BAT**に相当します。

#### **halt**

このスクリプトは、ランレベル**0**または**6**への移行時のみ実行され、**halt**または**reboot**として機能します。システムがシャットダウンするからブートするかは、**halt**の呼び出され方に依存します。シャットダウン時に特別なコマンドが必要な場合は、それらのコマンドを**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、現在のランレベルの適切な停止スクリプトと、新しく選択したランレベルの起動スクリプトを呼び出します。**Like the /etc/init.d/boot**スクリプトと同様、このスクリプトは、目的のランレベルをパラメータとして使用して、**/etc/inittab**から呼び出します。

独自のスクリプトを作成して、先に説明したスキーマに容易に組み込むことができます。カスタムスクリプトの形式、名前付け、および構成方法は、**LSB**の仕様と、**init**、**init.d**、**chkconfig**、および**insserv**のマニュアルペー

ジを参照してください。加えて、startprocおよびkillprocのマニュアルページも参照してください。

---

### 警告: initスクリプトのエラーはシステムの停止につながる

initスクリプトに問題があると、コンピュータがハングアップします。このようなスクリプトは最大限の注意を払って編集し、可能であれば、マルチユーザ環境で徹底的にテストします。initスクリプトに関する他の情報は、**8.2.1項 「ランレベル」 (76 ページ)**を参照してください。

---

特定のプログラムまたはサービス用にカスタムのinitスクリプトを作成する場合は、テンプレートとしてファイル/etc/init.d/skeletonを使用します。このファイルのコピーを別名で保存し、関連のプログラムやファイル名、パス、その他の詳細を必要に応じて編集します。また場合によっては、initプロシージャで正しいアクションが実行されるように、独自の改良をスクリプトに加える必要があります。

最初に記載されているINIT INFOブロックはスクリプトの必須部分で、次のように編集する必要があります。詳細については、**例 8.1. 「最低限のINIT INFOブロック」 (82 ページ)**を参照してください。

#### 例 8.1 最低限の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ブロックの最初の行では、Provides:の後に、このinitスクリプトで制御するプログラムまたはサービスの名前を指定します。「Required-Start:」および「Required-Stop:」行には、このサービスを開始/停止する前に開始/停止する必要があるすべてのサービスを指定します。この情報は後で、ランレベルディレクトリに表示するスクリプト名に対し、番号を生成するために使用します。Default-Start:およびDefault-Stop:の後に、サービスが自動的に起動または停止する際のランレベルを指定します。最後に、Description:の下に、対象のサービスについての簡単な説明を記載します。



ランレベルディレクトリ(/etc/init.d/rc?.d/)から/etc/init.d/内の対応するスクリプトへのリンクを作成するには、コマンド`insserv <新しいスクリプト名>`を入力します。`insserv`プログラムは、INIT INFOヘッダを評価して、ランレベルディレクトリ(/etc/init.d/rc?.d/)のスクリプトを起動、停止するために必要なリンクを作成します。このプログラムはまた、必要な番号をこれらのリンクの名前に取り込むことによって、ランレベルごとに正しい起動、停止の順序を管理します。グラフィックツールを使用してリンクを作成する場合は、[8.2.3項「YaSTでのシステムサービス\(ランレベル\)の設定;](#)」(83 ページ)の説明に従って、YaSTのランレベルエディタを使用します。

/etc/init.d/にすでに存在するスクリプトを既存のランレベルスキーマに統合する場合は、はじめに`insserv`を使用するか、YaSTのランレベルエディタで対応するサービスを有効にすることにより、ランレベルディレクトリにリンクを作成します。変更内容は、次のブート時に適用され、新しいサービスが自動的に起動します。

作成したリンクは手動で設定しないでください。INFOブロック内に誤りがある場合は、後で他のサービスに対して`insserv`を実行すると問題が生じます。手動で追加されたサービスは、このスクリプトに対する`insserv`の次回実行時に削除されます。

## 8.2.3 YaSTでのシステムサービス(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)] の順に選択して、このYaST moduleを起動すると、利用可能なすべてのサービスの概要と、各サービスの現在のステータス(有効か無効か)が表示されます。モジュールを[単純モード]と[エキスパートモード]のどちらで使用するかを決定します。ほとんどの場合、デフォルトの[単純モード]で十分です。左の列にはサービスの名前が、中央の列にはその現在のステータスが、右の列には簡単な説明が表示されます。ウィンドウの下部には、選択したサービスについての詳細な説明が表示されます。サービスを有効にするには、表でそれを選択し、[有効にする]を選択します。同じ手順で、サービスを無効にできます。

サービスの起動または停止時のランレベルを詳細に制御する場合、またはデフォルトのランレベルを変更する場合は、最初に[エキスパートモード]を選択します。上部には、現在のデフォルトのランレベル、つまり「initdefault」

(システムのブート時にデフォルトで入るランレベル)が表示されます。通常、SUSE Linux Enterprise Desktopシステムのデフォルトのランレベルは、5(ネットワークありフルマルチユーザモードおよびX)です。適切な代替の設定は、ランレベル3(ネットワークありフルマルチユーザモード)です。

YaSTのダイアログボックスでは、ランレベルのいずれか1つを新しいデフォルトとして選択できます(表 8.1. 「ランレベルの種類」 (76 ページ)を参照)。また、このウィンドウのテーブルを使用して、個々のサービスやデーモンを有効、無効にできます。テーブルには、利用可能なサービスとデーモンが一覧表示され、現在システム上で有効かどうかと、有効な場合はそのランレベルが表示されます。マウスで行を選択し、ランレベルを表すチェックボックス( *[B]*、*[0]*、*[1]*、*[2]*、*[3]*、*[5]*、*[6]*、*[S]* )をクリックして、選択しているサービスまたはデーモンが実行されるランレベルを定義します。ランレベル4は、カスタムランレベルを作成できるように未定義になっています。最後に現在選択しているサービスまたはデーモンの簡単な説明が、テーブルの概要の下に表示されます。

---

**警告: ランレベルの設定を誤るとシステムに害が及ぶことがある**

ランレベルの設定が誤っていると、システムを使用できなくなることがあります。変更を実際に適用する前に、どういう結果が出るかをよく確認してください。

---

## ☒ 8.1 システムサービス(ランレベル)

 **システムサービス (ランレベル): サービス**  
ここではどのシステムサービスを起動するかを指定してください。注意: システムサービス (ランレベルエディタ) は熟練者... [続き](#)

☒ 簡易モード (S)    ☐ 熟練者モード (E)

サービス	有効	説明
SuSEfirewall2_init	はい	SuSEfirewall2 phase 1
SuSEfirewall2_setup	はい	SuSEfirewall2 phase 2
a11y	はい	enables a11y support on livecd
aaeventd	いいえ*	AppArmor Notification and Reporting
acpid	はい	Listen and dispatch ACPI events from the kernel
alsasound	はい	Set up ALSA sound system
amavis	いいえ	Start amavisd-new
atd	いいえ	Start AT batch job daemon
auditd	はい	auditd daemon providing core auditing services
autofs	いいえ	automatic mounting of filesystems
autoyast	いいえ*	A start script to execute autoyast scripts
avahi-daemon	はい	ZeroConf daemon
avahi-daemon	いいえ	ZeroConf daemon

[スタート/中止/更新] をクリックして、サービスを有効化するかどうかを決定します。現在の状態が自動的に確認されなかった場合は、[状態を更新] を使用して確認することができます。[設定/リセット] をクリックすると、変更をシステムに適用するか、ランレベルエディタの起動前に存在していた設定を復元するかを選択できます。[完了] を選択すると、設定の変更がディスクに保存されます。

## 8.3 /etc/sysconfigによるシステム設定

SUSE Linux Enterprise Desktopの主な設定は、/etc/sysconfigに格納されている設定ファイルで指定できます。/etc/sysconfigディレクトリの個々のファイルは、それらが関係するスクリプトによってのみ読み込まれます。これにより、たとえば、ネットワークはネットワーク関連のスクリプトでのみ解析されるようになります。

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のsysconfigエディターを使う方法と、設定ファイルを手動で編集する方法です。

## 8.3.1 YaSTのsysconfigエディターを使ってシステム設定を変更する

YaSTのsysconfigエディタは、使いやすいシステム設定のフロントエンドです。変更する必要がある設定用変数の実際の場所が分からなくても、このモジュールに内蔵された検索機能を使うだけで、必要に応じて設定用変数の値を変更できますし、これらの変更の適用、sysconfigで設定されている値に基づく設定の更新、サービスのリスタートは、YaSTが行います。

---

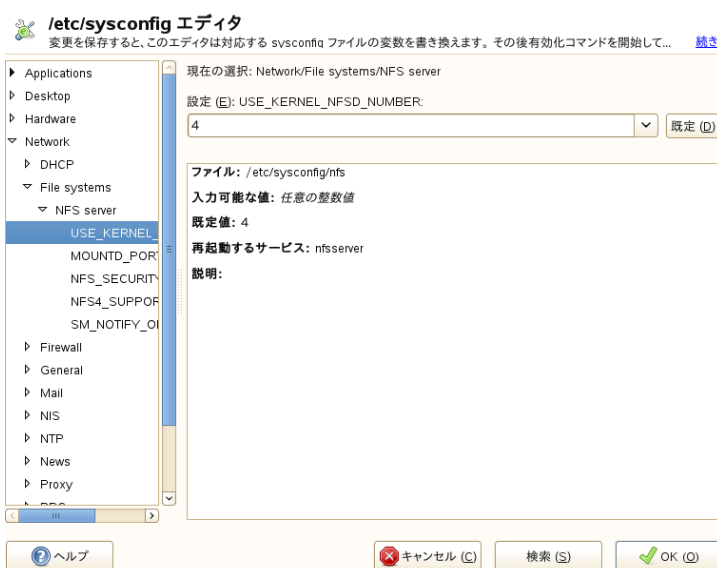
**警告:** `/etc/sysconfig/*`ファイルの変更はインストールに害を及ぼすことがある

---

知識や経験が豊富でない限り、`/etc/sysconfig`ファイルは変更しないでください。場合によっては、システムに相当なダメージを与えることがあります。`/etc/sysconfig`のファイルには、各変数が持つ実際の効果を説明する簡単なコメントが付いています。

---

図 8.2 sysconfigエディタを使用したシステム設定



YaSTのsysconfigダイアログは、3つの部分に分かれています。ダイアログの左側には、すべての設定変数がツリー表示されます。変数を選択した段階で、右側に現在選択されている変数と、この変数の現在の設定が表示されます。その下の3番目のウィンドウには、変数の目的、有効な値、デフォルト値、およびこの変数が設定されている実際の設定ファイルについての簡単な説明が表示されます。このダイアログボックスには、変数の変更後に実行された設定スクリプトや、変更の結果起動された新しいサービスについての情報も表示されます。YaSTにより変更の確認が求められ、**[完了]**を選択してダイアログを終了した後にはどのスクリプトが実行されるかが通知されます。現在は実行しないサービスやスクリプトを選択すると、それらが後で実行されます。YaSTはすべての変更を自動的に適用し、変更と関係のあるすべてのサービスをリスタートします。

## 8.3.2 システム設定を手動で変更する

システム設定を手動で変更するには、以下の手順に従います。

- 1 rootになります。
- 2 `telinit 1`コマンドで、システムをシングルユーザモード(ランレベル 1)にします。
- 3 必要に応じて、設定ファイルを、自分が使っているエディタで変更します。

`/etc/sysconfig`の設定ファイルの変更にYaSTを使用しない場合、空の変数値は2つの引用符(`KEYTABLE=""`)によって表し、空白を含む値へ引用符で囲むことに注意してください。語の値は、引用符で囲む必要はありません。

- 4 `SuSEconfig`を実行して、変更が有効になっていることを確認します。
- 5 `telinit default_runlevel`などのコマンドで、システムを以前のランレベルに戻します。`default_runlevel`の部分は、システムのデフォルトのランレベルで置き換えてください。ネットワークとXのあるフルマルチユーザモードに戻るには5を、ネットワークのあるフルマルチユーザa[d]に戻るには3を選択します。

この手順は主に、ネットワーク設定など、システム全体の設定を変更する場合に必要です。小さな変更であれば、シングルユーザモードに移行する必要はありませんが、関与するすべてのプログラムが正しく再起動することを絶対的に保証する必要がある場合は、移行しても差し支えありません。

---

#### ティップ: 自動システム設定機能の設定

**SuSEconfig**の自動システム設定機能を無効にするに

は、`/etc/sysconfig/suseconfig`の`ENABLE_SUSECONFIG`を`no`に設定します。**SUSE**のインストールサポートを使用する場合は、**SuSEconfig**を無効にしないでください。無効にすると、自動設定も部分的に無効になる可能性があります。

---

# ブートローダGRUB

この章では、SUSE® Linux Enterprise Desktopで使用されているブートローダGRUBの設定方法について説明します。すべての設定操作には、特殊なYaSTモジュールを使用できます。Linuxでのブートに不慣れな場合は、以降の各セクションを読んで背景情報を理解してください。また、この章では、GRUBでのブート時に頻繁に発生する問題とその解決策についても説明します。

---

**注意: UEFIを使用するコンピュータ上にGRUBがない**

通常GRUBは従来のBIOSを備え、UEFI (Unified Extensible Firmware Interface) 上にあるコンピュータにインストールされます。CSMが有効になっていないUEFIコンピュータでは、eLILOが自動的にインストールされます (DVD1が正常に起動した場合)。詳細については、ご使用のシステムの/usr/share/doc/packages/elilo/にあるeLILOマニュアルを参照してください。

---

この章は、ブート管理とGRUBブートローダの設定に重点を置いています。ブート手順は、総じて第8章 **Linuxシステムのブートと設定** (71 ページ) で説明しています。ブートローダは、マシン(BIOS)とオペレーティングシステム(SUSE Linux Enterprise Desktop)の間のインタフェースになります。ブートローダの設定は、オペレーティングシステムの起動に直接影響を及ぼします。

次の用語は、この章で頻繁に使用されており、少し説明を加えた方がよいと思われるものです。

マスターブートレコード

MBRの構造は、オペレーティングシステムに依存しない規則に従って定義されます。最初の446バイトは、プログラムコード用に予約されています。

す。通常、ここにはブートローダプログラムやオペレーティングシステムセクタの一部が保管されています。次の64バイトは、最大4つのエントリからなるパーティションテーブル用のスペースです。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分割とファイルシステムのタイプに関する情報が含まれています。オペレーティングシステムでハードディスクを処理するには、このテーブルが必要です。MBRの従来の汎用コードでは、1つのパーティションにだけアクティブのマークを付ける必要があります。MBRの最後の2バイトは、静的な「マジックナンバー」(AA55)を含む必要があります。一部のBIOSでは、異なる値を持つMBRは無効とみなされ、ブートの対象とはみなされません。

### ブートセクタ

ブートセクタは、拡張パーティションを除くハードディスクパーティションの最初のセクタであり、その他のパーティションの「コンテナ」として機能するだけです。これらのブートセクタのうち512バイトのスペースは、関連パーティションにインストールされているオペレーティングシステムをブートするためのコードが占有します。これは、フォーマット済みのDOS、Windows、およびOS/2パーティションのブートセクタに該当し、ファイルシステムの重要な基本データも一部含まれています。これに対して、Linuxパーティションのブートセクタは、XFS以外のファイルシステムの設定直後は当初空になっています。そのため、Linuxパーティションは、カーネルと有効なルートファイルシステムが含まれている場合にも、単独ではブートできません。システムブート用の有効なコードを含むブートセクタの場合、最後の2バイトにはMBRと同じマジックナンバー(AA55)があります。

## 9.1 GRUBによるブート

GRUB(Grand Unified Bootloader)は、2つのステージから成り立っています。stage1は512バイトで構成され、そのタスクは、ブートローダのstage2をロードすることだけです。その後、stage2が読み込まれます。このステージには、ブートローダの主要部分が含まれています。

一部の設定では、適切なファイルシステムからステージ2を検出し、ロードする中間ステージの1.5を使用できます。可能であれば、デフォルトでインストール時、またはYaSTを使用したGRUBの初回セットアップ時に、こ



stage2は、多くのファイルシステムにアクセスできます。現在、Windowsで使用されているExt2、Ext3、ReiserFS、Minix、およびDOS FATファイルシステムがサポートされます。BSDシステムで使用されているXFS、UFS、およびFFSも、特定の範囲までサポートされます。バージョン 0.95以降のGRUBには、「El Torito」仕様に準拠するISO 9660標準ファイルシステムを含むCDまたはDVDからブートする機能も用意されています。システムをブートする前にも、GRUBはサポートされているBIOSディスクデバイス(BIOSにより検出されるフロッピーディスクまたはハードディスク、CDドライブ、およびDVDドライブ)のファイルシステムにアクセスできます。したがって、GRUBの設定ファイル(menu.lst)を変更しても、ブートマネージャを新たにインストールする必要はありません。システムをブートすると、GRUBはメニューファイルと共にカーネルまたは初期RAMディスク(initrd)の有効なパスとパーティションデータを再読み込みし、これらのファイルを検索します。

GRUBの実際の設定は、以下の3つのファイルに基づきます。

/boot/grub/menu.lst

このファイルには、GRUBでブートできるパーティションまたはオペレーティングシステムに関する情報がすべて含まれています。この情報がない場合、GRUBコマンドラインは、どのように処理を続行するかユーザの指示を求めます(詳細については、「[ブート手順実行中のメニューエントリの編集項 \(97 ページ\)](#)」を参照してください)。

/boot/grub/device.map

このファイルは、デバイス名をGRUBとBIOSの表記法からLinuxデバイス名に変換するために使います。

/etc/grub.conf

このファイルには、GRUBシェルでブートローダを正常にインストールするために必要なコマンド、パラメータおよびオプションが含まれています。

GRUBは、さまざまな方法で制御できます。グラフィカルメニュー(スプラッシュ画面)を使用して、既存の設定からブートエントリを選択できます。設定は、ファイルmenu.lstから読み込まれます。

GRUBでは、すべてのブートパラメータをブート前に変更できます。たとえば、メニューファイルを間違えて編集した場合は、この方法で訂正できます。また、一種の入力プロンプトからブートコマンドを対話形式で入力することもできます(「[ブート手順実行中のメニューエントリの編集項 \(97 ページ\)](#)」を

参照してください)。GRUBには、ブート前にカーネルとinitrdの位置を判別する機能が用意されています。この機能を使用すると、ブートローダ設定にエントリが存在しないインストール済みオペレーティングシステムでもブートできます。

GRUBは、2種類のバージョンで存在します。ブートローダとして、または/usr/sbin/grub中のLinuxプログラムとしてです。このプログラムをGRUBシェルと呼びます。GRUBシェルは、インストールされたシステムにGRUBのエミュレーションを提供し、GRUBのインストールまたは新規設定の適用前のテストに使用できます。ハードディスクやフロッピーディスクにGRUBをブートローダとしてインストールする機能は、installコマンドとsetupコマンドの形でGRUBに組み込まれています。この機能は、Linuxの読み込み時にGRUBシェル内で使用できます。

## 9.1.1 GRUBのブートメニュー

ブートメニューを含むグラフィカルスプラッシュ画面は、GRUBの設定ファイル/boot/grub/menu.lstに基づいており、このファイルにはメニューを使用してブートできるパーティションまたはオペレーティングシステムに関する情報がすべて含まれています。

システムをブートするたびに、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。このため、ファイルを変更するたびにGRUBを再インストールする必要があります。9.2項「YaSTによるブートローダの設定;」(100ページ)で説明しているように、YaSTのブートローダを使用してGRUBの設定を変更します。

メニューファイルにはコマンドが含まれています。構文はきわめて単純です。各行には、コマンド1つとオプションのパラメータがシェルと同様にスペースで区切って指定されています。これまでの経緯が理由で、一部のコマンドでは最初の引数の前に等号(=)を使用することができます。コメントを記述するには、行頭にシャープ記号(#)を入力します。

メニュー概要の中にあるメニュー項目を識別できるように、各エントリに対してtitle(タイトル)を設定します。キーワードtitleの後に続くテキスト(半角スペースも使用できます)は、メニューの中で、選択可能なオプションとして表示されます。そのメニュー項目が表示された場合、次のtitleまでに記述されているすべてのコマンドが実行されます。

最も簡単な例は、他のオペレーティングシステムのブートローダにリダイレクトすることです。該当するコマンドはchainloaderであり、引数は通常、他のパーティション内にあるブートブロックをGRUBのブロック表記に従って記述したものです。たとえば、次のようにします。

```
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に関する命名規則項 (94 ページ)**を参照してください。この例では、1台目のハードディスクの4番目のパーティションの最初のブロックを指定しています。

カーネルイメージを指定するには、kernelコマンドを使用します。最初の引数は、パーティションにあるカーネルイメージを表すパスです。他の引数は、そのコマンドラインでカーネルに渡されます。

ルートパーティションへのアクセスに必要なビルトインドライバがカーネルに用意されていない場合、または高度なhotplug機能のある新しいLinuxシステムが使用されていない場合は、initrdファイルへのパスを示す引数だけを指定して、別のGRUBコマンドでinitrdを指定する必要があります。initrdのロードアドレスは、ロードされるカーネルイメージに書き込まれるので、initrdコマンドは、kernelコマンドの後に記述する必要があります。

rootコマンドは、kernelとinitrdの各ファイルの指定を簡略化します。rootの引数は、デバイスまたはパーティションだけです。このデバイスは、すべてのカーネル、initrd、または次のrootコマンドまでデバイスが明示的に指定されて「ない他のファイルのパスに使用されます。

bootコマンドは各メニューエントリの最後に必ず含まれています。そのため、メニューファイルにこのコマンドを記述する必要はありません。ただし、GRUBをブート時に対話形式で使用する場合は、bootコマンドを最後に入力する必要があります。このコマンド自体は、引数を使用しません。単純に、読み込み済みのカーネルイメージ、または指定のチェーンローダをブートします。

すべてのメニューエントリを記述した後、その1つをdefaultエントリとして定義します。デフォルトエントリを指定しなかった場合、最初のエントリ(エントリ0)が使用されます。デフォルトエントリがブートされるまでのタイムアウトを秒単位で指定することもできます。通常、timeout およびdefaultは、メニューエントリより先に記述します。サンプルファイルについては、**メニューファイルの例項 (95 ページ)**を参照してください。

## ハードディスクとパーティションに関する命名規則

GRUBでのハードディスクとパーティションの命名規則は、通常のLinuxデバイスの命名規則と異なっています。どちらかという、BIOSが使用する単純なディスクエミュレーションに似ており、構文は一部のBSDデリバティブで使用されているものに類似しています。GRUBでは、パーティション番号は0から始まります。これは、(hd0, 0)は最初のハードディスクの最初のパーティションになります。ハードディスクがプライマリマスタとして接続されている一般的なデスクトップマシンでは、対応するLinuxデバイス名は/dev/sda1になります。

可能な4つの基本パーティションに、パーティション番号}0～3が割り当てられます。論理パーティション番号は4から始まります。

```
(hd0,0)    first primary partition of the first hard disk
(hd0,1)    second primary partition
(hd0,2)    third primary partition
(hd0,3)    fourth primary partition (usually an extended partition)
(hd0,4)    first logical partition
(hd0,5)    second logical partition
```

GRUBは、BIOSデバイスに依存するため、IDE、SATA、SCSIおよびハードウェアRAIDデバイス間を区別しません。BIOSまたは他のディスクコントローラで認識されるすべてのハードディスクには、BIOSの中で事前に設定されたブートシーケンスに従って番号が割り当てられます。

一般に、GRUBには、Linuxデバイス名をBIOSデバイス名に正確にマップする機能がありません。このマッピングはアルゴリズムを使用して生成され、device.mapファイルに保存されるため、必要に応じて編集できます。ファイルdevice.mapについては、[9.1.2項「device.mapファイル」](#) (98 ページ)を参照してください。

GRUBのフルパスは、カッコ内のデバイス名と、指定のパーティションにあるファイルシステム内のファイルへのパスで構成されます。このパスはスラッシュで始まります。たとえば、単一IDEハードディスクの最初のパーティションにLinuxを含んでいるシステムでは、ブート可能カーネルを次のように指定できます。

```
(hd0,0)/boot/vmlinuz
```

## メニューファイルの例

次の例は、**GRUB**のメニューファイルの構造を示しています。このインストール例では、**Linux**のブートパーティションが/dev/sda5、ルートパーティションが/dev/sda7、および**Windows**のインストールファイルが/dev/sda1にあります。

```
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
```

最初のブロックは、スプラッシュ画面の設定を定義します。

**gfxmenu (hd0,4)/message**

背景画像messageは、/dev/sda5パーティションの最上位ディレクトリにあります。

**color white/blue black/light-gray**

カラースキーム:白(前景色)、青(背景色)、黒(選択項目)、明るい灰色(選択項目の背景色)です。配色はスプラッシュ画面には影響しません。影響を受けるのは、**Esc**キーを押してスプラッシュ画面を終了するとアクセスできるカスタマイズ可能な**GRUB**メニューだけです。

**default 0**

最初のメニューエントリ**title linux**は、デフォルトでのブート対象です。

## timeout 8

ユーザ入力がないまま8秒が経過した場合、**GRUB**は自動的にデフォルトエントリをブートします。自動ブートを無効にするには、`timeout`の行を削除します。`timeout 0`と設定すると、**GRUB**は待ち時間なしでデフォルトのエントリをブートします。

2番目の(最大)ブロックは、ブート可能な各種オペレーティングシステムを示します。個々のオペレーティングシステムに関するセクションはtitleで始まります。

- 最初のエントリ(`title linux`)は、**SUSE Linux Enterprise Desktop**をブートする役割を果たします。カーネル(`vmlinuz`)は、1台目のハードディスクの最初の論理パーティション(ブートパーティション)内に配置されています。ルートパーティションや**VGA**モードなどのカーネルパーティションは、ここに追加されます。ルートパーティションは、**Linux**の命名規則に従って指定されたものです(`/dev/sda7`)。この情報を読み込むのは**Linux**カーネルであり、**GRUB**は関係しないからです。`initrd`も、1台目のハードディスクの最初の論理パーティション内に配置されています。
- 第2のエントリは、**Windows**を読み込む役割を果たします。**Windows**は、1台目のハードディスク(`hd0,,0`)の最初のパーティションからブートされます。`chainloader +1`コマンドは、指定されたパーティションの最初のセクタを読み取って実行するよう**GRUB**に指示します。
- 次のエントリは、**BIOS**設定を変更することなく、フロッピーディスクからブートすることを可能にします。
- ブートオプション`failsafe`は、問題のあるシステム上でも**Linux**のブートを可能にするカーネルパラメータを選択して**Linux**を起動します。

メニューファイルは必要に応じて変更できます。その場合、**GRUB**は変更後の設定を次のブート時に使用します。このファイルを永続的に編集するには、**YaST**または好みのエディタを使用します。また、対話形式で一時的に変更するには、**GRUB**の編集機能を使用します。詳細については、**ブート手順実行中のメニューエントリの編集項** (97 ページ)を参照してください。

## ブート手順実行中のメニューエントリの編集

グラフィカルブートメニューでは、ブートするオペレーティングシステムを矢印キーで選択します。Linuxシステムを選択した場合は、ブートプロンプトからブートパラメータを追加入力できます。個々のメニューエントリを直接編集するには、<Esc>キーを押してスプラッシュ画面を終了し、GRUBテキストベースメニューを表示してから<E>キーを押します。この方法で加えた変更は、現在のブートだけに適用され、永続的に採用されることはありません。

---

### 重要項目: ブート手順実行中のキーボードレイアウト

ブート時は、USキーボードレイアウトだけが使用可能です。図「US Keyboard Layout」(↑システム分析およびチューニングガイド)の図を参照してください。

---

メニューエントリの編集により、障害が発生してブートできなくなったシステムを容易に修復できます。これは、ブートローダの設定ファイルの誤りをパラメータの手動入力により回避できるからです。ブート手順の中でパラメータを手動で入力する方法は、ネイティブシステムを損傷せずに新規設定をテストする際にも役立ちます。

編集モードを有効にした後、矢印キーを使用して、設定を編集するメニューエントリを選択します。設定を編集可能にするには、もう一度<E>キーを押します。このようにして、不正なパーティションまたはパス指定を、ブートプロセスに悪影響を及ぼす前に編集します。<Enter>キーを押して編集モードを終了し、メニューに戻ります。次に、<B>キーを押してこのエントリをブートします。下部のヘルプテキストに、さらに可能なアクションが表示されます。

変更後のブートオプションを永続的に入力してカーネルに渡すには、ユーザのrootでファイルmenu.lstを開き、関連カーネルパラメータをスペースで区切って既存の行に追加します。

```
title linux
    root(hd0,0)
    kernel /vmlinuz root=/dev/sda3 additional parameter
    initrd /initrd
```

GRUBは、次のシステムブート時に新規パラメータを自動的に使用します。または、この変更をYaSTのブートローダモジュールで行うこともできます。新規パラメータをスペースで区切って既存の行に追加します。

## 9.1.2 device.mapファイル

device.mapファイルは、GRUBおよびBIOSのデバイス名をLinuxのデバイス名にマップします。IDEとSCSIの各ハードディスクが混在するシステムでは、GRUBは特殊プロシージャを使用してブートシーケンスの判定を試みる必要があります。これは、GRUBはBIOSのブートシーケンス情報にアクセスできない場合があるためです。GRUBはこの分析の結果をファイル/boot/grub/device.mapに保存します。BIOS内でブートシーケンスがIDE、SCSIの順に設定されているシステムの場合、ファイルdevice.mapは次のようになります。

```
(fd0)  /dev/fd0
(hd0)  /dev/sda
(hd1)  /dev/sdb
```

IDE、SCSI、および他のハードディスクのシーケンス(順序)は、さまざまな要因によって異なり、Linuxではマッピングを識別できないため、device.mapファイル内のシーケンスは手動で設定できます。ブート時に問題に直面した場合、このファイル内のシーケンスが、BIOS内のシーケンスに対応しているかどうかチェックします。さらに、必要に応じてGRUBは、前者を一時的に変更するように指示します。Linuxシステムのブート後に、YaSTブートローダモジュールまたは好みのエディタを使用して、device.mapファイルを永続的に変更できます。

device.mapを手動で編集した後、次のコマンドを実行してGRUBを再インストールします。このコマンドにより、device.mapファイルが再読み込みされ、grub.confに指定されているコマンドが実行されます。

```
grub --batch < /etc/grub.conf
```

## 9.1.3 /etc/grub.confファイル

menu.lstおよびdevice.mapの次に重要な第3のGRUB設定ファイルは、/etc/grub.confです。このファイルには、GRUBシェルでブートローダを正常にインストールするために必要なコマンド、パラメータおよびオプションが含まれています。

```
setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)
quit
```



このコマンドは、同じパーティションに存在するブートイメージを使用して、最初のハードディスク(hd0,1)の第2パーティションにブートローダを自動的にインストールするようにGRUBに指示します。マウントされたファイルシステムからstage2イメージをインストールするには、`--stage2=/boot/grub/stage2`パラメータが必要です。一部のBIOSは、LBAサポート実装に欠陥があります。これを無視する解決策として、`--force-lba`を使用します。

## 9.1.4 ブートパスワードの設定

オペレーティングシステムのブート前でも、GRUBはファイルシステムへのアクセスを可能にします。rootパーミッションを持たないユーザは、システムのブート後、アクセス権のないLinuxシステム上のファイルにアクセスできます。この種のアクセスを阻止したり、ユーザによる特定のオペレーティングシステムのブートを防止するために、ブートパスワードを設定できます。

---

### 重要項目: ブートパスワードとスプラッシュ画面

GRUBにブートパスワードを使用する場合、通常のスプラッシュ画面は表示されません。

---

ユーザrootとして、次の手順に従ってブートパスワードを設定します。

- 1 rootプロンプトで、`grub-md5-crypt`を使ってパスワードを暗号化します。

```
# grub-md5-crypt
Password: ****
Retype password: ****
Encrypted: $1$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 2 暗号化後の文字列を、`menu.lst`ファイル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$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

これで、ブートプロンプトからGRUBコマンドを実行するには、先にPキーを押してパスワードを入力する操作が必要になります。しかし、

ユーザはブートメニューから引き続き任意のオペレーティングシステムをブートすることができます。

- 3 ブートメニューから1つまたは複数のオペレーティングシステムをブートする操作を禁止するには、menu.lst内で、パスワードを入力しなければブートできないようにする必要のある各セクションにエントリlockを追加します。たとえば、次のようにします。

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

システムをリブートしてブートメニューからLinuxエントリを選択すると、次のエラーメッセージが表示されます。

```
Error 32: Must be authenticated
```

<Enter>キーを押してメニューを表示します。次に、<P>キーを押してパスワードプロンプトを表示します。パスワードを入力して<Enter>キーを押すと、選択したオペレーティングシステム(この場合はLinux)がブートします。

## 9.2 YaSTによるブートローダの設定;

SUSE Linux Enterprise Desktopシステムでブートローダを設定する最も簡単な方法は、YaSTモジュールを使用することです。YaSTコントロールセンタで、**[システム] > [ブートローダ]**の順に選択します。図 9.1. 「ブートローダの設定」 (101 ページ)で説明しているように、システムの現在のブートローダ設定が表示され、設定を変更できます。

## 図 9.1 ブートローダの設定



[セクション管理] タブでは、各オペレーティングシステムのブートローダセクションの編集、変更、削除を行うことができます。オプションを追加するには、[追加] をクリックします。既存のオプションの値を変更するには、マウスで選択してから [編集] をクリックします。既存のエントリを削除するには、エントリを選択して [削除] をクリックします。ブートローダのオプションをよくご存知でない場合には、はじめに9.1項「GRUBによるブート」(90 ページ)を参照してください。

[ブートローダのインストール] タブで、タイプ、場所、高度なローダ設定に関する設定を表示および変更できます。

[その他] をクリックすると開くドロップダウンメニューから、高度な設定オプションにアクセスします。組み込みエディタでGRUB設定ファイルを変更できます(詳細は9.1項「GRUBによるブート」(90 ページ)を参照してください)。既存の設定を削除して新しい設定を作成したり、YaSTで新しい設定を提案できます。設定をディスクに書き込んだり、ディスクから設定を読み直すこともできます。インストール時に保存した最初のMBR(Master Boot Record) j を復元するには、[ハードディスクのMBRの復元] を選択します。

## 9.2.1 デフォルトブートエントリの調整

デフォルトでブートされるシステムを変更するには、次の手順に従います。

### 手順 9.1 標準のシステムの設定

- 1 [セクション管理] タブを開きます。
- 2 リストから目的の項目を選択します。
- 3 [デフォルトにする] をクリックします。
- 4 [完了] をクリックして、変更を有効にします。

## 9.2.2 ブートローダの場所の変更

ブートローダの場所を変更するには、次の手順に従います。

### 手順 9.2 ブートローダの場所の変更

- 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] で、次のオプションの1つを選択します。

ブートパーティションからブート

/bootパーティションのブートセクタです。

拡張パーティションからブート

拡張パーティションコンテナにブートローダがインストールされます。

マスタブートレコードからブート

最初のディスクのMBRにブートローダをインストールします(BIOS中のブートシーケンスプリセットによる)。

ルートパーティションからブート

/パーティションのブートセクタにブートローダがインストールされます。

カスタムブートパーティション

このオプションを選択すると、手動でブートローダの場所を指定できます。

- 2 [完了] をクリックして、変更を適用します。

## 9.2.3 ブートローダのタイムアウトの変更

ブートローダは、標準のシステムを直ちにブートするわけではありません。タイムアウト中、ブートまたはカーネルパラメータを書き込むシステムを選択できます。ブートローダのタイムアウトを設定するには、次の手順に従います。

**手順 9.3** ブートローダのタイムアウトの変更

- 1 [ブートローダのインストール] タブを開きます。
- 2 [ブートローダのオプション] をクリックします。
- 3 新しい値を入力するか、マウスで矢印キーをクリックするか、またはキーボードの矢印キーを使って、*[Time-Out in Seconds]* の値を変更します。
- 4 [OK] をクリックします。
- 5 [完了] をクリックして、変更を保存します。

## 9.2.4 ブートパスワードの設定

このYaSTモジュールでは、ブートを保護するためのパスワードを設定することもできます。そうすれば、セキュリティに付加的なレベルを追加できます。

**手順 9.4** ブートローダパスワードの設定

- 1 [ブートローダのインストール] タブを開きます。
- 2 [ブートローダのオプション] をクリックします。

- 3 [メニューインタフェースのパスワード] でパスワードを設定します。
- 4 [OK] をクリックします。
- 5 [完了] をクリックして、変更を保存します。

## 9.2.5 ディスク順序の変更

コンピュータに複数のハードディスクがある場合、ディスクのブートシーケンスを、コンピュータのBIOSセットアップと一致するように指定できます (「9.1.2項 「device.map ファイル」 (98 ページ)」を参照してください)。次の手順に従います。

### 手順 9.5 ディスクの順序の設定

- 1 [ブートローダのインストール] タブを開きます。
- 2 [ブートローダのインストールの詳細] をクリックします。
- 3 複数のディスクが表示されている場合には、ディスクを選択してから [上へ] または [下へ] をクリックして、ディスクの表示順を変更します。
- 4 [OK] をクリックして、変更を保存します。
- 5 [完了] をクリックして、変更を保存します。

## 9.2.6 詳細オプションの設定

詳細なブートオプションは、[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] の順に選択して、設定できます。通常は、デフォルト設定を変更する必要はありません。

[ブートパーティション用パーティションテーブルにアクティブフラグを設定]

ブートローダを含むパーティションをアクティブにします。Windows 98 のような一部のレガシーオペレーティングシステムは、アクティブパーティションからブートできます。

#### デバッグフラグ

**Sets GRUB**を、ディスクアクティビティを示すメッセージを表示するデバッグモードに設定します。

#### MBRに汎用ブートコードを書き込む

現在の**MBR**を、オペレーティングシステムに依存しない独立した汎用コードで置換します。

#### ブートメニューを隠す

ブートメニューを隠し、デフォルトエントリをブートします。

#### 信頼できるGRUBを使用する

信頼性の高いコンピューティング機能をサポートする信頼できる**GRUB**を起動します。

#### グラフィカルメニューファイル

ブート画面の表示時に使用されるグラフィックファイルへのパスを設定します。

#### シリアル接続パラメータ

コンピュータがシリアルコンソールで制御されている場合は、どの**COM**ポートをどの速度で使用するか指定できます。さらに、[ターミナル定義]を「**serial**」に設定します。詳細については、[info grub](http://www.gnu.org/software/grub/manual/grub.html)または<http://www.gnu.org/software/grub/manual/grub.html>を参照してください。

#### ターミナル定義

シリアルコンソールからブートするには、ここで「**serial**」を入力します(そうでない場合は、ブランクのままにします)。シリアルコンソールからブートする場合は、[シリアル接続パラメータ]を指定する必要もあります。

## 9.2.7 ブートローダタイプの変更

ブートローダのインストールでブートローダのタイプを設定します。**SUSE Linux Enterprise Desktop**のデフォルトブートローダは**GRUB**です。**LILO**を使用するには、以下の手順に従います。

## 手順 9.6 ブートローダのタイプの変更

- 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] で、 *[LILO]* を選択します。
- 3 表示されるダイアログボックスで、次のオプションのうち、いずれかを選択します。

新しい設定を提案する

YaSTは新しい設定を提案します。

Convert Current Configuration (現在の設定を変換する)

YaSTは現在の設定を変換します。設定を変換すると、いくつかの設定内容が失われることがあります。

Start New Configuration from Scratch (新しい設定を新規に作成する)

カスタム設定を書き込みます。この操作は、SUSE Linux Enterprise Desktopのインストール時には利用できません。

Read Configuration Saved on Disk (ディスクに保存されている設定を読み込む)

独自の/etc/lilo.confをロードします。この操作は、SUSE Linux Enterprise Desktopのインストール時には利用できません。

- 4 *[OK]* をクリックして、変更内容を保存します。
- 5 メインのダイアログで *[完了]* をクリックして、変更を適用します。

変換中に、古いGRUB設定はディスクに保存されます。これを使用するには、ブートローダのタイプをGRUBに戻し、 *[Restore Configuration Saved before Conversion]* を選択します。この操作は、インストール済みのシステムでのみ実行可能です。

---

### 注意: カスタムのブートローダ

GRUBやLILO以外のブートローダを使用する場合は、 *[ブートローダはインストールしないでください]* を選択します。このオプションを選択する場合には、あらかじめ、ブートローダのドキュメントをよくお読みください。

---



## 9.3 Linuxブートローダのアンインストール

YaSTを使用してLinuxブートローダをアンインストールし、MBRをLinuxインストール前の状態に戻すことができます。インストール中に、YaSTは自動的にオリジナルMBRのバックアップコピーを作成しており、要求があるとMBRを復元します。

GRUBをアンインストールするには、YaSTブートローダモジュールを起動します(システム> ブートローダの設定)。その他> ハードディスクのMBRの復元を選択し、はい、上書きしますで確認します。

## 9.4 ブートCDの作成

ブートマネージャを使用してシステムをブートできない場合、またはハードディスクやフロッピーディスクのMBRにブートマネージャをインストールできない場合は、Linuxに必要なすべての起動ファイルを使用してブート可能CDを作成することもできます。そのためには、システムにCDライターがインストールされている必要があります。

GRUBでは、`stage2_eltorito`という特殊形式の `stage2`とカスタマイズされた `menu.lst`(オプション)を使用するだけで、ブート可能CDROMを作成することができます。従来のファイル `stage1`および `stage2`は不要です。

### 手順 9.7 ブートCDの作成

- 1 以下のように、ISOイメージの作成先ディレクトリに移動します。`cd /tmp`
- 2 `&grub`のサブディレクトリを作成し、新たに作成された `iso`ディレクトリに移動します。  

```
mkdir -p iso/boot/grub && cd iso
```
- 3 カーネル、`stage2_eltorito`、`initrd`、`menu.lst`および `message` ファイルを `iso/boot/`にコピーします。

```
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** CD-ROMデバイスを指すように/boot/grub/menu.lstのパスエントリを調整します。そのためには、パス名に(hdx,y)形式で表示されるハードディスクのデバイス名を、**CD-ROM**ドライブのデバイス名(cd)で置き換えます。また、メッセージファイル、カーネル、およびinitrdに対するパスを調整することが必要になる場合があります。これらのパスはそれぞれ、/boot/message、/boot/vmlinuz、および/boot/initrdを参照する必要があります。調整を行った後、menu.lstは次の例のようになります。

```
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
```

ブート処理時にブートメッセージの表示を防止するには、「splash=verbose」の代わりに「splash=silent」を使用します。

- 5** 次のコマンドで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** 好みのユーティリティを使用して、生成されたファイルgrub.isoをCDに書き込みます。ISOイメージをデータファイルとして書き込まず、お使いのCD書き込みユーティリティのCDイメージ作成オプションを使用します。

## 9.5 SUSEのグラフィカル画面

オプション `vga=value` がカーネルパラメータとして使用されている場合、SUSEのグラフィカル画面が1番目のコンソール上に表示されます。YaSTを使用してインストールする場合、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基づいて自動的に使用されます。必要な場合にSUSEの画面を無効にするには、3つの方法があります。

必要に応じてSUSE画面を無効にする。

コマンドラインでコマンド `echo 0 >/proc/splash` を入力し、グラフィカル画面を無効にします。画面を再度有効にするには、`echo 1 >/proc/splash` コマンドを入力します。

デフォルトでSUSE画面を無効にする。

カーネルパラメータ `splash=0` をブートローダの設定に追加します。これについては、[第9章 ブートローダGRUB](#) (89 ページ) を参照してください。ただし、前のバージョンでデフォルトとなっていたテキストモードを選択する場合は、`vga=normal` を設定します。

SUSE画面を完全に無効にする。

新しいカーネルをコンパイルし、*framebuffer support* でオプション *Use splash screen instead of boot logo* を無効にします。

---

### ティップ

カーネルでフレームバッファのサポートを無効にすると、スプラッシュ画面も自動的に無効になります。システムをカスタムカーネルで実行した場合、SUSEはサポートを何も提供することができません。

---

## 9.6 トラブルシューティング

ここでは、GRUBを使用してブートする際に頻繁に発生する一部の問題と、考えられる解決策の概略について説明します。一部の問題については、<http://support.novell.com/> の Knowledgebase (ナレッジベース) に記事が提供されています。「GRUB」、「ブート」、および「ブートローダ」などのキーワードを使って検索を行うには、検索ダイアログを使用します。

## GRUBとXFS

XFSの場合、パーティションブートブロックにはstage1のための余地がありません。そのため、ブートローダの位置としてXFSパーティションを指定しないでください。この問題は、XFSでフォーマットされていない別のブートパーティションを作成することで解決できます。

## GRUBがGRUB Geomエラーを報告した

GRUBは、システムのブート時に、接続されているハードディスクのジオメトリを検査します。ときには、BIOSから一貫性のない情報が戻され、GRUBがGRUB Geom Errorをレポートする場合があります。そのような場合は、BIOSをアップデートしてください。

また、LinuxがBIOSに登録されていない追加ハードディスクにインストールされている場合にも、GRUBはこのエラーメッセージを戻します。ブートローダのstage1は正常に検出されロードされますが、stage2は検出されません。この問題は、新規ハードディスクをBIOSに登録することで解消できます。

## いくつかのハードディスクを搭載したシステムがブートしない

インストール中、YaSTは、ハードディスクのブートシーケンスを誤って判断する場合があります。たとえば、GRUBがIDEディスクをhd0、SCSIディスクをhd1と見なしても、BIOS内ではブートシーケンスが逆(IDEの前にSCSI)になっている場合があります。

この場合は、ブートプロセス中にGRUBコマンドラインを使用してハードディスクを訂正します。システムのブート後に、device.mapファイルを編集して新規マッピングを永続的に適用します。次に、/boot/grub/menu.lstファイルと/boot/grub/device.mapファイルでGRUBデバイス名を検査し、次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
```

## 2台目のハードディスクからのWindowsのブート

Windowsのような一部のオペレーティングシステムは、1台目のハードディスクからのみブートできます。この種のオペレーティングシステムが2台目以降のハードディスクにインストールされている場合は、関連メニューエントリに対して論理的な変更を加えることができます。

```
...
title windows
    map (hd0) (hd1)
    map (hd1) (hd0)
```

```
chainloader (hd1,0)+1  
...
```

この例では、**Windows**は2台目のハードディスクから起動されます。この目的で、`map`を使用して、ハードディスクの論理的な順序を変更します。この変更は、**GRUB**のメニューファイル内のロジックには影響を及ぼしません。したがって、2台目のハードディスクは`chainloader`に対して指定する必要があります。

## 9.7 詳細情報

**GRUB**の詳細情報は、<http://www.gnu.org/software/grub/>で入手できます。また、`grub`情報ページも参照してください。<http://www.novell.com/support>にある**Technical Information Search**(技術情報検索)で、キーワード「**GRUB**」を検索して、特別な事項に関する情報を入手することもできます。



## 特別なシステム機能

この章では、はじめに、さまざまなソフトウェアパッケージAバーチャルコンソール、およびキーボードレイアウトについて説明します。bash、cron、およびlogrotateといったソフトウェアコンポーネントについても説明します。これらは、前回のリリースサイクルで変更または強化されたからです。これらのコンポーネントはそれほど重要ではないと思われるかもしれませんが、システムと密接に結びついているものなので、デフォルトの動作を変更したい場合もあることでしょう。この章の最後では、言語および国固有設定(I18NおよびL10N)について説明します。

### 10.1 特殊ソフトウェアパッケージ

bashcronlogrotatelocateulimitfreeといったプログラムは、システム管理者および多くのユーザにとって非常に重要です。manのページとinfoのページは、コマンドについての2つの役立つ情報源ですが、その両方が常に利用できるとは限りません。GNU Emacsは、人気のある、自由度に設定できるテキストエディタです。

#### 10.1.1 bashパッケージと/etc/profile

Bashはデフォルトのシステムシェルです。ログインシェルとして使用する場合には、いくつかの初期化ファイルを読み込みます。Bashは、各ファイルを次の順序で処理します:

1. /etc/profile
2. ~/.profile
3. /etc/bash.bashrc
4. ~/.bashrc

~/.profileまたは~/.bashrcに、カスタム設定を行います。これらのファイルを正しく処理するには、基本設定ファイル/etc/skel/.profileまたは/etc/skel/.bashrcを、ユーザのホームディレクトリにコピーする必要があります。更新後、/etc/skelから設定ファイルをコピーすることをお勧めします。次のシェルコマンドを実行して、既存の個人別設定が失われるのを防止します。

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

それから、個人的な調整点を、\*.oldファイルから書き戻します。

## 10.1.2 cronパッケージ

コマンドを、前もって決めた時間に、定期的かつ自動的にバックグラウンドで実行したい場合、**cron**を用います。**cron**は特別な形式のタイムテーブルに従って起動します。その一部はシステムに付属しています。ユーザは必要に応じ、自分自身のテーブルを書くことができます。

**cron**テーブルは、/var/cron/tabsにあります。/etc/crontabはシステム全体の**cron**テーブルとして機能します。ユーザ名を入力して、タイムテーブルの後、コマンドの前に直接コマンドを実行するようにします。では、**例 10.1**「**/etc/crontab内のエントリ**」(114ページ)rootが入力されています。/etc/cron.dにあるパッケージ固有のテーブルも同じ形式です。**cron**のマニュアルページを参照してください(man cron使用)。

### 例 10.1 /etc/crontab内のエントリ

```
1-59/5 * * * * root    test -x /usr/sbin/atrun && /usr/sbin/atrun
```



/etc/crontabを、`crontab -e`コマンドで編集することはできません。これは、エディタに直接ロードして、変更し、保存する必要があります。

複数のパッケージによりシェルスクリプトが/etc/cron.hourly、/etc/cron.daily、/etc/cron.weekly、および/etc/cron.monthlyの各ディレクトリにインストールされます。これらの実行は、/usr/lib/cron/run-cronsによって制御されます。/usr/lib/cron/run-cronsは、15分おきにメインテーブル(/etc/crontab)から実行されます。これにより、無視されていたプロセスが、適切な時刻に実行されることが保証されます。

管理用のスクリプトを1時間ごと、毎日、または他の特定の周期で実行するには、/etc/crontabのエントリで、定期的に、使用するタイムスタンプファイルを削除します(「**例 10.2. 「/etc/crontab:タイムスタンプファイルの削除」** (115ページ)」を参照してください。そこでは、hourlyという名前の付いているファイルが毎時59分に、dailyという名前の付いているファイルが毎日午前 2::14に削除されるようになっていきます)。

### 例 10.2 /etc/crontab:タイムスタンプファイルの削除

```
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
```

または、/etc/sysconfig/cronのDAILY\_TIMEをcron.dailyを起動する時刻に設定します。MAX\_NOT\_RUNの設定では、ユーザが長期にわたってコンピュータを指定したDAILY\_TIMEに起動しなくても、毎日のジョブの実行がトリガされるようにします。MAX\_NOT\_RUNの最大値は14日です。

日常のシステムメンテナンスジョブは、わかりやすいようにさまざまなスクリプトに分散されています。これらはパッケージaaa\_baseに含まれています。たとえば、/etc/cron.dailyディレクトリには、コンポーネントsuse.de-backup-rpmdb、suse.de-clean-tmp、またはsuse.de-cron-localがあります。

## 10.1.3 ログファイル:パッケージlogrotate

カーネル自体に加え、定期的にシステムステータスおよび特定のイベントをログファイルに記録する多数のシステムサービス(デーモン)があります。これ

により、管理者は、ある特定時期のシステムステータスを定期的を確認し、エラーまたは問題のある機能を認識し、正確にトラブルシューティングできます。通常、これらのログファイルは、FHSで指定されるように/var/log内に格納され、毎日記録が追加されるためにサイズが増大します。logrotateパッケージを使用して、これらのファイルが増大するのを制御できます。

/etc/logrotate.confファイルを使用して、logrotateを設定します。特に、includeには、最初に読み込む追加ファイルを設定します。ログファイルを生成しないプログラムは、個別の環境設定ファイルを/etc/logrotate.dにインストールします。たとえば、そのようなファイルは、apache2 (/etc/logrotate.d/apache2)およびsyslogd (/etc/logrotate.d/syslog) パッケージに含まれています。

### 例 10.3 /etc/logrotate.confの例

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#     monthly
#     create 0664 root utmp
#     rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotateは、cronによって制御され、/etc/cron.daily/logrotateにより毎日呼び出されます。

---

## 重要項目

createオプションは、管理者によって/etc/permissions\*内に作成されるすべての設定を読み取ります。個人的な変更によっていずれの競合も発生することがないようにしてください。

---

### 10.1.4 locateコマンド

ファイルをすばやく検索するためのlocateコマンドは、標準のインストール済みソフトウェアには含まれていません。必要であれば、パッケージfind-locateをインストールしてください。updatedbプロセスは、毎晩、またはシステムをブートしてから約15分で自動的に起動します。

### 10.1.5 ulimitコマンド

ulimit (使用制限)コマンドを使用すると、システムリソースの使用量に制限を設けたり、これらの制限を表示したりすることができます。ulimitは、アプリケーションでの使用可能メモリを制限する場合に特に便利です。1つのアプリケーションが大量のメモリを独占するとシステムが停止してしまいますが、これを使用することで、それが避けられます。

ulimitコマンドには、さまざまなオプションがあります。メモリの使用量を制限するには、表10.1.「ulimit:ユーザのためのリソースの設定」(117 ページ)に示すオプションを使用します。

**表 10.1** ulimit: ユーザのためのリソースの設定

---

-m	最大常駐セットサイズ
-v	シェルが使用できる仮想メモリの最大量
-s	最大スタックサイズ
-c	作成されるコアファイルの最大サイズ
-a	すべての現在の制限値の報告

---

システム全体のエントリは、`/etc/profile`で設定できます。コアファイルの作成を有効にします。これはプログラマがデバッグを行うために必要です。通常のユーザは、`/etc/profile`ファイルでシステム管理者が指定した値を大きくすることはできませんが、`~/.bashrc`に特別なエントリを作成することは可能です。

#### 例 10.4 `ulimit:~/.bashrc`中の設定

```
# Limits maximum resident set size (physical memory):
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

メモリの量は、KB単位で指定する必要があります。詳細については、`man bash`コマンドで`man`ページを参照してください。

---

#### 重要項目

すべてのシェルが`ulimit`ディレクティブをサポートするわけではありません。ユーザが制約を包括的に設定する必要がある場合、**PAM**(たとえば、`pam_limits`)を使用すれば、包括的な調整が可能になります。

---

## 10.1.6 `free`コマンド

現在使用されている**RAM**の容量を確認することが目的ならば、`free`コマンドは、少々誤解を招くかもしれません。そのような情報は、`/proc/meminfo`で表示できます。今日では、**Linux**のような最新のオペレーティングシステムにアクセスする場合、ユーザはメモリについてそれほど深刻に考える必要はありません。利用可能な**RAM**という概念は、統一的なメモリ管理が生まれる以前の遺物です。空きメモリは悪いメモリというスローガンは、**Linux**にぴったりで、結果として、**Linux**では、空きメモリや未使用メモリを実質的に発生させず、キャッシュの量を調整するよう努力が重ねられてきました。

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ついての直接的な知識はありません。その代わりにカーネルは、ページキャッシュのアプリケーションとユーザデータを管理します。メモリが不足すると、その一部はスワップパーティションかファイルに書き込まれ、そこから**mmap**コマンドで読み込まれます(`man mmap`コマンドで`man`ページを参照)。

カーネルには、たとえば、ネットワークアクセスに使用されたキャッシュが格納されている`slab`キャッシュなどの別のキャッシュがあります。これが`/proc/meminfo`のカウンタ間の違いになります。全部ではありませんが、これらのキャッシュのほとんどは、`/proc/slabinfo`でアクセスできます。

## 10.1.7 manページとinfoページ

一部のGNUアプリケーション(`tar`など)では、`man`ページが提供されなくなりました。`man`ページが用意されていたコマンドについては、`--help`オプションを使用して簡単な概要を表示するか、詳細な手順を説明する`info`ページを使用します。`info`は、GNUのハイパーテキストシステムです。このシステムについての説明は、`info info`と入力してください。`Info` ページは、`emacs -f info`コマンドを入力してEmacsを起動するか、コンソールで直接`info`と入力します。あるいは、`tkinfo`、`xinfo`、またはヘルプシステムを使用して、`info`ページを表示します。

## 10.1.8 manコマンドを使用したマニュアルページの選択

`man man-page`を使用すると通常、マニュアルページを表示してすぐに読むことができます。同じ名前のマニュアルページが別のセクションにも存在する場合に、`man`コマンドを使用すると、どのセクションのページを表示するか尋ねるメッセージが表示されます。ユーザは答えとしてセクションを入力してください。

前の手順に戻るには、`~/.bashrc`などのシェル初期化ファイルで`MAN_POSIXLY_CORRECT=1`を設定します。

## 10.1.9 GNU Emacs用の設定

GNU Emacsは、複合作業環境です。ここでは、GNU Emacsを起動する際に処理される設定ファイルについて説明します。詳細については、<http://www.gnu.org/software/emacs/>を参照してください。

Emacsは起動時に、ユーザ、システム管理者、およびカスタマイズまたは事前設定のディストリビュータに関する設定が含まれているいくつかのファイルを読み取ります。~/emacs初期化ファイルは、/etc/skelから各ユーザのホームディレクトリにインストールされます。その後、.emacsは、/etc/skel/.gnu-emacsファイルを読み取ります。プログラムをカスタマイズするには、.gnu-emacsをホームディレクトリにコピーし(cp /etc/skel/.gnu-emacs ~/.gnu-emacsを使用)、このディレクトリで希望どおりに設定します。

.gnu-emacsは、~/gnu-emacs-customファイルをcustom-fileとして定義します。Emacsでcustomizeを使用して設定を行う場合、この設定は、~/gnu-emacs-customに保存されます。

SUSE Linux Enterprise Desktopでは、emacsパッケージはsite-start.elファイルを /usr/share/emacs/site-lispディレクトリにインストールします。site-start.elファイルは、~/emacs初期化ファイルの前にロードされます。site-start.elは、psgmlなどのEmacsアドオンパッケージと共に配布される特殊な設定ファイルが自動的にロードされるようにします。この種類の設定ファイルも /usr/share/emacs/site-lispに置かれ、ファイル名は常にsuse-start-で始まります。ローカルのシステム管理者は、default.elでシステム全体の設定を指定できます。

これらのファイルに関する詳しい説明は、*Init File*: [info:emacs/InitFile](http://info:emacs/InitFile). これらのファイルを無効にする(必要な場合)方法についても記載されています。

Emacsのコンポーネントは、いくつかのパッケージに分かれています。

- 基本パッケージのemacs。
- emacs-x11(通常インストールされている): X11をサポートしているプログラム。
- emacs-nox: X11をサポートしていないプログラム。
- emacs-info:info形式のオンラインマニュアル。
- emacs-el:Emacs Lisp内のコンパイルされていないライブラリファイル。これらは、実行時には必要ありません。

- 必要に応じて`emacs-auctex`(**LaTeX**用)、`psgml`(**SGML**および**XML**用)、`gnuserv`(クライアント/サーバ操作用)など、さまざまなアドオンパッケージをインストールできます。

## 10.2 バーチャルコンソール

**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、スタンドアロンのPCシステム上でも利用できます。テキストモードでは、6つのバーチャルコンソールが使用できます。これらの切り替えには、**Alt+F1** から **Alt+F6** を使用します。7番目のコンソールはX用に予約されており、10番目のコンソールにはカーネルメッセージが表示されます。コンソールの割り当て数は、`/etc/inittab` ファイルを修正すれば変更できます。

Xを終了せずにXからコンソールに切り替えるには、**Ctrl+Alt+F1** から **Ctrl+Alt+F6** を使用します。Xに戻るには、**Alt+F7** を押します。

## 10.3 キーボードマッピング

プログラムのキーボードマッピングを標準化するために、次のファイルに変更が行われました。

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

これらの変更は、`terminfo` エントリを使用するか、その設定ファイルが直接変更されるアプリケーション(`vi`、`less`など)にのみ影響します。.. システムに付随しないアプリケーションは、これらのデフォルト値に合わせる必要があります。

Xでは、**Compose**キー(マルチキー)は、**<Ctrl+Shift**キー(右)を使用してアクセスできます。対応するエントリも`/etc/X11/Xmodmap`に示されます。

詳しい設定は、Xキーボード拡張(XKB)を使って行うことができます。この拡張機能は、デスクトップ環境GNOME(gswitchit)およびKDE (kxkb)によっても使用されます。

---

### ティップ: 詳細情報

XKBに関する説明は、`/etc/X11/xkb/README`とそこにリストされた文書にあります。

中国語、日本語、および韓国語(CJK)に関する詳しい説明は、<http://www.suse.de/~mfabian/suse-cjk/input.html>のMike Fabianのページにあります。

---

## 10.4 言語および国固有の設定

本システムは、非常に広い範囲で国際化されており、現地の状況に合わせて柔軟に変更できます。言い換えれば、国際化(I18N)によって具体的なローカライズ(L10N)が可能になっています。I18NとL10Nという略語は、語の最初と最後の文字の間に、省略されている文字数を挟み込んだ表記です。

設定は、ファイル`/etc/sysconfig/language`の変数`LC_`で定義します。これは、単なる現地語サポートだけでなく、*Messages* (メッセージ) (言語)、*Character Set* (文字セット)、*Sort Order* (ソート順)、*Time and Date* (時刻と日付)、*Numbers* (数字)および*Money* (通貨)の各カテゴリも指します。これらのカテゴリはそれぞれ、独自の変数を使用して直接定義することも、ファイル`language`にあるマスタ変数を使用して間接的に定義することも可能です(`man locale`コマンドで`man`ページを参照)。

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`,  
`RC_LC_NUMERIC`, `RC_LC_MONETARY`

これらの変数は、プレフィクス`RC_`を付けずにシェルに渡され、前述のカテゴリを表します。関連するシェルスクリプトファイルについては後で説明します。現在の設定は、コマンド`locale`を使用して表示できます。

`RC_LC_ALL`

この変数は、すでに参照された変数の値を上書きします。



RC\_LANG

前述の変数がまったく設定されていない場合、これがフォールバックとなります。デフォルトでは、RC\_LANGだけが設定されます。これにより、ユーザが独自の変数を入力しやすくなります。

ROOT\_USES\_LANG

yesまたはno変数。noに設定すると、rootが常にPOSIX環境で動作します。

変数は、YaSTのsysconfigエディタで設定できます(8.3.1項「YaSTのsysconfigエディターを使ってシステム設定を変更する」(86 ページ)を参照してください)。このような変数の値には、言語コード、国コード、エンコーディング、および修飾子が入っています。個々のコンポーネントは特殊文字で接続されます。

```
LANG=<language>[_<COUNTRY>].<Encoding>[@<Modifier>]
```

## 10.4.1 例

言語コードと国コードは必ず一緒に設定する必要があります。言語の設定は、<http://www.evertype.com/standards/iso639/iso639-en.html>および<http://www.loc.gov/standards/iso639-2/>で入手できる、ISO 639規格に従います。国コードは、[http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en\\_listp1.html](http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html)で入手できる、ISO 3166にリストされています。

使用可能な説明ファイルが/usr/lib/localeに存在する場合のみ、値を設定する意味があります。追加の記述ファイルは、/usr/share/i18n のファイルを使用し、コマンド localedef を実行して作成できます。記述ファイルは、glibc-i18ndataパッケージに含まれています。en\_US.UTF-8の説明ファイル(英語および米国)は以下のように作成します。

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

LANG=en\_US.UTF-8

インストール時にAmerican Englishを選択すると、これがデフォルトの設定になります。他の言語を選択した場合、その言語が有効になりますが、文字コードはUTF-8が使用されます。

```
LANG=en_US.ISO-8859-1
```

これにより、言語が英語、国が米国、文字セットがISO-8859-1に設定されます。この文字セットは、ユーロ記号をサポートしませんが、UTF-8がサポートされていない、更新前のプログラムを使用する方が便利なこともあります。文字セット(この状況ではISO-8859-1)を定義する文字列は、Emacsのようなプログラムによって評価されます。

```
LANG=en_IE@euro
```

上記の例では、ユーロ記号が言語設定に明示的に組み込まれています。厳密に言うと、この設定は今では古くなっています。UTF-8もユーロ記号を扱うからです。この設定が役立つのは、アプリケーションがUTF-8ではなく、ISO-8859-15しかサポートしない場合だけです。

SuSEconfigは、/etc/sysconfig/languageにある変数を読み込み、必要な変更を/etc/SuSEconfig/profileと/etc/SuSEconfig/csh.cshrcに書き込みます。/etc/SuSEconfig/profileは/etc/profileによって読み込まれます。つまり、ソースとして使用されます。/etc/SuSEconfig/csh.cshrcは/etc/csh.cshrcのソースとして使用されます。これにより、設定はシステム全体に渡って使用できるようになります。

ユーザは、同様に ~/.bashrc ファイルを編集して、システムのデフォルトを上書きすることができます。たとえば、システム設定の en\_US をプログラムメッセージに使用しない場合は、LC\_MESSAGES=es\_ES を指定してメッセージが英語の代わりにスペイン語で表示されるようにします。

## 10.4.2 ~/.i18nでのロケール設定

ロケールシステムのデフォルトが不十分な場合、Bash スクリプトの構文に従って ~/.i18n の設定を変更してください。 ~/.i18n 内のエントリは、/etc/sysconfig/language のシステムデフォルトを上書きします。同じ変数名を RC\_ネームスペースプレフィクスなしで使用します。たとえば、RC\_LANG ではなく、LANG を使用します。

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

## 10.4.3 言語サポートの設定

カテゴリ *Messages* のファイルは、フォールバックを確保するため、対応する言語ディレクトリ(たとえば、en)にのみ格納されることになっています。たとえばLANGをen\_USに設定したが、**message**ファイルが/usr/share/locale/en\_US/LC\_MESSAGESに存在しない場合は、/usr/share/locale/en/LC\_MESSAGESにフォールバックされます。

フォールバックチェーンも定義できます。たとえば、ブルターニュ語、次いでフランス語、またはガリシア語、次いでスペイン語、次いでポルトガル語の順にフォールバックするには、次のように設定します。

```
LANGUAGE="br_FR:fr_FR"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

必要に応じて、次のようにノルウェー語の方言であるニーノシクやブークモールをノルウェー語の代わりに使用できます(noへのフォールバックを追加します)。

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

または

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

ノルウェー語では、LC\_TIMEの扱いも違うので注意してください。

生じる可能性のある1つの問題は、数字の桁を区切るための文字が正しく認識されないことです。このことは、LANGがdeのような2文字の言語コードにのみ設定されているのに、**glibc**が使用している定義ファイル/usr/share/lib/de\_DE/LC\_NUMERICに存在している場合に生じます。それで、区切り文字の定義がシステムに認識されるようにするには、LC\_NUMERICをde\_DEに設定する必要があります。

## 10.4.4 詳細情報

- 『*The GNU C Library Reference Manual*』の「Locales and Internationalization」の章。glibc-infoパッケージに格納されています。
- 『UTF-8 and Unicode FAQ for Unix/Linux』、Markus Kuhn 著。Web ページ <http://www.cl.cam.ac.uk/~mgk25/unicode.html> (現在のアドレス)を参照してください。
- *Unicode-Howto*(Bruno Haible著):/usr/share/doc/howto/en/html/Unicode-HOWTO.html。

## プリンタの運用

SUSE® Linux Enterprise Desktopは、リモートネットワークプリンタも含め、さまざまな種類のプリンタを使った印刷をサポートしています。プリンタはYaSTを使って、または手動で設定することができます。設定の詳細については、項「Setting Up a Printer」(第6章 *Setting Up Hardware Components with YaST*, ↑導入ガイド)を参照してください。プリントジョブの開始、管理には、グラフィカルインタフェースまたはコマンドラインユーティリティの両方を利用できます。プリンタが正常に動作しない場合は、**11.8項「トラブルシューティング」**(138 ページ)を参照してください。

CUPSは、SUSE Linux Enterprise Desktopでの標準的な印刷システムです。CUPSは、特にユーザ中心の構造(ユーザ志向の設計)です。多くの状況ではLPRngとの互換性があるか、比較的少ない作業で適応させることができます。LPRngは、互換性を維持する理由でのみ、SUSE Linux Enterprise Desktopに付属しています。

プリンタは、インタフェース(USB、ネットワークなど)と、プリンタ言語によって区別できます。プリンタの購入時には、プリンタにご利用のハードウェアで利用できるインタフェース(USBやパラレルポートなど)が搭載されていること、およびプリンタの対応言語が正しいことをご確認ください。プリンタは、次の3つのプリンタ言語クラスに基づいて分類できます。

### PostScriptプリンタ

PostScriptは、LinuxとUnix環境のほとんどの印刷ジョブを生成する際に使用されるプリンタ言語であり、内部の印刷システムもこの言語を使用して処理を行います。この言語はかなり古いですが、ひじょうに効率的です。使用中のプリンタがPostScriptドキュメントを直接処理でき、印刷システム側で追加のステージを使用して変換を行う必要がない場合、潜在的なエ

ラーの原因の数が減少します。PostScriptプリンタは多額のライセンスコストの対象になるので、通常、これらのプリンタは、PostScriptインタプリタを内蔵しないプリンタよりコストが高くなります。

#### 標準的なプリンタ(PCLおよびESC/Pなどの言語)

これらのプリンタ言語はかなり古いのですが、プリンタで新機能を実現するために、引き続き拡張が行われています。既知のプリンタ言語の場合、印刷システムはGhostscriptの支援により、PostScriptのジョブを該当のプリンタ言語へ変換できます。この処理ステージを「解釈」(interpreting)と呼びます。非常によく知られている言語は、ほとんどのHPのプリンタおよび互換モデルが採用しているPCLと、Epsonのプリンタが採用しているESC/Pです。これらのプリンタ言語は、通常はLinuxによってサポートされていて、まずまずの印刷結果をもたらします。最新のプリンタや特殊なプリンタの機能は、Linuxがサポートしていないことがあります。オープンソースの開発者は、それらの機能に関してまだ作業をしている可能性もあります。HPが開発したHPLIPを除き、現時点では、Linuxドライバを開発してオープンソース条項に基づきそれらをLinuxのディストリビュータに提供しているプリンタメーカは存在しません。これらのプリンタのほとんどは、中間の価格帯にあります。

#### 独自規格のプリンタ(GDIプリンタ)

これらのプリンタは、共通のプリンタ言語をサポートしていません。これらのプリンタは独自のプリンタ言語を使用しており、新しいエディション/モデルがリリースされると、プリンタ言語も変更される可能性があります。一般的にこのようなプリンタでは、Windowsドライバしか利用できません。詳細については、**11.8.1項「標準的なプリンタ言語をサポートしないプリンタ」** (138 ページ)を参照してください。

新しいプリンタを購入する前に、次の各ソース(情報源)を参照し、購入を予定しているプリンタがどの程度までサポートされているかを確認してください。

<http://www.linuxfoundation.org/en/OpenPrinting/>  
OpenPrinting.orgのプリンタデータベース。

<http://www.cs.wisc.edu/~ghost/>  
GhostscriptのWebページ。

/usr/share/doc/packages/ghostscript-library/catalog.devices  
付属するドライバのリスト

オンラインデータベースはいつでも、Linuxによるサポートの最新のステータスを示しています。しかし、Linuxのディストリビューションが統合できるのは、製造の時点で使用可能だったドライバだけです。したがって、現時点で「完全にサポート済み」と評価されているプリンタであっても、最新バージョンのSUSE Linux Enterprise Desktopがリリースされた時点では、そのステータスに達していなかった可能性があります。そのため、これらのデータベースは必ずしも正しいステータスを表しているとは限らず、おおよその状況を提示するだけにとどまっています。

## 11.1 印刷システムのワークフロー

ユーザが印刷ジョブを作成します。印刷ジョブは、印刷するデータとスプーラに対する情報から構成されますが、その情報には、プリンタの名前またはプリンタキューの名前だけでなく、必要に応じて、プリンタ固有のオプションなど、フィルタに関する情報も含まれます。

各プリンタには、1つ以上の専用プリンタキューが存在しています。指定のプリンタがデータを受け取れるようになるまで、スプーラは印刷ジョブをキュー内に留めています。プリンタの準備が整うと、スプーラはフィルタおよびバックエンドを経由して、プリンタにデータを送信します。

このフィルタは、印刷中のアプリケーションが生成したデータ(通常的にはPostScriptやPDFですが、ASCII、JPEGなどの場合もあります)を、プリンタ固有のデータ(PostScript、PCL、ESC/Pなど)に変換します。プリンタの機能については、PPDファイルに記述されています。PPDファイルには、プリンタ固有のオプションが記述されています。各オプションに対しては、プリンタでそのオプションを有効にするために必要なパラメータが指定されています。フィルタシステムは、ユーザが有効として選択したオプションを確認します。

PostScriptプリンタを選択すると、フィルタシステムがデータをプリンタ固有のPostScriptに変換します。この変換にプリンタドライバは必要ありません。PostScript非対応プリンタを使用すると、フィルタシステムがデータをプリンタ固有データに変換します。この変換には、使用しているプリンタに適応したプリンタドライバが必要です。バックエンドは、プリンタ固有データをフィルタから受信し、そのデータをプリンタに送信します。

## 11.2 プリンタに接続するための方法とプロトコル

プリンタをシステムに接続するには、さまざまな方法があります。CUPS印刷システムの設定は、ローカルプリンタと、ネットワーク経由でシステムに接続されているプリンタを区別しません。Linux環境では、ローカルプリンタは、プリンタメーカーのマニュアルに記載されているとおりに接続する必要があります。CUPSは、シリアル、USB、パラレル、およびSCSI接続をサポートしています。

---

### 警告: 稼働中システムのケーブル接続の変更

プリンタをコンピュータに接続する場合、コンピュータの動作中に接続と取り外しを行って良いのはUSBデバイスだけであることに注意してください。システムやプリンタの損傷を回避するために、USB以外の接続を変更する場合は、あらかじめシステムをシャットダウンしてください。

---

## 11.3 ソフトウェアのインストール

PPD (PostScript printer description、PostScriptプリンタ記述)は、PostScriptプリンタの特性(解像度など)やオプション(両面印刷ユニットなど)を記述するコンピュータ言語です。これらの記述は、CUPS側でさまざまなプリンタオプションを使用するために必須です。PPDファイルがない場合、印刷データは「raw」(ロー、未加工)状態でプリンタへ送信されますが、そのことは通常は望ましくありません。SUSE Linux Enterprise Desktopのインストール時に、多数のPPDファイルがプレインストールされます。

PostScriptプリンタを設定する場合、最善のアプローチは、適切なPPDファイルを入手することです。この種の多数のPPDファイルは、標準インストールの範囲内で自動的にインストールされるパッケージmanufacturer-PPDsに用意されています。および[11.8.2項「特定のPostScriptプリンタに適したPPDファイルが入手できない」](#) (139 ページ)を参照してください。[11.7.2項「各種パッケージ内のPPDファイル」](#) (136 ページ)

新しいPPDファイルは、`/usr/share/cups/model/`ディレクトリ内に保存するか、YaSTで印刷システムに追加できます(Adding Drivers with YaST項(第



6章 *Setting Up Hardware Components with YaST*, ↑導入ガイド)を参照)。その結果、インストールの際にPPDファイルを選択できるようになります。

ユーザが設定ファイルを変更するのみでなくソフトウェアパッケージ全体をインストールすることを、プリンタメーカーが望んでいるかどうかに注意してください。第一に、このようなタイプのインストールを行うと、SUSE Linux Enterprise Desktopによって提供されているサポートが失われる結果になります。第二に、印刷コマンドが異なる方法で機能する可能性があり、システムは他のメーカーのデバイスに対応できなくなる可能性もあります。この理由で、メーカーのソフトウェアをインストールすることをお勧めしません。

## 11.4 ネットワークプリンタ

ネットワークプリンタは、さまざまなプロトコルをサポートできますし、その複数を同時にサポートすることも可能です。サポートされているプロトコルのほとんどは標準化されたものですが、いくつかのメーカーはその標準に拡張(変更)を加えました。それらのメーカーは標準を正しく実装していないシステムのテストや、標準では使用できない特定の機能を提供することを望んでいます。そのような場合、メーカーは少数のオペレーティングシステム用ののみドライバを提供し、自社のシステムにつきまとう課題を排除します。残念なことに、Linuxドライバはめったに提供されません。現在の状況では、あらゆるプロトコルがLinux環境で円滑に動作するという仮定に基づいて行動することはできません。したがって、機能する設定を実現するために、さまざまなオプションを実験する必要があります。

CUPSはsocket、LPD、IPP、およびsmbの各プロトコルをサポートしています。

### socket

*socketl*は、データのハンドシェイクを最初に行うことなく、データをインターネットソケットへ送信する接続を意味します。一般的に使用されるsocketのポート番号のいくつかは、9100または35です。デバイスURI (uniform resource identifier)の構文は、`socket://プリンタのIP:ポート`です。たとえば、`socket://192.168.2.202:9100/`のようになります。

### LPD (line printer daemon、ラインプリンタデーモン)

実証されてきたLPDプロトコルは、RFC1179で説明されています。このプロトコルを使用する場合、プリンタキューのIDのようなジョブ関連データ

の一部は、実際の印刷データより先に送信されます。したがって、データを送信するために、**LPD**プロトコルを設定する際にプリンタキューを指定する必要があります。さまざまなプリンタメーカーによる実装は、プリンタキューとして任意の名前を受け入れる柔軟性を備えています。必要に応じて、使用可能な名前がプリンタのマニュアルに提示されています。多くの場合、**LPT**、**LPT1**、**LP1**、または他の類似した名前が使用されています。**CUPS**システムを採用している他の**Linux**ホストまたは**Unix**ホスト上で、**LPD**キューを設定することもできます。**LPD**サービスが使用するポート番号は515です。デバイスURIの例は、`lpd://192.168.2.202/LPT1`です。

#### **IPP (Internet Printing Protocol、インターネット印刷プロトコル)**

**IPP**は比較的新しい(1999年)プロトコルであり、**HTTP**プロトコルに基づいています。**IPP**を使用する場合、他のプロトコルより、ジョブとの関連性が高いデータが送信されます。**CUPS**は、**IPP**を使用して内部のデータ送信を行います。これは、2台の**CUPS**サーバ間でキューを転送する上で優先されるプロトコルです。**IPP**を正しく設定するには、印刷キューの名前は必須です。**IPP**のポート番号は631です。デバイスURIの例は、

`ipp://192.168.2.202/ps`および  
`ipp://192.168.2.202/printers/ps`です。

#### **SMB (Windows共有)**

**CUPS**は、**Windows**共有に接続されたプリンタへの印刷もサポートしています。この目的で使用されるプロトコルは、**SMB**です。**SMB**は、ポート番号137、138、および139を使用します。デバイスURIの例は、

`smb://user:password@workgroup/smb.example.com/printer`、  
`smb://user:password@smb.example.com/printer`、および  
`smb://smb.example.com/printer`です。

設定を行う前に、プリンタがサポートしているプロトコルを決定する必要があります。必要な情報をメーカーが提供していない場合、`nmap`コマンド(`nmap`パッケージ)を使用して、プロトコルを推定します。`nmap`は、ホストでオープンしているポートを確認します。たとえば、次のようにします。

```
nmap -p 35,137-139,515,631,9100-10000 printerIP
```

## 11.4.1 コマンドラインツールによるCUPS設定

YaSTでのCUPSオプションとは別に、ネットワークプリンタの設定時に `lpadmin` や `lpoptions` などのコマンドラインツールを使ってCUPSを設定することができます。バックエンド(パラレルなど)とパラメータで構成されるデバイスURIが必要です。システム上の有効なデバイスURIを決定するには、コマンド `lpinfo -v | grep ":/"` を使用します。

```
# lpinfo -v | grep ":/"  
direct usb://ACME/FunPrinter%20XL  
direct parallel:/dev/lp0
```

`lpadmin` で、CUPSサーバ管理者の追加、削除、またはクラスおよび印刷キューの管理ができます。プリントキューを追加するには、次の構文を使用します。

```
lpadmin -p queue -v device-URI -P PPD-file -E
```

このデバイス(-v)は、指定したPPDファイル(-P)を使用して、queue(-p)として使用できます。プリンタを手動で設定する場合は、このPPDファイルとデバイスのURIを把握しておく必要があります。

-Eは、最初のオプションとして使用しないでください。どのCUPSコマンドでも、-Eを最初の引数として使用した場合、暗号化接続を使用することを暗示的に意味します。プリンタを使用可能にするには、次の例に示す方法で-Eを使用する必要があります。

```
lpadmin -p ps -v parallel:/dev/lp0 -P \  
/usr/share/cups/model/Postscript.ppd.gz -E
```

ネットワークプリンタの設定例:

```
lpadmin -p ps -v socket://192.168.2.202:9100/ -P \  
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

`lpadmin` のオプションの詳細は、`lpadmin(1)` のマニュアルページを参照してください。

プリンタのセットアップ時には、一部のオプションがデフォルトとして設定されています。これらのオプションは、各印刷ジョブ用に変更できます(使用される印刷ツールに依存)。YaSTを使用して、これらのデフォルト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を使用して、デフォルトオプションを次のように設定します。

- 1 最初に、すべてのオプションを列挙します。

```
lpoptions -p queue -l
```

例:

```
Resolution/Output Resolution: 150dpi *300dpi 600dpi
```

アクティブになったデフォルトオプションは、先頭にアスタリスク(\*)が付いています。

- 2 次のようにlpadminを使用してオプションを変更します。

```
lpadmin -p queue -o Resolution=600dpi
```

- 3 新しい設定値の確認:

```
lpoptions -p queue -l
```

```
Resolution/Output Resolution: 150dpi 300dpi *600dpi
```

標準ユーザがlpoptionsを実行すると、設定が~/.cups/lpoptionsに書き込まれます。ただし、root設定は/etc/cups/lpoptionsに書き込まれます。

## 11.5 グラフィカルな印刷インタフェース

xppやKDEプログラムKprinterなどのツールは、キューを選択したり、CUPS標準オプションとPPDファイルを介して使用可能になるプリンタ固有オプションの両方を設定するための、グラフィカルなインタフェースを提供します。Kprinterを、KDE以外のアプリケーションの標準印刷インタフェースとして使用することもできます。これらのアプリケーションの印刷ダイアログで、プリンタコマンドとしてkprinterまたはkprinter --stdinを指定してください。使用するコマンドは、アプリケーションのデータ転送方法によって異なります。両方のコマンドを試して、機能する方を使用してください。設定が適切であれば、アプリケーションから印刷ジョブが発行されると、アプリケーションはKprinterのダイアログを表示します。このダイアログを使用してキューを選択し、他の印刷オプションを設定できます。この場合、アプリケーション自身の印刷設定がkprinterの印刷設定と競合が発生せず、Kprinterが使用

可能になった後で、印刷オプションの変更がKprinterによってのみ行われる必要があります。KPrinterに関する詳細については、第6章 *Managing Print Jobs* (↑*KDE User Guide*)を参照してください。

## 11.6 コマンドラインからの印刷

コマンドラインから印刷するには、コマンド `lp -d queuefilename` を入力し、`queuefilename`および`filename`を対応する名前置き換えます。

一部のアプリケーションでは、印刷処理を`lp`コマンドに依存しています。この場合、アプリケーションの印刷ダイアログで正しいコマンドを入力します。ただし、通常は`filename`を指定しません。たとえば、`lp -d queuefilename`と入力します。

## 11.7 SUSE Linux Enterprise Desktopでの特殊機能

CUPSの多くの機能は、SUSE Linux Enterprise Desktopでできるように調整されています。ここでは、最も重要な変更点について説明します。

### 11.7.1 CUPSとファイアウォール

デフォルトのSUSE Linux Enterprise Desktopインストールを実行した後、`SuSEfirewall2`はアクティブになり、ネットワークインタフェースは着信トラフィックをブロックする外部ゾーンに設定されます。これらのデフォルト設定は、CUPSを使用するときに調整します。`SuSEfirewall2`設定の詳細については、項「`SuSEfirewall2`」(第9章 *Masquerading and Firewalls*, ↑セキュリティガイド)を参照してください。

### 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内部の信頼されるネットワーク環境の通常のワークステーションで実行されます。この場合、ネットワーク

インタフェースを内部ゾーンに設定し、ワークステーションにネットワーク内部から到達できるようにすることを推奨します。

## CUPSサーバ

CUPSサーバがファイアウォールで保護された信頼済みネットワーク環境の一部の場合、ネットワークインタフェースはファイアウォールの内部ゾーンに設定します。CUPS設定で特別なファイアウォールルールおよびセキュア設定により保護する場合を除いて、信頼できないネットワーク環境でCUPSサーバを設定することはお勧めできません。

### 11.7.2 各種パッケージ内のPPDファイル

YaSTのプリンタ設定機能は、`/usr/share/cups/model/`内に記述されたPPDファイルのみを使用して、CUPS用のキューをセットアップします。プリンタモデルに適したPPDファイルを決定するために、YaSTはハードウェア検出の際に判断されたベンダおよびモデルを、システムの`/usr/share/cups/model/`内で使用可能なすべてのPPDファイル内にあるベンダおよびモデルと比較します。この目的で、YaSTのプリンタ設定機能は、PPDファイルから抽出したベンダおよびモデルの情報に基づいて、データベースを生成します。プリンタを選択した場合、そのベンダおよびモデルに対応するPPDファイルを受け取ることになります。

PPDファイルのみを使用し、他の情報ソースを使用しない設定には、`/usr/share/cups/model/`内のPPDファイルを自由に変更できるという利点があります。YaSTのプリンタ設定機能は、変更結果を認識し、ベンダおよびモデルからなるデータベースを再生成します。たとえば、PostScriptプリンタのみを使用している場合、通常は`cups-drivers`パッケージ内にあるFoomatic PPDファイルや、`gutenprint`パッケージ内にあるGutenprint PPDファイルを必要としません。代わりに、使用中のPostScriptプリンタ用のPPDファイルを`/usr/share/cups/model/`へ直接コピーし(それらがまだ`manufacturer-PPDs`パッケージ内に存在していない場合)、使用中のプリンタに合わせて最適な設定を行うこともできます。

## cupsパッケージ内のCUPS PPDファイル

cupsパッケージ内にある基本PPDファイルは、PostScript Level 1およびLevel 2プリンタに適応したFoomatic PPDファイルによって補足されます。

- /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プリンタフィルタのfoomatic-ripは、PostScript非対応プリンタ用のGhostscriptと組み合わせて使用されます。適切なFoomatic PPDファイルには、`*NickName: ... Foomatic/Ghostscript driver`および`*cupsFilter: ... foomatic-rip`のエントリがあります。これらのPPDファイルは、cups-driversパッケージ内にあります。

YaSTでは一般に、manufacturer-PPDファイルが優先されます。ただし、適切なmanufacturer-PPDファイルが存在しない場合は、エントリ`*NickName: ... Foomatic ... (recommended)`のFoomatic PPDファイルが選択されます。

## gutenprintパッケージのGutenprint PPDファイル

多くのPostScript非対応プリンタでは、foomatic-ripの代わりに、Gutenprint(以前のGIMP-Print)から取得したCUPSフィルタrastertogutenprintを使用できます。このフィルタと、適切なGutenprint PPDファイルは、gutenprintパッケージ内に用意されています。Gutenprint PPDファイルは/usr/share/cups/model/gutenprint/内に配置されていて、そのファイル内にエントリ`*NickName: ... CUPS+Gutenprint`および`*cupsFilter: ... rastertogutenprint`があります。

## manufacturer-PPDsパッケージ内にあるプリンタメーカーからのPPDファイル

manufacturer-PPDsパッケージには、十分自由なライセンスに基づいてプリンタメーカーから提供されたPPDファイルが含まれています。PostScriptプリンタは、プリンタメーカーの適切なPPDファイルを使用して設定するのが妥当です。このファイルを使用すると、そのPostScriptプリンタの機能すべてを活用できるためからです。manufacturer-PPDsパッケージから得られたPPDファイルが優先されます。モデル名が一致しない場合、YaSTではmanufacturer-PPDパッケージからPPDファイルを使用できません。これは、Funprinter 12xxシリーズなど、類似モデルについて1つのPPDファイルのみがmanufacturer-PPDパッケージに含まれる場合に該当します。この場合は、YaSTで対応するPPDファイルを手動で選択します。

## 11.8 トラブルシューティング

ここでは、プリンタハードウェアおよびソフトウェアに最も一般的に発生する問題と、それを解決または回避する方法について説明します。GDIプリンタ、PPDファイル、およびポート設定などのトピックをカバーしています。一般的なネットワークプリンタに関する問題、印刷に問題がある場合、およびキュー処理についても対処しています。

### 11.8.1 標準的なプリンタ言語をサポートしないプリンタ

これらのプリンタは、共通のプリンタ言語をサポートしておらず、独自のコントロールシーケンスを使用しないと対処できません。そのため、これらのプリンタは、メーカーがドライバを添付した特定のバージョンの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用に開発したプログラミングインタフェースです。通常、メーカーはWindows用のドライバだけを提供しています。また、WindowsドライバはGDIインタフェースを使用しているため、これらのプリンタは「*GDI*プリンタ」と呼ばれることもあります。実質的な問題は、このプログラミングインタフェースではなく、これらのプリンタを制御できるのは、各プリンタモデルが採用している独自のプリンタ言語のみという事実にあります。



いくつかのGDIプリンタは、GDIモードと標準的なプリンタ言語のいずれかの間で切り替えることができます。マニュアルがある場合は、プリンタのマニュアルを参照してください。モデルによっては、切り替えを行うために特別なWindowsソフトウェアが必要なこともあります(Windowsから印刷する場合、Windowsプリンタドライバは常にプリンタをGDIモードに切り替える場合があることに注意してください)。他のGDIプリンタでは、標準のプリンタ言語を利用するための拡張モジュールが用意されています。

一部のメーカーは、プリンタに独自規格のドライバを提供しています。独自規格のプリンタドライバの欠点は、インストール済みの印刷システムとそのドライバを組み合わせたときに動作するという保証も、さまざまなハードウェアプラットフォームに適しているという保証もないことです。一方、標準的なプリンタ言語をサポートするプリンタは、特殊なバージョンの印刷システムや特殊なハードウェアプラットフォームに依存しません。

独自規格に対応するLinuxドライバを正常に機能させるために時間を費やすより、サポートされているプリンタを購入する方がコスト効率が良いこともあります。この方法により、ドライバの問題を一度だけで、そしてあらゆる状況で解決できます。特殊なドライバソフトウェアのインストールと設定を行う必要はなく、新しい印刷システムの開発に伴ってドライバのアップデートを入手する必要もありません。

## 11.8.2 特定のPostScriptプリンタに適したPPDファイルが入手できない

manufacturer-PPDsパッケージの中に、特定のPostScriptプリンタに適したPPDファイルが含まれていない場合、プリンタメーカー製のドライバCDに収録されているPPDファイルを使用すること、またはプリンタメーカーのWebページから適切なPPDファイルをダウンロードすることができるはずです。

PPDファイルがzipアーカイブ(.zip)または自己展開zipアーカイブ(.exe)の形で提供されている場合、unzipを使用してそのファイルを展開します。最初に、PPDファイルのライセンス(許諾契約)条項を読みます。次にcupstestppdユーティリティを使って、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ます。

「FAIL」ユーティリティから失敗が返された場合は、PPDファイル中のエラーは深刻なもので、問題を引き起こす可能性があります。cupstestppdによって報告された問題点は、取り除く必要があります。必要に応じて、適切なPPD

ファイルが入手できるかどうかをプリンタメーカーに問い合わせることも考えられます。

## 11.8.3 パラレルポート

最も安全なアプローチは、プリンタを最初のパラレルポートに直接接続し、BIOS内で次のパラレルポート設定値を選択することです。

- I/Oアドレス:378 (16進)
- 割り込み:無関係
- モード:Normal (通常)、SPP、またはOutput Only (出力専用)
- DMA:無効

これらの設定値を使用した場合でも、パラレルポートに接続したプリンタを使用できない場合、BIOS内での設定値に合わせて、I/Oアドレスを0x378という形で/etc/modprobe.conf内に明示的に入力します。2つのパラレルポートが存在し、それぞれのI/Oアドレスが378と278 (16進)に設定されている場合、それらを0x378, 0x278という形で入力します。

割り込み(IRQ) 7が空いている場合、例 11.1. 「**/etc/modprobe.conf:最初のパラレルポートの割り込みモード**」 (140 ページ)に示すエントリを使用して、その割り込みを有効にすることもできます。割り込みモードを有効にする前に、/proc/interruptsファイルを参照して、すでに使用中の割り込みを調べます。現時点で使用中の割り込みだけが表示されます。どのハードウェアコンポーネントがアクティブになっているかに応じて、この表示は変化することがあります。パラレルポート用の割り込みは、他のどのデバイスも使用してはなりません。自信がない場合、irq=noneを指定してポーリングモードを使用します。

**例 11.1** /etc/modprobe.conf:最初のパラレルポートの割り込みモード

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

## 11.8.4 ネットワークプリンタ接続

### ネットワークの問題の識別

プリンタをコンピュータに直接接続します。テストの目的で、そのプリンタをローカルプリンタとして設定します。この方法で動作する場合、問題はネットワークに関連しています。

### TCP/IPネットワークの確認

TCP/IPネットワークと名前解決が正しく機能していることが必要です。

### リモートlpdの確認

次のコマンドを使用して、*host*上のlpd(ポート515)に対するTCP接続を確立できるかどうかをテストします。

```
netcat -z host 515 && echo ok || echo failed
```

lpdへの接続を確立できない場合、lpdがアクティブになっていないか、ネットワークの基本的な問題があります。

rootユーザで次のコマンドを使用し、リモート*host*上の*queue*に関するステータスレポート(おそらく、非常に長い)を照会することもできます。これは、該当のlpdがアクティブで、そのホストが照会を受け付けることを前提にしています。

```
echo -e "\004queue" \  
| netcat -w 2 -p 722 host 515
```

lpdが応答しない場合、それがアクティブになっていないか、ネットワークの基本的な問題が発生している可能性があります。lpdが応答する場合、その応答は、*host*上にある*queue*を介して印刷ができない理由を示すはずです。例 11.2. 「lpdからのエラーメッセージ」(141 ページ)でこうした応答を受け取った場合、問題はリモートのlpdにあります。

### 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
```

## リモート cupsd の確認

デフォルトでは、CUPS ネットワークサーバはUDPポート631を使用して、自らのキューを30秒ごとにブロードキャストします。したがって、次のコマンドを使用して、ネットワーク内にCUPS ネットワークサーバが存在しているかどうかをテストすることができます。コマンドを実行する前に、ローカルCUPSデーモンが終了していることを確認します。

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

ブロードキャストを行っているCUPS ネットワークサーバが存在している場合、出力は例 11.3. 「CUPS ネットワークサーバからのブロードキャスト」 (142 ページ) に示すようになります。

### 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
```

次のコマンドを使用して、*host* 上の cupsd (ポート631) に対するTCP接続を確立できるかどうかをテストすることができます。

```
netcat -z host 631 && echo ok || echo failed
```

cupsd への接続を確立できない場合は、cupsd が有効になっていないか、基本的なネットワークの問題が発生している可能性があります。lpstat -h *host* -l -t は、*host* 上のすべてのキューに関するステータスレポート (非常に長い場合がある) を返しますが、それぞれの cupsd が有効になっていて、ホストがクエリを受け入れることが前提になります。

次のコマンドを使用して、*host* 上の *queue* が、1つのキャリッジリターン (CR、改行) 文字からなる印刷ジョブを受け付けるかどうかをテストできます。何も印刷されないのが妥当です。おそらく、空白のページが排出されるはずです。

```
echo -en "\r" \  
| lp -d queue -h host
```

ネットワークプリンタまたは印刷サーバボックスのトラブルシューティング  
印刷サーバボックス上のスプーラは時々、大量の印刷ジョブを処理する必要が生じた場合、問題を引き起こすことがあります。これは印刷サーバボックス内のスプーラに起因しているので、ほとんどの場合、管理者が実行できる対策はありません。回避策として、印刷サーバボックス内のスプーラを使用することを避け、TCPソケットを使用して、印刷サーバボッ

クスに接続されているプリンタに直接送信できます。詳細については、[11.4項「ネットワークプリンタ」](#) (131 ページ)を参照してください。

この方法により、印刷サーバボックスは異なる形式のデータ転送(TCP/IP ネットワークとローカルプリンタ接続)間の単純なコンバータになります。この方法を使用するには、印刷サーバボックス内にある、該当するTCP ポートについて把握する必要があります。プリンタが印刷サーバボックスに接続されていて、電源がオンになっている場合、印刷サーバボックスの電源をオンにした後、しばらく経過した時点で、nmapパッケージのnmapユーティリティを使用することにより、このTCPポートを特定できます。たとえば、nmap *IP-address*は、印刷サーバボックスに関して次のような出力をすることがあります。

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

この出力は、印刷サーバボックスに接続されているプリンタが、ポート9100上のTCPソケットを介して使用できることを示します。nmapはデフォルトでは、`/usr/share/nmap/nmap-services`内でリストされている多数の一般的な既知のポートだけを確認します。可能性のあるすべてのポートをチェックするには、nmap

`-p from_port-to_portIP-address`コマンドを使用します。これは、ある程度の時間を要することがあります。詳細な情報については、nmapのマニュアルページを参照してください。

次のようなコマンドを入力します。

```
echo -en "\rHello\r\f" | netcat -w 1 IP-address port
cat file | netcat -w 1 IP-address port
```

これは、このポートを通してプリンタを使用できるかどうかをテストするために、該当のポートへ文字列またはファイルを直接送信します。

## 11.8.5 エラーメッセージを生成しない異常なプリントアウト

印刷システムの観点では、CUPSバックエンドが受信側(プリンタ)へのデータ転送を完了した段階で、印刷ジョブは完了します。受信側でそれ以降の処理が失敗した場合(たとえば、プリンタがそのプリンタ固有のデータを印刷できない)、印刷システムはそのことを検出しません。プリンタがそのプリンタ固有のデータを印刷できない場合、そのプリンタにより適していると考えられる他のPPDファイルを選択します。

## 11.8.6 無効にされたキュー

受信側へのデータ転送が数回の試行後に完全に失敗した場合、usbやsocketなどのCUPSバックエンドは印刷システム(より正確にはcupsd)にエラーを報告します。データ転送が不可能であると報告する前に、バックエンドは、試行に意味があるかどうか、また何回の試行に意味があるかを判断します。それ以上の試行は無駄に終わる可能性があるため、cupsdは該当のキューへの印刷を無効にします。問題の原因を取り除いた後、システム管理者はcupsenableコマンドを使用して、印刷を再度有効にする必要があります。

## 11.8.7 CUPS参照:印刷ジョブの削除

CUPSネットワークサーバが参照機能を使用して自らのキューをクライアントホストへブロードキャストし、クライアントホスト側で適切なローカルcupsdがアクティブになっている場合、クライアント側のcupsdはアプリケーションから印刷ジョブを受け付け、サーバ側のcupsdへそれらを転送します。cupsdが印刷ジョブを受け付けた段階で、そのジョブに新しいジョブ番号が割り当てられます。したがって、クライアントホスト上のジョブ番号は、サーバ上のジョブ番号とは異なっています。印刷ジョブは通常、即座に転送されるので、クライアントホスト上でジョブ番号を使用してそのジョブを削除することはできません。クライアント側のcupsdは、サーバ側のcupsdへの転送が完了した段階で、その印刷ジョブは完了したと考えるからです。

サーバ上にある印刷ジョブを削除するには、`lpstat -h cups.example.com -o`などのコマンドを使用してサーバ上でジョブ番号を判断します。サーバ

がまだその印刷ジョブを完了していない(つまり、プリンタへ完全に送信していない)ことが前提条件です。このジョブ番号を使用して、サーバ上にある印刷ジョブを削除できます。

```
cancel -h cups.example.com queue-jobnumber
```

## 11.8.8 異常な印刷ジョブとデータ転送エラー

印刷プロセスの実行中に、管理者がプリンタの電源をオフにして再度オンにした場合、またはコンピュータをシャットダウンしてリブートした場合、印刷ジョブはキュー内にとどまっていて、印刷が再開されます。異常な印刷ジョブは、cancelを使用してキューから削除する必要があります。

印刷ジョブが異常な場合、またはホストとプリンタの間で通信エラーが発生した場合、プリンタはデータを正しく処理できなくなるので、文字化けのような大量のページを印刷することがあります。この状態を処理するには、次の処理を実行します。

- 1 プリンタの動作を停止するために、インクジェットプリンタの場合、すべての用紙を取り除きます。レーザープリンタの場合、用紙トレイを開けます。上位機種プリンタでは、現在のプリントアウトをキャンセルするボタンを用意していることもあります。
- 2 この時点で、印刷ジョブはキューに残っている可能性があります。ジョブがキューから削除されるのは、ジョブ全体をプリンタへ送信した後に限られるからです。lpstat -o(またはlpstat -h cups.example.com -o)を使用して、どのキューが現在印刷に使用されているかを確認します。cancel queue-jobnumber(またはcancel -h cups.example.com queue-jobnumber)を使用して、該当の印刷ジョブを削除します。
- 3 印刷ジョブがすでにキューから削除されたにもかかわらず、一部のデータが依然として、プリンタへ送信され続けることもあります。CUPSバックエンドプロセスが、引き続き該当のキューを対象として動作しているかどうかをチェックし、その処理を終了します。たとえば、プリンタがパラレルポートに接続されている場合、fuser -k /dev/lp0コマンドを使用して、引き続きそのプリンタ(より正確に表現すると、パラレルポート)にアクセスしているすべてのプロセスを終了することができます。

- 4 ある程度の時間にわたって電源をオフにして、プリンタを完全にリセットします。その後、紙を元に戻し、プリンタの電源をオンにします。

## 11.8.9 CUPS印刷システムのデバッグ

CUPS印刷システムの問題を特定するために、次の一般的な処理を実行してください。

- 1 `/etc/cups/cupsd.conf`内に、`LogLevel debug`を設定します。
- 2 `cupsd`コマンドを停止します。
- 3 `/var/log/cups/error_log*`を削除して、大規模なログファイルから検索を行うことを避けます。
- 4 `cupsd`を起動します。
- 5 問題の原因となったアクションをもう一度実行します。
- 6 `/var/log/cups/error_log*`内のメッセージを確認し、問題の原因を識別します。

## 11.8.10 詳細情報

Novell Knowledgebase (<http://support.novell.com/>)では、さまざまな個別の問題のソリューションが紹介されています。CUPSのテキスト検索機能により関連する記事を見つけてください。



# udevを使用した動的カーネル デバイス管理

# 12

実行中のシステムで、カーネルは、ほとんどどのデバイスでも追加または削除できます。デバイス状態の変更(デバイスが接続されているか、または取り外されたか)をユーザスペースに反映させる必要があります。デバイスは、接続後、検出されるとすぐに設定されなければなりません。また、特定のデバイスのユーザは、このデバイスの状態に関する変更について通知を受ける必要があります。udevは、/devディレクトリ内にデバイスノードファイルおよびシンボリックリンクを動的に管理することにより、必要なインフラストラクチャを提供します。udevルールによって、外部ツールをカーネルデバイスイベント処理に含めることができます。これにより、カーネルデバイス処理の一部として実行する特定のスクリプトを追加するなど、udevデバイス処理をカスタマイズしたり、デバイス処理中に評価する他のデータを要求およびインポートしたりできます。

## 12.1 /devディレクトリ

/devディレクトリ内のデバイスノードを使用して、対応するカーネルデバイスにアクセスできます。udevにより、/devディレクトリは、カーネルの現在の状態を反映します。カーネルデバイスは、それぞれ1つの対応するデバイスファイルを持ちます。デバイスがシステムから取り外されると、そのデバイスノードは削除されます。

/devディレクトリのコンテンツは一時的なファイルシステム内で管理され、すべてのファイルはシステムの起動時に新規に作成されます。意図的に、手動で作成または変更されたファイルは再起動時に復元されません。対応するカー

ネルデバイスの状態にかかわらず、/devディレクトリ内に常駐する静的ファイルおよびディレクトリは、/lib/udev/devicesディレクトリ内に保管できます。システムの起動時、そのディレクトリのコンテンツは、/lib/udev/devices内のファイルと同じ所有者およびパーミッションの/devディレクトリ内にコピーされます。

## 12.2 カーネルのueventおよびudev

必要なデバイス情報は、システムファイルシステムによってエクスポートされます。カーネルが検出および初期化するすべてのデバイスについて、そのデバイス名を含んだディレクトリが作成されます。このディレクトリには、デバイス固有のプロパティのある属性ファイルが含まれます。

デバイスが追加または削除されるたびに、カーネルはueventを送信して、udevに変更を通知します。udevデーモンは、起動時に/etc/udev/rules.d/\*.rulesから提示されたすべての規則を読み、解析し、メモリ内に保管します。規則ファイルが変更、追加、または削除されると、このデーモンは、udevadm control reload\_rulesコマンドで、すべての規則をメモリに再ロードできます。これは、/etc/init.d/boot.udev reloadの実行時にも行われます。udevルールとその構文の詳細については、[12.6項「udevルールを処理するカーネルデバイスイベントへの影響」](#) (151 ページ)を参照してください。

着信したイベントは、すべて一連のプロバイダルールと一致します。規則によって、イベント環境キーを追加または変更したり、作成するデバイスノードに特定の名前を要求したり、ノードを指すシンボリックリンクを追加したり、またはデバイスノードの作成後に実行するプログラムを追加したりできます。ドライバのコアueventは、カーネルのネットリンクソケットから受信されます。

## 12.3 ドライバ、カーネルモジュールおよびデバイス

カーネルバスドライバは、デバイスを検出します。検出されたデバイスごとに、カーネルは内部デバイス構造を作成し、ドライバコアは、ueventをudev

デーモンを送信します。バスデバイスは、デバイスの種類を示す特別な形式のIDを識別します。通常、これらのIDは、ベンダー、製品IDおよびサブシステム固有の値で構成されています。各バスには、これらのIDに対してMODALIASという独自のスキームを持ちます。カーネルは、デバイス情報を読み取り、この情報からMODALIASID文字列を作成し、イベントとともに文字列を送信します。USBマウスの場合、次のようになります。

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

各デバイスドライバは、既知の処理可能デバイスのエイリアスのリストを持ちます。このリストは、カーネルモジュールファイル自体にも含まれています。depmodプログラムは、IDリストを読み取り、現在使用可能なすべてのモジュールについて、カーネルの/lib/modulesディレクトリ内にmodules.aliasを作成します。このインフラストラクチャにより、MODALIASキーを持つイベントごとにmodprobeを呼び出すだけで簡単にモジュールをロードできます。modprobe \$MODALIASが呼び出されると、そのデバイスに付けられたデバイスエイリアスとモジュールによって提示されるエイリアスとが一致します。一致したエントリが見つかり、そのモジュールがロードされます。これらの処理はすべて、udevによってトリガされ、自動的に行われます。

## 12.4 ブートおよび初期デバイスセットアップ

udevデーモンが実行する前にブートプロセス中に発生するすべてのデバイスイベントは、rootファイルシステムに常駐し、ブート時にはアクセスできないため、これらのイベントは消失します。その消失の埋め合せに、カーネルは、sysfsファイルシステム内の各デバイスのデバイスディレクトリにueventファイルを生成します。そのファイルにaddと書き込むことにより、カーネルは、ブート時に消失したものと同一イベントを再送します。/sys内のすべてのueventファイルを含む単純なループにより、すべてのイベントが再びデバイスノードを作成し、デバイスセットアップを実行します。

たとえば、ブート時に存在するUSBマウスは、ドライバがその時点で使用できないため、初期のブートロジックでは初期化されない場合があります。デバイス検出イベントは、消失し、そのデバイスのカーネルモジュールは検出されません。接続されている可能性があるデバイスを手動で検索する代わりに、udevは、rootファイルシステムが使用可能になった後でカーネルからすべ

てのデバイスイベントを要求するだけです。これにより、このUSBマウスデバイスのイベントが再び実行します。これで、マウントされたrootファイルシステム上のカーネルモジュールが検出され、USBマウスが初期化されます。

ユーザスペースでは、実行時のデバイスのcoldplugシーケンスとデバイス検出との間に明らかな違いはありません。両方の場合も、同じ規則を使用して一致検出が行われ、同じ設定されたプログラムが実行されます。

## 12.5 実行中のudevデーモンの監視

udevadm monitorプログラムを使用すると、ドライバのコアイベントとudevイベントプロセスのタイミングをビジュアル化できます。

```
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行は、カーネルがnetlinkで送信したイベントを示します。UDEV行は、完了したudevイベントハンドラを示します。タイミングは、マイクロ秒で出力されます。UEVENTおよびUDEV間の時間は、udevがこのイベントの処理に要した時間、またはudevデーモンがこのイベントと関連する実行中のイベントとの同期の実行に遅れた時間です。たとえば、ハードディスクパーティションのイベントは常に、メインディスクイベントがハードウェアに問い合わせたデータに依存する可能性があるため、メインデバイスイベントが完了するのを待ちます。

udevadm monitor --envは、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。いずれのメッセージがsyslogに送信されるかを制御するデフォルトのsyslogの優先度は、udev設定ファイルの/etc/udev/udev.conf内で指定されています。実行中のデーモンのログ優先度は、udevadm control log\_priority=level/numberで変更できます。

## 12.6 udevルールを処理するカーネルデバイスイベントへの影響

udevルールは、カーネルがイベント自体に追加するすべてのプロパティ、またはカーネルがsysfsにエクスポートするすべての情報と一致します。また、この規則で、外部プログラムからの追加情報を要求することもできます。各イベントは、指定されたすべての規則と一致します。すべての規則は、/etc/udev/rules.dディレクトリにあります。

規則ファイル内の各行には、少なくとも1つのキー値ペアが含まれています。これらは、一致と割り当てキーという2種類のキーです。すべての一致キーが各値と一致する場合、その規則が適用され、割り当てキーに指定された値が割り当てられます。一致する規則がある場合、デバイスノードの名前を指定、ノードを指すシンボリックリンクを追加、またはイベント処理の一部として指定されたプログラムを実行できます。一致する規則がない場合、デフォルトのデバイスノード名を使用して、デバイスノードが作成されます。データのマッチまたはインポートのためのルール構文およびキーの詳細については、udevのマニュアルページで説明されています。以下に示す規則の例では、udevルール構文の基本を紹介します。これらの規則の例は、すべて、/etc/udev/rules.d/50-udev-default.rulesにあるudevデフォルトルールセットに含まれています。

## 例 12.1 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ルールは、3つのキーで構成されています。その内訳は、一致キーが1つ(KERNEL)、割り当てキーが2つ(MODE、OPTIONS)です。KERNEL一致ルールはconsoleタイプのアイテムをデバイスリストから検索します。正確な一致だけが有効であり、このルールの実行をトリガします。MODEキーは、特別パーミッションをデバイスノードに割り当てます。この例では、読み取り/書き込みパーミッションをこのデバイスの所有者にのみ割り当てます。OPTIONSキーは、この規則をこのタイプのデバイスに適用される最後の規則にします。以降の規則は、この特定デバイスタイプとマッチしても、どのような結果も生じません。

serial devicesルールは、50-udev-default.rulesには存在しなくなりましたが、依然その知識は重要です。この規則は、2つの一致キー(KERNEL、ATTRS)および1つの割り当てキー(SYMLINK)で構成されます。KERNELキーは、ttyUSBタイプのすべてのデバイスを検索します。このキーで\*ワイルドカードを使用すると、これらのデバイスのいくつかとマッチします。2つ目の一致キーATTRSは、ttyUSBデバイスのsysfsにあるproduct属性ファイルに一定の文字列が含まれているかどうかをチェックします。割り当てキー(SYMLINK)は、/dev/pilotの下に、このデバイスへのシンボリックリンクを追加します。このキーで演算子(+=)を使用すると、前/後の規則が他のシンボリックリンクを追加した場合でも、udevはこの操作を追加実行します。この規則は、2つの一致キーを含むので、両方の条件が満たされる場合のみ適用されます。

printerルールは、USBプリンタを対象とし、2つの一致キー(SUBSYSTEM、KERNEL)を含みます。規則全体を適用するには、これらのキーを両方とも適用する必要があります。3つの割り当てキーは、このデバイスタイプの名前付け(NAME)、シンボリックデバイスリンクの作成、(SYMLINK)、およびこのデバイスタイプのグループメンバーシップ(GROUP)を処理します。KERNELキー

で\*ワイルドカードを使用すると、いくつかのlpプリンタデバイスとマッチします。NAMEおよびSYMLINKの両キーで置き換えを使用すると、これらの文字列を内部デバイス名で拡張できます。たとえば、最初のlpUSBプリンタへのシンボリックリンクは/dev/usb/lp0となります。

kernel firmware loaderルールでは、ランタイム時の外部ヘルパースクリプトで、udevが追加ファームウェアをロードします。SUBSYSTEM一致キーは、firmwareサブシステムを検索します。ACTIONキーは、firmwareサブシステムに属するデバイスが追加されているかどうかをチェックします。RUN+=キーは、firmware.shスクリプトの実行をトリガして、ファームウェアを見つけます。

すべての規則に共通する一般的特性は次の通りです。

- 各規則は、カンマで区切られた1つ以上のキー値ペアで構成されます。
- キーの動作は、演算子で決定されます。udevルールは、いくつかの異なる演算子をサポートします。
- 指定する各値は、引用符で囲む必要があります。
- 規則ファイルの各行が1つの規則に相当します。規則が1行を超える場合は、shell構文のように、\を使用して異なる行を結合してください。
- udevルールは、shell型パターンをサポートします。このパターンは、\*、?、および[]の各パターンとマッチします。
- udevルールは、置換をサポートします。

## 12.6.1 udevルールでの演算子の使用

キーを作成する場合は、作成するキーのタイプによって、いくつかの異なる演算子から選択できます。一致キーは、通常、検索値とマッチするか、明示的にミスマッチする値を見つけるためにだけ使用されます。一致キーは、次の演算子のいずれかを含みます。

==

等価の比較。キーに検索パターンが含まれている場合は、そのパターンと一致するすべての結果が有効です。

!=

非等価の比較。キーに検索パターンが含まれている場合は、そのパターンと一致するすべての結果が有効です。

割り当てキーでは、次のどの演算子でも使用できます。

=

値をキーに割り当てます。既に値のリストで構成されているキーはリセットされ、指定した1つの値だけが割り当てられます。

+=

エントリのリストを含むキーに値を追加します。

:=

最終値を割り当てます。以降の規則による変更は許可されません。

## 12.6.2 udevルールでの置換の使用

udevルールは、ブレースホルダと置換の使用をサポートします。それらは、他のスクリプトでの使用と同様な方法で使用します。udevルールでは、次の置換を使用できます。

%r、\$root

デフォルトのデバイスディレクトリ/dev。

%p、\$devpath

DEVPATHの値。

%k、\$kernel

KERNELの値または内部デバイス名。

%n、\$number

デバイス番号。

%N、\$stemnode

デバイスファイルの一時名。

%M、\$major

デバイスのメジャー番号。



%m、\$minor

デバイスのマイナー番号。

%s{attribute}、\$attr{attribute}

sysfs属性の値(attributeで指定)。

%E{variable}、\$attr{variable}

環境変数の値(variableで指定)。

%c、\$result

PROGRAMの出力。

%%

%文字。

\$\$

\$文字。

## 12.6.3 udev一致キーの使用

一致キーは、udevルールの適用前に満たす必要のある条件を記述します。次の一致キーが使用可能です。

ACTION

イベント動作の名前。たとえば、addまたはremove(デバイスの追加または削除の場合)。

DEVPATH

イベントデバイスのデバイスパス。たとえば、  
DEVPATH=/bus/pci/drivers/ipw3945(ipw3945ドライバに関連するすべてのイベントを検索する場合)。

KERNEL

イベントデバイスの内部(カーネル)名。

SUBSYSTEM

イベントデバイスのサブシステム。たとえば、SUBSYSTEM=usb(USBデバイスに関連するすべてのイベント用)。

ATTR{*filename*}

イベントデバイスのsysfs属性。vendor属性ファイル名に含まれた文字列とマッチするには、たとえば、ATTR{vendor}=="On[sS]tream"を使用できます。

KERNELS

デバイスパスを上方に検索して、一致するデバイス名を見つけます。

SUBSYSTEMS

デバイスパスを上方に検索して、一致するデバイスサブシステム名を見つけます。

DRIVERS

デバイスパスを上方に検索して、一致するデバイスドライバ名を見つけます。

ATTRS{*filename*}

デバイスパスを上方に検索して、一致するsysfs属性値を持つデバイスを見つけます。

ENV{*key*}

環境変数の値。たとえば、ENV{ID\_BUS}="ieee1394"でFireWire bus IDに関連するすべてのイベントを検索します。

PROGRAM

外部プログラムを実行します。成功の場合は、プログラムが終了コードとしてゼロを返します。stdoutに印刷されるプログラムの出力は、RESULTキーで使用できます。

RESULT

最後のPROGRAM呼び出しの出力文字列とマッチします。このキーは、PROGRAMキーと同じ規則に含めるか、それ以降のキーに含めてください。.

## 12.6.4 udev割り当てキーの使用

上記で説明した一致キーとは対照的に、割り当てキーは、満たすべき条件を記述しません。割り当てキーは、udevにより保守されるデバイスノードに値、名前、および動作を割り当てます。

#### NAME

作成するデバイスノードの名前。いったん規則でノード名が設定されると、このノードのNAMEキーを持つ他の規則はすべて無視されます。

#### SYMLINK

作成するノードに関連するシンボリックリンクの名前。複数の一致ルールで、デバイスノードとともに作成するシンボリックリンクを追加できます。1つのルール内で、スペース文字でシンボリックリンク名を区切ることで、1つのノードに複数のシンボリックリンクを指定することもできます。

#### OWNER、GROUP、MODE

新しいデバイスノードのパーミッションここで指定する値は、既にコンパイルされている値を上書きします。

#### ATTR{key}

イベントデバイスのsysfs属性に書き込む値を指定します。==演算子を使用すると、このキーは、sysfs属性の値とのマッチングにも使用されます。

#### ENV{key}

環境への変数のエクスポートをudevに指示します。==演算子を指定すると、このキーは、環境変数とのマッチングにも使用されます。

#### RUN

このデバイスに対して実行されるプログラムのリストにプログラムを追加するように、udevに指示します。このデバイスのイベントをブロックしないようにするため、これは非常に短いタスクに限定してください。

#### LABEL

GOTOのジャンプ先にするラベルを追加します。

#### GOTO

いくつかの規則をスキップし、GOTOキーで参照されるラベルを含む規則から続行するように、udevに指示します。

#### IMPORT{type}

変数をイベント環境(外部プログラムの出力など)にロードします。udevは、いくつかの異なるタイプの変数をインポートします。タイプが指定されていない場合、udevは、ファイルパーミッションの実行可能ビットに基づいてタイプを決定しようとします。

- `program` - 外部プログラムを実行し、その出力をインポートします。
- `file` - テキストファイルをインポートします。
- `parent` - 親デバイスから保存されたキーをインポートします。

#### WAIT\_FOR\_SYSFS

一定のデバイスに指定された`sysfs`ファイルが作成されるまで、`udev`を待機させます。たとえば、`WAIT_FOR_SYSFS="ioerr_cnt"`では、`ioerr_cnt`ファイルが作成されるまで、`udev`を待機させます。

#### オプション

`OPTION`キーには、次の可能な値があります。

- `last_rule` - 以降のすべての規則を無視します。
- `ignore_device` - このイベントを完全に無視します。
- `ignore_remove` - このデバイスの以降のすべての削除イベントを無視します。
- `all_partitions` - ブロックデバイス上のすべての使用可能なパーティションにデバイスノードを作成します。

## 12.7 永続的なデバイス名の使用

動的デバイスディレクトリおよび`udev`ルールインフラストラクチャによって、認識順序やデバイスの接続手段にかかわらず、すべてのディスクデバイスに一定の名前を指定できるようになりました。カーネルが作成する適切なブロックデバイスはすべて、特定のバス、ドライブタイプまたはファイルシステムに関する特別な知識を備えたツールによって診断されます。動的カーネルによって指定されるデバイスノード名とともに、`udev`は、デバイスを指す永続的なシンボリックリンクのクラスを維持します。

```
/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
```

## 12.8 udevで使用するファイル

/sys/\*

Linuxカーネルによって提供される仮想ファイルシステム。現在知られているデバイスをすべてエクスポートします。この情報は、udevによって使用され、/dev内にデバイスノードが作成されます。

/dev/\*

動的に作成されるデバイスノードと静的コンテンツ。ブート時に/lib/udev/devices/\*からコピーされます。

次のファイルおよびディレクトリには、udevインフラストラクチャの重要な要素が含まれています。

/etc/udev/udev.conf

メインのudev設定ファイル。

/etc/udev/rules.d/\*

規則と一致するudevイベント。

/lib/udev/devices/\*

静的/devコンテンツ。

`/lib/udev/*`

udevルールから呼び出されるヘルパープログラム.

## 12.9 詳細情報

udevインフラストラクチャの詳細については、次のマニュアルページを参照してください。

**udev**

udev、キー、規則およびその他の重要な設定問題に関する一般情報。

**udevadm**

udevadmは、udevのランタイム動作を制御し、カーネルイベントを要求し、イベントキューを管理し、簡単なデバッグメカニズムを提供します。

**udev**

udevイベント管理デーモンに関する情報。

## X Windowシステム

X Window System (X11)は、UNIX系のグラフィカルユーザインタフェースで、事実上の標準となっています。Xはネットワークベースであり、あるホスト上で起動されたアプリケーションを、任意のネットワーク(LANやインターネット)を介して接続されている他のホスト上で表示できるようにします。この章ではX Window System環境のセットアップと最適化について説明し、SUSE® Linux Enterprise Desktopでのフォント使用の背景情報を提供します。

### 13.1 X Window システムの手動設定

デフォルトでは、X Windowシステムは項「Setting Up Graphics Card and Monitor」(第6章 *Setting Up Hardware Components with YaST*, ↑導入ガイド)に説明されているSaX2インタフェースを使って設定されます。代わりに環境設定ファイルを編集して、手動設定することもできます。

---

**警告: X環境設定ファイルに不適切な設定を行うとハードウェアが損傷する可能性があります**

X Window Systemの設定は慎重に行う必要があります。設定が完了するまでは、X Window Systemを起動しないでください。システムが正しく設定されていないと、ハードウェアが復元不能な損傷を受ける可能性があります(特に固定周波数モニタの場合)。本書およびSUSE Linux Enterprise Desktopの作成者は、このような原因による損傷や損害に対していかなる責任も負いません。この情報は慎重に調査されたものですが、ここで説明する方法がすべて正しく、ハードウェアが損傷を受けないという保証はありません。

---

コマンドsax2で/etc/X11/xorg.confファイルが作成されます。これはX Window Systemの基本設定ファイルです。このファイルには、グラフィックカード、マウス、およびモニタに関する設定がすべて含まれています。

---

**重要項目: X -configureの使用**

SUSE Linux Enterprise DesktopのSaX2で失敗した場合は、X -configureを使ってXセットアップの設定を行ってください。セットアップ専有(ソフトウェア)

---

ここでは、設定ファイル/etc/X11/xorg.confの構造について説明します。xorg.confは複数のセクションで構成され、各セクションは設定の特定の側面を取り扱います。各セクションは、キーワードSection <designation>で始まってキーワードEndSectionで終わります。すべてのセクションで、以下の表記規則を使用します。

```
Section "designation"
    entry 1
    entry 2
    entry n
EndSection
```

使用可能なセクションのタイプのリストは表 13.1. 「**/etc/X11/xorg.confのセクション**」 (162 ページ)にあります。

**表 13.1** /etc/X11/xorg.confのセクション

タイプ	意味
ファイル	フォントとRGBカラーテーブルで使用するパス。
ServerFlags	サーバ動作の汎用スイッチ。
モジュール	サーバがロードする必要があるモジュールリスト
InputDevice	キーボードや特殊入力デバイス(タッチパッド、ジョイスティックなど)といった入力デバイスを設定します。このセクションで重要なパラメータはDriverと、ProtocolおよびDeviceを定義するオプションです。通常、コン



タイプ	意味
	<p>コンピュータに接続した1つのデバイスごとに1つの <code>InputDevice</code> があります。</p>
Monitor	<p>使用するモニタ。このセクションの重要な要素は、後で <code>Screen</code> の定義で参照する ID、リフレッシュレートの <code>VertRefresh</code>、および同期周波数の制限(<code>HorizSync</code> および <code>VertRefresh</code>) です。設定値は MHz、kHz、および Hz 単位です。通常、サーバはモニタ仕様に対応しない <b>modeline</b> を拒否します。このため、意図せずに高すぎる周波数がモニタに送信されるのを防止できます。</p>
Modes	<p>特定の画面解像度の <b>modeline</b> パラメータ。これらのパラメータは、ユーザ指定の値に基づいて <b>SaX2</b> で計算でき、通常は変更不要です。固定周波数モニタに接続する場合は、この時点で手動で介入します。個々の数値の意味の詳細については、<b>HOWTO</b> ファイル <code>/usr/share/doc/howto/en/html/XFree86-Video-Timings-HOWTO</code> を参照してください (<code>howtoenh</code> パッケージ内)。<b>VESA</b> モードを手動で計算する場合は、ツール <code>cvt</code> を使用できます。たとえば、<b>1680x1050@60Hz</b> モニタの <b>modeline</b> を計算する場合は、コマンド <code>cvt 1680 1050 60</code> を使用します。</p>
Device	<p>特定のグラフィックカード。グラフィックカードは記述名で参照されます。このセクションで利用可能なオプションは、使用するドライバに大きく依存します。たとえば、<code>i810</code> ドライバを使用する場合は、マニュアルページ <code>man 4 i810</code> に使用可能なオプションの詳細が記載されています。</p>
画面	<p><code>Monitor</code> と <code>Device</code> を組み合わせて、<b>X.Org</b> に必要な設定を形成します。<code>Display</code> サブセクションでは、仮想画面のサイズ(<code>Virtual</code>)、<code>ViewPort</code>、およびこの画面で使用する <code>Modes</code> を指定します。</p>

タイプ	意味
	一部のドライバでは、いずれかの場所にあるDisplayセクションにすべての使用設定が存在しなければならないことに注意してください。たとえば、ラップトップを使用している場合で、内部LCDより大きい外部モニタを使用するときは、内部LCDによりサポートされる以上の分解能をModes行の最後に追加することが必要になる場合があります。
ServerLayout	シングルまたはマルチヘッド設定のレイアウト。このセクションにより、入力デバイスInputDeviceと表示デバイスScreenがバインドされます。
DRI	DRI(Direct Rendering Infrastructure)の情報を提供します。

ここでは、Monitor、Device、およびScreenについて詳しく説明します。他のセクションの詳細については、X.Orgおよびxorg.confのマニュアルページを参照してください。

xorg.confには、複数の異なるMonitorおよびDeviceセクションを記述できます。複数のScreenセクションを記述することも可能です。ServerLayoutセクションでは、このセクションのうち使用するものを判定します。

## 13.1.1 Screenセクション

Screenセクションでは、MonitorセクションとDeviceセクションを組み合わせ、どの解像度とカラー設定を使用するかを決定します。Screenセクションは例 13.1. 「ファイル/etc/X11/xorg.confのScreenセクション」 (165 ページ) のようになります。

### 例 13.1 ファイル/etc/X11/xorg.confのScreenセクション

```
Section "Screen"❶
    DefaultDepth 16❷
    SubSection "Display"❸
        Depth 16❹
        Modes "1152x864" "1024x768" "800x600"❺
        Virtual 1152x864❻
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1280x1024"
    EndSubSection
    Device "Device[0]"
    Identifier "Screen[0]"❼
    Monitor "Monitor[0]"
EndSection
```

- ❶ Sectionはセクションタイプを判定し、この場合はScreenになります。
- ❷ DefaultDepthは、色深度が明示的に指定されていない場合にデフォルトで使用する色深度を示します。
- ❸ 各色深度に対して、異なるDisplayサブセクションが指定されます。
- ❹ Depthは、このセットのDisplay設定とともに使用する色深度を示します。8、15、16、24、および32を指定できますが、すべてのXサーバモジュールまたは解像度がこれらの値をすべてサポートしている訳ではありません。
- ❺ Modesセクションは、可能な画面解像度のリストから成り立っています。Xサーバは、このリストを左から右に検査します。解像度ごとに、XサーバはModesセクション内で適切なModelineを検索します。Modelineは、モニタとグラフィックカード両方の機能に応じて異なります。Monitor設定により、Modelineが決まります。

最初に検出される解像度はDefault modeです。Ctrl + Alt + +(数字パッド上のキー)を使用すると、リスト内で右隣の解像度に切り替えることが

できます。以前の値に切り替えるには、**Ctrl+Alt+-**(数字パッド上のキー)を使用します。これにより、**X**の実行中に解像度を変更できます。

- ⑥ Depth 16が指定されているDisplayサブセクションの最終行は、仮想画面のサイズを指します。仮想画面の最大許容サイズは、モニタの最大解像度ではなく、グラフィックカードにインストールされているメモリの容量と必要なカラー設定に応じて異なります。この行を省略すると、仮想解像度は物理解像度と同じになります。最近のグラフィックカードはビデオメモリ容量が大きくなってきているため、きわめて大型の仮想デスクトップを作成できます。ただし、ビデオメモリのほとんどが仮想デスクトップを占めると、3D機能を使用できなくなる場合があります。たとえば、カードのビデオRAMが16MBであれば、仮想画面には8ビットカラー深度で最大4096x4096ピクセルのサイズを設定できます。ただし、特にアクセラレータカードの場合は、仮想画面にメモリすべてを使用しないことをお勧めします。この種のカードのメモリは、複数のフォントやグラフィックキャッシュにも使用されるからです。
- ⑦ Identifier行(ここではScreen[0])では、このセクションに以降のServerLayoutセクションで一意に参照できる定義済みの名前を割り当てています。Device行とMonitor行では、この定義に属しているグラフィックカードとモニタを指定しています。これらは、対応する名前または識別子を持つDeviceおよびMonitorセクションにリンクされます。これらのセクションの詳細については、以下を参照してください。

## 13.1.2 Deviceセクション

Deviceセクションでは、特定のグラフィックカードを記述します。名前が異なっていれば、キーワードIdentifierを使用してxorg.conf内で必要な数だけデバイスエントリを指定できます。複数のグラフィックカードをインストールしている場合、セクションには順番に番号が付けられます。最初のセクションはDevice[0]、2番目のセクションはDevice[[1]]となります。次のファイルは、Matrox Millennium PCIグラフィックカードが搭載されているコンピュータのDeviceセクションから抜粋したものです(SaX2が設定)。

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"①
    Driver         "mga"②
    Identifier     "Device[0]"
    VendorName     "Matrox"
```

```
Option      "sw_cursor"  
EndSection
```

- ❶ BusIDは、グラフィックカードがインストールされているPCIスロットまたはAGPスロットの定義です。これは、`lspci`コマンドで表示されるIDと一致します。Xサーバは10進形式による詳細を必要としますが、`lspci`ではこれらが16進形式で表示されます。BusIDの値は、SaX2が自動検出します。
- ❷ Driverの値はSaX2が自動的に検出し、グラフィックカードで使用するドライバを指定します。カードがMatrox Millenniumである場合は、ドライバモジュールはmgaと呼ばれます。Xサーバは、driversサブディレクトリのFilesセクションで定義されているModulePathを検索します。標準インストールでは、これは/usr/lib/xorg/modules/driversディレクトリ、または64ビットオペレーティングシステムディレクトリでは/usr/lib64/xorg/modules/driversディレクトリです。名前には\_drv.oが追加されるので、mgaドライバの場合は、ドライバファイルmga\_drv.oがロードされます。

Xサーバやドライバの動作は、その他のオプションを使用して変更することもできます。その一例がDeviceセクションで設定するオプションsw\_cursorです。このオプションは、ハードウェアのマウスカーソルを無効にし、ソフトウェアを使用してマウスカーソルを示します。ドライバモジュールによっては、さまざまなオプションを使用できます。各オプションは、ディレクトリ/usr/share/doc/package\_name内のドライバモジュール記述ファイル内にあります。通常、有効なオプションについてはマニュアルページ(`man xorg.conf`, `man 4 <ドライバモジュール>`および`man 4 chips`)でも確認できます。

グラフィックカードに複数のビデオコネクタがある場合、この1枚のカードの異なるデバイスを単一ビューとして設定できます。SaX2を使用してグラフィックインタフェースをこのように設定します。

### 13.1.3 MonitorセクションとModesセクション

Deviceセクションと同様に、MonitorセクションとModesセクションでもモニタを1つずつ記述します。設定ファイル/etc/X11/xorg.confでは、Monitor

セクションを必要な数だけ指定できます。Monitorセクションはそれぞれ、UseModes行があるModesセクションを参照します。MonitorセクションにModesセクションがない場合、Xサーバは該当する値を一般的な同期の値から計算します。サーバレイアウトセクションでは、どのMonitorセクションが関係するかを指定します。

熟練者以外は、モニタ定義を設定しないでください。**modeline**は、Monitorセクションで重要な役割を果たします。**modeline**では、関連解像度の水平と垂直のタイミングを設定します。モニタ特性、特に許容周波数は、Monitorセクションに格納されます。標準VESAモードは、ユーティリティcvtにより生成できます。詳細については、マニュアルページcvtman cvtを参照してください。

---

## 警告

モニタおよびグラフィックカード機能の詳細な知識がない場合は、**modeline**を変更しないでください。モニタに重大な損傷が生じることがあります。

---

独自のモニタ記述を作成する場合は、/usr/share/X11/doc内のドキュメントを熟読する必要があります。PDFおよびHTMLページを参照するために、パッケージxorg-x11-docをインストールします。

**modeline**の手動指定が必要になることはほとんどありません。最新のマルチシンクモニタを使用している場合、許容周波数と最適解像度は、**SaX2**設定のセクションで説明したように、原則としてXサーバがDDCを介してモニタから直接読み込みます。何らかの原因で直接読み込めない場合は、Xサーバに付属するVESAモードの1つを使用してください。このモードは、実際にはグラフィックカードとモニタの大半の組み合わせに機能します。

## 13.2 フォントのインストールと設定

SUSE Linux Enterprise Desktopで追加のフォントをインストールするのは簡単です。フォントを、X 11フォントパスにある任意のディレクトリにコピーするだけです(13.2.1項「X11コアフォント」(170 ページ)を参照)。フォントの使用を有効にするには、インストールディレクトリが/etc/fonts/fonts.confに設定されているディレクトリのサブディレクトリでなければなりません

(13.2.2項「Xft」(171 ページ)を参照)。または、このファイルを/etc/fonts/suse-font-dirs.confに入れなければなりません。

以下は、/etc/fonts/fonts.confから抜粋したものです。このファイルは、大半の設定に適合する標準設定ファイルです。また、インクルード済みのディレクトリ/etc/fonts/conf.dを定義します。このディレクトリには、すべてのファイルまたは2桁の数字で始まるシンボリックリンクがfontconfigによりロードされます。この機能の詳細な説明は、/etc/fonts/conf.d/READMEを参照してください。

```
<!-- 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>
<include ignore_missing="yes">conf.d</include>
```

/etc/fonts/suse-font-dirs.confは、OpenOffice.org、Java、またはAdobe Acrobat Readerなどのアプリケーション(多くはサードパーティ製)に付属のフォントを取り込むために、自動的に生成されます。/etc/fonts/suse-font-dirs.confの一般的なエントリは次のようになります:

```
<dir>/usr/lib64/ooo-2.0/share/fonts</dir>
<dir>/usr/lib/jvm/java-1_4_2-sun-1.4.2.11/jre/lib/fonts</dir>
<dir>/usr/lib64/jvm/java-1.5.0-sun-1.5.0_07/jre/lib/fonts</dir>
<dir>/usr/X11R6/lib/Adobe/7/Resource/Font</dir>
<dir>/usr/X11R6/lib/Adobe/7/Resource/Font/PFM</dir>
```

システム全体に追加フォントを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手動コピーしてください(rootとして)。また、この作業は、KDEコントロールセンターでKDEフォントインストーラを使用して行うこともできます。結果は同じです。

フォントを実際に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、マウントされているWindowsパーティション上にライセンスを取得しているフォントがあり、それらのフォントを使用したい場合は、シンボリックリンクを作成します。次に、SuSEconfig--module fontsコマンドを実行します。

`SuSEconfig--module fonts` コマンドは、フォントを設定するスクリプト、`/usr/sbin/fonts-config` を実行します。このスクリプトの詳細については、マニュアルページ(`man fonts-config`)を参照してください。

手順は、ビットマップフォント、TrueTypeフォントとOpenTypeフォント、およびType1 (PostScript)フォントの場合と同様です。これらのタイプのフォントはすべて、任意のディレクトリにインストールできます。

X.Orgには、従来の *[X11 コアフォントシステム]* と、新たに設計された *[Xft およびfontconfig]* システムの2種類のまったく異なるフォントシステムが含まれています。以降のセクションでは、これらの2つのシステムについて簡単に説明します。

## 13.2.1 X11 コアフォント

今日、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、Type1フォント、TrueTypeとOpenTypeフォントなどのスケーラブルフォントもサポートしています。スケーラブルフォントは、アンチエイリアスとサブピクセルレンダリングなしでサポートされており、多数の言語用のグリフを持つ大きいスケーラブルフォントのロードには時間がかかります。Unicodeフォントもサポートされていますが、使用すると時間がかかり、より多くのメモリが必要になります。

X11 コアフォントシステムには、その他にも固有の弱点がいくつかあります。時代遅れになっており、これ以上拡張することはできません。下位互換性のために保持されていますが、可能なときはいつでも、新しいXftおよびfontconfigシステムを使用してください。

Xサーバは、操作のためにどのようなフォントが使用可能で、そのフォントがシステム内のどこにあるかを認識する必要があります。この情報は、有効なすべてのシステムフォントディレクトリへのパスを含むFontPath変数で処理されます。これらの各ディレクトリでは、ファイル`fonts.dir`にそのディレクトリ内で使用可能なフォントのリストがあります。FontPathは、起動時にXサーバにより生成されます。設定ファイル`/etc/X11/xorg.conf`の各FontPathエントリ内で、有効なファイル`fonts.dir`が検索されます。これらのエントリは、Filesセクションにあります。実際のFontPathを表示するには、`xset q`を使用します。このパスは、`xset`を使用して実行時に変更す



ることもできます。パスを追加するには、`xset+fp <path>`を使用します。不要なパスを削除するには、`xset-fp <path>`を使用します。

Xサーバがすでにアクティブである場合、マウントされたディレクトリに新たにインストールされたフォントは、コマンド`xsetfp rehash`で使用可能にできます。このコマンドは、`SuSEconfig--module fonts`によって実行されます。コマンド`xset`が実行中のXサーバにアクセスする必要がある場合、これは、`SuSEconfig--module fonts`が実行中のXサーバにアクセスできるシェルから起動されている場合にのみ可能です。これを実行する簡単な方法は、`su`と`root`パスワードを入力して、ルートパーミッションを取得することです。`su`によってXサーバを起動したユーザのアクセス許可が`root`シェルに転送されます。フォントが正しくインストールされ、X11コアフォントシステムを介して使用可能かどうか検査するには、コマンド`xlsfonts`を使用して、すべての使用可能なフォントのリストを表示します。

デフォルトでは、SUSE Linux Enterprise DesktopはUTF-8ロケールを使用します。そのため、Unicodeフォントを使用するようにします(`xlsfonts`の出力中で`iso10646-1`で終了するフォント名)。使用可能なすべてのUnicodeフォントは、`xlsfonts | grep iso10646-1`コマンドでリストを表示できます。SUSE Linux Enterprise Desktopで使用可能なほとんどすべてのUnicodeフォントには、少なくともヨーロッパ言語に必要なグリフが含まれています(以前は`iso-8859-*`としてエンコードされていました)。

## 13.2.2 Xft

最初から、Xftのプログラマは、アンチエイリアスを含むスケーラブルフォントが適切にサポートされるようにしています。Xftが使用された場合、フォントは、X11コアフォントシステムにおけるXサーバではなく、そのフォントを使用するアプリケーションによってレンダリングされます。このようにすると、それぞれの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、グリフのレンダリング方法を完全に制御できます。これが、多数の言語においてテキストを正しく表示するための基本となっています。フォントファイルに直接アクセスできることは、印刷のためにフォントを組み込んで、画面出力と同じ印刷出力を得るのに役立ちます。

SUSE Linux Enterprise Desktopでは、2種類のデスクトップ環境KDEとGNOME、Mozilla、および他の多くのアプリケーションが、すでにXftをデフォルトで使

用しています。そのため、Xftはすでに、古いX11コアフォントシステムよりも多くのアプリケーションで使用されています。

Xftは、fontconfigライブラリを使ってフォントを検索し、フォントのレンダリング方法を制御します。fontconfigのプロパティは、グローバルな設定ファイル/etc/fonts/fonts.confによって制御されます。特別設定は、/etc/fonts/local.confおよびユーザ固有の設定ファイル~/.fonts.confに追加する必要があります。これらのfontconfig設定ファイルはどちらも、以下の行で始まっていなければなりません。

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

さらに、以下の行で終わっていなければなりません。

```
</fontconfig>
```

フォントを検索するためのディレクトリを追加するには、以下のような行を付加します。

```
<dir>/usr/local/share/fonts/</dir>
```

ただし、これは通常、必要ありません。デフォルトで、ユーザ固有のディレクトリ~/.fontsは、すでに/etc/fonts/fonts.confに入っています。その結果、追加のフォントをインストールするには、それらのフォントを~/.fontsにコピーするだけです。

また、フォントの見栄えを制御する規則を導入することもできます。例えば、次のように入力して、すべてのフォントについてアンチエイリアスを無効にします。

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

あるいは次のように入力します。

```
<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>
```

この場合、特定のフォントのアンチエイリアスが無効になります。

デフォルトで、ほとんどのアプリケーションは、フォント名のsans-serif (または等価のsans)、serif、あるいはmonospaceを使用します。これらは、実際のフォントではなく、言語設定に応じて適切なフォントに解決されるエイリアスにすぎません。

ユーザは、規則を~/.fonts.confファイルに追加して、それらのエイリアスを簡単に好みのフォントに変換できます。

```
<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>
```

ほとんどすべてのアプリケーションで、これらのエイリアスがデフォルトで使われるので、システム全体が影響を受けます。そのため、個々のアプリケーションでフォント設定を変更しなくても、ほとんどどこでも好みのフォントを簡単に使用できます。

fc-listを使用して、どのフォントがインストールされており、使用可能になっているか調べます。たとえば、fc-listコマンドを実行すると、すべてのフォントのリストが表示されます。使用可能なスケーラブルフォント (:scalable=true)の内、どのフォントがHebrew (:lang=he)に必要なすべてのグリフ、それらのフォント名(family)、それらのスタイル(style)、それらの幅(weight)、およびフォントを含むファイルの名前を含んでいるか調べるには、次のコマンドを入力します。

```
fc-list ":lang=he:scalable=true" family style weight
```

上記のコマンドの出力は、以下のようになります。

```
Lucida Sans:style=Demibold:weight=200
DejaVu Sans:style=Bold Oblique:weight=200
Lucida Sans Typewriter:style=Bold:weight=200
FreeSerif:style=Bold, polkrepko:weight=200
FreeSerif:style=Italic, ležeče:weight=80
FreeSans:style=Medium, navadno:weight=80
DejaVu Sans:style=Oblique:weight=80
FreeSans:style=Oblique, ležeče:weight=80
```

`fc-list`で調べることができる重要なパラメータ：

**表 13.2** *fc-list*のパラメータ

パラメータ	意味と有効な値
family	フォントファミリの名前。たとえば、FreeSans
foundry	フォントメーカー。たとえば、urw
style (スタイル)	フォントスタイル。たとえば、Medium、Regular、Bold、Italic、Heavy。
lang	フォントがサポートする言語。例えば、ドイツ語にはde、日本語にはja、繁体字中国語にはzh-TW、簡体字中国語にはzh-CN
weight	フォント幅。たとえば、通常では80、ボールドでは200。
slant	スラント。通常、なしでは0、イタリックでは100
file	フォントを含むファイルの名前
outline	アウトラインフォントではtrue、他のフォントではfalse。

パラメータ	意味と有効な値
scalable	スケーラブルフォントではtrue、他のフォントではfalse。
bitmap	ビットマップフォントではtrue、他のフォントではfalse。
pixelsize	ピクセル単位でのフォントサイズ。fc-listとの関連で、このオプションはビットマップフォントでのみ有効

## 13.3 詳細情報

X11に関する詳細情報を入手するには、xorg-x11-docおよびhowtoenhパッケージをインストールしてください。X11開発の詳細情報は、プロジェクトのホームページ<http://www.x.org>で参照できます。

パッケージxorg-x11-driver-videoで配布されるドライバの大半については、マニュアルページに詳細が記載されてます。たとえば、radeonドライバを使用する場合は、`man 4 radeon`でドライバの詳細を参照できます。

サードパーティーのドライバ情報は、`/usr/share/doc/packages/<package_name>`に記載されています。たとえば、x11-video-nvidiaG01の場合、パッケージのインストール後は、`/usr/share/doc/packages/x11-video-nvidiaG01`でマニュアルを参照できます。



# FUSEによるファイルシステム へのアクセス

# 14

FUSEは、*file system in userspace*の頭字語です。これは、特権のないユーザとしてファイルシステムを設定およびマウントできることを意味します。通常、このタスクを行うためには、rootにいる必要があります。FUSE自体は、カーネルモジュールです。FUSEは、プラグインと組み合わせることで、ほとんどすべてのファイルシステムにアクセスするように拡張できます(リモートSSH接続、ISOイメージなど)。

## 14.1 FUSEの設定

FUSEを使用するには、まず、fuseパッケージをインストールする必要があります。使用するファイルシステムによって、異なるパッケージからプラグインを追加する必要があります。これらのパッケージの検索には、YaSTを使用し、キーワードにはfuseまたはfile systemを使用します。

一般的には、FUSEは設定の必要がなく、そのまま使用します。ただし、すべてのマウントポイントを結合するディレクトリの作成をお勧めします。たとえば、ディレクトリ~/mountsを作成し、そこに、各種のファイルシステムのサブディレクトリを挿入します。

## 14.2 NTFSパーティションのマウント

NTFS(*New Technology File System*)は、一部のWindowsバージョン(Windows NT、Windows 2000、Windows XP、Windows Vista)のデフォルトファイルシステム

です。NTFSはFATファイルシステムに置き換わります。通常ユーザとしてWindowsパーティションをマウントするには、次の手順に従います。

- 1 rootになって、パッケージntfs-3gをインストールします。
- 2 ディレクトリ/media/windowsを作成します。
- 3 必要なWindowsパーティションを見つけます。YaSTを使用し、パーティションモジュールを起動して、どのパーティションがWindowsに属するかチェックします(ただし、何も変更しないでください)。代替として、rootになって、/sbin/fdisk -lを実行することもできます。パーティションタイプHPFS/NTFSのパーティションを捜します。
- 4 読み書きモードでパーティションをマウントします。ブレースホルダ*DEVICE*を各Windowsパーティションで置き換えます。

```
ntfs-3g /dev/DEVICE /media/windows
```

Windowsパーティションを読み込み専用モードで使いたい場合は、`-o`を追加します。

```
ntfs-3g /dev/DEVICE /media/windows -o ro
```

コマンドntfs-3gは、現在のユーザ(uid)とグループID(gid)を使用して、所定のデバイスをマウントします。書き込みパーミッションを別のユーザに設定したい場合は、コマンドid USERを使用して、uid値とgid値の出力を取得します。次のコードで設定してください。

```
id tux
uid=1000(tux) gid=100(users) Gruppen=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE /media/windows -o uid=1000,gid=100
```

追加オプションをマニュアルページで見つけます。

リソースをアンマウントするには、次のコードを使用します。

```
fusermount -u /media/windows
```



## 14.3 SSHFSによるリモートファイルシステムのマウント

安全なシェルネットワークプロトコルSSHを使用すると、安全なチャネルを使用して2つのコンピュータ間でデータを交換できます。FUSEによってSSH接続を確立するには、次の手順に従います。

- 1 パッケージsshfsをインストールします。
- 2 リモートコンピュータへのアクセス元とするディレクトリを作成します。これには、`~/mounts/HOST`を使用することをお勧めします。`HOST`はリモートコンピュータの名前で置き換えます。
- 3 リモートファイルシステムをマウントします。

```
sshfs USER:HOST ~/mounts/HOST
```

`USER`と`HOST`を該当する値でそれぞれ置き換えます。

- 4 リモートコンピュータ用のパスワードを入力します。

## 14.4 ISOファイルシステムのマウント

ISOイメージにアクセスするには、次の手順で、`fuseiso`パッケージでISOイメージをマウントできます。

- 1 パッケージfuseisoをインストールします。
- 2 ディレクトリ`~/mounts/iso`を作成します。
- 3 ISOイメージをマウントします。

```
fuseiso ISO_IMAGE ~/mounts/iso
```

ISOイメージのコンテンツは、読み込み専用で、書き込むことはできません。

## 14.5 利用可能なFUSEプラグイン

FUSEは、プラグインに依存しています。次のテーブルに、よく利用されるプラグインを一覧します。

**表 14.1** 利用可能なFUSEプラグイン

<code>fuseiso</code>	ISO9660ファイルシステムを含むCD-ROMをマウントします。
<code>ntfs-3g</code>	NTFSボリュームをマウントします（読み込み/書き込みサポート付き）。
<code>sshfs</code>	SSHファイル転送プロトコルに基づくファイルシステムクライアント。
<code>wdfs</code>	WebDAVファイルシステムをマウントします。

## 14.6 詳細情報

詳細については、FUSEのホームページ「<http://fuse.sourceforge.net>」を参照してください。

## パート III. モバイルコンピュータ



# Linuxでのモバイルコンピューティング

# 15

モバイルコンピューティングという言葉から連想されるのはラップトップ、PDA、携帯電話、そしてそれらとのデータ交換ではないでしょうか。外付けハードディスク、フラッシュドライブ、デジタルカメラなどのモバイルハードウェアコンポーネントは、ラップトップやデスクトップシステムに接続できます。多くのソフトウェアコンポーネントで、モバイルコンピューティングを想定しており、一部のアプリケーションは、モバイル使用に合わせて特別に作成されています。

## 15.1 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システムとは異なります。これは交換可能性、占有スペース、消費電力などの基準が関係するためです。モバイルハードウェアメーカーは、ラップトップハードウェアを拡張するために使用可能なPCMCIA(Personal Computer Memory Card International Association)、Mini PCI、Mini PCIeなどの標準インタフェースを開発してきました。この標準にはメモリカード、ネットワークインタフェースカード、ISDNおよびモデムカード、そして外部ハードディスクなどが含まれます。

---

### ティップ: SUSE Linux Enterprise DesktopおよびタブレットPC

SUSE Linux Enterprise Desktopはまた、タブレットPCをサポートします。タブレットPCには、タッチパッド/デジタイザが付属しており、マウスとキーボードを使用するのではなく、デジタルペンやさらに指による操作で画面上で直接データを編集できます。タブレットPCは、他のシステムとまったく同じようにインストールおよび設定されます。タブレットPCのインストー

ルおよび設定について詳しくは、[第17章 タブレットPCの使用](#) (207 ページ) を参照してください。

---

## 15.1.1 電源消費量

ラップトップの製造時、消費電力を最適化したシステムコンポーネントを組み込むことで、電源網にアクセスすることなくシステムを快適に使用できるようにしています。電源の管理に関するこうした貢献は少なくともオペレーティングシステムの貢献度と同じくらい重要です。SUSE® Linux Enterprise Desktopはラップトップの電源消費量に影響する様々なメソッドをサポートすることで、バッテリー使用時の操作に数々の効果をあげています。次のリストでは電源消費量節約への貢献度の高い順に各項目を示します。

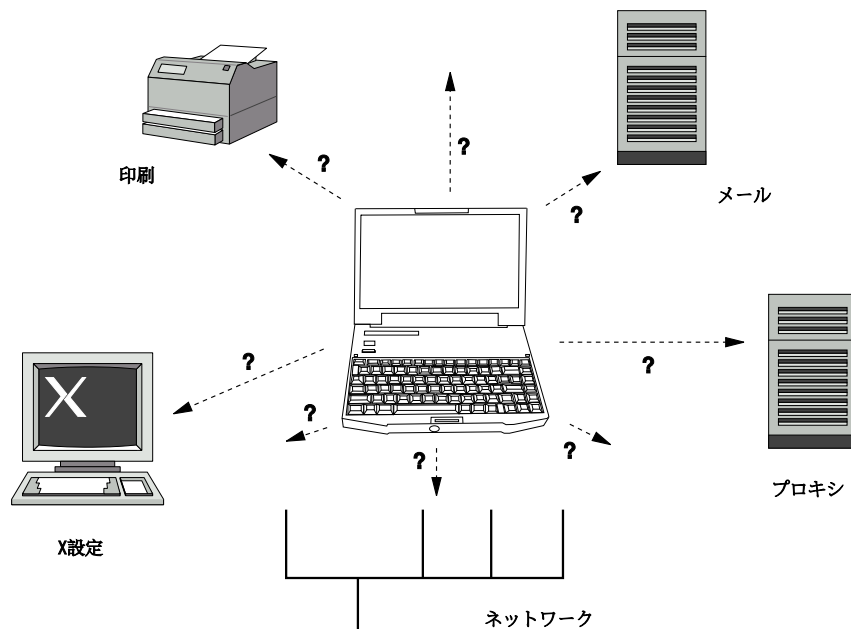
- CPUの速度を落とす
- 休止中にディスプレイの照明を切る.
- ディスプレイの明るさを手動で調節する.
- ホットプラグ対応の使用していないアクセサリを切断する(USB CD-ROM、外付けマウス、使用していないPCMCIAカード、WLANなど)
- アイドル中にはハードウェアディスクをスピンドアウンする.

デスクトップ固有の電源管理に関する詳細については、項「[Controlling Your Desktop's Power Management](#)」(第1章 *Getting Started with the GNOME Desktop*, ↑*GNOME User Guide*)のGNOME電源マネージャの使用方法を参照してください。KDE電源管理アプレットの詳細については、第8章 *Controlling Your Desktop's Power Management with KPowerSave* (↑*KDE User Guide*)を参照してください。

## 15.1.2 操作環境の変化の統合

モバイルコンピューティングに使用する場合、ご使用のシステムを操作環境の変化に順応させる必要があります。環境とそこに存在するクライアントに応じて、多くのサービスを再設定する必要があります。SUSE Linux Enterprise Desktopはこうしたタスクをユーザに代わって実行します。

図 15.1 既存環境でのモバイルコンピュータの統合



スモールホームネットワークとオフィスネットワーク間でラップトップを持ち運びする場合に影響のあるサービスは次のとおりです。

### ネットワーク

IPアドレスの割り振り、名前解決、インターネット接続、およびその他のネットワークへの接続が含まれます。

### 印刷

使用可能なプリンタの現在のデータベース、および使用可能なプリントサーバが、ネットワークに応じて表示されなければなりません。

電子メールとプロキシ

印刷と同様、現在の環境に対応するサーバが表示されなければなりません。

X(グラフィック環境)

ご使用のラップトップが一時的にプロジェクトまたは外付けモニタに接続されている場合、異なるディスプレイ設定が使用可能になっていなければなりません。

SUSE Linux Enterprise Desktopではラップトップを既存の操作環境に統合させる複数の方法を提供しています。

NetworkManager

NetworkManagerは、ラップトップでのモバイルネットワーキング用に特別に作成されています。ネットワーク環境、またはワイヤレスLANおよびイーサネットなど、別のタイプのネットワーク環境間で簡単に自動的に切り替える手段を提供します。NetworkManagerは、ワイヤレスLANでのWEPおよびWPA-PSKの暗号化をサポートします。また、(smpppdにより)ダイヤルアップ接続をサポートします。両方のデスクトップ環境(GNOMEおよびKDE)には、NetworkManagerへのフロントエンドが含まれています。デスクトップアプレットの詳細については、[22.4項「KDE NetworkManager ウィジェットの使用」](#) (310 ページ)および[22.5項「GNOME NetworkManager アプレットの使用」](#) (311 ページ)を参照してください。

表 15.1 NetworkManagerの使用

マイコンピュータ...	NetworkManagerの使用
ラップトップである	対応
別のネットワークに接続される場合がある	対応
ネットワークサービスを提供する(DNSまたはDHCP)	非対応
スタティックIPアドレスのみを使用する	非対応

NetworkManagerがネットワーク設定を扱うのが適切でない場合、YaSTツールを使用してネットワークを設定します。



## SCPM

SCPM(システム設定プロファイル管理)では任意のシステム設定状態をプロファイルと呼ばれる一種の「スナップショット」として格納することができます。プロファイルは異なる状況でも作成できます。プロファイルはシステムが異なる環境(ホームネットワーク、オフィスネットワーク)で作作される場合に便利です。常にプロファイルを切り替えることができます。システムでSCPMを実行するには、パッケージkscpmをインストールし、Profile Chooser KDEアプレットをパネルに追加します。そして、YaSTプロファイル管理モジュールを使用してSCPMを有効にし、rootパスワードを入力しなくてもプロファイルを切り替えることができるようにユーザーを設定します。システム再起動後にもプロファイルの変更を持ち越すか、シャットダウン時に破棄するかを決定します。すべてのリソースグループ(つまりネットワークやプリンタなどのサービス)がアクティブであることを確認します。プロファイルチューザーにより開始される

SUMF(SCPM Unified Management Front-End)ツールを使用して、実際のプロファイルの作成手順に進みます。このシステムを使用するすべての異なる設定についてプロファイルを作成します。実行中のシステムでプロファイル間の切り替えは、プロファイルチューザーアプレットにより、またはF3キーを使用してシステムブート時に行うことができます。選択したプロファイルに含まれる新しい環境に対して、SCPMはプロファイルの切り替え時に自動的にシステム設定を調整します。

## SLP

サービスローケーションプロトコル(SLP)は既存のネットワークでのラップトップの接続を容易にします。SLPがなければラップトップの管理者は通常ネットワークで使用可能なサービスに関する詳細な知識が必要になります。SLPはローカルネットワーク上のすべてのクライアントに対し、使用可能な特定のタイプのサービスについてブロードキャストします。SLPをサポートするアプリケーションはSLPとは別に情報を処理し、自動的に設定することが可能です。SLPはシステムのインストールに使用することもできます。これを使用することで適切なインストールソースの検索を行う必要がなくなります。SLPの詳細については、[第20章 ネットワーク上のSLPサービス](#) (297 ページ)を参照してください。

## 15.1.3 ソフトウェアオプション

モバイル用途には、専用ソフトウェアにより対応されるシステムモニタリング(特にバッテリーの充電)、データ同期、周辺機器との無線通信、インターネットなど、さまざまな特別タスク領域が存在します。次のセクションでは、SUSE Linux Enterprise Desktopが各タスクに提供する最も重要なアプリケーションについて説明します。

### システムモニタリング

SUSE Linux Enterprise Desktopでは2種類のKDEシステムモニタリングツールを提供しています。

#### KPowersave

KPowersaveはコントロールパネルで再充電可能なバッテリーの状態を表示するアプレットです。アイコンは電源のタイプを表示するように設計されています。AC電源で作業する場合、小さな電源のアイコンが表示されます。バッテリーで作業する場合は、アイコンがバッテリーに変わります。rootパスワードの入力が要求された後、対応するメニューにより電源管理用のYaSTモジュールが開きます。これにより異なる電源でもシステムの動作を設定することができます。

#### KSysguard

KSysguardは重要なシステムパラメータをすべて、モニタリング環境に集める独立したアプリケーションです。KSysguardはACPI(バッテリー状態)、CPUのロード、ネットワーク、パーティショニング、メモリ使用状況などを監視します。また、すべてのシステムプロセスを監視し、表示することも可能です。また、収集した情報の表示およびフィルタリングをカスタマイズできます。様々なデータページにある異なるシステムパラメータを監視したり、ネットワーク上で別々のマシンにあるデータを同時に収集することも可能です。KSysguardはKDE環境がなくてもマシン上でデーモンとして実行できます。このプログラムについての詳細な情報は、プログラムに組み込まれたヘルプ機能かSUSEヘルプページを参照してください。

GNOMEデスクトップでは、GNOME電源管理設定およびシステムモニタを使用します。

## データの同期化

ネットワークから切断されたモバイルマシンと、オフィスのネットワーク上にあるワークステーションの両方で作業を行う場合、すべての場合で処理したデータを同期しておくことが必要になります。これには電子メールフォルダ、ディレクトリ、個別の各ファイルなど、オフィスでの作業時と同様、オフィス外で作業する場合にも必須となるものが含まれます。両方の場合のソリューションを次に示します。

### 電子メールの同期化

オフィスネットワークで電子メールを保存するためにIMAPアカウントを使用します。これで電子メールは、KMail、Evolution、またはMozilla Thunderbird Mailなどのような切断型IMAP対応電子メールクライアントを使用するワークステーションからアクセスできるようになります。これについては*GNOME User Guide* (↑*GNOME User Guide*)と*KDE User Guide* (↑*KDE User Guide*)に説明があります。送信メッセージで常に同じフォルダを使用するには、電子メールクライアントでの設定が必要になります。また、この機能により、同期プロセスが完了した時点でステータス情報とともにすべてのメッセージが使用可能になります。未送信メールについての信頼できるフィードバックを受信するためには、システム全体で使用するMTA postfixまたはsendmailの代わりに、メッセージ送信用のメールクライアントに実装されたSMTPサーバーを使用します。

### ファイルと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のデータの同期に対応するユーティリティが複数あります。

## 無線通信

ラップトップはケーブルを使用して自宅やオフィスのネットワークに接続すると同様に、他のコンピュータ、周辺機器、携帯電話、PDAなどに無線接続することもできます。Linuxは3種類のタイプの無線通信をサポートします。

### WLAN

これらの無線テクノロジーの中では最大規模で、特にWLANは規模が大きく、ときに物理的に離れているネットワークでの運用に適している唯一のテクノロジーと言えます。個々のマシンを相互に接続して、独立した無線ネットワークを構築することも、インターネットにアクセスすることも可能です。アクセスポイントと呼ばれるデバイスがWLAN対応デバイスの基

地局として機能し、インターネットへの中継点としての役目を果たします。モバイルユーザは、場所や、どのアクセスポイントが最適な接続を提供するかに応じて様々なアクセスポイントを切り替えることができます。WLANユーザは携帯電話網と同様の、特定のアクセス場所にとらわれる必要のない大規模ネットワークを使用できます。WLANの詳細については、「[19.1項「無線LAN」](#) (285 ページ)」を参照してください。

## Bluetooth

Bluetoothはすべての無線テクノロジーに対するブロードキャストアプリケーション周波数を使用します。BluetoothはIrDAのように、コンピュータ(ラップトップ)およびPDAまたは携帯電話間で通信するために使用できます。また視界内に存在する別のコンピュータと接続するために使用することもできます。Bluetoothはまたキーボードやマウスなど無線システムコンポーネントとの接続にも用いられます。ただし、このテクノロジーはリモートシステムをネットワークに接続するほどには至っていません。壁のような物理的な障害物をはさんで行う通信にはWLANテクノロジーが適しています。

## IrDA

IrDAは狭い範囲での無線テクノロジーです。通信を行う両者は相手の見える位置にいてはなりません。壁のような障害物をはさむことはできません。IrDAで利用できるアプリケーションはラップトップと携帯電話間でファイルの転送を行うアプリケーションです。ラップトップから携帯電話までの距離が短い場合はIrDAを使用できます。ファイル受信者への長距離におよぶファイルの転送はモバイルネットワークが送信します。IrDAのもう1つのアプリケーションは、オフィスでの印刷ジョブを無線転送するアプリケーションです。

# 15.1.4 データのセキュリティ

無認証のアクセスに対し、複数の方法でラップトップ上のデータを保護するのが理想的です。実行可能なセキュリティ対策は次の領域になります。

## 盗難からの保護

常にシステムを物理的な盗難から守ることを心がけます。チェーンなどのような様々な防犯ツールが小売店で販売されています。

## 強力な認証

ログインとパスワードによる標準の認証に加えて、生体認証を使用します。SUSE Linux Enterprise Desktopでは、指紋認証がサポートされます。詳

細については、第3章 *Using the Fingerprint Reader* (↑セキュリティガイド)を参照してください。

#### システム上のデータの保護

重要なデータは転送時のみでなく、ハードディスク上に存在する時点でも暗号化すべきです。これは盗難時の安全性確保にも有効な手段です。SUSE Linux Enterprise Desktopでの暗号化パーティションの作成については第12章 *Encrypting Partitions and Files* (↑セキュリティガイド)に記載されています。また、YaSTによりユーザーを追加するときに暗号化されたホームディレクトリを作成する場合があります。

---

#### 重要項目: データの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のイベントの際にもアンマウントされません。それで、これらのパーティション上のデータは、ハードウェアが盗まれた場合、ハードディスクのレジュームを行うことで、誰にでも入手できるようになります。

---

#### ネットワークセキュリティ

データの転送はどのような状況下でも、必ず保護されていなくてはなりません。Linuxおよび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)を参照してください。無線ネットワークについてのセキュリティ対策は第19章 **無線通信** (285 ページ)に記載されています。

## 15.2 モバイルハードウェア

SUSE Linux Enterprise DesktopはFireWire(IEEE 1394)またはUSB経由のモバイルストレージデバイスを自動検出します。モバイルストレージデバイスという用語は、FireWire、USBハードディスク、USBフラッシュドライブ、デジタルカメラのいずれにも適用されます。これらのデバイスは、対応するインタフェースを介してシステムに接続されるとすぐに自動的に検出されて設定されます。GNOMEとKDEのファイルマネージャは、いずれもモバイルハードウェアアイテムを柔軟に処理します。これらのメディアを安全にアンマウントするには、いずれかのファイルマネージャの [安全に取り外す] (KDE) 機能または [Unmount Volume] (GNOME) 機能を使用します。デスクトップを使用したリムーバブルメディアの処理に関する詳細は、*GNOME User Guide*

(↑*GNOME User Guide*)と*KDE User Guide* (↑*KDE User Guide*)を参照してください。

#### 外付けハードディスク(USBおよびFireWire)

システムが外付けハードディスクを正しく認識するとすぐに、外付けハードディスクのアイコンがファイルマネージャに表示されます。アイコンをクリックすると、ドライブの内容が表示されます。ここでフォルダやファイルの作成および編集、削除を実行できます。システムに指定されたハードディスクの名前を変更するには、アイコンを右クリックしたときに開くメニューから、対応するメニューアイテムを選択します。この名前変更はファイルマネージャでの表示に限られています。/mediaにマウントされているデバイスのデスクリプタは、これには影響されません。

#### USBフラッシュドライブ

システムはこれらのデバイスを外付けハードディスクと同じように扱います。同様にファイルマネージャでエントリの名前変更をすることが可能です。

#### デジタルカメラ (USBおよびFireWire)

システムによって識別されたデジタルカメラもまた、ファイルマネージャの概要に外付けドライブのように表示されます。KDEではURL `camera:/` から写真を読み取ったり、アクセスしたりできます。さらに画像はdigiKamまたはf-spotを使用して処理できます。高度な写真処理はGIMPを使用して行います。digiKam、f-spot、GIMPに関する簡単な説明は、第24章 *Managing Your Digital Image Collection* (↑アプリケーションガイド)、第25章 *Managing Your Digital Image Collection with F-Spot* (↑アプリケーションガイド)および第23章 *Manipulating Graphics with The GIMP* (↑アプリケーションガイド)を参照してください。

## 15.3 携帯電話とPDA

デスクトップシステムまたはラップトップはbluetoothまたはIrDAを介して携帯電話と通信できます。一部のモデルで両方のプロトコルをサポートしていますが、どちらか一方のみしかサポートしていないものもあります。これら2つのプロトコルの使用可能エリア、およびそれぞれの拡張マニュアルは無線通信項(189ページ)ですでに説明しました。携帯電話側のこれらのプロトコルの設定はそれぞれのマニュアルに記載されています。

Plam社製のハンドヘルドデバイスを用いた同期のサポートはEvolutionおよびKontaktにすでに組み込まれています。どちらの場合もデバイスとの初期接続はウィザードを利用して簡単に実行できます。PalmPilotsのサポートを設定し終えたら、同期するデータのタイプ(アドレス、アポイントなど)を決定する必要があります。詳細については、*GNOME User Guide* (↑*GNOME User Guide*)と*KDE User Guide* (↑*KDE User Guide*)を参照してください。

プログラムopensyncでは、より高度な同期ソリューションが利用可能です(異なるデバイスのパッケージlibopensync、msynctool、および対応するプラグインを参照してください)。

## 15.4 詳細情報

モバイルデバイスおよびLinuxに関連するすべてのお問い合わせは<http://tuxmobil.org/>を参照してください。このWebサイトではラップトップのハードウェア、ソフトウェア、PDA、携帯電話、その他のモバイルハードウェアについて複数のセクションで取り扱います。

<http://tuxmobil.org/>では<http://www.linux-on-laptops.com/>、と同様の内容について参照できます。ラップトップおよびハンドヘルドデバイスについての情報はここを参照してください。

SUSEはラップトップを主題としたドイツ語の専用メーリングリストを運営しています。詳細については、<http://lists.opensuse.org/opensuse-mobile-de/>を参照してください。このリストではユーザと開発者がSUSE Linux Enterprise Desktopでのモバイルコンピューティングに関するあらゆるテーマを話題にしています。英語での投稿には回答されますが、アーカイブされた情報のほとんどはドイツ語です。英語の投稿では<http://lists.opensuse.org/opensuse-mobile/>を使用します。

OpenSyncの詳細については、<http://en.opensuse.org/OpenSync>を参照してください。





## 電源管理

電源管理はラップトップコンピュータで特に重要ですが、他のシステムでも役に立ちます。ACPI(Advanced Configuration and Power Interface)は現在のすべてのコンピュータ(ラップトップ、デスクトップ、サーバ)で使用できます。電源管理テクノロジーでは、適切なハードウェアとBIOSルーチンを必要とします。ほとんどのラップトップと多くの新型デスクトップおよびサーバは、これらの必要条件を満たしています。電源の節約や騒音の低減のために、CPU周波数を制御することもできます。

### 16.1 省電力機能

省電力機能はラップトップをモバイル使用する場合に限らず、デスクトップシステムでも重要です。ACPIの主要な機能と、その使用目的は、以下のとおりです。

スタンバイ

サポートされていない。

サスペンド(メモリに保存)

このモードでは、システム状態をすべてRAMに書き込みます。その後、RAMを除くシステム全体がスリープします。この状態では、コンピュータの消費電力が非常に小さくなります。この状態の利点は、ブートやアプリケーションの再起動をせずに、数秒でスリープ前の作業をスリープの時点から再開できることです。この機能はACPI状態S3に対応します。この状態のサポートはまだ開発中なので、ハードウェアに大幅に依存します。

ハイバーネーション(ディスクに保存)

この動作モードでは、システム状態がすべてハードディスクに書き込まれ、システムの電源がオフになります。すべてのアクティブデータを書き込むには、少なくともRAMの大きさのスワップパーティションが必要です。この状態から再開するには、30～90秒かかります。サスペンド前の状態が復元されます。メーカーの中には、このモードを便利なハイブリッド仕様にして提供するものもあります(たとえば、IBM ThinkpadのRediSafe)。対応するACPI状態は、S4です。Linux環境では、suspend to diskはACPIから独立したカーネルルーチンにより実行されます。

バッテリーモニタ

ACPIは、バッテリーをチェックして、充電ステータスに関する情報を提供します。また、システムは、重要な充電ステータスに達した時点で実行するようにアクションを調整します。

自動電源オフ

シャットダウンの後、コンピュータの電源が切れます。これは、バッテリーが空になる直前に自動シャットダウンが行われる場合に特に重要です。

プロセッサ速度の制御

CPUに関して、エネルギーを節約するには3種類の方法があります。周波数と電圧のスケーリング(PowerNow!またはSpeedstepとしても知られています)、スロットル、およびプロセッサのスリープ状態(Cステート)への移行の3種類です。コンピュータの動作モードによっては、この3つの方法を併用することもできます。

## 16.2 ACPI

ACPI (advanced configuration and power interface)は、オペレーティングシステムが個々のハードウェアコンポーネントをセットアップ、および制御できるように設計されています。ACPIは、PnPとAPMの両方の後継となります。また、ACPIはバッテリー、ACアダプタ、温度、ファン、および「close lid」や「battery low」などのシステムイベントに関する情報も提供します。

BIOSには個々のコンポーネントとハードウェアアクセス方法についての情報が入ったテーブルがあります。オペレーティングシステムは、この情報を使用して、割り込みまたはコンポーネントの有効化と無効化などのタスクを実行します。BIOSに格納されているコマンドを、オペレーティングシステムが実行するとき、機能はBIOSの実装方法に依存します。ACPIが検出可能で、

ロードできるテーブルは、`/var/log/boot.msg`にレポートされます。ACPIに生じた問題のトラブルシューティングについては、[16.2.3項「トラブルシューティング」](#) (199 ページ)を参照してください。

## 16.2.1 CPUパフォーマンスの制御

CPUには、3つの省電力方法があります。コンピュータの動作モードによっては、この3つの方法を併用することもできます。また、省電力とは、システムの温度上昇が少なく、ファンが頻繁にアクティブにならないことを意味します。

### 周波数と電圧の調節

**PowerNow!**と**Speedstep**は、AMD社とIntel社が使用するこのテクノロジーの名称です。ただし、このテクノロジーは他のメーカーのプロセッサにも適用されます。CPUのクロック周波数とそのコア電圧が同時に低下し、段階的な省エネよりも大きな効果が得られます。つまり、周波数が半分になると(半分のパフォーマンス)、消費電力も半分以下になります。このテクノロジーは、ACPIには依存していません。CPU周波数制御には、カーネル自体か、またはユーザスペースアプリケーションを使用した2つのアプローチがあります。従って、`/sys/devices/system/cpu/cpu*/cpufreq/`で設定できる各種カーネルガバナがあります。

### ユーザスペースガバナ

ユーザスペースガバナが設定されている場合、カーネルは、通常、デーモンなどのユーザスペースアプリケーションにCPU周波数制御を譲渡します。SUSE Linux Enterprise Desktopでは、このデーモンは **powersaved** パッケージにあります。この実装が使用されるとき、CPU周波数は現在のシステム負荷に応じて調整されます。デフォルトでは、カーネル実装の1つが使用されます。ただし、一部のハードウェア、特定のプロセッサまたはドライバによっては、ユーザスペースの実装が唯一の対策となっています。

### オンデマンドガバナ

これは、動的CPU周波数ポリシーのカーネル実装で、ほとんどのシステムで使用できます。システムの負荷が高くなるとすぐに、CPU周波数が直ちに上がります。システム負荷が低い場合は、周波数が下がります。

### 保守的ガバナ

このガバナは、より保守的なポリシーが使用される以外は、オンデマンド実装と類似しています。CPU周波数が上がる前に、一定の時間、システムの負荷が高くなっている必要があります。

### 省電力ガバナ

CPU周波数を一定にできるだけ低く設定します。

### パフォーマンスガバナ

CPU周波数を一定にできるだけ高く設定します。

### クロック周波数のスロットリング(速度を抑える)

このテクノロジーでは、CPUのクロック信号インパルスが一定割合だけ省略されます。25%のスロットリングでは、4回に1回の割合でインパルスが省略されます。87.5%では、プロセッサにインパルスが届くのは8回に1回だけになります。ただし、省エネ度が減速の割合に比例して増えることはありません。通常、スロットリングが使用されるのは、周波数調節を使用できない場合、または省電力を最大限に使用する場合だけです。このテクノロジーも、特殊なプロセスで制御する必要があります。システムインタフェースは、`/proc/acpi/processor/*/throttling`です。

### プロセッサのスリープ状態への切り替え

オペレーティングシステムは、何も実行することがない場合にプロセッサをスリープ状態にします。この場合、オペレーティングシステムはCPUにhaltコマンドを送ります。C1、C2、およびC3の、3つのオプションがあります。最も経済的な状態C3では、プロセッサキャッシュとメインメモリとの同期も停止します。そのため、この状態を適用できるのは、バスマスタアクティビティを介してメインメモリの内容を変更している他のデバイスが存在しない場合だけです。一部のドライバでは、C3を使用できません。現在の状態は、`/proc/acpi/processor/*/power`に表示されます。

周波数調節とスロットリングが関係するのは、プロセッサがビジー状態の場合だけです。これは、プロセッサがアイドル状態のときには、最も経済的なC状態が常に適用されるためです。CPUがビジー状態の場合、省電力方式として周波数調節を使用することをお勧めします。通常、プロセッサは部分的な負荷でのみ動作します。この場合は、低周波数で実行できます。一般に、カーネルのオンデマンドガバナまたはpowersavedのようなデーモンで制御される動的な周波数調節が最善の方法といえます。低周波数をスタティックに設定す

る方法は、バッテリー使用時やコンピュータを冷却または静止させたい場合に役立ちます。

スロットリングは、システムが高負荷であるにもかかわらずバッテリー使用時間を延長する場合など、最後の手段として使用する必要があります。ただし、スロットリングの割合が高すぎると、スムーズに動作しないシステムがあります。さらに、CPUの負荷が小さければ、CPUスロットリングは無意味です。

## 16.2.2 ACPIツール

総合的に呼べるACPIユーティリティには、バッテリー充電レベルや温度などの情報を表示するだけのツール(`acpi`、`klaptopdaemon`、など)、`/proc/acpi`内の構造へのアクセスを容易にするツール、変化の監視を補助するツール(`akpi`、`acpiw`、`gtkacpiw`)、BIOS内のACPIテーブルを編集するためのツール(パッケージ `pmttools`)などが含まれています。

## 16.2.3 トラブルシューティング

問題を2つに大別できます。1つはカーネルのACPIコードに、未検出のバグが存在する可能性があることです。この場合は、いずれ修正プログラムがダウンロードできるようになります。ただし、問題の多くはBIOSが原因になっています。また、場合によっては、他の広く普及しているオペレーティングシステムにACPIを実装した場合にエラーが起きないように、BIOSにおけるACPIの指定を故意に変えていることがあります。ACPIに実装すると重大なエラーを生じるハードウェアコンポーネントは、ブラックリストに記録され、これらのコンポーネントに対してLinuxカーネルがACPIを使用しないようにします。

問題に遭遇したときに最初に実行することは、BIOSの更新です。コンピュータがまったくブートしない場合、次のブートパラメータは有用です。

`pci=noacpi`

PCIデバイスの設定にACPIを使用しません。

`acpi=ht`

単純なリソース設定のみを実行します。ACPIを他の目的には使用しません。

`acpi=off`

ACPIを無効にします。

---

### 警告: ACPIなしに起動できない場合

一部の新型のコンピュータは(特に、SMPシステムとAMD64システム)、ハードウェアを正しく設定するためにACPIが必要です。これらのコンピュータでACPIを無効にすると、問題が生じます。

---

コンピュータは時折、USBまたはFireWireを介して接続されたハードウェアと混同されることがあります。コンピュータが起動を拒否した場合、必要のないハードウェアのプラグをすべてはずして再試行してください。

システムのブートメッセージを調べてみましょう。そのためには、ブート後にコマンド `dmesg | grep -2i acpi` を使用します(または、問題の原因がACPIだとは限らないので、すべてのメッセージを調べます)。ACPIテーブルの解析時にエラーが発生した場合、重要なDSDTテーブルを改善版と置換することができます。この場合、BIOSで障害のあるDSDTが無視されます。具体的な手順については16.4項「トラブルシューティング」(203ページ)を参照してください。

カーネルの設定には、ACPIデバッグメッセージを有効にするスイッチがあります。ACPIデバッグを有効にした状態でカーネルをコンパイルし、インストールすると、詳細な情報を表示するエラーのエクスポート検索がサポートできるようになります。

BIOSまたはハードウェアに問題がある場合は、常にメーカーに連絡することをお勧めします。特に、Linuxに関するサポートを常に提供していないメーカーには、問題を通知する必要があります。なぜなら、メーカーは、自社の顧客の無視できない数がLinuxを使用しているとわかってやっと、問題を真剣に受け止めるからです。

## 詳細情報

- <http://www.cpqlinux.com/acpi-howto.html> (詳細なACPI HOWTO、DSDTバッチが含まれています)
- <http://www.intel.com/technology/iapc/acpi/index.htm> (Advanced Configuration and Power Interface)

- <http://www.lesswatts.org/projects/acpi/> (SourceforgeによるACPI4Linuxプロジェクト)
- <http://www.poupinou.org/acpi/> (Bruno DucrotによるDSDTパッチ)

## 16.3 ハードディスクの休止

Linux環境では、不要な場合にハードディスクを完全にスリープ状態にしたり、より経済的な静止モードで動作させることができます。最近のラップトップの場合、ハードディスクを手動でオフに切り替える必要はありません。不要な場合は自動的に経済的な動作モードになります。ただし、省電力レベルを最大限にする場合は、次の方法をいくつかテストしてください。

`hdparm`アプリケーションを使用して、各種のハードディスク設定を変更できます。`-y`オプションは、簡単にハードディスクをスタンバイモードに切り替えます。`-Y`を指定すると、スリープ状態になります。`hdparm -S x`を使用すると、一定時間アクティビティがなければハードディスクが回転を停止します。`x`は、次のように置換します:`0`を指定するとこの機構が無効になり、ハードディスクは常時稼働します。`1`から`240`までの値を指定すると、指定した値`x` 5秒が設定値になります。`241`から`251`は、30分の1倍から11倍(30分から5.5時間)に相当します。

ハードディスクの内部省電力オプションは、オプション`-B`で制御できます。`0` (最大限の省電力)～`255` (最大限のスループット)の値を選択します。結果は使用するハードディスクに応じて異なり、査定するのは困難です。ハードディスクを静止状態に近づけるにはオプション`-M`を使用します。`128` (静止)～`254` (高速)の値を選択します。

ハードディスクをスリープにするのは、多くの場合簡単ではありません。Linuxでは、多数のプロセスがハードディスクに書き込むので、ウェイクアップが常に繰り返されています。したがって、ハードディスクに書き込むデータを、Linuxがどのように処理するかを理解することは重要です。はじめに、すべてのデータがRAMにバッファされます。このバッファは、`pdflush`デーモンによって監視されます。データが一定の寿命に達するか、バッファがある程度一杯になると、バッファの内容がハードディスクにフラッシュされます。バッファサイズはダイナミックであり、メモリサイズとシステム負荷に対応して変化します。デフォルトでは、データの完全性を最大まで高めるように、`pdflush`

の間隔が短く設定されています。**pdflush**デーモンはバッファを5秒おきにチェックし、データをハードディスクに書き込みます。次の変数が使用できます。

`/proc/sys/vm/dirty_writeback_centisecs`

**pdflush**スレッドが起動するまでの100分の1秒の遅延を含みます。

`/proc/sys/vm/dirty_expire_centisecs`

ダーティページが次に最新の変更を書き込まれるまでの時間枠を定義します。デフォルト値は3000(つまり30秒)です。

`/proc/sys/vm/dirty_background_ratio`

**pdflush**が書き込みを始めるまでのダーティページの最大割合。デフォルトは5%です。

`/proc/sys/vm/dirty_ratio`

メモリ全体の中でダーティページの割合がこの値を超えると、プロセスは書き込みを続けずに、短時間でダーティバッファを書き込むように強制されます。

---

### 警告: データの完全性に関する障害

**pdflush**デーモンの設定を変更すると、データの完全性が損なわれる可能性があります。

---

これらのプロセスとは別に、**ReiserFS**や**Ext3**などのジャーナリングファイルシステムは、それらが持つメタデータを**pdflush**とは無関係に書き込むので、ハードディスクが回転を停止しなくなります。モバイル機器では、これを避けるために特別なカーネル拡張が開発されています。詳細については、`/usr/src/linux/Documentation/laptop-mode.txt`を参照してください。

もう1つの重要な要因は、アクティブプログラムが動作する方法です。たとえば、優れたエディタは、変更中のファイルを定期的にハードディスクに自動バックアップし、これによってディスクがウェイクアップされます。データの完全性を犠牲にすれば、このような機能を無効にできます。

この接続では、メールデーモン**postfix**が変数**POSTFIX\_LAPTOP**を使用します。この変数を**yes**に設定すると、**postfix**がハードディスクにアクセスする頻度は大幅に減少します。



SUSE Linux Enterprise Desktopでは、これらのテクノロジーは `laptop-mode-tools` で制御されます。

## 16.4 トラブルシューティング

すべてのエラーメッセージおよびアラートはファイル `/var/log/messages` に記録されます。必要な情報が得られない場合、ファイル `/etc/sysconfig/powersave/common` にある、`DEBUG` を使用して `powersave` に関連するメッセージの冗長度を上げます。変数の値を7または15まで増やし、デーモンを再起動します。`/var/log/messages` で利用可能なより詳しいエラーメッセージは、エラーの発見に役立ちます。以下のセクションでは `powersave` およびさまざまなスリープモードで最も頻繁に起こる問題について解説します。

### 16.4.1 ACPIはハードウェアサポートで有効になっていますが、各機能を使用できません。

ACPIで問題が発生した場合は、コマンド `dmesg | grep -i acpi` を使用し、`dmesg` の出力からACPI固有のメッセージを検索します。問題を解決するためにBIOSのアップデートが必要になる場合があります。ラップトップメーカーのホームページにアクセスし、BIOSの更新バージョンを検索してインストールします。メーカーに最新のACPI仕様に準拠していることを確認してください。BIOSの更新後もエラーが継続する場合は、以下の手順に従い、BIOS内で問題が発生しているDSDTテーブルを更新されたDSDTに置き換えます。

- 1 <http://acpi.sourceforge.net/dsdt/index.php> からシステムに適したDSDTをダウンロードします。以下に示すようにファイルを解凍し、コンパイル後ファイル拡張子が `.aml` (ACPI machine language) になっていることを確認します。拡張子が `.aml` の場合はステップ3に進みます。
- 2 ダウンロードしたテーブルのファイル拡張子が `.asl` (ACPI source language) である場合は、`iasl` (`pmttools` パッケージ) でコンパイルします。コマンド `iasl -sa file.asl` を入力してください。

- 3 ファイルDSDT.amlをいずれかのロケーション(/etc/DSDT.amlが推奨されています)にコピーします。/etc/sysconfig/kernelを編集し、DSDTファイルに応じてパスを変更します。mkinitrd (mkinitrdパッケージ)を開始します。カーネルをアンインストールし、mkinitrdを使用してinitrdを作成する場合は常に、変更されたDSDTが組み込まれ、システムブート時にロードされます。

## 16.4.2 CPU周波数調節が機能しません。

カーネルソース(kernel-source)を参照して、ご使用のプロセッサがサポートされているか確認してください。CPU周波数制御を有効にするには特別なカーネルモジュールまたはモジュールオプションが必要になる場合があります。この情報については/usr/src/linux/Documentation/cpu-freq/\*を参照してください。

## 16.4.3 サスペンドとスタンバイが機能しません。

ACPIシステムでは問題のあるDSDTを実装していることにより(BIOS)、サスペンドとスタンバイに関する問題が発生する可能性があります。そのような場合は、BIOSをアップデートしてください。

システムが不具合のあるモジュールをアンロードしようとする、システムは停止するか、またはサスペンドイベントがトリガされません。また、サスペンドに入らない原因となるモジュールをアンロードしない、またはそうしたサービスを停止しない場合、同様の状態に陥る可能性があります。どちらの場合でも、スリープモードに入らない原因となっている障害モジュールを識別してください。ログファイル/var/log/pm-suspend.logには、エラーの内容と場所に関する詳細情報が含まれます。/usr/lib/pm-utils/defaultsのSUSPEND\_MODULES変数を変更し、サスペンドまたはスタンバイがトリガされる前に問題のあるモジュールをアンロードします。

<http://www.opensuse.org/Pm-utils>および<http://www.opensuse.org/S2ram>で、サスペンドを変更してプロセスを再開する方法についての詳細情報を参照してください。

## 16.5 詳細情報

- <http://www.opensuse.org/S2ram>—「How to get Suspend to RAM working」
- <http://www.opensuse.org/Pm-utils>—「How to modify the general suspend framework」



## タブレットPCの使用

SUSE® Linux Enterprise Desktopでは、タブレットPCをサポートします。ここでは、タブレットPCのインストールと設定の方法を学び、デジタルペンで入力できるLinux\* アプリケーションの利便性を理解します。

次のタブレットPCが使用できます。

- シリアルWacomデバイス、たとえばACER TMC30x series、Fujitsu Lifebook T series (T30xx/T40xx/T50xx)、Gateway C-140X/E-295C、HP Compaq TC1100/TC4200/TC4400、2710p/2730p、IBM/Lenovo X41t/X61t、LG LT20、Motion M1200/M1400、OQO 02、Panasonic Toughbook CF-18、Toshiba Portege/Tecra M series、Satellite R15/R20などを含むタブレットPC。
- WacomUSBデバイス、たとえばASUS R1E/R1F、Gateway C-120X/E-155C、HP Pavilion tx2000/tx2100/tx2500 seriesなどを含むタブレットPC。
- FinePointデバイス、たとえばGateway C210X/M280E/CX2724、HP Compaq TC1000などを含むタブレットPC。
- Asus R2H、Clevo TN120R、Fujitsu Siemens Computers P-Series、LG C1、Samsung Q1/Q1-Ultraなどのタッチスクリーンデバイスを含むタブレットPC。

タブレットPCパッケージをインストールしてデジタイザを正しく設定すると、スタイラスと呼ばれるペンによる入力を、以下のアクションとアプリケーションに使用できます。

- KDMまたはGDMへのログイン
- KDEとGNOMEデスクトップの画面のロック解除
- カーソルの画面上の移動、アプリケーションの起動、終了、サイズ変更、ウィンドウの移動、ウィンドウのフォーカス移動、オブジェクトのドラッグ&ドロップなど、その他のポインティングデバイス(マウスやタッチパッドなど)によって起動されるアクション
- X Window Systemのアプリケーションのジェスチャ認識の使用
- GIMPによる描画
- JarnalまたはXournalなどのアプリケーションでのメモ作成またはスケッチ、またはDasherによる大量のテキストの編集

---

**注意:** インストールに必要なキーボードまたはマウス

SUSE Linux Enterprise Desktopのインストール時は、ペンを入力デバイスとして使用できません。タブレットPCに組み込みのキーボードまたはタッチパッドが装備されていない場合、システムをインストールするために外部キーボードまたはマウスをタブレットPCに接続します。

---

## 17.1 タブレットPCパッケージのインストール

タブレットPC用に必要なパッケージは、TabletPCインストールパターンに含まれています。インストール時にTabletPCを選択した場合は、次のパッケージがすでにシステムにインストールされているはずです。

- cellwriter: 文字ベースの手書き入力パネル
- jarnal: Javaベースのメモ作成用アプリケーション

- `wacom-kmp (-default)`: USB Wacomデバイスを含むタブレットPCのカーネルドライバ
- `xournal`: メモ作成およびスケッチ用アプリケーション
- `xstroke`: X Windows System向けジェスチャー認識プログラム
- `xvkbd`: X Window System向け仮想キーボード
- `x11-input-fujitsu`: Fujitsu P-Seriesタブレット向けX入力モジュール
- `x11-input-evtouch`: タッチスクリーンのある一部のタブレットPCのX入力モジュール
- `x11-input-wacom`: Wacomタブレット向けX入力モジュール
- `x11-input-wacom-tools`: Wacomタブレット向け設定、診断、ライブラリ

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、必要なパッケージをコマンドラインから手動でインストールするか、YaST内でTabletPCインストール用パターンを選択します。

## 17.2 タブレットデバイスの設定

[ハードウェア設定] スクリーンでインストールプロセス中に [グラフィックカード] オプションを変更することによって、タブレットPCを設定できます(ここではタッチスクリーンを使用するタブレットPCは含みません)。また、タブレットデバイス(内蔵でも外付けでも)は、インストール後にいつでも設定することができます。

- 1 **SaX2**を起動するには、コマンドラインを使用するか、または**Alt + F2**を押して**sax2**を入力します。
- 2 **Wacom**または**Finepoint**デバイスを使用する場合は、[タブレット] をクリックして、[タブレットのプロパティ] を表示します。

タッチスクリーンのあるタブレットPCを使用する場合は、[タッチスクリーン] をクリックしてください。

- 3 右側のリストから、ベンダとして *[Tablet PCs]*、およびタブレットの名前を選択し、*[このタブレットを有効にする]* をオンにします。

ご使用のコンピュータがリストになく、Wacomデバイスがあることが確かな場合は、*[Wacom ISDV4 Tablet PC (SERIAL)]* または *[Wacom ISDV4 Tablet PC(USB)]* を選択します。

- 4 *[電子ペン]* タブに切り替え、次のオプションが有効になっていることを確認します: *[ペンの追加]*、*[消しゴムの追加]*。タッチスクリーンのあるタブレットPCの場合、*[Add Touch]* もアクティブ化します。

- 5 *[OK]* をクリックし、変更を保存します。

X Window Systemの設定を完了したら、ログアウトでXサーバを再起動します。または、ユーザインタフェースを終了して、仮想コンソールで `init 3 && init 5` を実行します。

タブレットデバイスを設定し、ペンを入力デバイスとして使用できるようになりました(タブレットPCによっては指を使用)。

## 17.3 仮想キーボードの使用

KDEまたはGNOMEデスクトップにログインしたり、画面のロックを解除するには、ユーザ名とパスワードを、通常通りに入力するか、ログインフィールドの下に表示される仮想キーボード `xvkbd` から入力します。キーボードを設定するには、または統合ヘルプにアクセスするには、左下隅の `xvkbd` フィールドをクリックして `xvkbd` メインメニューを開きます。

入力が表示されない場合(または表示されるべきウィンドウに転送されない場合)、`xvkbd` で *[Focus]* キーをクリックしてフォーカスをリダイレクトしてから、キーボードイベントを反映させるウィンドウをクリックします。



## 図 17.1 xvkbd 仮想キーボード



ログイン後にxvkbdを使用するには、メインメニューから起動するか、またはシェルからxvkbdで起動します。

## 17.4 ディスプレイの回転

KRandRTray(KDE)またはgnome-display-properties(GNOME)を使用すると、オンザフライで、ディスプレイの回転やサイズ変更を手動で行うことができます。KRandRTrayおよびgnome-display-propertiesは両方とも、XサーバのRANDR拡張用アプレットです。

メインメニューからKRandRTrayまたはgnome-display-propertiesを起動するか、krandrtrayまたはgnome-display-propertiesを入力して、シェルからアプレットを起動します。アプレットを起動すると、通常、アプレットアイコンがシステムトレイに追加されます。gnome-display-propertiesアイコンがシステムトレイに自動的に表示されない場合は、*[Show Displays in Panel]* が *[Monitor Resolution Settings]* ダイアログでオンになっているかどうかを確認してください。

KRandRTrayでディスプレイを回転するには、アイコンを右クリックし、*[ディスプレイの設定]* を選択します。設定ダイアログから、該当する向きを選択します。

gnome-display-propertiesでディスプレイを回転するには、アイコンを右クリックし、該当する向きを選択します。ディスプレイが新しい方向にすぐに回転します。また、グラフィックタブレットの向きも変更されるので、(ディスプレイの向きが変わっても)ペンの動きを正しく解釈できます。

デスクトップの向きの変更で問題がある場合は、[17.7項「トラブルシューティング」](#) (217 ページ)で詳細を参照してください。

## 17.5 ジェスチャ認識の使用

SUSE Linux Enterprise Desktopには、ジェスチャ認識用にCellWriterおよびxstrokeの両方が含まれます。どちらのアプリケーションでも、ペンまたはその他のポインティングデバイスによるジェスチャを、X Window Systemのアプリケーションへの入力として使用できます。

### 17.5.1 CellWriterの使用

CellWriterを使用すると、セルのグリッドに文字を書き込むことができ、書き込んだ内容は文字ベースで即座に認識されます。書き込みが終了したら、入力を現在フォーカスされているアプリケーションに送信できます。ジェスチャ認識にCellWriterを使用できるようにするには、最初にアプリケーションがユーザの手書き文字を認識できるよう学習させる必要があります。文字を1つずつ特定のキーのマップで覚えさせます(覚えさせていない文字はアクティブ化されないため使用できません)。

#### 手順 17.1 CellWriterのトレーニング

- 1 メインメニューから、またはコマンドラインからcellwriterを入力してCellWriterを起動します。最初の起動時には、CellWriterは自動的にトレーニングモードで起動します。トレーニングモードでは、現在選択されているキーマップの文字セットが示されます。
- 2 各文字のセルに文字に使用するジェスチャを入力します。最初の入力時に背景の色が白に変わり、文字は薄いグレーで表示されます。文字の色が黒に変わるまでジェスチャを複数回繰り返します。トレーニングされていない文字は薄いグレーまたは茶色の背景上に表示されます(デスクトップのカラースキームによります)。
- 3 CellWriterが必要な文字をすべて覚えるまでこの手順を繰り返します。
- 4 CellWriterに別の言語を覚えさせるには、**[Setup]** ボタンをクリックして**[言語]** タブから言語を選択します。**[閉じる]** をクリックして設定ダイアログを閉じます。**[Train]** ボタンをクリックし、**[CellWriter]**

ウインドウの右下にあるドロップダウンボックスからキーマップを選択します。新しいキーのマップについてトレーニングを繰り返します。

- 5 キーマップのトレーニングが終了したら、**[Train]** ボタンをクリックして通常モードに切り替えます。

通常モードでは、CellWriterウィンドウにジェスチャを入力するための空のセルがいくつか表示されます。**[Enter]** ボタンをクリックするまで文字は別のアプリケーションには送信されません。文字を入力として使用する前に修正したり削除できます。認識の確実度が低い文字はハイライト表示されます。入力を修正するには、セルを右クリックすると表示されるコンテキストメニューを使用します。文字を削除するには、ペンの消しゴムを使用するか、マウスで中央をクリックしてセルをクリアします。CellWriterで入力が終わったら、アプリケーションのウィンドウをクリックして入力の送信先となるアプリケーションを定義します。**[Enter]** をクリックしてアプリケーションに入力を送信します。

図 17.2 CellWriterのジェスチャ認識



CellWriterの**[Keys]** ボタンをクリックすると、仮想キーボードが表示され、手書き認識の代わりに使用できます。

CellWriterを非表示にするには、CellWriterウィンドウを閉じます。これでこのアプリケーションはシステムトレイ内にアイコンで表示されます。入力ウィンドウを再表示するには、システムトレイのアイコンをクリックします。

## 17.5.2 Xstrokeの使用

xstrokeでは、ペンまたはその他のポインティングデバイスでのジェスチャを、X Window Systemのアプリケーションへの入力として使用できます。xstrokeアルファベットは、Graffiti\*アプレットに類似のユニストロークアルファベット

です。有効にすると、**xstroke**は入力を現在フォーカスされているウィンドウに送信します。

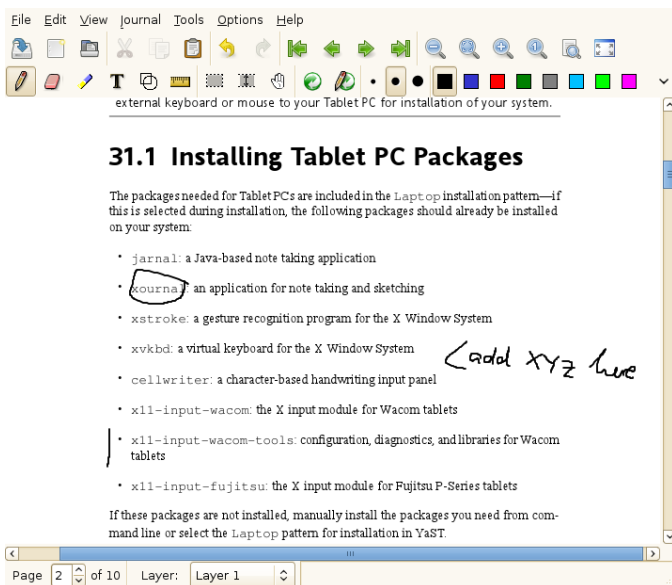
- 1 メインメニューから、またはシェルから**xstroke**を使用して、**xstroke**を起動します。これで、ペンシルアイコンがシステムトレイに追加されます。
- 2 ペンでテキスト入力を作成したいアプリケーション(ターミナルウィンドウ、テキストエディタ、OpenOffice.org Writerなど)を起動します。
- 3 ジェスチャー認識モードを有効にするため、ペンシルアイコンを1回クリックします。
- 4 ペンまたは別のポインティングデバイスで、グラフィックタブレット上で何らかのジェスチャを行います。**xstroke**はジェスチャをキャプチャし、テキストに転送してフォーカスのあるアプリケーションウィンドウに表示します。
- 5 フォーカスを別のウィンドウに移すには、目的のウィンドウをペンでクリックしてしばらくそのままにします(または、デスクトップのコントロールセンターで定義したキーボードショートカットを使用します)。
- 6 ジェスチャー認識モードを無効にするには、ペンシルアイコンをもう一度クリックします。

## 17.6 ペンを使用したメモの作成とスケッチ

ペンで図を作成するには、GIMPなどのプロ向けグラフィックエディタを使用したり、XournalまたはJarnalなどのメモ作成アプリケーションを使用します。XournalとJarnalの両方を使って、ペンでメモや図を作成し、PDFファイルにコメントを付けられます。いくつかのプラットフォームで利用できるJavaベースのアプリケーションとして、Jarnalには基本的なコラボレーション機能もあります。詳細については、<http://www.dklevine.com/general/software/tcl1000/jarnal-net.htm>を参照してください。コンテンツを保存するとき、Jarnalはデータをアーカイブ形式(\*.jaj)にデータを保存し、これにはSVG形式のファイルも含まれます。

JarnalまたはXournalをメインメニューから、またはシェルにjarnalまたはxournalと入力して起動します。XournalでPDFファイルにコメントを付けるには、[ファイル] > [Annotate PDF] を選択して、ファイルシステムからPDFファイルを開きます。ペンまたは別のポインティングデバイスを使用してPDFに注釈を付け、[ファイル] > [Print to PDF] の順に選択して変更内容を保存します。

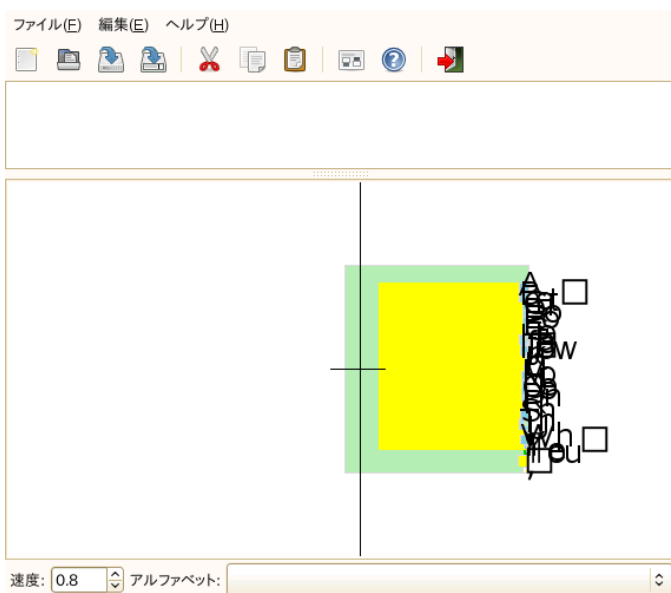
## 図 17.3 XournalによるPDFへの注釈



Dasherも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はない、または利用できない場合に適しています。少し訓練することで、ペンだけで大量のテキストを高速に入力できるようになります(または、視線追跡手段などによるペン以外の入力デバイス)。

メインメニューから、またはシェルからdasherと入力してDasherを起動します。ペンをある方向に動かすと、アプリケーションが右側の文字にズームインし始めます。中央の十字を過ぎた文字から、テキストが作成または予測され、ウィンドウ上部に出力されます。書き込みを停止または開始するには、ディスプレイをペンで1回クリックします。ウィンドウ下部でズーム速度を変更します。

図 17.4 Dasherによるテキストの編集



Dasherの概念は、多くの言語で動作します。詳細はDasherのWebサイトを参照してください。包括的なドキュメント、デモ、トレーニング用テキストがあります。<http://www.inference.phy.cam.ac.uk/dasher/>をご覧ください。

## 17.7 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表示されない

時々、ログイン画面が仮想キーボードに表示されないことがあります。これを解決するには、**Ctrl + Alt + <**を押すか、またはタブレットPCの該当するキー(内蔵キーボードのないスレートモデルを使用している場合)を押して、**X Server**を再起動します。仮想キーボードがまだ表示されない場合は、外部キーボードをスレートモデルに接続し、ハードウェアキーボードを使用してログインします。

## Wacomグラフィックタブレットの向きが変わらない

xrandrコマンドで、シェルからディスプレイの向きを変更できます。  
xrandr--helpと入力すると、使用できるオプションが表示されます。

グラフィックタブレットの向きも同時に変更するには、コマンドを以下のように変更します。

- 通常の方向(0度回転):

```
xrandr --output LVDS ---rotate normal && xsetwacom set "Mouse[7]" Rotate NONE
```

- 90度回転(時計回り、縦):

```
xrandr --output LVDS ---rotate right && xsetwacom set "Mouse[7]" Rotate CW
```

- 180度回転(横):

```
xrandr --output LVDS --rotate inverted && xsetwacom set "Mouse[7]" Rotate HALF
```

- 270度回転(反時計回り、縦):

```
xrandr --output LVDS --rotate left && xsetwacom set "Mouse[7]" Rotate CCW
```

上記のコマンドは、`/etc/X11/xorg.conf`設定ファイルの内容によって異なります。[17.2項「タブレットデバイスの設定」](#) (209 ページ)の説明どおりに、`&sax`でデバイスを設定した場合、コマンドは作成された通りに機能するはずです。`xorg.conf`にあるタブレットスタイラス入力デバイスのIdentifierを手動で変更した場合は、`"Mouse[7]"`を新しいIdentifierで置き換えてください。タッチサポート(指を使ってカーソルを移動できる)の備わったWacomデバイスの場合、タッチデバイスを回転させることも必要です。



## 17.8 詳細情報

ここで説明したアプリケーションの一部には統合オンラインヘルプがありませんが、使用方法および設定についての便利な情報が、インストールしたシステムの `/usr/share/doc/package/packageName` または Web 上にあります。

- Xournalのマニュアルは、<http://xournal.sourceforge.net/manual.html>を参照してください。
- Jarnalのドキュメントは、<http://www.dklevine.com/general/software/tcl000/jarnal.htm#documentation>にあります。
- xstrokeのマニュアルページは、<http://davesource.com/Projects/xstroke/xstroke.txt>にあります。
- Linux上でXを設定する方法は、Wacom Webサイト(<http://linuxwacom.sourceforge.net/index.php/howto/x11>)を参照してください。
- Dasherプロジェクトについては、Webサイト<http://www.inference.phy.cam.ac.uk/dasher/>に詳細な情報があります。
- CellWriterの詳細およびドキュメントについては、<http://risujin.org/cellwriter/>を参照してください。
- gnome-display-propertiesについては、<http://en.opensuse.org/GNOME/Multiscreen>を参照してください。



## パート IV. サービス



## ネットワークの基礎

Linuxには、あらゆるタイプのネットワークストラクチャに統合するために必要なネットワークツールと機能が用意されています。ここでは、一般に使用されるLinuxプロトコルであるTCP/IPについて説明します。このプロトコルが持つさまざまなサービスや特別な機能について述べます。ネットワークカード、モデム、その他のデバイスを使用したネットワークアクセスは、YaSTによって設定できます。手動による環境設定も可能です。この章では、基本的メカニズムと関連のネットワーク設定ファイルのみを解説します。

Linuxおよび他のUnix系オペレーティングシステムは、TCP/IPプロトコルを使用します。これは1つのネットワークプロトコルではなく、さまざまなサービスを提供する複数のネットワークプロトコルのファミリーです。TCP/IPを使用して2台のコンピュータ間でデータをやり取りするために、表 18.1、「TCP/IP プロトコルファミリーを構成する主要なプロトコル」(224ページ)に示した各プロトコルが提供されています。TCP/IPによって結合された世界規模のネットワークを「インターネット」と呼びます。

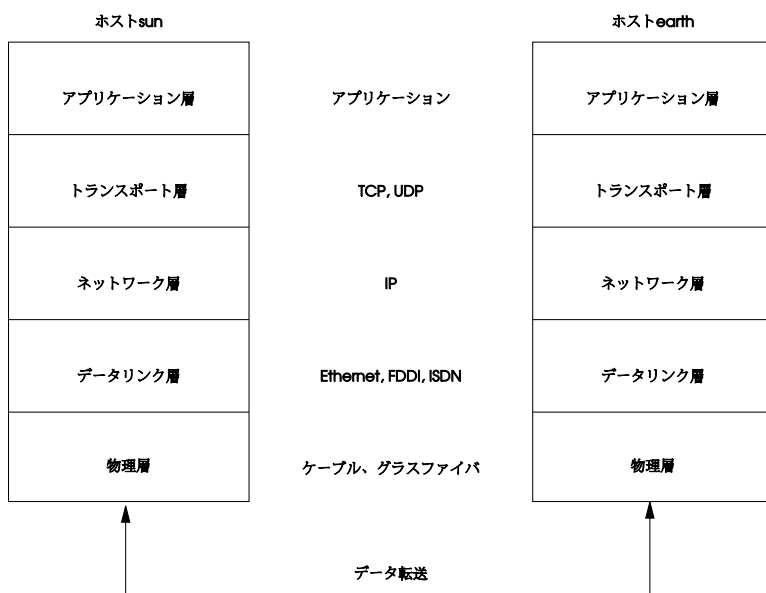
RFCは、*Request for Comments*の略です。RFCは、さまざまなインターネットプロトコルとそれをオペレーティングシステムとそのアプリケーションに実装する手順を定めています。RFC文書ではインターネットプロトコルのセットアップについて説明しています。プロトコルについての知識を広めるには、その種類にかかわらず、適切なRFC文書を参照してください。これらは、<http://www.ietf.org/rfc.html>から入手できます。

表 18.1 TCP/IP プロトコルファミリーを構成する主要なプロトコル

プロトコル	説明
TCP	TCP(Transmission Control Protocol): 接続指向型の安全なプロトコルです。転送データは、まず、アプリケーションによってデータストリームとして送信され、オペレーティングシステムによって適切なフォーマットに変換されます。データは、送信当初のデータストリーム形式で、宛先ホストのアプリケーションに着信します。TCPは、転送中にデータが失われていないか、データの順序に狂いがないかを判別します。データの順序が意味を持つ場合は常にTCP/IPが実装されます。
UDP	UDP(User Datagram Protocol): コネクションレスで安全でないプロトコルです。転送されるデータは、アプリケーションで生成されたパケットの形で送信されます。データが受信側に到着する順序は保証されず、データの損失の可能性があります。UDPはレコード指向のアプリケーションに適しています。TCPよりも遅延時間が小さいことが特徴です。
ICMP	ICMP (Internet Control Message Protocol): 基本的にはエンドユーザ向けのプロトコルではありませんが、エラーレポートを発行し、TCP/IPデータ転送にかかわるマシンの動作を制御できる特別な制御プロトコルです。またICMPには特別なエコーモードがあります。エコーモードは、pingで使用されています。
IGMP	IGMP (Internet Group Management Protocol): このプロトコルは、IPマルチキャストを実装した場合のコンピュータの動作を制御します。

に示したように、データのやり取りはさまざまなレイヤで実行されます。  
**図 18.1. 「TCP/IPの簡易レイヤモデル」** (225 ページ)実際のネットワークレイヤは、IP (インターネットプロトコル)によって実現される確実性のないデータ転送です。IPの上で動作するTCP(転送制御プロトコル)によって、ある程度の確実性のあるデータ転送が保証されます。IPレイヤの下層には、イーサネットなどのハードウェア依存プロトコルがあります。

図 18.1 TCP/IPの簡易レイヤモデル



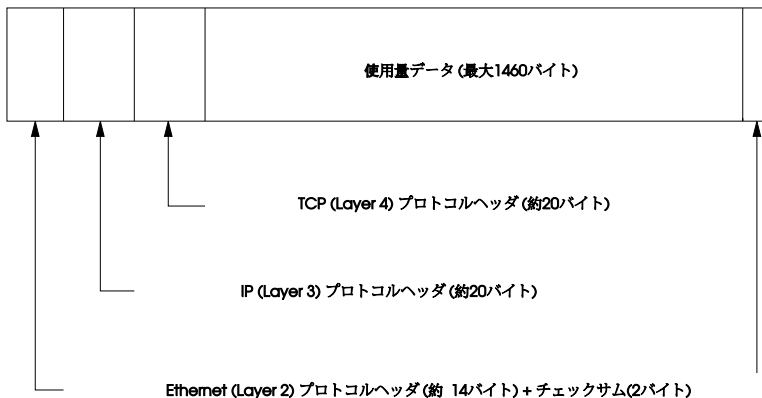
図では、各レイヤに対応する例を1つまたは2つ示しています。レイヤは抽象化レベルに従って並べられています。最下位レイヤは最もハードウェアに近い部分です。一方、最上位レイヤは、ハードウェアがまったく見えないほぼ完全な抽象化になります。各レイヤにはそれぞれの固有の機能があります。各レイヤ固有の機能は、上記の主要プロトコルの説明を読めば大体わかります。データリンクレイヤと物理レイヤは、使用される物理ネットワーク（たとえばイーサネット）を表します。

ほとんどすべてのハードウェアプロトコルは、パケット単位で動作します。転送されるデータは、一度にすべて送信できないので、パケットに分割されます。TCP/IPパケットの最大サイズは約64KBです。しかし、パケットサイズは通常、64KBよりもかなり小さな値になります。これは、ネットワークハードウェアでサポートされているパケットサイズに制限があるからです。イーサネットの最大パケットサイズは、約1500バイトです。イーサネット上に送出されるTCP/IPパケットは、このサイズに制限されます。転送するデータ量が大きくなると、それだけ多くのパケットがオペレーティングシステムによって送信されます。

すべてのレイヤがそれぞれの機能を果たすためには、各レイヤに対応する情報を各データパケットに追加する必要があります。この情報はパケットのヘッ

ダとして追加されます。各レイヤでは、プロトコルヘッダと呼ばれる小さなデータブロックが、作成されたパケットに付加されます。図 18.2. 「TCP/IP イーサネットパケット」 (226 ページ)に、イーサネットケーブル上に送出されるTCP/IPデータパケットの例を示します。誤り検出のためのチェックサムは、パケットの先頭ではなく最後に付加されます。これによりネットワークハードウェアの処理が簡素化されます。

図 18.2 TCP/IP イーサネットパケット



アプリケーションがデータをネットワーク経由で送信すると、データは各レイヤを通過します。これらのレイヤは、物理レイヤを除き、すべてLinuxカーネルに実装されています。各レイヤは、隣接する下位レイヤに渡せるようにデータを処理します。最下位レイヤは、最終的にデータを送信する責任を負います。データを受信したときには、この手順全体が逆の順序で実行されます。重なり合ったたまねぎの皮のように、各レイヤで伝送データからプロトコルヘッダが除去されていきます。最後に、トランスポートレイヤが、着信側のアプリケーションがデータを利用できるように処理します。この方法では、1つのレイヤが直接やり取りを行うのは隣接する上下のレイヤのみです。データが伝送される物理的なネットワークは、100MBit/sのFDDIかもしれませんし、56Kbit/sのモデム回線かもしれませんが、アプリケーションがその違いを意識することはありません。同様に、物理ネットワークは、パケットの形式さえ正しければよく、伝送されるデータの種類を意識することはありません。



# 18.1 IPアドレスとルーティング

ここでは、IPv4ネットワークについてのみ説明しています。IPv4の後継バージョンであるIPv6については、**18.2項「IPv6 —次世代のインターネット」** (230 ページ)を参照してください。

## 18.1.1 IPアドレス

インターネット上のすべてのコンピュータは、一意の32ビットアドレスを持っています。この32ビット(4バイト)は、通常、**例 18.1. 「IPアドレスの表記」** (227 ページ)の2行目に示すような形式で表記されます。

### 例 18.1 IPアドレスの表記

```
IP Address (binary): 11000000 10101000 00000000 00010100
IP Address (decimal): 192. 168. 0. 20
```

10進表記では、4つの各バイトが10進数で表記され、ピリオドで区切られます。IPアドレスは、ホストまたはネットワークインタフェースに割り当てられます。各アドレスは世界で唯一のアドレスであり、重複して使用されることはありません。このルールには例外もありますが、以下の説明には直接関係していません。

IPアドレスにあるピリオドは、階層構造を表しています。1990年代まで、IPアドレスは、各クラスに固定的に分類されていました。しかし、このシステムがあまりに柔軟性に乏しいことがわかったので、今日、そのような分類は行われていません。現在採用されているのは、クラスレスルーティング(CIDR: classless inter domain routing)です。

## 18.1.2 ネットマスクとルーティング

ネットマスクは、サブネットワークのアドレス範囲を定義するために用いられます。2台のホストが同一のサブネットワークに属している場合には、それらは相互に直接連絡できますが、そうでない場合には、サブネットワークとそれ以外の場所との間のトラフィックを処理するゲートウェイのアドレスを必要とします。2つのIPアドレスが同じサブネットワークに属しているかどうかを確認するには、両方のアドレスとネットマスクの「AND」を求めます。結果が同一であれば、両方のIPアドレスは同じローカルネットワークに属し

ています。相違があれば、それらのIPアドレス、そしてそれらに対応するインタフェースが連絡するには、ゲートウェイを通過する必要があります。

ネットマスクの役割を理解するには、例 18.2. 「IPアドレスとネットマスクの論理積(AND)」 (228 ページ)を参照してください。ネットマスクは、そのネットワークにいくつのIPアドレスが属しているかを示す、32ビットの値から成っています。1になっているビットは、IPアドレスのうち、特定のネットワークに属することを示すビットに対応します。0になっているビットは、サブネットワーク内での識別に使われるビットに対応します。これは、1になっているビット数が多いほど、サブネットワークが小さいことを意味します。ネットマスクは常に連続する1のビットから構成されているので、その数だけでネットマスクを指定することができます。例 18.2. 「IPアドレスとネットマスクの論理積(AND)」 (228 ページ)の、24ビットからなる第1のネットワークは、192.168.0.0/24と書くこともできます。

### 例 18.2 IPアドレスとネットマスクの論理積(AND)

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
-----
Result of the link:        11000000 10101000 00000000 00000000
In the decimal system:      192.      168.      0.      0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:        11010101 10111111 00001111 00000000
In the decimal system:      213.      95.      15.      0
```

また、たとえば同じイーサネットケーブルに接続しているすべてのマシンは、普通同じサブネットに属し、直接アクセスできます。サブネットがスイッチまたはブリッジで物理的に分割されていても、これらのホストは直接アクセス可能です。

ローカルサブネットの外部のIPアドレスには、ターゲットネットワーク用のゲートウェイが設定されている場合にのみ、連絡できます。最も一般的には、外部からのすべてのトラフィックを扱うゲートウェイを1台だけ設置します。ただし、異なるサブネット用に、複数のゲートウェイを設定することも可能です。

ゲートウェイを設定すると、外部からのすべてのIPパケットは適切なゲートウェイに送信されます。このゲートウェイは、パケットを複数のホストを経

由して転送し、それは最終的に宛先ホストに到着します。ただし、途中でTTL (time to live)に達した場合は破棄されます。

表 18.2 特殊なアドレス

アドレスのタイプ	説明
基本ネットワークアドレス	ネットマスクとネットワーク内の任意のアドレスの論理積をとったもの。例 18.2. 「IPアドレスとネットマスクの論理積(AND)」 (228 ページ)のANDをとった結果を参照。このアドレスは、どのホストにも割り当てることができません。
ブロードキャストアドレス	ブロードキャストアドレスは、基本的には「サブネットワーク内のすべてのホストにアクセスする」ためのアドレスです。」このアドレスを生成するには、2進数形式のネットマスクを反転させ、基本ネットワークアドレスと論理和をとります。そのため上記の例では、192.168.0.255になります。このアドレスをホストに割り当て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、各ホストの「ループバックデバイス」に割り当てられます。このアドレスを使用すると、自分のマシンに対して接続を確立できます。

IPアドレスは、世界中で一意でなければならぬので、自分勝手にアドレスを選択して使うことはできません。IPベースのプライベートネットワークをセットアップする場合のために、3つのアドレスドメインが用意されています。これらは、外部のインターネットに直接接続することはできません。インターネット上で転送されることがないからです。このようなアドレスドメインは、RFC 1597で、表 18.3. 「プライベートIPアドレスドメイン」 (229 ページ)に示すとおりに定められています。

表 18.3 プライベートIPアドレスドメイン

ネットワーク/ネットマスク	ドメイン
10.0.0.0/255.0.0.0	10.x.x.x

ネットワーク/ネットマスク	ドメイン
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

## 18.2 IPv6 一次世代のインターネット

WWW(ワールドワイドウェブ)の出現により、ここ10年間でTCP/IP経由で通信を行うコンピュータの数が増大し、インターネットは爆発的に拡大しました。CERN(<http://public.web.cern.ch>)のTim Berners-Leeが1990年にWWWを発明して以来、インターネットホストは、数千から約1億まで増加しました。

前述のように、IPv4のアドレスはわずか32ビットで構成されています。しかも、多くのIPアドレスが失われています。というのは、ネットワークの編成方法のせいで、使われないIPアドレスが無駄に割り当てられてしまうからです。サブネットで利用できるアドレスの数は、 $(2^{\text{ビット数}} - 2)$ で与えられます。たとえば、1つのサブネットワークでは、2、6、または14個のアドレスが使用可能です。たとえば128台のホストをインターネットに接続するには、256個のIPアドレスを持つサブネットワークが必要ですが、そのうち2つのIPアドレスは、サブネットワーク自体を構成するのに必要なブロードキャストアドレスと基本ネットワークアドレスになるので、実際に使用できるのは254個だけです。

現在のIPv4プロトコルでは、アドレスの不足を避けるために、DHCPとNAT(ネットワークアドレス変換)の2つのメカニズムが使用されています。これらの方法をパブリックアドレスとプライベートアドレスを分離するという慣習と組み合わせて使用することで、確かにアドレス不足の問題を緩和することができます。問題は、セットアップが面倒で保守しにくいその環境設定方法にあります。IPv4ネットワークでホストをセットアップするには、ホスト自体のIPアドレス、サブネットマスク、ゲートウェイアドレス、そして場合によってはネームサーバアドレスなど、相当数のアドレス項目が必要になります。管理者は、これらをすべて自分で設定しなければなりません。これらのアドレスをどこかから取得することはできません。

IPv6では、アドレス不足と複雑な環境設定方法はもはや過去のもので。ここでは、IPv6がもたらした進歩と恩恵について説明し、古いプロトコルから新しいプロトコルへの移行について述べます。

## 18.2.1 長所

この新しいプロトコルがもたらした最大かつ最もわかりやすい進歩は、利用可能なアドレス空間の飛躍的な増加です。IPv6アドレスは、従来の32ビットではなく、128ビットで構成されています。これにより、2の128乗、つまり、約 $3.4 \times 10^{38}$ 個のIPアドレスが得られます。

しかしながら、IPv6アドレスがその先行プロトコルと異なるのはアドレス長だけではありません。IPv6アドレスは内部構造も異なっており、それが属するシステムやネットワークに関してより具体的な情報を有しています。詳細については、[18.2.2項「アドレスのタイプと構造」](#) (232 ページ)を参照してください。

以下に、この新しいプロトコルの利点をいくつか紹介します。

### 自動環境設定機能

IPv6を使用すると、ネットワークが「プラグアンドプレイ」対応になります。つまり、新しくシステムをセットアップすると、手動で環境設定しなくても、(ローカル)ネットワークに統合されます。新しいホストは自動環境設定メカニズムを使用して、ネイバーディスカバリ (ND) と呼ばれるプロトコルにより、近隣のルータから得られる情報を元に自身のアドレスを生成します。この方法は、管理者の介入が不要だけでなく、サアドレス割り当てを1台のサーバで一元的に管理する必要もありません。これもIPv4より優れている点の1つです。IPv4では、自動アドレス割り当てを行うために、DHCPサーバを実行する必要があります。

### モバイル性

IPv6を使用すると、複数のアドレスを1つのネットワークインタフェースに同時に割り当てることができます。これにより、ユーザは複数ネットワークに簡単にアクセスできます。このことは、携帯電話会社が提供する国際ローミングサービスにたとえられます。携帯電話を海外に持って行った場合、現地会社のサービス提供エリアに入ると自動的に携帯電話はそのサービスにログインし、同じ番号で普段と同じように電話をかけることができます。

## 安全な通信

IPv4では、ネットワークセキュリティは追加機能です。IPv6にはIPSecが中核的機能の1つとして含まれているので、システムが安全なトンネル経由で通信でき、インターネット上での部外者による通信傍受を防止します。

## 後方互換性

現実的に考えて、インターネット全体を一気にIPv4からIPv6に切り替えるのは不可能です。したがって、両方のプロトコルが、インターネット上だけでなく1つのシステム上でも共存できることが不可欠です。これは、一方ではアドレスの互換性によって(IPv4アドレスは容易にIPv6アドレスに変換できます)、他方ではトンネルの使用によって保証されています。参照先 [18.2.3項「IPv4とIPv6の共存」](#) (237 ページ)。また、システムはデュアルスタックIPテクニックによって、両方のプロトコルを同時にサポートできるので、2つのプロトコルバージョン間に相互干渉のない、完全に分離された2つのネットワークスタックが作成されます。

## マルチキャストによるサービスの詳細なカスタマイズ

IPv4では、いくつかのサービス(SMBなど)が、ローカルネットワークのすべてのホストにパケットをブロードキャストする必要があります。IPv6では、これよりはるかにきめ細かいアプローチが取られ、サーバがマルチキャストという、複数のホストをグループの一部として扱う技術によって、ホストにデータを送信します(これは、すべてのホストにデータを送信するブロードキャストとも、各ホストに個別に送信するユニキャストとも異なります)。どのホストを対象グループに含めるかは、個々のアプリケーションによって異なります。事前定義のグループには、たとえば、すべてのネームサーバを対象とするグループ(全ネームサーバマルチキャストグループ)やすべてのルータを対象とするグループ(全ルータマルチキャストグループ)があります。

## 18.2.2 アドレスのタイプと構造

これまでに述べたように、現在のIPプロトコルには、IPアドレス数が急激に不足し始めているということと、ネットワーク設定とルーティングテーブルの管理がより複雑で煩雑な作業になっているという、2つの重要な問題があります。IPv6では、1つ目の問題を、アドレス空間を拡張することによって解決しています。2番目の問題には、階層的なアドレス構造を導入し、ネットワークアドレスを割り当てる高度なテクニックとマルチホーミング(1つのデバイス

に複数のアドレスを割り当てることによって、複数のネットワークへのアクセスを可能にします)を組み合わせて対応しています。

IPv6を扱う場合は、次の3種類のアドレスについて知っておくと役に立ちます。

#### ユニキャスト

このタイプのアドレスは、1つのネットワークインタフェースだけに関連付けられます。このようなアドレスを持つパケットは、1つの宛先にのみ配信されます。したがって、ユニキャストアドレスは、パケットをローカルネットワークまたはインターネット上の個々のホストに転送する場合に使用します。

#### マルチキャスト

このタイプのアドレスは、ネットワークインタフェースのグループに関連します。このようなアドレスを持つパケットは、そのグループに属するすべての宛先に配信されます。マルチキャストアドレスは、主に、特定のネットワークサービスが、相手を特定のグループに属するホストに絞って通信を行う場合に使用されます。

#### エニーキャスト

このタイプのアドレスは、インタフェースのグループに関連します。このようなアドレスを持つパケットは、基盤となるルーティングプロトコルの原則に従い、送信側に最も近いグループのメンバに配信されます。エニーキャストアドレスは、特定のネットワーク領域で特定のサービスを提供するサーバについて、ホストが情報を得られるようにするために使用します。同じタイプのすべてのサーバは、エニーキャストアドレスが同じになります。ホストがサービスを要求すると、ルーティングプロトコルによって最も近い場所にあるサーバが判断され、そのサーバが応答します。何らかの理由でこのサーバが応答できない場合、プロトコルが自動的に2番目のサーバを選択し、それが失敗した場合は3番目、4番目が選択されます。

IPv6アドレスは、4桁の英数字が入った8つのフィールドで構成され、それぞれのフィールドが16進数表記の16ビットを表します。各フィールドは、コロン(:)で区切られます。各フィールドで先頭の0は省略できますが、数字の間にある0や末尾の0は省略できません。もう1つの規則として、0のバイトが5つ以上連続する場合は、まとめて2つのコロン(::)で表すことができます。ただし、アドレスごとに::は1回しか使用できません。この省略表記の例については、[例 18.3. 「IPv6アドレスの例」](#) (234 ページ)を参照してください。この3行はすべて同じアドレスを表します。

### 例 18.3 IPv6アドレスの例

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                                : 10 : 1000 : 1a4
```

IPv6アドレスの各部の機能は個別に定められています。最初の4バイトはプレフィックスを形成し、アドレスのタイプを指定します。中間部分はアドレスのネットワーク部分ですが、使用しなくてもかまいません。アドレスの最後の4桁はホスト部分です。IPv6でのネットマスクは、アドレスの末尾のスラッシュの後にプレフィックスの長さを指定して定義します。に示すアドレスには、最初の64ビットがアドレスのネットワーク部分を構成する情報、最後の64ビットにホスト部分を構成する情報が入っています。例18.4、「プレフィックスの長さを指定したIPv6アドレス」(234 ページ)言い換えると、64は、ネットマスクに64個の1ビット値が左から埋められていることを意味します。IPv4と同様、IPアドレスとネットマスクのANDをとることにより、ホストが同じサブネットワークにあるかそうでないかを判定します。

### 例 18.4 プレフィックスの長さを指定したIPv6アドレス

```
fe80::10:1000:1a4/64
```

IPv6は、事前に定義された複数タイプのプレフィックスを認識します。に、一部のプレフィックスタイプを示します。表18.4、「IPv6のプレフィックス」(234 ページ)

表 18.4 IPv6のプレフィックス

プレフィックス (16進)	定義
00	IPv4アドレスおよびIPv4 over IPv6互換性アドレス。これらは、IPv4との互換性を保つために使用します。これらを使用した場合でも、IPv6パケットをIPv4パケットに変換できるルータが必要です。いくつかの特殊なアドレス(たとえばループバックデバイスのアドレス)もこのプレフィックスを持ちます。
先頭桁が2または3	集約可能なグローバルユニキャストアドレス。IPv4と同様、インタフェースを割り当てて特定のサブネットワークの一部を構成することができます。現在、2001::/16(実



プレフィクス (16進)	定義
	稼働品質のアドレス空間)と2002::/16 (6to4アドレス空間)の2つのアドレス空間があります。
fe80::/10	リンクローカルアドレス。このプレフィクスを持つアドレスは、ルーティングしてはなりません。したがって、同じサブネットワーク内からのみ到達可能です。
fec0::/10	サイトローカルアドレス。ルーティングはできますが、それが属する組織のネットワーク内に限られます。要するに、IPv6版のプライベートネットワークアドレス空間です(たとえば、10.x.x.x)。
ff	マルチキャストアドレス。

ユニキャストアドレスは、以下の3つの基本構成要素からなります。

#### パブリックトポロジ

最初の部分(前述のいずれかのプレフィクスが含まれる部分)は、パブリックインターネット内でパケットをルーティングするために使用します。ここには、インターネットアクセスを提供する企業または団体に関する情報が入っています。

#### サイトトポロジ

2番目の部分には、パケットの配信先のサブネットワークに関するルーティング情報が入っています。

#### インタフェースID

3番目の部分は、パケットの配信先のインタフェースを示します。これを使用して、MACをアドレスの一部に含めることができます。MACは、世界中で重複がない固定の識別子であり、ハードウェアメカによってデバイスにコーディングされるので、環境設定手順が大幅に簡素化されます。実際には、最初の64アドレスビットが統合されてEUI-64トークンを構成します。このうち、最後の48ビットにはMACアドレス、残りの24ビットにはトークンタイプに関する特別な情報が入ります。これにより、PPPやISDNのインタフェースのようにMACを持たないインタフェースにEUI-64トークンを割り当てられるようになります。

IPv6は、この基本構造の上で、以下の5種類のユニキャストアドレスを区別します。

**:: (未指定)**

このアドレスは、インタフェースが初めて初期化される時、すなわち、アドレスが他の方法で判定できないときに、ホストがそのソースアドレスとして使用します。

**:::1 (ループバック)**

ループバックデバイスのアドレス。

**IPv4互換アドレス**

IPv6アドレスが、IPv4アドレスおよび96個の0ビットからなるプレフィックスで作成されます。このタイプの互換アドレスは、IPv4とIPv6のホストが、純粋なIPv4環境で動作している他のホストと通信するためのトンネリング(18.2.3項「IPv4とIPv6の共存」(237 ページ)を参照)として使用されます。

**IPv6にマッピングされたIPv4アドレス**

このタイプのアドレスは、IPv6表記で純粋なIPv4アドレスを指定します。

**ローカルアドレス**

ローカルで使用するアドレスのタイプには、以下の2種類があります。

**リンクローカル**

リンクローカルこのタイプのアドレスは、ローカルのサブネットワークでのみ使用できます。このタイプの送信元または宛先アドレスを持つパケットをインターネットまたは他のサブネットワークにルーティングしてはなりません。これらのアドレスは、特別なプレフィックス (fe80:::/10)とネットワークカードのインタフェースID、およびヌルバイトからなる中間部分からなります。このタイプのアドレスは、自動環境設定のとき、同じサブネットワークに属する他のホストと通信するために使用されます。

**サイトローカル**

このタイプのアドレスを持つパケットは、他のサブネットワークにはルーティングできますが、それより広いインターネットにはルーティングしてはなりません。つまり、組織自体のネットワークの内側だけで使用するよう制限する必要があります。このようなアドレスはイントラネット用に使用され、IPv4によって定義されているプライベート

トアドレス空間に相当します。これらのアドレスは、特殊なプレフィクス(fec0::/10)とインタフェースID、およびサブネットワークIDを指定する16ビットのフィールドからなります。

IPv6では、各ネットワークインタフェースが複数のIPアドレスを持つことができるといったまったく新しい機能が導入されました。これにより、同じインタフェースで複数のネットワークにアクセスできます。これらのネットワークは、MACと既知のプレフィクスを使用して完全に自動設定できるので、IPv6を有効にするとすぐに、(リンクローカルアドレスを使用して)ローカルネットワーク上のすべてのホストに接続できるようになります。IPアドレスにMACが組み込まれているので、使用されるIPアドレスは世界中で唯一のアドレスになります。アドレスの唯一の可変部分は、ホストが現在動作している実際のネットワークによって、サイトトポロジとパブリックトポロジを指定する部分になります。

複数のネットワークに接続するホストの場合、少なくとも2つのアドレスが必要です。1つはホームアドレスです。ホームアドレスには、インタフェースIDだけでなく、それが通常属するホームネットワークの識別子(および対応するプレフィクス)も含まれています。ホームアドレスは静的アドレスなので、通常は変更されません。しかし、モバイルホスト宛てのパケットは、それがホームネットワーク内にあるかどうかにかかわらず、すべてそのホストに配信できます。これは、IPv6で導入されたステートレス自動環境設定やネイバーディスカバリのようなまったく新しい機能によって実現されました。モバイルホストは、ホームアドレスに加え、ローミング先の外部ネットワークに属するアドレスも取得します。これらはケアオブアドレスと呼ばれます。ホームネットワークには、ホストが対象エリア外をローミングしている間、そのホスト宛てのすべてのパケットを転送する機能があります。IPv6環境において、このタスクは、ホームエージェントによって実行されます。ホームエージェントは、ホームアドレスに届くすべてのパケットを取得してトンネルにリレーします。一方、ケアオブアドレスに届いたパケットは、特別迂回することなく、直接モバイルホストに転送されます。

## 18.2.3 IPv4とIPv6の共存

インターネットに接続されている全ホストをIPv4からIPv6に移行する作業は、段階的に行われます。両方のプロトコルは今後しばらく共存することになります。両方のプロトコルをデュアルスタックで実装すれば、同じシステム上に共存することが保証されます。しかし、それでもなお、IPv6対応のホストがどのようにしてIPv4ホストと通信するか、また多くがIPv4ベースの現行ネッ

トワークでIPv6パケットをどのように伝送するかなど、解決すべき問題が残ります。最善のソリューションは、トンネリングと互換アドレスです(18.2.2項「アドレスのタイプと構造」(232 ページ)を参照)。

ワールドワイドなIPv4ネットワークと隔離されているIPv6ホストは、トンネルを使って通信を行うことができます。IPv6パケットをIPv4パケットにカプセル化すれば、それをIPv4ネットワークに送ることができます。2つのIPv4ホスト間のこのような接続をトンネルと呼びます。これを行うには、パケットにIPv6の宛先アドレス(または対応するプレフィクス)とともに、トンネルの受信側にあるリモートホストのIPv4アドレスも含める必要があります。基本的なトンネルは、ホストの管理者間が合意すれば、手動で設定が可能です。これは、静的トンネリングとも呼ばれます。

ただし、静的トンネルの環境設定とメンテナンスは、あまりに手間がかかるので、多くの場合、日常の通信には向きません。そこで、IPv6は、動的トンネリングを実現する3つの異なる方法を提供しています。

#### 6over4

IPv6パケットが自動的にIPv4パケットとしてカプセル化され、マルチキャスト対応のIPv4ネットワークによって送信されます。IPv6は、ネットワーク全体(インターネット)を巨大なLAN (local area network)だと思い込んで動作することになります。これにより、IPv4トンネルの着信側の端を自動的に判定できます。ただし、この方法は拡張性に欠けているだけではなく、IPマルチキャストがインターネット上で広く普及しているとはいえないという事実も障害となります。したがってこの解決方法を採用できるのは、マルチキャストが利用できる小規模な企業内ネットワークだけです。この方式の仕様は、RFC 2529に規定されています。

#### 6to4

この方式では、IPv6アドレスからIPv4アドレスを自動的に生成することで、隔離されたIPv6ホストがIPv4ネットワーク経由で通信できるようにします。しかし、隔離されたIPv6ホストとインターネットの間の通信に関して、多くの問題が報告されています。この方式は、RFC 3056で規定されています。

#### IPv6トンネルブローカー

この方式は、IPv6ホスト専用のトンネルを提供する特殊なサーバに依存します。この方式は、RFC 3053で規定されています。

## 18.2.4 IPv6の設定

IPv6を設定するには、通常、個々のワークステーションの設定を変更する必要はありません。IPv6は、デフォルトで有効になっています。インストール時にネットワーク設定ステップで、これを無効にすることができます。Network Configuration項 (第4章 *Installation with YaST*, ↑導入ガイド)を参照してください。インストール済みシステムでIPv6を有効または無効にするには、YaSTの [Network Settings] モジュールを使用します。[グローバルオプション] タブで、必要に応じて [IPv6を有効にする] オプションをオン/オフします。IPv6を手作業で有効にするには、rootとしてmodprobe ipv6と入力します。

IPv6の自動環境設定の概念があるため、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、ワークステーション上ではルーティングテーブルの管理を実行しません。ワークステーションは、ルータアダプタイズプロトコルを使用して、実装する必要のあるプレフィクスとゲートウェイをネットワークルータに問い合わせます。IPv6ルータは、radvdプログラムを使用して設定できます。このプログラムは、IPv6アドレスに使用するプレフィクスとルータをワークステーションに通知します。または、zebra/quaggaを使用してアドレスとルーティングの両方を自動設定することもできます。

/etc/sysconfig/networkファイルを使用して各種のトンネルを設定する方法については、ifcfg-tunnel (5)のマニュアルページを参照してください。

## 18.2.5 詳細情報

ここでの概要は、IPv6に関する情報を網羅しているわけではありません。IPv6の詳細については、次のオンラインドキュメントや書籍を参照してください。

<http://www.ipv6.org/>

IPv6のあらゆる情報にここからリンクできます。

<http://www.ipv6day.org>

独自のIPv6ネットワークを開始するには、すべての情報が必要です。

<http://www.ipv6-to-standard.org/>

IPv6対応製品のリスト。

<http://www.bieringer.de/linux/IPv6/>

Linux IPv6-HOWTOと多くの関連トピックへのリンクが用意されています。

#### RFC 2640

IPv6に関する基本的なRFCです。

#### IPv6 Essentials

Silvia Hagenによる*IPv6 Essentials* (ISBN 0-596-00125-8)は、このトピックに関するあらゆる重要な面を扱っている本です。

## 18.3 ネームレゾリューション

DNSはIPアドレスに1つまたは複数のホスト名を割り当てるとともに、ホスト名をIPアドレスに割り当てます。Linuxでは、この変換は通常、`bind`という特別な種類のソフトウェアによって行われます。また、この変換を行うマシンをネームサーバと呼びます。ホスト名は、その名前構成要素がピリオド(.)で区切られた階層システムを構成しています。しかしながら名前の階層構造は、先に述べたIPアドレスの階層構造とは無関係です。

`hostname.domain`形式で書かれた完全な名前(たとえば、`jupiter.example.com`)について検討してみましょう。完全修飾ドメイン名(FQDN: *fully qualified domain name*)と呼ばれるフルネームは、ホスト名とドメイン名(`example.com`)で構成されます。ドメイン名には最上位ドメイン(TLD) (`de`)が含まれます。

TLDの割り当ては、これまでの経緯もあって、非常に複雑になっています。従来から、米国では、3文字のドメイン名が使用されています。他の国では、ISOで制定された2文字の国コードが標準です。これに加えて、2000年には、特定の活動領域を表す、より長いTLDが導入されました(たとえば、`.info`、`.name`、`.museum`)。

インターネットの初期(1990年より前)には、ファイル`/etc/hosts`に、インターネットで利用されるすべてのマシン名を記述していました。しかし、インターネットに接続されるコンピュータ数の急激な増加により、この方法はすぐに現実的でなくなりました。このため、ホスト名を広く分散して保存するための分散データベースが開発されました。このデータベースは、ネーム

サーバと同様、インターネット上のすべてのホストに関するデータがいつでも用意されているわけではなく、他のネームサーバに問い合わせを行います。

この階層の最上位には、複数のルートネームサーバがあります。ルートネームサーバは、**Network Information Center (NIC)**によって運用されており、最上位レベルドメインを管理します。各ルートネームサーバは、特定の最上位ドメインを管理するネームサーバについての情報を持っています。最上位ドメインNICの詳細については、<http://www.internic.net>を参照してください。

DNSには、ホスト名の解決以外の機能もあります。ネームサーバには、特定のドメイン宛の電子メールをどのホストに転送するかも管理しています(メールエクスチェンジャ(MX))。

マシンがIPアドレスを解決するには、少なくとも1台のネームサーバとそのIPアドレスを知っている必要があります。YaSTを使用すれば、このようなネームサーバを簡単に指定できます。モデムを使ったダイヤルアップ接続の場合は、ネームサーバを手動で設定する必要はありません。接続が設定されるときに、ダイヤルアッププロトコルによってネームサーバのアドレスが提供されるからです。

whoisプロトコルは、DNSと密接な関係があります。このプログラムを使用すると、特定のドメインの登録者名をすぐに検索できます。

---

### 注意: MDNSおよび.localドメイン名

.localトップレベルドメインは、リゾルバではリンクローカルドメインとして処理されます。DNS要求は通常のDNS要求ではなく、マルチキャスト要求として送信されます。ネームサーバ構成で.localドメインをすでに使用している場合は、このオプションを/etc/host.confでオフに変更する必要があります。host.confマニュアルページも参照してください。

インストール中にMDNSをオフにするには、nomdns=1をブートパラメータとして使用してください。

マルチキャストDNSの詳細は、<http://www.multicastdns.org>を参照してください。

---

## 18.4 YaSTによるネットワーク接続の設定;

Linuxでは多くのタイプのネットワーク接続がサポートされています。その多くは、異なるデバイス名と、ファイルシステム内の複数の場所に分散した設定ファイルを使用しています。手動によるネットワーク設定のさまざまな面についての詳細は、[18.6項「ネットワークの手動環境設定」](#) (265 ページ)を参照してください。

NetworkManagerがデフォルトでアクティブなSUSE Linux Enterprise Desktop上では、すべてのネットワークカードが設定されます。NetworkManagerがアクティブでない場合は、リンクアップしている(つまり、ネットワークケーブルが接続している)最初のインタフェースだけが自動的に設定されます。インストール済みのシステムには、付加的なハードウェアを設定することができます。以下のセクションでは、SUSE Linux Enterprise Desktopがサポートするすべてのタイプのネットワーク接続について、その設定方法を説明します。

### 18.4.1 YaSTでのネットワークカードの設定;

YaSTで無線/有線ネットワークカードを設定するには、[\[ネットワークデバイス\]](#) > [\[ネットワーク設定\]](#) の順に選択します。モジュールの開始後に、YaSTで4つのタブ([\[グローバルオプション\]](#)、[\[概要\]](#)、[\[ホスト名/DNS\]](#)、および [\[ルーティング\]](#))を持つ [\[ネットワーク設定\]](#) ダイアログが表示されます。

[\[グローバルオプション\]](#) タブでは、NetworkManager、IPv6、一般的なDHCPオプションの使用など、一般的なネットワークオプションを設定できます。詳細については、[グローバルネットワークオプションの設定項](#) (243 ページ)を参照してください。

[\[概要\]](#) タブには、インストールされたネットワークインタフェースと環境設定に関する情報が含まれています。正しく検出されたネットワークカードの名前が表示されます。このダイアログでは、手動で新しいカードを設定し、それらの設定内容を削除または変更できます。自動検出されなかったカードを手動で設定する場合は、[検出されないネットワークカードの設定項](#) (251 ページ)を参照してください。既に設定済みのカードの設定を変更する場合には、[ネットワークカードの設定の変更項](#) (245 ページ)を参照してください。



[*ホスト名/DNS*] タブでは、マシンのホスト名を設定し、使用サーバに名前を付けることができます。詳細については、[ホスト名とDNSの設定項 \(251 ページ\)](#)を参照してください。

[*ルーティング*] タブは、ルーティングの設定で使います。詳細については、[ルーティングの設定項 \(253 ページ\)](#)を参照してください。

### 図 18.3 ネットワーク設定の実行

## グローバルネットワークオプションの設定

YaST ネットワーク設定モジュールの [*グローバルオプション*] タブでは、NetworkManager、IPv6、およびDHCPクライアントオプションの使用など、重要なグローバルネットワークオプションを設定できます。この設定は、すべてのネットワークインタフェースに適用されます。

[*ネットワークのセットアップ方法*] では、ネットワーク接続を管理する方法を選択します。NetworkManagerデスクトップアプレットですべてのインタフェースの接続を管理する場合は、 [*NetworkManagerでユーザを制御*] を選

択します。このオプションは、複数の有線ネットワークおよび無線ネットワーク間の切り替えに適しています。デスクトップ環境(GNOMEまたはKDE)を実行しない場合、またはコンピュータがXenサーバ(仮想システム)であるか、ネットワーク内でDHCPやDNSなどのネットワークサービスを提供する場合は、*[ifupを使用した従来の方法]*を使用します。NetworkManagerを使用する場合は、nm-appletを使用して、ネットワークオプションを設定する必要があります。*[概要]*、*[Hostname/DNS]*、および*[ルーティング]* (これらは*[Network Settings]* モジュールのタブ)が無効になります。NetworkManagerの詳細については、[第22章 NetworkManagerの使用](#) (307 ページ)を参照してください。

IPv6プロトコル設定で、IPv6プロトコルを使用するかどうかを選択します。IPv4とともにIPv6を使用できます。デフォルトでは、IPv6が選択されています。ただし、IPv6プロトコルを使用しないネットワークでは、IPv6プロトコルを無効にした方が応答時間がより短くなる場合があります。IPv6を無効にする場合は、*[IPv6を有効にする]* オプションをオフにします。これにより、IPv6のカーネルモジュールの自動ロードが無効になります。これは、再起動後に適用されます。

*[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を設定します。常にその応答をブロードキャストするようにサーバに要求することをDHCPクライアントに求める場合は、*[ブロードキャスト応答の要求]* をオンにします。この機能は、マシンが異なるネットワーク間を移動する場合に必要なことがあります。DHCPクライアントIDは、単一ネットワーク上の各DHCPクライアントで異なる必要があります。空白のままにした場合は、デフォルトでネットワークインタフェースのハードウェアアドレスになります。ただし、同じネットワークインタフェース、したがって同じハードウェアアドレスを使用して複数の仮想マシンを実行している場合は、ここで自由形式の固有識別子を指定します。

*[送信するホスト名]* では、dhcpcdがDHCPサーバにメッセージを送信するときに、ホスト名オプションフィールドで使われる文字列を指定します。一部のDHCPサーバでは、このホスト名(ダイナミックDNS)に応じて、ネームサーバゾーン(順レコードおよび逆レコード)を更新します。また一部のDHCPサーバでは、クライアントからのDHCPメッセージで、*[送信するホスト名]* オプションフィールドに特定の文字列が含まれることが必要です。現在のホスト名(/etc/HOSTNAMEで定義されたホスト名)を送信する場合は、*[自動]* のままにします。ホスト名を送信しない場合は、このオプションフィールドを空のままにします。DHCPからの情報に従ったデフォルトのルートを変更しない場合は、*[Change Default Route via DHCP]* をオフにします。

## ネットワークカードの設定の変更

ネットワークカードの設定を変更するには、YaSTの [ネットワーク設定]、> [概要] で検出されたカードのリストから目的のカードを選択し、[編集] をクリックします。[Network Card Setup] ダイアログが表示されます。このダイアログの [一般]、[アドレス]、および [ハードウェア] タブを使って、カードの設定を変更します。無線カードの設定については、[19.1.2項「YaSTでの設定;」](#) (289 ページ)を参照してください。

### IPアドレスの設定

[Network Card Setup] ダイアログの [アドレス] タブで、ネットワークカードのIPアドレス、またはそのIPアドレスの決定方法を設定できます。IPv4およびIPv6の両アドレスがサポートされます。ネットワークカードは、[IP アドレスなし] (ボンドデバイスで有用)の場合や、[Statically Assigned IP Address] (IPv4またはIPv6)、またはDHCPおよび(または)Zeroconf経由で割り当てられる [Dynamic Address] を持つ場合もあります。

[Dynamic Address] を使用する場合は、[DHCP Version 4 Only] (IPv4の場合)、[DHCP Version 6 Only] (IPv6の場合)、または [DHCP Both Version 4 and 6] のいずれを使用するかを選択します。

可能であれば、インストール時に利用可能なリンクを持つ最初のネットワークカードがDHCPによる自動アドレス設定を使用するように自動的に設定されます。NetworkManagerがデフォルトでアクティブなSUSE Linux Enterprise Desktop上では、すべてのネットワークカードが設定されます。

DSL回線を使用していてISP(Internet Service Provider)からスタティックIPが割り当てられていない場合も、DHCPを使用する必要があります。DHCPを使用することを選択する場合は、YaSTネットワークカード設定モジュールの [ネットワーク設定] ダイアログにある [グローバルオプション] タブの [DHCPクライアントオプション] で詳細を設定します。常にその応答をブロードキャストするようにサーバにDHCPクライアントが要求するかどうかを [ブロードキャスト応答の要求] で指定します。このオプションは、マシンがネットワーク間を移動するモバイルクライアントである場合に必要になることがあります。さまざまなホストが同じインタフェースを介して通信するようにバーチャルホストがセットアップされている場合は、各ホストの識別に [DHCPクライアントID] が必要になります。

DHCPは、クライアント設定には適していますが、サーバ設定には適していません。静的なIPアドレスを設定するには、以下の手順に従ってください。

- 1 YaSTネットワークカード設定モジュールの **[概要]** タブの検出されたカード一覧から目的のカードを選択し、 **[編集]** をクリックします。
- 2 **[アドレス]** タブで、 **[Statically Assigned IP Address]** を選択します。
- 3 IPアドレスを入力します。IPv4およびIPv6の両アドレスを使用できます。 **[サブネットマスク]** にネットワークマスクを入力します。IPv6アドレスが使用されている場合は、フォーマット/64のプレフィックス長に対する **[サブネットマスク]** を使用します。

オプションで、このアドレスの完全修飾 **[ホスト名]** を入力できます。このホスト名は、 `/etc/hosts` 設定ファイルに書き込まれます。

- 4 **[次へ]** をクリックします。
- 5 環境設定を有効にするには、 **[OK]** をクリックします。

静的アドレスを使用する場合、ネームサーバとデフォルトゲートウェイは、自動的に設定されません。ネームサーバを設定するには、 **ホスト名とDNSの設定項 (251 ページ)** に従って手順を進めます。ゲートウェイを設定するには、 **ルーティングの設定項 (253 ページ)** に従って手順を進めます。

## エイリアスの設定

1台のネットワークデバイスに、複数のIPアドレスを割り当てることをできます。追加するIPアドレスは、エイリアスと呼ばれます。ネットワークカードにエイリアスを設定するには、以下の手順に従ってください。

- 1 YaSTネットワークカード設定モジュールの **[概要]** タブの検出されたカード一覧から目的のカードを選択し、 **[編集]** をクリックします。
- 2 **[アドレス]** 、 **> [追加アドレス]** タブで、 **[追加]** をクリックします。
- 3 **[エイリアス名]** 、 **[IPアドレス]** 、および **[ネットマスク]** に適切な値を入力します。エイリアス名にはインタフェースを含めないでください。

- 4 [OK] をクリックします。
- 5 [次へ] をクリックします。
- 6 環境設定を有効にするには、[OK] をクリックします。

## デバイス名およびUdevルールの変更

ネットワークカードのデバイス名が使用されている場合、ネットワークカードのデバイス名を変更できます。また、ハードウェア(MAC)アドレスまたはバスIDを介してudevによりネットワークカードを識別するかどうかを選択できます。大型のサーバでは、カードのホットスワッピングを容易にするために後者のオプションが適しています。YaSTを使ってこうしたオプションを設定するには、次の手順に従います。

- 1 YaST ネットワーク設定モジュールの [概要] タブの検出されたカード一覧から目的のカードを選択し、[編集] をクリックします。
- 2 [ハードウェア] タブを開きます。現在のデバイス名がUdevルールに表示されます。[変更] をクリックします。
- 3 udevで [MACアドレス] または [バスID] によりカードを識別するかどうかを選択します。カードの現在のMACアドレスおよびバスIDがダイアログに表示されます。
- 4 デバイス名を変更するには、[Change Device Name] オプションをオンにし、名前を編集します。
- 5 [OK] および [次へ] をクリックします。
- 6 環境設定を有効にするには、[OK] をクリックします。

## ネットワークカードカーネルドライバの変更

一部のネットワークカードには、複数のカーネルドライバを使用できます。カードが既に設定されている場合は、YaSTで利用可能で適切なドライバのリストから、使用するカーネルドライバを選択できます。また、カーネルドライバのオプションを指定することもできます。YaSTを使ってこうしたオプションを設定するには、次の手順に従います。

- 1 YaSTネットワークカード設定モジュールの **[概要]** タブの検出されたカード一覧から目的のカードを選択し、 **[編集]** をクリックします。
- 2 **[ハードウェア]** タブを開きます。
- 3 **[モジュール名]** で、使用するカーネルドライバを選択します。選択したドライバのオプションを、 **[オプション]** に `option=value` の形式で入力します。他にもオプションを使用する場合は、スペースで区切る必要があります。
- 4 **[OK]** および **[次へ]** をクリックします。
- 5 環境設定を有効にするには、 **[OK]** をクリックします。

## ネットワークデバイスの有効化

ifupを使った従来の方法を使用している場合、デバイスをブート時、ケーブル接続時、カード検出時、または手動で起動するように設定したり、起動しないように設定することができます。デバイスの起動方法を変更するには、以下の手順に従ってください。

- 1 YaSTで、 **[ネットワークデバイス]** 、 > **[ネットワーク設定]** で検出されたカードの一覧からカードを選択し、 **[編集]** をクリックします。
- 2 **[一般]** タブの **[デバイスの起動]** から、適切な項目を選択します。

システムブート中にデバイスを起動するには、 **[ブート時]** を選択します。 **[ケーブル接続時]** では、インタフェースで物理接続が存在するかどうか監視されます。 **[ホットプラグ時]** では、インタフェースは可能な限り早急に設定されます。これは、 **[ブート時]** オプションに似ていますが、インタフェースがブート時に存在しない場合にエラーが発生しない点のみが異なります。ifupまたはKInternetによりインタフェースを手動で制御する場合は、 **[手動]** を選択します。デバイスを全く起動しない場合は、 **[起動しない]** を選択します。 **[NFSrootオン]** は **[ブート時]** に似ていますが、インタフェースはrcnetwork stopコマンドによりシャットダウンしません。このオプションは、nfsまたはiscsiのルートファイルシステムを使用する場合に選択します。

- 3 **[次へ]** をクリックします。

- 4 環境設定を有効にするには、**[OK]** をクリックします。

通常、システム管理者のみがネットワークインタフェースを有効および無効にできます。いずれかのユーザがKInternetによりこのインタフェースを有効化できるようにする場合は、**[Enable Device Control for Non-root User via Kinternet]** を選択します。

## 最大転送単位サイズの設定

インタフェースの最大転送単位(MTU)を設定できます。MTUでは、最大許容パケットサイズ(バイト)を参照します。MTUが大きいと、帯域幅の効率が高くなります。ただし、パケットが大きくなると、低速なインタフェースの処理がしばらく阻止され、以降のパケットの遅延が増加する場合があります。

- 1 YaSTで、**[ネットワークデバイス]**、> **[ネットワーク設定]** で検出されたカードの一覧からカードを選択し、**[編集]** をクリックします。
- 2 **[一般]** タブの **[Set MTU]** リストから、適切な項目を選択します。
- 3 **[次へ]** をクリックします。
- 4 環境設定を有効にするには、**[OK]** をクリックします。

## ファイアウォールの設定

項「Configuring the Firewall with YaST」(第9章 *Masquerading and Firewalls*, ↑セキュリティガイド)で説明しているような詳細なファイアウォール設定を行わずに、デバイスに基本的なファイアウォールを設定することができます。次の手順に従います。

- 1 YaST **[ネットワークデバイス]** の> **[ネットワーク設定]** モジュールを開きます。**[概要]** タブで、検出されたカードの一覧からカードを選択し、**[編集]** をクリックします。
- 2 **[ネットワーク設定]** ダイアログの **[一般]** タブを表示します。
- 3 インタフェースを割り当てるファイアウォールゾーンを指定します。次のオプションを指定できます。

## Firewall Disabled

このオプションは、ファイアウォールが無効であり、ファイアウォールがまったく実行しない場合にのみ利用可能です。コンピュータが、外部ファイアウォールにより保護されている、より規模の大きいネットワークに接続している場合にのみ、このオプションを使用してください。

## 自動割り当てゾーン

このオプションは、ファイアウォールが有効になっている場合のみ、利用できます。ファイアウォールが実行中であり、インタフェースがファイアウォールゾーンに自動的に割り当てられます。こうしたインタフェースには、anyキーワードを含むゾーンまたは外部ゾーンが使用されます。

## 内部ゾーン(未保護)

ファイアウォールを実行しますが、このインタフェースを保護するルールは使いません。コンピュータが、外部ファイアウォールにより保護されている、より規模の大きいネットワークに接続している場合に、このオプションを使用してください。また、マシンに追加ネットワークインタフェースが存在する場合、内部ネットワークに接続するインタフェースで使用できます。

## 非武装地帯(DMZ)

非武装地帯ゾーンは、内部ネットワークと(悪意のある)インターネットとの中間にあたるゾーンです。このゾーンに割り当てられたホストは、内部ネットワークおよびインターネットからアクセスされますが、ホストから内部ネットワークにアクセスすることはできません。

## 外部ゾーン

このインタフェースでファイアウォールを実行し、(危険な可能性のある)他のネットワークトラフィックからインタフェースを保護します。これはデフォルトの設定です。

**4** [次へ] をクリックします。

**5** 環境設定を有効にするには、[OK] をクリックします。



## 検出されないネットワークカードの設定

カードは適切に検出されない場合があります。このような場合、検出されたカードのリストに、そのカードは表示されません。システムにそのカード用のドライバが間違いなく含まれている場合は、そのようなカードを手動で設定することができます。特殊なネットワークデバイスタイプ(ブリッジ、ボンド、TUN、TAPなど)も設定できます。未検出のネットワークカードまたは特殊なデバイスを設定するには、次の手順に従います。

- 1 YaSTの [ネットワークデバイス] の > [ネットワーク設定]、> [概要] ダイアログで [追加] をクリックします。
- 2 [ハードウェア] ダイアログで、使用可能なオプションからインタフェースの [デバイスの型] と [環境設定名] を設定します。ネットワークカードが、PCMCIAデバイスかUSBデバイスの場合、それぞれのチェックボックスを選択して、[次へ] をクリックしダイアログを終了します。それ以外の方法では、必要に応じて、カードとその [オプション] で使用されるカーネルの [モジュール名] を定義できます。
- 3 [次へ] をクリックします。
- 4 [一般]、[アドレス]、および [ハードウェア] タブで、インタフェースのIPアドレス、デバイス起動方法、ファイアウォールゾーンなどの必要なオプションを設定します。環境設定オプションの詳細については、[ネットワークカードの設定の変更項](#) (245 ページ)を参照してください。
- 5 インタフェースのデバイスタイプとして、[ワイヤレス] を選択した場合は、次のダイアログで無線接続の設定を行います。無線デバイスの設定方法の詳細は、[19.1項「無線LAN」](#) (285 ページ)を参照してください。
- 6 [次へ] をクリックします。
- 7 ネットワーク設定を有効にするには、[OK] をクリックします。

## ホスト名とDNSの設定

有線ネットワークカードが既に利用できる状態で、インストール時にネットワーク設定を変更しなかった場合、コンピュータのホスト名が自動的に生成され、DHCPが有効になります。また、ホストがネットワークに参加するため

に必要なネームサービス情報も自動的に生成されます。ネットワークアドレス設定にDHCPを使用している場合は、ドメインネームサーバのリストは自動的に記入されます。静的設定を利用する場合は、これらの項目を手動で設定してください。

コンピュータ名を変更し、ネームサーバの検索リストを修正するには、以下の手順に従ってください。

- 1 YaST内のネットワークデバイスモジュールの **[ネットワーク設定]**、  
    > **[ホスト名/DNS]** タブに移動します。
- 2 **[ホスト名]** にホスト名を入力し、必要に応じて **[ドメイン名]** にドメイン名を入力します。マシンがメールサーバである場合、ドメインは特に重要です。ホスト名はグローバルであり、すべての設定ネットワークインタフェースに適用されることに注意してください。

IPアドレスを取得するためにDHCPを使用している場合、DHCPによりコンピュータのホスト名が自動的に設定されます。異なるネットワークに接続する場合は、異なるホスト名が割り当てられることがあり、ランタイムにホスト名が変更されるとグラフィックデスクトップが混同される可能性があるので、この機能を無効にした方が良い場合もあります。DHCPを使用したIPアドレスの取得を無効にするには、**[DHCPでホスト名を変更する]** をオフにします。

IPアドレスを取得するためにDHCPを使用している場合、ホスト名はデフォルトで/etc/hostsに書き込まれ、IPアドレス127.0.0.2として解決可能になります。この機能を無効にするには、**[ホスト名を/etc/hostsに書き込む]** をオフにします。ただし、アクティブなネットワークがない場合、ホスト名は解決可能にならないことに注意してください。

- 3 **[Modify DNS Configuration]** では、DNS設定(ネームサーバ、検索リスト、/etc/resolv.confファイルの内容)を変更する方法を選択します。

**[Use Default Policy]** オプションを選択した場合、(DHCPクライアントまたはNetworkManagerから)動的に取得されたデータと、(YaSTまたは設定ファイルで)静的に定義されたデータをマージするnetconfigスクリプトにより設定が処理されます。ほとんどの場合、デフォルトのポリシーで十分です。

[手動でのみ] オプションを選択した場合、netconfigでは/etc/resolv.confファイルを変更できません。ただし、このファイルは手動で編集できます。

[Custom Policy] オプションを選択した場合、マージポリシーを定義する [Custom Policy Rule] 文字列を指定する必要があります。この文字列は、設定の有効なソースとみなされるインタフェース名のカンマで区切られたリストから構成されます。完全なインタフェース名の場合を除いて、複数のインタフェースに一致する基本的なワイルドカードを使用することもできます。たとえばeth\* ppp?は、先頭がethであり、以降にppp0-ppp9を含むすべてのインタフェースが対象になります。/etc/sysconfig/network/configファイルで定義された静的な設定を適用する方法を示す次の2つの特別なポリシー値が存在します。

#### STATIC

静的な設定は、動的な設定とマージされる必要があります。

#### STATIC\_FALLBACK

静的な設定は、ダイナミックコンフィギュレーションが利用可能でない場合に限り使用されます。

詳細については、man 8 netconfigを参照してください。

- 4 [ネームサーバ] および [ドメイン検索] リストに入力します。ネームサーバは、ホスト名ではなく、192.168.1.116などのIPアドレスにより指定する必要があります。[ドメイン検索] タブで指定した名前は、ドメインが指定されていないホスト名の解決のために使用されるドメイン名です。複数のドメイン検索を使用する場合は、カンマまたは空白でドメインを区切ります。

- 5 環境設定を有効にするには、[OK] をクリックします。

## ルーティングの設定

コンピュータを他のコンピュータやネットワークと通信させるには、ネットワークトラフィックが正しい経路を通過するように、ルーティング情報を設定する必要があります。DHCPを使用している場合、この情報は自動的に設定されます。静的アドレスを使用する場合は、このデータを手作業で追加する必要があります。

- 1 YaSTで、 [ネットワーク設定]、 > [ルーティング] に移動します。
- 2 [デフォルトゲートウェイ] のIPアドレスを指定します。デフォルトゲートウェイは、すべての宛先に一致しますが、必要なアドレスに一致する他のエントリが存在する場合は、デフォルトルートの代わりにそのエントリが使用されます。
- 3 [ルーティングテーブル] には、さらに追加エントリを入力できます。  
[宛先] のネットワークIPアドレス、 [ゲートウェイ] のIPアドレス、および [ネットマスク] を入力します。定義されたネットワークにトラフィックがルーティングされる [デバイス] を選択します(マイナス記号はデバイスを表わします)。このいずれかの値を省略する場合は、マイナス記号(-)を使用します。デフォルトゲートウェイをテーブルに入力するには、 [宛先] フィールドをdefaultのままにします。

---

### 注意

追加のデフォルトルートが使用されている場合、より高い優先度を持つルートを決定するためのメトリックオプションを指定できます。メトリックオプションを指定するには、 [オプション] に `- metric` 番号を入力します。最も高いメトリックを持つルートがデフォルトとして使用されます。ネットワークデバイスが切断している場合は、そのルートが削除され、次のルートが使用されます。ただし、現在のカーネルは静的なルーティングでメトリックを使用せず、`multipathd`などのルーティングデーモンのみがメトリックを使用します。

---

- 4 システムがルータである場合は、 [ネットワーク設定] で [IP転送を有効にする] オプションをオンにします。
- 5 環境設定を有効にするには、 [OK] をクリックします。

## 18.4.2 モデム

YaSTコントロールセンターで、 [ネットワークデバイス] > [モデム] の順に選択して、モデム設定にアクセスします。モデムが自動的に検出されなかった場合は、 [モデムデバイス] タブに移動し、手動設定用のダイアログを [追加] のクリックで開きます。 [モデムデバイス] に、モデムの接続先インタフェースを入力します。

## ティップ: CDMAおよびGPRSモデム

YaSTのモデムモジュールを使って、通常のモデムの設定と同様に、サポートするCDMAおよびGPRSモデムを設定します。

### 図 18.4 モデム設定

 **モデムのパラメータ**  
すべてのモデム設定値を入力してください。モデムデバイス にはモデムが接続されているボー... [続き](#)

モデムデバイス (V):

ダイヤルプレフィックス (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)

 ヘルプ

 キャンセル (C)

 次へ (N)

構内交換機(PBX)経由で接続している場合は、ダイヤルプレフィックスの入力が必要な場合があります。通常、このプレフィックスは0(ゼロ)です。PBX付属の指示書で確認してください。また、トーンダイヤル方式とパルスダイヤル方式のどちらを使用するか、スピーカをオンにするかどうか、およびモデムをダイヤルトーンの検出まで待機させるかどうかを選択します。モデムが交換機に接続されている場合、後者のオプションは無効です。

[詳細] で、ボーレートとモデムの初期化文字列を設定します。これらの設定は、モデムが自動検出されなかった場合、またはデータ転送を動作させるために特殊な設定が必要な場合にのみ変更してください。これは、主にISDN

端末アダプタを使用する場合です。[OK] をクリックしてこのダイアログを閉じます。モデムの制御権をroot権限のない通常のユーザに委任するには、

[*Enable Device Control for Non-root User via Kinternet*] を有効にします。このようにすると、管理者権限のないユーザがインタフェースを有効化または無効化できるようになります。[*Dial Prefix Regular Expression*] には、正規表現を指定します。この正規表現とKInternetで設定する[ダイヤルプレフィックス] が一致する必要があります。このフィールドを空のままにした場合、管理者権限のないユーザは[ダイヤルプレフィックス] を変更できません。

次のダイアログで、ISPを選択します。事前定義済みの国内ISPリストから選択するには、[国] を選択します。または、[新規] をクリックしてダイアログを開き、独自ISPのデータを入力します。これには、ダイヤルアップ接続名、ISP名、ISPから提供されるログインとパスワードが含まれます。接続するたびにパスワードを要求させるには、[常にパスワードを要求する] を選択します。

最後のダイアログでは、次のようにその他の接続オプションを指定できます。

#### [必要に応じてダイヤルする]

[ダイヤルオンデマンド] を有効にする場合は、ネームサーバを少なくとも1つ指定します。インターネットに定期的にデータを要求するプログラムが存在するために、インターネット接続が低コストである場合にのみこの機能を使用します。

#### [接続時にDNSを変更する]

このオプションはデフォルトでオンになっていて、インターネットに接続するたびにネームサーバアドレスが更新されます。

#### [自動でDNS情報を取得]

接続後にプロバイダからドメインネームサーバの情報が送信されない場合は、このオプションをオフにしてDNSの情報を手動で入力します。

#### [Automatically Reconnect]

このオプションが有効である場合、障害の後で接続が自動的に再確立されます。

#### [ドライブを無視する]

このオプションは、ダイヤルアップサーバからのプロンプトの検出を無効にします。接続の構成が低速であるか、まったく機能しない場合は、このオプションを試みてください。

#### [外部ファイアウォールインタフェース]

このオプションを選択すると、ファイアウォールが有効になり、インタフェースが外部として設定されます。このようにして、インターネット接続時に外部からの攻撃から保護されます。

#### [アイドルタイムアウト(秒)]

このオプションでは、ネットワークがアイドル状態になってからモデムが自動的に切断されるまでの時間を指定します。

#### [IP Details(IP詳細設定)]

このオプションを選択すると、アドレス設定ダイアログが開きます。ISPからホストにダイナミックIPアドレスが割り当てられていない場合は、

[ダイナミックIPアドレス]を無効にして、ホストのローカルIPアドレスとリモートIPアドレスを入力します。この情報については、ISPにお問い合わせください。[デフォルトルート]は有効なままにし、[OK]を選択してダイアログを閉じます。

[次へ]を選択すると、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。[OK]をクリックしてこのダイアログを閉じます。

## 18.4.3 ISDN

このモジュールは、システムの1つ以上のISDNカードを設定します。YaSTによってISDNカードが検出されなかった場合は、[ISDNデバイス]タブで□

[追加]をクリックして手動で選択してください。複数のインタフェースを設定することも可能ですが、1つのインタフェースに複数のISPを設定することも可能です。以降のダイアログでは、カードが正しく機能するために必要なISDNオプションを設定します。

## 図 18.5 ISDNの設定

**contr0 に関する ISDN のローレベル設定**  
を選択すると、ドライバは起動時に読み込まれます。手動を選択すると、ドライバは rcisdn start コマンド... [続き](#)

**ISDN カードの情報**  
製造元 Abocom/Magitek  
ISDN カード 2BD1

ドライバ (V):  
HiSax\_driver

**ISDN プロトコル**  
☒ Euro-ISDN (EDSSI) (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):  
ダイヤルプレフィックス (D):  
☒ ISDN 記録を開始する (I)

デバイスの有効化 (D):  
起動時

ヘルプ キャンセル (C) 戻る (B) OK (O)

図 18.5. 「ISDNの設定」 (258 ページ)に示すダイアログでは、使用するプロトコルを選択します。デフォルトは、*[Euro-ISDN (EDSSI)]* ですが、旧式または大型の交換機の場合は、*[1TR6]* を選択します。米国では、*[NI1]* を選択します。関連するフィールドで国を選択してください。隣接するフィールドに対応する国コードが表示されます。最後に、必要に応じて *[市外局番]* と *[ダイヤルプレフィックス]* を入力します。すべてのISDNトラフィックをログに記録しない場合は、*[ISDN記録を開始する]* オプションをオフにします。

*[デバイスの起動]* は、ISDNインタフェースの起動方法を定義します。*[ブート時]* を選択すると、システムブート時にISDNドライバが毎回初期化されます。*[Manually]* を選択した場合は、rootとしてrcisdn startコマンドを実行して、ISDNドライバをロードする必要があります。*[On Hotplug]* は、PCMCIAやUSBデバイスに使用します。デバイスを装着したときにドライバがロードされます。これらの設定が完了したら、*[OK]* を選択します。



次のダイアログでは、ISDNカードのインタフェースタイプを指定し、既存のインタフェースにISPを追加します。インタフェースタイプには、SyncPPPまたはRawIPのどちらかを指定できますが、たいていのISPは、SyncPPPモードで運用しています。このモードについては後述します。

## 図 18.6 ISDNインタフェースの設定



[自分の電話番号] に入力する番号は、次の設定によって異なります。

### 電話線引出口に直接接続されたISDNカード

標準のISDN回線では、3つの電話番号を使用できます(MSN(multiple subscriber number)と呼ばれる)。加入者によっては、最大10個までである場合もあります。これらの電話番号の1つをここに入力します。ただし、市外局番は入力しないでください。間違った番号を入力すると、お使いのISDN回線に付与された最初のMSNが、電話交換手によって自動的に使用されます。

### PBX (Private Branch Exchange)に接続されたISDNカード

この場合も、設定方法は設置された装置によって異なります。

1. 小型のPBX (private branch exchanges)ではたいてい、内線通話にEuro-ISDN (EDSS1)プロトコルを使用します。これらの交換機にはS0バスが内蔵されており、交換機に接続された装置に内線番号を付与します。

内線番号の1つをMSNとして使用してください。外線用に付与されたMSNの少なくとも1つは外線用に使用できるはずですが、もし使用できない場合は、1つのゼロを試してください。詳細については、交換機付属のマニュアルを参照してください。

2. ビジネス向けに設計された大型の交換機では通常、外線通話に1TR6プロトコルを使用します。このタイプの交換機に付与されるMSNはEAAZと呼ばれ、通常直通番号に対応しています。Linuxでの設定では、EAAZの最後の数字を入力するだけで十分なはずですが、どうしてもうまくいかない場合は、1から9までの数字をすべて試してみてください。

次の課金単位の直前に接続を切断するようにする場合は、**[ChargeHUP(課金HUP)]**を有効にします。ただし、このオプションはすべてのISPで使用できるわけではないため注意してください。チャンネルバンドル(マルチリンクPPP)を有効にするオプションも用意されています。最後に、使用している回線でファイアウォールを有効にするには、**[外部ファイアウォールインタフェース]**と**[ファイアウォールの再起動]**を選択します。管理者権限のない通常のユーザがインタフェースの有効化と無効化を行えるようにするには、**[Enable Device Control for Non-root User via Kinternet]**を選択します。

**[詳細]**を選択すると、詳細な接続方式を実装するためのダイアログが開きます。ただし、これらの設定は、通常の個人ユーザには不要です。**[OK]**をクリックして**[Details]**ダイアログを閉じます。

次のダイアログでは、IPアドレスを設定します。プロバイダからスタティックなIPアドレスを与えられていない場合は、**[ダイナミックIPアドレス]**を選択します。スタティックなIPアドレスを与えられている場合は、ISPの指示に従って、ホストのローカルIPアドレスとリモートIPアドレスを該当するフィールドに入力します。このインタフェースをインターネットへのデフォルトルートにする必要がある場合は、**[デフォルトルート]**を選択します。各ホストは、デフォルトルートとして設定されたインタフェースを1つだけ持つことができます。**[次へ]**をクリックして次のダイアログに進みます。

次のダイアログでは、国を設定し、ISPを選択できます。リストに登録されているISPは、call-by-callプロバイダだけです。契約しているISPがリストに登録されていない場合は、**[新規]**を選択します。**[プロバイダパラメータ]**ダイアログが開き、契約しているISPの詳細な情報を入力できます。電話番号を入力するときは、各数字の間に空白やカンマを挿入しないように注意してください。最後に、ISPから提供されたログインIDとパスワードを入力します。入力したら、**[次へ]**をクリックします。

スタンドアロンワークステーションで「必要に応じてダイヤルする」を使用するには、ネームサーバ(DNSサーバ)も指定します。ほとんどのISPはダイナミックDNSをサポートしており、接続するたびにISPからネームサーバのIPアドレスが送信されます。ただし、単一ワークステーションの場合は、192.168.22.99のようなプレースホルダアドレスを入力してください。ISPがダイナミックDNSをサポートしていない場合は、ISPから提供されたネームサーバIPアドレスを入力します。必要に応じて、接続タイムアウト、すなわち、ネットワークがアイドル状態になってから接続を自動的に切断するまでの時間(秒)を指定します。「次へ」をクリックすると設定が確定し、YaSTは、設定されたインタフェースの概要を表示します。これらの設定を有効にするには、「OK」を選択します。

## 18.4.4 ケーブルモデム

一部の国では、ケーブルテレビネットワークを介したインターネット接続が広く普及しています。ケーブルテレビ加入者は通常、モデムを貸与されます。このモデムは、ケーブルテレビの引出線とネットワークカード(10Base-TGより対線を使用)に接続して使用します。ケーブルモデムを接続すると、固定IPアドレスが付与されたインターネット専用接続が提供されます。

契約しているISPから、ネットワークカードを設定する際に、「*Dynamic Address*」または「*Statically Assigned IP Address*」のどちらかを選択するように指示があります。最近では、大半のプロバイダがDHCPを使用しています。スタティックなIPアドレスは、多くの場合、特殊なビジネス用アカウントの一部として提供されます。

ケーブルモデムの設定に関する詳細については、[http://en.opensuse.org/SDB:Setting\\_Up\\_an\\_Internet\\_Connection\\_via\\_Cable\\_Modem\\_with\\_SuSE\\_Linux\\_8.0\\_or\\_Higher](http://en.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher)にある、サポートデータベースの記事を参照してください。

## 18.4.5 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」セクションから「DSL」モジュールを選択します。このモジュールは、次のいずれかのプロトコルに基づいてDSLリンクのパラメータを設定する複数のダイアログで構成されます。

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz Cards)
- ポイントツーポイントトンネリングプロトコル(PPTP)—オーストリア

*[DSLの環境設定の概要]* ダイアログの *[DSLデバイス]* タブに、インストール済みのDSLデバイスのリストが表示されます。DSLデバイスの設定を変更するには、リストでデバイスを選択し、*[編集]* をクリックします。*[追加]* をクリックすることで、新しいDSLデバイスを手動で設定できます。

PPPoEまたはPPTPに基づくDSL接続を設定するには、対応するネットワークカードが正しく設定されている必要があります。ネットワークカードをまだ設定していない場合は、はじめに、*[ネットワークカードの設定]* を選択してカードを設定してください(18.4.1項「*YaSTでのネットワークカードの設定;*」(242ページ)参照)。DSLリンクの場合は、IPアドレスが自動的に割り当てられる場合もありますが、その場合でもDHCPは使用されません。そのため、*[Dynamic Address]* オプションを有効にしないでください。その代わり、スタティックなダミーアドレス(192.168.22.1など)をインタフェースに入力します。*[サブネットマスク]* には、「255.255.255.0」を入力します。スタンドアロンのワークステーションを設定する場合は、*[デフォルトゲートウェイ]* を空白のままにします。

---

## ティップ

*[IPアドレス]* と *[サブネットマスク]* の値は単なるブレースホルダーです。これらはネットワークカードを初期化するために必要なだけであって、実際のDSLリンクを表しているわけではありません。

---

最初の *[DSLの環境設定]* ダイアログ(図 18.7.「*DSLの設定*」(263ページ)参照)で、まず、*[PPPモード]* と、DSLモデムが接続される *[イーサネットカード]* を選択します(ほとんどの場合、eth0)。次に、*[Activate Device]* で、ブート時にDSLリンクを確立する必要があるかどうかを指定します。管理者権限のない通常のユーザがインタフェースの有効化と無効化を行えるようにするには、*[Enable Device Control for Non-root User via Kinternet]* を選択します。

次のダイアログでは、国とその国で提供されている多くのISPの1つを選択できます。以降のダイアログの詳細は、ここまでで設定したオプションによって異なるため、簡単に触れるだけにとどめておきます。各オプションの詳細については、各ダイアログのヘルプを参照してください。

## ☒ 18.7 DSLの設定

 **DSL の設定**  
ここでは、DSL 接続に関する最も重要な設定を行います。まずはお使いの環境に合わせて PPP モード ... [続き](#)

**DSL 接続の設定**  
  
PPP モード (M):  
  
**PPP モード依存の設定**  
VPI/VCI (V):  
  
**Ethernet カード (E)**  
不明なデバイス  
不明 - IP アドレスが割り当てられていません。 [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

  
  
☒ ☐ Internet 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](#) [キャンセル \(C\)](#) [戻る \(B\)](#) [次へ \(N\)](#)

スタンドアロンワークステーションで「必要に応じてダイヤルする」を使用するには、ネームサーバ(DNSサーバ)も指定します。ほとんどのISPはダイナミックDNSをサポートしており、接続するたびにISPからネームサーバのIPアドレスが送信されます。ただし、単一ワークステーションの場合は、192.168.22.99のようなプレースホルダアドレスも入力する必要があります。ISPがダイナミックDNSをサポートしていない場合は、ISPのネームサーバIPアドレスを指定してください。

「切断するまでのアイドル時間(秒数)」には、ネットワークがアイドル状態になってからモデムを自動的に切断するまでの時間を指定します。タイムアウト値としては、60秒～300秒が妥当です。「必要に応じてダイヤルする」を無

効にしている場合は、このタイムアウト値をゼロに設定して自動的に接続が切断されないようにしておきます。

T-DSLの設定はDSLの設定とほぼ同じです。プロバイダとして [T-Online] を選択すると、T-DSL設定ダイアログが開きます。このダイアログで、T-DSLに必要な追加情報(ラインID、T-Online番号、ユーザコード、パスワードなど)を指定します。T-DSLに加入すると、プロバイダからこれらの情報がすべて提供されるはずです。

## 18.5 NetworkManager

NetworkManagerは、モバイルワークステーションに理想的なソリューションです。NetworkManagerを使用すると、移動時のネットワーク間の切り替えおよびネットワークインタフェースの設定について心配する必要がなくなります。NetworkManagerは、自動的に既知のWLANネットワークに接続できます。2つ以上の接続がある場合は、速い方に接続します。

ただし、NetworkManagerはすべての場合に適合するソリューションではありません。したがって、依然としてネットワーク接続管理のための伝統的方法(ifup)とNetworkManagerの間で選択を行うことができます。NetworkManagerでネットワーク接続を管理する場合は、[22.2項「NetworkManagerの有効化」](#) (308 ページ)に従ってYaSTネットワーク設定モジュールでNetworkManagerを有効にし、NetworkManagerでネットワーク接続を設定します。ユースケースのリスト、およびNetworkManagerを設定および使用する方法の詳細については、[第22章 NetworkManagerの使用](#) (307 ページ)を参照してください。

次に、ifupとNetworkManagerの相違をいくつか示します。

### root特権

ネットワークセットアップにNetworkManagerを使用する場合、アプレットを使用するデスクトップ環境内からいつでも簡単にネットワーク接続を切り替え、停止または開始できます。NetworkManagerでは、必要なroot権限なしに、ワイヤレスカード接続の変更および設定もできます。この理由から、NetworkManagerは、モバイルワークステーションに理想的なソリューションと言えます。

ifupを使用する従来の設定では、ユーザ管理デバイスのようなユーザの介入があってもなくても、接続を切り替え、停止または開始する方法がいくつか用意されていますが、ネットワークデバイスを変更または設定するの

にroot権限が常に必要とされます。このことは、多くの場合、考えられるすべての接続を事前に設定することができないモバイルコンピューティングでは問題になります。

### ネットワーク接続のタイプ

従来の設定およびNetworkManagerの両方とも、DHCPと静的設定を使用する有線ネットワーク、ダイヤルアップ、ワイヤレスネットワーク(WEP、WPA-PSK、WPA-Enterpriseアクセス)によるネットワーク接続を処理できます。また、VPNを介した接続もサポートしています。

NetworkManagerは、コンピュータが常に最適な接続を使用して接続されるようにします。ネットワークケーブルの接続が誤って切断された場合は、再接続しようとします。また、ワイヤレス接続のリストから信号強度が最高のネットワークを検出し、自動的にそれを使用して接続します。ifupと同じ機能を得るため、多くの設定作業が必要です。

## 18.6 ネットワークの手動環境設定

ネットワークソフトウェアの手動環境設定は、常に最後の手段です。設定には可能な限りYaSTを使用してください。しかし、ネットワークの環境設定に関する背景知識がYaSTでの設定作業に役立つことがあります。

カーネルは、ネットワークカードを検出し、対応するネットワークインタフェースを作成する際に、デバイスディスカバリの順序またはカーネルモジュールのロード順序によって、デバイスに名前を割り当てます。デフォルトのカーネルデバイス名は、非常にシンプルまたは厳しく制御されたハードウェア環境でのみ予測可能です。ランタイム時の追加や削除が可能であるか、またはデバイスの自動設定をサポートするシステムでは、カーネルにより割り当てられるネットワークデバイス名がリブート間で安定しているとは期待できません。

しかし、すべてのシステム設定ツールは、永続的なインタフェース名に依存しています。この問題は、udevによって解決されます。udevの永続的ネットジェネレータ(/etc/udev/rules.d/75-persistent-net-generator.rules)は、ハードウェアをマッチするルール(デフォルトでは、そのハードウェアアドレスを使用)を生成し、ハードウェアに永続的に一意のインタフェースを割り当てます。udevのネットワークインタフェースデータベースは、ファイル/etc/udev/rules.d/70-persistent-net.rulesに保存されます。

このファイルの行ごとに、1つのネットワークインタフェースが記述され、永続名が指定されます。システム管理者は、NAME=""項目を編集することにより、割り当て名を変更できます。永続的ルールも、YaSTで変更できます。

表 18.5. 「手動ネットワーク環境設定用スクリプト」 (266 ページ)に、ネットワークの環境設定関連の最も重要なスクリプトをまとめます。

表 18.5 手動ネットワーク環境設定用スクリプト

コマンド	機能
<code>if{up,down,status}</code>	<code>if*</code> スクリプトは、ネットワークインタフェースの起動と停止や、指定のインタフェースのステータスの表示を行います。詳細は、 <code>ifup</code> のマニュアルページを参照してください。
<code>rcnetwork</code>	<code>rcnetwork</code> スクリプトを使用すると、すべてのネットワークインタフェースまたは特定のネットワークインタフェースだけを起動、停止、または再起動できます。ネットワークインタフェースの停止には <code>rcnetwork stop</code> 、起動には <code>rcnetwork start</code> 、再起動には <code>rcnetwork restart</code> を使用します。1つのインタフェースだけを停止、起動、または再起動したい場合は、コマンドの後にインタフェース名を指定します(たとえば、 <code>rcnetwork restart eth0</code> )。 <code>rcnetwork status</code> コマンドを使用すると、インタフェースの状態、IPアドレス、およびDHCPクライアントが実行中かどうかが表示されます。 <code>rcnetwork stop-all-dhcp-clients</code> または <code>rcnetwork restart-all-dhcp-clients</code> を使用すると、ネットワークインタフェースで実行中のDHCPクライアントを停止または再起動できます。

`udev`および永続デバイス名の詳細については、第12章 *udevを使用した動的カーネルデバイス管理* (147 ページ)を参照してください。



## 18.6.1 環境設定ファイル

ここでは、ネットワークの環境設定ファイルの概要を紹介し、その目的と使用される形式について説明します。

### **/etc/sysconfig/network/ifcfg-\***

これらのファイルには、ネットワークインタフェースの環境設定が含まれています。これには、実行モード、IPアドレスなどが含まれます。指定可能なパラメータについては、ifupのマニュアルページを参照してください。また、一般的設定を1つのインタフェースだけに使用する場合は、dhcp、wireless、およびconfigの各ファイルにあるすべての変数が、ifcfg-\*ファイルで使用されます。

### **/etc/sysconfig/network/{config, dhcp, wireless}**

configファイルには、ifup、ifdown、およびifstatusの動作に関する汎用的な設定が記述されています。また、dhcpにはDHCPの設定が、wirelessには無線LANカードの設定が記述されています。3つの環境設定ファイル内の変数にはコメントが付きます。/etc/sysconfig/network/config内の一部の変数は、ifcfg-\*ファイルでも使用できます。このファイルでは、それらの変数がより高い優先順位で処理されます。/etc/sysconfig/network/ifcfg.templateファイルは、インタフェースごとに指定できる変数を一覧表示します。ただし、/etc/sysconfig/network/configの変数の大半はグローバル変数であり、ifcfg-ファイル内で無効にすることはできません。たとえば、NETWORKMANAGER変数やNETCONFIG\_\*変数はグローバルです。

### **/etc/sysconfig/network/{routes,ifroute-\*}**

TCP/IPパケットの静的ルーティングが設定されています。ホストへのルート、ゲートウェイ経由のホストへのルート、およびネットワークへのルートなど、さまざまなシステムタスクが必要とするすべての静的ルートは、/etc/sysconfig/network/routesファイルに指定できます。個別のルーティングが必要な各インタフェースにタイして、付加環境設定ファイル/etc/sysconfig/network/ifroute-\*を定義します。\*はインタフェース名で読

み替えてください。経路の環境設定ファイルのエントリは次のようになります。

# 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

第1列は、経路の宛先です。この列には、ネットワークまたはホストのIPアドレスが入ります。到達可能なネームサーバの場合は、完全に修飾されたネットワークまたはホスト名が入ります。

第2列は、デフォルトゲートウェイ、すなわちホストまたはネットワークにアクセスする際に経由するゲートウェイです。第3列は、ゲートウェイの背後にあるネットワークまたはホストのネットマスクです。たとえば、ゲートウェイの背後にあるホストのネットマスクは、255.255.255.255になります。

最後の列は、ローカルホスト(ループバック、イーサネット、ISDN、PPP、モデムデバイスなど)に接続されたネットワークのみに関連します。ここには、デバイス名を指定する必要があります。

(オプションの)5番目のコラムには、経路のタイプを指定することができます。必要ではないコラムには、マイナス記号-を記入してください。これは、パーサがコマンドを正しく解釈できるようにするためです。詳細は、「routes(5) マニュアル」 ページを参照してください。

## /etc/resolv.conf

このファイルには、ホストが属するドメインが指定されています(キーワード search)。また、アクセスするネームサーバアドレスのステータスのリストも記述されています(キーワード nameserver)。このファイルでは、複数のドメイン名を指定できます。完全修飾でない名前を解決する場合は、search の各エントリを付加して完全修飾名の生成が試みられます。複数のネームサーバを、nameserver で始まる複数行で指定できます。コメントの先頭には# 記号が付きます。例 18.5. 「/etc/resolv.conf」 (269 ページ)には、/etc/resolv.conf の可能な内容が示されています。

ただし、`/etc/resolv.conf`は、手動では編集しないでください。このファイルは、`netconfig`スクリプトで生成されます。`YaST`を使用せずに静的DNS設定を定義するには、`/etc/sysconfig/network/config`ファイルから該当する変数を手動で編集します。`(NETCONFIG_DNS_STATIC_SEARCHLIST`で、ホスト名のルックアップに使用されるDNSドメイン名のリスト、`NETCONFIG_DNS_STATIC_SERVERS`で、ホスト名のルックアップに使用されるネームサーバのIPアドレスのリスト、`NETCONFIG_DNS_FORWARDER`で、設定の必要のあるDNSフォワーダの名前を、それぞれ定義できます)。`netconfig`でDNS環境設定を無効にするには、`NETCONFIG_DNS_POLICY=''`を設定します。`netconfig`の詳細については、`man 8 netconfig`を参照してください。

### 例 18.5 `/etc/resolv.conf`

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

## `/sbin/netconfig`

`netconfig`は、追加のネットワーク環境設定を管理するモジュール式ツールです。このツールは、事前定義されたポリシーに従って、`dhcp`または`ppp`として自動設定メカニズムにより提供される設定と、静的に定義された設定をマージします。要求された変更は、`netconfig`モジュールの呼び出しによって適用されます。このモジュールは、環境設定ファイルの変更と、サービスまたは同様のアクションの再起動を行います。

`netconfig`は、3つの主要なアクションを認識します。`netconfig modify`コマンドと`netconfig remove`コマンドは、`dhcp`や`ppp`などのデーモンによって使用され、`netconfig`の設定値を提供したり、削除します。ユーザが使用できるのは、`netconfig update`コマンドだけです。

### 変更

`netconfig modify`コマンドは、現在のインタフェースとサービス固有の動的設定を変更し、ネットワーク設定を更新します。`netconfig`は、標準入力からか、または`--lease-file filename`オプションで指定されたファイルから設定を読み込み、システムのリブートまたは次の変更/削除アクションまで、それらの設定を内部的に保存します。同じインタフェー

スとサービスの組み合わせに関する既存設定は、上書きされます。インタフェースは、`-i interface_name`パラメータで指定されます。サービスは、`-s service_name`パラメータで指定されます。

## 削除

`netconfig remove`コマンドは、特定のインタフェースとコマンドの組み合わせに対する変更アクションによる動的設定を削除し、ネットワーク設定を更新します。インタフェースは、`-i interface_name`パラメータで指定されます。サービスは、`-s service_name`パラメータで指定されます。

## update

`netconfig update`コマンドは、現在の設定で、ネットワーク設定を更新します。これは、ポリシーや静的環境設定が変更された場合に便利です。

`netconfig`ポリシーおよび静的環境設定は、手動またはYaSTを使用して、`/etc/sysconfig/network/config`ファイル内で定義します。`dhcp`または`ppp`として、自動設定ツールで提供された動的環境設定は、`netconfig modify`および`netconfig remove`のアクションでこれらのツールによって直接配信されます。`NetworkManager`は、`netconfig modify`および`netconfig remove`アクションも使用します。`NetworkManager`が有効な場合、`netconfig`(ポリシーモード-`auto`)は、`NetworkManager`の設定のみを使用し、従来の`ifup`方式で設定された他のインタフェースからの設定を無視します。`NetworkManager`が設定を提供しない場合は、静的設定がフォールバックとして使用されます。`NetworkManager`と従来の`ifup`方式の混合使用はサポートされません。

`netconfig`の詳細については、`man 8 netconfig`を参照してください。

## /etc/hosts

このファイル(例 18.6. 「`/etc/hosts`」 (271 ページ)を参照)では、`IIP`アドレスがホスト名に割り当てられています。ネームサーバが実装されていない場合は、`IP`接続をセットアップするすべてのホストをここにリストする必要があります。ファイルには、各ホストについて1行を入力し、`IP`アドレス、完全修飾ホスト名、およびホスト名を指定します。`IP`アドレスは、行頭に指定し、各エントリは空白とタブで区切ります。コメントは常に`#`記号の後に記入します。

### 例 18.6 /etc/hosts

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

## /etc/networks

このファイルには、ネットワーク名とネットワークアドレスの対応が記述されています。形式は、ネットワーク名をアドレスの前に指定すること以外は、hostsファイルと同様です。詳細については、[例 18.7. 「/etc/networks」 \(271 ページ\)](#)を参照してください。

### 例 18.7 /etc/networks

```
loopback      127.0.0.0
localnet      192.168.0.0
```

## /etc/host.conf

名前解決(リゾルバライブラリを介したホストおよびネットワーク名の解釈)は、このファイルにより制御されます。このファイルは、**libc4**または**libc5**にリンクされているプログラムについてのみ使用されます。最新の**glibc**プログラムについては、`/etc/nsswitch.conf`の設定を参照してください。パラメータは、その行内で常に独立しています。コメントは#記号の後に記入します。[表 18.6. 「/etc/host.confファイルのパラメータ」 \(271 ページ\)](#)に、利用可能なパラメータを示します。`/etc/host.conf`の例については、[例 18.8. 「/etc/host.conf」 \(272 ページ\)](#)を参照してください。

**表 18.6** /etc/host.confファイルのパラメータ

---

<b>order hosts,bind</b>	名前の解決の際、サービスがアクセスされる順序を指定します。有効な引数は次のとおりです(空白またはカンマで区切ります)。
-------------------------	---

**hosts:** /etc/hostsファイルを検索します。

**bind:** ネームサーバにアクセスします。

**nis:** NISを使用します。

<code>multi on/off</code>	<code>/etc/hosts</code> に指定されているホストが、複数のIPアドレスを持てるかどうかを定義します。
<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spoofingに影響を与えますが、ネットワークの環境設定にはまったく影響を与えません。
<code>trim domainname</code>	ホスト名が解決された後、指定したドメイン名をホスト名から切り離します(ホスト名にドメイン名が含まれている場合)。このオプションは、ローカルドメインにある名前だけが <code>/etc/hosts</code> ファイルに指定されているが、付加されるドメイン名でも認識する必要がある場合に便利です。

---

#### 例 18.8 `/etc/host.conf`

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

## `/etc/nsswitch.conf`

GNU C Library 2.0を導入すると、*Name Service Switch* (NSS)も合わせて導入されます。詳細については、`nsswitch.conf(5)` manページおよび『*The GNU C Library Reference Manual*』を参照してください。

クエリの順序は、ファイル`/etc/nsswitch.conf`で定義します。`nsswitch.conf`の例については、を参照してください。例 18.9. 「`/etc/nsswitch.conf`」(273 ページ)コメントは#記号の後に記入します。この例では、`hosts`データベースのエントリによると、要求がDNS経由で`/etc/hosts (files)`に送信されています。

### 例 18.9 /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で利用できる「データベース」については、表 18.7. 「[/etc/nsswitch.confで利用できるデータベース](#)」 (273 ページ)を参照してください。それらに加えて、automount、bootparams、netmasks、およびpublickeyが近い将来導入される予定です。NSSデータベースの環境設定オプションについては、表 18.8. 「[NSS 「データベース」 の環境設定オプション](#)」 (274 ページ)を参照してください。

表 18.7 /etc/nsswitch.confで利用できるデータベース

aliases	sendmailによって実行されたメールエイリアス。man5 aliasesコマンドで、マニュアルページを参照してください。
ethers	イーサネットアドレス
group (グループ)	getgrentによって使用されるユーザグループ。groupのマニュアルページも参照してください。
hosts	gethostbynameおよび同類の関数によって使用されるホスト名とIPアドレス。
netgroup	アクセス許可を制御するための、ネットワーク内にあ る有効なホストとユーザのリスト。netgroup(5) man ページを参照してください。
networks	ネットワーク名とアドレス。getnetentによって使用 されます。

passwd	ユーザパスワード。getpwentによって使用されます。 passwd(5) <b>man</b> ページを参照してください。
protocols	ネットワークプロトコル。getprotoentによって使用されます。protocols(5) <b>man</b> ページを参照してください。
rpc	リモートプロシージャコール名とアドレス。 getrpcbynameおよび同様の関数によって使用されます。
services	ネットワークサービス。getserventによって使用されます。
shadow	ユーザのシャドウパスワード。getspnamによって使用されます。shadow(5) <b>man</b> ページを参照してください。

**表 18.8** NSS 「データベース」 の環境設定オプション

ファイル	たとえば/etc/aliasesのような直接アクセスファイル。
db	データベース経由のアクセス。
nis、nisplus	NIS。第4章 <i>Using NIS</i> (↑セキュリティガイド)を参照。
dns	hostsおよびnetworksの拡張としてのみ使用できます。
compat	passwd、shadow、およびgroupの拡張としてのみ使用できます。

## /etc/nscd.conf

このファイルは、nscd (name service cache daemon)の環境設定に使用します。  
nscd(8)およびnscd.conf(5) **man**ページを参照してください。デフォルト



では、`nscd`によって`passwd`と`groups`のシステムエントリがキャッシュされます。キャッシュが行われないと名前やグループにアクセスするたびにネットワーク接続が必要になるため、このキャッシュ処理は **NIS** や **LDAP** といったディレクトリサービスのパフォーマンスに関して重要な意味を持ちます。`hosts`はデフォルトではキャッシュされません。これは、`nscd` でホストをキャッシュすると、ローカルシステムで正引き参照と逆引き参照のルックアップチェックを信頼できなくなるからです。したがって、`nscd`を使用して名前をキャッシュするのではなく、キャッシュ**DNS**サーバをセットアップします。

`passwd`オプションのキャッシュを有効にすると、新しく追加したローカルユーザが認識されるまで、通常、約15秒かかります。この待ち時間を短縮するには、コマンド`rcnscdrestart`を使用して`nscd`を再起動します。

## **/etc/HOSTNAME**

このファイルには、ドメイン名の付いていないホスト名が記述されています。このファイルは、マシンの起動時に複数のスクリプトによって読み込まれます。指定できるのは、ホスト名が設定されている1行のみです。

## **18.6.2 設定のテスト**

設定内容を設定ファイルに書き込む前に、それをテストすることができます。テスト環境を設定するには、`ip`コマンドを使用します。接続をテストするには、`ping`コマンドを使用します。また、以前の設定ツールの`ifconfig`や`route`も使用することができます。

`ip`、`ifconfig`、および`route`コマンドは、ネットワーク設定を直接変更します。ただし、設定ファイルに変更内容は保存されません。正しい設定ファイルに変更内容を保存しない限り、変更したネットワーク設定は再起動時に失われてしまいます。

## **ipコマンドを使ったネットワークインタフェースの設定**

`ip`は、ルーティング、ネットワークデバイス、ルーティングポリシー、およびトンネルに関する設定を行ったり、設定内容を表示したりするコマンドで

す。ipは、以前のifconfigコマンド、およびrouteコマンドに代わるコマンドとして設計されました。

ipは非常に複雑なツールです。一般的には、`ip options object command`の形式で指定します。objectの部分には、次のオブジェクトを指定することができます。

#### リンク

ネットワークデバイスを表します。

#### アドレス

デバイスのIPアドレスを表します。

#### neighbour

ARPまたはNDISCキャッシュエントリを表します。

#### route

ルーティングテーブルエントリを表します。

#### ルール

ルーティングポリシーデータベース中のルールを表します。

#### maddress

マルチキャストアドレスを表します。

#### mroute

マルチキャストルーティングキャッシュエントリを表します。

#### tunnel

IPトンネルを表します。

commandの部分に何も指定しないと、デフォルトのコマンド(通常はlist)が使用されます。

デバイスの状態を変更するには、`ip link set device_name command`コマンドを使用します。たとえば、デバイスeth0を無効にするには、`ip link set eth0 down`を実行します。このデバイスを再び有効にする場合は、`ip link set eth0 up`を実行します。

デバイスを有効にしたら、そのデバイスを設定することができます。デバイスのIPアドレスを使用する場合は、`ip addr add ip_address + dev`

`device_name`を使用します。たとえば、インタフェース`eth0`にアドレス「`192.168.12.154/30`」を設定し、標準のブロードキャスト(`brd`オプション)を使用する場合は、`ip addradd 192.168.12.154/30 brd + dev eth0`と入力します。

接続を実際に利用可能にするには、デフォルトゲートウェイの設定も必要です。システムのゲートウェイを設定するには、`ip route addgateway_ip_address`を入力します。あるIPアドレスを別のIPアドレスに変換するには、`nat:ip route add nat ip_address via other_ip_address`を使用します。

すべてのデバイスを表示する場合は、`ip link ls`を使用します。動作しているインタフェースだけを表示する場合は、`ip link ls up`を使用します。デバイスのインタフェース統計情報を印刷する場合は、`ip -s link lsdevice_name`と入力します。デバイスのアドレスを表示する場合は、`ip addr`と入力します。`ip addr`の出力には、デバイスのMACアドレスに関する情報も表示されます。すべてのルートを表示する場合は、`ip route show`を使用します。

`ip`の使用方法の詳細は、`iphelp`を入力するか、または`ip(8)`マニュアルページを参照してください。`help`オプションは、すべての`ip`オブジェクトで利用することができます。たとえば、`ipaddr`に関するヘルプを表示する場合は、「`ipaddr help`」と入力します。`ip`のマニュアルは、`/usr/share/doc/packages/iproute2/ip-cref.pdf`に用意されています。

## pingを使った接続のテスト

`ping`コマンドは、TCP/IP接続が正常に動作しているかどうかを調べるための、標準ツールです。`ping`コマンドはICMPプロトコルを使って、小さなデータパケットECHO\_REQUESTデータグラムを、宛先ホストに送信し、即時応答を要求します。この作業が成功した場合、`ping`コマンドは、その結果を知らせるメッセージを表示します。これは、ネットワークリンクが基本的に機能していることを意味します。

`ping`は、2台のコンピュータ間の接続をテストするだけでなく、接続品質に関する基本的な情報も提供します。**ping例 18.10. 「pingコマンドの出力」**(278 ページ)コマンドの実行結果例は、を参照してください。2番目の行から最

後の行には、転送パケット数、失われたパケット数、およびpingの実行時間の合計が記載されています。

PINGの宛先には、ホスト名またはIPアドレスを指定することができます。たとえば、`pingexample.com`または`ping192.168.3.100`のように指定します。`ping`コマンドを実行すると、**Ctrl+C**キーを押すまでの間、継続的にパケットが送信されます。

接続されているかどうかを確認するだけで良い場合は、`-c`オプションを使って送信するパケット数を指定することができます。たとえば、PINGを3パケットに制限する場合は、`ter ping -c 3 example.com`を入力します。

### 例 18.10 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
```

デフォルトでは、pingは1秒ごとにパケットを送信します。送信間隔を変更するには、`-i`オプションを指定します。たとえば、PING間隔を10秒に増大する場合は、`enter ping -i 10 example.com`を入力します。

複数のネットワークデバイスを持つシステムの場合、特定のインタフェースアドレスを指定してpingを実行することができます。その場合は、`-I`オプションを、選択したデバイスの名前とともに使用します。たとえば、`ping -I wlan1 example.com`と指定します。.

pingのオプションと使用方法の詳細は、`ping-h`を入力するか、または`ping(8)`マニュアルページを参照してください。

## ifconfigを使ったネットワークの設定

ifconfigは、従来のネットワーク設定ツールです。ipと違い、このコマンドはインタフェースを設定する場合にのみ使用します。ルーティングを設定する場合は、`route`を使用します。

---

## 注意: ifconfigとip

ifconfigプログラムは廃止されました。かわりにipを使用します。

---

ifconfigに引数を指定しないと、現在アクティブなインタフェースのステータスが表示されます。例 18.11. 「ifconfigコマンドの出力」 (279 ページ)のように、ifconfigでは、詳細な情報がわかりやすく表示されています。この出力では、デバイスのMACアドレス(HWaddrの値)も1行目に表示されています。

### 例 18.11 ifconfig コマンドの出力

```
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のオプションと使用方法の詳細については、ifconfig-hを入力するか、またはifconfig(8)マニュアルページを参照してください。

## routeを使ったルーティングの設定

routeは、IPルーティングテーブルを操作するプログラムです。このコマンドを使って、ルーティングの設定内容を表示したり、ルートを追加または削除することができます。

---

**注意: routeとip**

routeプログラムは廃止されました。かわりにipを使用します。

---

routeは、総合的なルーティング情報を素早く参照して、ルーティングに関する問題を探す場合などに役立ちます。現在のルーティング設定を表示するには、rootとしてroute-nを入力します。

**例 18.12 route -n コマンドの実行結果**

```
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のオプションと使用方法の詳細については、route-hを入力するか、またはroute (8) マニュアルページを参照してください。

### 18.6.3 スタートアップスクリプト

前述の環境設定ファイルに加え、マシンのブート時にネットワークプログラムをロードするさまざまなスクリプトも用意されています。これらは、システムがマルチユーザランレベルのいずれかに切り替わったときに起動します。これらのスクリプトの一部は、表18.9、「ネットワークプログラム用スタートアップスクリプト」(280 ページ)で説明されています。

**表 18.9 ネットワークプログラム用スタートアップスクリプト**

/etc/init.d/network	このスクリプトは、ネットワークインタフェースの環境設定を処理します。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フェースは実装されません。
/etc/init.d/xinetd	xinetdを開始します。xinetdを使用すると、サーバサービスがシステム上で利用できるようになります。たとえば、FTP接続の開始時に必ずvsftpdを起動するといったことができます。

<code>/etc/init.d/portmap</code>	NFSサーバなどのRPCサーバに必要なポートマップを起動します。
<code>/etc/init.d/nfsserver</code>	NFSサーバを起動します。
<code>/etc/init.d/postfix</code>	postfixプロセスを制御します。
<code>/etc/init.d/ypserv</code>	NISサーバを起動します。
<code>/etc/init.d/ypbind</code>	NISクライアントを起動します。

---

## 18.7 ダイアルアップアシスタントとしてのsmpppd

一部のホームユーザは、インターネット接続専用の回線を持っていません。代わりにダイアルアップ接続を使用しています。接続は、ダイアルアップ方法(ISDNまたはDSL)に応じてipppdまたはpppdで制御されます。基本的には、これらのプログラムを正常に起動するだけでオンラインで接続できます。

ダイアルアップ接続時に追加費用が発生しない定額接続を使用している場合は、単に該当するデーモンを起動します。ダイアルアップ接続の管理には、KDEアプレットまたはコマンドラインインタフェースを使用します。インターネットゲートウェイ以外のホストを使用している場合は、ネットワークホスト経由でダイアルアップ接続を管理できます。

smpppdが関係するのはこの部分です。このプログラムは補助プログラム用に一様なインタフェースを提供し、双方向に動作します。第1に、必要なpppdまたはipppdをプログラミングし、そのダイアルアッププロパティを制御します。第2に、各種プロバイダをユーザプログラムで使えるようにして、現在の接続ステータスに関する情報を送信します。smpppdはネットワーク経由で制御することもできるため、プライベートサブネットワーク内のワークステーションからインターネットへのダイアルアップ接続の制御に適しています。

## 18.7.1 smpppdの設定

smpppdによる接続は、YaSTにより自動的に設定されます。実際のダイアルアッププログラムであるkinternetとcinternetも事前に設定済みです。手動設定が必要となるのは、リモート制御など、smpppdの付加的機能を設定する場合のみです。

smpppdの設定ファイルは/etc/smpppd.confです。デフォルトでは、このファイルによるリモート制御はできません。この設定ファイルの最も重要なオプションを次に示します。

`open-inet-socket = yes/no`

smpppdをネットワーク経由で制御するには、このオプションをyesに設定する必要があります。smpppdがリッスンするポートは3185です。このパラメータをyesに設定した場合は、パラメータbind-address、host-rangeおよびpasswordもそれに応じて設定する必要があります。

`bind-address = ip address`

ホストに複数のIPアドレスがある場合は、このパラメータを使用してsmpppdで接続の受け入れに使用するIPアドレスを指定します。デフォルトでは、すべてのアドレスでリッスンします。

`host-range = min ipmax ip`

パラメータhost-rangeを使用して、ネットワーク範囲を定義します。この範囲内のIPアドレスを持つホストには、smpppdへのアクセス権が付与されます。この範囲外のホストはすべてアクセスを拒否されます。

`password = password`

パスワードを割り当てることで、クライアントを認可されたホストに限定できます。これはプレーンテキストによるパスワードのため、このパスワードによるセキュリティを過大評価しないでください。パスワードを割り当てないと、すべてのクライアントがsmpppdへのアクセスを許可されます。

`slp-register = yes/no`

このパラメータにより、smpppdサービスがSLPによってネットワーク上にアナウンスされます。



smpppdについての詳細は、smpppd(8)およびsmpppd.conf(5) manページを参照してください。

## 18.7.2 リモート用kinternatおよびcinternatの設定

KInternetとcinternatは、ローカルまたはリモートのsmpppdの制御に使用できます。cinternatは、グラフィックKInternetのコマンドライン版です。これらのユーティリティをリモートsmpppdに使用するには、設定ファイル/etc/smpppd-c.confを手動で、またはkinternatを使用して編集します。このファイルでは、次の4つのオプションのみを使用します。

`sites = list of sites`

このオプションでは、フロントエンドがsmpppdを検索する場所を指定します。フロントエンドは、ここに記述されている順序でオプションをテストします。localは、ローカルsmpppdへの接続の確立を指定します。gatewayオプションは、ゲートウェイ上のsmpppdをポイントします。config-fileは、/etc/smpppd-c.confファイルのserverオプションとportオプションで指定されたsmpppdに対して接続を確立することを指定しています。slpは、フロントエンドを、SLPで検出されたsmpppdに接続することを指定します。

`server = server`

このオプションでは、smpppdを実行するホストを指定します。

`port = port`

このオプションでは、smpppdを実行するポートを指定します。

`password = password`

このオプションでは、smpppd用に選択したパスワードを挿入します。

smpppdがアクティブな場合は、これでコマンドcinternat--verbose --interface-listなどのコマンドを使用してアクセスを試行できます。この時点でアクセスできない場合は、smpppd-c.conf(5)およびcinternat(8) manページを参照してください。



## 無線通信

Linuxシステムを使用して他のコンピュータ、携帯電話または周辺機器と通信するにはいくつかの方法があります。WLAN(ワイヤレスLAN)は、ネットワークラップトップに使用できます。Bluetoothは、個別のシステムコンポーネント(マウス、キーボード)、周辺機器、携帯電話、PDAとの通信、および各コンピュータ間の通信に使用できます。IrDAは、主にPDAsまたは携帯電話との通信に使用できます。UMTS (Universal Mobile Telecommunications System) は、3Gとしても知られていますが、Webのブラウズやメッセージの送受信など、複数のマルチメディアサービスを提供します。この章では、これらのテクノロジーおよび設定について説明します。

### 19.1 無線LAN

無線LANは、モバイルコンピューティングに不可欠な側面となってきています。現在、ほとんどのラップトップにはWLANカードが内蔵されています。WLANカードによる無線通信に関する802.11規格がIEEEにより策定されました。当初、この規格は最大伝送速度2MBit/sについて提供されましたが、その後、データ伝送速度を高めるために複数の補足事項が追加されています。これらの補足事項では、モジュレーション、伝送出力、および伝送速度などの詳細が定義されています(表 19.1. 「各種WLAN規格の概要」 (286 ページ)参照)。さらに、多数の企業が専有権またはドラフト機能を持つハードウェアを実装しています。

表 19.1 各種WLAN規格の概要

名前	帯域(GHz)	最大伝送速度(MBit/s)	メモ
802.11レガシー	2.4	2	廃止、実質上、使用可能なエンドデバイスはなし
802.11a	5	54	干渉が少ない
802.11b	2.4	11	あまり普及せず
28.29oz	2.4	54	広く普及、11bと後方互換
802.11nドラフト	2.4および/または5	300	Common(通常のネットワーキング)

802.11レガシーカードは、SUSE® Linux Enterprise Desktopではサポートされません。802.11a、802.11b、802.11g、および802.11nドラフトを使用する大半のカードは、サポートされています。通常、新しいカードは802.11nドラフト規格に準拠していますが、802.11gを使用するカードも使用可能です。

## 19.1.1 機能

無線ネットワークでは、高速で高品質、そして安全な接続を確保するために、さまざまなテクニックや設定が使用されています。動作のタイプが違えば、それに適したセットアップ方式も異なります。適切な認証方式を選択するのは難しいことがあります。利用可能な暗号化方式には、それぞれ異なる利点と欠点があります。

基本的に、無線ネットワークは管理ネットワークとAd-hocネットワークに分類できます。管理ネットワークには、管理要素のアクセスポイントがあります。このモード(インフラストラクチャモードとも呼ばれます)では、ネットワーク内のWLAN局の接続はすべてアクセスポイント経由で行われ、イーサネットへの接続としても機能できます。Ad-hocネットワークには、アクセスポイントはありません。アドホックネットワークでは、局同士が直接に通信するので、通常、アドホックネットワークは管理ネットワークより高速です。ただし、アドホックネットワークでは、参加局の伝送範囲と数が大幅に制限されます。それらのネットワークでは、WPA認証もサポートしません。その

ため、通常はアクセスポイントを使用します。また、WLANカードをアクセスポイントとして使用することも可能です。この機能をサポートしているカードも存在します。

## 認証

有線ネットワークよりも無線ネットワークの方がはるかに盗聴や侵入が容易なので、各種の規格には認証方式と暗号化方式が含まれています。IEEE 802.11規格のオリジナルバージョンでは、これらがWEPという用語で説明されています。ただし、WEPは安全でないことが判明したので(セキュリティ項(294ページ))、WLAN業界(Wi-Fi Allianceという団体名で協力)はWPAという新規の拡張機能を定義しており、これによりWEPの弱点がなくなるものと思われます。その後のIEEE 802.11i規格には、WPAと他の認証方式および暗号化方式が含まれています(WPAはドラフトバージョンの802.11iに基づいているので、この規格はWPA2と呼ばれることもあります)。

認可された局だけが接続できるように、管理ネットワークでは各種の認証メカニズムが使用されます。

### オープン

オープンシステムとは、認証を必要としないシステムです。任意の局がネットワークに参加できます。ただし、WEP暗号化(暗号化項(288ページ)を参照)は使用できます。

### 共有キー(IEEE 802.11に準拠)

この方式では、認証にWEPキーが使用されます。ただし、WEPキーが攻撃にさらされやすくなるので、この方式はお勧めしません。攻撃者は、局とアクセスポイント間の通信を長時間リスニングするだけで、WEPキーを奪取できます。認証処理中には、通信の両側が1度は暗号化形式、1度は暗号化されていない形式で同じ情報を交換します。そのため、適当なツールを使えば、キーを再構成することが可能です。この方式では認証と暗号化にWEPキーを使用するので、ネットワークのセキュリティは強化されません。適切なWEPキーを持っている局は、認証、暗号化および復号化を行うことができます。キーを持たない局は、受信したパケットを復号化できません。したがって、自己認証を行ったかどうかに関係なく、通信を行うことができません。

### WPA-PSK (IEEE 802.1xに準拠)

WPA-PSK (PSKはpreshared keyの略)の機能は、共有キー方式と同様です。すべての参加局とアクセスポイントは、同じキーを必要とします。キーの

長さは256ビットで、通常はパスフレーズとして入力されます。この方式では、WPA-EAPのような複雑なキー管理を必要とせず、個人で使用するのに適しています。したがって、WPA-PSKはWPA「Home」とも呼ばれます。

#### WPA-EAP (IEEE 802.1xに準拠)

実際には、WPA-EAP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、企業内の無線ネットワークを保護するために使用されます。プライベートネットワークでは、ほとんど使用されていません。このため、WPA-EAPはWPA「Enterprise」とも呼ばれます。

WPA-EAPは、ユーザを認証するのにRadiusサーバを必要とします。EAPには、サーバへの接続と認証手段として、TLS(Transport Layer Security)、TTLS(Tunneled Transport Layer Security)、およびPEAP(Protected Extensible Authentication Protocol)の、3種類の方法が用意されています。簡単に説明すると、これらのオプションは以下のように働きます。

#### EAP-TLS

TLSの認証は、サーバとクライアント両方の、証明書の相互交換に依存しています。はじめに、サーバがクライアントに対して証明書を提示し、それが評価されます。証明書が有効であるとみなされた場合には、今度がクライアントがサーバに対して証明書を提示します。TLSはセキュアですが、ネットワーク内で証明書管理のインフラストラクチャを運用することが必要になります。このインフラストラクチャは、プライベートネットワークでは通常存在しません。

#### EAP-TTLSとPEAP

TTLSとPEAPは両方とも、2段階からなるプロトコルです。最初の段階ではセキュリティ接続が確立され、2番目の段階ではクライアントの認証データが交換されます。これらの証明書管理のオーバーヘッドは、もしあるとしても、TLSよりずっと小さいものです。

## 暗号化

権限のないユーザが無線ネットワークで交換されるデータパケットを読み込んだりネットワークにアクセスしたりできないように、さまざまな暗号化方式が存在しています。

### WEP (IEEE 802.11で定義)

この規格では、RC4暗号化アルゴリズムを使用します。当初のキー長は40ビットでしたが、その後104ビットも使用されています。通常、初期化ベクタの24ビットを含めるものとして、長さは64ビットまたは128ビットとして宣言されます。ただし、この規格には一部弱点があります。このシステムで生成されたキーに対する攻撃が成功する場合があります。それでも、ネットワークをまったく暗号化しないよりはWEPを使用する方が適切です。

非標準の「ダイナミックWEP」を実装しているベンダーもあります。これは、WEPとまったく同様に機能し、同じ弱点を共有しますが、キーがキー管理サービスによって定期的に変更されます。

### TKIP (WPA/IEEE 802.11iで定義)

このキー管理プロトコルはWPA規格で定義されており、WEPと同じ暗号化アルゴリズムを使用しますが、弱点は排除されています。データパケットごとに新しいキーが生成されるので、これらのキーに対する攻撃は無駄になります。TKIPはWPA-PSKと併用されます。

### CCMP (IEEE 802.11iで定義)

CCMPは、キー管理を記述したものです。通常は、WPA-EAPに関連して使用されますが、WPA-PSKとも併用できます。暗号化はAESに従って行われ、WEP規格のRC4暗号化よりも厳密です。

## 19.1.2 YaSTでの設定;


無線ネットワークカードを設定するには、YaSTコントロールセンターで、[\[ネットワークデバイス\] > \[ネットワーク設定\]](#)の順に選択します。一般的なネットワーク設定を構成できるネットワーク設定ダイアログが開きます。一般的なネットワーク設定の詳細については、[18.4項「YaSTによるネットワーク接続の設定」](#) (242ページ)を参照してください。システムにより検出されたすべてのネットワークカードが[\[概要\]](#)タブの下にリストされます。

リストから目的のワイヤレスカードを選択し、[\[編集\]](#)をクリックして、ネットワークカード設定ダイアログを開きます。[\[Address\]](#)タブの下で、IPアドレスとして動的/静的のどちらを使用するかを決定します。[\[全般\]](#)と[\[ハードウェア\]](#)の設定([\[デバイスの起動\]](#)または[\[Firewall Zone\]](#)など)、およびドライバ設定も調整できます。大抵の場合、事前設定値を変更する必要はありません。

[次へ] をクリックして、無線ネットワークカードの詳細設定ダイアログに進みます。NetworkManagerを使用している場合(詳細は18.5項

「NetworkManager」(264 ページ)参照)は、無線デバイスの設定を調整する必要はありません。これは、それらの設定がオンデマンドでNetworkManagerによって設定されるからです。[次へ] と [はい] を使用して設定を完了します。コンピュータを特定の無線ネットワーク内でのみ使用する場合は、ここでWLAN操作の基本設定を行ってください。

### 図 19.1 YaST:無線ネットワークカードの設定

 **無線ネットワークカードの設定**  
ここでは、無線ネットワークに関する最も重要な設定を行います。動作モードではネットワーク方式... [続き](#)

**無線デバイスの設定**  
動作モード (P):  

管理

  
ネットワーク名 (ESSID) (I):  

ネットワークの検索


  
認証モード (A):  


WEP - オープン


  
鍵入力種類  
☒ パスフレーズ (P)   ☐ ASCII (A)   ☐ 16 進数 (H)  
暗号鍵 (E):  
  


熟練者向け設定 (X)

WEPキー (W)

 ヘルプ

 中止 (R)

 戻る (B)

 次へ (N)

#### 動作モード

WLANでは、局を3つのモードで統合できます。最適なモードは、アドホック(アクセスポイントのないピアツーピアネットワーク)、管理(アクセスポイントにより管理されるネットワーク)、またはマスタ(アクセスポイントとしてネットワークカードを使用)など、通信するネットワークによって異なります。WPA-PSKまたはWPA-EAPモードを使用するには、動作モードを [管理] に設定する必要があります。



## ネットワーク名(ESSID)

無線ネットワークのすべての局が相互に通信するには、同じESSIDが必要です。何も指定しなければ、カードは自動的にアクセスポイントを選択できますが、それが意図したアクセスポイントとは異なる場合があります。

Use **[Scan Network]** で使用可能な無線ネットワークのリストを取得します。

## 認証モード

ネットワークに適した認証方法を選択します( **[暗号化しない]**、 **[WEP-Open]**、 **[WEP-Shared Key]**、 **[WPA-EAP]**、または **[WPA-PSK]** )。

WPA認証を選択した場合は、ネットワーク名(ESSID)を設定する必要があります。

## キーの入力タイプ

WEPおよびWPA-PSKの認証方法では、キーの入力が必要です。キーは、 **[パスフレーズ]** ( **[ASCII]** 文字列として)入力するか、 **[16進]** 文字列として入力します)。

## WEPキー

ここでデフォルトキーを入力するか、 **[WEPキー]** をクリックして高度なキー設定ダイアログに入ります。キー長を設定します( **[128ビット]** または **[64ビット]** )。デフォルト設定は、 **[128ビット]** ビットです。ダイアログ下部にあるリスト領域では、局で暗号化に使用するキーを最大4つまで指定できます。 **[デフォルト設定とする]** を押して、4つのうち1つをデフォルトキーとして定義します。この方法で変更しない限り、YaSTでは最初に入力したキーがデフォルトキーとして使用されます。標準キーが削除された場合は、残りのキーの1つを手動でデフォルトキーに設定する必要があります。 **[編集]** をクリックし、既存のリストエントリを変更するか、新規のキーを作成します。新規作成の場合、ポップアップウィンドウが表示され、キーの入力タイプ( **[パスフレーズ]**、 **[ASCII]**、または **[16進]** )を選択する必要があります。 **[パスフレーズ]** を選択した場合は、前に指定した長さに従ってキーの生成に使用するワードまたは文字列を入力します。

**[ASCII]** を選択した場合は、64ビットキーであれば5文字、128ビットキーであれば13文字を入力する必要があります。 **[Hexadecimal]** を選択した場合は、64ビットキーであれば10文字、128ビットキーであれば26文字を16進表記で入力します。

## WPA-PSK

WPA-PSK用のキーを入力するには、入力方法として [パスフレーズ] または [16進] を選択します。 [Passphrase] モードでは、8から63文字を入力する必要があります。 [16進] モードでは、64文字を入力します。

## エキスパート設定

このボタンをクリックすると、WLAN接続の詳細設定用ダイアログが開きます。通常、事前設定値を変更する必要はありません。

## チャンネル

WLAN局が使用するチャンネルの指定を必要とするのは、 [Ad-hoc] モードと [マスタ] モードだけです。 [管理] モードでは、カードはアクセスポイントに使用可能なチャンネルを自動的に検索します。 [アドホック] モードでは、自局と他局との通信用に提供されているチャンネル(国によって11から14局)から1つを選択します。 [マスタ] モードでは、使用するカードがアクセスポイント機能を提供する必要があるチャンネルを指定します。このオプションのデフォルト設定は [自動] です。

## 転送ビットレート

ネットワークのパフォーマンスに応じて、あるポイントから別のポイントへの伝送について特定のビットレートを設定できます。デフォルト設定の [自動] では、システムは最大許容データ伝送速度を使用しようとしています。ビットレートの設定をサポートしていないWLANカードもあります。

## Access Point

複数のアクセスポイントがある環境では、MACアドレスを指定することで、その1つを事前に選択できます。

## 電源管理の使用

旅行中は、電力節減技術でバッテリーの動作時間を最大限に延ばしてください。電源管理に関する詳細については第16章 **電源管理** (195 ページ) を参照してください。電源管理を使用すると、接続品質に影響したり、ネットワーク待ち時間が増大する場合があります。

[次へ] をクリックして、セットアップを完了します。WPA-EAP認証を選択した場合は、自局をWLANに展開する前にもう1つ必要な設定ステップがあります。ネットワーク管理者から受け取った証明書を設定します。TLSの場合

は、[Identity]、[Client Certificate]、[Client Key]、および[Server Certificate]に適切な値を入力します。TTLSとPEAPでは、[Identity]と[Password]が必要です。[Server Certificate]と[Anonymous Identity]は、必要に応じて指定してください。YaSTは/etc/certから証明書を検索します。したがって、付与された証明書はこの場所に保存し、これらのファイルへのアクセスを0600(所有者による読み取り/書き込み)に制限してください。[詳細]をクリックして、WPA-EAPセットアップ用の高度認証ダイアログを入力します。EAP-TTLSまたはEAP-PEAP通信の第2ステージ用の認証方法を選択します。前のダイアログでTTLSを選択した場合は、any、MD5、GTC、CHAP、PAP、MSCHAPv1またはMSCHAPv2を選択します。PEAPを選択した場合は、any、MD5、GTCまたはMSCHAPv2を選択します。自動的に決定された設定を変更する必要がある場合は、[PEAP version]を使用して特定のPEAP実装を使用するように強制できます。

---

### 重要項目: 無線ネットワークでのセキュリティ

ネットワークトラフィックを保護するために、サポートされている認証方式と暗号化方式の1つを必ず使用してください。暗号化されていないWLAN接続では、第三者がすべてのネットワークデータを盗聴することができます。弱い暗号化(WEP)でも、まったく暗号化しないよりはましです。詳細については、[暗号化項 \(288 ページ\)](#)と[セキュリティ項 \(294 ページ\)](#)を参照してください。

---

## 19.1.3 ユーティリティ

パッケージwireless-toolsには、無線LAN固有のパラメータの設定と統計の取得を可能にするユーティリティが含まれています。詳細については、[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html)を参照してください。

## 19.1.4 WLANのセットアップに関するヒントとテクニック

これらのヒントでは、速度と安定性を微調整する方法や、WLANのセキュリティの側面について説明します。

## 安定性と速度

無線ネットワークのパフォーマンスと信頼性は、主として参加局が他局からクリーンな信号を受信するかどうか依存します。壁などの障害物があると、信号が大幅に弱くなります。信号強度が低下するほど、伝送速度も低下します。操作中には、コマンドライン(Link Qualityフィールド)で*iwconfig*ユーティリティを使用するか、またはNetworkManagerかKNetworkManagerを使用して、信号強度を確認します。信号品質に問題がある場合は、他の場所でデバイスをセットアップするか、またはアクセスポイントのアンテナ位置を調整してください。多くのPCMCIA WLANカードの場合、受信品質を実質的に向上させる補助アンテナを利用できます。メーカー指定のレート(54MBit/sなど)は、理論上の上限を表す公称値です。実際の最大データスループットは、この値の半分以下です。

## セキュリティ

無線ネットワークをセットアップする際には、セキュリティ対策を導入しなければ、伝送範囲内の誰もが簡単にアクセスできることを忘れないください。したがって、必ず暗号化方式をアクティブにする必要があります。すべてのWLANカードとアクセスポイントが、WEP暗号化をサポートしています。それでも完全に安全とは言えませんが、潜在的な攻撃者に対する障害物は存在することになります。通常、プライベート用であればWEPで十分です。WPA-PSKも適していますが、WLAN機能を持つ古いアクセスポイントやルータには実装されていません。デバイスによっては、ファームウェア更新を使用してWPAを実装できます。さらに、Linuxは大半のハードウェアコンポーネント上でWPAをサポートしますが、WPAサポートのないドライバもあります。WPAが使用できない場合、暗号化しないよりはWEPを使用することをお勧めします。高度なセキュリティ要件を持つ企業では、無線ネットワークの運用にWPAを使用する必要があります。

### 19.1.5 トラブルシューティング

WLANカードが応答しない場合は、必須ファームウェアをダウンロードしたかどうかを確認します。詳細については、`/usr/share/doc/packages/wireless-tools/README.firmware`を参照してください。

## 複数のネットワークデバイス

通常、最近のラップトップにはネットワークカードとWLANカードが搭載されています。DHCP(自動アドレス割り当て)を使用して両方のデバイスを構成すると、名前解決とデフォルトゲートウェイに問題が発生することがあります。これは、ルータはpingできるがインターネット上でナビゲーションできないことを示しています。詳細については、[http://en.opensuse.org/SDB:Name\\_Resolution\\_Does\\_Not\\_Work\\_with\\_Several\\_Concurrent\\_DHCP\\_Clients](http://en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients)にあるSupport Databas (サポートデータベース)を参照してください。

## Prism2カードの問題

Prism2チップ搭載のデバイスには、複数のドライバが用意されています。各種カードがスムーズに動作するかどうかは、ドライバに応じて異なります。この種のカードの場合、WPAに使用できるのはhostapドライバだけです。この種のカードが正常に動作しない場合、まったく動作しない場合、またはWPAを使用する必要がある場合は、`/usr/share/doc/packages/wireless-tools/README.prism2`を参照してください。

### 19.1.6 詳細情報

Linux用の無線ツールを開発したJean Tourrilhesのインターネットページには、無線ネットワークに関して役立つ情報が多数提供されています。詳細については、[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html)を参照してください。



## ネットワーク上のSLPサービス

サービスロケーションプロトコル(*SLP*)は、ローカルネットワークに接続されているクライアントの構成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には、すべての必要なサービスを含め、管理者はネットワークで利用できるサーバに関する詳しい知識が必要とされました。SLPは、ローカルネットワーク上にあるすべてのクライアントに対して特定のサービスを利用できることを通知します。このような通知情報を利用してSLPをサポートする各種アプリケーションを自動的に設定することができます。

SUSE® Linux Enterprise Desktopは、SLPによって提供されるインストールソースを使用するインストールをサポートしています。また、多くのシステムサービスは、統合SLPをサポートしています。YaSTとKonquerorは、どちらもSLP用の適切なフロントエンドを持っています。ご利用のシステムでインストールサーバ、ファイルサーバ、印刷サーバなどのSLPを使用することにより、ネットワークに接続されたクライアントに一元的な管理機能を提供します。

---

### 重要項目: SUSE Linux Enterprise DesktopでのSLPサポート

SLPサポートを提供するサービスにはcupsd、rsyncd、ypserv、openldap2、ksysguardd、saned、kdm、vnc、login、smpppd、rpasswd、postfix、およびsshd(fish経由)があります。

---

## 20.1 インストール

SLPクライアントおよびslptoolsのみがデフォルトでインストールされます。SLPによりサービスを提供する場合は、パッケージopenslp-serverをイン

ストールします。このパッケージをインストールするには、YaSTを起動し、**[ソフトウェア]**、> **[ソフトウェアの管理]** の順に選択します。ここで、**[フィルタ]**、> **[パターン]** の順に選択し、**[Misc.]** をクリックします。サーバ **[openslp-server]** を選択します。必要なパッケージのインストールを確認して、インストールプロセスを完了します。

## 20.2 SLPをアクティブ化する

SLPサービスを提供するには、システム上でslpdを実行する必要があります。マシンがクライアントとしてのみ動作し、サービスを提供しない場合は、slpdを実行する必要はありません。SUSE Linux Enterprise Desktop中のほとんどのシステムサービスと同様、slpdデーモンは別のinitスクリプトを使用して制御されます。インストール後に、このデーモンはデフォルトで非アクティブになります。一時的にこのデーモンを有効化するには、`rcslpd start`をrootで実行し、`rcslpd stop`で停止します。`restart`で再始動、または`status`でステータスチェックを実行します。ブート後にslpdを常にアクティブにする必要がある場合は、YaSTで**[システム]**、> **[システムサービス(ランレベル)]** の順に選択してslpdを有効にするか、または`insserv slpd`コマンドをrootとして実行します。これにより、ブート時に起動するサービスの設定にslpdが含まれます。

## 20.3 SUSE Linux Enterprise DesktopのSLPフロントエンド

ネットワーク内でSLPが提供するサービスを検索するには、SLPフロントエンドを使用します。SUSE Linux Enterprise Desktopには、さまざまなフロントエンドが用意されています。

### slptool

slptoolはネットワーク上のSLP照会を通知するため、またはプロプライエタリサービスを通知するために使用される単純なコマンドラインプログラムです。`slptool --help`はすべての利用可能なオプションと機能をリストします。slptoolはSLP情報を処理するスクリプトから呼び出すことができます。たとえば、現在のネットワークに自身をアナウンスするすべてのネットワーク時間サーバを検索するには、次のコマンドを実行します。



```
slptool findsrvs service:ntp
```

## YaST

YaST内では、SLPブラウザも利用できます。ただし、このブラウザをYaSTコントロールセンタを介して利用することはできません。このYaSTモジュールを起動するには、`yast2 slp`をrootユーザとして実行します。ユーザインタフェースの左側にある各種のプロトコルをクリックして、各サービスの詳細を取得してください。

# 20.4 SLPによるサービスの提供

SUSE Linux Enterprise Desktopのアプリケーションの多くはlibslpライブラリを使用することで、最初から統合SLPをサポートしています。サービスがSLPサポートでコンパイルされていない場合は、SLPを介して利用できるように次の方法のいずれかを使用してください。

## /etc/slp.reg.dによる静的登録

新規サービスに個別の登録ファイルを作成します。次はスキャナサービスを登録するためのファイルの例です。

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
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:`から開始するサービスURLです。このURLにはサービスタイプ(`scanner.sane`)および、サーバ上でサービスが使用可能になるアドレスが含まれます。`$HOSTNAME`は自動的に完全ホスト名で置き換えられます。その後ろにはサービスごとのTCPポートの名前がコロンで区切られる形で続きます。さらにサービスを表示する場合に使用される言語、登録の期間を秒単位で入力します。これらはコンマを使用してサービスURLと分けるようにします。0から65535で登録期間の値を設定します。0の場合は登録する必要がありません。65535はすべての制限を削除します。

登録ファイルにはまた、2つの変数`watch-port-tcp`および`description`が含まれます。`watch-tcp-port`はSLPサービスアナウンスとリンクし

て、slpdにサービスのステータスをチェックさせることにより、関連サービスがアクティブかどうか確認します。descriptionには、正しいブラウザを使用している場合に表示される、さらに詳細なシステム名が含まれています。

/etc/slp.regによる静的登録

この方法と、/etc/slp.reg.dによる手続きの唯一の違いは、すべてのサービスが中央のファイルにグループ化されることです。

slptoolによる動的登録

設定ファイルなしでサービスを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します。同じユーティリティを使用して、slpdを再起動しないで、既存の提供サービスの登録を取り消すことができます。

## 20.5 詳細情報

次のソースではSLPについての詳しい情報が提供されています。

RFC 2608、2609、2610

一般的にRFC 2608はSLPの定義を取り扱います。RFC 2609は、使用されるサービスURLの構文を詳細に扱います。またRFC 2610ではSLPを使用したDHCPについて説明しています。

<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、SUSE Linux Enterprise Desktopの詳細、RFC、および2つの入門用HTMLマニュアルが記載されているREADME.SuSEを含め、SLPに関する利用可能なマニュアルがすべて用意されています。SLP機能を使用するプログラマに役立つより詳細な情報については、openslp-develパッケージに含まれる『プログラマガイド』を参照してください。

## NTPによる時刻の同期

NTP (network time protocol)メカニズムは、システムの時刻をネットワーク上で同期させるためのプロトコルです。最初に、マシンは信頼できる時刻を持つサーバに時刻を照会できます。次に、ネットワーク上の他のコンピュータがこのマシン自体に対し、時刻を照会できます。目的は2つあり、絶対的な時間を維持することと、ネットワーク内のすべてのマシンのシステム時刻を同期させることです。

正確なシステムタイムを維持することはさまざまな場で重要です。ハードウェア組み込み型(BIOS)クロックがデータベースなどのアプリケーション要件に合致しないことがよくあります。システムタイムを手動で修正することは時に問題を発生させる可能性があります。たとえば、時間を逆廻りに戻すことで重要なアプリケーションの誤動作を誘発することもあります。ネットワーク内では、すべてのマシンのシステムタイムを同期させることが通常必要とされますが、手動での時刻調整はよい方法ではありません。ntpには、これらの問題を解決するメカニズムがあります。このメカニズムは常にネットワーク上の信頼できるタイムサーバに照会することで、システムタイムを調整します。さらに、電波時計のようなローカルリファレンスクロックを管理する機能があります。

---

### 注意

アクティブディレクトリによる時間同期を有効にするには、Joining an AD Domain (↑セキュリティガイド)にある指示に従います。

---

## 21.1 YaSTでのNTPクライアントの設定;

ntpは、ローカルのコンピュータクロックを時刻の基準として使用するように事前設定されています。ただし、BIOSクロックの使用は、それ以上に正確な時刻ソースが利用できない場合の代替として以外は避けるようにしてください。YaSTを利用すれば、NTPクライアントを簡単に設定することができます。ファイアウォールを利用していないシステムの場合は、クイック設定または詳細設定のいずれかを使用してください。ファイアウォールで保護されているシステムの場合は、詳細設定を使ってSuSEfirewall2の適切なポートを開きます。

### 21.1.1 NTPクライアントの詳細設定

DHCPによりネットワークで利用可能なNTPサーバのリストを取得するようにNTPクライアントを手動または自動で設定できます。[DHCP経由でNTPデーモンを環境設定]を選択した場合、後述する手動オプションは使用できません。

[一般の設定] タブの下部には、クライアントに対するサーバおよび時刻情報のその他の情報源が表示されます。必要に応じて、[追加]、[削除]、および[編集]を使用してこのリストを変更します。[Display Log] では、クライアントのログファイルを表示できます。

時刻情報の情報源を追加するには、[追加]をクリックします。表示されるダイアログで、時刻同期に使用する情報源のタイプを選択します。次のオプションを指定できます。

サーバ

別のダイアログでは、NTPサーバを選択できます。システムのブート時にサーバとクライアント間で時刻情報の同期を実行するには、[初期同期に用いる]を有効にします。[オプション] では、ntpdの追加オプションを指定できます。

[Access Control Options] を使用すると、コンピュータ上で実行するデーモンによりリモートコンピュータが実行可能なアクションを制限できます。このフィールドは、[セキュリティの設定] タブの [Restrict NTP Service to Configured Servers Only] をオンにした場合にのみ有効になります。このオプションは、/etc/ntp.conf内のrestrict節に対応しま

す。たとえば`nomodify notrap noquery`は、サーバがコンピュータのNTP設定を変更し、NTPデーモンのトラップ機能(リモートイベントのログ記録機能)を使用することを拒否します。自身の管理下でないサーバについては(たとえばインターネット上のサーバなど)、こうした制限を適用することをお勧めします。

詳細については、`/usr/share/doc/packages/ntp-doc`(`ntp-doc`パッケージの一部)を参照してください。

## ピア

ピアは、対称的な関係が確立されたコンピュータで、タイムサーバとクライアントの両方の役割を果たします。サーバの代わりに、同じネットワーク内のピアを使用するには、そのピアシステムのアドレスを入力します。ダイアログのそれ以外の内容は「サーバ」ダイアログと同じです。

## ラジオクロック

時刻同期にシステムのラジオクロックを使用するには、クロックタイプ、ユニット番号、デバイス名、およびその他のオプションをこのダイアログで指定します。ドライバを微調整するには、「ドライバの調整」をクリックします。ローカルラジオクロックの動作の詳細については、`/usr/share/doc/packages/ntp-doc/refclock.html`を参照してください。

## ブロードキャストの発信

時刻情報とクエリは、ネットワーク上にブロードキャストすることができます。このダイアログでは、このブロードキャストの送信先を指定します。電波時計のような信頼できる時刻ソースがない限りブロードキャストをアクティブにしないでください。

## ブロードキャストの着信

クライアントで情報をブロードキャスト経由で受け取る場合は、どのアドレスからのパケットを受け入れるかをこのフィールドに指定します。

## 図 21.1 高度なNTP設定:セキュリティの設定

The screenshot shows a window titled "高度な NTP 設定" (Advanced NTP Settings). Below the title bar, there is a subtitle: "chroot 環境下で NTP デモンを起動するには NTP デモンを chroot 環境下で実行する を..." followed by a blue link "続き" (Continue). The window has two tabs: "一般設定" (General Settings) and "セキュリティ設定" (Security Settings), with the latter being selected. Inside the "セキュリティ設定" tab, there are two checkboxes: the first is checked and labeled "NTP デモンを chroot 環境下で実行する (J)" (Run NTP daemon in chroot environment (J)); the second is unchecked and labeled "NTP サービスを設定したサーバに制限する (R)" (Restrict NTP service to configured servers (R)). Below these is a section titled "ファイアウォールの設定" (Firewall Settings) with an unchecked checkbox "ファイアウォールでポートを開く (E)" (Open ports in firewall (E)) and a button "ファイアウォールの詳細 (D)..." (Firewall details (D)...). Below this checkbox is the text "ファイアウォールは無効に設定されています" (Firewall is disabled). At the bottom of the window, there are three buttons: "ヘルプ" (Help) with a question mark icon, "キャンセル (C)" (Cancel) with a red X icon, and "OK (O)" (OK) with a green checkmark icon.

[セキュリティの設定]タブで、`ntpd`を`chroot jail`で起動するかどうかを指定します。デフォルトでは、*[Run NTP Daemon in Chroot Jail]* が選択されています。このオプションは、攻撃によってシステム全体が危険な状態に陥ることを防ぐので、`ntpd`が攻撃された場合のセキュリティを強化します。

*[Restrict NTP Service to Configured Servers Only]* は、リモートコンピュータがユーザのコンピュータのNTP設定を表示および変更すること、およびリモートイベントログのトラップ機能を使用することを拒否し、それによってシステムのセキュリティを向上させます。[一般の設定] タブの時間ソースのリストで、個別のコンピュータに対するアクセス制御オプションを上書きしない限り、こうした制限は有効になるとすべてのリモートコンピュータに適用されます。他のすべてのリモートコンピュータでは、ローカルタイムのクエリのみが許可されます。

SuSEfirewall2がアクティブな場合、[ファイアウォール内でポートを開く]を有効にします(デフォルト)。ポートを閉じたままにすると、タイムサーバと接続を確立することはできません。

## 21.2 ネットワークでのntpの手動設定

ネットワーク内のタイムサーバを使用するには、`server`パラメータを設定するのが最も簡単です。たとえば、タイムサーバ`ntp.example.com`がネットワークから接続可能な場合、その名前をファイル`/etc/ntp.conf`に行;として追加します。

```
server ntp.example.com
```

別のタイムサーバを追加するには、別の行にキーワードの「`server`」を挿入します。`rcntpdstart`コマンドで`ntpd`を初期化後、時間が安定し、ローカルコンピュータのクロックを修正するドリフトファイルが作成されるまで、約1時間かかります。ドリフトファイルを用いることで、バードウェアクロックの定誤差はコンピュータの電源が入った時点で、すぐに算出されます。修正はすぐに反映されるため、システム時刻がより安定します。

NTP機構をクライアントとして使用するには、2種類の方法があります。まず、クライアントは既知のサーバに定期的に時間を照会することができます。クライアント数が多い場合、この方法はサーバの過負荷を引き起こす可能性があります。2つ目は、ネットワークでブロードキャストを行う時刻サーバから送信されるNTPブロードキャストを、クライアントが待機する方法です。この方法には不利な面があります。サーバの精度が不明なこと、そしてサーバから送信される情報が誤っていた場合、深刻な問題が発生する可能性があります。

ブロードキャスト経由で時刻を取得する場合、サーバ名は必要ではありません。この場合は、設定ファイル`/etc/ntp.conf`に行`broadcastclient`を記述します。1つ以上の信頼された時刻サーバのみを使用するには、`servers`で始まる行にサーバの名前を記述します。

## 21.3 ローカルリファレンスクロックの設定

ntpソフトウェアパッケージには、ローカルリファレンスクロックに接続するためのドライバが含まれています。サポートされているクロックのリストは、ntp-docパッケージの/usr/share/doc/packages/ntp-doc/refclock.htmファイルに記載されています。各ドライバには、番号が関連付けられています。ntpでは、実際の設定は疑似IPアドレスを使用して行われます。クロックは、ネットワークに存在しているものとして/etc/ntp.confファイルに入力されます。このため、これらのクロックには127.127.t.uという形式の特別なIPアドレスが割り当てられます。ここで、tはクロックのタイプを示し、使用されているドライバを決定します。uはユニットのタイプを示し、使用されているインタフェースを決定します。

通常、各ドライバは設定をより詳細に記述する特別なパラメータを持っています。/usr/share/doc/packages/ntp-doc/driverNN.html(ここでNNはドライバの番号)ファイルは、特定のクロックタイプの情報を提供します。たとえば、「タイプ8」クロック(シリアルインタフェース経由のラジオクロック)はクロックをさらに細かく指定する追加モードを必要とします。また、Conrad DCF77レシーバモジュールはモード5です。このクロックを優先参照として使用するには、キーワードpreferを指定します。Conrad DCF77レシーバモジュールの完全なserver行は次のようになります。

```
server 127.127.8.0 mode 5 prefer
```

他のクロックも同じパターンで記述されます。ntp-docパッケージのインストール後は、ntpのマニュアルを/usr/share/doc/packages/ntp-docディレクトリで参照できます。ドライバパラメータについて説明するドライバページへのリンクは、ファイル/usr/share/doc/packages/ntp-doc/refclock.htmに記述されています。



## NetworkManagerの使用

NetworkManagerは、ラップトップなどの携帯用コンピュータのための理想的ソリューションです。NetworkManagerを使用すれば、ネットワークインタフェースの設定や移動時の有線/無線ネットワーク間の切り替について心配する必要がなくなります。NetworkManagerでは、自動的に既知の無線ネットワークに接続できます。いくつものネットワーク接続を平行に管理し、もっとも高速な接続をデフォルトとして使用することもできます。さらに、利用可能なネットワーク間を手動で切り換えたり、システムトレイのアプレットやウィジェットを使用してネットワーク接続を管理できます。

ラップトップコンピュータでは、NetworkManagerがデフォルトで有効です。ただし、NetworkManagerは、&yastを使用して、いつでも、有効または無効にできます(22.2項「NetworkManagerの有効化」(308 ページ)参照)。

### 22.1 NetworkManagerの使用

NetworkManagerは、高度で直感的なユーザインタフェースを提供します。このインタフェースを使用すると、ネットワーク環境を簡単に切り換えることができます。ただし、NetworkManagerは、次の場合には適しません。

- コンピュータが、DHCPまたはDNSサーバなど、ネットワーク内で他のコンピュータにネットワークサービスを提供している場合。
- コンピュータがXenサーバの場合、またはシステムがXen内の仮想システムである場合。

- ネットワーク設定管理にSCPMを使用する場合。SCPMとNetworkManagerを同時に使用する場合は、SCPM設定でネットワークリソースを無効にします。

## 22.2 NetworkManagerの有効化

NetworkManagerでネットワーク接続を管理する場合は、YaSTネットワーク設定モジュール内でNetworkManagerを有効にします。NetworkManagerを有効にするには、次の手順に従います:

- 1 YaSTを実行し、[ネットワークデバイス] > [NetworkSettings] の順に選択します。.
- 2 [NetworkSettings] ダイアログが開きます。[グローバルオプション] タブを開きます。.
- 3 [ネットワークのセットアップ方法] フィールドで、[User Controlled with NetworkManager] を有効にします。
- 4 [完了] をクリックします。
- 5 ネットワーク接続を管理する方法を選択したら、DHCPまたは静的IPアドレスによる自動設定でネットワークカードを設定するか、またはモデムを設定します(ダイアルアップ接続の場合は、[ネットワークデバイス] > [モデム] の順に選択)。内部またはUSB ISDNモデムを設定するには、[ネットワークデバイス] > [ISDN] の順に選択します。内部またはUSB DSLモデムを設定するには、[ネットワークデバイス] > [DSL] の順に選択します。.

YaSTを使用したネットワーク接続の詳細については、[18.4項「YaSTによるネットワーク接続の設定」](#) (242 ページ)および[19.1項「無線LAN」](#) (285 ページ)を参照してください。

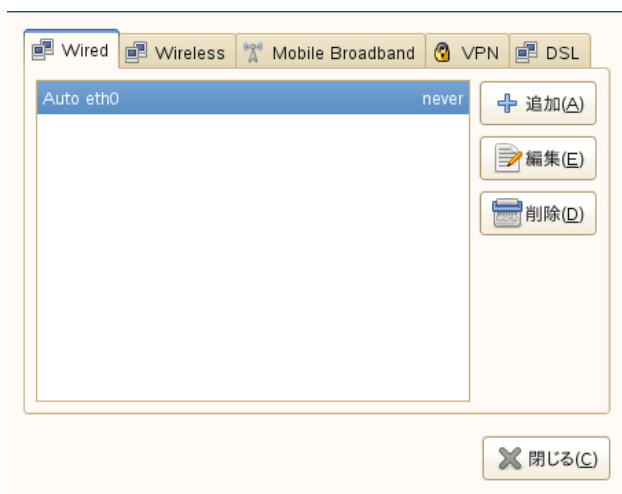
NetworkManagerを有効にした後、NetworkManagerでネットワーク接続を設定します ([22.3項「ネットワーク接続の設定」](#) (309 ページ)参照)。

NetworkManagerを無効にして、ネットワークを従来の方法で制御したい場合は、[ネットワークのセットアップ方法] フィールドの [ifupを使用した従来の方法] オプションを選択します。

## 22.3 ネットワーク接続の設定

YaST内でNetworkManagerを有効にしたら、GNOMEコントロールセンターまたはKDE 4の個人設定からのダイアログでネットワーク接続を設定します。GNOMEを使用する場合は、メインメニューからGNOMEコントロールセンターを起動し、次に、[システム] > [ネットワーク設定] の順に選択すると、[ネットワーク設定] ダイアログが開きます。KDEを使用する場合は、メインメニューから、[個人設定] を起動します。このためには、まず、[Configure Desktop] をクリックし、次に、[詳細] > [Network Settings] の順に選択して [Network Settings] ダイアログを開きます。

図 22.1 GNOME ネットワーク環境設定ダイアログ



この環境設定ダイアログは、システムトレイ内のアプレット/ウィジェットからも起動できます。このためには、[設定] (KDE)をクリックするか、またはGNOMEアプレットを右クリックして [Edit Connections] を選択します。

GNOMEおよびKDE 4の環境設定ダイアログには、すべてのタイプのネットワーク接続(有線、無線、モバイル、ブロードバンド、DSL、VPNなど)のタブが表示されます。NetworkManagerは、802.1Xで保護されたネットワークへの接続もサポートします。

新しい接続を追加するには、使用したい接続タイプのタブをクリックしてから、[追加] をクリックします。[接続名] および接続の詳細を入力します。

接続タイプごとに複数の物理デバイスを利用できる(たとえば、コンピュータに2つのイーサネットカードまたは2つのワイヤレスカードが搭載されている)場合、デバイスに接続するには、デバイスの **[MACアドレス]** (つまり、ハードウェアアドレス)を指定します。 **[OK]** または **[適用]** をクリックして設定を確定します。これで、NetworkManagerまたはウィジェットの左クリックで取得できる新規設定のネットワーク接続が、利用可能なネットワークのリストに表示されます。

---

### 注意: 非表示のネットワーク

「非表示の」ネットワーク(サービスをブロードキャストしないネットワーク)に接続するには、そのネットワークのESSID(Extended Service Set Identifier)を知る必要があります。これは、隠れたネットワークは自動的に検出できないためです。この場合、ESSIDと、必要に応じて、暗号化のパラメータを入力します。

---

各接続の編集時には、NetworkManagerでその接続を自動的に使用するか ( **[Connect Automatically]** を有効にする)、またはシステム全体で使用するか ( **[Available to all users]** を有効にする)を定義できます。そのようなシステム接続は、すべてのユーザが共有でき、NetworkManagerの起動直後で、どのユーザもログインしていないとき、利用可能になります。システム接続を作成および編集するには、rootパーミッションが必要です。

## 22.4 KDE NetworkManagerウィジェットの使用

KDE 4では、NetworkManagerを制御するためのKNetworkManagerアプレットがNetworkManagerウィジェットに置き換えられました。ウィジェットは、デスクトップまたはパネルに統合できる小さなアプリケーションです。NetworkManagerコントロールに対してネットワークが設定されている場合、ウィジェットは通常デスクトップ環境で自動的に開始し、システムトレイにアイコンとして表示されます。

NetworkManagerウィジェットは、現在のネットワークステータスをアイコンとして表示し、通知を使用して変更をレポートします。ウィジェットを使用することにより、必要に応じて新しいネットワーク接続の設定、手動による別のネットワーク接続の選択、ワイヤレスネットワークの使用の無効化、ま

たはオフラインモードへの全切り替えを行うことができます。アイコンが表示されるかどうかは、現在のネットワーク接続のタイプおよび状態に依存します。マウスカーソルをアイコンに合わせると、接続に関する詳細が表示されます。

NetworkManagerは、接続を「信頼された」と「信頼なし」という2種類で区別します。「信頼された」接続とは、明示的に選択したネットワークです。その他は「信頼なし」です。接続アイコンを右クリックすると、過去に少なくとも1回は使用したことがある接続のリストが表示されます。現在使用されている接続は、メニュー内でチェックマークが付きます。

別のネットワーク接続の選択は、いずれかの接続アプレットを左クリックしていつでも行えます。ここで選択したネットワークは、自動的に選択されるネットワークよりも優先されます。選択したネットワークは使用可能な限り使用されます。ネットワークケーブルを接続しても自動的に有線ネットワーク接続には切り替わりません。

## 22.5 GNOME NetworkManager アプレットの使用

In GNOMEでは、NetworkManagerはGNOME NetworkManagerアプレットを使用して制御できます。ネットワークがNetworkManagerコントロール用に設定されている場合、通常、アプレットはデスクトップ環境とともに自動的に起動し、システムトレイにアイコンとして表示されます。

システムトレイにGNOME NetworkManagerアプレットが表示されない場合は、おそらくアプレットが起動していません。アプレットを手動で起動するには、Alt + F2を押し、nm-appletを入力します。

### 22.5.1 有線ネットワークへの接続

コンピュータがネットワークケーブルで既存のネットワークに接続している場合、NetworkManagerアプレットを使用してネットワーク接続を選択します。

- 1 アプレットアイコンで左クリックすると、使用可能なネットワークがメニューに表示されます。メニューでは、現在使用されている接続が選択されています。

- 2 別のネットワークに切り替えるには、リストから選択します。
- 3 有線と無線のすべてのネットワーク接続を切り替えるには、アプレットアイコンを右クリックして *[Enable Networking]* を選択解除します。

## 22.5.2 ワイヤレスネットワークへの接続

使用可能な可視のワイヤレスネットワークは、*[Wireless Networks]* の下の **GNOME NetworkManager** アプレットメニューにリストされます。各ネットワークの信号強度もメニューに表示されます。暗号化された無線ネットワークには、シールドアイコンが付きます。

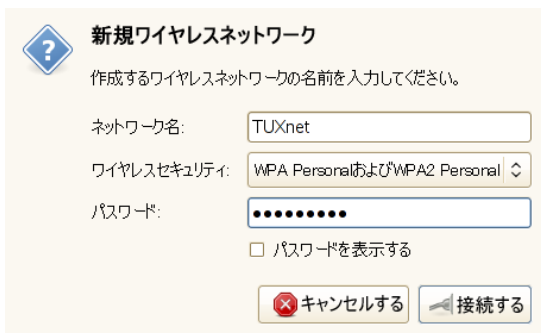
### 手順 22.1 ワイヤレスネットワークへの接続

- 1 ワイヤレスネットワークに接続するには、アプレットアイコンを左クリックして、使用できるワイヤレスネットワークのリストからエントリを選択します。
- 2 ネットワークが暗号化されている場合は、ダイアログが開きます。ネットワークが使用する *[Wireless Security]* のタイプを選択し、適切な *[パスワード]* を入力します。
- 3 ESSID (サービスセット識別子) をブロードキャストしないため自動的に検出されないネットワークに接続するには、**NetworkManager** アイコンを左クリックし、*[他のワイヤレスネットワークへの接続]* を選択します。
- 4 表示されるダイアログで、ESSIDを入力し、必要に応じて暗号化パラメータを設定します。
- 5 ワイヤレスネットワークングを無効にするには、アプレットアイコンで右クリックし、*[ワイヤレスの有効化]* のチェックを外します。これは飛行機内など、ワイヤレスネットワークングが使用できない環境にいる場合に非常に便利です。

## 22.5.3 ワイヤレスカードのアクセスポイントとしての設定

お使いのワイヤレスカードでアクセスポイントモードがサポートされている場合、NetworkManagerを使用して設定できます。

- 1 **[Create New Wireless Network]** をクリックします。



- 2 **[Wireless Security]** ダイアログで、ネットワーク名を追加し、暗号化を設定します。

---

**重要項目:** 保護されていないワイヤレスネットワークによるセキュリティリスク

**[Wireless Security]** を **[なし]** に設定した場合、誰でもネットワークに接続し、コネクティビティを再利用し、ネットワーク接続を傍受できるようになります。アクセスをアクセスポイントに制限して接続を安全なものにするには、暗号化を使用します。さまざまなWEP/WPAベースの暗号化を選択できます。いずれのテクノロジーが最適であるか不明な場合は、**認証項** (287 ページ)を参照してください。

---

## 22.6 NetworkManagerとVPN

NetworkManagerは、数種類のVPN(Virtual Private Network)技術をサポートしています:

- **NovellVPN**—パッケージ `NetworkManager-novellvpn`
- **OpenVPN**—パッケージ `NetworkManager-openvpn`
- **vpnc(Cisco)**—パッケージ `NetworkManager-vpnc`
- **PPTP(Point-to-Point Tunneling Protocol)**—パッケージ `NetworkManager-pptp`

**NetworkManager**でVPNを使用するには、まず、適切なVPNパッケージをインストールします。VPN技術ごとに、2つのパッケージが必要です(**NetworkManager**の汎用サポートを提供する上記のパッケージの1つと、アプレット用のデスクトップ固有パッケージ)。

KDEの場合は、次の中から1つ選択します。

- **NovellVPN サポート(KNetworkManager用)**—パッケージ `NetworkManager-novellvpn-kde4`
- **OpenVPNサポート(KNetworkManager用)**—パッケージ `NetworkManager-openvpn-kde4`
- **vpnc (Cisco)サポート(KNetworkManager用)**—パッケージ `NetworkManager-vpnc-kde4`

**PPTPサポート(KDE用)**はまだ利用できませんが、準備中です。

GNOMEの場合は、次の中から1つ選択します:

- **NovellVPNサポート(GNOME NetworkManagerアプレット用)**—パッケージ `NetworkManager-novellvpn-gnome`
- **OpenVPNサポート(GNOME NetworkManagerアプレット用)**—パッケージ `NetworkManager-openvpn-gnome`
- **vpnc(Cisco)サポート(GNOME NetworkManagerアプレット用)**—パッケージ `NetworkManager-vpnc-gnome`
- **PPTP(Point-to-Point Tunneling Protocol)サポート(GNOME NetworkManagerアプレット用)**—パッケージ `NetworkManager-pptp-gnome`



パッケージのインストールを完了したら、VPN接続を設定します(22.3項「ネットワーク接続の設定」 (309 ページ)参照)

## 22.7 NetworkManagerとセキュリティ

NetworkManagerは、ワイヤレス接続を「信頼された」と「信頼なし」という2種類で区別します。「信頼された」接続とは、過去に明示的に選択したネットワークです。その他は「信頼なし」です。信頼された接続は、アクセスポイントのMACアドレスと名前で識別されます。MACアドレスを使用して、信頼された接続が同じ名前でも、異なるアクセスポイントを使用できないようにすることができます。

NetworkManagerにより、定期的に、使用可能なネットワークがスキャンされます。信頼されたネットワークが複数検出された場合、最近使用されたものが自動的に選択されます。すべてのネットワークが信頼されないネットワークの場合は、NetworkManagerはユーザ選択を待機します。

暗号化設定が変更されても、名前とMACアドレスが同じままの場合は、NetworkManagerは接続を試みますが、まず、新しい暗号化設定の確認とアップデート(新しいキーなど)の提供を求めるプロンプトが表示されます。

NetworkManagerは、userおよびsystemという2種類の接続を認識します。ユーザ接続は、最初のユーザがログインしたとき、NetworkManagerで利用可能になる接続です。ユーザは、必要な資格情報を要求されます。ユーザがログアウトすると、接続は切断され、NetworkManagerから削除されます。システム接続として定義された接続は、すべてのユーザが共有でき、NetworkManagerの起動直後で、どのユーザもまだログインしていないとき、利用可能になります。システム接続の場合、すべての資格情報を接続作成時に提供する必要があります。そのようなシステム接続は、認証を要求するネットワークへの自動接続に使用することができます。ユーザ接続またはシステム接続を設定する方法については、22.3項「ネットワーク接続の設定」(309 ページ)を参照してください。

ワイヤレス接続の使用からオフラインモードへ切り替える場合は、NetworkManagerでESSIDが空白になります。これにより、カードの接続解除が確保されます。

## 22.7.1 パスワードと資格情報の保存

暗号化されたネットワークに接続するたびに資格情報を入力したくない場合は、デスクトップ固有ツールのGNOMEキーリングマネージャまたはKWalletManagerを使用して、資格情報を暗号化してディスク上に保存し、マスターパスワードで安全を確保できます。GNOMEキーリングマネージャについての詳細は、「項「Managing Keyrings」(第2章 *Customizing Your Settings*, ↑*GNOME User Guide*)」を参照してください。

NetworkManagerは、安全な接続(暗号化された有線、無線、またはVPNの接続など)のための証明書を証明書ストアから取得することもできます。詳細については、第13章 *Certificate Store* (↑セキュリティガイド)を参照してください。

もう1つのオプションとして、Novell CASAのSingle Sign-onを使用できます。Single Sign-onはアクセス制御の方法で、ユーザが1度認証すれば、複数のソフトウェアシステムのリソースにアクセスできるようにするものです。Novell CASAがシステムに設定されていると、GNOMEキーリングマネージャをロック解除するためにNetworkManagerからパスワードを要求されることはなくなります。代わりに、ユーザがデスクトップにログインするときにキーリングが自動的にロック解除されます。Novell CASAについての詳細は、「<http://developer.novell.com/wiki/index.php/Special:Downloads/casa>」を参照してください。

## 22.8 よくある質問とその回答

NetworkManagerによる特別なネットワークオプションの設定に関するよくある質問は、次のとおりです。

特定のデバイスには、どのようにして接続しますか?

デフォルトでは、NetworkManager内の接続は、デバイスタイプ固有の接続であり、同じタイプのすべての物理デバイスに適用されます。接続タイプごとに複数の物理デバイスを利用できる(たとえば、コンピュータに2つのイーサネットカードが搭載されている)場合は、所定のデバイスのハードウェアアドレス(つまり、MACアドレス)を明示的に指定することで、そのデバイスに接続できます。

デバイスのMACアドレスを調べるには、アプレット/ウィジェットから入手できる [*Connection Information*] か、またはコマンドラインツール

(nm-toolまたはifconfigなどの)の出力を使用します。次に、GNOMEコントロールセンタから [システム] > [ネットワークの環境設定] の順に選択するか、またはKDE 4内で [個人設定] から [詳細] > [Network Settings] の順に選択して、ネットワーク接続を設定するダイアログを開始します。変更したい接続を選択し、[編集] をクリックします。

[Wired] タブまたは [ワイヤレス] タブで、デバイスの [MACアドレス] を入力し、[OK] で変更を確定します。

同じESSIDを持つ複数のアクセスポイントが検出された場合、どのようにして特定のアクセスポイントを指定しますか？

異なる無線帯域(a/b/g/n)を持つ複数のアクセスポイントが利用可能な場合、デフォルトでは、最も強い信号を持つアクセスポイントが自動的に選択されます。このデフォルトを無効にするには、ワイヤレス接続の設定時に [BSSID] フィールドを使用します。

BSSID(Basic Service Set Identifier)は、各Basic Service Setを一意に識別します。インフラストラクチャBasic Service Setでは、BSSIDは、ワイヤレスアクセスポイントのMACアドレスです。独立型(アドホック)Basic Service Setでは、BSSIDは、46ビットの乱数から生成されローカルに管理されるMACアドレスです。

GNOMEコントロールセンタから [システム] > [Network Configurations] の順に選択するか、またはKDE 4内で [個人設定] から [詳細] > [Network Settings] の順に選択して、ネットワーク接続を設定するダイアログを開始します。変更したいワイヤレス接続を選択し、[編集] をクリックします。[ワイヤレス] タブで、BSSIDを入力します。

どのようにして、ネットワーク接続を他のコンピュータと共有しますか？

プライマリデバイス(インターネットに接続するデバイス)には、特別な設定はありません。ただし、ローカルハブまたはローカルコンピュータに接続するデバイスは、次の手順で設定する必要があります。

1. GNOMEコントロールセンタから [システム] > [Network Configurations] の順に選択するか、またはKDE 4内で [個人設定] から [詳細] > [Network Settings] の順に選択して、ネットワーク接続を設定するダイアログを開始します。変更したい接続を選択し、[編集] をクリックします。[IPv4 Settings] タブに切り替えます。[方法] ドロップダウンリストから、[Shared to other computers] を選択します。これで、IPトラフィックの転送が有効になり、デバイス上でDHCPサーバが実行されます。NetworkManagerで変更内容を確認します。

2. DHCPサーバは、ポート67を使用するので、サーバがファイアウォールでブロックされないことを確認してください。そのためには、まず、接続を共有するコンピュータでYaSTを起動し、[セキュリティとユーザ] > [ファイアウォール] の順に選択します。[許可されるサービス] カテゴリに切り替えます。[DHCP Server] が [許可されるサービス] として表示されていない場合は、[Services to Allow] から [DHCP Server] を選択し、[追加] をクリックします。YaSTで変更内容を確認してください。

静的DNSアドレスに、どのようにして自動(DHCP, PPP, VPN)アドレスを提供しますか？

DHCPサーバが無効なDNS情報(および/またはルート)を提供する場合は、次の手順でそれを無効にできます。GNOMEコントロールセンタから [システム] > [Network Configurations] の順に選択するか、またはKDE 4内で [個人設定] から [詳細] > [Network Settings] の順に選択して、ネットワーク接続を設定するダイアログを開始します。変更したい接続を選択し、[編集] をクリックします。[IPv4 Settings] タブに切り替え、[方法] ドロップダウンリストから、[Automatic (DHCP) addresses only] を選択します。[DNS Servers] および [Search Domains] のフィールドにDNS情報を入力します。[Routes] をクリックして、追加ルートを追加して、自動ルートを上書きします。変更内容を確認します。

どのようにしたら、ユーザがログインする前に、パスワード保護されたネットワークにNetworkManagerを接続できますか？

そのような目的に使用できるsystem connectionを定義します。詳細については、[22.7項「NetworkManagerとセキュリティ」](#) (315 ページ)を参照してください。

## 22.9 トラブルシューティング

接続に関する問題が発生する可能性があります。NetworkManagerに関する一般的な問題には、アプレットが起動しない、VPNオプションが表示されない、SCPMに関する問題などがあります。これらの問題の解決、防止方法は、使用ツールによって異なります。

NetworkManagerデスクトップアプレット/ウィジェットが起動しない

ネットワークがNetworkManagerによる制御に設定されている場合は、GNOME NetworkManagerアプレットまたはKDE NetworkManagerウィジェッ

トが自動的に起動するはずですが、アプレット/ウィジェットが起動しない場合は、YaST内でNetworkManagerが有効になっているかどうかチェックしてください(22.2項「NetworkManagerの有効化」(308 ページ)参照)。次に、デスクトップ環境に適切なパッケージがインストールされていることを確認します。KDE 4を使用する場合、該当するパッケージはNetworkManager-kde4です。GNOMEを使用する場合、該当のパッケージはNetworkManager-gnomeです

デスクトップアプレットがインストールされていても、何らかの理由で実行されていない場合(おそらく、誤って終了した場合)、nm-appletコマンドで手動で起動してください。

KDE4システムトレイにネットワーク接続のアイコンが表示されない場合は(YaSTでNetworkManagerを使用して静的ネットワーク設定からユーザ制御に切り替え後のケースなど)、NetworkManagerウィジェットをパネルに追加します。このためには、パネルの空きパッチを右クリックし、*[Panel Options]* > *[Add Widgets]* の順に選択します。(デスクトップオブジェクトが現在ロックされている場合は、オブジェクトの追加前に、まず、*[Unlock Widgets]* のクリックが必要な場合があります。表示されたダイアログボックスで、*[NetworkManager]* を選択し、*[Add Widget]* をクリックします。

NetworkManager アプレット/ウィジェットにVPNオプションがない

NetworkManager,アプレットとNetworkManager用VPNのサポートは、個別のパッケージで配布されます。NetworkManagerアプレット/ウィジェットにVPNオプションがない場合は、ご使用のVPN技術のNetworkManagerサポートを含むパッケージがインストールされているかどうか確認してください。詳細については、22.6項「NetworkManagerとVPN」(313 ページ)を参照してください。

SCPMで、ネットワーク設定が切り替わらない

SCPMをNetworkManagerとともに使用している可能性があります。現在、NetworkManagerでは、SCPMプロファイルを処理できません。SCPMもネットワーク設定を変更する場合、NetworkManagerをSCPMとともに使用しないでください。SCPMとNetworkManagerを同時に使用する場合は、SCPM設定でネットワークリソースを無効にします。

ネットワーク接続を使用できない

ネットワーク接続が正しく設定され、ネットワーク接続の他のすべてのコンポーネントも(ルータなど)、正常に機能している場合は、コンピュータ

上でネットワークインタフェースを再起動すると、問題が解決する場合があります。そのためには、コマンドラインでrootとしてログインし、`rcnetwork restart`を実行します。.

## 22.10 詳細情報

NetworkManagerの詳細は、次のウェブサイトおよびディレクトリから入手できます。

- <http://www.gnome.org/projects/NetworkManager/>—NetworkManagerプロジェクトページ
- KDE NetworkManagerウィジェットの詳細については、<http://userbase.kde.org/KNetworkManager>を参照してください。.
- NetworkManagerとGNOME NetworkManagerアプレットおよびKDE NetworkManagerウィジェットの最新情報については、次のディレクトリの情報も参照してください: `/usr/share/doc/packages/NetworkManager/`、`/usr/share/doc/packages/NetworkManager-kde4/`、`/usr/share/doc/packages/NetworkManager-gnome/`

# Samba

Sambaを使用すると、Mac OS X、Windows、OS/2マシンに対するファイルサーバおよびプリントサーバをUnixマシン上に構築できます。Sambaは、今や成熟の域に達したかなり複雑な製品です。Sambaは、YaST、SWAT(Webインタフェース)を使用するか設定ファイルを手動で編集して設定します。

## 23.1 用語

ここでは、SambaのマニュアルやYaSTモジュールで使用される用語について説明します。

### SMBプロトコル

SambaはSMB(サーバメッセージブロック)プロトコルを使用します。SMBはNetBIOSサービスを基にしています。Microsoftがこのプロトコルをリリースしたので、他のソフトウェアメーカーはMicrosoftドメインネットワークに接続できるようになりました。Sambaでは、SMBプロトコルがTCP/IPプロトコルの上で動作するので、すべてのクライアントにTCP/IPプロトコルをインストールする必要があります。

### CIFSプロトコル

CIFS (common Internet file system)プロトコルは、Sambaがサポートしているプロトコルです。CIFSは、ネットワーク上で使用する標準のリモートファイルシステムで、ユーザグループによる共同作業およびネットワーク間でのドキュメントの共有ができるようにします。

## NetBIOS

NetBIOSは、マシン間通信用に設計された、ネームサービスを提供するソフトウェアインタフェース(API)です。これにより、ネットワークに接続されたマシンが、それ自体の名前を維持できます。予約を行えば、これらのマシンを名前によって指定できます。名前を確認する一元的なプロセスはありません。ネットワーク上のマシンでは、すでに使用済みの名前でない限り、名前をいくつでも予約できます。NetBIOSインタフェースは、異なるネットワークアーキテクチャに実装できるようになっています。ネットワークハードウェアと比較的密接に機能する実装はNetBEUIと呼ばれますが、これはよくNetBIOSとも呼ばれます。NetBIOSとともに実装されるネットワークプロトコルは、Novell IPX (TCP/IP経由の NetBIOS)とTCP/IPです。

TCP/IP経由で送信されたNetBIOS名は、`/etc/hosts`で使用されている名前、またはDNSで定義された名前とまったく共通点がありません。NetBIOSは独自の、完全に独立した名前付け規則を使用しています。しかし、管理を容易にするために、DNSホスト名に対応する名前を使用するか、DNSをネイティブでを使用することをお勧めします。これはSambaが使用するデフォルトでもあります。

## Sambaサーバ

Sambaサーバは、SMB/CIFSサービスおよびNetBIOS over IPネーミングサービスをクライアントに提供します。Linuxの場合、SMB/CIFSサービス用の`smnd`と、ネーミングサービス用の`nmbd`、認証用の`winbind`の3種類のSambaサーバデーモンが用意されています。

## Sambaクライアント

Sambaクライアントは、SMBプロトコルを介してSambaサーバからSambaサービスを使用するシステムです。Mac OS X、Windows、OS/2などの一般的なオペレーティングシステムは、すべてSMBプロトコルをサポートしています。TCP/IPプロトコルは、すべてのコンピュータにインストールする必要があります。Sambaは、異なるUNIXフレーバーに対してクライアントを提供します。Linuxでは、SMB用のカーネルモジュールがあり、LinuxシステムレベルでのSMBリソースの統合が可能です。Sambaクライアントに対していずれのデーモンも実行する必要はありません。

## 共有

SMBサーバは、そのクライアントに対し、共有によってリソースを提供します。共有は、サーバ上のサブディレクトリのあるディレクトリおよびプリンタです。これは名前によってエクスポートされ、名前によってアク



セスされます。共有名にはどのような名前も設定できます。エクスポートディレクトリの名前である必要はありません。プリンタにも名前が割り当てられます。クライアントはプリンタに名前アクセスできます。

## DC

ドメインコントローラ(DC)はドメインのアカウントを処理するサーバです。データレプリケーションには、1つのドメインの中で追加のドメインコントローラが使用できます。

## 23.2 Sambaサーバの設定

Sambaサーバの設定についてはSUSE Linux Enterprise Serverマニュアルを参照してください。

## 23.3 クライアントの設定

クライアントは、TCP/IP経由でのみSambaサーバにアクセスできます。IPX経由のNetBEUIおよびNetBIOSは、Sambaで使用できません。

### 23.3.1 YaSTによるSambaクライアントの設定;

Sambaサーバ上の共有リソース(ファイルまたはプリンタ)にアクセスするSambaクライアントを設定します。[ネットワークサービス] > [Windows Domain Membership]を選択して表示されるダイアログに、ドメインまたはワークグループを入力します。[Linuxの認証にもSMBの情報を用いる]を有効にした場合、ユーザ認証はSambaサーバによって行われます。設定が終わったら、[完了]をクリックします。

## 23.4 ログインサーバとしてのSamba

Windowsクライアントが大部分を占めるネットワークでは、ユーザが有効なアカウントとパスワードを持つ場合のみ登録できることが求められるのが普

通です。Windowsベースのネットワークでは、このタスクはPDC(プライマリドメインコントローラ)によって処理されます。Windows NTサーバをPDCとして使用することもできますが、Sambaサーバを使用しても処理できます。

**例 23.1. 「smb.confファイルのグローバルセクション」** (324 ページ)に示すように、smb.confの[global]セクションにエントリを追加する必要があります。

### **例 23.1** smb.confファイルのグローバルセクション

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

暗号化されたパスワードが検証目的で使用される場合、Sambaサーバはこれを処理できるはずですが、これには、[global]セクションでエントリencrypt passwords = yesを指定します(Sambaバージョン3ではデフォルト)。また、ユーザアカウントとパスワードをWindowsに準拠した暗号化形式で作成する必要があります。そのためにはコマンドsmbpasswd -a nameを実行します。さらに次のコマンドを使用して、Windows ドメイン概念で必要になるコンピュータのドメインアカウントを作成します。

### **例 23.2** マシンアカウントのセットアップ

```
useradd hostname\$$
smbpasswd -a -m hostname
```

useraddコマンドを使用すると、ドル記号が追加されます。コマンドsmbpasswdを指定すると、パラメータ-mを使用したときにドル記号が自動的に挿入されます。コメント付きの設定例(/usr/share/doc/packages/Samba/examples/smb.conf.SuSE)には、この作業を自動化するための設定が含まれています。

### **例 23.3** マシンアカウントの自動セットアップ

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$$
```

Sambaがこのスクリプトを確実に正しく実行できるようにするため、必要な管理者許可を持つSambaユーザを選択します。これには、1人のユーザを選択してntadminグループに追加します。これにより、このLinuxグループに属する

すべてのユーザに対し、次のコマンドによってDomain Adminステータスを割り当てることができます。

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

この詳細については、`/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`の『**Samba HOWTO Collection**』の第12章を参照してください。

## 23.5 詳細情報

Sambaについての詳細な情報は、デジタルドキュメントの形で入手できます。コマンドラインから`apropossamba`と入力するとマニュアルページを参照できます。または、**Samba**マニュアルがインストールされている場合は、`/usr/share/doc/packages/samba`ディレクトリに格納されているオンラインマニュアルと例を参照できます。また、コメント付きの設定例(`smb.conf.SuSE`)が`examples`サブディレクトリに用意されています。

Sambaチームが作成した『**Samba HOWTO Collection**』にはトラブルシューティングについても説明されています。またマニュアルの**Part V**では、手順を追って設定を確認するためのガイドが用意されています。`samba-doc`パッケージのインストール後、`/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`で、『**Samba HOWTO Collection**』を参照できます。

openSUSE wikiのSambaページ<http://en.opensuse.org/Samba>を参照してください。



## NFS共有ファイルシステム

ネットワーク上でファイルシステムを分散して共有することは、企業環境では一般的なタスクです。NFSは実績のあるシステムで、イエローページプロトコルNISとも連携します。LDAPと連携する暗号化されたより安全なプロトコルについては、NFSv4を確認してください。

NFSをNISと連携して使用すると、ユーザに対してネットワークを透過的にすることができます。NFSでは、ネットワーク経由で任意のファイルシステムを分散できます。適切なセットアップを行えば、現在どの端末を使用しているかに係わりなく、常に同じ環境で作業できます。

### 24.1 必要なソフトウェアのインストール

ホストをNFSクライアントとして設定する場合、他のソフトウェアをインストールする必要はありません。デフォルトで、NFSクライアントを設定するために必要なすべてのパッケージがインストールされています。

## 24.2 YaSTによるファイルシステムのインポート

適切な権限があれば、NFSディレクトリをNFSサーバから自分のファイルツリーにマウントできます。これには、YaSTの `[NFSクライアント]` モジュールを使用します。 `[追加]` をクリックし、NFSサーバのホスト名、インポートするディレクトリ、およびディレクトリをローカルにマウントするマウントポイントを入力します。最初のダイアログで `[完了]` をクリックすると、変更が反映されます。

`[NFS Settings]` タブで、 `[ファイアウォールでポートを開く]` をクリックし、ファイアウォールを開き、リモートコンピュータからのサービスアクセスを許可します。チェックボックスの下には、ファイアウォールのステータスが表示されます。NFSv4を使用する場合は、チェックボックス `[NFSv4を有効にする]` がオンで、 `[NFSv4 Domain Name]` にNFSv4サーバが使用する値と同じ値が入力されていることを確認してください。デフォルトドメインは、`localdomain`です。

`[完了]` をクリックして、変更を保存します。詳細については、[図24.1. 「YaSTによるNFSクライアント設定」](#) (329 ページ)を参照してください。

設定は`/etc/fstab`に書かれ、指定されたファイルシステムがマウントされます。後でYaST設定クライアントを起動した時に、このファイルから既存の設定が取得されます。

## 図 24.1 YaSTによるNFSクライアント設定



## 24.3 ファイルシステムの手動インポート

ファイルシステムは、NFSサーバから手動でもインポートできます。この前提条件はRPCポートマッパーが実行中であることです。RPCポートマッパーを起動するには、rootユーザとして`rcrpcbind start`を入力します。この前提条件を満たせば、エクスポートされたリモートファイルシステムを、ローカルのハードディスクと同じようにマウントすることができます。マウントするには、次のように`mount`コマンドを使用します。

```
mount host:remote-path local-path
```

たとえば、`nfs.example.com`コンピュータからユーザディレクトリをインポートする場合は、次のコマンドを使用します。

```
mount nfs.example.com:/home /home
```

## 24.3.1 自動マウントサービスの使用

通常のローカルデバイスをマウントする場合と同様に、**autofs**デーモンを使ってリモートファイルシステムを自動的にマウントすることもできます。そのためには、`/etc/auto.master`ファイルに次のエントリを追加してください。

```
/nfsmounts /etc/auto.nfs
```

これで、`/nfsmounts`ディレクトリがクライアント上のすべての**NFS**マウントのルートディレクトリの役割を果たすようになります(`auto.nfs`ファイルが正しく設定されている場合)。ここでは、`auto.nfs`と言う名前を使用しましたが、任意の名前を選択することができます。選択したファイルに(存在しない場合はファイルを作成してください)、次の例のようにすべての**NFS**マウントのエントリを追加します。

```
localdata -fstype=nfs server1:/data  
nfs4mount -fstype=nfs4 server2:/
```

`rcautofs start`を実行して、設定をアクティブにします。この例で、`server1`の`/data`ディレクトリの`/nfsmounts/localdata`は、**NFS**でマウントされ、`server2`の`/nfsmounts/nfs4mount`は**NFSv4**でマウントされます。

**autofs**サービスの動作中に`/etc/auto.master`ファイルを編集した場合、変更内容を反映するには自動マウント機能を再起動する必要があります。再起動するには、`rcautofs restart`を実行します。

## 24.3.2 `/etc/fstab`の手動編集

`/etc/fstab`内の典型的な**NFSv3**マウントエントリは、次のようになります:

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

`/etc/fstab`ファイルに**NFSv4**マウントを手動で追加することもできます。これらのマウントの場合、3列目に**nfs**の代わりに**nfs4**を指定します。また、1列目の`nfs.example.com:`の後に、リモートファイルシステムを`/`として必ず指定してください。たとえば、`/etc/fstab`内の**NFSv4**マウント行は、次のようになります。

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```



noautoオプションを使用すると、起動時にファイルシステムが自動マウントされません。各ファイルシステムを手動でマウントする場合は、次のように、マウント用コマンドを短縮してマウントポイントの指定だけで済ませることができます。

```
mount /local/path
```

ただし、noautoオプションを入力しないと、起動時に、システムのインストールスクリプトによって、それらのファイルシステムがマウントされます。

## 24.4 NFSでのKerberosの使用

NFSでKerberos認証を使用するには、GSSセキュリティを有効にする必要があります。有効にするには、YaSTの初期ダイアログで「GSSセキュリティを有効にする」を選択します。ただし、この機能を使用するには、正常に機能するKerberosサーバが必要です。&yastはKerberosサーバの設定は行いません。その提供機能を使用するだけです。YaST環境設定に加えて、Kerberos認証も使用する場合は、NFS設定を実行する前に、少なくとも次の手順を完了してください：

- 1 サーバとクライアントが両方とも、同じKerberosドメインにあることを確認します。つまり、クライアントとサーバが同じKDC(Key Distribution Center)サーバにアクセスし、krb5.keytabファイル(the default location on any machine is /etc/krb5.keytab)を共有していなければなりません。
- 2 クライアントでrcgssd startコマンドを実行して、gssdサービスを開始します。

NFSでのKerberosの設定の詳細は、「[24.5項「詳細情報」](#) (331 ページ)」を参照してください。

## 24.5 詳細情報

NFSサーバ/クライアントの環境設定の詳細は、exports、nfs、およびmountのマニュアルページのほか、/usr/share/doc/packages/nfsidmap/

READMEにも記載されています。オンラインヘルプは、次のウェブドキュメントで参照できます。

- 詳細な技術ヘルプについては、SourceForge [<http://nfs.sourceforge.net/>]を参照してください。
- NFSでのKerberosの設定方法は、「NFS Version 4 Open Source Reference Implementation [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>]]」を参照してください。
- 「Linux NFSv4 Frequently Asked Questions [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>]]」には、NFSv4に関するFAQが用意されています。

## ファイルの同期

今日、多くの人々が複数のコンピュータを使用しています。自宅に1台、職場に1台またはそれ以上、外出時にラップトップやPDAを携帯することも珍しくありません。これらすべてのコンピュータには、多くのファイルが必要です。どのコンピュータでも作業して、ファイルを変更した後は、すべてのコンピュータで最新バージョンを使用したいと考えるでしょう。

### 25.1 使用可能なデータ同期ソフトウェア

データの同期は、高速ネットワークで固定接続されているコンピュータ間ではまったく問題なく実現できます。この場合、NFSなどのネットワークファイルシステムを使用し、ファイルをサーバに保存して、すべてのホストがネットワーク経由で同じデータにアクセスすればよいわけです。ところがこの方法は、ネットワーク接続が低速な場合、または固定でない場合には不可能です。ラップトップをもって外出しているとき、必要なファイルをローカルハードディスクにコピーする必要があります。しかし、そうすると今度は、変更したファイルを同期させる必要があります。1台のコンピュータでファイルを変更したときは、必ず他のすべてのコンピュータでファイルを更新しなければなりません。たまにコピーする程度なら、手動で`scp`または`rsync`を使用してコピーすればよいでしょう。しかし、ファイルが多い場合、手順が複雑になるだけでなく、新しいファイルを古いファイルで上書きしてしまうといった間違いを防ぐために細心の注意が必要になります。

---

## 警告: データ損失の危険

データを同期システムで管理する前に、使用するプログラムをよく理解し、機能をテストしておく必要があります。重要なファイルのバックアップは不可欠です。

---

このように手動によるデータの同期は、時間がかかる上に間違いが起りやすい作業ですが、この作業を自動化するためのさまざまな方法を採用したプログラムを使用することで手動による作業は行わずに済みます。ここでの説明は、このようなプログラムの仕組みと使用法について、一般的な理解を図ることを目的としています。実際に使用する場合は、プログラムのマニュアルを参照してください。

### 25.1.1 CVS

CVSは、多くの場合プログラムソースのバージョン管理に使用されるプログラムで、複数のコンピュータでファイルのコピーを保存する機能を持っています。したがって、データ同期にも適しています。CVSはサーバ上に一元的なリポジトリを設定し、ファイルおよびファイルの変更内容を保存します。ローカルに実行された変更はリポジトリにコミットされ、更新によって他のコンピュータに取得されます。両方の処理はユーザによって実行される必要があります。

CVSは、複数のコンピュータで変更が行われた場合、非常に優れたエラー回復力を発揮します。変更内容がマージされ、同じ行が変更された場合は、競合がレポートされます。競合が生じて、データベースは一貫した状態のままです。競合はクライアントホストで解決するためにのみ表示されます。

### 25.1.2 rsync

バージョン管理は不要であっても、低速ネットワーク接続を使用して大きなディレクトリ構造を同期させる必要がある場合は、ツールrsyncの適切に開発されたメカニズムを使用して、ファイル内の変更箇所のみを送信できます。この処理では、テキストファイルのみでなくバイナリファイルも対象となります。ファイル間の差分を検出するために、rsyncはファイルをブロック単位で分割してチェックサムを計算します。

変更内容の検出処理は高コストを伴います。`rsync`の使用量に合わせて、同期対象となるシステムの規模を調整する必要があります。特に、RAMが重要です。

## 25.2 プログラムを選択する場合の決定要因

使用するプログラムを決定する際に重要な要因がいくつかあります。

### 25.2.1 クライアントサーバか、ピアツーピアか

一般に、データの配信には2種類のモデルが使用されます。1つは、すべてのクライアントが、そのファイルを一元的なサーバによって同期させるモデルです。サーバはすべてのクライアントから、少なくともいずれかの時点でアクセスできる必要があります。このモデルは、CVSが使用します。

もう1つは、すべてのネットワークホストがそれぞれのデータをピアとして相互に同期させるモデルです。`rsync`は、実際にクライアントモードで動作しますが、任意のクライアントがサーバとして動作できます。

### 25.2.2 移植性

CVS、および`rsync`は、各種のUNIXおよびWindowsシステムなど、他の多くのオペレーティングシステムでも使用できます。

### 25.2.3 インタラクティブと自動制御

CVSでは、ユーザが手動によってデータの同期を開始します。これにより、データの同期を詳細に制御でき、競合の処理も容易です。ただし、同期の間隔が長すぎると、競合が起こりやすくなります。

## 25.2.4 競合:問題と解決策

複数のユーザが大きなプログラミングプロジェクトにかかわっている場合も、CVSでは、競合はまれにしか発生しません。これはドキュメントが個別の行単位でマージされるためです。競合が起これば、影響を受けるのは1台のクライアントだけです。CVSでは、通常、競合が容易に解決できます。

rsyncには、競合処理の機能はありません。ユーザは、意図せずにファイルを上書きしないように注意し、考えられる競合はすべて手動で解決する必要があります。安全のために、RCSなどのバージョンングシステムを追加採用できます。

## 25.2.5 ファイルの選択と追加

CVSでは、新しいディレクトリやファイルは、コマンド`cvs add`を使って明示的に追加する必要があります。これにより、同期の対象となるファイルについて、ユーザがより詳細に制御できます。しかし他方で、新しいファイルが見過ごされることが多く、特に`cvs update`の出力に表示される疑問符は、ファイルの数が多いためにたびたび無視されます。

## 25.2.6 履歴

CVSは追加機能として、古いバージョンのファイルが再構成できます。変更を行うたびに簡単な編集コメントを挿入しておくことで、内容とコメントからファイルの作成状況を後で簡単に追跡できます。これは論文やプログラムテキストを作成する際、貴重な支援となります。

## 25.2.7 データ量と必要なハードディスク容量

同期の対象となるすべてのホストには、分散されたデータを処理できるだけの十分なハードディスクの空き容量が必要です。CVSでは、サーバ上のリポジトリデータベースに余分な容量が必要となります。ファイルの履歴もサーバに保存されるため、このための容量も別に必要です。テキスト形式のファイルが変更されたときには、変更された行だけを保存すれば足ります。バイナリファイルは、ファイルが変更されるたびに、ファイルのサイズと同じだけの容量が必要なため、テキストより必要な容量が多くなります。

## 25.2.8 GUI

CVSを使い慣れたユーザは、通常、コマンドラインでプログラムを制御します。しかし、*cervisia*のようなLinux用のグラフィカルユーザインタフェースがあり、また他のオペレーティングシステム用に*wincvs*なども用意されています。*kdevelop*などの開発ツールや*Emacs*などのテキストエディタの多くが、CVSをサポートしています。競合の解決は、これらのフロントエンドの方が、はるかに容易です。

## 25.2.9 使いやすさ

*rsync*は、より使いやすく初心者向けです。CVSは、より操作が難しくなっています。ユーザはレポジトリとローカルデータの間のインタラクションを理解する必要があります。データを変更すると、最初にローカルでレポジトリとマージする必要があります。これはコマンド*cvs*または*update*で実行します。次にコマンド*cvs*または*commit*でデータをレポジトリに送信する必要があります。この手順をいったん理解すれば、初心者の方でもCVSを簡単に利用できるようになります。

## 25.2.10 攻撃に備えるセキュリティ

伝送中、データは妨害や改ざんから保護される必要があります。CVSや*rsync*はいずれも*ssh*(セキュアシェル)経由で容易に使用できるため、この種の攻撃からセキュリティ保護されます。CVSを*rsh*(リモートシェル)経由で実行するのは避けるべきです。また、安全でないネットワークで*pserver*メカニズムを使用してCVSにアクセスすることもお勧めできません。

## 25.2.11 データ損失からの保護

CVSは、プログラミングプロジェクト管理のため長期間にわたって開発者に使用されてきたため、きわめて安定しています。CVSでは開発履歴が保存されるため、誤ってファイルを削除するといったユーザの誤操作にも対応できます。

表 25.1 ファイル同期化ツールの機能: -- = とても悪い、- = 悪い、または利用不可、o = 普通、+ = 良好、++ = とても良好、x = 利用可能

	CVS	rsync
クライアント/サーバ	C-S	C-S
移植性	Lin、Un*x、Win	Lin、Un*x、Win
対話処理	x	x
Speed	o	+
競合	++	o
ファイル選択	Sel./file, dir.	ディレクトリ
履歴	x	-
ハードディスクスペース	--	o
GUI	o	-
難度	o	+
攻撃	+ (ssh)	+(ssh)
データ損失	++	+

### 25.3 CVSの概要

CVSは、個々のファイルが頻繁に編集され、ASCIIテキストやプログラムソーステキストのようなファイル形式で保存される場合の同期に適しています。CVSを使用して他の形式、たとえばJPEGファイルのデータを同期させることは可能ですが、データ量が膨大になるとともに、生成される数多くのファイルをCVSサーバに恒久的に保存する必要があります。このような場合、CVSの機能のほとんどが利用できません。CVSを使用したファイルの同期は、すべてのワークステーションが同じサーバにアクセスできる場合のみ可能です。



## 25.3.1 CVSサーバの設定

サーバとは、すべてのファイルの最新バージョンを含め、有効なファイルが配置されるホストです。固定のワークステーションであれば、どれでもサーバとして使用できます。可能であれば、CVSレポジトリのデータを定期バックアップに含めます。

CVSサーバを設定するとき、できればユーザアクセスをSSH経由で許可します。ユーザがサーバにtuxとして認識され、CVSソフトウェアがサーバとクライアントにインストールされている場合、次の環境変数をクライアント側に設定する必要があります。

```
CVS_RSH=ssh CVSROOT=tux@server:/serverdir
```

コマンドcvsinitを使用して、クライアント側からCVSサーバを初期化します。これは一度だけ実行すれば、後は必要ありません。

最後に、同期に名前を付ける必要があります。クライアント上で、CVSで管理するファイル専用のディレクトリ(空のディレクトリ)を選択するか作成します。ディレクトリには、同期用の名前を付けます。この例で、ディレクトリ名はsynchomeです。このディレクトリに移動し、次のコマンドを入力して、同期名をsynchomeと設定します。

```
cvs import synchome tux wilber
```

CVSの多くはコメントが必要です。このため、CVSはエディタを起動します(環境変数\$EDITORで定義されたエディタか、エディタが定義されていない場合はvi)。事前に次の例のようなコマンドラインにコメントを入力しておけば、エディタ呼び出しが避けられます。

```
cvs import -m 'this is a test' synchome tux wilber
```

## 25.3.2 CVSの使用

これで、すべてのホストがcvsco synchomeを使用して同期リポジトリからチェックアウトできます。これにより、クライアントに新しいサブディレク

トリsynchomeが作成されます。変更内容をサーバにコミットするには、ディレクトリsynchome(またはそのサブディレクトリ)に移動し、「cvs commit」と入力します。

デフォルトでは、すべてのファイル(サブディレクトリを含め)がサーバにコミットされます。個別のファイルまたはディレクトリだけをコミットするには、`cvs commit file1 directory1`のように指定します。新しいファイルとディレクトリは、サーバにコミットする前に、`cvs add file1 directory1`のようなコマンドを使用してレポジトリに追加する必要があります。この後、`cvs commit file1 directory1`を実行して、新しく追加したファイルとディレクトリをコミットします。

他のワークステーションに移動する場合、同じワークステーションの以前のセッションで同期レポジトリからチェックアウトしていない場合は、ここでチェックアウトします。

サーバとの同期は、`cvs update`を使用して起動します。`cvs update file1 directory1`を使用すると、ファイルやディレクトリを個別に更新できます。現行のファイルとサーバに格納されているバージョンとの違いを確認するには、コマンド`cvs diff`または`cvs diff file1 directory1`を使用します。更新によって変更されたファイルを確認する場合は、`cvs -nq update`を使用します。

更新時に表示されるステータス記号の例を次に示します。

## U

ローカルバージョンが更新されました。この更新はサーバが提供しているすべてのファイル、およびローカルにシステムに存在しないすべてのファイルに影響します。

## M

ローカルバージョンが変更されました。サーバ上で変更があれば、その差分がローカルコピーに取り込まれていることがあります。

## P

ローカルバージョンに対し、サーバ上のバージョンからパッチが適用されました。

## C

ローカルファイルが、レポジトリの現在のバージョンと競合しています。

?

このファイルがCVSに存在しません。

ステータスMは、ローカルで変更されたファイルを示します。ローカルコピーをサーバにコミットするか、ローカルファイルを削除して更新を再実行します。この場合、不足しているファイルは、サーバから取得されます。ローカルに変更したファイルをコミットしたが、そのファイルで同じ行に変更があり以前にコミットされている場合は、競合がCで示されて表示されることがあります。

この場合、ファイル内の競合マーク(「>>」および「<<」)を確認し、2つのバージョンのどちらを採用するか決定します。これは厄介な作業のため、変更を破棄し、ローカルファイルを削除して「cvs up」と入力し、現在のバージョンをサーバから取得することもできます。

## 25.4 rsyncの概要

rsyncは、大量のデータを定期的に変送する必要があるが、変更量はあまり多くない場合に便利だ。たとえば、バックアップの作成時などが該当します。もう1つのアプリケーションはステージングサーバに関係します。この種のサーバには、DMZでWebサーバに定期的に変ミラー化されるWebサーバの完全なディレクトリツリーが格納されます。

### 25.4.1 設定と操作

rsyncには2つの操作モードがあります。このプログラムを使用してデータをアーカイブまたはコピーできます。そのためには、ターゲットシステム上にsshなどのリモートシェルがあれば十分です。ただし、rsyncをdaemonとして使用し、ネットワークにディレクトリを提供することもできます。

rsyncの基本操作モードの場合、特別な設定は不要です。rsyncでは、ディレクトリ全体を別のシステムに直接ミラー化できます。たとえば、次のコマンドでは、tuxのホームディレクトリのバックアップがバックアップサーバsun上に作成されます。

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

次のコマンドは、ディレクトリを復元する場合に使用します。

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

ここまでの操作は、`scp`のような通常のコピーツールの場合とほぼ同じです。

「rsync」のすべての機能を完全に使用可能にするには、「rsync」モードで操作する必要があります。そのためには、いずれかのシステムでrsyncdデーモンを起動します。設定はファイル/etc/rsyncd.conf内で行います。たとえば、rsyncでディレクトリ/srv/ftpを使用可能にするには、次の設定を使用します。

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

```
[FTP]
```

```
path = /srv/ftp
comment = An Example
```

次に、`rcrsyncdstart`を使用してrsyncdを起動します。また、ブート処理中にrsyncdを自動的に起動する方法もあります。このようにセットアップするには、このサービスをYaSTのランラベルエディタで有効にするか、またはコマンド`insservrsyncd`を入力します。かわりに、`xinetd`からrsyncdを起動することもできます。ただし、この方法はrsyncdの使用頻度が低いサーバの場合にのみ使用してください。

この例では、すべての接続を示すログファイルも作成されます。このファイルは/var/log/rsyncd.logに格納されます。

これで、クライアントシステムからの転送をテストできます。そのためには次のコマンドを使用します。

```
rsync -avz sun::FTP
```

このコマンドを入力すると、サーバのディレクトリ/srv/ftpにあるファイルがすべてリストされます。このリクエストはログファイル/var/log/rsyncd.logにも記録されます。実際の転送を開始するには、ターゲットディレクトリを指定します。現在のディレクトリには..`..`を使用してください。たとえば、次のようにします。

```
rsync -avz sun::FTP .
```

デフォルトでは、`rsync`での同期中にファイルは削除されません。ファイルを削除する必要がある場合は、オプション`--delete`を追加してください。新しい方のファイルが削除されないように、代わりにオプション`--update`を使用することもできます。競合が発生した場合は、手動で解決する必要があります。

## 25.5 詳細情報

### CVS

CVSの重要情報については、ホームページ「<http://www.cvshome.org>」を参照してください。

### rsync

`rsync`に関する重要な情報は、マニュアルページ`manrsync`および`manrsyncd.conf`を参照してください。`rsync`の基本原則に関する技術情報については、`/usr/share/doc/packages/rsync/tech_report.ps`を参照してください。`rsync`の最新ニュースについては、このプロジェクトのWebサイト<http://rsync.samba.org/>を参照してください。

