# SUSE LINUX

SUSE LINUX PROFESSIONAL 9.2 アドミニストレーションガイド

Edition 4 2005



ご意見やご感想は、<documentation@suse.de>までお寄せください。

*Authors:* Stefan Behlert, Frank Bodammer, Stefan Dirsch, Olaf Donjak, Roman Drahtmller, Torsten Duwe, Thorsten Dubiel, Thomas Fehr, Stefan Fent, Werner Fink, Kurt Garloff, Carsten Gro, Joachim Gleiner, Andreas Grnbacher, Franz Hassels, Andreas Jaeger, Klaus Kmpf, Andi Kleen, Hubert Mantel, Lars Marowsky-Bree, Chris Mason, Johannes Meixner, Lars Mller, Matthias Nagorni, Anas Nashif, Siegfried Olschner, Peter Pml, Thomas Renninger, Heiko Rommel, Marcus Schfer, Nicolaus Schler, Klaus Singvogel, Hendrik Vogelsang, Klaus G. Wagner, Rebecca Walter, Christian Zoz

*Editors:* Jrg Arndt, Antje Faber, Berthold Gunreben, Roland Haidl, Jana Jaeger, Edith Parzefall, Ines Pozo, Thomas Rlz, Thomas Schraitle, Rebecca Walter

*Layout:* Manuela Piotrowski, Thomas Schraitle

*Setting:* DocBook-XML, LaTeX

This book has been printed on 100 % chlorine-free bleached paper.

# はじめに

新しいLinuxオペレーティングシステム、SUSE LINUX 9.3をお選びいただき、ありがとうございます。このバージョンをご購入いただいたお客様は、http://www.novell.com/products/linuxprofessional/support/conditions.htmlに記載されているインストールサポートを電話および電子メールでご利用いただけます。このサービスをご利用いただくには、CDケースに印字されているコードを使用して、SUSE LINUX Portal (http://portal.suse.com)でサポートの認証を受けてください。

ご利用のシステムを常に安全かつ最新の状態に保つために、YaSTオンラインアップデートを使用して定期的にシステムをアップデートすることをお勧めします。SUSEは、SUSE LINUXのセキュリティ関連の情報や、ヒントおよびテクニックを記載した無料の電子ニュースレターを配信しています。http://www.novell.com/company/subscribe/に電子メールアドレスを入力するだけで、このニュースレターを購読できます。

『SUSE LINUX SUSE LINUX Professional 9.2 アドミニストレーションガイド』には、SUSE LINUXシステムの動作に関する背景情報が記載されています。Linuxシステム管理の基本であるファイルシステム、カーネル、ブートプロセス、およびApache Webサーバの環境設定に関する説明があります。『SUSE LINUX SUSE LINUX Professional 9.2 アドミニストレーションガイド』は、次の5つカテゴリで構成されています。

**インストール** YaSTを使用したシステムのインストールと環境設定、特殊なインストールタイプ、LVM、RAID、アップデート、およびシステム回復について説明します。

**システム** SUSE LINUXの特別な機能、カーネルの詳細、ブートの概念、initプロセス、ブートローダとX Window Systemの環境設定、印刷、およびLinuxでのモバイルコンピューティングについて説明します。

**サービス** 異種ネットワークへの統合、Apache Webサーバの環境設定、ファイル同期、およびセキュリティについて説明します。

**管理** ファイルシステムのACLおよび重要なシステム監視ツールについて説明します。

**付録** Linuxに関する情報の重要なソースを紹介します。

デジタル版のSUSE LINUXマニュアルは、`/usr/share/doc/manual/`ディレクトリにあります。

# アドミニストレーションガイドの変更点

以前のバージョン(SUSE LINUX 9.2)のマニュアルから変更された点は、次のとおりです。

- LVMとパーティション分割に関するセクションが改訂されました。詳細については、項3.7. 「LVMの設定」および項2.7.5. 「パーティション」を参照してください。
- 章 8. ブートローダが改訂され、YaSTモジュールの説明が追加されました。ワイルドカードの使用に関する新しいセクション(項8.3.1. 「ワイルドカードを使用したブートカーネルの選択」)も追加されました。
- ファイルシステムの章に、Reiser4ファイルシステムに関する情報が追加されました。詳細については、項20.2.5. 「Reiser4」を参照してください。
- ネットワークの部分が完全に改訂され、再編成されました。詳細については、章 22. ネットワークの基礎以降の章を参照してください。
- SuSEfirewall2がアップデートされ、新しいYaSTモジュールの説明が追加されました。詳細については、項34.1.4. 「YaSTによる設定」を参照してください。
- 章 36. システムモニタリングユーティリティに、新しいプログラムが追加されました。
- 用語集が改訂され、更新されました。詳細については、も参照してください。

# 書体

本書では、次の書体を使用しています。

- `/etc/passwd`:ファイル名またはディレクトリ名
- ⟨*placeholder*⟩:⟨*placeholder*⟩は、実際の値で置き換えられます。
- `PATH`:環境変数PATH
- `ls`:コマンド
- `--help`:オプションとパラメータ
- `user`:ユーザ
- Alt:使用するキー
- ［'ファイル'］:メニュー項目、ボタン
- Process killed:システムメッセージ
- man man(1):マニュアルページへの参照
- ► **x86, AMD64**
  この項は、指定されたアーキテクチャにのみ関連しています。矢印は、テキストブロックの先頭と終わりを示します。◄

# 謝辞

Linuxの開発は、世界中で多数のLinux開発者がボランティアとして参加することにより、進められています。世界中のLinux開発者の貢献に感謝します。このディストリビューションは、このような人々の協力なしには存在し得ませんでした。加えて、Frank ZappaとPawarにも感謝します。当然のことですが、Linus Torvaldsにも深く感謝します。

大いに楽しんでください。

チーム一同

# Contents

# Part I

# インストール

# YaSTによるインストール

この章では、システムアシスタントYaSTを使用したSUSE LINUXシステムのインストールを系統的に解説します。インストール手順の準備に関する記述には、個別の環境設定で適切な指定を行うための背景情報が含まれます。.

# 1.1 インストール時のシステム起動

ドライブに、1枚目のSUSE LINUX CDまたはDVDを挿入します。続いて、コンピュータを再起動し、ドライブのメディアからインストールプログラムを開始します。

## 1.1.1 ブートオプション

CDまたはDVD以外からブートする方法もあり、何らかの障害でCDやDVDからブートできない場合に使用できます。これらのオプションについては、表 1.1. 「ブートオプション」に記載されています。

*Table 1.1: ブートオプション*

| ブートオプション | 説明 |
|---|---|
| CD-ROM | これが最も簡単なブートオプションです。このオプションは、Linuxでサポートされ ているCD-ROMが、システムのローカルにある場合に使用できます。 |
| フロッピー | ブートフロッピーを作成するためのイメージは、CD 1の/boot/ディレクトリにあります。READMEも同じディレクトリに格納されています。 |
| PXEまたはBOOTP | このオプションが使用できるのは、システムのBIOSまたはファームウェアにサポートされている場合に限られます。また、ブートサーバはネットワーク内にあることが前提です。別のSUSE LINUXシステムで、このタスクを実行させることもできます。 |
| ハードディスク | SUSE LINUXは、ハードディスクからブートすることもできます。ハードディスクからブートするには、CD 1の/boot/loaderディレクトリから、カーネル(linux)とインストールシステム(initrd)をハードディスクにコピーし、ブートローダに適切なエントリを追加します。 |

### 1.1.2 システムをブートするときに起こり得る問題

古いハードウェアまたはサポートされていないハードウェアを使用してCDまたはDVDからブートすると問題が発生する可能性があります。ご使用のCD-ROMドライブがCD 1 上のブートイメージを読み込めない場合、CD2を使用してシステムをブートしてください。CD2には従来の2.88 MBブートイメージが格納されており、サポートされていないドライブでも読み込むことができます。それによりネットワークを介してのインストールできます。

BIOS上でブートシーケンスが正常に設定されていない場合があります。BIOSの設定を変更する方法については、ご使用のマザーボードのマニュアルを参照してください。以降では、基本的な手順について説明します。

BIOSとはコンピュータの非常に基本的な機能を有効にするソフトウェアです。マザーボードを供給するベンダが、独自のハードウェア用のBIOSを供給します。通常、BIOSセットアップは特別な時(マシンのブート時)にだけアクセスされます。この初期化段階の間に、マシンは数多くのハードウェア診断テストを実行します。そのうちの1つとして、メモリカウンタにより示されるメモリチェックがあります。メモリカウンタが表示されたとき、通常カウンタの下または画面の下部の辺りに、BIOSセットアップにアクセスするために押すキーについて表示されています。通常押すキーは、Del、F1、またはEscです。BIOSセットアップ画面が表示されるまでこのキーを押します。

**Important**

**BIOSでのキーボードレイアウト**

BIOS設定では、通常USキーボードレイアウトが使用されます。

**Important**

AWARD BIOSでブートシーケンスを変更するには、［'BIOS FEATURES SETUP']エントリを探してください。他のメーカでは、［'ADVANCED CMOS SETUP'］といった違う名前が使用されています。エントリが見つかったなら、そのエントリを選択して、Enterを押して確定します。

開いた画面で、［'BOOT SEQUENCE'］というサブエントリを探します。ブートシーケンスは、通常C,AまたはA,Cのように設定されています。C,Aの場合、マシンは最初にハードディスク(C)を検索し、次にフロッピーディスクドライブ(A)を検索して、ブート可能なメディアを検出します。ブートシーケンスがA,CDROM,Cになるまで PgUp または PgDown を押して、設定を変更します。

Escを押してBIOS設定画面を終了します。設定を保存するには、［'SAVE & EXIT SETUP'］を選択し、F10を押します。設定が保存されていることを確認するには、Yを押します。

SCSI CD-ROMドライブを使用している場合、SCSI BIOSの設定を変更します。たとえばAdaptecホストアダプタの場合、Ctrl-Aを押してセットアップを開きます。次に'Disk Utilities'を選択します。接続されているハードウェアコンポーネントが表示されます。ご使用のCD-ROMドライブに割り当てられているSCSI IDの記録をとります。Escを押してメニューを終了し、'アダプタセッティングの設定'を開きます。'追加オプション'で、'Boot Device Options(ブートデバイスオプション)'を選択し、Enterを押します。CD-ROMドライブのIDを入力して、再度Enterを押します。次に、Escを2回押して、SCSI BIOSの起動画面に戻ります。'Yes'を押してコンピュータをブートする設定を確定し、この画面を終了します。

## 1.2 ブート画面

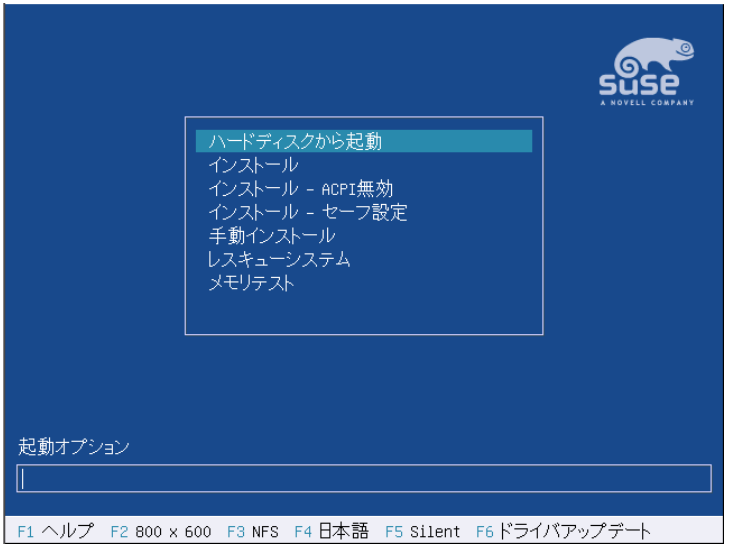


*Figure 1.1: ブ ー ト 画 面*

ブート画面には、インストール手順の複数のオプションが表示されます。['Boot from Hard Disk (ハードディスクからブート)'] は、すでにインストールされたシステムをブートします。CDがドライブに挿入されたままになって

いる場合が多いため、デフォルトでこのオプションが選択されています。システムをインストールするには、矢印キーで移動し、インストールオプションを選択します。関連するオプションは次のとおりです。

**インストール** 通常のインストールモード。最新のハードウェア機能のすべてが有効になります。

**Installation—ACPI Disabled(インストール—ACPI無効)**
通常のインストールが失敗する場合、システムのハードウェアがACPI (advanced configuration and power interface)をサポートしないことが原因である可能性があります。ACPIが原因と考えられる場合は、このオプションを使用し、ACPIのサポートを省略してインストールします。

**Installation—Safe Settings(インストール—セーフ設定)**
システムをDMAモード(CD-ROMドライブ用)でブートし、電源管理機能は無効になります。上級者はコマンドラインを使用して、カーネルのパラメータを入力、変更することもできます。

インストールの設定番号を変更するには、画面下部のバーに表示されているファンクションキーを使用します。

F1 ブート画面上にあるアクティブな要素の状況依存ヘルプ。

F2 インストールに使用するグラフィカルディスプレイモードの多彩な選択肢。グラフィカルインストールが障害の原因になる場合は、テキストモードを選択することも可能です。

F3 通常、インストールはデバイスに挿入されたメディアから実行されます。FTPまたはNFSサーバなど、他のソースを使用する場合は、ここで選択します。SLPサーバを利用し、ネットワーク経由でインストールする場合、サーバでインストールに使用できるソースの1つとして、このオプションを選択することができます。SLPに関する詳細は、章 23. ネットワーク上のSLPサービスを参照してください。

F4 インストール時の表示言語を選択します。

F5 デフォルトでは、システム起動時に、Linuxカーネルの診断メッセージは表示されません。進行状況バーのみが表示されます。これらのメッセージを表示するには、［'Verbose (詳細)'］を選択します。グラフィカルフレームなしで詳細情報を表示するには、［'Native (ネイティブ)'］を選択してください。

F6 このキーを使用し、SUSE LINUX用のドライバアップデートを含むディスクがあることを、システムに通知します。インストール手順が適切な段階に達すると、アップデートディスクの挿入を促すメッセージが表示されます。

インストールを開始すると間もなく、SUSE LINUXはインストール手順の実行に必要な最低限のLinuxシステムをロードします。［'Native(ネイティブ)'］または［'Verbose(詳細)'］を有効にした場合、メッセージと著作権表示をスクロールし、プロセスのロードが完了すると、YaSTインストールプログラムが開始します。その数秒後、画面にグラフィカルインストーラが表示されます。

SUSE LINUXの実際のインストールは、この時点から開始します。YaSTには、全画面で共通のレイアウトがあります。マウスかキーボードを使用することで、ボタン、入力フィールド、リストのすべてにアクセスできます。マウスポインタが動作しない場合、そのマウスは自動検出されていません。このような場合は、一時的にキーボードを使用してください。キーボードを用いたナビゲーションは、項2.9.1.「モジュールでのナビゲーション」の記述と同様になります。

## 1.3 言語の選択

YaSTおよびSUSE LINUXは通常、必要に応じて、設定に多様な言語を使用できます。ここで選択された言語は、キーボード配列にも使用されます。さらに、YaSTはシステムクロックのタイムゾーンを推測するためにも、この言語設定を使用します。これらの設定は、システムにインストールする2番目の言語の選択にとともに、後で変更することができます。マウスが機能しない場合は、矢印キーで言語を選択し、［'了解'］が選択されるまでTabキーを押します。続いてEnterキーを押し、言語の選択を確定します。

## 1.4 インストールモード

［'新規インストール'］または［'既存のシステムの更新'］を選択します。更新は、SUSE LINUXシステムが、既にインストールされている場合のみ有効です。この場合、［'インストールしたシステムの起動'］で、インストールしたシステムを起動できます。インストールしたシステムの起動に失敗する場合は、重要なシステム設定が壊れている可能性があるため、［'インストールしたシステムの修復'］で、再度、システムをブートできるようにすることがで


*Figure 1.2: 言 語 の 選 択*

きます。SUSE LINUXシステムがインストールされていない状態では、新規インストールだけが実行可能です。図 1.3.「インストールモードの選択」を参照してください。

次のシナリオでは、新規システムのインストール手順を解説します。システムアップデートの詳細な手順については、項2.2.5.「システムのアップデート」に記載されています。システム修復オプションの説明は、章 5. システムの修復を参照してください。

## 1.5 インストールの提案

ハードウェアの検出が完了すると、図 1.4.「提案ウィンドウ」のような提案ウィンドウが表示されます。この画面では、認識されたハードウェアに関する情報、およびインストールとパーティション設定のオプションが多数表示されます。これらの項目を選択し、それぞれのダイアログでの設定を完了すると、常にこの提案ウィンドウに戻ります。この画面の内容は、設定に応じて常に更新されます。それぞれの設定については、この後のセクションで解説します。

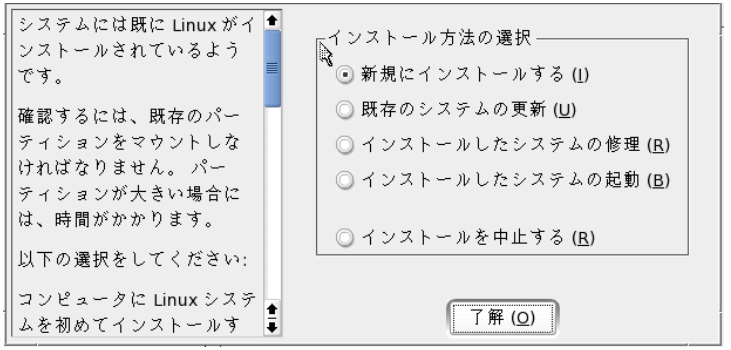


*Figure 1.3: インストールモードの選択*

### 1.5.1 インストールモード

これは、先に選択したインストールモードを変更するために使用します。オプションは項1.4. 「インストールモード」に記載されている内容と同じです。

### 1.5.2 キーボード配列

キーボード配列を選択します。デフォルトでは、この配列は選択言語に対応します。配列の変更を行った後は、Ⓨ、Ⓩ、および特殊文字の入力テストを行い、正しく選択されていることを確認します。ここまでの手順が完了したら、［'次へ'］を選択して、提案ウィンドウに戻ります。

### 1.5.3 マウス

YaSTが自動的にマウスを検出できなかった場合、提案ウィンドウで'マウス'が選択されるまで、Tabキーを数回押します。続いて、Spaceキーを使用して、マウスのタイプを設定するダイアログを開きます。図 1.5. 「マウスのタイプの選択」に示すダイアログボックスが表示されます。

マウスのタイプを選択するには、↑と↓を使用します。ご使用のマウスのタイプについては、マウスに添付のマニュアルを参照してください。マウスのタイプを選択した後は、Alt-Tを使用し、選択を確定する前にデバイスが正しく機能するかテストします。マウスが正しく動作しない場合は、キーボー


*Figure 1.4: 提 案 ウ ィ ン ド ウ*

ドで別のタイプを選択し、再度テストを行います。現在の選択を確定するには、Tabと Enterを使用します。

## 1.5.4 パーティション

通常、YaSTが提案するパーティション設定は適切で、変更せずにそのまま適用することができます。しかし、YaSTを使用して、パーティション設定をカスタマイズすることも可能です。このセクションでは、必要なステップを解説します。

### パーティションのタイプ

どのハードディスクにも、パーティションテーブルがあり、4つのエントリ領域が設けられています。パーティションテーブルのエントリは、基本パーティションまたは拡張パーティションのいずれかに使用されます。ただし、拡張パーティションとして指定できるエントリは、1つだけです。

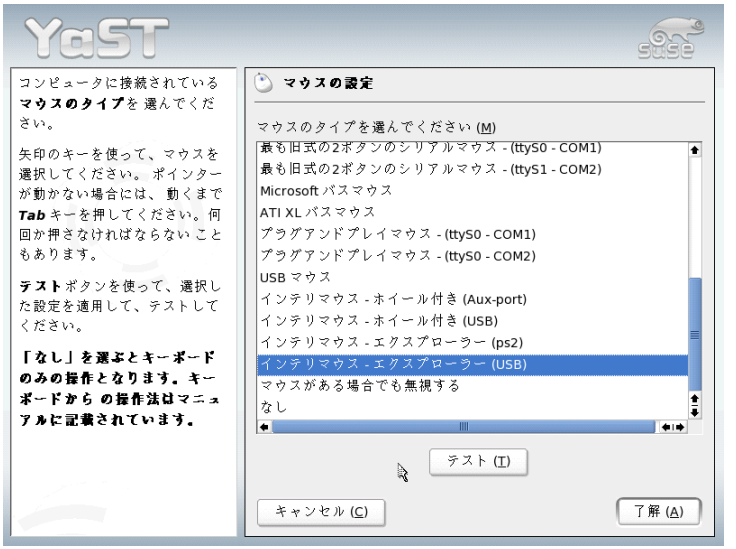


*Figure 1.5: マウスのタイプの選択*

基本パーティションは、単純にシリンダの連続した領域(物理ディスク領域)で構成され、これらのシリンダは、特定のオペレーティングシステムに割り当てられています。パーティションテーブルの制限に伴い、基本パーティションの場合、1台のハードディスクで作成できるパーティションの数が4つに限られます。このような理由から、拡張パーティションが使用されます。拡張パーティションもディスクの連続シリンダから構成されますが、拡張パーティションの場合は、パーティション自体を分割して、*論理*パーティションを作成できます。論理パーティションは、必ずしもパーティションテーブルに存在している必要はありません。つまり、拡張パーティションは論理パーティションのコンテナということになります。

パーティションが4つ以上必要な場合は、4つ目(またはそれ以前)に拡張パーティションを1つ作成します。この拡張パーティションには、残りの空きシリンダ領域全体を使用するのが妥当です。さらに、この拡張パーティションを複数の論理パーティションに区切ります。SCSI、SATA、Firewireなどのディスクで作成可能な論理パーティションは、最大で15個、(E)IDEディスクの場合は、最大63個です。どのタイプのパーティションを使用しても、Linuxへの影響はありません。基本パーティション、論理パーティションのいずれも、正常

に動作します。

**Tip**

**GPTディスクラベル付きのハードディスク**

GPTディスクラベルを使用しているアーキテクチャの場合、基本パーティションの数に制限がありません。したがって、この場合、論理パーティションはありません。

**Tip**

### 必須ディスクスペース

YaSTは、通常、十分なディスク領域を確保した適切なパーティション設定スキーマを提案します。独自のパーティション設定スキーマを実装する場合、以下に示す、システムタイプ別ごとの要件を考慮した推奨値も参照してください。

**最小システム:500MB** グラフィカルインタフェース(X Window System)はインストールしません。これは、使用できるのがコンソールアプリケーションのみであることを示します。また、限られたごく基本的なソフトウェアのみがインストールされます。

**最小システムとグラフィカルインタフェース:700MB** この構成には、X Window Systemと一部のアプリケーションが含まれます。

**標準システム:2.5GB** この構成には、KDEやGNOMEなど、最新のデスクトップ環境が含まれ、OpenOffice.org、NetscapeまたはMozillaなどのような、サイズの大きなアプリケーションスイートにも十分対応できるスペースが確保されています。

作成するパーティションは、使用可能な領域によって異なります。次に基本的なパーティション設定に関するガイドラインを示します。

**4GB以下:** スワップ領域パーティションと、ルートパーティションを1つずつ作成します(/)。この場合、使用可能領域に余裕があれば、通常は独自のパーティションに配置するディレクトリも、ルートパーティションに配置するようにします。

**4GB以上:** スワップパーティション、ルートパーティション(1GB)を各1つ作成し、必要に応じて以下のディレクトリごと各1つのパーティションを作成します。/usr (4GB以上)、/opt (4GB以上)、/var (1GB)。これらのディレクトリを別々のパーティションとして割り当てない場合は、先に提示したディスク領域をルートパーティションに追加します。残りの使用可能領域は、/homeとして使用できます。

ハードウェアによっては、ブートパーティション(/boot)を作成し、ブートメカニズムとLinuxカーネルを配置する方が便利な場合もあります。このパーティションはディスクの先頭に配置し、少なくとも8MB、または1シリンダ分を割り当てるのが妥当です。一般的な規則として、このようなパーティションがYaSTの元々の提案に含まれていた場合、必ず、このパーティションを作成するようにします。この件について確実でない場合は、念のためブートパーティションを作成してください。

また、一部の(特に市販の)プログラムは、独自のデータを/optにインストールすることに注意しなければなりません。このような理由から、/optとして別のパーティションを作成するか、ルートパーティションに十分な大きさを割り当てるようにします。KDEおよびGNOMEも/optにインストールされます。

### YaSTによるパーティション設定

提案ウィンドウで最初にパーティションの項目を選択した場合、YaSTにより、現時点で提案されるパーティション設定を示したパーティション設定ダイアログが表示されます。現在の設定をそのまま適用するか、設定を変更して続行します。また、ここまでの設定をすべて破棄し、最初から設定し直すこともできます。

［'推奨案でパーティションを構成する'］を選択した場合、パーティション設定は提案どおりのまま、変更されません。［'提案を変更して基本的なパーティションの設定をする'］を選択した場合、［'パーティションのエキスパート設定'］が表示されます。この画面では、あらゆるパーティション設定を詳細に渡って調整できます。このダイアログについては、項2.7.5. 「パーティション」で解説されています。この解説の中では、YaSTによって提案されたオリジナル設定を起点として使用しています。

［'カスタムパーティション設定をする'］を選択すると、図 1.7. 「ハードディスクの選択」に示すダイアログが表示されます。システム上の既存のハードディスクを選択する場合は、このリストを使用します。このダイアログで選択されたディスクに、SUSE LINUXがインストールされます。

次のステップでは、ディスク全体を使用してインストールするか( ［'ハードディスクの全体を使う'］)、既存のパーティションがある場合はそのいずれか

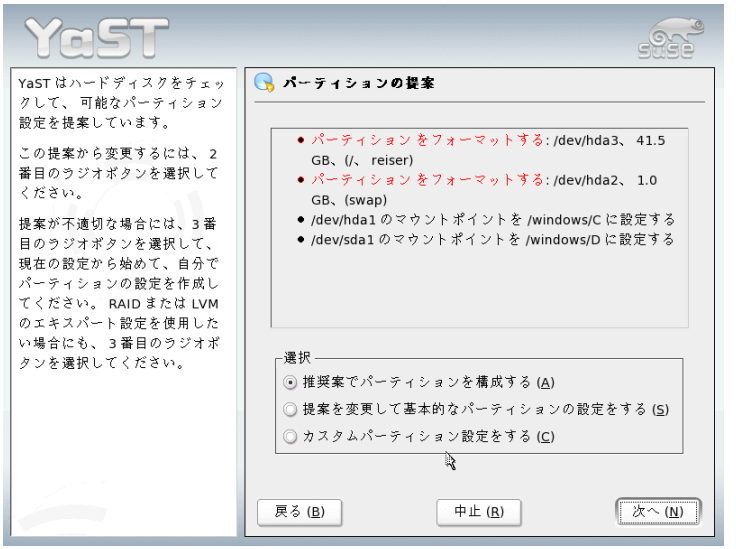


*Figure 1.6: パーティション設定の編集*

を使用してインストールするか、を選択します。ディスク上にWindowsオペレーティングシステムが検出された場合、このパーティションを削除、またはサイズ変更するか確認するメッセージが表示されます。この決定を行う前に、項1.5.4. 「Windowsパーティションのサイズ変更」をお読みください。必要に応じて、この時点で［'パーティションのエキスパート設定'］ダイアログを開き、カスタムパーティションの設定を作成します(項2.7.5. 「パーティション」を参照してください)。

**Warning**

**インストールにハードディスク全体を使用する場合**

［'ハードディスクの全体を使う'］を選択した場合、ディスク上にある既存のデータは、続くインストールプロセス中に完全に消去され、失われます。

**Warning**

インストール中、選択されたソフトウェアに対し、ディスク領域が十分あるかどうか、YaSTにより、チェックされます。不十分な場合、YaSTは、ソフト

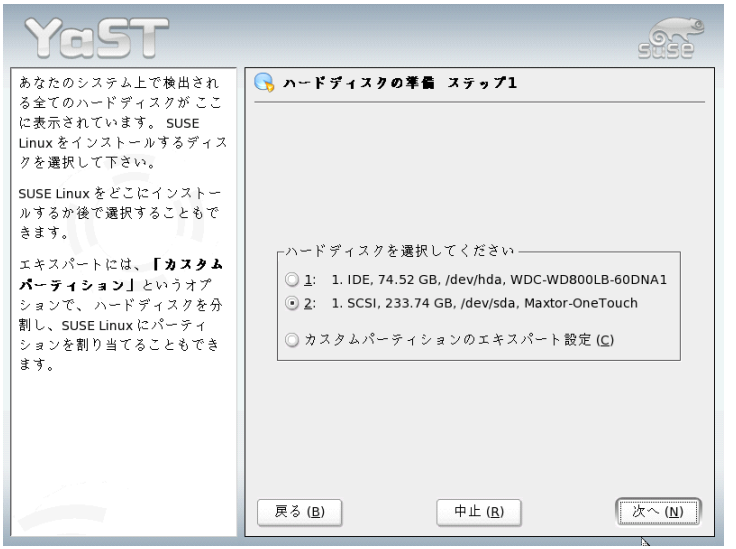


*Figure 1.7: ハードディスクの選択*

ウェアの選択を自動的に変更します。提案ダイアログが表示され、ユーザにこの情報を通知します。ディスクスペースが十分にある限り、YaSTはユーザの設定をそのまま受け入れ、ハードディスク上に設定どおり、パーティションを作成します。

### Windowsパーティションのサイズ変更

Windows FATまたはNTFSパーティションを含むハードディスクがインストール対象として選択された場合、YaSTを使用してこのパーティションを削除または縮小できます。この方法を使用すると、現状ではハードディスクに十分な容量がない場合でもSUSE LINUXをインストールできます。この機能は、ハードディスク全体で1つのWindowsパーティションを構成するハードディスクを選択する場合に特に有用です。Windowsがプリインストールされたコンピュータで、しばしばこの状態が見られます。YaSTが、選択されたハードディスクに十分な空き容量はないが、Windowsパーティションを削除または縮小すれば空き容量を利用できると判断した場合、次の2つのオプションのうち1つを選択するダイアログが表示されます。

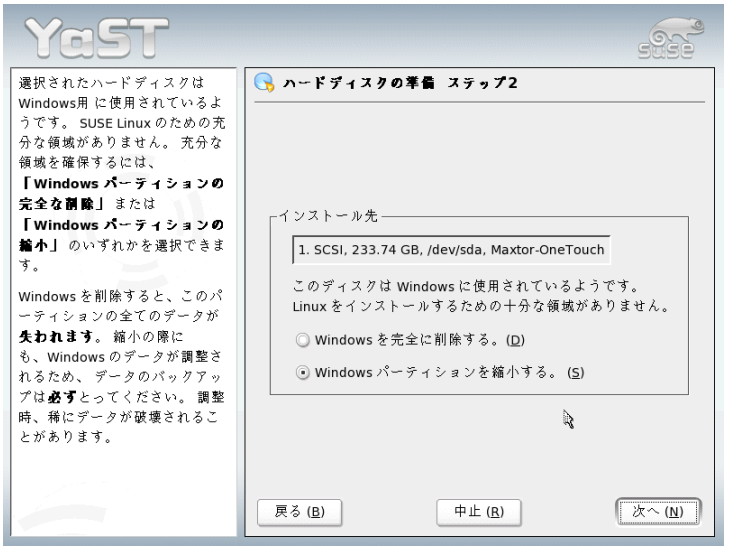


***Figure 1.8:*** *Windowsパーティションで使用可能なオプション*

［'Delete Windows Completely(Windowsを完全に削除)'］を選択した場合、Windowsパーティションには削除マークが付けられ、空き容量はSUSE LINUXのインストールに使用されます。

**Warning**

**Windowsの削除**

Windowsを削除した場合、フォーマットがすぐに開始され、すべてのデータが削除されるためリカバリー不能になります。

**Warning**

Windowsパーティションを縮小するためには、インストールを中断し、縮小したパーティションを準備するためにWindowsをブートします。この手順はFATパーティションでは必須ではありませんが、実行するとサイズ変更プロセスは高速化しより安全になります。これらの手順はNTFSパーティションでは必須です。

**FATファイルシステム** Windowsでは、最初にスキャンディスクを実行し、FATパーティションに断片化され失われたファイルおよびクロ

スリンクがないことを確認します。次に、デフラグを実行しファイルをパーティションの最初に移動します。これによりLinuxでのサイズ変更処理を迅速化します。

Windows用に仮想メモリ設定を最適化し、連続するスワップファイルが仮想メモリと同じ初期(最小)および最大のサイズ制限を使用する場合、他の手順を検討します。このWindows設定では、サイズ変更はスワップファイルを多くの小さな部分に分割し、すべてのFATパーティションに散在させる可能性があります。スワップファイル全体をサイズ変更中に移動する必要があり、プロセスに時間がかかります。したがって、差し当たりこれらのWindowsの最適化を無効にして、サイズ変更が完了後に再度有効にします。

**NTFSファイルシステム** Windowsでは、スキャンディスクとデフラグを実行してファイルをハードディスクの最初に移動します。FATファイルシステムとは異なり、次の手順を実行する必要があります。そうでない場合、NTFSパーティションのサイズ変更はできません。

**Important**

**Windowsのスワップファイルを無効にする**

NTFSファイルシステムで永続的スワップファイルを使用してシステムを運用している場合、このファイルはハードディスクの末尾に格納され、デフラグ後も残ります。そのため、パーティションを十分に縮小することは不可能です。この場合、スワップファイル(Windowsでは仮想メモリ)を一時的に無効化します。パーティションをサイズ変更した後に、仮想メモリを再設定します。

**Important**

これらの準備後に、Linuxパーティションセットアップに戻り、［'Shrink Windows Partition(Windowsパーティションの縮小)'］を選択します。パーティションを簡単に確認した後に、YaSTはWindowsパーティションのサイズ変更を推奨するダイアログを開きます。

最初の棒グラフはWindowsにより現在使用中のディスク容量および利用可能なディスク容量を示します。2番目の棒グラフは、YaSTの提示内容に基づいて、サイズ変更後に割り当てられる容量を示します。図 1.9.「Windowsパーティションのサイズ変更」を参照してください。提示された設定を了承するか、スライダを使用してパーティションサイズ(一定の制限以内で)を変更します。

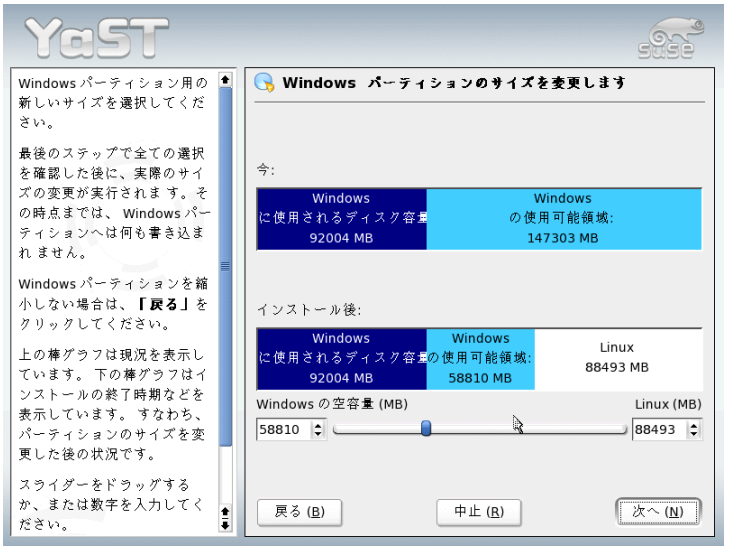


*Figure 1.9: Windowsパーティションのサイズ変更*

このダイアログを［'次へ'］を選択して離れる場合、設定は保存され前のダイアログに戻ります。実際のサイズ変更は後で、ハードディスクがフォーマットされる前に実行されます。

**Important**

**NTFSパーティションにインストールされたWindowsシステム**

デフォルトでは、Windows NT、2000、およびXPのバージョンはNTFSファイルシステムを使用します。FATファイルシステムとは異なり、NTFSファイルシステムはLinuxからしか読み込めません。これは、WindowsファイルをLinuxから読み込めますが、編集できないという意味です。Windowsデータに対して書き込みアクセスが必要で、NTFSファイルシステムが必要ない場合は、FAT32ファイルシステムを用いてWindowsを再インストールしてください。この場合、SUSE LINUXからWindowsデータに対してフルアクセスを持ちます。

**Important**

### 1.5.5 ソフトウェア

SUSE LINUXには、さまざまな用途に適したソフトウェアパッケージが付属しています。必要なパッケージを1つずつ選択するのは手間のかかる作業であるため、SUSE LINUXには、多様なインストールスコープごとに、3つのシステムタイプが用意されています。使用可能なディスク領域によって、YaSTはこれらの事前定義システムを選択し、それを提案ウィンドウに表示します。

**最小システム(特別な目的の場合のみ推奨)**
基本的にこの構成には、コアオペレーティングシステムと各種サービスが含まれますが、グラフィックユーザインタフェースは含まれません。コンピュータの操作に使用できるのは、ASCIIコンソールのみになります。このシステムタイプは、直接的なユーザインタフェースをほとんど必要としないサーバ用として、特に、適しています。

**最小のグラフィカルシステム(GNOMEまたはKDEを除く)**
KDEまたはGNOMEのデスクトップ環境が必要ない場合、またはディスクスペースが十分でない場合は、このシステムタイプをインストールします。この構成では、X Window Systemと基本的なウィンドウマネージャがインストールされます。プログラムで独自のグラフィカルユーザインタフェースを備えている場合、それらのインタフェースを使用することが可能です。オフィスプログラムはインストールされません。

**標準システムとGNOMEおよびオフィススイート**
これは事前設定システムの中でも最大の構成になります。GNOMEデスクトップ環境、GNOMEプログラムのほとんど、さらにオフィスプログラムが含まれます。

**標準システムとKDEおよびオフィススイート**
このシステムには、KDEデスクトップ環境、KDEプログラムのほとんど、さらにオフィスプログラムが含まれます。

提案ウィンドウで［'ソフトウェア'］をクリックし、表示されたダイアログで事前定義されたシステムのいずれかを選択します。ソフトウェアインストールモジュール(パッケージマネージャ)を開始し、インストールスコープを編集するには、［'詳細な選択'］をクリックします。詳細については、図 1.10.「YaSTパッケージマネージャによるソフトウェアのインストールと削除」を参照してください。

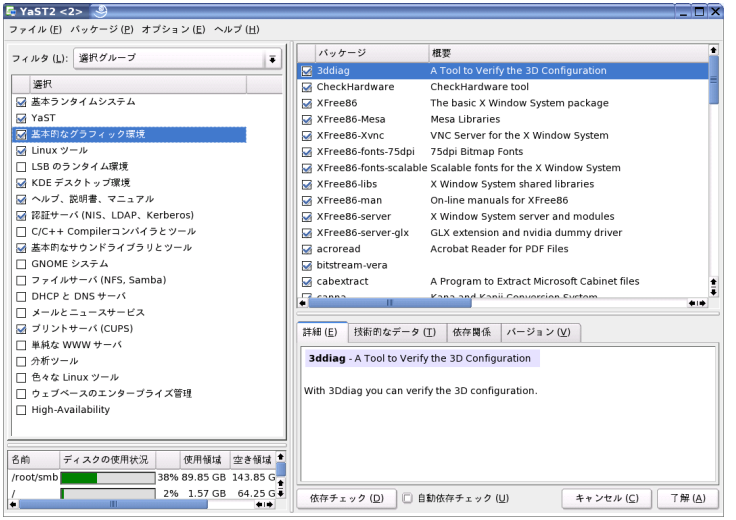


***Figure 1.10:*** *YaSTパッケージマネージャによるソフトウェアのインストールと削除*

### インストールスコープの変更

標準システムをインストールする場合、通常、個別のパッケージを追加、削除する必要はありません。この構成で選択されるソフトウェア群は一般的な要件を満たし、変更の必要はありません。特定の要件がある場合は、パッケージマネージャを用いてこの選択を変更します。パッケージマネージャを使用すると、このタスクは非常に容易になります。このツールには、SUSE LINUXに付属する多数のパッケージの選択作業を簡略化するために用いる、多様なフィルタ条件が用意されています。

フィルタ選択ボックスは、メニューバーの下、左上部にあります。開始後、有効なフィルタは、［'選択'］です。このフィルタにより、マルチメディア、オフィスアプリケーションなど、アプリケーションの用途別にプログラムパッケージがソートされます。これらの分類は、フィルタ選択ボックスの下にリストされています。現在のシステムタイプに含まれているパッケージは、あらかじめ選択されています。個別のチェックボックスをクリックして、インストール対象とする項目全体、グループを選択または選択解除します。

ウィンドウの右側に、現在選択されているパッケージを個別にリストしたテーブルが表示されます。一番左側の列には、各パッケージの現在の状態が表示されています。次に示す2種類のステータスフラグは、インストールに特に関

係するものです。［'インストールする'］(パッケージ名の左にあるボックスがチェックされた状態)と［'インストールしない'］(ボックスはチェックされていない状態)です。個別のソフトウェアパッケージを選択または選択解除するには、希望するステータスになるまでステータスボックスをクリックします。また、それ以外の方法としては、パッケージの行を右クリックしてポップアップメニューを表示し、使用可能なステータス設定をすべてリストします。さらに詳しい情報については、項2.2.1.「ソフトウェアのインストールと削除」に記載されているこのモジュールの詳細説明を参照してください。

### その他のフィルタ

フィルタ選択ボックスをクリックして、その他の使用可能なフィルタを表示します。［'パッケージグループ'］ごとの選択を使用して、インストールすることもできます。このフィルタは内容別にプログラムパッケージをソートし、左側にツリー構造で表示します。ツリーのブランチ部分を展開していくと、パッケージの選択基準がさらに絞り込まれ、右側に表示される関連パッケージリストのパッケージ数も減っていきます。

［'検索'］を使用し、特定のパッケージを検索することもできます。この機能については、項2.2.1.「ソフトウェアのインストールと削除」で詳細に説明しています。

### パッケージの依存関係と競合

ソフトウェアパッケージの組み合せを考慮せず、あらゆるパッケージをインストールできるわけではありません。それぞれのソフトウェアパッケージは、互換性を備えている必要があります。このような互換性がない場合、パッケージが互いに干渉し、競合を引き起こす原因となり、システム全体に影響を与えます。したがって、ソフトウェアパッケージを選択、または選択解除した後、パッケージで検出された未解決の依存関係や競合を示す警告が、ダイアログに表示されることがあります。SUSE LINUXを初めてインストールする場合、または警告が完全に理解できない場合は、項2.2.1.「ソフトウェアのインストールと削除」を確認してください。パッケージマネージャの操作に関する詳細情報および、Linuxでのソフトウェア編成の要約が解説されています。

**Warning**

インストール時に事前選択されるソフトウェア群は、長年にわたる経験を基にしているため、通常、新しいユーザと上級個人ユーザのほとんどのニーズは満たされるはずです。一般的に、ここでの変更は必要ありません。しかし、パッケージのいずれかをあえて追加選択、または選択解除する場合は、その結果について十分考慮する必要があります。特に、あらゆる警告に注意を払い、基本システムのパッケージを選択解除することのないようにしてください。

**Warning**

**ソフトウェア選択の終了**

ソフトウェアの選択をすべて完了し、全パッケージの依存関係と競合が解決されたら、［'了解'］をクリックして、変更を反映し、このモジュールを終了します。インストール中、これらの変更は内部的に保存され、実際のインストールが開始された後、適用されます。

### 1.5.6 ブートの設定

インストール中、YaSTにより、システムのブート設定が提案されます。通常、設定を変更せずに、そのまま適用することができます。しかし、カスタムセットアップが必要な場合、ご使用のシステムに応じ、提案された設定を変更します。

特別なブートフロッピーを使用したブートメカニズムを設定することも可能です。ブート時には、常にそのブートフロッピーをドライブに挿入しておかなければならないという欠点がありますが、既存のブートメカニズムをそのまま活用できるという利点もあります。ただし、他の既存のオペレーティングシステムからもブートできるよう、YaSTでブートローダを設定できるため、改めてフロッピーディスクを作成する必要はありません。この設定を利用するもう1つの可能性は、ハードディスク上のブートメカニズムの位置を変更する場合です。

YaSTによって提案されたブート設定を変更するには、［'ブート'］を選択します。これによりダイアログが表示され、ブートメカニズムに関する多くの詳細を変更できるようになります。詳細については、項8.4.「YaSTを使用するブートローダの設定」を参照してください。経験のあるユーザ以外、ブートメソッドを変更しないようにしてください。

### 1.5.7 タイムゾーン

このダイアログでは、図 1.11.「タイムゾーンの選択」で示すように、［'ハードウェア時計の時間設定'］をローカルタイムか世界協定時間(UTC)のいずれかに指定します。ご使用のコンピュータのハードウェア(BIOS)クロックの設定により、この選択肢は変わります。クロックがGMTに設定されている場合は、UTCに対応していることを意味し、標準時間と夏時間への切替えは、SUSE LINUXが自動的に行います。

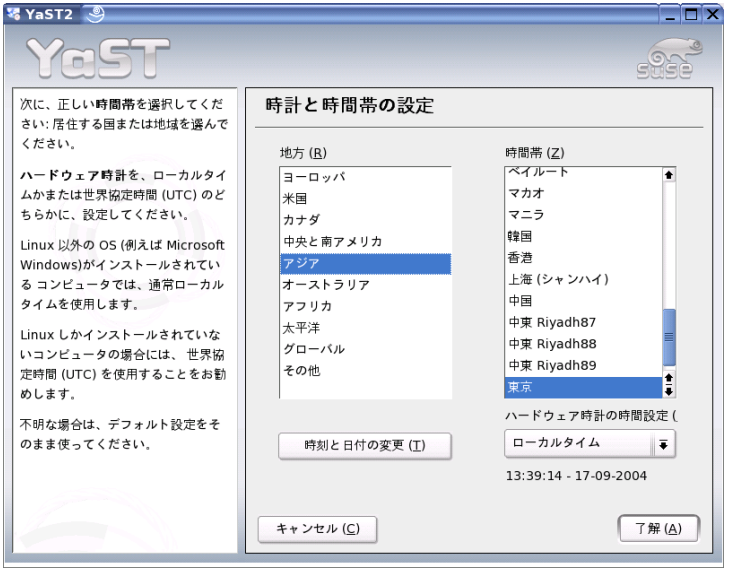


*Figure 1.11: タイムゾーンの選択*

### 1.5.8 言語

言語は項1.3.「言語の選択」で解説したように、インストール開始後、既に選択しました。しかし、ここでは、その設定を変更したり、ご使用のシステムにインストールする追加言語を選択したりできます。このダイアログの上部で、第一言語を選択します。ここで指定する第一言語が、インストール後に有効になります。必要に応じ、それぞれのチェックマークを選択して、キー

ボードとタイムゾーンの設定を調整します。オプションで、［'詳細'］を使用して、rootユーザの言語を設定します。以下、3つのオプションがあります。

**ctypeのみ** ファイル/etc/sysconfig/languageで指定されている、変数LC_CTYPEの値は、rootユーザ用に適用されます。この値により、言語特有の機能呼び出しを指定します。

**yes(はい)** rootユーザは、ローカルユーザと同じ言語設定を使用します。

**no(いいえ)** rootユーザの言語設定は、ここで行う言語設定に影響されません。すべてのロケール変数は、設定されません。

システム管理者の中には、rootアカウントの実行時に、UTF-8マルチ言語のサポートを好まないユーザもいます。そのような場合は、['UTF-8エンコーディングを用いる']のチェックを外します。

ダイアログの下端にあるリストで、インストールする追加言語を選択することができます。このリストで選択された言語すべてに対し、YaSTは、現在選択されているソフトウェアパッケージに、言語特定のパッケージが含まれているどうか確認します。言語特有のパッケージがある場合、これらのパッケージもインストールされます。

['了解']をクリックして設定を完了します。変更を破棄する場合は、['キャンセル']をクリックします。

### 1.5.9 インストールの開始

インストール設定を完了した時点で、提案ウィンドウで['次へ']をクリックし、インストールを開始します。続いて表示されるダイアログでは['はい']をクリックし、開始を確認します。システムのパフォーマンスと選択したソフトウェアによっても異なりますが、通常インストールには15分から30分程度かかります。すべてのパッケージのインストールが完了すると、YaSTは新しいLinuxシステムをブートします。ここまで完了した後、ハードウェアおよびシステムサービスの設定に移ります。

## 1.6 インストールの完了

基本的なシステム設定と選択したソフトウェアパッケージのインストールが完了した後は、システム管理者用アカウント(rootユーザ)のパスワードを指定し

ます。続いて、インターネットアクセスとネットワーク接続を設定することができます。インターネット接続が機能する環境では、インストールの一環として、システムアップデートを実行することが可能です。さらに、ローカルネットワーク内のユーザを集中的に管理するため、認証サーバを設定することもできます。最後に、コンピュータに接続されているハードウェアデバイスの設定を行います。

### 1.6.1 rootのパスワード

rootとは、スーパーユーザ、つまり、システム管理者の名前です。システムでの特定の作業によって、パーミッションを持っていたり、許可されていない場合のある一般ユーザと異なり、rootには、あらゆることを行うための権利が無制限で付与されています。これらの権利には以下のものがあります。システム設定の変更、プログラムのインストール、新規ハードウェアの設定などです。ユーザがパスワードを忘れてしまった場合、システムに関連する他の問題がある場合、rootは支援することができます。rootアカウントは、システム管理、メンテナンス、修復のみに限って使用するのが妥当です。日常的な作業のためにrootでログインすると、次に示すようなリスクが高まります。ただ1度のミスが、多くのシステムファイルの損失を招き、回復不能な障害につながる可能性があります。

rootのパスワードは、確認の目的で図 1.12. 「rootパスワードの設定」で示すように、2度入力しなければなりません。rootのパスワードは、決して忘れないでください。1度入力すると、このパスワードを取得することはできません。

**Warning**

**rootユーザ**

rootユーザには、システムに変更を加えるために必要なすべての権限が付与されています。システムを変更するためのタスクには、rootパスワードが必要になります。このパスワードがなければ、いかなる管理タスクも実行できません。

**Warning**

### 1.6.2 ネットワークの設定

この時点で、ネットワークカード、モデム、ISDNまたはDSLハードウェアなど、外部ネットワークと接続とするためのネットワークデバイスを接続するこ

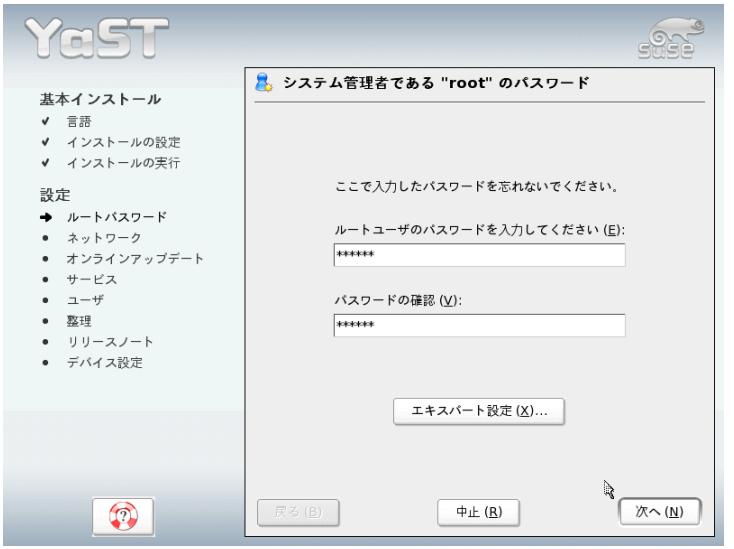


*Figure 1.12: rootパスワードの設定*

とができます。インターネット接続があれば、YaSTによりSUSE LINUXのアップデートを取得し、インストールに組み込むことができるため、 デバイスを接続する場合は、この時点で設定するのが望ましいでしょう。

この段階でネットワークハードウェアを設定する場合は、項22.4.「ネットワーク統合」を参照してください。設定を行わない場合は、［'設定をスキップする'］を選択して［'次へ'］をクリックします。システムのインストールが完了した後、ネットワークハードウェアを設定することもできます。

### 1.6.3 ファイアウォール設定

ネットワークに接続すると、ファイアウォールが設定済みインタフェース上で自動的に開始されます。ファイアウォールの設定がネットワーク設定ダイアログに表示されます。ファイアウォールの設定についての提案は、インタフェースやサービスが設定が変更される度に、自動的に更新されます。変更を、自分の設定に自動的に反映させるには、'変更' → 'ファイアウォール'をクリックします。新規ダイアログでは、ファイアウォールを開始するかどうかを決定します。ファイアウォールを開始しない場合は、適切なオプションを選択

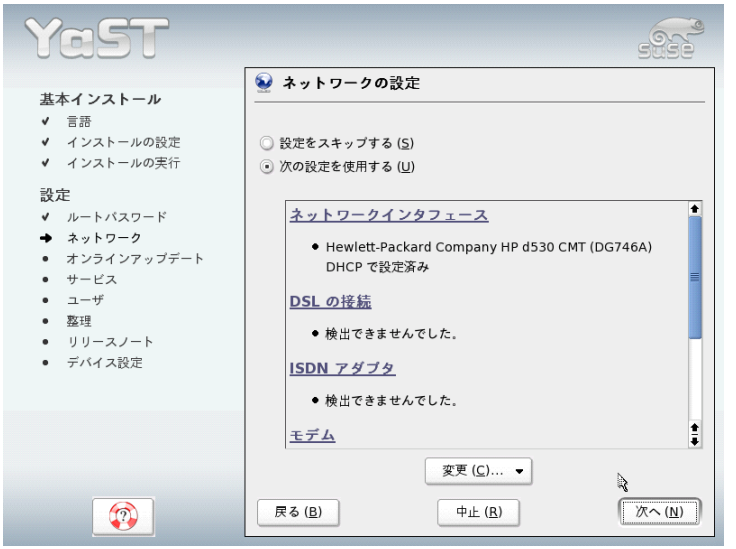


*Figure 1.13: ネットワークデバイスの設定*

し、ダイアログを終了します。ファイアウォールを開始し、設定するには、項34.1.4.「YaSTによる設定」で説明されているダイアログに類似した各ダイアログで、［'次へ'］をクリックします。

### 1.6.4 インターネット接続のテスト

インターネット接続を設定した場合は、この時点でテストできます。テスト用に、YaSTはSUSEサーバとの接続を確立し、ご使用のSUSE LINUXのバージョンに使用できる製品アップデートがないか確認します。更新があれば、インストールに含めることができます。また、最新のリリースノートもダウンロードされます。これらは、インストールの最後に参照できます。

この時点でテストを行わない場合は、［'Skip Test(テストをスキップする)'］を選択し、［'次へ'］をクリックします。ここでスキップすると、製品アップデートおよびリリースノートのダウンロードも省略されます。

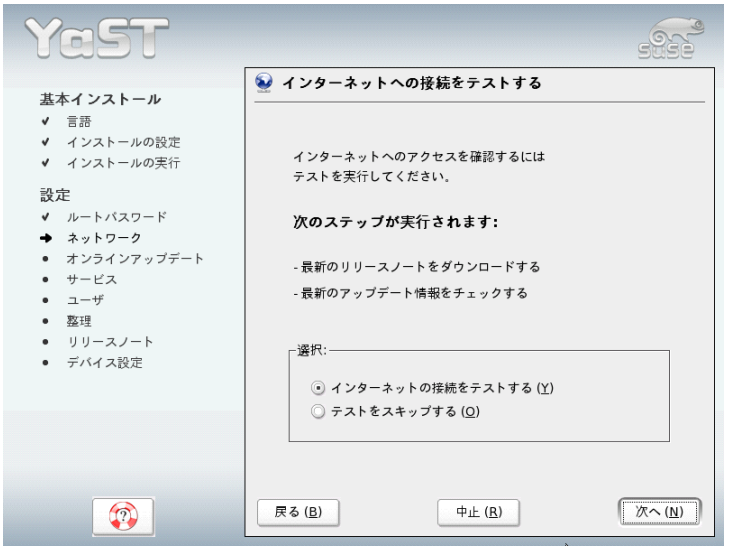


*Figure 1.14: インターネット接続のテスト*

### 1.6.5 ソフトウェアアップデートのロード

YaSTがSUSEのサーバに接続できた場合、YaSTオンラインアップデートを実行するか選択します。サーバ上に利用可能なパッチ付きパッケージがある場合、既知のバグやセキュリティ問題を修正するために、ここでそれらをダウンロードしてインストールします。

**Important**

**ソフトウェアアップデートのダウンロード**

各アップデートのダウンロードには、ある程度時間がかかる場合があります。インターネット接続の帯域幅、アップデートファイルのサイズによって、ダウンロードに要する時間は異なります。

**Important**

ソフトウェアアップデートを直ちに実行するには、['Perform Update Now(アップデートを実行する)']を選択し、['OK']をクリックします。これにより、YaSTのオンラインアップデートダイアログが開き、使用可能な

パッチがあれば、選択およびロード可能なパッチのリストを表示します。このプロセスについての詳細は、項2.2.3. 「YaSTオンラインアップデート」を参照してください。この種のアップデートはインストール後、いつでも実行することができます。この時点でアップデートしない場合は、['アップデートしない']を選択し、['OK']をクリックします。

### 1.6.6 ユーザの認証

これまでのインストールステップで、ネットワークアクセスが正常に設定された場合、システム上に存在するユーザアカウントを管理するために、4種類の方法が使用可能になります。

**ローカルユーザの管理** ユーザはインストールされたホストで、ローカルで管理されます。これはスタンドアロンのワークステーションに向いています。ユーザのデータは、ローカルファイル`/etc/passwd`で管理されます。

**LDAP** ユーザはネットワーク上のすべてのシステムに対し、1台のLDAPサーバ上で集中的に管理されます。

**NIS** ユーザはネットワーク上のすべてのシステムに対し、1台のNISサーバ上で集中的に管理されます。

**Samba** SMB認証は、通常、LinuxとWindowsが混在するネットワークで使用されます。

すべての要件が満たされると、YaSTはユーザの管理メソッドを選択するダイアログを開きます。このツールを図 1.15. 「ユーザの認証」に示します。必要なネットワーク接続が確立されていない場合は、ローカルユーザアカウントを作成します。

### 1.6.7 NISクライアントとしてホストを設定する場合

NISを使用してユーザ管理を実装するには、次のステップでNISクライアントを設定します。このセクションでは、クライアント側の設定のみ解説します。YaSTでNISサーバを設定する方法については、章 25. NISの使用に記載されています。

続くダイアログでは、図 1.16. 「NISクライアントの設定」で示すように、まず始めに、ホストがスタティックなIPアドレスを持っているか、DHCPから


*Figure 1.15: ユーザの認証*

アドレスを取得するかを選択します。DHCPを選択すると、NISドメインまたはNISサーバアドレスを指定することはできません。これらはDHCPサーバにより割り当てられるためです。DHCPに関する詳細は、章 27. DHCPを参照してください。固定IPアドレスを使用する場合は、NISドメインとNISサーバを手動で指定します。

ネットワークでNISサーバのブロードキャストを検索するには、関連するオプションを選択します。また、複数のNISドメインを指定して、デフォルトドメインを設定することもできます。各ドメインごとに['編集']を選択し、複数のサーバアドレスを指定するか、ドメインごとのブロードキャスト機能を有効にします。

クライアントが使用しているサーバを他のネットワークホストから照会できないようにするには、エキスパート設定で['ローカルホストにのみ応答する']を使用します。['ブロークンサーバ']を有効にすると、特権のないポート上のサーバからの応答も受け入れるようになります。詳細な情報については、`ypbind`のマニュアルページを参照してください。


*Figure 1.16: NISクライアントの設定*

## 1.6.8 ローカルユーザアカウントの作成

ユーザの認証に認証サーバを使用しない場合は、ローカルユーザを作成します。ユーザアカウントに関連するあらゆるデータ(名前、ログイン、パスワード、その他)は、インストールしたシステムに格納され、管理も同じシステム上で行われます。

Linuxは、複数のユーザが同じシステムで、同時に作業することが可能なオペレーティングシステムです。各ユーザには、システムにログインするためのユーザアカウントが必要になります。ユーザアカウントを使用することにより、システムのセキュリティは大幅に向上します。たとえば、システムが正常に機能するために必要なファイルを、一般ユーザが変更したり、削除したりすることはできません。さらに、ユーザの個人データを他のユーザが変更、表示、改ざんすることも不可能です。ユーザは自分の作業環境をセットアップすることが可能です。そして、いつログインしても、それらが変更されていることはありません。

図 1.17. 「ユーザ名とパスワードの入力」に示すように、ダイアログを使用してユーザアカウントを作成できます。名前(ファーストネーム)と姓名(ファミ


*Figure 1.17: ユーザ名とパスワードの入力*

リネーム)を入力した後、ユーザ名(login)を指定します。['推奨ユーザ名']をクリックすると、システムはユーザ名を自動生成します。

最後にユーザのパスワードを入力します。確認用に(入力内容が誤っていないことを再確認する目的で)、パスワードを再入力します。ユーザ名には、ユーザを識別し、このIDを確認するために使用するパスワードを、システムに指定する働きがあります。

**Warning**

**ユーザ名とパスワード**

ユーザ名とパスワードは、システムにログインする際、毎回必要になるため、どちらも記憶しておいてください。

**Warning**

安全に運用するため、パスワードは5文字から8文字の長さで指定してます。パスワードに指定できる最大文字数は128字です。ただし、特別なセキュリティモジュールをロードしていない限り、パスワードを識別するために使用されるのは、最初の8字のみです。パスワードでは、大文字小文字が区別されます。

ウムラウトなどの特殊文字は使用できません。他の特殊文字(7ビットASCII)と数字は使用できます。
ローカルユーザは、以下に示す2つの追加オプションを使用できます。

**'Receive System Messages via E-Mail(電子メール経由でのシステムメッセージの受信)'**
このボックスにチェックを入れると、システムサービスによって作成されたメッセージがユーザに送信されます。これらのメッセージは通常、root、つまりシステム管理者にのみ、送信されます。このオプションは、主に使用するアカウントに設定すると便利です。rootを使用してログインするのは、特殊な場合に限るよう推奨されているためです。

**'自動ログイン'** このオプションを使用できるのは、デフォルトのデスクトップがKDEの場合に限られます。システムの起動時に、現在のユーザは自動的にシステムにログインします。この機能は、主に、コンピュータを使用するユーザが1人に限定されている場合、有用です。

**Warning**

**自動ログイン**
自動ログインが有効になっている場合、システムは認証をまったく行うことなく、ユーザのデスクトップをそのまま開始します。システム上に機密データを格納していて、他のユーザがコンピュータにアクセスできる場合は、このオプションを有効にすべきではありません。

**Warning**

### 1.6.9 リリースノート

ユーザ認証のセットアップを完了した後、YaSTはリリースノートを表示します。リリースノートには、マニュアルの印刷時には利用できなかった、最新の重要情報が含まれているため確認するようにしてください。アップデートパッケージをインストールした場合は、SUSEのサーバから取得した、最新のリリースノートが利用できます。

## 1.7 ハードウェア設定

インストールの最後に、グラフィックカードやプリンタ、サウンドカードなど、システムに接続されているハードウェアコンポーネントの設定ダイアロ

グが、YaSTにより表示されます。個別のコンポーネントをクリックすると、ハードウェア設定が開始されます。多くの場合、デバイスはYaSTにより、自動的に検出され、設定されます。


*Figure 1.18:* システムコンポーネントの設定

すべての周辺デバイスの設定を省略し、後で設定することもできます。ただし、グラフィックカードの設定は、直ちに行うのが妥当です。YaSTが自動設定したディスプレイの設定は、通常、適用して問題ありません。ただし、解像度、色深度、その他のグラフィック機能の設定については好みが分かれる点でもあるため、設定はユーザごとにまったく異なることがあります。これらの設定を変更するには、['グラフィックカード']を選択します。この設定については、項11.1. 「SaX2によるX11の設定」を参照してください。YaSTが設定データの書き込みを完了した後、最後のダイアログに'終了'が表示され、SUSE LINUXのインストールが完了します。

## 1.8 グラフィカルログイン

これでSUSE LINUXがインストールされました。ローカルユーザの管理モ

ジュールで自動ログインを有効にした場合は、ログインを省略して開始します。有効にしなかった場合は、画面上にグラフィカルログインが表示されます。図 1.19. 「KDMのログイン画面」に示します。システムにログイン用ユーザ名と対応するパスワードを入力します。

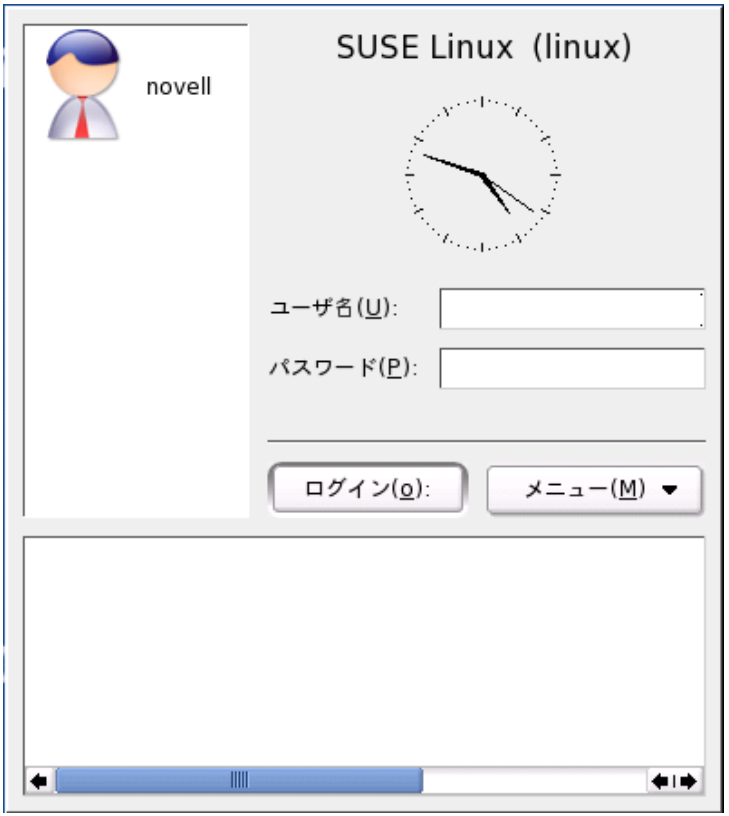


*Figure 1.19: KDMのログイン画面*

# YaSTでのシステム設定

インストールに使用されるセットアップツールのYaSTは、SUSE LINUXの設定ツールでもあります。この章では、YaSTを使用してシステムを設定する方法について説明します。この章で説明する内容には、ハードウェア、グラフィカルユーザインタフェース、インターネットアクセス、セキュリティ設定、ユーザー管理、ソフトウェアのインストール、システムの更新、およびシステム情報のほとんどが含まれます。この章では、YaSTをテキストモードで使用する方法についても説明します。

YaSTでのシステム設定は、さまざまなYaSTモジュールを使用して実行されます。ハードウェアプラットフォームおよびインストール済みのソフトウェアに応じて、YaSTがインストールされたシステムへのアクセス方法は異なります。

KDEまたはGNOMEでは、SUSEメニュー(‘システム’ → ‘YaST’の順に選択)から、YaSTコントロールセンターを起動します。さらに、個々のYaST設定モジュールは、KDEコントロールセンターに統合されています。YaSTを起動する前に、ルートパスワードの入力を求めるプロンプトが表示されます。これは、YaSTがシステムファイルを変更するために、システム管理者のアクセス権が必要なためです。

コマンドラインからYaSTを起動するには、コマンド「su」(rootユーザに変更するため)と入力してから、「yast2」と入力します。YaSTのテキストバージョンを起動するには、「yast2」ではなく、「yast」と入力します。また、「yast」コマンドを使用すると、仮想コンソールの1つからプログラムを起動することもできます。

**Tip**

YaSTの言語を変更するには、YaSTコントロールセンターで、‘システム’ → ‘Select Language (言語選択)’の順に選択します。言語を選択した後、YaSTコントロールセンターを終了し、システムからログアウトしてから再度ログインします。次回YaSTを起動したときから、新しい言語設定が有効になります。

**Tip**

独自のディスプレイデバイスをサポートしないハードウェアプラットフォームの場合、または他のホストをリモート管理する場合は、YaSTをリモートで実行します。最初に、YaSTを表示するホスト上のコンソールを開き、「ssh -X root@<system-to-configure>」コマンドを入力して、rootを設定するためにシステムにログインし、Xサーバ出力を自分の端末にリダイレクトします。SSHログインが成功したら「yast2」と入力して、グラフィカルモードでYaSTを起動します。

他のシステム上で、YaSTをテキストモードで起動するには、「ssh root@<system-to-configure>」コマンドを使用して接続を開きます。次に、「yast」と入力してYaSTを起動します。

## 2.1 YaSTコントロールセンター

グラフィカルモードでYaSTを起動する場合、図 2.1. 「YaSTコントロールセン

ター」に示すように、YaSTコントロールセンターが開きます。左側のフレームには、［'ソフトウェア'］、［'ハードウェア'］、［'システム'］、［'ネットワーク装置'］、［'ネットワークサービス'］、［'セキュリティとユーザ'］、［'システム'］、および［'その他'］のカテゴリが含まれています。アイコンをクリックすると、右側のフレームにコンテンツが表示されます。次に、目的の要素を選択します。たとえば、［'ハードウェア'］を選択し［'サウンド'］をクリックすると、右側にサウンドカードの設定ダイアログが開きます。各項目を設定するには、通常複数の処理を実行する必要があります。［'次へ'］をクリックして、次の処理手順に進みます。

ほとんどのモジュールでは、左側のフレームにヘルプテキストが表示されます。内容は、必要なエントリについての説明です。ヘルプのフレームなしでモジュールのヘルプを表示するには、F1を押すか、メニューの［'ヘルプ'］を選択します。必要な設定を行った後、設定ダイアログの最後で［'完了'］をクリックして処理を完了します。この時点で設定が保存されます。

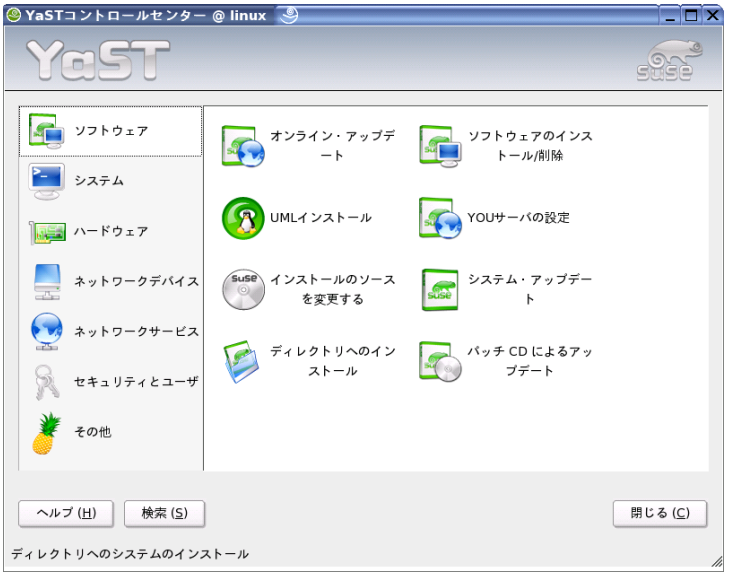


*Figure 2.1: YaSTコントロールセンター*

## 2.2 ソフトウェア

### 2.2.1 ソフトウェアのインストールと削除

このモジュールでは、使用中のマシンに対する、ソフトウェアのインスール、アンインストール、および更新を行います。Linuxでは、ソフトウェアはパッケージの形で用意されています。通常、パッケージにはプログラムに必要なものがすべて含まれています。つまりプログラム自身、設定ファイル、およびマニュアルが含まれています。パッケージにはプログラムのソースファイルが含まれており、通常はソースファイルも使用できます。ソースファイルはプログラムを実行するためには必要ありませんが、プログラムのカスタムバージョンをコンパイルするには、ソースをインストールします。

一部のパッケージは他のパッケージに依存しています。つまり、パッケージの一部のソフトウェアは、他のパッケージもインストールされている場合にのみ適切に動作します。さらに一部のパッケージは、他の特定のパッケージがインストールされていないとインストールできません。インストールルーチンで特定のツールが必要なためです。したがって、これらのパッケージは正しい順序でインストールする必要があります。一部のパッケージは、同一または類似する機能を持っています。これらのパッケージが同じシステムリソースを使用する場合は、同時にインストールしないでください(パッケージの競合)。パッケージの依存関係と競合は複数のパッケージ間で発生し、時にはかなり複雑になることもあります。円滑なインストール処理のために、特定のパッケージバージョンが必要な場合があるという事実が、より複雑さを増します。

これらの要因のすべてを、ソフトウェアのインストール、アンインストール、および更新を実行するときに考慮する必要があります。YaSTには、この問題を考慮にいれた非常に効果的なツールが備えられています。それは通常パッケージマネージャと呼ばれる、ソフトウェアインストールモジュールです。パッケージマネージャが起動すると、システムを検査し、インストール済みのパッケージを表示します。インストールを行う追加のパッケージを選択した場合、パッケージマネージャは自動的に依存関係を確認し、必要な他のパッケージを選択します(依存関係の解決)。競合するパッケージを選択した場合、パッケージマネージャは競合を示し、問題を解決するための提案を行います(競合の解決)。他のインストール済みのパッケージで必要とされるパッケージが削除としてマーク付けされた場合、パッケージマネージャは警告メッセージと、詳細な情報および代替の解決策を表示します。

これらの純粋に技術的な側面だけでなく、パッケージマネージャでは、SUSE LINUXに含まれる多様なパッケージのわかりやすい概要が提供されます。パッケージは対象別に配置され、これらのグループの表示は、適切なフィルタを使用して制限されます。

### パッケージマネージャ

パッケージマネージャを使用して、システムで選択されたソフトウェアを変更するには、YaSTコントロールセンターの［'Install or Remove Software (ソフトウェアのインストールまたは削除)'］を選択します。図 2.2.「YaSTパッケージマネージャ」に、パッケージマネージャのダイアログウィンドウを示します。ウィンドウはさまざまなフレームにより構成されます。領域を区切るラインをクリックして移動すると、フレームのサイズを変更できます。各フレームの内容と使用目的については後で説明します。

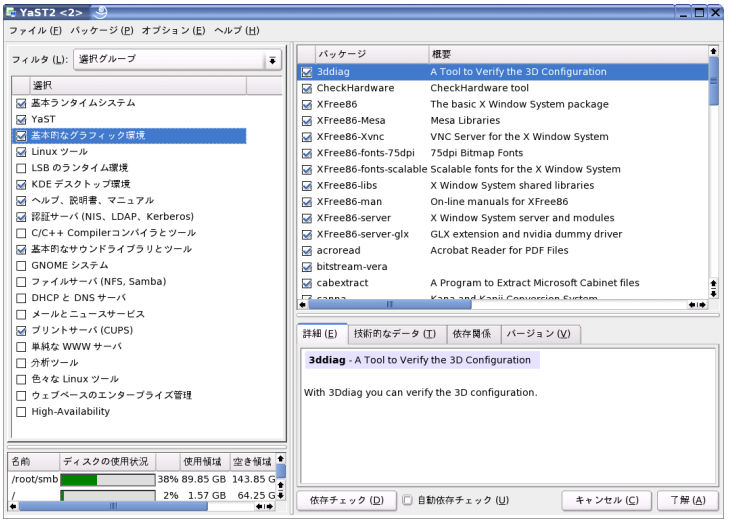


*Figure 2.2: YaSTパッケージマネージャ*

### フィルタウィンドウ

パッケージマネージャには、さまざまなフィルタが備えられています。それによりパッケージをカテゴリごとに整列したり、表示するパッケージ数を制限できます。フィルタウィンドウはメニューバーの左下にあります。フィルタウィンドウを使用して、さまざまなフィルタ方法を制御し表示します。最上部にあるフィルタ選択ボックスで、フィルタウィンドウの下部に表示する項目を決定します。フィルタ選択ボックスをクリックして、使用可能なフィルタのリストから任意のフィルタを選択します。

**選択グループフィルタ**　起動時には、［'選択グループ'］フィルタが有効になっています。このフィルタは、マルチメディアやオフィスアプリケーションなどのアプリケーションの目的に従って、プログラムパッケージをグループ化します。フィルタ選択ボックスには、［'選択グループ'］フィルタのさまざまなグループがリストされています。システムに既にインストールされているパッケージは既に選択された状態になっています。行の先頭にあるステータスボックスをクリックすると、選択項目のステータスフラグが切り替わります。選択項目を右クリックして直接ステータスを選択すると、コンテキストメニューを使用できます。右側の個々のパッケージウィンドウには、パッケージのリストが表示されます。パッケージのリストには、現在の選択項目、有効にする選択項目、除外する選択項目の個々のパッケージが含まれます。

**パッケージグループフィルタ**　［'パッケージグループ'］フィルタは、多様なパッケージのより技術的な概要を提供します。これはSUSE LINUXのパッケージ構造に精通しているユーザに適しています。このフィルタはプログラムパッケージを対象ごとにソートします。対象には、左側のツリー構造にある、アプリケーション、開発、およびハードウェアなどがあります。分岐を展開するほど選択項目は特定化され、右側に表示される個々のパッケージウィンドウに表示されるパッケージが少なくなります。

さらに、このフィルタは、分類することなくすべてのパッケージをアルファベット順に表示することが可能です。これを行うには、最上位レベルで、［'zzz全て'］を選択します。SUSE LINUXには多くのパッケージが含まれているため、この長いリストを表示するには時間がかかる場合があります。

**検索機能**　［'検索'］機能を使用することは、特定のパッケージを見つけるための最も簡単な方法です。さまざまな検索条件を指定することにより、個々のパッケージウィンドウに1つのパッケージだけを表示して、フィルタを制限することもできます。検索文字列を入力し、チェックボックスを使用して対象文字列を検索する場所(名前で、説明で、パッケージ依存関係内で)を決定します。熟練したユーザは、ワイルドカードと正規表現を使用して特別な検索パターンを設定したり、［'提供する機能'］および［'依存する機能'］フィールドに指定した項目によって、パッケージの依存関係を検索できます。たとえばこの機能は、どのパッケージに特定のライブラリが含まれているかを判別するために使用できます。

**Tip**

**クイックサーチ**

［‘検索’］フィルタに加えて、パッケージマネージャのリストすべてにクイックサーチ機能があります。1文字入力すると、入力した文字で始まる、リスト内の最初のパッケージにカーソルが移動します。カーソルはリスト内になければなりません(リストをクリックする)。

**Tip**

**言語** SUSE LINUXの一部のパッケージでは、言語固有のパッケージを使用できます。このパッケージは、プログラムのユーザインターフェース、マニュアルで使用されるテキストが翻訳され、フォントも変更されています。このフィルタでは、左側のウィンドウに、SUSE LINUXによりサポートされるすべての言語のリストが表示されます。これらのうちの1つを選択すると、右側のウィンドウに、選択した言語で使用可能なすべてのパッケージが表示されます。これらの中で、現在のソフトウェア選択にあてはまるすべてのパッケージに、自動的にインストール用のタグが付けられます。

**Note**

言語が指定されたパッケージは他のパッケージに依存するため、パッケージマネージャは状況によって、インストールする追加のパッケージを選択する場合があります。

**Note**

**インストール概要** インストール、更新、または削除するパッケージを選択した後に、フィルタセレクションを使用してインストール概要を表示します。これにより、［‘了解’］をクリックした場合にパッケージに生じる変更点が表示されます。左側のチェックボックスを使用してパッケージをフィルタし、個々のパッケージウィンドウを表示します。たとえば、どのパッケージが既にインストールされているかを確認するには、パッケージマネージャを起動して、［‘保持’］を除くすべてのチェックボックスを無効にします。

通常、個々のパッケージウィンドウのパッケージステータスは変更できます。ただし、変更したパッケージは検索条件に当てはまらなくなる可能性があります。そのようなパッケージをリストから削除するには、［‘リストの更新’］を使用してリストを更新します。

### 個々のパッケージウィンドウ

前述のように、個々のパッケージのリストが、個々のパッケージウィンドウの右側に表示されます。このリストの内容は現在選択されているフィルタにより決定されます。たとえば、［'選択'］フィルタが選択されている場合、個々のパッケージウィンドウは現在選択されているすべてのパッケージを表示します。

パッケージマネージャでは、各パッケージは、パッケージで実行する事柄を決定するステータスを持ちます。ステータスには、「インストール」や「削除」などがあります。このステータスは行の先頭にあるステータスボックス内に記号で表示されます。項目を右クリックしたときに表示されるメニューから、該当のステータスをクリックまたは選択することにより、ステータスを切り替えます。状況によっては、いくつかの潜在的なステータスフラグを選択できません。たとえば、まだインストールしていないパッケージに、「削除」フラグを設定することはできません。使用可能なステータスフラグを表示するには、'ヘルプ' → 'シンボル'の順に選択します。

パッケージマネージャには、次のパッケージステータスフラグがあります。

**インストールしない** このパッケージはインストールされず、今後もインストールされません。

**インストールする** このパッケージはインストールされていませんが、今後インストールされます。

**保持** このパッケージは既にインストールされていて、今後変更されません。

**アップデート** このパッケージは既にインストールされていて、インストールメディアにあるバージョンにより置き換えられます。

**削除** このパッケージは既にインストールされていて、今後削除されます。

**禁止: インストールを禁止する** このパッケージはインストールされず、決してインストールされません。インストールメディアのどこにも存在しないかのように扱われます。依存関係を解決するために、パッケージが自動的に選択される場合、パッケージを「禁止」と設定することにより防ぐことができます。ただし、手動で解決する(依存関係チェック)必要がある不整合が発生する可能性があります。したがって、「禁止」は主に上級ユーザ向けです。

**保護** このパッケージはインストールされていますが、編集しないでください。サードパーティ製のパッケージ(SUSEの署名がないパッケージ)には、自動的にこのステータスが割り当てられ、インストールメディアに

存在する最新のバージョンにより上書きされることを防ぎます。これにより、手動で解決する必要があるパッケージの競合が発生する可能性があります。

**Automatic Installation(自動インストール)**
このパッケージは、他のパッケージに必要(パッケージ依存関係の解決)なため、インストールするために自動的に選択されます。この種のパッケージを選択解除するには、「禁止」ステータスを使用する必要があります。

**Automatic Update(自動アップデート)**
このパッケージは既にインストールされています。ただし、他のパッケージがこのパッケージのより新しいバージョンを必要とするため、インストール済みのバージョンは自動的にアップデートされます。

**Delete Automatically(自動削除)** このパッケージは既にインストールされていますが、存在するパッケージの競合のために、このパッケージを削除する必要があります。たとえば、現在のパッケージが異なるパッケージにより置き換えられた場合が考えられます。

**Automatic Installation (after selection)(自動インストール(選択後))**
このパッケージはインストールするために自動的に選択されました。「マルチメディア」または「開発」など、定義済みの選択の一部であるためです。

**Automatic Update (after selection)(自動アップデート(選択後))**
このパッケージは既にインストールされましたが、インストールメディアにより新しいバージョンが存在します。このパッケージは、「マルチメディア」または「開発」など、定義済みの選択の一部で、アップデートのために選択され、自動的にアップデートされます。

**Delete Automatically (after selection)(自動削除(選択後))**
このパッケージは既にインストールされていますが、定義済みの選択(「マルチメディア」または「開発」など)では、このパッケージが削除されている必要があります。この状況は頻繁に発生しません。

さらに、パッケージのソースをインストールするかどうかを決定します。この情報は、現在のパッケージステータスを補完します。マウスを使用して切り替えたり、コンテキストメニューから直接選択することはできません。代わりに、パッケージ行の最後のチェックボックスにより、ソースパッケージの選択が可能です。このオプションは、［'パッケージ'］からもアクセス可能です。

**ソースをインストールする** ソースコードもインストールします。

**ソースをインストールしない** ソースはインストールされません。

個々のパッケージウィンドウのさまざまなパッケージに使用されるフォントカラーは、追加の情報を提供します。インストールメディア上にあるより新しいバージョンが使用できるインストール済みのパッケージは、青で表示されます。インストールメディア上にあるパッケージのバージョン番号がインストール済みのパッケージよりも新しい場合、赤で表示されます。ただし、パッケージのバージョン番号は、新しいバージョン番号が大きくなるとは限らないため、バージョン情報は正しくない可能性がありますが、問題を引き起こすパッケージを示すには十分なはずです。必要に応じて、情報ウィンドウのバージョン番号を確認します。

### 情報ウィンドウ

フレームの右下にあるタブは、選択されたパッケージについてのさまざまな情報を提供します。選択されたパッケージの説明が自動的にアクティブになります。他のタブをクリックして、技術的なデータ(パッケージのサイズ、グループなど)、依存する他のパッケージのリスト、またはバージョン情報を表示します。

### リソースウィンドウ

ソフトウェアの選択中、左下のリソースウィンドウには、すべてのマウントされたファイルシステムの使用方法を事前に表示します。配色されたバーグラフが選択ごとに上昇します。緑の状態は、十分な容量があることを示します。ディスク容量の限界に近づくと、バーの色が次第に赤くなります。インストールするパッケージを選択しすぎると、警告が表示されます。

### メニューバー

ウィンドウの左上にあるメニューバーから、前述のほとんどの機能にアクセスできます。さらにメニューバーには次の4つのメニューが含まれます。

**ファイル** 'ファイル' → 'エクスポート'の順に選択して、インストール済みのパッケージすべてのリストをテキストファイルに保存します。後で、または他のシステム上で、特定のインストールスコープをレプリケートする場合は、この処理をお勧めします。この方法で生成されたファイルは、［'インポート'］を使用してインポートし、保存されたパッケージと

同じパッケージ選択を生成できます。どちらの場合でも、ファイルの位置を指定するか、推奨を選択します。

パッケージ選択の変更を保存せずにパッケージマネージャを終了するには、［'保存しないで終了する'］をクリックします。変更を保存するには、［'保存して終了する'］を選択します。この場合、すべての変更が適応されてプログラムが終了します。

**パッケージ** ［'パッケージ'］メニューの項目は、個々のパッケージウィンドウに現在選択されているパッケージを常に参照します。すべてのステータスフラグが表示されますが、現在のパッケージに対して使用可能なステータスフラグだけを選択できます。チェックボックスを使用してパッケージのソースをインストールするかどうかを決定します。［'このリストの全て'］は、すべてのパッケージステータスフラグをリストするサブメニューを開きます。ただし、これらは現在のパッケージだけでなく、このリスト内のすべてのパッケージに影響を与えます。

**オプション** ［'オプション'］メニューには、パッケージの依存関係と競合を処理するためのオプションが用意されています。インストールするパッケージを手動で選択した場合、［'自動パッケージ変更を表示する'］をクリックして、パッケージマネージャが依存関係を解決するために自動的に選択したパッケージマネージャのリストを表示します。解決できないパッケージ競合がまだある場合、警告と推奨される解決策が表示されます。

パッケージの競合を［'無視する'］に設定した場合、この情報はシステムに永続的に保存されます。それ以外の場合、パッケージマネージャを起動するたびに、同じパッケージを［'無視する'］に設定する必要があります。依存関係を無視しない場合は、［'無視している依存の競合をリセットする'］をクリックします。

**ヘルプ** 'ヘルプ' → '概要'の順に選択して、パッケージマネージャの機能についての簡単な説明を表示します。さまざまなパッケージフラグについての詳細は、［'シンボル'］をクリックすると表示されます。マウスなしでプログラムを操作するには、［'キー'］をクリックしてショートカットのリストを表示します。

### 依存チェック

［'依存チェック'］および［'自動依存チェック'］は、情報ウィンドウの下部にあります。［'依存チェック'］をクリックすると、パッケージマネージャは、現在のパッケージ選択により解決していないパッケージの依存関係または競合

が発生していないかどうかをチェックします。解決していない依存関係がある場合、必要となる追加のパッケージが自動的に選択されます。パッケージの競合の場合、パッケージマネージャは競合を示すダイアログを開き、問題を解決するためのさまざまなオプションを提供します。

［'自動依存チェック'］を有効にした場合、パッケージのステータスを変更したときに、必ず自動チェックが行われます。これは便利な機能です。パッケージ選択の整合性が永続的に監視されるためです。ただし、このプロセスはリソースを消費し、パッケージマネージャの動作が遅くなります。この理由により、デフォルトでは自動依存チェックは有効ではありません。どちらの場合でも、整合性の確認は［'了解'］をクリックして選択を確定した場合に実行されます。

次の例では、sendmailおよびpostfixは同時にインストールされません。図 2.3. 「パッケージマネージャの競合管理」に、どちらをインストールするのかの決定を要求する、競合メッセージを示します。postfixは既にインストールされています。選択肢としては、sendmailのインストールを無効にする、postfixを削除する、危険を承知で競合メッセージを無視する、があります。

**Warning**

**パッケージの競合の処理**

パッケージの競合を処理する場合は、YaSTの提案に従うようにお勧めします。提案を受け入れなかった場合、システムの安定性と機能性が存在する競合により失われる可能性があります。

**Warning**

## 2.2.2 インストールソースの変更

YaSTは、さまざまな異なるインストールソースを管理できます。用途に応じて、インストール目的または更新目的として選択することができます。このモジュールを起動すると、以前に登録したソースすべてのリストが表示されます。CDからの通常のインストールが終了すると、インストールCDのみがリストされます。［'追加'］をクリックして、このリストにある追加のソースを含めます。CDまたはDVDなどのリムーバブルメディアと同様に、NFSおよびFTPサーバなどのネットワークソースも追加できます。ローカルハードディスク上のディレクトリもインストールメディアとして選択できます。詳細については、YaSTヘルプテキストを参照してください。

登録されたソースはすべて、リストの最初の列に有効状態が表示されます。［'Activate or Deactivate(有効化または無効化)'］をクリックして、個々のイン

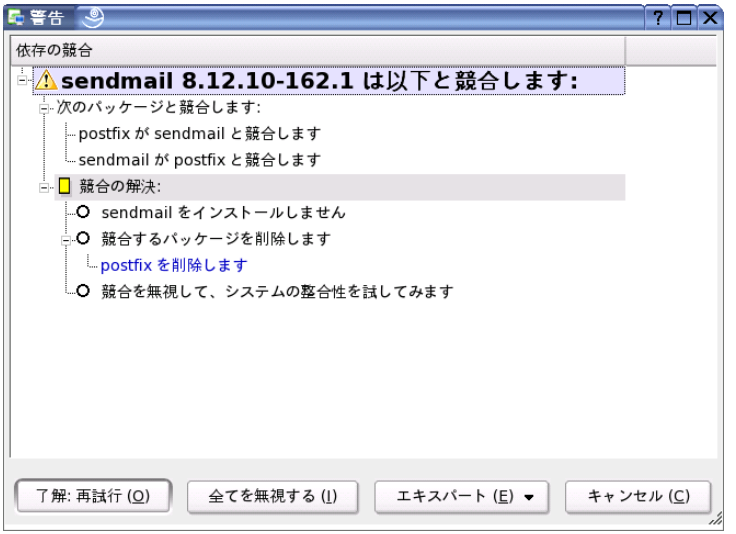


*Figure 2.3: パッケージマネージャの競合管理*

ストールソースを有効化または無効化します。ソフトウェアパッケージのインストールまたはアップデート中に、YaSTは、有効化されたインストールソースのリストから適切なエントリを選択します。［'閉じる'］をクリックしてモジュールを終了した時点で、現在の設定が保存され、設定モジュールの［'ソフトウェアのインストール／削除'］および［'System Update(システム更新)'］に適応されます。

### 2.2.3 YaSTオンラインアップデート

YaSTオンラインアップデート(YOU)は、重要なアップデートと改善のインストールを可能にします。これらのパッチは、SUSEのFTPサーバおよびさまざまなミラーサーバ上でダウンロードできます。

［'インストールソース'］で、さまざまなサーバの1つを選択します。サーバを選択したら、サーバのURLが編集可能な入力フィールドにコピーされます。`file:/my/path`または、`/my/path`の形式でローカルなURLも指定できます。［'新規サーバ'］を使用してサーバを追加した既存のリストを展開しま

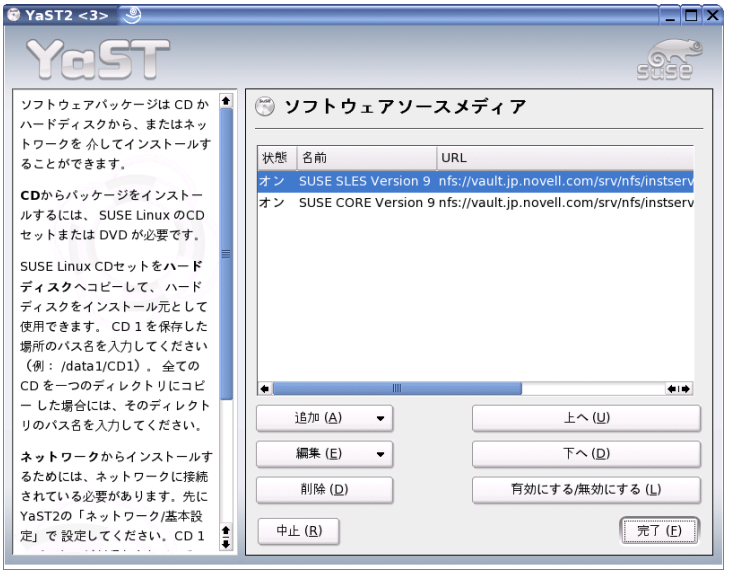


*Figure 2.4: インストールソースの変更*

す。［‘サーバの編集’］をクリックして、現在選択されているサーバの設定を編集します。

モジュールを起動すると、［‘Manual Selection of Patches (パッチの手動選択)’］が有効になり、取得するパッチを選択できるようになります。使用可能なすべてのアップデートパッケージを適用するには、このオプションを無効にします。ただし、接続の帯域幅と送信するデータ量によっては、この設定によりダウンロードに必要な時間が長くなる可能性があります。

［‘Download All Patches Again(すべてのパッチを再度ダウンロード)’］を有効にすると、すべての使用可能なパッチ、インストール可能なパッケージ、および説明がサーバからダウンロードされます。これが有効化されていない場合(デフォルト)、システムにインストールされていないパッチだけを取得します。

さらに、システムを自動的にアップデートすることも可能です。［‘完全自動アップデートの設定’］をクリックして、定期的にアップデートを検出し、適用する自動プロセスを設定します。この処理は完全に自動化されています。システムはスケジュール設定された時間に、アップデートサーバに接続できる必

要があります。

アップデートを実行するには、［'次へ'］をクリックします。手動でアップデートする場合、このクリックによりすべての使用可能なパッチのリストがロードされ、パッケージマネージャが起動します。詳細については、項2.2.1.「ソフトウェアのインストールと削除」を参照してください。パッケージマネージャでは、YOUパッチに対するフィルタが有効化されています。これによりインストールするアップデートの選択が可能です。起動時に、使用可能なセキュリティパッチおよび推奨されるパッチが事前に選択されています。これによりシステムに関連するパッケージがインストールされます。この提案は受け入れる必要があります。

選択をした後に、パッケージマネージャの［'了解'］をクリックします。選択されたアップデートがすべてサーバからダウンロードされ、マシンにインストールされます。接続の速度とハードウェアのパフォーマンスによっては、時間がかかる場合があります。エラーはすべてウィンドウに表示されます。必要に応じて、問題のあるパッケージをスキップします。インストールの前に、詳細について表示するウィンドウが開くパッチもあります。

アップデートをダウンロードおよびインストール中に、ログウィンドウですべてのアクションを追跡できます。すべてのパッチのインストールが成功した後に、［'完了'］をクリックしてYOUを終了します。インストール後にアップデートファイルが必要でない場合、［'アップデート後、ソースパッケージを削除する'］を選択すると、アップデート後にファイルが削除されます。最後に、SuSEconfigが実行され、必要に応じてシステム設定が調整されます。

### 2.2.4 パッチCDによるアップデート

このオプションはFTPサーバからではなく、CDからパッチをインストールします。CDを使用するほうがより速くアップデートできることが利点です。パッチCDを挿入すると、CDに保存されているすべてのパッチがスキャンされ、ダイアログに表示されます。パッチのリストから該当するパッケージを、インストール対象として選択します。モジュールは、パッチCDが存在しない場合にエラーメッセージを表示します。パッチCDを挿入してモジュールを再起動します。

### 2.2.5 システムのアップデート

このモジュールはシステムにインストールされたバージョンのアップデートを可能にします。操作中には、SUSE LINUXベースシステムではなく、アプリケーションソフトウェアだけをアップデートできます。ベースシステムを

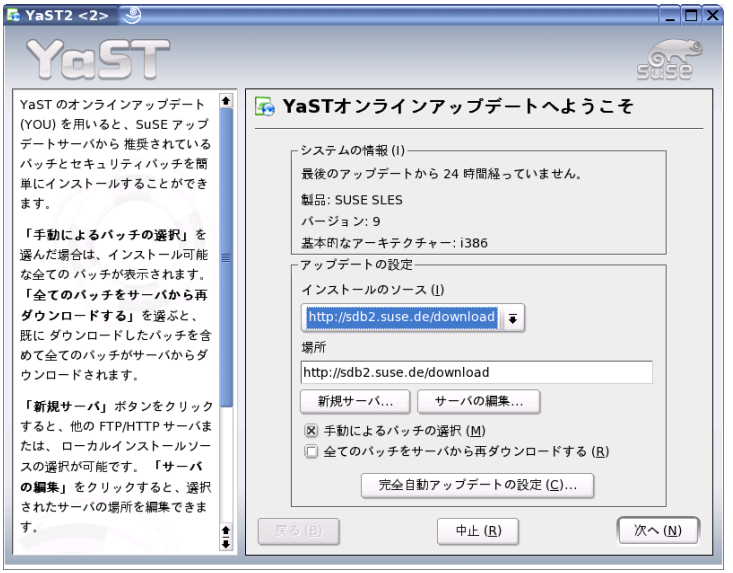


*Figure 2.5: YaSTオンラインアップデート*

アップデートするには、CDなどのインストールメディアからコンピュータをブートします。YaSTのインストールモードを選択する場合は、［'新規にインストールする'］ではなく、［'既存のシステムの更新'］を選択します。

システムをアップデートする処理手順は、新規のインストールと類似しています。最初に、YaSTはシステムを検査し、適切なアップデートの方針を決定し、推奨ダイアログに結果を表示します。詳細を変更するには、マウスを使用して個々の項目をクリックします。［'言語'］および［'キーボード配列'］などの項目のいくつかは、インストールの処理手順について説明するセクションで説明します(項1.3. 「言語の選択」を参照)。このセクションでは、アップデートに関連した設定だけを説明します。

### アップデートに関する選択

いくつかのバージョンのSUSE LINUXがシステムにインストールされている場合、この項目を使用するとリストからアップデートするパーティションを選択できます。

**アップデートオプション**

システムに対するアップデート方法を設定します。2つのオプションが使用可能です。

**Update with Installation of New Software(新しいソフトを含むアップデート)**
システム全体を最新のソフトウェアバージョンにアップデートするには、定義済みの選択グループの1つを選択します。これらの選択グループは、インストール中に用意されるものと同じです。これにより以前に存在しなかったパッケージもインストールされることが確認されます。

**インストール済みパッケージのみアップデート**
このオプションはシステムに既に存在するパッケージだけをアップデートします。新しい機能はインストールされません。

さらに、［'Delete Outdated Packages(廃止されたパッケージの削除)'］を使用して、新しいバージョンが存在しないパッケージを削除します。このオプションは、廃止されたパッケージが不必要にハードディスクの容量を使用しないように、デフォルトで事前に選択されています。

**パッケージ**

［'パッケージ'］をクリックして、パッケージマネージャを起動し、アップデートする個々のパッケージを選択または選択解除します。整合性チェックを実行すると、すべてのパッケージの競合が解決されます。パッケージマネージャの詳細な使用方法については、項2.2.1.「ソフトウェアのインストールと削除」を参照してください。

**バックアップ**

アップデート中に、いくつかのパッケージの設定ファイルは、新しいバージョンの設定ファイルにより置き換えられます。現在のシステムでいくつかのファイルを変更した場合、通常、パッケージマネージャは、置き換えるファイルのバックアップコピーを作成します。このダイアログを使用して、これらのバックアップの範囲を決定します。

**Important**

**バックアップの範囲**

このバックアップにはソフトウェアは含まれません。設定ファイルだけが含まれます。

**Important**

**アップデートに関する重要な情報**

システムのアップデートはとても複雑な処理です。各プログラムパッケージごとに、YaSTは最初にコンピュータにインストールされているバージョンを確認し、旧バージョンを新バージョンを正常に置き換えるのに必要な事柄を判断します。YaSTはまた、インストール済みパッケージ独自の設定をすべて使用するように試みます。いくつかの設定が問題の原因となる可能性があります。旧バージョンの設定では新しいプログラムのバージョンを処理できない場合があり、また、さまざまな設定の間で予期せぬ不整合が発生する可能性があるためです。

既存のバージョンが古いほど、またアップデートするパッケージの設定が標準設定から変更されているほど、アップデートで問題が発生しやすくなります。古い設定が、正常に継承されない場合もあります。この場合、完全に新しい設定を作成する必要があります。アップデートを開始する前に、既存の設定を保存してください。

### 2.2.6 メディアチェック

SUSE LINUXインストールメディアの使用中に問題が発生した場合、このモジュールを使用するCDまたはDVDをチェックします。まれに、特定のメディアを読み込むときに問題が発生するデバイスがあります。これは、「独自に作成した」メディアを使用する場合により発生します。SUSE LINUX CDまたはDVDにエラーがないことをチェックするには、メディアをドライブに挿入してこのモジュールを実行します。［'開始'］をクリックすると、YaSTはメディアのMD5チェックサムをチェックします。これには少し時間がかかります。問題が検出された場合、インストール用にこのメディアを使用しないでください。

## 2.3 ハードウェア

新しいハードウェアは、最初にインストールされているか、ベンダが指定する方法で接続されている必要があります。プリンタやモデムなどの外部デバイスの電源をオンにして、関連するYaSTモジュールを起動します。ほとんどのデバイスは自動的にYaSTにより検出され、技術的なデータが表示されます。自動検出が失敗した場合、YaSTはデバイスのリスト(モデル、ベンダなど)を表示するので、その中から適切なデバイスを選択します。詳細については、ハードウェアに付属しているマニュアルを参照してください。

**Important**

**モデルの指定**

使用中のモデルがデバイスリストに含まれていない場合、類似するモデルを指定します。ただし、モデルは正確に適合しなければならない場合があります。類似するモデルは互換性があるとは限らないためです。

**Important**

### 2.3.1 CD-ROMおよびDVDドライブ

インストール中に、すべての検出されたCD-ROMドライブが、`/etc/fstab`ファイルのエントリを使用して、インストール済みのシステムに統合されます。関連するサブディレクトリが`/media`内に作成されます。YaSTモジュールを使用してシステムに追加のドライブが統合されます。

モジュールが起動すると、検出されたドライブすべてのリストが表示されます。行の最初にあるチェックボックスを使用して新しいドライブにマークを付け、［'完了'］をクリックして統合を完了します。これで、新しいドライブがシステムに統合されます。

### 2.3.2 プリンタ

Linuxでの印刷に関する詳細については、章 12. プリンタの運用を参照してください。一般的な印刷に関する事項が説明されています。YaSTはプリンタを自動的に設定するか、プリンタの手動による設定を支援する設定ダイアログを提供します。次に、コマンド行から印刷するか、アプリケーションが印刷システムを使用するように設定します。YaSTにおけるプリンタの設定の詳細については、項12.5.1. 「ローカルプリンタ」を参照してください。

### 2.3.3 ハードディスクコントローラ

通常、YaSTはインストール中にシステムのハードディスクコントローラを設定します。コントローラを追加すると、このYaSTモジュールを使用してシステムにコントローラを統合します。既存の設定も変更できますが、通常は必要ありません。

検出されたハードディスクコントローラのリストがダイアログに表示され、特定のパラメータを使用して適切なカーネルモジュールを割り当てることができ

ます。［'モジュールのロードをテストする'］を使用して現在の設定が動作することを確認してから、システムに設定を恒久的に保存します。

**Warning**

**ハードディスクコントローラの設定**

これは経験者のための設定ツールです。適切でない設定をするとシステムがブートしなくなります。変更する場合は、テストオプションを使用してください。

**Warning**

### 2.3.4 ハードウェア情報

YaSTは、ハードウェアを検出し、ハードウェアコンポーネントの設定を行います。検出された技術的なデータが個別の画面に表示されます。たとえば、サポートを依頼するときに、ハードウェアに関する情報が必要な場合などに、特に役立ちます。

### 2.3.5 IDE DMAモード

このモジュールを使用して、インストール済みシステムのIDEハードディスク、IDE CDまたはDVDドライブ用に、DMAモードを有効化および無効化します。このモジュールは、SCSIデバイスには影響を与えません。DMAモードは、パフォーマンスとシステム内でのデータ転送スピードを大幅に向上します。

インストール中に、現在のSUSE LINUXカーネルは自動的にハードディスク用のDMAを有効化しますが、CDドライブ用のDMAは有効化しません。すべてのドライブに対してDMAを有効化すると、CDドライブに問題が発生する場合が多いためです。DMAモジュールを使用して、ドライブに対してDMAを有効化します。ドライブが問題なくDMAモードをサポートする場合、ドライブのデータ転送率はDMAを有効化することにより向上します。

**Important**

DMA(ダイレクトメモリアクセス)は、プロセッサの制御を回避して、データがRAMに直接転送されることを意味します。

**Important**

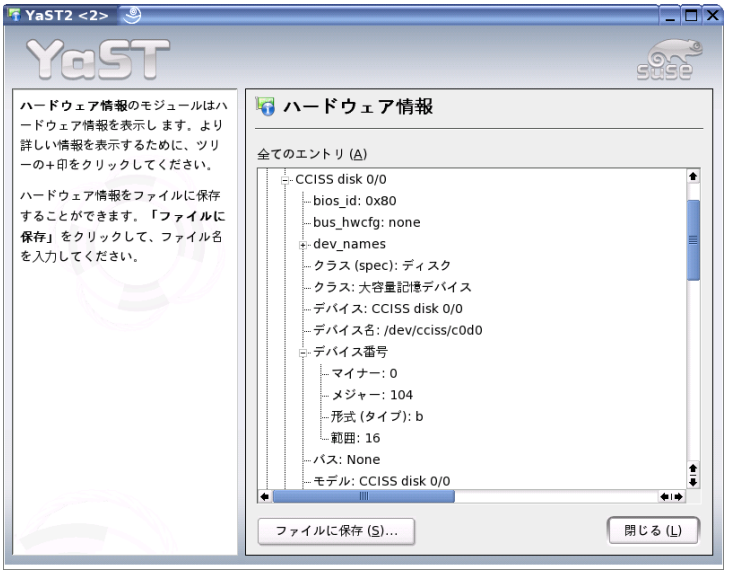


*Figure 2.6: ハードウェア情報の表示*

## 2.3.6 スキャナ

スキャナが接続され、スイッチがオンになっている場合、このYaSTモジュールが起動すると自動的に検出されます。この場合、スキャナのインストール用のダイアログが表示されます。スキャナが検出されない場合、手動設定ダイアログが表示されます。1つまたは複数のスキャナを既にインストールしている場合、変更または削除できる既存のスキャナのリストが表示されます。［'追加'］をクリックして、新しいデバイスを設定します。

次に、デフォルト設定を使用するインストールが実行されます。インストールが成功すると、対応するメッセージが表示されます。ここで、文書をスキャナに挿入して、［'テスト'］をクリックし、スキャナをテストします。

### 検出されないスキャナ

サポートされるスキャナだけが自動検出されます。他のネットワークホストに接続されているスキャナは検出されません。手動で設定する場合、USBスキャ

ナ、SCSIスキャナ、およびネットワークスキャナの3種類のスキャナを区別する必要があります。

**USBスキャナ** ベンダおよびモデルを指定します。YaSTは、次にUSBモジュールのロードを試みます。スキャナが非常に新しい場合、モジュールが自動的にロードされない可能性があります。この場合、USBモジュールを手動でロードするダイアログに自動的に表示されます。詳細については、YaSTヘルプを参照してください。

**SCSIスキャナ** `/dev/sg0`などのデバイスを指定します。SCSIスキャナは、システムの実行中に接続したり、切断しないでください。先にシステムをシャットダウンします。

**ネットワークスキャナ** IPアドレスまたはホスト名を入力します。ネットワークスキャナを設定するには、Support Databaseの記事*Scanning in Linux*を参照してください(`http://portal.suse.com/sdb/en/index.html`、キーワード*scanner*)。

スキャナが検出されない場合、デバイスはサポートされていない可能性があります。ただし、サポートされているスキャナでも検出されない場合があります。その場合は、手動でスキャナを選択して続行します。ベンダおよびモデルのリストに、使用中のスキャナがある場合は、選択します。ない場合は、［'キャンセル'］を選択します。Linuxで動作するスキャナに関する情報については、`http://cdb.suse.de/`および`http://www.mostang.com/sane`を参照してください。

**Warning**

**スキャナを手動で割り当てる**

確実な場合にだけ手動でスキャナを割り当てます。適切でない選択は、ハードウェアが損傷を受ける可能性があります。

**Warning**

### トラブルシューティング

使用中のスキャナが、次に示す理由の1つのために検出されなかった可能性があります。

- 使用中のスキャナはサポートされていません。`http://cdb.suse.de/`で、Linux互換デバイスのリストを確認してください。

- SCSIコントローラが正常にインストールされていません。
- 使用中のSCSIポートには、終端が関係する問題があります。
- SCSIケーブルが長すぎます。
- 使用中のスキャナには、LinuxではサポートされていないSCSI lightコントローラがあります。
- 使用中のスキャナに不具合があります。

**Warning**

SCSIスキャナは、システムの実行中に接続したり、切断しないでください。先にシステムをシャットダウンします。

**Warning**

スキャンの詳細については、SUSE LINUX Professional 9.2 ユーザガイドのkookaについての章を参照してください。

### 2.3.7 サウンド

サウンド設定ツールが起動すると、YaSTはサウンドカードの自動検出を試みます。1つまたは複数のサウンドカードを設定します。複数のサウンドカードを使用するためには、設定する1つのカードの選択から開始します。［'設定'］をクリックして、［'セットアップ'］ダイアログを表示します。［'編集'］をクリックすると以前に設定したサウンドカードを編集するダイアログが開きます。［'完了'］をクリックすると現在の設定が保存され、サウンドカードの設定が完了します。

YaSTが、サウンドカードを自動的に検出できない場合、［'サウンドの設定'］の［'追加'］をクリックしてダイアログを開き、サウンドカードおよびモジュールを選択します。必要な情報については、使用中のサウンドカードのマニュアルを参照してください。対応するサウンドモジュールを使用するALSAによってサポートされるサウンドカードの参照リストについては、/usr/share/doc/packages/alsa/cards.txtおよびhttp://www.alsa-project.org/~goemon/を参照してください。選択した後に、［'次へ'］をクリックして、［'セットアップ'］に戻ります。

### 設定

最初のセットアップ画面で設定レベルを選択します。［'簡易設定'］を使用すると、さらに設定処理を続行する必要はありません。またサウンドテストも実行されません。サウンドカードは自動的に設定されます。［'標準の設定'］を使用すると、出力するボリュームの調節ができます。またテストサウンドを再生できます。［'エキスパート設定'］を使用すると、サウンドカードのオプションを手動でカスタマイズできます。

このダイアログでは、ジョイスティックの設定へのショートカットも用意されています。対応するチェックボックスをクリックします。次に表示されたダイアログボックスでジョイスティックのタイプを選択し、［'次へ'］をクリックします。

### サウンドカードのボリューム

このテスト画面でサウンドの設定をテストします。ボリュームを変更するには、［'+'］と［'-'］を使用します。ボリュームを10%程度にして、スピーカーにダメージを与えたり、耳を損傷することがないようにしてください。テストサウンドは、［'テスト'］をクリックすると聞くことができます。何も聞こえない場合、ボリュームを上げます。［'続行'］をクリックして、サウンド設定を完了します。この時点でボリューム設定が保存されます。

### サウンド設定

［'削除'］を使用して、サウンドカードを削除します。設定されたサウンドカードの既存のエントリが、/etc/modprobe.d/soundファイルで無効化されます。［'オプション'］をクリックしてダイアログを開き、手動でサウンドモジュールのオプションをカスタマイズします。［'追加'］で、追加のサウンドカードを設定します。YaSTが他のサウンドカードを検出した場合は、［'サウンドカードの設定'］が表示されます。YaSTが、サウンドカードを検出しない場合、自動的に［'サウンドカードを手動で選択する'］が表示されます。

Creative Soundblaster LiveまたはAWEサウンドカードを使用する場合、［'Install Sound Fonts (サウンドフォントのインストール)'］を使用して、オリジナルのSoundblasterドライバCD-ROMから、SF2サウンドフォントをハードディスクにコピーします。サウンドフォントは、/usr/share/sfbank/creative/ディレクトリに保存されます。

MIDIファイルを再生する場合は、［'シーケンサーの実行'］を有効化します。この方法で、シーケンサをサポートするモジュールが、サウンドモジュールと共にロードされます。

インストールされたすべてのサウンドカードのボリュームと設定は、［‘完了’］をクリックしたときに保存されます。ミキサー設定は/etc/asound.confファイルに保存され、ALSA設定データ は/etc/modprobe.confファイルの最後に追加されます。

### 2.3.8 テレビとラジオカード

このモジュールを起動および初期化した後に、［‘TV and Radio Cards(テレビカードとラジオカード)’］ダイアログが表示されます。使用中のカードが自動的に検出され、リストの先頭に表示されます。この場合、マウスでその行を強調表示し、［‘設定’］を選択します。使用中のカードが検出されない場合、［‘Other (not recognized) その他(未認識)’］を選択します。［‘設定’］をクリックして、手動設定に進み、ベンダおよびモデルのリストから使用中のカードを選択します。

テレビカードまたはラジオカードを既に設定した場合、既存の設定を［‘変更’］をクリックして変更します。この場合、ダイアログにすべての設定済みカードのリストが表示されます。カードを選択し、［‘編集’］をクリックして手動設定を開始します。

自動ハードウェア検出中に、YaSTは使用中のカードに対して正しいチューナの割り当てを試みます。自信がない場合、［‘Default (recognized)(デフォルト(認識済み))’］の設定を続けて、動作するかどうかを確認します。すべてのチャネルを設定できない場合、チューナタイプの自動検出の失敗が原因である可能性があります。この場合、［‘チューナーの選択’］をクリックして、リストの中から正しいチューナタイプを強調表示します。

技術的な詳細について精通している場合は、エキスパートダイアログを使用して、テレビカードまたはラジオカードに対する設定を指定できます。このダイアログで、カーネルモジュールおよびパラメータを選択します。テレビカードドライバのパラメータもすべてチェックします。これを行うには、対応するパラメータを選択し、パラメータ行に新しい値を入力します。新しい値を［‘適用’］をクリックして確定するか、または［‘リセット’］をクリックしてデフォルトの値に戻します。

［‘TV and Radio Cards, Audio(テレビカードとラジオカード、オーディオ)’］ダイアログでは、インストール済みのサウンドカードを使用する、テレビカードまたはラジオカードに接続できます。サウンドカードの外部オーディオ入力を使用する、テレビカードまたはラジオカードの出力にケーブルを接続する必要があります。これはサウンドカードが既に設定されていて外部入力が有効化されている場合にのみ動作します。サウンドカードを設定していない場合、

項2.3.7. 「サウンド」で説明されるように、［'サウンドカードの設定'］を選択して対応するダイアログを表示します。

テレビカードまたはラジオカードにスピーカのジャックがある場合、サウンドカードを設定しないで直接スピーカを接続することもできます。サウンド機能がないテレビカードもあります。この場合オーディオの設定は必要ありません。たとえばCCDカメラ用のテレビカードなどです。

## 2.4 ネットワークデバイス

システムのネットワークデバイスはすべて、サービスにより使用される前に初期化する必要があります。これらのデバイスの検出および設定は、［'ネットワークデバイス'］モジュールグループで行われます。ネットワーク接続についての背景情報を含む、YaSTでサポートされるネットワークアダプタのタイプの設定に関する詳細については、項22.4. 「ネットワーク統合」を参照してください。無線通信に対するネットワークデバイスの設定については、章 17. 無線通信を参照してください。

## 2.5 ネットワークサービス

このグループには、ネットワークにあるすべての種類のサービスを設定するツールが含まれています。これには名前解決、ユーザー認証、およびファイルサービスが含まれます。

### 2.5.1 メール転送エージェント

このモジュールは、使用中のプロバイダのsendmail、postfix、またはSMTPサーバを使用して電子メールを送信する場合のメール設定を行います。You can fetch mail via the fetchmail program, for which you can also enter the details of the POP3 server or IMAP server of your provider.または、KMailまたはEvolutionなど、任意のメールプログラムを使用して、通常どおり(POP3を使用してメールを受信し、SMTPを使用してメールを送信)POPおよびSMTPアクセスデータを設定します。この場合、このモジュールは必要ありません。

YaSTを使用してメールを設定するには、電子メール設定モジュールの最初のダイアログで、インターネットに接続するための任意のタイプを指定します。次のオプションのうちの1つを選択します。

**'常にネットワークと接続している'** インターネット接続専用回線がある場合に、このオプションを選択します。マシンは永続的にオンラインであり、ダイアルアップ接続は必要ありません。システムが、一元的な電子メールサーバを使用するローカルネットワークの一部であれば、電子メールメッセージに永続的にアクセスするためにこのオプションを選択します。

**'ダイヤルアップ'** この項目は、ネットワーク上ではなく自宅にコンピュータがあり、時々インターネットに接続するユーザが対象です。

**ネットワークと接続していない** インターネットへアクセスする方法がなく、ネットワークにも接続していない場合は、電子メールを送受信できません。

さらに、関連するチェックボックスを有効にすることにより、AMaViSを使用して着信および発信する電子メールに対してウィルススキャンを有効化することができます。メールフィルタリング機能を有効化すると、即座にまた自動的にこのパッケージがインストールされます。次のダイアログでは、発信メールサーバ(通常は使用中のプロバイダのSMTPサーバ)、および着信メールに対するパラメータを指定します。ダイアルアップ接続を使用する場合、さまざまなユーザからのメール受信に対応するために、さまざまなPOPまたはIMAPサーバを指定します。このダイアログを使用して、エイリアス、マスカレードの使用、バーチャルドメインの設定も可能です。［'完了'］をクリックして、メール設定を終了します。

### 2.5.2 他の使用可能なサービス

YaSTでは、他の多くのネットワークモジュールを使用できます。

**DHCPサーバ** YaSTは、少しの処理だけでカスタムDHCPサーバをセットアップできます。章 27. DHCPには、この処理に関する基本的な情報と、YaSTの設定プロセスに関する処理手順の段階的な説明が記載されています。

**DNSサーバ** 大規模なネットワークの場合、名前解決の役割を果すDNSサーバを設定するようにお勧めします。YaSTを使用する設定については、項24.1. 「YaSTによる設定」を参照してください。章 24. ドメインネームシステムには、DNSの背景情報が記述されています。

**DNSとホスト名** このモジュールを使用して、ネットワークデバイス設定中に設定されなかった場合の、ホスト名とDNSを設定します。ホスト名とドメイン名を変更する場合も、このモジュールを使用します。使用中のプロバイダがDSL、モデム、またはISDNアクセスを正常に設定した場合、ネームサーバのリストにはプロバイダのデータから自動的に抽出されたエントリが含まれています。ローカルネットワークに配置されている場合、ホスト名をDHCP経由で入手する場合があり、その場合は名前を変更しません。

**HTTPサーバ** 独自のWebサーバを稼動するには、YaSTを使用してApacheを設定します。詳細については、章 30. Apache Webサーバを参照してください。

**ホスト名** ブート時および小規模なネットワークでは、ホスト名解決はDNSを使用する代わりにこのモジュールでも実行できます。このモジュールのエントリは、`/etc/hosts`ファイルのデータに反映されます。詳細については、項22.5.1. 「/etc/hosts」を参照してください。

**LDAPクライアント** LDAPは、ネットワーク内のユーザ認証のために、NISの代わりに使用します。LDAPに関連する背景情報、およびYaSTを使用するクライアント設定の詳細については、章 29. LDAP—ディレクトリサービスを参照してください。

**NFSクライアントとNFSサーバ** NFSにより、ネットワークのすべてのメンバーがアクセス可能なファイルサーバを稼動できます。ファイルサーバは、特定のアプリケーション、ファイル、および記憶域の容量を、ユーザに対して使用可能にするために使用されます。［'NFSサーバ'］モジュール内で、使用中のホストをNFSサーバとして設定し、ネットワークユーザにより一般的に使用されるエクスポートディレクトリを決定します。適切なアクセス権限を持つすべてのユーザは、それらのディレクトリを、自分のファイルツリーにマウントできます。YaSTモジュールについての説明と、NFSについての背景情報は、章 26. NFS共有ファイルシステムを参照してください。

**NISクライアントとNISサーバ** 複数のシステムを運用している場合、ローカルユーザ管理(`/etc/passwd`と`/etc/shadow`ファイルの使用)は現実的ではなく、管理に手間がかかります。この場合、ユーザデータは一元的なサーバによって管理され、そこからデータをクライアントに配布する必要があります。LDAPおよびSambaと同様に、NISも有効な解決策になります。NISについての詳細な情報、およびYaSTを使用する設定については、章 25. NISの使用を参照してください。

**NTPクライアント** NTP(network time protocol)は、ネットワーク経由でハードウェアクロックを同期するためのプロトコルです。NTPの背景情報、およびYaSTを使用する設定については、章 28. xntpによる時刻の同期を参照してください。

**ネットワークサービス(inetd)** このツールを使用して、SUSE LINUXブート時に開始する、ネットワークサービス(finger、talk、ftpなど)を決定します。これらのサービスは外部ホストを有効にして、コンピュータに接続します。さまざまなパラメータが、すべてのサービスに対して設定できます。デフォルトでは、個々のサービス(inetdまたはxinetd)を管理するマスタサービスは起動しません。

このモジュールが起動すると、inetdまたはxinetdを起動するかどうかを選択します。選択されたデーモンは一般的なサービスを選択して起動します。または、［'追加'］、［'削除'］、［'編集'］を使用して起動するサービスを独自に選択および構成します。

**Warning**

**ネットワークサービス(inetd)の設定**

システムのネットワークサービスの構成と調整は、処理が複雑で、Linuxサービスの概念を包括的に理解している必要があります。

**Warning**

**Proxy** このモジュールでは、システム全体のプロキシ設定を編集できます。プロキシの詳細な情報については、章 33. Squidプロキシサーバを参照してください。

**リモートホストからの管理** リモートホストからVNC接続を介するシステム管理を許可するには、このYaSTモジュールを使用する接続の確立を許可します。詳細については、項3.2.2. 「VNCインストール用のクライアント」を参照してください。

**ルーティング** このツールは、ローカルネットワーク内のゲートウェイを経由してインターネットに接続している場合に必要です。DSLの場合、ゲートウェイのデータはネットワークカードを設定する場合にのみ必要です。しかし、DSLに関するエントリは機能がなにもない単なるダミーです。

**Sambaサーバおよびクライアントの設定**
LinuxとWindowsホストにより構成される異種ネットワークで

は、Sambaが2つの環境間の通信を制御します。Sambaに関して、またクライアントとサーバの設定情報については、章 32. Sambaを参照してください。

## 2.6 セキュリティとユーザ

Linuxの基本的な特徴の1つは、マルチユーザ機能です。つまり、複数のユーザが同じLinuxシステム上で個別に作業することができます。各ユーザは、システムにログインするためのログイン名と個人パスワードにより識別されるユーザアカウントを持ちます。すべてのユーザは独自のホームディレクトリを持ち、そこに個人的なファイルと設定を保存します。

### 2.6.1 ユーザ管理

ユーザの編集を選択した後に、YaSTはシステム内のローカルユーザすべての概要を提供します。大規模なネットワークに属している場合、［'フィルタを設定する'］をクリックしてすべてのシステムユーザ(rootなど)またはNISユーザをリストします。カスタムフィルタ設定を作成することもできます。個々のユーザグループを切り替える代わりに、必要に応じてそれらを組み合わせます。新しいユーザを追加するには、次の画面で必要な情報を入力します。その後、新しいユーザはログイン名とパスワードを使用してホストにログインします。ユーザプロファイルは、［'詳細'］を使用して調整します。ユーザID、ホームディレクトリ、デフォルトのログインシェルを手動で設定します。新しいユーザを特定のグループに割り当てます。パスワードの妥当性を［'パスワードの設定'］で設定します。［'編集'］をクリックして、これらの設定を必要に応じて変更します。ユーザを削除するには、リストからユーザを選択して、［'削除'］をクリックします。

高度なネットワーク管理の場合、［''］を使用して新しいユーザを作成する場合のデフォルト設定を定義します。認証方法(NIS、LDAP、Kerberos、またはSamba)、およびパスワードを暗号化するためのアルゴリズムを選択します。これらの設定は、大規模なネットワークにおいて重要です。

### 2.6.2 グループ管理

YaSTコントロールセンターからグループ管理モジュールを起動するか、ユーザ管理の［'グループ'］をクリックします。どちらのダイアログも、グループの作成、編集、削除という同じ機能を提供します。

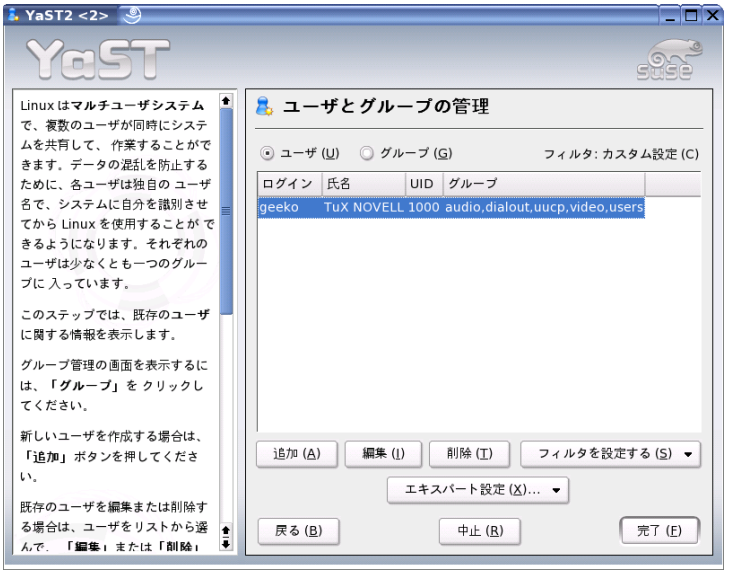


*Figure 2.7: ユ ー ザ 管 理*

YaSTはすべてのグループのリストを提供します。グループを削除するには、削除するグループをリストから選択して(選択した行が紺色で強調表示される)［‘削除’］をクリックします。［‘追加’］と［‘編集’］で、名前、グループID(gid)、およびグループのメンバーを、関連するYaST画面内で入力します。必要に応じて、このグループに変更するためのパスワードを設定します。フィルタ設定は［‘User Administration(ユーザ管理)’］ダイアログでの設定と同じです。

### 2.6.3 セキュリティ設定

［‘Security&Users(セキュリティとユーザ)’］からアクセスできる、［‘ローカルセキュリティ設定’］で、次の4つのオプションのうち1つを選択します。レベル1はスタンドアロンコンピュータに対して設定します(事前設定済み)。レベル2はネットワークを使用するワークステーションに対して設定します(事前設定済み)。レベル3はネットワークを使用するサーバに対して設定します(事前に設定されている)。独自の設定を行うには、［‘カスタム設定’］を使用しま

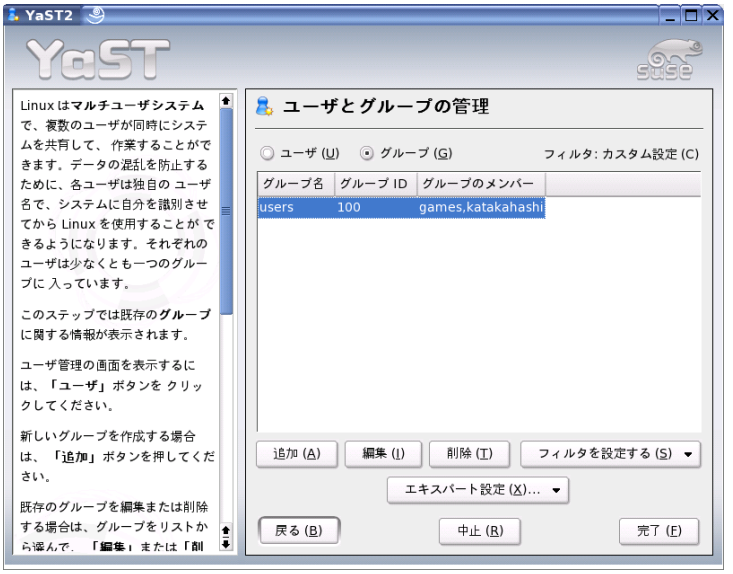


*Figure 2.8: グループ管理*

す。

最初の3つの項目のうちの1つをクリックした場合、事前に設定されたシステムセキュリティオプションのレベルの1つが取り込まれます。これを行うには、［'完了'］をクリックするだけです。［'詳細'］では、変更可能な個々の設定にアクセスします。［'カスタム設定'］を選択した場合、［'次へ'］をクリックして別のダイアログへ進みます。ここでは、デフォルトのインストール値を確認します。

**'パスワードの設定'** 新しいパスワードをシステムがチェックしてから承認する場合は、［'新しいパスワードをチェックする'］と［'パスワードの適性をチェックする'］をオンにします。新しく作成されたユーザに対するパスワードの長さの、最小値および最大値を設定します。パスワードを有効とする期限、またユーザがテキストコンソールにログインしたときに発行する期限切れの警告を、期限切れの何日前に表示するかを定義します。

**'ブート設定'** キーの組み合わせCtrl-Alt-Delを使用して、どのようなアク

ションを実行するかを指定します。通常、この組み合わせが、テキストコンソールで押されると、システムは再起動されます。使用中のマシンまたはサーバが、誰でも触ることができる場所にあり、誰かが承認なしにこのアクションを行う恐れがない限り、この変更を行わないでください。［'中止'］を選択すると、このキーの組み合わせを押すとシステムがシャットダウンします。［'無視する'］を選択すると、このキーの組み合わせは無視されます。

KDEディスプレイマネージャ、KDEのグラフィカルログインからシステムをシャットダウンする権限を与えるには、［'KDMのシャットダウン'］を指定します。［'ルートのみ'］(システム管理者)、［'全てのユーザ'］、［'該当者なし'］、または［'ローカルユーザ'］に権限を与えます。［'該当者なし'］が選択された場合、システムはテキストコンソール経由からのみシャットダウンできます。

**'ログイン設定'** 一般的に、ログイン試行が失敗した後、数秒待ってから、再度ログインが可能になります。これによりパスワードスニファのログインはさらに難しくなります。必要に応じて、［'失敗したログインを記録する'］と［'成功したログインを記録する'］を有効化します。誰かがパスワードを見破ろうとしている可能性がある場合、/var/logにあるシステムログファイルのエントリを確認します。［'リモートグラフィカルログインを許可する'］を使用すると、あるユーザのグラフィカルログイン画面に、ネットワーク経由で別のユーザがアクセスできるようになります。このアクセス手段には潜在的なセキュリティリスクがあるため、デフォルトでは無効になっています。

**'ユーザ設定の追加'** すべてのユーザに数値とアルファベットで構成されたユーザIDが割り当てられます。これらの相関関係は、/etc/passwdファイルを介して確立され、可能な限り一意になります。この画面のデータを使用して、新しいユーザを追加するときに、ユーザIDの数値部分に割り当てる数字の範囲を定義します。ユーザに最適な最小値は500です。自動的に生成される数字は1000から始まります。グループID設定も同じ方法で設定します。

**'その他の設定'** ［'ファイルのアクセス許可の設定'］には、［'簡易'］、［'安全'］、［'被害妄想'］の3つの選択オプションがあります。最初のオプションで、ほとんどのユーザに対しては十分です。YaSTヘルプテキストには、3つのセキュリティレベルについての情報が記載されています。［'被害妄想'］設定は非常に制約が強く、システム管理者の設定に対しても、操作の基本レベルしか設定できません。［'被害妄想'］を選択する場合、一部のプログラムは動作しないか、適切に動作しない可能性があ

ります。これは、ユーザが特定のファイルへのアクセス権を失うためです。

このダイアログでは、updatedbプログラムを起動するユーザも定義します。このプログラムは、毎日またはブート後に自動的に実行され、コンピュータ上の各ファイルの場所が保存されるデータベース(locatedb)を生成します。［‘該当者なし’］を選択する場合、すべてのユーザは、他の(アクセス権のない)ユーザが参照可能なデータベースへのパスのみを参照できます。rootが選択された場合、すべてのローカルファイルにインデックスが付けられます。これは、スーパーユーザであるrootユーザがすべてのディレクトリにアクセスするためです。最後に、［‘カレントディレクトリをrootユーザのパスに追加する’］オプションが無効化(デフォルト)されていることを確認します。

［‘完了’］をクリックして、セキュリティ設定を完了します。

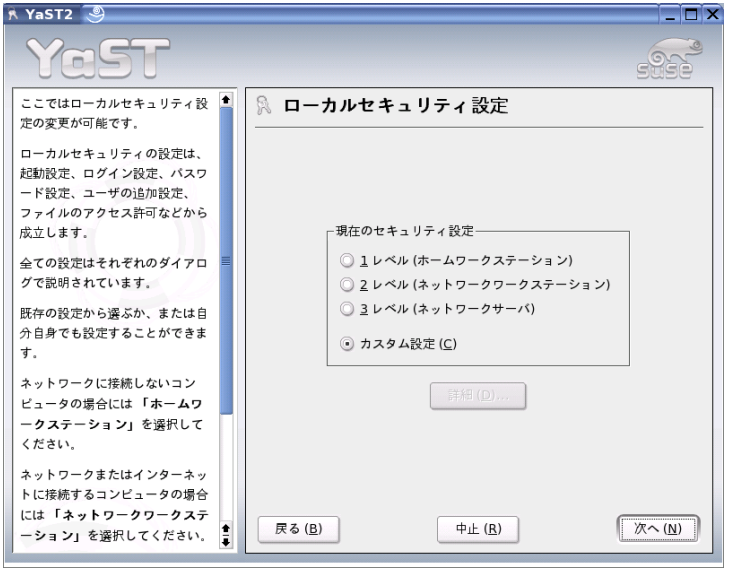


*Figure 2.9:* セキュリティ設定

### 2.6.4 ファイアウォール

このモジュールを使用してSuSEfirewall2を設定し、インターネットからの攻撃に対してマシンを保護します。SuSEfirewall2の詳細については、項34.1. 「マスカレードとファイアウォール」を参照してください。

**Tip**

**ファイアウォールの自動有効化**

YaSTは、すべての設定済みネットワークインターフェース上で、適切な設定を使用してファイアウォールを自動的に起動します。カスタム設定を使用してファイアウォールを再設定するか、無効にする場合にのみ、このモジュールを起動します。

**Tip**

## 2.7 システム

### 2.7.1 システム領域のバックアップコピー

YaSTバックアップモジュールを使用して、システムのバックアップを作成できます。モジュールによって作成されるバックアップには、システム全体は含まれません。変更されたパッケージに関する重要な情報と、重要な記憶領域のコピー、および設定ファイルだけを保存します。

バックアップに保存するデータの種類を定義します。デフォルトでは、最後にインストールした状態から変更されたパッケージに関する情報だけが、バックアップに含まれます。さらに、パッケージ自体には含まれないデータもバックアップに含まれます。たとえば/etcの設定ファイル、または/homeの下位ディレクトリなどです。また、バックアップにはハードディスク上の重要な記憶領域も含まれます。それはパーティションテーブルまたはマスタブートレコード(MBR)など、システムを復元するときに不可欠な記憶領域です。

### 2.7.2 システムの復元

図 2.10. 「復元モジュールのウィンドウの起動」に示す復元モジュールは、バックアップアーカイブからシステムの復元を可能にします。YaSTの説明に従います。［'次へ'］をクリックして、個々のダイアログに進みます。最初に、アーカイブが格納されている場所(リムーバブルメディア、ローカルハー

ドディスク、ネットワークファイルシステム)を指定します。説明と個々のアーカイブの内容が表示されるため、アーカイブからリストアする対象を決定します。

さらに、最後にバックアップしたときから追加されたパッケージのうちアンインストールしたパッケージを示すダイアログと、最後にバックアップしたときから削除されて再インストールしたパッケージを示すダイアログが表示されます。これらの2つの処理により最後にバックアップしたときと完全に同じシステムを復元できます。

**Warning**

**システムの復元**

このモジュールは、通常多くのパッケージとファイルをインストール、置換、アンインストールするため、必ず事前にバックアップ処理を実行してから使用してください。バックアップ処理を実行しなかった場合、データを失う可能性があります。

**Warning**


*Figure 2.10: 復元モジュールのウィンドウの起動*

### 2.7.3 ブートおよびレスキューディスクの作成

このYaSTモジュールを使用して、ブートディスクおよびレスキューディスクを作成します。これらのフロッピーディスクはシステムのブート設定が破損した場合に有用です。レスキューディスクは特に、ルートパーティションのファイルシステムが破損した場合に必要です。

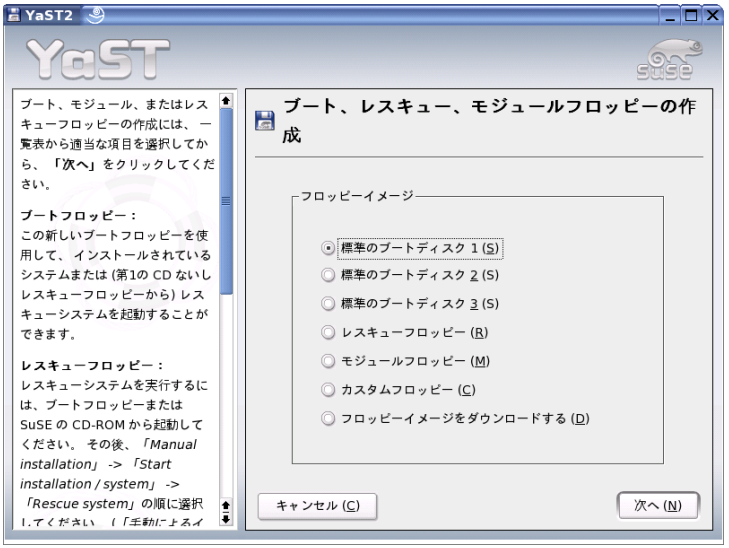


*Figure 2.11: ブートおよびレスキューディスクの作成*

使用可能なオプションは次のとおりです。

**'標準のブートディスク'** このオプションを使用してインストール済みシステムをブートするために使用する標準のブートディスクを作成します。アーキテクチャによっては、ブートディスクの枚数は異なります。すべてのディスクがブート時に必要なため、ダイアログに表示されるすべてのブートディスクを作成する必要があります。それらは、レスキューシステムを起動するためにも必要です。

**'レスキューフロッピー'** このディスクにはインストール済みシステムにおいて、管理タスクを実行するための特別な環境が含まれています。たとえばファイルシステムの確認と修復、およびブートローダの更新などで

す。レスキューシステムを起動するには、標準ブートディスクを使用してブートし、次に、'手動によるインストール' → 'Start Installation or System(インストールを開始 / システム)' → 'レスキューシステム'の順に選択します。次にレスキューディスクの挿入を求めるプロンプトが表示されます。

**'カスタムフロッピー'** これを使用してハードディスクからフロッピーディスクに、既存のフロッピーディスクイメージを書き込みます。

**'フロッピーイメージをダウンロードする'**
これを使用して、URLと認証データを入力し、インターネットからフロッピーディスクイメージをダウンロードします。

これらのフロッピーディスクの1つを作成するには、対応するオプションを選択して、［'次へ'］をクリックします。プロンプトが表示されたらフロッピーディスクを挿入します。［'次へ'］を再度クリックすると、フロッピーディスクが作成されます。

### 2.7.4 LVM

論理ボリュームマネージャ(LVM)は、論理ドライブを使用するハードディスクのカスタムパーティション用ツールです。LVMの詳細については、項3.7.「LVMの設定」を参照してください。

### 2.7.5 パーティション

図 2.12.「YaST Expert Partitioner」に示す［上級者向けのパーティション設定］ダイアログを使って、1つまたは複数のハードディスクのパーティションを手動で設定します。パーティションを追加、削除、および編集することができます。このYaSTモジュールからソフトウェアRAID設定、およびVM設定にもアクセスできます。

**Warning**

インストールされたシステムのパーティションを変更することもできますが、上級者以外は行わないでください。さもないと、間違いを犯した場合に、データ消失の危険性が非常に高くなります。使用中のハードディスクのパーティション設定を変更した場合、その直後にシステムをリブートしてください。稼働中にシステムのパーティションを再設定するより、レスキューシステムを使用したほうが安全です。

**Warning**

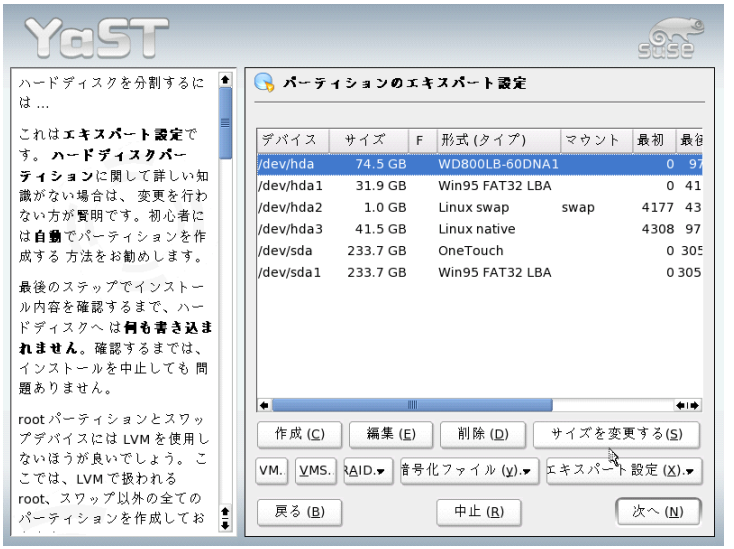


***Figure 2.12:*** *YaST Expert Partitioner*

接続されているすべてのハードディスクの既存パーティションまたは提案パーティションのリストが、［上級者向けのパーティション設定］ダイアログに表示されます。このリストでは、ハードディスク全体は、/dev/hdaや/dev/sdaなど、番号のないデバイスとして表されます。パーティションは、/dev/hda1や/dev/sda1など、それらのデバイスの一部として表されます。ハードディスクのサイズ、形式(タイプ)、ファイルシステム、マウントポイントと、ハードディスクのパーティションも表示されます。マウントポイントには、パーティションのLinuxファイルシステムツリー内のマウント先が指定されています。

インストール中、エキスパートダイアログで作業中の場合は、未使用のハードディスクスペースも表示され、自動的に選択されます。追加のディスクスペースをSUSE LINUX用に用意するには、リストの下から上に、必要なスペースが確保できるまで、領域を解放します(ハードディスクの最後のパーティションから始めて、最初のパーティションの方に向かいます)。たとえば、パーティションが3つある場合、2番目のパーティションをSUSE LINUX専用で使用し、1番目と3番目のパーティションを別のオペレーティングシステム用に保持しておくことはできません。

### パーティションの作成

［‘作成’］を選択します。複数のハードディスクが接続されている場合、新規パーティションの作成先ハードディスクの選択ダイアログが表示されます。次に、パーティションの形式(基本か拡張)を指定します。最大4つの基本パーティションを作成するか、最大3つの基本パーティションと1つの拡張パーティションを作成します。拡張パーティション内に、複数の論理パーティションを作成できます(詳細については、項1.5.4.「パーティションのタイプ」を参照してください)。

使用するファイルシステムとマウントポイント(必要な場合)を選択します。YaSTは、作成されたパーティションごとにマウントポイントを推奨します。パラメータの詳細については、次の項を参照してください。［‘OK’］を選択して、変更内容を適用します。新規パーティションが、パーティションテーブルに表示されます。［‘次へ’］をクリックすると、現在の値が採用され、提案画面に戻ります。インストール中に提案画面に戻ります。

### パーティション設定のパラメータ

新規パーティションを作成する場合、または既存パーティションを変更する場合、多数のパラメータを設定することができます。新規パーティションの場合、適切なパラメータがYaSTによって設定されるので、通常は変更の必要はありません。手動設定を実行する場合、以下の手順に従います。

1. パーティションを選択します。
2. ［‘編集’］を使って、パーティションにパラメータを設定します。

   **ファイルシステムID** この段階でパーティションをフォーマットしたくない場合であっても、パーティションにファイルシステムIDを割り当て、パーティションが正しく登録されるようにします。可能な値は、［‘Linux’］、［‘Linux swap’］、［‘Linux LVM’］、または［‘Linux RAID’］です。LVMとRAIDの詳細については、項3.7.「LVMの設定」および項3.8.「ソフトウェアRAID設定」を参照してください。

   **ファイルシステム** インストールの範囲内でパーティションをすぐにフォーマットするには、パーティション用に次のファイルシステムの1つを選択します。‘Swap’、‘Ext2’、‘Ext3’、‘ReiserFS’、または‘JFS’のいずれかです。各種ファイルシステムの詳細については、章 20. Linuxのファイルシステムを参照してください。

   ［Swap］は特別なフォーマットであり、パーティションを仮想メモリとして使用可能にします。［ReiserFS］ファイルシステム

は、Linuxパーティションのデフォルトファイルシステムです。［ReiserFS］、［JFS］、［Ext3］ファイルシステムは、ジャーナルファイルシステムです。これらのファイルシステムでは、運用中に書き込み処理がログに出力されるので、システムでクラッシュが発生した後、システムを迅速にリストアすることができます。さらに、［ReiserFS］ファイルシステムでは、多数の小容量ファイルが非常に高速に処理されます。［Ext2］はジャーナルファイルシステムではありません。ただし、このファイルシステムは堅牢で、管理に必要なディスクスペースが少ないので、小容量のパーティションに向いています。

**ファイルシステムのオプション** ［ファイルシステムのオプション］画面では、選択したファイルシステムのパラメータを指定します。使用するファイルシステムによって、使用可能なオプションは変わります。

**暗号化ファイルシステム** 暗号化を有効にした場合、すべてのデータは暗号化された状態で、ハードディスクに書き込まれます。これにより、機密データのセキュリティが向上しますが、暗号化に時間がかかるので、システムの処理速度はわずかに低下します。ファイルシステムの暗号化の詳細については、項34.3.「パーティションとファイルの暗号化」を参照してください。

**fstabのオプション** ［fstabのオプション］画面では、ファイルシステムの管理ファイル(`/etc/fstab`)の多数のパラメータを指定します。

**マウントポイント** パーティションのファイルシステムツリー内でのマウント先ディレクトリを指定します。YaSTで表示されるディレクトリから選択するか、または他のディレクトリ名を指定します。

3. ［'次へ'］を選択して、パーティションをアクティブにします。

パーティションを手動で設定する場合は、最低256MBのswapパーティションを作成します。swapパーティションは、その時点で使用されているデータからメモリを解放するために使用されます。これにより、メインメモリを、使用頻度の高い重要なデータ用に使用することができます。

### エキスパート用オプション

［'エキスパート設定'］は、次のコマンドを含むメニューを開きます。

**パーティションテーブルの再読み込み**
ディスクからパーティション設定を再読み込みします。たとえば、テキ

ストコンソールで手動パーティション設定を行った後で、これが必要になります。

**パーティションテーブルとディスクラベルの削除**
この処理では、古いパーティションテーブルが完全に上書きされます。たとえば、独自のディスクラベルに問題がある場合に役立ちます。この方法を用いると、ハードディスク上のすべてのデータが失われます。

### パーティション設定に関するヒント

YaSTによってパーティション設定が実行され、システム内に他のパーティションが検出された場合、検出されたパーティションも/etc/fstabファイルに入れられ、この設定データへのアクセスが簡単になります。このファイルには、システム内のすべてのパーティションとそのプロパティ（ファイルシステム、マウントポイント、ユーザのパーミッションなど)が記載されています。

***Example 2.1:*** */etc/fstab:*パーティションデータ

```
/dev/sda1    /data1    auto      noauto,user 0 0
/dev/sda5    /data2    auto      noauto,user 0 0
/dev/sda6    /data3    auto      noauto,user 0 0
```

LinuxパーティションかFATパーティションかに関係なく、パーティションは、noautoオプションとuserオプションを使って指定されます。これにより、すべてのユーザがこれらのパーティションを、必要に応じてマウントまたはアンマウントすることができます。セキュリティ上の理由で、YaSTでは、プログラムを関連位置で実行するのに必要なexecオプションは、ここに自動的に入力されません。ただし、そこからプログラムを実行するために、このオプションを手動で入力できます。 不正インタプリタやパーミッションの拒否などのシステムメッセージが出されたら、この方法が必要になります。

### パーティション設定とLVM

Expert Partitionerから［'LVM'］を選択してLVM設定にアクセスします( 項3.7.「LVMの設定」を参照)。ただし、作業するLVM設定がシステムにすでに存在している場合は、セッションで初めてLVM設定を入力した時点でただちに、自動的にその設定がアクティブになちます。この場合、アクティブになったボリュームグループに属するパーティションを含むすべてのディスクは、パーティションを再設定できません。Linuxカーネルは、ハードディスクの変更さ

れたパーティションテーブルを、このディスク上のいずれかのパーティションが使用中になった時点では、再読みすることができないからです。ただし、機能しているLVM設定がシステム上にがすでにある場合は、物理的なパーティション再設定は必要になりません。代わりに、論理ボリュームの設定を変更します。

物理ボリューム(PV)の先頭では、そのボリュームに関する情報がパーティションに書き込まれます。このようにすれば、PVは、その所属するボリュームグループを"認識"します。こうしたパーティションをLVM以外の目的で再使用するには、このボリュームの先頭を削除しておくようにお勧めします。たとえば、VG `system`およびPV `/dev/sda2`では、これは、コマンド`dd if=/dev/zero of=/dev/sda2 bs=512 count=1`で行うことができます。

**Warning**

**ブート用ファイルシステム**

ブートに使用するファイルシステム(rootファイルシステムまたは`/boot`)をLVM論理ボリュームに格納しないでください。通常の物理パーティションに格納してください。

**Warning**

## 2.7.6 プロファイルマネージャ(SCPM)

SCPM (system configuration profile management)モジュールには、システム設定の作成、管理、切り替えなどの機能が用意されています。これは、さまざまな場所(さまざまなネットワーク)で、さまざまなユーザにより使用される、モバイルコンピュータにとって特に有用です。それでも、この機能はデスクトップマシンにとっても有用です。これによりさまざまなハードウェアコンポーネントの使用や、テスト設定の使用が可能になるためです。SCPMの基礎と処理内容の詳細については、章 15. システム設定プロファイル管理を参照してください。

## 2.7.7

SUSE LINUXは、複数のランレベルで実行できます。デフォルトでは、システムはランレベル5でブートします。それにより、マルチユーザモード、ネットワークアクセス、およびグラフィカルユーザインターフェース(X Window System)が提供されます。他のランレベルで提供される機能は次のとおりで

す。ランレベル3では、マルチユーザモードとネットワークアクセスは提供されますが、X Window Systemは提供されません。ランレベル2では、マルチユーザモードは提供されますが、ネットワークアクセスは提供されません。ランレベル1とSではシングルユーザモードが提供されます。ランレベル0はシステム停止、ランレベル6はシステムのリブートになります。

さまざまなランレベルは、より高いランレベルの特定のサービス(Xまたはネットワーク)に関連する問題が発生した場合に有用です。この場合、サービスを修復するために、システムをより低いランレベルでブートすることができます。多くのサーバはグラフィカルユーザインターフェースなしで動作するため、Xなしのランレベル、たとえばランレベル3でブートします。

通常、標準のランレベル(5)で問題ありません。ただし、グラフィカルユーザインターフェースがフリーズしたときはいつでも、Ctrl-Alt-F1を押してテキストコンソールに切り替えてX Window systemを再起動し、ルートとしてログインして、`init 3`コマンドを使用してランレベル3に切り替えます。これにより、X Window Systemはシャットダウンし、テキストコンソールに切り替わります。グラフィカルシステムを再起動するには、「`init 5`」と入力します。

SUSE LINUXでのランレベルの詳細について、およびYaSTランレベルエディタについての説明は、章 7. Linuxシステムのブートと設定を参照してください。

### 2.7.8 Sysconfigエディタ

`/etc/sysconfig`ディレクトリには、SUSE LINUXにとって最も重要な設定ファイルが含まれています。sysconfigエディタはすべての設定をよく整えられた形式で表示します。値を変更して、個々の設定ファイルに保存できます。一般的に、ファイルを手動で編集する必要はありません。パッケージがインストールされたとき、またはサービスが設定されたときにファイルは自動的に変更されるためです。`/etc/sysconfig`とYaST sysconfigエディタの詳細については、章 7. Linuxシステムのブートと設定を参照してください。

### 2.7.9 タイムゾーンの選択

タイムゾーンはインストール中に設定されていますが、ここで変更できます。リストから国または地域をクリックして、［'ローカルタイム'］または［'UTC'］(世界協定時刻、以前のグリニッジ標準時)を選択します。［'UTC'］は、Linuxシステムで多くの場合使用されます。Microsoft Windowsなど、追加のオペレーティングシステムを使用するマシンでは、多くの場合ローカルタイムを使用します。

### 2.7.10 言語選択

ここでは、Linuxシステムで使用する言語を選択します。YaSTで選択された言語は、YaSTおよびデスクトップ環境を含む、システム全体に適応されます。

## 2.8 その他

### 2.8.1 サポートリクエストの送信

SUSE LINUXを購入すると、無償インストールサポートを受ける権利が与えられます。サポートする範囲、住所、電話番号の詳細については、Webサイトhttp://www.novell.com/linux/suse/にアクセスしてください。

YaSTには、電子メールにより、サポートリクエストを直接SUSEチームに送信する機能が備えられています。最初に登録が必要です。必要なデータの入力から開始してください。登録コードはCDカバーの背面に記載されています。質問内容に応じて、次のウィンドウで問題のカテゴリを選択し、問題の詳細を入力してください。図 2.13. 「サポートリクエストの送信」を参照してください。YaSTヘルプテキストも参照してください。サポートチームが援助するために必要な、問題を記述する最善の方法について説明されています。

**Tip**

特別な問題へのサポートなど、高度なサポートが必要な場合は、http://support.novell.com/linux/を参照してください。

**Tip**

### 2.8.2 ブートログ

ブートログ/var/log/boot.msgには、コンピュータが起動したときに表示される画面のメッセージが含まれています。このYaSTモジュールを使用してログを表示します。たとえば、すべてのサービスと機能が意図したとおりに起動したかどうかをログにより確認します。

### 2.8.3 システムログ

システムログは、コンピュータの操作を/var/log/messagesログに出力します。カーネルメッセージはここに記録され、日時に基づいてソートされます。

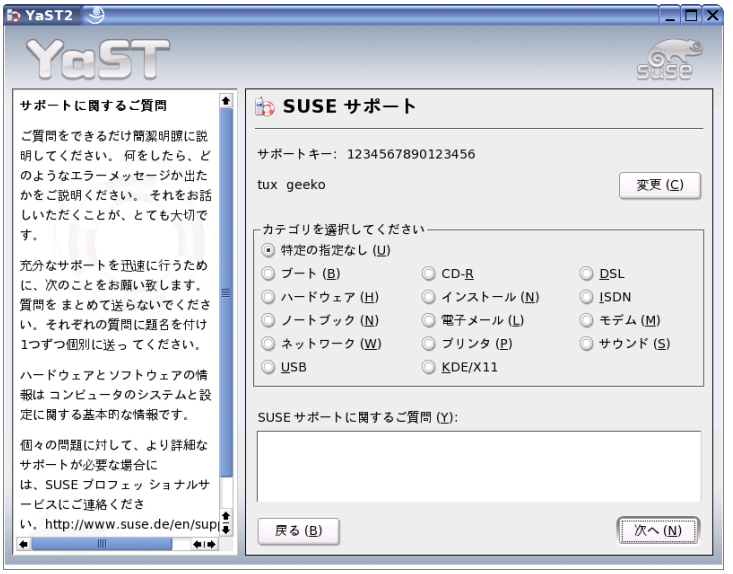


*Figure 2.13: サポートリクエストの送信*

### 2.8.4 ベンダのドライバCDのロード

このモジュールでは、SUSE LINUX用のドライバが含まれているLinuxドライバCDから、デバイスドライバを自動的にインストールします。SUSE LINUXを最初からインストールした場合、このYaSTモジュールを使用して、インストール後にベンダが提供するCDから必要なドライバをロードします。

## 2.9 テキストモードのYaST (ncurses)

ここでは、自身のシステムでXサーバを稼動しないシステム管理者および上級者を主に対象としており、テキストベースのインストールツールに基づいています。テキストモードのYaST (ncurses)の起動と運用に関する基本的な情報を示しています。

YaSTをテキストモードで起動すると、YaSTのコントロールセンタが最初に表示されます。図 2.14. 「テキストモードのYaSTのメインウィンドウ」を参照し

てください。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’］用のボタンがあります。

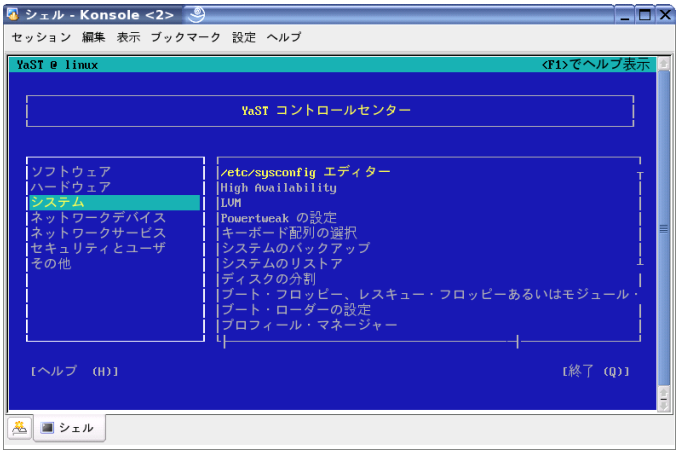


*Figure 2.14:* テキストモードのYaSTのメインウィンドウ

YaSTのコントロールセンタが起動されると、カテゴリ［‘Software’］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、モジュールのタイトルが背景色付きで表示され、簡単な説明が下方のフレームが表示されます。

Enter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でナビゲートする代わりとなるボタンを選択するには、Alt-yellow_letterを使用します。YaSTのコントロールセンタを終了するには、［‘終了’］ボタンを押すか、カテゴリ概要で［‘終了’］を選択してEnterを押します。

### 2.9.1 モジュールでのナビゲーション

以降では、YaSTモジュール内のコントロール要素について、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説明します。可能性のある例外事項については、項2.9.2. 「キーの組み合わせの制約」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**
ボタン間および選択リストを含むフレーム間でナビゲートするには、TabとAlt–TabまたはShift–Tab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**
［］が付いていつボタン(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またはEnter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による確認は不要です。Tabでアイテムにナビゲートする場合は、Enter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(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。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、［'OK'］、［'次へ'］、および［'完了'］の代用として使用します。F1を押すと、YaSTのヘルプが表示され、個々のFキーにマップされた機能がそのヘルプに表示されます。

### 2.9.2 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の組み合わせを使用している

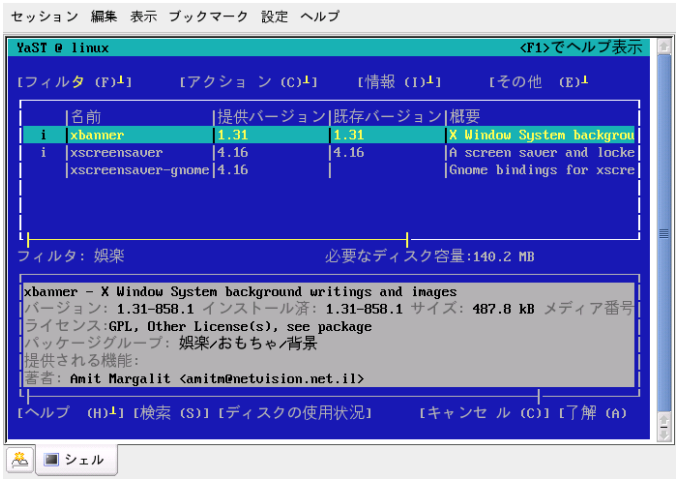


*Figure 2.15: ソフトウェアインストールモジュール*

と、YaSTでのAltの組み合わせが機能しない場合があります。AltやShift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。

**Altを Escの代用とする** Altショートカットは、Altの代わりにEsc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。

**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。

## 2.9.3 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、`yast <module_name>`と入力します。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。`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。

### 2.9.4 YOUモジュール

YaSTオンラインアップデート(YOU)モジュールは、他のYaSTモジュールのように、rootでコ次のように入力すると、コマンドラインから起動できます。

```
yast online_update .url <url>
```

yast online_updateは、対応するモジュールを起動します。オプションurlは、サーバ(ローカルまたはインターネット上）を指定するために使用できます。YOUによってそこからすべての情報およびパッチが取り出されます。モジュールの起動時にサーバを指定しない場合は、YaSTダイアログでサーバまたはディレクトリを選択します。［'完全自動アップデートの設定'］を選択して、更新を自動化するための、クローンジョブを設定します。

## 2.10 コマンドラインからのオンラインアップデート

コマンドラインツールのonline_updateを使用することにより、スクリプトなどを使用して、システムを自動的にアップデートすることができます。たとえば、定期的な指定した時間に、システムが特定のサーバでアップデートを検索し、パッチおよびパッチ情報をダウンロードするという設定にするとします。ただし、パッチは自動的にインストールしない設定にします。代わりに、パッチを参照し、インストールするパッチを後で選択するとします。

ツールを使用するには、次のコマンドを実行するcronジョブを最初に設定します。

```
online_update -u <URL> -g <type_specification>
```

-uは、パッチがダウンロードされるディレクトリツリーの基になるURLを示します。サポートされるプロトコルには、http、ftp、smb、nfs、cd、dvd、およびdirが含まれます。-gは、パッチをローカルディレクトリにダウンロードしますが、インストールはしないことを示します。必要に応じて、［セキュリティ］、［推奨］、［オプショ

ン］のタイプを指定することによりパッチをフィルタします。フィルタを使用しない場合、online_updateはすべての新しい、［セキュリティ］と［推奨］のパッチをダウンロードします。

ダウンロードされたパッケージは個々のパッケージを確認することなく即座にインストールできます。online_updateは、パッチを/var/lib/YaST2/you/mntディレクトリに保存します。パッチをインストールするには、次のコマンドを実行します。

```
online_update -u /var/lib/YaST2/you/mnt/ -i
```

-uパラメータは、インストールするパッチのローカルなURLを指定します。-iは、インストール処理を開始します。

ダウンロードされたパッチを確認してからインストールを開始するには、次のようにしてYOUダイアログを起動します。

```
yast online_update .url /var/lib/YaST2/you/mnt/
```

インターネット上のリモートディレクトリの代わりに、ダウンロード済みのパッチを含むローカルディレクトリを使用して、YOUはパッチのインストールを開始します。パッケージマネージャでのパッケージのインストールでも、同じ方法でインストールするパッチを選択します。

YaSTオンラインアップデートの動作は、コマンドラインパラメータを使用して制御できます。構文は、online_update [command-line parameter]です。使用可能なパラメータとその機能を次にリストします。

**-u URL** ディレクトリツリーのURLに基づいて、パッチがダウンロードされます。

**-g** パッチだけをダウンロードします。インストールしません。

**-i** ダウンロード済みパッチをインストールします。ダウンロードしません。

**-k** 新しいパッチが使用可能かどうか確認します。

**-c** 現在の設定を表示します。アクションは起こしません。

**-p product** パッチをダウンロードする対象の製品を指定します。

**-v version** パッチをダウンロードする対象の製品バージョンを指定します。

**-a architecture** パッチをダウンロードする対象の製品の基本アーキテクチャを指定します。

**-d** ドライラン。パッチをダウンロードしインストールをシミュレートします(システムは変更されずテストだけが行われる)。

**-n** ダウンロード済みファイルの署名確認をしません。

**-s** 使用可能なパッチのリストを表示します。

**-V** 詳細表示モード。

**-D** 上級者用、およびトラブルシューティング用のデバッグモード。

online_updateの詳細については、「`online_update -h`」と入力してください。

# 特殊なインストール手順

SUSE LINUXは、さまざまな方法でインストールできます。たとえば、グラフィカルモードによるクイックインストールや、さまざまな手動調整が可能なテキストベースのインストールなどの方法があります。ここでは、さまざまなインストール手順と、CD-ROMやNFSなど多様なインストールソースの使用方法について説明します。この章では、インストール中に発生する問題の解決方法およびパーティション設定の詳細についても説明します。

# 3.1 linuxrc

すべてのマシンには、ハードウェアを初期化するために起動時に実行される特殊なBIOSルーチンがあります。実際のブートプロセスでは、これらのルーチンがコンピュータによって実行されるイメージをロードし、後続のブートプロセスを制御する必要があります。このイメージはブートマネージャでもかまいませんが、カーネルを直接ロードすることもできます。SUSE LINUXのインストール中には、カーネルとlinuxrcというプログラムを含むブートイメージがロードされます。

linuxrcは、実際のブートプロセスの前に、カーネルの起動段階で動作するプログラムです。デフォルトでは、このプログラムは、ユーザの指示なしに実行し、終了後YaSTを起動します。特殊なパラメーターをモジュールに渡す必要がある場合、あるいはハードウェア検出が失敗した場合は、手動インストールで開始して、linuxrcを対話方式で実行する必要があります。

linuxrcの使用は、インストールだけに制限されているわけではありません。インストール済みのシステム、また独立したRAMディスクベースのレスキューシステムのブートツールとしても使用できます。詳細については、項5.4.「SUSEレスキューシステム」を参照してください。

システムが初期RAMディスク(initrd)を使用する場合、linuxrcというシェルスクリプトもブート時のモジュールのロードを処理します。このスクリプトは、スクリプト`/sbin/mkinitrd`によって動的に生成されます。このスクリプトは、インストールに使用されるプログラムlinuxrcとはまったく異なるものであり、それと混同しないようにしてください。

## 3.1.1 linuxrcへのパラメータの転送

起動動作を変更するパラメーターをlinuxrcに渡すことができます。linuxrcは、infoファイルをフロッピーディスクで検索するか、`/info`内の`initrd`で検索します。次にlinuxrcは、カーネルプロンプトでパラメータをロードします。ファイル`/linuxrc.config`内でデフォルト値を編集できます。しかし、変更する場合は、infoファイルで行うことをお勧めします。

**Tip**

linuxrcを手動モードで実行できます。このためには、インストール・プロンプトでパラメーター「manual=1」を使用します。

**Tip**

infoファイルには、キーワードと値がkey: valueの形式で示されています。このようなキーと値のペアは、インストールメディアで表示されるブートプロンプトから、このkey=valueの形式で入力することもできます。利用できるキーのリストについては、ファイル/usr/share/doc/packages/linuxrc/linuxrc.htmlを参照してください。次のリストでは、いくつかの重要なキーと値の例を示します。

**Install:URL (nfs、ftp、hd、など)** インストールソースをURLの形で指定します。使用可能なプロトコルには、cd、hd、nfs、smb、ftp、http、およびtftpが含まれています。このURLの構文は、Webブラウザで使用される一般的な形式に対応しています。 例を示します。

- nfs://<server>/<directory>
- ftp://[user[:password]@]<server>/<directory>

**Netdevice:<eth0>** Netdevice:キーワードは、インストール先ホストで複数のイーサネットインタフェースが使用可能な場合に、linuxrcが使用する必要のあるインタフェースを指定します。

**HostIP: <10.10.0.2>** ホストのIPアドレスを指定します。

**Gateway: <10.10.0.128>** これは、インストールサーバがホストと同じサブネットワーク内に存在していない場合に、そのサーバに到達するためのゲートウェイを指定します。

**Proxy: <10.10.0.1>** Proxy:キーワードは、FTPまたはHTTPプロトコルに対するプロキシを定義します。

**ProxyPort: <3128>** これは、プロキシがデフォルトのポートを使用しない場合に、プロキシが使用するポートを指定します。

**Textmode: <0|1>** このキーワードは、YaSTをテキストモードで起動できるようにします。

**VNC: <0|1>** VNCパラメータは、VNC経由のインストールプロセスを制御します。 VNCは、グラフィカルコンソールを備えていないホストへのインストールをより便利にします。これを有効にした場合、対応するサービスが、インストール先ホストで有効になります。VNCPasswordキーワードも参照してください。

**VNCPassword:<password>** これは、VNCインストールがセッションへのアクセスを制御するためのパスワードを設定します。

**UseSSH: <0|1>** このキーワードは、テキストモードのYaSTを使用してインストールを実行する際に、SSHを使用してlinuxrcにアクセスできるようにします。

**SSHPassword:<password>** これは、`root`ユーザがlinuxrcにアクセスするためのパスワードを設定します。

**Insmod:<module parameters>** これは、カーネルがロードする必要のあるモジュール、およびそのモジュールが必要とするすべてのパラメータを指定します。複数のモジュールパラメータは、半角スペースで区切る必要があります。

**AddSwap:<0|3|/dev/hda5>** 0に設定した場合、システムはスワップパーティションを有効にしようとしません。正の数に設定した場合、この番号に対応するパーティションがスワップパーティションとして有効になります。代わりに、パーティションの完全なデバイス名を指定することもできます。

## 3.2 VNCによるインストール

VNC (*Virtual Network Computing*、仮想ネットワークコンピューティング)は、スリムで使いやすいクライアントを通して、リモートXサーバにアクセスできるようにするクライアントサーバソリューションです。このクライアントは、Microsoft Windows、Apple社のMacOS、およびLinuxを含むさまざまなオペレーティングシステムで使用できます。

VNCクライアントであるvncviewerは、インストールプロセスの実行中に、グラフィカル表示とYaSTの処理を保証する目的で使用されます。インストール対象のシステムをブートする前に、リモートコンピュータの準備をし、インストール対象のシステムにネットワーク経由でアクセスできるようにしてください。

### 3.2.1 VNCインストール用の準備

VNCインストールを実行するには、カーネルに特定のパラメータを渡します。カーネルを起動する前に、この作業を行う必要があります。そうするには、ブートプロンプトに次のコマンドを入力します。

```
vnc=1 vncpassword=<xyz> install=<source>
```

vnc=1は、インストール先システムに対して、VNCサーバを起動するようシグナルを送信します。vncpasswordは、後で使用するパスワードです。インストールソース(install)は、手動で指定する(プロトコルと、該当のディレクトリに対応するURLを入力)か、その中にslp:/命令を含めることができます。slp:/命令を含める場合、SLP照会により、インストールソースは自動的に決定されます。SLPに関する詳細は、章 23. ネットワーク上のSLPサービスを参照してください。

### 3.2.2 VNCインストール用のクライアント

インストール先コンピュータと、その上で動作するVNCサーバへの接続を確立するには、VNCクライアントを使用します。SUSE LINUX環境では、vncviewerを使用します。これは、xorg-x11-Xvncパッケージの一部です。Windowsクライアントからインストール先システムへの接続を確立するには、Windowsシステムにtightvncプログラムをインストールします。このプログラムは、最初のSUSE LINUXCDの、/dosutils/tightvncディレクトリにあります。

好みのVNCクライアントを起動します。次に、プロンプトが表示されたら、インストール先システムのIPアドレスと、VNCパスワードを入力します。

代わりに、Java対応ブラウザを使用してVNC接続を確立することもできます。この作業を行うには、ブラウザのアドレスフィールドに、次の内容を入力します。

```
http://<IP address of the installation system>:5801/
```

接続が確立された後で、YaSTが起動され、インストールを開始できます。

## 3.3 YaSTを使用するテキストベースのインストール

グラフィックインターフェースを用いたインストールに加えて、SUSE LINUXはYaSTのテキストバージョン(コンソールモード)を用いてインストールすることもできます。YaSTモジュールはすべてこのテキストモードで利用できます。テキストモードは特にグラフィカルインターフェースを必要としない場合、たとえばサーバシステムにインストールする場合、またはX Window Systemでグラフィックカードがサポートされていない場合に役立ちます。こ

のインストールモードを使用すると、視覚障害のあるユーザが、適切な出力デバイスの助けを得てSUSE LINUXをインストールできます。

最初に、BIOS内のブートシーケンスを設定し、CD-ROMドライブからの起動を可能にします。DVDまたはCD 1をドライブに挿入し、マシンを再起動します。数秒後にインストール開始画面が表示されます。

↑と↓を使用して［'手動によるインストール'］を選択します。これを10秒以内に行わないとシステムが自動的にインストールシステムを起動します。一般的ではありませんが、ハードウェアが特別なパラメータを必要とする場合、それらをブートオプションに入力します。使用されているキーボードの言語をインストール言語として選択した場合、キーボードレイアウトは正しくなります。これによりパラメータの入力ができます。

F2 (［'Video Mode(ビデオモード)'］)を使用してインストール用に画面解像度を設定します。インストール中にグラフィックカードが問題を起こすと想定される場合は、［'テキストモード'］を選択します。次に、Enterを押します。`Loading Linux kernel`(Linuxカーネルをロードしています)という進捗バーがついたボックスが表示されます。カーネルがブートし、linuxrcが起動します。linuxrcのメニューを使用してインストールを続行します。

他のブートの問題は、通常カーネルパラメータを使用することで回避できます。DMAが問題を引き起こす場合は、起動オプション［'Installation—Safe Settings(インストール-安全な設定)'］を使用します。ACPI (advanced configuration and power interface)の使用により問題が発生する場合は、次のカーネルパラメータを使用できます。

**`acpi=off`** このパラメータは、コンピュータ上の完全ACPIサブシステムを無効にします。これはコンピュータがACPIをまったく処理できない場合、またはコンピュータのACPIが問題を引き起こしていると考えられる場合に役に立ちます。

**`acpi=oldboot`** ブートに必要な部分以外のACPIをオフにします。

**`acpi=force`** 2000年より前の日付が付けられた古いBIOSを持つコンピュータとしても、常にACPIを有効にします。このパラメータは、`acpi=off`に加えて設定されるとしても、ACPIを有効にします。

**`pci=noacpi`** 新しいACPIシステムのPCI IRQルーティングを無効にします。

この接続では、`https://portal.suse.com`で、「acpi」キーワードを使用してSupport Database (サポートデータベース)の記事を検索します。

カーネルのロード中、またはインストール中に説明できないエラーが発生した場合は、ブートメニューから［'メモリテスト'］を選択し、メモリを確認しま

す。Linuxはハードウェアに高いスペックを必要とするため、メモリおよびそのタイミングも正確に設定されている必要があります。詳しい情報は、キーワード「memtest86」を使用してSupport Database (サポートデータベース)を検索すると入手できます。可能な場合、夜間にメモリテストを実行します。

## 3.4 SUSE LINUXの起動

インストールが終了したら、日々の操作でのLinixのブート方法を決定します。以下に、Linuxをブートするさまざまな方法について簡単に説明します。ブート方法は、使用目的によって異なります。

**Linuxブートローダ** システムをブートするに当たって、最も用途が広く技術的にエレガントなソリューションは、GRUB (Grand Unified Bootloader)やLILO (Linuxローダ)のようなLinuxブートマネージャを使用することです。 これらはブートする前にオペレーティングシステムを選択できます。ブートローダは、インストール時に設定することも、YaSTを使用して後で設定することもできます。

**ブートディスク** Linuxは、ブートディスクからブートできます。この方法を使用できるのは、システムにフロッピードライブがある場合だけです。ブートディスクは、YaSTで作成できます。項2.7.3. 「ブートおよびレスキューディスクの作成」を参照してください。

ブートディスクは、他の方法でブートするのが困難な場合や、最終的なブートメカニズムの決定を延期したい場合に、暫定的なソリューションとなります。またブートディスクは、OS/2やWindowsNTでの接続に対する適切なソリューションにもなります。

**Warning**

ブートセクタ(MBR)の構造をチェックし、GRUBまたはLILOのインストール後にウイルスに関する間違った警告を表示する、改良版のBIOSがあります。この問題を解決するには、BIOSに入り、該当する設定を見つけ出します。たとえば、［'virus protection'］をオフにします。このオプションは、後でオンに戻すことができます。ただし、使用している唯一のオペレーティングシステムがLinuxである場合は、上記の作業は不要です。

**Warning**

各種ブート方法の詳細については、章 8. ブートローダを参照してください。

### 3.4.1 グラフィカルSUSE画面

SUSE LINUX 7.2以降は、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。

### 3.4.2 SUSE画面の無効化

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、章 8. ブートローダ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画面を完全に無効化** 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。

**Tip**

カーネル内のframebufferサポートを自動的に無効にすると、スプラッシュスクリーンも無効になります。カスタムカーネルで実行する場合は、SUSEはシステムにサポートを提供しません。

**Tip**

## 3.5 ヒントとコツ

コンピュータの中には、CD-ROMドライブがなく、ブート可能なフロッピーディスクドライブが利用可能なものがあります。そのようなコンピュータにイ

ンストールするには、ブートディスクを作成し、ブートディスクを使用してシステムをブートする必要があります。

3.5インチHDフロッピーディスクをフォーマットし、ブート可能な3.5インチフロッピーディスクを作成する必要があります。CD 1上のbootディレクトリには、多数のディスクイメージが含まれています。適切なユーティリティを使用すると、それらのイメージをフロッピーディスクにコピーできます。そのようにして作成されたフロッピーディスクをブートディスクと言います。

ディスクイメージには、ローダSYSLINUXとプログラムlinuxrcも含まれています。SYSLINUXを使用すると、ブート時にカーネルを選択し、使用するハードウェアに必要なパラメータを指定できます。プログラムlinuxrcは、使用するハードウェア用のカーネルモジュールのローディングをサポートし、その後インストールを開始します。

### 3.5.1 rawwritewinによるブートディスクの作成

Windowsでは、グラフィカルユーティリティrawwritewinを使用してブートディスクを作成できます。このユーティリティはCD1上のディレクトリdosutils/rawwritewinにあります。

スタートアップ時に、イメージファイルを指定します。イメージファイルは、CD 1上のディレクトリbootにあります。少なくとも、イメージbootdiskとmodules1が必要ですファイルブラウザでこれらのイメージを一覧表示するには、ファイルタイプを［すべてのファイル］に設定します。次に、フロッピーディスクドライブにフロッピーディスクを挿入し、［'書き込み'］をクリックします。

他のディスクイメージ(modules1、modules2、modules3、およびmodules4)も同様に作成できます。それらのフロッピーディスクは、インストール時に準備を行いたいUSBまたはSCSIデバイス、ネットワークまたはPCMCIAカードがある場合に必要になります。インストール時に特殊なファイルシステムを使用する場合には、モジュールディスクも必要になります。

### 3.5.2 rawriteによるブートディスクの作成

DOSユーティリティのrawrite.exe (CD 1上のディレクトリdosutils/rawriteに格納されている)を使用すると、SUSEのブートおよびモジュールディスクを作成できます。このユーティリティを使用するには、DOS(たとえばFreeDOS)またはWindowsが稼働するコンピュータが必要です。

Windows XPでは、以下の手順に従います。

1. SUSE LINUXCD 1を挿入します。
2. DOSウィンドウを開きます(スタートメニューで、'アクセサリ' → 'コマンドプロンプト'の順に選択します)。
3. CDドライブのパスを指定して、rawrite.exeを実行します。以下の例では、現在のディレクトリはハードディスクC:上のWindowsであり、CDドライブはD:とします。

   ```
   d:\dosutils\rawrite\rawrite
   ```

4. スタートアップ時に、このユーティリティはファイルのコピー元とコピー先を尋ねます。ブートディスクのイメージは、CD 1上のディレクトリbootに格納されています。 ファイル名はbootdiskです。CDドライブのパスを指定することを忘れないでください。

   ```
   d:\dosutils\rawrite\rawrite
   RaWrite 1.2 - Write disk file to raw floppy diskette

   Enter source filename: d:\boot\bootdisk
   Enter destination drive: a:
   ```

コピー先ドライブa:を入力すると、rawriteは、フォーマット済みのフロッピーディスクを入れて、Enterを押すように促します。その後、コピー処理の進行状況が表示されます。この処理は、Ctrl-Cで終了できます。複数のフロッピーディスクを作成するには、同じ手順を繰り返します。

### 3.5.3 UNIX系システムでのブートディスクの作成

UNIXまたはLinuxシステムでは、CD-ROMとフォーマット済みのフロッピーディスクが必要です。以下の手順に従って、ブートディスクを作成します。

1. 最初にディスクをフォーマットする必要がある場合は、次のコマンドを入力します。

   ```
   fdformat /dev/fd0u1440
   ```

   このコマンドにより、フロッピーディスクにエラーがないかどうかも確認できます。エラーがあるメディアを使用して続行しないでください。

2. CD-ROMドライブにCD 1を挿入し、次の手順でCD上のbootディレクトリに変更します。現行のSUSEバージョンでは、CDをマウントする必要はありません。

```
cd /media/cdrom/boot
```

3. 次のコマンドでブートディスクを作成します。

```
dd if=bootdsk1 of=/dev/fd0 bs=8k
```

4. イメージbootdsk2およびbootdsk3で同様の手順を繰り返します。

bootディレクトリにあるREADMEファイルは、フロッピーディスクイメージに関する詳細情報を提供します。それらのファイルを読むには、moreコマンドまたはlessコマンドを使用します。

他のディスクイメージ(modules1、modules2、modules3、およびmodules4)も同様に作成できます。それらのフロッピーディスクは、インストール時に準備を行いたいUSBまたはSCSIデバイス、ネットワークまたはPCMCIAカードがある場合に必要になります。インストール時に特殊なファイルシステムを使用する場合には、モジュールディスクも必要になります。

モジュールディスクの作成は些細な作業ではありません。モジュールディスクの作成に関する詳細については、/usr/share/doc/packages/yast2-installation/vendor.htmlを参照してください。

### 3.5.4 フロッピーディスク(SYSLINUX)からのブート

ブートディスクを使用すると、特殊なインストール要件に対処できます(たとえば、CD-ROMドライブが使用できない場合)。ブート処理は、ブートローダSYSLINUX (パッケージsyslinux)によって開始されます。システムが起動すると、SYSLINUXは、以下のステップで構成される、最小限のハードウェア検出検査を実行します。

1. ブートローダは、BIOSがVESA2.0準拠のフレームバッファサポートを提供しているかどうかを調べ、適宜、カーネルを起動します。
2. モニタデータ(DDC info)が読み込まれます。
3. 1番目のハードディスクの最初のブロック(MBR)が読み込まれ、BIOS IDとLinuxのデバイス名がブートローダの設定時に対応付けられます。ブートローダは、BIOSのlba32関数を使用して当該ブロックを読み込み、BIOSがそれらの関数をサポートしているかどうかを判別します。

**Tip**

SYSLINUXの開始時に、Shiftキーを押したままにすると、上記のステップはすべてスキップされます。トラブルシューティングの目的で、

```
verbose 1
```

`syslinux.cfg`に次の行を挿入した場合、ブートローダは、現在実行中のアクションを表示します。

**Tip**

マシンがフロッピーディスクからブートしない場合は、BIOS内のブートシーケンスを`A,C,CDROM`に変更しなければならないことがあります。

► ***x86***
x86システムでは、CD 2からもブートできます。CD 1とは異なり、CD 2はブート可能なISOイメージを使用します。 CD 2は2.88 MBディスクイメージを使用してブートされます。CDからブートできることはわかっているが、CD 1からはブートできない場合、CD 2を使用します(代替ソリューション)。 ◄

### 3.5.5 サポート対象外のCD-ROMドライブ

ほとんどのCD-ROMドライブがサポートされています。CD-ROMドライブからブートできない場合は、CD-SetのCD 2からブートを試みてください。

システムにCD-ROMもフロッピーディスクもない場合でも、USB、FireWire、またはSCSI を使用して外部接続したCD-ROMを使用してシステムをブートできます。これは、BIOSおよびご利用のハードウェアのインタラクションに大きく依存します。問題が発生した場合、BIOSアップデートにより解決する場合があります。

### 3.5.6 ネットワークソースからのインストール

CD-ROMドライブを使用して標準インストールをできないことがときどきあります。たとえば、古い独自規格ドライブであるために、CD-ROMがサポートされない場合です。ラップトップのような補助的コンピュータは、CD-ROMドライブがまったく装備されておらず、Ethernetアダプタしかない場合があります。SUSE LINUXの場合、CD-ROMドライブが装備されていないコンピュータでも、ネットワーク接続を介してインストールを実行できます。通

常、こうしたインストールは、Ethernet経由でNFSまたはFTPを使用して行います。

この方法の場合のインストールはサポートされていません。したがって、次に示す手順を実行するユーザには、コンピュータの経験を積んでいることが求められます。

ネットワークソースからSUSE LINUXをインストールするために必要なステップは次の2つのステップです。

1. インストールに必要なデータ(CDやDVD)を、インストールソースとして機能するコンピュータで使用できるようにしておく。
2. インストールするシステムを、フロッピーディスク、CD、またはネットワークからブートし、ネットワークを設定する。

インストールソースは、NFSやFTPなど、さまざまなプロトコルを介して使用できるようになります。実際のインストールに関しては、項3.1.1. 「linuxrcへのパラメータの転送」を参照してください。

## 3.6 永続的デバイスファイル名のSCSI デバイスへの割り当て

システムをブートすると、SCSIデバイスに動的な方法でデバイスファイル名が割り当てられます。これはデバイス数または設定に変更がない限り問題にはなりません。ただし、新しいSCSIハードディスクが追加され、古いハードディスクが検出される前に新しいハードディスクが検出されると、古いディスクに新しい名前が割り当てられるためマウントテーブルのエントリ/etc/fstabは一致しません。

この問題を避けるため、システムスタートアップスクリプトboot.scsidevを使用できます。/sbin/insservを使用して、このスクリプトを有効化し、/etc/sysconfig/scsidevにあるこのスクリプト用のパラメータを設定します。スクリプト/etc/rc.d/boot.scsidevは、ブート処理中にSCSIデバイスのセットアップを処理し、/dev/scsi/下に永続的デバイス名を入力します。次にこれらの名前が、/etc/fstab内で使用されます。/etc/scsi.aliasが、SCSI設定における固定的な名前を定義するために使用されます。/etc/scsi内でのデバイスの命名規則は、man scsidevを参照してください。

ランレベルエディタのエキスパートモードで、レベルBに対してboot.scsidevを有効化します。ブート処理中に名前を生成するために必要なリンクが、次に/etc/init.d/boot.dに作成されます。

**Tip**

**デバイス名とudev**

SUSE LINUXの場合は、boot.scsidevが依然としてサポートされていますが、永続的なデバイス名を作成する適切な方法は、udevを使用して/dev/by-id/内の永続的な名前を使用するデバイスノードを作成することです。

**Tip**

# 3.7 LVMの設定

このセクションでは、LVMの基本原則と様々な状況で役立つ基本的な機能を簡単に説明します。 項3.7.2. 「YaSTを使用したLVMの設定」にてYaSTを使用したLVMのセットアップ方法を学びます。

**Warning**

LVMを使用することでデータ損失などの危険性が増加する恐れがあります。この危険性にはアプリケーションのクラッシュ、電源障害、誤ったコマンドなども含まれます。LVMまたはボリュームの再設定を実施する前にデータを保存してください。バックアップなしでは作業を実行しないでください。

**Warning**

## 3.7.1 論理ボリュームマネージャ(LVM)

論理ボリュームマネージャ(LVM)は、複数のファイルシステム上でハードディスクスペースを柔軟に割り振ることができます。これは、インストール中の初期パーティショニングを終了した後になってハードディスクスペースの区分を変更する必要が時として発生するために開発されました。稼動中のシステムでパーティションを変更することは困難なため、LVMは必要に応じて論理ボリューム(LV)を作成できるメモリスペースの仮想プール(ボリュームグループ(VG))を提供します。オペレーティングシステムは物理パーティションの代

わりにこれらのLVにアクセスします。ボリュームグループは2つ以上のディスクを使用することができます。また、複数のディスクまたはその一部が連続した1つのVGを形成することも可能です。この方法でLVMは物理ディスクスペースから一種の抽象層を提供します。この抽象層により、物理的にパーティショニングを再度行うよりもより簡単かつ安全な方法で区分に変更を加えられるようになります。物理パーティショニングに関連する背景情報については項1.5.4. 「パーティションのタイプ」および項2.7.5. 「パーティション」を参照してください。

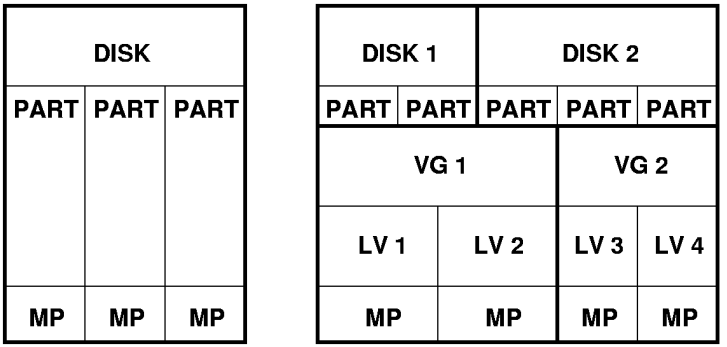


*Figure 3.1: 物理パーティショニング対LVM*

図 3.1. 「物理パーティショニング対LVM」では物理パーティショニング(左)とLVM区分(右)を比較しています。左側は、1つのディスクが割り当てられたマウントポイント(MP)をもつ3つの物理パーティション(PART)に分かれています。これによりオペレーティングシステムはそれぞれのパーティションにアクセスできます。右側では2つのディスクがそれぞれ3つの物理パーティションに分かれています。2つのLVMボリュームグループ(VG 1およびVG 2)が定義されています。VG 1にはDISK 1とDISK 2の2つのパーティションが含まれます。VG 2はDISK 2の2つのパーティションを除いた残り部分になります。LVMではボリュームグループに組み込まれた物理ディスクパーティションは物理ボリューム(PV)と呼ばれます。ボリュームグループ内に4つの論理ボリューム(LV 1からLV 4)が定義されています。これらのボリュームは、それぞれに関連づけられたマウントポイントを介してオペレーティングシステムに使用されます。 別の論理ボリュームとの境界とパーティションの境界を並べることはできません。この例ではLV 1およびLV 2の間に境界があります。

LVMの機能:

- 複数のハードディスクまたはパーティションを大きな論理ボリュームに組み込むことができる。
- 提供された設定が適切であれば、LV(/usrなど)は空きスペースがなくなったときに拡張することができます。
- LVMを使用することで、実行中のシステムにハードディスクまたはLVが追加されます。ただし、こうしたディスクやLVを追加するには、ホットスワップ可能なハードウェアが必要になります。
- 複数の物理ボリューム上に論理ボリュームのデータストリームを割り当てる「ストライピングモード」を有効にすることもできます。これらの物理ボリュームが別のディスクに存在する場合、RAID 0と同様に読み込みおよび書き込みのパフォーマンスを向上できます。
- スナップショット機能は稼動中のシステムで一貫性のある(特にサーバ)バックアップを取得できます。

これらの機能とともにLVMを使用することは、頻繁に使用されるホームPCや小規模サーバではそれだけでも意義があります。データベース、音楽アーカイブ、ユーザディレクトリなどの増え続けるデータストックがある場合は、LVMが最適と言えます。LVMは物理ハードディスクより大きなファイルシステムを利用できます。LVMのもう1つの利点は最大256個のLVを追加できることです。ただし、LVMでの作業は従来のパーティションでの作業とは異なることに留意してください。LVMの設定についての指示および詳しい情報はhttp://tldp.org/HOWTO/LVM-HOWTO/の公式LVM HOWTOからご利用いただけます。

カーネルバージョン2.6から開始して、LVMバージョン2を利用することができます。これはLVMの前バージョンとの下方互換になり、これまでのボリュームグループを管理できるようにします。新しいボリュームグループを作成する場合は、新しいフォーマットまたは下方互換バージョンのどちらを使用するか決定します。LVM 2にはいずれのカーネルパッチも必要ありません。LVM 2によりデバイスマッパーがカーネル2.6に統合されます。このカーネルはLVMバージョン2のみをサポートします。そのため、このセクションでLVMについて記述している場合は常にLVMバージョン2を参照します。

### 3.7.2 YaSTを使用したLVMの設定

YaSTLVM設定には、YaSTパーティションモジュールのエキスパートページ(項2.7.5. 「パーティション」を参照)からアクセスできます。この専用パー

ティショニングツールにより、既存のパーティションを編集、および削除できます。また、LVMで使用する新規パーティションを作成することもできます。次に'作成' → 'Do not format(フォーマットしない)'を最初にクリックし、続いて'0x8E Linux LVM'をパーティションIDとして選択します。LVMで使用するすべてのパーティションを作成した後に、'LVM'をクリックして、LVMの設定を開始します。

### ボリュームグループの作成

システムにまだボリュームグループが存在しない場合、ボリュームグループを追加するようにプロンプトされます(図 3.2. 「ボリュームグループの作成」を参照)。'Add group(グループを追加)'で追加グループを作成することができますが、通常はボリュームグループは1つで十分です。SUSE LINUXシステムファイルがあるボリュームグループの名前としては`system`が推奨されます。物理エクステントサイズではボリュームグループの物理ブロックサイズを定義します。ボリュームグループにある全ディスクスペースはこの物理ブロックサイズ内で使用されます。 この値は通常4MBに設定され、物理ボリュームおよび論理ボリュームには最大サイズとして256GB使用できます。物理エクステントは論理ボリュームとして256GB以上必要な場合のみ、8、16、32MBのように増やしてください。

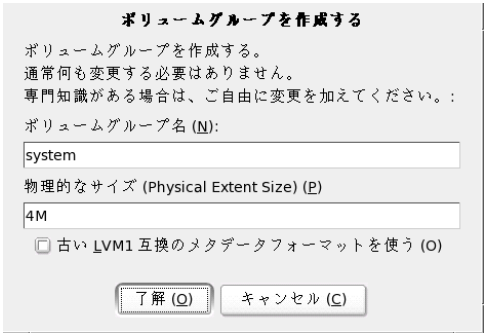


*Figure 3.2: ボリュームグループの作成*

### 物理ボリュームの設定

いったんボリュームグループが作成されると、続くダイアログで「Linux LVM」または「Linux Native」のすべてのパーティションがリストされます。

スワップパーティションまたはDOSパーティションは表示されません。パーティションがボリュームグループにすでに割り振られている場合、ボリュームグループの名前がリストに表示されます。未割り当てのパーティションは「--」で表示されます。

複数のボリュームグループが存在する場合は、選択ボックスで現在のボリュームグループを左上に設定します。右上にあるボタンは追加ボリュームグループの作成および既存ボリュームグループの削除を実行します。ボリュームグループのパーティションが未割り当ての場合のみ、そのボリュームグループを削除できます。ボリュームグループに割り当てられたすべてのパーティションも、同様に物理ボリューム(PV)として参照されます。

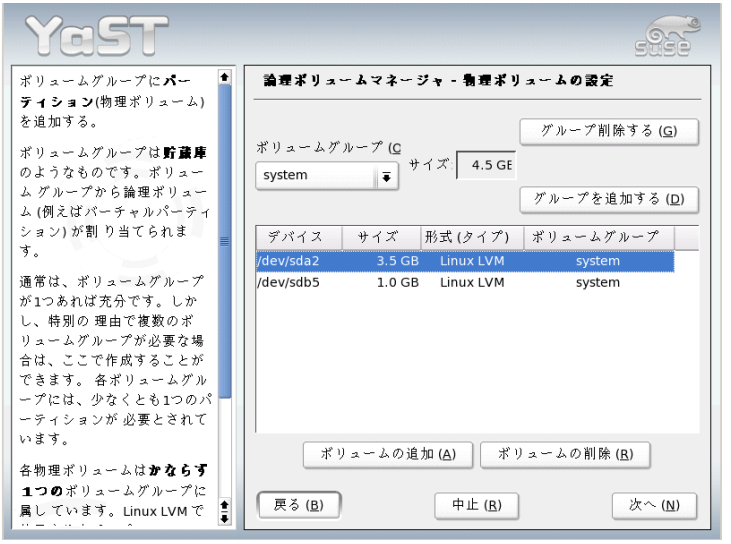


*Figure 3.3: 物理ボリュームの設定*

これまで未割り当てだったパーティションを選択したボリュームグループに追加するには、そのパーティションをクリックしてから［'ボリュームの追加'］をクリックします。この時点で、そのボリュームグループの名前が選択したパーティションの隣に入力されます。LVM用に予約されているパーティションをすべて1つのボリュームグループに割り当ててください。複数のボリュームグループに割り当てると、パーティションのスペースが未使用のまま残ります。ダイアログを終了する前に、すべてのボリュームグループを少なくと

も1つの物理ボリュームに割り当てる必要があります。すべての物理ボリュームを割り当て終えた後、［'次へ'］をクリックして論理ボリュームの設定に進みます。

### 物理ボリュームの設定

物理ボリュームにボリュームグループを設定し終えた後、次のダイアログでオペレーティングシステムが使用する論理ボリュームを定義します。現在のボリュームグループを選択ボックスで左上に設定します。設定したボリュームグループの隣に現在の空き領域が表示されます。下のリストにはボリュームグループの全論理ボリュームが表示されます。マウントポイントが割り当てられている通常の全Linuxパーティション、全スワップパーティション、既存の全論理ボリュームがここにリストされています。ボリュームグループのすべての領域がなくなるまで、必要に応じて論理ボリュームの［'追加'］、［'編集'］、［'削除'］を実行します。各ボリュームグループに少なくとも1つの論理ボリュームを割り当ててください。

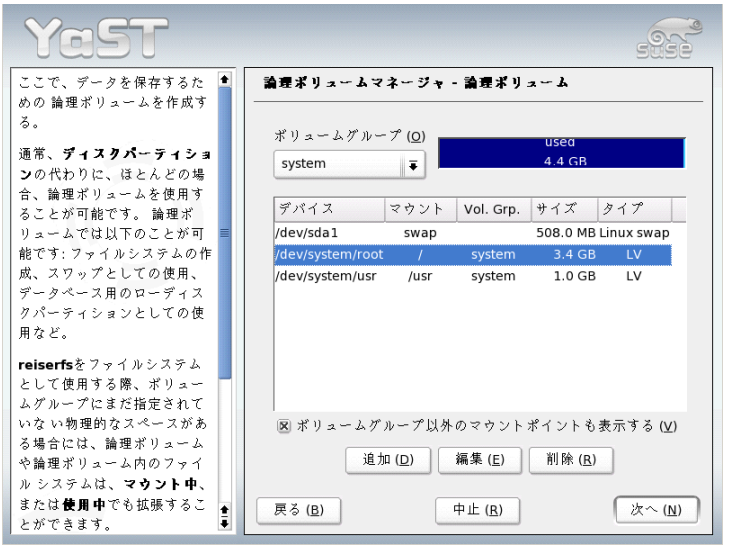


*Figure 3.4: 論理ボリューム管理*

新しい論理ボリュームを作成するには［'追加'］をクリックし、開いたポップアップの内容を埋めます。パーティショニングの場合、サイズ、ファイルシス

テム、およびマウントポイントを入力できます。通常、ReiserFSまたはExt2などのファイルシステムは論理ボリューム上に作成され、マウントポイントを指定します。この論理ボリューム上に格納されたファイルは、インストールしたシステム上の該当するマウントポイントで検出することができます。さらに、複数の物理ボリューム上(ストライピング)に存在する論理ボリュームにデータストリームを分配することも可能です。これらの物理ボリュームが別のハードディスクに存在する場合、この性質により、読み込みおよび書込みのパフォーマンスが向上します(RAID 0など)。ただし、nストライプでLVをストライピングする場合、LVが必要とするハードディスクスペースが物理ボリュームn個に等しく配分されている場合にのみ、ストライプが正しく作成されます。たとえば、使用可能な物理ボリュームが2個だけの場合、3個の論理ボリュームを持つことはできません。

**Warning**

**ストライピング**

YaSTには、現時点でストライピングの観点からエントリの正確性を確認する機会はありません。何か間違いがあった場合、それが明らかになるのはLVMがディスクに実装された後です。

**Warning**

すでにシステム上にLVMを設定した場合、ここで既存の論理ボリュームを指定することができます。続行する前に、これらの論理ボリュームを適切なマウントポイントに割り当てます。'次へ'でYaSTのパーティションモジュールのエキスパートページに戻り、ここでの設定作業を完了します。

#### LVMの直接管理

LVMをすでに設定し、一部に変更を加えるのみの場合は、別の方法でLVMにアクセスすることができます。YaSTコントロールセンターで'システム' → 'LVM'を選択します。基本的にこのダイアログで先に説明した、物理パーティショニングを除くアクションと同じことを実行できます。2つのリストに既存の物理ボリュームと論理ボリュームが表示されます。これにより、先に説明した方法を使用して、LVMシステムを管理できます。

## 3.8 ソフトウェアRAID設定

RAID (redundant arrays of inexpensive disks)の目的は、複数のハードディスクパーティションを1つの大きい仮想ハードディスクに結合し、パフォーマン


*Figure 3.5: 論 理 ボ リ ュ ー ム の 作 成*

スとデータのセキュリティを最適化することです。ただし、この方法を用いると、1つの利点が生かされるために他の利点が犠牲になります。ほとんどのRAIDコントローラはSCSIプロトコルを使用します。 これは、IDEプロトコルも効率的な方法で多数のハードディスクのアドレスを指定でき、コマンドのパラレル処理に適しているからです。一方、IDEまたはSATAハードディスクをサポートしているRAIDコントローラもあります。`http://cdb.suse.de`にアクセスして「Hardware Database」(ハードウェアデータベース)を参照してください。

### 3.8.1 ソフトウェアRAID

非常に高価な RAID コントローラと同様に、ソフトウェア RAID も上記のタスクを実行できます。SUSE LINUX では、YaST を使用することにより、複数のハードディスクを1つのソフトウェア RAID システムに結合するオプション、つまり、非常にリーズナブルな、ハードウェア RAID の代替機能を提供します。RAIDは、それぞれが異なる目標、利点、および属性を持ついくつかの

ハードディスクを1つのRAIDシステムに結合するためのいくつかの戦略を含んでいます。これらは通常、*RAID レベル*と呼ばれます。

一般的なRAIDレベルは次のとおりです。

**RAID 0** このレベルでは、各ファイルのブロックが複数のディスクドライブに分散されるので、データアクセスのパフォーマンスが向上します。 このレベルはデータのバックアップを提供しないため、実際にはRAIDではありませんが、この種のシステムでは*RAID 0*という名前が一般的です。RAID 0では、2つ以上のハードディスクが互いにプールします。高いパフォーマンスが得られます。 ただし、1つのハードディスクに障害が発生しただけで、RAIDシステムが破壊され、データは失われます。

**RAID 1** このレベルでは、データが他のハードディスクに一対一でコピーされるため、データに対する適切なセキュリティが提供されます。 これは、ハードディスクミラーリングとして知られています。一方のディスクが破壊された場合、そのディスク内容のコピーが他方のディスク上で利用できます。一方のディスクが破壊されても、データが危険にさらされることはありません。単一ディスクアクセスを使用した場合を比較すると、コピー処理において書き込みのパフォーマンスが若干、低下しますが(10から20%遅くなる)、読み取りアクセスは通常の物理ハードディスクに比べ、大幅に速くなります。 これは、データが複製されており、並列にスキャンできるためです。一般的に、レベル1は、単一ディスクのほぼ2倍の読み取りトランザクション速度と、単一ディスクとほぼ同じ書き込みトランザクション速度を提供します。

**RAID 2およびRAID 3** これらは、一般的なRAID実装ではありません。レベル2では、データは、ブロックレベルではなく、ビットレベルでストライプ化されます。レベル3は、専用パリティディスクによってバイトレベルのストライプ化を提供しますが、複数の要求を同時にサービスすることはできません。両方のレベルとも、使用されることはまれです。

**RAID 4** レベル4は、専用パリティディスクと結合されたレベル0と同様に、ブロックレベルのストライプ化を提供します。データディスク障害の場合、交換用ディスクを作成するために、パリティデータが使用されます。 ただし、パリティディスクは、書き込みアクセスの場合に障害となる可能性があります。にもかかわらず、レベル4は時々使用されます。

**RAID 5** RAID 5は、レベル0とレベル1の間をパフォーマンスおよび冗長性の面で調整して、最適化したものです。ハードディスクスペースは、使用されるディスク数から1を引いたものに等しくなります。データは、RAID 0の場合のようにハードディスク間で分散されます。 パーティ

ションの1つで作成されたパリティブロックがあるのは、セキュリティ上の理由からです。各パーティションはXORによって互いにリンクされているので、システム障害の場合に、XORを介して内容が対応するパリティブロックによって再構築されます。RAID 5の場合、同時に複数のハードディスクが障害を起こすことはありません。1つのハードディスクに障害がある場合は、そのハードディスクをできるだけ早く交換して、データ消失の危険性をなくす必要があります。

**その他のRAIDレベル** 他のRAIDレベル(RAIDn、RAID 10、RAID 0+1、RAID 30、RAID 50など)が開発されていますが、そのうちのいくつかはハードウェアベンダによって独自規格で作成される実装となります。これらのレベルは、広く使用されてはいないため、ここでの説明は省略します。

### 3.8.2 YaSTによるソフトウェアRAID設定

YaSTソフトウェアRAID設定には、YaSTのExpert Partitioner (項2.7.5. 「パーティション」を参照)からアクセスできます。このプロフェッショナル向けのパーティション設定ツールを使用すると、既存のパーティションを編集および削除したり、ソフトウェアRAIDで使用する新規パーティションを作成できます。ここでは、RAIDパーティションを作成します。最初に'作成' → 'Do not format (フォーマットしない)'の順にクリックし、次にパーティション識別子として'0xFD Linux RAID'を選択します。RAID 0およびRAID 1の場合、少なくとも2つのパーティションが必要です。RAID 1の場合、パーティションは2つだけです。RAID 5を使用する場合、少なくとも3つのパーティションが必要です。同じサイズのパーティションだけを使用するようにお勧めします。RAIDパーティションを異なるハードディスクに保存すると、1つが損傷した場合のデータ消失のリスクが削減され(RAID 1と5)、またRAID 0のパフォーマンスを最適化できます。RAIDで使用するすべてのパーティションを作成したら、'RAID' → 'Create RAID (RAIDの作成)'の順にクリックして、RAID設定を開始します。

次のダイアログでは、RAIDレベル 0、1、および5の間で選択します。詳細については、項3.8.1. 「ソフトウェアRAID」を参照してください。'次へ'をクリックすると、次のダイアログにタイプが「Linux RAID」または「Linux Native」であるすべてのパーティションのリストが表示されます(図 3.6. 「RAIDパーティション」を参照)。スワップパーティションまたはDOSパーティションは表示されません。パーティションがRAIDボリュームにすでに割り当てられている場合は、RAIDデバイスの名前(たとえば/dev/md0)がリストに表示されます。割り当てられていないパーティションは、"--"で示されます。


*Figure 3.6: RAIDパーティション*

前に割り当てを解除したパーティションを、選択したRAIDボリュームに追加するには、そのパーティションをクリックしてから、［'ボリュームの追加'］をクリックします。この時点で、そのRAIDデバイスの名前が選択したパーティションの隣に入力されます。すべてのパーティションをRAID用の予約パーティションとして割り当てます。すべてのパーティションを割り当てないと、パーティションのスペースが未使用のまま残ります。すべてのパーティションを割り当てたら、［'次へ'］をクリックして、設定ダイアログに進みます。このダイアログではパフォーマンスを微調整できます(図 3.7.「ファイルシステム設定」を参照)。

従来のパーティションの場合と同様の設定以外だけでなく、暗号化とRAIDボリュームのマウントポイントを使用するように、ファイルシステムを設定します。［'Persistent Superblock'］チェックボックスを有効にすると、起動時などにRAIDパーティションが認識されるようになります。［'完了'］をクリックして設定を完了した後で、パーティションモジュールのエキスパートページ上にある［*RAID*］と示された`/dev/md0`デバイスと他のデバイスを観察してください。


*Figure 3.7:* ファイルシステム設定

## 3.8.3 トラブルシューティング

/proc/mdstatsファイルを調べて、RAIDパーティションが破壊されているかどうかを調べます。システム障害が発生した場合は、Linuxシステムをシャットダウンして、問題のあるハードディスクを、同じ方法でパーティション分割されている新しいハードディスクで置き換えます。次に、システムを再起動して、mdadm /dev/mdX --add /dev/sdXコマンドを入力します。ここで、「X」を使用しているデバイス識別子に置き換えてください。これにより、ハードディスクがRAIDシステムに自動的に統合され、そのRAIDシステムが完全に再構築されます。

## 3.8.4 関連資料

ソフトウェアRAIDの設定方法と詳細情報が、次のHOWTOにあります。

- /usr/share/doc/packages/raidtools/Software-RAID-HOWTO.html

- http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html

Linux RAIDメーリングリストも使用できます。たとえば、http://www.mail-archive.com/linux-raid@vger.rutgers.eduなどがあります。

# システムおよびパッケージマネージメントの更新

SUSE LINUXには、完全な再インストールを行わないで、既存のシステムを更新できるオプションがあります。更新には次の2種類があります。*個別のソフトウェアパッケージの更新*とシステム*全体の更新*です。パッケージは、パッケージマネージャRPMを使用することにより、手動でインストールすることもできます。

# 4.1 SUSE LINUXの更新

ソフトウェアは、バージョンが上がるたびに「増加する」傾向があります。そのため、更新する前に、まずdfコマンドで、利用できるパーティションの容量を調べてください。ディスク容量が不足していると思われる場合は、システムの更新とパーティション設定を行う前に、データをバックアップしておきます。各パーティションに必要な容量を決定する一般的な規則はありません。必要な容量は、特定のパーティションプロファイル、選択したソフトウェア、およびSUSE LINUXのバージョン番号によって変わります。

**Important**

CDに収録されているREADMEファイル、あるいはDOSまたはWindows環境にあるREADME.DOSファイルを読んでください。このファイルには、このマニュアルが出版された後で加えられた変更点も収録されています。

**Important**

## 4.1.1 準備作業

更新を開始する前に、データを確保するために、古い設定ファイルを別のメディア(ストリーマ、取り外し可能なハードディスク、ZIPドライブなど)にコピーしておきます。主に、/etcの下に格納されているファイル、また、/varと/optの下にあるディレクトリとファイルの一部に当てはまります。さらに、/home (HOMEディレクトリ)下のユーザデータをバックアップメディアに書き込むようにします。このデータは、rootユーザでバックアップします。rootだけがすべてのローカルファイルを読み取るパーミッションを持っています。

更新を開始する前に、ルートパーティションの記録をとります。df /コマンドは、ルートパーティションのデバイス名リストを表示します。例 4.1.「df -hの出力例」に示すように、書き留めておくルートパーティションは、/dev/hda2です(/としてマウントされています)。

*Example 4.1: df -hの出力例*

```
Filesystem  Size  Used  Avail Use% Mounted on
/dev/hda1   1.9G  189M  1.7G  10%  /dos
/dev/hda2   8.9G  7.1G  1.4G  84%  /
/dev/hda5   9.5G  8.3G  829M  92%  /home
```

### 4.1.2 起こり得る問題

#### /etc内のpasswdとgroupのチェック

システムを更新する前に、/etc/passwdと/etc/groupに、構文エラーがまったく存在していないことを確認してください。この目的で、rootになって検証ユーティリティpwckとgrpckを起動し、報告されたエラーを取り除きます。

#### PostgreSQL

PostgreSQL (postgres)を更新する前に、データベースをダンプします。詳細については、pg_dumpのマニュアルページを参照してください。この作業が必要になるのは、更新の前にPostgreSQLを実際に使用している場合だけです。

### 4.1.3 YaSTによる更新

項4.1.1. 「準備作業」に概要を示した準備手順を実行しましたから、ここでシステムを更新できるようになります。

1. インストールの目的でシステムをブートします(項1.1. 「インストール時のシステム起動」を参照)。YaSTで言語を選択し、［'既存のシステムの更新'］を選択します。［'新規インストール'］を選択しないようにします。
2. YaSTは、複数のルートパーティションが存在するかどうか判定します。1つだけであれば、次のステップに進みます。いくつかある場合、正しいパーティションを選択して、［'次へ'］をクリックして確認します(項4.1.1. 「準備作業」の例で/dev/hda2を選択)。YaSTは、分析するこのパーティションの古いfstabを読み取り、そこにリストされたファイルシステムをマウントします。
3. 更新時にシステムファイルのバックアップコピーが作成される場合があります。このオプションは、更新処理を低速化します。このオプションは、最近バックアップを作成していない場合に使用します。
4. 次のダイアログで、すでにインストール済みのソフトウェアだけを更新するか、新規ソフトウェアコンポーネントをシステムに追加するか(アップグレードモード)のいずれかを選択します。提示されたコンポジション(例えば、［'デフォルトのシステム'］)を受け入れるようにします。後でYaSTを使用して調整できます。

### 4.1.4 個々のパッケージの更新

全体的な更新環境に関係なく、個々のパッケージは常に更新できます。この観点から言うと、システムで一貫性を維持するのはユーザの責任です。更新に関するアドバイスは、http://www.novell.com/linux/download/updates/から入手できます。

必要に応じてYaSTパッケージ選択リストからコンポーネントを選択します。システムの動作全般に必須のパッケージを選択した場合、YaSTでは警告が表示されます。そのようなパッケージは、更新モードでのみ更新します。たとえば、*共有ライブラリ*は多くのパッケージに含まれています。それらのプログラムとアプリケーションを稼働中のシステムで更新した場合、誤動作が起きることがあります。

## 4.2 バージョンごとのソフトウェアの変更点

バージョンごとの個別の変更は、以降で要約されています。この要約には、基本設定が完全に変更されているかどうか、設定ファイルが他の場所に移されているかどうか、共通アプリケーションが大幅に変更されているかどうかなどの情報が示されています。ユーザレベルまたは管理者レベルで日々のシステムの使用に影響を与える重要な変更が、ここに記載されています。

個別のバージョンの問題と特別な課題は、確認され次第、オンラインで公開されます。以下のリンクを参照してください。個々のパッケージに関する重要な更新については、http://www.novell.com/products/linuxprofessional/downloads/でYaSTオンラインアップデート(YOU)を使用してアクセスできます。項2.2.3. 「YaSTオンラインアップデート」を参照してください。

### 4.2.1 8.1から8.2への更新

問題と特別な課題: http://portal.suse.com/sdb/en/2003/04/bugs82.html

- nVidiaベースのグラフィックカード用の3Dサポート(変更): RPM NVIDIA_GLX/NVIDIA_kernel(スクリプトswitch2nvidia_glxを含む)は、含まれなくなりました。Linux IA32用のnVidiaインストー

ラをnVidia Webサイト(http://www.nvidia.com)からダウンロードして、このインストーラでドライバをインストールし、SaX2またはYaSTを使って3Dサポートを有効にします。

- 新規インストールでは、xinetdがinetdの代わりにインストールされ、安全な値で設定されます。ディレクトリ/etc/xinetd.dを参照してください。ただし、システムの更新の場合は、inetdが保持されます。

- 現在のPostgreSQLのバージョンは7.3です。バージョン7.2.xから移行するときは、pg_dumpで*dump/restore*を実行します。バージョン7.3ではスキーマが導入されているため、アプリケーションがシステムカタログでクエリを行う場合は、さらに作業が必要になります。詳細については、http://www.ca.postgresql.org/docs/momjian/upgrade_tips_7.3を参照してください。

- バージョン4のstunnelでは、コマンドラインオプションはサポートされなくなりました。ただし、付属のスクリプト/usr/sbin/stunnel3_wrapperによってコマンドラインオプションをstunnelに適した設定ファイルに変換できるので、プログラムの起動時にそのファイルを使用します(OPTIONSを自分のオプションで置き換えます)。

  ```
  /usr/sbin/stunnel3_wrapper stunnel OPTIONS
  ```

  作成された設定ファイルはデフォルトの出力に送られ、永続的な設定ファイルを作成するためにそれらの仕様を使用できるようにします。

- openjade (openjade)は、db2x.sh (docbook-toys)が実行される場合に、jade (jade_dsl)の代わりに現在使用されているDSSSLエンジンです。互換性のために、個別のプログラムもプレフィックスoなしで使用できます。

  自身のアプリケーションがディレクトリjade_dslおよび前にそこにインストールされたファイルに依存している場合は、そのアプリケーションを新しいディレクトリ/usr/share/sgml/openjadeに適用するか、リンクをrootとして作成します。

  ```
  cd /usr/share/sgml
  rm jade_dsl
  ln -s openjade jade_dsl
  ```

  rzszとの競合を避けるために、コマンドラインツールsxは、引き続きs2x、sgml2xml、またはosxという名前になります。

### 4.2.2 8.2から9.0への更新

問題と特別な課題: `http://portal.suse.com/sdb/en/2003/07/bugs90.html`

- バージョン4のRPMパッケージマネージャが利用できるようになりました。パッケージをビルドする機能は、別のプログラム`rpmbuild`に移されました。`rpm`は、引き続き、インストール、更新、およびデータベースクエリに使用されます。項4.3. 「RPM—パッケージマネージャ」を参照してください。
- パッケージ`foomatic-filters`が印刷に使用できるようになりました。CUPS がインストールされていなくても印刷に使用できるように、コンテンツが`cups-drivers`から分離されています。このように、YaSTは、印刷システム(CUPS、LPRng)から独立している設定をサポートします。 このパッケージの設定ファイルは`/etc/foomatic/filter.conf`です。
- パッケージ`foomatic-filters`と`cups-drivers`が、LPRngとlpdfilterにも必要になりました。
- 付属のソフトウェアパッケージのXMLリソースには、`/etc/xml/suse-catalog.xml`内のエントリを用いてアクセスできます。このファイルを`xmlcatalog`コマンドで編集してはなりません。編集すると、正しい更新に必要な構造用コメントが削除されます。`/etc/xml/suse-catalog.xml`は、`/etc/xml/catalog`内の`nextCatalog`ステートメントでアクセスされ、`xmllint`や`xsltproc`のようなXMLツールによってローカルのリソースが自動的に検出できるようになります。

### 4.2.3 9.0から9.1への更新

`http://portal.suse.com`にアクセスし、キーワード*special features*を使用して、“SUSE Support Database (サポートデータベース)で記事「Known Problems and Special Features in SUSE LINUX 9.1」(SUSE LINUX 9.1″で判明している問題と特殊機能)を参照してください。これらの記事は、SUSE LINUXのバージョンごとに公開されます。

### カーネル2.6への更新

SUSE LINUXは、完全にカーネル2.6に基づいています。以前のバージョン2.4は、付属のアプリケーションがカーネル2.4では動作しないので、使用してはいけません。以下の情報にも注意してください。

- モジュールのロードは、ファイル/etc/modprobe.confで設定されるようになりました。ファイル/etc/modules.confは廃止されました。YaSTは、このファイルの変換を試みます(スクリプト/sbin/generate-modprobe.confも参照してください)。
- モジュールにはサフィックス.koが付けられています。
- CDを作成するのに、モジュールide-scsiは必要なくなりました。
- プレフィックスsnd_は、ALSAサウンドモジュールオプションから削除されました。
- sysfsは、/procファイルシステムを補完するようになりました。
- 電源管理(特にACPI)が改善され、YaSTモジュールで設定できるようになりました。

### VFATパーティションのマウント

VFATパーティションをマウントする場合、code=パラメータをcodepage=に変更する必要があります。VFATパーティションを容易にマウントできない場合、/etc/fstabファイル内に、古いパラメータ名が含まれているかどうか確認します。

### ACPIによるスタンバイとサスペンド

新しいカーネル2.6は、ACPIによるスタンバイとサスペンドをサポートするようになりました。この機能は依然として実験段階にあり、いくつかのハードウェアコンポーネントはサポートしていない可能性があります。この機能を使用するには、powersaveパッケージが必要です。このパッケージについては、/usr/share/doc/packages/powersaveを参照してください。グラフィカルフロントエンドは、kpowersaveパッケージにより使用できます。

### 入力デバイス

入力デバイス関連の変更については、http://portal.suse.comにアクセスし、Support Databaseでキーワード*special features*を使用して上記のPortalの記事"Known Problems and Special Features in SUSE LINUX 9.1" (SUSE LINUX 9.1で判明している問題と特殊機能)を参照してください。

### ネイティブPOSIXスレッドライブラリとglibc2.3.x

NGPT (*Next Generation POSIX Threading*)にリンクされているアプリケーションは、glibc2.3.xでは動作しません。SUSE LINUXで提供されていない、影響を受けるすべてのアプリケーションは、linuxthreadsまたはNPTL (*Native POSIX Thread Library*)でコンパイルしなければなりません。将来は標準となるので、NPTLを使用してください。

NPTLで問題が発生した場合は、以下の環境変数を設定することにより、以前のlinuxthreadsの実装を使用できます(⟨*kernel-version*⟩をカーネルのバージョン番号で置き換えます)。

```
LD_ASSUME_KERNEL=kernel-version
```

以下のバージョン番号を指定できます。

**2.2.5 (i386、i586):** フローティングスタックのないlinuxthreads

**2.4.1 (AMD64, i586, i686):** フローティングスタックのあるlinuxthread

カーネルとフローティングスタックのあるinuxthreadsに関する注意事項:`errno`、`h_errno`、および`_res`を使用するアプリケーションは、`#include`でヘッダファイル(`errno.h`、`netdb.h`、および`resolv.h`)をインクルードしなければなりません。*thread cancellation*を使用するマルチスレッド対応のC++プログラムでは、環境変数`LD_ASSUME_KERNEL=2.4.1`を使用して、linuxthreadsライブラリを使用するようにします。

### ネイティブPOSIXスレッドライブラリへの適応

NPTLは、SUSE LINUX9.1にスレッドパッケージとして含まれています。NPTLは、以前のlinuxthreadsライブラリとバイナリ互換性があります。ただし、linuxthreadsがPOSIX標準に準拠していない場合は、NPTLを使用する必要があります。これにはシグナル処理が含まれます。`getpid`はすべてのスレッドで同じ値を返し、pthread_atforkに登録されたスレッドハンドラは、`vfork`が使用された場合には動作しません。

### ネットワークインタフェース設定

ネットワークインタフェースの設定方法は変更されました。従来、存在しないインタフェースを設定した後で、ハードウェアの初期化が行われていました。現在は、システムは新しいハードウェアの検索を行い、それを即座に初期化し、新しいネットワークインタフェースの設定を有効にします。

設定ファイルに関して、新しい名前が採用されました。ネットワークインタフェイスの名前は動的に生成され、ホットプラグデバイスの使用はいっそう安定します。eth0、eth1などの名前は、設定の目的では適切ではなくなりました。この理由により、インタフェース設定に名前を付ける際に、MACアドレスまたはPCIスロットのような一意の呼び名が使用されています。インタフェース名が表示された直後にそれを使用できます。ifup eth0やifdown eth0のようなコマンドは引き続き使用可能です。

デバイス設定は、/etc/sysconfig/hardwareに配置されます。これらのデバイスによって実現されるインタフェースは通常、/etc/sysconfig/network内に(互いに異なる名前で)配置されます。詳細については、/usr/share/doc/packages/sysconfig/READMEを参照してください。

### サウンド設定

更新の後、サウンドカードを再設定する必要があります。これは、YaSTサウンドモジュールを使用して行うことができます。rootで、yast2 soundコマンドを入力します。

### トップレベルドメイン.localをリンクローカルドメインとして扱う

リゾルバライブラリは、トップレベルドメイン.localを"link-local" (リンクローカル)ドメインとして扱い、通常のDNSクエリと異なり、マルチキャストDNSクエリをマルチキャストアドレス224.0.0.251のポート5353へ送信します。これは、互換性のない変更です。ドメイン.localがネームサーバ設定の際にすでに使用されている場合、他のドメイン名を使用してください。マルチキャストDNSの詳細については、http://www.multicastdns.orgを参照してください。

### システム全体のUTF-8エンコード

現在、システムのデフォルトのエンコーディングはUTF-8です。そのため、標準インストールを実行すると、ロケールは.UTF-8エンコーディングに設定されます(en_US.UTF-8など)。詳細については、http://www.suse.de/~mfabian/suse-cjk/locales.htmlを参照してください。

### UTF-8へのファイル名変換

以前に作成されたファイルシステム内にあるファイルは、(明示的に指定した場合を除き)ファイル名のエンコードとしてUTF-8を使用していません。ASCII以外の文字がファイル名の一部として使用されている場合、それらは文字化けになります。この点を訂正するには、convmvスクリプトを使用します。これは、ファイル名のエンコードをUTF-8に変換します。

### 2001年のPOSIX標準と互換性のあるシェルツール

デフォルト設定では、coreutilsパッケージに含まれるシェルツール(tail、chown、head、sortなど)は、1992年のPOSIX標準には準拠せず、2001年のPOSIX標準(*統一UNIX仕様バージョン3 == IEEE Std 1003.1-2001 == ISO/IEC 9945:2002*)に準拠します。環境変数を使用することにより、以前のバージョンの動作を強制できます。

```
_POSIX2_VERSION=199209
```

新しい値は200112であり、_POSIX2_VERSIONのデフォルト値として使用されます。SUS標準は、http://www.unix.orgから入手できます(無償ですが、登録が必要です)。

***Table 4.1:*** *POSIX 1992とPOSIX 2001の比較*

| POSIX 1992 | POSIX 2001 |
|---|---|
| chown tux.users | chown tux:users |
| tail +3 | tail -n 3 |
| head -1 | head -n 1 |
| sort +3 | sort -k 4 |
| nice -10 | nice -n 10 |
| split -10 | split -l 10 |

**Tip**

サードパーティ製のソフトウェアは、まだ新しい標準に準拠していない場合があります。その場合は、環境変数を上記で説明したように設定してください。

**Tip**

### /etc/gshadowの廃止

/etc/gshadowは、以下の理由により余分であるため、廃棄および削除されました。

- glibcでサポートされていない。

- このファイル用の公式インタフェースがない。シャドウスイートにも、この種のインタフェースは含まれていません。
- グループパスワードをチェックするほとんどのツールは、このファイルをサポートしておらず、上記の理由で無視する。

### OpenLDAP

データベースの形式が変更されたので、データベースを再生成する必要があります。更新の実行中に、システムはこの変換を自動的に行うことを試みます。しかし、変換が失敗する状況が存在するのも確かです。

スキーマチェックには大掛かりな改良が加えられました。したがって、従来のLDAPサーバでは実行可能だった多くの(標準に準拠していない)操作は、現在は実行できません。

設定ファイルの構文は、ACLの観点に基づいて一部変更されました。インストール後、更新に関する情報は、`/usr/share/doc/packages/openldap2/README.update`ファイル内で参照できます。

### Apache 1.3からApache 2への置き換え

Apache Webサーバ(バージョン1.3)は、Apache 2によって置き換えられました。バージョン2.0の詳細マニュアルは、Webページ`http://httpd.apache.org/docs-2.0/en/`に用意されています。HTTPサーバがインストールされているシステムでは、更新によりApacheパッケージが削除され、Apache 2がインストールされます。その後、YaSTを使用するか手動でシステムを調整する必要があります。`/etc/httpd`内にあった設定ファイルは、更新後は`/etc/apache2`内に配置されています。

複数の同時クエリを処理するために、スレッドまたはプロセスを選択できます。バージョン2.0では、プロセス管理が、マルチプロセシングモジュール(MPM)という独立したモジュールに移動されています。したがって、Apache 2は、`apache2-prefork`パッケージ(安定性のためにこちらを推奨)、または`apache2-worker`パッケージを必要とします。クエリに対するApache2の反応は、MPMに応じて異なります。これは、パフォーマンスおよびモジュールの使用方法に影響します。これらの特性の詳細については、項30.4. 「スレッド」を参照してください。

Apache 2では、次世代のインターネットプロトコルIPv6もサポートされるようになりました。

モジュールのプログラマがモジュールに必要なロードシーケンスを指定し、ユーザをこのタスクから解放できるようにするためのメカニズムが実装されて

います。通常、モジュールの実行順序が重要であり、ロードシーケンスを使用して決定されます。たとえば、アクセス権を持たないユーザはページを表示できないように、認証済みユーザにのみ特定のリソースへのアクセス権を与えるモジュールを最初にロードする必要があります。

Apacheとの間のクエリが、フィルタによって処理できます。

### Samba~2.xからSamba~3.xへの更新

Samba~2.xからSamba~3.xへの更新を行った後、winbind認証はもう使用できません。他の認証方法は引き続き使用できます。この理由で、次のプログラムは削除されました。

```
/usr/sbin/wb_auth
/usr/sbin/wb_ntlmauth
/usr/sbin/wb_info_group.pl
```

http://www.squid-cache.org/Doc/FAQ/FAQ-23.html#ss23.5も参照してください。

### OpenSSHの更新(バージョン3.8p1)

潜在的なMITM攻撃を防止するために、gssapiのサポートは、gssapi-with-micによって置き換えられました。これら2つのバージョンは、互いに互換性がありません。このことは、古いディストリビューションのKerberosチケットを認証できないことを意味します。異なる認証方法が使用されているからです。

### SSHと端末アプリケーション

バージョン9 (UTF-8を有効にした標準設定)と、それより古いシステム(SUSE LINUX9.0とそれより前のバージョンですが、これらではUTF-8 がデフォルトでは有効になっていないか、サポートされていません)の間で、リモートホスト(特にSSH、telnet、およびRSH)からの接続を確立する場合、端末アプリケーションが正しくない文字を表示することがあります。

これは、OpenSSHがローカル設定を送信しないことが原因です。したがって、デフォルトシステム設定が使用されていますが、それはリモート端末の設定と一致しない可能性があります。これは、テキストモードのYaSTと、通常のユーザ(rootではない)がリモートホストから実行するアプリケーションに影響を及ぼします。rootによって開始されたアプリケーションは、root用の標準ロケール(デフォルトではLC_CTYPEのみが設定されています)をユーザが変更した場合にのみ、影響を受けます。

#### libiodbcの廃棄

FreeRADIUSのユーザは、現在はunixODBCにリンクする必要があります。libiodbcは廃棄されたからです。

#### /usr/share/xml内のXML リソース

FHS (項A. 「規格と仕様」を参照)では、XMLリソース(DTD、スタイルシートなど)が /usr/share/xmlにインストールされていることが要求されます。そのため、いくつかのディレクトリが/usr/share/sgmlで使用できなくなりました。問題が発生した場合は、スクリプトおよびmakefileを修正するか、オフィシャルカタログ(特に /etc/xml/catalogまたは/etc/sgml/catalog)を使用します。

#### リムーバブルメディアとsubfs

リムーバブルメディア(取り外し可能メディア)は、subfsに統合されました。mountを使用してメディアを手動でマウントする必要はなくなりました。メディアをマウントするには、単に/media内の関連デバイスのディレクトリに移動します。プログラムがメディアにアクセスしている間、そのメディアをイジェクト(取り出し)することはできません。

### 4.2.4 9.1から9.2への更新

http://portal.suse.comにアクセスし、キーワード*special features*を使用して、"SUSE Support Database(サポートデータベース)で記事「Known Problems and Special Features in SUSE LINUX 9.2」(SUSE LINUX 9.1"で判明している問題と特殊機能)を参照してください。

#### インストール時の提案ダイアログでのファイアウォールの有効化

セキュリティレベルを上げるために、提案ダイアログでインストールを終了すると、同梱のファイアウォールソリューションSuSEFirewall2が有効になります。これは、最初はすべてのポートがクローズされており、必要に応じて提案ダイアログでオープンできることを意味します。デフォルトでは、リモートシステムからログインできません。SLP、Samba (「ネットワークコンピュータ」)、ある種のゲームなど、ネットワーク参照アプリケーションおよびマルチキャストアプリケーションとのインタフェースにもなります。YaSTを使用してファイアウォールを微調整できます。

サービスのインストールまたは設定中にネットワークへのアクセスを必要とする場合は、関連YaSTモジュールにより、すべての内部インタフェースと外部

インタフェースの必須TCPポートおよびUDPポートがオープンされます。これが必要ない場合は、YaSTモジュールでユーザはポートを閉じるか、他の詳しいファイアウォール設定を指定できます。

*Table 4.2: 重要なサービスで使用されるポート*

| サービス | ポート |
|---|---|
| HTTPサーバ | ファイアウォールは"list"ステートメントに従って設定(TCPのみ) |
| メール(postfix) | smtp 25/TCP |
| Sambaサーバ | netbios-ns 137/TCP、netbios-dgm 138/TCP、netbios-ssn 139/TCP、microsoft-ds 445/TCP |
| DHCPサーバ | bootpc 68/TCP |
| DNSサーバ | domain 53/TCP、domain 53/UDP |
| DNSサーバ | SuSEFirewall2のポートマッパに対する特殊サポートをプラス |
| ポートマッパ | sunrpc 111/TCP、sunrpc 111/UDP |
| NFSサーバ | nfs 2049/TCP |
| NFSサーバ | ポートマッパをプラス |
| NISサーバ | portmapを有効化 |
| TFTP | tftp 69/TCP |
| CUPS (IPP) | ipp 631/TCP、ipp 631/UDP |

### KDEとGNOMEのサポート

デフォルトでは、KDEにはIPv6サポートは有効でありません。YaSTの`/etc/sysconfig`エディタを使用して有効にすることができます。この機能を無効にする理由は、一部のインターネットサービスプロバイダではIPv6アドレスが正しくサポートされないからです。結果として、Webの検索中にエラーメッセージが表示され、Webページの表示で遅れが生じます。

### YaSTオンラインアップデートと「デルタパッケージ」

YaSTオンラインアップデートは、基本パッケージからの差分のみを格納する特殊なRPMパッケージをサポートするようになっています。この方法の

場合、最終的なパッケージの再構成のためCPUの負荷が高くなるという欠点はありますが、パッケージのサイズとダウンロード時間の面では大幅な削減が見られます。/etc/sysconfig/onlineupdateでは、YOUがこうした「デルタパッケージ」を使用するかどうかを設定します。詳細については、file:///usr/share/doc/packages/deltarpm/READMEを参照してください。

### 印刷システムの設定

インストールの終了時に(提案ダイアログ)、印刷システムに必要なポートをファイアウォール設定でオープンする必要があります。ポート631/TCPとポート631/UDPはCUPSに必須であり、通常の動作ではクローズしないでください。LPDまたはSMBを介して印刷を行うには、ポート515/TCP (古いLPDプロトコル用)とSambaで使用されるポートもオープンする必要があります。

### X.Orgへの移行

互換リンクを使用すると、XFree86からX.Orgに容易に移行できます。このリンクにより、重要なファイルとコマンドに古い名前でアクセスできます。

*Table 4.3: コマンド*

| XFree86 | X.Org |
|---|---|
| XFree86 | Xorg |
| xf86config | xorgconfig |
| xf86cfg | xorgcfg |

*Table 4.4: /var/log内のログファイル*

| XFree86 | X.Org |
|---|---|
| XFree86.0.log | Xorg.0.log |
| XFree86.0.log.old | Xorg.0.log.old |

X.Orgに移行する過程で、パッケージ名がXFree86*からxorg-x11*に変更されました。

### X11用のターミナルエミュレータ

多くのターミナルエミュレータを削除しました。それらのターミナルエミュレータはデフォルト環境ではメンテナンスされず、また機能しません。特にUTF-8をサポートしていません。SUSE LINUXは、xterm、KDE、GNOMEといった端末やmlterm (Multilingual Terminal Emulator for X)などの標準端末を提供しています。これらは、atermおよびetermに置き換わるものです。

### powersaveパッケージの変更

`/etc/sysconfig/powersave`内の設定ファイルが変更されています。

*Table 4.5: /etc/sysconfig/powersave内の設定ファイルの分割*

| 旧 | 分割後 |
|---|---|
| `/etc/sysconfig/powersave/common` | `common` |
| | `cpufreq` |
| | `events` |
| | `battery` |
| | `sleep` |
| | `thermal` |

`/etc/powersave.conf`は、廃棄されました。既存の変数は表 4.5.「/etc/sysconfig/powersave内の設定ファイルの分割」に示すファイルに移動されています。`/etc/powersave.conf`内で"event"を変数を変更している場合は、これらの変数を`/etc/sysconfig/powersave/events`内で調整する必要があります。

スリープ状態の名前が次のように変更されました。変更前の名前は次のとおりです。

- suspend (ACPI S4,、APMサスペンド)
- standby (ACPI S3、APMスタンドバイ)

変更後の名前は次のとおりです。

- suspend to disk (ACPI S4、APMサスペンド)

- suspend to ram (ACPI S3, APM サスペンド)
- standby (ACPI S1、APMスタンドバイ)

### OpenOffice.org (OOo)

**ディレクトリ:** OOoは、/opt/OpenOffice.orgの代わりに/usr/lib/ooo-1.1にインストールされます。ユーザ設定用のデフォルトディレクトリは、~/OpenOffice.org1.1ではなく~/.ooo-1.1です。

**ラッパー:** OOoコンポーネントの起動用に、いくつか新規ラッパーが用意されています。新しい名前を表 4.6. 「ラッパー」に示します。

*Table 4.6:* ラッパー

| 旧 | 新 |
|---|---|
| /usr/X11R6/bin/OOo-calc | /usr/bin/oocalc |
| /usr/X11R6/bin/OOo-draw | /usr/bin/oodraw |
| /usr/X11R6/bin/OOo-impress | /usr/bin/ooimpress |
| /usr/X11R6/bin/OOo-math | /usr/bin/oomath |
| /usr/X11R6/bin/OOo-padmin | /usr/sbin/oopadmin |
| /usr/X11R6/bin/OOo-setup | – |
| /usr/X11R6/bin/OOo-template | /usr/bin/oofromtemplate |
| /usr/X11R6/bin/OOo-web | /usr/bin/ooweb |
| /usr/X11R6/bin/OOo-writer | /usr/bin/oowriter |
| /usr/X11R6/bin/OOo | /usr/bin/ooffice |
| /usr/X11R6/bin/OOo-wrapper | /usr/bin/ooo-wrapper |

ラッパーは、KDEアイコンとGNOMEアイコンの間で切り替えるためのオプション--icons-setをサポートするようになりました。次のオプションはもうサポートさｒていません。--default-configuration、--gui, --java-path、--skip-check、--lang (言語はロケールにより決定)、--messages-in-window、および--quiet。

**KDEとGNOMEのサポート** OpenOffice_org-kdeおよびOpenOffice_-

org-gnomeパッケージ内でKDEおよびGNOME拡張を使用できます。

### サウンドミキサーkmix

サウンドミキサーkmixがデフォルトとして事前設定されています。ハイエンドハードウェアの場合は、他にもQAMix/KAMix、envy24control (ICE1712のみ)、またはhdspmixer (RME Hammerfallのみ)などのミキサーがあります。

### DVD作成

従来、DVD作成をサポートするために、パッチをcdrecordパッケージからcdrecordバイナリに適用しました。現在、このパッチ付きの新規バイナリcdrecord-dvdがインストールされています。

dvd+rw-toolsパッケージのgrowisofsプログラムは、現在ではすべてのDVDメディア(DVD+R、DVD-R、DVD+RW、DVD-RW、DVD+RL)を作成できるようになりました。パッチされたcdrecord-dvdの代わりにこれを使用するようにお勧めします。

### 複数カーネル

複数のカーネルを並べてインストールできます。この機能の目的は、管理者が新規カーネルをインストールし、新規カーネルが期待通りに機能することを検証したのち、古いカーネルをアンインストールすることによって、カーネルをアップグレードできるようにすることです。YaSTはまだこの機能をサポートしていませんが、rpm -i ⟨*package*⟩.rpmを使用すれば、カーネルのインストールとアンインストールはシェルから簡単に行うことができます。コマンドラインからのパッケージの管理については、項4.3. 「RPM—パッケージマネージャ」を参照してください。

デフォルトのブートローダメニューには、1つのカーネルエントリがあります。複数のカーネルをインストールする場合、追加するカーネルごとに1つのエントリを追加し、それらを簡単に選択できるようにしておくと便利です。新規カーネルのインストール前にアクティブであったカーネルは、vmlinuz.previousおよびinitrd.previousとしてアクセスできます。デフォルトエントリに似たブートローダエントリを作成し、このエントリがvmlinuzおよびinitrdではなく、vmlinuz.previousおよびinitrd.previousを参照するようにすると、前にアクティブであったカーネルにアクセスできます。またGRUBおよびLILOも、ワイルドカードのブートローダエントリをサポートします。詳細については、GRUBのinfoページ(info grub)およびlilo.conf (5)マニュアルページを参照してください。

### 4.2.5 9.2から9.3への更新

http://portal.suse.comにアクセスし、キーワード*special features*を使用して、"SUSE Support Database (サポートデータベース)で記事「Known Problems and Special Features in SUSE LINUX 9.3」(SUSE LINUX 9.1"で判明している問題と特殊機能)を参照してください。

#### カーネルプロンプトでの手動インストールの開始

［'手動インストール'］モードは、ブートローダの画面からなくなっています。それでも、ブートプロンプトでmanual=1を使用すれば、inuxrcを手動モードにすることはできます。通常ではこれは必要ありません。textmode=1のようにインストールオプションをカーネルプロンプトで直接設定するか、インストールソースとしてURLを設定できるからです。

#### ネットワーク認証用Kerberos

Kerberosがネットワーク認証のデフォルトです。heimdalではありません。既存のheimdal設定の自動変換は行えません。システム更新の場合、設定ファイルのバックアップコピーが表 4.7.「バックアップファイル」に示すように作成されます。

*Table 4.7: バックアップファイル*

| 古いファイル | バックアップファイル |
|---|---|
| /etc/krb5.conf | /etc/krb5.conf.heimdal |
| /etc/krb5.keytab | /etc/krb5.keytab.heimdal |

クライアント設定(/etc/krb5.conf)は、heimdaの1つによく似ています。特に何も設定することがなかった場合は、パラメータkpasswd_-serverをadmin_serverへ置き換えることで十分です。

サーバ(kdc/kadmind)関連データを引き継ぐことはできません。システム更新後、古いheimdalデータベースは、/var/heimdalの下でまだ使用可能です。MIT kerberosは、/var/lib/kerberos/krb5kdcの下のデータベースをメンテナンスします。

### X.Org設定ファイル

設定ツールSaX2は、X.Org設定を/etc/X11/xorg.confに書き込みます。最初からのインストール時には、XF86Configからxorg.confへの互換性のないリンクが作成されます。

### PAM設定

**common-auth** 認証セクション用のデフォルトPAM設定

**common-account** アカウントセクション用のデフォルトPAM設定

**common-password** パスワード変更用デフォルトPAM設定

**common-session** セッション管理用デフォルトPAM設定

アプリケーション固有設定ファイル内からこれらのデフォルト設定ファイルをインクルードします。システム上での存在のためにほぼ40ファイルを使用するより、1つの設定ファイルを変更して、メンテナンスするほうが簡単であるからです。後でアプリケーションをインストールすると、そのアプリケーションはすでに適用済みの変更を継承するので、管理者は、設定を調整するために覚えておく必要がありません。

変更は次のように簡単です。次の設定ファイルがあるとします(このファイルは、ほとんどのアプリケーションのデフォルトです)。

```
#%PAM-1.0 auth     required       pam_unix2.so account  required       pam_un
```

次のものに変更できます。

```
#%PAM-1.0
auth     include        common-auth
account  include        common-account
password include        common-password
session  include        common-session
```

## 4.3 RPM—パッケージマネージャ

SUSE LINUXでは、RPM (Red Hat Package Manager)がソフトウェアパッケージの管理に使用されます。RPMの主要なプログラムは、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ーを行って、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。

rpmの基本的なモードは、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、またはアップデート、RPMデータベースの再構築、RPMベースまたは個別RPMアーカイブのクエリー、パッケージの整合性チェック、パッケージの署名の5つです。 rpmbuildは、基本ソースからインストール可能パッケージをビルドするために使用できます。

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。

rpmは、LSB互換のパッケージを管理するのに使用できます。LSBについては、項A. 「規格と仕様」を参照してください。

**Tip**

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られています。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、kdelibs-develなどのように、名前の拡張子-develで識別できます。

**Tip**

### 4.3.1 パッケージの信頼性の検証

SUSE LINUX 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
```

コマンドrpm --checksig apache-1.3.12.rpm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。

### 4.3.2 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

一般に、RPMアーカイブのインストールは非常に簡単です。rpm -i ⟨*package*⟩.rpmを使用します。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。それ以外の人が行うと、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。

オプション-Uまたは--upgradeと-Fまたは--freshen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ます。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求めます。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。

### 4.3.3 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、まず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で簡単に見つけられます。rpmのパラメータ-P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。

### 4.3.4 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差分(これが“デルタ”)が含まれています。 古いRPMにデルタRPMを適用すると、完全に新しいRPMが作成されますが、古いRPMのコピーは必要なくなるので、デルタRPMはインストール済みのRPMでも機能できます。deltarpmパッケージは、パッチRPMと比べてそのサイズが小さくても、それは、インターネット上でアップデートパッケージを転送する必要が生じた場合には利点になります。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を使用する場合に比べて、CPUサイクルの消費が目立って多くなることです。YOUセッション中にYaSTでデルタRPMパッケージを使用するには、/etc/sysconfig/onlineupdate内でYOU_USE_DELTASを“yes”に設定します。

prepdeltarpm、writedeltarpm、およびapplydeltarpmバイナリはdeltarpmスイートの一部であり、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すると、new.delta.rpmという名前のデルタRPMを作成できます(この場合、old.rpmとnew.rpmが存在することが前提です)。

```
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
rm old.cpio new.cpio delta
```

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
```

あるいはファイルシステムにアクセスせずに、古いRPMから構築したい場合は、-rオプションを使用します。

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

詳細については、file:///usr/share/doc/packages/deltarpm/READMEを参照してください。

### 4.3.5 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。表 4.8.「最も重要なRPMクエリーのオプション」を参照してください。

*Table 4.8: 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 |
| --scripts | 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 4.2. 「rpm -q -i wget」に示された情報を表示します。

***Example 4.2:*** *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 4.3. 「パッケージを検索するスクリプト」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。

*Example 4.3: 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
```

コマンドrpm -q --changelog rpmは、特定のパッケージに関する詳細な情報(アップデート、設定、変更など)を表示します。この例は、パッケージrpmに関する情報を示します。ただし、RPMデータベース内の最後の5つの変更エントリだけがリストされます。(過去2年間に渡る)すべてのエントリは、パッケージ自体に含まれます。このクエリーは、CD1が/media/cdromにマウントされている場合にのみ、機能します。

```
rpm -qp --changelog /media/cdrom/suse/i586/rpm-4*.rpm
```

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。

*Table 4.9: 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1 GB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でパックされる)を毎日作成し、/var/adm/backup/rpmdbに格納します。コピーの数は、変数MAX_RPMDB_BACKUPS (デフォルト:5)によって制御されます。この変数は/etc/sysconfig/backupにあります。1つのバックアップのサイズは、1GBの/usrに対しておよそ3MBです。

### 4.3.6 ソースパッケージのインストールとコンパイル

SUSE LINUXの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。

**Tip**

ソースパッケージは、他のパッケージと同様に、YaSTでインストールできます。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。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。ソフトウェア自体は、コンパイルする必要があります。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。

**Tip**

(/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でソースパッケージをインストールすると、必要なコンポーネントがすべて/usr/src/packagesにインストールされます。SOURCES内のソースおよび調整ファイルとSPECS内の関連.specファイルです。

**Warning**

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。

**Warning**

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、以下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にソースを配置し、アンパックしてパッチを適用します。

**-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。注意: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。

### 4.3.7 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。それぞれの位置は、build --rpms ⟨*directory*⟩で指定されます。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 anewをビルドするには、rootユーザーで以下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。

### 4.3.8 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でHEADERを表示します。カーソルキーとEnterを使ってアーカイブ構造を表示します。F5で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項2.2.1. 「ソフトウェアのインストールと削除」を参照)。

# システムの修復

システムのインストールと設定を行う多くのYaSTモジュールに加えて、SUSE LINUXはインストール済みシステムの修復を行う機能も用意しています。この章では、システム修復のさまざまなタイプとステップについて説明します。SUSEレスキューシステムでは、パーティションにアクセスできます。熟練したシステム管理者は、レスキューシステムを使用して損傷したシステムを修復できます。

壊れたシステムが自身でブートできることは想定できず、また稼働中のシステムの修復は簡単でないので、新規インストールに臨むときのようにシステムをブートして修復します。章 1. YaSTによるインストールに概要を示したステップに従って、多くのインストールオプションを提供するダイアログを表示してから、［'インストールしたシステムの修復'］を選択します。

**Important**

**適切なインストールメディアの使用**

システム修復が正しく機能するには、システムをブートするために使用されるインストールメディアが、インストールしたシステムと正確に一致する必要があります。

**Important**

次の手順で、システム修復の実行方法を選択します。［自動的に修復する］、［カスタム修復］、［エキスパート設定用ツール］が使用可能であり、以降では各オプションについて説明します。

## 5.1 自動修復

この方法は、原因不明で損傷したシステムを修復するのに最適な方法です。この方法を選択すると、インストール済みシステムの広範囲の解析が開始されます。多数のテストと検証が実施されるため、解析には時間がかかります。このプロシージャの進捗状況は、画面下部にある2つの進捗バーで表示されます。上のバーは現在実行中のテストの進捗状況を示します。下のバーは解析プロセス全体の進捗状況を示します。上部のログウィンドウで、現在実行中のアクティビティとそのテスト結果を追跡することができます。図 5.1.「自動修復モード」を参照してください。以下のメインテストは、自動修復を実行すると毎回実行されます。言い換えれば、自動修復には、多数の個別サブテストが含まれています。

**全ハードディスクのパーティションテーブル**
検出された全ハードディスクのパーティションテーブルの妥当性と一貫性が検査されます。

**スワップパーティション** インストール済みのシステムのスワップ(swap)パーティションが検出され、テストされ、適用可能な場合は、スワップエリアを有効にする機会が提供されます。スワップエリアを有効にすると、システムの修復の処理速度が向上します。

**ファイルシステム** 検出されたすべてのファイルシステムがファイルシステム固有の検査の対象となります。

**/etc/fstabファイルのエントリ** このファイルのエントリの完全性と一貫性が検査されます。有効なパーティションは、すべてマウントされます。

**ブートローダの設定** インストールされているシステムのブートローダ設定(GRUBかLILO)の完全性と一貫性が検査されます。ブートデバイスとrootデバイスが調べられ、initrdモジュールの可用性が検査されます。

**パッケージデータベース** 最小構成のインストールの運用に必要なすべてのパッケージが存在しているか、検査されます。オプションで基本パッケージの解析も可能なので、基本パッケージの数が多いことが原因で、この検査には長時間かかります。


*Figure 5.1: 自 動 修 復 モ ー ド*

エラーを検出するたびに、プロシージャが一時停止し、エラーの詳細と、可能な解決策を提示するダイアログが表示されます。このような事例をすべて記述することは不可能です。画面に表示されるメッセージを注意深く読み、オプションのリストから必要な対応策を選択してください。また、提示された対応

策が適切なものかどうか疑わしい場合には、修復処理を拒否することもできます。この状況では、システムは変更されないままとなり、ユーザの対応を要求しないシステムの自動修復が実行されることはありません。

## 5.2 カスタム修復

前のセクションで説明した自動修復では、すべてのテストが実行されます。これは、システムの損傷範囲が不明な場合に役立ちます。しかし、影響を受けているシステムの部分が既に判明している場合には、実施するテストの範囲を狭めることができます。［'カスタム修復'］では、実行可能なテストのリストが、最初は、すべて実行対象として選択された状態で表示されます。全部のテスト範囲は、自動修復と合致します。損傷が存在していない個所が、既に判明している場合、対応するテストのチェックマークを消します。［'続行'］をクリックすると、より狭い範囲のテストプロシージャが開始され、実行時間が大幅に短縮されます。

すべてのテストグループを個別に実行できるわけではありません。fstabエントリの分析は、既存のスワップパーティションを含め、常にファイルシステムの検査と結び付いています。YaSTは、必要最小限の回数テストを実行して、自動的にこうした依存関係を満足させます。

## 5.3 エキスパート設定用ツール

SUSE LINUXに関する知識が豊富で、システムの修復に必要な対応策が既に明確な場合、［'エキスパート設定用ツール'］を選択して、必要な修復用ツールを直接適用します。

**新しいブートローダをインストールする**
YaSTの［ブートローダの設定］モジュールを起動します。詳細については項8.4.「YaSTを使用するブートローダの設定」を参照してください。

**パーティションツールを起動する** YaSTの［パーティションのエキスパート設定］ツールが起動します。詳細については項2.7.5.「パーティション」を参照してください。

**ファイルシステムを修復する** インストール済みのシステムのファイルシステムを検査します。まず、検出された全パーティションの中から1つを選択するダイアログが表示され、検査対象を選択することができます。

**失われたパーティションを復元する** 損傷したパーティションテーブルの再構築を試みることができます。まず、検出されたハードディスクのリストが表示され、対象を選択します。［'OK'］をクリックすると検証が開始されます。検証には、処理能力とハードディスクのサイズに応じて、時間がかかります。

**Important**

**パーティションテーブルの再構築**

パーティションテーブルの再構築は、難しい処理です。YaSTでは、ハードディスクのデータセクターを解析することにより、失われたパーティションの認識が試みられます。認識が成功すると、失われたパーティションが再構築したパーティションテーブルに追加されます。ただし、これは予想可能なすべての事例で成功するわけではありません。

**Important**

**システム設定をフロッピーに保存する**
このオプションは、重要なシステムファイルをフロッピーディスクに保存します。システムファイルの1つが損傷した場合には、作成しておいたフロッピディスクからリストアできます。

**インストールされたソフトウェアの確認**
パッケージデータベースの整合性と、最も重要なパッケージの可用性を検査します。このツールを使うと、損傷しているインストールパッケージを再インストールできます。

## 5.4 SUSEレスキューシステム

SUSE LINUXには、緊急の場合に、外部からLinuxパーティションにアクセスするためのレスキューシステムがあります。レスキューシステムは、CD、ネットワーク、またはSUSE FTPサーバからロードできます。レスキューシステムには、ヘルププログラムがいくつかあり、これらのプログラムによって、ハードディスクのアクセス不能、設定ファイルの設定間違いなどの重大な問題、あるいは他の同様の問題を解消できます。

レスキューシステムのもう1つのコンポーネントにPartedがあります。これは、パーティションのサイズ変更に使用されます。YaSTに内蔵されているリサイザーを使用したくない場合は、このプログラムをレスキューシス

テム内から起動できます。Partedに関する説明は、http://www.gnu.org/software/parted/にあります。

### 5.4.1 レスキューシステムの起動

インストールの場合と同様にシステムをブートします。ブートメニューから‘［レスキューシステム］’を選択します。これでレスキューシステムが解凍されて、新規ルートファイルシステムとしてRAMディスクにロードされたのち、マウントされ、起動されます。

### 5.4.2 レスキューシステムの使用

Alt-F1からAlt-F3では、レスキューシステムは3つのバーチャルコンソールを提供します。パスワードなしで、rootとしてログインできます。Alt-F10を押してシステムコンソールに入ると、カーネルメッセージおよびsyslogメッセージが表示されます。

シェルおよび他の多くの便利なユーティリティ(マウントプログラムなど)は、/binディレクトリにあります。sbinディレクトリには、ファイルシステムを検討し、修復するための重要なファイルおよびネットワークユーティリティが入っています。その中にはreiserfsckおよびe2fsckが含まれます。このディレクトリには、最も重要なバイナリも入っています。 たとえばステムメンテナンス用にはfdisk、mkfs、mkswap、mount、mount、init、およびshutdownがあり、ネットワークメンテナンス用にはifconfig、route、およびnetstatがあります。ディレクトリ/usr/binには、vi editor、grep、find、less、およびtelnetが入っています。

#### 通常システムのアクセス

レスキューシステムを使用してSUSE LINUXシステムをマウントするには、マウントポイント/mntを使用します。別のディレクトリを使用したり、作成したりもできます。次の例は、/etc/fstab詳細が例 5.1.「/etc/fstabの例」のようになっているシステムの場合のこの手順を示しています。

***Example 5.1:** /etc/fstabの 例*

```
/dev/sdb5    swap   swap    defaults   0   0
/dev/sdb3    /      ext2    defaults   1   1
/dev/sdb6    /usr   ext2    defaults   1   2
```

**Warning**

さまざなデバイスをマウントする場合には、次の項で概要を示したステップの順序に注意を払う必要があります。

**Warning**

システム全体にアクセスするには、次のコマンドを使用して、ステップバイステップでシステムを/mntディレクトリにマウントします。

```
mount /dev/sdb3 /mnt
mount /dev/sdb6 /mnt/usr
```

ここでシステム全体にアクセスし、/etc/fstab、/etc/passwd、/etc/inittab、などの設定ファイル中の間違いを訂正するなどします。ここで設定ファイルは、/etcではなく、/mnt/etcディレクトリに存在しています。fdiskプログラムを使用し、パーティションを再度設定するだけで、失われたパーティションを回復できますが、その場合、前もって/etc/fstabのプリントアウトおよびfdisk -lの出力を作成します。

### ファイルシステムの修復

ファイルシステムが壊れている状態は、レスキューシステムにとってやっかいな問題です。一般的に、稼動システムではファイルシステムを修復できません。重大な問題が見つかった場合、ルートファイルシステムをブートできなくなることさえあります。この場合、システムブートはカーネルパニックで終了します。この場合、外部からシステムを修復する唯一の方法は、レスキューシステムを使用することです。

SUSE LINUXレスキューシステムには、ユーティリティreiserfsck、e2fsck、およびdumpe2fs(診断用)が入っています。これらのユーティリティによって大半の問題が解消されるはずです。緊急時にmanページが使用不能となることがよくあります。このため、manページは本書の項B. 「Manual Page of reiserfsck」および項B. 「e2fsckのマニュアルページ」に含まれています。

スーパーブロックが無効であるために、ext2ファイルシステムのマウントが失敗した場合、e2fsckプログラムも失敗します。この状態が生じた場合、スーパーブロックも壊れることがあります。8192ブロックごと(8193、16385、など)スーパーブロックのコピーがあります。スーパーブロックが壊れた場合は、これらのコピーの1つを試してください。コマンドe2fsck -f -b 8193 /dev/damaged_partitionを入力すれば、これを行えます。-fオプションでは、ファイルシステムチェックが強制的に行われ、e2fsckのエラーが上書きされた結果、スーパーブロックが元の状態に復帰し、すべて良好な状態になります。

# Part 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

SUSE LINUXは、複数の64ビットプラットフォームで利用できます。ただし、用意されているすべてのアプリケーションがすでに64ビットプラットフォームに移植されているとは限りません。そのため、SUSE LINUXでは、64ビットシステム環境で32ビットアプリケーションを利用できます。この章では、このサポートを64ビットSUSE LINUX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の説明もあります。

64ビットプラットフォームAMD64、およびEM64Tに対応したSUSE LINUX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。

# 6.1 ランタイムサポート

**Important**

**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。

**Important**

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様にシステムの同じ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`、`/usr/lib`、および`/usr/X11R6/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`、`/usr/lib64`、および`/usr/X11R6/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`、`/usr/lib`、および`/usr/X11R6/lib`の下になっています。

データの内容がワードサイズに左右されないオブジェクトディレクトリのサブディレクトリは、移動されません。たとえば、X11フォントは、これまでどおり`/usr/X11R6/lib/X11/fonts`の下の通常の場所にあります。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。

## 6.2 ソフトウェア開発

biarch開発ツールチェーンでは、32ビットと64ビットのオブジェクトを生成できます。デフォルトは、64ビットオブジェクトのコンパイルです。特殊なフラグを使用すれば、32ビットオブジェクトを生成できます。GCCの場合、この特殊なフラグは`-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。

## 6.3 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。こうしたパッケージは、さらに、`rpmname-devel`パッケージからそれぞれのヘッダとライブラリ、また、`rpmname-devel-32bit`から対象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AMD64またはEM64Tのシステムを示しています。

1. 32ビットコンパイラを使用するために`autoconf`を設定します。

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

2. 32ビットオブジェクトを処理するようにリンカーに指示します。

   ```
   LD="ld -m elf64_i386"
   ```

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/X11R6/lib/
   ```

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 -m64"            \
LDFLAGS="-L/usr/lib64;"  \
   .configure       \
   --prefix=/usr  \
   --libdir=/usr/lib64 make make install
```

## 6.4 カーネル仕様

AMD64およびEM64T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される多くの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。このため、lspciやLVM管理プログラムなどの少数のアプリケーションは、正しく機能するように64ビットプログラムとして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。

**Tip**

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およびSUSEのプロバイダに問い合わせて、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。

**Tip**

# Linuxシステムのブートと設定

Linuxシステムのブート手順は複雑です。多くの異なるコンポーネントが関与し、それらが問題なく相互動作する必要があります。この章では、基本原理と関与するコンポーネントについて簡単に説明します。ランレベルの概念およびsysconfigによるSUSEのシステム設定についても、この章で説明します。

# 7.1 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(*CMOS*セットアップ)。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、章 8. ブートローダを参照してください。

3. **カーネルとinitrd**

   システムに制御を渡すために、ブートローダは、カーネルと初期RAMディスク(initrd)をメモリにロードします。Linuxカーネルには、実際のルートファイルシステムをマウントする前に、小さいファイルシステムをRAMディスクにロードしてプログラムを実行するオプションが用意されています。次に、カーネルはinitrdを展開し、それを一時的なルートファイルシステムとしてマウントします。initrdの中身は、linuxrcという実行可能ファイルを含む最低限のLinuxシステムです。この実行可能ファイルは、実際のルートファイルシステムがマウントされる前に実行されます。可能であれば、カーネルはinitrdで占有されたメモリを解放し、linuxrcが正常に終了した後にinitを起動します。initrdの詳細については、項7.1.1. 「initrd」を参照してください。

4. **linuxrc**

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、大容量ストレージコントローラに必要なファイルシステムとデバイスドライバ用のカーネル機能を提供します。実際のルートファイルシステムが正常にマウントされるとすぐに、linuxrcは停止し、カーネルがinitプログラムを起動します。linuxrcの詳細については、項7.1.2. 「linuxrc」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、項7.2. 「initプログラム」で説明しています。

### 7.1.1 initrd

initrdは、カーネルがRAMディスクにロードできる小さい(通常は圧縮された)ファイルシステムであり、一時的なルートファイルシステムとしてマウントできます。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、十分な容量のメモリ以外に具体的なハードウェア要件はありません。initrdは、エラーなく終了するlinuxrcという名前の実行可能ファイルを常に提供する必要があります。

実際の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も含まれる場合があります(を参照)。また、カーネルには、initrdのファイルシステムを読み取るために必要なコードも含まれていなければなりません。ルートファイルシステムに必要なモジュールは、linuxrcでロードすることができます。

スクリプト`mkinitrd`で`initrd`を作成します。SUSE LINUXでは、ロードするモジュールは、`/etc/sysconfig/kernel`の変数`INITRD_MODULES`で指定されます。インストール後に、この変数は自動的に正しい値に設定されます(インストールされたlinuxrcで、ロードされたモジュールの情報が保存されています)。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。複数のSCSIドライバが使用されている場合は、この順序が特に重要です。その理由は、順序が正しくなければハードディスクの名前が変わってしまうためです。厳密に言えば、ルートファイルシステムにアクセスするために必要なドライバをロードするだけで十分でしょう。しかし、後でロードすると

問題が生じることがあるため、インストールに必要なすべてのSCSIドライバがinitrdでロードされます。

**Important**

**initrdの更新**

ブートローダは、カーネルと同じようにinitrdをロードします。ブート時にGRUBが権限ファイルのディレクトリを検索するため、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。

**Important**

### 7.1.2 linuxrc

linuxrcの主要目的は、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。実際のシステム設定に基いて、linuxrc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。

**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維持するようにシステムを設定した場合、linuxrcは後でルートファイルシステムにアクセスできるようにLVMまたはRAIDをセットアップします。RAIDについては、項3.8. 「ソフトウェアRAID設定」を参照してください。LVMについては、項3.7. 「LVMの設定」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、そのメディアの実際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** 項7.1.1.「initrd」で説明したように、ブートプロセスは、ほとんどのハードウェア設定で使用できる最小限のドライバセットで開始されます。linuxrc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。こうした値は、それ以降のブートプロセスでカスタムinitrdを使用できるように、後で`/etc/sysconfig/kernel`の`INITRD_MODULES`に書き込まれます。インストールプロセスの間に、linuxrcはこの変数で指定されたモジュールセットをロードします。

**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の起動** 最終的に、linuxrcがYaSTを起動すると、パッケージのインストールとシステム設定が開始されます。

### 7.1.3 詳細情報

詳細については、`/usr/src/linux/Documentation/ramdisk.txt`、`/usr/src/linux/Documentation/initrd.txt`、およびマニュアルページのinitrd(4)とmkinitrd(8)を参照してください。

## 7.2 initプログラム

プログラムinitは、プロセス番号1のプロセスであり、必要な方法でシステムの初期化を実行します。他のすべてのプロセスは、initの子プロセスまたはそのプロセスのいずれかの子プロセスです。initは特別な役割を果たします。initはカーネルによって直接起動され、通常はプロセスを強制終了するシグナル9が使えないようにします。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は、/etc/inittabファイルによって一元的に設定されます。ここで、ランレベルが定義されます(項7.3.「ランレベル」を参照)。また各レベルで利用可能なサービスとデーモンが指定されます。/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。わかりやすくするために、これらのスクリプトはすべて、ディレクトリ/etc/init.dにあります。

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。

## 7.3 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動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。表 7.1.「ランレベルの種類」を参照してください。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で)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。

システムの稼動中にランレベルを変更するには、init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。システム管理者だけが、この操作を行うことができます。init 1(またはshutdown now)では、システムがシングルユーザモードに変更されます。このモードは、システムの保守と管理に使用されます。作業が完了したら、管理者はinit 3を入力すれば通常のランレベルに戻すことができます。このランレベルでは、必須のすべてのプログラムが起動され、通常のユーザがXなしでシステムにログインしてシステムを操作できます。GNOME、KDE、または他のウィンドウマネージャと同様にグラフィック環境を使用可能にするには、代わりにinit 5を使用します。init 0またはshutdown -h nowでは、システムが停止します。init 6またはshutdown -r nowでは、シャットダウンの後に再起動されます。

**Important**

**NFSマウントされた/usrパーティションによるランレベル2**

システムでNFSを介して/usr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。/usrディレクトリには、システムが正しく機能するために不可欠な重要なプログラムが入っています。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)で利用できるようになっていないので、システムが多くの側面で重大な制約を課されます。

**Important**

*Table 7.1: ランレベルの種類*

| ランレベル | 説明 |
|---|---|
| 0 | システム停止 |
| S | シングルユーザモード(ブートプロンプトからUSキーボードマッピングで入力された場合) |
| 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 | 未使用 |
| 5 | ネットワークとXディスプレイマネージャのKDM(デフォルト)、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 |

ランレベル5は、すべてのSUSE LINUX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。ユーザはグラフィックインタフェースで、直接ログインを求められます。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、章 11. X Windowシステムで説明するようにX Window Systemを正しく設定している必要があります。その後、`init 5`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する場合は、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。

**Warning**

**/etc/inittabの変更**

/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。したがって、/etc/inittabの編集はきわめて慎重に行う必要があります。また、必ず変更前のバージョンをバックアップしてください。破損したファイルを修復するには、ブートプロンプトのカーネル名の後に、init=/bin/shと入力して、直接シェルからブートしてみます。その後、コマンドmount -o remount,rw /でルートファイルシステムを書き込み可能にし、cpを使用して/etc/inittabをバックアップバージョンで置き換えます。ファイルシステムエラーを防止するには、再起動する前にルートファイルシステムを読み取り専用に変更します(mount -o remount,ro /)。

**Warning**

## 7.4 ランレベルの変更

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init 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はその設定ファイル(/etc/inittab)を調べ、新しいランレベルをパラメータとして使用して/etc/init.d/rcを起動する必要があると判断します。
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だけをすべて呼び出します。この例では、元のランレベルが3なので、/etc/init.d/rc3.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。依存性を考慮する必要がありますが、Kの後の数字によって、一定の起動順序を指定します。
4. 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。順序に関して、それらを起動したときに適用されたのと同じ手順が、ここでも適用されます。

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。

## 7.5 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。ランレベルを変更するスクリプトも同じディレクトリにありますが、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 7.2.「initスクリプトのオプション」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。

*Table 7.2: initスクリプトのオプション*

| オプション | 説明 |
|---|---|
| `start` | サービスを起動します。実行中のサービスを起動すると、何事もなく正常に終了します。 |
| `stop` | サービスを停止します。 |
| `restart` | サービスが実行中の場合は、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。 |
| `status` | 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スクリプト`/usr/lib/lsb/install_initd`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。詳細については、マニュアルページでinsserv(8)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`と`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。これらのスクリプトによってトリガされたアクション(たとえば、複数のサブスクリプトを実行するなど)が完了すると停止します。blogdは、すべての画面出力をログファイル`/var/log/boot.msg`に書き込みますが、これは`/var`が読み書き権を設定してマウントされた場合のみです。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の詳細については、マニュアルページでblogd(8)を参照してください。

スクリプトboot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、スクリプトboot.localが実行されます。

**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。

**boot.setup** このスクリプトは、シングルユーザモードから他のランレベルへの移行時に実行され、キーボードレイアウトや仮想コンソールの初期化に関する基本的な設定をします。

**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リブートするかは、haltの呼び出され方に依存します。

**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。

### 7.5.1 initスクリプトの追加

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、および構成方法については、LSBの仕様と、マニュアルページでinit(8)、init.d(7)、およびinsserv(8)を参照してください。また、マニュアルページでstartproc(8)とkillproc(8)も参照してください。

**Warning**

**カスタムのinitスクリプトの作成**

initスクリプトに問題があると、コンピュータがフリーズします。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スクリプトについては、項7.3. 「ランレベル」の情報が役立ちます。

**Warning**

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。例 7.1. 「最低限のINIT INFOブロック」を参照してください。

*Example 7.1: 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:とRequired-Stop:の2行に、サービス自体が起動または停止する前に、それぞれ起動または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:の下に、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、項7.6. 「」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、まず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、次回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。

## 7.6

このYaSTモジュールを起動すると、利用可能なすべてのサービスと各サービスの現在のステータス(有効か無効か)を示す概要が表示されます。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、まず［'エキスパートモード'］を選択します。このモードでは、ダイアログボックスが現在のデフォルトのランレベルまたは「initdefault」(システムブート時のデフォルトランレベル)が上部に表示されます。通常、SUSE LINUX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 7.1. 「ランレベルの種類」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、初期設定は未定義です。

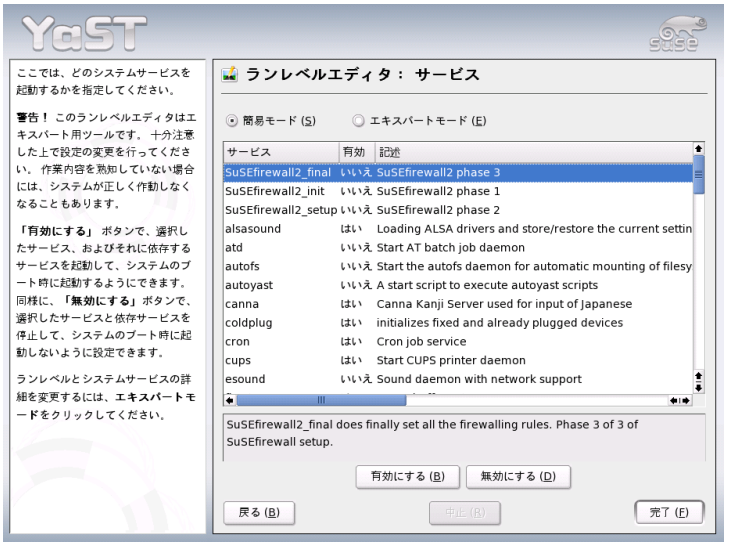


*Figure 7.1:*

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。

［‘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。

**Warning**

**ランレベル設定の変更**

ランレベルの設定が誤っていると、システムが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。

**Warning**

## 7.7 SuSEconfigと/etc/sysconfig

SUSE LINUXの主な設定は、/etc/sysconfigディレクトリ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。/etc/sysconfigディレクトリ内の設定に従って生成される他の多くのシステム設定ファイルも存在します。この作業は、SuSEconfigによって実行されます。たとえば、ネットワーク設定を変更すると、ネットワーク設定に関連するファイルの1つである/etc/host.confも、SuSEconfigによって変更されます。

これらのファイルを手動で変更した場合は、後でSuSEconfigを実行して、すべての必要な変更がすべての関連する場所で行われていることを確認します。YaSTのsysconfigエディタで設定を変更すると、すべての変更が自動的に適用されます。YaSTが自動的にSuSEconfigを起動し、必要に応じて設定ファイルを更新するからです。

この概念により、ユーザはシステムを再起動せずに基本的な設定を変更できます。変更の中には複雑なものもあるので、プログラムによっては変更を有効にするためにシステムの再起動が必要な場合もあります。たとえば、ネットワーク設定への変更に、関係するネットワークプログラムの再起動が必要なことがあります。これには、rcnetwork stopコマンドとrcnetwork startコマンドを使用します。

システム設定の変更は、次の手順で実行することをお勧めします。

1. init 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。
2. 必要に応じて設定ファイルを変更します。これには、任意のエディタを使用するか、YaSTのsysconfigエディタを使用します(項7.8. 「YaST sysconfigエディタ」を参照)。

**Warning**

**手動によるシステム設定の変更**

/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、空白を含む値は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。

**Warning**

3. 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。YaSTで設定ファイルを変更した場合は、この手順が自動的に行われます。

4. 元のランレベルに戻す場合は、init 3のようなコマンドなどを使用します(3の部分を元のランレベルに置き換えます)。

この手順は、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特に必要です。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。

**Tip**

**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。

**Tip**

# 7.8 YaST sysconfigエディタ

ほとんどの重要なSUSE LINUXの設定が格納されているファイルは、/etc/sysconfigディレクトリにあります。sysconfigエディタでは、オプションが読みやすく表示されます。値を変更したり、それをディレクトリの設定ファイルに追加したりできます。しかし、これらのファイルはパッケージのインストールやサービスの設定の際に自動的に調整されるので、一般に設定を手動で編集する必要はありません。

**Warning**

**/etc/sysconfig/*ファイルの変更**

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。

**Warning**

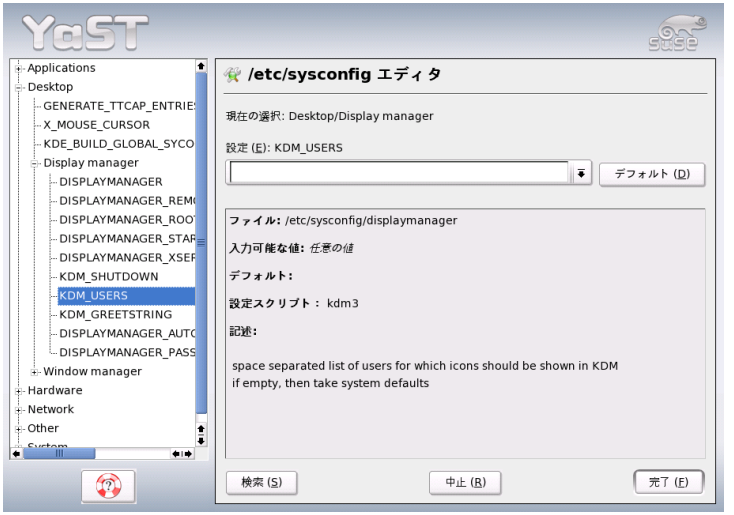


***Figure 7.2:*** *sysconfigエディタを使用したシステム設定*

YaSTsysconfig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。

8

ブートローダ

# ブートローダ

この章では、SUSE LINUXで現在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。

この章は、ブート管理とブートローダGRUBの設定に重点を置いています。ブート手順は、総じて章 7. Linuxシステムのブートと設定で説明しています。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)の間のインタフェースになります。ブートローダの設定では、起動するオペレーティングシステムとそのオプションが決定されます。

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。通常、この領域にはブートローダプログラム(この場合はGRUB)が保持されます。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(項1.5.4. 「パーティションのタイプ」を参照)。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。異なる値を含むMBRは、BIOSおよびすべてのPCのオペレーティングシステムにより無効と見なされます。

**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、ファイルシステムの設定直後は空になっています。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。

# 8.1 ブート管理

最も単純なケース、つまりコンピュータにオペレーティングシステムが1つしかインストールされていない場合には、ブート管理はこれまでに説明したよう

に行われます。コンピュータに複数のオペレーティングシステムがインストールされている場合は、次の選択肢があります。

**2番目以降のシステムを外部メディアからブートする**
オペレーティングシステムのいずれかをハードディスクからブートします。他のオペレーティングシステムは、外部メディア(フロッピーディスク、CD、USBストレージメディア)にインストールされているブートマネージャを使用してブートします。GRUBは他のすべてのオペレーティングシステムをブートできるため、外部ブートローダを使用する必要はありません。

**ブートマネージャをMBRにインストールする**
ブートマネージャを使用すると、1台のコンピュータに同時に複数のシステムをインストールし、それらを切り替えて使用できます。ユーザは、ブートプロセス中にブートするシステムを選択できます。別のシステムに切り替えるには、コンピュータを再起動する必要があります。この操作が可能なのは、選択したブートマネージャにインストール済みオペレーティングシステムとの互換性がある場合だけです。SUSE LINUXで使用されるブートマネージャGRUBには、一般的なすべてのオペレーティングシステムをブートする機能があります。デフォルトでは、選択したブートマネージャはSUSE LINUXによりMBRにインストールされます。

## 8.2 ブートローダの選択

SUSE LINUXでは、デフォルトでブートローダGRUBが使用されます。ただし、特殊なハードウェアやソフトウェアなど、状況によっては、LILOの方が適している場合があります。LILOで使用されていた古いバージョンのSUSE LINUXをアップデートすると、LILOがインストールされます。新規インストールの場合は、ルートパーティションが次のシステムにインストールされていない限り、GRUBがインストールされます。

- CPUに依存するRAIDコントローラ(たとえば、Promiseまたはの数多くのコントローラ)
- ソフトウェアRAID
- LVM

LILOのインストールおよび設定については、Support Database (サポートデータベース)でキーワード*LILO*を使用すると検索できます。

## 8.3 GRUBによるブート

GRUB (Grand Unified Bootloader)は、2つのステージで構成されています。stage1は512バイトから成り、MBR、またはハードディスクパーティションやフロッピーディスクのブートセクタに書き込まれます。その後、stage2が読み込まれます。このステージには、実際のプログラムコードが含まれています。最初のステージのタスクは、ブートローダの第2ステージを読み込むことだけです。

stage2には、ファイルシステムにアクセスする機能があります。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JFS、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再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。この情報がなければ、システム制御をオペレーティングシステムに渡すことができません。

**`/boot/grub/device.map`** このファイルでは、デバイス名がGRUBとBIOSの表記法からLinuxデバイス名に変換されます。

**`/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パラメータと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って編集した場合は、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(項8.3.1.「ブート手順実行中のメニューエントリの編集」を参照)。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シェル*は、インストール済みシステムでGRUBをエミュレートします。このシェルを使用すると、GRUBをインストールしたり、適用前に新規設定をテストできます。項8.3.4.「GRUBシェル」を参照してください。

### 8.3.1 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、このファイルにはブート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、GRUBは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せん。項8.4.「YaSTを使用するブートローダの設定」で説明するように、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)を指定します。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。次に例を示します。

```
chainloader (hd0,3)+1
```

GRUBでのデバイス名については、項8.3.1.「ハードディスクとパーティションに関する命名規則」を参照してください。先ほどの例では、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、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の読み込みアドレスは、読み込まれるカーネルイメージに書き込まれているので、initrdコマンドは、kernelコマンドの直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、GRUBデバイス、またはGRUBデバイス上の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いない他のファイルのパスに使用されます。インストール時に生成されるmenu.lstファイル内では、このコマンドは使用されていません。単純に手動で編集する際に使用するものです。

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、項8.3.1.「メニューファイルの例」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。GRUBでは、パーティション番号が0から始まります。そのため、(hd0,0)は最初のハードディスクの最初のパーティションとなります。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/hda1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)    最初のハードディスクの最初の基本パーティション
(hd0,1)    2番目の基本パーティション
(hd0,2)    3番目の基本パーティション
(hd0,3)    4番目の基本パーティション
(通常は拡張パーティション)
(hd0,4)    最初の論理パーティション
(hd0,5)    2番目の論理パーティション
```

GRUBは、IDE、SCSI、RAIDの各デバイスを区別しません。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。

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、項8.3.2.「device.mapファイル」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/hda5`、ルートパーティションが`/dev/hda7`、およびWindowsのインストールファイルが`/dev/hda1`にあります。

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
```

```
title linux
  kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd

title windows
  chainloader(hd0,0)+1

title floppy
  chainloader(fd0)+1

title failsafe
  kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma \
  apm=off acpi=off vga=normal nosmp maxcpus=0 3
  initrd (hd0,4)/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/hda5`にあります。

**color white/blue black/light-gray** 配色は、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。

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/hda7`)。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回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。項8.3.1.「ブート手順実行中のメニューエントリの編集」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

GRUBの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、現在のブート手順だけに適用され、永続的に採用されることはありません。

**Important**

**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。

**Important**

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
  kernel (hd0,0)/vmlinuz root=/dev/hda3 additional parameter
  initrd (hd0,0)/initrd
```

GRUBは、次回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。

### ワイルドカードを使用したブートカーネルの選択

特にカスタムカーネルを開発または使用する場合は、menu.lst内のエントリを変更するか、またはコマンドラインを編集して現在のカーネルとinitrdファイル名を反映する必要があります。この手順を単純化するには、GRUBのカーネルリストを動的に更新するために*ワイルドカード*を使用します。その結果、特定のパターンと一致するすべてのカーネルイメージが、ブート可能なイメージのリストに自動的に追加されます。この機能についてはサポートがないので注意してください。

ワイルドカードオプションを有効にするには、menu.lstにさらにメニューエントリを入力します。実用になるように、すべてのカーネルイメージとinitrdイメージには、カーネルをその関連するinitrdに対応させる共通の基本名と識別子が必要です。次のセットアップを考えてみます。

```
initrd-default
initrd-test
vmlinuz-default
vmlinuz-test
```

この場合には、1つのGRUB設定で両方のブートイメージを追加できます。メニューエントリlinux-defaultとlinux-testを取得するには、menu.lstに次のエントリが必要です。

```
title linux-*
  wildcard (hd0,4)/vmlinuz-*
  kernel (hd0,4)/vmlinuz-* root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-*
```

この例では、GRUBはワイルドカードに対応するエントリのパーティション(hd0,4)を検索します。こうしたエントリを使用してGRUBメニューの新しいエントリが生成されます。先ほどの例では、GRUBは、次のエントリがmenu.lstにあるかのように動作します。

```
title linux-default
  wildcard (hd0,4)/vmlinuz-default
  kernel (hd0,4)/vmlinuz-default root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-default
title linux-test
  wildcard (hd0,4)/vmlinuz-test
  kernel (hd0,4)/vmlinuz-test root=/dev/hda7 vga=791
  initrd (hd0,4)/initrd-test
```

このような設定で、ファイル名が一貫して使用されていない場合や、展開されたファイルのいずれか(initrdイメージなど)が失われている場合には、問題が発生するおそれがあります。

### 8.3.2 device.mapファイル

device.mapファイルは、GRUB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別を試みる必要があります。これは、GRUBにはブートシーケンスのBIOS情報へのアクセス権がないからです。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/hda
(hd1)  /dev/sda
```

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は、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をチェックします。さらに、必要に応じてGRUBシェルを使用し、ファイル内のシーケンスを一時的に変更します(項8.3.4. 「GRUBシェル」を参照)。Linuxシステムのブート後に、YaSTブートローダ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
```

### 8.3.3 /etc/grub.confファイル

menu.lstおよびdevice.mapのほかに重要な第3のGRUB設定ファイルは、/etc/grub.confです。このファイルには、grubコマンド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パラメータとオプションが含まれています。

```
root (hd0,4)
  install /grub/stage1 d (hd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
  quit
```

各エントリの意味:

**root (hd0,4)** このコマンドは、GRUBに対して後続のコマンドを1台目のハードディスクの最初の論理パーティション(ブートファイルの位置)に適用するように指示します。

**installパラメータ** grubコマンドは、installパラメータを指定して実行する必要があります。ブートローダのstage1は、1台目のハードディスクのMBR内にインストールする必要があります(/grub/stage1 d (hd0))。stage2は、メモリアドレス0x8000に読み込む必要があります(/grub/stage2 0x8000)。最後のエントリ((hd0,4)/grub/menu.lst)は、メニューファイルを探す場所をGRUBに伝えます。

### 8.3.4 GRUBシェル

GRUBには実質的に2つのバージョンがあります。ブートローダと、/usr/sbin/grubにある通常のLinuxプログラムです。このプログラムを*GRUB*シェルと呼びます。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。

ただし、setupコマンドとinstallコマンドは、Linux起動前のブート手順でも使用できます。これにより、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。単に、menu.lstの場合と同様の構文を使用して、実験的な設定ファイルを入力します。次に、既存の設定ファイルは変更せずに、このエントリの機能をテストします。たとえば、新規カーネルをテストするには、kernelコマンドと新規カーネルへのパスを入力します。ブートプロシージャが失敗した場合、次回のブート時にはオリジナルのmenu.lstを引き続き使用できます。同様に、訂正後のパラメータを入力することで、menu.lstファイルの誤りに関係なくコマンドラインインタフェースを使用してシステムをブートすることもできます。稼働中のシステムでは、menu.lstに正しいパラメータを入力して、システムを永続的にブート可能にすることができます。

GRUBデバイスからLinuxデバイスへのマッピングが関係するのは、GRUBシェルを(項8.3.2. 「device.mapファイル」の説明にしたがってgrubを入力して)Linuxプログラムとして実行する場合だけです。この目的で、このプログラムはdevice.mapファイルを読み取ります。詳細については、項8.3.2. 「device.mapファイル」を参照してください。

### 8.3.5 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。

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と入力します。
2. GRUBシェル内でパスワードを暗号化します。

```
grub> md5crypt
Password:****
Encrypted:$1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

3. 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 $1$lS2dv/$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。

4. 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。次に例を示します。

```
title linux
kernel (hd0,4)/vmlinuz root=/dev/hda7 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。

**Important**

**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。

**Important**

## 8.4 YaSTを使用するブートローダの設定

SUSE LINUX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ーで、'システム' → 'ブートローダの設定'の順に選択します。システムの現在のブートローダ設定が表示され、必要な変更が可能になります。図 8.1. 「YaSTを使用するブートローダの設定」を参照してください。

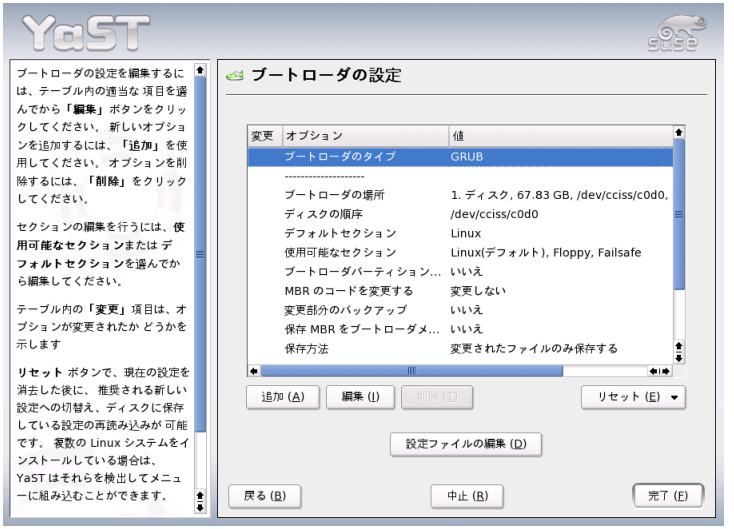


*Figure 8.1: YaSTを使用するブートローダの設定*

## 8.4.1 メインウィンドウ

設定データをリストとして表示するテーブルは、3つの列で構成されます。［'Changed(変更済み)'］(左側)では、中央の列にリストされる変更されたオプションにフラグマークが付きます。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、マウスをクリックして選択し、［'編集'］をクリックします。既存のオプションをまったく使用しない場合は、選択して［'削除'］をクリックします。 ［'リセット'］には次のオプションがあります。

**新しい設定を提案する** 推奨される新しい設定を生成します。他のパーティションで検出された、古いLinuxバージョンまたは他のオペレーティングシステムがブートメニューに追加され、Linuxまたは古いブートローダからのブートを可能にします。古いブートローダからブートする場合、2番目のブートメニューが表示されます。

**新規作成** すべての設定を新規に作成できます。推奨される設定は生成されません。

**ディスクから設定を再読み込みする** 既に設定変更を実行し、その結果に満足していない場合、このオプションを使用して現在の設定を再読み込みします。

**既存のGRUBメニューを使用して提案およびマージする**
他のオペレーティングシステムおよび古いLinuxバージョンが他のパーティションにインストールされている場合、エントリから新しいSUSE LINUXに対してメニューが生成されます。そこには他のシステムに対するエントリおよび古いブートローダメニューのすべてのエントリが追加されます。この処理には少し時間がかかることがあります。これは、LILOが使用されている場合利用できません。

**ハードディスクのMBRを戻す** ハードディスクに保存されている、バックアップされたMBRが書き戻されます。

［'Edit Configuration Files(設定ファイルの編集)'］を使用して、エディタを使用して関連する設定ファイルを編集します。ファイルを編集するには、選択したフィールドを使用してロードします。［'OK'］をクリックして、変更内容を保存します。ブートローダ設定を終了するには、［'キャンセル'］をクリックします。［'戻る'］をクリックして、メインウィンドウに戻ります。

### 8.4.2 ブートローダの設定オプション

YaSTを使用する設定はファイルを直接編集するよりはるかに容易です。オプションを選択し、［'編集'］をクリックして必要に応じた設定を変更するダイアログを開きます。［'OK'］をクリックして変更を確定し、メインメニューに戻り、他のオプションを編集します。利用可能なオプションは使用しているブートローダによって異なります。以下にブートローダGRUBのいくつかのオプションを紹介します。

**ブートローダのタイプ** このオプションを使用して、GRUBとLILOを切り替えます。切り替える方法を指定し、この変更が実行される他のダイアログに続きます。たとえば、現在のGRUB設定を類似したLILO設定に変換します。ただし、同等のオプションが利用できない場合、いくつかの変更点は失われます。新規に新しい設定を作成し、推奨される設定を編集することもできます。

稼動中のシステムでブートローダの設定を開始する場合、ハードディスクから設定をロードできます。最後の手段として、元のブートローダに戻る設定をロードすることができます。ただし、この手段はブートローダモジュールを終了しない場合に限り有効です。

**ブートローダの場所** このダイアログを使用してブートローダをインストールする場所を指定します。マスターブートレコード(MBR)内、ブートパーティションのブートセクタ(使用可能な場合)内、ルートパーティションのブートセクタ内、またはフロッピーディスク内に指定できます。［'その他'］を使用して異なる場所を指定します。

**ディスク順序** コンピュータに1つ以上のハードディスクがある場合、マシンのBIOSセットアップで定義したディスクのブートシーケンスを指定します。

**デフォルトのセクション** このオプションを使用すると、デフォルトでブートするカーネルまたはオペレーティングシステムを設定できます。選択されたシステムがタイムアウト時間を経過した後にブートします。このメニューでは、［'編集'］ボタンを使用すると、すべてのブートメニューエントリのリストを入手できます。リストから1つのエントリを選択し、［'デフォルトに設定'］をクリックします。この段階で、［'編集'］を使用することにより、すべてのエントリを編集することもできます。

**利用可能なセクション** ブートメニューの既存のエントリが、メインウィンドウ内のこのオプション下にリストされます。このオプションを選択し、［'編集'］をクリックすると、［'デフォルトエントリ'］ダイアログと同じ名前のダイアログが開きます。

**ブートローダパーティションをアクティブにする**
このオプションを使用して、ブートローダを保持するブートセクタがあるパーティションをアクティブにできます。これはブートローダのヘルパーファイルが保存されているディレクトリ(/bootまたはルートディレクトリ/)があるパーティションには依存しません。

**MBR内で置き換えられるコード** MBR内にGRUBがインストールされている場合、または新しいハードディスク上にシステムをインストール中にMBR内にGRUBをインストールしない場合は、このオプションを使用して汎用ブートコードを戻します。

**バックアップファイルおよびハードディスクの一部**
変更されたハードディスク領域をバックアップします。

**保存したMBRをブートローダメニューに追加する**
保存したMBRをブートローダメニューに追加します。

［'タイムアウト'］を使用して、デフォルトのシステムがブートする前に、ブートローダが何秒キーボードによる入力を待つかを指定します。他のオ

プションの多くは、［'追加'］を使用して指定できます。利用可能なオプションに関する詳細は、対応するマニュアルページ(grub(8)または lilo(8))、またはhttp://www.gnu.org/software/grub/manual/にあるオンラインドキュメントを参照してください。

## 8.5 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GRUBを上書きして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')。最初のダイアログで、'リセット' → 'ハードディスクのMBRを戻す'を選択し、'完了'を選択してダイアログを終了します。MBR内で、GRUBがオリジナルMBRのデータで上書きされます。

## 8.6 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CD-ROM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。

cd /tmpおよびmkdir isoなどを使用して、ISOイメージの作成場所となるディレクトリを作成します。また、mkdir -p iso/boot/grubでGRUBのサブディレクトリも作成します。ファイルstage2_eltoritoをディレクトリgrubにコピーします。

```
cp /usr/lib/grub/stage2_eltorito iso/boot/grub
```

また、カーネル(/boot/vmlinuz)、initrd (/boot/initrd)、およびファイル/boot/messageをiso/boot/にコピーします。

```
cp /boot/vmlinuz iso/boot/
cp /boot/initrd iso/boot/
cp /boot/message iso/boot/
```

これらをGRUBで使用できるように、ファイルmenu.lstをiso/boot/grubにコピーし、CD-ROMデバイスを指すようにパスエントリを調整します。そのためには、パス名に(hd*)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。

```
gfxmenu (cd)/boot/message
timeout 8
default 0

title Linux
  kernel (cd)/boot/vmlinuz root=/dev/hda5   vga=794 resume=/dev/hda1
splash=verbose showopts
  initrd (cd)/boot/initrd
```

最後に、次のコマンドでISOイメージを作成します。

```
mkisofs -R -b boot\grub\stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -o grub.iso iso
```

次に、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。

## 8.7 SUSEのグラフィカル画面

SUSE LINUX7.2以降は、オプション“vga=vga=<value>”がカーネルパラメータとして有効になっている場合、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。必要に応じて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。詳細については、章 8. ブートローダを参照してください。しかし、以前のバージョンではデフォルトになっていたテキストモードを使用したい場合は、vga=normalを設定します。

**SUSEの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 (フレームバッファサポート)'］でオプション［'Use splash screen instead of boot logo (ブートロゴの代わりにスプラッシュ画面を使用する)'］を無効にします。

**Tip**

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。これをカスタムカーネルで行った場合、SUSEはサポートを何も提供することができません。

**Tip**

## 8.8 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://portal.suse.de/sdb/en/index.htmlのSupport Database (サポートデータベース)に記事が提供されています。特定の問題がこのリストに含まれていない場合は、https://portal.suse.com/PM/page/search.pmにアクセスしてSupport Databaseの検索ダイアログを使用し、*GRUB*、*boot*、*boot loader*などのキーワードで検索してください。

**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とJFS** GRUBとJFS を組み合わせることは技術的には可能ですが、問題があります。この場合は、別のブートパーティション(/boot)を作成し、Ext2でフォーマットします。このパーティションにGRUBをインストールしてください。

**GRUBでGRUB Geom Errorがレポートされる**

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。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。このような場合は、LILOを使用するか、BIOSを更新します。LILOのインストール、設定、および保守の詳細については、Support Database (サポートデータベース)でキーワードLILOを使用すると検索できます。

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*は正常に検出され読み込まれますが、*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。

**IDEハードディスクとSCSIハードディスクを搭載したシステムがブートしない**

インストール中に、YaSTがハードディスクのブートシーケンスを誤って判別する(およびユーザがそれを訂正していない)場合があります。たとえば、GRUBが`/dev/hda`を`hd0`、`/dev/sda`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に存在する論理に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。

## 8.9 詳細情報

GRUBの詳細情報は、http://www.gnu.org/software/grub/で入手できます。使用中のコンピュータにtexinfoがインストールされている場合、info grubと入力して、シェルの中でGRUBの情報ページを参照できます。http://portal.suse.de/sdb/en/index.htmlにあるSupprt Database (サポートデータベース)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。

# Linuxカーネル

カーネルは、すべてのLinuxシステムのハードウェアを管理し、さまざまな処理に使用できるようにします。この章では、カーネルハッカーになるための情報ではなく、カーネル更新を実行する方法と、カスタムカーネルをコンパイルしてインストールする方法について説明します。この章で説明する手順に従うと、以前のカーネルは引き続き動作し、必要に応じてブートできます。

/bootディレクトリにインストールされているカーネルは、多様なハードウェアに使用できるように設定されています。通常、実験的な機能やドライバをテストする場合を除き、カスタムカーネルをコンパイルする必要はありません。

インストール済みカーネルの動作は、通常はカーネルパラメータを使用して変更できます。たとえば、desktopパラメータではスケジューラ用に短いタイムスライスを設定し、システムを実質的に高速化できます。kernel-sourceパッケージがインストールされていることを想定し、/usr/src/linux/Documentationディレクトリにあるカーネルドキュメントで情報を参照できます。

プロセスを自動化するために、カーネルとともに複数のMakefileが提供されます。ハードウェア設定や他のカーネル機能を選択してください。適切な選択を行うにはコンピュータシステムを熟知している必要があるため、初めて試行する際には現在使用中の設定ファイルを変更することをお勧めします。

## 9.1 カーネル更新

オフィシャルのSUSEアップデートカーネルをインストールするには、YaSTのオンラインアップデート機能を使用します。カーネルをアップデートした後はシステムをリブートする必要があります。アップデート前から稼動しているカーネルが、必要な機能を提供するために適切なモジュールを検出しない場合があるためです。YaSTオンラインアップデートについての詳細は項2.2.3.「YaSTオンラインアップデート」を参照してください。

アップデートの実行中は、必要なアクションをすべて説明するポップアップが表示されます。これらのコマンドに従い、整合的にシステムを保守します。

## 9.2 カーネルソース

カーネルを作成するには、kernel-sourceパッケージをインストールする必要があります。Cコンパイラ(gccパッケージ)、GNU binutils (binutilsパッケージ)、およびCコンパイラ用のインクルードファイル(glibc-develパッケージ)などの追加パッケージが、YaSTによりインストール対象として自動的に選択され、インストールされます。

インストール後、カーネルソースは/usr/src/linux-<kernel-version>に配置されます。別々のカーネルで実行する場合は、それぞれを異なるサブディレクトリ内で解凍し、現行のカーネルソースへのシンボリッ

クリンクを作成します。ソースが/usr/src/linuxに存在することに依存するソフトウェアパッケージがあるため、このディレクトリは現行のカーネルソースへのシンボリックリンクとして保持してください。YaSTでは、これが自動的に実行されます。

## 9.3 カーネル設定

現行カーネルの設定は、/proc/config.gzファイルに格納されています。この設定を変更するには、rootで/usr/src/linuxディレクトリにアクセスして次のコマンドを実行します。

```
zcat /proc/config.gz > .config
make oldconfig
```

make oldconfigコマンドは、/usr/src/linux/.configファイルを現行カーネル設定のテンプレートとして使用します。現行のカーネルソース用の新規オプションがクエリされます。.configファイルが存在しない場合は、カーネルソースに含まれているデフォルト設定が使用されます。

カーネルの設定オプションについての詳細をここで扱うことはできません。カーネルの設定時に利用できるヘルプテキストを参照してください。/usr/src/linux/Documentationでは常に最新のカーネルマニュアルをご利用いただけます。

### 9.3.1 コマンドラインでの設定

カーネルを設定するには、ディレクトリ/usr/src/linuxに移動してmake configコマンドを入力します。カーネルでサポートが必要な機能を選択します。通常次のような2つまたは3つのオプションがあります。Ⓨ、Ⓝ、Ⓜ。Ⓜは、このデバイスをカーネルに直接コンパイルするのではなく、モジュールとしてロードすることを意味します。システムのブートに必要なドライバは、Ⓨを使用してカーネルに統合する必要があります。Enterを押して、.configファイルから読み込まれたデフォルト設定を確認します。他のキーを押すと、関連オプションに関する簡潔なヘルプテキストが表示されます。

### 9.3.2 テキストモードでの設定

カーネルを設定するには、menuconfigを使用する方が簡単です。必要な場合は、YaSTでncurses-develをインストールしてください。make menuconfigコマンドを使用してカーネル設定を開始します。

設定変更が軽微な場合、すべてのプロンプトに応える必要はありません。代わりに、メニューを使用して特定のセクションに直接アクセスします。デフォルト設定は.configファイルからロードされます。異なる設定をロードするには、［'Load an Alternate Configuration File'］を選択してファイル名を入力します。

### 9.3.3 X Window Systemでの設定

X Window System(xf86パッケージ)とQT開発パッケージ(qt3-devel)のインストールと設定を完了している場合は、make xconfigコマンドを使用して設定用グラフィカルユーザインタフェースにアクセスできます。X Window Systemにrootでログインしていない場合は、suxコマンドを入力し、ディスプレイへのアクセス権付きroot-shellを取得します。デフォルト設定は.configファイルからロードされます。make xconfigを使用した設定は他の設定ほど適切に維持されないので、この設定方法を使用した後にはmake oldconfigコマンドを実行してください。

## 9.4 カーネルモジュール

PCハードウェアコンポーネントは多様性に富んでいます。このハードウェアを正しく使用するには、オペレーティングシステム(Linuxではカーネル)がこのハードウェアへのアクセスに使用できる「ドライバ」が必要です。システムにドライバを統合するには、基本的に2つの方法があります。

- ドライバをカーネルに直接コンパイルできます。この種のカーネルは(1個で)、*モノリシックカーネル*と呼ばれます。一部のドライバでは、使用できるのはこの形式だけです。
- ドライバを必要に応じてカーネルにロードできます。この場合、カーネルは*モジュール化されたカーネル*と呼ばれます。この種のカーネルには、実際に必要なドライバだけがロードされるため、カーネルに不要なものが含まれないというメリットがあります。

どのドライバをカーネルにコンパイルするか、どのドライバをランタイムモジュールとしてロードするかは、カーネル設定で定義されます。基本的に、システムのブートに不要なコンポーネントはモジュールとしてビルドする必要があります。これにより、カーネルが大きすぎてBIOSやブートローダでロードできないということがなくなります。ext2用のドライバ、SCSIベースシステムのSCSIドライバ、および類似のドライバは、カーネルにコンパイルする必要があります。これに対して、isofs、msdos、soundなど、コンピュータシステムの起動に不要なアイテムは、モジュールとしてビルドする必要があります。

**Tip**

システムのブートに必要なドライバでもモジュールとしてビルドすることができます。この場合、ブート時に初期RAMディスクがこれらのモジュールで使用されます。

**Tip**

カーネルモジュールは/lib/modules/<version>にあります。ここでversionは現在のカーネルバージョンを表します。

### 9.4.1 hwinfoを使用したハードウェア検出

hwinfoはシステムのハードウェアを検出し、そのハードウェアの実行に必要なドライバを選択できます。hwinfo --helpと入力すると、このコマンドに関する短い説明が表示されます。たとえば、SCSIデバイスに関する情報が必要な場合は、hwinfo --scsiコマンドを使用します。この情報はすべて、YaSTのハードウェア情報モジュールにも用意されています。

### 9.4.2 モジュールの処理

カーネルにモジュールをロードするユーティリティはパッケージmodule-init-toolsから利用できます。使用可能なコマンドは次のとおりです。

**insmod** insmodは、要求されたモジュールを/lib/modules/<version>のサブディレクトリ内で検索してからロードします。ただし、insmodではなくmodprobeを使用する方が適切です。modprobeは同時にモジュールの依存性もチェックするためです。

**rmmod** 要求されたモジュールをアンロードします。これは、このモジュールが不要になった場合にのみ可能です。たとえば、まだCDがマウントされている間は、isofsモジュールをアンロードできません。

**depmod** /lib/modules/<version>に、すべてのモジュールの依存関係を定義するmodules.depファイルを作成します。選択したモジュールと共に依存モジュールをすべて確実にロードするには、このファイルが必要です。 このファイルが存在しない場合は、システムの起動後に作成されます。

**modprobe** 指定のモジュールの依存関係を考慮して、そのモジュールをロードまたはアンロードします。このコマンドはきわめて強力であり、いずれかのモジュールが正常にロードされるまで、特定のタイプの全モジュールを検出するなど、さまざまな用途に使用できます。insmodとは異なり、modprobeは/etc/modprobe.confを検査します。このため、モジュールの推奨ロード方法となっています。この項目の詳細については、対応するマニュアルページを参照してください。

**lsmod** 現在ロードされているモジュールと、そのモジュールを使用している他のモジュールの数を表示します。カーネルデーモンにより起動されるモジュールは、autocleanタグで示されます。このラベルは、これらのモジュールがアイドル時間の上限に達すると自動的に削除されることを示します。

**modinfo** モジュール情報を表示します。この情報はモジュール自体から抽出されるので、表示できるのはドライバ開発者が作成した情報だけです。たとえば、作成者、説明、ライセンス、モジュールパラメータ、依存関係、エイリアスなどが表示されます。

### 9.4.3 /etc/modprobe.conf

モジュールのロードは、/etc/modprobe.confファイル、/etc/modprobe.conf.localファイルおよび/etc/modprobe.dディレクトリの影響を受けます。man modprobe.confのマニュアルページを参照してください。このファイルには、ハードウェアに直接アクセスするモジュールのパラメータを入力する必要があります。たとえば、CD-ROMドライバやネットワークドライバなどのモジュールは、システム固有のオプションを必要とする場合があります。ここで使用されるパラメータの説明は、カーネルソースに含まれています。kernel-sourceパッケージをインストールし、/usr/src/linux/Documentationディレクトリにあるドキュメントを参照してください。

### 9.4.4 Kmod—カーネルモジュールローダ

カーネルモジュールローダは、モジュールを使用する上で最も優れた手段です。Kmodはバックグラウンドモニタリングを実行し、カーネル内で関連機能が必要になると、必須モジュールがmodprobeによりロードされていることを確認します。

Kmodを使用するには、カーネル設定で［'Kernel module loader'］オプション(CONFIG_KMOD)を有効にします。Kmodはモジュールを自動的にアンロードするようには設計されていません。現在のRAM容量の観点からは、メモリ使用量を節約できるというメリットがあります。

## 9.5 カーネルのコンパイル

► **x86, AMD64, EM64T**
「bzImage」をコンパイルすることをお勧めします。原則として、このコンパイルによりカーネルが大きくなりすぎるという問題が回避されます。「zImage」の作成時に選択する機能が多すぎると、このような問題が発生しがちです。その場合は、kernel too big や System is too big などのエラーメッセージが表示されます。◄

カーネル設定をカスタマイズした後項9.3.「カーネル設定」で説明されているように、次のコマンドを入力してコンパイルを開始します(必ず先に/usr/src/linuxディレクトリに移動してください)。

```
make clean
make bzImage
```

この2つのコマンドは、1つのコマンドラインとして入力できます。

```
make clean bzImage
```

コンパイルに成功すると、圧縮済みカーネルが/usr/src/linux/arch/<arch>/bootに配置されます。カーネルイメージ(カーネルを含むファイル)はbzImageと呼ばれます。

このファイルが見つからない場合は、カーネルのコンパイル中にエラーが発生している場合があります。Bashシェルで、次のコマンドを入力してカーネルのコンパイルを再開し、出力をkernel.outファイルに書き込みます。

```
make bzImage V=1 2>&1 | tee kernel.out
```

カーネルの各部をモジュールとしてロードするように設定した場合は、モジュールのコンパイルを開始します。そのためには、make modulesを使用します。

## 9.6 カーネルのインストール

コンパイルしたカーネルは、ブートできるようにインストールする必要があります。カーネルは、必ず/bootディレクトリにインストールします。そのためには次のコマンドを使用します。

```
INSTALL_PATH=/boot make install
```

ここで、コンパイル済みのモジュールをインストールする必要があります。make modules_installを入力して、/lib/modules/<version>内の正しいターゲットディレクトリにコピーします。カーネルのバージョンが同一の場合は、古いモジュールが上書きされます。ただし、オリジナルモジュールは、CDからカーネルと共に再インストールできます。

**Tip**

予期しない影響を回避するために、機能をカーネルに直接コンパイルしたモジュールが/lib/modules/<version>から削除されていることを確認してください。これは、あまり経験のないユーザにカーネルのコンパイルをできるだけ避けるようにお勧めする理由の1つです。

**Tip**

GRUBで古いカーネル(/boot/vmlinuz.old)をブートできるように、/boot/grub/menu.lstファイルにブートイメージとしてラベルLinux.oldを追加します。この手順の詳細については章 8. ブートローダを参照してください。GRUBを再インストールする必要はありません。

/boot/System.mapファイルには、モジュールがカーネル機能を確実に起動するために必要なカーネルシンボルが含まれています。このファイルは現行カーネルに依存します。したがって、カーネルのコンパイルとインストールを完了した後、/usr/src/linux/System.mapを/bootディレクトリにコピーしてください。このファイルはカーネルをコンパイルするたびに再生成されます。「System.map does not match current kernel」のようなエラーメッセージが表示される場合、最も可能性の高い原因は、カーネルのコンパイル後にSystem.mapを/bootにコピーし忘れていることです。

## 9.7 コンパイル後のハードディスクのクリア

ハードディスクの空きスペースが少なくなっている場合は、/usr/src/linuxディレクトリでmake cleanを使用して、カーネルのコンパイル時に生成されたオブジェクトファイルを削除します。ディスクスペースに余裕があって、カーネルを定期的に再設定する予定の場合は、この操作をスキップしてかまいません。変更による影響を受ける部分だけが実際に再コンパイルされるので、カーネルの再コンパイルが大幅に高速になります。

# SUSE LINUXの特殊機能

この章では、さまざまなソフトウェアパッケージ、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。この章の後には、言語および国固有設定(I18NおよびL10N)に関する説明が続きます。

# 10.1 特殊ソフトウェアパッケージ

## 10.1.1 パッケージBashと/etc/profile

Bashがログインシェルとして使用されている場合は、次のinitファイルがすべて読み込まれます。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

ユーザは個人のエントリを~/.profileファイルまたは~/.bashrcファイルに作成できます。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ディレクトリ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
```

個人別設定は、*.oldファイルにコピーしておく必要があります。

## 10.1.2 cronパッケージ

cronテーブルは、/var/cron/tabsにあります。/etc/crontabがシステム全体のcronテーブルとして機能します。タイムテーブルの後に、コマンドを直接実行する必要のあるユーザの名前を入力します。例 10.1.「/etc/crontab内のエントリの例」では、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を持ちます。詳細については、man cronコマンドでmanページを参照してください。

***Example 10.1:*** */etc/crontab内のエントリの例*

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

/etc/crontabファイルは、crontab -eコマンドと一緒に処理できません。これは、エディタに直接ロードして、変更し、保存します。

複数のパッケージによりシェルスクリプトがディレクトリ/etc/cron.hourly、/etc/cron.daily、/etc/cron.weekly、および/etc/cron.monthlyにインストールされます。 これらの命令は、/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。

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ました。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backup-rpmdb、clean-tmp、またはclean-viがあります。

## 10.1.3 ログファイル:パッケージlogrotate

カーネルそのものと一緒になって、定期的にシステムスのテータスおよび特定イベントをログファイルに記録するシステムサービス(デーモン)が数多くあります。これにより、管理者は、一定時におけるシステムのステータスを定期的にチェックし、エラーまたは障害のある機能を認識し、そのトラブルシューティングをピンポイントの精度で実行できます。これらのログファイルは、通常、FHSによって指定された/var/logに格納され、日々大きさを増していきます。こうしたログファイルの増大の制御には、logrotateパッケージが役立ちます。

### 環境設定

logrotateは、 /etc/logrotate.confファイルを使って設定します。特に、include指定は、読み取るファイルを主に設定します。SUSE LINUXは、個々のパッケージで/etc/logrotate.d内のファイル(たとえばsyslogやyast)がインストールされるようにします。

***Example 10.2:*** */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/logrotat

# 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を通じて制御され、毎日/etc/cron.daily/logrotateによって呼び出されます。

**Important**

作成オプションでは、管理者が/etc/permissions*内に定義したすべての設定が読み取られます。 人による変更で矛盾が出ないことを確認してください。

**Important**

### 10.1.4 man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を入力して表示します。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SUSEヘルプシステムを使用して、info ページを表示します。

### 10.1.5 locate コマンド

ファイルをすばやく検索するためのlocateは、インストール済みソフトウェアの標準スコープには含まれていません。必要であれば、パッケー

ジ(find-locate)を手動でインストールします。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。

### 10.1.6 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」に示すオプションを使用します。

***Table 10.1:*** *ulimit:ユーザのためのリソースの設定*

| | |
|---|---|
| -m | 物理メモリの最大サイズ |
| -v | 仮想メモリの最大サイズ |
| -s | スタックの最大サイズ |
| -c | コアファイルの最大サイズ |
| -a | 制限セットの表示 |

システム全体の設定は、/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、独自の~/.bashrcに特別なエントリを作成することは可能です。

***Example 10.3:*** *ulimit:~/.bashrc での設定*

```
# Limits of 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。

**Important**

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。

**Important**

### 10.1.7 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にぴったりです。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)。

さらに、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。

### 10.1.8 ファイル/etc/resolv.conf

ドメイン名は、ファイル/etc/resolv.confを使用して管理されます。詳細については、章 24. ドメインネームシステムを参照してください。

このファイルを更新できるのは、スクリプト/sbin/modify_resolvconfのみで、他のプログラムには/etc/resolv.confファイルを直接変更するパーミッションがありません。このルールを強制することによってのみ、システムのネットワークの環境設定と関連のファイルが一貫性のある状態に維持されます。

### 10.1.9 GNU Emacs用の設定

GNU Emacsは、複合作業環境です。詳しい説明は、`http://www.gnu.org/software/emacs/`にあります。次の項では、GNU Emacsの起動時に処理される設定ファイルについて説明します。

起動時に、Emacsは、カスタマイズおよび事前環境設定用のユーザ、システム管理者、およびディストリビュータ設定が入っているいくつかのファイルを読み取ります。初期化ファイル`~/.emacs`が`/etc/skel`から個々のユーザのホームディレクトリにインストールされます。代わって`.emacs`がファイル`/etc/skel/.gnu-emacs`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(`cp /etc/skel/.gnu-emacs ~/.gnu-emacs`を伴う)にコピーして、必要な設定をそこに作成します。

`.gnu-emacs`は、ファイル`~/.gnu-emacs-custom`を`custom-file`として定義します。ユーザが`customize`オプションで設定を作成すると、その設定が`~/.gnu-emacs-custom`に保存されます。

SUSE LINUXを使用すると、`emacs`パッケージでファイル`site-start.el`がディレクトリー`/usr/share/emacs/site-lisp`にインストールされます。ファイル`site-start.el`は、初期化ファイル`~/.emacs`の前に読み込まれます。その他では、`site-start.el`によって、`psgml`など、Emacsアドオンパッケージによって配布された特殊な設定ファイルが自動的に読み込まれます。このタイプの設定ファイルも`/usr/share/emacs/site-lisp`にあり、常に`suse-start-`で始まる名前になっています。ローカルシステム管理者は、`default.el`内に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`の下のEmacs infoファイルにあります。必要な場合、これらのファイルの読み込みを防止する方法に関する説明もこの場所にあります。

Emacsのコンポーネントは、次の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`emacs`。
- `emacs-x11`(通常インストール済み):X11サポートのあるプログラム。
- `emacs-nox`:X11サポートのないプログラム。
- `emacs-nox`:info形式のオンライン文書。
- `emacs-el`:Emacs Lisp内の未コンパイルの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。
- 必要なら多くのアドオンパッケージをインストールできます。`emacs-auctex` (LaTeX版),、`psgml` (SGMLおよびXML版)、`gnuserv` (クライアント/サーバ運用用)、その他。

### 10.1.10 viの簡単な紹介

プログラミングのみでなく、多くのシステム管理タスクにも、相変わらずテキストエディタが使用されています。Unixでは、viは使いやすい編集機能を提供し、マウスサポート機能を持つ多数のエディタに比べて人間工学の面から優れたエディタとなっています。

#### 動作モード

基本的にviは、次の3つの動作モードを使用します。それは*挿入*モード、コマンドモード、および*拡張*モードです。キーの機能は動作モードに応じて異なります。起動時には、通常、viはコマンドモードに設定されます。まず、モード間で切り替える方法について説明します。

**コマンドモードから挿入モードへ** さまざまな方法があり、追加の場合はⓐ、挿入の場合はⓘ、現在行の下に新規行を挿入する場合はⓞを使用します。

**挿入モードからコマンドモードへ** *挿入*モードを終了するには、Escを押します。viは、*挿入*モードになっていると、終了できません。Escを押す習慣を付けることが大切です。

**コマンドモードから拡張モードへ** viの*拡張*モードを有効にするには、コロン(:)を入力します。*拡張*(*ex*)モードは、単純なものから複雑なものまで各種タスクに使用できる行単位のエディタです。

**拡張モードからコマンドモードへ** *拡張*モードでコマンドを実行した後、エディタは自動的にコマンドモードに戻ります。*拡張*モードでコマンドを実行しないことにした場合は、←を使用してコロンを削除します。エディタはコマンドモードに戻ります。

*挿入*モードから*拡張*モードに直接切り替えることはできません。まず、コマンドモードに切り替える必要があります。

他のエディタと同様に、viにも独自の終了手順があります。*挿入*モードではviを終了できません。最初に、Escを押して*挿入*モードを終了します。その後は、次の2つの選択肢があります。

1. *変更内容を保存せずに終了:* 変更内容を保存せずにエディタを終了するには、コマンドモードで:ⓠ!と入力します。感嘆符!を付けると、viでは変更内容が無視されます。

2. *変更内容を保存して終了:* 変更内容を保存してエディタを終了するには、複数の方法があります。*コマンドモード*では Z Z を使用します。*拡張モード*で変更内容をすべて保存してエディタを終了するには、: w q を入力します。*拡張モード*では、wは「書き込み」、qは「終了」を表します。

### 操作中のvi

viを標準エディタとして使用できます。*挿入モード*で、テキストの入力と削除に←とDelの削除キーを使用します。*カーソル移動*には矢印キーを使用します。

ただし、これらのコントロールキーを使用するとしばしば問題が発生します。これは、特殊なキーコードを使用する端末タイプが多数存在するからです。これは、*コマンドモード*に影響します。Escを押して*挿入モード*からコマンドモードに切り替えます。*コマンドモード*では、H、J、K、およびLを使用してカーソルを移動します。各キーの機能は、以下のとおりです。

H　左に1文字分移動します。

J　下に1行分移動します。

K　上に1行分移動します。

L　右に1文字分移動します。

*コマンドモード*では、コマンドを使用してさまざまな操作を行うことができます。コマンドを2度以上実行するには、単に反復回数を入力してから実際のコマンドを入力します。たとえば、5 L と入力すると、カーソルは右に5文字分移動します。

### 関連資料

viは多様なコマンドをサポートしています。マクロ、ショートカット、名前付きバッファ、および他の多数の便利な機能を使用できます。さまざまなオプションについて詳しく説明することは、本書の範疇を超えています。SUSE LINUXにはviの改良版であるvim (vi improved)が付随しています。このアプリケーションについては、さまざまな情報源があります。

- vimtutorは、vimの対話形式のチュートリアルです。

- vimで:helpコマンドを入力すると、さまざまなヘルプトピックが表示されます。
- vimに関するマニュアルは、http://www.truth.sk/vim/vimbook-OPL.pdfからオンラインで入手できます。
- http://www.vim.orgにあるvimプロジェクトのWebページでは、あらゆる種類のニュース、メーリングリスト、およびその他のドキュメントが提供されます。
- インターネットでは、多数のvimソースが提供されています。http://www.selflinux.org/selflinux/html/vim.html、http://www.linuxgazette.com/node/view/9039、http://www.apmaths.uwo.ca/~xli/vim/vim_tutorial.htmlなど。チュートリアルへのリンクについては、http://linux-universe.com/HOWTO/Vim-HOWTO/vim-tutorial.htmlを参照してください。

**Important**

**VIMライセンス**

vimは、「無償ソフトウェア」です。つまり、作者からはソフトウェアの代金を請求されませんが、資金援助による非営利プロジェクトの支援が奨励されます。このプロジェクトは、ウガンダの貧しい子供たちに対する援助を求めています。詳細については、http://iccf-holland.org/index.html、http://www.vim.org/iccf/、およびhttp://www.iccf.nl/でオンライン情報を参照してください。

**Important**

## 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用に予約されています。 割り当てるコンソールの数を変更する場合は、ファイル/etc/inittabを変更します。

Xを終了せずにXからコンソールを切り替えるには、Ctrl-Alt-F1からCtrl-Alt-F6までを使用します。その後、Alt-F7を押すとXに戻ります。

## 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 /usr/X11R6/lib/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/X11R6/lib/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。 .SUSE LINUX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 (右)を使用してアクセスできます。対応するエントリも/usr/X11R6/lib/X11/Xmodmapに示されます。

詳しい設定は、"Xキーボード拡張"(XKB)上で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。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、Mike Fabianのページにあります。 http://www.suse.de/~mfabian/suse-cjk/input.html.

## 10.4 言語および国固有の設定

SUSE LINUX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RC_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、前述のカテゴリを管理します。関連する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、前述の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、SUSE LINUXのみがRC_LANGを設定します。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。

他の変数は、YaSTの［/etc/sysconfig エディタ］で設定できます。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。言語設定は、ISO規格639 (http://www.evertype.com/standards/iso639/iso639-en.htmlとhttp://www.loc.gov/standards/iso639-2/)に準拠しています。国コード設定は、ISO 3166に規定されています(http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html)。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。localedefコマンドを使用すると、/usr/share/i18n内のファイルからさらに説明ファイルを作成できます。この記述ファイルは、glibc-i18ndataパッケージの一部となります。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 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。

### 起こり得る問題

1000の桁を表すカンマが認識されません。LANGがおそらくenに設定されているにもかかわらず、glibcが使用する記述が/usr/share/lib/en_US/LC_NUMERICに配置されているからです。たとえばLC_NUMERICをen_USに設定する必要があります。

### 関連資料

- *『The GNU C Library Reference Manual』* の 章"Locales and Internationalization"。 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)を参照してください。 file:/usr/share/doc/howto/en/html/Unicode-HOWTO.html.

11

X Windowシステム

# 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、SUSE LINUXでのフォントの使用に関する設定、最適化方法、背景情報に説明するとともに、OpenGLと3Dの設定について説明します。

# 11.1 SaX2によるX11の設定

グラフィカルユーザインタフェースまたはXサーバによって、ハードウェアとソフトウェアの間の通信が処理されます。KDEやGNOMEなどのデスクトップと、多様なウィンドウマネージャで、ユーザとの対話にXサーバが使用されています。

グラフィカルユーザインタフェースは、最初はインストール中に設定されます。その後、このSaX2モジュールを起動して、設定内容を変更します。現在の設定内容を保存して、いつでも再設定できます。現在値が表示され、この値を変更できます。変更する値には、画面解像度、カラー設定、リフレッシュレート、およびモニタのメーカと型式(自動検出される場合)。

新しいグラフィックカードをインストールした直後には、そのグラフィックカード用の3Dアクセラレータをアクティブにするか確認する小さなダイアログが表示されます。［'編集'］をクリックします。入力デバイスおよびディスプレイデバイス用の設定ツールSaX2が別ウィンドウで起動されます。このウィンドウを図 11.1. 「SaX2のメインウィンドウ」に示します。

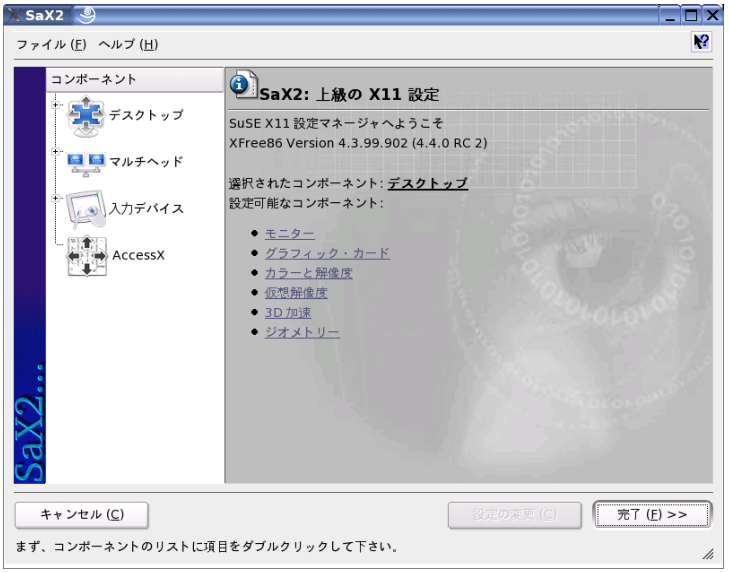


***Figure 11.1:*** *SaX2のメインウィンドウ*

左のナビゲーションバーには4つのメインアイテムがあります。その4つとは［'デスクトップ'］、［'マルチヘッド'］、［'入力デバイス'］、および［'AccessX'］です。［'デスクトップ'］の配下では、モニタ、グラフィックカード、カラー、解像度、画面の位置とサイズを設定します。［'入力デバイス'］の配下で、キーボード、マウス、タッチスクリーンモニタ、グラフィックタブレットを設定できます。［'マルチヘッド'］を使用して、マルチスクリーンを設定します(詳細については、項11.1.7.「マルチヘッド」を参照してください)。［'AccessX'］は、テンキーを使ってマウスポインタを制御する便利なツールです。

モニタとグラフィックカードを選択します。通常、モニタとグラフィックカードは、システムによって自動検出されます。モニタが自動検出されなかった場合、モニタ選択画面が自動的に表示されます。メーカとデバイスのリストからご利用のモニタを選択するか、モニタのマニュアルに記載されているモニタのメーカと型式の値を手動で入力します。モニタのメーカと型式を設定する代わりに、事前に設定されているVESAモードを選択することもできます。

モニタおよびグラフィックカードの設定を完了したら、メインウィンドウで'完了'をクリックしてから、設定をテストします。これにより、設定をデバイスに適合させることができます。画面が安定しない場合、Escを押してテストを直ちに終了させ、リフレッシュレート、解像度、カラーの設定値を減らします。テストを実行したかどうかに関係なく、変更内容はXサーバーの再起動時に有効になります。

### 11.1.1 デスクトップ

'設定を編集する'→'特性'をクリックすると、'モニタのモデル'タブ、'周波数'タブ、および'エキスパート'タブがあるウィンドウが表示されます。

**'モニタのモデル'** ウィンドウの左側リストで、メーカを選択します。右側リストで、型式を選択します。ご利用のモニタ用のLinuxドライバを格納したCDまたはフロッピーディスクがある場合、'メーカ製のディスク'をクリックして、これらをインストールします。

**'［周波数］'** 画面の水平周波数と垂直周波数を入力します。垂直周波数は、画像のリフレッシュレートのもう一つの呼び方です。標準的な場合、許容可能な値が型式から読み込まれ、ここに入力されています。通常、この2つの値を変更する必要はありません。

**'［エキスパート］'** 画面に関するいくつかのオプションを入力します。上部の選択フィールドで、画面解像度と画面のジオメトリの計算に使用する


*Figure 11.2: [ モニタの選択 ]*

方法を定義します。モニタが正しく調整されていないというメッセージが表示され、表示が安定していない場合以外は、変更してはいけません。このタブではさらに、［画面のサイズ］で表示イメージのサイズを変更することと、省電力モード DPMS を有効にすることができます。

**Warning**

**モニタ周波数の設定**

安全機構がありますが、周波数を手動で変更する場合は、注意しなければなりません。不正な値を設定した結果、モニタが壊れることがあります。確信がもてない場合、モニタのマニュアルを参照してください。

**Warning**

## 11.1.2 グラフィックカード

［グラフィックカード］ダイアログには2つのタブがあります。［'一般'］と［'エキスパート'］です。［'一般'］タブでは、左側のリストでグラフィックカードのメーカを選択し、右側のリストで型式を選択します。

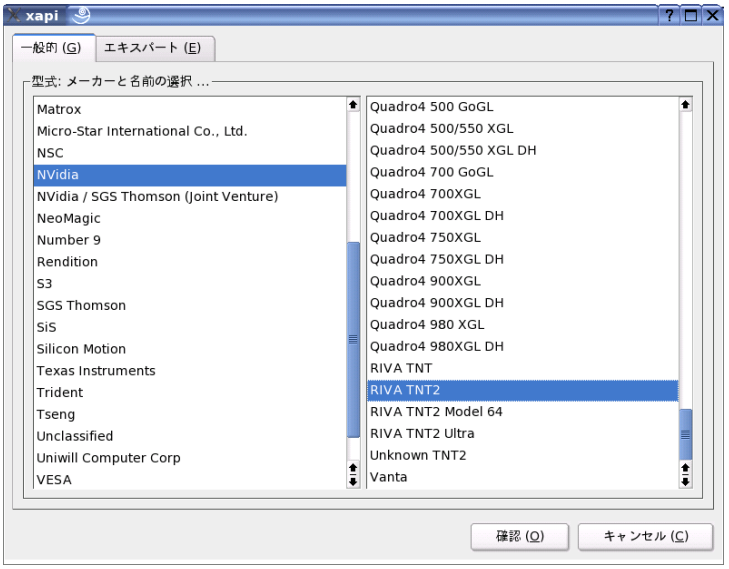


*Figure 11.3: グラフィックカードの選択*

［'エキスパート'］タブには、より詳細な設定機能があります。右側には、画面を左回りまたは右回りに回転して縦長にするかどうかを切り替えるラジオボ

タンがあります(調整可能なTFT画面の一部で役立ちます)。［BusID］のエントリは、複数の画面を操作する場合にのみ関連します。通常、このタブでは何も変更する必要はありません。［カードのオプション］フィールドは、熟練していてオプションの意味を知っている場合を除き、変更してはいけません。変更が必要な場合は、グラフィックカードのドキュメントを調べてください。

### 11.1.3 カラーと解像度

［カラーと解像度］には、［'色'］、［'解像度'］、［'エキスパート'］という3つのタブがあります。

'［**色**］' ご利用のハードウェアに応じて、16色(4ビット)、256色(8ビット)、32768色(16ビット)、1670万色(24ビット)の中から適切な色深度を選択します。妥当な表示品質を保つには、少なくとも256色を設定します。

'［**解像度**］' このモジュールでは、ご利用のハードウェアで正常に表示できる解像度と色の組み合わせがすべて提示されます。これにより、不正な設定が原因でハードウェアを損傷する可能性をSUSE LINUXでは低く抑えています。解像度を手動で変更する場合、ハードウェアの付属ドキュメントを調べて、表示可能な値を設定してください。

'［**エキスパート**］' ［解像度］タブで表示される解像度に加えて、このタブで、独自の解像度を追加することができます。追加した解像度は、［解像度］タブのリストに加えられます。

### 11.1.4 仮想解像度

各デスクトップには、フルスクリーン上で表示される一定の解像度があります。加えて、画面に表示可能な領域よりも大きい解像度を設定することができます。デスクトップのマージンを超えてマウスカーソルを動かすと、デスクトップの仮想部分が画面に表示されます。これにより、使用可能な作業スペースが大きくなります。

仮想解像度は2つの方法で設定できます。［'ドラッグ&ドロップ'］を使用する場合、マウスポインタをモニタの画像の上に移動すると、マウスポインタが十字形に変わります。マウスの左ボタンを押したままマウスを動かして、ラスタイメージ(仮想解像度)を拡大します。この方法は、デスクトップで必要な仮想スペースのサイズが不明な場合に適しています。

［'ポップアップメニューから選択'］を使用する場合、ラスタイメージの中程にあるポップアップメニューには、現在使用されている仮想解像度が表示さ

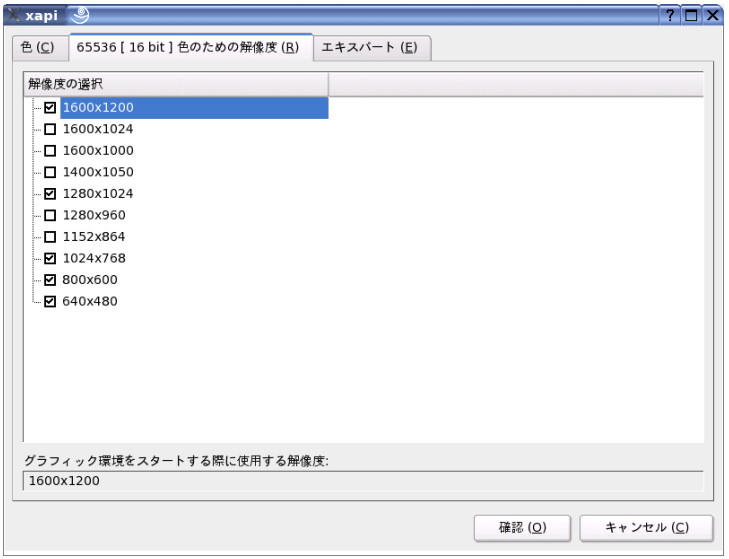


*Figure 11.4: 解 像 度 の 設 定*

れています。デフォルトの仮想解像度の1つを使用する場合は、このメニューで1つを選択します。

### 11.1.5 3D加速

初期インストール時または新規グラフィックカードのインストール時に3Dアクセラレータをアクティブにしなかった場合は、ここでアクティブにすることができます。

### 11.1.6 画面のレイアウト

［位置の変更］と［サイズの変更］という2つのタブで、矢印を使って画面の位置とサイズを精密に調整します。図 11.6. 「画面のジオメトリーの調整」を参照してください。マルチヘッド環境(複数のスクリーン)の場合、［'次の画面'］を使って、別のモニタに切り替えて、そのモニタの位置とサイズを調整します。［'保存'］を押して設定内容を保存します。


*Figure 11.5:* ［仮想解像度］の設定

### 11.1.7　マルチヘッド

ご利用のコンピュータに複数のグラフィックカードが搭載されている場合、または複数出力のグラフィックカードが搭載されている場合、複数のスクリーンをシステムに接続することができます。スクリーンが2台の場合、「デュアルヘッド」と言います。3台以上の場合をマルチヘッドと言います。SaX2は、システム内の複数のグラフィックカードを自動的に検出して、それにしたがって設定を準備します。［マルチヘッドモード］と［スクリーンのレイアウト］を［マルチヘッド］ダイアログで設定します。3つのモードが用意されています。［'Traditional'］(デフォルト)、［'One screen (Xinerama)'］、［'Clone mode'］です。

**［Traditional Multihead］**　各モニタは、個別ユニットを表します。マウスポインタをスクリーン間で切り換えることができます。

**［Cloned Multihead］**　このモードでは、すべてのモニタに同じ内容が表示されます。マウスポインタが見えるのは、メインスクリーンだけです。


*Figure 11.6: 画面のジオメトリーの調整*

**[Xinerama Multihead]** すべてのスクリーンを組み合わせて、1つの大画面が形成されます。プログラムウィンドウは、すべてのスクリーンに自由自在に配置することや、複数のモニタにまたがるサイズに拡大することができます。

マルチヘッド環境のレイアウトでは、個々のスクリーン間の配置と関係を記述します。デフォルトでは、グラフィック・カードの検出順に従って標準レイアウトがSaX2によって設定され、すべての画面が左から右に1列に並べられます。［マルチヘッド］ツールの［'Layout'］ダイアログで、マウスを使ってスクリーンのシンボルをグリッドに移動することにより、モニタの配置を決めます。レイアウトを設定し終えたら、［'Test'］をクリックして、新しい設定を検証します。

現行では、LinuxでのXineramaマルチスレッド環境の3Dサポートはありません。この場合、SaX2で3Dサポートは無効にされます。

### 11.1.8 入力デバイス

**マウス** 自動検出が失敗した場合、このダイアログを使ってマウスを手動で設定します。マウスの形式の説明については、マウスのドキュメントを参照してください。サポート対象のマウスのリストから、ご利用の型式を選択して、テンキーパッドの⑤を押して確認します。

**キーボード** ［キーボード・タイプ］ダイアログの最上部にある［タイプ］選択リストで、使用するキーボードのタイプを選択します。続いて、［配列］リストで、キーボード配列の言語(国固有のキーの配列)を選択します。テストフィールドで、特殊文字が正しく表示されるかを確認します。

対応言語でのアクセント記号付き文字入力の有効と無効の切り替え用チェックボックスのステータスを変更する必要はありません。［'完了'］をクリックして、新しい設定内容をシステムに適用します。

**タッチスクリーン** 現行では、X.Orgは、MicrotouchおよびElo TouchSystemsのタッチスクリーンしかサポートしません。SaX2はモニタを自動検出するだけで、タッチスクリーンは検出しません。タッチスクリーンは、入力デバイスとして処理されます。

タッチスクリーンを設定するには、SaX2を起動して、'入力デバイス' → 'タッチスクリーン'を選択します。'タッチスクリーンの追加'をクリックして、タッチスクリーンを追加します。'完了'をクリックして、設定を保存します。設定をテストする必要はありません。

タッチスクリーンには多様なオプションがあるので、最初に調整する必要があるのが一般的です。残念ながら、Linux環境には、この用途に合う汎用ツールは存在していません。標準設定には、タッチスクリーンの次元に合ったデフォルト値があります。通常は、追加の設定は必要ありません。

**タブレット** 現行では、X.Orgは、限定数のタブレットしかサポートしていません。SaX2では、USBポートまたはシリアルポートに接続されたタブレットを設定できます。設定の観点からすると、タブレットは、マウスと類似している入力デバイスの1つにすぎません。

SaX2を起動して、'入力デバイス' → 'タブレット'を選択します。'追加'をクリックし、表示されたダイアログでメーカを選択し、選択リストから

タブレットを追加します。ペンまたはイレーザを接続している場合、右にあるチェックボックスを有効にします。タブレットがシリアルポートに接続されている場合、ポートを確認します。/dev/ttyS0は1番目のシリアルポートを表します。/dev/ttyS1は2番目のシリアルポートを表します。追加のポートも同様の命名規則が適用されます。'完了'をクリックして、設定を保存します。

### 11.1.9 AccessX

ご使用のコンピュータでマウスを使用しない場合は、SaX2を起動し、AccessXを有効にして、マウントポインタをテンキーパッドのキーで制御できるようにします。各種のキーの機能については、表 11.1.「AccessX—テンキーパッドによるマウスの操作」を参照してください。スライダを使って、キーを押したときのマウスポインタの移動速度を設定します。

***Table 11.1:*** *AccessX—テンキーパッドによるマウスの操作*

| キー | 説明 |
|---|---|
| ÷ | マウスの左ボタンを選択します。 |
| × | マウスの中央ボタンを選択します。 |
| − | マウスの右ボタンを選択します。 |
| 5 | 事前に選択したマウスボタンのクリックイベントを起動します。他のボタンを選択しない場合、マウスの左ボタンがプリセットされます。イベントの後、選択内容はデフォルトにリセットされます。 |
| + | ダブルクリックイベント以外は5と同じように動きます。 |
| 0 | クリックアンドホールドイベント以外は5と同じように動きます。 |
| Del | 事前に0で起動したクリックアンドホールドイベントを解放します。 |
| 7 | カーソルを左上に移動します。 |
| 8 | カーソルを上に移動します。 |
| 9 | カーソルを右上に移動します。 |
| 4 | カーソルを左に移動します。 |

⑥ カーソルを右に移動します。

① カーソルを左下に移動します。

② カーソルを下に移動します。

③ カーソルを右下に移動します。

---

### 11.1.10 関連資料

X Window Systemに関する詳細は、章 11. X Windowシステムを参照してください。

### 11.1.11 ジョイスティック

このモジュールを使用すると、表示リストからメーカと型式を選択してジョイスティックを設定できます。［'テスト'］をクリックして、ジョイスティックが正しく応答するかを確認してください。テストダイアログにジョイスティックのアナログ軸の3つの図が表示され、4つの標準ボタンにマークが入っています。ジョイスティックを移動するか、ボタンを押すと、テストダイアログで反応を調べることができます。ジョイスティックは通常、サウンドカードに接続されているので、サウンドカード設定からもこのモジュールにアクセスできます。

### 11.1.12 キーボード配列の選択

必要なキーボード配列は通常選択した言語に対応していますが、言語に関係なく選択することもできます。テストフィールドを使用して、縦棒|のような特殊記号が正しく表示されるかを確認してください。

### 11.1.13 マウス

このYaSTモジュールでマウスを設定します。マウスの選択手順は、インストールですでに説明済みです。項1.5.3. 「マウス」を参照してください。

## 11.2 X設定の最適化

X.Orgは、X Window Systemのオープンソース実装です。X Window Systemの新規テクノロジおよび標準の開発にも責任を負っているX.Org Foundationにより、さらに開発が続けられています。

マウス、グラフィックカード、モニタ、キーボードなど、使用可能なハードウェアを最適の方法で使用するために、設定を手動で最適化することができます。ここでは、この最適化の一部の側面について説明します。X Window Systemの設定の詳細については、ディレクトリ/usr/share/doc/packages/Xorgにある各種ファイルとman xorg.confのマニュアルページを参照してください。

**Warning**

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 AGの作成者は、損傷に責任を負うことはできません。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。

**Warning**

プログラムSaX2およびxf86configは、デフォルトで/etc/X11にファイルxorg.confを作成します。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。

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。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。最も重要なセクションの概要は、以下のとおりです。

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。セクションは、常に以下の形式になっています。

```
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 11.2. 「/etc/X11/xorg.confのセクション」にあります。

*Table 11.2: /etc/X11/xorg.confのセクション*

| タイプ | 意味 |
|---|---|
| Files | ここではフォントおよびRGBカラーテーブルに使用されるパスを記述します。 |
| ServerFlags | 一般スイッチはここに設定します。 |
| 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。 |
| モニタのモデル | 使用するモニタを記述します。このセクションの要素は、後でScreenの定義で参照する名称、bandwidth、および同期周波数の制限(HorizSyncおよびVertRefresh)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modeline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。 |
| Modes | 特定の画面解像度に関するmodelineパラメータが格納されます。これらのパラメータは、ユーザ指定の値に基づいてSaX2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などは、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、HOWTOファイル/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gzを参照してください。 |
| Device | 特定のグラフィックカードを定義します。グラフィックカードは記述名で参照されます。 |
| Screen | MonitorとDeviceを使用してX.Orgに必要なすべての設定を指定します。Displayサブセクションでは、仮想画面(Virtual)のサイズ、ViewPort、およびこの画面で使用する Modesを指定します。 |

| | |
|---|---|
| ServerLayout | シングルヘッド設定またはマルチヘッド設定のレイアウトを定義します。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。 |

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。次のServerLayoutセクションでは、どれを使用するかが決定されます。

### 11.2.1 Screenセクション

最初に、Screenセクションを調べます。このセクションでは、Monitorセクションと Deviceセクションを組み合わせて、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 11.1.「ファイル/etc/X11/xorg.confのScreenセクション」のようになります。

*Example 11.1: ファイル/etc/X11/xorg.confのScreenセクション*

```
Section "Screen"
 DefaultDepth  16
 SubSection "Display"
 Depth       16
 Modes       "1152x864" "1024x768" "800x600"
 Virtual     1152x864
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]"
Monitor       "Monitor[0]"
EndSection
```

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。

DefaultDepth設定を使用して、特定のカラー設定で起動されない場合にサーバで使用するカラー設定を選択します。各カラー設定ごとにDisplayサブセクションがあります。キーワードDepthで、このサブセクションに有効なカラー設定を割り当てます。Depthに有効な値は8、15、16、および24です。必ずしもすべてのXサーバモジュールがこれらの値をすべてサポートしているわけではありません。

Modesセクションでは、カラー設定に続いて解像度のリストを設定します。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)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。左隣に切り替えるには、Ctrl-Alt--(数字パッド上)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。

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。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16 MBであれば、仮想画面には8ビットカラーで最大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。

### 11.2.2 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。原則として、複数のグラフィックカードがインストールされている場合は、各セクションは順に番号を付けられるだけです。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。

```
Section "Device"
 BoardName     "MGA2064W"
 BusID         "0:19:0"
 Driver        "mga"
 Identifier    "Device[0]"
 VendorName    "Matrox"
 Option        "sw_cursor"
EndSection
```

設定にSaX2を使用すると、Deviceセクションは上記の例のようになります。DriverおよびBusIDは、どちらもコンピュータにインストールされているハードウェアに応じて異なり、SaX2により自動的に検出されます。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。

Driverパラメータにより、このグラフィックカードに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。標準インストールの場合、これはディレクトリ/usr/X11R6/lib/modules/drivers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/X11R6/lib/X11/doc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およびman X.Org)でも確認できます。

### 11.2.3 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。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。

**Warning**

モニタとグラフィックカードの機能を熟知していない場合は、モニタが深刻な損傷を受ける恐れがあるので、modelineを変更しないでください。

**Warning**

独自のモニタ記述を作成する場合は、/usr/X11/lib/X11/doc内のドキュメントを熟読する必要があります。ビデオモード関連のセクションには、特に注意する必要があります。ハードウェアの動作とmodelineの作成方法が詳しく記述されています。

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すべての組み合わせに機能します。

## 11.3 フォントのインストールと設定

SUSE LINUX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、X11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(項11.3.2. 「X11コアフォント」を参照)。新しいxftフォントレンダリングシステムでフォントを使用できるようにするには、インストール先ディレクトリが、/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(項11.3.1. 「Xft」を参照)。

フォントファイルは、/usr/X11R6/lib/X11/fonts/truetypeなどの適切なディレクトリに(rootユーザで)手動でコピーできます。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。このスクリプトが実行する事柄については、スプリプトのマニュアルページ(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。CID-keyedフォントでは、若干異なる手順が必要です。詳細については、項11.3.3. 「CID-Keyedフォント」を参照してください。

XFreeには、完全に異なる2つのフォントシステムがあります。それは、古い*X11*コアフォントシステムと新しく設計された*Xft*および*fontconfig*システムです。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。

### 11.3.1 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とユーザ固有の設定ファイル~/.fonts.confによって制御されます。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用されるので、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(:outline=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:outline=true" family style weight file
```

上記のコマンドの出力は、以下のようになります。

```
 /usr/X11R6/lib/X11/fonts/truetype/FreeSansBold.ttf:FreeSans:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBoldOblique.ttf:FreeMono:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSerif.ttf:FreeSerif:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBoldItalic.ttf:FreeSerif:style=BoldItalic:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansOblique.ttf:FreeSans:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifItalic.ttf:FreeSerif:style=Italic:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoOblique.ttf:FreeMono:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMono.ttf:FreeMono:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSans.ttf:FreeSans:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBold.ttf:FreeSerif:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBoldOblique.ttf:FreeSans:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBold.ttf:FreeMono:style=Bold:weight=200
```

fc-listで調べることができる重要なパラメータ:

***Table 11.3:*** *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。たとえば、urw |
| style | 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 |
| 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 |

### 11.3.2 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、CID-keyed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。Unicodeフォントもかな

り前からサポートされています。X11 コアフォントシステムは1987年に、モノクロのビットマップフォントを処理する目的でX11 R1用に開発されました。上記で説明した拡張機能は、後から追加されたものです。

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フォントを使用した場合にも時間がかかり、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`xset fp 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。これを行う最も簡単な方法は、`sux`コマンドとrootパスワードを入力して、パーミッションを`root`にすることです。`sux`は、Xサーバを起動したユーザのアクセスパーミッションをrootシェルに移すことです。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)。

### 11.3.3 CID-Keyedフォント

他のフォントタイプとは異なり、CID-keyedフォントは任意のディレクトリに簡単にインストールすることはできません。CID-keyedフォントは、/usr/share/ghostscript/Resource/CIDFontディレクトリにインストールしなければなりません。これは、XftおよびfontconfigとX11コアフォントシステムには必要です。

**Tip**

X11で使用可能なフォントの詳細については、http://www.xfree86.org/current/fonts.htmlを参照してください。

## 11.4 OpenGL - 3D 設定

### 11.4.1 ハードウェアサポート

SUSE LINUXには、3Dハードウェアのサポート用に複数のOpenGLドライバが用意されています。 表 11.4. 「サポートされている3Dハードウェア」 で、概要を説明します。

*Table 11.4: サポートされている3Dハードウェア*

| OpenGLドライバ | サポートされているハードウェア |
|---|---|
| nVidia | nVidia Chips:all except Riva 128(ZX) |
| DRI | 3Dfx Voodoo Banshee、<br>3Dfx Voodoo-3/4/5、<br>Intel i810/i815/i830M、<br>Intel 845G/852GM/855GM/865G、915、<br>Matrox G200/G400/G450/G550、<br>ATI Rage 128(Pro)/Radeon (9250まで) |

初めてYaSTを使用してインストールしている場合には、インストール時にYaSTが3Dサポートを検出すると、3Dアクセラレーションをアクティブにすることができます。nVidiaグラフィックチップの場合は、nVidiaドライバを最初にインストールする必要があります。そのためには、YOU (YaST Online Update)でnVidiaドライバパッチを選択します。ライセンス上の制限により、nVidiaドライバはディストリビューションには含まれていません。

新規インストールではなくアップデートを行う場合、または3Dfxアドオングラフィックアダプタ(Voodoo GraphicsまたはVoodoo-2)を設定する必要がある場合は、3Dハードウェアサポートの設定手順は異なります。手順は、使用するOpenGLドライバによって異なります。次のセクションでさらに詳しく説明します。

### 11.4.2 OpenGLドライバ

nVidiaとDRIのOpenGLドライバは、SaX2で簡単に設定できます。nVidiaアダプタの場合は、nVidiaドライバを最初にインストールする必要があります。コマンド`3Ddiag`を入力し、nVidiaまたはDRIの設定が正しいかどうか検査します。

セキュリティ上の理由から、グループ`video`に属しているユーザのみに、3Dハードウェアに対するアクセスが許可されています。そのため、すべてのローカルユーザを、このグループにメンバーとして所属させます。`video`グループに属していない場合は、OpenGLアプリケーションに対してOpenGLドライバの低速な*software rendering fallback (ソフトウェアレンダリング代替機能)*が使用されます。コマンド`id`を使用して、現在のユーザが`video`グループに属しているかどうか検査します。属していない場合は、YaSTを使用してそのユーザをグループに追加します。

### 11.4.3 診断ツール3Ddiag

診断ツール3Ddiagを使用すると、SUSE LINUXにおける3D設定を検証できます。3Ddiagは、ターミナルから起動する必要があるコマンドラインツールです。`3Ddiag -h`を入力すると、3Ddiagで使用可能なオプションをリストできます。

X.Org設定を検証するために、このツールで、3Dサポートに必要なパッケージがインストールされており、正しいOpenGLライブラリとGLX拡張機能が使用されているかどうかがチェックされます。エラーメッセージが表示された場合は、3Ddiagの指示に従います。すべての設定が正しければ、画面上に表示されるのは完了メッセージのみです。

### 11.4.4 OpenGLテストユーティリティ

OpenGLをテストするには、プログラムglxgearsと、tuxracerおよびarmagetronのようなゲーム(両者のパッケージ名は同じ)が役に立ちます。3Dサポートがアクティブになっている場合、比較的新しいコンピュータ上ではそれらのゲームをスムーズに実行できるはずです。3Dサポートがないと、それらのゲームは非常に遅くなります(コマ送り状態)。glxinfoコマンドを使用して、3Dがアクティブであることを確認します。アクティブであれば、direct rendering:  Yesを含む行が出力されます。

### 11.4.5 トラブルシューティング

OpenGL 3Dのテスト結果が良くない場合(ゲームがスムーズに実行できない場合)は、3Ddiagを使用して、設定に(エラーメッセージ中に)エラーがないか確認します。エラーを訂正しても状況が変わらない場合、あるいは、エラーメッセージが表示されない場合は、X.Orgのログファイルを参照してください。

多くの場合、X.Orgのログファイル/var/log/Xorg.0.logの中に、DRI is disabledという行が見つかります。正確な原因は、ログファイルを厳密に調べない限り見つかりません。この作業にはある程度の経験が必要になります。

この場合には、3Ddiagですでにエラーが検出されているため、設定エラーは存在しません。そのため、この時点での唯一の選択肢は、DRIドライバのsoftware rendering fallback (ソフトウェアレンダリング代替機能)を使用することです。これは、3Dハードウェアサポートを提供していません。OpenGLの表示エラーが発生したり、OpenGLが不安定な場合にも、3Dサポートを使用することはできません。SaX2を使用して、3Dサポートを完全に使用不可能にします。

### 11.4.6 インストールのサポート

DRIドライバのsoftware rendering fallback (ソフトウェアレンダリング代替機能)を除き、Linuxにおけるすべての OpenGLドライバは開発段階にあり、実験段階のドライバであると見なす必要があります。Linux用の3Dハードウェアアクセラレーションに対する要望が多いため、このドライバは、ディストリビューションに含まれています。OpenGLドライバが実験的な段階にあることを考慮すると、SUSEは、3Dハードウェアアクセラレーションに関するインストールのサポートも提供できず、関連する問題に対する支援も行えません。グラフィカルユーザインタフェース(X Window System)の基本設定に

は、3Dハードウェアアクセラレーションの設定は含まれていません。3Dハードウェアアクセラレーションで問題が発生した場合は、3Dサポートを完全に使用不可にすることをお勧めします。

### 11.4.7 その他のオンラインドキュメント

DRIの詳細については、/usr/X11R6/lib/X11/doc/README.DRI (xorg-x11-doc)を参照してください。nvidiaドライバインストールの詳細については、http://ftp.suse.com/pub/suse/i386/supplementary/X/nvidia-installer-HOWTO.htmlを参照してください。

# プリンタの運用

この章では、プリンタの運用に関する全般的な情報を提供し、ネットワーク内でプリンタを運用するための適切なソリューションを見出す作業を支援します。特に、CUPSの運用について重点的に説明します。詳細なトラブルシューティングの項では、プリンタの運用における最も一般的な問題と、その回避策について説明します。

# 12.1 準備と他の考慮事項

CUPSは、SUSE LINUX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 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。プリンタを購入するときは、ハードウェア(コンピュータ)がサポートしているインタフェースを採用していること、および適切なプリンタ言語が使用できることを確認してください。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のですが、かなり効率的です。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hpijsドライバを除き、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(通常はGDIプリンタ)**
独自規格のプリンタでは通常、1つまたは複数のWindowsドライバだけが使用可能です。これらのプリンタは、一般的なプリンタ言語をどれもサポートしていませんし、これらのプリンタが使用しているプリンタ言語は、新しいモデルがリリースされた段階で変更される可能性もあります。詳細については、項12.8.1.「標準的なプリンタ言語をサポートしないプリンタ」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チェックしてください。

- `http://cdb.suse.de/`—SUSE LINUXプリンタデータベース
- `http://www.linuxprinting.org/`—LinuxPrinting.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/catalog.devices`—ドライバに付属

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「perfectly supported」(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。

## 12.2 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。

すべてのプリンタには専用のプリンタキューが存在します。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。

フィルタを使用すると、ユーザが印刷するデータ(ASCII、PostScript、PDF、JPEGなど)がプリンタ固有のデータ(PostScript、PCL、ESC/Pなど)に変換されます。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Ghostscriptを使用して、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Ghostscript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。

## 12.3 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。プリンタ接続の詳細については、`http://portal.suse.com`にアクセスしてSupport Database (サポートデータベース)で記事「*CUPS in a Nutshell*」を参照してください。検索ダイアログに「*cups*」と入力すると、この記事を検索できます。

**Warning**

**コンピュータへのケーブル接続**

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。他の種類の接続を使用する場合、変更を加える前に、システム(コンピュータ)をシャットダウンする必要があります。

**Warning**

## 12.4 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のインストール時には、PostScriptサポート機能のないプリンタでも使用できるように、多数のPPDファイルが事前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。詳細については、項12.7.4. 「各種パッケージ内のPPDファイル」および項12.8.2. 「特定のPostScriptプリンタに適したPPDファイルが入手できない」を参照してください。

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(項12.5.1. 「手動設定」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。

## 12.5 プリンタの設定

プリンタをコンピュータに接続し、ソフトウェアをインストールした後、システム内でプリンタ(論理プリンタ)をインストールする必要があります。可能であれば、この操作にはSUSE LINUXに付属のツールを使用してください。SUSE LINUXはセキュリティに最大の注意を払っていますが、サードパーティのツールは多くの場合、セキュリティ上の制限に関して課題を残しており、最終的には、利点より煩雑さが勝る傾向があります。

### 12.5.1 ローカルプリンタ

ログイン時に未設定のローカルプリンタが検出されると、YaSTはその設定を開始します。この場合、設定についての次の説明と同じダイアログが使用されます。

プリンタを設定するには、YaSTコントロールセンターで'ハードウェア' → 'プリンタ'の順に選択します。これでプリンタ設定のメインウィンドウが開きます。このウィンドウでは、検出されたデバイスのリストが上部に表示されます。下の部分にはこれまでに設定されているすべてのキューのリストが表示されます。使用しているプリンタが検出されない場合は、手動で設定します。

**Important**

'プリンタ'エントリがYaSTコントロールセンターで使用できない場合は、その原因としてほとんどの場合、`yast2-printer`パッケージがインストールされていない可能性があります。この問題を解決するには、`yast2-printer`パッケージをインストールして、YaSTを再起動します。

**Important**

#### 自動設定

パラレルまたはUSBポートを自動的に設定し、接続されたプリンタを検出できる場合、YaSTはプリンタを自動で構成できます。プリンタデータベース内には、自動ハードウェア検出のときにYaSTが検索するプリンタID文字列も必要です。ハードウェアIDがモデル指定と異なっている場合は、手動でモデルを選択します。

すべてが正しく機能できるようにするために、各設定をYaSTの印刷テスト機能でチェックしてください。YaSTのテストページには、テストする設定についての重要な情報もあります。

### 手動設定

自動設定の条件が満たされない場合、またはカスタム設定が必要な場合は、プリンタを手動で設定します。自動検出の成功の条件およびプリンタモデルについてデータベース中に見つかる情報の量に応じて、YaSTが正しい設定を自動的に判別できる場合もあれば、最低限の適切な事前選択を行う場合もあります。

設定する必要があるパラメータは次のとおりです。

**ハードウェア接続(ポート)** ハードウェア接続の設定は、YaSTがハードウェアの自動検出のときに、プリンタを検出できるようになっていたかどうかによって異なります。プリンタモデルを自動的に検出できる場合、YaSTは、プリンタ接続がハードウェアレベルで機能し、この点では設定変更が必要ないと判断します。YaSTがプリンタモデルを自動検出できない場合は、ハードウェアレベルでの接続に問題がある可能性があります。この場合、接続の設定にある程度の手動介入が必要になります。

**キューの名前** キュー名は、印刷コマンドを実行するときに使用します。名前は比較的短いものにし、英小文字と数字だけを使用して指定してください。

**プリンタモデルおよびPPDファイル** 使用するGhostscriptドライバやドライバ用のプリンタフィルタに関するパラメータなど、プリンタ固有パラメータはすべて、PPD (PostScript Printer Description)ファイルに保存されます。PPDファイルの詳細については、項12.4.「ソフトウェアのインストール」を参照してください。

多くのプリンタモデルでは、いくつかのGhostscriptドライバが特定のモデルで機能する場合などに、複数のPPDファイルを使用できます。メーカおよびモデルを選択すると、YaSTがプリンタに対応するPPDファイルを選択します。モデルに対して複数のPPDファイルを使用できる場合、YaSTはそれらのうちの1つ(通常は、推奨とマーク付けされています)をデフォルトとします。［'編集'］を選択すると、デフォルトのPPDファイルを変更できます。

PostScript以外のモデルの場合、プリンタ固有のデータはすべて、Ghostscriptドライバによって作成されます。この理由から、ドライバ設定は、出力品質を左右する最も重要な要因の1つです。印刷出力は、選択したGhostscriptドライバの種類(PPDファイル)とその指定したオプションの両方の影響を受けます。必要に応じて、［'編集'］を選択して(PPDファイルで使用できるようにした場合のように)追加オプションを変更します。

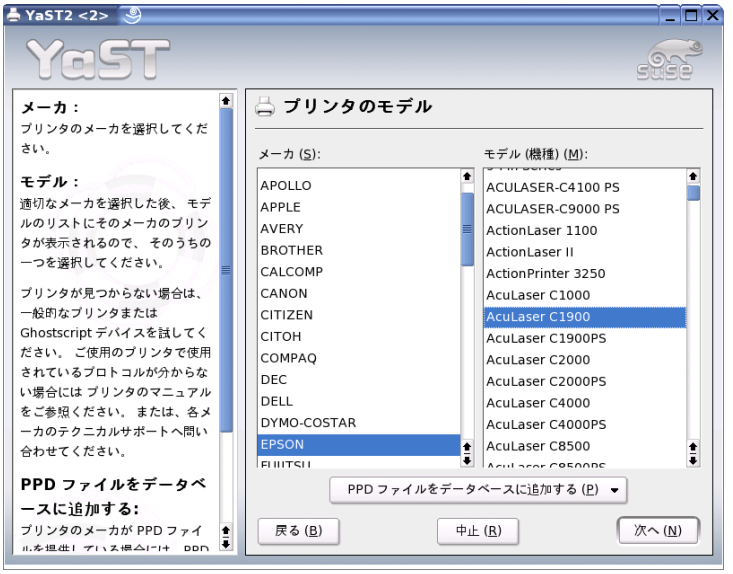


*Figure 12.1: プリンタモデルの選択*

テストページを印刷して、設定が予想どおりに機能するかどうかを確認してください。たとえば、数ページがほとんど何も印刷されない状態になるなど、出力が文字化けした場合は、最初にすべての用紙を取り出し、YaSTがテストを実行できないようにすると、プリンタを停止できます。

プリンタデータベースに使用中のモデルのエントリが含まれていない場合、［'Add PPD File to Database (PPDファイルをデータベースに追加)'］を選択して、新しいPPDファイルを追加したり、基本PPDファイルのコレクションを使用して、プリンタを標準プリンタ言語で機能させることができます。このためには、プリンタメーカとして［'UNKNOWN MANUFACTURER (不明なメーカ)'］を選択します。

**詳細な設定**　通常、詳細設定を変更する必要はありません。

### コマンドラインツールによるプリンタの設定

項12.5.3.「コマンドラインツールによる設定」で説明しているように、コ

マンドラインツールを使用してプリンタを手動で設定するには、バックエンド(usbなど)とパラメータ(/dev/usb/lpなど)で構成されるデバイスURI (uniform resource identifier)が必要です。たとえば、完全URIは`parallel:/dev/lp0` (パラレルポート1に接続されているプリンタ)または`usb:/dev/usb/lp1` (USBポートに接続されている最初に検出されたプリンタ)などとなります。

## 12.5.2 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、いくつかのメーカはその標準に拡張(変更)を加えました。それらのメーカ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、メーカは少数のオペレーティングシステム用に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。ここでは、これらのプロトコルに関する詳細な情報について説明します。

**socket** *socket*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の例は、`socket://host-printer:9100/`です。

**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。もちろん、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://host-printer/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://host-printer/psおよびipp://host-cupsserver/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/server/printer、smb://user:passwおよびsmb://server/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チェックします。次に例を示します。

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

### 12.5.3 設定タスク

設定タスクは、YaSTまたはコマンドラインツールを使用して実行できます。

#### YaSTを使用するネットワーク内でのCUPSの設定

ネットワークプリンタは、YaSTを使用して構成する必要があります。YaSTは、設定を簡単にし、CUPSでのセキュリティ制約の扱いに最も優れています(項12.7.2. 「CUPS Webフロントエンドの管理」を参照)。

ネットワークでのCUPSのインストールのガイドラインについては、http://portal.suse.comにアクセスしてSupport Database (サポートデータベース)で記事「*CUPS in a Nutshell*」を参照してください。

### コマンドラインツールによる設定

代わりに、lpadminやlpoptionsなどのコマンドラインツールを使用してCUPSを設定することもできます。準備作業が完了した後に(PPDファイルとデバイス名が把握できた場合は)、次のステップを実行する必要があります。

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

-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://192.168.1.0:9100/ -P \
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

### オプションの変更

システムのインストール時には、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。すべての印刷ジョブに対してこれらのオプションを変更すること(使用する印刷ツールにもよります)、または後で指定すること(たとえば、YaSTを使用)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:

   ```
   解像度/出力解像度:150dpi *300dpi 600dpi
   ```

   有効になっているデフォルトのオプションは、左側にあるアスタリスク(*)によって明示されます。

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. 新しい設定値のチェック:

   ```
   lpoptions -p queue -l

   解像度/出力解像度:150dpi 300dpi *600dpi
   ```

# 12.6 アプリケーション用の設定

アプリケーションは、コマンドラインツールの場合と同様に、既存のプリンタキューに依存しています。 通常は、特定のアプリケーション用にプリンタを再設定する必要はありません。使用可能なキューを使用して、アプリケーションから印刷できるようになっています。

## 12.6.1 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` ⟨*queuename*⟩ ⟨*filename*⟩」を入力し、⟨*queuename*⟩および⟨*filename*⟩を対応する名前で置き換えます。

## 12.6.2 コマンドラインツールを使用するアプリケーションからの印刷

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` ⟨*queuename*⟩と入力します。KDEプログラムでこの処理を行うには、［'Print through an external program (外部プログラムによる印刷)'］を有効にします。このオプションを有効にしないと、印刷コマンドを入力できません。

## 12.6.3 CUPS印刷システムの使用

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は、KDE以外のアプリケーションの印刷ダイアログでkprinterまたはkprinter --stdinを印刷コマンドとして指定すると、そのような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できます。これら2つのコマンドのどちらが選択されるかは、アプリケーション自身の動作によって決定されます。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。

## 12.7 SUSE LINUXでの特殊機能

CUPSの多くの機能は、SUSE LINUXで使用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。

### 12.7.1 CUPSサーバとファイアウォール

CUPSをネットワークサーバのクライアントとして設定するには、複数の方法があります。

- ネットワークサーバ上のキューごとに、すべてのジョブを対応するネットワークサーバに転送するときに使用するローカルキューを設定できます。ネットワークサーバの設定に変更があるたびに、すべてのクライアントマシンの再設定が必要になるため、通常、この方法はお勧めしません。
- 印刷ジョブを1つのネットワークサーバに直接転送することもできます。このタイプの設定では、CUPSデーモンを実行しないでください。lpr(または他のプログラムの対応ライブラリコール)により、ジョブをネットワークサーバに直接送信できます。ただし、ローカルプリンタで印刷する必要がある場合、この設定は機能しません。
- CUPSデーモンは、使用可能なキューを通知するために他のネットワークサーバから送信されるIPPブロードキャストパケットをリスニングできます。これは、リモートCUPSサーバを介した印刷に最適のCUPS設定です。ただし、攻撃者がキューと共にデーモンのIPPブロードキャストを

送信し、ローカルデーモンが偽のキューにアクセスする危険性があります。その場合、問題のキューがローカルサーバ上の別のキューと同じ名前で表示され、IPPパケットが早期に受信されると、ジョブが実際には攻撃者のサーバに送信されているのに、ジョブの所有者はローカルサーバに送信されていると考える可能性があります。この方法を使用するには、ポート631/UDPを着信パケット用にオープンしておく必要があります。

YaSTは、すべてのネットワークホストをスキャンして、このサービスが提供されているかどうかを確認し、IPPブロードキャストをリスニングして、CUPSサーバを検索できます。第2の方法は、提案用にCUPSサーバを検出するために、システムインストールのときに使用されます。この方法を使用するには、ポート631/UDPを着信パケット用にオープンしておく必要があります。

提案ダイアログに表示されるファイアウォールのデフォルト設定では、インタフェースでのIPPブロードキャストは許可されていません。したがって、リモートキューを検出する第2の方法と、リモートキューにアクセスする第3の方法は機能しません。そのため、インタフェースの1つを`internal`(デフォルトでポートをオープン)として指定するか、`external`インタフェースのポートを明示的にオープンして、ファイアウォール設定を変更する必要があります。セキュリティ上の理由から、デフォルトではポートはいずれもオープンされません。第2の方法でポートをオープンしてリモートキューへのアクセスを設定する操作には、セキュリティ上のリスクを伴います。これは、ユーザが受け入れるサーバを攻撃者がブロードキャストする可能性があるからです。

CUPSがインストール時にリモートキューを検出し、通常の操作中にローカルシステムから各種リモートサーバにアクセスできるように、提案されるファイアウォール設定を変更する必要があります。または、ユーザがローカルネットワークホストをスキャンするか、すべてのキューを手動で設定することにより、CUPSサーバを検出することもできます。ただし、前述の理由から、この方法はお勧めしません。

### 12.7.2 CUPS Webフロントエンドの管理

Webフロントエンド(CUPS)またはプリンタ管理ツール(KDE)を使用して管理を行うために、`root`ユーザは、CUPS管理グループである`sys`に所属するCUPS管理者をセットアップし、CUPSパスワードを割り当てる必要があります。そのためには`root`で次のコマンドを使用します。

```
lppasswd -g sys -a root
```

この作業をまだ行っていない場合、Webインタフェースまたは管理ツールを使用した管理はできません。CUPS管理者がまだ設定されていない場合、認証が失敗するためです。rootの代わりに、他のユーザのいずれかをCUPS管理者として任命することができます(項12.7.3. 「CUPS印刷サービス(cupsd)の変更点」を参照)。

## 12.7.3 CUPS印刷サービス(cupsd)の変更点

これらの変更は、初めにSUSE LINUX 9.1に適用されました。

### lpユーザとしてのcupsdの実行

起動時に、cupsdはrootユーザからlpユーザへの切り替えを行います。これは、セキュリティをより高いレベルに引き上げます。CUPS印刷サービスは、無制限のパーミッションを使用する代わりに、印刷サービスで必要とされるパーミッションだけを使用して動作するためです。

しかし、認証(より正確に表現すると、パスワードのチェック)は、/etc/shadowを介して実行することはできません。lpには、/etc/shadowへのアクセス権がないためです。代わりに、/etc/cups/passwd.md5を介したCUPS特有の認証を使用する必要があります。この目的で、CUPS管理グループであるsysに所属していて、CUPSパスワードの割り当てを受けたCUPS管理者を、/etc/cups/passwd.md5ファイル内に記述する必要があります。この作業を行うには、rootとして、次のコマンドを入力します。

```
lppasswd -g sys -a CUPS-admin-name
```

lpでcupsdを実行した場合、/etc/printcapを生成することはできません。lpは、/etc/直下にファイルを作成することを許可されていないためです。そのため、cupsdは/etc/cups/printcapを生成します。アプリケーションが/etc/printcapからキュー名の読み取りのみを実行して引き続き正常に動作できることを保証するために、/etc/printcapは、/etc/cups/printcapを指すシンボリックリンクになっています。

lpでcupsdを実行した場合、ポート631をオープンすることはできません。そのため、rccups reloadを使用してcupsdを再ロードすることはできません。代わりに、rccups restartを使用してください。

#### BrowseAllowとBrowseDenyの一般化された機能

BrowseAllowとBrowseDenyに対して設定されたアクセスパーミッションは、cupsdに対して送信されたすべてのタイプのパッケージに適用されます。/etc/cups/cupsd.conf内にあるデフォルトの設定値は、次のとおりです。

```
BrowseAllow @LOCAL
BrowseDeny All
```

および

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 127.0.0.2
Allow From @LOCAL </Location>
```

この方法では、LOCALホストだけが、CUPSサーバ上のcupsdにアクセスできます。LOCALホストとは、PPPインタフェース以外(より正確に表現すると、IFF_POINTOPOINTフラグがセットされていないインタフェース)に所属するIPアドレスを使用し、そのIPアドレスがCUPSサーバと同じネットワークに所属しているホストのことです。他のすべてのホストから着信したパケットは、即座に拒否されます。

#### cupsdがデフォルトで有効化

標準的なインストールでは、cupsdは自動的に有効になり、追加で手動の操作を行うことなく、CUPSネットワークサーバのキューに対して適切にアクセスすることができます。この機能を使用するには、前の2つの項目(項12.7.3.「lpユーザとしてのcupsdの実行」と項12.7.3.「BrowseAllowとBrowseDenyの一般化された機能」を参照)が必須の前提条件になります。それらが満たされていない場合、cupsdを自動的に有効にする状況で、セキュリティが不十分になります。

### 12.7.4 各種パッケージ内のPPDファイル

#### PPDファイルのみを使用したプリンタ設定

YaSTのプリンタ設定機能は、システムの/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。ベンダおよびモデルのリストから特定の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、cups-drivers-stpパッケージ内にあるGimp-Print 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、エントリ*NickName: ... Foomatic/Ghostscript driverおよび*cupsFilter: ... foomatic-rip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がFoomatic PPDファイルを採用するのは、Foomatic PPDファイルにエントリ*NickName: ... Foomatic ...(推奨)があり、そのファイルがプリンタモデルに一致し、manufacturer-PPDsパッケージにより適したPPDファイルが含まれない場合(次を参照)です。

### cups-drivers-stpパッケージ内のGimp-Print 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、Gimp-Printから取得したCUPSフィルタrastertoprinterを使用できます。このフィルタと、適切なGimp-Print PPDファイルは、cups-drivers-stpパッケージ内に用意されています。Gimp-Print PPDファイルは /usr/share/cups/model/stp/ 内に配置されていて、そのファイル内にエントリ*NickName: ... CUPS+Gimp-Printおよび*cupsFilter: ... rastertoprinterが記述されています。

### manufacturer-PPDsパッケージ内にあるプリンタメーカからの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。YaSTは、次の各条件が満たされている場合、manufacturer-PPDsパッケージから得られたPPDファイルを優先します。

- ハードウェア検出の際に決定されたベンダおよびモデルが、manufacturer-PPDsパッケージから得られたPPDファイル内にあるベンダおよびモデルと一致しています。
- manufacturer-PPDsパッケージから得られたPPDファイルは、そのプリンタモデルに適した唯一のPPDファイルです。または、同じくそのプリンタモデルに一致する*NickName:... Foomatic/Postscript (推奨)エントリがあるFoomatic PPDファイルです。

したがって、YaSTは次のような状況では、manufacturer-PPDsパッケージから得られたどのPPDファイルも使用しません。

- manufacturer-PPDsパッケージから得られたPPDファイルが、プリンタのベンダおよびモデルに一致していません。これは、manufacturer-PPDsパッケージに同様のモデル用にPPDファイルが1つしかない場合、たとえば、一連のモデルの個々のモデルに別々のPPDファイルがないが、PPDファイル内にFunprinter 1000 seriesのような形式でモデル名が指定されている場合に発生します。
- Foomatic PostScript PPDファイルは、推奨されていません。プリンタモデルは、PostScriptモードでは十分効率よく動作しないことがあるからで

す(たとえば、メモリが少なすぎるためにこのモードではプリンタの信頼性が低い、またはプリンタ内蔵プロセッサの能力が低いために動作が遅すぎる、などです)。デフォルトでは、プリンタがPostScriptをサポートしていないこともあります。たとえば、PostScriptサポートがオプションのモジュールという形でしか使用できない場合などです。

`manufacturer-PPDs`パッケージから得られたPPDファイルが特定のPostScriptプリンタに適しているが、上記で説明された理由によってYaSTがそのファイルを設定できない場合、YaST内で該当のプリンタモデルを手動で選択してください。

## 12.8 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。

### 12.8.1 標準的なプリンタ言語をサポートしないプリンタ

一般的なプリンタ言語をどれもサポートせず、特殊な制御シーケンスのみに依存して動作するプリンタを、「*GDI プリンタ*」と呼びます。これらのプリンタは、メーカ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。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、GDI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。一部のメーカは、GDI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。

### 12.8.2 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用して、そのPPDファイルが"Adobe PostScript Printer Description File Format Specification, version 4.3"(Adobe PostScriptプリンタ記述ファイルフォーマット仕様、バージョン4.3)に準拠しているかどうかを確認します。このユーティリティが"FAIL"を返した場合、PPDファイル内のエラーは重大なものであり、おそらく大きな問題を引き起こすと考えられます。`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に問い合わせることも考えられます。

### 12.8.3 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 address (I/O アドレス):378 (16進)
- Interrupt (割り込み):無関係
- Mode (モード):`Normal` (通常)、`SPP`、または`Output Only` (出力専用)
- DMA:disabled (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 12.1. 「/etc/modprobe.conf:最初のパラレルポート用の割り込みモード」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。

***Example 12.1:*** */etc/modprobe.conf:最初のパラレルポート用の割り込みモード*

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

### 12.8.4 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ネットワークのチェック** 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のチェック** 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 12.2.「lpdからのエラーメッセージ」のような応答を受け取った場合、問題はリモートのlpdにあります。

*Example 12.2: lpdからのエラーメッセージ*

```
lpd:ホストにラインプリンタへのアクセス権
lpdがない。キューにプリンタが存在しない。
スプーリングでプリンタが使用できない。
印刷が無効である。
```

**リモートcupsdのチェック** 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。

```
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 12.3.「CUPSネットワークサーバからのブロードキャスト」に示すようになります。

*Example 12.3: CUPSネットワークサーバからのブロードキャスト*

```
ipp://host.domain: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に対する接続を確立できない場合、cupsd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。 lpstat -h host -l -tこのコマンドは、⟨*host*⟩上にあるすべてのキューに関するステータスレポート(おそらく、非常に長い) を返します。これは、該当のcupsd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。

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、項12.5.2. 「ネットワークプリンタ」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チェックします。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_port*⟩ ⟨*IP-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。詳細については、man nmap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。

### 12.8.5 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。

### 12.8.6 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はずなので、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/usr/bin/enable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。

### 12.8.7 CUPSの参照: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 print-server -o`などのコマンドを使用してサーバ上での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 print-server queue-jobnnumber
```

### 12.8.8 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。上位機種のプリンタでは、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` ⟨*print-server*⟩ `-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チェックします。`cancel` ⟨*queue*⟩`-`⟨*jobnumber*⟩ (または`cancel -h` ⟨*print-server*⟩ ⟨*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。

### 12.8.9 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*`内のメッセージをチェックし、問題の原因を識別します。

### 12.8.10 補足情報

多くの具体的問題に対する解決策は、Support Database (サポートデータベース)にあります。プリンタに関する問題が発生した場合は、Support Database (サポートデータベース)の記事「*Installing a Printer*」および「*Printer Configuration from SUSE LINUX 9.2*」を参照してください。キーワード「printer」を使用して、これらの記事を検索できます。

# Linuxでのモバイルコンピューティング

この章ではモバイルコンピューティングにLinuxを使用した様々な例について説明します。様々な分野での使用例を簡潔に紹介し、使用されているハードウェアの基本的な機能についても解説します。電源消費量を最小限に抑える実現性とともに、パフォーマンスを最大限に引き出す特別な要件とオプションに適したソフトウェアソリューションを提案します。この章の終わりでは、今回のテーマに関する最も重要な情報ソースについて説明します。

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。この章ではラップトップやデスクトップシステムに接続可能な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にまで範囲を広げて追っていきます。

# 13.1 ラップトップ

## 13.1.1 ラップトップハードウェアの特性

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の製造元によりPCMCIA標準(*Personal Computer Memory Card International Association*)が開発されました。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。このようなハードウェアのサポートをどのようにLinuxに実装するか、構成中に考慮すべきことは何か、PCMCIAを制御するために使用可能なソフトウェアは何か、起こりうる障害をどのようにトラブルシューティングするか、などの情報は章 14. PCMCIAに記述されています。

## 13.1.2 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必要性を極力減らし、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
- ディスプレイの照明を手動で調整する
- 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カードなど)

- アイドル中にはハードウェアディスクをスピンダウンする

SUSE LINUXでの電源管理およびYaSTの電源管理モジュールに関する背景情報は章 16. 電源管理に記載されています。

### 13.1.3 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はこうした作業をユーザに代わって実行します。

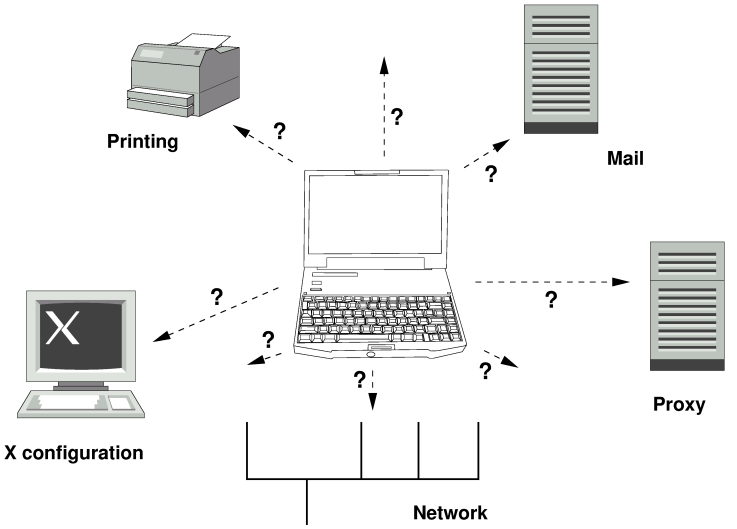


*Figure 13.1: ネットワークでのラップトップの統合*

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。

**ネットワークの設定** 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。

**E-Mail (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 Window Systemの設定** ご使用のラップトップが一時的にプロジェクタ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ではラップトップを組み合せが可能な既存の操作環境に統合させる2つの方法を提供しています。

**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)で操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。SCPMについての情報は章 15. システム設定プロファイル管理を参照してください。kickerアプレットであるKDEのProfile Chooserによりプロファイル間での切り替えが可能です。アプリケーションは切り替える際にrootパスワードを要求します。

**SLP** *サービスロ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についての詳細な情報は章 23. ネットワーク上のSLPサービスを参照してください。

SCPMの重要性は再現可能なシステム条件を有効にし、保持することです。SLPはネットワークに接続されたコンピュータの設定のほとんどを自動化することで設定自体を容易にしています。

### 13.1.4 ソフトウェアオプション

モバイルでの使用を前提として専用ソフトウェアによってカバーされる様々な特殊タスク領域があります。次に例をあげます。システムモニタリング(特に

バッテリの充電)、データ同期、周辺機器との無線通信、インターネット。次のセクションでは、SUSE LINUX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。ラップトップに備えられている再充電可能バッテリの単純状態の表示は、kickerのアプレットKPowersaveによって処理されます。複雑なシステムモニタリングはKSysguardによって実行されます。GNOMEを使用している場合、前述の機能はGNOME ACPI (パネルアプレットとして)およびSystem Monitorによって提供されます。

**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。バッテリで作業する場合は、アイコンがバッテリに変わります。rootパスワードの入力を要求した後、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。これにより異なるタイプの電源でもシステムの動作を設定することができます。電源管理および対応するYaSTモジュールについての情報は章 16. 電源管理を参照してください。

**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。監視するだけでなく、すべてのシステムプロセスを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。

### データの同期

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、

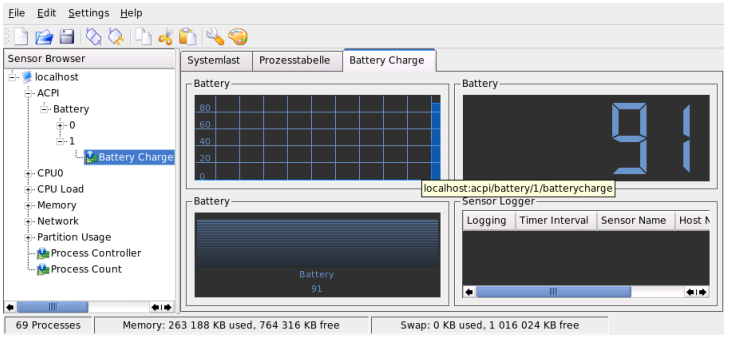


*Figure 13.2: KSysguardでのバッテリ状況のモニタリング*

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** オフィスのネットワークの電子メールを格納するためにIMAPアカウントを使用します。これで電子メールはMozilla Thunderbird Mail、Evolution、またはSUSE LINUX Professional 9.2 ユーザガイドで説明されているKMailなどのような切断型IMAP対応(電子メールクライアント)のいずれかでアクセスできるようになります。電子メールクライアントでの設定が必要になります。これにより送信メッセージには常に同じフォルダを使用できます。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信頼できるフィードバックを受信するために、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として実装されたSMTPサービスを使用します。

**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。詳細については、章 31. ファイルの同期を参照してください。

### 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のと同様に、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と言えます。1台のマシンは独立した無線ネットワークやインターネットを介して互いに接続することができます。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つなぎとしての役目を果たします。モバイルユーザは場所や最適な接続を提供するアクセスポイントによっ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についての詳細は項17.1. 「無線LAN」を参照してください。

**Bluetooth** Bluetoothはすべての無線テクノロジ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テクノロジが適しています。bluetooth、専用アプリケーション、および設定についての詳細は項17.2. 「Bluetooth」を参照してください。

**IrDA** IrDAは狭い範囲での無線テクノロジです。通信を行う両者は相手の見える位置にいなくてはなりません。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。IrDAの詳細情報については、項17.3. 「赤外線データ通信」を参照してください。

### 13.1.5 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。

**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るべきです。これは盗難時の安全性確保にも有効な手段です。SUSE LINUXでの暗号化パーティションの作成については項34.3. 「パーティションとファイルの暗号化」に記載されています。

**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上での一般的なセキュリティ問題については項34.4. 「セキュリティと機密性」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章 17. 無線通信に記載されています。

## 13.2 モバイルハードウェア

SUSE LINUX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にはFireWire、USBハードディスク、USBフラッシュドライブ、デジタルカメラなど、あらゆる種類があります。これらのデバイスは専用のインタフェースからシステムに接続されると同時にホットプラグを介して自動的に検出、設定されます。subfsおよびsubmountはこれらのデバイスがファイルシステムの対応するロケーションに確実にマウントされたことを確認します。ユーザはSUSE LINUXの前バージョンのような手動によるマウント、アンマウントから完全に開放されます。プログラムがデバイスへのアクセスを終了した段階ですぐにデバイスを切断できます。

**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'マイコンピュータ'(KDE)または'コンピュータ'(GNOME)のマウント済みリストに表示されます。アイコンを左クリックすると、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/usb-xxx`または`/media/ieee1394-xxx`でデバイスがマウ

ントされているデバイスのデスクリプタは変更されず、そのままになります。

**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The GIMPを使用して処理できます。GNOMEを使用している場合は、Nautilusのフォルダに写真が表示されます。簡単な画像処理および管理ユーティリティはGThumbです。高度な写真処理はThe GIMPで行います。GThumb以外のプログラムはSUSE LINUX Professional 9.2 ユーザガイドに記載されています。また、デジタルカメラについての章もあります。

**Important**

**モバイルデータドライブの保護**

モバイルハードディスクまたはフラッシュドライブはラップトップと同様、盗難に合う恐れがあります。第三者による不正使用を防ぐため、項34.3. 「パーティションとファイルの暗号化」に記載されているように、ドライブに暗号化パーティションを作成することを推奨します。

**Important**

## 13.3 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項13.1.4. 「無線通信」ですでに説明しました。携帯電話側における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。Linux側の設定は項17.2. 「Bluetooth」および項17.3. 「赤外線データ通信」を参照してください。

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Kontact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。Palm Pilotsのサポートがいったん設定されると、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。どちらのグループウェアについてもSUSE LINUX Professional 9.2 ユーザガイドに記載されています。

Kontactに統合プログラムであるKPilotは独立したユーティリティとしても利用可能です。これについてはSUSE LINUX Professional 9.2 ユーザガイドを参照してください。プログラムKitchenSyncもアドレスデータの同期に使用することができます。

EvolutionおよびKontactについての詳細情報はSUSE LINUX Professional 9.2 ユーザガイドを参照してください。

## 13.4 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://www.linux-on-laptops.com/`では、`http://tuxmobil.org/`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。`http://lists.suse.com/archive/suse-laptop/`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。

ラップトップでのSUSE LINUXの電源管理に関して問題がある場合は、`/usr/share/doc/packages/powersave`にある`README`ファイルを確認することを推奨します。このディレクトリにはテスターや開発者からの最終段階でのフィードバックが盛り込まれます。そのため問題のソリューションについて、有用なヒントが含まれている場合があります。

# PCMCIA

ここでは、PCMCIA ハードウェアおよびソフトウェアのラップトップ固有の側面について説明します。PCMCIAは*Personal Computer Memory Card International Association* の頭文字で、関連するハードウェアとソフトウェアの総称として使用されています。

# 14.1 ハードウェア

最も重要なコンポーネントはPCMCIAカードです。次の2つのタイプがあります。

**PCカード** このタイプのカードは、PCMCIAの黎明期から存在しています。データ伝送に16ビットバスを使用するため、通常はごく廉価です。最近の一部のPCMCIAブリッジは、この種のカードを検出できない場合があります。しかし、検出されれば通常は円滑に動作し、問題が発生することはありません。

**CardBusカード** PCカードより新しい標準です。ビットバスを使用しているため高速ですが、価格も高価です。この種のカードもPCIカードと同様にシステムに統合され、円滑に動作します。

PCMCIAサービスがアクティブの場合、コマンド`cardctl ident`を実行すると挿入されているカードのタイプが表示されます。サポートされているカードのリストは、ディレクトリ`/usr/share/doc/packages/pcmcia`にあるファイル`SUPPORTED.CARDS`で確認してください。このディレクトリには、PCMCIA HOWTOの最新バージョンも用意されています。

2番目に重要なコンポーネントは、カードとPCIバス間の接続を確立するPCMCIAコントローラ、PCカードまたはCardBusブリッジです。共通モデルはすべてサポートされています。コントローラのタイプは、コマンド`pcic_probe`で判別できます。PCIデバイスの場合は、コマンド`lspci -vt`を実行すると、詳細情報が表示されます。

# 14.2 ソフトウェア

## 14.2.1 基本モジュール

必要なカーネルモジュールは、カーネルパッケージに含まれています。加えて、`pcmcia`パッケージと`hotplug`パッケージの2 つが必要です。PCMCIAを起動すると、モジュール`pcmcia_core`、`yenta_socket`、および`ds`がロードされます。非常にまれに、`yenta_socket`の代わりにモジュール`tcic`が必要なことがあります。これらのモジュールは、既存のPCMCIAコントローラを初期化し、基本機能を提供します。

### 14.2.2 カードマネージャ

PCMCIAカードはシステムの実行中に交換できるので、PCMCIAスロットのアクティビティをモニタする必要があります。このタスクは、基本モジュールに実装されているカードサービスにより処理されます。挿入されているカードの初期化は、カードマネージャ(PCカードの場合)またはカーネルのホットプラグシステム(CardBusカードの場合)により処理されます。カードマネージャは、基本モジュールがロードされた後、PCMCIAの起動スクリプトによって起動されます。ホットプラグは自動的にアクティブになります。

カードが挿入されている場合、カードマネージャまたはホットプラグがカードの種類や機能を判定してから、関連モジュールをロードします。モジュールが正常にロードされると、カードマネージャまたはホットプラグはカードの機能に応じて特定の初期化スクリプトを起動します。初期化スクリプトにより、ネットワーク接続の確立、外付けSCSIハードディスクのパーティションのマウント、またはハードウェア固有の他のアクションが実行されます。カードマネージャ用のスクリプトは、ディレクトリ/etc/pcmciaにあります。ホットプラグ用のスクリプトは、/etc/hotplugにあります。カードを取り外すと、カードマネージャまたはホットプラグが、同じスクリプトを使用してすべてのカードアクティビティを終了します。その後、不要になったモジュールがアンロードされます。

これらのアクションをホットプラグイベントと呼びます。ハードディスクまたはパーティションを追加すると(ブロックイベント)、ホットプラグスクリプトはsubfsを使用して新規メディアを/media内ですぐに使用可能にします。古いPCMCIAスクリプトを使用してメディアをマウントするには、ホットプラグ内でsubfsを無効にする必要があります。

PCMCIAとカードイベントの起動は、いずれもシステムログファイル(/var/log/messages)に記録されます。このログファイルには、ロードされたモジュールと実行されたスクリプトが記録されます。

理論上、PCMCIAカードは付加的なアクションなしで取り外すことができます。他にアクティブなネットワーク接続がなければ、この方法はネットワーク、モデム、およびISDNカードに対して完全に機能します。ただし、外付けハードディスクのマウント済みパーティションやNFSディレクトリの場合、この方法は使用できません。この種のユニットは、正しく同期させてアンマウントする必要があります。当然、カードを取り外すと、このような同期とアンマウントはできません。不明点がある場合は、コマンドcardctl ejectを使用して、まだラップトップに挿入されているカードをすべて無効にします。カードを1つだけ無効にするには、cardctl eject 0のようにスロット番号を指定します。

## 14.3 環境設定

YaSTのランレベルエディタを使用して、システムのブート時にPCMCIAを起動するかどうかを指定します。このモジュールを起動するには、'システム' → 'ランレベルエディタ'を使用します。

ファイル/etc/sysconfig/pcmcia内で、次の3つの変数が定義されています。

**PCMCIA_PCIC** PCMCIAコントローラを制御するモジュールを指定します。通常、モジュールは起動スクリプトによって自動的に判別されます。判別されない場合は、ここにモジュールを入力します。自動検出が可能な場合は、この変数は空にしておきます。

**PCMCIA_CORE_OPTS** この変数は、pcmcia_coreモジュールのパラメータ用に設計されました。しかし、これらのパラメータはほとんど使用されていません。オプションについては、pcmcia_core(4)のマニュアルページを参照してください。このマニュアルページはDavid Hindsのpcmcia-csパッケージからの同種モジュールについて言及しているので、カーネルが実際にサポートしているモジュールよりも多数のパラメータが記載されています。cb_およびpc_debugで始まるパラメータは、すべてこれに該当します。

**PCMCIA_BEEP** カードマネージャのアコースティック信号を有効または無効にします。

ファイル/etc/pcmcia/configと/etc/pcmcia/*.confには、PCカードへのドライバ割り当てが含まれています。まず、configが読み込まれてから、*.confファイルがアルファベット順に読み込まれます。カードのための最後のエントリが使用されます。これらのファイルの構文の詳細については、pcmcia(5)のマニュアルページを参照してください。

/etc/sysconfig/hardware/hwcfg-<configurationname>として指定された各ファイルには、CardBusカードへのドライバ割り当てが含まれています。これらのカードは設定時にYaSTによって作成されます。設定名の詳細については、/usr/share/doc/packages/sysconfig/README、またはgetcfg(8)のマニュアルページを参照してください。.

### 14.3.1 ネットワークカード

イーサネット、無線LAN、およびトークンリングの各ネットワークカードは、通常のネットワークカードと同様にYaSTで設定できます。カードが検出

されない場合は、ハードウェア設定でカードタイプPCMCIAを選択してください。ネットワークの設定に関する他の詳細については、項22.4. 「ネットワーク統合」を参照してください。

### 14.3.2 ISDN

他のISDNカードと同様に、ISDN PCカードもYaSTで広範囲に設定できます。PCMCIAカードであれば、どのPCMCIA ISDNカードをリストから選択しても同じです。ハードウェアを設定してプロバイダを選択する場合、動作モードは常にonbootではなくhotplugにする必要があります。PCMCIAカード形式のISDNモデムも存在します。これらは、追加のISDN接続キット付きのモデムカードまたは多機能カードです。この種のカードはモデムと同様に扱われます。

### 14.3.3 モデム

通常、モデムPCカードにはPCMCIA固有の設定はありません。モデムは、挿入されている限り、/dev/modemで利用可能です。一部のPCMCIAモデムカードは、Linuxでサポートされていないソフトモデムです。この種のカードのドライバを使用可能な場合は、システムにインストールする必要があります。

### 14.3.4 SCSIとIDE

適切なドライバモジュールは、カードマネージャまたはホットプラグによってロードされます。SCSIまたはIDEカードを挿入すると、接続されているデバイスが使用可能になります。デバイス名はダイナミックに判別されます。使用可能なSCSIおよびIDEデバイスについての情報は、/proc/scsiおよび/proc/ideに用意されています。

外付け外部ハードディスク、CD-ROMドライブ、およびそれらに類するデバイスは、PCMCIAカードをスロットに挿入する前にスイッチをオンにする必要があります。SCSIデバイスのアクティブ終了を使用します。

**Warning**

**SCSIまたはIDEカードの取り外し**

SCSIまたはIDEカードを取り外す前に、コマンドumountを使用して、接続されているデバイスのパーティションをすべてアンマウントする必要があります。先にアンマウントしない場合、これらのデバイスにアクセスするにはシステムをリブートする必要があります。

**Warning**

## 14.4 ユーティリティ

前述したcardctlユーティリティは、PCMCIA 情報を取得したり特定のアクションを実行するためのメインツールです。詳細については、cardctl(8)のマニュアルページを参照してください。`cardctl`と入力すると、有効なオプションのリストが表示されます。グラフィカルフロントエンドcardinfoを使用すると、cardctlの主機能を制御できます。このグラフィカルフロントエンドを使用するには、`pcmcia-cardinfo`パッケージをインストールします。

その他、`pcmcia`パッケージには、`ifport`、`ifuser`、`probe`および`rcpcmcia`などのユーティリティが含まれています。ただし、通常は使用しません。`pcmcia`パッケージに含まれているファイルを確認するには、コマンド`rpm -ql pcmcia`を入力します。

## 14.5 トラブルシューティング

ラップトップや特定のカードに発生するPCMCIA関連した問題のほとんどは、問題を系統的に調べることで特定して解決できます。最初に、問題がカードにあるのかPCMCIA基本システムにあるのかを確認します。そのためには、カードを挿入していない状態でコンピュータをブートします。基本システムが正常に動作しているように思われる場合は、カードを挿入します。すべてのメッセージのログは、`/var/log/messages`ファイルに出力されます。`tail -f /var/log/messages`を使用して、このファイルをモニタしながら問題の原因を調べます。これにより、問題点を次の2つのいずれかに絞り込むことができます。

### 14.5.1 PCMCIA基本システムが動作しない

システムが「PCMCIA:Starting services」というメッセージを表示してハングする場合、またはブート時に他の異常な現象が発生する場合、システムをリブートし、ブートプロンプトからNOPCMCIA=yesと入力してPCMCIAを無効にします。エラーを分離するには、PCMCIAシステムの3つの基本モジュールを手動で1つずつロードします。

PCMCIAモジュールを手動でロードするには、ユーザrootでコマンドmodprobe pcmcia_core、modprobe yenta_socket、およびmodprobe dsを実行します。ごくまれに、yenta_socketの代わりにtcic、i82365、またはi82092の使用が必要になる場合があります。最初にロードされる2つのモジュールが重要です。

pcmcia_coreのロード中にエラーが発生する場合は、pcmcia_coreのマニュアルページを参照してください。このマニュアルページに記載されているオプションは、最初にコマンドmodprobeでテストできます。たとえば、空きI/O領域をテストするとします。他のハードウェアコンポーネントを妨害している場合は、このテストで問題が発生することがあります。オプションprobe_io=0を指定すると、この問題を回避できます。

```
modprobe  pcmcia_core probe_io=0
```

選択したオプションが正常に実行される場合は、ファイル/etc/sysconfig/pcmcia内で変数PCMCIA_CORE_OPTSに、値probe_io=0を設定します。複数のオプションを指定する場合は、次のようにスペースで区切ってください。

```
PCMCIA_CORE_OPTS="probe_io=0 setup_delay=10"
```

yenta_socketモジュールのロード中に発生するエラーは、ACPIによるリソース割り当てなど、より基本的な問題があることを示します。

ファイル/etc/pcmcia/configと/etc/pcmcia/config.optsは、カードマネージャにより解析されます。この2つのファイル内の設定は、一部がcardmgrの起動に使用され、一部がPCカード用ドライバモジュールのロードに使用されます。IRQ、I/Oポート、およびメモリアドレスの範囲は、ファイル/etc/pcmcia/config.opts内で含めるか除外できます。まれに、不正なI/O領域へのアクセスによりシステムがクラッシュする場合があります。このような場合は、これらの領域を制限してみてください。

### 14.5.2 PCMCIAカードが正常に動作しない

基本的に次の3つのタイプのエラーがあります。カードが検出されない、ドライバをロードできない、またはドライバによるインタフェースの設定が誤っている、などです。カードを制御しているのがカードマネージャであるかホットプラグであるかを知ることが重要です。カードマネージャはPCカードを制御し、ホットプラグはCardBusカードを制御します。

**カードを挿入しても反応がない** カードの挿入時にシステムが反応せず、コマンド`cardctl insert`を手動で実行しても効果がない場合は、PCIデバイスへの割り込みの割り当てが不正である可能性があります。このような場合、ネットワークカードなどの他のPCIデバイスにも問題が発生している可能性があります。その場合は、ブートパラメータ`pci=noacpi`または他のPCIまたはACPIパラメータが役立ちます。

**カードが検出されない** カードが検出されない場合は、`/var/log/messages`にメッセージ「Unsupported Card in Slot x」が記録されます。このメッセージは、単にカードマネージャがカードへのドライバ割り当てに失敗したことを示します。この割り当てには、ファイル`/etc/pcmcia/config`または`/etc/pcmcia/*.conf`が必要です。このドライバデータベースは、既存のエントリをテンプレートとして使用することで、簡単に拡張できます。コマンド`cardctl ident`を入力して、カードの詳細を確認します。詳細については、PCMCIA HOWTO (セクション6)およびpcmcia(5)のマニュアルページを参照してください。`/etc/pcmcia/config`または`/etc/pcmcia/*.conf`を編集した後、コマンド`rcpcmcia reload`を使用してドライバ割り当てを再ロードします。

**ドライバがロードされない** 原因の1つとして、ドライバデータベースに不正な割り当てが含まれていることが考えられます。たとえば、メーカが表面的には修正のないカードモデルに異なるチップを使用していると、この問題が発生する場合があります。一部のモデルは、事前に選択されたものとは異なるドライバでのみ動作したり、異なるドライバを使用する方が動作状況がよい場合があります。このような場合は、カードの詳細情報が必要になります。必要な場合は、問題をメーリングリストにポストするか、Advanced Supportに問い合わせてください。

CardBusカードの場合は、ファイル`/etc/sysconfig/hotplug`にエントリ`HOTPLUG_DEBUG=yes`を挿入する必要があります。その後、ドライバが(正常に)ロードされたかどうかを示すメッセージをシステムログで確認します。

もう1つの原因としてリソースの競合が考えられます。ほとんどのPCMCIAカードでは、どのIRQ、I/Oポート、またはメモリ領域で動作しているかは問題ではありませんが、中には例外もあります。この場合は、一度に1つずつカードをテストし、他のシステムコンポーネント(サウンドカード、IrDA、モデム、プリンタなど)を一時的に無効にします。ユーザrootでコマンドlsdevを実行し、システムのリソース割り当てを確認します。複数のPCIデバイスで同じIRQを使用していても、まったく問題はありません。

可能な解決策の1つは、カードドライバモジュール用の適切なオプションを調べることです。modinfo ⟨*driver*⟩と入力してオプションリストを表示します。ほとんどのモジュールの場合はマニュアルページを使用可能です。rpm -ql pcmcia | grep manと入力すると、pcmciaパッケージに含まれて入る全マニュアルページのリストが表示されます。.オプションをテストできるよう、カードドライバも手動でアンロードできます。

解決策が見つかった時点で、通常、/etc/pcmcia/config.optsにある特定のリソースの使用を許可または禁止することができます。カードドライバ用のオプションも、このファイルに入力できます。たとえば、pcnet_csモジュールをIRQ5で排他的に使用するには、次のエントリを追加します。

```
module pcnet_cs opts irq_list=5
```

**インタフェースの設定の誤り** この場合は、getcfgでインタフェースの設定と設定名を慎重に確認し、設定エラーを訂正します。ファイル/etc/sysconfig/network/config内の変数DEBUGとファイル/etc/sysconfig/hotplug内の変数HOTPLUG_DEBUGをyesに設定する必要があります。他のカードの場合や、この方法で解決できない場合は、カードマネージャまたはホットプラグにより実行されるスクリプトにset -vxという行を追加できます(/var/log/messagesを参照)。この方法を使用すると、スクリプトのコマンドがシステムログに1つずつ記録されます。スクリプト内で重要なセクションを見つけた場合は、端末で関連コマンドを入力してテストします。

## 14.6 関連資料

特定のラップトップモデルに関して役立つ情報は、Linux Laptopホームページhttp://linux-laptop.netで提供されています。またMobilixの

ホームページ(http://tuxmobil.org/)も役に立ちます。これらのページでは、ラップトップのHOWTO、IrDAのHOWTO、および他の多数の関連情報が提供されます。また、SUSE LINUXのSupport Database (http://portal.suse.com)には、モバイルデバイスでSUSE LINUXを使用する場合についての記事があります。これらの記事を検索するには、検索ダイアログにキーワード*notebook*または*laptop*を入力します。

15

システム設定プロファイル管理

# システム設定プロファイル管理

この章では、SCPM (system configuration profile management)について説明します。SCPMを使用して、さまざまな操作環境やハードウェア設定に合わせてコンピュータの設定を調整します。SCPMは、さまざまなシナリオに合わせた一連のシステムプロファイルを管理します。SCPMを使用すると、2つのシステムプロファイル間での切り替えが容易になり、システムを手動で再設定する必要がなくなります。

環境に応じてシステム設定を変更しなければならない場合があります。さまざまな場所で使用されるモバイルコンピュータなどは特にこのケースに該当します。また、デスクトップシステムでも通常使用しているのとは違うハードウェアコンポーネントを一時的に使用する必要がある場合などに、SCPMは有用です。元のシステム設定を簡単に復元できることはもとより、システム設定の変更を再現することも可能です。SCPMでは、システム設定のどの部分でもカスタマイズしたプロファイルとして保存できます。

SCPMが主に利用されるのは、ラップトップのネットワーク設定です。多くの場合、ネットワーク設定が違えば、電子メールやプロキシなど、他のサービスの設定も変更が必要になります。他の要素も同様です。自宅と会社で異なるプリンタを使用する、会議のマルチメディアプロジェクタ用にカスタマイズされたXサーバ設定を行う、外出先で特別な電源管理を適用する、外国支店で異なるタイムゾーンを使用するなど、さまざまな要因が考えられます。

## 15.1 用語

ここでは、SCPM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。

- *システム設定*という用語は、コンピュータの設定全体を指します。たとえば、ハードディスクパーティションの使用、ネットワーク設定、タイムゾーンの選択、およびキーボードマッピングなどの基礎的な設定をすべて含みます。
- *プロファイル*は、*設定プロファイル*とも呼ばれ、保存されていつでも復元可能な状態を指します。
- *有効なプロファイル*とは、最後に選択したプロファイルです。設定はいつでも変更できるので、現在のシステム設定が有効なプロファイルと同じだとは限りません。
- SCPMでいう*リソース*とは、システム設定に影響する要素を指します。これは、ファイルまたはソフトリンク(ユーザ、許可、またはアクセス時間などのメタデータを含む)です。また、このプロファイルでは動作するが、他のプロファイルでは動作しないシステムサービスも含まれます。
- すべてのリソースは、特定の*リソースグループ*に属します。リソースグループは論理的に共通なリソースで構成されます。ほとんどのグループには、サービスとその設定ファイルの両方が含まれています。SCPMに

よって管理されるリソースは、対象のサービスの設定ファイルに関する知識を必要としないため、ごく簡単に組み合わせることができます。SCPMには事前にリソースグループが設定されており、ほとんどの場合はそれらで対応できます。

## 15.2 コマンドラインを使用したSCPMの設定

このセクションではコマンドラインを使用してSCPMを設定する方法を説明します。まず初めに開始方法を説明し、続いて設定方法、そしてプロファイルの利用方法について説明します。

### 15.2.1 SCPMの起動とリソースグループの定義

SCPMは使用する前に有効にする必要があります。SCPMを有効にするには、コマンド`scpm enable`を使用します。SCPMを初めて実行すると初期化が行われますが、これには数秒かかります。SCPMはいつでも`scpm disable`で無効にできます。これにより誤ってプロファイルが切り替わらないようにすることができます。その後、再度有効にするには、初期化を再開するだけです。

デフォルトでは、SCPMはネットワークやプリンタだけでなく、X.Orgの設定も処理します。特別なサービスや設定ファイルを管理するには、それぞれのリソースグループを有効にします。事前定義のリソースグループを一覧表示するには、`scpm list_groups`を使用します。既に有効なグループのみを表示するには、`scpm list_groups -a`を使用します。これらのコマンドは、コマンドラインで`root`として実行します。

```
scpm list_groups -a

nis                 Network Information Service client
mail                Mail subsystem
ntpd                Network Time Protocol daemon
xf86                X Server settings
autofs              Automounter service
network             Basic network settings
printer             Printer settings
```

グループを有効または無効にするには、それぞれscpm activate_group NAMEとscpm deactivate_group NAMEを使用します。ここでNAMEは、対象のグループ名に置き換えて使用してください。

### 15.2.2 プロファイルの作成と管理

SCPMを有効にすると、defaultという名前のプロファイルが自動的に作成されます。利用可能なプロファイルを一覧表示するには、scpm listを実行します。この既存プロファイルが有効なプロファイルでもあります。コマンドscpm activeで確認できます。プロファイルdefaultは基本設定であり、これを元に他のプロファイルを作成することが可能です。このため、すべてのプロファイルで同一になる設定をあらかじめすべて設定してしまいます。続いてこれらの変更内容をコマンドscpm reloadを使用して、有効なプロファイルに保存します。次回からはdefaultプロファイルを新規プロファイルのベースとしてコピーし、名前を変更して使用できます。

新しいプロファイルを追加する方法は2つあります。プロファイルdefaultをベースにして、新規プロファイル(ここではworkとします)を作成する場合は、コマンドscpm copy default workを実行します。コマンドscpm switch workを実行すると、新しいプロファイルに切り替わり、変更が可能になります。特別な用途に合わせてシステム設定を変更し、変更内容を新しいプロファイルに保存できます。この場合、コマンドscpm add workを実行すると、新しいプロファイルが作成されるとともに、作成されたプロファイルworkに現在のシステム設定が保存され、有効を示すマークが付きます。scpm reloadを実行すると、変更内容がプロファイルworkに保存されます。

プロファイルの名前の変更または削除には、それぞれコマンドscpm rename x yおよびscpm delete zを使用します。たとえば、workという名前をprojectに変更するには、scpm rename work projectと入力します。projectを削除するには、コマンドscpm delete projectを入力します。有効なプロファイルは削除できません。

### 15.2.3 設定プロファイルの切り替え

コマンドscpm switch workを実行すると、別のプロファイル(ここでは、プロファイルwork)に切り替えることができます。有効なプロファイルに切り替えると、変更したシステム設定が組み込まれます。これは、コマンドscpm reloadに対応します。

プロファイルを切り替えると、まずSCPMは有効なプロファイルのリソースが変更されているかを確認します。次に、各リソースの変更内容を有効なプロファイルに追加するか、削除するかを確認するメッセージが表示されます。リソースを別に一覧表示する場合(以前のバージョンのSCPMのように)は、スイッチコマンドで-rパラメータを使用して、`scpm switch -r work`のように指定してください。

```
scpm switch -r work
```

変更したリソースの確認、
開始/シャットダウンするリソースの確認、
依存関係の確認、
デフォルトプロファイルの復元

次にSCPMは、現在のシステム設定と切り替えた後のプロファイルを比較します。この段階で、SCPMは相互依存への対応や設定変更の反映のために停止または再起動が必要なシステムサービスを評価します。これは、部分的なシステムリブートのようなもので、システムのごく一部だけがリブートし、残りの部分は変更なく動作し続けます。システムサービスが停止し、変更されたすべてのリソース(たとえば、設定ファイル)が書き込まれ、システムサービスが再起動されるのは、この時点のみです。

### 15.2.4 詳細なプロファイル設定

すべてのプロファイルには説明を入力できます。説明は`scpmlist`を使って表示できます。有効なプロファイルに説明を入力するには、`scpm set description "text"`を実行します。有効なプロファイル以外のプロファイル名を使用するには、たとえば、`scpm set description "text" work`のように指定します。場合によっては、プロファイルの切り替え時に、SCPMが提供する以外のアクションを実行することがあります。各プロファイルには、最高4つの実行可能ファイルを添付することができます。これらはそれぞれ、切り替え処理において起動する段階が異なります。これら4つの段階を次に示します。

**prestop** 切り替え前、サービス停止前

**poststop** 切り替え前、サービス停止後

**prestart** 切り替え後、サービス停止前

**poststart** 切り替え後、サービス停止後

これらのアクションを挿入するには、コマンドsetを、scpm set prestop filename、scpm set poststop filename、scpm set prestart filename、scpm set poststart filenameのように使用します。スクリプトが実行可能ファイルであることと、正しいインタプリタを参照することが必要です。

**Warning**

**カスタムスクリプトの統合**

SCPMにより実行されるその他のスクリプトを、スーパーユーザ(root)が読み取って実行できるようにする必要があります。他のユーザは誰もこれらのファイルにアクセスできないようにします。コマンドchmod 700 filenameおよびchown root:root filenameを入力して、rootにファイルへの排他アクセス権を付与します。

**Warning**

setコマンドで追加した設定を表示するには、コマンドgetを使用します。たとえば、コマンドscpm get poststartを実行すると、poststartの名前が返されるか、何も添付していない場合は何も返されません。このような設定は、""を使って上書きできます。scpm set prestop ""を実行すると、添付したprestopプログラムが削除されます。

すべてのsetコマンドとgetコマンドは、コメントを追加する場合と同じように任意のプロファイルに適用できます。たとえば、scpm get prestop filename workまたはscpm get prestop workとなります。

## 15.3 YaSTプロファイル管理

YaSTコントロールセンターからYaSTプロファイル管理を開始します('システム' → 'プロファイル管理')。または開始時に［'SCPMのオプション'］ダイアログで［'作動'］を選択することでSCPMを明示的に有効にします。図 15.1.「YaSTSCPMオプション」を参照してください。［'設定'］では、ご使用のSCPMで進展ポップアップを自動的に閉じるか、進展に関するメッセージに詳しいメッセージを指定するかなどを決定します。［'切替えモード'では、プロファイルが切り替えられたとき、有効なプロファイルの変更されたリソースを保存するか、破棄するかを指定します。［'切替えモード'］が［'通常'］に設定されている場合、有効なプロファイルで行われたすべての変更は、プロファ

イルが切り替えられる時点で保存されます。ブート時のSCPMの動作を定義するには［'ブートモード'］を［'変更を保存する'］(デフォルト設定)または［'変更を無視する'］に設定します。

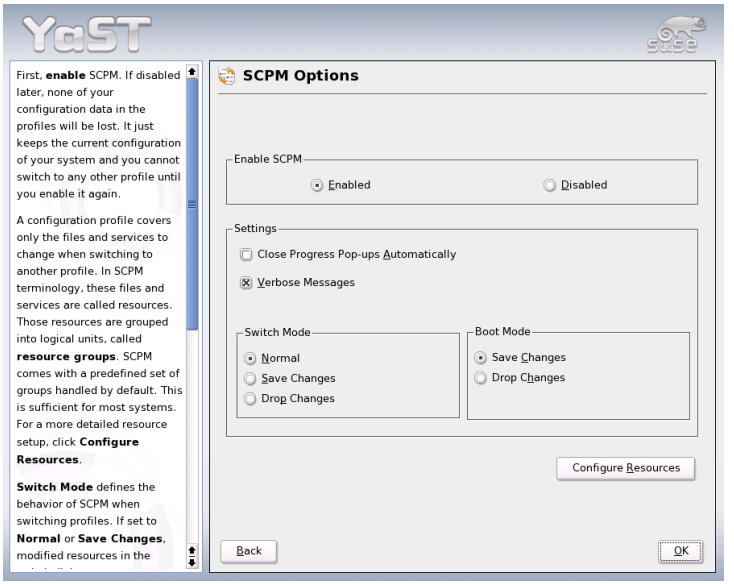


*Figure 15.1: YaSTSCPMオプション*

## 15.3.1 リソースグループの設定

現在のリソース設定を変更するには［'SCPMのオプション'］ダイアログで［'リソースを設定する'］オプションを選択します。その後のダイアログ、［'リソースグループの設定'］(図 15.2. 「リソースグループの設定」を参照)には、システムで使用可能なすべてのリソースグループが一覧表示されます。リソースグループを追加または編集するには、［'リソースグループ'］および［'記述'］を指定、または編集します。LDAPサービスの場合は、たとえば［'リソースグループ'］にldap、さらに［'記述'］にはLDAPクライアントサービスと入力します。続いて適切なリソース(サービス、設定ファイル、または両方)を入力するか、既存の設定を変更します。次に使用されていないリソースグループを削除します。選択したリソースの状態をリセットするには、つま

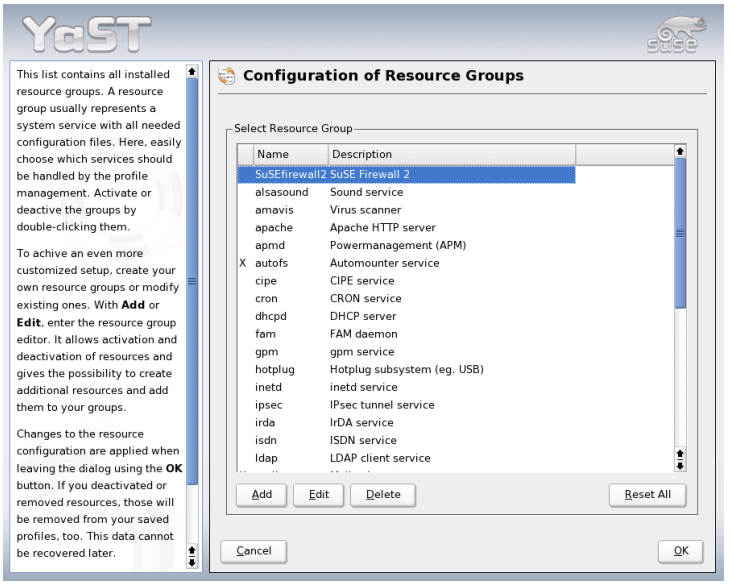


*Figure 15.2: リソースグループの設定*

りこれまでの変更内容をすべて破棄し、初期の設定値に戻すには、［'グループをリセットする'］を選択します。変更は有効なプロファイルに保存されています。

### 15.3.2 新規プロファイルの作成

新規プロファイルを作成するには開始ダイアログ(［'システム設定のプロファイル管理'］)で［'追加'］をクリックします。ウィンドウが開きます。ここでは新規プロファイルで、現在のシステム設定(SCPMが自動的に現在の設定を取得し、プロファイルが入力されます)をベースにするか、または既存のプロファイルを使用するかを選択します。新規プロファイルのベースとして現在のシステム設定を使用する場合、新規プロファイルを新規の有効なプロファイルに指定できます。この方法では旧プロファイルに対する変更は行われず、サービスを開始したり停止したりすることはありません。

続くダイアログで新規プロファイルの名前と簡潔な記述を入力します。SCPMでプロファイルを切り替えるために特別なスクリプトを実行す

るには、各実行可能ファイルのパスを入力します(図 15.3. 「特別なプロファイルの設定」を参照してください)。詳細については、項15.2.4. 「詳細なプロファイル設定」を参照してください。SCPMは新規プロファイルのリソースチェックを実行します。テストが正常に終了すると、新規プロファイルが使用可能になります。

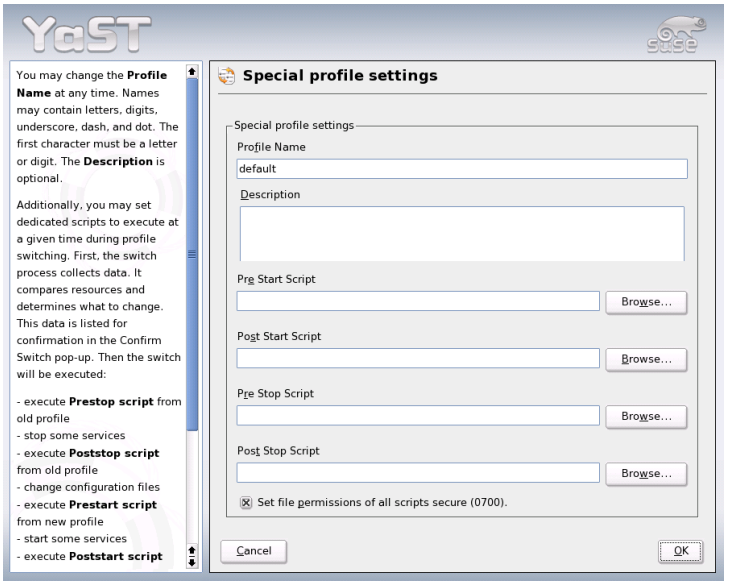


*Figure 15.3: 特別なプロファイルの設定*

### 15.3.3 既存のプロファイルの設定

既存のプロファイルを変更するには開始ダイアログ( [‘システム設定のプロファイル管理’] )で [‘編集’] をクリックします。次に必要に応じて名前、記述、スクリプト、およびリソースを編集します。

### 15.3.4 プロファイルの切り替え

プロファイルを切り替えるには、プロファイルマネージャを開きます。有効なプロファイルは矢印でマークされています。切り替えるプロファイルを選択

し、［'切替え'］をクリックします。SCPMは新規または変更されたリソースを確認し、必要に応じてそれらを追加します。

リソースが変更されると、YaSTにより［'切替えを確認する'］ダイアログが表示されます。［'有効なプロファイル中の変更されたリソースグループ'］は、変更されただけでまだ有効なプロファイルに保存されていないリソースグループを一覧表示します。現在選択されているリソースグループの［'保存または無視する'］では、このリソースグループへの変更を有効なプロファイルに保存するか、または破棄するかを指定します。または、各リソースを選択して［'詳細'］をクリックし、変更点を詳細に確認します。これにより編集されたリソースグループに含まれる設定ファイル、または実行可能ファイルがすべて一覧表示されます。新旧のバージョンを1行ずつ比較するには［'変更を表示する'］をクリックします。変更の確認が終了したら、［'アクション'］で変更へのアクションを指定します。アクションには以下のような項目があります。

**リソースを保存する** このリソースを有効なプロファイルに保存しますが、他のすべてのプロファイルへの変更は行いません。

**リソースを無視する** 有効なリソースへの変更を行いません。この変更は破棄されます。

**すべてのプロファイルに保存する** このリソースの設定全体をすべてのプロファイルにコピーします。

**全てのプロファイルにパッチを当てる**
すべてのプロファイルに最新の変更のみを適用します。

［'全てを保存または無視する'］はそのまま保存するか、またはこのダイアログに表示されているすべてのリソースの変更を破棄します。

有効なプロファイルの変更を確認した後、［'了解'］をクリックして［'切替えを確認する'］ダイアログを終了します。これでSCPMが新規プロファイルに切り替わります。切り替えている間、旧プロファイルに対しprestopおよびpoststopスクリプトを実行し、新規プロファイルにprestartおよびpoststartスクリプトを実行します。

## 15.4 トラブルシューティング

このセクションではSCPMでよく起こる問題について説明します。どのようにして問題が起こるか、そしてどのようにこれらの問題を解決するかについても説明します。

### 15.4.1 切り替えプロセス中の終了

SCPMが切り替えプロシージャ中に動作を停止することがあります。ユーザによる中止操作や電源障害、外部要因によって起こることもありますし、SCPM自体のエラーによることもあります。このような場合、次回SCPMを起動すると、SCPMがロックされているというエラーメッセージが表示されます。これはシステムの安全を確保するための措置です。なぜならデータベースに格納されているデータと、システムの状態とが食い違うことがあるからです。この問題を解決するには、`scpm recover`を実行します。SCPMが前回実行できなかったすべての操作を実行します。また、`scpm recover -b`を実行することもできます。これは前回実行時に実行された操作のアンドゥを試みます。YaSTプロファイルマネージャを使用している場合は、開始時に修復ダイアログが表示されます。このダイアログでは前述のコマンドを実行できます。

### 15.4.2 リソースグループ設定の変更

SCPMの初期化が完了した後にリソースグループの設定を変更するには、グループを追加または削除してから`scpm rebuild`と入力します。これにより、すべてのプロファイルに新規リソースが追加され、削除したリソースは完全に削除されます。削除したリソースの設定が各種プロファイル内で異なっている場合、この設定データは失われます。ただし、システム内の最新バージョンはそのまま残ります。YaSTで設定を変更する場合、rebuildコマンドを入力する必要はありません。これはYaSTにより処理されます。

## 15.5 システムブート時のプロファイル選択

システムのブート時にプロファイルを選択するには、ブート画面上で(F4)を押して、使用可能なプロファイルのリストを表示します。矢印キーを使用してプロファイルを選択し、(Enter)キーで決定します。これで選択したプロファイルがブートオプションとして使用されます。

## 15.6 関連資料

最新ドキュメントがSCPMのinfoページに用意されており、KonquerorやEmacs (`konqueror info:scpm`)などのツールで表示できます。コンソール

で、infoまたはpinfoと入力します。開発者向けの詳細については、/usr/share/doc/packages/scpmを参照してください。

# 電 源 管 理

ここでは、Linuxのさまざまな電源管理テクノロジの概要について説明します。Linuxで使用されるすべてのAPM (advanced power management)、ACPI (advanced configuration and power interface)、およびCPU周波数調節の設定について詳細に説明します。

従来、電源管理用としてラップトップのみで使用されてきたAPMとは異なり、ハードウェアの情報および設定を管理するツールであるACPIは、近年、ラップトップ、デスクトップ、サーバなど、あらゆるコンピュータ上で利用されています。近年のコンピュータの多くは、CPU周波数を状況に応じて調節できるため、特にモバイルデバイスで貴重なバッテリ時間を節約できます(*CPU周波数調節*)。

どのような電源管理テクノロジでも、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。APMは、従来型のコンピュータで多く使われてきました。APMは、ほとんどがBIOSに実装された関数セットからなるため、APMサポートのレベルはハードウェアによって異なります。より複雑なACPIでは、この傾向がさらに強まります。このため、どちらか片方を推奨することは無理です。さまざまな手順をハードウェア上でテストし、最も適切なサポートが実現できるテクノロジを選択してください。

**Important**

**AMD64プロセッサの電源管理**

64ビットカーネルを搭載したAMD64プロセッサは、ACPIのみをサポートしています。

**Important**

# 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。ここでは、主な機能と電源管理システムAPMおよびACPIでの使用の概要について説明します。

**スタンバイ** この動作モードは、ディスプレイの電源をオフにします。プロセッサのパフォーマンスを低下させるコンピュータも一部あります。この機能は、すべてのAPMの実装で利用可能とは限りません。この機能は、ACPI状態S1またはS2に対応します。

**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。通常、APMを

使用するデバイスは、ふたを閉じればサスペンドし、開ければ再開します。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。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APMおよびACPIから独立したカーネルルーチンにより実行されます。

**バッテリモニタ** ACPIとAPMは、バッテリをチェックして、充電ステータスに関する情報を提供します。また、どちらのシステムも、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。

**システムコンポーネントのシャットダウン**
システム全体を考えた場合、電力消費量を抑えるという点では、ハードディスクをオフにすることが最も重要です。システム全体の信頼性に応じて、しばらくハードディスクをスリープ状態にすることは可能です。ただし、スリープ時間が長くなれば、データ損失のリスクも高くなります。他のコンポーネントは、(少なくとも理論的には)ACPIによって無効にでき、またBIOSセットアップによって永久に無効にすることもできます。

**プロセッサ速度の制御** CPUに関連する省エネルギー方法は次の3つです。周波数調節と電圧調節(PowerNow!またはSpeedstep)、スロットリング、およびプロセッサのスリープ状態(C状態)への切り替えです。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 APM

省電力機能の中には、APM BIOS自体によって実行される機能もあります。多くのラップトップにおいて、スタンバイ状態とサスペンド状態は、特別なオペレーティングシステムの機能を使用するのではなく、キーの組み合わせによっ

て、またはふたを閉じることによって有効になります。しかし、コマンドを使用してこれらのモードを有効にするには、システムがサスペンドする前に、特定のアクションがトリガされなければなりません。さらに、バッテリの充電レベルを表示するには、特殊なプログラムパッケージと適切なカーネルが必要になります。

SUSE LINUXのカーネルでは、ビルトインのAPMをサポートしています。しかし、APMが有効になるのは、ACPIがBIOSに実装されておらず、APM BIOSが検出された場合に限られます。APMサポートを有効にするには、ブートプロンプトで`acpi=off`を実行してACPIを無効にする必要があります。APMが有効かどうかを確認するには、`cat /proc/apm`を入力します。ここでさまざまな値が出力されれば、すべて正常であることを意味します。ここで、コマンド`shutdown -h`を実行して、コンピュータをシャットダウンします。

BIOS実装が規格に完全に準拠していないと、APMに問題が発生することがあります。一部の問題は、特殊なブートパラメータで回避できます。すべてのパラメータは、ブートプロンプトで、`apm=parameter`の形式で入力します。

**onまたはoff** APMサポートの有効化または無効化

**(no-)allow-ints** BIOS機能の実行中の中断を許可します。

**(no-)broken-psr** BIOSの"GetPowerStatus"機能が正しく動作しません。

**(no-)realmode-power-off** シャットダウンの前にプロセッサをリアルモードにリセットします。

**(no-)debug** APMイベントをシステムログに記録します。

**(no-)realmode-power-off** シャットダウンの後、システムの電源を切断します。

**bounce-interval=⟨*n*⟩** サスペンドイベントの後、別のサスペンドイベントが無視される時間を1/100秒単位で表した数値です。

**idle-threshold=⟨*n*⟩** システムのアイドル状態がこの値に達するとBIOSの`idle`関数が実行されます(0=常時、100=実行しない)。

**idle-period=⟨*n*⟩** システムアクティビティを測定した後の時間を1/100秒単位で表した数値。

APMデーモン(apmd)は廃止になりました。その機能はACPIおよびCPUの周波数調節もサポートする新しいpowersavedで処理されます。

## 16.3 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、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、項16.3.4. 「トラブルシューティング」を参照してください。

### 16.3.1 動作中のACPI

システムのブート時に、カーネルがACPI BIOSを検出する場合、ACPIが自動的に有効になり、APMが無効になります。旧式のコンピュータでは、ブートパラメータ`acpi=on`を指定しなければならない場合があります。コンピュータは、ACPI 2.0以降をサポートする必要があります。`/var/log/boot.msg`のカーネルブートメッセージで、ACPIが有効にされていることを確認します。

続いて、複数のモジュールがロードされます。これは、ACPIデーモンの起動スクリプトによって行われます。これらのモジュールのいずれかに問題が発生すると、対応するモジュールが`/etc/sysconfig/powersave/common`でのロードまたはアンロードから除外されます。 システムログ(`/var/log/messages`)には、モジュールのメッセージが入っているので、どのコンポーネントが検出されたことがわかります。

`/proc/acpi`には、システム状態に関する情報を提供するファイルや、状態を変更するために使用できるファイルが多数含まれています。一部の機能はまだ開発中であるため動作しません。また、一部の機能はメーカの実装状況に大きく依存するためサポートされていない場合もあります。

すべてのファイル(`dsdt`および`fadt`)は、コマンド`cat`で読み取ることができます。一部のファイルでは、`echo`で設定を変更できます。たとえば、`echo X > file`でXに適した値を指定します。この情報と制御オプションにアクセスするには、常にコマンド`powersave`を使用します。以下で最も重要なファイルについて説明します。

**/proc/acpi/info** ACPIに関する一般的な情報

**/proc/acpi/alarm** システムがいつスリープ状態から回復するかを指定します。現在、この機能は完全にはサポートされていません。

**/proc/acpi/sleep** さまざまなスリープ状態に関する情報を提供します。

**/proc/acpi/event** すべてのイベントがここにレポートされ、Powersaveデーモン(powersaved)で処理されます。(Power)ボタンの押下げ、またはふたを閉じるなど、いずれのデーモンもこのファイルにアクセスしないイベントは、cat /proc/acpi/eventによって読み取ることができます((Ctrl)-(C)で終了します)。

**/proc/acpi/dsdtおよび/proc/acpi/fadt**
これらのファイルにはACPIテーブルのDSDT (*differentiated system description table*)とFADT (*fixed ACPI description table*)が含まれています。これらは、acpidmp、acpidisasm、およびdmdecodeで読み取ることができます。これらのプログラムとマニュアルは、パッケージpmtoolsにあります。たとえば、acpidmp DSDT | acpidisasmなどです。

**/proc/acpi/ac_adapter/AC/state**
ACアダプタが接続されているかを示します。

**/proc/acpi/battery/BAT*/{alarm,info,state}**
バッテリ状態についての詳細情報です。充電レベルを読み取るには、infoのlast full capacityとstateのremaining capacityを比較します。これをもっと円滑に行うには、項16.3.3.「ACPIツール」で説明する特別なプログラムの1つを使用します。バッテリイベントがトリガされる充電レベルは、alarmで指定できます。

**/proc/acpi/info** このディレクトリには、さまざまなスイッチについての情報が入っています。

**/proc/acpi/fan/FAN/state** ファンが現在、作動しているかを示します。ファンは、このファイルに0 (オン)か3 (オフ)かを書き込むことによって、オンまたはオフにできます。ただし熱が上がりすぎた場合は、カーネルとハードウェア(またはBIOS)の両方のACPIコードによってこの設定が上書きされます。

**/proc/acpi/processor/CPU*/info**
プロセッサの省エネオプションに関する情報。

**/proc/acpi/processor/CPU*/power**
現在のプロセッサ状態に関する情報。C2の横にアスタリスクが付いている場合、プロセッサがアイドル状態です。usageに示すように、これが最もよくある状態です。

**/proc/acpi/processor/CPU*/throttling**
プロセッサクロックの減速の設定に使用できます。通常、スロットリングは8つのレベルで使用できます。これは、CPUの周波数制御に依存しません。

**/proc/acpi/processor/CPU*/limit**
パフォーマンス(廃止)とスロットリングがデーモンによって自動的に制御される場合、上限をここで指定できます。上限の一部はシステムによって定義されますが、ユーザが調整できる上限もあります。

**/proc/acpi/thermal_zone/** すべてのサーマルゾーンに対し、個別の下位ゾーンが存在します。サーマルゾーンとは、よく似たサーマルプロパティを持ち、ハードウェアメーカによって番号と名前が指定された領域です。しかし、ACPIが持つ可能性の多くは、ほとんど実装されていません。そして、温度制御は相変わらずBIOSによって管理されています。オペレーティングシステムが介入すると、ハードウェアの寿命が短くなるので、オペレーティングシステムが介入する機会はあまりありません。したがって、次の説明には、理論上だけの部分があります。

**/proc/acpi/thermal_zone/*/temperature**
サーマルゾーンの現在の温度です。

**/proc/acpi/thermal_zone/*/state**
この状態は、すべてがokなのか、またはACPIがアクティブまたはパッシブ冷却を適用しているかを示します。ACPI独立のファン制御の場合、この状態は常にokです。

**/proc/acpi/thermal_zone/*/cooling_mode**
ACPIで制御される冷却化方式を選択します。パッシブ(パフォーマンスは低いが経済的)またはアクティブ(フルパフォーマンス、ファンノイズ)のどちらかを選択できます。

**/proc/acpi/thermal_zone/*/trip_points**
パッシブ/アクティブ冷却、サスペンション(hot)、またはシャットダウン(critical)など、特定のアクションをトリガする温度を設定します。可能なアクションは、DSDT(デバイス依存)内で定義されます。ACPI仕様で定義されているトリップポイント

は、critical、hot、passive、active1、およびactive2です。実装されていないトリップポイントがあっても、このファイルにはすべてを常にこの順序で入力する必要があります。たとえば、エントリecho 90:0:70:0:0 > trip_pointsは、criticalの温度を90、passiveの温度を70に設定します(温度はすべて摂氏)。

**/proc/acpi/thermal_zone/*/polling_frequency**
温度が変化してもtemperatureの値が自動的に更新されない場合は、ポーリングモードをここでオンにします。コマンドechoX > /proc/acpi/thermal_zone/*/polling_frequencyを使用すると、X秒ごとに温度の問い合わせが行われます。ポーリングを無効にするには、X=0に設定します。

これらの設定、情報、イベントは、いずれも手動で編集する必要はありません。編集はPowersaveデーモン(powersaved)および各種アプリケーション(powersave、kpowersave、wmpowersaveなど)で実行できます。項16.3.3.「ACPIツール」を参照してください。powersavedには古いacpidの機能が含まれているため、acpidは不要です。

### 16.3.2 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の名称です。ただし、このテクノロジは他のメーカ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。このテクノロジはAPMにもACPIにも依存せず、周波数と所要電流をパフォーマンスに合わせて調整するデーモンを必要とします。設定は、ディレクトリ/sys/devices/system/cpu/cpu*/cpufreq/内で行うことができます。

**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番目ごとのインパルスが省略され、87.5%の場合は8番目ごとのインパルスだけがプロセッサに到達し

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、C3という3つの状態があります。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、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。

SUSE LINUXでは、これらのテクノロジはPowersaveデーモンで制御されます。この設定については、項16.5. 「powersaveパッケージ」を参照してください。

### 16.3.3 ACPIツール

総合的と呼べるACPIユーティリティには、バッテリ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、wmacpimon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pmtools)などが含まれています。

### 16.3.4 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、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=oldboot** 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。

**Warning**

**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。

**Warning**

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テーブルの解析中に問題が発生した場合は、最も重要なテーブル(DSDT)を改良版に置き換えます。この場合、BIOSで障害のあるDSDTが無視されます。処理手順については、項16.5.4.「トラブルシューティング」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キスパ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に

は、問題を通知する必要があります。なぜなら、メーカは、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。

**関連資料**

ACPIに関する補足資料とヘルプ

- `http://www.cpqlinux.com/acpi-howto.html`(詳細なACPI HOWTO、DSDTパッチが含まれています)
- `http://www.intel.com/technology/iapc/acpi/faq.htm`(IntelのACPIに関するFAQ)
- `http://acpi.sourceforge.net/`(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)

## 16.4 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。ほとんどの機能はpowersavedで制御できます。

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。オプション`-y`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。またオプション`-Y` (caution)はハードディスクをスリープにします。コマンド`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、次のように使用します。`0`にすると、このメカニズムが無効になり、ハードディスクがずっと回り続けます。`1`から`240`までの値を指定すると、指定した値x5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。まず、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、カーネル更新デーモン(kupdated)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、kupdatedの間隔が短く設定されて、完全性を最大まで高めます。バッファが5秒毎にチェックされ、データが30秒以上経過していたり、バッファの使用レベルが30%に達すると、bdflushデーモンに通知されます。するとbdflushデーモンが、データをハードディスクに書き込みます。このデーモンはまた、たとえば、バッファが一杯のときに、kupdatedと無関係に書き込みます。

**Warning**

**データの完全性に関する障害**

カーネル更新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。

**Warning**

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bdflush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。しかしながら、kupdatedの間隔が広くなると、このことは重要でなくなります。

## 16.5 powersaveパッケージ

powersaveパッケージは、ラップトップでバッテリ使用時に省電力機能をサポートします。この機能の一部はサスペンド、スタンバイ、ACPIボタン機能、およびIDEハードディスクをスリープ状態に切り替える場合など、通常のワークステーションやサーバでも有用です。

このパッケージにはご使用のコンピュータの電源管理機能がすべて含まれます。電源管理機能はACPI、APM、IDEハードディスク、PowerNow!またはSpeedStepテクノロジを使用してハードウェアをサポートします。apmd、acpid、ospmd、cpufreqd (現在はcpuspeed)などの各パッケージの機能がpowersaveパッケージに統合されています。これらのパッケージのデーモンをpowersaveデーモンと同時に実行することは避けてください。

ご使用のシステムに前述のハードウェア要素の一部が含まれていないとしても、省電力機能の制御にはpowersaveデーモンを使用してください。ACPIおよびAPMは相互排他的であるため、ご使用のシステムではこれらのシステムのどちらか一方しか使用できません。このデーモンはハードウェア構成に変更があった場合、これを自動的に検出します。

**Important**

**powersaveに関する情報**

powersaveパッケージは/usr/share/doc/packages/powersaveでも利用できます。

**Important**

### 16.5.1 powersaveパッケージの設定

通常、powersaveの設定は複数のファイルに分散されています。

**/etc/sysconfig/powersave/common**
このファイルにはpowersaveデーモンの一般的な設定が含まれます。たとえば、エラーメッセージの量(/var/log/messages内の)は、変数POWERSAVE_DEBUGの値を増加させることで増やせます。

**/etc/sysconfig/powersave/events**
powersaveデーモンはシステムイベントを処理するためにこのファイルを必要とします。1つのイベントには外部アクションまたはデーモン自体が実行したアクションを割り当てることができます。外部アクションの

場合、デーモンは/usr/lib/powersave/scripts/にある外部ファイルを実行しようとします。事前定義された内部アクションは次のとおりです。

- ignore
- throttle
- dethrottle
- suspend_to_disk
- suspend_to_ram
- standby
- do_suspend_to_disk
- do_suspend_to_ram
- do_standby

throttleはPOWERSAVE_MAX_THROTTLINGに定義された値に従ってプロセッサを減速させます。この値は現在のスキーマに依存します。dethrottleはプロセッサをフルパフォーマンスに設定します。suspend_to_disk、suspend_to_ram、およびstandbyはスリープモード用のシステムイベントを生成します。これら3つのアクションは一般的にスリープモードのトリガとなりますが、常に、これらを特定のシステムイベントと関連付けるようにしてください。

ディレクトリ/usr/lib/powersave/scriptsにはイベントを処理するための各種スクリプトが含まれます。

**notify** コンソール、X Window、またはアコースティック(音による)シグナルのイベントに関する通知です。

**screen_saver** スクリーンセーバを作動させます。

**switch_vt** サスペンドやスタンバイの後に画面が戻らない場合に有用です。

**wm_logout** 設定を保存し、GNOME、KDE、または他のウィンドウマネージャからログアウトします。

**wm_shutdown** GNOMEまたはKDEの設定を保存し、システムをシャットダウンします。

たとえば、変数POWERSAVE_EVENT_GLOBAL_-
SUSPEND2DISK="prepare_suspend_to_disk do_suspend_-
to_disk"が設定されている場合、ユーザがスリープモード用

のコマンドであるsuspend to diskをpowersavedに対して発行するとすぐに、2つのスクリプトまたはアクションが指定された順番で処理されます。デーモンは外部スクリプトである/usr/lib/powersave/scripts/prepare_suspend_to_diskを実行します。このスクリプトが正常に実行されると、重要なモジュールがスクリプトによりアンロードされ、各種サービスが停止された後、デーモンが内部アクションであるdo_suspend_to_diskを実行し、コンピュータをスリープモードにします。

sleepボタンのイベントを処理するアクションはPOWERSAVE_EVENT_BUTTON_SLEEP="notify suspend_to_disk"のように変更することができます。この場合、外部スクリプトnotifyがユーザにサスペンドを通知します。続いて、イベントPOWERSAVE_EVENT_GLOBAL_SUSPEND2DISKが生成されます。これにより前述のアクションが実行され、システムサスペンドモードに入ります。スクリプトnotifyは/etc/sysconfig/powersave/commonにある変数POWERSAVE_NOTIFY_METHODを使用してカスタマイズできます。

**/etc/sysconfig/powersave/cpufreq**
動的CPU周波数の設定を最適化するための変数が含まれます。

**/etc/sysconfig/powersave/battery**
バッテリの制限とその他のバッテリ固有設定が含まれます。

**/etc/sysconfig/powersave/sleep**
このファイルでは、スリープモードを有効にし、サスペンドイベントまたはスタンバイイベントの前にアンロードすべき重要なモジュールと、停止すべき各種サービスを指定します。システムが再開されるとこれらのモジュールは再ロードされ、各種サービスも再開されます。またトリガされたスリープモードを遅らせる(ファイルを保存するために)ことも可能です。デフォルト設定は主にUSBおよびPCMCIAモジュールに関係しています。サスペンドまたはスタンバイの障害は通常、ある一定のモジュールによって発生します。エラーの特定の詳細については項16.5.4.「トラブルシューティング」を参照してください。

**/etc/sysconfig/powersave/thermal**
冷却コントロールおよびサーマルコントロールを有効にします。このテーマの詳細については、ファイル/usr/share/doc/packages/powersave/README.thermalを参照してください。

**/etc/sysconfig/powersave/scheme_***
これらは特定の導入シナリオに応じて消費電力を最適化するさまざまな

スキーマです。多くのスキーマが事前に設定され、そのまま使用できます。また、カスタムスクリプトをここに保存することもできます。

### 16.5.2 APMおよびACPIの設定

**サスペンドおよびスタンバイ**

デフォルトではスリープモードは無効になっています。これはスリープモードが一部のコンピュータではまだ機能しないためです。次に示すように、3種類の基本ACPIスリープモードおよび2種類のAPMスリープモードがあります。

**サスペンド(ディスク)(ACPI S4、APMサスペンド)**
メモリの内容全部をハードディスクに保存します。コンピュータは完全に電源オフの状態になり、電力は消費されません。

**サスペンド(RAM)(ACPI S3、APMサスペンド)**
デバイス全体の状態をメインメモリに保存します。メインメモリ以外からの電力消費はありません。

**スタンバイ(ACPI S1、APMスタンバイ)**
一部のデバイスの電源をオフにします(メーカにより異なる)。

サスペンド、スタンバイ、再開を正しく処理するため、ファイル/etc/sysconfig/powersave/eventsで次のデフォルトオプションが設定されていることを確認してください(SUSE LINUXインストール時のデフォルト設定)。

```
POWERSAVE_EVENT_GLOBAL_SUSPEND2DISK=
    "prepare_suspend_to_disk do_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_SUSPEND2RAM=
    "prepare_suspend_to_ram do_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_STANDBY=
    "prepare_standby do_standby"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2DISK=
    "restore_after_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2RAM=
    "restore_after_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_RESUME_STANDBY=
    "restore_after_standby"
```

### バッテリ状態のカスタマイズ

ファイル/etc/sysconfig/powersave/batteryで3通りのバッテリ充電レベル(パーセント指定)を定義します。バッテリの充電量がこれらのレベルに達すると、システムアラートが生成されたり、特定のアクションが実行されたりします。

```
POWERSAVED_BATTERY_WARNING=20
POWERSAVED_BATTERY_LOW=10
POWERSAVED_BATTERY_CRITICAL=5
```

充電レベルが指定された制限値を下回った場合に実行されるアクションまたはスクリプトは、設定ファイル/etc/sysconfig/powersave/eventsに定義されています。各種ボタンの標準アクションは項16.5.1. 「powersaveパッケージの設定」に示されているように変更できます。

```
POWERSAVE_EVENT_BATTERY_NORMAL="ignore"
POWERSAVE_EVENT_BATTERY_WARNING="notify"
POWERSAVE_EVENT_BATTERY_LOW="notify"
POWERSAVE_EVENT_BATTERY_CRITICAL="wm_shutdown"
```

### さまざまな条件に応じた電力消費の最適化

システムの動作は電源のタイプによって調整することができます。システムがAC電源を使用せずバッテリで稼動している場合は、システムの電力消費を抑えねばなりません。また、システムがAC電源に接続された場合はすぐ、自動的にパフォーマンスを上げる必要があります。このように、CPUの周波数、IDEの省電力機能、他のさまざまなパラメータを変更することができます。

コンピュータがAC電源に接続されている場合、またはされていない場合に実行される各種アクションは、/etc/sysconfig/powersave/eventsで定義されています。以下で/etc/sysconfig/powersave/commonで使用するスキーマを選択します。

```
POWERSAVE_AC_SCHEME="performance"
POWERSAVE_BATTERY_SCHEME="powersave"
```

各スキーマは/etc/sysconfig/powersaveにあるファイルに保存されています。ファイル名はスキーマごとにscheme_nameという形式になっています。次に2つのスキーマを例としてあげます。scheme_performanceおよびscheme_powersaveには、performance、powersave、presentation、およびacousticが事前定義されています。YaST電源管理モジュール(項16.6. 「YaST電源管理モジュール」を参照)を使用して、既存のスキーマの編集、作成、削除、別の電源状態との関連付けを行うことができます。

### 16.5.3 その他のACPI機能

ACPIを使用している場合、*ACPI*ボタン(電源、スリープ、ラップトップを開く、ラップトップを閉じる)に対するシステム応答を制御することができます。/etc/sysconfig/powersave/eventsでアクションの実行を設定します。個別のオプションについての説明はこの設定ファイルを参照してください。

**POWERSAVE_EVENT_BUTTON_POWER="wm_shutdown"**
電源ボタンが押されると、システムは応答して該当するウィンドウマネージャ(KDE、GNOME、fvwmなど)を閉じます。

**POWERSAVE_EVENT_BUTTON_SLEEP="suspend_to_disk"**
スリープボタンが押されると、システムはsuspend-to-diskモードに設定されます。

**POWERSAVE_EVENT_BUTTON_LID_OPEN="ignore"**
ラップトップのふたが開いている状態でアクションは発生しません。

**POWERSAVE_EVENT_BUTTON_LID_CLOSED="screen_saver"**
ラップトップが閉じられると、スクリーンセーバが有効になります。

指定した時間内に、CPUの負荷が指定した制限を越えない場合、さらにCPUのパフォーマンスを低下させることも可能です。負荷制限値をPOWERSAVED_-CPU_LOW_LIMITに、タイムアウトをPOWERSAVED_CPU_IDLE_TIMEOUTに指定します。

### 16.5.4 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で最も頻繁に起こる問題について解説します。

**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、次のコマンドを使用し、dmesgの出力からACPI固有のメッセージを検索します。`dmesg|grep -i acpi`。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/tables`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 (`pmtools`パッケージ)でコンパイルする必要があります。iaslでコンパイルするには、コマンド`iasl -sa file.asl`を入力します。iasl (Intel ACPIコンパイラ)の最新バージョンは、`http://developer.intel.com/technology/iapc/acpi/downloads.htm`で入手できます。

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。

**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。特別なモジュールまたはモジュールオプションが必要な場合その設定は、ファイル`/etc/sysconfig/powersave/cpufreq`にある変数`CPUFREQD_MODULE`および`CPUFREQD_MODULE_OPTS`で行います。

**サスペンドとスタンバイが機能しません。**

ACPIシステムにはサスペンドおよびスタンバイの不具合の原因になる、カーネル関連の問題が複数報告されています。以下を参照してください。

- 現在、RAMが1GB以上のシステムでは、サスペンドはサポートされていません。
- 現在、マルチプロセッサシステムおよびP4プロセッサ(ハイパースレッディング搭載)では、サスペンドはサポートされていません。

エラーメッセージはDSDTの実装(BIOS)の不具合が原因である可能性もあります。この場合、新しいDSDTをインストールします。

ACPIおよびAPMシステムの場合: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と、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。このモジュールの判別には/var/log/sleepmodeにあるpowersaveによって生成されたログファイルが大変有用です。コンピュータがスリープモードにならない場合、その原因は最後にアンロードされたモジュールに関係しています。/etc/sysconfig/powersave/sleepにある以下の設定を変更し、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。

```
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND2DISK=""
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND2RAM=""
POWERSAVE_UNLOAD_MODULES_BEFORE_STANDBY=""
POWERSAVE_SUSPEND2DISK_RESTART_SERVICES=""
POWERSAVE_SUSPEND2RAM_RESTART_SERVICES=""
POWERSAVE_STANDBY_RESTART_SERVICES=""
```

SambaやNISといったネットワーク環境の変更時、またはリモートでマウントされたファイルシステムとの接続時にサスペンドまたはスタンバイを使用する場合、オートマウンタを使用してそれらをマウントするか、それぞれのサービスを追加するようにします。たとえば、前述の変数では、smbfsまたはnfsなどが該当します。サスペンドまたはスタンバイの前に、アプリケーションがリモートでマウントされたファイルシステムにアクセスすると、このサービスは正常に停止されません。このためファイルシステムを正常にアンマウントすることができなくなります。このようなことが原因で、システムを再開した後、ファイルシステムに障害が発生したり、再マウントが必要になったりする場合があります。

**ACPIを使用した場合、Powersaveがバッテリレベルの低下を識別しません。**

ACPIを使用する場合、オペレーティングシステムはBIOSに対し、バッテリの充電レベルが一定の基準を下回った場合にメッセージを送信するよう要求できます。バッテリを定期的に確認することはコンピュータのパフォーマンス低下につながる恐れがあります。このメソッドの利点はその必要がないことです。ただし、充電レベルが指定値を下回った場合、BIOSがこの機能をサポートしているのにもかかわらず、通知されない場合があります。この現象がご使用のシステムで発生する場合は、ファイル`/etc/sysconfig/powersave/battery`にある変数`POWERSAVED_FORCE_BATTERY_POLLING`の値を`yes`に設定し、バッテリの確認を強制的に行うようにします。

## 16.6 YaST電源管理モジュール

YaST電源管理モジュールではこれまで解説してきたすべての電源管理を設定できます。YaSTコントロールセンターから'システム' → '電源管理'でモジュールを開始すると、モジュールの最初のダイアログが開きます。このダイアログを図 16.1.「スキーマの選択」に示します。

このダイアログでバッテリ使用時およびAC電源使用時に適用するスキーマを選択します。スキーマを追加、または変更するには［'スキーマ編集'］をクリックします。これにより、既存のスキーマの概要が表示されます。図 16.2.「既存のスキーマの概要」を参照してください。

スキーマの概要で変更するスキーマを選択し、［'編集'］をクリックします。新しいスキーマを追加するには、['追加']をクリックします。どちらを使用した場合でも、図 16.3.「スキーマの追加」に示す同じダイアログが表示されます。

まず初めに、新規または編集するスキーマに適切な名前と説明を指定します。このスキーマを使用した場合、CPUパフォーマンスを制御するか、さらにどのように制御するかを決定します。また、周波数の調整およびスロットル(減速)の使用の有無、およびその使用範囲を指定します。続くダイアログはハードディスクの設定です。ここでは最大パフォーマンス使用時または省電力時の［'スタンバイポリシー'］を定義します。［'音のポリシー'］ではハードディスクのノイズレベルを制御します(ハードディスクによってはサポートされていません)。［'冷却ポリシー'］は使用する冷却メソッドを決定します。残念ながら、このタイプの温度制御をサポートしているBIOSはほとんどありません。`/usr/share/doc/packages/powersave/README.thermal`で、ファンおよびパッシブ冷却メソッドの使用方法を参照してください。［'次へ'］をク


*Figure 16.1: スキーマの選択*

リックし、次のダイアログに進みます。このダイアログでは接続されたディスプレイの省電力モードを設定します。コンピュータ未使用時のディスプレイの電力消費量を抑えるため、［'スクリーンセーバを有効化'］チェックボックスをオンにします。［'省電力ディスプレイの有効化'］ではスタンバイ、サスペンド、電源断モードに入るまでの時間を制御できます。スキーマの設定がすべて終了したら、［'了解'］をクリックして開始ダイアログに戻ります。開始ダイアログでこのカスタムスキーマを2つの動作モードのいずれかに割り当てます。設定を有効にするには、［'了解'］を押してこのダイアログを終了します。

また、最初のダイアログの［'バッテリ警告'］、［'ACPIの設定'］、または［'サスペンドを有効化'］を使用して、全体的な電源管理設定を行うことも可能です。［'バッテリ警告'］をクリックし、図 16.4.「バッテリ充電レベル」に示す、バッテリ充電レベルのダイアログを開きます。

充電レベルが一定の基準値を下回った段階で、システムのBIOSはオペレーティングシステムに通知します。このダイアログでは、［'警告容量'］、［'低容量'］、および［'致命的容量'］の3つの基準値を定義します。充電レベルが

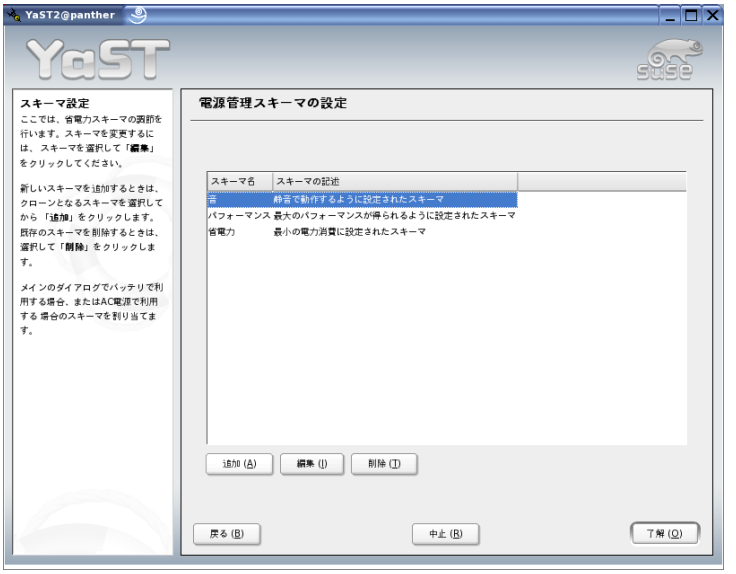


*Figure 16.2: 既存のスキーマの概要*

これらの基準値を下回ると特定のアクションがトリガされます。通常、初めの2つの状態ではユーザへの通知がトリガされるのみです。3つめの致命的なレベルではシャットダウンをトリガします。これは残りの電力ではシステムのオペレーションを維持することが困難であるためです。適切な充電レベルとそれに応じて実行するアクションを選択し、［'了解'］をクリックして開始ダイアログに戻ります。

［'ACPIの設定'］を使用してACPIボタンを設定するダイアログを開きます。このダイアログを図 16.5.「ACPIの設定」に示します。ACPIボタンの設定は特定のスイッチに対するシステムの応答を決定します。電源ボタンが押された場合、スリープボタンが押された場合、ラップトップが閉じられた場合のそれぞれに応じて、システムがどのように応答するかを設定します。［'了解'］をクリックして設定を終了し、開始ダイアログに戻ります。

［'サスペンドを有効化'］ダイアログを開きます。このダイアログではこのシステムのユーザがサスペンドまたはスタンバイ機能を使用できるか、さらにそれらをどのように使用するかを決定します。［'了解'］をクリックしてメインダイアログに戻ります。［'了解'］を再度クリックしてモジュールを終了し、


*Figure 16.3: スキーマの追加*

電源管理設定を確認してください。


*Figure 16.4: バッテリ充電レベル*


*Figure 16.5: ACPIの 設 定*

# 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、または周辺デバイスと通信するには、いくつかの方法があります。WLAN (無線LAN)は、ラップトップをネットワーク化するために使用できます。Bluetoothを使用すると、個々のシステムコンポーネント(マウス、キーボード)、周辺デバイス、携帯電話、PDA、および個々のコンピュータを互いに接続できます。PDAまたは携帯電話との通信には、IrDAがよく使用されます。この章では、3つのテクノロジとその設定のすべてを紹介します。

# 17.1 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。

*Table 17.1: 各種WLAN規格の概要*

| 名称 | 帯域(GHz) | 最大伝送速度(MBit/s) | 注 |
|---|---|---|---|
| 802.11 | 2.4 | 2 | 廃止、実質上、使用可能なエンドデバイスはなし |
| 802.11b | 2.4 | 11 | 普及 |
| 802.11a | 5 | 54 | あまり普及せず |
| 802.11g | 2.4 | 54 | 11bとの下位互換性あり |

また、最大伝送速度22MBit/sのTexas Instrumentsの802.11bバージョン(802.11b+)のような独自規格もあります。ただし、この規格を使用するカードは一般的ではありません。

## 17.1.1 ハードウェア

802.11カードは、SUSE LINUXでサポートされていません。802.11a、802.11b、および802.11gを使用するカードのほとんどは、サポートされています。通常、新しいカードは802.11g規格に準拠していますが、802.11bを使用するカードも使用可能です。一般に、次のチップを内蔵したカードがサポートされています。

- Lucent/Agere Hermes
- Intel PRO/Wireless 2100、2200BG、2915ABG
- Intersil Prism2/2.5/3

- Intersil PrismGT
- Atheros 5210、5211、5212
- Atmel at76c502、at76c503、at76c504、at76c506
- Texas Instruments ACX100、ACX111

普及していたが廃止になった古いカードも、多数サポートされています。WLANカードと使用チップの詳細リストについては、*AbsoluteValue Systems* (`http://www.linux-wlan.org/docs/wlan_adapters.html.gz`)にあるWebサイトを参照してください。`http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz`は、各種WLANチップの概要を提供しています。

一部のカードの場合は、ドライバの初期化時にファームウェアイメージをカードにロードする必要があります。これは、Intersil PrismGT、Atmel ACX100、ACX111を使用した場合です。ファームウェアは、YaSTオンラインアップデートを使用して簡単にインストールできます。Intel PRO-Wirelessカード用のファームウェアはSUSE LINUXに内蔵されており、この種のカードが検出されるとただちに、YaSTによって自動的にインストールされます。このトピックに関する詳細は、インストール済みシステムの`/usr/share/doc/packages/wireless-tools/README.firmware`を参照してください。

ネイティブLinuxサポートのないカードは、ndiswrapperアプリケーションを実行すれば使用できます。ndiswrapperは、ほとんどのWLANカードに同梱されるWindowsドライバを使用します。ndiswrapperについては、`/usr/share/doc/packages/ndiswrapper/README.SUSE`を参照してください(ただし`ndiswrapper`がインストールされている場合)。ndiswrapperの詳細については、プロジェクトのWebサイト`http://ndiswrapper.sourceforge.net/support.html`を参照してください。

### 17.1.2 機能

ここでは、無線ネットワーキングの基本的側面について説明します。各種の動作モード、認証、および暗号化方式について説明します。

#### 動作モード

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用の要素である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。局は相互に直接通信します。Ad-hocネットワークの場合は、伝送範囲と参加局の数が大幅に制限されます。そのため、通常はアクセスポイントを使用する方が効率的です。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。ほとんどのカードは、この機能をサポートしています。

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(項17.1.5. 「セキュリティ」)、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)。

### 認証

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(項17.1.2. 「暗号化」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、キーは適切なツールで再構成できます。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Pre-Shared Key)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。

#### 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ビットが含まれるかどうかに応じて、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。

**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水泡に帰します。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。

### 17.1.3 YaSTでの設定

無線ネットワークカードを設定するには、YaSTの［‘ネットワークカード’］モジュールを起動します。［‘ネットワークアドレスの設定’］で、デバイスタイプ［‘無線’］を選択して［‘次へ’］をクリックします。

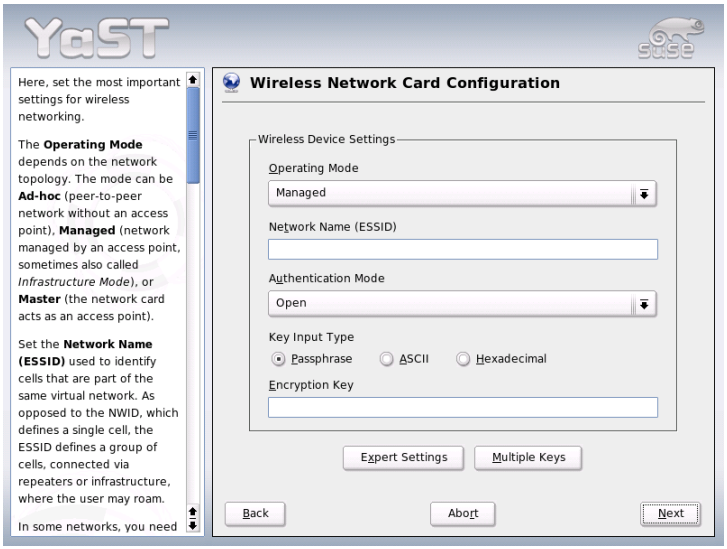


*Figure 17.1: YaST: 無線ネットワークカードの設定*

［‘無線ネットワークカードの設定’］で(図 17.1.「YaST: 無線ネットワークカードの設定」を参照)、WLAN操作の基本設定を行います。

**動作モード** WLANでは、局を3つのモードで統合できます。適切なモードは、通信に使用するネットワークに応じて異なります。選択肢は、［‘Ad-hoc’］(アクセスポイントなしのpeer-to-peerネットワーク)、［‘管理’］(アクセスポイントで管理されているネットワーク)、または［‘マスタ’］(ネットワークカードをアクセスポイントとして使用)です。

**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しますが、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。

**認証モード** ネットワークに合った認証方式を選択します。選択できるものは、［‘オープン’］、［‘共有キー’］、または［‘WPA-PSK’］です。［‘WPA-PSK’］を選択した場合は、ネットワーク名を設定する必要があります。

**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。このダイアログの詳細については後述します。

基本設定を完了すると、自局がWLANで運用可能になります。

**Important**

**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、項17.1.2.「暗号化」と項17.1.5.「セキュリティ」を参照してください。

**Important**

選択した認証方式によっては、YaSTの別のダイアログで設定を微調整するように要求されます。［'オープン'］を選択した場合、何も設定項目はありません。この設定では、認証なしの暗号化されない動作が実装されるからです。

**キーの入力タイプ** キーの入力タイプを設定します。［'パスフレーズ'］、［'ASCII'］、［'16進'］のいずれかを選択します。最大4つの異なるキーを使用して伝送データを暗号化できます。［'複数のキー'］をクリックしてキー設定ダイアログを開きます。キーの長さを選択します。選択できるのは、［'128ビット'］または［'64ビット'］です。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。［'16進'］を選択した場合は、64ビットキーに10文字、128ビットキーに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。［'パスフレーズ'］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'］をクリックしてWLAN接続の基本設定ダイアログを終了し、上級者用の設定に入ります。このダイアログでは、次のオプションを使用できます。

**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ネルを自動的に検索します。［'Ad-hoc'］モードでは、自局と他局との通信用に提供されている12のチャンネルから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のあるチャネルを指定します。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。

**アクセスポイント** 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。

**電源管理を使用** 外出先では、省電力テクノロジを使用してバッテリの動作時間を最長にします。電源管理の詳細については、章 16. 電源管理を参照してください。

### 17.1.4 ユーティリティ

WLANカードをアクセスポイントとして使用するには、hostap (`hostap`パッケージ)を使用します。このパッケージの詳細については、プロジェクトのホームページ(`http://hostap.epitest.fi/`)を参照してください。

kismet (`kismet`パッケージ)は、WLANパケットトラフィックのリスニングに使用するネットワーク診断ツールです。このツールを使用すると、ネットワーク内の侵入試行も検出できます。詳細については、`http://www.kismetwireless.net/`とマニュアルページを参照してください。

### 17.1.5 WLANのセットアップに関するヒントとテクニック

LANのセキュリティの側面だけでなく、速度と安定性を微調整する方法についても、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に依存します。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KDEでkwifimanagerを使用して、信号強度をチェックします。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。メーカ指定のレート(54MBit/sなど)は、理論上の上限を表す名目値です。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でください。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。こ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、すべてのハードウェアコンポーネントでWPAをサポートしているわけではありません。このマニュアルの制作時点では、WPAが機能するのは、AtherosまたはPrism2/2.5/3チップを使用するカードの場合だけです。後者の場合、WAPが機能するのはhostapドライバが使用されている場合だけです(項17.1.6. 「Prism2カードの問題」を参照)。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。

### 17.1.6 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、項17.1.1. 「ハードウェア」を参照してください。ここでは、判明している一部の問題について説明します。

#### 複数のネットワークデバイス

最新のラップトップでは、通常ネットワークカードとWLANカードが装備されています。両端のデバイスをDHCP (自動アドレス割り当て)で設定している場合は、名前解決およびデフォルトゲートウェイで問題が発生する可能性があります。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://portal.suse.com`にあるSupport Databas (サポートデータベース)を参照してください。この記事を検索するには、検索ダイアログに“DHCP”と入力します。

#### 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。

#### WPA

WPAサポートはSUSE LINUXに初めて実装されました。Linux環境では、WPAサポートはまだ開発中です。したがって、YaST ではWPA-PSKしか設定できません。WPAが機能するカードは多くありません。この種のカードを使用する場合、WPAを有効にするにはファムウェア更新が必要です。WPAを使用する場合は、`/usr/share/doc/packages/wireless-tools/README.wpa`を参照してください。

### 17.1.7 関連資料

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。`http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html`を参照してください。

## 17.2 Bluetooth

Bluetoothは、各種デバイス(携帯電話、PDA、周辺装置、ラップトップなど)やシステムコンポーネント(キーボードやマウスなど)の接続に使用される無線テクノロジです。このテクノロジ名は、スカンジナビア紛争でさまざまな対立党派を統一したデンマーク王Harold Bluetoothに由来しています。Bluetoothのロゴは、“H” (星型)と“B”を表すルーン文字に基づいています。

BruetoothをIrDAと比較すると、さまざまな側面に重要な違いがあります。まず初めに個々のデバイスが相互を直接「認識」する必要がなく、次に複数のデバイスをネットワーク上で接続することができます。ただし、最大データ転送速度は720Kbps (現行バージョン1.2の場合)です。理論上、Bluethoothは壁を隔てた通信が可能です。ただし、実際には壁の性質やデバイスクラスに依存します。10mから100mの通信範囲に、3つのデバイスクラスがあります。

### 17.2.1 基本事項

ここではBluetoothの機能に関する基本原則一般について概説します。必要なソフトウェア要件、Bluetoothによるシステムとの対話方法、Bluetoothプロファイルの機能などについて説明します。

#### ソフトウェア

Bluetoothを使用するためには、Bluetoothアダプタ(内蔵アダプタまたは外部デバイス)、ドライバ、およびBluetoothプロトコルスタックが必要です。Linuxカーネルには、すでにBluetooth用の基本ドライバが組み込まれています。Bluezシステムがプロトコルスタックとして使用されます。アプリケーションをBluetoothと確実に機能させるためには、基本パッケージ`bluez-libs`および`bluez-utils`をインストールする必要があります。この2つのパッケージには、必須サービスとユーティリティが多数用意されています。また、`bluez-firmware`パッケージのインストールを必要とするアダプタ(Broadcom、AVM BlueFritz!)もあります。`bluez-cups`パッケージは、Bluetooth接続を介した印刷処理を可能にします。

#### 一般的な相互作用

Bluetoothシステムは、必要な機能を提供する、4種類の関連するレイヤーから構成されています。

**ハードウェア** ハードウェアアダプタと、Linuxカーネルによるサポートに適したドライバです。

**設定ファイル** Bluetoothシステムの制御に使用されます。

**デーモン** 設定ファイルにより制御されるサービスで、各種機能を提供します。

**アプリケーション** アプリケーションにより、ユーザはデーモンが提供する機能を使用、および制御できます。

Bluetoothアダプタを挿入すると、ホットプラグシステムにより関連ドライバがロードされます。ドライバがロードされた後、システムは設定ファイルを検査してBluetoothを起動する必要があるかどうかを確認します。起動を必要とする場合は、どのサービスを起動するかが判別されます。この情報に基づいて、関連デーモンが起動されます。Bluetoothアダプタはインストール中に認識されます。1つ以上のアダプタが検出されると、Bluetoothを有効にします。ここで検出されない場合は、Bluetoothシステムは無効のままになります。そのため後から追加されたBluetoothデバイスは手動で有効にする必要があります。

#### プロファイル

Bluetoothでは、プロファイルによってサービスが定義されます。プロファイルには、ファイル転送プロファイル、基本印刷プロファイル、およびパーソナルエリアネットワークプロファイルなどがあります。デバイスパッケージやマニュアルで、この情報について説明されていないことがよくありますが、他方のデバイスのサービスを使用できるように設定するには、双方のデバイスが同じプロファイルを認識する必要があります。残念ながら、メーカが個々のプロファイルの定義に厳密に準拠していない場合もあります。こうした背景にもかかわらず、デバイス間の通信は通常、円滑に行われます。

次の説明で、ローカルデバイスとはコンピュータに物理的に接続された状態のデバイスを指します。アクセスに無線接続を必要とする他のデバイスはすべて、リモートデバイスと呼びます。

### 17.2.2 設定

ここではBluetoothの設定について説明します。関係する各種設定ファイル、必要な各種ツール、YaSTを使用して、または手動によるBluetoothの設定方法について解説します。

### YaSTによるBluetoothの設定

YaSTBluetoothモジュール(図 17.2. 「YaSTBluetooth設定」を参照)を使用して、システム上でBluetoothサポートを設定します。ホットプラグがシステム上でBluetoothを検出すると、(たとえば、ブート中アダプタをプラグインした場合)、このモジュールで設定された値を使用して、Bluetoothが自動的に起動されます。

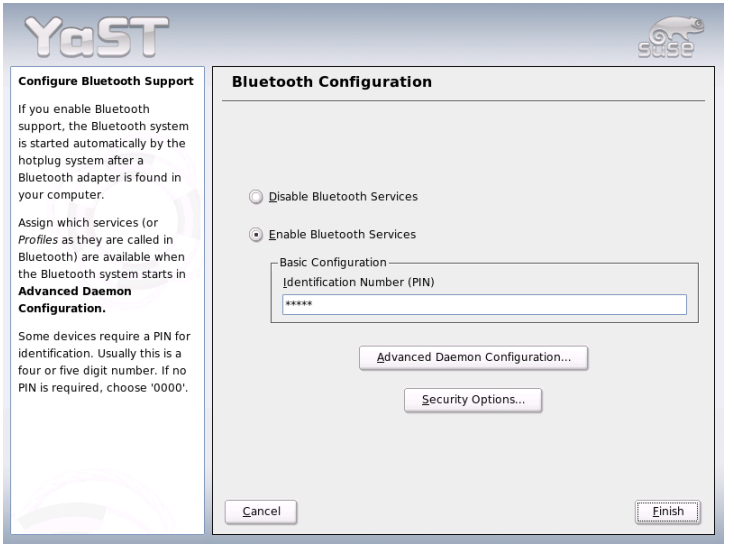


*Figure 17.2: YaSTBluetooth設 定*

最初の設定ステップでは、システムでBluetoothサービスを開始する必要があるかどうかを決定します。Bluetoothサービスが有効になっている場合、次の2つを設定できます。最初は［'デバイス名'］です。この名前はご使用のコンピュータが他のデバイスに検出された場合に、他のデバイス側に表示されるデバイス名です。これには次の2種類のプレースホルダを組み合わせることも可能です。%hはシステムのホスト名を表します(たとえば、DHCPによって動的に割り当てられる場合に有用)。%dはインタフェース番号を指定します。(ご使用のコンピュータに複数のBluetoothアダプタがある場合にのみ有用)。たとえば、フィールドに`Laptop %h`と入力し、ご使用のコンピュータがDHCPから`unit123`を割り当てられた場合、他のリモートデバイスはご使用のコンピュータを`Laptop unit123`として識別します。

2つ目のパラメータ［'セキュリティマネージャ'］は、リモートデバイスからの接続を検知した場合にローカルシステムがどのように対処するか、ということに関係しています。違いはPIN番号の取り扱いです。PINなしですべてのデバイスに接続を許可するか、PINが必要な場合に正しいPINが指定されたかを判別します。PIN (設定ファイルに格納)は、適切な入力フィールドに指定できます。あるデバイスが接続を試みた場合、最初にこのPINが使用されます。ここで認証に失敗すると、PINを使用しないように変更されます。セキュリティを高めるためには、3番目のオプションである「Always ask user for PIN(常時ユーザにPINを確認する)」を指定するのが最善と言えます。このオプションにより、異なる(リモート)デバイスに対して別のPINを使用できるようになります。

次に['拡張デーモン設定']をクリックすると、使用可能なサービス(Bluetoothでは*プロファイル*)の選択および設定用ダイアログが表示されます。使用可能なサービスがすべてリストに表示されます。ここでは['有効にする']または['無効にする']をクリックしてサービスを有効または無効にすることができます。［'編集'］をクリックするとダイアログが開き、選択したサービス(デーモン)に対する引数を追加指定できます。変更は、サービスを十分に理解している場合にのみ行ってください。デーモンの設定を完了後に、['了解']をクリックしてこのダイアログを終了します。

メインダイアログに戻り、['セキュリティオプション']をクリックしてセキュリティダイアログを表示し、暗号化、認証、およびスキャン設定を行います。次に、セキュリティダイアログを終了してメインダイアログに戻ります。['完了']をクリックしてメインダイアログを閉じると、Bluetoothシステムが使用可能になっています。

メインダイアログから'デバイスおよびサービスクラス'ダイアログにも移動できます。各Bluetoothデバイスはさまざまな「デバイスクラス」に分類されています。このダイアログで「デスクトップ」、「ラップトップ」など、ご使用のコンピュータに適したデバイスクラスを選択してください。デバイスクラスはここで同時に設定できる「サービスクラス」とは異なり、それほど重要ではありません。携帯電話のようなリモートのBluetoothデバイスでは、相手のシステム上で適切なサービスクラスを検出できた場合にのみ使用できるようになる、特定の機能を備えている場合があります。これは多くの場合携帯電話に当てるケースです。携帯電話では「オブジェクト転送」と呼ばれるクラスを待機した後、コンピュータ間のファイル転送を許可します。ここでユーザは複数のクラスを選択できます。ただし、「念のため」とすべてのクラスを選択することは実用的ではありません。通常の場合はデフォルト設定で問題ありません。

Bluetoothを使用してネットワークをセットアップする場合は、［'拡張デーモン設定'］ダイアログで［'PAND'］を有効にし、'編集'でデーモンのモードを設定します。Bluetoothネットワーク接続を機能させるには、一方

のpandが['ネットワーク接続(受信モード)']で動作し、ピアが['ネットワーク接続(検索モード)']で動作する必要があります。デフォルトでは、['ネットワークデーモンをリッスンモードに設定']モードが事前に設定されています。ローカルpandの動作を調整します。さらに、YaSTの['ネットワークカード']モジュールでbnepXインタフェース(Xはシステム内のデバイス番号)を設定します。

### Bluetoothの手動設定

Bluezシステムの各コンポーネントの設定ファイルは、ディレクトリ/etc/bluetoothにあります。ただし、コンポーネント起動用のファイル/etc/sysconfig/bluetoothは、YaSTモジュールにより変更されます。

ここで説明する設定ファイルは、ユーザrootのみが変更できます。現在のところ、すべての設定を変更できるグラフィカルユーザインタフェースはありません。YaST Bluetoothモジュールを使用して指定できる最も重要な設定は項17.2.2. 「YaSTによるBluetoothの設定」に記載されています。その他のすべての設定は経験のあるユーザ向けであり、特殊な場合以外、必要ではありません。通常、デフォルト設定はそのままで適切です。

PIN番号は、不正な接続を防止するための基本的な保護手段です。携帯電話は、最初に接続するときに(またはデバイスから電話への接続をセットアップするときに)、通常PIN番号を問い合わせます。2つのデバイスが通信を行うためには、両方が同じPIN番号で互いを識別する必要があります。コンピュータ上では、PINはファイル/etc/bluetooth/pinにあります。

**Important**

**Bluetooth接続のセキュリティ**

PINを使用しても、2つのデバイス間の通信が完全に安全なわけではありません。デフォルトでは、Bluetooth接続の認証と暗号化は無効になっています。認証と暗号化を有効にすることで、結果的に一部のBluetoothデバイス間では、通信時の問題につながる可能性があります。

**Important**

デバイス名やセキュリティモードなど、さまざまな設定を設定ファイル/etc/bluetooth/hcid.confで変更できます。通常、デフォルト設定はそのままで適切です。このファイルには、さまざまな設定のオプションを説明するコメントが含まれています。

インクルードファイルの2つのセクションがoptionsおよびdeviceとして指定されています。最初のセクションには、hcidで起動に使用される一般情報が含

まれています。次のセクションには、個々のローカルBluetoothデバイスの設定が含まれています。

optionsセクションで最も重要な設定の1つがsecurity auto;です。autoに設定すると、hcidは着信接続にローカルPINを使用します。ローカルPINで接続に失敗すると、PINがnoneに切り替わり、いずれにしても接続を確立します。セキュリティレベルを高めるために、このデフォルト設定をuserに指定し、接続の確立時にユーザに対して必ずPINの入力を促すようにする必要があります。

deviceセクションでは、接続先のデバイスで表示される、このコンピュータの表示名を設定します。デバイスクラス(Desktop、Laptop、Serverなど)も、このセクションで定義します。認証と暗号化も、ここで有効または無効にします。

### 17.2.3 システムコンポーネントとユーティリティ

Bluetoothの操作性は、さまざまなサービスとの対話に依存します。これには次のようなバックグラウンドデーモンが少なくとも2つ必要です。1つは、Bluetoothデバイスのインタフェースとして機能し、デバイスを制御するhcid(*ホストコントローラインタフェース*)、もう1つは、ホストが提供するサービスをデバイス側で確認するためのsdpd(*サービスディスカバリプロトコル*)です。システムを起動したときにこれらが自動的に有効になっていない場合は、rcbluetooth startを使用してhcidとsdpdを有効にできます。このコマンドは、rootとして実行する必要があります。

ここでは、Bluetoothの操作に使用できる最も重要なシェルツールについて説明します。現在、Bluetoothの制御に使用できるグラフィカルコンポーネントが多数出回ってはいますが、これらのツールプログラムについても調べてみるだけの価値はあります。

コマンドの中には、rootとしてのみ実行できるコマンドもあります。リモートデバイスのテストに使用するl2ping ⟨*device_address*⟩もそのようなコマンドの1つです。

#### hcitool

hcitoolは、ローカルおよびリモートのデバイスが検出されたかどうかを判断するために使用します。コマンドhcitool devを実行すると、ローカルデバイスが一覧表示されます。出力には、検出されたすべてのローカルデバイスについて、⟨*interface_name*⟩ ⟨*device_address*⟩という形式で1行に1つのデバイスが表示されます。

コマンドhcitool inqを使用してリモートデバイスを検索します。検出されたすべてのデバイスについて、デバイスアドレス、クロックオフセット、およびデバイスクラスの3つの値が表示されます。デバイスアドレスは、他のコマンドでターゲットデバイスを識別するために使用する重要な値です。クロックオフセットは、主に技術的な目的で使用されます。クラスには、デバイスタイプとサービスタイプが16進数で指定されます。

コマンドhcitool name ⟨*device-address*⟩は、リモートデバイスのデバイス名を確認するために使用します。リモートコンピュータの場合、クラスとデバイス名は/etc/bluetooth/hcid.conf内の情報に対応します。ローカルデバイスのアドレスを指定すると、エラーが出力されます。

#### hciconfig

コマンド/usr/sbin/hciconfigを実行すると、ローカルデバイスの詳細情報が表示されます。引数を指定せずにhciconfigを実行すると、出力にはデバイス名(hciX)、物理デバイスアドレス(00:12:34:56:78形式の12桁の番号)などのデバイス情報と、伝送済みデータ量に関する情報が表示されます。

hciconfig hci0 nameを実行すると、リモートデバイスから要求を受信したときにコンピュータから戻される名前が表示されます。hciconfigは、ローカルデバイスの設定のクエリだけでなく、これらの設定の変更にも使用できます。たとえば、hciconfig hci0 name TESTを実行すると、名前がTESTに設定されます。

#### sdptool

プログラムsdptoolは、特定のデバイスでどのサービスが利用可能かを確認するために使用します。コマンドsdptool browse ⟨*device_address*⟩を実行すると、デバイスのすべてのサービスが一覧表示されます。コマンドsdptool search ⟨*service_code*⟩を使用して特定のサービスを検索します。このコマンドを実行すると、要求したサービスからアクセスできるすべてのデバイスがスキャンされます。そのデバイスのいずれかがサービスを提供している場合、プログラムはこのデバイスから返された(完全な)サービス名と簡単な説明を出力します。パラメータなしでsdptoolと入力することにより、提供されている全サービスコードの一覧を表示します。

### 17.2.4 グラフィックアプリケーション

Konquerorで、URLsdp:/を入力してローカルとリモートのBluetoothデバイスのリストを表示します。デバイスをダブルクリックすると、そのデバイスが提

供するサービスの概要が表示されます。指定したサービスの1つにマウスを合わせると、そのサービスに使用されているプロファイルがブラウザのステータスバーに表示されます。サービスをクリックするとダイアログが開き、実行する操作を選択できます。保存、サービスの使用(アプリケーションの起動が必要)、またはアクションの取り消しのいずれかを選択します。ダイアログを再表示せず、選択したアクションを常に実行する場合は、チェックボックスをオンにします。一部、まだサポートが使用可能になっていないサービスや、追加パッケージのインストールを必要とするサービスがあります。

### 17.2.5 例

このセクションではBluetoothのシナリオとして想定される典型的な例を2つ取り上げます。最初の例では2つのホスト間でBluetooth経由のネットワーク接続がどのように確立されるかについて説明します。次の例ではコンピュータと携帯電話間の接続について説明します。

#### 2台のホスト間のネットワーク接続

最初の例では、ホスト*H1*と*H2*の間にネットワーク接続が確立されます。この2つのホストのBluetoothデバイスアドレスは*baddr1*と*baddr2*(前述のように両方のホスト上でコマンド`hcitool dev`を使用して判別)です。この2つのホストをIPアドレス`192.168.1.3` (*H1*)および`192.168.1.4` (*H2*)で識別する必要があります。

Bluetooth接続は、pand (パーソナルエリアネットワーキング)を使用して確立されます。次に示す各コマンドは、ユーザ`root`として実行する必要があります。ここでは、Bluetooth固有のアクションを中心に説明し、ネットワークコマンド`ip`の詳細は省略します。

コマンド`pand -s`を入力して、ホスト*H1*で`pand`を起動します。次に、コマンド`pand -c` ⟨*baddr1*⟩を使用することで、ホスト*H2*での接続を確立できます。一方のホストで`ip link show`と入力すると、使用可能なネットワークインタフェースのリストが表示されます。出力には、次のようなエントリが含まれています。

```
bnep0:<BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

出力には、`00:12:34:56:89:90`の代わりにローカルデバイスのアドレス*baddr1*または*baddr2*が含まれているはずです。ここで、このインタフェースにIPアドレスを割り当て、有効にする必要があります。*H1*で、そのために以下の2つのコマンドを使用します。

```
ip addr add 192.168.1.3/24 dev bnep0
ip link set bnep0 up
```

*H2*で実行するコマンド:

```
ip addr add 192.168.1.4/24 dev bnep0
ip link set bnep0 up
```

これで*H1*は、*H2*からIP `192.168.1.3`で、アクセスできます。コマンド`ssh 192.168.1.4`を使用して、*H1*から*H2*にアクセスします(*H2*が、SUSE LINUXによってデフォルトで有効にされたsshdを実行していると仮定します)。コマンド`ssh 192.168.1.4`は、一般ユーザとしても実行できます。

**携帯電話からコンピュータへのデータ転送**

2つ目の例では、携帯電話内蔵のデジタルカメラで撮った写真を、(マルチメディアメッセージの転送に必要になる余分なコストをかけずに)コンピュータに転送する方法を示します。携帯電話は機種によってメニュー構造が異なりますが、手順は通常、ほとんど同じです。必要であれば、携帯電話のマニュアルを参照してください。この例では、Sony Ericssonの携帯電話からラップトップに写真を転送する方法について説明します。サービスObex-Pushがコンピュータ上で利用でき、コンピュータが携帯電話からのアクセスを許可している必要があります。最初のステップでは、サービスをラップトップで利用できるようにします。これには、パッケージ`bluez-utils`にあるopdデーモンを使用します。次のコマンドでデーモンを起動します。

```
opd --mode OBEX --channel 10 --daemonize --path /tmp --sdp
```

ここで使用される2つのパラメータは重要です。`--sdp`はsdpdを使用してサービスを登録します。`--path /tmp`は、プログラムに対し、受信データの保存場所(ここでは、`/tmp`)を指定します。書き込みアクセス権を持つ他のディレクトリを指定することもできます。

次に、携帯電話がコンピュータを認識するように設定する必要があります。これには、電話側で['Connect(接続)']メニューの['Bluetooth']を選択します。必要に応じて、['My devices(マイデバイス)']を選択する前に、['Turn On(オン)']をクリックします。 ['New device(新規デバイス)']を選択すると、携帯電話がラップトップを探査します。デバイスが検出されると、ディスプレイに名前が表示されます。ラップトップに関連付けられているデバイスを選択します。PINの入力を求められたら、`/etc/bluetooth/pin`に指定されてい

るPINを入力します。これで、携帯電話がラップトップを認識し、ラップトップとデータを交換できるようになりました。現在のメニューを終了し、イメージメニューに移動します。転送するイメージを選択し、['More(詳細)']を押します。次のメニューでは、['Send(送信)']を押して通信モードを選択します。['Via Bluetooth(Bluetooth経由)']を選択します。ラップトップがターゲットデバイスとして表示されます。ラップトップを選択し、通信を開始します。イメージは、opdコマンドで指定したディレクトリに保存されます。オーディオトラックも、同じ方法でラップトップに転送できます。

### 17.2.6 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。エラーは接続の片端または両端で発生する可能性があることに注意してください。できれば、別のBluetoothデバイスで問題を再生し、そのデバイスに問題がないことを確認してください。

**`hcitool dev`の出力にローカルデバイスが表示されますか？**
: この出力にローカルデバイスが表示されない場合は、hcidが起動していないか、デバイスがBluetoothデバイスとして認識されていません。これにはさまざまな原因が考えられます。デバイスに不具合がある、正しいドライバがない、などです。Bluetooth内蔵ラップトップの場合、通常は無線デバイス(WLANやBluetoothなど)用のオン/オフスイッチが備えられています。この種のスイッチがデバイスにあるかどうかをラップトップのマニュアルで確認してください。コマンド`rcbluetooth restart`でBluetoothシステムを再起動し、`/var/log/messages`にエラーが報告されるかどうかを調べます。

**Bluetoothアダプタにはファームウェアファイルが必要ですか？**
: ファームウェアファイルが必要な場合は、`bluez-bluefw`をインストールし、`rcbluetooth restart`でBluetoothシステムを再起動します。

**`hcitool inq`の出力に他のデバイスが表示されますか？**
: このコマンドは何度かテストしてください。Bluetoothの周波数帯が他のデバイスでも使用されているため、接続に干渉が発生している可能性があります。

**PINは一致していますか？** コンピュータのPIN番号(`/etc/bluetooth/pin`内)がターゲットデバイスと一致しているかどうかを確認してください。

**リモートデバイスは使用中のコンピュータを「認識」できますか？**
リモートデバイスから接続を行ってみます。このデバイスがコンピュータを認識するかを確認します。

**ネットワーク接続を確立できますか(例1を参照)？**
最初の例(ネットワーク接続)が機能しない場合は、いくつかの理由が考えられます。たとえば、2台のコンピュータの一方がsshプロトコルをサポートしていない可能性があります。`ping 192.168.1.3`または`ping 192.168.1.4`を試してみます。このコマンドが実行できる場合は、次にsshdが有効かどうかを確認します。他に考えられる原因としては、例で使用しているアドレス`192.168.1.X`と競合するネットワーク設定が既に一方のデバイスで使用されている場合です。この場合、`10.123.1.2`と`10.123.1.3`のような異なるアドレスを試してみます。

**ラップトップはターゲットデバイスとして表示されますか(例2)？携帯デバイスは、ラップトップのObex-Pushを認識しますか？**
['My devices(マイデバイス)']で各デバイスを選択し、['Services(サービス)']のリストを表示します。(リストを更新しても) Obex-Pushが表示されない場合は、ラップトップ上のopdが原因です。opdはアクティブですか？指定したディレクトリに対して、書き込みアクセスができますか？

**例2は他の方法でも機能しますか？** `obexftp`パッケージがインストールされている場合は、一部のデバイスでこの操作にコマンド`obexftp -b ⟨device_address⟩ -B 10 -p ⟨image⟩`を使用できます。複数のSiemensおよびSony Ericssonモデルではテストを完了し、機能することが確認されています。`/usr/share/doc/packages/obexftp`にあるパッケージのドキュメントを参照してください。

## 17.2.7 関連資料

Bluetoothの使用方法と設定についての詳細な説明は、`http://www.holtmann.org/linux/bluetooth/`で入手可能です。その他、次の情報および指示が役立ちます。

- カーネルに統合されているBluetoothプロトコルスタックの公式HOWTO: `http://bluez.sourceforge.net/howto/index.html`
- PalmOS PDAへの接続: `http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html`

# 17.3 赤外線データ通信

IrDA (*Infrared Data Association*)は赤外線による無線通信の標準規格名です。現在販売されている多くのラップトップにはプリンタ、モデム、LAN、他のラップトップなど、リモートデバイスとの通信を可能にするIrDA互換トランシーバが装備されています。通信速度は2400 bpsから4M bpsの範囲になります。

IrDA操作モードには2種類あります。SIRは標準モードで、シリアルインタフェースを使用して赤外線ポートにアクセスします。このモードはほぼどのシステムでも機能し、ほとんどの要件に対応します。一方、より高速なFIRモードにはIrDAチップ用の特別なドライバが必要になります。しかるべきドライバがないため、FIRモードでサポートされていないチップタイプもあります。コンピュータのBIOSで必要なIrDAモードを設定します。また、BIOSではSIRモードで使用されるシリアルインタフェースも表示します。

IrDAについての情報は、`http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html`でWerner Heuser著の『Linux Infrared HOWTO』を参照してください。また、`http://irda.sourceforge.net/`のLinux IrDA Project Webサイトも参照してください。

## 17.3.1 ソフトウェア

必要なカーネルモジュールは、カーネルパッケージに組み込まれています。`irda`パッケージでは赤外線通信インタフェースをサポートするために必要な各種アプリケーションを提供しています。このマニュアルはパッケージをインストールした後、`/usr/share/doc/packages/irda/README`で参照できます。

## 17.3.2 設定

IrDAシステムサービスはシステムのブート時に自動的に開始されません。YaST IrDAモジュールを使用して有効にします。このモジュールで変更できる設定は1つだけです。赤外線デバイスのシリアルインタフェースのみが変更可能です。テストウィンドウに2種類の出力が表示されます。1つは`irdadump`の出力です。IrDAを介したすべての送受信パケットを記録します。この出力にはコンピュータ名および、通信圏内にあるすべての赤外線デバイス名が表示されます。項17.3.4. 「トラブルシューティング」で、これらのメッセージ例を参照できます。IrDA接続を持つすべてのデバイスがウィンドウの下部に表示されます。

IrDAは相当な量の電力を消費します。周辺機器を検出するために数秒間隔でディスカバリパケットを送信するためです。そのため、電源がバッテリのみの場合は、必要なとき以外はIrDAを開始しないようにします。コマンドrcirda startでIrDAを有効に、またコマンドrcirda stopで無効にします。インタフェースが有効なった時点で、必要なカーネルモジュールがすべて自動的にロードされます。

ファイル/etc/sysconfig/irdaを使用して、手動による設定が可能です。このファイルに含まれるのは変数IRDA_PORTのみです。この変数はSIRモードで使用するインタフェースを決定します。

### 17.3.3 使用方法

印刷用のデータをデバイスファイル/dev/irlpt0に送信できます。デバイスファイル/dev/irlpt0は、データ送信が赤外線を使用して無線で行われる以外、通常のケーブル接続されたインタフェースである、/dev/lp0と同様に動作します。印刷時には、プリンタがコンピュータの赤外線インタフェースから見える範囲にあり、赤外線サポートが開始されていることを確認してください。

赤外線インタフェースを介してプリンタを操作する場合、YaSTプリンタモジュールを使用してプリンタを設定できます。プリンタは自動検出されないため、［'その他(未検出)'をクリックし、手動で設定してください。その後にダイアログで［'IrDAプリンタ'］を選択します。通常、irlpt0が正しい接続になります。Linuxでのプリンタ操作に関する詳細については、章 12. プリンタの運用を参照してください。

その他のホストおよび、携帯電話またはそれに類するデバイスとの通信は、デバイスファイル/dev/ircomm0を介して行われます。たとえば、携帯電話のSiemens S25、Nokia 6210といった機種では、赤外線インタフェースを使用するwvdialアプリケーションで、ダイヤルおよびインターネットへの接続が可能です。Palm Pilotとのデータ同期も可能です。対応するアプリケーションのデバイス設定は、/dev/ircomm0に指定されています。

必要に応じて、プリンタまたはIrCOMMプロトコルをサポートするデバイスのみ指定できます。3Com Palm PilotのようなIROBEXプロトコルをサポートするデバイスにはirobexpalmおよびirobexreceiveなどの特別なアプリケーションを使用してアクセスできます。.詳細については、*IR-HOWTO* (http://tldp.org/HOWTO/Infrared-HOWTO/)を参照してください。irdadumpの出力で、デバイス名の隣にあるカッコ内にデバイス別にサポートされるプロトコルが一覧表示されます。IrLANプロトコルのサポートは現在「開発中」です。

### 17.3.4 トラブルシューティング

赤外線ポートに接続されたデバイスが応答しない場合は、コマンドirdadump (rootで)を使用して、コンピュータが他のデバイスを認識しているか確認します。Canon BJC-80プリンタがコンピュータの通信可能範囲内にあると、例 17.1. 「irdadumpの出力」のような内容が定期的に表示されます。

***Example 17.1:*** *irdadumpの 出 力*

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
          hint=8804 [Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* earth
          hint=0500 [ PnP Computer ] (21)
```

出力が得られない、または他のデバイスが応答しない場合は、インタフェースの設定を確認してください。正しいインタフェースが使用されていることを確認します。赤外線インタフェースが/dev/ttyS2または/dev/ttyS3にあったり、IRQ 3以外の割り込みが使用されていたりする場合もあります。ほぼすべてのラップトップのBIOS設定メニューから、これらの設定を確認および変更できます。

簡単なビデオカメラでも赤外線LEDライトが点灯しているかどうかを確認できます。多くのビデオカメラは人間の眼に見えない赤外線を感知することができるためです。

# ホットプラグシステム

ホットプラグシステムでコンピュータのほとんどのデバイスの初期化を管理します。ホットプラグシステムは、稼動中に取り付け/取り外しできるデバイスに使用されるだけでなく、システムのブート中に検出されるすべてのデバイスにも使用されます。また、sysfsファイルシステムやudev(章 19. udevをもつ動的デバイスノードを参照)と密接に連携します。

カーネルのブートが完了するまでは、バスシステム、ブートディスク、およびキーボードのような絶対に必要なデバイスだけが初期化されます。カーネルは、検出されたすべてのデバイスのホットプラグイベントをトリガします。udevdデーモンは、こうしたイベントをリッスンして、検出されたデバイスを初期化するためにそれぞれのホットプラグスクリプトを呼び出します。自動的に検出できないデバイスや、初期ブート時にイベントが失われたデバイスのために、coldplugがあります。coldplugでは、記録されたイベントを再生するか、または初期化されていないデバイスがないかシステムを調べて、ISAのような古いデバイスの静的な設定を使用します。

過去の少数の例外を除き、ほとんどのデバイスは、システムのブート時またはデバイスの接続時にアクセス可能になり次第、すぐに初期化されます。初期化中に、インタフェースはカーネルに登録されます。この登録によって、それぞれのインタフェースを自動的に設定させるためのホットプラグイベントがさらにトリガされます。

SUSE LINUXの前バージョンでは、デバイスを初期化する基礎として一連の静的な設定データが使用されていました。現在、このシステムでは、使用可能な各デバイスを調べて適切な設定データを検索するか、またはそれを生成します。

非常に重要なホットプラグ機能は、2つのファイル内で設定されます。そのうち最初のファイルである/etc/sysconfig/hotplugには、hotplugとcoldplugの動作に影響を及ぼす変数が含まれています。すべての変数にはコメントが付いています。2番目のファイルである/proc/sys/kernel/hotplugには、カーネルによって呼び出される実行可能プログラムの名前が含まれています。デバイス設定は、/etc/sysconfig/hardware内に配置されます。SUSE LINUX 9.3から登場したこのファイルが通常空であるのは、udevdがnetlinkソケットを介してホットプラグメッセージを受信するためです。

## 18.1 デバイスとインタフェース

ホットプラグシステムでは、デバイスだけでなくインタフェースも管理します。デバイスは、バスまたはインタフェースのどちらかにリンクされます。バスは複数インタフェースと見なすことができます。インタフェースは、デバイス同士をリンクするか、デバイスをアプリケーションにリンクします。ネットワークトンネルなどの仮想デバイスもあります。通常、デバイスは、カーネルモジュールの形でドライバを必要とします。ほとんどのインタフェースは、udevによって作成されたデバイスノードで表されます。全体的な概念を把握するには、デバイスとインタフェースを区別することが重要です。

sysfsファイルシステム内に入力されたデバイスは、/sys/devices内で見つかります。インタフェースは、/sys/classまたは/sys/blockの下に配置されています。sysfs内のすべてのインタフェースには、それぞれのデバイスへのリンクが必要です。ただし、このリンクを自動的に追加しないドライバもあります。該当するリンクがない場合は、このインタフェースがどのデバイスに属しているかが不明なので、適切な設定を検索することができません。

デバイスのアドレス指定には、デバイス記述が使用されます。これには、sysfs内のデバイスパス(/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0)、接続ポイントの記述(bus-pci-0000:02:00.0)、および個々のID (id-32311AE03FB82538)などがあります。従来、インタフェースは名前で処理されていました。こうした名前は既存のデバイスの単純な番号を表しており、デバイスが追加または削除されると変更されることがありました。

関連デバイスの記述を使用してインタフェースをアドレス指定することもできます。通常、コンテキストは、記述がデバイス自体を指すのか、またはそのインタフェースを指すのかを示します。デバイス、インタフェース、および記述の標準的な例として、次のものを挙げることができます。

**PCIネットワークカード** PCIバス(/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0またはbus-pci-0000:02:00.0)に接続され、ネットワークインタフェース(eth0、id-00:0d:60:7f:0b:22またはbus-pci-0000:02:00.0)を持つ1つのデバイス。ネットワークインタフェースはネットワークサービスに使用されるか、トンネルやVLANなど、インタフェースを持つ仮想ネットワークデバイスに接続されます。

**PCI SCSIコントローラ** 複数の物理インタフェースをバス(/sys/class/scsi_host/host1)の形式で使用可能にする1つのデバイス(/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0またはbus-scsi-1:0:0:0)。

**SCSIハードディスク** 複数のインタフェース(/sys/block/sda*)を使用する1つのデバイス(/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0またはbus-scsi-1:0:0:0)。

## 18.2 ホットプラグイベント

各デバイスと各インタフェースには、ホットプラグイベントが関連付けられており、udevおよび担当ホットプラグエージェントがそれらのイベントを処理

します。ホットプラグイベントは、デバイスへのリンクの確立時または解除時、あるいはドライバによるインタフェースの登録時または削除時に、カーネルによりトリガされます。SUSE LINUX 9.3以降は、udevdでホットプラグイベントが受信および分配されます。udevdがカーネルからnetlinkメッセージを直接リッスンするか、または/sbin/udevsendを/proc/sys/kernel/hotplugに指定する必要があります。udevdがそのジョブ(章 19. udevをもつ動的デバイスノードを参照)を完了したら、該当するイベントタイプに対応するホットプラグエージェントが/etc/hotplug.d/にないか検索します。

## 18.3 ホットプラグエージェント

ホットプラグエージェントとは、イベントに対して適切なアクションを実行する実行可能なプログラムのことです。デバイスイベントのエージェントは、/etc/hotplug.d/⟨イベント名⟩および/etc/hotplug.d/defaultにあります。これらのディレクトリ内の、拡張子が.hotplugのすべてのプログラムは、アルファベット順に実行されます。

特定の種類のイベントを確実に無視するために、それぞれのホットプラグエージェントから実行可能ビットを削除します。あるいは、拡張子.hotplugをほかの拡張子に変更します。

通常、デバイスエージェントはカーネルモジュールを読み込みますが、時には、追加のコマンドを呼び出すこともあります。SUSE LINUX環境では、/sbin/hwupまたは/sbin/hwdownがこの処理を行います。これらのプログラムは、デバイスに適した設定をディレクトリ/etc/sysconfig/hardware内で検索して適用します。特定のデバイスが初期化されないようにするには、起動モードmanualまたはoffを使用して適切な設定ファイルを作成します。/sbin/hwupが何も設定を見つけられなかった場合、モジュールはエージェントにより自動的に読み込まれます。この場合には、一部のエージェントがhwupの設定ファイルを自動的に生成します。このため、次回このコマンドを実行したときのエージェントの処理が高速になります。詳細については、項18.4. 「自動的なモジュール読み込み」を参照してください。/sbin/hwupの詳細については、/usr/share/doc/packages/sysconfig/README ファイルおよびマニュアルページ(man hwup)を参照してください。

インタフェースエージェントが呼び出される前に、通常は、udevでシステムがアクセスできるデバイスノードが生成されます。udevを使用すると、インタフェースに永続的な名前を割り当てることができます。詳細については、章 19. udevをもつ動的デバイスノードを参照してください。その後、個々の

イベントによりインタフェースがセットアップされます。ここでは、一部のインタフェースに関する手順を説明します。

### 18.3.1 ネットワークインタフェースの有効化

/sbin/ifupを使用してネットワークインタフェースを初期化し、/sbin/ifdownを使用して無効にします。詳細については、ファイル/usr/share/doc/packages/sysconfig/READMEおよびマニュアルページ(man ifup)を参照してください。

異なるドライバを使用する複数のネットワークデバイスがコンピュータに存在する場合、別のドライバを読み込む方が高速であれば、システムのブート中にインタフェース指定を変更できます。このため、SUSE LINUXはキューを使用してPCIネットワークデバイスに対するイベントを管理します。この機能は、ファイル/etc/sysconfig/hotplug内で変数HOTPLUG_PCI_QUEUE_NIC_EVENTS=noを使用して無効にすることができます。

ただし、最善の解決策は永続的なインタフェース指定を活用することです。設定ファイルで個々のインタフェースの名前を指定できます。この方法の詳細については、/usr/share/doc/packages/sysconfig/READMEを参照してください。SUSE LINUX 9.3以降、デバイスノードではありませんが、udevでネットワークインタフェースも処理されます。このため、より標準化された方法で永続的なインタフェース名を使用できます。

### 18.3.2 ストレージデバイスの有効化

ストレージデバイスにアクセスするには、そのインタフェースをマウントする必要があります。この作業は、完全に自動化するか事前に設定できます。設定には、/etc/sysconfig/hotplug内と/etc/fstab内の変数HOTPLUG_DO_MOUNT、HOTPLUG_MOUNT_TYPE、およびHOTPLUG_MOUNT_SYNCを使用します。変数HOTPLUG_DO_MOUNT=yesを設定すると、完全に自動化された操作を有効にすることができます。完全に自動化された操作を無効にするには、この変数をnoに設定します。

2つのモード(subfsおよびfstab)間で切り替えるには、変数HOTPLUG_MOUNT_TYPEを使用します。HOTPLUG_MOUNT_TYPE=subfsモードでは、ディレクトリ/media内にサブディレクトリが作成されます。サブディレクトリ名は、デバイスプロパティから導出されます。メディアは、アクセス時にsubmountdにより自動的にマウントおよびアンマウントされます。このモードのデバイスは、アクセスできなくなると簡単に削除できます。HOTPLUG_MOUNT_TYPE=fstabモードでは、ストレージデバイスはファイ

ル/etc/fstab内の適切なエントリを使用して従来の方法でマウントされます。

変数HOTPLUG_MOUNT_SYNCを設定すると、同期モードまたは非同期モードでのアクセスを可能にすることができます。非同期モードでは、結果がバッファに入れられるので、書き込みアクセスの方が高速です。ただし、データメディアを不注意に削除すると、データの書き込みが不完全になる場合があります。同期モードでは、すべてのデータが即時に書き込まれますが、アクセス時間は長くなります。デバイスは、umountを使用して手動でアンマウントする必要があります。

永続的なデバイス名を使用することをお勧めします。この理由は、初期化シーケンスによっては従来のデバイス名が変更される場合があるためです。永続的なデバイス名の詳細については、章 19. udevをもつ動的デバイスノードを参照してください。

## 18.4 自動的なモジュール読み込み

デバイスを/sbin/hwupで初期化できなければ、エージェントはモジュールマップで適切なドライバを検索します。エージェントが最初に検索するのは、/etc/hotplug/*.handmap内に含まれているマップです。そこでドライバが見つからなかった場合、エージェントは/lib/modules/<kernelversion>/modules.*map内も検索します。カーネル用の標準ドライバとは異なるドライバを使用するには、そのドライバを/etc/hotplug/*.handmapに入力します。これは最初に読み取られるファイルだからです。

USBエージェントは、/etc/hotplug/usb.usermapと/etc/hotplug/usb/*.usermapの各ファイル内でユーザモードドライバの検索も行います。ユーザモードドライバは、カーネルモジュールの代わりにデバイスへのアクセスを制御するプログラムです。この方法で、特定のデバイスに対して実行可能プログラムを呼び出すことも可能です。

PCIデバイスの場合、pci.agentが、最初にhwinfoにドライバモジュールについて問い合わせます。hwinfoがどのドライバも認識していない場合に限って、エージェントはpci.handmapとカーネルマップを参照します。hwinfoはそこを既に検索済みなので、この照会は失敗します。hwinfoには、ドライバマッピング用の追加データベースがあります。ただし、このファイル内の個々のマッピングが確実に適用されるように、エージェントはpci.handmapも読み込みます。

/lib/modules/<kernelversion>/kernel/driversの特定のサブディレクトリ内にある特定のタイプのデバイスまたはドライバモジュールだけを使用するように、pci.agentエージェントに制限を加えることもできます。

デバイスの場合は、ファイル/usr/share/pci.idsの最後にあるPCIデバイスクラスを、ファイル/etc/sysconfig/hotplug内の変数HOTPLUG_PCI_-CLASSES_WHITELISTおよびHOTPLUG_PCI_CLASSES_BLACKLISTに入力できます。ドライバモジュールの場合は、変数HOTPLUG_PCI_DRIVERTYPE_-WHITELISTおよびHOTPLUG_PCI_DRIVERTYPE_BLACKLISTで1つまたは複数のディレクトリを指定します。除外したディレクトリからはモジュールが読み込まれません。どちらの場合も、空のホワイトリストは、ブラックリストで除外した以外の候補はすべて読み込み可能であることを意味します。また、1つのモジュールを読み込みから除外することもできます。エージェントで読み込まないモジュールだけをファイル/etc/hotplug/blacklistに入力してください。各モジュール名は、個別の行で記述します。

1つのマップファイル内で、適切なモジュールが複数見つかった場合は、最初のモジュールだけが読み込まれます。モジュールすべてを読み込むには、変数HOTPLUG_LOAD_MULTIPLE_MODULES=yesを設定します。ただし、このデバイス用に別のデバイス設定/etc/sysconfig/hardware/hwcfg-*を作成する方が適切です。

hwupを使用して読み込まれたモジュールは、この影響を受けません。自動モジュール読み込みが発生するのは例外的な場合だけで、SUSE LINUXの将来のバージョンではさらに限定されます。しかし、適切なモジュールが見つかった場合、エージェントは、hwup設定ファイルを作成します。このファイルが次回に使用されます。これにより、デバイス初期化の速度が向上します。

## 18.5 PCIのホットプラグ

一部のコンピュータでは、PCIデバイスのホットプラグをサポートしています。これをフルに活用するには、特殊なカーネルモジュールを読み込む必要があります。しかし、PCIホットプラグ非対応のコンピュータでは、これらのモジュールが問題を引き起こす可能性があります。残念ながら、ホットプラグPCIスロットは自動検出できません。PCIホットプラグ対応を手動で設定するには、/etc/sysconfig/hotplugファイル内で変数HOTPLUG_DO_REAL_PCI_HOTPLUGをyesに設定します。

## 18.6 ブートスクリプトcoldplug

boot.coldplugは、自動検出されないデバイスと、ホットプラグイベントが生成されないデバイスをすべて受け持ちます。単に、/etc/sysconfig/

hardware/hwcfg-static-*として指定されているスタティックなデバイス設定ごとにhwupを呼び出します。このスクリプトを使用して、hotplugの場合とは異なる順序で内蔵デバイスを初期化することもできます。これは、coldplugがhotplugの前に実行されるからです。

# 18.7 エラーの解析

## 18.7.1 ログファイル

特に指定がない限り、hotplugは少数の重要なメッセージをsyslogに送信します。詳細情報を取得するには、/etc/sysconfig/hotplugファイル内で変数HOTPLUG_DEBUGをyesに設定します。この変数をmaxという値に設定した場合、あらゆるホットプラグスクリプトに関して、すべてのシェルコマンドがログに記録されます。これは、syslogによるすべてのメッセージの格納先となる/var/log/messagesが非常に大きくなることを意味します。しかし、hotplugとcoldplugが終了した後、ブートプロセスの最中にsyslogが起動されるため、最初のメッセージはログに記録されない場合があります。それらのメッセージが重要な場合、HOTPLUG_SYSLOG変数を通して他のログファイルを指定します。このトピックに関する情報は、/etc/sysconfig/hotplug内に記載されています。

## 18.7.2 ブートの問題

コンピュータがブートプロセスの最中にハングする場合、ブートプロンプトでNOHOTPLUG=yesまたはNOCOLDPLUG=yesと入力し、hotplugまたはcoldplugを無効にします。hotplugが無効になると、カーネルはホットプラグイベントを発行しません。稼働中のシステムでは、コマンド/etc/init.d/boot.hotplug startを入力してホットプラグを有効にすることができます。その時点までに生成されたイベントはすべて発行され、処理されます。キューにあるイベントを拒否するには、まず/proc/sys/kernel/hotplugに/bin/trueと入力し、後でこのエントリを/sbin/hotplugにリセットします。coldplugが無効になると、スタティックな設定は適用されません。スタティックな設定を適用するには、後で/etc/init.d/boot.coldplug startを入力します。

hotplugによって読み込まれた特定のモジュールがこの問題に関係しているかどうかを調べるには、ブートプロンプトでHOTPLUG_TRACE=<N>と入力します。読み込むすべてのモジュールの名前が画面に表示され、⟨*N*⟩秒後に実際に読み込まれます。この動作の進行中は、介入(操作)ができません。

### 18.7.3 イベントレコーダ

/sbin/hotplugにより、イベントごとにスクリプト/sbin/hotplugeventrecorderが実行されます。/eventsディレクトリが存在している場合、すべてのホットプラグイベントは個別のファイルとしてこのディレクトリ内に格納されます。これにより、テストのためにイベントを再生できます。このディレクトリが存在しなければ、何も記録されません。

# udevをもつ動的デバイスノード

Linuxカーネル2.6は、動的デバイスのディレクトリ/devに対して、新しいユーザ空間ソリューションを導入し、持続的なデバイス指定を実現しました。udevがこれに該当します。これは、実際に存在するデバイスにのみファイルを提供します。通常、/devディレクトリに存在するデバイスノードファイルを作成または削除し、ネットワークインタフェースの名前を変更します。devfsによる/devの以前の実装は機能しなくなり、udevへ置き換えられました。

従来は、デバイスノードはLinuxシステム上の/devディレクトリ内に格納されていました。システム内に実際に存在しているかどうかにかかわりなく、使用可能なすべてのデバイスタイプに対して1つのノードが存在していました。その結果、このディレクトリは多くのスペースを使用していました。devfsコマンドは大幅な改良を加えられました。実際に存在するデバイスだけが、/dev内でデバイスノードを割り当てられています。

udevは、デバイスノードを作成する新しい方法を採用しました。sysfsによって使用可能になる情報を、ユーザがルールの形で提供したデータと比較します。sysfsは、カーネル2.6での新しいファイルシステムです。システムに接続されたデバイスに関する基本情報を提供します。sysfsは、/sysの下にマウントされます。

ユーザは、ルールを作成する必要はありません。あるデバイスが接続された場合、適切なデバイスノードが作成されます。しかし、ルールは、ノードの名前を変更するための手段を導入します。これは、あいまいなデバイス名を覚えやすい名前へ置き換える規則を提供し、さらに同じタイプのデバイスが2台接続された状況で、持続するデバイス名を実現します。

特に指定がない限り、2台のプリンタは、/dev/lp0および/dev/lp1という指定を受け取ります。各デバイスの電源をオンにした順序に従って、どのデバイスがどのノードを割り当てられるかが決まります。もう1つの例は、USBハードディスクのような外付けの大容量ストレージデバイスです。udevコマンドを使用する場合、正確なデバイスパスを/etc/fstab内に入力できます。

## 19.1 ルールの作成

udevは、/devの下にデバイスノードを作成する前に、/etc/udev/rules.d内で拡張子.rulesをもつすべてのファイルをアルファベット順に読み取ります。デバイスに当てはまる最初のルールが使用されます。たとえ、他のルールも当てはまる場合であってもです。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。ルールは、次の形式を使用します。

```
key, [key,...]NAME [, SYMLINK]
```

少なくとも1つのキーを指定する必要があります。これらのキーに基づいて、ルールがデバイスに対して割り当てられるからです。また、名前を指定することも重要です。この名前は、/dev内で作成されるデバイスノードに使用されます。オプションのsymlinkパラメータを使用すると、他の場所にノードを作成できます。プリンタに関するルールは、次の形式を使用します。

```
BUS="usb", SYSFS{serial}="12345", NAME="lp_hp", SYMLINK="printers/hp"
```

この例では、BUSとSYSFS{serial}という2つのキーがあります。udevは、シリアル番号をUSBバスに接続されたデバイスのシリアル番号と比較します。名前lp_hpを/devディレクトリ内のデバイスに割り当てるには、すべてのキーが同じである必要があります。さらに、このデバイスノードを参照する、シンボリックリンク/dev/printers/hpも作成されます。この操作を実行している間に、printersディレクトリが自動的に作成されます。その後、印刷ジョブは/dev/printers/hpまたは/dev/lp_hpへ送信できます。

## 19.2 NAMEとSYMLINKを使用した自動化

NAMEとSYMLINKの各パラメータを使用すると、自動的な割り当てを行うための演算子が利用できます。これらの演算子は、該当のデバイスに関してカーネルが保持しているデータを表します。簡単な例を使用して、この手順を説明します。

```
BUS="usb", SYSFS{vendor}="abc", SYSFS{model}="xyz", NAME="camera%n"
```

名前の一部として使用されている演算子%nは、カメラデバイスの番号へ置き換えられます。たとえば、camera0やcamera1のようになります。役に立つもう1つの演算子は、%kです。これは、カーネルが保持している標準的なデバイス名へ置き換えられます。たとえば、hda1のようになります。udevルール内で外部プログラムを呼び出し、NAMEおよびSYMLINK値に返される文字列を使用することもできます。すべての演算子からなるリストについては、udevのマニュアルページを参照してください。

## 19.3 キーの中での正規表現

udevルールのキーでは、ワイルドカードとして知られるシェルスタイルのパターンマッチングを使用できます。たとえば、*字は任意の複数文字を表すプレースホルダ、?は任意の1文字を表すプレースホルダとして使用できます。

```
KERNEL="ts*", NAME="input/%k"
```

このルールは、文字tsで始まる指定を持つデバイスに対して、標準的なディレクトリ内でカーネルが使用する標準的な名前を割り当てます。udevルールの中で正規表現を使用する方法の詳細については、man udevのマニュアルページを参照してください。

## 19.4 キーの選択

適切なキーは、すべての作業用udevルールにとって重要です。ここでは、標準的なキーに関するいくつかの例を示します。

**BUS** デバイスのバスタイプ

**KERNEL** カーネルが使用するデバイス名

**ID** バスのデバイス番号(たとえば、PCIバスのID)

**PLACE** デバイスの接続先である物理ポイント(たとえば、USB上)

**SYSFS{...}** sysfs device attributes like label, vendor, serial number, etc.

IDとPLACEの各キーは役に立つこともありますが、通常はBUS、KERNEL、およびSYSFS{...}の各キーが使用されています。udevの設定は、外部スクリプトを呼び出してその結果を評価するためのキーも用意しています。詳細は、man udevのマニュアルページを参照してください。

ファイルシステムsysfsは、小規模なファイルをハードウェア情報とともにディレクトリツリー内に格納しています。一般的に、各ファイルにはデバイス名、メーカ、シリアル番号など、ただ1項目の情報が含まれます。これらの各ファイルを、キーの値として使用できます。しかし、1つのルールで複数のSYSFSキーを使用する場合、同じディレクトリの中にあるファイルだけがキー値として使用できます。有効なキー値を見つけるには、ツールudevinfoが役に立ちます。

該当のデバイスを参照していて、devファイルが含まれている/sysの1つのサブディレクトリを見つける必要があります。これらのディレクトリはいずれも、/sys/blockまたは/sys/classの下に配置されています。デバイスにデバイスノードがすでに存在している場合は、udevinfoが正しいサブディレクトリを見つけられます。udevinfo -q path -n /dev/sdaコマンドは、/block/sdaという出力をします。これは、必要とするディレクトリが/sys/block/sdaであることを意味します。今度は、コマンドudevinfo -a -p /sys/block/sdaを使用して、udevinfoを呼び出します。これら2つ

のコマンドを組み合わせて、たとえばudevinfo -a -p ‘udevinfo -q path -n /dev/sda‘のようにすることもできます。次に、出力の一部を抜粋します。

```
BUS="scsi"
ID="0:0:0:0"
SYSFS{detach_state}="0"
SYSFS{type}="0"
SYSFS{max_sectors}="240"
SYSFS{device_blocked}="0"
SYSFS{queue_depth}="1"
SYSFS{scsi_level}="3"
SYSFS{vendor}="        "
SYSFS{model}="USB 2.0M DSC    "
SYSFS{rev}="1.00"
SYSFS{online}="1"
```

出力情報から、変化しない適切なキーを探します。他のディレクトリにあるキーを使用できないことに注意してください。

## 19.5 大容量ストレージデバイスの持続的な名前

SUSE LINUXには、初期化の順序に関係なく、ハードディスクと他のストレージデバイスに同じ指定を割り当てられるようにするスクリプトが装備されています。/sbin/udev.get_persistent_device_name.shは、ラッパースクリプトです。これは最初に/sbin/udev.get_unique_hardware_path.shを呼び出します。後者は、指定されたデバイスに対応するハードウェアパスを判断します。/sbin/udev.get_unique_drive_id.shは、シリアル番号を取得します。どちらの出力も、udevに渡され、そこで、/devの下に、そのデバイスへのシンボリックリンクが作成されます。ラッパーは、udevルールの中で直接使用することもできます。次に、SCSIの例を示します。これを一般化して、USBまたはIDEに適用することもできます(1行で記述してください)。

```
BUS="scsi",
PROGRAM="/sbin/udev.get_persistent_device_name.sh",
NAME="%k", SYMLINK="%c{1+}"
```

大容量ストレージデバイス用のドライバは、ロードされた直後に、使用可能なハードディスクすべてをカーネルに登録します。各ハードディスクはホットプラグブロックイベントをトリガし、各イベントはudevを呼び出します。次にudevはルールを読み取って、symlink(シンボリックリンク)を作成する必要があるかどうかを判断します。

initrdをとおしてドライバをロードした場合、ホットプラグイベントは失われます。しかし、すべての情報はsysfs内に格納されます。udevstartユーティリティは、/sys/blockおよび/sys/classの下にあるすべてのデバイスファイルを見つけ、udevを起動します。

ブートプロセスの間にすべてのデバイスノードを再作成する開始スクリプトboot.udevも存在します。しかし、開始スクリプトは、YaSTのランレベルエディタ、またはinsserv boot.udevコマンドを使用してアクティブにする必要があります。

**Tip**

/dev/sdaがSCSIハードディスクであること、および/dev/hdaがIDEディスクであることを前提としているツールとプログラムは多数存在しています。これらの前提が満たされていない場合、これらのプログラムは動作しません。YaSTはこれらのツールを使用しているので、カーネルのデバイス指定が実行されている場合にのみ、YaSTは動作します。

**Tip**

# Linuxのファイルシステム

Linuxは、互いに異なる多数のファイルシステムをサポートしています。この章では、非常に一般的なLinuxファイルシステムの短い概要を紹介し、それらの設計の概念、利点、および適用分野について説明します。Linux環境でのLFS (Large File Support、大規模ファイルのサポート)に関する詳細情報も説明します。

## 20.1 用語

**メタデータ(metadata)** ファイルシステムの内部にあるデータ構造で、ディスク上にあるすべてのデータが適切に編成され、アクセス可能であることを保証します。本質的には、「データに関するデータ」です。ほぼすべてのファイルシステムが、メタデータからなる独自の構造を採用していますが、各ファイルシステムが互いに異なるパフォーマンス特性を示すのは、それが1つの理由になっています。メタデータが破損しないよう維持するのは、非常に重要なことです。もし破損した場合、ファイルシステム内にあるすべてのデータがアクセス不能になる可能性があるからです。

**inode** inodeには、ファイルに関するさまざまな情報、たとえばサイズ、リンクの数、作成された日時、変更された日時、アクセスされた日時、およびファイルの内容を実際に格納しているディスクブロックへのポインタなどが記録されています。

**ジャーナル(journal)** ファイルシステムの用語では、ジャーナルとはディスク上に存在する構造であり、ファイルシステムのメタデータに対して加えられた変更を記録するためにファイルシステムが格納するさまざまなログを保持しています。ジャーナル機能は、Linuxシステムの回復時間を大幅に短縮します。システム起動時に、ファイルシステム全体をチェックする冗長な検索プロセスを不要にするからです。ただし、それはジャーナルが再現できる場合に限定されます。

## 20.2 Linuxの主要なファイルシステム

2、3年前とは異なり、Linuxシステムで使用するファイルシステムを選択するのは、数秒で済む問題(Ext2とReiserFSのどちらにするか)ではありません。カーネル2.4およびそれ以降では、さまざまなファイルシステムが選択できるようになりました。この後で、各ファイルシステムの基本的な動作原理、およびそれらが提供する利点の概要について説明します。

あらゆる用途で最適な単一のファイルシステムなど存在しない、ということを考慮しておくことが重要です。各ファイルシステムには特定の利点と欠点があり、それらを考慮する必要があります。最も洗練されたファイルシステムであっても、妥当なバックアップの方針を何か他の機能で置き換えることはできません。

この章で「データの完全性」および「データの一貫性」という用語が登場した場合、それらはユーザスペースのデータ(アプリケーションが自らのファイルに書き込むデータ)の一貫性を指していません(メタデータの一貫性を指します)。ユーザスペースのデータが一貫しているかどうかは、アプリケーション自体が管理する必要があります。

**Important**

**ファイルシステムのセットアップ**

この章では特に注記がない限り、パーティションやファイルシステムのセットアップまたは変更するために必要なステップすべては、YaSTのモジュールを使用して実行できます。

**Important**

## 20.2.1 ReiserFS

ReiserFSは、公式にはカーネル2.4リリース時の主要な機能の1つですが、SUSE LINUXバージョン6.4以降で、ReiserFSはSUSE 2.2.xカーネルに対するカーネルパッチとして使用可能でした。ReiserFSは、Hans Reiser(ハンス・ライザー)とNamesys社の開発チームによって設計されました。ReiserFSは、古いExt2に代わる強力な選択肢であることを実証してきました。その主要な利点は、より良いディスクスペース使用効率、より良いディスクアクセスパフォーマンス、およびより高速なクラッシュ回復機能です。しかし、小さな欠点もあります。ReiserFSはメタデータに強い注意を払いますが、データ自体に注意を払いません。ReiserFSの将来の世代は、データジャーナル処理(メタデータと実際のデータの両方をジャーナルに書き込む)と、順序付け書き込みをサポートする予定です。

ReiserFSの利点をより詳細に記述すると、以下のようになります。

**より良いディスクスペース使用効率** ReiserFSでは、すべてのデータは、$B^*$-Tree(バランスドツリー)と呼ばれる構造内で編成されています。このツリー構造は、より良いディスクスペース使用効率に貢献しています。小さなファイルは、$B^*$-Treeのリーフノードに直接格納されるからです。そのようなファイルをどこか他の場所に格納して、ディスク上の実際の場所を指すポインタを維持するより優れています。それに加えて、ストレージ(記憶領域)は1KBまたは4KBのチャンク単位で割り当てられるのではなく、実際に必要なサイズの構成部分(エクステント)を割り当てられます。もう1つの利点は、inodeの動的割り当てに関係しています。

これは、ファイルシステムの作成時にinodeの密度を指定する必要のある、Ext2のような従来のファイルシステムに比べて、ファイルシステムの柔軟性を高めます。

**より良いディスクアクセスパフォーマンス**
小規模なファイルでは、多くの場合、ファイルのデータと"stat_data" (inode)情報が互いに隣り合って保存されます。これらは1回のディスクI/O操作で読み取れるので、ただ1回のディスクアクセスで、必要な情報すべてを取得できることを意味します。

**高速なクラッシュ回復機能** ジャーナルを使用して、メタデータに加えられた最新の変更結果を記録しているので、ファイルシステムが大規模な場合を含め、ファイルシステムを数秒でチェックできます。

**データジャーナリングによる信頼性** ReiserFSは、Ext3のセクション項20.2.3. 「Ext3」で説明した概念に類似したデータジャーナリングおよび順序データモードをサポートしています。デフォルトのモードは、`data=ordered`です。このモードでは、データとメタデータの完全性は保証されますが、メタデータのジャーナリングだけが行われます。

### 20.2.2 Ext2

Ext2の起源は、Linuxの歴史の初期にさかのぼります。その前身であったExtended File Systemは、1992年4月に実装され、Linux 0.96cに統合されました。Extended File Systemは多くの変更を加えられ、Ext2として数年にわたって、最も人気のあるLinuxファイルシステムになりました。その後、他のジャーナルファイルシステムが作成され、非常に短い回復時間を達成したため、Ext2の重要性は低下しました。

Ext2の利点に関する短い要約を読むと、かつて幅広く好まれ、そして今でも一部の分野で多くのLinuxユーザから好まれるLinuxファイルシステムである理由を理解するのに役立ちます。

**堅牢性** 「古くからある標準」として、Ext2は過去に多くの改良を受け、集中的にテストされてきました。このような理由で、多くの人はExt2を岩のように堅牢(rock-solid)と呼びます。ファイルシステムが正常にアンマウントできず、システムが機能停止した場合、e2fsckはファイルシステムのデータの分析を開始します。メタデータは一貫した状態に戻り、保留されていたファイルとデータブロックは、指定のディレクト

リ(lost+foundという名前)に書き込まれます。ジャーナルファイルシステムとは対照的に、e2fsckは、最近変更されたわずかなメタデータだけではなく、ファイルシステム全体を分析します。この結果、ジャーナルファイルシステムがログデータだけをチェックするのに比べて、かなり長い時間を要します。ファイルシステムのサイズにもよりますが、この手順は30分またはそれ以上を要することがあります。したがって、高可用性を必要とするどのようなサーバでも、Ext2を選択することは望ましくありません。ただし、Ext2はジャーナルを維持せず、非常にわずかなメモリを使用するだけなので、時には他のファイルシステムより高速なことがあります。

**容易なアップグレード性** Ext2のコードは、Ext3が次世代ファイルシステムであることを明確に主張するための強力な土台になりました。Ext2の信頼性および堅牢性が、ジャーナルファイルシステムの利点と見事に融合されました。

### 20.2.3 Ext3

Ext3は、Stephen Tweedieによって設計されました。他のすべての次世代ファイルシステムとは異なり、Ext3は完全に新しい設計理念に基づいているわけではありません。Ext3は、Ext2をベースとしています。これら2つのファイルシステムは、互いに非常に似通っています。Ext3ファイルシステムを、Ext2ファイルシステムの上に構築することも容易です。Ext2とExt3の間にある最も重要な違いは、Ext3がジャーナルをサポートしていることです。要約すると、Ext3には、次の3つの主要な利点があります。

**Ext2からの容易で信頼性の高いアップグレード**
Ext3はExt2のコードをベースとし、ディスクフォーマットとメタデータフォーマットが共通しているので、Ext2からExt3へのアップグレードは非常に容易です。ReiserFS、JFS、またはXFSのような他のファイルシステムへの移行はかなり手間がかかります(ファイルシステム全体のバックアップを作成し、移行先ファイルシステムを新規に作成する必要があります)が、それとは異なり、Ext3への移行は数分で完了します。ファイルシステム全体を新規に作成する作業は障害なしで完了するとは限りませんが、Ext3への移行にはそのような作業が伴わないので、非常に安全でもあります。ジャーナルファイルシステムへのアップグレードを待つ既存のExt2システムの数を考慮すると、多くのシステム管理者にとってExt3が重要な選択肢になっていることが容易に想像できるはずです。Ext3からExt2へのダウングレードも、アップグレードと同じほど容

易です。Ext3ファイルシステムのアンマウントを正常に行い、Ext2ファイルシステムとして再マウントするだけです。

**信頼性とパフォーマンス** 他のジャーナルファイルシステムは、「メタデータのみ」のジャーナルアプローチに従っています。これは、使用中のメタデータは常に一貫した状態を維持されていますが、ファイルシステムのデータ自体に関しては同じことが自動的に保証されるわけではない、という意味です。Ext3は、メタデータとデータの両方に注意するよう設計されています。「注意」の度合いはカスタマイズできます。Ext3の`data=journal`モードを有効にした場合、最大の保護(データの完全性)を実現しますが、メタデータとデータの両方がジャーナル化されるので、システムの動作が遅くなります。比較的新しいアプローチは、`data=ordered`モードを使用することです。これは、データとメタデータ両方の完全性を保証しますが、ジャーナルを適用するのはメタデータのみです。ファイルシステムドライバは、1つのメタデータの更新に対応するすべてのデータブロックを収集します。これらのブロックは、メタデータの更新前にディスクに書き込まれます。その結果、パフォーマンスを犠牲にすることなく、メタデータとデータの両方に関する一貫性を達成できます。3番目のオプションは、`data=writeback`を使用することです。これは、対応するメタデータをジャーナルにコミットした後で、データをメインファイルシステムに書き込むことを可能にします。多くの場合、このオプションは、パフォーマンスの点で最善と考えられています。しかし、内部のファイルシステムの完全性が維持される一方で、クラッシュと回復を実施した後では、古いデータがファイル内に再登場することを許してしまう可能性があります。管理者が他のオプションを指定しない限り、Ext3はデフォルトで`data=ordered`を使用して動作します。

### 20.2.4 Ext2ファイルシステムからExt3への変換

Ext2からExt3への変換には、2つの個別のステップが関係しています。

**ジャーナルの作成** `root`としてログインし、`tune2fs -j`を実行します。この結果、デフォルトのパラメータを使用してExt3ジャーナルが作成されます。ジャーナルの大きさや、どのデバイスにジャーナルを配置するかを自分で決定するには、代わりに`tune2fs -J`を実行し、希望のジャーナルオプションである`size=`および`device=`を指定します。tune2fsプログラムの詳細については、そのmanページ(tune2fs(8))を参照してください。

**/etc/fstab内でのファイルシステムタイプの指定**
Ext3ファイルシステムがExt3として正しく認識されることを保証するために、/etc/fstabファイルを編集し、対応するパーティションに対して指定されているファイルシステムタイプをext2からext3に変更します。この変更結果は、次回の再起動後に有効になります。

**ルートディレクトリとしてのExt3の使用**
Ext3パーティションとしてセットアップされたファイルシステムからブートするには、ext3とjbdの各モジュールをinitrd内に含めます。この作業を行うには、/etc/sysconfig/kernelファイルを編集し、INITRD_MODULESの下でこれら2つのモジュールを記述し、mk_initrdコマンドを実行します。

### 20.2.5 Reiser4

カーネル2.6のリリース直後に、ジャーナリングファイルシステムのファミリには、もう1つのメンバーReiser4が追加されました。Reiser4は、基本的にその前任のReiserFS(バージョン3.6)と異なります。Reiser4により、ファイルシステム機能を調整するためのプラグインの概念、およびきめ細かなセキュリティの概念が導入されます。

**きめ細かなセキュリティの概念** Reiser4を設計する際に、開発者は、セキュリティ関連機能の実装に重点を置きました。したがって、Reiser4には、専用のセキュリティプラグインセットが付属しています。最も重要なプラグインには、ファイル「項目」の概念が導入されています。現在、ファイルアクセス制御はファイル単位に定義されています。複数のユーザ、グループ、またはアプリケーションに関連する情報を格納している大きなファイルがある場合、すべての関係者を含めるには、アクセス権が不明確でした。Reiser4では、そのような大きいファイルをより小さい部分(「項目」)に分割できます。次に、項目ごとおよびユーザごとにアクセス権を個別に設定できるため、ファイルのセキュリティをはるかに正確に管理できます。ちょうどいい例は、/etc/passwdです。今のところ、rootだけはファイルを読み取りおよび編集できますが、root以外のユーザはこのファイルに読み取り専用でしかアクセスできません。Reiser4の項目の概念を取り入れると、このファイルを複数の項目(ユーザにつき1項目)に分割できるため、ユーザやアプリケーションがそれぞれのデータを変更できます。ただし、他のユーザのデータにはアクセスできません。この概念では、セキュリティと柔軟性のどちらも付加されます。

**プラグインによる拡張性** ファイルシステムで通常使用される多くのファイルシステム関数と外部関数は、Reiser4ではプラグインとして実装されます。このようなプラグインは、ベースシステムに簡単に追加できます。そのため、ファイルシステムに新しい機能を追加するために、カーネルを再コンパイルしたり、ハードディスクを再フォーマットしたりする必要はなくなりました。

**遅延アロケーションによる優れたファイルシステムレイアウト** XFSと同様に、Reiser4では遅延アロケーションをサポートしています。項20.2.7.「XFS」を参照してください。メタデータにも遅延アロケーションを使用すれば、レイアウト全体がさらに優れたものになる可能性があります。

### 20.2.6 JFS

JFSは、*Journaling File System*(ジャーナルファイルシステム)の略で、IBMが開発したものです。Linuxに移植されたJFSの最初のベータ版は、2000年夏にLinuxコミュニティに対して提供されました。バージョン1.0.0は、2001年にリリースされました。JFSは、パフォーマンスを最重要な目標とする高スループットサーバ環境のニーズを満たすようカスタマイズされています。完全64ビットファイルシステムとして、JFSは大規模なファイルと大規模なパーティションの両方をサポートしています。これは、JFSがサーバ環境で使用されるもう1つの理由です。

JFSを詳細に観察すると、このファイルシステムがLinuxサーバにとって適切な選択肢になる理由を知ることができます。

**効率的なジャーナル処理** JFSは、「メタデータのみ」のアプローチに従っています。包括的なチェックではなく、ファイルシステムに関する最近の操作によって発生したメタデータのみの変更結果をチェックしますが、それは回復時の時間を大幅に節約します。並列操作は、複数の並列ログエントリを必要としますが、それらの操作は結合されて1つのグループコミットになり、複数の書き込み操作に伴うファイルシステムのパフォーマンス低下を大幅に低減します。

**効率的なディレクトリ編成** JFSは、2つの異なるディレクトリ編成を採用しています。小規模なディレクトリに対しては、ディレクトリの内容をinodeに直接格納することを許可します。大規模なディレクトリに対しては、$B^+$-Treeを使用し、優れたディレクトリ管理を行います。

**inodeの動的割り当てによるより良いスペース使用効率**

Ext2の場合、inodeの密度(管理情報が占有するスペース)を事前に定義する必要があります。これは、ファイルシステムに記録できるファイルまたはディレクトリの最大数を限定します。JFSは、このような事前の考慮を不要にします。JFSはinodeのスペースを動的に割り当て、必要がなくなった場合はそれらを解放します。

### 20.2.7 XFS

本来は、IRIX OS用のファイルシステムを意図してSGIがXFSの開発を開始したのは、1990年代初期です。XFSの背後にある考えは、ハイパフォーマンスの64ビットジャーナルファイルシステムを作成し、非常に要求の多い今日のコンピューティングの課題を満たすことです。XFSは大規模なファイルを操作する点で非常に優れていて、ハイエンドのハードウェアを適切に活用します。しかし、XFSには1つの欠点があります。ReiserFSと同様、XFSはメタデータの完全性には最大の注意を払いますが、データの完全性にはそれほど注意を払いません。

XFSの主要な機能を簡単に観察することにより、ハイエンドのコンピューティング分野で、XFSが他のジャーナルファイルシステムの強力な競合相手という立場を実証している理由を説明できます。

**アロケーショングループの採用による高いスケーラビリティ**

XFSファイルシステムの作成時に、ファイルシステムの基にあるブロックデバイスは、等しいサイズを持つ8つ以上の線形の領域に分割されます。これらをアロケーショングループと呼びます。各アロケーショングループは、独自のinodeと空きディスクスペースを管理します。実用的には、アロケーショングループを、1つのファイルシステムの中にある複数のファイルシステムと見なすこともできます。アロケーショングループは互いに独立しているのではなく、カーネルから複数を同時にアドレス指定できる、という特徴があります。この特徴は、XFSの高いスケーラビリティの鍵です。独立性の高いアロケーショングループは、性質上、マルチプロセッサシステムのニーズに適しています。

**ディスクスペースの効率的な管理によるハイパフォーマンス**

空きスペースとinodeは、各アロケーショングループ内の$B^+$-Treeによって処理されます。$B^+$-Treeの採用は、XFSのパフォーマンスとスケーラビリティに大きく貢献しています。XFSでは、*遅延アロケーション*を採用しています。XFSはアロケーション(割り当て)を2つのパートに分割し

て、この操作を処理します。保留されているトランザクションはRAMの中に保存され、適切な量のスペースが確保されます。XFSはこの時点では、データの格納場所(言い換えると、ファイルシステムのどのブロックか)を決定しません。決定可能な最後の瞬間まで、この決定は遅延(先送り)されます。短時間だけ存続する一部の一時データは、ディスクに書き込まれません。XFSがデータの実際の保存場所を決定する時点で、それらのデータは不要になっているからです。したがって、XFSは書き込みのパフォーマンスを向上させ、ファイルシステムの断片化(フラグメンテーション)を減らします。遅延アロケーションは、他のファイルシステムより書き込みイベントの頻度を下げる結果をもたらすので、書き込み中にクラッシュが発生した場合、データ損失が深刻になる可能性が高くなります。

**事前割り当てによるファイルシステムの断片化の回避**
データをファイルシステムに書き込む前に、XFSはファイルが必要とする空きスペースを予約(プリアロケート、事前割り当て)します。したがって、ファイルシステムの断片化は大幅に減少します。ファイルの内容がファイルシステム全体に分散することがないので、パフォーマンスが向上します。

## 20.3 サポートされている他のいくつかのファイルシステム

表 20.1. 「Linux環境でのファイルシステムのタイプ」は、Linuxがサポートしている他のいくつかのファイルシステムを要約したものです。これらは主に、他の種類のメディアや外部オペレーティングシステムとの互換性およびデータの相互交換を保証することを目的としてサポートされています。

***Table 20.1:*** *Linux環境でのファイルシステムのタイプ*

| | |
|---|---|
| `cramfs` | *Compressed ROM file system* (圧縮ROMファイルシステム): ROM用に圧縮された読み込み専用ファイルシステムです。 |
| `hpfs` | *High Performance File System* (ハイパフォーマンスファイルシステム): IBM OS/2の標準ファイルシステムです。読み取り専用モードでサポートされています。 |

| | |
|---|---|
| `iso9660` | CD-ROMの標準ファイルシステム。 |
| `minix` | このファイルシステムは、オペレーティングシステムに関する学術的なプロジェクトを起源とするもので、Linuxで最初に使用されたファイルシステムです。現在では、フロッピーディスク用のファイルシステムとして使用されています。 |
| `msdos` | *fat*、つまり当初はDOSで使用されていたファイルシステムであり、現在はさまざまなオペレーティングシステムで使用されています。 |
| `ncpfs` | Novellのボリュームをネットワーク経由でマウントするためのファイルシステム。 |
| `nfs` | *Network File System* (ネットワークファイルシステム): この場合、ネットワーク内にある任意のコンピュータにデータを格納でき、ネットワーク経由でアクセスを許可できるファイルシステムを指します。 |
| `smbfs` | *Server Message Block* (サーバメッセージブロック): Windowsのような製品が、ネットワーク経由でのファイルアクセスを可能にする目的で採用しています。 |
| `sysv` | SCO UNIX、Xenix、およびCoherent (PC用の商用UNIXシステム)が採用。 |
| `ufs` | BSD、SunOS、およびNeXTstepが採用。読み取り専用モードでサポートされています。 |
| `umsdos` | *UNIX on MSDOS* (MSDOS上のUNIX): 通常の`fat`ファイルシステムに対して適用されるもので、特別なファイルを作成することにより、UNIXの機能(パーミッション、リンク、長いファイル名)を実現します。 |
| `vfat` | *Virtual FAT* (仮想FAT): `fat`ファイルシステムを拡張したものです(長いファイル名をサポートします)。 |
| `ntfs` | *Windows NT file system* (Windows NTファイルシステム)、読み取り専用です。 |

## 20.4 Linux環境での大規模ファイルサポート

当初、Linuxは最大2GBのファイルサイズをサポートしていました。マルチメディアが爆発的に普及する前、およびLinux環境で大規模データベースを運用することを誰も試みていないうちは、これで十分でした。サーバコンピューティングの重要性がますます高くなるにつれて、カーネルとCライブラリが変更され、2GBを超えるファイルサイズをサポートするようになりました。現在、ほぼすべてのファイルシステムはLFS (Large File Support、大規模ファイルサポート)に対応しているので、管理者やユーザはハイエンドのコンピューティングを実現できます。 表 20.2. 「ファイルシステムの最大サイズ(ディスクフォーマット時)」は、Linuxのファイルとファイルシステムに関する現在の制限の概要を示しています。

*Table 20.2: ファイルシステムの最大サイズ(ディスクフォーマット時)*

| ファイルシステム | ファイルサイズ(バイト) | ファイルシステムのサイズ(バイト) |
|---|---|---|
| Ext2またはExt3 (ブロックサイズ1KB) | $2^{34}$ (16GB) | $2^{41}$ (2TB) |
| Ext2またはExt3 (ブロックサイズ2KB) | $2^{38}$ (256GB) | $2^{43}$ (8TB) |
| Ext2またはExt3 (ブロックサイズ4KB) | $2^{41}$ (2TB) | $2^{44}$ (16TB) |
| Ext2またはExt3 (ブロックサイズ8KB) (Alphaのような8KBページ採用のシステム) | $2^{46}$ (64TB) | $2^{45}$ (32TB) |
| ReiserFS v3 | $2^{46}$ (64GB) | $2^{45}$ (32TB) |
| XFS | $2^{63}$ (8EB) | $2^{63}$ (8EB) |
| JFS (ブロックサイズ512バイト) | $2^{63}$ (8EB) | $2^{49}$ (512TB) |
| JFS (ブロックサイズ4Kバイト) | $2^{63}$ (8EB) | $2^{52}$ (4PB) |
| NFSv2 (クライアント側) | $2^{31}$ (2GB) | $2^{63}$ (8EB) |

| | | |
|---|---|---|
| NFSv3 (クライアント側) | $2^{63}$ (8EB) | $2^{63}$ (8EB) |

**Important**

**Linuxカーネルの制限**

表 20.2.「ファイルシステムの最大サイズ(ディスクフォーマット時)」は、ディスクフォーマット時の制限について記載しています。カーネル2.6は、自らが取り扱うファイルとファイルシステムのサイズについて、独自の制限を課しています。それらは、次のとおりです。

**ファイルのサイズ** 32ビットシステムでは、ファイルは2TB ($2^{41}$バイト)のサイズを上回ることができません。

**ファイルシステムのサイズ** ファイルシステムは最大$2^{73}$バイトの大きさまでサポートします。しかし、この制限は、現在使用可能なハードウェアが到達可能な範囲を上回っています。

**Important**

## 20.5 詳細情報

ここまでに説明した各ファイルシステムのプロジェクトには、独自のWebページがあります。そこで詳しいドキュメントとFAQ、さらにメーリングリストを参照することができます。

- `http://e2fsprogs.sourceforge.net/`
- `http://www.zipworld.com.au/~akpm/linux/ext3/`
- `http://www.namesys.com/`
- `http://oss.software.ibm.com/developerworks/opensource/jfs/`
- `http://oss.sgi.com/projects/xfs/`

Linuxのファイルシステムに関する包括的で複数のパートからなるチュートリアルは、*IBM developerWorks*のWebページ`http://www-106.ibm.com/developerworks/library/l-fs.html`から入手できます。Linux環境のさまざまなジャーナルファイルシステムに関する比較は、Juan I. Santos Floridoによる*Linuxgazette*のWebページ`http://www.linuxgazette.com/issue55/florido.html`を参照してください。Linux環境のLFSに関する興味深い詳しい分析については、Andreas JaegerによるLFS in Linuxサイト`http://www.suse.de/~aj/linux_lfs.html`を参照してください。

# PAMを使用した認証

Linuxは、ユーザとアプリケーションを仲介するレイヤとして認証プロセスでPAM (Pluggable Authentication Modules)を使用します。PAMモジュールはシステム単位で使用できるため、どのアプリケーションからもリクエストできます。この章では、モジュラー認証メカニズムの機能とその設定方法について説明します。

通常、システム管理者とプログラマは、システムの一定部分へのアクセスを制限することや、アプリケーションの一定の機能の使用を制限することを望みます。PAMを使用しなければ、新規の認証メカニズム(LDAPやSAMBAなど)が導入されるたびにアプリケーションを調整する必要があります。ただし、このプロセスには時間がかかり、ミスが発生する可能性があります。このような難点を回避する方法の1つは、アプリケーションを認証メカニズムから切り離し、残りは集中管理されるモジュールに任せることです。新しい認証方式が必要になった場合は、問題のプログラムで使用できるように適切なPAMモジュールを調整または記述するだけで済みます。

PAMメカニズムに依存するすべてのプログラムについて、ディレクトリ/etc/pam.d/<programname>に専用の設定ファイルがあります。これらのファイルでは、認証に使用するPAMモジュールが定義されます。また/etc/securityにはほとんどのPAMモジュール用のグローバル設定ファイルがあり、これらのモジュール(pam_env.conf、pam_pwcheck.conf、pam_unix2.conf、time.confなど)の正確な動作が定義されます。PAMモジュールを使用する各アプリケーションは、実際には一連のPAM関数を呼び出し、各PAM関数は各種設定ファイルの情報を処理して、その結果を呼び出し元のアプリケーションに戻します。

## 21.1 PAM設定ファイルの構造

PAM設定ファイルの各行は、次のように最大4列で構成されています。

<モジュールのタイプ> <制御フラグ> <モジュールパス> <オプション>

PAMモジュールはスタックとして処理されます。モジュールの用途はタイプごとに異なり、パスワードをチェックするモジュール、システムのアクセス元ロケーションを検証するモジュール、ユーザ固有の設定を読み込むモジュールなどがあります。PAMは、次の4タイプのモジュールを認識します。

**auth** このタイプのモジュールの目的は、ユーザの信憑性をチェックすることです。従来、このチェックのためにパスワードの問い合わせが行われていましたが、チップカードやバイオメトリクス(指紋や虹彩のスキャン)の助けを借りて行うこともできます。

**account** このタイプのモジュールは、ユーザがリクエストしたサービスを使用するための一般許可を付与されているかどうかをチェックします。たとえば、失効したアカウントのユーザ名では誰もログインできないようにするには、この種のチェックを実行する必要があります。

**password** このタイプのモジュールの目的は、認証トークンを変更可能にすることです。ほとんどの場合、このトークンはパスワードです。

**session** このタイプのモジュールは、ユーザセッションの管理と設定を受け持ちます。認証の前後に起動され、ログイン試行をシステムログに記録し、ユーザ固有の環境(メールアカウント、ホームディレクトリ、システム制限など)を設定します。

2列目には、起動されたモジュールの動作に影響する制御フラグが含まれています。

**required** このフラグが付いているモジュールは、認証を進める前に正常に処理される必要があります。requiredフラグが付いたモジュールが失敗した後、同じフラグが付いた他のモジュールがすべて処理されてから、ユーザが認証試行の失敗メッセージを受け取ります。

**requisite** このフラグが付いているモジュールも、requiredフラグが付いている場合とほぼ同様に、正常に処理される必要があります。ただし、このフラグが付いたモジュールが失敗した場合は、ユーザに即座にフィードバックが送られ、他のモジュールは処理されません。成功すると、requiredフラグが付いているモジュールの場合とほぼ同様に、続いて他のモジュールが処理されます。requisiteフラグは、正しい認証に不可欠な一定条件の有無をチェックするための基本フィルタとして使用できます。

**sufficient** このフラグが付いたモジュールが正常に処理されると、呼び出し元アプリケーションは即時に成功メッセージを受け取り、前にrequiredフラグが付いたモジュールが失敗していなければ、他のモジュールは処理されません。sufficientフラグが付いたモジュールが失敗しても、直接的な結果は発生せず、以降のモジュールはそれぞれの順序で処理されます。

**optional** このフラグが付いたモジュールが成功しても失敗しても、直接的な影響はありません。このフラグは、それ以上はアクションを実行しない、メッセージ表示(ユーザへのメール着信通知など)専用のモジュールに便利です。

**include** このフラグが設定された場合、引数として指定されたファイルがこの場所に挿入されます。

モジュールがデフォルトディレクトリ/lib/securityにあれば、そのパスを明示的に指定する必要はありません(SUSE LINUXでサポートされるす

べての64ビットプラットフォームの場合、このディレクトリは/lib64/securityです)。4列目には、debug(デバッグの有効化)やnullok(空のパスワードの使用を許可)など、対応するモジュール用のオプションが表示される場合があります。

## 21.2 sshdのPAM設定

PAMの裏付けとなっている理論の機能を示すために、実務的な例としてsshdのPAM設定を考えてみましょう。

***Example 21.1:** sshdのPAM設 定*

```
#%PAM-1.0
auth     include        common-auth
auth     required       pam_nologin.so
account  include        common-account
password include        common-password
session  include        common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README.SuSE)
#session  optional      pam_resmgr.so fake_ttyname
```

アプリケーション(この場合sshd)の通常のPAM設定には、次に示す4つのモジュールタイプの設定ファイルを参照するinclude文が含まれます。common-auth、common-account、common-password、およびcommon-session。これら4つのファイルにはそれぞれのモジュールタイプ用のデフォルト設定があります。各PAMアプリケーションごとにそれぞれのモジュールを個別に呼び出す代わりとしてこれらを組み込むことで、管理者がデフォルトを変更した場合、更新されたPAM設定を自動的に取得します。これまでPAMへの変更があった場合、または新規アプリケーションをインストールした場合には、すべてのアプリケーションの全設定ファイルを手動で調整しなければなりませんでした。PAM設定とPAMで行われたすべての変更は、デフォルトの構成ファイルを使用して継承されました。

最初のincludeファイル(common-auth)はauthタイプの2つのモジュール、pam_envおよびpam_unix2を呼び出します。詳細については、例 21.2.「authセクションのデフォルト設定」を参照してください。

***Example 21.2:** authセクションのデフォルト設定*

```
auth    required        pam_env.so
auth    required        pam_unix2.so
```

1つ目はpam_envで、ファイル/etc/security/pam_env.confをロードし、このファイルに指定されている環境変数を設定します。pam_envモジュールはログイン元ロケーションを認識するため、このファイルを使用するとDISPLAY変数を適切な値に設定できます。2つ目のpam_unix2は、ユーザのログインとパスワードを/etc/passwdおよび/etc/shadowと比較対照してチェックします。

common-authで指定されたモジュールが正常に呼び出された後、pam_nologinという3番目のモジュールがファイル/etc/nologinの存在する場所をチェックします。このファイルが存在する場合、root以外のユーザはログインできません。authモジュールのスタック全体が処理された後に、sshdがログインの成否に関するフィードバックを取得します。スタックの全モジュールにrequired制御フラグが付いている場合は、すべてが正常に処理されなければ、sshdには成功メッセージが送られません。モジュールが1つでも失敗すると、モジュールスタック全体が処理され、その後にのみsshdに失敗が通知されます。

authタイプのすべてのモジュールが正常に処理された時点で、別のinclude文が処理されます。この例では例 21.3. 「accountセクションのデフォルト設定」になります。common-accountに含まれるモジュールはpam_unix2のみです。pam_unix2からユーザが存在するという結果が戻されると、sshdは成功したことを通知するメッセージを受信し、モジュールの次のスタック(password)が処理されます。この処理を例 21.4. 「passwordセクションのデフォルト設定」に示します。

*Example 21.3: accountセクションのデフォルト設定*

```
account required        pam_unix2.so
```

*Example 21.4: passwordセクションのデフォルト設定*

```
password required       pam_pwcheck.so  nullok
password required       pam_unix2.so    nullok use_first_pass use_authtok
#password required      pam_make.so     /var/yp
```

繰り返しになりますが、sshdのPAM設定はcommon-passwordにあるpasswordモジュールのデフォルト設定を参照する1つのinclude文にのみ関係します。アプリケーションが認証トークンの変更をリクエストするたびに、これらのモジュールを正常に完了する必要があります。(制御フラグrequired)。パスワード変更や別の認証トークンについてはセキュリティチェックが必要です。これはpam_pwcheckモジュールで実現可能です。その後に使用されたpam_unix2モジュールがpam_pwcheckから新旧のパスワードを引き継ぐため、ユーザが再認証する必要はありません。また、これでpam_pwcheckによるチェックを回避することもできなくなります。accountまたはauthタイプのモジュールが期限切れパスワードに関するメッセージを送るように設定されている場合は、passwordタイプのモジュールを使用する必要があります。

*Example 21.5: sessionセクションのデフォルト設定*

```
session required        pam_limits.so
session required        pam_unix2.so
```

最終ステップとして、common-sessionに組み込まれたsessionタイプのモジュールが呼び出され、問題のユーザ用の設定に従ってセッションが設定されます。pam_unix2が再び処理されますが、このモジュール、pam_unix2.confが関連する設定ファイルにnoneオプションが指定されているため、実際の結果はありません。pam_limitsモジュールはファイル/etc/security/limits.confをロードします。このファイルでは、特定のシステムリソースの使用制限が定義されている場合があります。sessionモジュールはユーザのログアウト時に再び呼び出されます。

## 21.3 PAMモジュールの設定

PAMモジュールの一部は設定可能です。対応する設定ファイルは/etc/securityにあります。この項では、sshdの例(pam_unix2.conf、pam_env.conf、pam_pwcheck.confおよびlimits.conf)について簡単に説明します。

### 21.3.1 pam_unix2.conf

従来のパスワードベースの認証方式は、PAMモジュールpam_unix2によって制御されます。このモジュールは、必要なデータを/etc/passwd、/etc/shadow、NISマップ、NIS+テーブル、またはLDAPデータベースから読み込むことができます。このモジュールの動作は、アプリケーション自体のPAMオプションを設定して個別に変更するか、/etc/security/pam_unix2.confを編集してグローバルに変更できます。例 21.6. 「pam_unix2.conf」に、このモジュールの最も基本的な設定ファイルを示します。

***Example 21.6:** pam_unix2.conf*

```
auth:nullok
account:
password:nullok
session:none
```

モジュールタイプauthおよびpasswordのnullokオプションは、対応するタイプのアカウントに空のパスワードを許可するように指定します。また、ユーザは自分のアカウントのパスワード変更を許可されます。モジュールタイプsessionのnoneオプションは、代わりにメッセージが記録されないように指定します(デフォルト)。その他の設定オプションの詳細については、ファイル自体のコメントまたはpam_unix2のマニュアルページを参照してください。

### 21.3.2 pam_env.conf

このファイルを使用すると、pam_envモジュールが呼び出されるたびに設定される、ユーザ用に標準化された環境を定義できます。それにより、次の構文を使用して環境変数を事前設定できます。

```
VARIABLE  [DEFAULT=[value]]  [OVERRIDE=[value]]
```

**VARIABLE** 設定する環境変数の名前です。

**[DEFAULT=[value]]** 管理者が設定するデフォルト値です。

**[OVERRIDE=[value]]** 問い合わせ可能でpam_envによって設定される値です。この値でデフォルト値が上書きされます。

pam_envによるデフォルトの上書きが必要となる一般的な例がDISPLAY変数です。この変数は、リモートログインが発生するたびに変更されます。例 21.7. 「pam_env.conf」を参照してください。

***Example 21.7:*** *pam_env.conf*

```
REMOTEHOST     DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY        DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

1行目では、REMOTEHOST変数の値がlocalhostに設定されており、pam_envが他の値を判別できない場合にこの値が使用されます。DISPLAY変数には、REMOTEHOSTの値が含まれています。ファイル/etc/security/pam_env.confでより詳細な情報を参照してください。

### 21.3.3 pam_pwcheck.conf

この設定ファイルから、pam_pwcheckモジュールがpasswordタイプの全モジュールのオプションを読み込みます。このファイルに格納されている設定は、個々のアプリケーションのPAM設定よりも優先されます。アプリケーション固有の設定が定義されていない場合、アプリケーションではグローバル設定が使用されます。例 21.8. 「pam_pwcheck.conf」はpam_pwcheckに対して空のパスワードとパスワード変更を許可するように指示しています。このモジュールの他のオプションについては、ファイル/etc/security/pam_pwcheck.confを参照してください。

***Example 21.8:*** *pam_pwcheck.conf*

```
password:nullok
```

### 21.3.4 limits.conf

ファイルlimits.confでは、ユーザ別またはグループ別のシステム制限を設定できます。このファイルは、pam_limitsモジュールで読み込まれます。このファイルを使用すると、絶対に超過できない厳密な制限と、一時的な超過が許される緩やかな制限を設定できます。構文および使用可能なオプションの詳細については、ファイルに含まれているコメントを参照してください。

## 21.4 関連資料

インストール済みシステムのディレクトリ/usr/share/doc/packages/pamには、次のドキュメントが用意されています。

**READMEs** このディレクトリの最上位レベルには、一般的なREADMEファイルがいくつか入っています。サブディレクトリmodulesには、使用可能なPAMモジュールのREADMEファイルがあります。

**『Linux-PAM System Administrators' Guide』**
このマニュアルには、システム管理者を対象としたPAMに関する必須情報がすべて含まれています。設定ファイルの構文からPAMのセキュリティ面に至るまで、広範囲な項目を説明しています。このマニュアルは、PDFファイル、HTML形式およびプレーンテキストで提供されます。

**『Linux-PAM Module Writers' Manual』**
このマニュアルには、開発者を対象として標準準拠のPAMモジュールを記述する方法の概要が記載されています。このマニュアルは、PDFファイル、HTML形式およびプレーンテキストで提供されます。

**『The Linux-PAM Application Developers' Guide』**
このマニュアルには、PAMライブラリを使用するアプリケーション開発者に必要な情報がすべて含まれています。このマニュアルは、PDFファイル、HTML形式およびプレーンテキストで提供されます。

Thorsten KukukはSUSE LINUX用のPAMモジュールを多数開発しており、その一部の情報はhttp://www.suse.de/~kukuk/pam/で公開されています。

# Part III

# サ ー ビ ス

# 22



# ネットワークの基礎

Linuxはまさにインターネットの申し子であり、あらゆる種類のネットワーク構造を統合するために必要なネットワークツールと機能をすべて備えています。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、基本的なメカニズムと関連のネットワークの環境設定ファイルのみを扱います。

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 22.1.「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」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全体のことを“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。RFC文書は、`http://www.ietf.org/rfc.html`で参照してください。

***Table 22.1:*** *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 | 転送制御プロトコル。接続指向の安全なプロトコルです。転送されるデータはまずアプリケーションによってデータのストリームとして送信され、次にオペレーティングシステムによって適切な形式に変換されます。データは、それが送信されたときの元のデータ形式で、宛先ホストのそれぞれのアプリケーションに到着します。TCPは、伝送中にデータに損失がなかったか、データの混同がないかどうかを確認します。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 | ユーザデータグラムプロトコル。コネクションレスの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もあります。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 | インターネット制御メッセージプロトコル。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 | インターネットグループ管理プロトコル。このプロトコルは、IPマルチキャストを実装している場合に、マシンの動作を制御します。 |

図 22.1. 「TCP/IPの簡易レイヤモデル」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。実際の通信レイヤは、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。

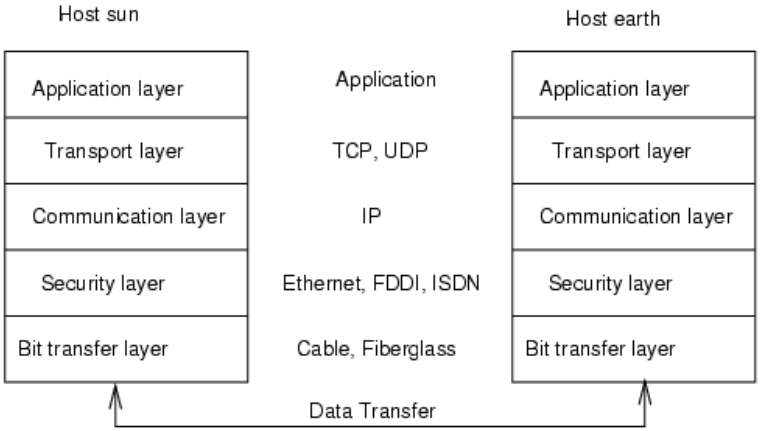


***Figure 22.1:*** *TCP/IPの簡易レイヤモデル*

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。この図からわかるように、各レイヤは、*抽象化レベル*に従った順序で並んでいます。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。ビット転送レイヤとセキュリティレイヤは、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 22.2. 「TCP/IPイーサネットパケット」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。

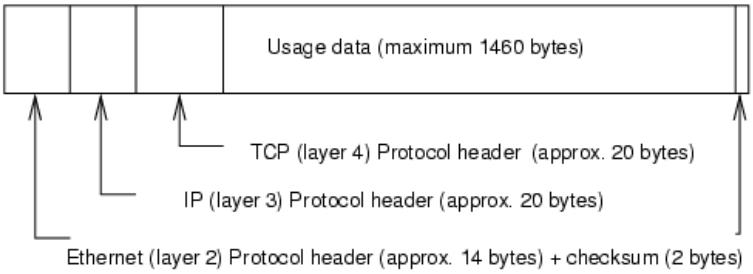


***Figure 22.2:*** *TCP/IPイーサネットパケット*

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、ネットワークカードに実装されたレイヤ1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、レイヤ4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、56-kbit/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。

# 22.1 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、項22.2. 「IPv6—次世代のインターネット」を参照してください。

## 22.1.1 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 22.1. 「IPアドレスの表記法」の2行目に示すような形式で表記されます。

***Example 22.1:** IPアドレスの表記法*

```
IPアドレス(2進表記):11000000 10101000 00000000 00010100
IPアドレス(10進表記):      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。この規則にはもちろん例外もありますが、これらの例外はここの説明で重要ではありません。

イーサネットカード自体も一意のアドレスを持っています。このアドレスを*MAC* (media access control)アドレスと呼びます。MACアドレスは長さ

が48ビットで、国際的に一意であり、ネットワークカードベンダによってハードウェアに書き込まれています。ただし、ベンダ割り当てによるアドレスである*MAC*アドレスには、階層的な体系がなく、多かれ少なかれ無作為に割り当てられているという欠点があります。したがって、このアドレスは、リモートマシンのアドレス指定には使用できません。しかし*MAC*アドレスは、ローカルネットワークのホスト間通信で重要な役割を果たしており、レイヤ2のプロトコルヘッダの主要構成要素です。

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。

## 22.1.2 ネットマスクとルーティング

ネットマスクは、IPアドレスが192.168.0.1のホストに対し、IPアドレスが192.168.0.20のホストの位置を通知するために使用します。簡単にいうと、あるIPアドレスを持つホストは、ネットワークマスクによって、外部と内部を定義します。内部に存在するホスト(同じサブネットワークにあるホスト)は、直接応答します。外部に存在するホスト(同じサブネットワークにないホスト)はゲートウェイまたはルータを経由しなければ応答できません。すべてのネットワークインタフェースは、自身のIPアドレスを受信できるので、非常に状況が複雑になることがあります。

ネットワークパケットが送信される前に、コンピュータ上で、IPアドレスとネットマスクの論理積(AND)がとられ、同様に、送信側ホストのアドレスもネットマスクと論理積(AND)がとられます。複数のネットワークインタフェースが利用できる場合は、通常、考えられるすべての送信側アドレスが確認されます。次に、2つのAND演算の結果が比較されます。比較の結果、違いがなければ、宛先すなわち受信側ホストは自分(つまり送信元)と同じサブネットワークに存在します。そうでなければ、ゲートウェイ経由でアクセスする必要があります。ネットマスクが"1"ビット長くなると、直接アクセスできるホストが減り、逆にゲートウェイ経由で接続できるホストが増えます。例 22.2. 「IPアドレスとネットマスクの論理積(AND)」に、いくつかの例を示します。

***Example 22.2:** IPアドレスとネットマスクの論理積(AND)*

```
IPアドレス(192.168.0.20):11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):11111111 11111111 11111111 00000000
```

```
------------------------------------------------------------
ANDをとった結果:11000000 10101000 00000000 00000000 10
進表記:           192.     168.       0.       0

IPアドレス(213.95.15.200):11010101 10111111 00001111 11001000
ネットマスク(255.255.255.0):11111111 11111111 11111111 00000000
------------------------------------------------------------
ANDをとった結果:11010101 10111111 00000000 00000000 10
進表記:           213.      95.      15.       0
```

ネットマスクは、IPアドレスと同様、ピリオドで区切った10進数で表記します。ネットマスクも32ビットの値なので、4つの数値が隣り合って表記されます。どのホストがゲートウェイか、どのアドレスドメインがどのネットワークインタフェース経由でアクセスできるかを設定する必要があります。

別の例を挙げましょう。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。イーサネットがスイッチまたはブリッジで分割されても、これらのホストは直接アクセス可能です。

イーサネットはコストを節約できる安いメディアですが、長距離を接続するのに不向きです。長距離接続では、IPパケットを別のメディア(FDDIやISDN)に転送する必要が出てきます。この転送のために使用されるデバイスのことを、ルータまたはゲートウェイといいます。Linuxマシンは、ルータまたはゲートウェイとして動作できます。これを実現するのが、`ip_forwarding`というオプションです。

ゲートウェイを設定すると、IPパケットは適切なゲートウェイに送信されます。そこからさらにパケットは、複数のホストを経由して転送され、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。

*Table 22.2: 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 22.2. 「IPアドレスとネットマスクの論理積(AND)」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。上の例の場合、この結果は192.168.0.255になります。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。 |
| 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"のみ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ないので、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用に、3つのアドレスドメインが用意されています。これらのアドレスはインターネット経由で伝送されるパケットに設定できないので、何らかの工夫をしない限り、これらのアドレスを持つホストとインターネット上の他のマシンとで接続を確立することはできません。このようなアドレスドメインは、RFC 1597で、表 22.3. 「プライベートIPアドレスドメイン」に示すとおりに定められています。

*Table 22.3: 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 |
| 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 |

## 22.2 IPv6—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、IP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のビット数乗- 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、4、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す。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。

### 22.2.1 利点

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3.4×1038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、項22.2.2.「アドレスのタイプと構造」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なだけでなく、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、ユーザは複数のネットワークに簡単にアクセスできます。 これは携帯電話会社が提供する国際ローミングサービスに似ています。国際ローミングサービスとは、携帯電話を国外に持ち出し、現地サービスのサービス地域に入ると、電話が自動的に現地サービスにログインするというサービスで、これによりどこにいても同じ番号で電話を受けられ、また自国にいるのと同様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。

**下位互換性** 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。項22.2.3.「IPv4とIPv6の共存」を参照してください。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。

### 22.2.2 アドレスのタイプと構造

前述のように、現行のIPプロトコルには、2つの重要な側面が欠けています。つはIPアドレスの不足の問題が表面化していること、もう1つはネットワークの設定とルーティングテーブルの保守がますます複雑で厄介な作業になりつつあることです。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、この2つのコロン::は、1つのアドレスで一度しか使用できません。この省略表記の例については、例 22.3. 「IPv6アドレスの例」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。

***Example 22.3:** 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クスの長さを指定して定義します。例 22.4. 「プレフィクスの長さを指定したIPv6アドレス」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。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。

***Example 22.4:** プレフィ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クスを認識します。表 22.4. 「IPv6のプレフィクス」に、一部のプレフィクスタイプを示します。

*Table 22.4: IPv6のプレフィクス*

| プレフィクス(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 (実稼動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、ハードウェアメー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とき、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(項22.2.3.「IPv4とIPv6の共存」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に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う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。

### 22.2.3 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(項22.2.2.「アドレスのタイプと構造」を参照)。

(世界的な) IPv4ネットワークから隔離されたIPv6ホストですが、トンネルを使用して通信することができます。つまり、IPv6パケットをIPv4パケットとしてカプセル化し、IPv4ネットワークを通じて伝送します。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** 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。

**Important**

**6boneイニシアチブ**

“旧式の”インターネットの中核部分では、トンネル経由で接続されたIPv6サブネットのネットワークが既に世界中に広がっています。これは*6bone*ネットワーク(`http://www.6bone.net`)というIPv6テスト環境で、新しいプロトコルの実装に必要な経験を積むために、IPv6ベースのサービスを開発して提供するプログラマやインターネットプロバイダを対象としています。詳細については、同プロジェクトのインターネットサイトを参照してください。

**Important**

### 22.2.4 IPv6の設定

通常、IPv6を設定するために、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。ただし、IPv6サポートをロードする必要はあります。それには、`root`権限で、`modprobe ipv6`コマンドを実行します。

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、*ルータアドバタイズプロトコル*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を使用してアドレスとルーティングを自動環境設定することもできます。

詳細については、ifupのマニュアルページを参照してください(`man ifup`)。`/etc/sysconfig/network`ファイルを使用してさまざまなタイプのトンネルを設定する方法が説明されています。

### 22.2.5 関連資料

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.ngnet.it/e/cosa-ipv6.php**
IPv6の基本的な事項についての詳細を紹介した記事が収録されています。IPv6に関する優れた入門書です。

**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。

**http://www.6bone.net/** トンネルIPv6ネットワークに参加するには、このサイト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。

**RFC 2640** IPv6に関する基本的なRFCです。

**IPv6 Essentials** IPv6のあらゆる重要な側面について説明した書籍です。*IPv6 Essentials* (Silvia Hagen著)O'Reilly & Associates, 2002 (ISBN 0-596-00125-8)

## 22.3 名前解決

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`という形式で書かれた完全な名前、たとえば、`laurent.suse.de`を考えてみましょう。フルネーム(*完全修飾ドメイン名* (FQDN: fully qualified domain name))は、ホスト名とドメイン名(`suse.de`)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ア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、ダイア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。SUSE LINUXでのネームサーバアクセスの環境設定については、章 24. ドメインネームシステムを参照してください。

`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。

## 22.4 ネットワーク統合

ホストには、サポートされているネットワークカードを装着する必要があります。通常、ネットワークカードはインストール時に検出され、適切なドライバがロードされます。適切なドライバによってカードが正しく統合されているかを確認するには、コマンド`ipaddress list eth0`を実行します。出力には、ネットワークデバイス`eth0`の上のすべての情報が一覧表示されるか、エラーメッセージが表示されます。

SUSEカーネルのデフォルトで、カーネルによるネットワークカードのサポートがモジュールとして実装されている場合は、モジュール名を`/etc/sysconfig/hardware/hwcfg-*`にエイリアスとして入力する必要があります。このファイルにモジュール名が指定されていない場合は、`hotplug`によって自動的にドライバが選択されます。ネットワークカードのタイプ(ホットプラグ可能カードまたは内蔵カード)にかかわらず、`hotplug`によってドライバが割り当てられます。

### 22.4.1 YaSTでのネットワークカード設定

モジュールを起動すると、汎用のネットワーク設定ダイアログが表示されます。上部には、未設定の全ネットワークカードのリストが表示されます。ブート処理中に正常に自動検出されたカードの場合は、リストに名前が示されます。検出できなかったデバイスは、［'その他(未検出)'］と表示されます。ダイアログの下部には、設定済みのデバイスがネットワークタイプおよびアドレスと共にリスト表示されます。これで、新規のネットワークカードを設定するか、既存の設定を変更できます。

#### ネットワークカードの手動設定

自動検出されなかった(［'Other'］としてリストされた)ネットワークカードについては、次の項目を設定する必要があります。

**［ネットワークの設定］** インタフェースのデバイスタイプおよび設定名を設定します。表示されるオプションからデバイスタイプを選択します。また、必要に応じて設定名を指定します。通常は、デフォルト設定のままで問題ありません。設定名の命名規則については、`getcfg`のマニュアルページを参照してください。

**［カーネルモジュール］** ［'ハードウェア設定名'］には、ネットワークカードのハードウェア設定が記述されている`/etc/sysconfig/hardware/hwcfg-*`ファイルの名前、たとえば適切なカーネルモジュールの名前を指定します。通常、PCMCIAおよびUSBハードウェアに対しては、妥当な名前がYaSTによって提示されます。その他のハードウェアに対しては、カードが`hwcfg-static-0`で設定されている場合、通常は`0`だけが妥当な名前です。

ネットワークカードが、PCMCIAデバイスかUSBデバイスの場合、［ネットワークカードの手動設定］ダイアログで、［PCMCIA］または［USB］チェックボックスを有効にして、［'次へ'］をクリックしてダイアログを終了します。PCMCIAデバイスまたはUSBデバイスではない場合、［'一覧表から選択する'］を使って、ネットワークカードの型式を選択します。［'次へ'］をクリックして、このダイアログを終了します。

#### ネットワークアドレスの設定

インタフェースのデバイスタイプおよび設定名を設定します。［デバイスの型］で表示されるオプションから該当するデバイスタイプを選択します。ま

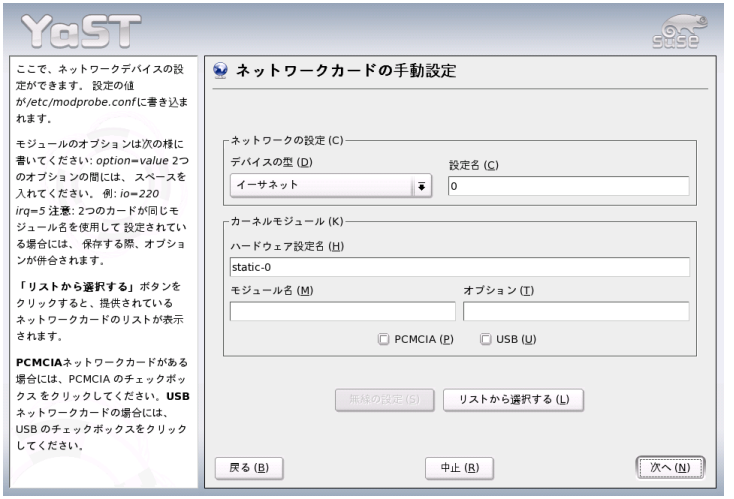


*Figure 22.3: ネットワークカードの設定*

た、必要に応じて設定名を指定します。通常は、デフォルト設定のままで問題ありません。設定名の命名規則については、`getcfg`のマニュアルページを参照してください。

［ネットワークの設定］の［デバイスの型］で［'無線'］を選択した場合は、次の［'無線ネットワークカードの設定'］ダイアログで、動作モード、ネットワーク名(ESSID)、および暗号化方式を設定します。［'了解'］をクリックして、カードの設定を完了します。WLANカードの設定の詳細については、項17.1.3. 「YaSTでの設定」を参照してください。その他のインタフェースタイプの場合は、ネットワークアドレスの設定に進んでください。

**'［自動アドレス設定(DHCPを介して)］'**
ネットワーク上でDHCPサーバを稼働している場合は、DHCPサーバからネットワークアドレスを自動的に取得できます。DSL回線を使用していてISPからスタティックIPが割り当てられていなければ、オプションも使用する必要があります。DHCPを使用する場合は、［'DHCPクライアントオプション'］を選択して詳細を設定します。DHCPサーバが常にブロードキャストリクエストを受け付けるかどうか、および使用するIDを指定します。デフォルトでは、DHCPサーバはカードのハードウェアアドレスを使用してインタフェースを識別します。 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IDが必要になります。

**‘［スタティックなアドレスの設定］’**
スタティックなアドレスを使用する場合は、対応するチェックボックスを選択します。次に、ネットワークのアドレスとサブネットマスクを入力します。事前設定されているサブネットマスクは、典型的なホームネットワークの要件と一致している必要があります。

［‘次へ’］を選択して、このダイアログを終了するか、ホスト名、ネームサーバ、ルーティングの設定に進みます(およびを参照)。

［‘詳細’］で、より詳細な設定を指定できます。［‘詳細設定’］の［‘ユーザコントロール’］を使用してネットワークカードの制御を管理者(root)から一般ユーザに委任できます。これにより、モバイル環境で、ユーザがインタフェースの有効/無効を自身で制御してネットワーク接続を柔軟に変更できるようになります。MTU (Maximum Transmission Unit)および［‘デバイスの起動’］のタイプもこのダイアログで設定できます。

## 22.4.2 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’］を使用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い場合は、手動設定用のダイアログを開きます。開いたダイアログの［‘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。

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も選択します。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、［‘ユーザコントロール’］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。［‘ダイアルプレフィクス正規表現’］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

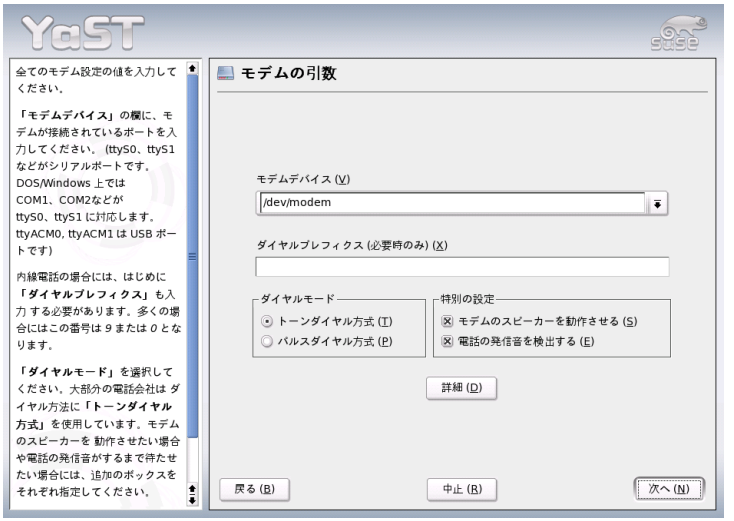


*Figure 22.4:* モ デ ム 設 定

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 (インターネットサービスプロバイダ)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。

'**［接続時にDNSを変更する］**' このチェックボックスはデフォルトで有効であり、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。ただし、［'必要に応じてダイヤルする'］を有効にした場合は、このチェックボックスを無効にし、ネームサーバの固定アドレスを指定してください。

‘［**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。

‘［**スチューピッドモード**］’ このオプションは、デフォルトで有効です。その場合、接続プロセスを妨げないように、ISPのサーバから送信される入力プロンプトは無視されます。

‘［**ファイアウォールを有効にする**］’
このオプションを選択すると、SUSEのファイアウォールが有効になり、インターネット接続中は外部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。［‘完了’］を選択し、このダイアログを閉じます。

### 22.4.3 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、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 22.5.「ISDNの設定」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 (EDSS1)’］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、［‘市外局番’］と、必要に応じてダイヤルプレフィックスを入力します。

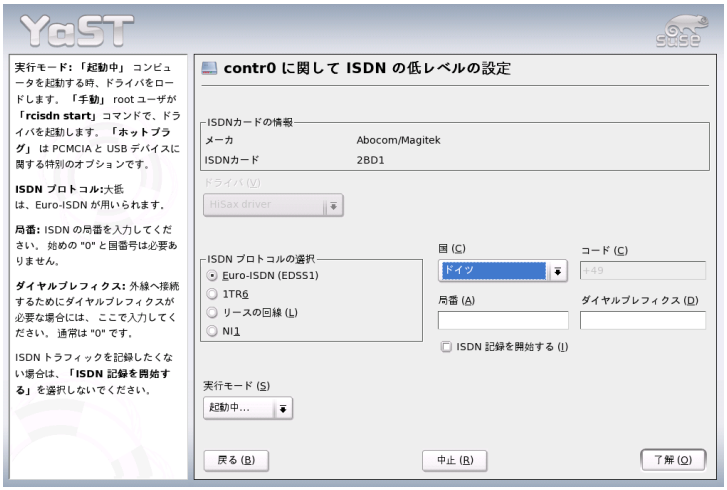


*Figure 22.5: ISDNの 設 定*

［‘実行モード’］では、ISDNインタフェースの起動方法を定義します。［‘起動中’］を選択すると、システムが起動されるたびにISDNドライバが初期化されます。［‘手動’］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。［‘ホットプラグ’］は、PCMCIAやUSBデバイスに使用します。このオプションを選択すると、デバイスが装着されたときにドライバがロードされます。 これらの設定がすべて完了したら、［‘OK’］をクリックします。

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。

［‘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)。加入者が望めば、最大10の電話番号を付与できます。ここには、いずれか1つのMSNを市外局番なしで入力します。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。

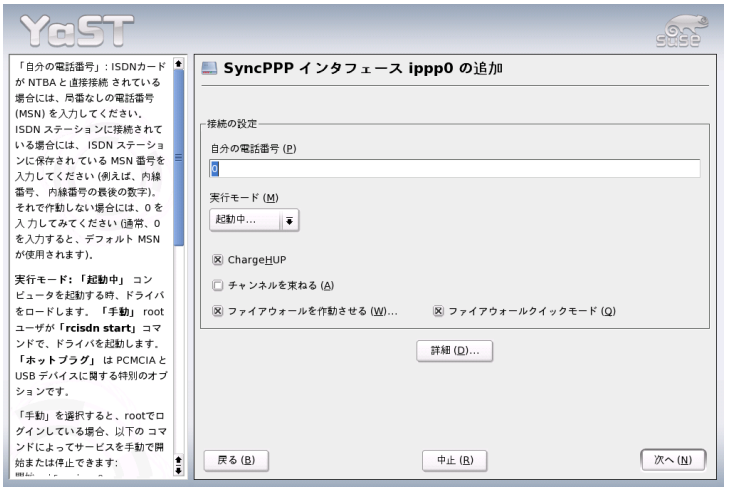


*Figure 22.6: ISDNインタフェースの設定*

**電話交換機に接続された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. 個人向けの小型の電話交換機ではたいてい、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つは内線用に使用できるはずです。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、内線通話に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EAZ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、EAZの最後の数字を入力するだけで十分なはずです。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。

次回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。チャネルバンドル(マルチリンクPPP)を有効にするチェックボックスも用意されています。最後に、使用している回線でSuSEfirewall2を有効にするには、［'ファイアウォールを有効にする'］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、［'ユーザコントロール'］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。［'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、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コ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、設定されたインタフェースのサマリーが表示されます。すべての設定を有効にするには、［'完了'］を選択します。

### 22.4.4 ケーブルモデム

一部の国(オーストリア、米国)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、［'自動アドレス設定(DHCPを介して)'］または［'スタティックなアドレスの設定'］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。

### 22.4.5 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YaST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

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、まず、［'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(項22.4.1. 「YaSTでのネットワークカード設定」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、［'自動アドレス設定(DHCPを介して)'］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。

**Tip**

［'IPアドレス'］と［'サブネットマスク'］の値は単なるプ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。

**Tip**

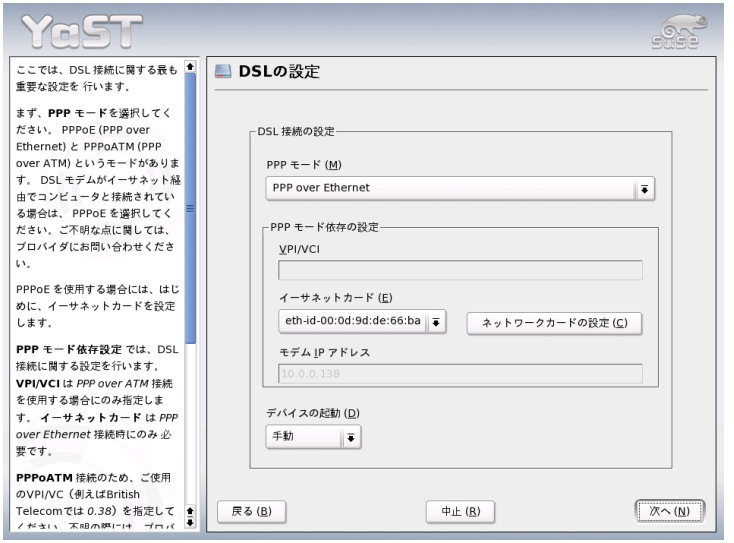


*Figure 22.7: DSLの 設 定*

DSLの設定を始めるには(図 22.7. 「DSLの設定」参照)、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、［'デバイスの起動'］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、［'ユーザコントロール'］を選択します。このダイアログでは、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。

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。

## 22.5 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。

ホットプラグネットワークカード(PCMCIA、USB、一部のPCIカード)に加え、すべての内蔵ネットワークカードがホットプラグ経由で検出、設定されます。以下に、この手順について説明します。システムは、ネットワークカードを2つの異なる方法で認識します。すなわち、物理的なデバイスとして認識する場合と、インタフェースとして認識する場合があります。デバイスが挿入または検出されると、ホットプラグイベントが生成されます。このホットプラグイベントによって、`/sbin/hwup`スクリプトが実行され、デバイスが初期化されます。ネットワークカードが新しいネットワークインタフェースとして初期化されると、カーネルによって別のホットプラグイベントが生成され、それにより`/sbin/ifup`が実行されてインタフェースがセットアップされます。

カーネルは、登録順に従ってインタフェース名に番号を付けます。割り当てられる名前は、初期化の順序によって決まります。あるネットワークカードの初期化に失敗した場合、その後に初期化されるカードの番号は1つずつずらされ

ます。実際のホットプラグ対応カードでは、デバイスを接続する順序が重要になります。

柔軟な環境設定を可能にするために、デバイス(ハードウェア)の環境設定とインタフェースの環境設定は切り分けられ、デバイスの環境設定とインタフェースの環境設定のマッピングをインタフェース名で管理する方式は廃止されました。デバイスの環境設定は、/etc/sysconfig/hardware/hwcfg-*に格納されます。インタフェースの環境設定は、/etc/sysconfig/network/ifcfg-*に格納されます。これらの環境設定ファイルには、そのファイルに関連付けられるデバイスまたはインタフェースを表す名前が付けられます。ドライバをインタフェース名にマッピングする従来の方式では静的なインタフェース名が必要なため、このマッピングを/etc/modprobe.confで行うことはできなくなりました。この新しい方式では、このファイルにエイリアスエントリが設定されていると、好ましくない副作用が発生することがあります。

環境設定名、すなわち、hwcfg-またはifcfg-の後の部分では、スロット、デバイス固有のID、インタフェース名などでデバイスを表します。たとえば、PCIカードの環境設定名は、bus-pci-0000:02:01.0 (PCIスロット)、vpid-0x8086-0x1014-0x0549 (メーカー名と製品ID)などになります。対応するインタフェース名は、bus-pci-0000:02:01.0、wlan-id-00:05:4e:42:31:7a (MACアドレス)などになります。

特定のカードではなく特定のタイプのカードにネットワークの環境設定を割り当てる場合は(ただし、同じタイプのカードを同時に2枚以上は装着しない)、もう少し汎用的な設定名を選択します。たとえば、すべてのPCMCIAカードに対してbus-pcmciaという設定名を使用できます。一方、先頭にインタフェースタイプが付いた限定的な設定名も使用できます。たとえば、USBポートに接続するWLANカードにはwlan-bus-usbという設定名を付けることができます。

システムは常に、インタフェースまたはそのインタフェースを提供するデバイスに最適な環境設定を使用します。最適な環境設定の検索は、/sbin/getcfgによって行われます。getcfgの出力には、デバイスを記述するために使用できるすべての情報が含まれています。環境設定名の指定の詳細については、getcfgのマニュアルページを参照してください。

この方法により、ネットワークデバイスは常に同じ順序で初期化されるとは限りませんが、ネットワークインタフェースは適切に設定されます。ただし、インタフェース名は、やはり初期化の順序によって決まります。特定のネットワークカードのインタフェースに確実にアクセスするには、次の2とおりの方法があります。

- /sbin/getcfg-interface〈環境設定名〉を実行すると、対応するネット

ワークインタフェース名が返されます。したがって、一部の環境設定ファイルでは(残念ながら現時点ではすべてではありませんが)、ファイアウォール、dhcpd、ルーティング、各種バーチャルネットワークインタフェース(トンネル)などの設定名を、固定的でないインタフェース名の代わりに指定できます。

- 環境設定にインタフェース名が含まれていないすべてのインタフェースには、固定的なインタフェース名を割り当てることができます。これを行うには、インタフェースの環境設定(`ifcfg-*`)に`PERSISTENT_NAME=<pname>`という名前のエントリを指定します。ただし、固定名〈*pname*〉は、カーネルによって自動的に割り当てられる名前とは異なっていなければなりません。したがって、`eth*`、`tr*`、`wlan*`、`qeth*`、`iucv*`などの名前は使用できません。このような名前ではなく、`net*`または`external`、`internal`、`dmz`.などの説明的な名前を使用します。固定名は、登録直後にのみインタフェースに割り当てることができます。つまり、ネットワークカードのドライバを再ロードするか、`hwup`デバイス記述を実行する必要があります。`rcnetworkrestart`コマンドを実行するだけでは不十分です。

**Important**

**固定的なインタフェース名の使用について**

固定的なインタフェース名の使用は、一部の領域ではテストされていません。したがって、アプリケーションによっては、自由に選択したインタフェース名を使用できないことがあります。この種の問題が発生した場合は、`http://www.suse.de/feedback`を使用して具体的な内容をお知らせください。

**Important**

`ifup`はハードウェアを初期化しないため、すでに存在しているインタフェースを必要とします。ハードウェアの初期化は、`hwup`コマンドによって行われます(このコマンドは`hotplug`または`coldplug`によって実行されます)。デバイスが初期化されると、`hotplug`によって`ifup` が新しいインタフェースに対して自動的に実行され、実行モードが`onboot`、`hotplug`、または`auto`であり`network`サービスが既に起動していれば、インタフェースがセットアップされます。従来は、`ifup`インタフェース名コマンドによってハードウェアの初期化が行われていましたが、新しいバージョンでは処理順序が逆になりました。まず、ハードウェアコンポーネントを初期化してから、その他の処理が行

われます。この方法により、可変数のデバイスを、既存の環境設定を用いてできる限り最適な方法で設定できます。

表 22.5.「手動ネットワーク環境設定用スクリプト」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。各スクリプトはハードウェアとインタフェースに分類してあります。

*Table 22.5: 手動ネットワーク環境設定用スクリプト*

| 環境設定段階 | コマンド | 機能 |
|---|---|---|
| ハードウェア | hw{up,down,status} | hw*スクリプトは、ホットプラグサブシステムによって実行され、デバイスの初期化、初期化の取り消し、デバイスのステータスの問い合わせを行います。詳細は、hwupのマニュアルページを参照してください。 |
| インタフェース | getcfg | getcfgは、環境設定名またはハードウェア記述に対応するインタフェース名の問い合わせに使用します。詳細は、getcfgのマニュアルページを参照してください。 |
| インタフェース | if{up,down,status} | if*スクリプトは、既存のネットワークインタフェースを起動したり、指定のインタフェースのステータスを表示したりします。詳細は、ifupのマニュアルページを参照してください。 |

ホットプラグおよび固定的なデバイス名の詳細については、章 18. ホットプラグシステムおよび章 19. udevをもつ動的デバイスノードを参照してください。

### 22.5.1 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/syconfig/hardware/hwcfg-*

これらのファイルには、ネットワークカードおよびその他のデバイスのハードウェアの環境設定が記述されています。これには、カーネルモジュール、実行モード、スクリプトの関連付けなどの必要なパラメータが含まれます。詳細については、hwupのマニュアルページを参照してください。存在しているハードウェアとは無関係に、coldplugの起動時にはhwcfg-static-*が適用されます。

### /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記述されています。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。これら3つの環境設定ファイルの変数にはコメントが付けられており、優先度の高い変数としてifcfg-*ファイルでも使用できます。

### /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。各種システムタスクで必要となるすべての静的経路(ホストへの経路、ゲートウェイを介したホストへの経路、ネットワークへの経路)は、/etc/sysconfig/network/routesファイルに指定します。個別のルーティングを必要とする各インタフェースには、追加の環境設定ファイル/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           GATEWAY NETMASK   INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION           GATEWAY PREFIXLEN INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION/PREFIXLEN GATEWAY -         INTERFACE [ TYPE ] [ OPTIONS ]
```

GATEWAY、NETMASK、PREFIXLEN、またはINTERFACEを省略する場合は、代わりに-を指定します。TYPEとOPTIONSは単に省略するだけでかまいません。

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、ダミーデバイスなど)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。

### /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`）。ドメイン名は複数指定することができます。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使用するには、`nameserver`で始まる行を複数行入力します。/etc/resolv.confコメントは#記号の後に記入します。YaSTは、指定されているネームサーバをこのファイルに記述します。例 22.5. 「/etc/resolv.conf」に`/etc/resolv.conf`の例を示します。

*Example 22.5: `/etc/resolv.conf`*

```
# Our domain search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

`pppd`(`wvdial`)、`ipppd`(`isdn`)、`dhcp`(`dhcpcd`、`dhclient`)、`pcmcia`、`hotplug`などの一部のサービスは、スクリプト`modify_resolvconf`を使用してファイル`/etc/resolv.conf`に変更を加えます。ファイル`/etc/resolv.conf`がこのスクリプトによって一時的に変更された場合、変更を加えたサービス、元のファイルがバックアップ

されている場所、および自動変更メカニズムを無効にする方法を示す事前定義のコメントが付されます。/etc/resolv.confが複数回変更された場合、ファイルには変更内容がネスト形式で保存されます。変更が行われた順序と異なる順序で復元を行った場合も、問題なく元通りに復元できます。このような柔軟性を必要とするサービスには、isdn、pcmcia、およびhotplugがあります。

サービスが通常のクリーンな状態で停止しなかった場合、modify_resolvconfを使用して元のファイルを復元することができます。また、システムブート時に、クリーンアップされていない変更されたresolv.confが存在しないかがチェックされ(たとえば、システムクラッシュがあった場合)、存在する場合は、元の(変更されていない) resolv.confが復元されます。

YaSTは、modify_resolvconf checkコマンドを使用して、resolv.confが変更されているかどうかを確認し、ユーザに対してファイルの復元後は変更内容が失われることを警告します。YaSTはこれ以外の作業でmodify_resolvconfに依存しないため、YaSTを使用して resolv.confを変更した場合の影響は、手動で変更した場合と同じです。どちらの場合も、変更は永久に有効です。一方、前述のサービスによって要求された変更は、一時的に有効なだけです。

### /etc/hosts

このファイル(例 22.6. 「/etc/hosts」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、各エントリはブランク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。

*Example 22.6: /etc/hosts*

```
127.0.0.1 localhost
192.168.0.20 sun.example.com sun
192.168.0.1 earth.example.com earth
```

### /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(例 22.7. 「/etc/networks」を参照)。

*Example 22.7: /etc/networks*

```
loopback     127.0.0.0
localnet     192.168.0.0
```

### /etc/host.conf

このファイルは、名前解決(*resolver*ライブラリによるホスト名とネットワーク名の変換)を制御します。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 22.6.「/etc/host.confファイルのパラメータ」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 22.8.「/etc/host.conf」を参照してください。

*Table 22.6: /etc/host.confファイルのパラメータ*

| | |
|---|---|
| *hosts*、*bind*の順序 | 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)。<br>*hosts*:`/etc/hosts`ファイルを検索します。<br>*bind*:ネームサーバにアクセスします。<br>*nis*:NISを経由します。 |
| multi *on*/*off* | `/etc/hosts`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。 |
| nospoof *on* spoofalert *on*/*off* | 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、それ以外のネットワークの環境設定に対してまったく影響を与えません。 |
| `trim domainname` | 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`/etc/hosts`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。 |

*Example 22.8: /etc/host.conf*

```
# We have named running
order hosts bind
# Allow multiple addrs
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、`man5 nsswitch.conf`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、例 22.9. 「/etc/nsswitch.conf」を参照してください。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 (章 24. ドメインネームシステムを参照)経由で`/etc/hosts` (`files`)に送信されています。

*Example 22.9: /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 22.7. 「/etc/nsswitch.confで利用できるデータベース」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 22.8. 「NSS データベースの環境設定オプション」を参照してください。

*Table 22.7: /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 |
| グループ | getgrentがユーザグループを調べるとき使用します。groupのマニュアルページも参照してください。 |
| hosts | gethostbynameおよび同様の関数が、ホスト名とIPアドレスを取得するために使用します。 |
| 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。man5 netgroup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。 man5 passwd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。 man5 protocols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。 man5 shadowを参照してください。 |

*Table 22.8: NSS データベースの環境設定オプション*

| | |
|---|---|
| files | 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。章 25. NISの使用を参照。 |
| 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。man8 nscdおよびman5 nscd.conf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。

### 22.5.2 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(表 22.9.「ネットワークプログラム用スタートアップスクリプト」も参照)。

*Table 22.9: 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。ハードウェアが事前に(hotplug経由で)/etc/init.d/coldplugによって初期化されている必要があります。networkサービスが起動していないと、ネットワークインタフェースは、ホットプラグ経由で挿入されたときに初期化されません。 |
| /etc/init.d/inetd | xinetdを起動します。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。 |
| /etc/init.d/portmap | NFSサーバなどのRPCサーバに必要なポートマッパを起動します。 |
| /etc/init.d/nfsserver | NFSサーバを起動します。 |
| /etc/init.d/sendmail | sendmailプロセスを制御します。 |
| /etc/init.d/ypserv | NISサーバを起動します。 |
| /etc/init.d/ypbind | NISクライアントを起動します。 |

## 22.6 ダイアルアップアシスタントとしてのsmpppd

ほとんどのユーザは、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用できるようにして、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。

### 22.6.1 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>** 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。

**host-range = <min ip> <max 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の詳細は、man 8 smpppdコマンドおよびman 5 smpppd.confコマンドを実行して、各マニュアルページを参照してください。

### 22.6.2 リモートで使用するためのkinternet、cinternet、およびqinternetの設定

kinternet、cinternet、およびqinternetは、ローカルに使用できるのみではなく、リモートsmpppdの制御にも使用できます。cinternetとはグラフィカルなkinternetに相当するコマンドラインプログラムです。qinternetは基本的にはkinternetと同じですが、KDEライブラリを使用しません。そのためKDEなしで使用することができ、また個別にインストールする必要があります。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kinternetを使用して編集します。このファイルでは、以下の3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。オプションlocalはローカルsmpppdへの接続の確立を指示します。gatewayはゲートウェイ上のsmpppdをポイントします。接続は、config-fileのserverの指定に従って確立する必要があります。slpは、フロントエンドに対してSLPによって検出されたsmpppdに接続するよう指示します。

**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。

**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cinternet --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、man 5 smpppd-c.confコマンドおよびman 8 cinternetコマンドを実行して、マニュアルページを参照してください。

# 23

ネットワーク上のSLPサービス

# 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では、SLPを介して提供されたインストールソースによるインストールをサポートしています。また、SLPサポートとして組み込まれた様々なシステムサービスが含まれています。YaSTおよびKonquerorのどちらもSLPに適切なフロントエンドを備えています。SUSE LINUXでインストールサーバ、YOU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。

## 23.1 独自のサービスを登録する

SUSE LINUX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。

登録ファイルにも、2つの変数である`watch-tcp-port`および`description`が含まれます。watch-tcp-portは、どの関連サービスがアクティブかによってSLPサービス通知にリンクします(slpdはサービスの状態をチェックします)。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** 前述の手順と異なるのは、一元的なファイルですべてのサービスをグループ化している点です。

**slptoolによる動的登録** 専用のスクリプトからサービスをSLPに登録するには、slptoolコマンドラインフロントエンドを使用します。

## 23.2 SUSE LINUXのSLPフロントエンド

SUSE LINUXには複数のフロントエンドが含まれます。これらはネットワークでチェック後に使用されるSLP情報を有効にします。

**slptool** slptoolはネットワーク上のSLP照会を通知するため、または適切な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。

**YaST SLPブラウザ** YaSTには個別のSLPブラウザが含まれており、'ネットワークサービス' → 'SLPブラウザ'の下に、SPLを介して通知されたローカルネットワークのすべてのネットワークがツリーダイアグラム形式でリストされます。

**Konqueror** ネットワークブラウザとして使用される場合、Konquerorは`slp:/`のローカルネットワークで使用可能なすべてのSLPサービスを表示できます。メインウィンドウにあるアイコンをクリックして、関連サービスについての詳細情報を参照してください。

Konquerorを`service:/`で使用する場合、ブラウザウィンドウで関連するアイコンをクリックして、選択したサービスとの接続をセットアップします。

## 23.3 SLPをアクティブ化する

サービスを提供する場合、システム上でslpdが実行している必要があります。サービスの照会を作成するだけの場合は、このデーモンを開始する必要はありません。SUSE LINUXのほとんどのシステムサービスと同様、slpdデーモンは別の初期化スクリプトを使用して制御されます。このデーモンはデフォルトで非アクティブになっています。セッション中にこのデーモンをアクティ

ブにするには、rcslpd startをrootで実行してデーモンを開始し、rcslpd stopで終了します。restartで再始動、またはstatusで状態チェックを実行します。slpdをデフォルトでアクティブにするには、insserv slpdコマンドをrootで、1度実行します。システムのブート時に開始するサービスセットとしてslpdが自動的に追加されます。

## 23.4 関連資料

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.com** OpenSLPプロジェクトのホームページです。

**file:/usr/share/doc/packages/openslp/*** このディレクトリには、SUE LINUXの詳細、前述の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を使用するプログラマはopenslp-develパッケージをインストールし、その中で提供される『*Programmers Guide*』を確認してください。

# 24

ドメインネームシステム

# ドメインネームシステム

DNS (ドメインネームシステム)は、ドメイン名とホスト名をIPアドレスに解決するために必要です。これにより、たとえばIPアドレス192.168.0.1がホスト名earthに割り当てられます。独自のネームサーバをセットアップする前に、項22.3. 「名前解決」でDNSに関する一般的な説明を参照してください。以降に示す設定例はBINDの場合のものです。

## 24.1 YaSTによる設定

YaSTのDNSモジュールを使用すると、ローカルネットワーク用のDNSサーバを設定できます。このモジュールを初めて起動すると、サーバ管理に関して少数の基本的な事項を決定するように要求されます。この初期セットアップを完了すると、必要最低限の機能が設定された基本的なサーバ設定が生成されます。エキスパートモードを使用すると、、より詳細な設定タスクを行うことができます。

### 24.1.1 ウィザードによる設定

ウィザードは3つのステップ(ダイアログ)で構成されています。各ダイアログの適切な箇所でエキスパート用の設定モードに入ることができます。

**フォワーダの設定** モジュールを初めて起動すると、図 24.1. 「DNSサーバのインストール:フォワーダの設定」のようなダイアログが表示されます。このダイアログでは、PPPデーモンがDSLまたはISDNを介してダイアルアップ時にフォワーダのリストを提供するか( [‘PPPデーモンがフォワーダを設定する’] )、または独自のリストを指定するか( [‘手動でフォワーダを設定する’] )を指定できます。

**DNSゾーン** 複数の部分で構成されるこのダイアログでは、項24.4. 「ゾーンファイル」で説明するゾーンファイルの管理に関する項目を設定します。新しいゾーンを作成する場合は、 [‘ゾーン名’] にその名前を入力します。逆引きゾーンを追加する場合は、`.in-addr.arpa`で終わる名前を入力しなければなりません。最後に、 [‘ゾーンのタイプ’] (マスターまたはスレーブ)を選択します。図 24.2. 「DNSサーバのインストール:DNSゾーン」を参照してください。既存のゾーンのその他の項目を設定するには、 [‘ゾーンの編集’] をクリックします。ゾーンを削除するには、 [‘ゾーンの削除’] をクリックします。

**ウィザードの完了** この最後のダイアログでは、インストール時にアクティブ化されるファイアウォールのDNSサービス用ポートを開いて、DNSを起動するかどうかを決定します。このダイアログからもエキスパート用の設定に入ることができます。図 24.3. 「DNSサーバのインストール:ウィザードの完了」を参照してください。

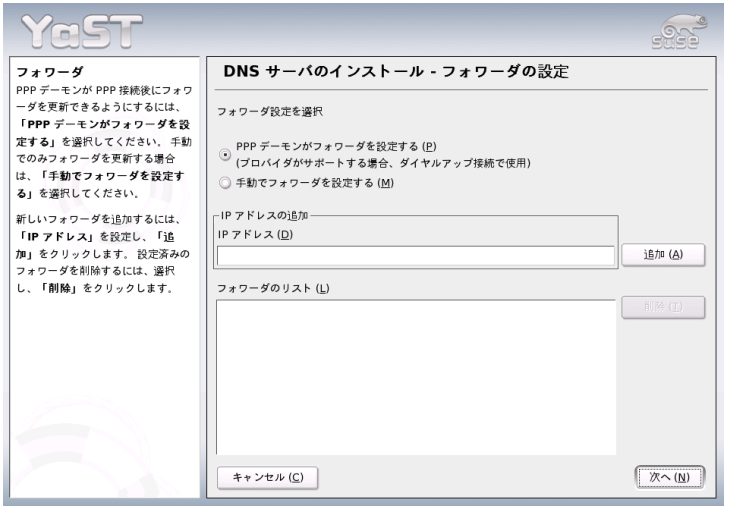


*Figure 24.1: DNSサーバのインストール:フォワーダの設定*

### 24.1.2 エキスパート環境設定

モジュールを起動するとウィンドウが開き、複数の設定オプションが表示されます。設定を完了すると、基本的な機能が組み込まれたDNSサーバ設定が作成されます。

**起動** ［'ブート'］には、DNSサーバをデフォルトで［'オン'］にするか［'オフ'］にするかを指定します。DNSサーバをすぐに起動するには、［'DNSサーバの開始'］を選択します。DNSサーバを停止するには、［'DNSサーバの停止'］を選択します。現在の設定を保存するには、［'設定を保存してDNSを再起動'］を選択します。ファイアウォールのDNSポートを開くには［'ファイアウォールで開いているポート'］を、ファイアウォールの設定を変更するには［'ファイアウォールの詳細'］をクリックします。

**フォワーダ** これは、ウィザードの設定を起動したときに開くダイアログと同じです(を参照)。

**ログ** このセクションでは、DNSサーバがログに記録する内容とログの方法を設定できます。［'ログタイプ'］に、DNSサーバがログデータを書

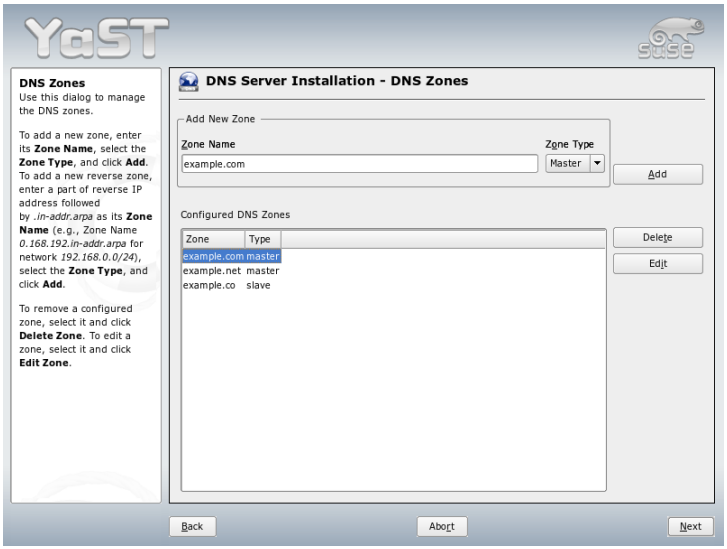


***Figure 24.2:*** *DNSサーバのインストール:DNSゾーン*

き込む場所を指定します。システム全体のログファイル［/var/log/messages］を使用する場合は［'システムログ'］を、別のファイルを指定する場合は［'ファイルに記録'］を選択します。別のファイルを指定する場合は、ログファイルの最大サイズ(メガバイト(MB))とログファイルの数も指定します。

［'追加のログ'］には、さらに詳細なオプションが用意されています。［'名前付きクエリをログに記録'］を有効にすると、すべてのクエリがログに記録されるため、ログファイルが非常に大きくなる可能性があります。ですから、このオプションを有効にするのはデバッグ時だけにすることをお勧めします。DHCPサーバとDNSサーバ間でのゾーン更新時のデータトラフィックをログに記録するには、［'ゾーン更新をログに記録'］を有効にします。マスタからスレーブへのゾーン転送時のデータトラフィックをログに記録するには、［'ゾーン転送をログに記録'］を有効にします。図 24.4. 「DNSサーバ:ログ」を参照してください。

**DNSゾーン** このダイアログでは、ウィザードの設定について説明します。項24.1.1. 「ウィザードによる設定」を参照してください。


*Figure 24.3: DNSサーバのインストール:ウィザードの完了*

**スレーブゾーンエディタ** で説明したステップで、ゾーンタイプとして［'スレーブ'］を選択すると、このダイアログが開きます。［'マスタDNSサーバ'］で、データの転送元としてスレーブが使用するマスタを指定します。サーバへのアクセスを制限するために、リストから定義済みのACLを1つ選択します。図 24.5. 「DNSサーバ:スレーブゾーンエディタ」を参照してください。

**マスタゾーンエディタ** で説明したステップで、ゾーンタイプとして［'マスタ'］を選択すると、このダイアログが開きます。このダイアログは、［'基本'］(最初に開くページ)、［'NSレコード'］、［'MXレコード'］［'SOA'］、［'レコード'］の各ページで構成されます。

**ゾーンエディタ(NSレコード)** このダイアログでは、指定したゾーンの代替ネームサーバを定義できます。リストに自分が使用しているネームサーバが含まれていることを確認してください。レコードを追加するには、［'追加するネームサーバ'］にレコード名を入力し、［'追加'］をクリックして確定します。図 24.7. 「DNSサーバ:ゾーンエディタ(NSレコード)」を参照してください。

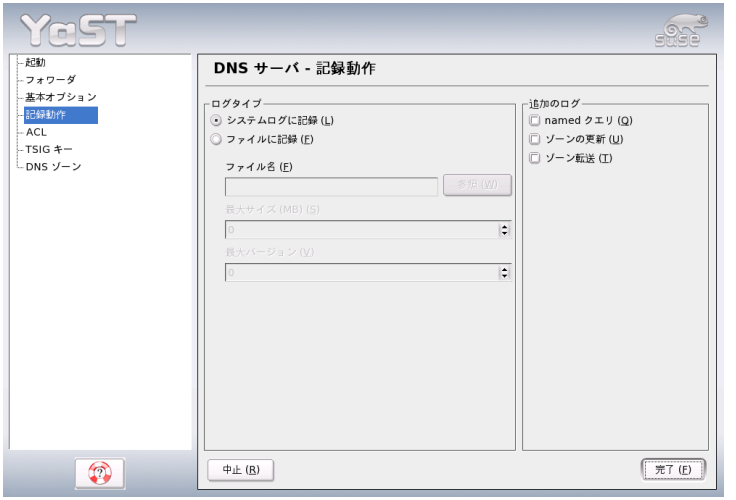


*Figure 24.4: DNSサーバ:ログ*

**ゾーンエディタ(MXレコード)** 現行ゾーンのメールサーバを既存のリストに追加するには、対応するアドレスと優先順位の値を入力します。その後、［'追加'］を選択して確定します。図 24.8. 「DNSサーバ:ゾーンエディタ(MXレコード)」を参照してください。

**ゾーンエディタ(SOA)** このページでは、SOA (start of authority)レコードを作成できます。個々のオプションについては、例 24.6. 「/var/lib/named/world.zoneファイル」を参照してください。LDAPを介して管理される動的ゾーンの場合、SOAレコードの変更がサポートされないので注意してください。

**ゾーンエディタ(レコード)** このダイアログでは、名前解決を管理します。［'レコードキー'］では、ホスト名を入力してレコードタイプを選択します。［'A-Record (Aレコード)'］はメインエントリを表します。この値はIPアドレスでなければなりません。［'CNAME'］はエイリアスです。［'NS'］および［'MX'］の各タイプを指定すると、［'NSレコード'］および［'MXレコード'］の各タブで提供される情報に基づいて、詳細レコードまたは部分レコードが展開されます。この3つのタイプのは、既存のAレコードに解決されます。［'PTR'］は逆引きゾーン用レコードです。これは、Aレコードとは反対にIPアドレスに対するホスト名を定義


*Figure 24.5: DNSサーバ:スレーブゾーンエディタ*

します。

## 24.2 ネームサーバBINDの起動

SUSE LINUXシステムでは、BIND (*Berkeley Internet name domain*)が事前に設定された状態で提供されているので、インストールが正常に完了すればすぐにネームサーバが起動されます。既にインターネットに接続し、`/etc/resolv.conf`の`localhost`にネームサーバアドレス`127.0.0.1`が入力されている場合、通常、プロバイダのDNSを知らなくても、既に機能する名前解決メカニズムが存在します。この場合、BINDは、ルートネームサーバを介して名前の解決を行うため、処理が非常に遅くなります。通常、効率的で安全な名前解決を実現するには、`forwarders`の下の設定ファイル`/etc/named.conf`にプロバイダのDNSとそのIPアドレスを入力する必要があります。いままでこれが機能している場合、ネームサーバは、純粋なキャッシュ専用ネームサーバとして動作しています。ネームサーバは、自身のゾーンを設定してはじめて、本当のDNSになります。これの簡単な例については、`/usr/share/doc/packages/bind/sample-config`のドキュメントを


*Figure 24.6: DNSサーバ:ゾーンエディタ(基本)*

参照してください。

**Tip**

**ネームサーバ情報の自動取得**

インターネット接続やネットワーク接続のタイプによっては、ネームサーバ情報を自動的に現在の状態に適合させることができます。これを行うには、`/etc/sysconfig/network/config`ファイル内で`MODIFY_NAMED_CONF_DYNAMICALLY`変数に`yes`を設定します。

**Tip**

ただし、公式ドメインは、管理団体から割り当てられるまでセットアップしないでください。独自のドメインを持っていて、プロバイダがそれを管理している場合でも、BINDはそのドメインに対する要求を転送しないので、そのドメインを使用しないほうが賢明です。たとえば、プロバイダのWebサーバは、このドメインからはアクセスできません。

ネームサーバを起動するには、`root`ユーザとして、コマンド`rcnamedstart`を入力します。右側に緑色で“done”と表示されたら、named (ネームサーバプロセス名)が正常に起動しています。サーバが正常に起動

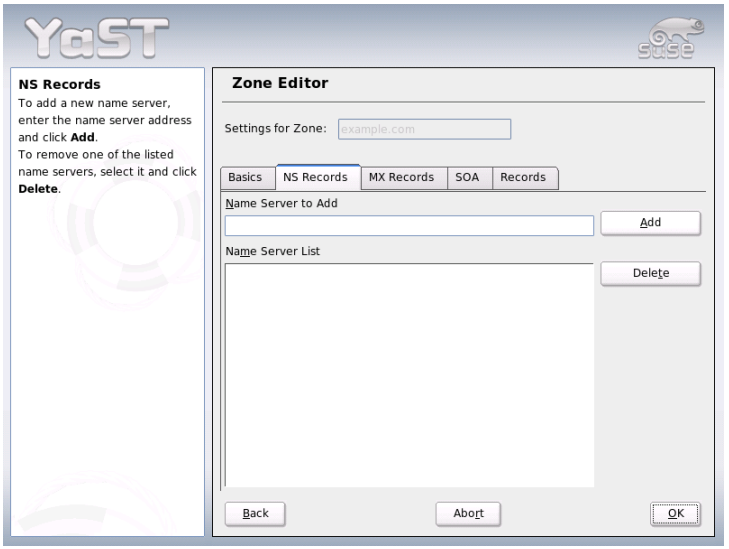


*Figure 24.7: DNSサーバ:ゾーンエディタ(NSレコード)*

したらすぐに、hostまたはdigプログラムを用いてローカルシステム上でネームサーバをテストしてください。デフォルトサーバlocalhostとそのアドレス127.0.0.1が返されるはずです。これが返されない場合は、/etc/resolv.confに含まれているネームサーバエントリが誤っているか、同ファイルが存在しないかのいずれかです。最初のテストとして、host127.0.0.1を入力します。これは常に機能するはずです。エラーメッセージが表示された場合は、rcnamedstatusを使用して、サーバが実際に起動されていることを確認します。ネームサーバが起動しない場合、または予想しない動作をしている場合、多くはログファイル/var/log/messagesでその原因が明らかになります。

プロバイダのネームサーバまたはフォワーダとして既にネットワーク上で動作しているネームサーバを使用する場合は、forwardersの下のoptionsセクションに、対応するIPアドレスまたはアドレスを入力します。例 24.1.「named.confファイルの転送オプション」に含まれているアドレスは、単なる例です。自サイトの設定に合わせて変更してください。

*Example 24.1: named.confファイルの転送オプション*

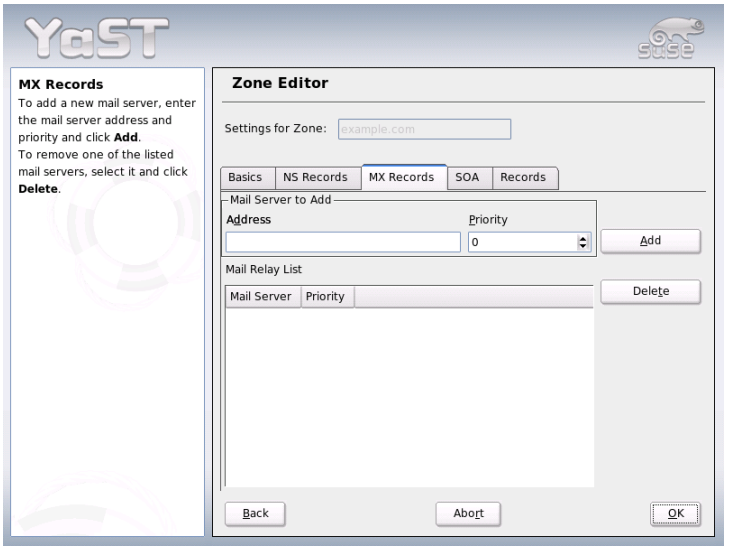


*Figure 24.8: DNSサーバ:ゾーンエディタ(MXレコード)*

```
options {
   directory "/var/lib/named";
   forwarders { 10.11.12.13; 10.11.12.14; };
   listen-on { 127.0.0.1; 192.168.0.99; };
   allow-query { 127/8; 192.168.0/24; };
   notify no;
   };
```

optionsエントリの後には、ゾーン用のエントリ、localhostと0.0.127.in-addr.arpaが続きます。“.”の下のtype hint(タイプヒント)は必ず存在しなければなりません。対応するファイルは、変更する必要がなく、そのままで機能します。また、各エントリの末尾が“;”で閉じられ、中カッコが適切な位置にあることを確認してください。設定ファイル/etc/named.confまたはゾーンファイルを変更したら、rcnamedreloadを使用して、BINDにそれらを再読み込みさせます。または、rcnamedrestartを使用してネームサーバを停止、再起動しても同じ結果が得られます。サーバはrcnamedstopを入力していつでも停止することができます。

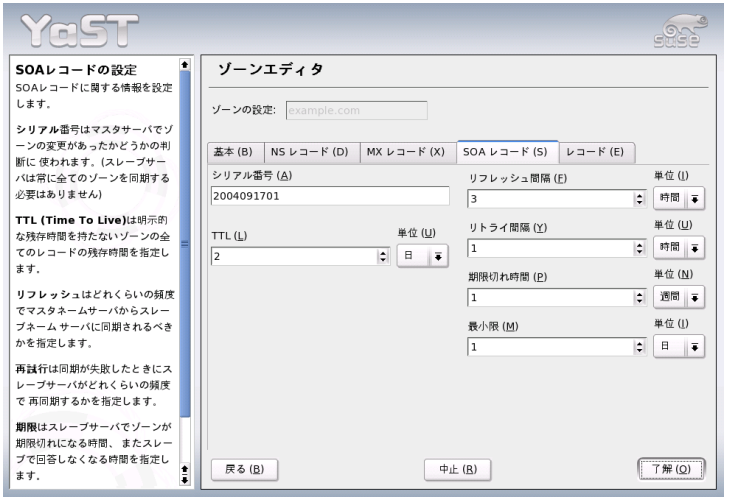


***Figure 24.9:*** *DNSサーバ:ゾーンエディタ(SOA)*

## 24.3 設定ファイル/etc/named.conf

BINDネームサーバ自体の設定はすべて、ファイル/etc/named.confに格納されます。ただし、ホスト名、IPアドレスなどで構成され、ドメインが処理するゾーンデータは、/var/lib/namedディレクトリ内の個別のファイルに格納されます。この詳細については、後述します。

/etc/named.confファイルは、大きく2つのエリアに分けられます。1つは一般的な設定用のoptionsセクション、もう1つは個々のドメインのzoneエントリで構成されるセクションです。ログセクションとacl(アクセス制御リスト)エントリは省略可能です。コメント行は、行頭に#記号または//を指定します。最も基本的な/etc/named.confファイルの例を、例 24.2. 「基本的な/etc/named.confファイル」に示します。

***Example 24.2:*** *基本的な/etc/named.confファイル*

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
```

```
    notify no;
    };

zone "localhost" in {
    type master;
    file "localhost.zone";
    };

zone "0.0.127.in-addr.arpa" in {
     type master;
     file "127.0.0.zone";
     };

zone "." in {
     type hint;
     file "root.hint";
};
```

### 24.3.1 重要な設定オプション

**directory "⟨*filename*⟩";** BINDが検索する、ゾーンファイルが格納されているディレクトリを指定します。通常は`/var/lib/named`です。

**forwarders { ⟨*ip-address*⟩; };** DNS要求が直接解決できない場合、それらが転送されるネームサーバ(ほとんどの場合、プロバイダのネームサーバ)を指定します。⟨*ip-address*⟩には、IPアドレスを`10.0.0.1`のように指定します。

**forward first;** ルートネームサーバでDNS要求の解決を試みる前に、それらを転送するようにします。`forward first`の代わりに`forward only`を指定すると、要求が転送されたままになり、ルートネームサーバには送り返されません。このオプションは、ファイアウォール構成で使用します。

**listen-on port 53 { 127.0.0.1; ⟨*ip-address*⟩; };**
BINDがクライアントからのクエリを受け取るネットワークインタフェースとポートを指定します。`port 53`はデフォルトポートであるため、明示的に指定する必要はありません。ローカルホストからの要求を許可するには、`127.0.0.1`と記述します。このエントリ全体を省略した場合は、すべてのインタフェースがデフォルトで使用されます。

**listen-on-v6 port 53 {any; };** BINDがIPv6クライアント要求をリッスンするポートを指定します。any以外で指定できるのはnoneだけです。IPv6に関して、サーバはワイルドカードアドレスのみ受け付けます。

**query-source address * port 53;** ファイアウォールが発信DNS要求をブロックする場合、このエントリが必要です。BINDに対し、外部への要求をポート53から発信し、1024を超える上位ポートからは発信しないように指示します。

**query-source address * port 53;** BINDがIPv6のクエリに使用するポートを指定します。

**allow-query { 127.0.0.1; ⟨*net*⟩; };** クライアントがDNS要求を発信できるネットワークを定義します。⟨*net*⟩には、アドレス情報を`192.168.1/24`のように指定します。末尾の`/24`は、ネットマスクの短縮表記で、この場合`255.255.255.0`を表します。

**allow-transfer ! *;;** ゾーン転送を要求できるホストを制御します。この例では、！ *が使用されているので、ゾーン転送要求は完全に拒否されます。このエントリがなければ、ゾーン転送をどこからでも制約なしに要求できます。

**statistics-interval 0;** このエントリがなければ、BINDは1時間ごとに数行の統計情報を生成して`/var/log/messages`に保存します。0を指定すると、統計情報をまったく生成しないか、時間間隔を分単位で指定します。

**cleaning-interval 720;** このオプションは、BINDがキャッシュをクリアする時間間隔を定義します。キャッシュがクリアされるたびに、`/var/log/messages`にエントリが追加されます。時間の指定は分単位です。デフォルトは60分です。

**statistics-interval 0;** forwardingBINDは定期的にインタフェースを検索して、新しいインタフェースや存在しなくなったインタフェースがないか確認します。この値を`0`に設定すると、この検索が行われなくなり、BINDは起動時に検出されたインタフェースのみをリッスンします。0以外の値を指定する場合は分単位で指定します。デフォルトは60分です。

**notify no;** `no`に設定すると、ゾーンデータを変更したとき、またはネームサーバが再起動されたときに、他のネームサーバに通知されなくなります。

### 24.3.2 ログ

BINDでは、何を、どのように、どこにログ出力するかを詳細に設定できます。通常は、デフォルト設定のままで十分です。 例 24.3.「ログを無効にするエントリ」に、このエントリの最も簡単な形式、すなわちログをまったく出力しない例を示します。

*Example 24.3: ログを無効にするエントリ*

```
logging {
category default { null; };
};
```

### 24.3.3 ゾーンエントリ

*Example 24.4: my-domain.deのゾーンエントリ*

```
zone "my-domain.de" in {
     type master;
     file "my-domain.zone";
     notify no;
 };
```

zoneの後、管理対象のドメイン名my-domain.deを指定し、次にinと関連のオプションを中カッコで囲んで指定します(例 24.4.「my-domain.deのゾーンエントリ」参照)。スレーブゾーンを定義するには、typeをslaveに変更し、このゾーンをmasterとして管理することをネームサーバに指定します(例 24.5.「other-domain.deのゾーンエントリ」参照)。これが他のマスタのスレーブとなることもあります。

*Example 24.5: other-domain.deのゾーンエントリ*

```
zone "other-domain.de" in {
     type slave;
     file "slave/other-domain.zone";
     masters { 10.0.0.1; };
 };
```

ゾーンオプション

**type master;** masterを指定して、BINDに対し、ゾーンがローカルネームサーバによって処理されるように指示します。これは、ゾーンファイルが正しい形式で作成されていることが前提となります。

**type slave;** このゾーンは別のネームサーバから転送されたものです。必ずmastersとともに使用します。

**type hint;** ルートネームサーバの設定には、hintタイプのゾーン.を使用します。このゾーン定義はそのまま使用できます。

**file my-domain.zoneまたはfile "slave/other-domain.zone";** このエントリは、ドメインのゾーンデータが格納されているファイルを指定します。スレーブの場合、このデータを他のネームサーバから取得するので、このファイルは不要です。マスタとスレーブのファイルを区別するには、スレーブファイルにディレクトリslaveを使用します。

**masters {** ⟨*server-ip-address*⟩**; };** このエントリは、スレーブゾーンにのみ必要です。ゾーンファイルの転送元となるネームサーバを指定します。

**allow-update {! *; };** このオプションは、外部書き込みアクセスを制御し、クライアントにDNSエントリへの書き込み権を付与することができます。ただし、これは通常、セキュリティ上の理由で好ましくありません。このエントリがなければ、ゾーンの更新は完全に拒否されます。上のエントリでは、！ *によって一切の書き込みを禁止しているので、変更が完全に拒否されるという結果はこのエントリを指定しない場合と同じです。

## 24.4 ゾーンファイル

ゾーンファイルは2種類必要です。1つはIPアドレスをホスト名に割り当てるゾーンファイル、もう1つは逆にホスト名をIPアドレスに割り当てるゾーンファイルです。

**Tip**

**ゾーンファイルでのピリオドの使用**

.は、ゾーンファイル内で重要な意味を持ちます。末尾に.のホスト名を指定すると、ゾーンが追加されます。完全なホスト名を完全なドメイン名とともに指定する場合は、末尾に.を付けて、ドメインが追加されないようにします。ピリオドの打ち忘れや位置の間違いは、ネームサーバ設定エラーの原因としておそらく最も頻繁に見られるものです。

**Tip**

最初に、ドメインworld.cosmosに責任を負うゾーンファイルworld.zoneについて示します(例 24.6. 「/var/lib/named/world.zoneファイル」参照)。

***Example 24.6:*** */var/lib/named/world.zoneファイル*

```
$TTL 2D
world.cosmos. IN SOA      gateway  root.world.cosmos. (
            2003072441  ; serial
            1D          ; refresh
            2H          ; retry
            1W          ; expiry
            2D )        ; minimum

            IN NS       gateway
            IN MX       10 sun

gateway     IN A        192.168.0.1
            IN A        192.168.1.1
sun         IN A        192.168.0.2
moon        IN A        192.168.0.3
earth       IN A        192.168.1.2
mars        IN A        192.168.1.3
www         IN CNAME    moon
```

**1行目:** $TTLは、このファイルのすべてのエントリに適用されるデフォルトの寿命(time to live)です。この例では、エントリは2日間(2 D)有効です。

**2行目:** ここから、SOA (start of authority)制御レコードが始まります。

- 管理対象のドメイン名は、先頭にあるworld.cosmosです。これは、末尾に.(ピリオド)が付いています。ピリオドを付けないと、ゾーンが再度末尾に追加されてしまいます。あるいはピリオドを@で置き換えることもできます。その場合は、ゾーンが/etc/named.confの対応するエントリから抽出されます。
- IN SOAの後には、このゾーンのマスタであるネームサーバの名前を指定します。これらの名前は末尾に.(ピリオド)が付いていないので、gatewayからgateway.world.cosmosに拡張されます。
- この後には、このネームサーバの責任者の電子メールアドレスが続きます。@記号は既に特別な意味を持つので、ここでは代わりに.(ピリオド)を使用します。root@world.cosmosの場合、エントリはroot.world.cosmos.となります。ここでもゾーンが追加されないよう、.を末尾につける必要があります。
- (は、)までの行をすべてSOAレコードに含める場合に使用します。

**3行目:** シリアル番号は任意の番号で、このファイルを変更するたびに増加します。変更があった場合、セカンダリネームサーバ(スレーブサーバ)に通知する必要があります。これには、日付と実行番号をYYYYMMDDNNという形式で表記した10桁の数値が、慣習的に使用されています。

**4行目:** リフレッシュレートは、セカンダリネームサーバがゾーンserial numberを確認する時間間隔を指定します。この例では1日です。

**5行目:** 再試行間隔は、エラーが生じた場合に、セカンダリネームサーバがプライマリサーバに再度通知を試みる時間間隔を指定します。この例では2時間です。

**6行目:** 有効期限は、セカンダリネームサーバがプライマリサーバに再通知できなかった場合に、キャッシュしたデータを廃棄するまでの時間枠を指定します。この例では1週間です。

**7行目:** SOAレコードの最後のエントリは、ネガティブキャッシュTTLです。これは、DNSクエリが解決できないという他のサーバからの結果をキャッシュしておく時間です。

**9行目:** IN NSは、このドメインを担当するネームサーバを指定します。これらの名前は末尾に.(ピリオド)が付いていないので、gatewayからgateway.world.cosmosに拡張されます。このように、プライマリネームサーバと各セカンダリネームサーバに1つずつ指定する行がいくつかあります。/etc/named.confでnotifyをnoに設定しない限り、ゾーンデータが変更されると、ここにリストされているすべてのネームサーバにそれが通知されます。

**10行目:** MXレコードは、ドメインworld.cosmos宛ての電子メールを受領、処理、および転送するメールサーバを指定します。この例では、ホストsun.world.cosmosが指定されています。ホスト名の前の数字は、プリファレンス値です。複数のMXエントリが存在する場合、値が最も小さいメールサーバが最初に選択され、このサーバへのメール配信ができなければ、次に小さい値のメールサーバが試みられます。

**12〜17行目:** これらは、ホスト名に1つ以上のIPアドレスが割り当てられている実際のアドレスレコードです。ここにリストされている名前にはドメインが含まれていないので、. (ピリオド)が付いておらず、その結果、すべての名前にworld.cosmosが追加されることになります。ホストgatewayは、ネットワークカードが2枚搭載されているので、2つのIPアドレスが割り当てられます。ホストアドレスが従来型のアドレス(IPv4)の場合、レコードにAが付きます。アドレスがIPv6アドレスの場合、エントリにA6が付きます。以前は、IPv6アドレスがAAAAで示されていましたが、現在では廃止されました。

**18行目:** エイリアスwwwをmondの別名として使用できます(CNAMEは*canonical name(*キャノニカル名*)*という意味です)。

擬似ドメインin-addr.arpaは、IPアドレスからホスト名への逆引き参照に使用されます。このドメインの前に、IPアドレスのネットワーク部分が逆順に指定されます。たとえば、192.168.1は、1.168.192.in-addr.arpaに解決されます。例 24.7. 「逆引き」を参照してください。

***Example 24.7: 逆 引 き***

```
1
2 $TTL 2D 1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
3             2003072441      ; serial
4             1D              ; refresh
5             2H              ; retry
6             1W              ; expiry
7             2D )            ; minimum
8
9             IN NS           gateway.world.cosmos.
10
11 1              IN PTR          gateway.world.cosmos.
12 2              IN PTR          earth.world.cosmos.
13 3              IN PTR          mars.world.cosmos.
```

**1行目:** $TTLは、このファイルのすべてのエントリに適用される標準のTTLです。

**2行目:** この設定ファイルは、ネットワーク192.168.1.0の逆引きを有効にします。ゾーン名は1.168.192.in-addr.arpaであり、これはホスト名に追加しません。したがって、すべてのホスト名は完全な形で、つまりドメインと末尾の. (ピリオド)が付いて指定されます。残りのエントリは、前のworld.cosmosの例の記述と同じです。

**3〜7行目:** 前の例のworld.cosmosを参照してください。

**9行目:** 正引きの場合と同様、この行は、このゾーンを担当するネームサーバを指定します。ただし、ホスト名はドメインと末尾の. (ピリオド)が付いた完全な形で指定されます。

**11〜13行目:** これらはそれぞれのホスト上でのIPアドレスを示すポインタレコードです。IPアドレスの最後のオクテットのみが、行の最初に入力され、末尾に. (ピリオド)は付きません。ゾーンをこれに追加すると(.in-addr.arpaを付けずに)、完全なIPアドレスが逆順で生成されます。

通常、異なるバージョンのBIND間のゾーン転送は、問題なく行えるはずです。

## 24.5 ゾーンデータの動的アップデート

*動的*アップデートという用語は、マスタサーバのゾーンファイル内のエントリが追加、変更、削除される操作を指します。この仕組みは、RFC 2136に記述されています。動的アップデートをゾーンごとに個別に構成するには、オプションのallow-updateルールまたはupdate-policyルールを追加します。動的に更新されるゾーンを手動で編集してはなりません。

サーバに更新エントリを転送するには、nsupdateコマンドを使用します。このコマンドの詳細な構文については、nsupdateのマニュアルページ(man 8 nsupdate)を参照してください。セキュリティ上の理由から、こうした更新はTSIGキーを使用して実行するようにしてください(項24.6. 「安全なトランザクション」参照)。

## 24.6 安全なトランザクション

安全なトランザクションは、共有秘密キー(TSIGキーとも呼ばれる)に基づくトランザクション署名(TSIG)を使用して実現できます。ここでは、このキーの生成方法と使用方法について説明します。

安全なトランザクションは、異なるサーバ間の通信、およびゾーンデータの動的アップデートに必要です。アクセス制御をキーに依存する方が、単にIPアドレスに依存するよりもはるかに安全です。

TSIGキーの生成には、次のコマンドを使用します(詳細については、mandnssec-keygenを参照)。

dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2

これにより、次のような形式の名前を持つファイルが2つ作成されます。

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

キー自体(ejIkuCyyGJwwuN3xAteKgg==のような文字列)は、両方のファイルにあります。キーをトランザクションで使用するには、2番目のファイル(Khost1-host2.+157+34265.key)を、できれば安全な方法で(たとえばscpを使用して)、リモートホストに転送する必要があります。host1とhost2の間で安全な通信ができるようにするには、リモートサーバでキーをファイル/etc/named.confに含める必要があります。

```
key host1-host2.{
algorithm hmac-md5;
secret ";ejIkuCyyGJwwuN3xAteKgg==;
};
```

**Warning**

**/etc/named.confのファイルパーミッション**

/etc/named.confのファイルパーミッションが適切に制限されていることを確認してください。このファイルのデフォルトのパーミッションは0640で、オーナーがroot、グループがnamedです。この代わりに、パーミッションが制限された別ファイルにキーを移動して、そのファイルを/etc/named.conf内にインクルードすることもできます。

**Warning**

サーバhost1がhost2 (この例では、アドレス192.168.2.3)のキーを使用できるようにするには、host1の/etc/named.confに次の規則が含まれている必要があります。

```
server 192.168.2.3 {
keys { host1-host2. ;};
};
```

同様のエントリがhost2の設定ファイルにも含まれている必要があります。IPアドレスとアドレス範囲に対して定義されているすべてのACL (アクセス制御リスト—ACLファイルシステムと混同しないこと)にTSIGキーを追加してトランザクションセキュリティを有効にします。対応するエントリは、次のようになります。

```
allow-update { key host1-host2. ;};
```

このトピックについての詳細は、update-policyの下の『*BIND Administrator Reference Manual*』を参照してください。

## 24.7 DNSセキュリティ

DNSSEC、すなわちDNSセキュリティは、RFC2535に記述されています。DNSSECに利用できるツールについては、BINDのマニュアルを参照してください。

ゾーンが安全だといえるためには、1つ以上のゾーンキーが関連付けられている必要があります。キーはホストキーと同様、dnssec-keygenによって生成されます。現在、これらのキーの生成には、DSA暗号化アルゴリズムが使用されています。生成されたパブリックキーは、$INCLUDEルールによって、対応するゾーンファイルにインクルードします。

生成したすべてのキーは、dnssec-makekeysetコマンドによって1つのセットにパッケージングし、安全な方法で親ゾーンに転送する必要があります。親ゾーンでは、dnssec-signkeyによってセットに署名が付されます。このコマンドによって複数のファイルが生成され、これらのファイルを使用してdnssec-signzoneが実行され、ゾーンに署名が付されます。 このときにファイルが生成されて、各ゾーンの/etc/named.confにインクルードされます。

## 24.8 関連資料

ここで扱ったトピックの詳細については、/usr/share/doc/packages/bind/ディレクトリの『*BIND Administrator Reference Manual*』を参照してください。BINDに付属のマニュアルやマニュアルページで紹介されているRFCも、必要に応じて参照してください。/usr/share/doc/packages/bind/README.SuSEには、SUSE LINUXのBINDに関する最新情報が含まれています。

# NISの 使 用

ネットワーク上の複数UNIXシステムが共通のリソースにアクセスするようになると、すべてのユーザおよびグループ識別情報がネットワーク上のすべてのコンピュータで一致していることが重要になります。ネットワークはユーザにとって透過的でなければなりません。すなわち、使用するコンピュータに関係なく、常に、まったく同じ環境で作業できる必要があります。これを実現するのが、NISおよびNFSサービスです。NFSはネットワーク上にファイルシステムを分散させるシステムです。これについては、章 26. NFS共有ファイルシステムで説明します。

NIS (Network Information Service)は、`/etc/passwd`、`/etc/shadow`、`/etc/group`の各ファイルにネットワーク越しにアクセスできるようにするデータベースサービスと考えることができます。NISの用途はこれ以外にもありますが(`/etc/hosts`や`/etc/services`といったファイルにアクセスできるようにするなど)、ここでは触れません。NISはよく*YP*と呼ばれますが、これは、NISがちょうどネットワークの「イエローページ」のような役割を果たすためです。

# 25.1 NISサーバの設定

NISサーバを設定するには、YaSTの［‘ネットワークサービス’］モジュールから［‘NISサーバ’］を選択します。ネットワーク上にNISサーバが存在しない場合は、次の画面で［‘Install and Set up a Master NIS Server(マスタNISサーバのインストールと設定)’］を有効にしてください。既にNISサーバ(マスタ)が存在する場合は、NISスレーブサーバを追加できます(たとえば、新しいサブネットワークを設定する場合など)。最初に、マスタサーバの設定について説明します。

必要なパッケージが不足している場合は、要求されたCDまたはDVDを装着すると自動的にインストールされます。設定ダイアログの先頭にはドメイン名を入力します(図 25.1. 「NIS サーバ設定ツール」参照)。すぐ下のチェックボックスで、このホストをNISクライアントとしても使用するかどうかを指定します。このチェックボックスをオンにすると、ユーザはこのホストにログインしてNISサーバのデータにアクセスできます。


*Figure 25.1: NIS サーバ設定ツール*

後でネットワーク上にNISサーバ(スレーブ)を追加設定できるように、ここで［‘NIS スレーブサーバが存在する’］を有効にしておいてください。また、マ

スタからスレーブにデータベースエントリが高速転送されるようにするには、［'高速マップ配布'］を選択します。

ネットワーク上のユーザがyppasswdコマンドを用いてNISサーバ上のパスワードを変更できるようにするには、対応するオプションを有効にします。これにより、［'GECOS エントリ変更を許可する'］および［'ログインシェルの変更を許可する'］オプションが選択可能になります。前者を選択すると、ユーザがypchfnコマンドを使用して自分の名前とアドレスの設定を変更できるようになります。後者を選択すると、ユーザが、ypchshコマンドを使用してデフォルトのシェルを(たとえばbashからshに)変更できるようになります。

［'その他のグローバル設定'］を選択すると、図 25.2.「ディレクトリの変更とNISサーバ用の各ファイルの同期化」に示す画面が表示されます。ここでは、NISサーバのソースディレクトリを変更できます(デフォルトは/etc)。また、パスワードとグループを結合することもできます。ここで［'はい'］を選択して、/etc/passwd、/etc/shadow、および/etc/groupの各ファイル間の同期をとるようにしてください。また、最小のユーザIDとグループIDも指定します。［'OK'］をクリックして設定内容を確定すると、前の画面に戻ります。次に、［'次へ'］をクリックします。

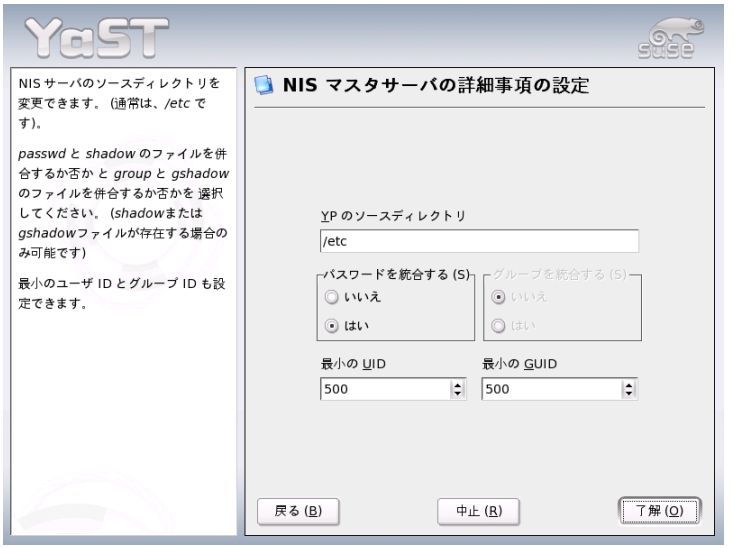


*Figure 25.2: ディレクトリの変更とNISサーバ用の各ファイルの同期化*

前の画面で［'NIS スレーブサーバが存在する'］を選択した場合は、スレーブとして使用するホスト名を入力して［'次へ'］をクリックします。スレーブサーバを使用しない場合は、スレーブ設定を省略して、データベース設定のダイアログに進んでください。ここでは、マップを指定します。マップとは、NISサーバからNISクライアントに転送される部分データベースのことです。通常は、デフォルトの設定のままで十分です。

［'次へ'］をクリックすると最後のダイアログ(図 25.3. 「NISサーバに対するリクエスト送信許可の設定」)に進みます。ここでは、NISサーバにリクエストを送信できるネットワークを指定します。通常は、内部ネットワークを指定します。その場合は、次の2つのエントリが必要です。

```
255.0.0.0       127.0.0.0
0.0.0.0         0.0.0.0
```

最初のエントリによって、自分自身、つまりNISサーバからの接続が許可されます。2つ目のエントリによって、同一ネットワークにアクセス可能なすべてのホストがNISサーバにリクエストを送信することを許可されます。

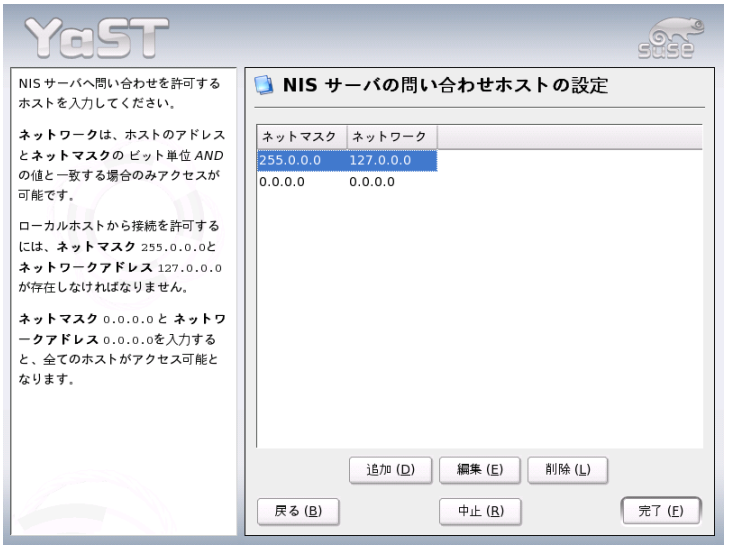


*Figure 25.3: NISサーバに対するリクエスト送信許可の設定*

**Important**

**自動ファイアウォール設定**

システムでファイアウォール(SuSEfirewall2) が有効になっている場合に、［'ファイアウォールで開いているポート'］を選択すると、YaSTは、`portmap`サービスを有効にすることでNISサーバ用にファイアウォール設定を変更します。

**Important**

## 25.2 NISクライアントの設定

このモジュールではNISクライアントを設定します。NIS、および、必要に応じてオートマウンタを使用するように選択すると、このダイアログが開きます。ホストに固定IPアドレスを割り当てるのか、DHCPから提供されたIPアドレスを使用するのかを選択してください。DHCPからは、NISドメインとNISサーバも提供されます。DHCPについては、章 27. DHCPを参照してください。固定IPアドレスを使用する場合は、NISドメインとNISサーバを手動で指定します。詳細については、図 25.4. 「NISサーバのドメインとアドレスの設定」を参照してください。［'検索'］をクリックすると、ネットワーク上でアクティブなNISサーバが検索されます。

1つのデフォルトドメインの他に、複数のドメインを指定することができます。［'追加'］を使用して、個々のドメインに対してブロードキャストを発行できる複数のサーバを指定してください。

クライアントが使用しているサーバを他のホストに知られたくない場合は、エキスパート設定で、［'ローカルホストにのみ応答する'］を有効にします。［'ブロークンサーバ'］を有効にすると、クライアントが、特権のないポートを介して通信するサーバからの応答を受信できるようになります。詳細については、`man ypbind`を参照してください。

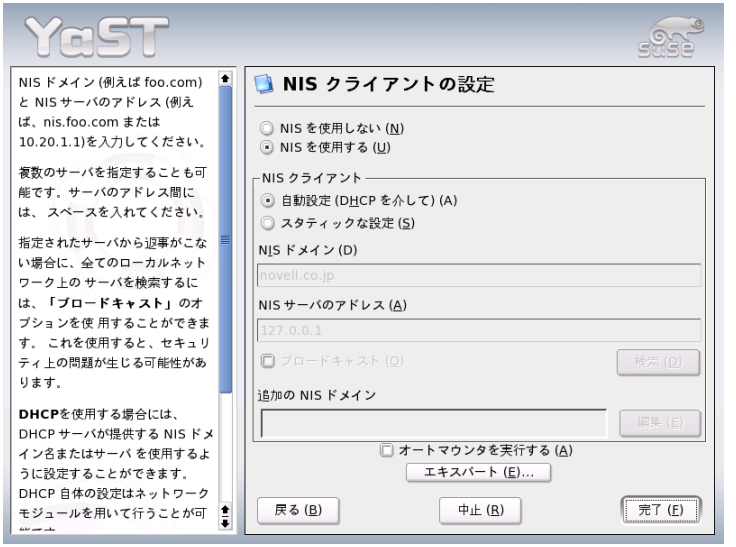


*Figure 25.4: NISサーバのドメインとアドレスの設定*

# NFS共有ファイルシステム

章 25. NISの使用で説明したように、NFSをNISと連係して使用すると、ネットワークをユーザにとって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ファイルシステムを分散できます。ユーザはどの端末からログインしても、常に、同じ環境で作業できます。

NISと同様、NFSは非対称サービスで、NFSサーバとNFSクライアントがあります。ファイルシステムをネットワーク経由で提供し(エクスポート)、同時に他のホストからファイルシステムをマウントする(インポート)ことができます。一般に、これらは大容量のハードディスクを搭載したサーバであり、そのファイルシステムが他のクライアントによってマウントされます。

**Important**

**DNSの必要性**

原則として、すべてのエクスポートはIPアドレスのみを使用して実行できます。ただし、タイムアウトを回避するために、実際に動作するDNSシステムを用意しておく必要があります。mountdデーモンは逆引きを行うため、少なくともログ目的にはDNSが必要です。

**Important**

# 26.1 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'］モジュールを使用するのが最も簡単です。NFSサーバのホスト名、インポートするディレクトリ、およびこのディレクトリをマウントするマウントポイントを入力するだけです。この操作はすべて、最初のダイアログボックス(図 26.1. 「YaSTによるNFSクライアント設定」)で［'追加'］をクリックした後に行います。

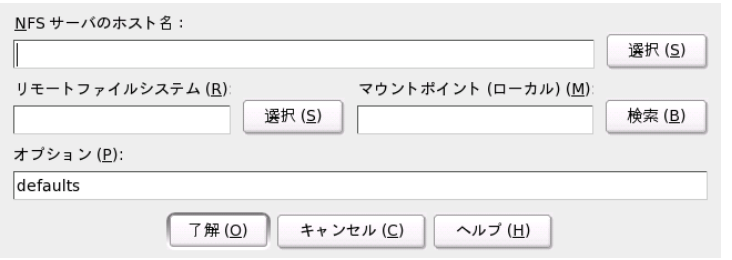


*Figure 26.1: YaSTによるNFSクライアント設定*

## 26.2 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サーバから手動で容易にインポートできます。唯一の前提条件はRPCを実行していることです。RPCを起動するにはrootユーザとして「rcportmapstart」コマンドを入力します。この前提条件さえ満たせば、それぞれのマシン上でエクスポートされたリモートファイルシステムを、自マシンのファイルシステムにマウントして、ローカルのハードディスクのように使用することができます。それには、mountコマンドを次の構文で使用します。

```
mount host:remote-path local-path
```

たとえば、マシンsun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 sun:/home /home
```

## 26.3 YaSTによるファイルシステムのエクスポート

YaSTを使用して、ネットワーク上のホストをNFSサーバに変更し、そのホストへのアクセスを許可されたすべてのホストに、ディレクトリやファイルをエクスポートすることができます。これにより、グループに属する全社員がアプリケーションをそれぞれのホストにローカルにインストールしなくても、全員にアプリケーションを提供できるようになります。NFSサーバをインストールするには、YaSTを起動して、'ネットワークサービス' → 'NFSサーバ'の順に選択します。図 26.2. 「NFSサーバ設定ツール」に示すダイアログが開きます。

次に、［'NFSサーバを起動する'］を有効にし、［'次へ'］をクリックします。上部のテキストフィールドに、エクスポートするディレクトリを入力します。下部に、それらのディレクトリへのアクセスを許可するホストを入力します。図 26.3. 「YaSTによるNFSサーバの設定」に示すダイアログボックスが表示されます。各ホストに対して設定できるオプションは、［単独のホスト］、［ネットグループ］、［ワイルドカード］、および［IPネットワーク］の4つです。これらのオプションの詳細については、man exportsを参照してください。［'完了'］を選択して、設定を完了させます。

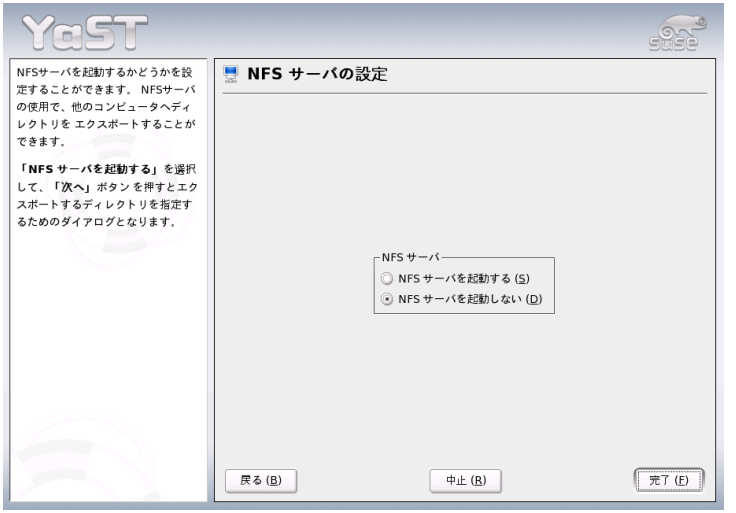


*Figure 26.2: NFSサーバ設定ツール*

**Important**

**自動ファイアウォール設定**

システムでファイアウォール(SuSEfirewall2)が有効になっている場合に、［'ファイアウォールで開いているポート'］を選択すると、YaSTは、`nfs`サービスを有効にすることでNFSサーバ用にファイアウォール設定を変更します。

**Important**

## 26.4 ファイルシステムの手動エクスポート

YaSTを使用しない場合は、以下のシステムがNFSサーバ上で稼動していることを確認します。

- RPCポートマッパー(portmap)
- RPCマウントデーモン(rpc.mountd)

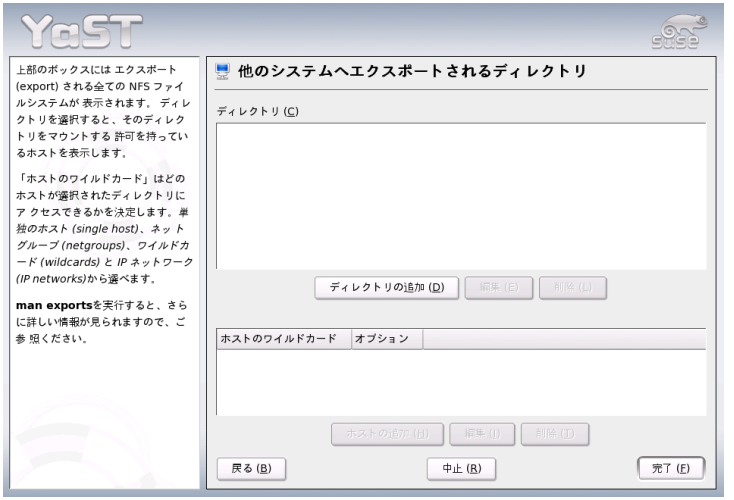


***Figure 26.3:*** *YaSTによるNFSサーバの設定*

- RPC NFSデーモン(rpc.nfsd)

`/etc/init.d/portmap`スクリプトと`/etc/init.d/nfsserver`スクリプトを使用して、システムの起動時にこれらのサービスを起動するには、「`insserv/etc/init.d/nfsserver`」コマンドと「`insserv/etc/init.d/portmap`」コマンドを入力します。 また、どのファイルシステムを、どのホストにエクスポートするかを設定ファイル`/etc/exports`に定義します。

エクスポートするディレクトリごとに1行を指定して、どのマシンがどのようなパーミッションでそのディレクトリにアクセスできるかを設定します。このディレクトリのすべてのサブディレクトリも、自動的にエクスポートされます。許可するマシンは、通常、フルネーム(ドメイン名付き)で指定しますが、*や?(Bashシェルと同様に展開)のようなワイルドカードを使用することもできます。ここでマシンを指定しない場合、指定したパーミッションで、すべてのマシンがこのファイルシステムにアクセスできます。

エクスポートファイルシステムのパーミッションを、マシン名の後にカッコで囲んで設定します。重要なオプションを表 26.1.「エクスポートされるファイルシステムのパーミッション」に示します。

*Table 26.1: エクスポートされるファイルシステムのパーミッション*

| オプション | 意味 |
|---|---|
| ro | ファイルシステムを読み込み専用(read only)パーミッションでエクスポートします(デフォルト)。 |
| rw | ファイルシステムを読み書き可能パーミッションでエクスポートします。 |
| root_squash | インポート側ホストのrootユーザが、このファイルシステムでrootパーミッションを持たないようにします。そのために、ユーザID 0 (rootユーザのID)に、ユーザID 65534が割り当てられます。このユーザIDは、nobody (デフォルト)に設定する必要があります。 |
| no_root_squash | ユーザID 0をユーザID 65534に割り当てず、rootユーザのパーミッションを有効なままにします。 |
| link_relative | 絶対リンク(/で始まるリンク)を../に変換します。これは、マシンのファイルシステム全体がマウントされている場合(デフォルト)のみ使用できます。 |
| link_absolute | シンボリックリンクを変更しません。 |
| map_identity | 各ユーザIDがクライアントとサーバの両方で一致します(デフォルト)。 |
| map_daemon | クライアントとサーバに、一致するユーザIDがありません。この結果、nfsdによってユーザIDの変換テーブルが作成されます。変換テーブルの作成には、ugiddデーモンが必要です。 |

exportsファイルの例を、例 26.1.「/etc/exports」に示します。/etc/exportsが、mountdとnfsdによって読み込まれます。ファイルをまったく変更しない場合は、mountdとnfsdを再起動して、変更内容を有効にします。再起動は、rcnfsserverrestartによって簡単に実行できます。

***Example 26.1: /etc/exports***

```
#
   # /etc/exports #
```

```
/home            sun(rw)   venus(rw)
/usr/X11         sun(ro)   venus(ro)
/usr/lib/texmf   sun(ro)   venus(rw)
/                earth(ro,root_squash)
/home/ftp        (ro)
# End of exports
```

# DHCP

*dynamic host configuration protocol* (DHCP)の目的は、ネットワーク環境設定を各ワークステーションでローカルに行うのではなく、サーバから一元的に割り当てることです。DHCPを使用するように設定されたクライアントは、自身の静的アドレスを制御できません。サーバからの指示に従って、すべてが自動的に設定されるからです。

DHCPの使用法の1つとして、ネットワークカードのハードウェアアドレス(ほとんどの場合、固定)を使用して各クライアントを識別し、そのクライアントがサーバに接続するたびに同じ設定を提供する方法があります。DHCPはまた、サーバが用意したアドレスプールから、アドレスを各クライアントに動的に割り当てるように設定することもできます。後者の場合、DHCPサーバはクライアントから要求を受信するたびに、接続が長期にわたる場合でも、クライアントに同じアドレスを割り当てようと試みます。当然ですが、これは、ホスト数がアドレス数を超えていない場合にのみ機能します。

DHCPはこれらの機能を提供することによって、システム管理者の作業負荷を2つの点で軽減します。サーバの環境設定ファイルを編集して、アドレスに関するあらゆる変更(大きな変更であっても)と一般的なネットワークの環境設定を一元的に実装できます。これは、多数のワークステーションをいちいち再設定するのに比べるてはるかに簡単です。また、特に新しいマシンをネットワークに統合する場合、IPアドレスをプールから割り当てられるので、作業が楽になります。適切なネットワークの環境設定をDHCPサーバから取得する方法は、日常的に、ラップトップをさまざまなネットワークで使用する場合に特に便利です。

DHCPサーバは、クライアントが使用するIPアドレスとネットマスクを供給するだけでなく、ホスト名、ドメイン名、ゲートウェイ、およびネームサーバアドレスも供給します。この他にも、DHCPを使用して一元的に設定できるパラメータがあり、たとえば、クライアントが現在時刻をポーリングするタイムサーバやプリントサーバも設定可能です。

## 27.1 YaSTによるSambaサーバの設定

モジュールを初めて起動すると、YaSTは4つのステップで構成される環境設定ウィザードを起動します。このウィザードの指示に従って基本的なDHCPサーバをセットアップできます。

**ネットワークインタフェースの選択** 最初のステップでは、YaSTによりシステムで使用可能なネットワークインタフェースが検査され、リスト形式で表示されます。このリストから、DHCPサーバがリッスンするインタフェースを選択し、［'選択されたインタフェースでファイアウォールを開く'］をクリックしてそのインタフェースのファイアウォールを開きます。図 27.1.「DHCPサーバ:ネットワークインタフェースの選択」を参照してください。

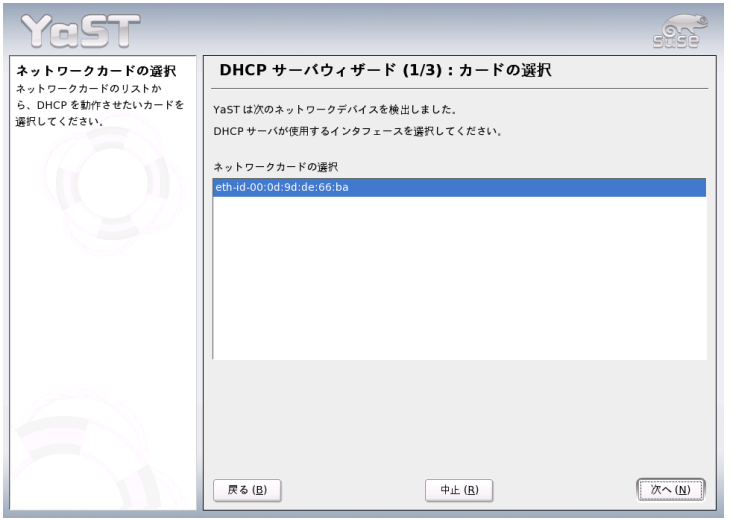


*Figure 27.1: DHCPサーバ:ネットワークインタフェースの選択*

**グローバル設定** エントリフィールドに、DHCPサーバで管理する全クライアントのネットワークを指定します。この指定には、ドメイン名、タイムサーバのアドレス、プライマリネームサーバとセカンダリネームサーバのアドレス、印刷サーバとWINSサーバのアドレス(Windowsクライアントと Linuxクライアントの両方が混在するネットワークを使用する場合)、ゲートウェイアドレスおよびリース期間が含まれます。図 27.2. 「DHCPサーバ:グローバル設定」を参照してください。

**動的DHCP** このステップでは、クライアントに対する動的IPアドレスの割り当て方法を設定します。そのためには、サーバがDHCPクライアントに割当て可能なIPアドレスの範囲を指定します。これらのアドレスは、すべて同じネットマスクを使用する必要があります。また、クライアントがリースの延長を要求せずにIPアドレスを維持できるリース期間も指定します。必要に応じて、最大リース期間、つまりサーバが特定のクライアントのIPアドレスを保持している期間を指定します(図 27.3. 「DHCPサーバ:動的DHCP」を参照)。

**環境設定の完了と実行モードの設定** 環境設定ウィザードの3つ目の手順を終了すると、最後にDHCPサーバの起動方法を定義するダイアログが表示

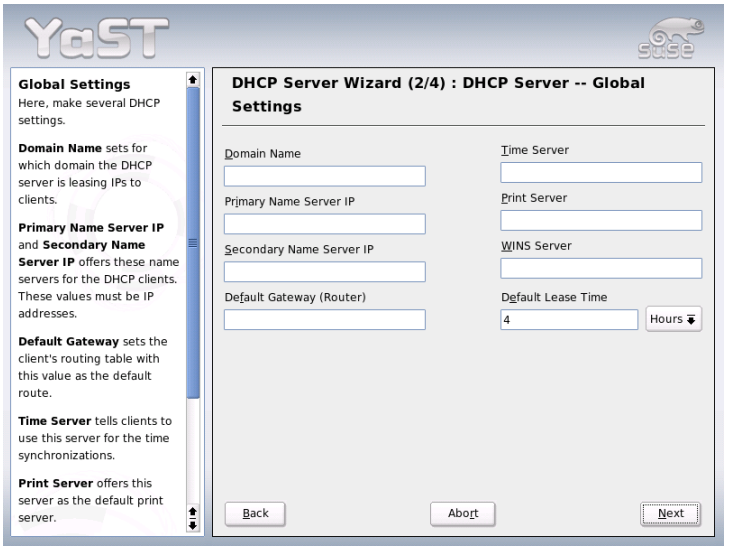


*Figure 27.2: DHCPサーバ:グローバル設定*

されます。ここでは、システムのブート時にDHCPサーバを自動的に起動するか、テスト時など必要に応じて手動で起動するかを指定します。［'完了'］をクリックして、サーバの環境設定を完了します。図 27.4.「DHCPサーバ:起動」を参照してください。

## 27.2 DHCPソフトウェアパッケージ

SUSE LINUXでは、DHCPサーバとDHCPクライアントのどちらも利用可能です。用意されているDHCPサーバは、dhcpd (Internet Software Consortium製)です。クライアント側では、DHCPクライアントプログラムとして、dhclient (同じくISC製)またはdhcpcdパッケージのDHCPクライアントデーモンのどちらかを選択できます。

SUSE LINUXは、デフォルトでdhcpcdをインストールします。このプログラムは非常に扱いやすく、システムブート時に自動的に起動して、DHCPサーバを監視します。環境設定ファイルは必要ありません。標準的な設定であればほとんどの場合、そのまま使用できます。複雑な状況で使用する場合は、環境設

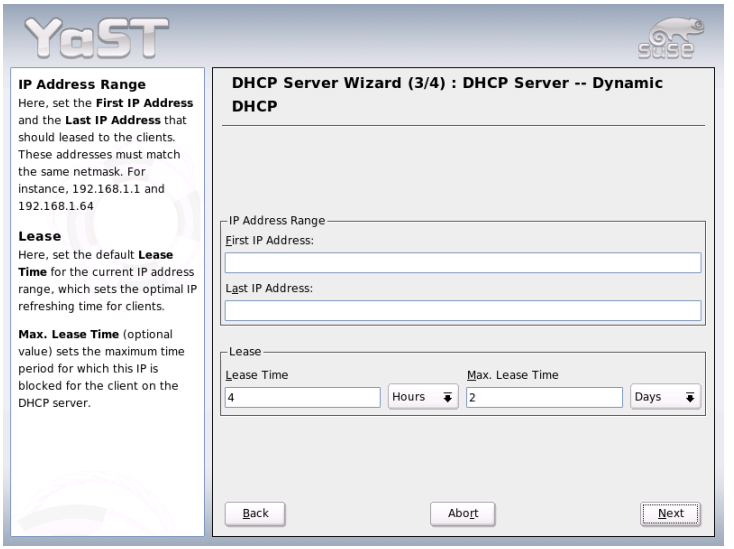


*Figure 27.3: DHCPサーバ:動的DHCP*

定ファイル/etc/dhclient.confによって制御されるISC dhclientを使用します。

## 27.3 DHCPサーバdhcpd

DHCPシステムの中核には、動的ホスト環境設定プロトコルデーモンがあります。このサーバは、環境設定ファイル/etc/dhcpd.confに定義された設定に従ってアドレスをリースし、その使用状況を監視します。システム管理者は、このファイルのパラメータと値を変更して、プログラムの動作をさまざまな方法で調整できます。例 27.1.「環境設定ファイル/etc/dhcpd.conf」で、/etc/dhcpd.confファイルの基本的な例を見てみましょう。

*Example 27.1: 環境設定ファイル/etc/dhcpd.conf*

```
default-lease-time 600;          # 10 minutes
```

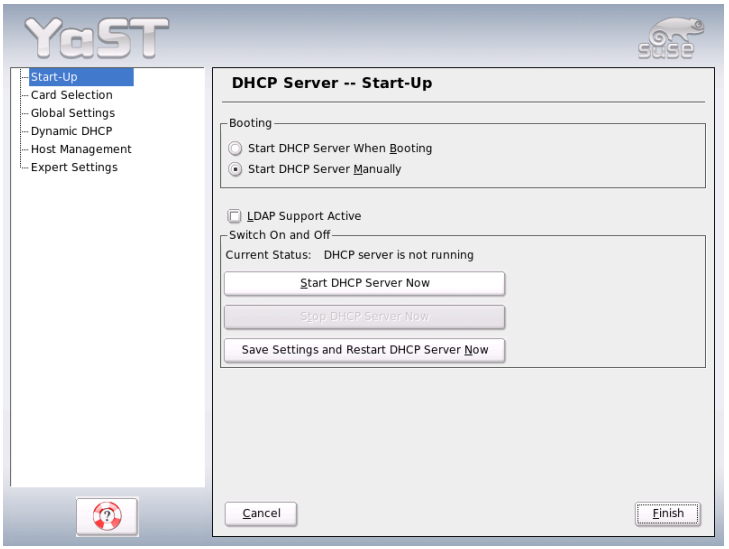


*Figure 27.4: DHCPサーバ:起動*

```
   max-lease-time 7200;            # 2  hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
{
 range 192.168.1.10 192.168.1.20;
 range 192.168.1.100 192.168.1.200;
}
```

DHCPサーバを用いてネットワーク内でIPアドレスを割り当てるには、このサンプルのような環境設定ファイルを用意すれば十分です。各行の末尾にセミコロンが付いていることに注意してください。これがなければ、dhcpdは起動しません。

このサンプルファイルは、3つのセクションに分けられます。最初のセクションは、要求側クライアントにIPアドレスがリースされた場合に、デフォルトで

最大何秒間経過すればリースの更新が必要になるか(デフォルトリース時間)が定義されます。このセクションには、DHCPサーバがマシンにIPアドレスを割り当てた場合に、マシンが更新を求めずにそのIPアドレスを保持できる最大時間(max-lease-time)も指定されています。

2つ目のセクションでは、基本的なネットワークパラメータがグローバルレベルで定義されています。

- option domain-nameの行は、ネットワークのデフォルトドメインを定義してます。
- option domain-name-serversエントリには、IPアドレスをホスト名(また逆方向に)に解決するためのDNSサーバを最高3つを指定します。ネームサーバは、DHCPをセットアップする前に、使用しているマシン上またはネットワーク上のどこか他の場所で設定するのが理想的です。ネームサーバではまた、各ダイナミックアドレスに対してホスト名を定義し、またその逆も定義する必要があります。独自のネームサーバを設定する方法については、章 24. ドメインネームシステムを参照してください。
- option broadcast-addressの行は、要求側クライアントが使用するブロードキャストアドレスを定義します。
- option routersの行では、ローカルネットワークでホストに配信できないデータパケットの送信先を(指定されたソース／ターゲットホストアドレスおよびサブネットに応じて)サーバに指示します。ほとんどの場合、特に小規模ネットワークでは、このルータはインターネットゲートウェイと同一です。
- option subnet-maskでは、クライアントに割り当てるネットマスクを指定します。

ファイルの最後のセクションでは、サブネットマスクを含め、ネットワークを定義します。最後に、DHCPが対象のクライアントにIPアドレスを割り当てるために使用するアドレス範囲を指定します。この例では、クライアントは192.168.1.10～192.168.1.20および192.168.1.100～192.168.1.200の範囲にある任意のアドレスを与えられます。

これら数行を編集すると、rcdhcpdstartコマンドを使用してDHCPデーモンを有効にできるようになります。DHCPデーモンはすぐに使用できます。rcdhcpdcheck-syntaxコマンドを使用すると、簡単な構文チェックを実

行できます。サーバでエラーが発生して中断する、起動時にdoneが返されないなど、環境設定に関して予期しない問題が発生した場合は、メインシステムログ/var/log/messagesまたはコンソール10 (Ctrl-Alt-F10)で情報を探せば、原因が突き止められます。

SUSE LINUXシステムで、セキュリティを確保するためにchroot環境からDHCPデーモンを起動します。デーモンが見つけられるように、環境設定ファイルは、chroot環境にコピーします。このファイルは、rcdhcpdstartコマンドによって自動的にこのファイルがコピーされるので、通常は、手動でコピーする必要はありません。

### 27.3.1 固定IPアドレスを持つホスト

前述のように、DHCPを使用すると、特定のクライアントが要求を行うたびに事前に定義した静的アドレスを割り当てることができます。明示的に割り当てられるアドレスは、プールから割り当てられる動的アドレスに常に優先します。また、たとえばアドレスが不足していて、サーバがクライアント間でアドレスを再配布する必要がある場合でも、静的アドレスは動的アドレスと違って期限切れになりません。

静的アドレスを割り当てられたホストを識別するために、dhcpdは、ハードウェアアドレスを使用します。ハードウェアアドレスは、6つのオクテットペアで構成される世界で唯一の固定数値コードで、すべてのネットワークデバイスの識別に使用されます(たとえば、00:00:45:12:EE:F4)。たとえば、例 27.2. 「環境設定ファイルへの追加」のような数行を例 27.1. 「環境設定ファイル/etc/dhcpd.conf」に示す環境設定ファイルに追加すると、DHCPデーモンはあらゆる状況で、対応するホストに常に同じデータのセットを割り当てます。

*Example 27.2: 環境設定ファイルへの追加*

```
host earth {
hardware ethernet 00:00:45:12:EE:F4;
fixed-address 192.168.1.21;
}
```

対応するクライアントの名前(host〈クライアント名〉、ここではearth)を1行目に、MACアドレスを2行目に入力します。Linuxホストでこのアドレスを確認するには、ifstatusコマンドの後にネットワークデバイス(たとえば、eth0)指定して実行します。必要に応じてifupeth0を実行し、ネットワークカードを有効にします。出力例を次に示します。

```
link/ether 00:00:45:12:EE:F4
```

上の例では、MACアドレス00:00:45:12:EE:F4を持つネットワークカードが装着されたクライアントに、IPアドレス192.168.1.21とホスト名earthが自動的に割り当てられます。指定するハードウェアの種類は、ほとんどの場合ethernetですが、IBMシステムでよく使用されるtoken-ringもサポートされています。

### 27.3.2 SUSE LINUXのバージョン

セキュリティ向上のため、SUSEバージョンのISC製DHCPサーバには、Ari Edelkind氏開発の非root/chrootパッチが付属しています。これにより、dhcpdをユーザID nobodyで実行したり、chroot環境で実行したりできます(/var/lib/dhcp)。これの機能を使用するには、環境設定ファイルdhcpd.confが/var/lib/dhcp/etcに存在する必要があります。initスクリプトは、起動時に環境設定ファイルをこのディレクトリに自動的にコピーします。

この機能に関するサーバの動作は、環境設定ファイル/etc/sysconfig/dhcpdのエントリを使用して制御できます。非chroot環境でdhcpdを実行するには、/etc/sysconfig/dhcpd内の変数DHCPD_RUN_CHROOTEDを“no”に設定します。

chroot環境内であっても、dhcpdを有効にしてホスト名を解決するには、次のような他の環境設定ファイルをコピーする必要があります。

- /etc/localtime
- /etc/host.conf
- /etc/hosts
- /etc/resolv.conf

これらのファイルは、initスクリプトの起動時に、/var/lib/dhcp/etc/にコピーされます。コピーされたファイルが/etc/ppp/ip-upのようなスクリプトによって動的に変更されている場合は、必要な変更箇所がないか注意する必要があります。ただし、環境設定ファイルに(ホスト名でなく) IPアドレスだけを指定している場合は、これについて考える必要はありません。

環境設定の中に、chroot環境にコピーすべき追加ファイルが存在する場合は、ファイルetc/sysconfig/dhcpdの変数DHCPD_CONF_INCLUDE_FILESに、

これらのファイルを指定します。syslogデーモンの再起動後もDHCPロギング機能が作動していることを確認するには、ファイル/etc/sysconfig/syslogのSYSLOGD_PARAMSにオプション"-a /var/lib/dhcp/dev/log"を追加する必要があります。

## 27.4 関連資料

DHCPの詳細については、*Internet Software Consortium*のWebサイト(http://www.isc.org/products/DHCP/)を参照してください。また、dhcpd、dhcpd.conf、dhcpd.leases、およびdhcp-optionsの各マニュアルページにも詳細が記載されています。

# xntpによる時刻の同期

NTP(ネットワーク時刻同期用プロトコル)メカニズムはネットワークを介してシステムタイムを同期す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。絶対時間の管理、ネットワーク上にあるマシン全体のシステム時間の同期が、2重の目標になっています。

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、通常すべてのマシンのシステムタイムを同期させておかなければなりませんが、手動での調整は適切な方法ではありません。xntpではこれらの問題を解決するメカニズムを備えています。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。

## 28.1 ネットワークでのxntp構成

xntpは、ローカルのコンピュータクロックを時刻の標準として参照するように事前に設定されています。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。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、ネットワークから`ntp.example.com`という名前のタイムサーバに到達できる場合は、`server ntp.example.com`という行を追加して、ファイル`/etc/ntp.conf`にこのサーバ名を追加します。別のタイムサーバを追加するには、別の行にキーワードserverを挿入します。`rcxntpd start`コマンドでxntpdを初期化すると、アプリケーションは時計が安定するまで1時間待機し、ドリフトファイルを作成してローカルコンピュータのクロックを修正します。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。

クライアントとしてNTPメカニズムを使用する方法が2つあります。1つ目はクライアントが定期的に既知のサーバに対し、時刻を照会する方法です。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ることです。

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。

## 28.2 ローカルリファレンスクロックの設定

ソフトウェアパッケージxntp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、xntp-docパッケージのファイル/usr/share/doc/packages/xntp-doc/html/refclock.htm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。xntpの実際の設定は、疑似IP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ファイル/etc/ntp.conf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。ファイル/usr/share/doc/packages/xntp-doc/html/driverNN.htm(ここでNNはドライバの番号)は特定のクロックタイプに関する情報を提供します。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。xntp-docパッケージのインストール後に、ディレクトリ/usr/share/doc/packages/xntp-doc/htmlにあるxntpのマニュアルを参照してください。これらのパラメータについては、説明のあるドライバページへのリンクがファイル/usr/share/doc/packages/xntp-doc/html/refclock.htmに記述されています。

## 28.3 YaSTでのNTPクライアントの設定

このようなxntpの手動設定に加え、SUSE LINUXではYaSTを使用してNTPクライアントを設定できます。簡易設定と詳細設定の2つの設定方法があります。これら2つの方法について次に説明します。

### 28.3.1 NTPクライアントの簡易設定

NTPクライアントの簡易設定では、2つのダイアログを使用します。最初のダイアログでは、xntpdの実行モードおよびクエリ先のNTPサーバを設定します。システムのブート時にxntpdを自動起動させるには、[‘システムブート時’]をクリックします。次に、[‘選択’]をクリックして2番目のダイアログを開きます。このダイアログでは、使用しているネットワークに適したタイムサーバを選択します。

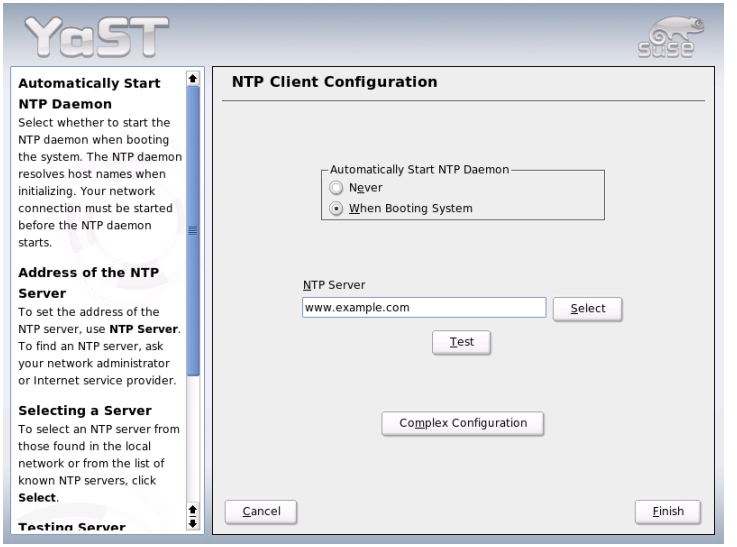


*Figure 28.1: YaST: NTPクライアントの設定*

サーバ選択用の詳細ダイアログでは、ローカルネットワーク上のタイムサーバとインターネット上のタイムサーバ([‘公開NTPサーバ’])のどちらを使用して時刻の同期を行うかを指定します。ローカルタイムサーバを使用する場合は、[‘検索’]をクリックして、ネットワーク上の利用可能なタイムサーバを問い合わせるSLPクエリを実行します。検索結果のリストから最適なタイムサーバを選択し、[‘了解’]をクリックしてダイアログを閉じます。インターネット上の公開タイムサーバを使用する場合は、国(タイムゾーン)および適切なタイムサーバを[‘公開NTPサーバ’]のリストから選択し、[‘了解’]をクリックしてダイアログを閉じます。メインダイアログで、[‘テスト’]をクリックして選択し

たサーバが利用可能かどうかをテストし、['完了']をクリックしてダイアログを閉じます。

### 28.3.2 NTPクライアントの詳細設定

NTPクライアントの詳細設定は、簡易設定の項目で説明した実行モードを選択した後、['NTPクライアント']モジュールのメインダイアログの['詳細設定'](図 28.1. 「YaST: NTPクライアントの設定」を参照)をクリックすると表示されます。

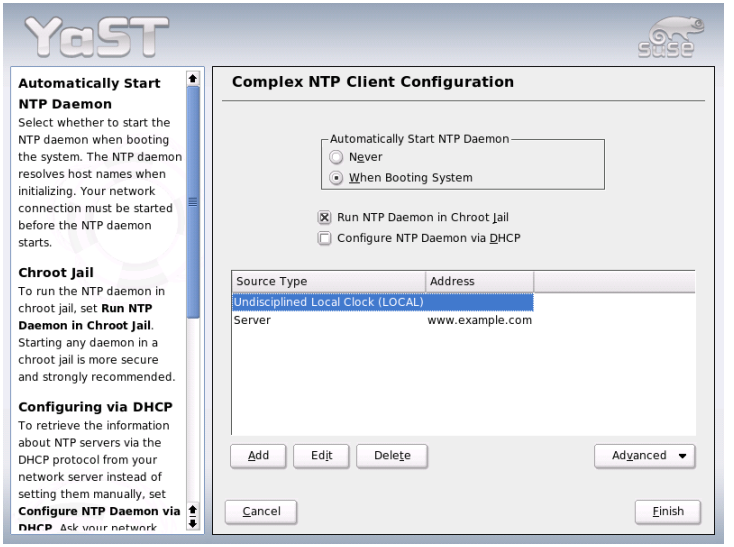


*Figure 28.2: YaST: NTPクライアントの詳細設定*

['NTPクライアントの詳細設定']で、xntpdをchroot jailで実行するかどうかを指定します。このオプションは、xntpd上の攻撃に対するセキュリティを強化し、不正ユーザによってシステム全体が危険な状態に陥ることを防ぎます。['DHCPからNTPデーモンを設定']は、ローカルネットワーク上のNTPサーバのリストをDHCP経由で取得するようにNTPクライアントを設定します。

ダイアログ下部には、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。 [‘詳細’]では、クライアントのログファイルを表示することや、NTPクライアント設定に合わせてファイアウォール設定を調整することができます。

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。使用可能なオプションは次のとおりです。

**サーバ:** [同期相手のタイプの選択]ダイアログで、(項28.3.1. 「NTPクライアントの簡易設定」で説明したように)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。入力フィールドでは、xntpdの追加オプションを指定できます。詳細については、`/usr/share/doc/packages/xntp-doc`を参照してください。

**ピア** ピアとは対の関係にあるマシンを意味します。たとえば、時刻サーバとクライアントのどちらの役割にもなります。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。ローカルのラジオクロックに関する詳細な情報は`/usr/share/doc/packages/xntp-doc/html/refclock.htm`を参照してください。

**ブロードキャスト** 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。

**ブロードキャストパケットを受け取る**
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。

29



# LDAP—ディレクトリサービス

LDAP (Lightweight Directory Access Protocol)は、情報ディレクトリへのアクセスと管理を行うために設計されたプロトコルセットです。LDAPは、ユーザおよびグループ管理、システム構成の管理、アドレス管理など、さまざまな目的に使用できます。この章では、LDAPの動作原理とYaSTを使用したLDAPデータの管理方法の基本事項について説明します。

ネットワーク環境では、重要な情報をすぐに利用できるように整理しておくことは不可欠です。そのため、一般的に使用されているイエローページのようなディレクトリサービスを使用して、情報を整理し、すぐに検索できる形式にしておくことができます。

理想的なケースは、一元的なサーバでデータをディレクトリに保持し、特定のプロトコルを使用してそれをすべてのクライアントに配布するという形態です。データはさまざまなアプリケーションがアクセスできる方法で整理されます。この方法では、個々のカレンダツールや電子メールクライアントが独自のデータベースを持つ必要はありません。一元的なリポジトリにアクセスすればよいためです。これにより、情報管理のための負荷も大幅に軽減されます。LDAP (lightweight directory access protocol)のようなオープンで標準化されたプロトコルを使用すれば、可能な限り多くの異なるクライアントアプリケーションが、このような情報にアクセスできるようになります。

この文脈でのディレクトリとは、高速かつ効果的に読み込みと検索ができるように最適化された一種のデータベースです。

- 膨大な(同時)読み込みとアクセスを可能にするため、書き込みアクセスは、管理者による少量の更新作業に限られます。従来のデータベースは、できる限り大量のデータを短時間に受け付けられるように最適化されます。
- 書き込みアクセスは制約された形でのみ可能なため、ディレクトリサービスは、ほとんどが変更のない静的情報の管理に使用されます。一般に、非常に頻繁に変更されるデータ(*動的*データ)は、従来のデータベースに保存されます。たとえば、企業ディレクトリにある電話番号は、経理で管理する数字ほど頻繁に変更されません。
- 静的データを管理する場合、既存のデータセットの更新は非常にまれです。動的データ、特に銀行口座や経理のデータセットが関与する場合、データの一貫性が最重要課題となります。たとえばある項目から差し引かれた金額を他の項目に加算する場合、データストックで残高を正しく維持するためには、1回のトランザクション内で両方の操作が同時に行われる必要があります。データベースはこのようなトランザクションをサポートしますが、ディレクトリではサポートされません。ディレクトリでは、短期的にデータの一貫性が崩れても大きな問題にはなりません。

LDAPなどのディレクトリサービスの設計には、複雑な更新やクエリメカニズムのサポートは含まれません。このサービスにアクセスするすべてのアプリケーションが、すばやく簡単にアクセスできることが主な課題です。

Unixでも他のシステムでも、多くのディレクトリサービスがこれまでに存在し、今なお存在しています。いくつか例を挙げると、Novell NDS、Microsoft ADS、BanyanのStreet Talk、OSI標準のX.500などがあります。LDAPは元々、DAP (directory access protocol)の無駄な機能を省略したサービスであり、X.500へのアクセスを目的として開発されました。X.500標準は、ディレクトリエントリの階層構造を規定しています。

LDAPは、DAPの簡易版です。LDAPでは、X.500エントリ階層が維持されているため、プラットフォーム非依存という特長を持ち、必要なリソースも少なくて済みます。TCP/IPを使用することにより、ドッキングアプリケーションとLDAPサービス間のインタフェースが、非常に簡単に確立できます。

一方でLDAPは、X.500サポートとは別に進化し、スタンドアロンソリューションとして採用されることが多くなっています。LDAPはLDAPv3の*照会*(パッケージ`openldap2`のプロトコルバージョン)をサポートすることによって、分散データベースを実現しています。SASL (simple authentication and security layer)も新しく採用されています。

LDAPの機能は、当初の計画ではX.500サーバにデータを問い合わせることだけでしたが、現在はそれだけにとどまりません。slapdというオープンソースサーバが存在し、オブジェクト情報をローカルデータベースに格納できます。また複数のLDAPサーバへのレプリケートを行うslurpdという拡張機能もあります。

`openldap2`パッケージの構成は、次のとおりです。

**slapd** スタンドアロンのLDAPv3サーバ。オブジェクト情報をBerkeleyDBベースのデータベースで管理します。

**slurpd** このプログラムは、データの変更をローカルLDAPサーバから、ネットワーク上にインストールされた他のLDAPサーバへレプリケートします。

**システム管理用の追加ツール** `slapcat`、`slapadd`、`slapindex`

## 29.1 LDAPとNISの比較

Unix系システムの管理者は、従来から、ネットワーク内の名前の解決やデータ配信にNISサービスを使用しています。設定データは/etc内のファイルに保存され、`group`、`hosts`、`mail`、`netgroup`、`networks`、`passwd`、`printcap`、`protocols`、`rpc`、および`services`の各ディレクトリは、ネットワーク内の複数のクライアント

に分散されています。これらのファイルはシンプルテキストファイルのため、保守にそれほどの手間はかかりません。しかし、構造化されていないため、大量のデータを処理することがますます困難になっています。NISはUnix系プラットフォーム専用に設計されているため、異種ネットワークでの一元的データ管理には採用できません。

LDAPサービスはNISと異なり、純粋なUnix系ネットワークに制限されていません。Windowsサーバ(2000以降)は、LDAPをディレクトリサービスとしてサポートします。NovellもまたLDAPサービスを提供します。前述のアプリケーションタスクは、Unix系以外のシステムでもサポートされます。

LDAPの原則は、一元管理が必要なあらゆるデータ構造に適用可能です。いくつかの例を次に示します。

- NISサービスの代替としての採用
- メールルーティング(postfix、sendmail)
- Mozilla、Evolution、およびOutlookなどのメールクライアント用アドレス帳
- BIND9ネームサーバのゾーン記述の管理

LDAPはNISと異なり拡張できるため、これら以外にも広範な用途が考えられます。データが明確に定義された階層構造になっているため、検索が容易であり、大量データの管理が非常に容易になります。

## 29.2 LDAPディレクトリツリーの構造

LDAPディレクトリは、ツリー構造です。ディレクトリのすべてのエントリ(オブジェクトと呼びます)には、この階層内に定義された位置があります。この階層は*ディレクトリ情報ツリー(Directory Information Tree)*またはその短縮形DITと呼ばれます。対象のエントリへの完全パスは、*識別名*(DN)と呼ばれ、確実にエントリを識別します。このエントリへのパス上にある個々のノードを*相対識別名*(RDN)と呼びます。オブジェクトは、一般的に、2つのタイプのいずれかに割り当てられます。

**コンテナ** これらのオブジェクトは、それ自体に他のオブジェクトを持っています。オブジェクトクラスには`root`(ディレクトリツリーのルート要素。実際には存在しません)、`c`(国)、`ou`(組織単位)、`dc`(ドメインコンポーネント)があります。このモデルは、ファイルシステムのディレクトリ(フォルダ)にあたります。

**リーフ** これらのオブジェクトは、ブランチの端にあり、下位のオブジェクトを持ちません。たとえば、person、InetOrgPerson、またはgroupofNamesがあります。

ディレクトリ階層の最上位には、ルート要素rootがあります。これには、下位要素として、c (国)、dc (ドメインコンポーネント)、またはo (組織)が含まれます。LDAPディレクトリ内ツリーの関係については、図 29.1. 「LDAPディレクトリの構造」に示す次の例で詳細に説明します。

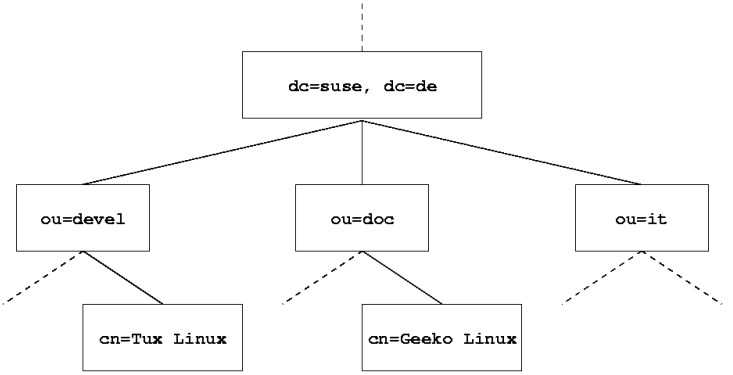


***Figure 29.1:*** *LDAPディレクトリの構造*

この図は、架空のディレクトリ情報ツリーです。3レベルのエントリが示されています。各エントリは、図内の1つの箱に対応します。最後に、このケースにおける架空のSUSE社員Geeko Linuxの*識別名*をcn=Geeko Linux,ou=doc,dc=suse,dc=deとします。この識別名は、RDN cn=Geeko Linuxを前のエントリのDN ou=doc,dc=suse,dc=deに追加して構成されます。

DITに格納するオブジェクトのタイプをグローバルに決定するには、次のスキーマが使用されます。オブジェクトタイプは、*オブジェクトクラス*によって決定されます。オブジェクトクラスは、オブジェクトに割り当てる、または割り当てられる属性を決定します。したがって、スキーマには、すべてのオブジェクトクラスと、想定したアプリケーションシナリオで使用される属性の定義を含む必要があります。RFC 2252と2256では、一般的なスキーマがいくつか用意されています。しかし、LDAPサーバの操作環境で必要になる場合は、カスタムスキーマを作成したり、複数のスキーマを相互補完的に使用することもできます。

表 29.1. 「一般的に使用されるオブジェクトクラスと属性」では、前述の例で使用されているcore.schemaとinetorgperson.schemaのオブジェクトクラスについて、必要な属性や有効な属性値などの簡単な概要を示します。

***Table 29.1:*** *一般的に使用されるオブジェクトクラスと属性*

| オブジェクトクラス | 意味 | 例で使用されているエントリ | 必須の属性 |
|---|---|---|---|
| dcObject | *domainComponent* (ドメインのコンポーネントの名前を指定します) | suse | dc |
| organizationalUnit | *organizationalUnit* (組織単位) | doc | ou |
| inetOrgPerson | *inetOrgPerson* (インターネットまたはイントラネット用の個人関連情報) | Geeko Linux | snとcn |

例 29.1. 「schema.coreの一部(行番号は説明用)」に、スキーマディレクティブの一部とその説明を示します。

***Example 29.1:*** *schema.coreの一部(行番号は説明用)*

```
   #1 attributetype (2.5.4.11 NAME ( 'ou' 'organizationalUnitName')
   #2   DESC 'RFC2256: organizational unit this object belongs to'
   #3        SUP name )

...
   #4 objectclass ( 2.5.6.5 NAME 'organizationalUnit'
   #5   DESC 'RFC2256: an organizational unit'
   #6   SUP top STRUCTURAL
   #7   MUST ou
   #8 MAY (userPassword $ searchGuide $ seeAlso $ businessCategory
   $ x121Address $ registeredAddress $ destinationIndicator
   $ preferredDeliveryMethod $ telexNumber
   $ teletexTerminalIdentifier $ telephoneNumber
   $ internationaliSDNNumber $ facsimileTelephoneNumber
   $ street $ postOfficeBox $ postalCode $ postalAddress
   $ physicalDeliveryOfficeName
   $ st $ l $ description) )
   ...
```

属性タイプorganizationalUnitNameとそれに対応するオブジェクトクラスorganizationalUnitがここで例として使用されています。1行目では、属性名、一意のOID (*オブジェクト識別子*) (数値)、および属性値の省略名が指定されています。

2行目には、DESCを使用して、属性の簡単な説明が記入されています。この定義がどのRFCに基づいているかもここに記載されます。3行目のSUPは、この属性が属する上位属性を示します。

オブジェクトクラスorganizationalUnitの定義は、4行目から始まり、属性の定義と同様、OEDとオブジェクトクラスが最初に定義されます。行目はオブジェクトクラスの簡単な説明です。SUP topで始まる6行目は、このオブジェクトクラスが他のオブジェクトクラスの上位でないことを示します。MUSTで始まる7行目は、タイプorganizationalUnitのオブジェクトで*使用する必要がある*属性値をすべてリストします。MAYで始まる8行目は、このオブジェクトクラスで使用できる属性値をすべてリストします。

スキーマの用途については、OpenLDAPのマニュアルにわかりやすく説明されています。これはインストール後に、/usr/share/doc/packages/openldap2/admin-guide/index.htmlで参照してください。

## 29.3 slapd.confを使用したサーバの設定

インストールされたシステムでは、/etc/openldap/slapd.confにLDAPサーバの完全な設定ファイルが用意されています。ここでは1つのエントリについて簡単に説明し、必要な調整について説明します。ハッシュ(#)で始まるエントリは無効です。エントリを有効にするには、このコメント文字を削除します。

### 29.3.1 slapd.conf内のグローバルエントリ

***Example 29.2:*** *slapd.conf:* スキーム用ディレクティブの取り込み

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/inetorgperson.schema
```

例 29.2. 「slapd.conf: スキーム用ディレクティブの取り込み」に示すように、slapd.confにある最初のディレクティブは、LDAPディレクト

リを編成するスキーマを指定します。エントリcore.schemaは必須です。追加の必須スキーマは、このディレクティブに追加されます(たとえばinetorgperson.schema)。ディレクトリ/etc/openldap/schemaには、利用可能なその他のスキーマが用意されています。NISを類似のLDAPサービスに置き換えるには、rfc2307.schemaとcosine.schemaという2つのスキーマを組み込みます。詳細については、OpenLDAPのマニュアルを参照してください。

***Example 29.3:*** *slapd.conf: pidfileとargsfile*

```
pidfile /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

この2つのファイルには、PID (プロセスID)とslapdプロセスの起動時に使用される引数が含まれています。これらを変更する必要はありません。

***Example 29.4:*** *slapd.conf: アクセス制御*

```
# Sample Access Control
#       Allow read access of root DSE
# Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
# access to dn="" by * read
   access to * by self write
   by users read
   by anonymous auth
#
# if no access controls are present, the default is:
#       Allow read by all
#
# rootdn can always write!
```

例 29.4. 「slapd.conf: アクセス制御」は、サーバ上のLDAPディレクトリへのアクセス許可を制御するslapd.confの一部です。slapd.confのグローバルセクションで行われている設定は、データベース固有のセクションで、カスタムのアクセス規則が宣言されていない限り有効です。これらはグローバル宣言を上書きするためです。ここで示すように、すべてのユーザはディレクトリの読み込みアクセスができますが、ディレクトリに書き込めるのは管理者(rootdn)のみです。LDAPのアクセス制御の管理は、非常に複雑なプロセスです。次のヒントが役立ちます。

- すべてのアクセス規則は、次の構造に従います。

```
access to <what> by <who> <access>
```

- ⟨*what*⟩には、アクセスを付与するオブジェクトまたは属性を指定します。個々のディレクトリブランチを、別の規則で明示的に保護することもできます。正規表現を使用して、ディレクトリのある部分を1つの規則で処理することも可能です。slapdは、設定ファイルでリストされている順序で、すべての規則を評価します。一般的な規則は、特定的な規則の後に指定する必要があります。slapdが有効だと考える最初の規則が評価され、それ以降のエントリは無視されます。

- ⟨*who*⟩には、⟨*what*⟩で指定された領域へのアクセスを付与されるユーザを指定します。ここでもslapdは、最初に一致するwhoを見つけた後、評価を行わないため、特定の規則は、一般的な規則より前に指定する必要があります。表 29.2.「ユーザグループと付与されるアクセス許可」に有効なエントリを示します。

*Table 29.2: ユーザグループと付与されるアクセス許可*

| タグ | スコープ |
|---|---|
| * | 例外なくすべてのユーザ |
| anonymous | 認証されていない(“匿名”)ユーザ |
| users | 認証済みユーザ |
| self | ターゲットオブジェクトに接続されているユーザ |
| dn.regex=<regex> | 正規表現に一致するすべてのユーザ |

- ⟨*access*⟩は、アクセスタイプを指定します。表 29.3.「アクセスのタイプ」に示すオプションを使用してください。

*Table 29.3: アクセスのタイプ*

| タグ | アクセスのスコープ |
|---|---|
| none | アクセス不可 |
| auth | サーバへの連絡用 |
| compare | 比較アクセス用のオブジェクト |

| | |
|---|---|
| search | 検索フィルタ設定用 |
| read | 読み込みアクセス |
| write | 書き込みアクセス |

slapdはクライアントが要求するアクセス権をslapd.confで付与されたアクセス権と比較します。要求された権限と比較して、同等または上位の権限が規則によって与えられている場合は、クライアントに対して、アクセスが許可されます。規則に宣言された権限を越える権限をクライアントが要求した場合、アクセスが拒否されます。

例 29.5. 「slapd.conf: アクセス制御の例」に、簡単なアクセス制御の例を示します。このように正規表現を用いて自由にアクセス制御できます。

***Example 29.5:*** *slapd.conf: アクセス制御の例*

```
access to  dn.regex="ou=([^,]+),dc=suse,dc=de"
by dn.regex="cn=administrator,ou=$1,dc=suse,dc=de" write
by user read
by * none
```

この規則は、担当の管理者のみが個別のouエントリに書き込みアクセスできることを宣言します。他のすべての認証済みユーザは読み込みアクセスができ、その他のユーザはアクセスできません。

**Tip**

**アクセス規則の設定**

access to規則または一致するbyディレクティブが存在しない場合、アクセスが拒否されます。付与されるのは、明示的に宣言されたアクセス権だけです。規則がまったく宣言されていない場合、デフォルトの原則として、管理者は書き込みアクセスができ、残りのユーザ全員は読み込みアクセスができます。

**Tip**

詳細な説明およびLDAPのアクセス権の設定例については、インストールしたopenldap2パッケージのオンラインマニュアルを参照してください。

アクセス許可を一元的なサーバ設定ファイル(slapd.conf)で管理する方法以外に、ACI (アクセス制御情報)を使用する方法があります。ACIは、個々のオブジェクトのアクセス情報をLDAPツリーに格納します。アクセス制御のタイプには共通のものがなく、開発者の間では未だ実験的だと考えられています。詳細については、http://www.openldap.org/faq/data/cache/758.htmlを参照してください。

### 29.3.2 slapd.conf内のデータベース固有のディレクティブ

***Example 29.6:** slapd.conf: データベース固有のディレクティブ*

```
database ldbm
suffix "dc=suse,dc=de"
rootdn "cn=admin,dc=suse,dc=de"
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index   objectClass     eq
```

データベースのタイプ(この例ではLDBM)は、このセクションの1行目に指定します(例 29.6. 「slapd.conf: データベース固有のディレクティブ」を参照)。suffixで始まる2行目には、このサーバが担当するLDAPツリーの部分を指定します。次のrootdnは、このサーバに対して、管理者権限を持つユーザを指定します。ここで宣言されるユーザは、LDAPエントリが必要ではなく、通常ユーザとして存在する必要もありません。管理者パスワードは、rootpwで設定します。ここでsecretを使用する代わりに、slappasswdによって作成した管理者パスワードのハッシュを入力することもできます。directoryディレクティブは、サーバ上でデータベースディレクトリが格納されている(ファイルシステム内の)ディレクトリを示します。最後のディレクティブindex objectClass eqは、すべてのオブジェクトクラスのインデックスを管理します。経験的に、ユーザが最も頻繁に検索しそうな属性をここに追加できます。データベースに対してここで定義されたカスタムのAccess規則は、グローバルAccess規則に代わって使用されます。

### 29.3.3 サーバの起動と停止

LDAPサーバが完全に設定され、項29.4. 「LDAPディレクトリのデータ処理」で説明するパターンに従ってすべてのエントリが作成されたら、rootユーザで「rcldap start」を入力し、LDAPサーバを起動します。実行されているかどうかわからない場合は、rcldap stopコマンドを実行します。実行しているLDAPサーバのステータスは、rcldap statusコマンドを実行して要求します。

YaSTランレベルエディタ(項7.6. 「」を参照)を使用して、システムのブートまたは停止時に、サーバを自動的に起動および停止することができます。またコマンドプロンプトでinsservコマンドを実行して、起動および停止スクリプトそれぞれへのリンクを作成することもできます。詳細については、項7.5.1. 「initスクリプトの追加」を参照してください。

## 29.4 LDAPディレクトリのデータ処理

OpenLDAPは、LDAPのデータを管理するためのツールを提供しています。ここでは、中でも重要な4つのツール、データストックの追加、削除、検索、および変更について説明します。

### 29.4.1 LDAPディレクトリへのデータの挿入

/etc/openldap/lsapd.confでLDAPサーバを正しく設定し、使用する準備ができたら(suffix、directory、rootdn、rootpw、およびindexについて適切なエントリが表示されることを確認)、レコード入力に進みます。OpenLDAPでは、ldapaddコマンドを使用してこのタスクを実行します。可能であれば、実践的な見地から、バンドルされたデータベースにオブジェクトを追加してください。LDAPは、LDIF形式(LDAP data interchange format)を処理してデータを入力します。LDIFは、任意の数の属性と値が指定されたシンプルテキストファイルです。指定できるオブジェクトクラスと属性については、slapd.confで宣言したスキーマファイルを参照してください。図 29.1. 「LDAPディレクトリの構造」の例のような簡単なフレームワークを作成するには、例 29.7. 「LDIFファイルの例」のLDIFを使用します。

***Example 29.7: LDIFファイルの例***

```
# The SUSE Organization
dn: dc=suse,dc=de
objectClass: dcObject
objectClass: organization
o: SUSE AG dc: suse

# The organizational unit development (devel)
dn: ou=devel,dc=suse,dc=de
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=suse,dc=de
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=suse,dc=de
objectClass: organizationalUnit
ou: it
```

**Important**

**LDIFファイルのエンコーディング**

LDAPでは、UTF-8 (Unicode)を使用します。ウムラウトは正しくエンコードする必要があります。UTF-8をサポートするエディタ(たとえばKateまたは最近のバージョンのEmacs)を使用してください。それ以外のエディタを使用する場合は、ウムラウトや他の特殊文字の使用を避けるか、recodeを使用してUTF-8をコード変換します。

**Important**

ファイルには.ldifというサフィックスを付けて保存し、次のコマンドでサーバに渡します。

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

-xは、認証(この例ではSASL)をオフにします。-Dは、操作を呼び出すユーザを宣言します。slapd.confでの設定と同様、管理者の有効なDNをここに入力します。この例では、cn=admin,dc=suse,dc=deです。-Wを指定

すると、コマンドライン(クリアテキスト)でのパスワード入力が不要になり、別のパスワードプロンプトがアクティブ化されます。このパスワードは、slapd.confのrootpwで事前に指定されています。-fはファイル名を渡します。ldapaddの実行方法の詳細については、例 29.8. 「example.ldifでのldapaddの使用」を参照してください。

***Example 29.8:*** *example.ldifで のldapaddの 使 用*

```
ldapadd -x -D cn=admin,dc=suse,dc=de -W -f example.ldif

Enter LDAP password:
adding new entry "dc=suse,dc=de"
adding new entry "ou=devel,dc=suse,dc=de"
adding new entry "ou=doc,dc=suse,dc=de"
adding new entry "ou=it,dc=suse,dc=de"
```

個人ごとのユーザデータは、別のLDIFファイルに分けて作成することができます。 例 29.9. 「TuxのLDIFデータ」では、Tuxというユーザを新しいLDAPディレクトリに追加します。

***Example 29.9:*** *TuxのLDIFデ ー タ*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@suse.de
uid:  tux
telephoneNumber: +49 1234 567-8
```

LDIFファイルには、任意の数のオブジェクトを指定できます。サーバのディレクトリブランチ全体を一度に渡すことも、個別のオブジェクトの例で示すように、その一部だけを渡すことも可能です。一部のデータを比較的頻繁に変更する必要がある場合は、1つのオブジェクトごとに細かく分割することをお勧めします。

### 29.4.2 LDAPディレクトリのデータの変更

データストックの変更用には、ツールldapmodifyが用意されています。最も簡単な方法は、対応するLDIFファイルを変更してから、変更したファイルをLDAPサーバに渡すことです。Tux社員の電話番号を+49 1234 567-8から+49 1234 567-10に変更するには、例 29.10. 「LDIFファイルtux.ldifの変更」のようにLDIFファイルを編集する必要があります。

***Example 29.10:** LDIFファイルtux.ldifの変更*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

次のコマンドを使用して、変更したファイルをLDAPディレクトリにインポートします。

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W -f tux.ldif
```

代わりの方法として、変更する属性を直接ldapmodifyに渡すこともできます。この処理手順を次に示します。

1. ldapmodifyを起動し、パスワードを入力します。

   ```
   ldapmodify -x -D cn=admin,dc=suse,dc=de -W
   Enter LDAP password:
   ```

2. 次に示す順序に従って、慎重に変更を入力します。

   ```
   dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
   changetype: modify
   replace: telephoneNumber
   telephoneNumber: +49 1234 567-10
   ```

ldapmodifyに関する詳細な情報とその構文については、対応するマニュアルページ(ldapmodify(1))を参照してください。

### 29.4.3 LDAPディレクトリでのデータの検索と読み込み

OpenLDAPには、ldapsearchを使用して、LDAPディレクトリでデータを検索して読み込むコマンドラインツールが用意されています。簡単なクエリの構文は次のとおりです。

```
ldapsearch -x -b dc=suse,dc=de "(objectClass=*)"
```

-bオプションは検索ベース、つまり、検索を実行するツリーのセクションを指定します。この例では、dc=suse,dc=deです。セクション内の特定の部分(たとえば、devel部門内のみ)で精度の高い検索を実行するには、-bを使用してこのセクションをldapsearchに渡します。-xは、簡単な認証を起動するよう要求します。(objectClass=*)は、対象のディレクトリにあるすべてのオブジェクトを読むように宣言します。このコマンドオプションは、新しいディレクトリツリーを作成した後に、すべてのエントリが正しく記録され、サーバが意図したとおりに応答することを確認するために使用されます。ldapsearchの使用の詳細については、対応するマニュアルページ(ldapsearch(1))を参照してください。

### 29.4.4 LDAPディレクトリでのデータの削除

不要なエントリを削除するには、ldapdeleteを使用します。構文は、これまでに説明した他のコマンドとほぼ同じです。たとえば、Tux Linuxに関するエントリをすべて削除するには、次のコマンドを実行します。

```
ldapdelete -x -D cn=admin,dc=suse,dc=de -W cn=Tux \
Linux,ou=devel,dc=suse,dc=de
```

## 29.5 YaST LDAPクライアント

YaSTには、LDAPベースのユーザ管理をセットアップするためのモジュールが組み込まれています。インストール時にこの機能を有効にしなかった場合は、'ネットワークサービス' → 'LDAPクライアント'の順に選択してモジュールを起動します。 LDAPに必要なPAMおよびNSS関連の変更が自動的に有効になり、必要なファイルがインストールされます。

### 29.5.1 標準的な処理手順

YaST LDAPクライアントモジュールの動作を理解するには、クライアントマシンのバックグラウンドで動作するプロセスを理解する必要があります。ネットワーク認証のためにLDAPがアクティブ化される、またはYaSTモジュールが呼び出されると、パッケージpam_ldapおよびnss_ldapがインストールされ、対応する2つの設定ファイルが調整されます。pam_ldapは、ログインプロセスと認証データのソースであるLDAPディレクトリとの間のネゴシエートを受け持つPAMモジュールです。専用モジュールpam_ldap.soがインストールされ、PAM設定が調整されます(例 29.11. 「LDAPに合わせて調整されたpam_unix2.conf」を参照)。

***Example 29.11:** LDAPに合わせて調整されたpam_unix2.conf*

```
auth:       use_ldap nullok
     account:    use_ldap
     password:   use_ldap nullok
     session:    none
```

LDAPを使用するようにサービスを手動で追加設定する場合は、/etc/pam.d内のサービスに対応するPAM設定ファイルにPAM LDAPモジュールを組み込みます。/usr/share/doc/packages/pam_ldap/pam.d/には、個々のサービスに合わせて調整済みの設定ファイルが用意されています。適切なファイルを/etc/pam.dにコピーしてください。

nsswitchメカニズムを介したglibcの名前解決は、LDAPと共にnss_-ldapを使用するように調整されています。新しく調整されたファイルnsswitch.confが、このパッケージのインストールと共に/etc/に作成されます。nsswitch.confの機能の詳細については、項22.5.1. 「環境設定ファイル」を参照してください。LDAPを使用してユーザ管理および認証を行うために、nsswitch.confに次の行が存在する必要があります。詳細については、例 29.12. 「nsswitch.confの調整」を参照してください。

***Example 29.12:** nsswitch.confの 調 整*

```
passwd: compat
     group: compat

passwd_compat: ldap
group_compat: ldap
```

この行は、glibcのリゾルバライブラリに対して、最初に/etc内で対応するファイルを評価し、次に認証およびユーザデータのソースとしてLDAPサーバにアクセスするように指示しています。このメカニズムをテストするために、たとえばgetent passwdコマンドを使用してユーザデータベースの内容を読み込みます。返されたセットには、システムのローカルユーザに関する情報と、LDAPサーバに格納されている全ユーザに関する情報が含まれているはずです。

LDAPで管理される通常のユーザがsshまたはloginを使用してサーバにログインするのを防止するには、ファイル/etc/passwdおよび/etc/groupにそれぞれ1行を追加する必要があります。 ★翻訳不要★この行は、/etc/passwdの場合は+::::::/sbin/nologin、/etc/groupの場合は+:::です。.

## 29.5.2 LDAPクライアントの設定

YaSTによりnss_ldap、pam_ldap、/etc/passwd、および/etc/groupを必要に応じて修正したら、最初のYaSTダイアログで実際の設定作業を開始できます。詳細については、図 29.2. 「YaST: LDAPクライアントの設定」を参照してください。

最初のダイアログで、LDAPをユーザ認証用に使用可能にします。［'LDAPベースDN'］に、LDAPサーバ上で全データが格納される下位サーバの検索ベースを入力します。［'LDAPサーバのアドレス'］には、LDAPサーバに到達できるアドレスを入力します。ディレクトリをリモートホストに自動的にマウントするには、［'オートマウンタを起動'］を選択します。サーバ上のデータを管理者として修正するには、［'詳細な設定'］をクリックします。詳細については、図 29.3. 「YaST: 詳細な設定」を参照してください。

次のダイアログは2つの部分に分かれています。上部領域では、YaSTユーザモジュールが反映されるユーザとグループの一般オプションを設定します。下部領域では、LDAPサーバへのアクセスに必要なデータを入力します。ユーザおよびグループ設定は、次の項目で構成されています。

**［ファイルサーバ］** 現行のシステムがファイルサーバで、個々のユーザのディレクトリが/homeに含まれている場合は、このオプションを有効にするとYaSTモジュールでユーザディレクトリが正しく扱われます。

**［LDAPユーザのログインを有効にする］**
このオプションを有効にすると、LDAP許可を介して管理されているユーザがシステムへのログインを許可されます。


***Figure 29.2:*** *YaST: LDAPクライアントの設定*

**［グループメンバー属性］** ここでは、使用するLDAPグループのタイプとして［'member'］(デフォルト設定)または［'uniquemember'］を指定します。

ここでは、LDAPサーバの設定変更に必要なアクセスデータを入力します。入力には、すべての設定オブジェクトが格納される［'ベースDNの設定'］と［'管理DN'］を使用します。

［'ユーザ管理の設定'］をクリックし、LDAPサーバ上のエントリを編集します。表示されるダイアログに、サーバとの認証に使用するLDAPパスワードを入力します。これにより、サーバに格納されているACLとACIに従って、サーバ上の設定モジュールへのアクセス権が付与されます。

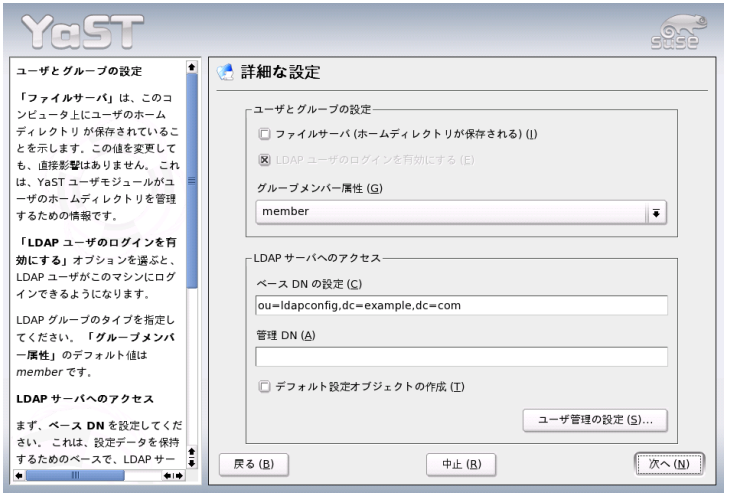


*Figure 29.3: YaST: 詳 細 な 設 定*

**Important**

**YaSTクライアントの使用**

YaST LDAPクライアントを使用して、YaSTモジュールをユーザとグループの管理用に調整し、それを必要に応じて拡張します。また、個々の属性にデフォルト値を使用してテンプレートを定義し、実際のデータ登録を簡素化できます。ここで作成した事前設定は、それ自体がLDAPディレクトリにLDAPオブジェクトとして格納されます。ユーザデータの登録には、通常のYaSTモジュール入力フォームが使用されます。登録された情報は、LDAPディレクトリにオブジェクトとして格納されます。

**Important**

モジュール設定ダイアログ(図 29.4. 「YaST: モジュールの設定」)を使用すると、既存の設定モジュールの選択と変更、新規モジュールの作成、および新規モジュールのテンプレートの設計と変更ができます。設定モジュールの値を変更したり、モジュール名を変更するには、現行モジュールのコンテンツビューの上のモジュールタイプを選択します。これにより、コンテンツビューに、このモジュールで許可されている全属性と、それぞれに割り当てられている値が

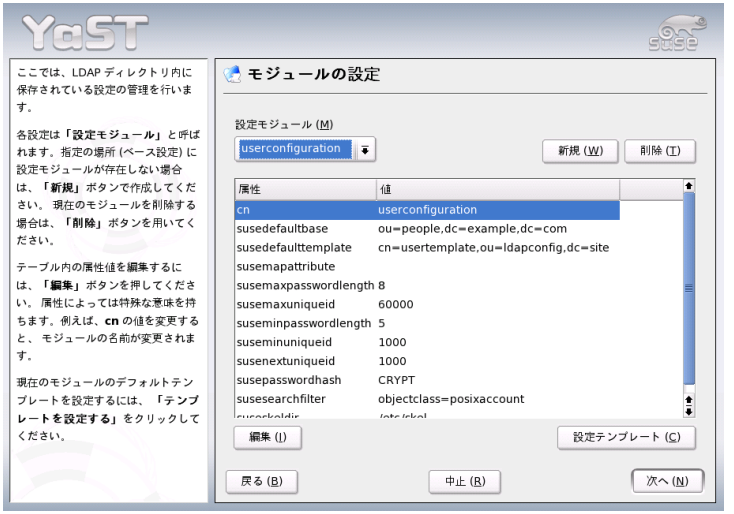


***Figure 29.4:*** *YaST: モジュールの設定*

テーブル形式のリストとして表示されます。このリストには、設定されているすべての属性に加えて、現行スキーマで許可されているが現在は使用されていない他の属性もすべて含まれています。

モジュールは、cnを変更するだけでコピーできます。個々の属性値を変更するには、コンテンツリストから選択して［'編集'］をクリックします。ダイアログが開き、その属性に属する設定をすべて変更できます。［'OK'］を選択して変更内容を受け入れます。

既存のモジュールに新規モジュールを追加する必要がある場合は、コンテンツ概要の上にある［'新規'］をクリックします。表示されるダイアログ(suseuserconfigurationまたはsusegroupconfiguration)に新規モジュールの名前とオブジェクトクラスを入力します。［'OK'］を選択してダイアログを閉じると、新規モジュールが既存モジュールの選択リストに追加され、選択または選択解除できるようになります。現在選択しているモジュールを削除するには［'削除'］をクリックします。

グループおよびユーザ管理用のYaSTモジュールは、重要な標準値が以前にYaST LDAPクライアントで定義されていれば、その値をテンプレートに埋め込みます。必要に応じてテンプレートを編集するには、［'設定テンプレート'］をクリックします。ドロップダウンメニューには、変更可能な既存のテ

ンプレートまたは空のエントリが含まれています。そのいずれか1つを選択し、［'オブジェクトテンプレートの設定'］フォームで、このテンプレートのプロパティを設定します(図 29.5. 「YaST: オブジェクトテンプレートの設定」を参照)。このフォームは、テーブル形式の2つの概要ウィンドウに分かれています。上のウィンドウには、すべての一般テンプレート属性が表示されます。必要に応じて値を指定するか、一部を空のままにしておきます。空の属性は、LDAPサーバ上で削除されます。

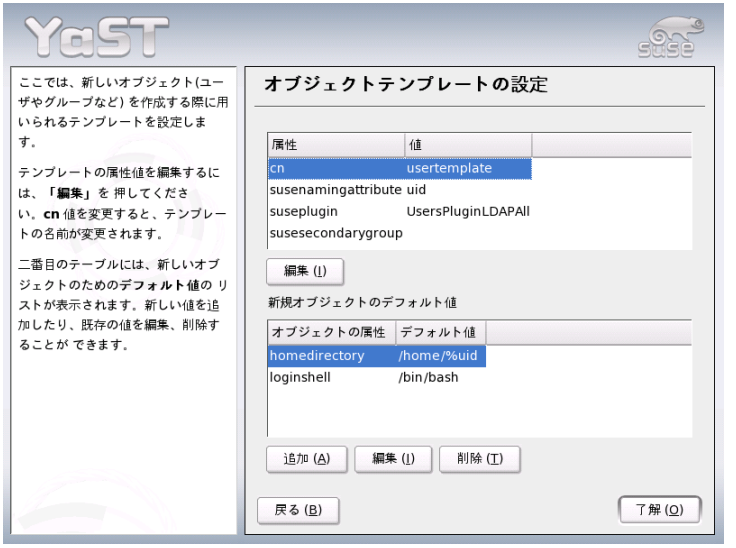


*Figure 29.5: YaST: オブジェクトテンプレートの設定*

第2のビュー(［'新規オブジェクトのデフォルト値'］)には、標準値が定義済みの対応するLDAPオブジェクト(この場合はグループ設定またはユーザ設定)の属性がすべて表示されます。さらに属性とその標準値を追加したり、既存の属性と値のペアを編集したり、属性全体を削除できます。cnエントリを変更してテンプレートをコピーしてください。前述のように、モジュールの`susedefaulttemplate`属性値を調整済みテンプレートのDNに設定し、テンプレートを対応するモジュールに接続します。

**Tip**

絶対値の代わりに変数スタイルを使用すると、属性のデフォルト値を他の属性から作成できます。たとえば、新規ユーザの作成時には、snとgivenNameの属性値からcn=%sn %givenNameが自動的に作成されます。

**Tip**

すべてのモジュールとテンプレートを適切に設定し、実行する準備が完了したら、新しいグループとユーザを通常の方法でYaSTに登録できます。

### 29.5.3 ユーザとグループ—YaSTによる設定

ユーザおよびグループデータの実際の登録手順は、LDAPを使用しない場合とほぼ同様です。次に、ユーザ管理に関連する手順の概略を示します。グループの管理手順も同様です。

'セキュリティとユーザ' → 'User Administration(ユーザ管理)'の順に選択し、YaSTユーザ管理にアクセスします。入力フォームが表示され、名前、ログインおよびパスワードなど、最も重要なユーザデータを登録できます。［'詳細'］を選択すると、グループメンバーシップ、ログインシェルおよびホームディレクトリの設定フォームにアクセスできます。デフォルト値は、項29.5.2.「LDAPクライアントの設定」で説明した手順で定義されています。LDAPの使用時には、このフォームから別のフォームにアクセスし、LDAP固有の属性を登録できます。このダイアログを図 29.6.「YaST: LDAPの追加設定」に示します。値を変更する属性をすべて選択し、［'編集'］をクリックします。開いたフォームで［'続行'］を選択して閉じると、ユーザ管理用の初期入力フォームに戻ります。

ユーザ管理の初期入力フォームには、［'LDAPオプション'］が用意されています。ここでは、使用可能なユーザのセットにLDAP検索フィルタを適用するか、［'LDAP User and Group Configuration(LDAPユーザとグループの設定)'］を選択してLDAPユーザおよびグループの設定モジュールにアクセスできます。

## 29.6 関連資料

SASL設定や、複数のスレーブ間で作業不可を分散するためのLDAPサーバのレプリケートの設定などの複雑なトピックについては、ここではあえて触れませ


***Figure 29.6:*** *YaST: LDAPの 追 加 設 定*

んでした。この2つの項目の詳細については、『*OpenLDAP 2.2 Administrator's Guide*』(下記参照)を参照してください。

OpenLDAPプロジェクトのWebサイトには、LDAPの初心者向けや熟練者向けのあらゆるマニュアルが用意されています。

**『OpenLDAP Faq-O-Matic』** OpenLDAPのインストール、設定、および運用に関する豊富なQAが集められています。 `http://www.openldap.org/faq/data/cache/1.html`.

**『Quick Start Guide』** LDAPサーバのインストール方法を手順を追って簡単に説明しています。

`http://www.openldap.org/doc/admin22/quickstart.html`またはインストール済みのシステムで、`/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`を参照してください。

**『OpenLDAP 2.2 Administrator's Guide』**
アクセス制御や暗号化など、LDAP設定の重要な側面を詳細に説明しています。`http://www.openldap.org/doc/admin22/`またはインス

トール済みのシステムで、/usr/share/doc/packages/openldap2/admin-guide/quickstart.htmlを参照してください。

IBMからLDAPに関する以下のレッドブックが出版されています。

**『Understanding LDAP』** LDAPの基本原則一般について、詳細に説明しています。 http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf.

**『LDAP Implementation Cookbook』**
この本の対象読者は、*IBM SecureWay Directory*の管理者です。ただし、LDAPに関する重要な一般情報についての記述もあります。 http://www.redbooks.ibm.com/redbooks/pdfs/sg245110.pdf.

LDAPに関する書籍

- 『*Understanding and Deploying LDAP Directory Services*』第2版、Howes、Smith、Good共著、2003年Addison-Wesley発行(ISBN 0-672-32316-8)
- 『*LDAP System Administration*』、Gerald Carter著、2003年、O'Reilly & Associates発行(ISBN 1-56592-491-6)。

LDAPに関する究極的な参考資料は、RFC 2251～2256です。

# Apache Webサーバ

Apacheは60%以上のシェアを誇る、世界で最も広く使われているWebサーバです(情報源: `http://www.netcraft.com`)。Webアプリケーションをサポートするために、Apacheは多くの場合、Linux、データベースMySQL、およびプログラミング言語のPHPやPerlと組み合わせられています。この組み合わせは、一般に*LAMP*と呼ばれます。

この章では、WebサーバApacheについて説明します。インストール方法と設定方法に加え、いくつかのモジュールについても説明します。また、仮想ホストの応用例についても紹介します。

# 30.1 基本事項

ここでは、WebサーバとWebサーバが使用するプロトコルの基本的な内容を説明します。また、ほとんどの重要な機能についても紹介します。

## 30.1.1 Webサーバ

Webサーバは、クライアントが要求するHTMLページを発行します。これらのページはディレクトリに格納されているページ(パッシブページまたはスタティックページ)か、クエリに対する応答として生成されます(アクティブコンテンツ)。

## 30.1.2 HTTP

クライアントは通常、KonquerorやMozillaなどのWebブラウザです。ブラウザとWebサーバ間の通信は、ハイパーテキスト転送プロトコル(HTTP)によって行われます。現行バージョンのHTTP 1.1は、RFC 2068および更新版のRFC 2616に文書化されています。これらのRFCについては、`http://www.w3.org`を参照してください。

## 30.1.3 URL

クライアントは、`http://www.novell.com/linux/suse/`などのURLを使用して、サーバにページを要求します。URLは、次の各部で構成されています。

**プロトコル** 次のプロトコルがよく使用されます。

**http://** HTTPプロトコル

**https://** HTTPに暗号化機能を付加したセキュアプロトコル

**ftp://** ファイルのアップロードとダウンロードに使用するファイル転送プロトコル

**ドメイン** この例の`www.suse.com`に対応する部分。ドメインは、2つの部分に分かれます。最初の部分(`www`)は、コンピュータを示します。2番目の部分(`suse.com`)が、実際のドメインです。これらを合わせてFQDN (fully qualified domain name: 完全修飾ドメイン名)と呼びます。

**リソース** この例のindex_us.htmlに対応する部分。この部分は、リソースへのフルパスを示します。リソースには、この例のようにファイルを指定できます。CGIスクリプト、JavaServer Page、その他のリソースも指定できます。

このためのインターネットメカニズム(たとえば、ドメインネームシステム、DNS)は、クエリをドメインwww.suse.comに転送し、リソースをホストする1つ以上のコンピュータに届けます。ここでApacheは、実際のリソース(この例ではページindex_us.html)をそのファイルディレクトリから配信します。この例ではファイルがディレクトリの最上位レベルにありますが、http://support.novell.com/linux/のようにリソースがサブディレクトリにあることもあります。

ファイルパスは、DocumentRootからの相対パスで指定します。DocumentRootは、設定ファイルで変更できます。この仕組みについては、項30.6.2. 「DocumentRoot」を参照してください。

### 30.1.4 デフォルトページの自動表示

デフォルトページが指定されていない場合、Apacheはよく使われる名前の1つをURLに自動的に追加します。このような場合に最もよく使用されるのは、index.htmlという名前です。この機能は、サーバが使用する実際のページ名とともに、項30.6.2. 「DirectoryIndex」の説明に従って設定可能です。この例で、http://www.suse.comというURLを使用すると、サーバに対してページhttp://www.novell.com/linux/suse/の配信が求められます。

## 30.2 YaSTによるHTTPサーバのセットアップ

Apacheは、YaSTを使用して簡単にセットアップできますが、この方法でWebサーバをセットアップするには、それについてある程度の知識が必要です。YaSTコントロールセンターで、'ネットワークサービス' → 'HTTPサーバ'の順に選択すると、まだインストールされていないパッケージをインストールするかどうかを尋ねるプロンプトが表示されることがあります。すべてがインストールされると、YaSTにより設定ダイアログ ('HTTP サーバの設定') が表示されます。

このダイアログで、まず'HTTPサービス'自体を有効にします。これにより、ファイアウォールで対応するポート(ポート80)が開かれます('選択したポート上でファイアウォールを開く')。ウィンドウ下部にある'設定'/'概要'で、ロー

カルHTTPサーバの情報'受信'(デフォルトはポート80)、'モジュール'、'デフォルトホスト''ホスト'を設定することができます。現在の設定値を変更するには、'編集'をクリックします。

まず、['デフォルトホスト']を確認し、必要に応じて設定を調整します。次に、['モジュール']で必要なモジュールを有効にします。その他、仮想ホストの作成など、より詳細な設定を行うためのダイアログがいくつか用意されています。

## 30.3 Apacheのモジュール

Apacheは、モジュールを使用することによって、広い範囲の機能を取り込んで拡張することができます。たとえば、Apacheではモジュールにアクセスするさまざまなプログラミング言語のCGIスクリプトを実行できます。PerlやPHPはもとより、PythonやRubyのような追加のスクリプト言語も利用できます。安全なデータ伝送用のモジュール(secure sockets layer: SSL)、ユーザ認証、拡張ログ出力など、さまざまなモジュールがあります。

必要なノウハウをご存知であれば、カスタムモジュールを使用してあらゆる種類の要件や好みに応じてApacheをカスタマイズできます。詳細については、項30.12.4. 「その他の情報源」を参照してください。

クエリを処理するために、複数の「ハンドラ」を指定できます(設定ファイルのディレクティブを使用します)。これらのハンドラは、Apacheの一部でも、クエリ処理のために呼び出されたモジュールでもかまわないため、この手順は非常に柔軟に調整することができます。また、Apacheとともにカスタムモジュールを使用して、要求が処理される方法を決定することもできます。

Apacheは高度にモジュール化されており、いくつかの小さいタスクを除いてすべてをモジュールによって処理できます。この方法で、HTTPさえもモジュールによって処理できます。Apacheは、必ずしもWebサーバである必要はありません。他のモジュールとともにまったく異なる目的に使用することもできます。たとえば、Apacheで動作するPOP3 (proof-of-concept)メールサーバもあります。

Apacheのモジュールは、次のようなさまざまな追加機能を提供します。

**仮想ホスト** 仮想ホストをサポートするとは、Apacheの1つのインスタンスと1台のコンピュータを複数のWebサイトに使用できるという意味です。ユーザにとっては、そのWebサーバが、複数の独立したWebサーバのように見えます。仮想ホストは、異なるIPアドレスまたは異なる名前に基づいて設定できます。これにより、コンピュータの追加に要する取得コストや管理負荷が不要になります。

**柔軟性の高いURL変更** Apacheには、URLの操作や変更を行う方法がいくつか用意されています。詳細については、Apacheのマニュアルを参照してください。

**コンテントネゴシエーション** Apacheでは、クライアント(ブラウザ)の機能に合わせてページを配信できます。たとえば、古いブラウザやLynxのようにテキストモードのみで動作するブラウザには、フレームのないシンプルなバージョンを配信します。JavaScriptにはさまざまなブラウザとの間で非互換性の問題がありますが、この問題も、それぞれのブラウザに適切なバージョンのページを配信することで回避できます。

**柔軟性の高いエラー処理** ページが存在しないなどのエラーが発生した場合も、柔軟な方法で適切な応答を提供できます。たとえば、CGIを使用して応答をアクティブに生成できます。

## 30.4 スレッド

スレッドとは、「軽い」形態のプロセスです。プロセスよりもスレッドが優れている点は、リソースの消費が少ないことです。このため、プロセスの代わりにスレッドを使用すれば、パフォーマンスが向上します。逆に短所は、スレッド環境で実行されるアプリケーションがスレッドセーフでなければならないことです。これは次のことを意味します。

- 関数(またはオブジェクト指向アプリケーションではメソッド)がリエントラントであること、つまり、他のスレッドが同時に同じ関数を実行していても、関数への入力が同じであれば同じ結果が得られることが保証されなければなりません。また、複数のスレッドで同時に実行できるような方法で、関数をプログラムする必要があります。
- 同時スレッドが競合しないような方法でリソースが用意されている必要があります。

Apache 2は、クエリを別のプロセスとして、またはプロセスとスレッドを組み合わせた混合モードで処理します。プロセスとしての実行は、MPM *prefork*が行います。スレッドとしての実行は、MPM *worker*が行います。使用するMPMは、インストール時に選択します(項30.5.「インストール」を参照)。3番目のモードである*perchild*はまだ完成されていないので、SUSE LINUXでは利用できません。

## 30.5 インストール

### 30.5.1 YaSTでのパッケージの選択

基本のインストールでは、Apacheパッケージである`apache2`を選択すれば十分です。これに加え、`apache2-prefork`、`apache2-worker`などのMPM(マルチプロセッシング)パッケージを1つインストールします。MPMを選択する場合、`mod_php4`のライブラリはまだスレッドセーフでないので、スレッドベースのworker MPMを`mod_php4`と一緒に使用できないことに注意してください。

### 30.5.2 Apacheの有効化

インストール後、ランレベルエディタでApacheをサービスとして有効にする必要があります。システムのブート時にApacheを起動するようにするには、ランレベルエディタで、ランレベル3と5をオンにします。Apacheが実行されているかどうかをテストするには、ブラウザで`http://localhost/`を開きます。Apacheが有効な場合、`apache2-example-pages`がインストールされていれば、サンプルページが表示されます。

### 30.5.3 有効なコンテンツのモジュール

モジュールを使用して有効なコンテンツを使用するには、それぞれのプログラミング言語のモジュールをインストールします。Perlの場合は`apache2-mod_perl`モジュール、PHPの場合は`mod_php4`モジュール、Pythonの場合は`mod_python`モジュールです。これらのモジュールの用途については、項30.8.4. 「モジュールを使用したアクティブコンテンツの生成」を参照してください。

### 30.5.4 その他の推奨パッケージ

マニュアルのパッケージ`apache2-doc`をインストールすることをお勧めします。このパッケージをインストールした後にサーバを項30.5.2. 「Apacheの有効化」の手順に従って起動すれば、マニュアルをURL`http://localhost/manual`で直接参照できます。

Apacheモジュールの開発やサードパーティ製モジュールのコンパイルを行う場合は、パッケージ`apache2-devel`を対応する開発ツールとともにインストールします。これらの開発ツールには、`apxs`ツール(項30.5.5. 「apxsによるモジュールのインストール」を参照)などがあります。

### 30.5.5 apxsによるモジュールのインストール

apxs2は、モジュール開発者にとって重要なツールです。このプログラムを使用すると、ソースコードからのモジュールのコンパイルとインストールが(設定ファイルでの必要な変更も含め)1つのコマンドで実行できます。さらに、オブジェクトファイルとして提供されているモジュール(拡張子.o)やスタティックライブラリとして提供されているモジュール(拡張子.a)もインストールできます。ソースからインストールすると、apxs2によって、Apacheがモジュールとして直接使用するダイナミック共有オブジェクト(DSO)が作成されます。

ソースコードからモジュールをインストールするには、たとえばapxs2 -c -i -a mod_foo.cというコマンドを実行します。apxs2のその他のオプションについては、マニュアルページを参照してください。その後、項30.6.1.「SuSEconfigでの設定」の手順に従って、/etc/sysconfig/apache2内のエントリAPACHE_MODULESを使用してモジュールを有効にします。

apxs2には、apxs2、apxs2-prefork、およびapxs2-workerという3つのバージョンがあります。apxs2は、どのMPMに対しても使用できるようにモジュールをインストールします。他の2つのプログラムは、preforkかworkerのどちらかのMPMのみが使用できるように、モジュールをインストールします。apxs2はモジュールを/usr/lib/apache2にインストールし、apxs2-preforkは/usr/lib/apache2-preforkにインストールします。

## 30.6 設定

Apacheをインストールした後は、特に必要な場合を除き、設定を変更する必要はありません。Apacheを設定するには、YaSTとSuSEconfigを使用するか、ファイル/etc/apache2/httpd.confを直接編集します。

### 30.6.1 SuSEconfigでの設定

/etc/sysconfig/apache2で行った設定は、SuSEconfigによってApache設定ファイルに適用されます。あらかじめ用意されている設定オプションだけで、ほとんどのシナリオに対応できます。ファイル内の各変数には、その効果を説明するコメントが付けられています。

#### カスタム設定ファイル

設定ファイル/etc/apache2/httpd.confを直接変更する代わりに、変数APACHE_CONF_INCLUDE_FILESを使用してhttpd.conf.localなどの独自

の設定ファイルを指定できます。これにより、ファイルがメイン設定ファイルによって解釈されます。この方法を使用すると、次にインストールを行ったときにファイル/etc/apache2/httpd.confが上書きされても、設定に加えた変更内容が維持されます。

#### モジュール

YaSTによってインストールしたモジュールは、変数APACHE_MODULESで指定したリストにモジュール名を追加することによって有効化できます。この変数は、ファイル/etc/sysconfig/apache2で定義します。

#### フラグ

APACHE_SERVER_FLAGSを使用して、設定ファイルの特定のセクションを有効化または無効化するフラグを指定できます。設定ファイルのセクションが、次のように囲まれている場合、

```
<IfDefine someflag>
.
.
.
</IfDefine>
```

それぞれのフラグが変数ACTIVE_SERVER_FLAGSで、ACTIVE_SERVER_-FLAGS = ... フラグ ...のように設定されている場合のみ、そのセクションが有効化されます。これにより、テストの際に設定ファイルの大きなセクションを簡単に有効化または無効化できます。

### 30.6.2 手動設定

/etc/sysconfig/apache2に定義されている設定以外の機能を有効にするには、ファイル/etc/apache2/httpd.confを編集します。以降のセクションで、このファイル内の設定可能なパラメータの一部を説明します。パラメータは、ファイルでの記載順に示してあります。

#### DocumentRoot

基本的な設定の1つであるDocumentRootには、サーバによって配信されるWebページの格納ディレクトリとしてApacheが想定するディレクトリを指定します。デフォルトの仮想ホストでは、/srv/www/htdocsに設定されています。通常、この設定の変更は不要です。

### Timeout

サーバが要求に対してタイムアウトをレポートするまでの待ち時間を指定します。

### MaxClients

Apacheが同時に処理できるクライアントの最大数を指定します。デフォルト値は`150`ですが、アクセス数の多いWebサイトの場合、この値では小さすぎる可能性があります。

### LoadModule

`LoadModule`ディレクティブには、ロードするモジュールを指定します。ローディングシーケンスはモジュール自体によって決定されます。これらのディレクティブには、モジュールを含むファイルも指定します。

### Port

Apacheがクエリをリスンするポートを指定します。一般に、HTTPのデフォルトポート80を指定します。通常、この設定は変更しません。新しいWebサイトをテストする場合などに、Apacheがリスンするポートを変更することがあります。この場合、実稼動バージョンのWebサイトは、引き続きポート80でアクセスできるようにします。

別の理由として考えられるのは、一般向けでない情報がページに含まれているために、イントラネット上でページを提供する場合です。この場合、ポートの値をたとえば`8080`に設定し、ファイアウォールを使用してこのポートへの外部からのアクセスをブロックします。これにより、サーバを外部アクセスから保護することができます。

### Directory

このディレクティブは、ディレクトリに対するアクセス権やその他のパーミッションを設定する場合に使用します。`DocumentRoot`に対しても同様のディレクティブがあります。`DocumentRoot`を変更した場合は、ここに指定したディレクトリ名も変更する必要があります。

### DirectoryIndex

ここでは、ファイル指定が欠けているURLを補うためにApacheが検索するファイルを指定します。デフォルト値は、`index.html`です。たとえば、クライアントがURL`http://www.example.com/foo/bar`を要求した場合、`DocumentRoot`の下のディレクトリ`foo/bar`にファイル`index.html`があれば、Apacheはこのページをクライアントに返します。

### AllowOverride

Apacheのすべてのドキュメント配布元ディレクトリに、グローバルアクセス権やこのディレクトリに対する他の設定を上書きできるファイルが入っていることがあります。これらの設定は、サブディレクトリにある他の同様のファイルによって上書きされるまで、現在のディレクトリとそのサブディレクトリに繰り返し適用されます。したがって、この種のファイルに指定された設定は、それがDocumentRootにある場合、グローバルに適用されます。このようなファイルは、通常.htaccessという名前ですが、項30.6.2.「AccessFileName」の説明に従って名前を変更することも可能です。

AllowOverrideは、ローカルファイルに指定された設定が、グローバル設定を上書きできるかを判定するために使用します。有効な値は、NoneとAllのほか、Options、FileInfo、AuthConfig、およびLimitの任意の組み合わせです。これらの値の意味は、Apacheのマニュアルに詳しく説明されています。安全なデフォルト値は、Noneです。

### Order

このオプションでは、AllowとDenyのアクセス権が適用される順序を指定します。デフォルトの設定は、次のとおりです。

```
Order allow,deny
```

つまり、許可するアクセスがまず適用され、次に、拒否するアクセスが適用されます。基本的なアプローチは、次の2つのいずれかです。

**allow all** すべてのアクセスを許可して例外を定義する

**deny all** すべてのアクセスを拒否して例外を定義する

deny allの例:

```
Order deny,allow
Deny from all
Allow from example.com
Allow from 10.1.0.0/255.255.0.0
```

### AccessFileName

ここでは、Apacheによって配信されるディレクトリのグローバルアクセス権やその他の設定を上書きできるファイルの名前を設定します(項30.6.2.「AllowOverride」を参照)。デフォルト値は、.htaccessです。

### ErrorLog

Apacheがエラーメッセージのログを出力するファイル名を指定します。デフォルト値は、/var/log/httpd/errorlogです。特別なログファイルを設定ファイルのVirtualHostセクションで指定しない限り、仮想ホストについてのエラーメッセージ(項30.9. 「仮想ホスト」を参照)もこのファイルに出力されます。

### LogLevel

エラーメッセージは、その重要度レベルに従って分類されています。この設定は、ログを出力するエラーメッセージの重要度の下限を指定します。これをあるレベルに設定すると、そのレベル以上の重要度を持つエラーメッセージがログに出力されます。デフォルト値は、warnです。

### Alias

エイリアスによってディレクトリへのショートカットを指定すると、このディレクトリに直接アクセスできるようになります。たとえば、エイリアス/manual/を使用すると、DocumentRootが/srv/www/htdocs以外のディレクトリに設定されていても、ディレクトリ/srv/www/htdocs/manualへのアクセスが可能になります(DocumentRootをそのディレクトリに設定した場合は、エイリアスを設定する意味がなくなります)。このエイリアスでは、http://localhost/manualを使用して、対応するディレクトリに直接アクセスします。Aliasディレクティブで指定した新しいターゲットディレクトリにパーミッションを定義するには、そのディレクトリにDirectoryディレクティブを指定します。項30.6.2. 「Directory」を参照してください。

### ScriptAlias

このディレクティブは、Aliasに似ています。加えて、ターゲットディレクトリのファイルがCGIスクリプトとして扱われる必要があることを示します。

### SSI (Server-Side Includes)

SSI (Server-side includes)は、SSI用の実行可能ファイルをすべて検索することによって有効化できます。これは次の命令によって行います。

```
<IfModule mod_include.c>
XBitHack on </IfModule>
```

SSIのファイルを検索するには、コマンドchmod +x ⟨*filename*⟩を使用してファイルを実行可能にします。または、SSI用に検索するファイルタイプを明示的に指定します。これは次の命令によって行います。

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

.htmlを指定すると、ApacheがSSI用のページを(何も含まれていないページも含めて)すべて検索することによりパフォーマンスが大幅に低下するので、この方法はお勧めできません。SUSE LINUXでは、これら2つのディレクティブがすでに設定ファイルに含まれているので、通常、変更は必要ありません。

#### UserDir

モジュールmod_userdirとディレクティブUserDirを使用して、Apacheでユーザのホームディレクトリ内のどのディレクトリからファイルを公開できるようにするかを指定します。これは、変数HTTPD_SEC_PUBLIC_-HTMLを設定することによって、SuSEconfigで設定できます。ファイルの発行を有効にするには、この変数をyesに設定します。これにより、ファイル/etc/apache2/mod_userdir.confに、次のエントリが追加されます。このエントリは、/etc/apache2/httpd.confによって解釈されます。

```
<IfModule mod_userdir.c>
UserDir public_html
</IfModule>
```

## 30.7 Apacheの使用

ApacheでスタティックWebページを表示するには、単にファイルを適切なディレクトリに配置します。SUSE LINUXでは、このディレクトリは/srv/www/htdocsになります。小さなサンプルページが、いくつかすでにそのディレクトリに格納されています。これらのページを使用して、Apacheが正しくインストールされ、現在有効であることを確認します。これ以降は、これらのページの上書きやアンインストールを行えます。カスタムCGIスクリプトは、/srv/www/cgi-binにインストールされます。

操作中にApacheは、ファイル/var/log/httpd/access_logまたは/var/log/apache2/access_logにログメッセージを書き込みます。これらのメッセージは、どのリソースが、いつ、どのメソッド(GET、POSTなど)で配信されたかを示します。エラーメッセージのログは、/var/log/apache2ファイルに出力されます。

## 30.8 アクティブコンテンツ

Apacheでは、複数の方法でアクティブコンテンツを配信できます。アクティブコンテンツは、クライアントからの変数入力データを元に生成されたHTMLページです。たとえば、1つ以上の検索文字列(ANDやORの論理演算子で接続されていることもある)を入力すると、それに応答して、これらの検索文字列が含まれたページのリストを返す検索エンジンなどです。

Apacheには、アクティブコンテンツの生成方法として次の3つがあります。

**Server Side Includes (SSI)** SSIは、特別なコメントによってHTMLページに埋め込まれたディレクティブです。Apacheは、コメントの内容を解釈し、結果をHTMLページの一部として配信します。

**CGI (Common Gateway Interface)** CGIは、特定のディレクトリに格納されているプログラムです。Apacheは、クライアントから伝送されたパラメータをこれらのプログラムに転送し、プログラムの出力を返します。特に既存のコマンドラインプログラムが、Apacheから入力を受け取り、Apacheに出力を返すという方法で設計できるので、この種のプログラミングは非常に簡単です。

**モジュール** Apacheは、要求処理に必要なあらゆるモジュールを実行するためのインタフェースを提供します。Apacheでは、これらのプログラムから要求やHTTPヘッダのような重要な情報にアクセスできます。プログラムはアクティブコンテンツの生成だけでなく、認証などの他の機能も担うことができます。これらのモジュールをプログラミングするには、専門知識が必要です。このアプローチの利点は、SSIやCGIを上回るパフォーマンスと可能性が得られることです。

CGIスクリプトは(その所有者のユーザIDの下で)Apacheによって直接実行されますが、モジュールはApacheに組み込まれている固定インタプリタによって制御されます。この方法では、要求のたびに個別のプロセスを開始、終了する必要はありません(プロセスの開始、終了には、プロセス管理、メモリ管理などのための多大なオーバーヘッドを要します)。スクリプトは、WebサーバのIDの下で実行されるインタプリタによって処理されます。

ただし、このアプローチにも注意すべき点はあります。CGIスクリプトは、モジュールに比べてプログラムミスに比較的寛容です。CGIスクリプトでは、プログラムは要求が処理されれば終了するので、誤ってリソースやメモリの解放をしなくても、後々まで影響が残るということがありません。最終的には、プログラムエラーのためにプログラムによって解放されなかったメモリがクリア

されるだけです。モジュールの場合は、インタプリタが固定なので、プログラミングエラーの影響が累積されます。サーバが再起動されず、インタプリタが数ヶ月間実行された場合、データベース接続のようなリソースが解放されていなければ、大きな影響が出ることがあります。

### 30.8.1 SSI (Server-Side Includes)

SSI (Server-side includes)は、特別なコメントに埋め込まれたディレクティブで、Apacheによって実行されます。結果は出力に埋め込まれます。たとえば現在の日付は、`<!--#echo var="DATE_LOCAL" -->`として出力されます。最初のコメントマーク`<!--`の最後にある#は、これが単純なコメントではなく、SSIディレクティブであることをApacheに示します。

SSIを有効化するには、いくつかの方法があります。最も簡単なアプローチは、SSI用の実行可能ファイルをすべて検索する方法です。別のアプローチとして、特定のファイルタイプを指定してSSI用のファイルを検索する方法もあります。これらの設定の詳細については、項30.6.2. 「SSI (Server-Side Includes)」を参照してください。

### 30.8.2 CGI (Common Gateway Interface)

CGIは、*Common Gteway Iterface*の略語です。CGIを使用すると、サーバはスタティックなHTMLページを配信するだけでなく、ページを生成するプログラムを実行します。これにより、計算の結果(データベース内の検索の結果)を表すページを生成できます。プログラムは、実行するプログラムに渡す引数を利用して、すべての要求に対して個別の応答ページを返すことができます。

CGIの最大の利点は、このテクノロジが極めてシンプルだということです。プログラムは単に特定のディレクトリに存在し、コマンドラインプログラムとまったく同様にWebサーバによって実行されます。サーバは、プログラム出力をクライアントへの標準出力チャネル(`stdout`)に送出します。

CGIプログラムは、理論的にはどのプログラム言語でも作成できます。しかし、通常、PerlやPHPなどのスクリプト言語(インタプリタ言語)が使用されます。速度が重要な場合は、CまたはC++のほうが適しています。

単純なケースでは、Apacheがこれらのプログラムを特定のディレクトリ(`cgi-bin`)で探します。このディレクトリは、項30.6. 「設定」の説明にあるように、設定ファイル内で設定できます。必要であれば、追加のディレクトリを指定できます。この場合、Apacheはこれらのディレクトリで実行可能ファイルを検索します。ただし、これにより、すべてのユーザがApacheを使用し

て(悪意のあるプログラムを含め)プログラムを実行できることになるので、セキュリティリスクが生じます。そこで実行可能プログラムをcgi-binだけに限定すれば、管理者は誰がどのスクリプトやプログラムをこのディレクトリに配置したかが簡単にわかるので、悪意があるものを見分けられます。

### 30.8.3 GETとPOST

入力パラメータは、GETまたはPOSTによってサーバに渡されます。どちらのメソッドを使用するかによって、サーバがスクリプトにパラメータを渡す方法も異なります。POSTの場合、サーバは標準入力(stdin)からパラメータをプログラムに渡します。プログラムは、コンソールから起動した場合と同じようにその入力を受け取ります。GETの場合、サーバは環境変数QUERY_STRINGを使用してパラメータをプログラムに渡します。

### 30.8.4 モジュールを使用したアクティブコンテンツの生成

Apacheではさまざまなモジュールを使用できます。「モジュール」という用語は2つの意味で使用されます。1つ目は、特定の処理を行うためにApacheに組み込まれるモジュールで、プログラミング言語の埋め込みモジュールなどがこれにあたります。

2つ目は、プログラミング言語と関連して、モジュールが独立の機能、クラス、変数を参照する場合です。これらのモジュールはプログラムに組み込まれ、特定の機能を提供します。たとえば、すべてのスクリプト言語で利用できるCGIモジュールがこれにあたります。これらのモジュールは、要求パラメータを読み込むメソッドやHTML出力のメソッドのようなさまざまな機能を提供して、CGIアプリケーションのプログラミングを支援します。

### 30.8.5 mod_perl

Perlは、広く普及している定評のあるスクリプト言語です。Perl用のモジュールやライブラリは、Apache設定ファイルの拡張用ライブラリを含め、数多く存在します。幅広いPerlライブラリについては、http://www.cpan.org/のComprehensive Perl Archive Network (CPAN)を参照してください。

#### mod_perlのセットアップ

SUSE LINUXでは、対応するパッケージをインストールするだけでmod_perlをセットアップできます(項30.5.「インストール」を参照)。インス

トールが終了すると、Apache設定ファイルに必要なエントリが追加されます(/etc/apache2/mod_perl-startup.plを参照)。mod_perlの情報は、http://perl.apache.org/で入手できます。

### mod_perlとCGI

最も単純なケースでは、以前のCGIスクリプトを別のURLで要求して、mod_perlスクリプトとして実行します。設定ファイルには、同じディレクトリを指定し、そのディレクトリにあるスクリプトをCGIまたはmod_perlによって実行するエイリアスがあります。これらのエントリはすべて、すでに設定ファイルに存在します。CGIのエイリアスエントリは次のとおりです。

```
ScriptAlias /cgi-bin/ "/srv/www/cgi/bin/"
```

mod_perlのエントリは次のとおりです。

```
<IfModule mod_perl.c>
# Provide two aliases to the same cgi-bin directory,
# to see the effects of the 2 different mod_perl modes.
# for Apache::Registry Mode
ScriptAlias /perl/          "/srv/www/cgi-bin/"
# for Apache::Perlrun Mode
ScriptAlias /cgi-perl/      "/srv/www/cgi-bin/"
</IfModule>
```

mod_perlでは、次のエントリも必要です。これらのエントリは、すでに設定ファイルに存在します。

```
#
# If mod_perl is activated, load configuration information
#
<IfModule mod_perl.c>
Perlrequire /usr/include/apache/modules/perl/startup.perl
PerlModule Apache::Registry

#
# set Apache::Registry Mode for /perl Alias
#
<Location /perl>
SetHandler  perl-script
PerlHandler Apache::Registry
Options ExecCGI
PerlSendHeader On
</Location>
```

```
#
# set Apache::PerlRun Mode for /cgi/perl Alias
#
<Location /cgi-perl>
SetHandler  perl-script
PerlHandler Apache::PerlRun
Options ExecCGI
PerlSendHeader On
</Location>

</IfModule>
```

これらのエントリは、Apache::RegistryモードとApache::PerlRunモード用にエイリアスを作成します。この2つのモードの違いは以下のとおりです。

**Apache::Registry** すべてのスクリプトはコンパイルされて、キャッシュに保存されます。すべてのスクリプトはサブルーチンのコンテンツとして適用されます。これはパフォーマンスを高めますが、要求が変わっても変数やサブルーチンはそのままなので、スクリプトを非常に慎重にプログラムしなければならないという短所もあります。このため、次の要求を処理できるように、変数をリセットする必要があります。たとえば、オンラインバンキングのスクリプトで顧客のクレジットカード番号を変数に格納した場合、次の顧客がアプリケーションを使用して同じスクリプトを要求したときに、前の顧客の番号がもう一度表示される可能性があります。

**Apache::PerlRun** スクリプトは要求ごとに再コンパイルされます。変数とサブルーチンは、要求の後、次の要求までにネームスペースから削除されます(ネームスペースとは、スクリプトの存在時に定義されるすべての変数名とルーチン名の総称です)。したがって、Apache::PerlRunでは、スクリプトの開始時にすべての変数が再初期化され、前回の要求の値が残ることはないので、綿密なプログラミングは必要ありません。このため、Apache::PerlRunはApache::Registryよりも低速ですが、CGIのようにインタプリタに対して別のプロセスを起動するわけではないので、(CGIと似たところはありますが) CGIよりはるかに高速です。

### 30.8.6 mod_php4

PHP は、Webサーバで使用するために特に開発されたプログラミング言語です。他の言語がコマンドを別のファイル(スクリプト)に格納するのに対

し、PHPコマンドは(SSIと同様) HTMLページに埋め込みます。PHPインタプリタは、PHPコマンドを処理し、処理結果をHTMLページに埋め込みます。

PHPのホームページはhttp://www.php.net/です。PHPを使用するには、mod_php4-coreに加え、Apache 2のapache2-mod_php4をインストールします。

### 30.8.7 mod_python

Pythonは、非常に明快で読みやすい構文を持つ、オブジェクト指向プログラミング言語です。一般とは異なりますが、便利な機能として、プログラム構造がインデントに依存することが挙げられます。ブロックは(CやPerlのように)中カッコや他の区切り要素(begin、endなど)で定義するのではなく、インデントのレベルで定義します。インストールするパッケージは、apache2-mod_pythonです。

この言語の詳細については、http://www.python.org/を参照してください。mod_pythonの詳細については、URL http://www.modpython.org/を参照してください。

### 30.8.8 mod_ruby

Rubyは、比較的新しいオブジェクト指向の高レベルプログラミング言語です。PerlやPythonと似た側面を持ち、スクリプト作成に最適です。Pythonと同様、明快で透明性の高い構文を持ちます。一方、Rubyでは入力ファイルの最終行の番号を$.rで表すなどの省略記法が採用されており、この特徴はプログラマによって賛否両論です。Rubyの基本コンセプトは、Smalltalkに非常によく似ています。

Rubyのホームページはhttp://www.ruby-lang.org/です。Ruby用のApacheモジュールが用意されています。ホームページはhttp://www.modruby.net/です。

## 30.9 仮想ホスト

仮想ホストを使用すると、1台のWebサーバを複数のドメインのホストとして機能させることができます。これにより、ドメインごとに別のサーバを用意するためのコストと管理作業負荷が不要になります。次の数種類の仮想ホストがあります。

- 名前ベースの仮想ホスト
- IPベースの仮想ホスト
- 1台のコンピュータ上でのApacheの複数インスタンスの使用

### 30.9.1 名前ベースの仮想ホスト

名前ベースの仮想ホストでは、Apacheの1つのインスタンスが複数のドメインのホストとして機能します。コンピュータに複数のIPアドレスを設定する必要はありません。これが最も簡単で、適切な選択肢です。名前ベースの仮想ホストが適切でない場合については、Apacheのマニュアルを参照してください。

この方法では、設定ファイル(/etc/apache2/httpd.conf)を使用して直接設定します。名前ベースの仮想ホストを有効にするには、適切なディレクティブを指定します。Apache がすべての着信要求を受け取るようにするには、NameVirtualHost *.* と指定します。その後、個々のホストを設定します。

```
<VirtualHost *>
 ServerName www.example.com
 DocumentRoot /srv/www/htdocs/example.com
 ServerAdmin webmaster@example.com
 ErrorLog /var/log/apache2/www.example.com-error_log
 CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>


<VirtualHost *>
 ServerName www.myothercompany.com
 DocumentRoot /srv/www/htdocs/myothercompany.com
 ServerAdmin webmaster@myothercompany.com
 ErrorLog /var/log/apache2/www.myothercompany.com-error_log
 CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>
```

そのサーバで元々提供していたドメイン(www.example.com)についても、VirtualHostエントリを設定する必要があります。この例では、元のドメインと追加のドメイン(www.myothercompany.com)の2つのホストとして、同じサーバが機能します。

NameVirtualHostと同様に、VirtualHostディレクティブでも*を使用します。Apacheは、HTTPヘッダのホストフィールドを使用して、要求を仮

想ホストに結び付けます。要求は、このフィールドに指定されたホスト名とServerNameが一致する仮想ホストに転送されます。

ディレクティブErrorLogとCustomLogでは、ログファイル名にドメイン名を含める必要はありません。ここでは、自由に名前を選択できます。

ServerAdminには、問題が発生したときに連絡先となる責任者の電子メールアドレスを指定します。エラーが生じた場合、Apacheはこのアドレスをエラーメッセージに含めてクライアントに送信します。

## 30.9.2 IPベースの仮想ホスト

この方法では、1つのコンピュータに対して複数のIPアドレスを設定する必要があります。この場合、Apacheのインスタンスは、複数のドメインにホストとしてサービスを提供し、各ドメインに別のIPアドレスが割り当てられることになります。次の例では、Apacheを設定して、元のIPアドレス(192.168.1.10)と追加IPアドレス(192.168.1.20と192.168.1.21)上の2つの追加ドメインのホストして機能させる方法を示します。IPアドレス192.168.0.0～192.168.255.0はインターネットにルーティングされないので、この例はイントラネットでのみ使用できます。

### IPエイリアスの設定

Apacheで複数のIPアドレスにサービスを提供するには、使用するコンピュータが複数のIPアドレスに対する要求を受け付ける必要があります。これをマルチIPホスティングと呼びます。このためには、カーネルでIPエイリアスを有効にする必要があります。これは、SUSE LINUXのデフォルトの設定です。

カーネルでIPエイリアスを設定すると、コマンドifconfigとrouteを使用して、ホスト上に追加のIPアドレスが設定されます。これらのコマンドは、rootユーザで実行する必要があります。次の例では、ホストがすでに独自のIPアドレス(192.168.1.10)を持っており、それがネットワークデバイスeth0に割り当てられていると想定しています。

コマンドifconfigを実行してホストのIPアドレスを確認します。また、次のコマンドでさらにIPを追加できます。

```
ip addr add 192.168.1.20.24 dev eth0
```

これらのIPアドレスはすべて、同じ物理ネットワークデバイス(eth0)に割り当てられています。

**IPアドレスを持つ仮想ホスト**

いったんIPエイリアスをシステムに設定するか、ホストに複数のネットワークカードを設定すれば、Apacheが設定可能になります。すべての仮想サーバについて、VirtualHostブロックを個別に指定します。

```
<VirtualHost 192.168.1.20>
 ServerName www.myothercompany.com
 DocumentRoot /srv/www/htdocs/myothercompany.com
 ServerAdmin webmaster@myothercompany.com
 ErrorLog /var/log/apache2/www.myothercompany.com-error_log
 CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.21> ServerName www.anothercompany.com DocumentRoot /srv/www/htdocs/anoth
```

VirtualHostディレクティブは、追加のドメインにのみ指定します。元のドメイン(www.example.com)は、VirtualHostブロックの外で独自の指定(DocumentRootなど)によって設定されます。

## 30.9.3 Apacheの複数インスタンス

上で述べた仮想ホスト用のメソッドを使用して、1つのドメインの管理者が他のドメインのデータを表示することができます。個々のドメインを隔離するには、Apacheのインスタンスを複数起動し、設定ファイルのUser、Group、および他のディレクティブに別個の設定を行います。

設定ファイルで、Listenディレクティブを使用して、Apacheのそれぞれのインスタンスが管理するIPアドレスを指定します。上の例では、Apacheの最初のインスタンスは次のようになります。

```
Listen 192.168.1.10:80
```

他の2つのインスタンスは、次のとおりです。

```
Listen 192.168.1.20:80
Listen 192.168.1.21:80
```

# 30.10 セキュリティ

## 30.10.1 リスクの最小化

コンピュータ上にWebサーバが必要なければ、ランレベルエディタでApacheを無効にし、アンインストールするか、最初からインストールしないようにします。リスクを最小化するには、不要なサーバをすべて無効化します。特に、ファイアウォールとして使用されているホストはこのことに注意が必要です。可能な限り、これらのホスト上でサーバを実行しないでください。

## 30.10.2 アクセス権

### DocumentRootの所有者はrootでなければならない

デフォルトでは、DocumentRootディレクトリ(/srv/www/htdocs)およびCGIディレクトリの所有者はユーザrootになっています。この設定は変更しないでください。これらのディレクトリにすべてのユーザが書き込めるようにすると、どのユーザでもそこにファイルを配置できるようになります。このようにすると、それらのファイルが、Apacheによってユーザ wwwrunのアクセス権で実行される可能性があります。また、Apacheは、配信するデータやスクリプトに対する書き込み権も持たないようにしなければなりません。したがって、これらのディレクトリの所有者は、ユーザwwwrunではなくrootなど別のユーザである必要があります。

ユーザがApacheのドキュメントディレクトリにファイルを配置できるようにする場合、全ユーザを書き込み可能に設定しないでください。その代わり、サブディレクトリを作成して(たとえば、/srv/www/htdocs/miscellaneous)、それを全ユーザが書き込めるように設定します。

### ホームディレクトリからのドキュメントの発行

ユーザにファイルの発行を許可する必要がある場合は、各ユーザのホームディレクトリのサブディレクトリをWeb発行用のディレクトリとして宣言することができます。このサブディレクトリには、慣習的に~/public_htmlという名前が付けられます。SUSE LINUXでは、デフォルトでこれが有効になっています。詳細については、項30.6.2. 「UserDir」を参照してください。

これらのWebページは、URLでユーザを指定してアクセスできます。このURLには、各ユーザのホームディレクトリにあるサブディレクトリへのショートカットを表す~usernameという要素を含めます。たとえば、ユー

ザtuxのホームディレクトリにあるサブディレクトリpublic_html内のファイルを一覧表示するには、ブラウザでhttp://localhost/~tuxと入力します。

### 30.10.3 最新情報の収集

Webサーバを運用している場合、特にそのWebサーバを一般に公開している場合は、バグや潜在的に脆弱な部分について、常に情報を収集しておく必要があります。情報源や修正プログラムについては、項30.12.3. 「セキュリティ」を参照してください。

## 30.11 トラブルシューティング

Apacheでページが表示されない、または表示が正しくないなどの問題が発生した場合は、次の手順に従って問題を特定します。まず、エラーログを見て、そこに含まれているメッセージがエラーの解明につながるかをチェックします。一般的なエラーログは、/var/log/apache2/error_logにあります。

確実なアプローチとして、コンソールでログファイルを追跡し、アクセスに対するサーバの対応を調べます。この作業は、root権限でコンソールから次のコマンドを実行することにより行えます。

```
tail -f /var/log/;apache2/*_log
```

http://bugs.apache.org/でオンラインバグデータベースをチェックします。関連のメーリングリストやニュースグループを読みます。ユーザ向けのメーリングリストは、http://httpd.apache.org/userslist.htmlにあります。ニュースグループの中では、comp.infosystems.www.servers.unixやその関連のグループを推奨します。

これらの方法を全部試みても問題が解決されず、バグがApacheで検出されたことが間違いない場合は、http://www.suse.de/feedback/にご連絡ください。

## 30.12 詳細情報

Apacheは、広く普及しているWebサーバです。そのため、多数のドキュメントがありますが、多くのWebサイトでApacheのヘルプとサポートを提供しています。

### 30.12.1 Apache

Apacheには詳細なマニュアルが付属しています。これらのマニュアルのインストール方法については、項30.5.「インストール」を参照してください。インストールすると、http://localhost/manualでマニュアルにアクセスできるようになります。最新のマニュアルは、Apacheホームページ(http://httpd.apache.org)で参照できます。

### 30.12.2 CGI

CGIの詳細については、次のWebページを参照してください。

- http://apache.perl.org/
- http://perl.apache.org/
- http://www.modperl.com/
- http://www.modperlcookbook.org/
- http://www.fastcgi.com/
- http://www.boutell.com/cgic/

### 30.12.3 セキュリティ

SUSE LINUXパッケージの最新のパッチは、http://www.novell.com/linux/security/securitysupport.htmlから入手可能です。このURLを定期的にチェックしてください。ここではまた、セキュリティ情報についてのSUSEメーリングリストへもご登録いただけます。

Apacheチームは、Apacheのバグについて情報を積極的に公開する方針をとっています。最新のバグレポートおよび潜在的な脆弱性については、http://httpd.apache.org/security_report.htmlに公開しています。セキュリティバグを発見した場合は、(既知のバグでないことを上のページで確認の上) security@suse.de または security@apache.orgにお知らせください。

Apacheのセキュリティ問題に関するその他の情報源(および他のインターネットプログラム)は、次のとおりです。

- http://www.cert.org/

- http://www.vnunet.com/
- http://www.securityfocus.com/

### 30.12.4 その他の情報源

問題が発生したときは、SUSEサポートデータベース(http://portal.suse.com/sdb/en/index.html)を参照してください。Apache関連のオンラインニュースペーパーが、http://www.apacheweek.com/で参照できます。

Apacheの沿革は、http://httpd.apache.org/ABOUT_APACHE.htmlで参照できます。このページでは、Apacheというサーバ名の由来についても説明しています。

バージョン1.3から2.0へのアップグレード情報もhttp://httpd.apache.org/docs-2.0/en/upgrading.htmlで参照できます。

# 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。

# 31.1 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。

**Warning**

**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。

**Warning**

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こ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すみます。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。

## 31.1.1 Unison

Unisonは、ネットワークファイルシステムではありません。ファイルは単にローカルで保存、編集されます。プログラムUnisonは、手動で実行してファイルを同期させます。同期を初めて実行すると、2台のホスト上にデータベースが作成され、チェックサム、タイムスタンプ、および選択したファイルへのアクセス許可が保存されます。次に実行すると、Unisonはどのファイルが変更されたかを認識でき、ホスト間の伝送を提案します。通常、すべての提案は了承できます。

### 31.1.2 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レポジトリを設定し、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レ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。競合が生じても、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。

### 31.1.3 subversion

進化を遂げたCVSとは異なり、subversionは一貫して設計されたプロジェクトです。 subversionは、技術面を改良したCVSの後継バージョンとして開発されました。

subversionは、従来のタイプに比べてさまざまな面で改良されています。このような経緯があるため、CVSで維持されるのはファイルのみで、ディレクトリは対象外です。subversionではディレクトリも同様にバージョン履歴をもつため、ファイルと同じようにコピーしたり、名前を変更することができます。また、すべてのファイルとすべてのディレクトリにメタデータを追加できます。このメタデータはバージョニング機能により完全に維持できます。CVSとは異なり、subversionではWebDAV (Web-based Distributed Authoring and Versioning)のような専用プロトコルを介した透過型ネットワークアクセスがサポートされます。WebDAVでは、HTTPプロトコルの機能を拡張して、リモートWebサーバ上のファイルへの書き込みアクセスを共同して行うことを可能にしています。

subversionは既存のソフトウェアパッケージとの組み合わせを念頭に置いて作られています。そのため、Apacheウェブサーバおよび拡張WebDAVは常にsubversionと組み合わせて実行されます。

### 31.1.4 mailsync

これまでに説明した同期ツールとは異なり、mailsyncはメールボックス間の電子メールの同期だけを実行します。プロシージャは、ローカルのメールボックスファイルとIMAPサーバのメールボックスの両方に適用されます。

電子メールのヘッダに記載されているメッセージIDに基づいて、個々のメッセージを同期させるか、削除します。同期は個別のメールボックス間およびメールボックスの階層間で実行できます。

### 31.1.5 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。

## 31.2 プログラムを選択する場合の決定要因

### 31.2.1 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、subversion、CVS、およびWebDAVで採用されています。

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。これは、unisonで採用されている概念です。実際には、rsyncはクライアントモードで動作しますが、すべてのクライアントがサーバとしても動作できます。

### 31.2.2 移植性

subversion、CVS、およびunison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。

### 31.2.3 インタラクティブと自動制御

subversion、CVS、WebDAV、およびUnison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。

### 31.2.4 競合: 発生と解決

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、subversionまたは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。競合が起こると、影響を受けるのは1台のクライアントだけです。subversionやCVSでは、普通、競合が容易に解決できます。

Unisonは、競合をレポートし、影響を受けたファイルを同期処理から排除します。しかしながら、subversionやCVSでは、変更のマージが容易ではありません。

競合時に変更を部分的に受け入れることができるsubversionやCVSとは対照的に、WebDAVでは、変更が全体的に成功したと見なせる場合にのみチェックインを行います。

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などのバージョニングシステムを追加採用できます。

### 31.2.5 ファイルの選択と追加

標準設定では、Unisonはディレクトリツリー全体の同期が行われます。ファイルシステムに新しく追加したファイルが、自動的に他のコンピュータに表示されます。

subversionまたはCVSでは、新しいディレクトリとファイルは、それぞれコマンド`svn add`または`cvs add`を使用し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`svn update`および`svn status`または`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。

### 31.2.6 履歴

subversionまたはCVSは追加機能として、古いバージョンのファイルを再構成できます。変更を行うたびに簡単な編集コメントを挿入しておくと、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。

### 31.2.7 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。subversionおよびCVSでは、サーバ上のレ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。

### 31.2.8 GUI

Unisonはグラフィカルユーザインタフェースを備え、Unisonが実行する同期手順を画面に表示します。提案を了承するか、個別のファイルを同期処理から排除します。テキストモードでは、個々の手順を対話型で確認します。

subversionまたは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。しかしながら、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やsubversion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。

### 31.2.9 使いやすさ

Unisonとrsyncは使いやすく、初心者にも適しています。CVSとsubversionは、やや操作が難しいプログラムです。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 update`または`svn update`で実行します。次にコマンド`cvs commit`または`svn 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、初心者でもCVSまたはsubversionを簡単に利用できるようになります。

### 31.2.10 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。Unison、CVS、rsync、およびsubversion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やUnison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。 subversionは、Apacheで実行することで既に必要なセキュリティ対策を提供しています。

### 31.2.11 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。subversionはCVSほど普及してはいませんが、生産的な環境(subversionプロジェクト自体など)に採用されつつあります。

Unisonはまだ比較的新しいプログラムですが、ハイレベルな安定性を誇っています。しかし、ユーザエラーには効果的に対応できません。いったんファイルを削除するという同期処理が確定されたら、そのファイルを復元する手立てはありません。

*Table 31.1: ファイル同期ツールの機能:-- = よくない - = あまりよくないまたはサポート対象外、o = 普通、+ = よい、++ = 非常によい、x = サポートされている*

| | unison | CVS/subv. | rsync |
|---|---|---|---|
| クライアント/サーバ | 同等 | C-S/C-S | C-S |
| 移植性 | Lin、Un*x、Win | Lin、Un*x、Win | Lin、Un*x、Win |
| 対話処理 | x | x/x | x |
| 速度 | - | o/+ | + |
| 競合 | o | ++/++ | o |
| ファイル選択 | ディレクトリ | 選択/ファイル、ディレクトリ | ディレクトリ |
| 履歴 | - | x/x | - |
| ハードディスクスペース | o | -- | o |
| GUI | + | o/o | - |

| | | | |
|---|---|---|---|
| 難度 | + | o/o | + |
| 攻撃 | +(ssh) | +/+(ssh) | +(ssh) |
| データ損失 | + | ++/++ | + |

## 31.3 Unisonの概要

Unisonは、ディレクトリツリー全体を同期させ、転送するための優れたソリューションです。同期は双方向に実行され、直観的なグラフィカルフロントエンドによって制御できます。コンソールバージョンも使用できます。同期処理を自動化できるため、ユーザとの対話が不要になりますが、使用するには経験が必要です。

### 31.3.1 必要条件

Unisonは、クライアントとサーバの両方にインストールする必要があります。ここでサーバとは、2番目のリモートホストを指します(CVSとは異なります。項31.1.2. 「CVS」を参照)。

ここでは、Unisonをsshと共に使用します。この場合、SSHクライアントをクライアントにインストールし、SSHサーバをサーバにインストールする必要があります。

### 31.3.2 Unisonの使用

Unisonで採用されているアプローチは、2つのディレクトリ(*roots*)を互いに関連付けるという方法です。この関連付けはシンボリックです。つまり、オンライン接続があるわけではありません。この例のディレクトリレイアウトは次のとおりです。

```
クライアント:  /home/tux/dir1
サーバ:        /home/geeko/dir2
```

それでは、これらの2つのディレクトリを同期させましょう。ユーザは、クライアント上のtuxおよびサーバ上のgeekoとして認識されています。最初に、クライアントサーバ間通信が有効かどうかをテストします。

```
unison -testserver /home/tux/dir1 ssh://geeko@server//homes/geeko/dir2
```

最もよくある問題は次のとおりです。

- クライアントとサーバで使用されるUnisonのバージョンに互換性がない。
- サーバがSHH接続を使用できない。
- 指定されたパスがどちらも存在しない。

これらに問題がない場合は、オプション-testserverを省略します。最初の同期では、Unisonがまだ2つのディレクトリ間の関係を把握していないため、個々のファイルやディレクトリの転送方向についての提案が表示されます。['Action']列の矢印は、転送方向を示します。疑問符は、両方のバージョンが変更されている、または両方が新規のため、転送方向についてUnisonが提案を行えないことを示します。

矢印キーを使用して、個々のエントリの転送方向を設定します。表示されているすべてのエントリについて転送方向が適切な場合は、['Go']をクリックします。

Unisonの特性(たとえば、問題のないケースについて同期を自動実行するかどうか)は、プログラムの開始時にコマンドラインパラメータで指定して制御することができます。すべてのパラメータのリストは、unison --helpコマンドで表示できます。

***Example 31.1:*** *ファイル ~/.unison/example.prefs*

```
root=/home/tux/dir1
root=ssh://wilber@server//homes/wilber/dir2
batch=true
```

各ペアについての同期ログが、ユーザディレクトリ~/.unisonに保存されます。~/.unison/example.prefsのような設定セットもこのディレクトリに保存できます。同期を開始するには、このファイルをコマンドラインパラメータとしてunison example.prefsのように指定します。

### 31.3.3 関連資料

Unisonの公式マニュアルは、非常に役に立ちます。そのため、ここでは簡単な概要だけを説明しました。このマニュアルは、`http://www.cis.upenn.edu/~bcpierce/unison/`とSUSEパッケージunisonに完全版が用意されています。

## 31.4 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。

### 31.4.1 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 CVS_ROOT=tux@server:/serverdir
```

コマンド`cvs init`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
```

### 31.4.2 CVSの使用

これで、すべてのホストがcvs co synchomeを使用して同期レ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(前述を参照)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、ファイルの競合マーク(>と<)を確認し、2つのバージョンのどちらを採用するかを決定します。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。

### 31.4.3　関連資料

ここでは、CVSが持つ多くの機能から、その概要だけを紹介しました。詳細については、多数のマニュアルが次のURLに用意されています。

```
http://www.cvshome.org/
http://www.gnu.org/manual/
```

## 31.5　subversionの概要

subversionは、無償で公開されているバージョン管理システムであり、一般にCVSの後継と見なされています。つまり、通常、CVSに導入済みの機能はsubversionにも組み込まれています。特に、CVSの長所を考慮しても短所を補いきれないと思われる場合に使用することをお勧めします。この種の機能のほとんどについては、既に項31.1.3. 「subversion」で簡単に紹介しています。

### 31.5.1 Subversionサーバのインストール

サーバにレポジトリーデータベースをインストールする処理は比較的簡単です。subversionには、そのための専用管理ツールが用意されています。新規レポジトリの作成コマンドは、次のとおりです。

```
svnadmin create /path/to/repository
```

svnadmin helpを使用すると、その他のオプションをリストできます。CVSとは異なり、subversionはRCSベースではなくBerkeley Databaseベースです。レポジトリはNFS、AFS、またはWindows SMBのようなリモートファイルシステムにインストールしないでください。データベースにはPOSIXロックメカニズムが必要ですが、これらのファイルシステムではこのメカニズムがサポートされていません。

コマンドsvnlookを実行すると、既存のレポジトリに関する情報が表示されます。

```
svnlook info /path/to/repository
```

異なる複数のユーザに対してレポジトリへのアクセスを許可するには、サーバを設定する必要があります。WebDAVとともにApache Webサーバを使用するか、subversionに含まれているサーバパッケージsvnserve を使用します。svnserveを起動すると、svn://またはsvn+ssh://というURLでレポジトリにアクセスできるようになります。svnを呼び出すときに自己認証が必要なユーザは、etc/svnserve.confで設定できます。

Apacheとsvnserveのどちらを使用するかについては、さまざまな判断基準があります。これについては、subversionのマニュアルを参照することをお勧めします。詳細については、項31.5.3. 「関連資料」を参照してください。

### 31.5.2 使用方法と操作

subversionレポジトリにアクセスするには、コマンドsvn(cvsに類似)を使用します。対応するレポジトリに合わせて適切に設定されたサーバから提供されるコンテンツには、どのクライアントからも次のいずれかのコマンドを使用してアクセスできます。

```
svn list http://svn.example.com/path/to/project
```

または

```
svn list http://svn.example.com/path/to/project
```

既存のプロジェクトを現行のディレクトリに保存(チェックアウト)するには、コマンドsvn checkoutを使用します。

```
svn list http://svn.example.com/path/to/project
```

チェックアウトすると、クライアント上に新規のサブディレクトリnameofprojectが作成されます。これで、そのサブディレクトリに対して操作(追加、コピー、名前の変更、削除)を実行できます。

```
svn add file
svn copy oldfile newfile
svn move oldfile newfile
svn delete file
```

これらのコマンドは、ディレクトリに対しても使用できます。 ubversionでは、ファイルやディレクトリのプロパティも記録できます。

```
svn propset license GPL foo.txt
```

この例では、プロパティlicenseの値GPLを設定しています。プロパティを表示するには、svn proplistを使用します。

```
svn proplist --verbose foo.txt
Properties on 'foo.txt':
license :GPL
```

変更内容をサーバに保存するにはsvn commitを使用します。他のユーザはsvn updateを使用してサーバと同期させることで、変更内容を自分の作業ディレクトリに取り込むことができます。

CVSとは異なり、svn statusを使用すると、レポジトリにアクセスしなくてもsubversionの作業ディレクトリのステータスを表示できます。ローカルの変更は5列に表示され、1列目が最も重要です。

″　変更はありません。

'A'　オブジェクトには追加マークが付いています。

'D'　オブジェクトには削除マークが付いています。

'M' オブジェクトは変更されています。

'C' オブジェクトは競合しています。

'I' オブジェクトは無視されました。

'?' オブジェクトはバージョン管理対象ではありません。

'!' オブジェクトは欠落としてレポートされています。このフラグが表示されるのは、オブジェクトがsvnコマンドを使用せずに削除または移動された場合です。

'~' ファイルとして扱われていたオブジェクトがディレクトリで置換された、またはその逆の処理が発生しました。

2列目はプロパティのステータスを示します。他の各列の意味については、subversionのマニュアルを参照してください。

コマンドパラメータの説明を表示するには、コマンドsvn helpを使用します。

```
svn help proplist
proplist (plist, pl):ファイル、ディレクトリ、修正のすべてのプロパティを表示します。
1. proplist [PATH...]
2. proplist --revprop -r REV [URL]

1. 作業用コピーのバージョン付きプロパティをリストします。
2. レポジトリー修正のバージョンなしリモートプロパティをリストします。
...
```

### 31.5.3 関連資料

最初に、http://subversion.tigris.org/にアクセスしてsubversionプロジェクトのホームページを参照してください。パッケージsubversion-docによってディレクトリfile:///usr/share/doc/packages/subversion/html/book.htmlにインストールされるマニュアルも非常に参考になります。このマニュアルはhttp://svnbook.red-bean.com/svnbook/index.htmlでも入手できます。

## 31.6 rsyncの概要

rsyncは、大量のデータを定期的に送信する必要があるが、変更量はあまり多くない場合に便利です。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。

### 31.6.1 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、ホームディレクトリtux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、rcrsyncd start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「insserv rsyncd」を入力します。また、rsyncdをxinetdで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。次に例を示します。

```
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。

### 31.6.2 関連資料

rsyncに関する重要な情報は、マニュアルページman rsyncおよびman rsyncd.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。

## 31.7 mailsyncの概要

mailsyncは、主に次の3種類のタスクに適しています。

- ローカルに保存されている電子メールをサーバに保存されているメールと同期させる。
- メールボックスを異なる形式または異なるサーバに移行する。
- メールボックスの完全性チェックまたは重複の検索を行う。

### 31.7.1 設定と使用

mailsyncは、メールボックス自体(ストア)と2つのメールボックス間の接続(チャネル)を区別します。ストアとチャネルの定義は、~/.mailsyncで説明されています。ここでは、ストアの例をいくつか示します。

単純な定義は次のようになります。

```
store saved-messages {
  pat Mail/saved-messages
prefix  Mail/
}
```

Mail/とは、ユーザのホームディレクトリのサブディレクトリであって、フォルダsaved-messagesをはじめとする電子メールフォルダが格納されています。mailsyncがmailsync -m saved-messagesで始まっている場合、すべてのメッセージのインデックスは、saved-messagesにリストされます。次のように定義されている場合、

```
store localdir {
pat     Mail/*
prefix  Mail/
}
```

コマンドmailsync -m localdirを実行すると、Mail/の下位に保存されているすべてのメッセージがリストされます。これとは異なり、コマンドmailsync localdirを実行するとフォルダ名がリストされます。IMAPサーバでのストアの指定は次のようになります。

```
store imapinbox {
server {mail.edu.harvard.com/user=gulliver}
ref    {mail.edu.harvard.com}
pat    INBOX }
```

上の例は、単にIMAPサーバ上のメインフォルダのアドレス指定です。サブフォルダのストアは次のように表示されます。

```
store imapdir {
server {mail.edu.harvard.com/user=gulliver}
ref {mail.edu.harvard.com}
pat INBOX.*
prefix  INBOX.
}
```

IMAPサーバが暗号化接続をサポートしている場合、サーバ指定を次のように変更する必要があります。

```
server {mail.edu.harvard.com/ssl/user=gulliver}
```

変更しなければ、サーバ証明書が次のサーバに認識されません。

```
server {mail.edu.harvard.com/ssl/novalidate-cert/user=gulliver}
```

プレフィクスについては、後で説明します。

ここでMail/の下位のフォルダをIMAPサーバのサブディレクトリに接続する必要があります。

```
channel folder localdir imapdir {
msinfo .mailsync.info
}
```

mailsyncはmsinfoファイルを使用して、既に同期されているメッセージを追跡します。

コマンドmailsync folderを実行すると、次の処理が行われます。

- メールボックスパターンが、両方の側で拡張されます。
- 作成されたフォルダ名からプレフィクスが取り除かれます。
- フォルダがペアとして同期されます(片方が存在しない場合は作成されます)。

これにより、IMAPサーバ上のINBOX.sent-mailが、ローカルフォルダMail/sent-mail(前述の定義が存在する場合)と同期されます。個々のフォルダ間の同期は、次のように実行されます。

- メッセージが両方の側に存在する場合は何も行いません。
- 片側にメッセージが存在せず、新規メッセージ(msinfoファイルに存在しない)の場合は送信されます。
- 単にメッセージが片側に存在し、古いメッセージ(msinfoファイルに存在する)の場合は削除されます(もう片方に以前存在したメッセージが削除されているため)。

同期によって、どのメッセージが送信送され、どのメッセージが削除されるかを事前に確認するには、mailsync folder localdirを使用して、チャネルとストアの*両方*に対しmailsyncを実行します。このコマンドを実行すると、ローカルホストにあるすべての新規メッセージのリストと共に、同期の際にIMAP側で削除されるすべてのメッセージのリストが作成されます。同様に、コマンドmailsync folder imapdirを実行すると、IMAP側にあるすべての新規メッセージのリストと共に、同期の際にローカルホストで削除されるすべてのメッセージのリストが作成されます。

### 31.7.2 起こり得る問題

データが損失した場合、最も安全な方法は、関連のチャネルログファイルmsinfoを削除することです。これにより、片方だけに存在するメッセージはすべて新規とみなされ、次回の同期の際に送信されます。

同期の対象となるのは、メッセージIDを持つメッセージのみです。メッセージIDのないメッセージは無視され、送信も削除もされません。メッセージIDのないメッセージは、通常、そのメッセージの送信または作成時にプログラムに障害が発生します。

IMAPサーバによっては、メインフォルダがINBOXとして識別され、そのサブフォルダが無作為に選択された名前(INBOXとINBOX.nameではなく)で識別されます。このようなIMAPサーバでは、サブフォルダだけに使用されるパターンを指定することができません。

IMAPサーバにメッセージを正常に送信すると、mailsyncが使用するメールボックスドライバ(c-client)が、特別なステータスフラグを設定します。このため、muttなど一部の電子メールプログラムでは、これらのメッセージを新規として認識できません。この特別なステータスフラグの設定を無効にするには、オプション-nを使用します。

### 31.7.3 関連資料

mailsyncの/usr/share/doc/packages/mailsync/にあるREADMEには、関連情報が記述されています。またこのトピックに関しては、RFC 2076“Common Internet Message Headers”が特に参考になります。

32

Samba

# Samba

Sambaを使用すると、DOS、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。この章では、Sambaの設定に関する基本事項について、およびネットワーク上でのSambaの設定に使用できるYaSTモジュールについて説明します。

Sambaは、今や成熟の域に達したかなり複雑な製品です。この章では、Sambaの基本機能について概説します。詳細は、付属のドキュメントに記載されています。コマンドラインからapropos samba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をインストール済であれば、/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。

付属のバージョン3のsambaパッケージは、次のような重要な新機能を備えています。

- Active Directoryのサポート
- Unicodeサポートの拡張
- 内部認証メカニズムの全面改訂
- Windows 200xおよびXP印刷システムの大幅な機能向上
- Active Directoryドメインのメンバサーバとしてのサーバのセットアップ
- NT4ドメインの採用と、NT4ドメインからSambaドメインへの移行の実現

**Tip**

**Samba3への移行**

Samba 2.xからSamba 3に移行するときは、いくつか特別な配慮が必要です。このトピックについては、『Samba HOWTO Collection』で1章全部を費やして説明されています。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HOWTOを確認してください。

**Tip**

SambaはSMBプロトコル(サーバメッセージブロック)を使用します。SMBはNetBIOSサービスを基にしています。IBMからの圧力によって、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。

は、マシン間通信用に設計されたソフトウェアインタフェース(API)です。ここではネームサービスが提供されています。これにより、ネットに接続された

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。名前をチェック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。現在、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用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。

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。

SMBサーバは、そのクライアントに対し、共有によってハードウェア空間を提供します。共有には、サーバ上のディレクトリとそのサブディレクトリが含まれます。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でアクセスできます。

## 32.1 サーバの設定

Sambaをサーバとして使用する場合は、`samba`をインストールする必要があります。Sambaに必要なサービスは、`rcnmb start && rcsmb start`で起動し、`rcsmb stop && rcnmb stop`で停止します。

Sambaの主となる設定ファイルは`/etc/samba/smb.conf`です。このファイルは2つの論理部分に分けられます。`[global]`セクションには、中心的なグローバル設定が含まれます。`[share]`セクションには、個別のファイルとプリンタ共有が入っています。このアプローチにより、共有に関する詳細は`[global]`セクションで個別に、またはグローバルに設定することができ、設定ファイルの構造的透過性が高まっています。

### 32.1.1 グローバルセクション

[global]の次のパラメータは、ネットワークの設定に応じた必要条件を満たし、Windows環境で他のマシンがSMBを経由してこのSambaサーバにアクセスできるようにするために多少の調整が必要です。

**workgroup = TUX-NET** この行は、Sambaサーバをワークグループに割り当てます。TUX-NETを実際のネットワーク環境にある適切なワークグループに置き換えてください。DNS名がネットワーク内の他のマシンに割り当てられていなければ、SambaサーバがDNS名の下に表示されます。DNS名が使用できない場合は、netbiosname=MYNAMEを使用してサーバ名を設定します。このパラメータについての詳細はmansmb.confを参照してください。

**os level = 2** このパラメータは、Sambaサーバがワークグループの LMB(ローカルマスタブラウザ)になるかどうかのきっかけとなります。Sambaサーバの設定が誤っていた場合に、既存のWindowsネットワークに支障が出ないよう、小さな値を選択します。この重要なトピックについての詳細は、パッケージマニュアルのtextdocsサブディレクトリにあるBROWSING.txtとBROWSING-Config.txtを参照してください。

ネットワーク内に他のSMBサーバ(たとえば、Windows NTまたは2000サーバ)が存在せず、ローカル環境に存在するすべてのシステムのリストをSambaサーバに保存する場合は、os levelの値を大きくします(たとえば、65)。これでSambaサーバが、ローカルネットワークのLMBとして選択されました。

この設定を変更するときは、それが既存のWindowsネットワーク環境にどう影響するかを慎重に検討する必要があります。まず、隔離されたネットワークで、または影響の少ない時間帯に、変更をテストしてください。

**wins supportとwins server** アクティブなWINSサーバをもつ既存のWindowsネットワークにSambaサーバを参加させる場合は、wins serverオプションを有効にし、その値をWINSサーバのIPアドレスに設定します。

各Windowsマシンの接続先サブネットが異なり、互いを認識させなければならない場合は、WINSサーバをセットアップする必要があります。SambaサーバをWINSサーバなどにするには、オプションwins support = Yesを設定します。ネットワーク内でこの設定が有効なSambaサーバは1台だけであることを確認します。smb.confファイル内で、オプションwins serverとwins supportは同時に有効にしないでください。

### 32.1.2 共有

次の例では、SMBクライアントがCD-ROMドライブとユーザディレクトリ(homes)を利用できるようにする方法を示します。

**[cdrom]** CD-ROMドライブが誤って利用可能になるのを避けるため、これらの行はコメントマーク(この場合はセミコロン)で無効にします。最初の列のセミコロンを削除し、CD-ROMドライブをSambaと共有します。

*Example 32.1: CD-ROMの 共 有*

```
;[cdrom]
;       comment = Linux CD-ROM
;       path = /media/cdrom
;       locking = No
```

**[cdrom]およびコメント** エントリ[cdrom]は、ネットワーク上のすべてのSMBクライアントが認識できる共有の名前です。さらにcommentを追加して、共有を説明することができます。

**path = /media/cdrom** pathオプションで、/media/cdromディレクトリをエクスポートします。

デフォルトを非常に制約的に設定することによって、このシステム上に存在するユーザのみがこの種の共有を利用できるようになります。この共有をあらゆるユーザに開放する場合は、設定にguest ok = yesという行を追加します。この設定は、ネットワーク上の全ユーザに読み込み許可を与えます。このパラメータを使用する場合には、相当な注意を払うことをお勧めします。またこのパラメータを[global]セクションで使用する場合には、さらに注意が必要です。

**[homes]** [home]共有は、ここでは特に重要です。ユーザがLinuxファイルサーバの有効なアカウントとパスワードを持ち、独自のホームディレクトリを持っていればそれに接続することができます。

*Example 32.2: homes共 有*

```
[homes]
    comment = Home Directories
    valid users = %S
    browseable = No
    read only = No
    create mask = 0640
    directory mask = 0750
```

**[homes]** SMBサーバに接続しているユーザの共有名を他の共有が使用していない限り、[homes]共有ディレクティブを使用して共有が動的に生成されます。生成された共有の名前は、ユーザ名と同じです。

**valid users = %S** %S は、接続が正常に確立されるとすぐに、具体的な共有名に置き換えられます。[homes]共有の場合、これは常にユーザ名と同じ名前です。したがって、ユーザの共有に対するアクセス権は、そのユーザだけに付与されます。

**browseable = No** この設定を行うと、共有がネットワーク環境で認識されなくなります。

**read only = No** デフォルトでは、Sambaはread only = Yesパラメータによって、エクスポートされた共有への書き込みアクセスを禁止します。共有に書き込めるように設定するには、値read only = Noを設定します。これはwriteable = Yesと同値です。

**create mask = 0640** MS Windows NTベース以外のシステムは、UNIXのパーミッションの概念を理解しないので、ファイルの作成時にアクセス権を割り当てることができません。パラメータcreate maskは、新しく作成されたファイルに割り当てられるアクセス権を定義します。これは書き込み可能な共有にのみ適用されます。実際、この設定はオーナーが読み書き権を持ち、オーナーの一次グループのメンバが読み込み権を持つことを意味します。valid users = %Sを設定すると、グループに読み込み権が与えられても、読み込みアクセスができなくなります。グループに読み書き権を付与する場合は、valid users = %Sという行を無効にしてください。

### 32.1.3 セキュリティレベル

SMBプロトコルはDOSやWindowsの世界から生まれ、セキュリティの問題についてもよく考慮されています。各共有へのアクセスは、パスワードによって保護されています。SMBには、パーミッションをチェックする方法が3つあります。

**共有レベルのセキュリティ(セキュリティ=共有):**
パスワードが共有に対し確実に割り当てられています。このパスワードを持っているユーザ全員が、その共有にアクセスできます。

**ユーザレベルのセキュリティ(セキュリティ=ユーザ):**
このセキュリティレベルは、ユーザという概念をSMBに取り入れていま

す。各ユーザは、サーバにパスワードを登録する必要があります。登録後、エクスポートされた個々の共有へのアクセスは、ユーザ名に応じてサーバが許可します。

**サーバレベルのセキュリティ(セキュリティ=サーバ):**
クライアントに対しては、Sambaがユーザレベルモードで動作しているように見えます。しかし、Sambaはすべてのパスワードクエリを別のユーザレベルモードサーバに渡し、ユーザレベルモードサーバが認証を行います。設定には追加のパラメータが必要です(`password server=`)。

共有、ユーザ、およびサーバレベルのセキュリティの区別は、サーバ全体に適用されます。個別の共有ごとに、ある共有には共有レベルのセキュリティ、別の共有にはユーザレベルセキュリティを設定するといったことはできません。しかし、システム上に設定したIPアドレスごとに、別のSambaサーバを実行することは可能です。

この詳細については、『Samba HOWTO Collection』を参照してください。つのシステムに複数のサーバをセットアップする場合は、オプション`interfaces`および`bind interfaces only`に注意してください。

**Tip**

Sambaサーバでの管理作業を簡単にするため、swatというプログラムも用意されています。このプログラムには、Sambaサーバを便利に設定するための簡単なWebインタフェースがあります。Webブラウザで、`http://localhost:901`を開き、rootユーザでログインします。ただし、swatをファイル`/etc/xinetd.d/samba`と`/etc/services`で有効にする必要があります。これには、`/etc/xinetd.d/samba`で`disable`の行を`disable = no`のように編集します。swatの詳細については、マニュアルページを参照してください。

**Tip**

## 32.2 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。これは、Sambaサーバで実現されます。Windowsベースのネットワー

クでは、このタスクはプライマリドメインコントローラ(PDC)として設定されたWindowsNTサーバによって処理されます。例 32.3. 「smb.confファイルのグローバルセクション」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。

***Example 32.3:** 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に使用されている場合は(強固に保守されたMS Windows 9xインストール、MS Windows NT 4.0(サービスパック3以降)、およびそれ以降にリリースされた製品でのデフォルト設定)、Sambaサーバでこれを処理できる必要があります。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 NT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。

***Example 32.4:** 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。

***Example 32.5:** 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。

## 32.3 YaSTでのSambaサーバの設定

サーバを設定するには、まず、新しいSambaサーバで制御するワークグループまたはドメインを選択します。['ワークグループまたはドメイン名']ドロップダウンメニューから既存のワークグループ/ドメインを選択するか、新しいワークグループ/ドメインを入力します。次に、サーバをPDC (プライマリドメインコントローラ)とBDC (バックアップドメインコントローラ)のどちらとして動作させるかを指定します。

['開始']タブで、Sambaを有効にします(図 32.1. 「Sambaの設定—起動」 を参照)。Sambaサーバを円滑に運用できるように、サーバ上のファイアウォールですべての外部インタフェースと内部インタフェースに対してnetbios-ns、netbios-dgm、netbios-ssn、microsoft-dsの各サービス用のポートを開くには、['ファイアウォールで開いているポート']を選択して、['ファイアウォールの詳細']を使用します。

['共有']タブ(図 32.2. 「Sambaの設定—共有」 )で、有効にするSambaの共有を指定します。 ['状態の変更']を使用して、['有効']と['無効']の間で切り替えます。新しい共有を追加するには['追加']をクリックします。

['ID']タブ(図 32.3. 「Sambaの設定—識別」 )で、ホストが関連付けられているドメイン(['基本設定'])と、ネットワークで代替ホスト名を使用するかどうか(['NetBIOSホスト名'])を指定します。

## 32.4 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。


*Figure 32.1: Sambaの 設 定—起 動*

### 32.4.1 YaSTでの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。 ['SAMBAワークグループ']ダイアログで、ドメインまたはワークグループを入力します。 ['検索']をクリックすると、使用可能なすべてのグループとドメインが表示され、マウスで選択することができます。 ['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。

### 32.4.2 Windows 9xおよびME

Windows 9xおよびMEには、あらかじめTCP/IPのサポートが組み込まれています。しかし、デフォルトでインストールされるわけではありません。TCP/IPを追加するには、'コントロール パネル' → 'システム'に移動し、'追加' → 'プロトコル' → 'TCP/IP'の順に選択します。Windowsマシンをリブートし、デスクトップでネットワーク環境のアイコンをダブルクリックしてSambaサーバを見つけます。

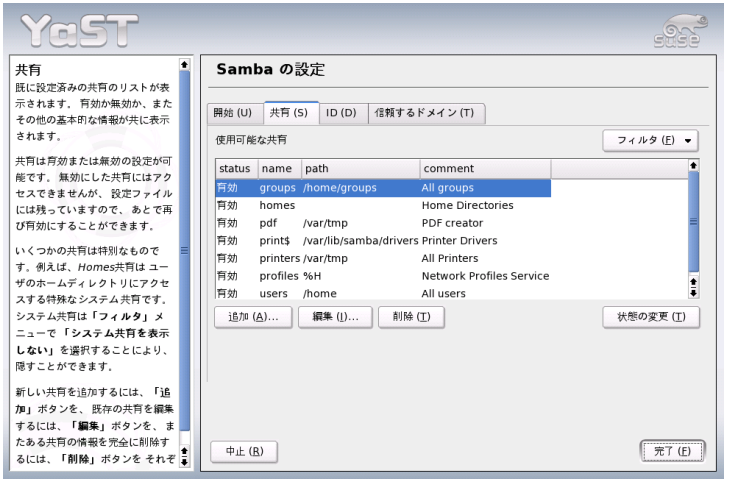


*Figure 32.2: Sambaの 設 定--共 有*

**Tip**

Sambaサーバ上でプリンタを使用するには、対応するWindowsバージョンから、標準のプリンタドライバまたはApple-PostScriptプリンタドライバをインストールします。これをLinuxプリンタキュー(Postscriptを入力形式として許容)にリンクするのが最適な方法です。

**Tip**

## 32.5 最適化

`socket options`は最適化の1つの選択肢であり、ご使用のバージョンのSambaにサンプル設定とともに付属しています。デフォルト設定は、ローカルのイーサネットネットワークを参照します。`socket options`の詳細については、`smb.conf`のマニュアルページの関連セクションと`socket(7)`のマニュアルページを参照してください。詳細については、『Samba HOWTO Collection』の「Samba performance tuning」の章を参照してください。


*Figure 32.3: Sambaの 設 定—識 別*

/etc/samba/smb.confの標準設定は、Sambaチームのデフォルト設定に基づいて、便利な設定ができるよう設計されています。ただし、ネットワーク設定やワークグループ名の点から、そのまま使える設定をあらかじめ提供するのは不可能です。コマンドサンプル設定examples/smb.conf.SuSEには、具体的な必要条件に合わせた調整に役立つ情報が含まれています。

**Tip**

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チェックするためのガイドが用意されています。

**Tip**

# 33

Squidプロキシサーバ

# Squidプロキシサーバ

Squidは、LinuxおよびUNIXプラットフォームで普及しているプロキシキャッシュです。この項では、Squidの設定、実行に必要な設定、透過型のプロキシ処理を実行するようにシステムを設定する方法、Calamarisやcachemgrなどのプログラムを使用してキャッシュの使用統計を収集する方法、およびsquidGuardを使用してWebコンテンツをフィルタする方法について説明します。

# 33.1 プロキシキャッシュとしてのSquid

Squidはプロキシキャッシュとして機能します。クライアント(この場合はWebブラウザ)からのオブジェクト要求をサーバにリダイレクトします。要求されたオブジェクトがサーバから到着すると、クライアントに配信され、そのコピーがディスクキャッシュに格納されます。キャッシングの利点の1つは、様々なクライアントが同じオブジェクトを要求した場合に、これらのオブジェクトをハードディスクのキャッシュから提供できることです。これにより、クライアントはインターネットから取得する場合に比べてはるかに高速にデータを受信できます。また、ネットワークトラフィックも減少します。

Squidは、実際のキャッシングのみでなく、プロキシサーバの通信階層にまたがる負荷の分散、プロキシにアクセスする全クライアントの厳密なアクセス制御リストの定義、他のアプリケーションを使用した特定のWebページへのアクセスの許可または拒否、ユーザのアクセスパターンの調査を目的としたアクセス回数の多いWebサイトに関する統計の生成など、多様な機能を備えています。Squidは汎用プロキシではありません。通常は、HTTP接続のみのプロキシを行います。また、FTP、Gopher、SSLおよびWAISの各プロトコルをサポートしていますが、Real Audio、newsまたはビデオ会議など、他のインターネットプロトコルはサポートしていません。Squidは様々なキャッシュ間に通信を提供するUDPプロトコルのみをサポートしているため、他の多くのマルチメディアプログラムはサポートされません。

# 33.2 プロキシキャッシュに関する注意事項

## 33.2.1 Squidとセキュリティ

Squidをファイアウォールと併用し、プロキシキャッシュを使用して社内ネットワークを外部から保護することもできます。ファイアウォールは、Squidを除く外部サービスに対する全クライアントのアクセスを拒否します。すべてのWeb接続は、プロキシを使用して確立する必要があります。

ファイアウォール設定にDMZが含まれている場合、プロキシはこのDMZ内で動作する必要があります。この場合、DMZ内のすべてのコンピュータがログファイルを安全なネットワーク内のホストに送信することが重要です。「*透過型*」プロキシの実装の可能性については、項33.6.「透過型プロキシの設定」を参照してください。

### 33.2.2 複数のキャッシュ

プロキシ間でオブジェクトを交換できるように複数のプロキシを設定できます。これにより、システム全体の負荷を削減し、ローカルネットワーク内の既存のオブジェクトの検出率を高めることができます。また、キャッシュから兄弟キャッシュまたは親キャッシュにオブジェクト要求を転送できるように、キャッシュ階層を設定することも可能です。これにより、ローカルネットワーク内の他のキャッシュから、またはソースから直接、オブジェクトを取得できるようになります。

ネットワークトラフィック全体が増大することは望ましくないため、キャッシュ階層に適切なトポロジを選択することがきわめて重要です。大規模ネットワークの場合は、サブネットワークごとにプロキシサーバを設定して親プロキシに接続し、親プロキシはISPのプロキシキャッシュに接続すると有効です。

この通信はすべて、UDPプロトコルの最上位で実行されるICP (Internet cache protocol)により処理されます。キャッシュ間のデータ転送は、TCPベースのHTTP (hyper text transmission protocol)により処理されます。

どのサーバからオブジェクトを取得するのが最も適切であるかを検出するために、あるキャッシュからすべての兄弟プロキシにICPリクエストが送信されます。各兄弟プロキシは、オブジェクトが検出された場合はHITコード、検出されなかった場合はMISSを使用し、ICPレスポンスを介してリクエストに応答します。複数のHITレスポンスが検出された場合、プロキシサーバは、最も短時間で応答したキャッシュまたは最も近接するキャッシュなどのファクタに従ってダウンロード元のサーバを決定します。リクエストを満たすレスポンスが受信されなければ、リクエストは親キャッシュに送信されます。

**Tip**

ネットワーク上の様々なキャッシュ内でオブジェクトの重複を回避するために、CARP (Cache Array Routing Protocol)やHTCP (Hypertext Cache Protocol)など、他のICPプロトコルが使用されます。ネットワーク上で維持されるオブジェクトが多くなるほど、必要なオブジェクトを検出できる可能性が高くなります。

**Tip**

### 33.2.3 インターネットオブジェクトのキャッシュ

ネットワーク上で使用可能なオブジェクトがすべてスタティックであるとは限りません。動的に生成されるCGIページ、アクセス件数カウンタ、暗号化され

たSSLコンテンツドキュメントが多数存在します。この種のオブジェクトは、アクセスされるたびに変化するためキャッシュされません。

その他のオブジェクトについても、キャッシュにどのくらいの期間残しておくかという問題があります。これを決定するために、オブジェクトが取り得るさまざまな状態を定義し、キャッシュ内のすべてのオブジェクトに1つの状態を割り当てます。Webサーバとプロキシサーバは、これらのオブジェクトに“Last modified”や“Expires”などのヘッダおよび対応する日付を追加することで、オブジェクトの状態を検出します。その他、オブジェクトをキャッシュしないように指定するヘッダも使用されます。

ハードディスクの空き容量不足が原因で、通常、キャッシュ内のオブジェクトはLRU (Least Recently Used)などのアルゴリズムを使用して置換されます。これは、基本的には、長期間要求されていないオブジェクトがプロキシにより消去されることを意味します。

## 33.3 システム要件

最も重要なのは、システムにかかる最大負荷を判断することです。したがって、負荷のピークに注意する必要があります。ピーク時の負荷が1日の平均負荷の4倍を超えることもあるためです。疑わしい場合は、システム要件を多めに見積もることをお勧めします。これは、Squidの動作状態が処理能力の限界に近づくと、サービス品質が著しく低下する可能性があるためです。次の各項では、システム要件を重要度に従って説明します。

### 33.3.1 ハードディスク

速度はキャッシュ処理に重要な役割を果たすため、この要件には特に注意する必要があります。ハードディスクの場合、このパラメータは*ランダムシーク時間*と呼ばれ、ミリ秒単位で計測されます。Squidがハードディスクとの間で読み書きするデータブロックは比較的少数である傾向があるため、データのスループットよりもハードディスクのシーク時間の方が重要です。プロキシに使用する場合は、回転速度の高い(つまり読取り/書込みヘッドが必要な位置に迅速に移動する)ハードディスクを選択するのが適切です。システムを高速化するには、同時に多数のディスクを使用する方法や、ストライピングRAIDアレイを使用する方法があります。

### 33.3.2 ディスクキャッシュのサイズ

キャッシュ容量が小さいと、簡単にいっぱいになってしまい、要求頻度の低いオブジェクトが新規オブジェクトで置換されるため、HIT (要求された既存のオブジェクトの検出)の可能性は低くなります。逆に、キャッシュに1GBが使用可能で、ユーザが1日に10MB分しかアクセスしなければ、キャッシュがいっぱいになるまでに100日以上かかることになります。

必要なキャッシュサイズを判断する場合に最も簡単なのは、接続の最大転送速度を考慮することです。1Mbit/sの接続の場合、最大転送速度は125KB/sです。このトラフィックがすべてキャッシュに入ると、1時間で合計450MBとなり、このトラフィックがすべて8時間の営業時間帯にのみ発生すると仮定すれば、1日に3.6GBに達します。通常、接続が上限まで使用されることはないため、キャッシュで処理される合計データ量は約2GBと想定できます。このため、Squidで1日にブラウズされたデータをキャッシュに保持する例では、2GBのディスク容量が必要となります。

### 33.3.3 RAM

Squidに必要なメモリ容量(RAM)は、キャッシュ内のオブジェクト数に比例します。また、Squidでは、キャッシュオブジェクト参照と要求頻度の高いオブジェクトの検索を高速化するために、これらのデータがメインメモリに格納されます。ランダムアクセスメモリの方が、ハードディスクよりも高速です。

その他、Squidでは、処理された全IPアドレスの表、正確なドメインネームキャッシュ、最もアクセス頻度の高いオブジェクト、アクセス制御リスト、バッファなどのデータもメモリに保持する必要があります。

ディスクにスワップする必要があるとシステムパフォーマンスが大幅に低下するため、Squidプロセス用に十分なメモリを用意する必要があります。キャッシュメモリの管理には、cachemgr.cgiツールを使用できます。このツールの詳細については、項33.7. 「cachemgr.cgi」を参照してください。

### 33.3.4 CPU

Squidは、CPU集約型のプログラムではありません。プロセッサの負荷が増大するのは、キャッシュの内容がロードまたはチェックされる間のみです。マルチプロセッサマシンを使用しても、システムパフォーマンスは向上しません。効率を高めるには、高速ディスクまたは増設メモリを購入することをお薦めします。

## 33.4 Squidの起動

SquidはSUSE LINUXで事前に設定されているため、インストール直後に起動できます。スムーズに起動するように、インターネットおよび少なくとも1つのネームサーバにアクセスできるようにネットワークを設定してください。ダイナミックDNS設定でダイヤルアップ接続を使用すると、問題が発生する可能性があります。このような場合は、少なくともネームサーバを明確に入力してください。というのは、/etc/resolv.conf内でDNSサーバが検出されないとSquidが起動しないためです。

### 33.4.1 Squidの起動コマンドと停止コマンド

Squidを起動するには、root権限でコマンドラインに「rcsquid start」と入力します。初期起動時には、最初に/var/squid/cache内でディレクトリ構造を定義する必要があります。この操作は、起動スクリプト/etc/init.d/squidにより自動的に実行され、完了までに数秒ないし数分かかります。右側に緑で完了と表示されたら、Squidは正常にロードされています。ローカルシステム上でSquidの機能をテストするには、ブラウザでプロキシとして「localhost」、ポートとして「3128」を入力します。

ユーザ全員にSquidへのアクセスとSquidを介したインターネットへのアクセスを許可するには、設定ファイル/etc/squid/squid.conf内のエントリをhttp_access deny allからhttp_access allow allに変更します。ただし、その場合は、この操作によりSquidが完全に誰でもアクセス可能になることに注意してください。したがって、プロキシへのアクセスを制御するACLを定義します。この詳細については、項33.5.2. 「アクセス制御オプション」ファイルを参照してください。

設定ファイル/etc/squid/squid.confを変更した後、Squidで変更後の設定ファイルを再ロードする必要があります。それには、rcsquid reloadコマンドを使用します。または、「rcsquid restart」と入力してSquidを完全に再起動します。

プロキシが稼動しているかどうかを確認するには、rcsquid statusコマンドを使用します。Squidをシャットダウンするには、rcsquidstopコマンドを使用します。Squidは、クライアントへの接続が切断されてデータがディスクに書き込まれるまで最大30秒(/etc/squid/squid.confのshutdown_lifetimeオプション）待機するため、終了までに少し時間がかかることがあります。

**Warning**

**Squidの終了**

killまたはkillallを使用してSquidを終了すると、キャッシュが破損する可能性があります。Squidを再起動できるようにするには、破損したキャッシュを完全に削除する必要があります。

**Warning**

Squidが正常に起動しても短時間で停止する場合は、ネームサーバエントリに誤りがないかどうかと、/etc/resolv.confファイルが欠落していないかどうかをチェックしてください。起動エラーの原因は、/var/squid/logs/cache.logファイルに記録されます。システムのブート時にSquidを自動的にロードする必要がある場合は、YaSTランレベルエディタを使用してSquidを必要なランレベルで有効にしてください。項2.7.7. 「」を参照してください。

Squidをアンインストールしても、キャッシュ階層やログファイルは削除されません。これらを削除するには、/var/cache/squidディレクトリを手動で削除します。

### 33.4.2 ローカルDNSサーバ

サーバで独自ドメインを管理しない場合も、ローカルDNSサーバをセットアップすると有効です。ローカルDNSサーバは単にキャッシュ専用ネームサーバとして機能し、特に設定しなくてもルートネームサーバを介してDNSリクエストを解決できます(項24.2. 「ネームサーバBINDの起動」を参照)。ローカルDNSサーバを有効にする方法は、インターネット接続の設定時にダイナミックDNSを選択したかどうかによって異なります。

**ダイナミックDNS** 通常、ダイナミックDNSを使用すると、インターネット接続が確立されるときプロバイダによってDNSサーバが設定され、ローカルの/etc/resolv.confファイルが自動的に変更されます。この動作は、MODIFY_RESOLV_CONF_DYNAMICALLYシステム変数をYESに設定すると実行されます。このシステム変数をYaSTsysconfigエディタを使用してNOに設定してください(項7.8. 「YaST sysconfigエディタ」参照)。そして、/etc/resolv.confファイルに、ローカルのDNSサーバとして「localhost」、そのIPアドレスとして「127.0.0.1」を入力します。このようにすれば、Squidは常に、起動時にローカルのネームサーバを検出できます。

プロバイダのネームサーバにアクセスするには、/etc/named.conf設定ファイルに、サーバ名forwardersとそのIPアドレスを入力する必要

があります。ダイナミックDNSを使用すると、この動作を接続の確立時に自動的に実行できます。それには、sysconfig変数MODIFY_NAMED_CONF_DYNAMICALLYをYESに設定しておく必要があります。

**スタティックDNS** スタティックDNSでは、接続の確立時に、DNSに関する自動設定を実行しません。したがって、sysconfig変数を変更する必要はありませんが、上記のとおり、/etc/resolv.confファイルにローカルのDNSサーバ名を入力する必要があります。プロバイダのスタティックなネームサーバにアクセスするには、/etc/named.conf設定ファイルに、サーバ名forwardersとそのIPアドレスを手動で入力する必要があります。

**Tip**

**DNSとファイアウォール**

ただし、ファイアウォールを実行している場合は、DNSリクエストがファイアウォールを通過できることを確認してください。

## 33.5 設定ファイル/etc/squid/squid.conf

Squidのプロキシサーバ設定は、すべて/etc/squid/squid.confファイル内で行います。Squidを初めて起動する場合、このファイル内で設定を変更する必要はありませんが、外部クライアントは最初はアクセスを拒否されます。プロキシはlocalhostに使用できます。デフォルトポートは3128です。プリインストール済みの/etc/squid/squid.confには、オプションの詳細と多数の例が用意されています。ほぼすべてのエントリは(コメント行を示す) #記号で始まり、関連する指定が行末にあります。示されている値は、ほぼ常にデフォルト値に関係しているため、パラメータを実際に変更せずにコメント記号を削除しても、ほとんどの場合に影響はありません。サンプルはそのまま残し、変更したパラメータと共にオプションを次の行に挿入することをお勧めします。これにより、デフォルト値と変更内容が一目でわかります。

**Tip**

**更新後の設定ファイルの変更について**

Squidを旧バージョンから更新した場合は、新規の/etc/squid/squid.confを編集し、旧バージョンのファイルで行った変更のみを適用することをお勧めします。旧バージョンのsquid.confファイルを実装すると、オプションが変更されたり新たな変更が加えられているために、設定が機能しなくなる危険性があります。

**Tip**

### 33.5.1 一般設定オプション(選択)

**http_port 3128** これは、Squidがクライアントリクエストをリスンするポートです。デフォルトポートは3128ですが、8080も一般的です。必要な場合は、複数のポート番号を空白で区切って指定します。

**cache_peer** ⟨***hostname***⟩ ⟨***type***⟩ ⟨***proxy-port***⟩ ⟨***icp-port***⟩
ここでは、たとえばISPのプロキシを使用する場合に、親プロキシを入力します。⟨*hostname*⟩には、使用するプロキシの名前とIPアドレスを入力し、⟨*type*⟩には親プロキシを入力します。⟨*proxy-port*⟩には、ポート番号(通常は8080)を入力します。親プロキシのユーザもこのポート番号をブラウザに設定します。⟨*icp-port*⟩は、7に設定するか、親のICPポートが不明で、その使用がプロバイダに無関係な場合は0に設定します。また、ICPプロトコルの使用を禁止するには、ポート番号に続けてdefaultおよびno-queryを指定する必要があります。このように指定すると、Squidはプロバイダのプロキシに関する限り通常のブラウザのように動作します。

**cache_mem 8 MB** このエントリは、Squidでキャッシュに使用できるメモリ容量を定義します。デフォルトは8MBです。

**cache_dir ufs /var/cache/squid/ 100 16 256**
*cache_dir*エントリは、すべてのオブジェクトが格納されるディスク上のディレクトリを定義します。末尾の数値は、使用される最大ディスク領域(単位MB)と第1レベルと第2レベルのディレクトリ数を示します。ufsパラメータは残しておく必要があります。デフォルトでは、/var/cache/squidディレクトリに100MBのディスク領域を使用して16個のサブディレクトリが作成され、各サブディレクトリにそれぞれ256個以上のサブディレクトリが含まれます。使用するディスク領域を

指定するときには、予備のディスク領域を十分に残しておきます。ここでは、使用可能ディスク領域の50～80パーセントが最も有効です。ディレクトリが多すぎるとパフォーマンスが低下する可能性があるため、ディレクトリに関する最後の2つの数値を増やす場合は注意してください。複数のディスクでキャッシュを共有する場合は、複数の*cache_dir*行を入力します。

**cache_access_log /var/log/squid/access.log**
メッセージログのパスです。

**cache_log /var/log/squid/cache.log** メッセージログのパスです。

**cache_store_log /var/log/squid/store.log**
メッセージログのパスです。

この3つのエントリには、Squidのすべてのアクションが記録されるパスを指定します。通常、ここでは何も変更しません。Squidの使用負荷が大きい場合は、キャッシュとログファイルを複数のディスクに分散すると有効な場合があります。

**emulate_httpd_log off** このエントリを*on*に設定すると、読込み可能なログファイルが生成されます。ただし、一部の評価プログラムではこの形式のログファイルを解釈できません。

**client_netmask 255.255.255.255** このエントリでは、ログファイル内のIPアドレスをマスクして、クライアントの識別情報を隠します。ここで「`255.255.255.0`」と入力すると、IPアドレスの最終桁はゼロに設定されます。

**ftp_user Squid@** このエントリでは、Squidで匿名FTPログインに使用する必要のあるパスワードを設定します。一部のFTPサーバには電子メールアドレスの妥当性がチェックされるため、ここでは有効な電子メールアドレスを指定できます。

**cache_mgr webmaster** Squidが予期せずにクラッシュした場合のメッセージ送信先となる電子メールアドレスを指定します。デフォルトは*webmaster*です。

**logfile_rotate 0** `squid -k rotate`を実行すると、Squidは保護されたログファイルを循環利用することができます。このプロセス中にファイルに番号が割り当てられ、指定した値に達すると最も古いファイルが上書きされます。ではログファイルのアーカイブと削除が設定ファイル`/etc/logrotate/squid`内で検出された自動実行ジョブにより実行されるため、デフォルト値は`0`です。

**append_domain <domain>** *append_domain*には、未指定の場合に自動的に追加されるドメインを指定します。通常、ブラウザに「*www*」と入力して独自Webサーバにアクセスできるように、このエントリには独自ドメインを入力します。

**forwarded_for on** このエントリを*off*に設定すると、SquidではHTTPリクエストからクライアントのIPアドレスとシステム名が削除されます。

**negative_ttl 5 minutes; negative_dns_ttl 5 minutes**
通常、これらの値を変更する必要はありません。ただし、ダイヤルアップ接続を使用する場合は、インターネットが一時的にアクセス不能になる場合があります。Squidは、失敗したリクエストを記録してから新規リクエストの発行を拒絶しますが、インターネット接続は再確立されています。このような場合は、*minutes*を*seconds*に変更し、ブラウザの*更新*機能を使用すると、数秒後にダイヤルアッププロセスが再開されます。

**never_direct allow ⟨*acl_name*⟩** Squidがインターネットからリクエストを直接取り込むのを防ぐには、上記のコマンドを使用して他のプロキシに強制的に接続します。このプロキシは、あらかじめ*cache_peer*に入力しておく必要があります。⟨*acl_name*⟩として`all`を指定すると、すべてのリクエストは「親」に直接転送されます。たとえば、プロキシの使用を奨励しているプロバイダや、ファイアウォールによるインターネットへのダイレクトアクセスを拒否しているプロバイダを使用している場合は、この設定が必要な場合があります。

### 33.5.2 アクセス制御オプション

Squidには、プロキシへのアクセスを制御する詳細システムが用意されています。ACLを実装することで、このシステムを簡単かつ包括的に設定できます。そのためには、順次処理されるルールを持ったリストが必要です。ACLは定義しなければ使用できません。*all*や*localhost*などのデフォルトACLがいくつか用意されています。ただし、ACLを定義しただけで、実際に適用されるわけではありません。実際に適用するには、*http_access*ルールも共に定義する必要があります。

**acl <acl_name> <type> <data>** ACLの定義には、3つ以上の指定が必要です。名前<*acl_name*>は任意に選択できます。<*type*>は、`/etc/squid/squid.conf`ファイルの*ACCESS CONTROLS*セクションにある多数のオプションから選択できます。<*data*>の指定は個々のACLタイプに応じて異なり、ホスト名、IPアドレスまたはURLを使用するなど、ファイルから読み込むこともできます。次に単純な例を示します。

```
acl mysurfers srcdomain .my-domain.com
acl teachers src 192.168.1.0/255.255.255.0
acl students src 192.168.7.0-192.168.9.0/255.255.255.0
acl lunch time MTWHF 12:00-15:00
```

**http_access allow <acl_name>** *http_access*では、プロキシの使用を許可されるユーザと、インターネット上でどのユーザが何にアクセスできるかを定義します。それには、ACLを指定する必要があります。*localhost*および*all*の定義はすでに前述しており、この2つのACLでは*deny*または*allow*を介してアクセスを拒否または許可できます。多数の*http_access*エントリを含むリストを作成できます。各エントリは上から下へと処理され、発生順に従って個々のURLへのアクセスが許可または拒否されます。最後のエントリは、常に*http_access deny all*にする必要があります。次の例では、*localhost*はすべてに自由にアクセスできますが、他のホストはいずれもアクセスを完全に拒否されます。

```
http_access allow localhost
http_access deny all
```

また、このルールの使用を示す次の例では、グループteachersは常にインターネットへのアクセス権を持ちます。グループstudentsは月曜日から金曜日のランチタイム中にのみアクセス権を取得します。

```
http_access deny localhost
http_access allow teachers
http_access allow students lunch time
http_access deny all
```

*http_access*エントリを含むリストは、読みやすいように/etc/squid/squid.confファイルの指定の位置にのみ入力してください。つまり、次の2つの間に入力します。

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS



http_access deny all
```

**redirect_program /usr/bin/squidGuard**
このオプションでは、squidGuardなど、望ましくないURLをブロックできるリダイレクタを指定します。インターネットアクセスは、プロキシ認証と適切なACLを使用してユーザグループごとに個別に制御できます。 squidGuardは別途にインストールして設定できるパッケージです。

**auth_param basic program /usr/sbin/pam_auth**

ユーザのプロキシ認証が必要な場合は、pam_authなどの対応するプログラムを設定します。ユーザがpam_authに初めてアクセスすると、ログインウィンドウが表示され、ユーザ名とパスワードを入力することになります。また、有効なログインを持つクライアント以外はインターネットを使用できないように、ACLも必要です。

```
acl password proxy_auth REQUIRED

http_access allow password
http_access deny all
```

*proxy_auth*の後の*REQUIRED*は、許可されるユーザ名のリストまたはそのリストへのパスで置き換えることができます。

**ident_lookup_access allow <acl_name>**

ここでは、ACLで定義されたクライアントすべてについてidentリクエストを実行させ、各ユーザの識別情報を検索させます。*<acl_name>*に*all*を適用すると、すべてのクライアントに対して有効になります。また、すべてのクライアントでidentデーモンを実行する必要があります。Linuxの場合、そのためにはpidentdパッケージをインストールします。Windowsの場合は、インターネットからダウンロードできるフリーソフトウェアが提供されています。identが正常に検索されたクライアントのみが許可されるように、対応するACLをここで定義します。

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

この場合も、*REQUIRED*を許可されるユーザ名のリストで置き換えることができます。*ident*を使用すると、その検索がリクエストごとに繰り返されるため、アクセス速度が少し低下する場合があります。

## 33.6 透過型プロキシの設定

プロキシサーバを使用する場合の通常の操作は次のとおりです。まず、Webブラウザからプロキシサーバの特定のポートにリクエストが送信され、プロキシは要求されたオブジェクトがキャッシュ内にあるかどうかに関係なく提供します。ネットワークで操作する場合には、次のような状況が発生することがあります。

- セキュリティ上の理由から、すべてのクライアントがインターネットでのナビゲーションにはプロキシを使用することを推奨される場合。
- すべてのクライアントが、認識するかどうかに関係なくプロキシを使用する必要がある場合。
- ネットワーク上でプロキシが移動しても、既存のクライアントは古い設定を保持する必要がある場合。

いずれの場合も、透過型プロキシを使用できます。原則はきわめて簡単で、プロキシはWebブラウザのリクエストを捕捉して応答するため、Webブラウザは要求したページを出所を認識せずに受信します。透過型プロキシと呼ばれるのは、このプロセス全体が透過的に実行されるためです。

### 33.6.1 カーネル設定

最初に、プロキシサーバのカーネルで透過型プロキシがサポートされているかどうかを確認します。SUSE LINUX付属のカーネルは、透過型プロキシをサポートされるように設定されています。サポートされていない場合は、これらのオプションをカーネルに追加して再コンパイルします。詳細については、章 9. Linuxカーネルを参照してください。

### 33.6.2 /etc/squid/squid.conf内の設定オプション

`/etc/squid/squid.conf`ファイル内で透過型プロキシの起動と実行に使用できるオプションは、次のとおりです。

- `httpd_accel_host` virtual
- `httpd_accel_port` 80
  実HTTPサーバが動作するポート番号
- `httpd_accel_with_proxy` on
- `httpd_accel_uses_host_header` on

### 33.6.3 SuSEfirewall2を使用したファイアウォール設定

ファイアウォールを介して受信するリクエストをすべて、Squidポートへのポート転送ルールに従ってリダイレクトします。そのためには、同梱のツールであるSuSEfirewall2を使用します。このツールの設定ファイルは/etc/sysconfig/SuSEfirewall2にあります。この設定ファイルは、適切なエントリで構成されています。透過型プロキシのみを設定する場合にも、次に示すように一部のファイアウォールオプションは設定する必要があります。

- インターネットを指すデバイス: FW_DEV_EXT="eth1"
- ネットワークを指すデバイス: FW_DEV_INT="eth0"

インターネットなど、信頼されない(外部)ネットワークからアクセスが許可される、ファイアウォール上のポートとサービスを定義します(/etc/servicesを参照)。次の例では、外部に対してWebサービスのみが提供されます。

```
FW_SERVICES_EXT_TCP="www"
```

安全な(内部)ネットワークからのアクセスが許可される、ファイアウォール上のポートとサービス(TCPサービスとUDPサービスの両方)を定義します(/etc/servicesを参照)。

```
FW_SERVICES_INT_TCP="domain www 3128" FW_SERVICES_INT_UDP="domain"
```

この例では、WebサービスとSquid (デフォルトポートは3128)へのアクセスが許可されます。“domain”サービスはDNS (ドメインネームサービス)を意味します。このサービスは一般に使用されます。一般に公開しない場合は、単に上記のエントリから削除して次のオプションをnoに設定します。

```
FW_SERVICE_DNS="yes"
```

最も重要なのは15番目のオプションです。

*Example 33.1: ファイアウォール設定: オプション15*

```
#
# 15.)
# Which accesses to services should be redirected to a local port
# on the firewall machine?
#
# This can be used to force all internal users to surf via your
# Squid proxy, or transparently redirect incoming web traffic to
# a secure web server.
#
# Choice: leave empty or use the following explained syntax of
# redirecting rules, separated with spaces.
# A redirecting rule consists of 1) source IP/net,
# 2) destination IP/net, 3) original destination port and
# 4) local port to redirect the traffic to, separated by a colon,
# e.g. "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"
#
```

上記のコメントは、次の構文を示しています。最初に、プロキシファイアウォールにアクセスする内部ネットワークのIPアドレスとネットマスクを入力します。次に、これらのクライアントからのリクエストの送信先となるIPアドレスとネットマスクを入力します。Webブラウザの場合は、ネットワーク`0/0`を指定します。これは、「あらゆる場所」を意味するワイルドカードです。その後、これらのリクエストの送信先となるオリジナルポートを入力し、最後に全リクエストのリダイレクト先となるポートを入力します。SquidはHTTP以外のプロトコルもサポートしているため、HTTP以外のポートからのリクエスト(FTP (ポート21)、HTTPSまたはSSL (ポート443)など)もリダイレクトされます。この例では、Webサービス(ポート`80`)がプロキシポート(ポート`3128`)にリダイレクトされます。他にも追加するネットワークやサービスがある場合は、対応するエントリに空白1個で区切って指定する必要があります。

```
FW_REDIRECT_TCP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
FW_REDIRECT_UDP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
```

ファイアウォールとそれを使用した新規設定を開始するには、`/etc/sysconfig/SuSEfirewall2`ファイル内のエントリを変更します。エントリ`START_FW`を`"yes"`に設定する必要があります。

項33.4. 「Squidの起動」の説明に従ってSquidを起動します。すべてが正常に動作しているかどうかをチェックするには、`/var/log/squid/access.log`内でSquidのログを調べます。

すべてのポートが正しく設定されていることを確認するには、ネットワーク外の任意のコンピュータからマシン上のポート検索を実行します。Webサービス(ポート80)のみがオープンしている必要があります。nmapコマンドを使用してポートを検索する場合の構文は、`nmap -O IP_address`です。

## 33.7 cachemgr.cgi

キャッシュマネージャ(cachemgr.cgi)は、実行中のSquidプロセスによるメモリ使用状況に関する統計を表示するCGIユーティリティです。また、キャッシュを管理し、サーバのロギングなしで統計を表示できる便利な手段でもあります。

### 33.7.1 設定

最初に、システムでWebサーバを稼働させる必要があります。Apacheがすでに稼働しているかどうかをチェックするには、「`root`として`rcapachestatus`」コマンドを入力します。次のようなメッセージが表示される場合は、マシンでApacheが実行されています。

```
Checking for service httpd: OK
Server uptime: 1 day 18 hours 29 minutes 39 seconds
```

それ以外の場合は、「`rcapachestart`」コマンドを入力して、SUSE LINUXのデフォルト設定でApacheを起動します。最後に、`cachemgr.cgi`ファイルをApacheのディレクトリ`cgi-bin`にコピーします。

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi /srv/www/cgi-bin/
```

### 33.7.2 /etc/squid/squid.conf内のキャッシュマネージャACL

キャッシュマネージャの場合は、オリジナルファイル内で次のようなデフォルト設定が必要です。キャッシュマネージャはcache_objectプロトコルを用いてSquidと通信するため、最初のACLが最も重要です。

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

次のルールを指定する必要があります。

```
http_access allow manager localhost
http_access deny manager
```

次のルールは、WebサーバとSquidが同じマシンで実行されている場合を想定しています。キャッシュマネージャとSquidとの通信が他のコンピュータ上のWebサーバで開始される場合は、例 33.2. 「アクセスルール」に示すACLを追加します。

*Example 33.2: アクセスルール*

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # webserver IP
```

次に、例 33.3. 「アクセスルール」に示すルールを追加します。

*Example 33.3: アクセスルール*

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

キャッシュのリモートクローズやキャッシュ詳細情報の表示など、より多数のオプションにアクセスする場合は、マネージャのパスワードを設定します。そのためには、マネージャ用のパスワードと表示するオプションのリストを指定してエントリcachemgr_passwdを設定します。このリストは、/etc/squid/squid.confにエントリのコメントの一部として表示されます。

設定ファイルを変更するたびにSquidを再起動してください。それには、rcsquid reloadコマンドを使用します。

### 33.7.3 統計情報の表示

対応するWebサイト— http://webserver.example.org/cgi-bin/cachemgr.cgiに移動します。［'続行'］をクリックして様々な統計情報をブラウズします。キャッシュマネージャに表示される各エントリの詳細は、http://www.squid-cache.org/Doc/FAQ/FAQ-9.htmlにあるSquidのFAQを参照してください。

## 33.8 squidGuard

このセクションでは、squidGuardの詳細な設定については説明しません。ごく基本的な設定のみを紹介し、squidGuardの使用法についていくつか助言するに留めます。詳細な設定については、squidGuardのWebサイトhttp://www.squidguard.orgを参照してください。

squidGuardは、Squid用の無償(GPL)で柔軟で高速なフィルタ、リダイレクタおよびアクセスコントローラプラグインです。このプラグインを使用すると、Squidキャッシュ上のユーザグループごとに様々な制限を指定して、複数のアクセスルールを定義できます。 squidGuardでは、Squidの標準リダイレクタインタフェースを使用します。

squidGuardには、次の用途があります。

- 一部のユーザによるWebアクセスを、許可されているか既知のWebサーバまたはURLのリストに限定します。
- リストまたはブラックリストに含まれたWebサーバまたはURLへの、一部のユーザによるアクセスをブロックします。
- 正規表現または語のリストと一致するURLへの、一部のユーザによるアクセスをブロックします。
- ブロックしたURLを「インテリジェント」CGIベースの情報ページにリダイレクトします。
- 未登録ユーザを登録フォームにリダイレクトします。
- バナーを空のGIFにリダイレクトします。
- 時刻、曜日、日付などに基づいて異なるアクセスルールを使用します。
- ユーザグループごとに異なるルールを使用します。

squidGuardとSquidは、以下の用途には使用できません。

- ドキュメント内のテキストの編集、フィルタ処理または検閲。
- JavaScriptやVBscriptなど、HTML埋込みスクリプト言語の編集、フィルタ処理または検閲。

squidGuardを使用するにはまず、インストールします。最小限の設定ファイルとして/etc/squidguard.confを設定します。http://www.squidguard.org/config/に設定例が用意されています。最小限の設定で正常に動作したら、より複雑な設定を試してみてください。

次に、クライアントがブラックリストに含まれるWebサイトを要求した場合にSquidをリダイレクトするために、ダミーの「アクセス拒否」ページまたは複雑度の異なるCGIページを作成します。Apacheを使用することをお薦めします。

ここで、squidGuardを使用するようにSquidを設定します。/etc/squid.confファイル内の次のエントリを使用してください。

```
redirect_program /usr/bin/squidGuard
```

もう1つのオプションredirect_childrenでは、マシン上で実行する「リダイレクト」(この場合はsquidGuard)プロセスの数を設定します。5,900のドメインと7,880のURL (合計13,780)であれば、500MHz Pentium上で10秒以内に100,000のリクエストを処理できます。したがって、プロセスを5つ以上設定しないようにお勧めします。これは、5つ以上設定すると、それらのプロセスの割り当てに大量のメモリが消費されるためです。

```
redirect_children 4
```

最後に、rcsquidreloadを実行し、Squidに新規設定をロードさせます。ここで、ブラウザで設定をテストします。

## 33.9 Calamarisを使用したキャッシュレポート生成

Calamarisは、ASCIIまたはHTML形式でキャッシュアクティビティレポートを生成するためのPerlスクリプトです。このスクリプトはネイティブのSquidアクセスログファイルを処理します。Calamarisのホームページはhttp://Calamaris.Cord.de/にあります。このプログラムの使用方法はきわめて簡単です。

rootとしてログインし、「cat access.log.files | calamaris ⟨*options*⟩ > reportfile」と入力します。複数のログファイルをパイプする場合は、各ログファイルを古いものから時系列順に指定する必要があります。このプログラムには、次のようなオプションがあります。

**-a** 使用可能な全レポートを出力

**-w** HTMLレポートとして出力

**-l** レポートヘッダにメッセージまたはロゴを挿入

各種オプションの詳細については、「man calamaris」と入力してプログラムのマニュアルページで参照できます。

典型的な例を次に示します。

```
cat access.log.2 access.log.1 access.log | calamaris -a -w \
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

このコマンドでは、レポートがWebサーバのディレクトリに生成されます。レポートを表示するにはApacheが必要です。

もう1つの強力なキャッシュレポート生成ツールとしてSARG (Squid Analysis Report Generator)があります。詳細については、http://web.onda.com.br/orso/を参照してください。

## 33.10 関連資料

http://www.squid-cache.org/にあるSquidのホームページにアクセスしてください。ここには『Squid User Guide』が置かれており、Squidに関する広範囲なFAQ集もあります。

透過型プロキシの使用方法に関する簡潔な情報は、/usr/share/doc/howto/en/txt/TransparentProxy.gzにhowtoenとして含まれています。また、squid-users@squid-cache.orgで、Squidに関するメーリングリストに登録できます。このアーカイブはhttp://www.squid-cache.org/mail-archive/squid-users/にあります。

# Part IV

# アドミニストレーション

# 34

Linuxのセキュリティ

# Linuxのセキュリティ

マスカレードとファイアウォールを使用すると、データフローとデータの送受信を確実に管理できるようになります。SSH (Secure Shell) を使用すると、暗号化された接続を介してリモートホストにログインできます。ファイルまたはパーティション全体を暗号化すれば、第三者によってシステムに侵入された場合でも、データを保護できます。最後のセクションでは、こうした純粋に技術的な側面から離れて、Linuxネットワークのセキュリティ面全般について説明します。

# 34.1 マスカレードとファイアウォール

ネットワーク環境でLinuxを使用する場合は常に、ネットワークパケットを操作するカーネル機能を使用して内部ネットワークと外部ネットワークを隔離できます。Linuxのnetfilterフレームワークは、複数のネットワークを隔離する効果的なファイアウォールを構築する手段を提供します。ルールセットを定義する汎用的なテーブル構造体であるiptablesを使用すれば、ネットワークインタフェースを通すパケットを詳細に制御することが可能です。このようなパケットフィルタは、SuSEfirewall2および対応するYaSTモジュールを使用して簡単にセットアップできます。

## 34.1.1 iptablesによるパケットフィルタリング

netfilterコンポーネントおよびiptablesコンポーネントは、ネットワークアドレス変換(NAT)に加え、ネットワークパケットのフィルタリングと操作の機能を備えています。フィルタ条件およびそれに関連付けられたアクションはルールセットとして格納され、受信したネットワークパケットに対して1つずつ個別に照合されます。使用されるフィルタ条件とアクションのセットはテーブルに格納されます。これらのテーブルおよびルールセットに変更を加えるには、`iptables`コマンドを使用します。

Linuxカーネルは、以下の3つのテーブルを管理します。各テーブルは、パケットフィルタの特定の機能カテゴリに対応しています。

**filter** このテーブルは、狭い意味での「パケットフィルタリング」メカニズムを実装するもので、フィルタルールの大半を含んでいます。たとえば、パケットを通すか(ACCEPT)破棄するか(DROP)を判定します。

**nat** このテーブルは、パケットの送信元アドレスと宛先アドレスに対する変更内容を定義します。これらの機能を使用して、「マスカレード」を実装できます。マスカレードは、プライベートネットワークとインターネットをリンクするNATの一種です。

**mangle** このテーブルのルールを使用して、IPヘッダ内の値(サービスタイプなど)を操作できます。

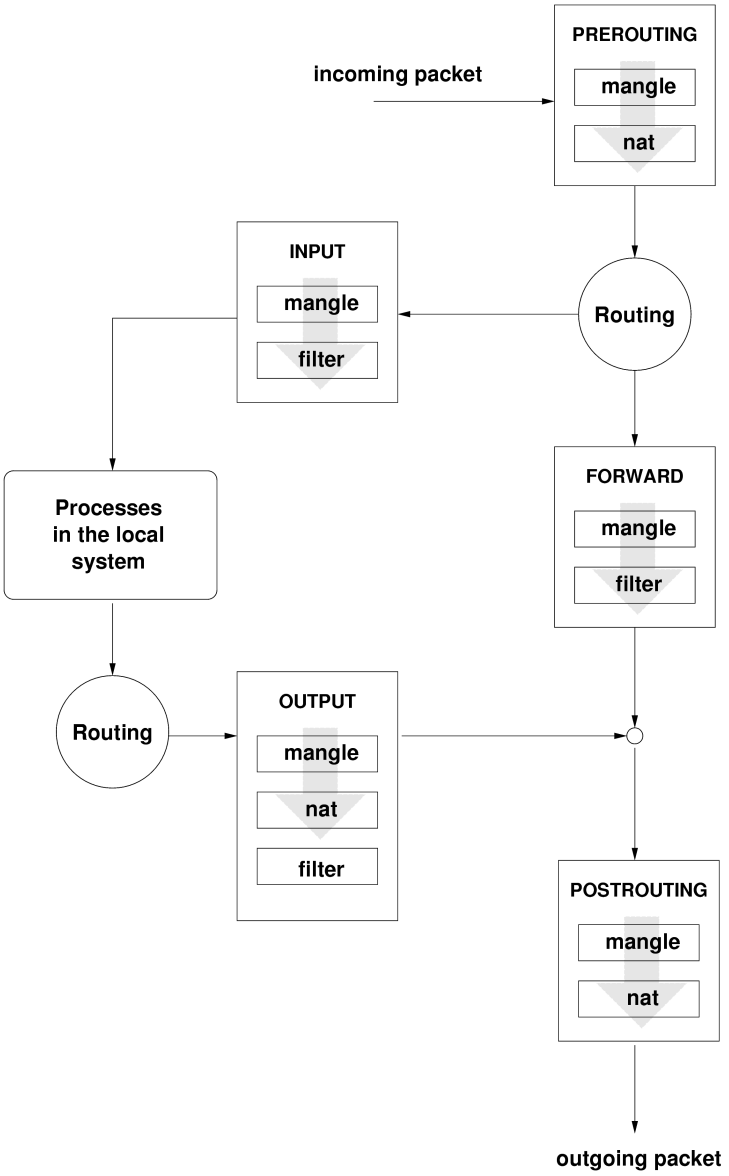


*Figure 34.1: iptables:パケットが取り得る経路*

上記のテーブルには、パケットと照合される次のような複数の事前定義ルールセットが含まれています。

**PREROUTING** このルールセットは、着信パケットに適用されます。

**INPUT** このルールセットは、システムの内部プロセス宛てのパケットに適用されます。

**FORWARD** このルールセットは、システムを通過するだけのパケットに適用されます。

**OUTPUT** このルールセットは、このシステム自身が送信元であるパケットに適用されます。

**POSTROUTING** このルールセットは、すべての発信パケットに適用されます。

あるシステムにおけるネットワークパケットの伝送経路を図 34.1.「iptables:パケットが取り得る経路」に示します。簡略化するために、この図ではテーブルをルールセットの一部として示してありますが、実際にはこれらのルールセットはテーブル自体に格納されています。

最も単純なケースとして、システム宛の着信パケットがeth0インタフェースに届いた場合を考えてみます。このパケットはまずmangleテーブルのPREROUTINGルールセットと照合され、次にnatテーブルのPREROUTINGルールセットと照合されます。パケットのルーティングに関する次のステップでは、パケットの実際の宛先がシステム自身のプロセスであることが確認されます。mangleテーブルおよびfilterテーブルのINPUTルールセットを経た後、このパケットは、filterテーブルのルールに実際に適合していれば、最終的に宛先に届きます。

### 34.1.2 マスカレードの基礎知識

マスカレードは、Linux固有のNAT (ネットワークアドレス変換)です。マスカレードを使用すると、小規模LAN (ホストがプライベート範囲のIPアドレスを使用するネットワーク—項22.1.2.「ネットマスクとルーティング」を参照)をインターネット(パブリックIPアドレスを使用するネットワーク)に接続することができます。このLANのホストをインターネットに接続するためには、プライベートアドレスをパブリックアドレスに変換する必要があります。この変換処理は、LANとインターネット間のゲートウェイとして動作するルータで行います。ルータの基本原理は単純です。ルータとは、複数のネットワークイ

ンタフェース(通常、ネットワークカードおよびそれとは別のインターネット接続用インタフェース)を備えたネットワーク装置です。インターネット接続用インタフェースは外部に接続し、その他のインタフェースはLAN上のホストに接続します。ルータのネットワークカード(eth0など)に接続されているローカルネットワーク内のホストは、ローカルネットワーク以外の宛先を持つすべてのパケットをデフォルトゲートウェイ、つまりルータに送信します。

**Important**

**正しいネットワークマスクの使用**

ネットワークを設定する際は、すべてのローカルホストに同じブロードキャストアドレスとネットマスクを設定する必要があります。そうしないと、パケットが正しくルーティングされず、ネットワークが正常に機能しません。

**Important**

前述のように、LAN上のホストがインターネット上のアドレス宛にパケットを送信すると、そのパケットは常にデフォルトルータに送信されます。しかし、そのためには、これらのパケットを転送できるようにルータを設定しておく必要があります。セキュリティ上の理由から、SUSE LINUXのインストール時のデフォルト設定では、この転送処理が有効になっていません。有効にするには、/etc/sysconfig/sysctlファイルのIP_FORWARD変数をIP_FORWARD=yesに設定します。

宛先ホストからは、ルータは参照できますが、内部ネットワーク内の送信元ホストに関する情報は一切分かりません。この技術がマスカレード(masquerading:「変装」の意)と呼ばれているのは、このためです。アドレス変換が行われているため、あらゆる応答パケットはまずルータに届きます。ルータはこれらの着信パケットを識別し、宛先アドレスを変換して、ローカルネットワーク内の正しいホストにパケットを転送します。

着信トラフィックのルーティングはマスカレードテーブルによって決まるため、外部から内部ホストへの接続を開く方法はありません。テーブルには、そのような接続に関するエントリがありません。また、確立済みの接続に対してはテーブルでステータスエントリが割り当てられるため、そのエントリは他の接続では使用されません。

このため、マスカレードを使用すると、ICQ、cucme、IRC (DCC、CTCP)、FTP (PORTモード)などいくつかのアプリケーションプロトコルで問題が発生する可能性があります。標準的なFTPプログラムであるNetscapeは、PASVモードを使用しています。PASVモードを使用すれば、パケットフィルタとマスカレードに関する問題が発生する可能性はかなり低くなります。

### 34.1.3 ファイアウォールの基礎知識

「ファイアウォール」は、ネットワーク間のリンクを提供、管理し、ネットワーク間のデータフローを制御するメカニズムを表す用語として、おそらくもっとも広く知られています。ただし、厳密にいうと、このセクションで説明するメカニズムは「パケットフィルタ」と呼ばれるものです。パケットフィルタは、プロトコル、ポート、IPアドレスなどに関する一定の条件に従ってデータフローを規制します。これにより、アドレスに応じて内部ネットワークに到達しないように定められているパケットが、ブロックされます。たとえば、社内のWebサーバを外部に公開するには、対応するポートを明示的に開きます。ただし、パケットフィルタは、社内のWebサーバ宛てのパケットなど、正当なアドレスを持つパケットの内容はスキャンしません。たとえば、着信パケットがWebサーバ上のCGIプログラムの破壊を目的としたものである場合でも、パケットフィルタはそれをそのまま通してしまいます。

より効果的な、しかしより複雑なメカニズムとして、いくつかのタイプのシステムを組み合わせる方法があります。たとえば、パケットフィルタと、プロキシと呼ばれるアプリケーションゲートウェイを連携動作させます。この場合、パケットフィルタは、無効なポートへのパケットをすべて拒否し、アプリケーションゲートウェイ宛てのパケットのみを受け入れます。このゲートウェイ、つまりプロキシは、サーバの実際のクライアントであるかのように振る舞います。ある意味で、このようなプロキシは、アプリケーションによって使用されるプロトコルレベルのマスカレードホストと見なすことができます。プロキシの例としては、HTTPプロキシサーバのSquidがあります。Squidを使用するには、プロキシ経由で通信するようにブラウザを設定する必要があります。要求したHTTPページはまずプロキシのキャッシュ内で検索され、キャッシュに見つからなかったページのみがプロキシによってインターネットから取得されます。別の例としては、FTPプロトコルのプロキシサーバであるSUSE proxy-suite (`proxy-suite`)があります。

次のセクションでは、SUSE LINUXに付属のパケットフィルタについて説明します。パケットフィルタとファイアウォールに関するより詳細な説明については、`howto`パッケージに含まれている『Firewall HOWTO』を参照してください。このパッケージがインストールされていれば、`less /usr/share/doc/howto/en/Firewall-HOWTO.gz`で『Firewall HOWTO』を参照できます。

### 34.1.4 SuSEfirewall2

SuSEfirewall2は、`/etc/sysconfig/SuSEfirewall2`から変数を読み取って一連のiptablesルールを生成するスクリプトです。このスクリプトは、次に

示す3つのセキュリティゾーンを定義します(ただし、以降のサンプル設定では1番目と2番目のセキュリティゾーンについてのみ考察します)。

**外部ゾーン** 外部ネットワークで何が発生しているかを制御できないことを考えれば、ホストを外部ネットワークから保護する必要があることがわかります。外部ネットワークはほとんどの場合インターネットですが、WLANなどそれ以外の安全でないネットワークであることもあります。

**内部ゾーン** これはプライベートネットワークを表します。ほとんどの場合はLANになります。内部ネットワーク内のホストがプライベート範囲のIPアドレス(項22.1.2. 「ネットマスクとルーティング」を参照)を使用している場合、ネットワークアドレス変換(NAT)を有効にして内部ネットワークのホストが外部ネットワークにアクセスできるようにします。

**非武装地帯(DMZ)** このゾーンのホストには外部ネットワークと内部ネットワークの両方からアクセスできますが、このゾーンのホストは自身では内部ネットワークにアクセスできません。DMZ内のシステムは内部ネットワークから隔離されるため、内部ネットワークの周りに追加の防衛線を設けたい場合にこのゾーンを設定します。

フィルタリングルールセットで明示的に許可されていないあらゆる種類のネットワークトラフィックは、iptablesによって抑止されます。したがって、着信トラフィックを持つそれぞれのインタフェースは、3つのゾーンのいずれかに配置する必要があります。各ゾーンに対して、許可するサービスやプロトコルを定義します。ルールセットは、外部ホストから送信されたパケットにのみ適用されます。ローカルに生成されたパケットは、ファイアウォールによって捕捉されません。

設定は、YaSTで行うことができます(項34.1.4. 「YaSTによる設定」を参照)。または、ファイル`/etc/sysconfig/SuSEfirewall2`に手動で設定することもできます。このファイルには、詳しい注釈が付けられています。また、さまざまな設定例が`/usr/share/doc/SuSEfirewall2/EXAMPLES`に格納されています。

### YaSTによる設定

**Important**

**自動ファイアウォール設定**

YaSTは、すべての設定済みインタフェース上で自動的にファイアウォールを起動します。システム上でサーバが設定されており有効になっていれば、YaSTは、サーバ設定モジュールの［'ファイアウォールで開いているポート'］オプションまたは［'Open Ports on Selected Interface in Firewall(選択したインタフェースでファイアウォールを開く)'］オプションを使用して、生成されたファイアウォール設定に自動的に変更を加えます。サーバモジュールの一部のダイアログでは、［'ファイアウォールの詳細'］ボタンをクリックすると、追加のサービスとポートを有効にできます。YaSTのファイアウォール設定モジュールは、ファイアウォールを有効または無効にする作業、あるいは個別に設定する作業に使用できます。

**Important**

グラフィカル設定用のYaSTダイアログには、YaSTコントロールセンターからアクセスできます。'セキュリティとユーザ' → 'ファイアウォール'を選択してください。設定は7つのセクションに分かれており、画面左側のツリー構造で各セクションに直接ジャンプすることができます。

**起動** このダイアログでは起動動作を設定します。デフォルトの設定では、SuSEfirewall2は、新しくインストールされたシステム上で自動的に起動します。このダイアログで、ファイアウォールを起動または停止することもできます。現在のファイアウォール設定をテストするには、［'設定を保存してファイアウォールを今すぐ再起動する'］をクリックします。

**［インタフェース］** ここには、認識されているすべてのネットワークインタフェースがリストされます。ゾーンからインタフェースを削除するには、削除するインタフェースを選択して、［'変更'］をクリックし、［'____no_zone____（ゾーンなし）'］を選択します。ゾーンにインタフェースを追加するには、追加するインタフェースを選択して、［'変更'］をクリックし、使用可能のいずれかのゾーンを選択します。［'ユーザ定義'］を使用して、ユーザ固有の設定で特殊なインタフェースを作成することもできます。

**［許可されるサービス］** このオプションは、保護されている(システムに対するアクセスが禁止されている)ゾーンにシステムサービスを提供するために使用します。デフォルトでは、外部ゾーンだけが保護されています。

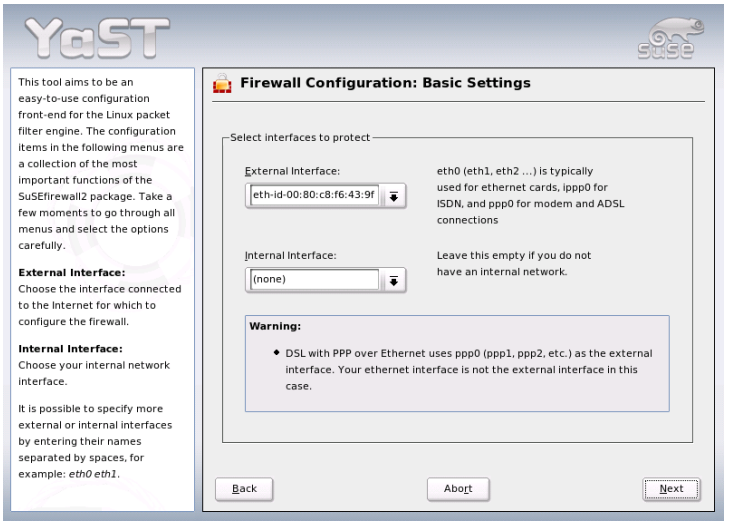


*Figure 34.2: YaSTによるファイアウォールの設定*

その場合は、外部ホストからアクセス可能にする必要があるサービスを明示的に許可する必要があります。［'選択されたゾーンで許可されるサービス'］で適切なゾーンを選択して、許可するサービスを有効化します。

**［マスカレード］** マスカレードを使用すると、内部ネットワークをインターネットなどの外部ネットワークから隠ぺいできます。また、内部ネットワークから外部ネットワークに透過的にアクセスできるようになります。外部ネットワークから内部ネットワークへの要求はブロックされますが、内部ネットワークからの要求は、外部から見ると、マスカレードサーバから発信されたように見えます。

内部ホストの特殊なサービスを外部ネットワークから利用可能にする必要がある場合は、そうしたサービス用の特殊なリダイレクトルールを追加します。

**［ブロードキャスト］** このダイアログでは、ブロードキャストが可能なUDPポートを設定します。各ゾーンに対して、必要なポート番号またはサービスを空白で区切って指定します。`/etc/services`も参照してください。

このダイアログで、禁止されたブロードキャストのログを有効にできます。ただし、Windowsホストは、互いを認識するためにブロードキャストを使用するため、大量のパケットが禁止されることになります。このため、ログを有効にすると大量パケットがすべてログに記録されてしまいます。

**[IPsecサポート]** このダイアログでは、外部ネットワークからのIPsecサービスを許可するかどうかを設定します。［'詳細'］で信頼するパケットを設定します。

**[ログレベル]** ログには、パケットの許可と禁止の2つのルールがあります。許可されたパケットはACCEPTED、禁止されたパケットはDROPPEDまたはREJECTEDと表記されます。その両方について、［'すべてログに記録する'］、［'Log Critical(重要なパケットを記録する)'］、［'ログに何も記録しない'］のいずれかを選択できます。

機能設定が終わったら、［'次へ'］をクリックしてダイアログを閉じます。ファイアウォール設定のゾーンごとのサマリーが表示されるので、すべての設定を再確認してください。このサマリーには、許可されたすべてのサービス、ポート、プロトコルがリストされます。設定し直す場合は、［'戻る'］をクリックしてください。そのままでよければ、［'了解'］をクリックして設定を保存します。

### 手動設定

以降では、適切に設定するための手順を順を追って説明します。各設定項目には、ファイアウォールとマスカレードのどちらに関連するかを示してあります。設定ファイルで述べられているDMZ (非武装地帯)関連の設定については、ここでは取り上げません。DMZは、大規模な組織に見られる複雑なネットワークインフラストラクチャ(企業ネットワークなど)でのみ使用されるものであり、広範な設定とこの分野に関する深い知識を必要とします。

まず、YaSTモジュールを使用して、使用中のランレベル(通常3または5)でSuSEfirewall2を有効にします。これにより、`/etc/init.d/rc?.d/`ディレクトリ内のSuSEfirewall2_*スクリプトへのシンボリックリンクが設定されます。

**FW_DEV_EXT (ファイアウォール、マスカレード)**
インターネットへの接続デバイス。モデム接続の場合は、`ppp0`を指定します。IDSNリンクの場合は、`ippp0`を指定します。DSL接続には、`dsl0`を指定します。デフォルトルートに対応するインタフェースを使用する場合は、`auto`を指定します。

**FW_DEV_INT (ファイアウォール、マスカレード)**
内部プライベートネットワークへの接続デバイス(eth0など)。内部ネットワークがなく、ファイアウォールが動作するホストのみを保護する場合は、空にします。

**FW_ROUTE (ファイアウォール、マスカレード)**
マスカレード機能が必要な場合は、yesに設定します。内部ホストのネットワークアドレス(例: 192.168.x.x)がインターネットルータで無視されるようになるため、内部ホストは外部から見えなくなります。

マスカレード機能なしのファイアウォールで、内部ネットワークへのアクセスを許可する場合は、これをyesに設定します。この場合、内部ホストでは公式のIPアドレスを使用する必要があります。ただし、外部ネットワークから内部ネットワークへのアクセスは許可しないのが普通です。

**FW_MASQUERADE (マスカレード)** マスカレード機能が必要な場合は、yesに設定します。これにより、内部ホストからインターネットへの仮想的な直接接続が実現されます。内部ネットワークのホストとインターネット間にプロキシを設定すると、セキュリティが強化されます。プロキシサーバが提供するサービスにはマスカレードは必要ありません。

**FW_MASQ_NETS (マスカレード)** マスカレードを行うホストやネットワークを指定します。各エントリはスペースで区切ります。次に例を示します。

```
FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
```

**FW_PROTECT_FROM_INT (ファイアウォール)**
内部ネットワークからの攻撃に対してファイアウォールホストを保護するには、yesに設定します。サービスは、明示的に有効にした場合にのみ、内部ネットワークに対して提供されます。FW_-SERVICES_INT_TCPおよびFW_SERVICES_INT_UDPも参照してください。

**FW_SERVICES_EXT_TCP (ファイアウォール)**
使用可能にするTCPポートを指定します。一般的な自宅用のワークステーションでは、通常サービスは提供していないため、空にします。

**FW_SERVICES_EXT_UDP (ファイアウォール)**
UDPサービスを実行しており、それを外部から使用できるようにする場合を除き、空にします。UDPを使用したサービスとしては、DNSサーバ、IPSec、TFTP、DHCPなどがあります。これらのサービスを使用可能にする場合は、使用するUDPポートを指定します。

**FW_SERVICES_INT_TCP (ファイアウォール)**
この変数には、内部ネットワークに対して使用可能にするサービスを指定します。記述形式はFW_SERVICES_EXT_TCPと同じですが、この設定は内部ネットワークに適用されます。この変数は、FW_PROTECT_FROM_INTをyesに設定した場合のみ設定します。

**FW_SERVICES_INT_UDP (ファイアウォール)**
FW_SERVICES_INT_TCPの項を参照してください。

ファイアウォールの設定が完了したら、設定をテストします。ファイアウォールのルールセットは、root権限でSuSEfirewall2 startを実行すると作成されます。次に、telnetを使用して、たとえば外部ホストから接続が実際に拒否されるかどうかを確認します。その後、/var/log/messagesを参照します。次のようなログが記録されているはずです。

```
Mar 15 13:21:38 linux kernel:SFW2-INext-DROP-DEFLT IN=eth0
OUT= MAC=00:80:c8:94:c3:e7:00:a0:c9:4d:27:56:08:00 SRC=192.168.10.0
DST=192.168.10.1 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=15330 DF PROTO=TCP
SPT=48091 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A061AFEBC0000000001030300)
```

他にも、nmapやnessusといったパッケージを使用して、ファイアウォールの設定をテストできます。パッケージをインストールすると、nmapのドキュメントは/usr/share/doc/packages/nmapに、nessusのドキュメントは/usr/share/doc/packages/nessus-coreに置かれます。

### 34.1.5 関連資料

SuSEfirewall2の最新情報およびその他のドキュメントは、/usr/share/doc/packages/SuSEfirewall2で参照できます。また、netfilter/iptablesプロジェクトのホームページhttp://www.netfilter.orgでは、さまざまな文書を多くの言語で参照できます。

## 34.2 SSH:安全なネットワーク操作

ネットワーク環境に多数のコンピュータがインストールされるほど、遠隔地からホストへのアクセスが必要となります。通常、これはユーザが認証のために

ログイン文字列とパスワード文字列を送信することを意味します。これらの文字列が平文で転送される限り、パケットが盗聴されて、転送元ユーザのアカウントにアクセスするために、そのアカウントを知る権限ユーザを使用せずに不正使用される恐れがあります。これはユーザのファイルがすべて攻撃者に公開されてしまうだけでなく、不正なアカウントを使用して管理者やrootユーザのアクセス権を取得したり、他のシステムに侵入できることにもなります。従来、リモート接続の確立にはtelnetが使用されていましたが、telnetには暗号化形式や他のセキュリティメカニズムのパケット盗聴に対する防護機能が用意されていません。その他にも、従来のFTPプロトコルや一部のリモートコピープログラムのように、保護機能のない通信チャネルが存在します。

SSHスイートは、認証文字列(通常はログイン名とパスワード)およびホスト間でやりとりされる他のすべてのデータを暗号化することで、必要な保護を提供します。SSHを使用した場合も、データフローを第三者に記録される可能性は残りますが、内容は暗号化されており、暗号鍵を知らない限り平文に戻すことはできません。そのため、SSHを使用すると、インターネットのように安全でないネットワーク上でも安全な通信が可能になります。SUSE LINUX付属のSSH機能はOpenSSHです。

### 34.2.1 OpenSSHパッケージ

SUSE LINUXでは、デフォルトでパッケージOpenSSHがインストールされます。これによりtelnet、rlogin、rsh、rcp、およびftpの代わりにプログラムssh、scp、およびsftpが使用可能になります。デフォルト設定では、SUSE LINUXシステムのシステムアクセスはOpenSSHユーティリティを使用し、ファイアウォールがアクセスを許可した場合にのみ可能になります。

### 34.2.2 sshプログラム

sshプログラムを使用すると、リモートシステムにログインして対話形式で作業できます。このプログラムは、telnetおよびrloginに代わるものです。sloginプログラムは、sshを指す単なるシンボリックリンクです。たとえば、コマンドssh sunを使用してホストsunにログインするとします。このホストはsunのパスワード入力を求めるプロンプトを表示します。

認証に成功すると、リモートのコマンドラインで作業したり、YaSTなどの対話型アプリケーションを使用できます。ローカルユーザ名がリモートユーザ名と異なる場合は、ssh -l augustine sunまたはssh augustine@sunを使用して、異なるログイン名でログインできます。

さらに、sshでは、rshから既知されるリモートシステム上でコマンドを実行できます。次の例では、ホストsun上でコマンドuptimeを実行し、tmpというディレクトリを作成します。プログラムの出力は、ホストearthのローカル端末に表示されます。

```
ssh otherplanet "uptime; mkdir tmp"
tux@otherplanet's password:
1:21pm  up  2:17,  9 users,  load average: 0.15, 0.04, 0.02
```

この例では、両方のコマンドを1つのコマンドで送信するために、引用符が必要です。2つ目のコマンドもsun上で実行するには、このように引用符で囲む必要があります。

### 34.2.3 scp—Secure Copy

scpは、ファイルをリモートマシンにコピーします。これは、rcpに対する安全で暗号化機能を持つ代替策です。たとえば、`scp MyLetter.tex sun:`と入力すると、ホストearthからホストsunにファイルMyLetter.texがコピーされます。earth上でのユーザ名がsun上でのユーザ名と異なる場合は、後者をusername@host形式で指定します。このコマンドには-lオプションがありません。

正しいパスワードを入力すると、scpによりデータ転送が開始され、進行状況バーをシミュレートする一連のアスタリスクが表示されます。また、進行状況バーの右端への到達予想時間も表示されます。すべての出力を抑制するには、オプション-qを指定します。

scpには、ディレクトリ全体の再帰コピー機能も用意されています。コマンド`scp -r src/ sun:backup/`を入力すると、ディレクトリsrcの内容全体がすべてのサブディレクトリを含めてホストsun上のbackupディレクトリにコピーされます。このサブディレクトリが存在しない場合は、自動的に作成されます。

オプション-pはscpに対して、変更のないファイルのタイムスタンプを残すように指示します。-Cを指定するとデータ転送が圧縮されます。この場合、データ転送量は最小限ですみますが、プロセッサにかかる負荷が大きくなります。

### 34.2.4 sftp—安全なファイル転送

安全なファイル転送のために、scpの代わりにsftpプログラムを使用できます。sftpセッション中は、ftpで認識される多数のコマンドを使用できます。特

にファイル名がわからないデータを転送する場合に、sftpプログラムはscpよりも優れた選択肢です。

### 34.2.5 SSHデーモン(sshd)—サーバ側

SSHのクライアントプログラムであるsshおよびscpを操作する場合は、サーバであるSSHデーモンをバックグラウンドで実行し、TCP/IP port 22で接続をリスンする必要があります。このデーモンは、初回起動時に鍵のペアを3組生成します。鍵のペアはそれぞれ、秘密鍵と公開鍵で構成されます。そのため、このプロシージャは公開鍵ベースと呼ばれます。SSHを介した通信のセキュリティを保証するために、秘密鍵ファイルへのアクセスはシステム管理者に限定する必要があります。ファイルアクセス権は、デフォルトインストールにより適切に設定されます。秘密鍵はSSHデーモンでローカルにのみ必要であり、他人には付与しないでください。公開鍵コンポーネント(拡張子.pubで識別)は、接続を要求しているクライアントに送信されます。これは、ユーザ全員が読み込み可能です。

接続はSSHクライアントにより開始されます。待機中のSSHデーモンと要求側のSSHクライアントは、プロトコルとソフトウェアのバージョンを比較して不正なポートを介した接続を防止するために、識別データを交換します。オリジナルのSSHデーモンの子プロセスが要求に応答するため、同時に複数のSSH接続を確立できます。

SSHサーバとSSHクライアントとの通信の場合、OpenSSHはバージョン1および、2のSSHプロトコルをサポートします。新規にインストールされたSUSE LINUXシステムは、デフォルトでバージョン2に設定されます。更新後も引き続きバージョン1を使用する場合は、/usr/share/doc/packages/openssh/README.SuSE内の指示に従ってください。このドキュメントには、SSH 1環境を数ステップでSSH 2作業環境に変換する方法も含まれています。

バージョン1のSSHを使用する場合、サーバはホスト公開鍵とSSHデーモンにより1時間ごとに再生成されるサーバ鍵を送信します。この両方を使用すると、SSHクライアントは自由に選択したセッション鍵を暗号化でき、この鍵がSSHサーバに送られます。また、SSHクライアントはサーバに対して、どの暗号化方式(暗号)を使用するかも指示します。

バージョン2のSSHプロトコルはサーバ鍵を必要としません。クライアント側とサーバ側は、Diffie-Helmanのアルゴリズムを使用して鍵を交換します。

セッション鍵を復号化するにはホストとサーバの秘密鍵が不可欠であり、公開部分からは導出できません。秘密鍵を使用してセッション鍵を復号化できるのは、接続相手のSSHデーモンのみです(man

/usr/share/doc/packages/openssh/RFC.nroffコマンドでマニュアルページを参照してください)。この初期接続フェーズは、SSHクライアントの詳細デバッグオプション-vをオンにすると緊密に監視できます。

デフォルトではバージョン2のSSHプロトコルが使用されます。バージョン1のプロトコルを使用するには、-1スイッチを指定してこの設定を上書きします。クライアントでは、すべてのホスト公開鍵がリモートホストとの初期接続後に~/.ssh/known_hostsに格納されます。このため、man-in-the-middle攻撃、つまり、外部SSHサーバが他の名前とIPアドレスを偽装して使用しようとする攻撃が防止されます。この種の攻撃は、~/.ssh/known_hostsに含まれていないホスト鍵が使用されたことで検出されるか、適切な秘密鍵がないためにサーバがセッション鍵を復号化できないことで検出されます。

/etc/ssh/に格納された秘密鍵と公開鍵のバックアップを、外部の安全な場所に保管することをお薦めします。これにより、鍵の変更を検出でき、再インストール後は古い鍵を再び使用できます。また、ユーザの動揺を招くような警告を出す必要もなくなります。警告にも関わらず実際には正しいSSHサーバであることが確認された場合は、このシステムに関する既存のエントリを~/.ssh/known_hostsから削除する必要があります。

### 34.2.6 SSHの認証メカニズム

この時点で実際の認証が発生します。最も単純な形式の認証は、前述のようにパスワードを入力することからなっています。SSHの目標は、使いやすく安全なソフトウェアを提供することでした。これは、rshおよびrloginにとって代わるという側面もあるため、SSHは日常的な使用に適した認証方式も提供できるようにする必要があります。そのために、SSHはもう1つ、ユーザが生成する鍵のペアを使用します。SSHパッケージには、そのためのヘルパープログラムが用意されています。ssh-keygenです。ssh-keygen -t rsaまたはssh-keygen -t dsaを入力すると鍵のペアが生成され、鍵を格納するベースファイルの名前を求めるプロンプトが表示されます。

デフォルト設定を確認し、パスフレーズ要求に応答します。ソフトウェアから空のパスフレーズが提示された場合も、ここで説明する手順には10〜30文字のテキストを使用することをお薦めします。短くて単純な語句は使用しないでください。また、パスフレーズを再入力して確認してください。その後、秘密鍵と公開鍵の格納場所(この例ではファイルid_rsaおよびid_rsa.pub)が表示されます。

古いパスフレーズを変更するには、ssh-keygen -p -t rsaまたはssh-keygen -p -t dsaを使用します。公開鍵コンポーネント(この例ではid_rsa.pubファイル)をリモートマシンにコピーし、~/.ssh/

authorized_keysファイルに保存します。次回の接続確立時には、パスフレーズで自己認証するように要求されます。このプロンプトが表示されない場合は、これらのファイルの位置と内容を確認してください。

長時間実行する場合、この手順はその都度パスワードを入力するよりも煩雑です。そのため、SSHパッケージにはssh-agentというツールが用意されており、Xセッションの存続期間中は秘密鍵が保持されます。Xセッション全体はssh-agentの子プロセスとして開始されます。この場合に最も簡単な方法は、.xsessionファイルの先頭にある変数usesshをyesに設定し、KDMやXDMなどのディスプレイマネージャを介してログインすることです。また、ssh-agent startxと入力する方法もあります。

これで、sshまたはscpを通常どおり使用できます。前述のように公開鍵を配布している場合、パスワードを求めるプロンプトは表示されなくなります。Xセッションを終了するか、xlockなどのパスワード保護アプリケーションでロックすることに注意してください。

バージョン2のSSHプロトコル導入に関連する変更は、すべてファイル/usr/share/doc/packages/openssh/README.SuSEにも記載されています。

### 34.2.7 X、認証および転送メカニズム

前述したセキュリティ関連の改善に加えて、SSHを使用するとリモートXアプリケーションの使用も簡略化されます。オプション-Xを指定してsshを実行すると、リモートマシン上でDISPLAY変数が自動的に設定され、すべてのX出力が既存のSSH接続を介してリモートマシンにエクスポートされます。それと同時に、権限のないユーザは、この方法でリモートで起動してローカルに表示していたXアプリケーションのパケットを盗聴できなくなります。

オプション-Aを追加すると、ssh-agentの認証メカニズムが次のマシンに繰り越されます。これにより、事前に接続先ホストに公開鍵を配布してそこで適切に保存している場合にのみ、パスワードを入力しなくても様々なマシンから作業できます。

デフォルト設定では両方のメカニズムが無効になっていますが、システム単位の設定ファイル/etc/ssh/sshd_configまたはユーザの~/.ssh/configファイル内でいつでも永続的に有効にすることができます。

sshを使用してTCP/IP接続をリダイレクトすることもできます。次の例では、SSHに対してそれぞれSMTPポートとPOP3ポートをリダイレクトするように指定しています。

```
ssh -L 25:sun:25 earth
```

このコマンドを使用すると、*earth port 25* (SMTP)に送られた接続は、すべて暗号化チャネルを介してsunのSMTPポートにリダイレクトされます。これが特に役立つのは、SMTP-AUTHまたはPOP-before-SMTP機能のないSMTPサーバを使用する場合です。ネットワークに接続している任意の場所から「ホーム」メールサーバに電子メールを転送して配信できます。同様に、次のコマンドを使用すると、earth上のすべてのPOP3要求(ポート 110)をsunのPOP3ポートに転送できます。

```
ssh -L 110:sun:110 earth
```

どちらのコマンドも、権限付きのローカルポートに接続するためrootユーザで実行する必要があります。電子メールは、既存のSSH接続で標準ユーザにより送受信されます。これを機能させるには、SMTPとPOP3のホストをlocalhostに設定する必要があります。追加情報は、前述の各プログラムのマニュアルページおよび/usr/share/doc/packages/openssshにある該当ファイルを参照してください。

## 34.3 パーティションとファイルの暗号化

### 34.3.1 適用事例

ユーザは皆、第三者に公開することを意図していない機密データを持っています。接続環境やモバイル環境が充実すればするほど、データの取り扱いに細心の注意を払わなければならなくなります。ネットワーク接続または直接的な物理アクセスにより第三者がデータにアクセスできる場合は、ファイルやパーティション全体を暗号化するのが効果的です。以下に、想定される使用状況をいくつか挙げます。

**ラップトップ** ラップトップを携えて出張する場合は、ハードディスク上の機密データを格納するパーティションを暗号化するとよいでしょう。データは暗号化ファイルシステムまたは1つの暗号化ファイル内にあるため、ラップトップの紛失や盗難の際にアクセスされる心配はありません。

**リムーバブルメディア** USBフラッシュドライブや外付けハードディスクには、ラップトップと同様、盗難の危険があります。暗号化ファイルシステムを利用することにより、第三者によるアクセスを防止できます。

### 34.3.2 YaSTによる暗号ファイルシステムのセットアップ

YaSTを使用して、インストール時に、またはインストール済みのシステムで、ファイルやパーティションを暗号化できます。暗号化ファイルは既存のパーティションレイアウトにうまく組み込めるために、常に作成することができます。パーティション全体を暗号化するには、パーティションレイアウト内に暗号化用の専用パーティションが必要になります。ただし、YaSTによって提示されるデフォルトの標準パーティション設定には、暗号化パーティションは含まれていません。暗号化パーティションは、パーティション設定用のダイアログで手動で設定します。

#### インストール時の暗号化パーティションの作成

**Warning**

**パスワード入力**

暗号化パーティションのパスワードを設定する際は、パスワードのセキュリティに関する警告をよく読み、パスワードをきちんと記憶してください。暗号化データには、パスワードを入力しないとアクセスできません。

**Warning**

YaSTの［パーティションのエキスパート設定設定］ダイアログ(項2.7.5. 「パーティション」を参照)には、暗号化パーティションの作成に必要なオプションが用意されています。通常のパーティションを作成するときと同様、［'作成'］をクリックします。表示されるダイアログで、フォーマット方式、マウントポイントなど、新しいパーティションのパーティションパラメータを指定します。［'暗号化ファイルシステム'］をクリックして暗号化パーティションを作成します。その後に表示されるダイアログで、パスワードを設定します。セキュリティ上の理由から、パスワードは2回入力します。パーティション作成ダイアログで［'了解'］をクリックすると、新しい暗号化パーティションが作成されます。ブート時にオペレーティングシステムによってパスワードの入力が求められ、その後パーティションがマウント可能になります。

起動時に暗号化パーティションをマウントしたくない場合は、パスワードの入力が求められたときに(Enter)キーを押します。その後、再度パスワードの入力が求められたらそれを拒否します。この場合、暗号化ファイルシステムはマウントされず、オペレーティングシステムはブート処理を継続します。これにより、データが保護されます。パーティションは、いったんマウントされるとすべてのユーザが使用できるようになります。

必要な場合にのみ暗号化ファイルシステムをマウントするには、［'Fstabのオプション'］ダイアログの［'システムスタート時にマウントしない'］をオンにします。対応するパーティションはシステム起動時にマウント対象外になります。その後、そのパーティションを使用可能にするには、mount ⟨*name_of_partition*⟩ ⟨*mount_point*⟩を実行して手動でマウントします。そのパーティションをマウントするプロンプトが表示されたら、パスワードを入力します。パーティションを使用し終わったら、umount name_of_partitionを実行してアンマウントし、他のユーザからアクセスされないようにします。

### 稼動中のシステムでの暗号化パーティションの作成

**Warning**

**稼動中のシステムでの暗号化のアクティブ化**

インストール時と同様に、稼動中のシステムに暗号化パーティションを作成することもできます。ただし、既存のパーティションを暗号化すると、そのパーティションの格納データはすべて失われます。

**Warning**

稼動中のシステムでは、YaSTコントロールセンターで'システム' → 'ディスクの分割'を選択します。'はい'をクリックして続行します。先ほどの'作成'ではなく、'編集'をクリックします。以降の手順は同じです。

### 暗号化ファイルのインストール

パーティションの使用に加え、単一のファイル内に機密データを保持する暗号化ファイルシステムを作成することもできます。暗号化ファイルシステムの作成にも同じYaSTダイアログを使用します。'暗号化ファイル'を選択し、作成するファイルのパスを予定サイズとともに入力します。フォーマット設定およびファイルシステム種別については、提示される設定をそのまま使用します。次に、マウントポイントを指定し、その暗号化ファイルシステムをブート時にマウントするかどうかを［Fstabのオプション］で指定します。

暗号化ファイルの利点は、ハードディスクのパーティションを変更せずにファイルを追加できることです。暗号化ファイルは、ループデバイスを活用してマウントされ、通常のパーティションのように動作します。

**viを使用したファイルの暗号化**

暗号化パーティションを使用した場合の欠点は、このパーティションをマウントしている間は、rootユーザしかデータにアクセスできないことです。このようにならないために、viを使用して暗号化することができます。

`vi -x filename`を使用して新しいファイルを編集します。viは、パスワードの入力を求めた後、ファイルの内容を暗号化します。再度このファイルにアクセスするときは必ず、viにより正しいパスワードの入力が求められます。

実際に保存する場合は、その暗号化テキストファイルをセキュリティ保護済みのパーティションに格納することもできます。viで使用されている暗号のメカニズムがあまり強固な暗号化でないとわかっているため、パスワード入力機能が役に立ちます。

### 34.3.3 リムーバブルメディアの内容の暗号化

外付けハードディスク、USBフラッシュデバイスなどのリムーバブルメディアは、YaSTによって他のハードディスクと同様に認識されます。リムーバブルメディアでも、ファイルやパーティションの暗号化をこの手順で行えます。ただし、リムーバブルメディアは、通常システムの稼動中に接続するため、ブート時にはマウントしないでください。

## 34.4 セキュリティと機密性

LinuxまたはUNIXシステムの主な特性は、同時に複数のユーザを処理できること(マルチユーザ)と、これらのユーザが同じコンピュータ上で同時に複数のタスクを実行できること(マルチタスキング)です。さらに、オペレーティングシステムはネットワークを意識させません。通常、ユーザは自分が使用しているデータやアプリケーションが各自のマシンからローカルに提供されているのか、ネットワークを介して使用可能になっているかを意識することはありません。

マルチユーザ機能を使用する場合、様々なユーザのデータを別々に格納する必要があります。また、セキュリティとプライバシーを保証する必要があります。データのセキュリティは、コンピュータをネットワーク経由でリンクできるようになる以前から、すでに重要な問題になっていました。現在と同様に、最重要課題は、データメディア(ほとんどの場合はハードディスク)が消失したり他の方法で破損した場合にも、データを使用可能な状態で維持する機能でした。

この章では、主として機密性の問題とユーザのプライバシーを保護する手段について重点的に説明します。ただし、定期的に更新されて作業可能なテスト済みバックアップを常に備えておくための手順を確立することが包括的なセキュリティの概念には不可欠であるという点については、詳しく説明しません。この手順がなければ、何らかのハードウェア障害が発生した場合のみでなく、誰かが不正にアクセスしてファイルを改ざんしたという疑いが生じた場合にも、データの復旧作業に手間と時間がかかることになります。

### 34.4.1 ローカルセキュリティとネットワークセキュリティ

データにアクセスするには、次のような複数の方法があります。

- 必要な情報を持っているユーザとの個人的な通信、またはコンピュータ上のデータへのアクセス
- コンピュータのコンソールからの直接アクセス(物理アクセス)
- シリアル回線を介したアクセス
- ネットワークリンクを使用したアクセス

いずれの場合も、ユーザが問題のリソースやデータにアクセスするには、認証を受ける必要があります。この点ではWebサーバはあまり限定的ではありませんが、すべての個人データを第三者に公開しないようにする必要はあります。

上記のリストのうち、最初の項目では、銀行の担当者に連絡したときに自分が口座名義人であるという証明を要求される場合のように、多くの対話を必要とします。この場合は、署名、PINまたはパスワードなど、自分の身元を証明する情報を提供するように要求されます。場合によっては、単に知っている情報を断片的に述べ、言葉巧みに信頼させて相手から情報を引き出す可能性もあります。その結果、情報が少しずつ明らかになり、そのことに気づかないことさえあります。ハッカーの間では、この行為を「*ソーシャルエンジニアリング*」と呼んでいます。この行為を防止するには、人々を教育し、言語や情報を自覚して取り扱うしかありません。通常、攻撃者はコンピュータシステムに侵入する前に、受付係、会社のサービススタッフ、家族などをターゲットにしようとします。多くの場合、このようなソーシャルエンジニアリングに基づく攻撃は、はるか後になるまで発見されません。

他人のデータに不正にアクセスしようとする第三者は、従来の方法を使用して他人のハードウェアに直接接続を試みることもあります。そのため、マシンは他人にコンポーネントを削除、交換または無効化されないように、あらゆる改

ざんから保護する必要があります。これは、バックアップ、ネットワークケーブル、電源コードにも当てはまります。また、一部のキーの組合せは異常動作を引き起こす場合があるため、ブート手順も保護してください。自己防衛のためには、BIOSとブートローダーのパスワードを設定する必要があります。

シリアルポートに接続されたシリアル端末は、従来から多くの場所で使用されています。ネットワークインタフェースとは異なり、ホストとの通信をネットワークプロトコルに依存しません。デバイス間での単なる文字のやりとりには、単純なケーブルまたは赤外線ポートが使用されます。このようなシステムでは、ケーブル自体が最大の弱点です。古いプリンタがケーブルに接続されていれば、ケーブル経由で伝達される情報を記録するのは簡単です。攻撃対象によっては、プリンタで実行できることであれば他の方法でも実行できます。

ホスト上でファイルをローカルに読み込むには、他のホスト上でサーバとのネットワーク接続をオープンする以外のアクセスルールが必要です。ローカルセキュリティとネットワークセキュリティには、違いがあります。つまり、データをどこかにパケット単位で送信する必要がある場合には、回線が使用されます。

### ローカルセキュリティ

ローカルセキュリティは、コンピュータが稼働している場所の物理環境から始まります。マシンは、セキュリティ上の要件やニーズに沿った場所に設置してください。ローカルセキュリティの主な目標は、誰も他人の権限や識別情報を偽れないように、常にユーザを相互に分離しておくことです。これは遵守すべき原則ですが、ユーザrootの場合はシステム上で最高の権限を持つため、このことが特に重要になります。rootユーザは、パスワードを求めるプロンプトなしで、他のすべてのローカルユーザの識別情報を使用して、ローカルに格納されているファイルをすべて読み込むことができます。

### パスワード

当然、Linuxシステムでパスワードが平文として格納されることはなく、入力されたテキスト文字列は単に保存されているパターンと照合されるのみではありません。平文として格納され、保存されているパターンと照合されるのみであれば、対応するファイルに誰かがアクセスした直後に、システム上のすべてのアカウントが危険にさらされることになります。代わりに、格納されているパスワードは暗号化されており、入力されるパスワードもそのたびに再び暗号化され、暗号化された2つの文字列が比較されます。この方法でセキュリティレベルが向上するのは、暗号化されたパスワードを逆算して元のテキスト文字列に戻せない場合のみです。

実際には、この処理は特殊なアルゴリズムによって達成されます。このアルゴリズムは、一方向にしか機能しないため、「*トラップドアアルゴリズム*」とも呼ばれます。暗号化された文字列を攻撃者が入手しても、単に同じアルゴリズムを再適用するだけでは他人のパスワードを取得できません。代わりに、暗号化すると他人のパスワードになる組合せが見つかるまで、考えられる文字の組合せをすべてテストする必要があります。パスワードが長さ8文字であれば、計算が必要な組合せの候補は膨大な数になります。

1970年代には、使用されていたアルゴリズムが比較的低速で、1つのパスワードを暗号化するだけで数秒かかっていたため、この方法が他の方法よりも安全であると言われていました。ただし、その後はPCのパフォーマンスが向上し、毎秒数10万～数100万回の暗号化を実行できるようになっています。このため、暗号化されたパスワードは通常のユーザが参照できないようにする必要があります(通常のユーザは`/etc/shadow`ファイルを読み込めません)。さらに重要なのは、何らかのエラーが原因でパスワードファイルが参照可能になった場合に備えて、簡単に推測できないパスワードを使用することです。したがって、「tantalise」のようなパスワードを「t@nt@1ls3」に「変換」したとしても、実際には役に立ちません。

語句の一部の文字を数字に同じパターンで置き換えただけでは、安全とは言えません。辞書を使用して語句を推測するパスワードクラックプログラムも、これと同様の置換を行います。そこで、「The Name of the Rose」 by Umbert Eco (ウンベルトエーコ著『薔薇の名前』)のように、文や書名に含まれる語句の頭文字など、一般的な意味はなく、自分にしか意味のない語句を作成するのが適切な方法です。こうして作成される次のようなパスワードは安全と言えます。"TNotRbUE9"。これに対して、"beerbuddy"や"jasmine76"のようなパスワードは、ユーザに関してわずかしか知識のない他人でさえ簡単に推測できます。

### ブート手順

システムは、ドライブ全体を取り外すか、BIOSパスワードを設定してハードディスクからでなければブートできないようにBIOSを設定し、フロッピーやCDからはブートできないように設定してください。通常、Linuxシステムはブートローダーから起動するため、ブートしたカーネルに追加のオプションを渡すことができます。`/boot/grub/menu.lst`内でパスワードを追加設定し、他のユーザがこの種のパラメータをブート時に使用できないようにしてください(章 8. ブートローダを参照)。これはシステムのセキュリティに不可欠です。カーネル自体が`root`権限で実行されるのみでなく、システム起動時に`root`権限を付与する最初の認可でもあります。

### ファイルのパーミッション

通常は、特定のタスクに可能な最も限定的な権限で作業します。たとえば、電子メールを読み書きするには、rootユーザである必要はありません。メールプログラムにバグがあると、このバグが攻撃にさらされ、起動時にプログラムのパーミッションが正確に処理されてしまう可能性があります。限定的な権限のルールに従って、考えられる損害を最小限に抑えてください。

SUSEディストリビューションパッケージに付属する200,000以上のファイルについては、パーミッションが慎重に選択されています。ソフトウェアや他のファイルを追加インストールするシステム管理者は、特にパーミッションビットの設定時には細心の注意を払う必要があります。経験豊富でセキュリティ意識の高いシステム管理者は、常にコマンドlsで-lオプションを使用して広範なファイルリストを取得します。これにより、不正なファイルパーミッションを即時に検出できます。不正なファイル属性は、そのファイルが変更または削除された可能性を意味するだけではありません。このように変更されたファイルがrootユーザにより実行される可能性や、設定ファイルの場合はこの種のファイルがプログラムでrootユーザの権限で使用される可能性があります。このため、攻撃者が侵入する可能性が大幅に増大します。このような攻撃は、カッコウが他の鳥をだまして自分の卵を孵化させるのと同様に、プログラム(卵)が他のユーザ(鳥)によって実行(孵化)されるため、カッコウの卵と呼ばれます。

SUSE LINUXシステムでは、ファイルpermissions、permissions.easy、permissions.secure、およびpermissions.paranoidがすべてディレクトリ/etcにあります。これらのファイルの目的は、world-writableなディレクトリなどの特殊な権限や、ファイルに対するsetuser IDビットを定義することです(setuser IDビットが設定されているプログラムは、それを起動したユーザの権限ではなく、ファイル所有者、ほとんどの場合はrootユーザの権限で実行されます)。管理者は、ファイル/etc/permissions.localを使用して自分専用の設定を追加できます。

前述のファイルのうち、SUSEの設定プログラムで権限の設定に使用されるファイルを定義するには、YaSTで['セキュリティ']を選択します。このトピックの詳細は、/etc/permissions内のコメントまたはchmodのマニュアルページを(manchmodコマンドを実行して)参照してください。

### バッファオーバーフローと書式文字列のバグ

プログラムがユーザによる変更が可能なデータを処理すると思われる場合は、特に注意する必要がありますが、これは通常のユーザよりもアプリケーションプログラマにとって問題です。プログラマは、小さすぎてデータを保持できないメモリ領域に書き込むことなく、自分のアプリケーションでデータが適切に

解析されることを確認する必要があります。また、プログラムでは、専用に定義されたインタフェースを使用して、データを一貫した方法で受け渡す必要があります。

実際のメモリバッファのサイズを考慮しないと、そのバッファへの書き込み時に「バッファオーバーフロー」が発生する可能性があります。また、このデータ(ユーザが生成)に使用される領域が、バッファ内で使用可能な領域を超える場合があります。その結果、データはそのバッファ領域の終わりを超えて書き込まれ、状況によってはプログラムで単にユーザデータが処理されるのではなく、ユーザ(プログラマではなく)が変更したプログラムシーケンスが実行される可能性があります。この種のバグは、特にプログラムが特殊な権限で実行されている場合には、重大な結果を招きます(項34.4.1.「ファイルのパーミッション」を参照)。

*書式文字列*のバグの場合、動作は少し異なりますが、プログラムの異常動作を引き起こす可能性のあるユーザ入力です。ほとんどの場合、この種のプログラミングエラーは、setuidプログラムやsetgidプログラムなど、特殊な権限で実行されるプログラムに見られます。これも、対応する実行権限をプログラムから削除することで、データとシステムをこの種のバグから保護できることを意味します。また、最善の方法は、考えられる最小権限を使用するというポリシーを適用することです(項34.4.1.「ファイルのパーミッション」を参照)。

バッファオーバーフローと書式文字列のバグがユーザデータの処理に関連するバグであるとすれば、アクセス権がローカルアカウントに付与されている場合にのみ発生するわけではありません。レポートされているバグの多くは、ネットワークリンク上でも利用される可能性があります。したがって、バッファオーバーフローと書式文字列のバグは、ローカルセキュリティとネットワークセキュリティの両方に関連する問題として分類する必要があります。

### ウィルス

通説とは異なり、Linux上で動作するウィルスは存在します。ただし、判明しているウィルスは、テクニックが意図したとおりに動作することを証明するために、作成者が*自分のアイデアの証明*としてリリースしたものです。この種のウィルスは、これまでのところいずれも*一般には*検出されていません。

ウィルスは、活動するホストがなければ存続も拡散もできません。たとえば、ホストがプログラムやシステムの重要な記憶領域(マスターブートレコードなど)であり、そこにウィルスのプログラムコードを書き込む必要があるとします。Linuxにはマルチユーザ機能があるため、特定のファイルへの書き込みアクセスを制限でき、これは特にシステムファイルの場合に重要です。したがって、`root`権限で通常の作業を実行すると、システムがウィルスに感染する可

能性が増大します。これに対して、考えられる最小権限を使用するという原則に従えば、ウィルスに感染する可能性は低下します。

それとは別に、実際には知らないインターネットサイトからはプログラムを実行しないようにする必要があります。SUSEのRPMパッケージは、その作成に必要な措置が講じられたデジタルラベルとして暗号署名を使用します。ウィルスは、管理者やユーザにセキュリティに関して必要な自覚が欠けており、設計によって高度に保護されているシステムであっても危険にさらす可能性があることを示す典型的な兆候です。

ウィルスをワームと混同しないようにする必要があります。ワームの対象はネットワーク全体です。ワームの拡散にはホストを必要としません。

### ネットワークセキュリティ

ネットワークセキュリティは、外部で開始される攻撃から保護する場合に重要です。ユーザ認証にユーザ名とパスワードを必要とする典型的なログイン手順は、ローカルセキュリティの課題です。ネットワーク経由の特殊なログインの場合は、2つのセキュリティの課題を区別してください。実際の認証までに発生する処理はネットワークセキュリティに関連し、その後に発生する処理はローカルセキュリティに関連します。

### X Window SystemとX認証

冒頭に述べたように、ネットワーク透過性は、UNIXシステムの中核的な特性の1つです。UNIXオペレーティングシステムのウィンドウシステムであるXは、この機能を優れた方法で実現します。Xを使用すると、リモートホストでログインしてグラフィカルプログラムを起動しても基本的には問題はなく、グラフィカルプログラムはネットワーク経由で送信されてコンピュータに表示されます。

Xサーバを使用してXクライアントをリモートで表示する必要がある場合、Xサーバは管理対象のリソース(ディスプレイ)を不正アクセスから保護する必要があります。より厳密には、クライアントプログラムに特定の権限を付与する必要があります。X Window Systemでは、この権限付与を ホストベースのアクセスコントロールおよびCookieベースのアクセスコントロールと呼ばれる2通りの方法で実行できます。前者は、クライアントが実行されるホストのIPアドレスに依存します。これを制御するプログラムがxhostです。xhostは正当なクライアントのIPアドレスをXサーバに属する小型データベースに入力します。ただし、認証はIPアドレスに依存するため、安全度は高くありません。たとえば、クライアントプログラムを送信中のホストで第2のユーザが作業している場合、そのユーザはXサーバにもアクセスできます。IPアドレスを

盗む第三者はこれと同じことをしているに過ぎません。このような欠点があるため、ここではこの認証方式について詳述しません。詳細は、man xhostを参照してください。

Cookieベースのアクセスコントロールの場合は、ある種のIDカードと同様に、Xサーバと正当なユーザにのみ認識される文字列が生成されます。このCookie(通常のクッキーを意味するのではなく、エピグラムが入っている中国のフォーチュンクッキー)は、ログイン時にユーザのホームディレクトリのファイル.Xauthorityに格納され、Xサーバを使用してウィンドウを表示しようとするすべてのXクライアントで使用できます。ファイル.Xauthorityは、ユーザがツールxauthを使用して検査できます。.Xauthorityの名前を変更したり、意図せずにホームディレクトリから削除すると、新規のウィンドウやXクライアントをオープンできなくなります。X Window Systemのセキュリティメカニズムの詳細は、Xsecurityのマニュアルページを(man Xsecurityコマンドを実行して)参照してください。

SSH(セキュアシェル)を使用すると、ユーザには暗号化メカニズムを意識させることなく、ネットワーク接続を完全に暗号化してXサーバに透過的に転送(フォーワード)することができます。この処理は「X転送」とも呼ばれます。X転送は、サーバ側でXサーバをシミュレートし、リモートホスト上でシェルのDISPLAY変数を設定することで行われます。SSHについての詳細な情報は項34.2. 「SSH:安全なネットワーク操作」を参照してください。

**Warning**

ログイン先のホストに対して、ホストの保護を考慮していない場合は、X転送を使用しないでください。X転送を有効にすると、攻撃者がユーザのSSH接続を介して認証し、Xサーバに侵入し、ユーザになりすましてキーボード入力などを行う可能性があります。

**Warning**

### バッファオーバーフローと書式文字列のバグ

項34.4.1. 「バッファオーバーフローと書式文字列のバグ」で説明したように、バッファオーバーフローと書式文字列のバグは、ローカルセキュリティとネットワークセキュリティの両方に関係する課題として分類する必要があります。この種のバグのローカルバリエーションと同様に、ネットワークプログラムでのバッファオーバーフローは、首尾よく利用されてしまうと、ほとんどの場合はroot権限の取得に使用されます。それ以外の場合にも、攻撃者がバグを利用して権限のないローカルアカウントにアクセスし、システムに存在する他の脆弱部分を利用する可能性があります。

ネットワークリンク経由で利用される恐れのあるバッファオーバーフローと書式文字列のバグは、リモート攻撃全体で最も頻度の高い形式であることは確実です。これらのセキュリティ上の弱点(これらの新しく見つかったセキュリティホールを攻撃するためのプログラム)はしばしばセキュリティ関連のメーリングリストに投稿されます。この情報を使用すると、コードの詳細を知らなくても脆弱部分を絞り込むことができます。多年の経験では、オペレーティングシステムメーカーは自社ソフトウェアの問題を修正せざるを得ないため、悪用可能なコードを知ることがオペレーティングシステムのセキュリティレベル向上に役立つことが判明しています。無償ソフトウェアを使用すれば、誰でもソースコードにアクセスでき(SUSE LINUXの場合は、ソースコードがすべて使用可能です)、脆弱部分とその悪用可能なコードを見つけたユーザは誰でも、対応するバグ修正のためのパッチを発行できます。

### DoS—サービス拒否

この種の攻撃の目的はサーバプログラムやシステム全体をブロックしてしまうことです。これを実行するには次のような様々な手段があります。サーバのオーバーロード、ガベージパケットによって絶えずビジー状態にする、リモートバッファオーバーフローを利用するなどがあげられます。通常、DoS攻撃はサービスの消失のみを目的として実行されます。ただし、特定のサービスが使用不能になると、*man-in-the-middle*攻撃(パケット盗聴、TCP接続のハイジャック、偽装攻撃)やDNSポイゾニングなどに対して脆弱になる可能性があります。

### Man in the Middle:パケット盗聴、ハイジャック、偽装攻撃

一般に、通信中のホスト間に割り込む攻撃者が実行するリモート攻撃は、「*man-in-the-middle*攻撃」と呼ばれます。ほぼすべてのタイプのman-in-the-middle攻撃に共通するのは、通常、ユーザは何が起きているのかに気づかないことです。攻撃者が接続要求を自分のマシン宛てに転送するなど、様々なバリエーションが考えられます。その場合、相手のマシンは有効な接続先マシンであるかのように偽装されているので、知らないうちに不正なホストとの接続が確立されることになります。

最も単純なタイプのman-in-the-middle攻撃は*sniffer*と呼ばれ、攻撃者はネットワークトラフィックをリスンするだけです。より複雑な“man in the middle”攻撃は、すでに確立された接続を乗っ取ろうとします(ハイジャック)。これを実現するため、攻撃者は一定時間だけパケットを分析し、接続に属するTCPシーケンス番号を予測できる必要があります。攻撃者が最終的にターゲットホストのロールを停止すると、エラーのため接続が終了したことを示すエラーメッセージが表示されるため、このことがわかります。

暗号化を介してハイジャックから保護されるプロトコルはなく、接続の確立時には単純認証手順しか実行されないことが、攻撃を容易にしています。

*偽装攻撃*は、パケットが偽のソースデータ(通常はIPアドレス)を含むように変更される攻撃です。攻撃に利用させる手段の多くは偽のパケット(Linuxマシンではスーパーユーザである`root`のみしか実行できないようなパケット)を送りつける方法です。

前述の攻撃の多くは、DoSと組み合わせて実行されます。特定のホストを短時間でも突然停止できることが攻撃者にわかれば、ホストは攻撃で一定時間は干渉できなくなるため、攻撃者は容易にアクティブ攻撃をかけられるようになります。

### DNSポイゾニング

DNSポイゾニングとは、攻撃者が偽装したDNSリプライパケットで応答し、サーバの情報を要求しているユーザに対して、そのサーバから特定のデータを送信するよう試みることにより、DNSサーバのキャッシュを破壊することを意味します。多くのサーバは、IPアドレスまたはホスト名に基づいて他のホストとの信頼関係を維持しています。攻撃者は、ホスト間の信頼関係の実際の構造を詳細に理解した上で、自分を信頼のおけるホストの1つとして偽装する必要があります。通常、攻撃者はサーバから受信した一部のパケットを分析し、必要な情報を取得します。また、しばしば攻撃者はネームサーバも適切なタイミングによるDoS攻撃のターゲットとする必要があります。接続先ホストの識別情報を確認できる、暗号化された接続を使用することで、自分自身を保護してください。

### ワーム

ワームはしばしばウィルスと混同されますが、両者には明らかな違いがあります。ウィルスとは異なり、ワームはホストプログラムに感染しなくても活動できます。むしろ、ネットワーク構造上でできるだけ迅速に拡散するように特化されています。Ramen、Lion、Adoreなど、これまでに出現したワームは、bind8やlprNGなどのサーバプログラムの周知のセキュリティホールを使用しています。ワームからの保護は、比較的容易です。セキュリティホールが検出されてからワームがサーバに侵入するまでにある程度の時間があれば、影響を受けるプログラムの更新バージョンが間に合う可能性が大きくなります。これが役立つのは、管理者が問題のシステムにセキュリティ更新を実際にインストールする場合のみです。

### 34.4.2 セキュリティ全般のヒントとテクニック

セキュリティを完全に処理するには、新規の開発に遅れをとらず、常に最新のセキュリティ問題に関する情報を入手することが重要です。システムをあらゆる種類の問題から保護するために、セキュリティ通知で推奨されているパッケージ更新版をできるだけ迅速に入手してインストールすることをお薦めします。SUSEのセキュリティ通知はメーリングリストにて公開されており、リンクhttp://www.novell.com/linux/security/securitysupport.htmlを使用してサブスクライブできます。リストsuse-security-announce@suse.deは、パッケージ更新版に関する最初の情報源であり、アクティブな貢献者の中でもSUSEのセキュリティチームのメンバーが含まれています。

メーリングリストsuse-security@suse.deは、必要なセキュリティ問題の説明の参照先として活用できます。suse-security-announce@suse.deには前述のURLにアクセスしてサブスクライブしてください。

bugtraq@securityfocus.comは、世界中で最もよく知られているセキュリティメーリングリストです。このリストは1日15～20件を受け付けているため、このリストを参照することをお薦めします。詳細は、http://www.securityfocus.comを参照してください。

ここでは、基本的なセキュリティ問題に対処する上で役立つルールについて説明します。

- ジョブごとに考えられる最も限定的な権限セットを使用するというルールに従い、日常的なジョブはrootユーザで実行しないようにします。これにより、カッコウの卵やウィルスに感染する危険性が減少し、自分自身のミスも防止できます。
- 可能な限り、リモートマシンでの作業には常に暗号化された接続を使用します。telnet、ftp、rshおよびrloginの代わりにssh(セキュアシェル)を使用することを、習慣づけてください。
- IPアドレスのみに基づく認証方式は使用しないでください。
- 最も重要なネットワーク関連パッケージは常に更新し、対応するメーリングリストにサブスクライブして、この種のプログラム(bind、sendmail、sshなど)の新バージョンに関する通知を受け取ります。これは、ローカルセキュリティに関連するソフトウェアの場合も同じです。
- /etc/permissionsファイルを変更し、システムのセキュリティに不可欠なファイルのパーミッションを最適化します。プログラムか

らsetuidビットを削除すると、そのジョブは意図した方法で実行できなくなる場合があります。一方、ほとんどの場合、プログラムの潜在的なセキュリティリスクもなくなることを考慮してください。同様のアプローチは、world-writableなディレクトリおよびファイルにも適用できます。

- サーバの正常動作に不可欠でないネットワークサービスを停止します。これにより、システムの安全性が向上します。ソケットがLISTEN状態のオープンポートは、プログラムnetstatで検出できます。オプションとして`netstat -ap`または`netstat -anp`を使用することをお薦めします。`-p`オプションを使用すると、指定した名前のポートを使用しているプロセスを確認できます。

  netstatの結果を、ホスト外部から実行したポートスキャンの結果と比較します。このジョブに適したプログラムはnmapで、マシンのポートがチェックされるのみでなく、その背後で待機中のサービスについてもある程度の情報が得られます。ただし、ポートスキャンは攻撃的な行為と解釈される場合があるため、管理者から明示的な承認を受けない限りホスト上では実行しないでください。最後に、TCPポートのみでなくUDPポートも検出することが重要であることを忘れないでください(オプション`-sS`および`-sU`を使用します)。

- システムのファイルの整合性を信頼できる方法で監視するには、SUSE LINUX配布パッケージに付属のプログラム`tripwire`を使用します。他人に改ざんされないように、`tripwire`で作成されたデータベースは暗号化します。さらに、マシン外部から使用可能なこのデータベースのバックアップは、ネットワークリンクで接続されていない外部データメディアに格納します。

- サードパーティソフトウェアのインストール時には、適切な措置を講じます。幸いにして迅速に発見されたものの、ハッカーがセキュリティソフトウェアパッケージのtarアーカイブにトロイの木馬を組み込んでいた事例があります。バイナリパッケージをインストールする場合は、それをダウンロードしたサイトを信頼します。

  SUSEのRPMパッケージにはgpgの署名が付いています。SUSEが署名に使用している鍵は、次のとおりです。

  ID:9C800ACA 2000-10-19 SUSE Package Signing Key <build@suse.de>

  Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA

  コマンド`rpm --checksig package.rpm`を実行すると、アンインストールされたパッケージのチェックサムと署名が正しいかどうかを確認できます。この鍵は、配布パッケージCDの1枚目と、世界中のほとんどのキーサーバにあります。

- ユーザファイルとシステムファイルのバックアップを定期的にチェックします。バックアップが動作するかどうかをテストしなければ、実際には役に立たない可能性があることを考慮してください。
- ログファイルをチェックします。可能な場合は、小型スクリプトを記述して疑わしいエントリを検索します。実際、これは些細な作業ではありません。結局のところ、どのエントリが例外的でどのエントリがそうでないかがわかるのは自分だけです。
- tcp_wrapperを使用して、サービスに接続できるIPアドレスを明示的に制御できるように、マシンで実行中の個々のサービスへのアクセスを制限します。tcp_wrapperの詳細は、tcpdおよびhosts_accessのマニュアルページを(man 8 tcpdコマンドと、manhosts_accessコマンドを実行して)参照してください。
- SuSEfirewallを使用して、tcpdが提供するセキュリティを強化します(tcp_wrapper)。
- 冗長性のあるセキュリティ対策を設計します。メッセージが2度表示される方が、まったく表示されないよりも有効です。

### 34.4.3 Central Security Reporting Address の使用

セキュリティ関連の問題を発見した場合は(まず使用可能な更新パッケージをチェックしてから)、security@suse.deに電子メールでお送りください。その際に、問題の詳しい説明と、関係するパッケージのバージョン番号をお知らせください。SUSEは、できる限り迅速にお答えするように努めています。電子メールメッセージはpgpで暗号化することをお薦めします。SUSEのpgp鍵は次のとおりです。

```
ID:3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

この鍵はhttp://www.novell.com/linux/security/securitysupport.htmlからもダウンロードできます。

# 35

Linuxのアクセス制御リスト

# Linuxのアクセス制御リスト

この章では、LinuxファイルシステムのPOSIX ACL(アクセス制御リスト)の背景と機能を簡潔に説明します。ACLは、ファイルシステムオブジェクトに対する従来のパーミッション概念の拡張として使用できます。ACLを使用すれば、従来のパーミッション概念で許されていた以上のパーミッションを柔軟に定義できます。

*POSIX ACL*という用語は、このACLが真のPOSIX(*Portable Operating System Interface*)規格であることを示唆しています。ドラフト規格のPOSIX 1003.1eとPOSIX 1003.2cは、いくつかの理由で白紙に戻されました。それにもかかわらず、UNIXファミリに属している多くのシステムに見られるACLは、これらのドラフト規格に基づいており、この章で説明するファイルシステムACLの実装も同様にこの2つの規格に従っています。これらの規格については、`http://wt.xpilot.org/publications/posix.1e/`を参照してください。

## 35.1 ACLの利点

従来どおり、Linuxシステムのファイルオブジェクトごとに3セットのパーミッションが定義されます。この3セットには、読み取り(r)、書き込み(w)、実行(x)の各パーミッションがあり、それぞれが3種類のユーザ(ファイル所有者、グループ、その他のユーザ)ごとに設定されます。そのほかに、*ユーザID*設定ビット、*グループID*設定ビット、スティッキビットを設定できます。この無駄のない概念は、ほとんどの実際的なケースに十分適しています。ただし、複雑なシナリオまたは高度なアプリケーションの場合、以前は、システム管理者が従来のパーミッション概念の制限を回避するために多くの仕掛けを施す必要がありました。

ACLは、従来のファイルパーミッション概念を拡張する必要がある場合に使用できます。ACLを使用すれば、パーミッションが元の所有者や所有者の所属グループに対応していない場合でも個々のユーザまたはグループにそうしたパーミッションを割り当てることができます。アクセス制御リストは、Linuxカーネルの機能であり、現在ReiserFS、Ext2、Ext3、JFS、およびXFSでサポートされています。ACLを使用すると、アプリケーションレベルで複雑なパーミッションモデルを実装しなくても複雑なシナリオを実現できます。

ACLの利点は、WindowsサーバをLinuxサーバに置き換えるような場合にはっきりします。接続した一部のワークステーションは、移行後も引き続きWindowsの下で動作できます。Linuxシステムは、Sambaを搭載したWindowsクライアントにファイルサービスと印刷サービスを提供します。Sambaがアクセス制御リストをサポートしている場合は、LinuxサーバおよびWindows(Windows NT以降のみ)のどちらでもグラフィカルユーザインタフェースでユーザパーミッションを設定できます。winbinddを使用すれば、Linuxサーバ上にアカウントのない、Windowsドメインにしか存在していないユーザにパーミッションを割り当てることもできます。

## 35.2 定義

**ユーザクラス** 従来のPOSIXのパーミッション概念では、ファイルシステムでパーミッションを割り当てるための3つのユーザクラス(所有者、所有者の所属グループ、その他のユーザ)が使用されます。読み取り(r)、書き込み(w)、および実行(x)を可能にする3つのパーミッションビットは、ユーザクラスごとに設定できます。

**アクセスACL** あらゆる種類のファイルシステムオブジェクト(ファイルやディレクトリ)のユーザアクセスパーミッションとグループアクセスパーミッションは、アクセスACLによって決定されます。

**デフォルトACL** デフォルトACLは、ディレクトリにしか適用できません。このACLでは、ファイルシステムオブジェクトが作成されたときにその親ディレクトリから継承するパーミッションが決定されます。

**ACLエントリ** 各ACLは、ACLエントリセットから成ります。ACLエントリには、タイプ(表 35.1.「ACLエントリタイプ」を参照)、エントリが参照するユーザまたはグループのクォリファイア、およびパーミッションセットが含まれます。一部のエントリタイプの場合、グループまたはユーザのクォリファイアは定義されていません。

## 35.3 ACLの処理

表 35.1.「ACLエントリタイプ」に、考えられる6つのACLエントリタイプをまとめています。各エントリで、ユーザまたはユーザグループのパーミッションが定義されます。*所有者*エントリでは、ファイルまたはディレクトリを所有しているユーザのパーミッションが定義されます。*所有者の所属グループ*エントリでは、ファイルの所有者の所属グループのパーミッションが定義されます。スーパーユーザは、chownまたはchgrpを使用して所有者または所有者の所属グループを変更できます。その場合、所有者と所有者の所属グループエントリは、新しい所有者と所有者の所属グループを参照します。各*名前付き*ユーザエントリでは、エントリのクォリファイアフィールドで指定されたユーザのパーミッションが定義されます。クォリファイアフィールドとは、表 35.1.「ACLエントリタイプ」に示すテキスト書式の中央のフィールドのことです。各*名前付き*グループエントリでは、エントリのクォリファイアフィールドで指定されたグループのパーミッションが定義されます。名前付きユーザと名前付きグループのエントリのクォリファイアフィールドだけが指定されます。その*他*のエントリでは、他のすべてのユーザのパーミッションが定義されます。

マスクエントリでは、*名前付きユーザ*、*名前付きグループ*、および*所有者の所属グループ*のエントリで与えられたパーミッションをさらに制限するために、それらのエントリのパーミッションのどれが有効で、どれをマスクするかが定義されます。パーミッションは、マスク内と同様にこのいずれかのエントリ内にも存在する場合に有効です。マスクだけまたは実際のエントリだけに指定されているパーミッションは有効ではありません。つまり、パーミッションは与えられません。*所有者*と*所有者の所属グループ*のエントリで定義されているすべてのパーミッションは常に有効です。表 35.2. 「アクセスパーミッションのマスキング」の例に、このメカニズムを示しています。

基本的なACLクラスには、次の2つがあります。*最小*ACLには、*所有者*、*所有者の所属グループ*、および*その他*というタイプのエントリだけが含まれます。これらのエントリは、ファイルやディレクトリの従来のパーミッションビットに対応しています。*拡張*ACLは、このACLを越えるものです。このACLには、マスクエントリが必ず含まれ、*名前付きユーザ*と*名前付きグループ*のタイプのエントリがいくつか含まれている場合があります。

***Table 35.1:*** *ACLエントリタイプ*

| タイプ | テキスト書式 |
|---|---|
| 所有者 | `user::rwx` |
| 名前付きユーザ | `user:name:rwx` |
| 所有者の所属グループ | `group::rwx` |
| 名前付きグループ | `group:name:rwx` |
| マスク | `mask::rwx` |
| その他 | `other::rwx` |

***Table 35.2:*** *アクセスパーミッションのマスキング*

| エントリタイプ | テキスト書式 | パーミッション |
|---|---|---|
| 名前付きユーザ | `user:geeko:r-x` | `r-x` |
| マスク | `mask::rw-` | `rw-` |
| | 有効なパーミッション: | `r--` |

### 35.3.1 ACLエントリとファイルモードのパーミッションビット

図 35.1. 「最小ACL: ACLエントリとパーミッションビットとの比較」と図 35.2. 「拡張ACL: ACLエントリとパーミッションビットとの比較」に、最小ACLと拡張ACLの2つのケースを示しています。図は、3つのブロックから成ります。左側のブロックはACLエントリのタイプを示し、中央のブロックはACLの例を示しています。右側のブロックは、従来のパーミッション概念に基くそれぞれのパーミッションビット(ls -lなどで表示される)を示します。どちらのケースも、*所有者*クラスのパーミッションは、ACLエントリ*所有者*に割り当てられます。*その他*のクラスのパーミッションは、それぞれのACLエントリに割り当てられます。しかし、*グループ*クラスのパーミッションの割り当ては、2つのケースで異なります。

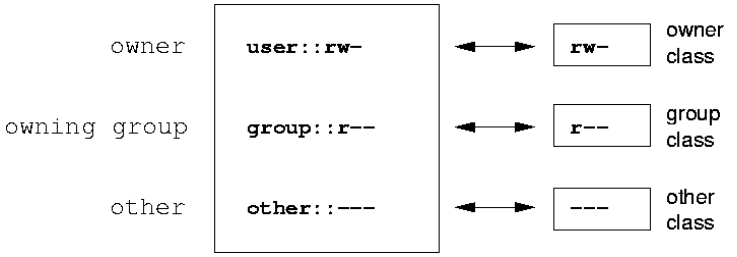


***Figure 35.1: 最小ACL: ACLエントリとパーミッションビットとの比較***

最小ACL(マスクなし)の場合、*グループ*クラスのパーミッションは、ACLエントリ*所有者の所属グループ*に割り当てられます。このようすを図 35.1. 「最小ACL: ACLエントリとパーミッションビットとの比較」に示しています。拡張ACL(マスクあり)の場合、*グループ*クラスのパーミッションは、マスクエントリに割り当てられます。このようすを図 35.2. 「拡張ACL: ACLエントリとパーミッションビットとの比較」に示しています。

この割り当て方法により、アプリケーションがACLをサポートしているかどうかにかかわらず、アプリケーションとのスムーズなインタラクションができます。パーミッションビットによって割り当てられたアクセスパーミッションは、ACLで他のすべてのパーミッションを「微調整」する場合の上限を表します。パーミッションビットの変更は、ACLに反映されます。その逆も同様です。

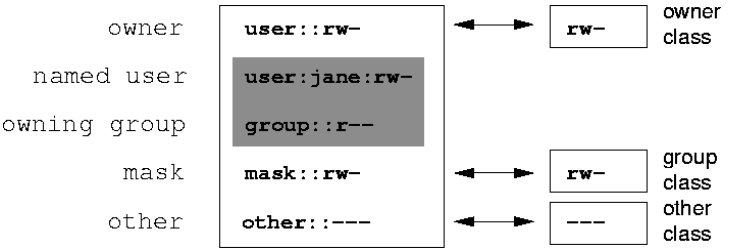


*Figure 35.2: 拡張ACL: ACLエントリとパーミッションビットとの比較*

### 35.3.2 アクセスACLが設定されたディレクトリ

アクセスACLの処理については、次の例に示します。

ディレクトリを作成する前に、umaskコマンドを使用して、ファイルオブジェクトを作成するたびにどのアクセスパーミッションをマスクする必要があるかを定義します。コマンドumask 027では、デフォルトパーミッションを設定するために、所有者にすべてのパーミッションを与え(0)、グループ書き込みアクセスを拒否して(2)、その他のユーザにはパーミッションを与えません(7)。umaskは、実際に対応するパーミッションビットをマスクするか、それらをオフにします。詳細については、対応するmanページ(man umask)を参照してください。

mkdir mydirは、umaskで設定されたデフォルトパーミッションでmydirディレクトリを作成する必要があります。ls -dl mydirを使用して、すべてのパーミッションが正しく割り当てられたかどうかをチェックします。このコマンドの出力例は、次のとおりです。

```
drwxr---- ... tux project3 ... mydir
```

getfacl mydirでは、ACLの初期状態をチェックします。このコマンドでは、次のような情報が得れられます。

```
# file: mydir
# owner; tux
# group: project3
user::rwx
group::r-x
other::---
```

getfaclの出力には、項35.3.1. 「ACLエントリとファイルモードのパーミッションビット」で説明したパーミッションビットの割り当てとACLエントリが正確に反映されます。最初の3つの出力行には、名前、所有者、およびディレクトリの所有者の所属グループが表示されています。次の3行には、*所有者*、*所有者の所属グループ*、および*その他*という3つのACLエントリが表示されています。実際には、この最小ACLの場合、getfaclコマンドではlsで取得できなかった情報は生成されません。

読み取り、書き込み、実行の各パーミッションをさらにユーザgekoとグループmascotsに割り当てるには、次のようにします。

```
setfacl -m user:geeko:rwx,group:mascots:rwx mydir
```

オプション-mを指定すると、setfaclに対して既存のACLの変更が求められます。このオプションの後の引き数は、変更するACLエントリを示します(複数のエントリはカンマで区切られます)。最後の部分には、こうした変更を適用するディレクトリの名前を指定します。設定されたACLを確認するには、getfaclコマンドを使用します。

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other::---
```

ユーザgekoとグループmascots向けのエントリのほかに、マスクエントリが生成されました。このマスクエントリは、すべてのパーミッションが有効になるように自動的に設定されます。setfaclは、既存のマスクエントリを変更済みの設定に自動的に合わせます。ただし、-nを指定してこの機能を無効にした場合は除きます。マスクでは、*グループクラス*のすべてのエントリに対する最大限に有効なアクセスパーミッションが定義されます。こうしたエントリには、*名前付きユーザ*、*名前付きグループ*、および*所有者の所属グループ*があります。ls -dl mydirで表示された*グループクラス*のパーミッションビットは、maskエントリに対応しています。

```
drwxrwx---+ ... tux project3 ... mydir
```

出力の最初のカラムには、この項目に*拡張*ACLがあることを示すためにさらに+が表示されます。

lsコマンドの出力に従って、マスクエントリのパーミッションには書き込みアクセスが追加されています。従来どおり、そのようなパーミッションビットは、*所有者の所属グループ*(ここではproject3)もディレクトリmydirに書き込みアクセスできることを表します。ただし、*所有者の所属グループ*の有効なアクセスパーミッションは、*所有者の所属グループ*向けおよびマスク用に定義されたパーミッションの重複部分に相当します。この部分は、この例ではr-xです(表 35.2.「アクセスパーミッションのマスキング」を参照)。この例の*所有者の所属グループ*の有効なパーミッションに関する限り、ACLエントリを追加した後も何も変わりませんでした。

マスクエントリを編集するには、setfaclまたはchmodを使用します。たとえば、chmod g-w mydirを使用すると、ls -dl mydirでは、次のように表示されます。

```
drwxr-x---+ ... tux project3 ... mydir
```

getfacl mydirでは、次の出力が得られます。

```
# file: mydir
# owner: tux
# group:project3
user::rwx
user:geeko:rwx        # effective: r-x
group::r-x
group:mascots:rwx     # effective: r-x
mask::r-x
other::---
```

chmodコマンドを実行して*グループ*クラスビットから書き込みパーミッションを削除した後に、lsコマンドの出力を見れば、マスクビットが相応に変更されている、つまり、書き込みパーミッションが再びmydirの所有者に制限されていることを十分に確認できます。getfaclの出力でこの確認を行います。この出力に、有効なパーミッションビットが元のパーミッションと一致しないすべてのエントリのコメントが含まれる理由は、それらのビットがマスクエントリに基いてフィルタ処理されるためです。chmod g+w mydirを使用すれば、いつでも元のパーミッションに戻すことができます。

### 35.3.3 デフォルトACLが設定されたディレクトリ

ディレクトリには、デフォルトACLを設定できます。デフォルトACLとは、ディレクトリのオブジェクトを作成するときにそうしたオブジェクトが継承するアクセスパーミッションを定義する特別な種類のACLのことです。デフォルトACLは、サブディレクトリとファイルに作用します。

#### デフォルトACLの作用

ディレクトリのデフォルトACLのパーミッションをそのディレクトリ内のファイルやサブディレクトリに渡す方法は、次の2種類があります。

- サブディレクトリは、そのデフォルトACLおよびアクセスACLとして親ディレクトリのデフォルトACLを継承します。
- ファイルは、そのアクセスACLとしてデフォルトACLを継承します。

ファイルシステムオブジェクトを作成するすべてのシステムコールは、新たに作成したファイルシステムオブジェクトのアクセスパーミッションを定義するmodeパラメータを使用します。親ディレクトリにデフォルトACLが設定されていない場合、umaskで定義されたパーミッションビットは、modeパラメータで渡されるパーミッションから取り去られ、その結果が新しいオブジェクトに割り当てられます。親ディレクトリのデフォルトACLが存在する場合、新しいオブジェクトに割り当てられるパーミッションビットは、modeパラメータのパーミッションとデフォルトACLで定義されているパーミッションの重複部分に相当します。この場合、umaskは無視されます。

#### デフォルトACLのアプリケーション

次の3つの例は、ディレクトリとデフォルトACLの主要な操作を示しています。

1. 次のコマンドで、既存のディレクトリmydirにデフォルトACLを追加します。

   ```
   setfacl -d -m group:mascots:r-x mydir
   ```

   setfaclコマンドのオプション-dを指定することによって、setfaclは、後続の変更(オプション-m)をデフォルトACLに加えるように求められます。

   このコマンドの結果を詳しく見てみます。

```
getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other::---
```

getfaclは、アクセスACLとデフォルトACLを返します。デフォルトACLは、defaultで始まるすべての行によって生成されます。デフォルトACLのmascotsグループのエントリでsetfaclコマンドを実行しただけですが、setfaclで他のすべてのエントリが自動的にアクセスACLからコピーされ、有効なデフォルトACLが作成されました。デフォルトACLが、アクセスパーミッションに即時に作用することはありません。デフォルトACLは、ファイルシステムオブジェクトが作成された場合にのみ作用し始めます。こうした新しいオブジェクトは、それぞれの親ディレクトリのデフォルトACLからのみパーミッションを継承します。

2. 次の例では、mkdirでmydirにサブディレクトリを作成しています。このサブディレクトリは、デフォルトACLを継承します。

```
mkdir mydir/mysubdir

getfacl mydir/mysubdir

# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx group::r-x
group:mascots:r-x
mask::r-x
other::---
default:user::rwx
```

```
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other::---
```

予想どおり、新たに作成されたサブディレクトリmysubdirには、親ディレクトリのデフォルトACLからのパーミッションが設定されています。mysubdirのアクセスACLは、mydirのデフォルトACLを正確に反映しています。このディレクトリからその下位オブジェクトにも同じデフォルトACLが継承されます。

3. touchコマンド(touch mydir/myfileなど)でmydirディレクトリにファイルを作成します。次に、ls -l mydir/myfileを実行すると、次のように表示されます。

```
-rw-r----+ ... tux project3 ... mydir/myfile
```

getfacl mydir/myfileの出力は、次のようになります。

```
# file: mydir/myfile
# owner: tux
# group: project3
user::rw-
group::r-x          # effective:r--
group:mascots:r-x   # effective:r--
mask::r--
other::---
```

touchコマンドは、新しいファイルの作成時に値0666を指定してmodeを使用します。この値は、umaskおよびデフォルトACLにほかに制限がなければ、すべてのユーザクラスのファイルが読み取りと書き込みのパーミッションで作成されることを表します(項35.3.3. 「デフォルトACLの作用」を参照)。つまり、mode値に含まれていないアクセスパーミッションはすべて、それぞれのACLエントリから削除されます。パーミッションは、グループクラスのACLエントリから削除されていませんが、マスクエントリは、modeで設定されていないパーミッションをマスクするように変更されています。

この方法により、ACLが設定されたアプリケーション(コンパイラなど)とのスムーズなインタラクションができます。制限付きアクセスパーミッションでファイルを作成し、作成したファイルを後で実行可能ファイルとしてマークすることができます。maskメカニズムでは、正当なユーザやグループが必要に応じて実行可能ファイルを実行できることが保証されます。

### 35.3.4 ACLチェックアルゴリズム

任意のプロセスまたはアプリケーションがACL保護されたファイルシステムオブジェクトにアクセスできるようになる前に、チェックアルゴリズムが適用されます。基本的なルールとして、ACLエントリは、*所有者*、*名前付きユーザ*、*所有者の所属グループ*または*名前付きグループ*、および*その他*の順に調べられます。アクセスは、プロセスに最も適したエントリに従って処理されます。パーミッションは累積しません。

プロセスが複数のグループに属し、複数のグループエントリに適する可能性がある場合は、さらに複雑になります。エントリは、必要なパーミッションを備えた適切なエントリから無作為に選択されます。どのエントリによって最終結果「アクセス許可」が実行されるかには関係ありません。同様に、適切なグループエントリのどれにも必要なパーミッションが設定されていない場合は、無作為に選択されたエントリによって最終結果「アクセス拒否」が実行されます。

## 35.4 アプリケーションでのACLサポート

ACLを使用すれば、最新のアプリケーションの要件を満たす非常に複雑なパーミッションシナリオを実現できます。従来のパーミッション概念とACLは、洗練された方法で組み合わせることができます。基本的なファイルコマンド(`cp`、`mv`、`ls`など)では、ACLをサポートします。Sambaでも同様です。

残念ながら、多くのエディタやファイルマネージャでは、依然としてACLをサポートしていません。たとえば、Konquerorでファイルをコピーすると、ファイルのACLは失われます。エディタでファイルを変更すると、使用するエディタのバックアップモードによっては、ファイルのACLが維持されるときもあれば、維持されないときもあります。エディタが元のファイルに変更を書き込む場合、アクセスACLは維持されます。エディタで更新内容を新しいファイルに保存し、そのファイルの名前を後で古いファイル名に変更しても、ACLは失われるおそれがあります。ただし、エディタがACLをサポートしている場合は除きます。starアーカイバ以外に、ACLを維持するバックアップアプリケーションは現在ありません。

## 35.5 詳細情報

ACLの詳細については、`http://acl.bestbits.at/`を参照してください。`getfacl(1)`、`acl(5)`、および`setfacl(1)`については、manページも参

照してください。

# 35 Linuxのアクセス制御リスト

# 36



# システムモニタリングユーティリティ

システムのステータスは、多数のプログラムやメカニズムを使用して検査できます。ここではその一部について説明します。また、日常作業に役立つ一部のユーティリティとその最も重要なパラメータについても説明します。

ここでは、コマンドごとに関連出力の例を示してあります。これらの例の1行目はコマンド自体です(ドル記号プロンプトの後)。コメントは、大カッコ[...]で示されており、長い行は必要に応じて折り返されています。長い行の改行はバックスラッシュ(\)で示されています。

```
$ command -x -y
output line 1
output line 2
output line 3 この行は少し長いので\
    次のように分割します。
output line 3
[...]
output line 98
output line 99
```

できるだけ多数のユーティリティを紹介できるように、簡潔に説明しています。すべてのコマンドの詳細は、マニュアルページで確認できます。また、ほとんどのコマンドではパラメータ--helpが認識されます。このパラメータを指定すると、使用可能なパラメータの簡略リストが表示されます。

## 36.1 開いているファイルのリスト:lsof

プロセスIDが⟨*PID*⟩のプロセスについて開いている全ファイルのリストを表示するには、-pを使用します。たとえば、現行のシェルで使用されている全ファイルを表示するには、次のように入力します。

```
$ lsof -p $$
COMMAND  PID USER   FD   TYPE DEVICE    SIZE     NODE NAME
zsh     4694   jj  cwd    DIR   0,18     144 25487368 /suse/jj/t (totan:/real-home/jj)
zsh     4694   jj  rtd    DIR    3,2     608        2 /
zsh     4694   jj  txt    REG    3,2  441296    20414 /bin/zsh
zsh     4694   jj  mem    REG    3,2  104484    10882 /lib/ld-2.3.3.so
zsh     4694   jj  mem    REG    3,2   11648    20610 /usr/lib/zsh/4.2.0/zsh/rlimits.so [...]
zsh     4694   jj  mem    REG    3,2   13647    10891 /lib/libdl.so.2
zsh     4694   jj  mem    REG    3,2   88036    10894 /lib/libnsl.so.1
zsh     4694   jj  mem    REG    3,2  316410   147725 /lib/libncurses.so.5.4
zsh     4694   jj  mem    REG    3,2  170563    10909 /lib/tls/libm.so.6
zsh     4694   jj  mem    REG    3,2 1349081    10908 /lib/tls/libc.so.6
zsh     4694   jj  mem    REG    3,2      56    12410 /usr/lib/locale/de_DE.utf8/LC_TELEPHONE [...]
zsh     4694   jj  mem    REG    3,2      59    14393 /usr/lib/locale/en_US/LC_NUMERIC
zsh     4694   jj  mem    REG    3,2  178476    14565 /usr/lib/locale/en_US/LC_CTYPE
zsh     4694   jj  mem    REG    3,2   56444    20598 /usr/lib/zsh/4.2.0/zsh/computil.so
zsh     4694   jj    0u   CHR 136,48               50 /dev/pts/48
zsh     4694   jj    1u   CHR 136,48               50 /dev/pts/48
zsh     4694   jj    2u   CHR 136,48               50 /dev/pts/48
zsh     4694   jj   10u   CHR 136,48               50 /dev/pts/48
```

この例では、値としてシェルのプロセスIDをとる特殊なシェル変数$$が使用されています。

パラメータを指定せずにコマンドlsofを入力すると、現在開いている全ファイルがリストされます。開いているファイルの数が何千にも達することがあるので、そのすべてをリストすることはほとんど無意味です。ただし、開いているすべてのファイルのリストを検索機能と組み合わせて使用すると、役立つリストが生成されます。たとえば、次のように使用されているすべてのキャラクタデバイスのリストを表示します。

```
$ lsof | grep CHR
sshd       4685    root  mem    CHR    1,5            45833 /dev/zero
sshd       4685    root  mem    CHR    1,5            45833 /dev/zero
sshd       4693      jj  mem    CHR    1,5            45833 /dev/zero
sshd       4693      jj  mem    CHR    1,5            45833 /dev/zero
zsh        4694      jj    0u   CHR 136,48               50 /dev/pts/48
zsh        4694      jj    1u   CHR 136,48               50 /dev/pts/48
zsh        4694      jj    2u   CHR 136,48               50 /dev/pts/48
zsh        4694      jj   10u   CHR 136,48               50 /dev/pts/48
X          6476    root  mem    CHR    1,1            38042 /dev/mem
lsof      13478      jj    0u   CHR 136,48               50 /dev/pts/48
lsof      13478      jj    2u   CHR 136,48               50 /dev/pts/48
grep      13480      jj    1u   CHR 136,48               50 /dev/pts/48
grep      13480      jj    2u   CHR 136,48               50 /dev/pts/48
```

## 36.2 ファイルにアクセス中のユーザ:fuser

現在一定のファイルにアクセスしているプロセスまたはユーザを判別しておくことは有効です。たとえば、/mntにマウントされているファイルシステムをアンマウントするとします。umountでは、「デバイスがビジー」状態が返されます。ここで次のようにコマンドfuserを使用すると、デバイスにアクセスしているプロセスを判断することができます。

```
$ fuser -v /mnt/*

         USER        PID ACCESS COMMAND
/mnt/notes.txt
         jj        26597 f....  less
```

別の端末で実行中であったlessプロセスの終了後は、ファイルシステムを正常にアンマウントできます。

## 36.3 ファイルのプロパティ:stat

コマンドstatは、ファイルのプロパティを表示します。

```
$ stat xml-doc.txt
  File:‘xml-doc.txt’
  Size:632             Blocks:8          IO Block:4096   regular file
  Device:eh/14d  Inode:5938009     Links:1
  Access:(0644/-rw-r--r--)  Uid:(11994/      jj)   Gid:(   50/    suse)
  Access:2004-04-27 20:08:58.000000000 +0200
  Modify:2003-06-03 15:29:34.000000000 +0200
  Change: 2003-07-23 17:48:27.000000000 +0200
```

パラメータ--filesystemを指定すると、指定したファイルが置かれているファイルシステムのプロパティの詳細が出力されます。

```
$ stat  .--filesystem
File:"."
  ID0        Namelen:255     Type:ext2/ext3
  Blocks:Total:19347388   Free:17831731   Available:16848938   Size:4096
  Inodes:Total:9830400   Free: 9663967
```

zシェル(zsh)を使用する場合、zシェルには異なるオプションと出力形式を使用するシェル内蔵statがあるため、/usr/bin/statと入力する必要があります。

```
% type stat
stat is a shell builtin
% stat .
device  769
inode   4554808
mode    16877
nlink   12
uid     11994
gid     50
rdev    0
size    4096
atime   1091536882
mtime   1091535740
ctime   1091535740
blksize 4096
blocks  8
link
```

## 36.4 USBデバイス:lsusb

コマンドlsusbは、すべてのUSBデバイスのリストを表示します。オプション-vを使用すると、詳細なリストが印刷されます。この詳細は、ディレクトリ/proc/bus/usb/から読み込まれます。次に示すものは、USBメモリスティックが取り付けられたあとのlsusbの出力です。最後のほうの行は、新規デバイスがあることを示します。

```
Bus 004 Device 001:ID 0000:0000
Bus 003 Device 001:ID 0000:0000
Bus 002 Device 001:ID 0000:0000
Bus 001 Device 001:ID 0000:0000
Bus 001 Device 018:ID 0402:5634 ALi Corp.
```

## 36.5 SCSIデバイスに関する情報:scsiinfo

コマンドscsiinfoは、SCSIデバイスに関する情報のリストを表示します。オプション-lを使用すると、システムに登録されているすべてのSCSIデバイスのリストが表示されます(同様の情報は、コマンドlsscsiでも入手できます)。次に示すものは、scsiinfo -i /dev/sdaの出力です。この場合、ハードディスクに関する情報が表示されます。オプション-aで、詳細が表示されます。

```
Inquiry command
---------------
Relative Address                    0
Wide bus 32                         0
Wide bus 16                         1
Synchronous neg.                    1
Linked Commands                     1
Command Queueing                    1
SftRe                               0
Device Type                         0
Peripheral Qualifier                0
Removable?0 Device Type Modifier                0
ISO Version                         0
ECMA Version                        0
ANSI Version                        3
AENC                                0
TrmIOP                              0
Response Data Format                2
Vendor:FUJITSU
Product:MAS3367NP
Revision level:0104A0K7P43002BE
```

ハードディスクの不良ブロックを示す次の2つのテーブルからなる欠陥リストがあります。1つはメーカによって提供されるもの(メーカテーブル)で、も

う1つは操作中に発生する不良ブロックのリスト(成長テーブル)です。成長テーブル内のエントリ数が増えた場合、ハードディスクを交換するほうが良いでしょう。

## 36.6 プロセス:top

コマンドtop ( "table of processes" =プロセステーブルを意味します)は、2秒間隔で更新されるプロセスリストを表示します。プログラムを終了するには、Qキーを押します。パラメータ-n 1を指定すると、プロセスリストが1回表示された後にプログラムが終了します。次に示すものは、コマンドtop -n 1の出力例です。

```
top - 14:19:53 up 62 days,  3:35, 14 users,  load average:0.01, 0.02, 0.00
Tasks:102 total,   7 running,  93 sleeping,   0 stopped,   2 zombie
Cpu(s):0.3% user,   0.1% system,   0.0% nice,  99.6% idle
Mem:514736k total,   497232k used,    17504k free,    56024k buffers
Swap:1794736k total,   104544k used,  1690192k free,   235872k cached

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  Command
1426 root      15   0  116m  41m  18m S  1.0  8.2  82:30.34 X
20836 jj        15   0   820  820  612 R  1.0  0.2   0:00.03 top
  1 root      15   0   100   96   72 S  0.0  0.0   0:08.43 init
  2 root      15   0     0    0    0 S  0.0  0.0   0:04.96 keventd
  3 root      34  19     0    0    0 S  0.0  0.0   0:00.99 ksoftirqd_CPU0
  4 root      15   0     0    0    0 S  0.0  0.0   0:33.63 kswapd
  5 root      15   0     0    0    0 S  0.0  0.0   0:00.71 bdflush
  [...]
  1362 root      15   0   488  452  404 S  0.0  0.1   0:00.02 nscd
  1363 root      15   0   488  452  404 S  0.0  0.1   0:00.04 nscd
  1377 root      17   0    56    4    4 S  0.0  0.0   0:00.00 mingetty
  1379 root      18   0    56    4    4 S  0.0  0.0   0:00.01 mingetty
  1380 root      18   0    56    4    4 S  0.0  0.0   0:00.01 mingetty
```

topの実行中にFキーを押すと、メニューが開き、出力形式を大幅に変更できます。

パラメータ-U UIDを指定すると、特定のユーザに関連したプロセスのみがモニタされます。⟨*UID*⟩は、ユーザのユーザIDに置き換えます。top -U $(id -u username)は、ユーザ名を基本としたユーザのUIDを返し、そのプロセスを表示します。

## 36.7 プロセスリスト:ps

コマンドpsは、プロセスのリストを作成します。パラメータrを追加すると、現在計算時間を使用しているプロセスだけが表示されます。

```
$ ps r
 PID TTY      STAT   TIME COMMAND
 22163 pts/7    R      0:01 -zsh
 3396 pts/3     R      0:03 emacs new-makedoc.txt
 20027 pts/7    R      0:25 emacs xml/common/utilities.xml
 20974 pts/7    R      0:01 emacs jj.xml
 27454 pts/7    R      0:00 ps r
```

このパラメータは、マイナス記号なしで指定する必要があります。さまざまなパラメータが、マイナス記号付きで指定されたり、マイナス記号なして指定されたりします。マニュアルページは潜在的ユーザを簡単に驚かせることもありますが、幸いなことに、ps --helpコマンドは簡単なヘルプページを作成します。

実行中のemacsプロセスの数をチェックするには、次のものを使用します。

```
$ ps x | grep emacs
  1288 ?S      0:07 emacs
  3396 pts/3    S      0:04 emacs new-makedoc.txt
  3475 ?S      0:03 emacs .Xresources
  20027 pts/7    S      0:40 emacs xml/common/utilities.xml
  20974 pts/7    S      0:02 emacs jj.xml

$ pidof emacs
 20974 20027 3475 3396 1288
```

パラメータ-pは、次のようにプロセスIDを通じてプロセスを選択します。

```
$ ps www -p $(pidof xterm)
  PID TTY      STAT   TIME COMMAND
  9025 ?S      0:01 xterm -g 100x45+0+200
  9176 ?S      0:00 xterm -g 100x45+0+200
  29854 ?S      0:21 xterm -g 100x75+20+0 -fn \
   -B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1
  4378 ?S      0:01 xterm -bg MistyRose1 -T root -n root -e su -l
  25543 ?S      0:02 xterm -g 100x45+0+200
  22161 ?R      0:14 xterm -g 100x45+0+200
  16832 ?S      0:01 xterm -bg MistyRose1 -T root -n root -e su -l
  16912 ?S      0:00 xterm -g 100x45+0+200
  17861 ?S      0:00 xterm -bg DarkSeaGreen1 -g 120x45+40+300
  19930 ?S      0:13 xterm -bg LightCyan
  21686 ?S      0:04 xterm -g 100x45+0+200 -fn \ lucidasanstypewriter-12
  23104 ?S      0:00 xterm -g 100x45+0+200
  26547 ?S      0:00 xterm -g 100x45+0+200
```

プロセスリストは、必要に応じてフォーマットできます。オプション-Lを指定すると、すべてのキーワードのリストが返されます。次のコマンドを入力すると、メモリ使用量順の全プロセスのリストが発行されます。

```
$ ps ax --format pid,rss,cmd --sort rss
PID  RSS CMD
 2    0 [ksoftirqd/0]
 3    0 [events/0]
17    0 [kblockd/0]
[...]
10164 5260 xterm
31110 5300 xterm
17010 5356 xterm
3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth /var/lib/xdm/authdir/au
```

## 36.8 プロセスツリー:pstree

pstreeコマンドは、プロセスリストをツリー形式で出力します。

```
$ pstree
init-+-atd
   |-3*[automount]
   |-bdflush |-cron
[...]
   |-usb-storage-1
   |-usb-storage-2
   |-10*[xterm---zsh]
   |-xterm---zsh---mutt
   |-2*[xterm---su---zsh]
   |-xterm---zsh---ssh
   |-xterm---zsh---pstree
   |-ypbind---ypbind---2*[ypbind]
   `-zsh---startx---xinit4-+-X
                           `-ctwm-+-xclock
                                  |-xload
                                  `-xosview.bin
```

パラメータ-pを指定すると、プロセス名にプロセスIDが追加されます。コマンドラインも表示させるには、-aパラメータを使用します。

```
$ pstree
```

```
-pa init,1
   |-atd,1255
[...]
   ‘-zsh,1404
    ‘-startx,1407 /usr/X11R6/bin/startx
      ‘-xinit4,1419 /suse/jj/.xinitrc [...]
         |-X,1426 :0 -auth /suse/jj/.Xauthority
         ‘-ctwm,1440
            |-xclock,1449 -d -geometry -0+0 -bg grey
            |-xload,1450 -scale 2
            ‘-xosview.bin,1451 +net -bat +net
```

## 36.9 実行者と実行内容:w

コマンドwを使用すると、システムにログオンしているユーザと、そのユーザが実行している操作を確認できます。次に例を示します。

```
$ w 15:17:26 up 62 days,  4:33, 14 users,  load average:0.00, 0.04, 0.01
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
jj       pts/0     30Mar04  4days  0.50s  0.54s xterm -e su -l
jj       pts/1     23Mar04  5days  0.20s  0.20s -zsh
jj       pts/2     23Mar04  5days  1.28s  1.28s -zsh
jj       pts/3     23Mar04  3:28m  3.21s  0.50s -zsh
[...]
jj       pts/7     07Apr04  0.00s  9.02s  0.01s w
jj       pts/9     25Mar04  3:24m  7.70s  7.38s mutt
[...]
jj       pts/14    12:49   37:34   0.20s  0.13s ssh totan
```

最後の行は、ユーザjjがコンピュータtotanへのセキュアシェル(ssh)接続を確立したことを示します。他のシステムのユーザがリモートログインしている場合は、パラメータ-fを指定すると、そのユーザがどのコンピュータから接続を確立したかが出力されます。

## 36.10 メモリの使用状況:free

ユーティリティfreeはRAMの使用状況を検査します。空きメモリと使用済みメモリ(およびスワップ領域)の両方について詳細が表示されます。

```
$ free
             total       used       free     shared    buffers     cached
Mem:514736     273964     240772          0      35920      42328
-/+ buffers/cache:195716     319020
Swap:      1794736     104096    1690640
```

-mを指定すると、すべてのサイズがMB単位で表されます。

```
$ free -m
                  total       used       free     shared    buffers     cached
Mem:502        267        235          0         35         41
-/+ buffers/cache:191        311
Swap:         1752        101       1651
```

実際に必要な情報は、次の行に含まれています。

```
-/+ buffers/cache:        191        311
```

ここでは、バッファとキャッシュで使用されているメモリの量が計算されます。パラメータ-d delayを指定すると、表示が⟨*delay*⟩秒間隔で確実に更新されます。たとえば、free -d 1.5と入力すると1.5秒ごとに更新されます。

## 36.11 カーネルリングバッファ:dmesg

Linuxカーネルは、リングバッファに一定のメッセージを保持します。これらのメッセージを表示するには、コマンドdmesgを入力します。

```
$ dmesg
[...]
sdc :READ CAPACITY failed.
sdc :status = 1, message = 00, host = 0, driver = 08
Info fld=0xa00 (nonstd), Current sd00:00:sense key Not Ready
sdc :block size assumed to be 512 bytes, disk size 1GB. sdc:test WP failed, assume Write Enable
sdc:I/O error:dev 08:20, sector 0
I/O error:dev 08:20, sector 0
I/O error:dev 08:20, sector 2097144
I/O error:dev 08:20, sector 2097144
I/O error:dev 08:20, sector 0
I/O error:dev 08:20, sector 0 unable to read partition table
I/O error:dev 08:20, sector 0 nfs:server totan not responding, still trying nfs:server totan OK
```

最終行は、NFSサーバtotanに一時的な問題が発生していたことを示しています。ここまでの行は、USBフラッシュドライブの挿入によって発生しています。古いイベントは、ファイル/var/log/messagesおよび/var/log/warnに記録されています。

## 36.12 ファイルシステムと使用状況:mount、df、およびdu

コマンドmountは、どのファイルシステム(デバイスとタイプ)がどのマウントポイントにマウントされているかを出力します。

```
$ mount
/dev/hdb2 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /data type ext2 (rw)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
automount(pid1012) on /suse type autofs \
  (rw,fd=5,pgrp=1012,minproto=2,maxproto=3)
totan:/real-home/jj on /suse/jj type nfs \
  (rw,nosuid,rsize=8192,wsize=8192,hard,intr,nolock,addr=10.10.0.1)
```

コマンドdfを使用して、ファイルシステムの使用状況に関する合計情報を入手してください。パラメータ-h(または--human-readable)を指定すると、出力は通常のユーザが理解できる形式に変換されます。

```
$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hdb2             7.4G  5.1G  2.0G  73% /
/dev/hda1              74G  5.8G   65G   9%
/data shmfs                  252M     0  252M   0% /dev/shm
totan:/real-home/jj   350G  324G   27G  93% /suse/jj
```

NFSファイルサーバtotanのユーザは、ホームディレクトリを遅延なしで空にする必要があります。指定したディレクトリとそのサブディレクトリの全ファイルの合計サイズを表示するには、コマンドduを使用します。パラメータ-sを指定すると、詳細情報は出力されません。-hを指定すると、データは通常のユーザが理解できる形式に再び変換されます。次のコマンドを使用したとします。

```
$ du -sh ~
361M    /suse/jj
```

自分のホームディレクトリに使用されている容量が表示されます。

## 36.13 /procファイルシステム

/procファイルシステムは、カーネルにより重要な情報が仮想ファイルの形式で保持される疑似ファイルシステムです。たとえば、次のコマンドを使用すると、CPUのタイプを確認できます。

```
$ cat /proc/cpuinfo
processor       :0
vendor_id       :AuthenticAMD
cpu family      :6
model           :8
model name      :AMD Athlon(tm) XP 2400+
stepping        :1
cpu MHz         :2009.343
cache size      :256 KB
fdiv_bug        :no
[...]
```

割り込みの割り当てと使用は、次のコマンドでクエリできます。

```
$ cat /proc/interrupts
      CPU0
 0:537544462          XT-PIC  timer
 1:820082          XT-PIC  keyboard
 2:0          XT-PIC  cascade
 8:2          XT-PIC  rtc
 9:0          XT-PIC  acpi
 10:13970          XT-PIC  usb-uhci, usb-uhci
 11:146467509          XT-PIC  ehci_hcd, usb-uhci, eth0
 12:8061393          XT-PIC  PS/2 Mouse
 14:2465743          XT-PIC  ide0
 15:1355          XT-PIC  ide1
 NMI:0
 LOC:0
 ERR:0
 MIS:0
```

重要なファイルとその内容の一部は次のとおりです。

**/proc/devices** 使用可能なデバイス

**/proc/modules** ロードされたカーネルモジュール

**/proc/cmdline** カーネルコマンドライン

**/proc/meminfo** メモリ使用状況に関する詳細情報

**/proc/config.gz** gzip-現在実行中のカーネルの圧縮設定ファイル

詳細は、テキストファイル/usr/src/linux/Documentation/filesystems/proc.txtにあります。現在実行中のプロセスについては、/proc/⟨*NNN*⟩ディレクトリで確認できます。この場合、⟨*NNN*⟩は関連プロセスのプロセスID (PID)です。/proc/self/を指定すると、プロセスとその特有の特性を確認できます。

```
$ ls -l /proc/self
lrwxrwxrwx  1 root root 64 Apr 29 13:52 /proc/self -> 27585

$ ls -l /proc/self/
total 0
dr-xr-xr-x  2 jj suse 0 Apr 29 13:52 attr
-r--------  1 jj suse 0 Apr 29 13:52 auxv
-r--r--r--  1 jj suse 0 Apr 29 13:52 cmdline
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 cwd -> /suse/jj/t
-r--r--r--  1 jj suse 0 Apr 29 13:52 delay
-r--------  1 jj suse 0 Apr 29 13:52 environ
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
dr-x------  2 jj suse 0 Apr 29 13:52 fd
-rw-------  1 jj suse 0 Apr 29 13:52 mapped_base
-r--r--r--  1 jj suse 0 Apr 29 13:52 maps
-rw-------  1 jj suse 0 Apr 29 13:52 mem
-r--r--r--  1 jj suse 0 Apr 29 13:52 mounts
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 root -> /
-r--r--r--  1 jj suse 0 Apr 29 13:52 stat
-r--r--r--  1 jj suse 0 Apr 29 13:52 statm
-r--r--r--  1 jj suse 0 Apr 29 13:52 status
dr-xr-xr-x  3 jj suse 0 Apr 29 13:52 task
-r--r--r--  1 jj suse 0 Apr 29 13:52 wchan
```

実行可能ファイルとライブラリのアドレス割り当ては、mapsファイルに含まれています。

```
$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 22890      /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 22890      /bin/cat
```

```
0804d000-0806e000 rwxp 0804d000 00:00 0
40000000-40016000 r-xp 00000000 03:02 10882      /lib/ld-2.3.3.so
40016000-40017000 rw-p 00015000 03:02 10882      /lib/ld-2.3.3.so
40017000-40018000 rw-p 40017000 00:00 0
4002b000-40135000 r-xp 00000000 03:02 10908      /lib/tls/libc.so.6
40135000-4013d000 rw-p 0010a000 03:02 10908      /lib/tls/libc.so.6
4013d000-40141000 rw-p 4013d000 00:00 0
bfffe000-c0000000 rw-p bfffe000 00:00 0
ffffe000-fffff000 ---p 00000000 00:00 0
```

## 36.14 vmstat、iostat、およびmpstat

ユーティリティvmstatは、仮想メモリ統計を報告します。このユーティリティは、ファイル/proc/meminfo、/proc/stat、および/proc/*/statを読み取ります。これは、システムのパフォーマンスの障害を判別するのに役立ちます。

コマンドiostatは、CPUとデバイスおよびパーティションでの入出力に関する統計を報告します。表示される情報は、ファイル/proc/statおよび/proc/partitionsから取られます。この出力は、ハードディスク間の入出力バランスを改善するために使用できます。コマンドmpstatは、CPU関連の統計を報告します。

## 36.15 procinfo

/procファイルシステムからの重要情報のサマリを確認するには、コマンドprocinfoを使用します。

```
$ procinfo
Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU [roth.suse.de]

Memory:Total        Used        Free      Shared     Buffers
Mem:516696      513200        3496           0       43284
Swap:       530136        1352      528784

Bootup:Wed Jul  7 14:29:08 2004    Load average: 0.07 0.04 0.01 1/126 5302

user  :2:42:28.08   1.3%  page in :0
nice  :0:31:57.13   0.2%  page out:0
system:0:38:32.23   0.3%  swap in :0
idle  :3d 19:26:05.93  97.7%  swap out:0
uptime:4d  0:22:25.84         context :207939498

irq  0:776561217 timer                 irq  8:2 rtc
```

```
irq  1:276048 i8042                irq  9:24300 VIA8233
irq  2:0 cascade [4]          irq 11:38610118 acpi, eth0, uhci_hcd
irq  3:3                      irq 12:3435071 i8042
irq  4:3                      irq 14:2236471 ide0
irq  6:2                      irq 15:251 ide1
```

すべての情報を表示するには、パラメータ-aを使用します。パラメータ-nNを指定すると、情報が⟨N⟩秒間隔で更新されます。この場合、プログラムを終了するには[Q]キーを押します。

デフォルトでは、累積値が表示されます。パラメータ-dを入力すると、別の値が作成されます。procinfo -dn5を入力すると、過去5秒間に変化した値が表示されます。

```
Memory:Total        Used        Free      Shared     Buffers      Cached
Mem:0           2          -2           0           0           0
Swap:            0           0           0

Bootup:Wed Feb 25 09:44:17 2004    Load average: 0.00 0.00 0.00 1/106 31902

user  :0:00:00.02   0.4%  page in :0  disk 1:0r       0w
nice  :0:00:00.00   0.0%  page out:0  disk 2:0r       0w
system:0:00:00.00   0.0%  swap in :0  disk 3:0r       0w
nice  :0:00:04.99  99.6%  swap out:0  disk 4:0r       0w
uptime:64d  3:59:12.62         context :     1087

irq  0:501 timer                irq 10:0 usb-uhci, usb-uhci
irq  1:1 keyboard              irq 11:32 ehci_hcd, usb-uhci,
irq  2:0 cascade [4]           irq 12:132 PS/2 Mouse
irq  6:0                       irq 14:0 ide0
irq  8:0 rtc                   irq 15:0 ide1 irq  9:0 acpi
```

# 36.16 PCI リソース:lspci

コマンドlspciはPCIリソースをリストします。

```
$ lspci
00:00.0 Host bridge:VIA Technologies, Inc. \
   VT8366/A/7 [Apollo KT266/A/333]
00:01.0 PCI bridge:VIA Technologies, Inc. \
   VT8366/A/7 [Apollo KT266/A/333 AGP]
00:0b.0 Ethernet controller:Digital Equipment Corporation \ DECchip 21140 [FasterNet] (rev 22)
00:10.0 USB Controller:VIA Technologies, Inc. USB (rev 80)
00:10.1 USB Controller:VIA Technologies, Inc. USB (rev 80)
00:10.2 USB Controller:VIA Technologies, Inc. USB (rev 80)
00:10.3 USB Controller:VIA Technologies, Inc. USB 2.0 (rev 82)
00:11.0 ISA bridge:VIA Technologies, Inc. VT8235 ISA Bridge
00:11.1 IDE interface:VIA Technologies, Inc. VT82C586/B/686A/B \
   PIPC Bus Master IDE (rev 06)
00:11.5 Multimedia audio controller:VIA Technologies, Inc. \ VT8233 AC97 Audio Controller (rev 50)
01:00.0 VGA compatible controller:Matrox Graphics, Inc. \ MGA G550 AGP (rev 01)
```

-vを使用すると、さらに詳細なリストが出力されます。

```
$ lspci -v
[...]
01:00.0 \
 VGA compatible controller:Matrox Graphics, Inc. MGA G550 AGP (rev 01) \
   (prog-if 00 [VGA])
Subsystem:Matrox Graphics, Inc. Millennium G550 Dual Head DDR 32Mb Flags:bus master, medium devsel, latency
Memory at d8000000 (32-bit, prefetchable) [size=32M]
Memory at da000000 (32-bit, non-prefetchable) [size=16K]
Memory at db000000 (32-bit, non-prefetchable) [size=8M]
Expansion ROM at <unassigned> [disabled] [size=128K]
Capabilities:<available only to root>
```

デバイス名の解決に関する情報は、ファイル/usr/share/pci.idsから取得されます。このファイルにないPCI IDは、“Unknown device”で示されます。

パラメータ-vvを指定すると、プログラムが問い合わせ可能な情報がすべて出力されます。数値のみを表示するには、パラメータ-nを指定する必要があります。

## 36.17 実行中のプログラムのシステム呼び出し:strace

ユーティリティstraceを使用すると、現在実行中のプロセスのシステム呼び出しをすべてトレースできます。行頭のstraceに続けてコマンドを通常どおり入力します。

```
$ strace -e open ls

execve("/bin/ls", ["ls"], [/* 88 vars */]) = 0
uname({sys="Linux", node="edison", ...})= 0
brk(0)                                  = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0x40017000
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or directory) open("/etc/ld.so.cache", O_R
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...})= 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...})= 0
ioctl(1, TIOCGWINSZ, {ws_row=53, ws_col=110, ws_xpixel=897, ws_ypixel=693}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...})= 0
fcntl64(3, F_SETFD, FD_CLOEXEC)         = 0
getdents64(3, /* 5 entries */, 4096)    = 160
```

```
getdents64(3, /* 0 entries */, 4096)    = 0
close(3)                                = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...})= 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \ = 0x40018000
write(1, "ltrace-ls.txt  myfile.txt  strac"..., 41) = 41
munmap(0x40018000, 4096)                = 0
exit_group(0)                           = ?
```

たとえば、特定のファイルを開く試みをすべてトレースするには、以下を入力します。

```
 $ strace -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or directory) open("/etc/ld.so.ca
open("/proc/mounts", O_RDONLY)          = 3
[...]
open("/proc/filesystems", O_RDONLY)     = 3 open("/proc/self/attr/current", O_RDONLY) = 4
```

すべての子プロセスをトレースするには、パラメータ-fを使用します。straceの動作と出力形式は厳密に制御できます。詳細については、man straceを参照してください。

## 36.18 実行されたプログラムによるライブラリ呼び出し:ltrace

コマンドltraceを使用すると、プロセスによるライブラリ呼び出しをトレースできます。このコマンドの使用方法は、straceと同様です。パラメータ-cを指定すると、発生したライブラリ呼び出しの回数と持続期間が出力されます。

```
$ ltrace -c find /usr/share/doc
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
86.27    1.071814          30     35327           write
10.15    0.126092          38      3297           getdents64
2.33    0.028931           3     10208           lstat64
0.55    0.006861           2      3122         1 chdir
0.39    0.004890           3      1567         2 open
[...]
0.00    0.000003           3         1           uname
0.00    0.000001           1         1           time
------ ----------- ----------- --------- --------- ----------------
100.00    1.242403                58269         3 total
```

## 36.19 必須ライブラリの指定:ldd

コマンドlddを使用すると、引数として指定した動的実行可能ファイルをロードするライブラリを確認できます。

```
$ ldd /bin/ls
linux-gate.so.1 =>  (0xffffe000)
librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)
libacl.so.1 => /lib/libacl.so.1 (0x40033000)
libselinux.so.1 => /lib/libselinux.so.1 (0x40039000)
libc.so.6 => /lib/tls/libc.so.6 (0x40048000)
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4015d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

静的バイナリファイルには、動的ライブラリは不要です。

```
$ ldd /bin/sash
    not a dynamic executable
$ file /bin/sash
/bin/sash:ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
for GNU/Linux 2.2.5, statically linked, stripped
```

## 36.20 ELF バイナリに関する補足情報

バイナリの内容は、readelfユーティリティを使用して読み込むことができます。このユーティリティは、他のハードウェアアーキテクチャ用に作成されたELFファイルにも使用できます。

```
$ readelf --file-header /bin/ls
ELF Header:
 Magic:7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
 Class:ELF32
 Data:2's complement, little endian
 Version:1 (current)
 OS/ABI:UNIX - System V
 ABI Version:0
 Type:EXEC (Executable file)
 Machine:Intel 80386
 Version:0x1
```

```
Entry point address:0x8049b40
Start of program headers:52 (bytes into file)
Start of section headers:76192 (bytes into file)
Flags:0x0 Size of this header:52 (bytes)
Size of program headers:32 (bytes)
Number of program headers:9
Size of section headers:40 (bytes)
Number of section headers:29
Section header string table index: 26
```

## 36.21 プロセス間通信:ipcs

コマンドipcsは、現在使用中のIPCリソースのリストを出力します。

```
$ ipcs
------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x000027d9 5734403    toms       660        64528      2
0x00000000 5767172    toms       666        37044      2
0x00000000 5799941    toms       666        37044      2

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
0x000027d9 0          toms       660        1

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
```

## 36.22 timeを使用した時間測定

コマンドの所要時間は、timeユーティリティで判断できます。このユーティリティには2つのバージョンがあります。一方はシェルビルトインで、他方はプログラム(/usr/bin/time)です。

```
$ time find .> /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s
```

# Part V

# 付　録

# 情報源とマニュアル

SUSE LINUXシステムについては、さまざまな情報源が存在しています。こうした情報源の一部はSUSE固有のものですが、多くは一般的な情報源です。また、システム上やインストール媒体上で利用できる情報や、インターネット経由でアクセスできる情報があります。

## SUSEのマニュアル

HTML形式またはPDF形式のマニュアルの詳細については、RPMパッケージの`suselinux-userguide_en`と`suselinux-adminguide_en`を参照してください。標準インストールの場合、マニュアルは`/usr/share/doc/manual/`ディレクトリにインストールされます。この情報へのアクセス権はSUSEヘルプセンターから提供されます。

## Linux Documentation Project (LDP)

Linux Documentation Project (`http://www.tldp.org/`を参照)は、Linuxのマニュアルを制作するボランティアチームです。LDPには、HOWTO (操作方法)、FAQ (よくある質問)、ガイドなど、無償で公開されるすべての情報が含まれます。

HOWTOは手順を段階的に説明したものであり、エンドユーザ、システム管理者、プログラマを対象としています。たとえば、HOWTOには、DHCPサーバの作成や注意事項の説明は含まれていますが、Linux自体のインストール方法は含まれていません。通常、この種のマニュアルはごく一般的な内容となるため、すべての場合に当てはまります。`howto`パッケージには、ASCII形式のHOWTOが組み込まれています。HTML形式を使用する場合は、`howtoenh`をインストールする必要があります。

FAQ(よくある質問)は、「LDAPとは?」や「RAIDとは?」など、メーリングリストにしばしば発生する特定の問題に関する質問と回答をまとめたものです。通常、この種の質問と回答は簡潔にまとめられています。

ガイドは、HOWTOやFAQよりも詳細な内容を取り扱っているマニュアルです。たとえば、カーネルのプログラミングやネットワーク管理などの内容が含まれます。基本的には、読者に詳細情報を提供することを意図しています。

LDPの一部のマニュアルは、PDF、1つ以上のHTMLページ、PostScript、SGMLやXMLソースなど、他の形式でも利用できます。さまざまな言語に翻訳されているマニュアルもあります。

## manページとinfoページ

manページ(マニュアルページ)とは、コマンド、システムコール、ファイル形式、類似項目に関するヘルプテキストです。通常、マニュアルページは名前、構文、説明、オプション、ファイルのように複数のセクションに分かれています。

マニュアルページを表示するには、manに続けてコマンド名を入力します。たとえば、`man ls`と入力すると`ls`コマンドのヘルプテキストが表示されます。表示領域を移動するにはカーソルキーを使用します。Ⓠキーを押すとmanが終了します。マニュアルページ(コマンド`ls`の場合)を印刷するには、`man -Tps | lpr`のようにコマンドを入力します。 manコマンドの詳細を表示するには、`--help`オプションまたはmanのmanページ(`man man`)を使用します。

一部のマニュアルは、infoページ(情報ページ)形式でも利用できます。たとえば、`grep`の場合は、`info grep`と指定すれば、その情報ページが表示されます。

情報ページの内容はマニュアルページよりも詳しくなっています。情報ページは、さまざまなノードに分かれており、各ページはWebブラウザに似たinfoリーダーで読むことができます。Ⓟキー(前ページ)とⓃキー(次ページ)を使用すると、情報ページ内で移動できます。`info`を終了するにはⓆキーを使用します。その他のキーについては、`info`のドキュメントを参照してください(`info info`を使用)。

マニュアルページと情報ページは、どちらもKonquerorで読むことができます。URL行に`man:`⟨コマンド⟩または`info:`⟨コマンド⟩と入力すると、必要なドキュメントが表示されます。

# 規格と仕様

規格と仕様に関する情報は、さまざまな情報源から提供されます。

**www.linuxbase.org** Free Standards Groupは、無償ソフトウェアとオープンソースソフトウェアの配布を促進する独立した非営利団体です。この団体は、ディストリビューションに依存しない規格を定義することで、この目標達成に努めています。また、重要なLSB (Linux Standard Base、Linux標準ベース)など、複数の規格の維持管理を監督しています。

**http://www.w3.org** World Wide Web Consortium (W3C)は、最もよく知られた標準化団体です。1994年10月にTim Berners-Leeによって設立され、Webテクノロジの標準化に専念しています。W3Cは、HTML、XHTML、XMLなど、メーカに依存しないオープン仕様の無償による普及を促進しています。これらのWeb規格はワーキンググループにおいて4段階のプロセスを経て開発され、*W3C勧告*(REC)として一般に公表されます。

**http://www.oasis-open.org** OASIS (構造化情報標準促進協会: Organization for the Advancement of Structured Information Standards)は、Webセキュリティ、Eビジネス、商取引、ロジスティクス、各種市場間の相互運用性に関する標準の開発を専門とする国際団体です。

**http://www.ietf.org** Internet Engineering Task Force (IETF)は、研究者、ネットワーク設計者、サプライヤ、ユーザが参加する国際的な団体です。インターネットアーキテクチャの開発とプロトコルを使用したインターネット運用の円滑化を目的としています。

IETFによる標準はすべてRFC (Request for Comments)として公開され、無償で入手できます。RFCには6つのタイプ(標準に関する提案、標準のドラフト、インターネット標準、実験的なプロトコル、情報ドキュメント、過去の標準)があります。より狭義では、最初の3タイプ(提案、ドラフト、完成版)のみがIETFの標準といえます(`http://www.ietf.org/rfc/rfc1796.txt`を参照)。

**http://www.ieee.org** 電気電子学会(Institute of Electrical and Electronics Engineers: IEEE)は、情報技術、通信、医薬、輸送などの分野における標準を策定する組織です。IEEEの標準は有償です。

**http://www.iso.org** 国際標準化機構委員会(ISO Committee: International Organization for Standards)は、世界最大の標準開発機関であり、世

界140カ国の標準化機関からなるネットワークを維持しています。ISOの標準は有償です。

**http://www.din.de, http://www.din.com**

Deutsches Institut fr Normung (DIN)は、1917年に設立され、登録された科学技術機関です。DINによれば、この組織は「ドイツにおける標準を取り扱い、各国およびヨーロッパの標準化団体に対してドイツの考えを提示することを目的とした組織」です。

この組織にはメーカ、消費者、貿易業者、サービス業者、科学者、標準の設立に関心を持つその他の人々が参加しています。標準は有償であり、DINのホームページから発注できます。

# ファイルシステムチェック

## Manual Page of reiserfsck

```
REISERFSCK(8)                                          REISERFSCK(8)

NAME
       reiserfsck - check a Linux Reiserfs file system

SYNOPSIS
       reiserfsck  [  -afprVy ] [ --rebuild-sb | --check | --fix-
       fixable | --rebuild-tree | --clean-attributes  ]  [  -j  |
       --journal device ] [ -z | --adjust-size ] [ -n | --nolog ]
       [ -l | --logfile file ] [ -q | --quiet ] [ -y | --yes ]  [
       -S  |  --scan-whole-partition ] [ --no-journal-available ]
       device

DESCRIPTION
       Reiserfsck searches for a Reiserfs filesystem on a device,
       replays  any  necessary transactions, and either checks or
       repairs the file system.

       device is the special file corresponding to the device  or
              partition  (e.g /dev/hdXX for IDE disk partition or
              /dev/sdXX for SCSI disk partition).

OPTIONS
       --rebuild-sb
              This option recovers the superblock on  a  Reiserfs
              partition.   Normally  you only need this option if
              mount reports "read_super_block: can't find a reis
              erfs  file system" and you are sure that a Reiserfs
              file system is there.

       --check
              This default action checks file system  consistency
```

and reports but does not repair any corruption that it finds. This option may be used on a read-only file system mount.

--fix-fixable
This option recovers certain kinds of corruption that do not require rebuilding the entire file sys tem tree (--rebuild-tree). Normally you only need this option if the --check option reports "corrup tion that can be fixed with --fix-fixable". This includes: zeroing invalid data-block pointers, cor recting st_size and st_blocks for directories, and deleting invalid directory entries.

--rebuild-tree
This option rebuilds the entire file system tree using leaf nodes found on the device. Normally you only need this option if the --check option reports "corruption that can be fixed only during --rebuild-tree". You are strongly encouraged to make a backup copy of the whole partition before attempting the --rebuild-tree option.

--clean-attributes
This option cleans reserved fields of Stat-Data items.

--journal device , -j device
This option supplies the device name of the current file system journal. This option is required when the journal resides on a separate device from the main data device (although it can be avoided with the expert option --no-journal-available).

--adjust-size, -z
This option causes reiserfsck to correct file sizes that are larger than the offset of the last discov ered byte. This implies that holes at the end of a file will be removed. File sizes that are smaller than the offset of the last discovered byte are corrected by --fix-fixable.

--logfile file, -l file
This option causes reiserfsck to report any corrup tion it finds to the specified log file rather than stderr.

--nolog, -n
This option prevents reiserfsck from reporting any kinds of corruption.

```
       --quiet, -q
              This  option prevents reiserfsck from reporting its
              rate of progress.

       --yes, -y
              This option inhibits reiserfsck from asking you for
              confirmation  after telling you what it is going to
              do, assuming yes. For safety, it does not work with
              the --rebuild-tree option.

       -a, -p These  options are usually passed by fsck -A during
              the automatic checking of those  partitions  listed
              in  /etc/fstab.  These  options cause reiserfsck to
              print some information  about  the  specified  file
              system,  check if error flags in the superblock are
              set and  do  some  light-weight  checks.  If  these
              checks reveal a corruption or the flag indicating a
              (possibly fixable) corruption is found set  in  the
              superblock,  then  reiserfsck  switches to the fix-
              fixable mode. If the flag indicating a  fatal  cor
              ruption  is found set in the superblock, then reis
              erfsck finishes with an error.

       -V     This option prints the  reiserfsprogs  version  and
              exit.

       -r, -f These options are ignored.

EXPERT OPTIONS
       DO  NOT  USE  THESE  OPTIONS  UNLESS YOU KNOW WHAT YOU ARE
       DOING.  WE ARE NOT RESPONSIBLE  IF  YOU  LOSE  DATA  AS  A
       RESULT OF THESE OPTIONS.

       --no-journal-available
              This  option  allows reiserfsck to proceed when the
              journal device is not available. This option has no
              effect when the journal is located on the main data
              device. NOTE: after this  operation  you  must  use
              reiserfstune to specify a new journal device.

       --scan-whole-partition, -S
              This option causes --rebuild-tree to scan the whole
              partition, not only used space on the partition.

EXAMPLE OF USING
       1. You think something may be wrong with a reiserfs parti
       tion  on  /dev/hda1  or  you  would just like to perform a
       periodic disk check.
```

2. Run reiserfsck --check --logfile check.log /dev/hda1. If reiserfsck --check exits with status 0 it means no errors were discovered.

3. If reiserfsck --check exits with status 1 (and reports about fixable corruptions) it means that you should run reiserfsck --fix-fixable --logfile fixable.log /dev/hda1.

4. If reiserfsck --check exits with status 2 (and reports about fatal corruptions) it means that you need to run reiserfsck --rebuild-tree. If reiserfsck --check fails in some way you should also run reiserfsck --rebuild-tree, but we also encourage you to submit this as a bug report.

5. Before running reiserfsck --rebuild-tree, please make a backup of the whole partition before proceeding. Then run reiserfsck --rebuild-tree --logfile rebuild.log /dev/hda1.

6. If the --rebuild-tree step fails or does not recover what you expected, please submit this as a bug report. Try to provide as much information as possible and we will try to help solve the problem.

EXIT CODES

reiserfsck uses the following exit codes:

0 - No errors.
1 - File system errors corrected.
4 - File system fatal errors left uncorrected,
reiserfsck --rebuild-tree needs to be launched.
6 - File system fixable errors left uncorrected,
reiserfsck --fix-fixable needs to be launched.
8 - Operational error.
16 - Usage or syntax error.

AUTHOR

This version of reiserfsck has been written by Vitaly Fertman <vitaly@namesys.com>.

BUGS

There are likely to be some bugs. Please report bugs to the ReiserFS mail-list <reiserfs-list@namesys.com>.

TODO

Faster recovering, signal handling, i/o error handling, etc.

SEE ALSO

mkreiserfs(8), reiserfstune(8) resize_reiserfs(8), debu

greiserfs(8),

Reiserfsprogs-3.6.9 April 2003 REISERFSCK(8)

# e2fsckのマニュアルページ

E2FSCK(8) E2FSCK(8)

Linuxの2番目の拡張ファイルシステムをチェックします

書式
e2fsck [ -pacnyrdfvstDFSV ] [ -b superblock ] [ -B block-size ] [ -l|-L bad_blocks_file ] [ -C fd ] [ -j external-journal ] [ -E extended_options ] device

説明 e2fsckは、Linuxの2番目の拡張ファイルシステム(ext2fs)のチェックに使用されます。e2fsckでは、ジャーナルを格納しているext2ファイルシステム(別名ext3ファイルシステム)もサポートします。そのために、最初にジャーナルをファイルシステムに適用した後に、通常のe2fsck処理を続行します。ジャーナルが適用されると、通常、ファイルシステムは問題なしとマークされます。したがってext3ファイルシステムの場合、通常、e2fsckはジャーナルを実行して終了します。ただし、そのスーパーブロックにさらにチェックを要することが示されている場合は除きます。

deviceは、ファイルシステムが保存されているデバイスファイル(/dev/hdc1など)です。

オプション -a
このオプションは-pオプションと動作が同じで、下位互換性のためだけに用意されています。そのため、できる限り-pオプションを使用することをお勧めします。

-b superblock
通常のスーパーブロックを使用するのではなく、superblockで指定された代替スーパーブロックを使用します。通常、このオプションは、プライマリスーパーブロックが壊れた場合に使用されます。バックアップスーパーブロックの場所は、ファイルシステムのブロックサイズによって異なります。ブロックサイズが1Kのファイルシステムの場合、バックアップスーパーブロックはブロック8193にあります。ブロックサイズが2Kの場合はブロック16384にあり、4Kの場合はブロック32768にあります。

さらにバックアップスーパーブロックの場所を特定するには、-nオプションを指定したmke2fsプログラムを使用して、スーパーブロックが作成された場所を出力します。スーパーブロックの正確な場所を出力するためには、

mke2fsの-bオプションでファイルシステムのブロックサイズを指定する必要があります。

代替スーパーブロックが指定され、ファイルシステムが読み取り専用で開かれていない場合、e2fsckは、ファイルシステムチェックの完了時にプライマリスーパーブロックが適切に更新されていることを確認します。

-B blocksize
通常、e2fsckは、適切なブロックサイズを検索する際に、さまざまなブロックサイズでスーパーブロックを検索します。この検索は、時間の無駄になる場合があります。このオプションでは、特別なブロックサイズでのスーパーブロックの検索だけがe2fsckで強制的に行われます。スーパーブロックが見つからない場合、e2fsckは致命的なエラーで終了します。

-c
このオプションを指定すると、e2fsckはbadblocks(8)プログラムを実行し、ファイルシステムの不良ブロックを検出してマークするために、検出されたブロックを不良ブロックinodeに追加します。このオプションが2回指定されると、非破壊読み書きテストを利用して不良ブロックのスキャンが行われます。

-C fd
このオプションを指定すると、e2fsckは、ファイルシステムチェックの進行状況を監視できるように、指定したファイル識別子に完了情報を書き込むことができます。通常、このオプションは、e2fsckの実行元プログラムで使用されます。指定されたファイル識別子が0の場合、e2fsckは、進行状況を表す完了バーを出力します。このためには、e2fsckがビデオコンソールまたは端末から実行されている必要があります。

-d　デバッグ情報を出力します(e2fsckをデバッグしていない限り役に立ちません)。

-D　ファイルシステム内のディレクトリを最適化します。このオプションを指定すると、e2fsckはすべてのディレクトリを最適化しようとします。そのために、ファイルシステムがディレクトリインデックス機能をサポートしている場合はインデックスを再作成します。あるいは、小さいディレクトリの場合や従来の線形ディレクトリを使用するファイルシステムの場合は、ディレクトリをソートおよび圧縮します。

-E e2fsck拡張オプションを設定します。拡張オプションはカンマで区切り、等号(=)を使用して引き数を指定できます。次のオプションがサポートされています。

ea_ver=extended_attribute_version
ファイルシステムの拡張属性ブロックのフォーマットが、指定したバージョン番号であると仮定します。バージョン番号には1または2を指定できます。デフォルトの拡張属性のバージョンフォーマットは2です。

-f　　ファイルシステムに問題がないと思える場合でも、強制的にチェックします。

-F　　チェック開始前にファイルシステムデバイスのバッファキャッシュをフラッシュします。e2fsckのタイムトライアルをする場合にのみ役に立ちます。

-j external-journal
　　このファイルシステムの外部ジャーナルが見つかるパス名を設定します。

-l filename
filenameで指定したファイルに登録されているブロック番号を不良ブロックのリストに追加します。このファイルのフォーマットは、badblocks(8)プログラムで生成されるファイルのフォーマットと同じです。ブロック番号は、ファイルシステムのブロックサイズに基づいています。したがって、正しい結果を得るためには、badblocks(8)にファイルシステムのブロックサイズを指定する必要があります。結果として、e2fsckに-cオプションを指定する方がはるかに簡単で安全です。この理由は、正しいパラメータがbadblocksプログラムに確実に渡されるためです。

-L 不良ブロックのリストを、filenameで指定したファイル内のブロックリストになるように設定します(このオプションは、-lオプションと同じです。ただし、不良ブロックリストを消去してから、ファイルに登録されているブロックを不良ブロックリストに追加する点は除きます)。

-n　　ファイルシステムを読み取り専用で開き、すべての質問に「no」と答えます。e2fsckを非対話型で使用できます(注意:　-nオプションのほかに、-c、-l、または-Lのオプションを指定した場合は、不良ブロックリストを更新できるようにファイルシステムが読み書き属性で開かれます)。

-p　　ファイルシステムを質問なしで自動的に修復(preen)します。

-r　　このオプションは、動作なしであり、下位互換性のためだけに用意されています。

-s　　このオプションを指定すると、標準的なバイト順序(i386またはリトルエンディアン)になるように、ファイルシステムのバイトスワップが行われます。ファイルシステムがすでに標準的なバイト順序になっている場合、e2fsckでは何も行われません。

-S　　このオプションを指定すると、現在のバイト順序に関係なく、ファイルシステムのバイトスワップが行われます。

-t　　e2fsckのタイミング統計を出力します。このオプションを2回指定すると、詳しいタイミング統計が次々に出力されます。

-v　　詳細表示モード。

-V　　バージョン情報を出力して終了します。

-y　　すべての質問に「yes」と答えることで、e2fsckを非対話型で使用できます。

終了コード e2fsckが返す終了コードは、次の状態を表す数値の合計です。

0    - エラーなし
1    - ファイルシステムのエラーが修正された
2    - ファイルシステムのエラーが修正されたので、システムを再起動する必要がある
4    - ファイルシステムのエラーが修正されないままである
8    - 操作エラー
16   - 使用方法または構文のエラー
32   - e2fsckがユーザ要求によってキャンセルされた
128  - 共有ライブラリエラー

シグナル 次のシグナルがe2fsckに送信されたときの効果は、記載されているとおりです。

SIGUSR1 このシグナルにより、e2fsckは、進行状況を表す完了バー
        の表示を開始します(-Cオプションの説明を参照)。

SIGUSR2 このシグナルにより、e2fsckは、進行状況を表す完了バー
        の表示を停止します

バグ報告
  ほとんどのソフトウェアにバグはあります。あるファイルシステムでe2fsck
  がクラッシュする場合や、e2fsckでファイルシステムを修復できない場合は、
  作者に報告してください。

  バグ報告にはできるだけ多くの情報を含めてください。理想的には、e2fsck
  の詳しい実行記録があれば、どのようなエラーメッセージが表示されているかを
  正確に確認できます。実行記録を保存できる書き込み可能なファイルシステムがあれば、
  script(1)プログラムで、e2fsckの出力を手軽にファイルに保存できます。

  dumpe2fs(8)の出力を送ることも役に立ちます。特定のinodeが原因でe2fsck
  に問題が生じていると思われる場合は、debugfs(8)コマンドを実行し、関連する
  inodeに対して実行したstat(1u)コマンドの出力を送ってください。inode
  がディレクトリである場合は、debugfs dumpコマンドを使用すれば、ディレクトリ
  inodeの内容を抽出できるので、この内容をuuencode(1)にかけて送ってください。

実行しているバージョンがわかるように、必ずe2fsck
の実行時に表示されるバージョン文字列すべてを含めてください。

著者このバージョンのe2fsckは、Theodore Ts'o <tytso@mit.edu>によって作成されました。

関連項目
  mke2fs(8)、tune2fs(8)、dumpe2fs(8)、debugfs(8)

# xfs_checkのマニュアルページ

```
xfs_check(8)                                                    xfs_check(8)


NAME
 xfs_check - XFSファイルシステムの整合性チェック

SYNOPSIS
 xfs_check [ -i ino ] ...[ -b bno ] ...[ -s ] [ -v ] xfs_special

 xfs_check -f [ -i ino ] ...[ -b bno ] ...[ -s ] [ -v ] file

DESCRIPTION
 xfs_checkは、XFSファイルシステムの整合性をチェックします。これを実行するのは、通常、
 ファイルシステムに整合性の問題があると確信できる　　理由が　　ある場合だけです。
  チェックするファイルシステムはxfs_special引数で指定します。
  これは、ファイルシステムのディスクデバイスまたはボリュームデバイスにする必要があります。
  ファイルに格納されたファイルシステムも-fフラグでチェックできます。
  xfs_checkの実行中は、ファイルシステムは通常、アンマウントされているか、
  読み取り専用になっている必要があります。さもないと、誤った問題が報告されます。

xfs_checkのオプションは次のとおりです。

-f        特殊デバイスが　　実際にはファイルであることを指定します
        (mkfs.xfs -d fileオプションを参照)。
        これは、ファイルシステムのイメージコピーが通常のファ
        イル内に作成された場合に起こることがあります。

-s        重大な　　エラーだけが　　報告される　　ことを指定します。
        重大なエラーとは、ファイルシステム内の主要データ構造を検出できなく
        させるようなエラーのこと　　です。
        このオプションは、重大な問題があり、出力で実際の問題を判別することが困難になる場合に、
        出力の量を切り詰めるために使用できます。

-v        冗長出力を指定します。一般的なサイズのファイルシステムの場合、
        これはあまりに長くなります。このオプションは、内部的に使用されるだけです。

-i ino      特定のinodeの冗長な動作を指定します。たとえば、一定のinode
        に関連したすべてのブロックを見つけるために使用できます。

-b bno      特定のファイルシステムブロックの冗長な動作を指定　　します。
        たとえば、特定のブロックの使用対象を判断するために使用できます。
        ブロック番号は、「ファイルシステムブロック番号」です。
        ディスクアドレス　　(すなわちxfs_bmapで報告されたアドレス)とファ
        イルシステムブロックの間の変換は、xfs_db's convert
        コマンドによって実行されます。
```

xfs_checからの冗長でない出力は、ファイルシステムに整合性がないことを意味します。
このファイルシステムは、 xfs_repair(8)を使用してファイルシステムを 現場で修復するか、
xfsdump(8)とmkfs.xfs(8)を使用してファイルシステムのダンプを取り、
新しいファイルシステムを作成したのち、xfsrestore(8)
を使用してデータを新規システムに復元すれば修復できます。
壊れたファイルシステムではxfsdumpは失敗する可能性があることに注意してください。
ただし、ファイルシステムがマウント可能である場合は、xfsdumpを使用して、
重要なデータを保存して から、 xfs_repairでファイルシステムを修復できます。
ファイルシステムがマウント可能でない場合は、実行できるオプションはxfs_repairだけです。

DIAGNOSTICS
ある環境においては、不幸なことに、xfs_checkは役立つ出力を作成する代わりに、
コアダンプを行うことがあります。
ファイルシステムが完全に壊れている場合は、「メッセージ x x x 有効なフ
ァイルシステムではありません」の代わりにコアダンプが作成されます。

ファイルシステムが大規模である(ファイルの数が多い)場合は、xfs_check
でメモリが使い果たされることがあります。この場合、メモリ切れのメッセージが印刷されます。

次に一般に起こり得る問題の説明とその関連メッセージを示します。
作成される診断のほとんどは、ファイルシステムの 構造を理解して初めて意味を持ちます。

agf_freeblks n, counted m in ag a
割り当てグループの割り当てグループヘッダ内の空きブロックカウント
が空きとしてカウントされたブロック数に一致しません。

agf_longest n, counted m in ag a
割り当てグループの割り当てグループヘッダ内の最長空きサイズが割り
当てグループ内に見つかった最長空きサイズと一致しません。

agi_count n, counted m in ag a
割り当てグループの割り当てグループヘッダ内の割り当て済みinode
カウントが割り当てグループ内でカウントされたinode数と一致しません。

agi_freecount n, counted m in ag a
割り当てグループの割り当てグループヘッダ内の空きinode
カウントが割り当てグループで空きとしてカウントされたinode数と一致しません。

block a/b expected inum 0 got i
ブロック番号は、ペア(割り当てグループ番号とその割り当てグル
ープ内のブロック番号)と して指定されます。ブロックは、複数のinode
間で複数回使用されます(共有)。このメッセージは、通常、次の タイプ
のメッセージの後に続きます。

block a/b expected type unknown got y
ブロックは、複数回使用されます(共有)。

block a/b type unknown not expected
ブロックは、考慮されません(空きリストになく、使用中でもありません)。

```
link count mismatch for inode nnn (name xxx), nlink m, counted n inode
           は、間違ったリンクカウント(ディレクトリ内の差分数)を持っています。

rtblock b expected inum 0 got i
            ブロックは、複数のinode間で複数回使用されました(共有)。
            このメッセージは、次のタイプのメッセージの後に続きます。

rtblock b expected type unknown got y
            リアルタイムブロックは、複数回使用されます(共有)。

rtblock b type unknown not expected
            このリアルタイムブロックは、考慮されません(空きリストになく、
            使用中でもありません)。

sb_fdblocks n, counted m
            スーパーブロック内に記録された空きデータブロックの数が
            ファイルシステム内で空きとしてカウントされた数と一致しません。

sb_frextents n, counted m
            スーパーブロック内に記録された空きリアルタイム データブロック
            の数がファイルシステム内で空きとしてカウントされた数と一致しません。

sb_icount n, counted m
            スーパーブロック内に記録された割り当て済みinodes          の数が
            ファイルシステム内で割り当てられた数と一致しません。

sb_ifree n, counted m
            スーパーブロック内に記録された空きinodes
            の数がファイルシステム内で空きとしてカウントされた数と一致しません。

SEE ALSO mkfs.xfs(8), xfsdump(8), xfsrestore(8), xfs_ncheck(8), xfs_repair(8), xfs(5).

                                                                    xfs_check(8)
```

# jfs_fsckのマニュアルページ

```
jfs_fsck(8)                    JFSユーティリティ  - ファイルシステムチェック                    jfs_fsck
```

名前
jfs_fsckはJFSトランザクションログの再生を開始して、JFS
フォーマットデバイスをチェックし、修復します。

概要
  jfs_fsck [ -afnpvV ] [  -j  journal_device  ]  [ --omit_journal_replay ]
  [ --replay_journal_only ] デバイス

説明 jfs_fsck
  JFSトランザクションログを再生するために使用されます。
  JFSフォーマットデバイスでエラーの有無を確認し、検出されたエ
  ラーを修復します。

  デバイスとはチェックされる実際のデバイスに対応する特別なフ
  ァイル名を意味します(/dev/hdb1など)。

  jfs_fsckはrootとして実行してください。

警告
  jfs_fsckはアンマウントされているファイルシステムまたはREAD
  ONLYでマウントされているファイルシステムをチェックするために
  のみ使用してください。jfs_fsckを使用してREAD
  ONLY以外でマウントされたファイルシステムをチェックすると、
  ファイルシステムが深刻な損傷を受ける恐れがあります。

オプション オプションが選択されていない場合、デフォルトとしてオプション-pが指定されます。

-a      自動チェック　　モード　-　トランザクションログを再生します。
        集約の状態がダーティ、またはログの再生に失敗した場合以外は、fsck
        処理を継続しないでください。　　機能的には-pと同じです。
        自動チェックモードは通常、jfs_fsckがブート時に呼び出された場合
        に使用されるデフォルトモードです。

-f      ファイルシステムがクリーンな状態だとしてもトランザクションログを再生し、
        強制的にチェックします。すべての問題を自動的に修復します。

-j journal_device
        ジャーナルデバイスを指定します。

-n      ファイルシステムを読み取り専用で開きます。
        トランザクションログを再生しません。
        エラーを報告しますが、エラーの修復はしません。

--omit_journal_replay
        トランザクションログの再生を省略します。
        このオプションは最終手段として以外は使用しな
        いでください(ログが完全に破損されて、
        ログの再生でさらに問題が発生する場合など)。

-p

自動的にファイルシステムを修復(preen)します。
トランザクションログを再生します。
集約の状態がダーティ、またはログの再生に失敗した場合以外はfsck処理を継続しないでください。
機能的には-aと同じです。

--replay_journal_only
トランザクションログのみを再生します。
再生が失敗した場合、またはジャーナルの再生が終了した後にファイルシステムがまだダーティな状態である場合は、フルファイルシステムで続行しないでください。
一般的にこのオプションは、ファイルシステムをアンマウント状態で放置できるようなデバッグ目的でのみ使用するようにしてください。
このオプションは-f、-n、--omit_journal_replayとともに使用できません。

-v 詳細メッセージング。stdoutに詳細とデバッグステートメントを表示します。

-V バージョン情報を表示し、終了します(その他の選択されたオプションは無視されます)。

例 2台目のハードディスクにある3番目のパーティションをチェックし、stdoutに拡張情報を出力。
トランザクションログを再生して、完全jfs_fsckチェックを強制実行し、さらにすべてのエラーの修復を許可する。

```
jfs_fsck -v -f /dev/hdb3
```

1台目のハードディスクにある5番目のパーティションをチェックし、レポートを作成する。ただしどのエラーも修復しない。

```
jfs_fsck -n /dev/hda5
```

終了コード jfs_fsckが戻す終了コードは次の条件のいずれかになります。

0 エラーなし。

1 ファイルシステムエラーが修復されました。またはトランザクションログが正常に再生されました。

2 ファイルシステムが修復されました。ファイルシステムがマウントされていた場合は、システムをリブートする必要があります。

4　　ファイルシステムエラーは修復されませんでした。

8　　操作に誤りがあります。

16　　使用方法または構文に誤りがあります。

128　　共有ライブラリエラーです。

バグのレポート JFSまたはjfs_fsckでバグがあった場合は、JFSプロジェクト
Webサイトのバグトラッキングシステムでレポートしてください(「Report
bugs」セクション)。http://oss.software.ibm.com/jfs

JFSデバイスでjfs_fsckを-vオプション付きで実行した場合の完全な出力も含め、
関連情報をできるだけ多く送信してください。

SEE ALSO
fsck(8), jfs_mkfs(8), jfs_fscklog(8), jfs_tune(8), jfs_logd

作成者 Barry Arndt(barndt@us.ibm.com) William Braswell, Jr.

jfs_fsckはIBMにより管理されています。詳細についてはJFSプロジェクトWebサイトを参照してください。
http://oss.software.ibm.com/jfs

2002年10月29日 jfs_fsck(8)

# GNU一般公開使用許諾

## GNU一般公開使用許諾



### はじめに

ほとんどのソフトウェアの使用許諾は、共有や変更が自由にできないようになっています。対照的に、*GNU一般公開使用許諾*は、フリーソフトウェアの共有や変更の自由を保証するように、つまり、フリーソフトウェアがそのすべてのユーザに対してフリーであることを保証するように考えられています。この*一般公開使用許諾*は、*フリーソフトウェア財団*の大半のソフトウェア、および作者がGPLを行使することにした他のプログラムに適用されます(*フリーソフトウェア財団*の他の一部のソフトウェアは、代わりに*GNUライブラリ一般公開使用許諾*の適用を受けています)。また、LGPLをユーザのプログラムに適用することができます。

「フリー」ソフトウェアを話題にしている場合は、価格ではなく、自由のことを言及しています。一般公開使用許諾は、利用者がフリーソフトウェアの複製物を間違いなく自由に頒布できるように考えられています(しかも、利用者が、希望に応じてこのようなサービスの対価も自由に設定できるようになっています)。また、必要に応じたソースコードの受領または入手、フリーソフト

ウェアの変更、あるいは新たなフリープログラムでのその部分的な流用が確実に行えるようになっています。さらに、利用者はこうした作業を実行できることが確実にわかるようになっています。

利用者の権利を保護するには、利用者にこうした権利を与えないようにすることや、利用者に権利の放棄を求めることを禁止するように制限を加える必要があります。このような制限のため、フリーソフトウェアの複製物を頒布する場合やソフトウェアを改変する場合には一定の責任が生じます。

たとえば、フリープログラムの複製物を頒布する場合は、無料または有料にかかわらず、自分のすべての権利を受領者に与える必要があります。また、受領者が確実にソースコードを受領するか、または入手できるようにする必要もあります。さらに、受領者に自分の権利を把握させるために、次の条項を受領者に示す必要があります。

利用者の権利の保護は、2つの手順で行われます。(1)ソフトウェアを著作権で保護します。次に、(2)利用者にソフトウェアの複製、頒布、改変を行うための法的な許可を与える本使用許諾を提供します。

また、作者やFSFの保護のために、このフリーソフトウェアには保証がないことを皆に確実に理解させたいと考えています。ソフトウェアが第三者によって改変されたり頒布されても、その受領者はオリジナル版を入手したわけではないため、第三者にもたらされた問題によってオリジナルの作者の評判が損なわれるものではないことも受領者に理解させる必要があります。

最後に、フリープログラムは、絶えずソフトウェア特許の脅威を受けています。そこで、フリープログラムの再頒布元が特許使用許諾を個々に取得してプログラムを事実上独占するという危険を回避したいと考えています。このようなことを防ぐために、特許を誰でも自由に使用できるように使用許諾を与える必要があるか、まったく使用許諾を与える必要がないかを明確にしました。

複製、頒布、改変の正確な条項は、この後に示します。

# GNU一般公開使用許諾

## 複製、頒布、改変の条項

**0.**本使用許諾は、この*一般公開使用許諾*の条項に従って頒布できることを定めた著作権者の通告が記載されているプログラムまたは他の著作物に適用されます。以降の「プログラム」とは、そのようなプログラムまたは著作物のことを指します。また、*プログラムに基づく著作物*とは、著作権法の下でプログラムまたは派生著作物(そのままの状態または改変した状態、あるいは別の言語に

翻訳した状態でプログラムまたはその一部を含む著作物)のことです(以下、翻訳は「改変」という用語に含まれます)。各被許諾者のことを「利用者」と呼びます。

複製、頒布、改変以外の活動は、本使用許諾では適用されません。これらは対象外です。プログラムを実行する行為は制限されません。また、プログラムからの出力は、その内容が(プログラムの実行によって作成されたこととは無関係に)プログラムに基づく著作物を構成する場合にのみ適用されます。それが本当かどうかは、プログラムの実行内容によって異なります。

**1.**利用者は、受領したプログラムのソースコードを任意のメディアでそのまま複製または頒布することができます。その際には、各複製物に適切な著作権表示と保証の放棄を適宜明記し、本使用許諾および一切の保証の否定に関する通告すべてを維持したまま、プログラムの他の受領者に本使用許諾の複製物をプログラムとともに頒布することを条件とします。

利用者は、複製物の移送に関する実際的な行為を有償にすることができます。また、自己裁量により有償の代わりに保証を提供してもかまいません。

**2.**利用者は、プログラムの複製物またはその一部を改変することができます。したがって、プログラムに基づいた著作物を形成し、第1項の条項に従ってそのような改変を複製または頒布することができます。ただし、次のすべての条件を満たしていることも前提とします。

1. 利用者は、ファイルを改変したことおよび改変日を示す情報を、改変したファイルに明記する必要があります。
2. 利用者は、利用者が頒布または公開する著作物、プログラムの全体または一部を含む著作物、あるいはプログラムまたはその一部から派生した著作物を、総じて本使用許諾の条項に従ってすべてのサードパーティに無償で使用許諾する必要があります。
3. 改変されたプログラムが、通常、実行時に対話形式でコマンドを読み取るようになっている場合に、そのプログラムを最も一般的な方法で対話形式での利用に向けて実行するときは、適切な著作権表示および無保証(あるいは利用者が保証)通告、ユーザがプログラムをここの条件の下で再頒布できること、本使用許諾の複製物をユーザが閲覧する方法の説明などの告知を出力または表示できるようにする必要があります(例外: プログラム自体が対話型にもかかわらず通常そのような告知を出力しない場合は、プログラムに基づいた利用者の著作物でも告知を出力する必要はありません)。

こうした要件は、総じて改変された著作物に適用されます。その著作物の一部がプログラムから派生していないと特定できて、それぞれが別の独立した著作

物であると合理的に考えられる場合、利用者がそれらを別の著作物として頒布するときには、本使用許諾およびその条項はそうした部分には適用されません。しかし、利用者が同じ部分をプログラムに基づいた著作物全体の一部として頒布する場合は、頒布物全体が、本使用許諾の条項に従わなければなりません。これは、本使用許諾の条項により他の被許諾者に与えられる許可はプログラム全体に及ぶため、作者にかかわらずあらゆる部分に本使用許諾が適用されるためです。

したがって、この項の趣旨は、利用者がすべて作成した著作物に対して権利を主張したり、利用者の権利に異議を申し立てることはではなく、プログラムに基いた派生著作物または集合著作物の頒布を管理する権利を行使することです。

さらに、プログラムに基づいていない別の著作物をプログラム(またはプログラムに基づいた著作物)とともに記憶装置のボリュームや頒布メディアにまとめただけでは、本使用許諾の適用範囲が他の著作物にまで及ぶことはありません。

**3.**利用者は第1項および第2項の条項に従って、プログラム(または第2項のプログラムに基づいた著作物)をオブジェクトコードまたは実行形式で複製または頒布することができます。ただし、次のいずれかを実施するという条件も付きます。

1. 著作物に、該当するマシンの読み取り可能なソースコード一式を添付します。このソースコードを第1項および第2項の条項に従ってソフトウェアの交換で習慣的に使われるメディアで頒布する必要があります。または、

2. マシンで読み取り可能な該当するソースコードの複製を、ソースの頒布に実際にかかるコストを上回らない程度の手数料で第三者に提供するために、著作物に、3年以上有効な書面による提案を添付します。このソースコードの複製を第1項および第2項の条項に従ってソフトウェアの交換で習慣的に使われるメディアで頒布します。あるいは、

3. 著作物に、該当するソースコードの頒布の提案に関して利用者が得た情報を添付します(この選択肢は、非営利目的の頒布の場合、および利用者が、先ほどのb項と一致するような提案とともにオブジェクトコードまたは実行可能ファイル形式でプログラムを受領した場合に限り選択できます)。

著作物のソースコードとは、それに改変を加える場合に好ましい著作物の形式のことです。実行形式の著作物の場合、「ソースコード一式」とは、その著作

物に含まれているすべてのモジュールのすべてのソースコードのほかに、関連するあらゆるインタフェース定義ファイル、および実行可能ファイルのコンパイルやインストールの制御に使用されるスクリプトを加えたものを意味します。しかし、特別な例外として、ソースコードを頒布する場合は、通常の頒布物(ソースまたはバイナリ形式)に実行可能ファイルの実行先オペレーティングシステムの主要なコンポーネント(コンパイラ、カーネルなど)まで加える必要はありません。ただし、そのコンポーネント自体に実行可能ファイルが付随する場合は除きます。

指定された場所から複製するためのアクセス手段を提供することによって、実行可能ファイルまたはオブジェクトコードを頒布している場合は、第三者がオブジェクトコードとともにソースを複製するように強制されなくても、ソースコードを同じ場所から複製するために同等のアクセス手段を提供していればソースコードの頒布と見なされます。

**4.**利用者は、本使用許諾に明示的に記載されている形態を除き、プログラムを複製、改変、二次使用許諾、および頒布してはなりません。別の方法でプログラムを複製、改変、二次使用許諾、または頒布しようとするのは無効であり、本使用許諾の下で利用者の権利は自動的に消滅します。ただし、本使用許諾の下で利用者から複製物または権利を受領した関係者は、条項を遵守している限り、権利が消滅することはありません。

**5.**利用者は、本使用許諾に署名しない限り、それを受け入れる必要はありません。しかし、利用者にプログラムまたはその派生著作物の改変または頒布を許可を与えるものは、本使用許諾以外にありません。こうした行為は、利用者が本使用許諾を受け入れなければ、法的に禁じられます。したがって、プログラム(またはプログラムに基づいた著作物)を改変または頒布すれば、利用者は、そうした行為に関して本使用許諾を受け入れ、プログラムまたはプログラムに基づいた著作物の複製、頒布、または改変に関するすべての条項を承認したことになります。

**6.**利用者がプログラム(またはプログラムに基づいた著作物)を再頒布するたびに、受領者は、本使用許諾の条項に従ってプログラムを複製、頒布、または改変するための使用許諾を元の許諾者から自動的に受領します。利用者は、本使用許諾で受領者に付与された権利の行使に関してさらに制約を課すことはできません。利用者は、第三者の本使用許諾の遵守に対して責任を負うものではありません。

**7.**特許侵害に関する裁判所の判決または申し立ての結果として、あるいはその他の理由(特許問題に限らない)で、利用者に課せられた条件(裁判所の命令、契約などを問わず)が本使用許諾の条件と矛盾している場合でも、利用者は本使用許諾の条件を免除されません。利用者が本使用許諾の下での義務と他の関連する義務を同時に満たすように頒布できない場合は、結果としてプログラムを

頒布してはなりません。たとえば、特許使用許諾で、利用者から直接的または間接的に複製物を受領した者によるプログラムの無償再頒布が許されていない場合、利用者が特許使用許諾と本使用許諾の両方を満足させる唯一の方法は、プログラムの頒布を全面的に中止することです。

本条項の任意の部分が、特定の状況の下で無効または実施不能になっている場合は、本条項の残りの部分が適用されるようになっていますが、他の状況では本条項は総じて適用されるようになっています。

本条項の目的は、特許または他の財産権を侵害したり、そのような権利の主張の正当性を争うことを利用者に勧めることではありません。本条項の唯一の目的は、フリーソフトウェア流通システムの完全性を保護することであり、一般使用許諾の実践によって実現されます。多くの人々が、このシステムの一貫した用途を信頼して、このシステムを介して頒布される広い範囲のソフトウェアに多大な貢献を果たしてきました。作者や寄贈者が他のシステムを介してソフトウェアを頒布するかどうかを決めるのは本人次第であり、被許諾者がその選択を強要することはできません。

本条項は、本使用許諾の他の条項の結果と考えられることを徹底的に明らかにすることを目的としています。

**8.**プログラムの頒布または使用が、特定の国の特許または著作権で保護されたインタフェースのどちらかで制限されている場合、プログラムに本使用許諾を適用した元の著作権者は、そうした国を除外する明示的な地域頒布制限を加えるため、頒布は除外されていない国の中やそうした国同士でしか許可されません。そのような場合は、地域頒布制限が本使用許諾の本文に記載されているのと同様に解釈されます。

**9.***フリーソフトウェア財団*は、*一般公開使用許諾*の改訂版または新版を随時公表することがあります。そのような新版は、性格的には現行版と似たものになりますが、新たな問題や懸案事項に対応するために細部が異なる可能性があります。

各版には、区別するための版番号が設定されます。プログラムに、それに適用される本使用許諾の版番号と「後継版」が指定されている場合、利用者は、選択によって現行版の条項または*フリーソフトウェア財団*から公開される後継版の条項に従うことになります。プログラムに、本使用許諾の版番号が指定されていない場合、利用者は、*フリーソフトウェア財団*からこれまでに公開された任意の版を選択することができます。

**10.**利用者が、頒布条件の異なる他のフリープログラムにプログラムの一部を組み込みたい場合は、作者に書面で許可を求めてください。*フリーソフトウェア財団*が著作権を有するソフトウェアについては、*フリーソフトウェア財団*に書面で問い合わせてください。例外を認める場合もあります。FSFの意思決定は、FSFのフリーソフトウェアのすべての派生物をフリーな状態に保つこと、

および一般にソフトウェアの共有と再利用を促進することという2つの目標を手がかりにしています。

## 無保証

**11. プログラムは無償で使用許諾されるため、適用される法律が許す範囲でプログラムの保証は一切ありません。別途書面に記載されている場合を除き、著作権者やその他の関係者は、明示的または暗黙的を問わず、一切の保証なしで「現状のまま」プログラムを提供します。保証には、市場性および特定目的への適合性に関する暗黙の保証が含まれますが、これに限定されるものではありません。プログラムの品質と性能に関するすべてのリスクは、利用者が負うものとします。プログラムに問題があると判明した場合、必要な点検、修復、修正にかかる費用はすべて利用者が負担するものとします。**

**12.適用法令または書面による合意で規定されている場合を除き、著作権者または上記の許諾を受けてプログラムを改変または再頒布できるその他の関係者は、プログラムを使用したことまたは使用できなかったことによる一般的損害、特別損害、付随的損害、間接的損害(データの消失や正確さの喪失、利用者や第三者から被った損失、他のプログラムとともに動作するプログラムの障害を含むがこれに限定されるものではない)に対して一切の責任を負いません。著作権者や第三者が、そのような損害が発生する可能性について忠告されていた場合でも同様です。**

**条項の終わり**

## こうした条項を新しいプログラムに適用する方法

利用者が新しいプログラムを開発し、できる限り広く一般に使用させたい場合は、本使用許諾の条項に従ってそのプログラムを誰でも再頒布および変更できるフリーソフトウェアにする方法が最も優れています。

そのためには、プログラムに次の通告を添付してください。保証のないことを最も効果的に伝えるには、各ソースファイルの冒頭に通告を添付するのが最も無難です。また、各ファイルには、少なくとも「著作権」の行を用意し、全文がある場所を示す必要があります。

<プログラムの名前とその機能についての簡単な説明。>

Copyright (C) <西暦年> <作者の名前>

このプログラムはフリーソフトウェアです。このプログラムはフリーソフトウェアです。

利用者は、フリーソフトウェア財団が発行したGNU一般公開使用許諾
(第2版または選択によってはそれ以降)
の条項に従って再頒布または改変することができます。

このプログラムは、役に立つことを願って頒布されますが、
市場性や特定目的への適合性についての暗黙の保証を含めて、
保証は一切ありません。詳細については、GNU一般公開使用許諾を参照してください。

利用者は、このプログラムとともにGNU一般公開使用許諾の写しを受領したはずです。
受領していない場合は、フリーソフトウェア財団
(Free Software Foundation, Inc., 59 Temple Place,
Suite 330, Boston, MA 02111-1307, USA)に問い合わせてください。

また、電子メールや郵送で利用者に連絡する方法に関する情報も加えてください。
対話型プログラムの場合は、それを対話モードで起動したときに次のような短い通告が出力されるようにしてください。

Gnomovision バージョン 69, Copyright (C) <西暦年> <作者の名前>

Gnomovisionには、一切保証がありません。詳細については、「show w」
と入力してください。このプログラムはフリーソフトウェアであり、
利用者は特定の条件の下でそれを自由に再頒布してかまいません。
詳細については、「show c」と入力してください。

仮想コマンドshow wとshow cは、一般公開使用許諾の該当する箇所を示すようになっている必要があります。言うまでもなく、利用者が使用するコマンドは、必ずしもshow wやshow cと呼ばれるわけではありません。こうしたコマンドには、利用者のプログラムに合わせたマウスクリックやメニュー項目も考えられます。
また、利用者は、必要に応じて雇用主(プログラマとして作業する場合)または場合によっては学校に、プログラムの「著作権放棄声明書」に署名してもらう必要があります。次に例を示しますが、名前は適当に変更してください。

Yoyodyne社は、James Hackerが作成したコンパイラ操作プログラム
「Gnomovision」の一切の著作権の所有権を放棄します。

Ty Coonの署名、1989年4月1日 副社長Ty Coon

本*一般公開使用許諾*では、著作権を有するプログラムに利用者のプログラムを組み込むことは許可していません。利用者のプログラムがサブルーチンライブ

ラリである場合、利用者は、著作権を有するアプリケーションをそのライブラリとリンクすることを許可した方が有用と考えるかもしれません。利用者がこのようにしたい場合は、本使用許諾ではなく、*GNUライブラリ一般公開使用許諾*を適用してください。

# 用 語 集

**アクセス権**
ファイルのアクセス権で、ユーザやグループがファイルやディレクトリを読み取り、書き込み、または実行できるかどうかが決定されます。通常、アクセス権はシステム管理者が設定します。

**アカウント**
アカウントは、ユーザ名またはログイン名とパスワードによって定義されます。アカウントは、ユーザID (UID)に対応します。

**ACL (Access Control List)**
ファイルやディレクトリの従来のパーミッション概念を拡張したもの。ACLを使用すれば、アクセス権をさらにきめ細かく設定できます。

**ADSL (Asymmetric Digital Subscriber Line)**
電話網を使用する高速転送プロトコル。

**AGP (Accelerated Graphics Port)**
PCIより帯域幅の広い、グラフィックスカード用の高速スロット。AGPグラフィックカードを使用すれば、データをプロセッサに伝送せずに直接ランダムアクセスメモリに戻ることができます。

**ATAPI (Advanced Technology Attachment Packet Interface)**
ATAPIは、(E)IDEコントローラに接続されるCD-ROMドライブのタイプです。ATAPIドライブと同様に、SCSI CD-ROMドライブがありますが、これはSCSIコントローラで処理されます。

**バックアップ**
バックアップとは、破損したデータや失われたデータの復元に使用されるデータのコピーのことです。重要なすべてのデータのバックアップは、定期的に行う必要があります。

**帯域幅**
データ転送用チャネルの最大転送速度。通常は、ネットワーク接続で使用されます。

**BIOS (Basic Input/Output System)**
システムの電源投入時またはコンピュータのリブート時に起動される小さいプログラム。このプログラムが、ハードウェアコンポーネントの初期化を実行します。ほとんどのBIOSでは、対話型のセットアッププログラムを介して低レベルのシステムパラメータを変更できます。このプログラムコードは、読み取り専用メモリ(ROM)チップに常駐します。

**ブックマーク(ブラウザで使用)**
URLのコレクション内の項目。

**ブート**
電源投入からシステムの使用準備が整うまでのコンピュータ動作のシーケンス。

**ブラウザ**
ローカルファイルまたはWebページの内容を表示するプログラム。

**クライアント**
ネットワーク環境において、サーバに接続してサーバの情報を要求するプログラムまたはコンピュータ。

**コマンドライン**
コンピュータにコマンドを発行するためのテキストベースのモード。

**コンソール**
以前は端末と同じ意味でした。Linuxでは、*仮想コンソール*により、グラフィカルディスプレイが実行していなくても、画面を複数の独立したワークセッションで並行して使用できます。

**CPU (Central Processing Unit)**
プロセッサを参照。

**カーソル**
カーソルとは、テキスト入力の場所をマークするブロック文字または下線文字のことです。

**デーモン**
デーモン(disk and execution monitor)とは、バックグラウンドで実行され、必要なときに自動的にアクティブにされるプログラムのことです。たとえば、HTTPデーモン(httpd)がHTTP要求に応答します。

**DDC (Direct Display Channel)**
モニタとグラフィックカードの間の通信規格です。この規格により、モニタ名、解像度などの特定のパラメータをグラフィックカードに転送できます。

**ディレクトリ(ファイルシステム内)**
ファイルや下位ディレクトリ(サブディレクトリ)を格納する構造。ファイルシステム内のディレクトリは、ファイルを編成するためにツリー型の構造を形成します。

**DNS (Domain Name System)**
名前ベースのアドレスをIPアドレスに変換したり、その逆に変換したりするためのプロトコル。

**ドライバ**
オペレーティングシステムの一部であり、ハードウェアコンポーネントとやりとりします。

**電子メール**
ネットワークを介してユーザ間で電子的にメールを転送する手段。電子メールアドレスは、`username@domain.org`という形式です。

**EIDE (Enhanced Integrated Drive Electronics)**
Enhanced IDE規格では、サイズが512MB以上のハードディスクを使用できます。

**環境**
シェルで保持される環境変数セットとその値。ユーザは、既存の環境変数の値を変更(または設定解除)したり、新しい変数を設定したりすることができます。永続的な割り当ては、シェルの設定ファイルによって行われます。

**環境変数**
シェルの環境の要素。

**イーサネット**
コンピュータネットワークにおけるデータ転送用の規格。

**EXT2 (Second Extended File System)**
Linuxでサポートされているファイルシステム。

**FAQ (Frequently Asked Questions)**
よくある質問への回答を提供するドキュメントの頭字語。

**ファイアウォール**
外部からの権限のないアクセスに対してローカルネットワークを保護するためにネットワークトラフィックをフィルタ処理するメカニズム。

**FTP (File Transfer Protocol)**
ネットワーク経由のファイル転送用のTCP/IPベースのプロトコル。

**GNOME (GNU Network Object Model Environment)**
Linux向けのグラフィカルデスクトップ環境。

**GNU (GNU is Not Unix)**
GNUとは、Free Software Foundation (FSF)のプロジェクトのことです。GNUプロジェクトの目的は、機能が充実したフリーのUNIXスタイルのオペレーティングシステムを作成することです。「フリー」には、*無料*という意味はあまり含まれておらず、*自由*という意味が含まれています。つまり、ソフトウェアを取得、修正、変更する権利があるということです。詳細については、権威あるGNU Manifesto (`http://www.gnu.org/gnu/manifesto.html`)を参照してください。法的には、GNUソフトウェアは、GNU General Public License(*GPL*) (`http://www.gnu.org/copyleft/gpl.html`)、およびGNU Lesser General Public License (*LGPL*) (`http://www.gnu.org/copyleft/lgpl.html`)で保護されています。Linuxカーネルは、GPLの対象であり、このプロジェクト(特にツール類)の恩恵を受けますが、それと同列に考えないでください。

**GPL (GNU General Public License)**
GNUを参照。

**ホームディレクトリ**
ファイルシステムにおいて特定のユーザに属しているプライベートディレクトリ(通常は/home/<username>)。スーパーユーザrootを除き、そのユーザだけが自分のホームディレクトリにフルにアクセスできます。

**ホスト名**
マシンの名前。一般に、この名前でネットワーク上のマシンに到達できます。

**HTML (Hypertext Markup Language)**
WWW (World Wide Web)で使用されるテキストドキュメント用のマークアップ言語。通常、HTMLドキュメントはブラウザで表示されます。

**HTTP (Hypertext Transfer Protocol)**
WWW (World Wide Web)でのドキュメントの要求方法や転送方法を定義しているネットワークプロトコル。通常、ドキュメントはサーバから提供されるHTMLページであり、それをユーザがブラウザを介して要求します。

**IDE (Integrated Drive Electronics)**
ハードディスクを接続するための規格。

**インターネット**
TCP/IPに基づく世界規模のコンピュータネットワーク

**IPアドレス**
TCP/IPネットワークにおけるコンピュータの一意の(32ビット)アドレス。一般に、ピリオド(.)で区切られた4つの10進数(192.168.10.1など)として記述されます。

**IRQ (Interrupt Request)**
ハードウェアまたはソフトウェアから実行できる何らかのアクションの(非同期)要求。ほとんどのIRQは、オペレーティングシステムで処理されます。

**ISDN (Integrated Services Digital Network)**
電話網でのデジタルデータ転送用の規格。

**KDE (K Desktop Environment)**
Linux向けのグラフィカルデスクトップ環境。

**カーネル**
カーネルとは、オペレーティングシステムのコアコンポーネントのことです。カーネルでは、メモリやファイルシステムの管理、ハードウェアデバイスとの通信用ドライバの組み込み、プロセスやネットワークの処理などが行われます。

**LAN (Local Area Network)**
LANとは、ある程度小規模なローカルネットワークのことです。

**LILO (Linux Loader)**
ハードディスクのブートセクタにインストールされる小さいプログラムであり、Linuxまたは別のオペレーティングシステムを起動します。

**リンク**
リンク(ファイルシステム内)とは、ファイルへのポインタのことです。ハードリンクとシンボリックリンクがあります。ハードリンクはファイルシステム内の正確な位置を示しますが、シンボリックリンクはそれぞれの名前を示すだけです。

**Linux**
GPL (GNU)の下で自由に配布される高性能なUNIX型のオペレーティングシステムコア。この名前は、頭字語(*Linus' uniX*)であり、作者のLinus Torvaldsを示しています。狭義には、この名前はカーネルそのものを示すだけですが、通常、*Linux*という用語は広くシステム全体の意味として理解されています。

**ログイン**
コンピュータシステムやネットワークにアクセスするための、ユーザ名とパスワードによるユーザの認証。

**ログアウト**
対話型Linuxセッションを閉じる手順。

**メインメモリ**
ほとんど遅延なくランダムにアクセスできる物理メモリ。一般に、メインメモリはRAM (Random Access Memory)と呼ばれます。

**manページ**
UNIXシステムの基本的な形式のドキュメント。コマンド`man`を使用して読むことができます。通常、manページはリファレンスのスタイルで作成されます。

**MBR (Master Boot Record)**
ハードディスクの最初の物理セクタ。その内容がメインメモリに読み込まれ、BIOSによって実行されます。次に、このコードで、ハードディスクパーティションのオペレーティングシステムまたは高機能のブートローダ(LILO、GRUBなど)を読み込みます。

**MD5**
ハッシュ値(ファイルのMD5チェックサム)を生成するためのアルゴリズム。このチェックサムの生成方法では、元のファイルとMD5チェックサムが同じで内容が異なるファイルを作成することはほぼ不可能です。

**マウント**
システムのディレクトリツリーにファイルシステムを結合するプロセス。

**MP3**
オーディオファイル用の(非可逆)圧縮アルゴリズム。このアルゴリズムでは、データサイズが非圧縮オーディオファイルに比べて約10分の1になります。

**マルチタスキング**
複数のプロセスを(実質的に)同時に実行するためのオペレーティングシステムの機能。

**マルチユーザ**
1台のコンピュータで同時に複数のユーザが作業できるようにするためのオペレーティングシステムの機能。

**ネットワーク**
複数のコンピュータ間の相互接続。これにより、コンピュータ間でデータ転送できるようになります。ネットワーク経由で要求を送信するコンピュータのことを一般にクライアントといい、要求(文書の配信など)に応じるコンピュータのことをサーバといいます。

**NFS (Network File System)**
ネットワーク経由でファイルシステムにアクセスするためのプロトコル。

**NIS (Network Information Service)**
ネットワークにおいて一元化されたユーザ管理システム。ユーザ名とパスワードは、NISによってネットワーク規模で管理できます。

**オペレーティングシステム**
カーネルを参照。

**パーティション**
ファイルシステムまたはスワップ領域を含むハードディスクのセクション。

**パス**
ファイルシステム内のファイルの位置を一意に示す記述。

**プラグアンドプレイ**
自動ハードウェア検出/設定プロトコル。

**プロセス**
実行中のプログラム。タスクと呼ばれる場合もあります。

**プロセッサ**
プロセッサ(CPU : Central Processing Unit)とは、メインメモリに格納されたマシンコードを実行するマイクロチップのことで、コンピュータの*中枢*です。

**プロンプト**
各コマンドラインの先頭に出力される短い(設定可能な)文字列です。通常、プロンプトには現在の作業ディレクトリが出力されます。

**プロトコル**
ハードウェア、ソフトウェア、またはネットワークのインタフェースおよび通信方法を定義する規格。たとえば、HTTP、FTPなどのプロトコルがあります。

**プロキシ**
一般に、インターネットから転送されるデータの中間記憶領域として機能するコンピュータのことをいいます。同じドキュメントが2度以上要求される場合、2回目の要求は非常に高速に処理されます。プロキシを利用するコンピュータは、それぞれの要求をプロキシを介して発行するように設定する必要があります。

**RAM (Random Access Memory)**
メインメモリを参照。

**ReiserFS**
潜在的な不整合を高速に修復できるファイルシステムタイプ。そのような不整合は、電源障害の発生などのため、ファイルシステムをマウント解除せずにオペレーティングシステムがシャットダウンされた場合に発生する可能性があります。

**root**
スーパーユーザアカウント。スーパーユーザは、すべてのパーミッションを持っています。このアカウントは、管理タスクに使用し、通常の作業には使用しないでください。

**ルートディレクトリ**
ファイルシステム階層のベースディレクトリ。UNIXでは、ルートディレクトリは/で表されます。

**SCSI (Small Computer Systems Interface)**
ハードディスクやその他のデバイス(スキャナ、テープなど)を接続するための規格。

**サーバ**
ネットワーク経由でのサービス提供専用のコンピュータまたはプログラム。サービスの例としては、HTTP、DNS、FTPなどがあります。

**シェル**
コマンドを実行できる対話型プログラム。Bash、zsh、tcshなど何種類かのシェルがあります。各タイプのシェルには、特定のプログラミング言語が用意されています。

**SMTP (Simple Mail Transfer Protocol)**
ネットワーク経由で電子メールを転送するためのプロトコル。

**SSL (Secure Socket Layer)**
HTTPデータを転送するための暗号化プロトコル。

**スーパーユーザ**
rootを参照。

**システム管理者**
rootを参照。

**タスク**
プロセスを参照。

**TCP/IP**
インターネットで使用される通信プロトコル。また、ほとんどのローカルネットワークにも使用されます。

**telnet**
telnetは、リモートホストとの通信用のプロトコルです。リモートログインの場合は、暗号化接続を提供するSSHの方が基本的にtelnetよりも優先されます。

**端末**
以前は、中央のコンピュータに接続されたキーボードとモニタ組み合わせの意味でした。現在、この用語は、実際の端末をエミュレートするプログラム(xtermなど)に使用されています。

**Tux**
Linux penguinの名前。`http://www.sjbaker.org/tux/`を参照。

**UNIX**
UNIXは、(商標かつ)オペレーティングシステムのタイプです。

**URL (Uniform Resource Locator)**
プロトコルを構成する、ネットワーク内のリソースの指定情報(`http://`など)、およびホストやドメイン(`www.suse.de`など)とドキュメント(`/us/company/index.html`など)の名前。すべて指定したURLの例は、`http://www.suse.de/us/company/index.html`のようになります。.

**ユーザディレクトリ**
ホームディレクトリを参照。

**VESA (Video Electronics Standard Association)**
特にビデオの規格を定義する業界団体。

**ワイルドカード**
1文字(記号: ?)または複数文字(記号: *)のプレースホルダ。これらは正規表現の要素です。

**ウィンドウマネージャ**
X Window Systemの最上位で実行するプログラムであり、ウィンドウのサイズの変更や移動などの操作を行うことができます。ウィンドウマネージャは、ウィンドウのタイトルやウィンドウ枠のようなウィンドウ装飾も担当します。動作と外観は、ユーザがカスタマイズできます。

**WWW (World Wide Web)**
HTTPプロトコルに基づく、Webブラウザで表示できるドキュメント、ファイル、イメージなどのハイパーリンクされたコレクションのことです。

**X Window System**
X Window Systemは、広範囲にわたるコンピュータで動作するネットワークベースのウィンドウシステムであり、線や長方形などの描画用のプリミティブを提供します。このシステムは、ハードウェアとウィンドウマネージャの中間レイヤーです。

**X11**
X Window Systemのバージョン11。

**YaST (Yet another Setup Tool)**
SUSE Linuxシステムアシスタント。

**YP**
NISを参照。

# Index

## A



## B



## C



## D

## M



## N



## O



## P



## R

## S

## T



## U



## V



## W



## X



## Y