管理ガイド

SUSE Linux Enterprise Server 12

管理ガイド

SUSE Linux Enterprise Server 12

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

発行日: Sep 30 2014

SUSE Linux Products GmbH
Maxfeldstr. 5
90409 Nürnberg
GERMANY
https://www.suse.com/documentation

Copyright © 2006–2014 SUSE LLC and contributors. All rights reserved.

この文書は、GNUフリー文書ライセンスのバージョン1.2または(オプションとして)バージョン1.3の条項に従って、複製、頒布、および/または改変が許可されています。ただし、この著作権表示およびライセンスは変更せずに記載すること。ライセンスバージョン1.2のコピーは、「GNUフリー文書ライセンス」セクションに含まれています。

SUSEおよびNovellの商標については、商標とサービスマークの一覧http://www.novell.com/company/legal/trademarks/tmlist.html を参照してください。他のすべての第三者の商標は、各商標権者が所有しています。商標記号 (®、#など) は、SUSEまたはNovellの商標を示し、アスタリスク(*)は、サードパーティの商標を示します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは絶対に正確であることを保証するものではありません。SUSE LLC、その関係者、著者、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

目次

このガイドについて

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

このガイドでは、次の内容が取り上げられています。

サポートと共通タスク

SUSE Linux Enterpriseには、システムのさまざまな側面をカスタマイズするための幅広いツールが用意されています。この部分では、これらのツールの一部を紹介しています。利用できるさまざまなデバイス技術、可用性の高い構成、および高度な管理機能など、管理者にとって役立つさまざまな機能を紹介します。

システム

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

サービス

SUSE Linux Enterpriseは、ネットワークオペレーティングシステムとして設計されています。このオペレーティングシステムは、DNS、DHCP、Web、プロキシ、および認証サービスなどの幅広いネットワークサービスを提供しています。また、MS Windowsクライアント/サーバなどとの混在環境にも、柔軟に対応することができます。

モバイルコンピュータ

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

トラブルシューティング

トラブルシューティングでは、詳細情報が必要な場合や特定のタスクを自分のシステムで実行する場合に、ヘルプや追加ドキュメントを見つけられる場所の概要がわかります。また、最も頻繁に発生する問題や厄介事も収録されており、それらの問題を自分で解決する方法を学ぶことができます。

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

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

1 利用可能なマニュアル

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

項目「インストールクイックスタート」

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

ブック「導入ガイド」

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

管理ガイド

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

Book “Virtualization Guide”

仮想化技術全般について説明し、仮想化統合インタフェースであるlibvirt、および特定のハイパーバイザの詳細情報を紹介します。

Book “Storage Administration Guide”

SUSE Linux Enterprise Serverサーバでストレージデバイスを管理する方法を説明します。

Book “AutoYaST”

AutoYaSTは、インストールデータと設定データを含むAutoYaSTプロファイルを使用して、ユーザの介入なしで、自動的に、1つ以上のSUSE Linux Enterpriseシステムをインストールするためのシステムです。マニュアルに従って、自動インストールの基本的な手順(準備、インストール、および設定)を実行できます。

Book “Security Guide”

システムセキュリティの基本概念を紹介し、ローカルセキュリティ/ネットワークセキュリティの両方の側面を説明します。AppArmorなど製品に付属するセキュリティソフトウェアや、セキュリティ関連イベントの情報を確実に収集する監査システムの使用方法を説明します。

Book “Security and Hardening Guide”

セキュアなSUSE Linux Enterprise Server、およびそのインストールのセキュリティを保護し強化するために必要なその他のポストインストールプロセスのインストールおよび設定について詳しく説明します。セキュリティ関連の選択や決定を行う管理者をサポートします。

Book “System Analysis and Tuning Guide”

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

ブック「GNOMEユーザガイド」

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

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

2 フィードバック

次のフィードバックチャネルがあります。

バグと機能拡張の要求

ご使用の製品に利用できるサービスとサポートのオプションについては、http://www.suse.com/support/ を参照してください。

製品のコンポーネントのバグをレポートするには、http://www.suse.com/mysupport にアクセスしてログインし、[Submit New SR (新規SRの送信)]を選択します。

ユーザからのコメント

本マニュアルおよびこの製品に含まれているその他のマニュアルについて、皆様のご意見やご要望をお寄せください。オンラインドキュメントの各ページの下にあるユーザコメント機能を使用するか、またはhttp://www.suse.com/doc/feedback.html にアクセスして、コメントを入力してください。

メール

この製品のドキュメントについてのフィードバックは、`doc-team@suse.de` 宛のメールでも送信できます。ドキュメントのタイトル、製品のバージョン、およびドキュメントの発行日を明記してください。エラーの報告または機能拡張の提案では、問題について簡潔に説明し、対応するセクション番号とページ(またはURL)をお知らせください。

3 マニュアルの表記規則

本書では、次の書体を使用しています。

- `/etc/passwd`: ディレクトリ名とファイル名
- `placeholder`: `placeholder` は、実際の値で置き換えられます
- `PATH`:環境変数PATH
- `ls`, `--help`:コマンド、オプション、およびパラメータ
- `user`:ユーザまたはグループ
- `Alt`、`Alt`–`F1`: 押すキーまたはキーの組み合わせ。キーはキーボード上の表記と同様に大文字で示します。
- [ファイル]、[ファイル] › [名前を付けて保存]: メニュー項目、ボタン
- `x86_64` この説明は、x86_64アーキテクチャにのみ当てはまります。矢印は、テキストブロックの先頭と終わりを示します。
  `POWER, System z` この説明は、`System z` および `POWER` の各アーキテクチャにのみ当てはまります。矢印は、テキストブロックの先頭と終わりを示します。
- Dancing Penguins (「Penguins」の章、↑他のマニュアル):他のマニュアルの章への参照です。

I サポートと共通タスク

1 YaSTオンラインアップデート

SUSEはお買い上げの製品に対し、継続的にソフトウェアセキュリティのアップデートを提供します。デフォルトでは、システムを最新の状態に維持するために更新アプレットが使用されます。アップデートアプレットの詳細については、ブック 「導入ガイド」 9 「ソフトウェアをインストールまたは削除する」9.4 「システムのアップデート」を参照してください。この章では、ソフトウェアパッケージをアップデートする代替ツールとして、YaSTオンラインアップデートを紹介します。

SUSE® Linux Enterprise Serverの現在のパッチは、アップデートソフトウェアリポジトリから入手できます。インストール時に製品を登録した場合、アップデートリポジトリはすでに設定されています。SUSE Linux Enterprise Serverを登録していない場合は、YaSTで[SUSEカスタマーセンターの環境設定]を起動して登録できます。または、信頼できるソースから、手動でアップデートリポジトリを追加することもできます。リポジトリを追加または削除するには、YaSTで、[ソフトウェア] › [ソフトウェアリポジトリ]の順に選択して、リポジトリマネージャを起動します。リポジトリマネージャの詳細については、ブック「導入ガイド」9「ソフトウェアをインストールまたは削除する」9.3「ソフトウェアリポジトリおよびサービスの操作」を参照してください。

注記: アップデートカタログのアクセス時のエラー

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

アップデートカタログへのアクセスが拒否される場合は、SUSEカスタマーセンターにアクセスして登録を確認することを推奨する警告メッセージが表示されます。SUSEカスタマーセンターには、https://scc.suse.com// でアクセスできます。

SUSEは、各種の関連性レベルを持つアップデートを提供します。

セキュリティアップデート

セキュリティアップデートは、重大なセキュリティハザードを修復するので、必ずインストールする必要があります。

推奨アップデート

コンピュータに損害を与える可能性のある問題を修復します。

オプションアップデート

セキュリティに関連しない問題を修復したり、拡張機能を提供します。

1.1 オンライン更新ダイアログ

[オンライン更新]ダイアログを開くには、YaSTを起動し、[ソフトウェア] › [オンライン更新]の順に選択します。または、`yast2 online_update`で、コマンドラインからオンラインアップデートを開始します。

[オンラインアップデート]ウィンドウは、4つのセクションから成り立っています。

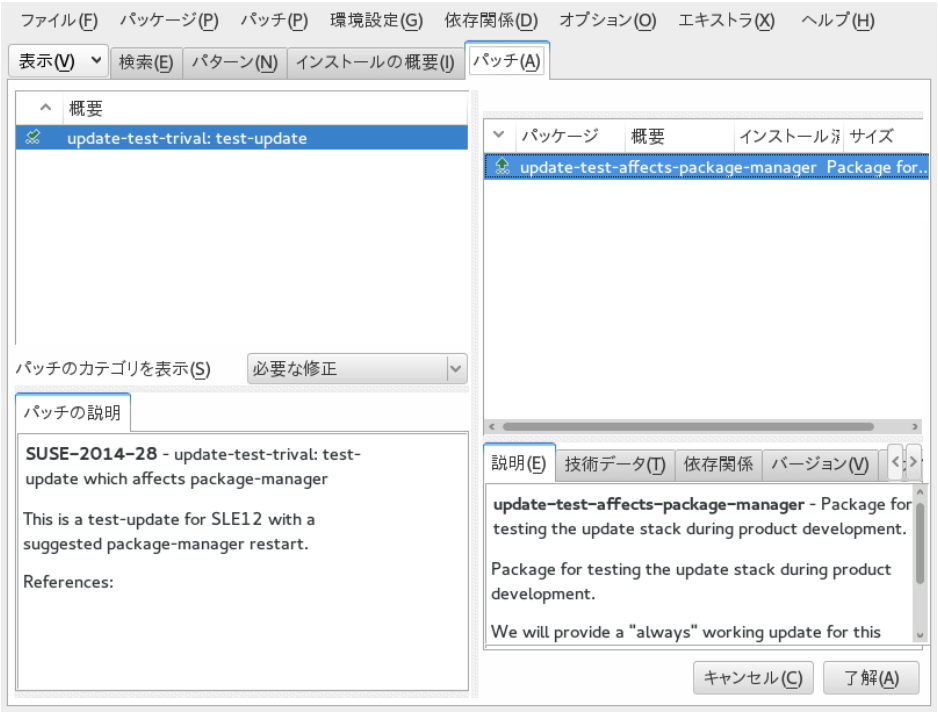


図 1.1 YASTオンラインアップデート

左側の[概要]セクションには、SUSE Linux Enterprise Serverの使用可能なパッチが一覧にされます。パッチはセキュリティの関連性によってソートされます(`security`、`recommended`、および`optional`)。[概要]セクションのビューは、[パッチのカテゴリを表示]から、以下のオプションの1つを選択することによって変更できます。

[Needed Patches](デフォルトビュー)
: システムにインストールされたパッケージに適用される、インストールされなかったパッチ。

[Unneeded Patches]
: システムにインストールされていないパッケージに適用されるパッチか、または(該当するセキュリティがすでに別のソースで更新されたので)要件がすでに満たされているパッチ。

[すべてのパッチ]
: SUSE Linux Enterprise Serverで使用可能なすべてのパッチ。

[概要]セクションの各リストエントリは、記号とパッチ名で構成されています。可能な記号とそれらの意味の概要については、Shift–F1を押してください。Securityパッチおよび Recommendedパッチで要求されるアクションは、自動的に設定されます。アクションは、[自動インストール]、[自動更新]、および[自動削除]です。

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

[概要]セクションでエントリを選択すると、ダイアログの左下隅に短い[パッチの説明]が表示されます。左上のセクションには、選択されたパッチに含まれているパッケージが一覧されます(パッチは複数のパッケージから成ることがあります)。右上のセクションでエントリをクリックすると、パッチに含まれている各パッケージの詳細が表示されます。

1.2 パッチのインストール

YaSTオンラインアップデートのダイアログでは、すべての利用可能なパッチを一度にインストールしたり、システムに適用したいパッチを手動で選択したりできます。システムに適用済みのパッチを元に戻すこともできます。

デフォルトでは、お使いのシステムで現在使用できる新しいパッチ(ただし、optional以外)　はすべて、すでにインストール用にマークされています。[受諾]または[適用]をクリックすると、これらのパッチが自動的に適用されます。1つまたは複数のパッチでシステムの再起動が必要な場合、パッチのインストールが開始される前にその旨が通知されます。選択したパッチのインストールを続行し、再起動が必要なすべてのパッチのインストールをスキップしてして残りをインストールするか、パッチの手動選択に戻ることを決定できます。

手順 1.1 YASTオンラインアップデートによるパッチの適用

1. YaSTを起動して、[ソフトウェア] › [オンライン更新]の順に選択します。
2. システムで現在使用可能なすべての新しいパッチ(ただし、optional以外)を自動的に適用するには、[適用]または[受諾]のクリックで続行して事前選択されているパッチのインストールを開始します。
3. まず、適用したいパッチの選択を変更します。
    a. インタフェースが提供する各フィルタとビューを使用します。詳細については、1.1項「オンライン更新ダイアログ」を参照してください。

b. ニーズと好みに従ってパッチを選択または選択解除するには、パッチを右クリックしてコンテキストメニューから各アクションを選択します。

> **重要: セキュリティアップデートは必ず適用する**
>
> 十分な理由がない限り、`security` 関係のパッチは選択解除しないでください。これらのパッチは、重大なセキュリティハザードを修復し、システムの悪用を防ぎます。

c. 大部分のパッチには、複数のパッケージのアップデートが含まれています。単一パッケージに対するアクションを変更する場合は、パッケージビューでパッケージを右クリックしてアクションを選択します。

d. 選択内容を確認し、選択したパッチを適用するには、[適用]または[受諾]をクリックして続行します。

4. インストールの完了後、[完了]をクリックして、YaSTの[オンライン更新]を終了します。これで、システムが最新の状態になりました。

1.3 自動オンラインアップデート

YaSTでは、毎日、毎週、または毎月のスケジュールで自動アップデートを設定することもできます。各モジュールを使用するには、まず、`yast2-online-update-configuration` パッケージをインストールする必要があります。

デフォルトでは、アップデートは、デルタRPMとしてダウンロードされます。デルタRPMからのRPMパッケージの再構築は、メモリとプロセッサを消費するので、セットアップまたはハードウェア構成によっては、パフォーマンス上の理由によりデルタRPMの使用を無効にする必要があります。

一部のパッチ(カーネルのアップデートやライセンス契約を必要とするパッケージなど)ではユーザによる操作が必要で、これによって自動アップデート手順が停止します。ユーザによる操作が必要なパッチをスキップするよう設定できます。

手順 1.2 自動オンラインアップデートの設定

1. インストール後、YaSTを起動し、[ソフトウェア] › [オンラインアップデートの設定]の順に選択します。
   または、コマンドラインから、`yast2 online_update_configuration` を使用してモジュールを起動します。
2. [自動オンラインアップデート]を有効にします。

3. 更新間隔として[毎日]、[毎週]、または[毎月]を選択します。
4. ライセンス契約に自動的に同意するには、[ライセンスに同意する]を有効にします。
5. 更新手順を完全に自動的に進行させたい場合は、[インタラクティブパッチをスキップする]を選択します。

   重要: パッチのスキップ

   介入を必要とするパッケージのスキップを選択した場合は、時折、[オンライン更新]を手動で実行して、それらのパッチもインストールしてください。さもないと、重要なパッチをインストールできないことがあります。

6. アップデートパッケージによって推奨されるすべてのパッケージを自動的にインストールするには、[推奨パッケージを含む]を有効にします。
7. デルタRPMの使用を無効にするには(パフォーマンス上の理由)、[delta rpmを使用する]を無効にします。
8. セキュリティや推奨など、カテゴリ別にパッチをフィルタリングするには、[Filter by Category]を有効にしてリストから適切なカテゴリを追加します。選択したカテゴリのパッチのみがインストールされます。それ以外はスキップされます。
9. 入力した設定を確認して、[[OK]をクリックします。

2 サポート用システム情報の収集

マシンに関連するすべてのシステム情報の概要を素早く参照できるよう、SUSE Linux Enterprise Serverでは `hostinfo` パッケージが提供されています。このパッケージは、システム管理者が汚染カーネル(サポートされていません)やサードパーティパッケージがマシンにインストールされていないかどうかを確認する場合にも役立ちます。

問題がある場合は、`supportconfig` コマンドラインツールまたはYaST[] サポートモジュールで詳細なシステムレポートを作成できます。どちらも、現在のカーネルのバージョン、ハードウェア、インストールされているパッケージ、パーティション設定などのシステム情報を収集します。結果は、複数のファイルのTARアーカイブになります。サービス要求(SR)を開いた後、そのTARアーカイブをグローバルテクニカルサポートにアップロードできます。これは、レポートされた問題を特定したり、問題解決を支援したりするのに役立ちます。

また、既知の問題がないかどうか `supportconfig` の出力を分析することで、問題解決を迅速化できます。このために、SUSE Linux Enterprise Serverでは、`Supportconfig Analysis` (SCA)用のアプライアンスとコマンドラインツールの両方が提供されています。

2.1 現在のシステム情報の表示

サーバへのログイン時に関連するすべてのシステム情報を素早く簡単に参照するには、パッケージ `hostinfo` を使用します。このパッケージをマシンにインストールすると、そのマシンにログインしたすべての `root` ユーザに対して、コンソールに次の情報が表示されます。

例 2.1 rootとしてログインしたときの`hostinfo`の出力

```
Hostname:                 earth
Current As Of:            Wed 12 Mar 2014 03:57:05 PM CET
Distribution:             SUSE Linux Enterprise Server 12
 -Service Pack:           0
Architecture:             x86_64
Kernel Version:           3.12.12-3-default
 -Installed:              Mon 10 Mar 2014 03:15:05 PM CET
 -Status:                 Not Tainted
Last Updated Package:     Wed 12 Mar 2014 03:56:43 PM CET
 -Patches Needed:         0
 -Security:               0
 -3rd Party Packages:     0
```

```
IPv4 Address:              ens3 192.168.1.1
Total/Free/+Cache Memory: 983/95/383 MB (38% Free)
Hard Disk:                 /dev/sda 10 GB
```

この出力でカーネルのステータスが`tainted`と表示される場合、詳細については、2.5項「カーネルモジュールのサポート」を参照してください。

2.2 Supportconfigによるシステム情報の収集

グローバルテクニカルサポートに提出できる詳細なシステム情報のTARアーカイブを作成するには、`supportconfig`コマンドラインツールまたはYaST[]サポートモジュールを使用します。このコマンドラインツールは、デフォルトでインストールされるパッケージ`supportutils`によって提供されます。YaST[]サポートモジュールも、このコマンドラインツールが基になっています。

2.2.1 サービス要求番号の作成

Supportconfigアーカイブはいつでも生成できます。ただし、supportconfigデータをグローバルテクニカルサポートに提出するには、まずサービス要求番号を生成する必要があります。サービス要求番号はアーカイブをサポートにアップロードするために必要です。

サービス要求を作成するには、http://www.novell.com/center/eservice にアクセスして、画面の指示に従います。11桁のサービス要求番号を記録します。

注記: プライバシーポリシー

SUSEおよびNovellは、システムレポートを機密データとして扱います。プライバシーに関する取り組みの詳細については、http://www.novell.com/company/legal/privacy/ を参照してください。

2.2.2 アップロード先

サービス要求番号を作成したら、supportconfigアーカイブをグローバルテクニカルサポートにアップロードできます。手順2.1「YaSTを使用したサポートへの情報の送信」または手順2.2「コマンドラインからのサポートへの情報の送信」を参照してください。次のいずれかのアップロードターゲットを使用します。

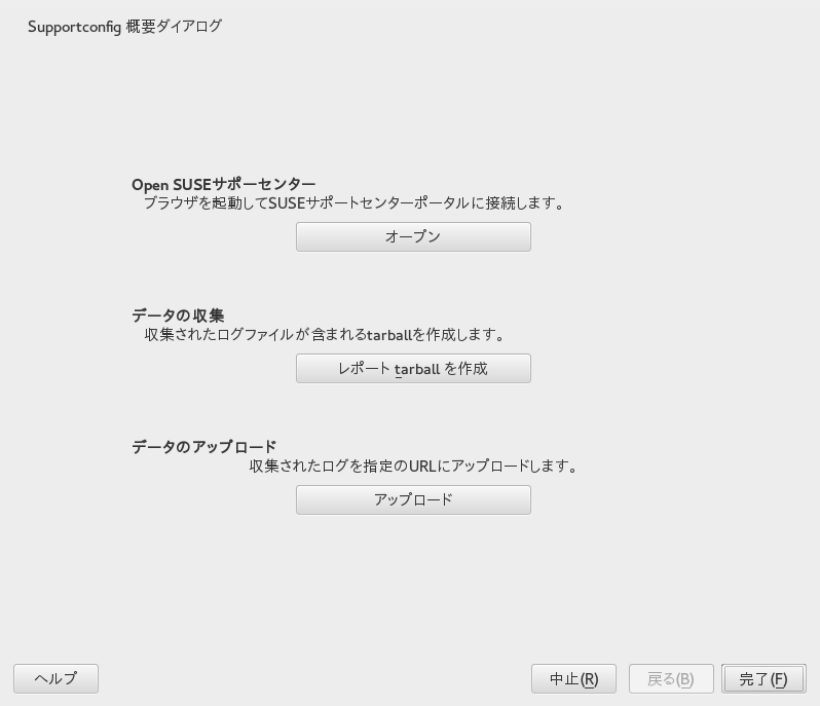
- 米国内のお客様: ftp://ftp.novell.com/incoming
- EMEA (ヨーロッパ、中東、およびアフリカ): ftp://support-ftp.suse.com/in

または、サービス要求URLhttp://www.novell.com/center/eservice を使用してTARアーカイブを手動でサービス要求に添付することもできます。

2.2.3 YaSTでのSupportconfigアーカイブの作成

YaSTでシステム情報を収集するには、次の手順に従います。

1. YaSTを起動して、[] サポートモジュールを開きます。



2. [Create report tarball (レポートtarballを作成)] をクリックします。

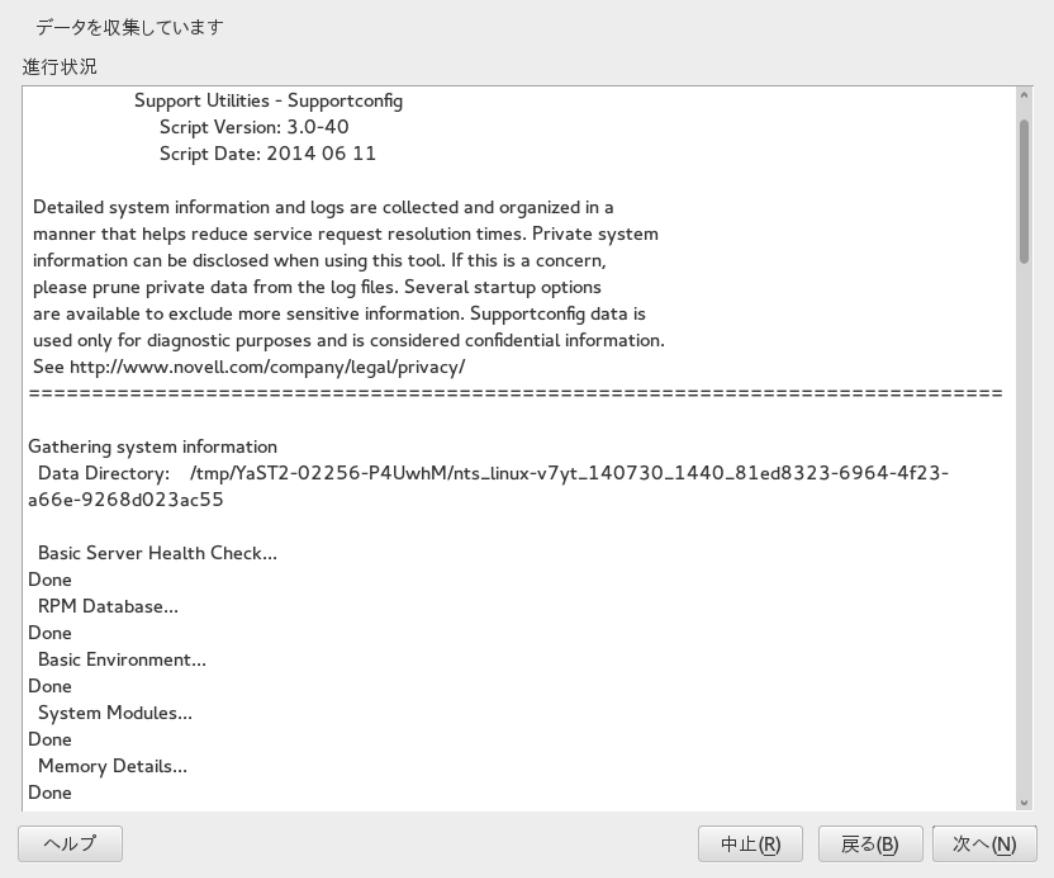
3. 次のウィンドウで、ラジオボタンリストからsupportconfigオプションを1つ選択します。デフォルトでは、[Use Custom (Expert) Settings (カスタム(エキスパート)設定を使用します)] が事前選択されています。最初にレポート機能をテストしたい場合は、[Only gather a minimum amount of info (最小限度の情報のみを収集する)] を使用します。その他のオプションに関する背景情報については、`supportconfig` のマニュアルページを参照してください。
[次へ] で続行します。

4. 連絡先情報を入力します。情報は`basic-environment.txt`という名前のファイルに書き込まれ、作成するアーカイブに組み込まれます。

5. 情報収集プロセスの終了時にアーカイブをグローバルテクニカルサポートに送信する場合、[Upload Information (情報のアップロード)]に入力する必要があります。YaSTによって自動的にアップロードサーバが提案されます。サーバを変更する場合、利用可能なアップロードサーバの詳細については、2.2.2項「アップロード先」を参照してください。
アーカイブを後で送信する場合、[Upload Information (情報のアップロード)]は空白のままで構いません。

6. [次へ]で続行します。

7. 情報の収集が開始します。



プロセスが完了したら、[次へ]で続行します。

8. データ収集を確認します。ログファイルのファイル名を[File Name (ファイル名)]で選択して、YaSTで内容を表示します。サポートへの送信前にTARアーカイブから除外したいファイルを削除するには、[Remove from Data (データから削除)]を使用します。[次へ]で続行します。

9. TARアーカイブを保存します。YaSTモジュールを `root` ユーザとして起動した場合、デフォルトではアーカイブを `/var/log` に保存するよう提案されます(そうでない場合はホームディレクトリ)。ファイル名の形式は、`nts_HOST_DATE_TIME.tbz` です。

10. アーカイブをサポートに直接アップロードする場合は、[Upload log files tarball to URL (ログファイルtarballをURLへアップロード)]が有効になっていることを確認してください。ここに表示される[Upload Target (アップロードターゲット)]は、ステップ 5でYaSTによって提案されたものです。アップロードターゲットを変更する場合は、2.2.2項「アップロード先」で、利用可能なアップロードサーバの詳細情報を参照します。

11. アップロードをスキップする場合は、[Upload log files tarball to URL (ログファイルtarballをURLへアップロード)]を無効にします。

12. 変更内容を確認し、YaSTモジュールを閉じます。

2.2.4 コマンドラインからのsupportconfigアーカイブの作成

次の手順は、supportconfigアーカイブをサポートに直接送信せずにアーカイブを作成する方法を示しています。アーカイブをアップロードするには、特定のオプションを指定してコマンドを実行する必要があります。手順2.2「コマンドラインからのサポートへの情報の送信」を参照してください。

1. シェルを開き `root` になります。

2. オプションなしで `supportconfig` を実行します。デフォルトのシステム情報が収集されます。

3. ツールが操作を完了するまで待機します。

4. デフォルトのアーカイブ場所は `/var/log` で、ファイル名の形式は `nts_HOST_DATE_TIME.tbz` です。

2.2.5 Supportconfigの一般的なオプション

`supportconfig` ユーティリティは、通常、オプションなしで呼び出されます。`supportconfig -h` で、すべてのオプションを一覧表示するか、マニュアルページを参照してください。よくある使用事例については、次のリストで簡単に説明します。

収集する情報のサイズを削減する

最小オプション(`-m`)を使用します。

```
supportconfig -m
```

情報を特定のトピックに限定する

すでにデフォルトの`supportconfig`出力で問題を特定し、その問題が特定の領域または機能セットにのみ関係することが判明している場合は、`supportconfig`の次回実行時に収集する情報を特定の領域に限定できます。たとえば、LVMに問題があることを検出した場合に、最近変更したLVMの設定をテストしたいときは、LVMに関連する最小限のsupportconfig情報のみを収集するのが適切です。

```
supportconfig -i LVM
```

収集する情報を特定の領域に限定する場合に使用できる機能のキーワードを網羅したリストについては、次のコマンドを実行します。

```
supportconfig -F
```

追加の連絡先情報を出力に含める

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

(すべてを1行に記述)

ローテーション済みログファイルの収集

```
supportconfig -l
```

これは、大規模なログを行う環境や、再起動後のsyslogによるログファイルのローテーション時にカーネルクラッシュが発生した場合に特に有効です。

2.3 グローバルテクニカルサポートへの情報の送信

システム情報をグローバルテクニカルサポートへ送信するには、YaST[]サポートモジュールまたは`supportconfig`コマンドラインユーティリティを使用します。サーバに問題がありサポートの支援が必要な場合、まずサービス要求を開く必要があります。詳細については、2.2.1項 「サービス要求番号の作成」を参照してください。

次の例では、実際のサービス要求番号のプレースホルダとして`12345678901`を使用しています。`12345678901`は、2.2.1項 「サービス要求番号の作成」で作成したサービス要求番号に置き換えてください。

手順 2.1 YASTを使用したサポートへの情報の送信

次の手順は、supportconfigアーカイブを作成済みであるものの、まだアップロードしていないことを想定しています。2.2.3項「YaSTでのSupportconfigアーカイブの作成」のステップ 4で説明されているように、アーカイブに連絡先情報が含まれていることを確認してください。supportconfigアーカイブの生成と送信を一度に行う方法については、2.2.3項「YaSTでのSupportconfigアーカイブの作成」を参照してください。

1. YaSTを起動して、[] サポートモジュールを開きます。
2. [アップロード]をクリックします。
3. 既存のsupportconfigアーカイブのパスを[Package with log files (ログファイルのあるパッケージ)]に指定するか、[参照]をクリックしてアーカイブを参照します。
4. YaSTによって自動的にアップロードサーバが提案されます。サーバを変更する場合、利用可能なアップロードサーバの詳細については、2.2.2項「アップロード先」を参照してください。

[次へ]で続行します。

5. [完了]をクリックします。

手順 2.2 コマンドラインからのサポートへの情報の送信

次の手順は、supportconfigアーカイブを作成済みであるものの、まだアップロードしていないことを想定しています。supportconfigアーカイブの生成と送信を一度に行う方法については、2.2.3項「YaSTでのSupportconfigアーカイブの作成」を参照してください。

1. インターネット接続のあるサーバの場合

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

    ```
    supportconfig -ur 12345678901
    ```

    b. 安全なアップロードターゲットには、次を使用します。

    ```
    supportconfig -ar 12345678901
    ```

2. インターネット接続のないサーバの場合

    a. 次を実行します。

    ```
    supportconfig -r 12345678901
    ```

    b. `/var/log/nts_SR12345678901 *tbz` アーカイブをいずれかのFTPサーバに手動でアップロードします。使用するサーバは世界のどの地域にいるかに応じて異なります。概要については、2.2.2項「アップロード先」を参照してください。

3. FTPサーバの着信ディレクトリにTARアーカイブが届くと、お客様のサービス要求に自動的に添付されます。

2.4 システム情報の分析

`supportconfig`で作成したシステムレポートで既知の問題がないかどうかを分析すると、問題の早期解決に役立ちます。このために、SUSE Linux Enterprise Serverでは、`Supportconfig Analysis` (SCA)用のアプライアンスとコマンドラインツールの両方が提供されています。SCAアプライアンスは非対話型のサーバサイドツールです。SCAツール(`scatool`)はクライアント側で動作し、コマンドラインから実行します。どちらのツールも、関係するサーバからのsupportconfigアーカイブを分析します。サーバでの初回の分析は、SCAアプライアンス、またはscatoolが実行されているワークステーションで行われます。分析サイクルは運用サーバ上では実行されません。

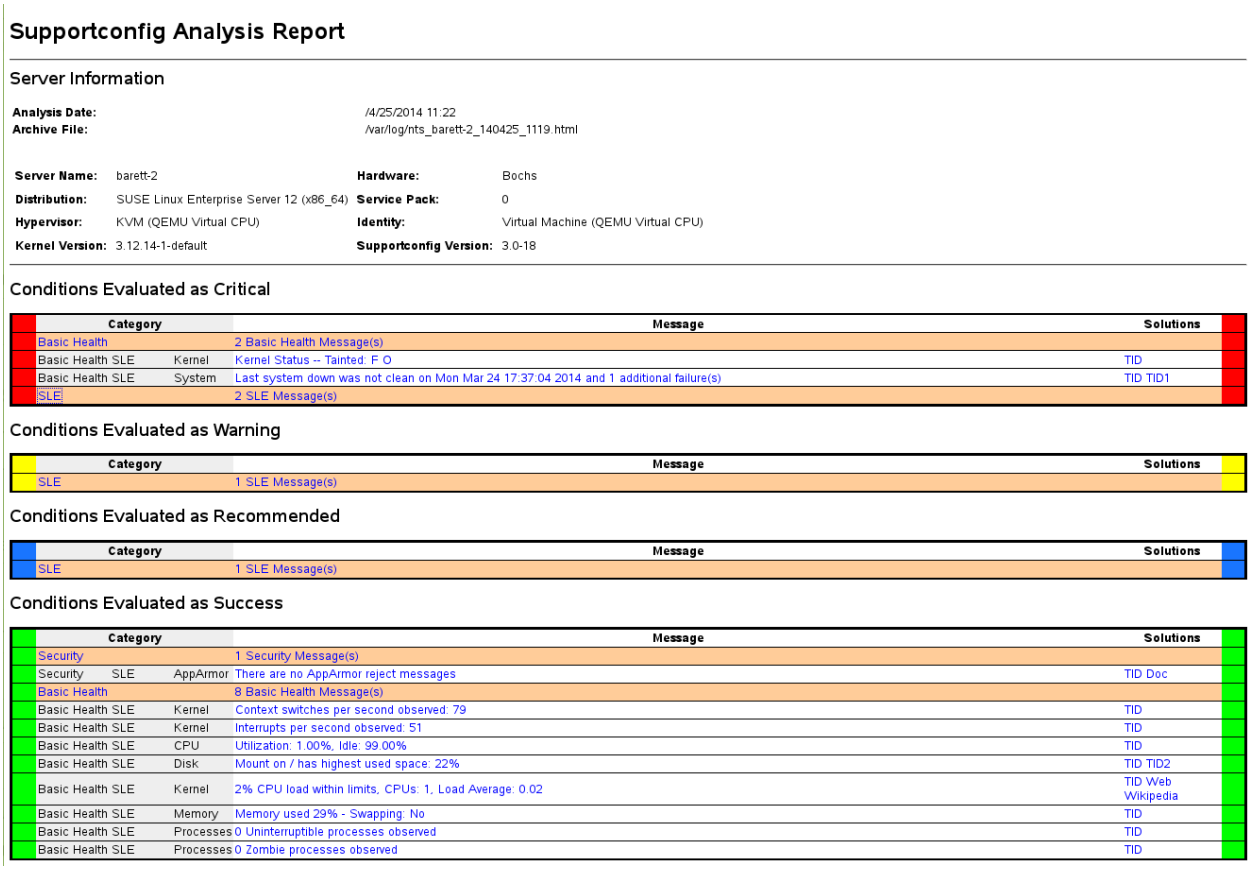
アプライアンスとコマンドラインツールのどちらにも、関連する製品のsupportconfig出力を分析できるようにする製品固有のパターンが追加で必要になります。各パターンは、特定の既知の問題がないかどうかsupportconfigアーカイブを解析して評価するスクリプトです。パターンはRPMパッケージとして提供されます。

たとえば、SUSE Linux Enterprise 11マシン(またはSCAアプライアンスサーバとして使用するマシン)上で生成されたsupportconfigアーカイブを分析する場合、SCAツールと共にパッケージ `sca-patterns-sle11` をインストールする必要があります。SUSE Linux Enterprise 10マシン上で生成されたsupportconfigアーカイブを分析するには、パッケージ `sca-patterns-sle10` が必要です。

独自のパターンを開発することもできます。これについては、2.4.3項 「カスタム分析パターンの開発」で簡単に説明されています。

2.4.1 SCAコマンドラインツール

SCAコマンドラインツールでは、`supportconfig` と、ローカルマシンにインストールされている特定の製品用の分析パターンの両方を使用してローカルマシンを分析できます。分析結果を示すHTMLレポートが作成されます。例については、図2.1「SCAツールによって生成されるHTMLレポート」を参照してください。



Supportconfig Analysis Report

Server Information

Analysis Date: /4/25/2014 11:22
Archive File: /var/log/nts_barett-2_140425_1119.html

Server Name:	barett-2	**Hardware:**	Bochs
Distribution:	SUSE Linux Enterprise Server 12 (x86_64)	**Service Pack:**	0
Hypervisor:	KVM (QEMU Virtual CPU)	**Identity:**	Virtual Machine (QEMU Virtual CPU)
Kernel Version:	3.12.14-1-default	**Supportconfig Version:**	3.0-18

Conditions Evaluated as Critical

Category			Message	Solutions
Basic Health			2 Basic Health Message(s)	
Basic Health	SLE	Kernel	Kernel Status -- Tainted: F O	TID
Basic Health	SLE	System	Last system down was not clean on Mon Mar 24 17:37:04 2014 and 1 additional failure(s)	TID TID1
SLE			2 SLE Message(s)	

Conditions Evaluated as Warning

Category	Message	Solutions
SLE	1 SLE Message(s)	

Conditions Evaluated as Recommended

Category	Message	Solutions
SLE	1 SLE Message(s)	

Conditions Evaluated as Success

Category			Message	Solutions
Security			1 Security Message(s)	
Security	SLE	AppArmor	There are no AppArmor reject messages	TID Doc
Basic Health			8 Basic Health Message(s)	
Basic Health	SLE	Kernel	Context switches per second observed: 79	TID
Basic Health	SLE	Kernel	Interrupts per second observed: 51	TID
Basic Health	SLE	CPU	Utilization: 1.00%, Idle: 99.00%	TID
Basic Health	SLE	Disk	Mount on / has highest used space: 22%	TID TID2
Basic Health	SLE	Kernel	2% CPU load within limits, CPUs: 1, Load Average: 0.02	TID Web Wikipedia
Basic Health	SLE	Memory	Memory used 29% - Swapping: No	TID
Basic Health	SLE	Processes	0 Uninterruptible processes observed	TID
Basic Health	SLE	Processes	0 Zombie processes observed	TID

図 2.1 SCAツールによって生成されるHTMLレポート

`scatool`コマンドは`sca-server-report`パッケージで提供されます。デフォルトではインストールされません。さらに、`sca-patterns-base`パッケージ、および`scatool`コマンドを実行するマシンにインストールされている製品に一致する製品固有の`sca-patterns-*`パッケージも必要です。

`scatool`コマンドは、`root`ユーザとして実行するか、`sudo`を使用して実行します。SCAツールを呼び出すときに、既存の`supportconfig` TARアーカイブを分析するか、新しいアーカイブの生成と分析を同時に行うことができます。このツールは対話型コンソール(タブ補完機能を使用)も備えており、`supportconfig`を外部マシンで実行したり、それ以降の分析をローカルマシンで実行したりできます。

次に、コマンドの例をいくつか示します。

`sudo scatool -s`

`supportconfig`を呼び出し、ローカルマシン上に新しいsupportconfigアーカイブを生成します。インストール済み製品に一致するSCA分析パターンを適用して、既知の問題がないかどうかアーカイブを分析します。分析結果から生成されたHTMLレポートのパスが表示されます。レポートは通常、supportconfigアーカイブのあるディレクトリと同じディレクトリに書き込まれます。

`sudo scatool -s -o /opt/sca/reports/`

`sudo scatool -s`と同じですが、HTMLレポートは`-o`で指定したパスに書き込まれる点が異なります。

`sudo scatool -a PATH_TO_TARBALL_OR_DIR`

指定したsupportconfigアーカイブファイル(またはsupportconfigアーカイブの展開先の指定ディレクトリ)を分析します。生成されたHTMLレポートは、supportconfigアーカイブまたはディレクトリと同じ場所に保存されます。

`sudo scatool -a sles_server.company.com`

外部サーバ`sles_server.company.com`とのSSH接続を確立し、そのサーバ上で`supportconfig`を実行します。その後、supportconfigアーカイブをローカルマシンにコピーし、そこで分析を行います。生成されたHTMLレポートは、デフォルトの`/var/log`ディレクトリに保存されます(`sles_server.company.com`にはsupportconfigアーカイブのみが作成されます)。

`sudo scatool -c`

`scatool`の対話型コンソールを起動します。利用可能なコマンドを参照するには、<kbd><Tab></kbd>を2回押します。

他のオプションおよび詳細については、`sudo scatool -h`を実行するか、`scatool`のマニュアルページを参照してください。

2.4.2 SCAアプライアンス

supportconfigアーカイブの分析にSCAアプライアンスを使用する場合は、専用のサーバ(または仮想マシン)をSCAアプライアンスサーバとして設定する必要があります。このSCAアプライアンスサーバを使用して、エンタープライズ内にある、SUSE Linux Enterprise ServerまたはSUSE Linux Enterprise Desktopが稼働するすべてのマシンからのsupportconfigアーカイブを分析できます。supportconfigアーカイブをアプライアンスサーバにアップロードするだけで分析を行うことができます。対話操作は必要ありません。MariaDBデータベースでは、SCAアプライアンスは、解析済みのsupportconfigアーカイブをすべて追跡します。アプライアンスのWebインタフェースからSCAレポートを直接参照できます。アプライアンスから管理者ユーザに電子メールでHTMLレポートを送信することもできます。詳細については、2.4.2.5.4項 「電子メールでのSCAレポートの送信」を参照してください。

コマンドラインから短時間でSCAアプライアンスをインストールしてセットアップするには、2.4.2.1項「インストールのクイックスタート」の手順に従います。この手順は上級者向けで、ベアインストールとセットアップコマンドに焦点を当てています。詳しい説明については、2.4.2.2項 「前提条件」～2.4.2.3項 「インストールと基本セットアップ」を参照してください。

2.4.2.1 インストールのクイックスタート

前提条件

- WebおよびLAMPパターン
- Webおよびスクリプティングモジュール(このモジュールを選択できるようにするにはマシンを登録する必要があります)

注記: root特権が必要

次のプロシージャのコマンドはすべて `root` として実行される必要があります。

手順 2.3 アップロードに匿名FTPを使用するインストール

アプライアンスをセットアップして稼働させた後は、手動での対話操作は必要ありません。したがって、cronジョブを使用してsupportconfigアーカイブを作成およびアップロードするには、この方法でアプライアンスをセットアップするのが理想的です。

1. アプライアンスをインストールするマシンでコンソールにログインし、次のコマンドを実行します。

```
zypper install sca-appliance-* sca-patterns-* vsftpd
systemctl enable apache2.service
systemctl start apache2.service
```

```
systemctl enable vsftpd.service
systemctl start vsftpd.service
yast ftp-server
```

2. YaST FTPサーバで、[Authentication (認証)] › [Enable Upload (アップロードの有効化)] › [Anonymous Can Upload (匿名ユーザのアップロード許可)] › [完了] › [はい]の順に選択し、[/srv/ftp/upload]を作成します。
3. 次のコマンドを実行します。

```
systemctl enable mysql.service
systemctl start mysql.service
mysql_secure_installation
setup-sca -f
```

このmysql_secure_installationにより、MariaDBの `root` パスワードが作成されます。

手順 2.4 アップロードにSCP/TMPを使用するインストール

この方法でアプライアンスをセットアップするには、SSHパスワードを入力する際に手動での対話操作が必要になります。

1. アプライアンスをインストールするマシンでコンソールにログインします。
2. 次のコマンドを実行します。

```
zypper install sca-appliance-* sca-patterns-*
systemctl enable apache2.service
systemctl start apache2.service
sudo systemctl enable mysql.service
systemctl start mysql.service
mysql_secure_installation
setup-sca
```

2.4.2.2 前提条件

SCAアプライアンスサーバを実行するには、次の前提条件が必要です。

- すべての `sca-appliance-*` パッケージ。
- `sca-patterns-base` パッケージ。さらに、アプライアンスで分析するsupportconfigアーカイブのタイプに合った、製品固有の `sca-patterns-*`。

- Apache
- PHP
- MariaDB
- 匿名FTPサーバ(オプション)

2.4.2.3 インストールと基本セットアップ

2.4.2.2項 「前提条件」に記載されているように、SCAアプライアンスには他のパッケージに対する依存関係がいくつかあります。そのため、SCAアプライアンスサーバをインストールしてセットアップする前に、次の手順で準備を行う必要があります。

1. ApacheおよびMariaDBに対して、`Web`および`LAMP`インストールパターンをインストールします。
2. Apache、MariaDB、および匿名FTPサーバ(オプション)をセットアップします。詳細については、第29章 Apache HTTPサーバと第30章 YaSTを使用したFTPサーバの設定を参照してください。
3. ApacheおよびMariaDBをブート時に起動するように設定します。

```
sudo systemctl enable apache2.service mysql.service
```

4. 両方のサービスを開始します。

```
sudo systemctl start apache2.service mysql.service
```

これで、手順2.5「SCAアプライアンスのインストールと設定」の説明に従ってSCAアプライアンスをインストールしてセットアップできます。

手順 2.5 SCAアプライアンスのインストールと設定

パッケージをインストールしたら、`setup-sca`スクリプトを使用して、SCAアプライアンスが使用するMariaDBの管理およびレポートデータベースの基本設定を行います。

このスクリプトを使用して、マシンからSCAアプライアンスにsupportconfigアーカイブをアップロードするための次のオプションを設定できます。

- `scp`
- 匿名FTPサーバ

1. アプライアンスとSCA基本パターンライブラリをインストールします。

```
sudo zypper install sca-appliance-* sca-patterns-base
```

2. さらに、分析するsupportconfigアーカイブのタイプに合ったパターンパッケージをインストールします。たとえば、現在の環境にSUSE Linux Enterprise Server 11のサーバとSUSE Linux Enterprise Server 12のサーバがある場合、`sca-patterns-sle11`パッケージと`sca-patterns-sle12`パッケージの両方をインストールします。
利用可能なすべてのパターンをインストールする

```
zypper install sca-patterns-*
```

3. SCAアプライアンスの基本セットアップには、`setup-sca`スクリプトを使用します。スクリプトの呼び出し方法は、supportconfigアーカイブをどのようにSCAアプライアンスサーバにアップロードするかによって異なります。

   - `/srv/ftp/upload`ディレクトリを使用する匿名FTPサーバを設定済みの場合は、`-f`オプションを指定してセットアップスクリプトを実行し、画面の指示に従います。

     ```
     setup-sca -f
     ```

     注記: 別のディレクトリを使用するFTPサーバ

     FTPサーバで`/srv/ftp/upload`以外のディレクトリを使用する場合は、まず、正しいディレクトリを指すように環境設定ファイル`/etc/sca/sdagent.conf`および`/etc/sca/sdbroker.conf`を調整します。

   - `scp`を使用してsupportconfigファイルをSCAアプライアンスサーバの`/tmp`ディレクトリにアップロードする場合は、パラメータを指定せずにセットアップスクリプトを呼び出して、画面の指示に従います。

     ```
     setup-sca
     ```

   セットアップスクリプトは要件チェックをいくつか実行し、必要なサブコンポーネントを設定します。2つのパスワードを入力するようプロンプトが表示されます。1つは、セットアップ済みのMariaDBのMySQL `root`パスワードで、もう1つは、SCAアプライアンスのWebインタフェースにログインするために使用するWebユーザのパスワードです。

4. 既存のMariaDBの`root`パスワードを入力します。これにより、SCAアプライアンスがMariaDBに接続できるようになります。

5. Webユーザのパスワードを定義します。パスワードは `/srv/www/htdocs/sca/web-config.php` に書き込まれ、ユーザ `scdiag` のパスワードとして設定されます。ユーザ名とパスワードは後で随時変更できます。2.4.2.5.1項 「Webインタフェースのパスワード」を参照してください。

インストールとセットアップが正常に完了したら、すぐにSCAアプライアンスを使用できます。2.4.2.4項「SCAアプライアンスの使用」を参照してください。ただし、Webインタフェースのパスワードの変更、SCAパターンのアップデートのソースの変更、アーカイブモードの有効化、電子メール通知の設定など、一部のオプションを変更できます。詳細については、2.4.2.5項「SCAアプライアンスのカスタマイズ」を参照してください。

警告: データの保護

SCAアプライアンスサーバ上のレポートには、分析済みのsupportconfigアーカイブが存在するマシンに関するセキュリティ関連情報が含まれているため、SCAアプライアンスサーバ上のデータを不正アクセスから保護してください。

2.4.2.4 SCAアプライアンスの使用

既存のsupportconfigアーカイブをSCAアプライアンスに手動でアップロードすることも、1つのステップで新しいsupportconfigアーカイブを作成してSCAアプライアンスにアップロードすることもできます。アップロードはFTPまたはSCP経由で行うことができます。どちらの場合も、SCAアプライアンスに接続できるURLがわかっている必要があります。FTP経由でのアップロードの場合、FTPサーバをSCAアプライアンス用に設定する必要があります。手順2.5「SCAアプライアンスのインストールと設定」を参照してください。

2.4.2.4.1 SCAアプライアンスへのSupportconfigアーカイブのアップロード

- supportconfigアーカイブを作成して(匿名) FTP経由でアップロードするには、次の手順に従います。

```
sudo supportconfig -U “ftp://sca-appliance.company.com/upload”
```

- supportconfigアーカイブを作成してSCP経由でアップロードするには、次の手順に従います。

```
sudo supportconfig -U “scp://sca-appliance.company.com/tmp”
```

SCAアプライアンスが動作しているサーバの `root` ユーザのパスワードを入力するようプロンプトが表示されます。

- 1つまたは複数のアーカイブを手動でアップロードする場合は、既存のアーカイブファイル(通常は `/var/log/nts_*.tbz` にあります)をSCAアプライアンスにコピーします。アップロード先には、アプライアンスサーバの `/tmp` ディレクトリまたは `/srv/ftp/upload` ディレクトリ(FTPがSCAアプライアンスサーバ用に設定されている場合)を使用します。

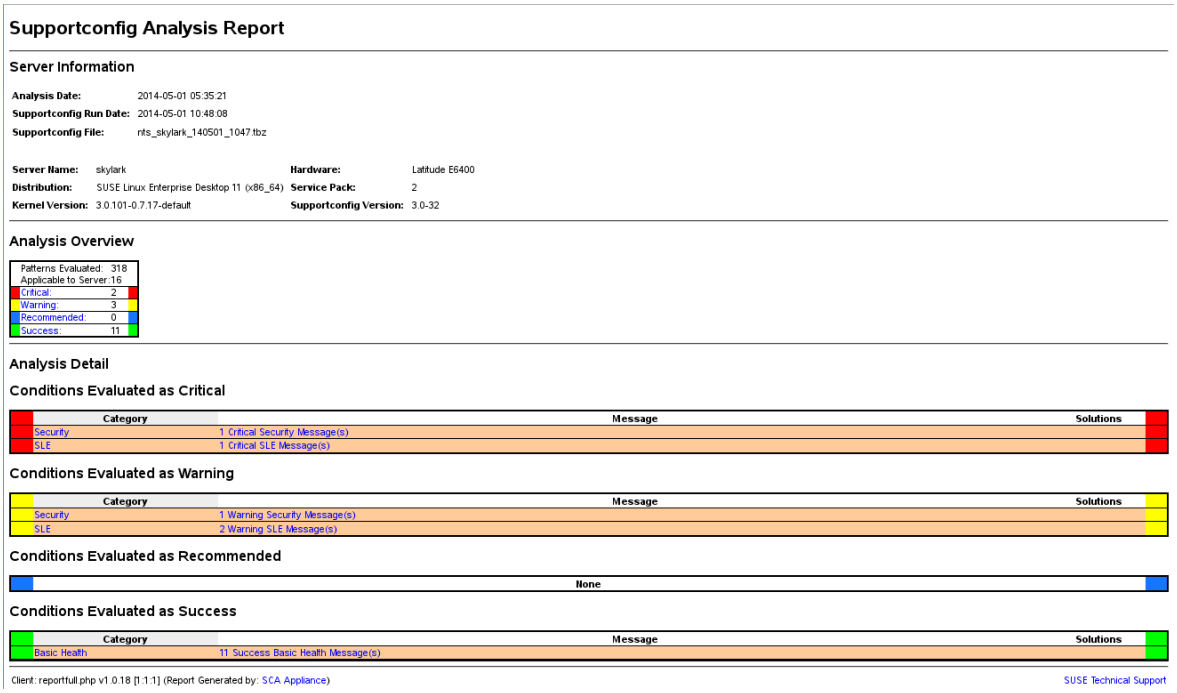
2.4.2.4.2 SCAレポートの表示

SCAレポートは、ブラウザがインストールされていて、SCAアプライアンスのレポートインデックスページにアクセス可能な任意のマシンから表示できます。

1. Webブラウザを起動し、JavaScriptとCookieが有効なことを確認します。
2. URLとして、SCAアプライアンスのレポートインデックスページを入力します。

   ```
   https://sca-appliance.company.com/sca
   ```

   不確かな場合は、システム管理者に問い合わせてください。
3. ログインするためのユーザ名とパスワードを入力するようプロンプトが表示されます。



Supportconfig Analysis Report

Server Information

Analysis Date: 2014-05-01 05:35:21
Supportconfig Run Date: 2014-05-01 10:48:08
Supportconfig File: nts_skylark_140501_1047.tbz

Server Name: skylark
Distribution: SUSE Linux Enterprise Desktop 11 (x86_64)
Kernel Version: 3.0.101-0.7.17-default
Hardware: Latitude E6400
Service Pack: 2
Supportconfig Version: 3.0-32

Analysis Overview

Patterns Evaluated:	318
Applicable to Server:	16
Critical:	2
Warning:	3
Recommended:	0
Success:	11

Analysis Detail

Conditions Evaluated as Critical

Category	Message	Solutions
Security	1 Critical Security Message(s)	
SLE	1 Critical SLE Message(s)	

Conditions Evaluated as Warning

Category	Message	Solutions
Security	1 Warning Security Message(s)	
SLE	2 Warning SLE Message(s)	

Conditions Evaluated as Recommended

None

Conditions Evaluated as Success

Category	Message	Solutions
Basic Health	11 Success Basic Health Message(s)	

Client: reportfull.php v1.0.18 [1.1.1] (Report Generated by: SCA Appliance) SUSE Technical Support

図 2.2 SCAアプライアンスによって生成されるHTMLレポート

4. ログイン後、参照するレポートの日付をクリックします。

5. 最初に[Basic Health (基本的なヘルス)]カテゴリをクリックして展開します。

6. [Message (メッセージ)]列で、個々のエントリをクリックします。SUSE Knowledgebaseの対応する記事が開きます。提案された解決方法を読み、指示に従います。

7. [Supportconfig Analysis Report (Supportconfig分析レポート)]の[Solutions (解決方法)]列に追加エントリが表示されている場合は、それらをクリックします。提案された解決方法を読み、指示に従います。

8. SCAによって特定された問題に直接関係する結果については、SUSE Knowledgebase (http://www.suse.com/support/kb/)を確認してください。問題解決に取り組みます。

9. 問題の再発防止のために事前に対処できる結果がないかどうかを確認します。

2.4.2.5 SCAアプライアンスのカスタマイズ

次の項では、Webインタフェースのパスワードを変更する方法、SCAパターンアップデートのソースを変更する方法、アーカイブモードを有効にする方法、および電子メール通知を設定する方法について説明します。

2.4.2.5.1 Webインタフェースのパスワード

SCAアプライアンスのWebインタフェースにログインするには、ユーザ名とパスワードが必要です。デフォルトのユーザ名は `scdiag` で、デフォルトのパスワードは `linux` です(特に指定されていない場合。手順2.5「SCAアプライアンスのインストールと設定」を参照してください)。パスワードを保護するため、デフォルトのパスワードはできる限り速やかに変更してください。ユーザ名を変更することもできます。

手順 2.6 WEBインタフェースのユーザ名またはパスワードの変更

1. SCAアプライアンスサーバのシステムコンソールで `root` ユーザとしてログインします。
2. エディタで `/srv/www/htdocs/sca/web-config.php` を開きます。
3. 必要に応じて、`$username` および `$password` の値を変更します。
4. ファイルを保存して終了します。

2.4.2.5.2 SCAパターンのアップデート

デフォルトでは、すべての `sca-patterns-*` パッケージは `root` cronジョブによって定期的にアップデートされます。このジョブは夜間に `sdagent-patterns` スクリプトを実行し、このスクリプトが `zypper update sca-patterns-*` を実行します。定期的なシステムアップデートにより、SCAアプライアンスおよびパターンのすべてのパッケージがアップデートされます。SCAアプライアンスとパターンを手動でアップデートするには、以下を実行します。

```
sudo zypper update sca-*
```

デフォルトでは、アップデートはSUSE Linux Enterprise 12のアップデートリポジトリからインストールされます。必要に応じて、アップデートのソースをSMTサーバに変更できます。`sdagent-patterns` は、`zypper update sca-patterns-*` を実行する際に、現在設定されているアップデートチャネルからアップデートを取得します。このチャネルがSMTサーバにある場合、パッケージはそこから取得されます。

手順 2.7 SCAパターンの自動アップデートの無効化

1. SCAアプライアンスサーバのシステムコンソールで `root` ユーザとしてログインします。
2. エディタで `/etc/sca/sdagent-patterns.conf` を開きます。
3. 次のエントリを変更します。

```
UPDATE_FROM_PATTERN_REPO=1
```

変更後:

```
UPDATE_FROM_PATTERN_REPO=0
```

4. ファイルを保存して終了します。変更を適用するためにマシンを再起動する必要はありません。

2.4.2.5.3 アーカイブモード

supportconfigアーカイブの分析が終了し、その結果がMariaDBデータベースに保存されると、アーカイブはすべてSCAアプライアンスから削除されます。ただし、トラブルシューティングのために、マシンからのsupportconfigアーカイブのコピーを保持しておくと便利です。デフォルトでは、アーカイブモードは無効になっています。

手順 2.8 SCAアプライアンスのアーカイブモードの有効化

1. SCAアプライアンスサーバのシステムコンソールで `root` ユーザとしてログインします。
2. エディタで `/etc/sca/sdagent.conf` を開きます。
3. 次のエントリを変更します。

```
ARCHIVE_MODE=0
```

変更後:

```
ARCHIVE_MODE=1
```

4. ファイルを保存して終了します。変更を適用するためにマシンを再起動する必要はありません。

アーカイブモードを有効にすると、SCAアプライアンスはsupportconfigファイルを削除せずに、`/var/log/archives/saved` ディレクトリに保存します。

2.4.2.5.4 電子メールでのSCAレポートの送信

分析された各supportconfigのレポートHTMLファイルを、SCAアプライアンスから電子メールで送信できます。デフォルトでは、この機能は無効になっています。これを有効にすると、レポートの送信先電子メールアドレスのリストを定義したり、レポートの送信をトリガするステータスメッセージのレベル (`STATUS_NOTIFY_LEVEL`)を定義したりできます。

STATUS_NOTIFY_LEVELに指定可能な値

$STATUS_OFF
: HTMLレポートの送信を無効にします。

$STATUS_CRITICAL
: CRITICALが含まれるSCAレポートのみを送信します。

$STATUS_WARNING
: WARNINGまたはCRITICALが含まれるSCAレポートのみを送信します。

$STATUS_RECOMMEND
: RECOMMEND、WARNING、またはCRITICALが含まれるSCAレポートのみを送信します。

$STATUS_SUCCESS
: SUCCESS、RECOMMEND、WARNING、またはCRITICALが含まれるSCAレポートを送信します。

手順 2.9 SCAレポートの電子メール通知の設定

1. SCAアプライアンスサーバのシステムコンソールで `root` ユーザとしてログインします。
2. エディタで `/etc/sca/sdagent.conf` を開きます。
3. `STATUS_NOTIFY_LEVEL` というエントリを探します。デフォルトでは、これは `$STATUS_OFF` に設定されています(電子メール通知は無効です)。
4. 電子メール通知を有効にするには、`$STATUS_OFF` を、電子メールレポートを要求するステータスメッセージのレベルに変更します。次に例を示します。

   ```
   STATUS_NOTIFY_LEVEL=$STATUS_SUCCESS
   ```

   詳細については、STATUS_NOTIFY_LEVELに指定可能な値を参照してください。
5. レポートの送信先の受信者リストを定義する

   a. `EMAIL_REPORT='root'` というエントリを探します。

   b. `root` を、SCAレポートの送信先電子メールアドレスのリストに置き換えます。複数の電子メールアドレスはそれぞれスペースで区切る必要があります。次に例を示します。

      ```
      EMAIL_REPORT='tux@my.company.com wilber@your.company.com'
      ```

6. ファイルを保存して終了します。変更を適用するためにマシンを再起動する必要はありません。今後、すべてのSCAレポートは指定したアドレスに電子メールで送信されます。

2.4.2.6 データベースのバックアップと復元

SCAレポートが保存されているMariaDBデータベースをバックアップおよび復元するには、次の説明に従って`scadb`コマンドを使用します。

手順 2.10 データベースのバックアップ

1. SCAアプライアンスが動作しているサーバのシステムコンソールで、`root`ユーザとしてログインします。
2. 次のコマンドを実行してアプライアンスを保守モードにします。

```
scadb maint
```

3. 次のコマンドを実行してバックアップを開始します。

```
scadb backup
```

   データはTARアーカイブ`sca-backup-*sql.gz`に保存されます。

4. パターン作成データベースを使用して独自のパターンを開発している場合は(2.4.3項 「カスタム分析パターンの開発」を参照)、そのデータもバックアップします。

```
sdpdb backup
```

   データはTARアーカイブ`sdp-backup-*sql.gz`に保存されます。

5. 次のデータを別のマシンまたは外部ストレージメディアにコピーします。
   - `sca-backup-*sql.gz`
   - `sdp-backup-*sql.gz`
   - `/usr/lib/sca/patterns/local` (カスタムパターンを作成している場合にのみ必要)
6. 次のコマンドを実行してSCAアプライアンスを再び有効にします。

```
scadb reset agents
```

手順 2.11 データベースの復元

バックアップからデータベースを復元するには、次の手順に従います。

1. SCAアプライアンスが動作しているサーバのシステムコンソールで、`root`ユーザとしてログインします。
2. 最も新しい`sca-backup-*sql.gz`および`sdp-backup-*sql.gz` TARアーカイブをSCAアプライアンスサーバにコピーします。
3. ファイルを圧縮解除するため、次のコマンドを実行します。

```
gzip -d *-backup-*sql.gz
```

4. データをデータベースにインポートするため、次のコマンドを実行します。

```
scadb import sca-backup-*sql
```

5. パターン作成データベースを使用して独自のパターンを開発している場合、次のデータもインポートします。

```
sdpdb import sdp-backup-*sql
```

6. カスタムパターンを使用している場合は、`/usr/lib/sca/patterns/local`もバックアップデータから復元します。
7. 次のコマンドを実行してSCAアプライアンスを再び有効にします。

```
scadb reset agents
```

8. データベース内のパターンモジュールを更新します。

```
sdagent-patterns -u
```

2.4.3 カスタム分析パターンの開発

SCAアプライアンスには、独自のカスタムパターンの開発を可能にする、充実したパターン開発環境(SCA Pattern Database)が付属しています。パターンは、どのプログラム言語でも作成できます。パターンをsupportconfig分析プロセスで利用できるようにするには、`/usr/lib/sca/patterns/local`に保存し、実行可能にする必要があります。SCAアプライアンスとSCAツールのどちらも、分析

レポートの一部として、新しいsupportconfigアーカイブに照らしてカスタムパターンを実行します。独自のパターンを作成(およびテスト)する方法の詳細については、http://www.suse.com/communities/conversations/sca-pattern-development/を参照してください。

2.5 カーネルモジュールのサポート

あらゆるエンタープライズ向けオペレーティングシステムにとって重要な要件は、利用環境に対して受けられるサポートのレベルです。カーネルモジュールは、ハードウェア(「コントローラ」)とオペレーティングシステムを結ぶものの中で最も重要です。SUSE Linux Enterpriseのカーネルモジュールにはすべて`supported`フラグが付いており、これは次の3つの値を取ります。

- 「yes」、したがって、`supported`
- 「external」、したがって`supported`
- 「」(空、未設定)、したがって`unsupported`

次のルールが適用されます。

- 自己再コンパイルしたカーネルのすべてのモジュールには、デフォルトで「unsupported」のマークが付きます。
- SUSEパートナーによってサポートされていて、`SUSE SolidDriverプログラム`を使用して配信されているカーネルモジュールには、「external」のマークが付きます。
- `supported`フラグが設定されていない場合、そのモジュールをロードすると、カーネルが汚染されます。汚染カーネルはサポートされません。サポート対象外のカーネルモジュールは追加のRPMパッケージ(`kernel-FLAVOR-extra`)に含まれており、デフォルトではロードされません(`FLAVOR`=`default`|`xen`|...)。さらに、これらのサポート対象外のモジュールはインストーラで利用できず、`kernel-FLAVOR-extra`パッケージはSUSE Linux Enterpriseのメディアに含まれていません。
- Linuxカーネルのライセンスと互換性があるライセンスに従って提供されていないカーネルモジュールを使用しても、カーネルが汚染されます。詳細については、`/usr/src/linux/Documentation/sysctl/kernel.txt`、および`/proc/sys/kernel/tainted`の状態を参照してください。

2.5.1 技術的背景

- Linuxカーネル: SUSE Linux Enterprise 12では、`/proc/sys/kernel/unsupported`の値はデフォルトで`2`に設定されています(`do not warn in syslog when loading unsupported modules （サポート対象外のカーネルのロード時にsyslogで警告しない）`)。このデフォルト値は、インストーラでも、インストールしたシステムでも使用されます。詳細については、`/usr/src/linux/Documentation/sysctl/kernel.txt`を参照してください。

- `modprobe`: モジュールの依存関係を確認して適切にモジュールをロードするための`modprobe`ユーティリティは、`supported`フラグの値を確認します。この値が「yes」または「external」であればモジュールはロードされ、他の値の場合はロードされません。この動作を無効にする方法については、2.5.2項 「サポート対象外のモジュールの使用」を参照してください。

注記

SUSEは一般的に、`modprobe -r`によるストレージモジュールの削除をサポートしていません。

2.5.2 サポート対象外のモジュールの使用

一般的なサポート可能性は重要ですが、サポート対象外のモジュールをロードしなければならないこともあります(たとえば、テストやデバッグを行う場合や、ハードウェアベンダーがホットフィックスを提供している場合など)。

- デフォルト値を無効にするには、`/etc/modprobe.d/10-unsupported-modules.conf`を編集して、変数 `allow_unsupported_modules` の値を`1`に変更します。initrdでサポート対象外のモジュールが必要な場合は、必ず`dracut -f`を実行してinitrdをアップデートしてください。モジュールを一度だけロードする場合は、`modprobe`で`--allow-unsupported-modules`オプションを使用できます。詳細については、`modprobe`のマニュアルページを参照してください。

- インストール時に、ドライバアップデートディスクを使用してサポート対象外のモジュールを追加できます。この場合、これらのモジュールはロードされます。ブート時およびそれ以降にサポート対象外のモジュールを強制的にロードするには、カーネルコマンドラインオプション`oem-modules`を使用します。`suse-module-tools`パッケージのインストールおよび初期化時に、カーネルフラグ`TAINT_NO_SUPPORT`（`/proc/sys/kernel/tainted`)が評価されます。カーネルがすでに汚染されている場合は、`allow_unsupported_modules`が有効になります。これにより、インストール中のシステムでサポート対象外のモジュールが失敗しないようにします。イン

ストール時にサポート対象外のモジュールが存在しておらず、もう1つの特殊なカーネルコマンドラインオプション(`oem-modules=1`)を使用していない場合は、引き続き、サポート対象外のモジュールを許可しない動作がデフォルトです。

サポート対象外のモジュールをロードおよび実行すると、カーネルとシステム全体がSUSEのサポート対象外になる点に注意してください。

2.6 その他の情報

- `man supportconfig`—`supportconfig`のマニュアルページ
- `man supportconfig.conf`—supportconfig環境設定ファイルのマニュアルページ
- `man scatool`—`scatool`のマニュアルページ
- `man scadb`—`scadb`のマニュアルページ
- `man setup-sca`—`setup-sca`のマニュアルページ
- https://mariadb.com/kb/en/—MariaDBのマニュアル
- http://httpd.apache.org/docs/および第29章 Apache HTTPサーバ—Apache Webサーバのマニュアル
- 第30章 YaSTを使用したFTPサーバの設定—FTPサーバのセットアップ方法のマニュアル
- http://www.suse.com/communities/conversations/sca-pattern-development/—独自のSCAパターンを作成(およびテスト)する方法
- http://www.suse.com/communities/conversations/basic-server-health-check-supportconfig/—「A Basic Server Health Check with Supportconfig」
- https://www.novell.com/communities/coolsolutions/cool_tools/create-your-own-supportconfig-plugin/—「Create Your Own Supportconfig Plugin」
- http://www.suse.com/communities/conversations/creating-a-central-supportconfig-repository/—「Creating a Central Supportconfig Repository」

3 テキストモードのYaST

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

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

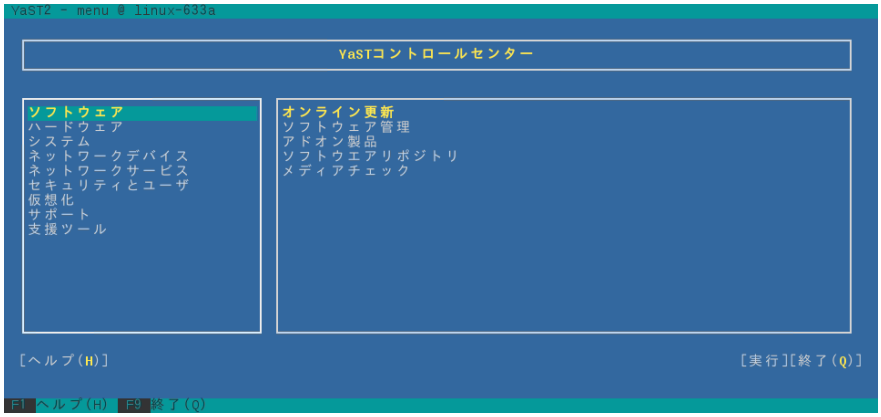


図 3.1 テキストモードのYASTのメインウィンドウ

YaSTをテキストモードで起動すると、YaSTコントロールセンターが表示されます(図　3.1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。左側のフレームのカテゴリには、さまざまなモジュールがあります。このフレームはYaSTが起動したときにアクティブになり、白い太線でマークされます。アクティブなカテゴリが選択されています。右側のフレームには、アクティブなカテゴリで使用できるモジュールの概要が表示されます。下方のフレームには、[ヘルプ]および[終了]用ボタンがあります。

YaSTコントロールセンターを起動すると、カテゴリ[Software　(ソフトウェア)]が自動的に選択されます。カテゴリを変更するには、<↓>と<↑>を使用します。カテゴリからモジュールを選択するには、<→>で右側のフレームをアクティブにして、<↓>と<↑>を使用してモジュールを選択します。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。選択したモジュールがハイライトされます。<Enter>を押してアクティブなモジュールを起動します。

モジュールのさまざまなボタンまたは選択フィールドで、文字がハイライト表示されています(デフォルトは黄色)。そのまま<Tab>キーでナビゲートするする代わりに、直接ボタンを選択するには、<Alt>-<表示された文字の>を使用します。<Alt>-<Q>を押すか、または[終了]を選択して<Enter>を押して、YaSTコントロールセンターを終了します。

ヒント: YaSTダイアログの更新

ウィンドウのサイズを変更した場合など、YaSTのダイアログの表示が乱れたり変形したりした場合は、< Ctrl >-< L >を押すとコンテンツを更新し復元できます。

3.1 モジュールでのナビゲーション

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

ボタンおよび選択リスト間のナビゲーター

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

選択リストでのナビゲーション

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をナビゲートするには、矢印キー(< ↑ >と< ↓ >)を使用します。フレーム内の個別エントリがその幅を超える場合は、< Shift >-< → >または< Shift >-< ← >を使用して、右または左にスクロールします。代わりに< Ctrl >-< E >または< Ctrl >-< A >を使用することもできます。この組み合わせは、コントロールセンターの場合のように、< → >または< ← >を使用すると、アクティブフレームまたは現在の選択リストが変更されてしまう場合に使用できます。

ボタン、ラジオボタン、およびチェックボックス

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

ファンクションキー

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

ncursesモードのナビゲーションツリーの使用

一部のYaSTモジュールでは、ウィンドウの左部分にあるナビゲーションツリーを使用して、設定ダイアログを選択します。矢印キー(< ↑ >と< ↓ >)を使用して、ツリー内を移動します。 Space を使用して、ツリー項目を開閉します。ncursesモードでは、ナビゲーションツリーでの選択後、選択したダイアログを表示するには< Enter >を押す必要があります。これは意図的な動作であり、これによって、ナビゲーションツリーのブラウズ時に時間のかかる再表示を節約できます。

ソフトウェアインストールモジュールでのソフトウェアの選択

表示されるパッケージの量を制限するには、左側のフィルタを使用します。インストール済みパッケージには、文字 i のマークが付いています。パッケージのステータスを変更するには、< Space >キーまたは< Enter >キーを押します。または、[Actions (アクション)]メニューを使用して、必要なステータスの変更(インストール、削除、更新、タブー、またはロック)を選択します。

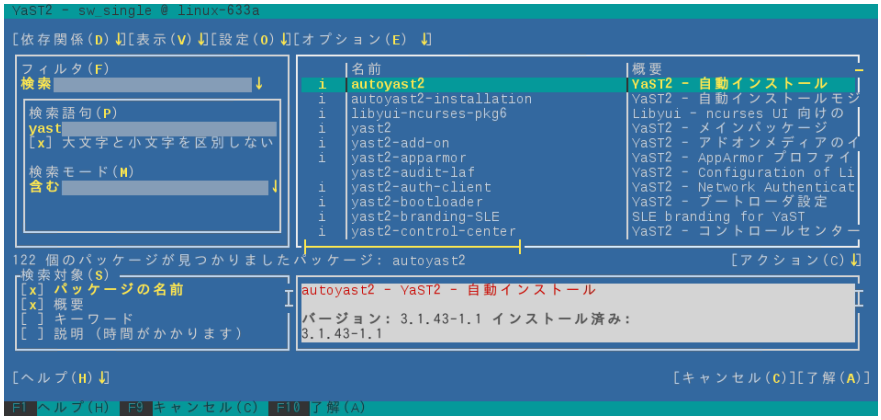


図 3.2 ソフトウェアインストールモジュール

3.2 キーの組み合わせの制約

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

< Alt >キーの< Esc >キーでの置き換え

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

< Ctrl >–< F >と< Ctrl >–< B >による前後のナビゲーション

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

ファンクションキーの制約

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

3.3 YaSTコマンドラインオプション

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

```
yast -h
```

3.3.1 個別モジュールの起動

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

```
yast <module_name>
```

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

3.3.2 コマンドラインからのパッケージのインストール

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

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

または

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

`package_name` には、`gvim` などの1つの短いパッケージ名を指定するか(この場合、依存関係を確認してインストールされます)、RPMパッケージのフルパスを指定できます(この場合、依存関係を確認せずにインストールされます)。

YaSTから提供される機能を超える機能を持つコマンドラインベースのソフトウェア管理ユーティリティを必要とする場合は、Zypperの使用をご検討ください。このユーティリティは、YaSTパッケージマネージャの基礎でもある同じソフトウェア管理ライブラリを使用します。Zypperの基本的使用法については、6.1項「Zypperの使用」で説明されています。

3.3.3 YaSTモジュールのコマンドラインパラメータ

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

```
yast <module_name> help
```

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

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

4 Snapperを使用したシステムの回復とスナップショット管理

Linuxでファイルシステムのスナップショットを作成し、ロールバックできるようにすることは、過去に要望の多かった機能です。Snapperを、Btrfsファイルシステムまたはシンプロビジョンのボリュームと併用することによって対応できます。

Btrfsは、Linux用の新しい書き込み時コピー方式のファイルシステムで、サブボリューム(各物理パーティション内の1つまたは複数の個別にマウント可能なファイルシステム)のファイルシステムスナップショット(特定時点におけるサブボリュームの状態のコピー)をサポートします。スナップショットは、XFS、Ext4、またはExt3でフォーマットされたシンプロビジョンボリュームでもサポートされています。Snapperを使用してこれらのスナップショットを作成および管理できます。Snapperには、コマンドラインおよびYaSTインタフェースがあります。SUSE Linux Enterprise Serverバージョン12から、Btrfsスナップショットからブートすることもできるようになりました。詳細については、4.3項「スナップショットからのブートによるシステムロールバック」を参照してください。

Snapperを使用して、次のタスクを実行できます。

- zypperやYaSTで行ったシステムの変更を元に戻す。詳細については、4.2項 「Snapperを使用した変更の取り消し」を参照してください。
- 古いスナップショットからファイルを復元する。詳細については、4.2.2項 「Snapperを使用したファイルの復元」を参照してください。
- スナップショットからブートすることによってシステムをロールバックする。詳細については、4.3項「スナップショットからのブートによるシステムロールバック」を参照してください。
- オンザフライでスナップショットを手動作成し、既存のスナップショットを管理する。詳細については、4.5項「スナップショットの手動での作成と管理」を参照してください。

4.1 デフォルト設定

SUSE Linux Enterprise Server上のSnapperは、システム変更の「取り消しおよび回復ツール」「」として機能するように設定されています。デフォルトでは、SUSE Linux Enterprise Serverのルートパーティション(/)はBtrfsでフォーマットされています。ルートパーティション(/)に十分な容量(約8GB以上)がある場合、スナップショットの作成が自動的に有効になります。デフォルトでは、/以外のパーティション上でスナップショットを作成することはできません。

スナップショットを作成すると、スナップショットとスナップショット元のファイルは、 いずれもファイルシステム内の同じブロックを指します。そのため、最初は、スナップショットが余分にディスク容量を占めることはありません。元のファイルシステムのデータが変更されると、変更されたデータブロックがコピー

され、古いデータブロックはスナップショットのように保持されます。このため、スナップショットは、変更されたデータと同じ容量を占めます。こうして、時間が経過するにつれて、スナップショットの領域は大きくなっていきます。その結果、スナップショットを含む `Btrfs` ファイルシステムからファイルを削除しても、ディスクの空き容量が増えないことがあります。

注記: スナップショットの場所

スナップショットは常に、スナップショット作成元と同じパーティションまたはサブボリュームに保存されます。別のパーティションまたはサブボリュームにスナップショットを保存することはできません。

その結果、スナップショットを含むパーティションは、「通常の」パーティションよりも大きくする必要があります。具体的な容量は、保持するスナップショット数やデータの変更頻度によって異なります。一般的には、通常のファイルシステムの2倍程度を検討してください。

スナップショット自体は技術的な意味では同じですが、作成された状況に基づいて、次の3種類のスナップショットを区別しています。

スナップショットの種類

タイムラインスナップショット

1時間ごとに1つのスナップショットが作成されます。古いスナップショットは自動的に削除されます。デフォルトで、最近10日間、10カ月間、10年間の最初のスナップショットが保持されます。タイムラインスナップショットは、ルートパーティションを除きデフォルトで有効になっています。

インストールスナップショット

YaSTまたはZypperで1つ以上のパッケージをインストールする場合、常にスナップショットのペアが作成されます。インストール開始前のスナップショット(「」「事前」)と、インストール完了後のスナップショット(「」「事後」)です。カーネルなどの重要なコンポーネントがインストールされた場合、スナップショットのペアは重要とマークされます(`important=yes`)。古いスナップショットは自動的に削除されます。デフォルトでは、最新の10個の重要なスナップショット、および最新の10個の「」「通常」のスナップショット(管理スナップショットを含む)が保持されます。インストールスナップショットはデフォルトで有効になっています。

管理スナップショット

システムをYaSTで管理する場合、常にスナップショットのペアが作成されます。YaSTモジュール開始時のスナップショット(「」「事前」)と、モジュール終了時のスナップショット(「」「事後」)です。古いスナップショットは自動的に削除されます。デフォルトでは、最新の10個の重要なスナップショットと最新の10個の「」「通常」のスナップショット(インストールスナップショットを含む)が保持されます。管理スナップショットはデフォルトで有効になっています。

一部のディレクトリは、さまざまな理由により、スナップショットから除外する必要があります。次のリストは、除外されるすべてのディレクトリを示しています。

スナップショットから除外されるディレクトリ

`/boot/grub2/x86_64-efi`、`/boot/grub2/power-ieee1275`、`/boot/grub2/s390x`
: ブートローダ設定のロールバックはサポートされていません。

`/home`
: `/home` が独立したパーティションに存在していない場合、ロールバック時にデータが失われのを避けるために除外されます。

`/opt`、`/var/opt`
: サードパーティ製品およびアドオンは通常、`/opt` にインストールされます。ロールバック時にこれらのアプリケーションがアンインストールされるのを避けるために除外されます。

`/srv`
: WebおよびFTPサーバ用のデータが含まれています。ロールバック時にデータが失われるのを避けるために除外されます。

`/tmp`、`/var/tmp`、`/var/crash`
: スナップショットから除外される一時ファイルを含むすべてのディレクトリ。

`/var/lib/named`
: DNSサーバ用のゾーンデータが含まれます。ネームサーバがロールバック後に確実に動作できるように、スナップショットから除外されます。

`/var/lib/mailman`、`/var/spool`
: 電子メールまたは電子メールキューを含むディレクトリは、ロールバック後に電子メールが失われるのを避けるために除外されます。

`/var/lib/pgqsl`
: PostgreSQLデータが含まれます。

`/var/log`
: ログファイルの場所。壊れたシステムのロールバック後にログファイルを分析できるようにスナップショットから除外されます。

4.1.1 設定のカスタマイズ

SUSE Linux Enterprise Serverには、適切なデフォルト設定が付属していて、ほとんどの使用事例ではこのままで十分です。ただし、スナップショットの自動作成およびスナップショットの維持管理のあらゆる側面をニーズに合わせて設定できます。

4.1.1.1 スナップショットの無効化/有効化

3つのスナップショットの種類(タイムライン、インストール、および管理)はそれぞれ独立して有効化または無効化することができます。

タイムラインスナップショットの有効化/無効化

有効化. `snapper -c root set-config "TIMELINE_CREATE=yes"`

無効化. `snapper -c root set-config "TIMELINE_CREATE=no"`
タイムラインスナップショットは、ルートパーティションを除きデフォルトで有効になっています。

インストールスナップショットの有効化/無効化

有効化. `snapper-zypp-plugin` パッケージをインストールします。

無効化. `snapper-zypp-plugin` パッケージをアンインストールします。
インストールスナップショットはデフォルトで有効になっています。

管理スナップショットの有効化/無効化

有効化. `/etc/sysconfig/yast2` で `USE_SNAPPER` を `yes (はい)` に設定します。

無効化. `/etc/sysconfig/yast2` で `USE_SNAPPER` を `no (いいえ)` に設定します。
管理スナップショットはデフォルトで有効になっています。

4.1.1.2 インストールスナップショットの制御

YaSTまたはZypperでパッケージをインストールする際にスナップショットペアを作成する処理は、`snapper-zypp-plugin` が扱います。XML環境設定ファイル `/etc/snapper/zypp-plugin.conf` で、スナップショットを作成するタイミングを定義します。デフォルトでは、ファイルは次のようになっています。

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <snapper-zypp-plugin-conf>
```

```
 3  <solvables>
 4   <solvable match="w" ❶ important="true" ❷>kernel-* ❸</solvable>
 5   <solvable match="w" important="true">dracut</solvable>
 6   <solvable match="w" important="true">glibc</solvable>
 7   <solvable match="w" important="true">systemd*</solvable>
 8   <solvable match="w" important="true">udev</solvable>
 9   <solvable match="w">*</solvable> ❹
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

❶ match属性は、パターンがUnixシェルと同様のワイルドカードであるか(`w`)、それともPython正規表現であるか(`re`)を定義します。

❷ 指定されたパターンが一致し、対応するパッケージに重要のマークが付いている場合(カーネルパッケージなど)、そのスナップショットにも重要のマークが付きます。

❸ パッケージ名に一致するパターン。特殊文字は、`match`属性の設定に基づいて、シェル風のワイルドカードまたは正規表現のいずれかとして解釈されます。このパターンは、`kernel-`で始まるすべてのパッケージ名に一致します。

❹ この行は、無条件にすべてのパッケージに一致します。

この設定スナップショットでは、パッケージのインストール時に常にペアが作成されます(9行目)。重要のマークが付いたKernel、dracut、glibc、systemd、またはudevパッケージがインストールされると、そのスナップショットペアにも重要のマークが付きます(4～8行目)。すべてのルールが評価されます。

ルールを無効にするには、削除するか、XMLコメントを使用して無効にします。パッケージがインストールされるたびにスナップショットペアが作成されないようにするには、次のようにします。たとえば、9行目のコメント行のように指定します。

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <snapper-zypp-plugin-conf>
 3  <solvables>
 4   <solvable match="w" important="true">kernel-*</solvable>
 5   <solvable match="w" important="true">dracut</solvable>
 6   <solvable match="w" important="true">glibc</solvable>
 7   <solvable match="w" important="true">systemd*</solvable>
 8   <solvable match="w" important="true">udev</solvable>
 9   <!-- <solvable match="w">*</solvable> -->
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

4.1.1.3 スナップショットのアーカイブの制御

スナップショットはディスク容量を占有します。ディスク容量不足によるシステム停止が発生しないようにするために、古いスナップショットは自動的に削除されます。デフォルトでは、次のスナップショットが保持されます。

- 最近10日間、10カ月間、および10年間の最初のスナップショット
- 重要のマークが付いている最近の10個のインストールスナップショットペア
- 最近の10個のインストール/管理スナップショット

これらの値の変更方法については、4.4.1項「既存の設定の管理」を参照してください。

4.1.1.4 シンプロビジョンLVMボリュームでのSnapperの使用

Snapperは、`Btrfs`ファイルシステムのスナップショット作成だけでなく、XFS、Ext4、またはExt3でフォーマットされたシンプロビジョンLVMボリュームのスナップショット作成にも対応しています(通常のLVMボリュームのスナップショットには「対応していません」)。LVMボリュームに関する詳細および設定の手順については、ブック「導入ガイド」15「高度なディスクセットアップ」15.2「LVMの設定」を参照してください。

シンプロビジョンLVMボリュームでSnapperを使用するには、そのようにSnapper設定を作成する必要があります。LVMで、`--fstype=lvm(FILESYSTEM)`を使用してファイルシステムを指定する必要があります。`ext3`、`etx4`、または`xfs`は、`FILESYSTEM`の有効な値です。例:

```
snapper -c lvm create-config --fstype="lvm(xfs)" /thin_lvm
```

4.4.1項「既存の設定の管理」で説明したように、必要に応じてこの設定を調整できます。

4.2 Snapperを使用した変更の取り消し

SUSE Linux Enterprise ServerのSnapperは、`zypper`やYaSTで行った変更を取り消すことができるツールとしてあらかじめ設定されています。このために、Snapperは、`zypper`およびYaSTの実行前後にスナップショットのペアを作成します。また、Snapperを使用して、誤って削除または変更したシステムファイルを復元することもできます。このためには、ルートパーティションのタイムラインスナップショットを有効にする必要があります。詳細については、4.1.1.1項 「スナップショットの無効化/有効化」を参照してください。

上記の自動スナップショットは、デフォルトでルートパーティションとそのサブボリュームに対して設定されます。カスタム設定を作成すれば、`/home` など、他のパーティションに対してスナップショット機能を利用できます。

重要: 変更の取り消しとロールバック

スナップショットを操作してデータを復元する場合、Snapperが処理可能なシナリオとして、根本的に異なる次の2つのシナリオがあることを理解することが重要です。

変更の取り消し

次に説明されているように、変更を取り消す際には、2つのスナップショットが比較され、これらの2つのスナップショット間の変更が取り消されます。この方法を使用して、復元する必要があるファイルを明示的に選択できます。

ロールバック

4.3項 「スナップショットからのブートによるシステムロールバック」で説明されているように、ロールバックを実行すると、システムはスナップショットが作成された状態にリセットされます。

変更を取り消す場合は、現在のシステムとスナップショットを比較することもできます。このような比較から「すべての」ファイルを復元すると、ロールバックを実行した場合と同じ結果になります。ただし、ロールバックについては、4.3項 「スナップショットからのブートによるシステムロールバック」で説明されている方法を使用することをお勧めします。この方法はより高速で、ロールバック実行前にシステムを確認できるためです。

警告: データの整合性

スナップショットを作成する際に、データの整合性を確保するメカニズムがありません。スナップショットを作成するのと同時にファイルが書き込まれると(データベースなど)、ファイルが破損したり、ファイルへの書き込みが部分的になったりします。このようなファイルを復元すると、問題が発生することがあります。また、`/etc/mtab` などの一部のシステムファイルは復元しないでください。このため、「必ず」、変更されたファイルとその差分をよく確認してください。どうしても元に戻すことが必要なファイルのみ復元してください。

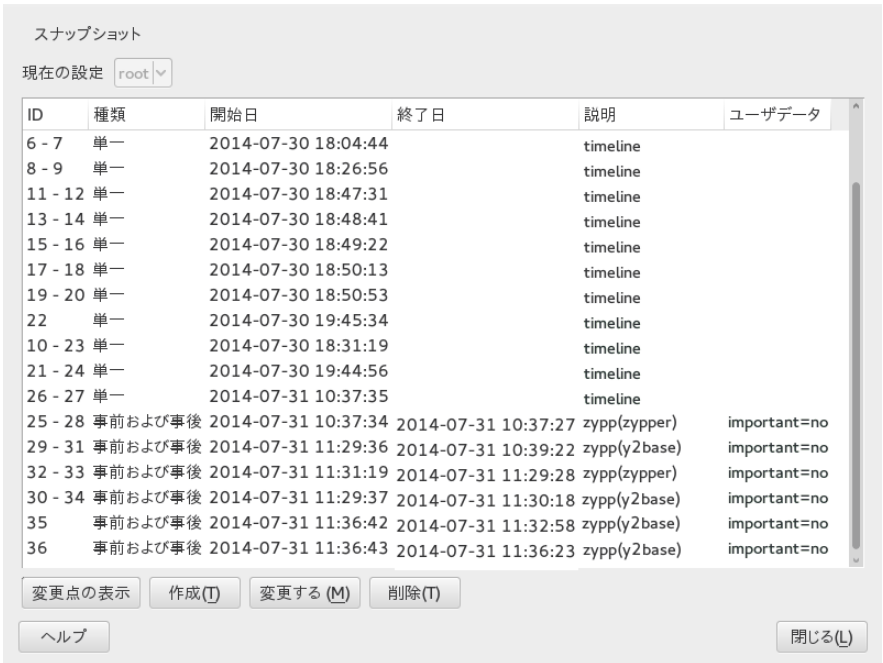
4.2.1 YaSTおよびZypperによる変更の取り消し

インストール時にルートパーティションを`Btrfs`で設定すると、Snapper (YaSTまたはZypperによる変更のロールバックがあらかじめ設定されている)が自動的にインストールされます。YaSTモジュールまたはZypperトランザクションを開始するたびに、2つのスナップショットが作成されます。モジュール開始前のファイルシステムの状態をキャプチャした「事前スナップショット」と、モジュール完了後の状態をキャプチャした「事後スナップショット」です。「」「」

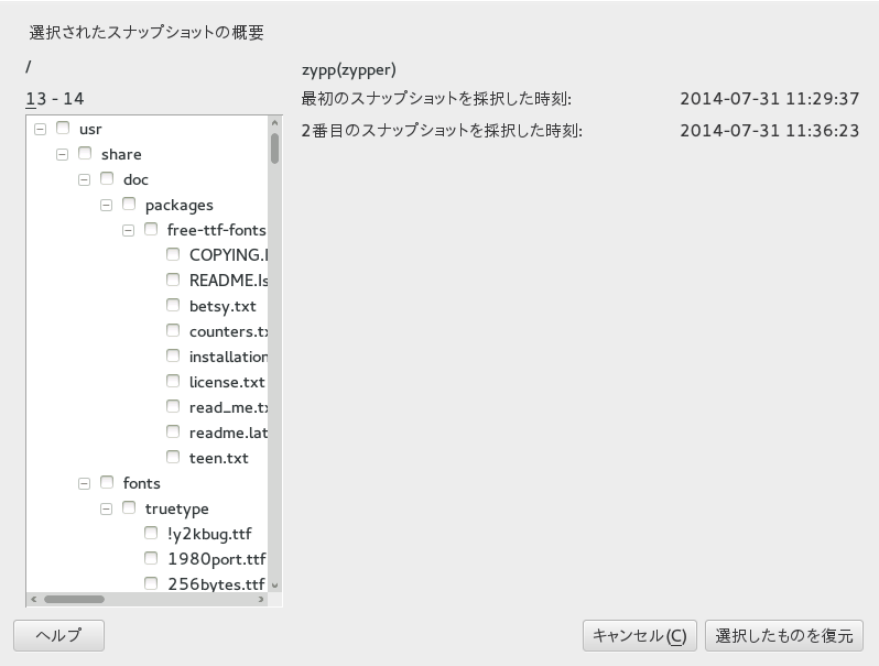
YaSTのSnapperモジュールまたは`snapper`コマンドラインツールを使用して、「事前スナップショット」「」からファイルを復元し、YaST/Zypperによる変更を元に戻すことができます。また、2つのスナップショットを比較して、どのファイルが変更されているか調べることができます。2つのバージョンのファイルの違いを表示することもできます(diff)。

手順 4.1 YASTの[SNAPPER]モジュールによる変更の取り消し

1. YaSTの[その他]セクションにある[Snapper]モジュールを起動するか、「`yast2 snapper`」と入力します。
2. [現在の設定]が[root]になっていることを確認します。 独自のSnapper設定を手動で追加していない限り、常にそのようになっています。
3. リストから事前スナップショットと事後スナップショットのペアを選択します。YaSTのスナップショットペアもZypperのスナップショットペアも、種類は[事前および事後]です。YaSTのスナップショットの場合は[説明]に「`zypp(y2base)`」と表示され、Zypperのスナップショットの場合は「`zypp(zypper)`」と表示されます。

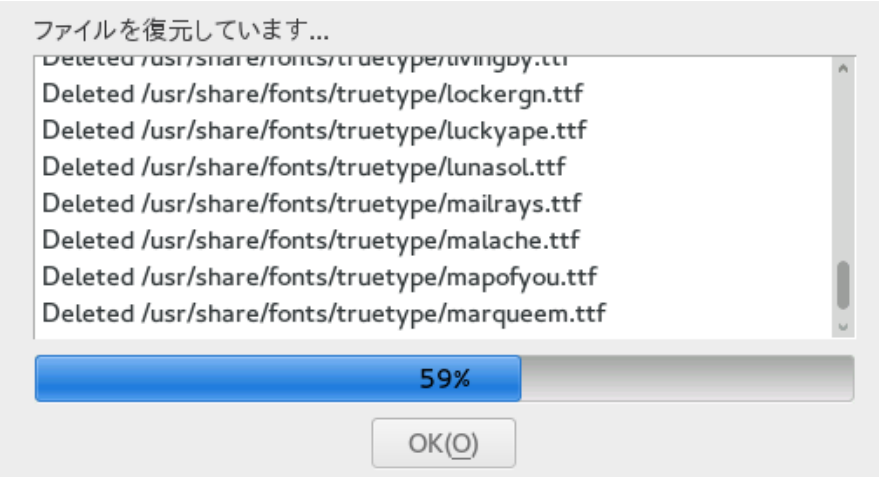


4. [変更点の表示]をクリックすると、2つのスナップショット間のファイルの差異がリストに表示されます。



5. ファイルのリストを確認します。事前および事後のファイル間の「差異」「」を表示するには、リストからファイルを選択します。

6. 1つまたは複数のファイルを復元するには、該当するチェックボックスをオンにして、関連するファイルまたはディレクトリを選択します。[選択したものを復元]をクリックし、[はい]をクリックして操作を確認します。



単一のファイルを復元する場合は、ファイル名をクリックして差分を表示します。[最初から復元する]をクリックし、[はい]をクリックして選択内容を確認します。

手順 4.2 `snapper`コマンドによる変更の取り消し

1. `snapper list -t pre-post`を実行すると、YaSTおよびZypperのスナップショットリストが表示されます。YaSTのスナップショットの場合は[説明]に「`yast`モジュール名」と表示され、Zypperのスナップショットの場合は「`zypp(zypper)`」と表示されます。

```
root # snapper list -t pre-post
Pre # | Post # | Pre Date                          | Post Date                         | Description
------+--------+-----------------------------------+-----------------------------------+-------------
311   | 312    | Tue 06 May 2014 14:05:46 CEST | Tue 06 May 2014 14:05:52 CEST | zypp(y2base)
340   | 341    | Wed 07 May 2014 16:15:10 CEST | Wed 07 May 2014 16:15:16 CEST | zypp(zypper)
342   | 343    | Wed 07 May 2014 16:20:38 CEST | Wed 07 May 2014 16:20:42 CEST | zypp(y2base)
344   | 345    | Wed 07 May 2014 16:21:23 CEST | Wed 07 May 2014 16:21:24 CEST | zypp(zypper)
346   | 347    | Wed 07 May 2014 16:41:06 CEST | Wed 07 May 2014 16:41:10 CEST | zypp(y2base)
348   | 349    | Wed 07 May 2014 16:44:50 CEST | Wed 07 May 2014 16:44:53 CEST | zypp(y2base)
350   | 351    | Wed 07 May 2014 16:46:27 CEST | Wed 07 May 2014 16:46:38 CEST | zypp(y2base)
```

2. スナップショットのペア間で変更されたファイルのリストを取得するには、以下を実行します。`snapper status PREPOST`. 内容が変更されたファイルには[c]のマーク、追加されたファイルには[+]のマーク、削除されたファイルには[-]のマークが付いています。

```
root # snapper status 350..351
+..... /usr/share/doc/packages/mikachan-fonts
+..... /usr/share/doc/packages/mikachan-fonts/COPYING
+..... /usr/share/doc/packages/mikachan-fonts/dl.html
c..... /usr/share/fonts/truetype/fonts.dir
c..... /usr/share/fonts/truetype/fonts.scale
+..... /usr/share/fonts/truetype/みかちゃん-p.ttf
+..... /usr/share/fonts/truetype/みかちゃん-pb.ttf
+..... /usr/share/fonts/truetype/みかちゃん-ps.ttf
+..... /usr/share/fonts/truetype/みかちゃん.ttf
c..... /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_64.cache-4
c..... /var/lib/rpm/Basenames
c..... /var/lib/rpm/Dirnames
c..... /var/lib/rpm/Group
c..... /var/lib/rpm/Installtid
c..... /var/lib/rpm/Name
c..... /var/lib/rpm/Packages
c..... /var/lib/rpm/Providename
c..... /var/lib/rpm/Requirename
c..... /var/lib/rpm/Sha1header
c..... /var/lib/rpm/Sigmd5
```

3. 特定のファイルの差異を表示するには、以下を実行します。`snapper diff PRE..POST ファイル名 ファイル名`を指定しない場合は、すべてのファイルの差異が表示されます。

```
root # snapper diff 350..351 /usr/share/fonts/truetype/fonts.scale
--- /.snapshots/350/snapshot/usr/share/fonts/truetype/fonts.scale 2014-04-23
 15:58:57.000000000 +0200
+++ /.snapshots/351/snapshot/usr/share/fonts/truetype/fonts.scale 2014-05-07
 16:46:31.000000000 +0200
@@ -1,4 +1,4 @@
-1174
+1486
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso10646-1
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso8859-1
[...]
```

4. 1つまたは複数のファイルを復元するには、以下を実行します。`snapper -v undochange PRE..POST ファイル名` `ファイル名`を指定しない場合は、変更されたすべてのファイルが復元されます。

```
root # snapper -v undochange 350..351
     create:0 modify:13 delete:7
     undoing change...
     deleting /usr/share/doc/packages/mikachan-fonts
     deleting /usr/share/doc/packages/mikachan-fonts/COPYING
     deleting /usr/share/doc/packages/mikachan-fonts/dl.html
     deleting /usr/share/fonts/truetype/みかちゃん-p.ttf
     deleting /usr/share/fonts/truetype/みかちゃん-pb.ttf
     deleting /usr/share/fonts/truetype/みかちゃん-ps.ttf
     deleting /usr/share/fonts/truetype/みかちゃん.ttf
     modifying /usr/share/fonts/truetype/fonts.dir
     modifying /usr/share/fonts/truetype/fonts.scale
     modifying /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-
x86_64.cache-4
     modifying /var/lib/rpm/Basenames
     modifying /var/lib/rpm/Dirnames
     modifying /var/lib/rpm/Group
     modifying /var/lib/rpm/Installtid
     modifying /var/lib/rpm/Name
     modifying /var/lib/rpm/Packages
     modifying /var/lib/rpm/Providename
     modifying /var/lib/rpm/Requirename
     modifying /var/lib/rpm/Sha1header
     modifying /var/lib/rpm/Sigmd5
     undoing change done
```

警告: ユーザ追加の取り消し

ユーザの追加を取り消す場合、Snapperで変更を取り消す方法はお勧めしません。特定のディレクトリはスナップショットから除外されているため、これらのユーザに属するファイルはファイルシステムに残ったままになります。削除済みユーザと同じユーザIDを持つユーザを作成した場合、このユーザはこれらのファイルを継承します。したがって、YaSTの［ユーザおよびグループ管理］ツールを使用して、ユーザを削除することを強くお勧めします。

4.2.2 Snapperを使用したファイルの復元

インストールスナップショットおよび管理スナップショットとは別に、Snapperはタイムラインスナップショットを作成します。このバックアップ用スナップショットを使用して、誤って削除したファイルを復元したり、ファイルの以前のバージョンを復元したりできます。Snapperの差分抽出機能を使用して、ある時点でどのファイルが変更されたのか調べることもできます。

ファイルの復元機能は、特に、デフォルトではスナップショットが作成されないサブボリュームまたはパーティションに存在するデータにとって重要です。ホームディレクトリからファイルを復元できるようにするには、たとえば、`/home`用に、自動的にタイムラインスナップショットを作成する別個のSnapper設定を作成します。手順については、4.4項「Snapper設定の作成と変更」を参照してください。

警告: ファイルの復元とロールバック

ルートファイルシステムから作成されたスナップショット(Snapperのルート設定で定義されています)を使用して、システムロールバックを実行できます。このようなロールバックを実行する場合にお勧めする方法は、そのスナップショットからブートしてからロールバックを実行する方法です。詳細については、4.3項 「スナップショットからのブートによるシステムロールバック」を参照してください。

次に説明するように、ルートファイルシステムスナップショットからすべてのファイルを復元することによってロールバックを実行することもできます。ただし、これはお勧めできません。たとえば、`/etc`ディレクトリから環境設定ファイルなど単一のファイルを復元できますが、スナップショットからファイルの完全なリストを復元することはできません。

この制限は、ルートファイルシステムから作成されたスナップショットにのみ影響します。

手順 4.3 YAST［SNAPPER］モジュールを使用したファイルの復元

1. YaSTの［その他］セクションにある［Snapper］モジュールを起動するか、「`yast2 snapper`」と入力します。
2. スナップショットの選択肢から［現在の設定］を選択します。
3. ファイルの復元元からタイムラインスナップショットを選択し、［変更点の表示］を選択します。タイムラインスナップショットは、タイプが［単一］で、説明の値が［timeline (タイムライン)］であるスナップショットです。
4. ファイル名をクリックしてテキストボックスからファイルを選択します。スナップショットバージョンと現在のシステムとの差分が表示されます。復元対象ファイルを選択するチェックボックスをオンにします。復元するすべてのファイルに対してこれを行います。
5. ［選択したものを復元］をクリックし、［はい］をクリックして操作を確認します。

手順 4.4 snapperコマンドを使用したファイルの復元

1. 次のコマンドを実行して、特定の設定のタイムラインスナップショットのリストを取得します。

```
snapper -c CONFIG list -t single | grep timeline
```

CONFIGは、既存のSnapper設定に置き換える必要があります。snapper list-configsを使用してリストを表示します。

2. 次のコマンドを実行して、指定のスナップショットの変更ファイルのリストを取得します。

```
snapper -c CONFIG status SNAPSHOT_ID>..0
```

SNAPSHOT_IDをファイルの復元元のスナップショットIDで置き換えます。

3. オプションで、次のコマンドを実行して、現在のファイルバージョンとスナップショットからのバージョンとの差分を一覧にします。

```
snapper -c CONFIG diff SNAPSHOT_ID..0 FILE NAME
```

<FILE NAME>を指定しない場合は、すべてのファイルの差分が表示されます。

4. 1つ以上のファイルを復元するには、以下を実行します

```
snapper -c CONFIG -v undochange
      SNAPSHOT_ID..0 FILENAME1 FILENAME2
```

ファイル名を指定しない場合は、変更されたすべてのファイルが復元されます。

4.3 スナップショットからのブートによるシステムロールバック

SUSE Linux Enterprise Serverに含まれているGRUB 2バージョンは、Btrfsスナップショットからブートできます。Snapperのロールバック機能と併用することで、誤設定されたシステムを回復できます。Snapperによって作成されたすべてのスナップショットはブートに使用することができ、ブートメニューから選択できます。

重要: ブート可能なスナップショット

ブート可能なスナップショットは、ルートファイルシステムからのスナップショット(Snapperのルート設定によって定義されます)のみです。

スナップショットをブートする場合、スナップショットに含まれているファイルシステムの該当部分が読み込み専用でマウントされます。スナップショットから除外されている他のすべてのファイルシステムと該当部分は読み書き可能でマウントされ、変更できます。

重要: 変更の取り消しとロールバック

スナップショットを操作してデータを復元する場合、Snapperが処理可能なシナリオとして、根本的に異なる次の2つのシナリオがあることを理解することが重要です。

変更の取り消し

4.2項 「Snapperを使用した変更の取り消し」で説明されているように、変更を取り消す場合は、2つのスナップショットが比較され、これらの2つのスナップショット間の変更が元に戻されます。この方法を使用すると、選択したファイルを復元から明示的に除外することもできます。

ロールバック

次に説明する方法でロールバックを実行すると、システムはスナップショットが作成された状態にリセットされます。

ブート可能なスナップショットからロールバックを行うには、次の要件を満たす必要があります。デフォルトインストールを行った場合、システムはそのように設定されます。

ブート可能なスナップショットからのロールバックの要件

- ルートファイルシステムは、Btrfsである必要があります。LVMボリュームスナップショットからのブートはサポートされていません。
- ルートファイルシステムは、単一のデバイス、単一のパーティション、および単一のサブボリューム上にある必要があります。`/srv` などスナップショットから除外されるディレクトリ(完全なリストについては、スナップショットから除外されるディレクトリを参照)は、別のパーティション上に存在していても構いません。
- システムは、インストールされたブートローダを介してブート可能である必要があります。

ブート可能なスナップショットからのロールバックを実行するには、次の手順に従います。

1. システムをブートします。ブートメニューから、[Bootable snapshots (ブート可能なスナップショット)]を選択して、ブートするスナップショットを選択します。スナップショットのリストが日別に一覧にされます。最新のスナップショットが先頭に表示されます。

2. システムにログインします。すべてが予期したとおりに動作しているかどうかを注意深く確認します。スナップショットの一部であるディレクトリには書き込むことができないことに注意してください。他のディレクトリに書き込むデータは、次に行う操作にかかわらず、失われることは「ありません」。

3. ロールバックを実行するかどうかに応じて、次のステップを選択します。

    a. システムがロールバックが不要な状態である場合、再起動して現在のシステム状態でブートするか、再起動して別のスナップショットを選択するか、または再起動してレスキューシステムを起動します。

    b. ロールバックを実行する場合は、次のコマンドを実行します。

    ```
    sudo snapper rollback
    ```

    その後、再起動します。ブート画面で、デフォルトのブートエントリを選択して、復元されたシステムで再起動します。

4.3.1 制限

システム全体をスナップショット作成時点とまったく同じ状態に復元する、システムの「完全な」ロールバックは不可能です。

4.3.1.1 スナップショットから除外されるディレクトリ

ルートファイルシステムのスナップショットには、すべてのディレクトリが含まれるわけではありません。詳細および理由については、スナップショットから除外されるディレクトリを参照してください。そのため、一般的にこれらのディレクトリのデータは復元されないため、次の制限が生じます。

ロールバック後、アドオンおよびサードパーティソフトウェアを使用できない場合がある

スナップショットから除外されるサブボリューム(`/opt`など)にデータをインストールするアプリケーションやアドオンは、アプリケーションデータの他の部分がスナップショットに含まれるサブボリュームにもインストールされている場合、ロールバック後に動作しない場合があります。この問題を解決するには、アプリケーションまたはアドオンを再インストールします。

ファイルアクセスの問題

スナップショットと現在のシステムでファイルのパーミッションまたは所有権、あるいはその両方がアプリケーションによって変更されている場合、そのアプリケーションは該当するファイルにアクセスできない場合があります。ロールバック後、影響を受けるファイルのパーミッションまたは所有権、あるいはその両方をリセットします。

互換性のないデータ形式

サービスまたはアプリケーションがスナップショットと現在のシステムとの間に新しいデータ形式を設定した場合、ロールバック後、そのアプリケーションは影響を受けたデータファイルを読み込めない場合があります。

コードとデータが混在するサブボリューム

`/srv` のようなサブボリュームには、コードとデータが混在する場合があります。ロールバックの結果、コードが機能しなくなる場合があります。たとえば、PHPのバージョンがダウングレードされ、WebサーバのPHPスクリプトが壊れる場合があります。

ユーザデータ

ロールバックによりシステムからユーザが削除された場合、これらのユーザが、スナップショットから除外されているディレクトリ内で所有していたデータは削除されません。同じユーザIDを持つユーザが作成された場合、そのユーザは該当ファイルを継承します。`find` のようなツールを使用して、孤立したファイルを検索して削除します。

4.3.1.2 ブートローダのデータはロールバックできない

ブートローダはロールバックできません。これは、ブートローダのすべての「」「ステージ」が整合している必要があるためです。これは、ロールバックを実行する際には保証できません。

4.4 Snapper設定の作成と変更

Snapperの動作は、各パーティションまたは `Btrfs` サブボリュームに固有の設定ファイルで定義できます。これらの設定ファイルは、`/etc/snapper/configs/` に保存されます。`/` ディレクトリに対してSnapperでインストールされるデフォルトの設定ファイルが `root` です。このファイルは、YaSTとZypperのスナップショットを作成し管理するほか、`/` に対する毎時のバックアップスナップショットも作成および管理します。

Btrfsでフォーマットされたその他のパーティションやBtrfsパーティション上の既存のサブボリュームに対して、独自の設定ファイルを作成できます。以下の例では、/srv/wwwにマウントされたBtrfsフォーマットのパーティションに保存されたWebサーバデータをバックアップするSnapper設定を作成します。

設定が作成された後で、snapper自体またはYaSTの[Snapper]モジュールを使用して、これらのスナップショットからファイルを復元できます。YaSTの場合は[現在の設定]を選択する必要があります。snapperの場合は、グローバルスイッチ-cを使用して設定を指定する必要があります(例: snapper -c myconfig list)。

新しいSnapper設定を作成するには、snapper create-configを実行します。

```
snapper -c www-data❶ create-config /srv/www❷
```

❶ 設定ファイルの名前。

❷ スナップショットを作成するパーティションまたはBtrfsサブボリュームのマウントポイント。

このコマンドにより、新しい設定ファイル/etc/snapper/configs/www-dataが作成され、/etc/snapper/config-templates/defaultから取得されたデフォルト値が使用されます。これらのデフォルトの調整方法については、4.4.1項「既存の設定の管理」を参照してください。

ヒント: 設定のデフォルト値

新しい設定ファイルのデフォルト値は/etc/snapper/config-templates/defaultから取得されます。独自のデフォルトセットを使用する場合は、同じディレクトリ内にこのファイルのコピーを作成し、必要に応じて調整してください。作成したファイルを使用するには、create-configコマンドで-tオプションを指定します。

```
snapper -c www-data create-config -t my_defaults /srv/www
```

4.4.1 既存の設定の管理

snapperは、既存の設定を管理するためのサブコマンドを備えています。これらの設定を一覧、表示、削除、および変更することができます。

設定の一覧

既存の設定をすべて取得するには、snapper list-configsコマンドを使用します。

```
root # snapper list-configs
```

```
Config | Subvolume
-------+----------
root   | /
usr    | /usr
local  | /local
```

設定の削除

設定を削除するには、`snapper -c CONFIG delete-config` サブコマンドを使用します。`Config` は、`snapper list-configs` で表示される設定名に置き換える必要があります。

設定の表示

指定した設定を表示するには、`snapper -c CONFIG get-config` サブコマンドを使用します。`Config` は、`snapper list-configs` で表示される設定名に置き換える必要があります。設定オプションの詳細については、4.4.1.1項「設定データ」を参照してください。

指定した設定のオプションを変更するには、`snapper -c CONFIG set-config OPTION=VALUE` サブコマンドを使用します。`Config` は、`snapper list-configs` で表示される設定名に置き換える必要があります。`OPTION` および `VALUE` に指定可能な値は、4.4.1.1項 「設定データ」に一覧にされています。

4.4.1.1 設定データ

各設定には、コマンドラインから変更可能なオプションのリストが含まれています。次のリストは、各オプションの詳細を示しています。

`ALLOW_GROUPS`、`ALLOW_USERS`

通常のユーザにスナップショットを使用するパーミッションを付与します。詳細については、4.4.1.2項「通常ユーザとしてのSnapperの使用」を参照してください。
デフォルト値は `""` です。

`BACKGROUND_COMPARISON`

事前および事後スナップショットを作成後にバックグラウンドで比較するかどうかを定義します。
デフォルト値は `yes (はい)` です。

EMPTY_PRE_POST_CLEANUP

`yes (はい)` に設定した場合、違いがない事前および事後スナップショットのペアは削除されます。
デフォルト値は `no (いいえ)` です。

EMPTY_PRE_POST_MIN_AGE

違いがない事前および事後スナップショットのペアが自動削除の対象となるまでの最短期間を秒単位で定義します。
デフォルト値は 1800 です。

FSTYPE

パーティションのファイルシステムタイプ。変更しません。
デフォルト値は btrfs です。

NUMBER_CLEANUP

合計スナップショット数が NUMBER_LIMIT で指定した数を超え、かつ NUMBER_MIN_AGE で指定した保存期間を超えた場合に、古いインストールスナップショットおよび管理スナップショットを自動的に削除するかどうかを定義します。有効な値: yes (はい)、no (いいえ)
デフォルト値は no (いいえ) です。

注記: 制限と保存期間

NUMBER_LIMIT、NUMBER_LIMIT_IMPORTANT、および NUMBER_MIN_AGE は常に評価されます。スナップショットが削除されるのは、「すべての」条件を満たしている場合のみです。保存期間に関係なく一定数のスナップショットを常に保持したい場合は、NUMBER_MIN_AGE を 0 に設定します。一方、一定の保存期間を超えたスナップショットをすべて削除したい場合は、NUMBER_LIMIT および NUMBER_LIMIT_IMPORTANT を 0 に設定します。

NUMBER_LIMIT

NUMBER_CLEANUP が yes に設定されている場合に保持する、重要とマークされていないインストールスナップショットおよび管理スナップショットのペア数を定義します。最も新しいスナップショットのみが保持されます。
デフォルト値は 50 です。

NUMBER_LIMIT_IMPORTANT

NUMBER_CLEANUP が yes に設定されている場合に保持する、重要とマークされたスナップショットペアの数を定義します。最も新しいスナップショットのみが保持されます。
デフォルト値は 10 です。

NUMBER_MIN_AGE

スナップショットペアが自動削除の対象になるまでの最短期間を秒単位で定義します。
デフォルト値は 1800 です。

`SUBVOLUME`

スナップショットを作成するパーティションまたはサブボリュームのマウントポイント。変更しません。

`SYNC_ACL`

Snapperが通常ユーザによって使用される場合(4.4.1.2項「通常ユーザとしてのSnapperの使用」を参照)、ユーザは `.snapshot` ディレクトリにアクセスして、そのディレクトリ内のファイルを読み取ることができる必要があります。SYNC_ACLを `yes (はい)` に設定した場合、Snapperは自動的に、ALLOW_USERSまたはALLOW_GROUPSエントリからACLを使用してユーザとグループがファイルにアクセスできるようにします。
デフォルト値は `no (いいえ)` です。

`TIMELINE_CLEANUP`

スナップショット数が `TIMELINE_LIMIT_*` オプションで指定した数を超え、かつ `TIMELINE_MIN_AGE` で指定した保存期間を超えた場合に、古いスナップショットを自動的に削除するかどうか定義します。有効な値: `yes(はい)`、`no(いいえ)`
デフォルト値は `no (いいえ)` です。

`TIMELINE_CREATE`

`yes (はい)` に設定されている場合は、毎時スナップショットが作成されます。現時点では、これがスナップショットを自動的に作成する唯一の方法なので、`yes(はい)` に設定することを強くお勧めします。有効な値: `yes(はい)`、`no(いいえ)`
デフォルト値は `no (いいえ)` です。

`TIMELINE_LIMIT_DAILY`、`TIMELINE_LIMIT_HOURLY`、`TIMELINE_LIMIT_MONTHLY`、`TIMELINE_LIMIT_YEARLY`

1時間、1日、1カ月間、1年間に保持するスナップショット数です。
各エントリのデフォルト値は `10` です。

例 4.1 タイムライン設定の例

```
TIMELINE_CLEANUP="yes"
TIMELINE_CREATE="yes"
TIMELINE_LIMIT_DAILY="10"
TIMELINE_LIMIT_HOURLY="10"
TIMELINE_LIMIT_MONTHLY="10"
TIMELINE_LIMIT_YEARLY="10"
TIMELINE_MIN_AGE="1800"
```

この設定例では、毎時スナップショットが自動的に削除されます。TIMELINE_MIN_AGEとTIMELINE_LIMIT_*は常に両方が評価されます。この例では、スナップショットが削除対象となるまでの最短期間が30分(180秒)に設定されています。毎時のスナップショットを作成するので、最新のスナップショットだけが保持されることになります。TIMELINE_LIMIT_DAILYをゼロ以外に設定すると、1日の最初のスナップショットが保持されることになります。

保持されるスナップショット

- 1時間ごと:最新の10個のスナップショットが保持されます。
- 1日ごと:それぞれの日の最初のスナップショットが、直近の10日分保持されます。
- 1カ月ごと:それぞれの月の最後の日に作成された最初のスナップショットが、直近の10カ月分保持されます。
- 1年ごと:それぞれの年の最後の日に作成された最初のスナップショットが、直近の10年分保持されます。

TIMELINE_MIN_AGE

スナップショットが自動削除の対象となるまでの最短期間を秒単位で定義します。デフォルト値は1800です。

4.4.1.2 通常ユーザとしてのSnapperの使用

デフォルトでは、rootしかSnapperを使用できません。しかし、以下のような場合、特定のグループまたはユーザがスナップショットを作成したり、スナップショットを使って変更を取り消したりできる必要があります。

- Webサイト管理者が/srv/wwwのスナップショットを作成したい場合
- ユーザが自身のホームディレクトリのスナップショットを作成したい場合

このような場合、ユーザやグループにパーミッションを与えるSnapper設定を作成できます。対応する.snapshotsディレクトリは、指定されたユーザによって読み込みおよびアクセス可能である必要があります。このための最も簡単な方法は、SYNC_ACLオプションをyes (はい)に設定することです。

手順 4.5 通常ユーザによるSNAPPER使用の有効化

次のすべての手順はrootとして実行する必要があります。

1. ユーザがSnapperを使用するパーティションまたはサブボリュームにSnapper設定がない場合は、作成します。手順については、4.4項「Snapper設定の作成と変更」を参照してください。例:

```
snapper --config web_data create /srv/www
```

2. `/etc/snapper/configs/CONFIG`に設定ファイルを作成します。CONFIGは、前の手順で`-c/--config`を使用して指定される値です(`/etc/snapper/configs/web_data`など)。必要に応じて設定ファイルを調整します。詳細は4.4.1項「既存の設定の管理」を参照してください。

3. `ALLOW_USERS`と`ALLOW_GROUPS`、またはその一方の値を設定し、ユーザやグループにパーミッションを与えます。複数のエントリは`Space`で区切ってください。たとえば、ユーザ`www_admin`にパーミッションを与えるには、次のように入力します。

```
snapper -c web_data set-config "ALLOW_USERS=www_admin" SYNC_ACL="yes"
```

4. これで、指定されたユーザやグループが特定のSnapper設定を使用できます。以下のように`list`コマンドを使ってテストできます。

```
www_admin:~ > snapper -c web_data list
```

4.5 スナップショットの手動での作成と管理

Snapperは設定によって自動的にスナップショットを作成および管理するだけのものではありません。コマンドラインツールまたはYaSTモジュールを使用して、手動でスナップショットのペア(「」「事前および事後」)や単一のスナップショットを作成することもできます。

Snapperのすべての操作は既存の設定に対して実行されます(詳細は4.4項 「Snapper設定の作成と変更」を参照)。スナップショットを作成するには、対象のパーティションまたはボリュームに対して設定が存在する必要があります。デフォルトで、システム設定(`root`)が使用されます。独自の設定に対してスナップショットを作成または管理する場合は、明示的にその設定を選択する必要があります。YaSTの[現在の設定]ドロップダウンボックスを使用するか、コマンドラインで`-c`を指定します(`snapper -c MYCONFIG COMMAND`)。

4.5.1 スナップショットのメタデータ

各スナップショットには、スナップショット自体とメタデータが含まれています。スナップショットを作成する場合は、メタデータも指定する必要があります。スナップショットを修正すると、メタデータが変更されます。コンテンツを修正することはできません。各スナップショットについて、以下のメタデータを利用できます。

- Type(種類):スナップショットの種類です。詳細は4.5.1.1項 「スナップショットの種類」を参照してください。このデータは変更できません。
- Number(番号):スナップショットの一意の番号。このデータは変更できません。
- Pre Number(前番号):対応する事前スナップショットの番号を指定します。事後スナップショットにのみ適用されます。このデータは変更できません。
- Description(説明):スナップショットの説明です。
- Userdata (ユーザデータ): カンマ区切りの「キー=値」のリスト形式でカスタムデータを指定できる、拡張用の項目です。(例: `reason=testing, project=foo`)。このフィールドは、スナップショットに重要のマークを付ける場合(`important=yes`)や、スナップショットを作成したユーザを一覧にする場合(user=tux)にも使用されます。
- Cleanup-Algorithm(クリーンアップアルゴリズム):スナップショットのクリーンアップアルゴリズムです。詳細は4.5.1.2項 「クリーンアップアルゴリズム」を参照してください。

4.5.1.1 スナップショットの種類

Snapperには、事前(pre)、事後(post)、および単一(single)の3種類のスナップショットがあります。これらは物理的には同じものですが、Snapperでは別のものとして扱われます。

`pre`(事前)

変更前のファイルシステムのスナップショットです。各`pre(事前)`スナップショットには、対応する`post(事後)`スナップショットがあります。たとえば、YaST/Zypperの自動スナップショットに対して使用します。

`post`(事後)

変更後のファイルシステムのスナップショットです。各`post(事後)`スナップショットには、対応する`pre(事前)`スナップショットがあります。たとえば、YaST/Zypperの自動スナップショットに対して使用します。

`single`(単一)

スタンドアロンのスナップショットです。たとえば、自動毎時スナップショットに使用します。これは、スナップショットを作成する際のデフォルトの種類です。

4.5.1.2 クリーンアップアルゴリズム

Snapperには、古いスナップショットのクリーンアップアルゴリズムが3種類あります。このアルゴリズムは、日次のcronジョブとして実行されます。クリーンアップの頻度は、パーティションまたはサブボリュームのSnapper設定で定義されます(詳細は4.4.1項「既存の設定の管理」を参照)。

number(番号)
: スナップショットが特定の数に達すると、古いスナップショットを削除します。

timeline (タイムライン)
: 特定の期間が経過したスナップショットを削除しますが、毎時、毎日、毎月、および毎年のスナップショットを一定数保持します。

empty-pre-post(事前事後の差分なし)
: 事前と事後のスナップショットに差分がない場合、そのペアを削除します。

4.5.2 スナップショットの作成

スナップショットを作成するには、`snapper create`を実行するか、YaSTの[Snapper]モジュールで[作成]をクリックします。以下は、コマンドラインを使ってスナップショットを作成する場合の例です。YaSTインタフェースを使用している場合、これらの例は簡単に採用できます。

ヒント: Snapshot Description

後で識別しやすくするため、わかりやすい説明を指定しておいてください。ユーザデータオプションを使って、さらに情報を指定することもできます。

`snapper create --description "Snapshot for week 2 2014"`
: 説明付きのスタンドアロンのスナップショット(種類はsingle)を、デフォルト(`root`)設定で作成します。クリーンアップアルゴリズムは指定されていないので、自動的にスナップショットが削除されることはありません。

`snapper --config home create --description "Cleanup in ~tux"`
: 説明付きのスタンドアロンのスナップショット(種類はsingle)を、カスタム設定`home`で作成します。クリーンアップアルゴリズムは指定されていないので、自動的にスナップショットが削除されることはありません。

`snapper --config home create --description "Daily data backup" --cleanup-algorithm timeline`

説明付きのスタンドアロンのスナップショット(種類はsingle)を、カスタム設定 `home` 設定で作成します。設定のタイムライン(timeline)クリーンアップアルゴリズムで指定された条件が満たされると、ファイルが自動的に削除されます。

`snapper create --type pre--print-number--description "Before the Apache config cleanup"--userdata "important=yes"`

種類が `pre` のスナップショットを作成し、スナップショット番号を出力します。「」「事前」と「」「事後」の状態を保存するために使用されるスナップショットペアを作成するために必要な、最初のコマンドです。スナップショットには重要のマークが付きます。

`snapper create --type post--pre-number 30--description "After the Apache config cleanup"--userdata "important=yes"`

番号 `30` の `pre` スナップショットとペアになる `post` スナップショットを作成します。「」「事前」と「」「事後」の状態を保存するために使用されるスナップショットペアを作成するために必要な、2番目のコマンドです。スナップショットには重要のマークが付きます。

`snapper create --command COMMAND--description "Before and after COMMAND"`

`COMMAND` の実行前後に自動的にスナップショットを作成します。このオプションを使用できるのは、コマンドラインでsnapperを使用する場合のみです。

4.5.3 スナップショットのメタデータ修正

Snapperでは、説明、クリーンアップアルゴリズム、およびスナップショットのユーザデータを修正できます。それ以外のメタデータは変更できません。以下は、コマンドラインを使ってスナップショットを修正する場合の例です。YaSTインタフェースを使用している場合、これらの例は簡単に採用できます。

コマンドラインでスナップショットを修正するには、スナップショットの番号がわかっている必要があります。`snapperlist` を実行すると、すべてのスナップショットとその番号が表示されます。

YaSTの[Snapper]モジュールでは、すでにすべてのスナップショットのリストが表示されています。リストからスナップショットを選択し、[Modify]をクリックします。

`snapper modify --cleanup-algorithm "timeline" 10`

デフォルト(`root`)設定のスナップショット10番のメタデータを修正します。クリーンアップアルゴリズムが `timeline` に設定されます。

`snapper --config home modify --description "daily backup" -cleanup-algorithm "timeline"120`

カスタム設定 `home` のスナップショット120番のメタデータを修正します。新しい説明が設定され、クリーンアップアルゴリズムを無しに設定します。

4.5.4 スナップショットの削除

YaSTの［Snapper］モジュールを使用してスナップショットを削除するには、リストからスナップショットを選択して［Delete (削除)］をクリックします。

コマンドラインツールを使ってスナップショットを削除するには、スナップショットの番号が分かっている必要があります。`snapper list` を実行して番号を調べます。スナップショットを削除するには、`snapper delete NUMBER` を実行します。

ヒント: スナップショットペアの削除

`pre` スナップショットを削除する場合は、必ず、対応する `post` スナップショットを削除する必要があります(逆も同様です)。

`snapper delete 65`

デフォルト(`root`)設定のスナップショット65番を削除します。

`snapper -c home delete 89 90`

カスタム設定 `home` のスナップショット89番および90番を削除します。

ヒント: 参照されていないスナップショットの削除

Btrfsスナップショットが存在していても、メタデータを持つ `snapper` のXMLファイルではない場合があります。したがって、`snapper` に対しては、スナップショットは存在しません。`SNAPSHOT_NUMBER` ディレクトリを削除できるようにするには、まずBtrfsサブボリュームを削除する必要があります。

```
btrfs subvolume delete /.snapshots/SNAPSHOTNUMBER/snapshot
    rm -rf /.snapshots/SNAPSHOTNUMBER
```

ヒント: 古いスナップショットほどディスク容量を使用

ハードディスクの容量を空けるためにスナップショットを削除する場合は、古いスナップショットから削除します。古いスナップショットほど、多くの容量を使用します。

スナップショットは、日次のcronジョブでも自動的に削除されます。詳細については、4.5.1.2項 「クリーンアップアルゴリズム」を参照してください。

4.6 よくある質問とその回答

問: Snapperでは `/var/log`、`/tmp` などのディレクトリの変更が表示されませんが、なぜですか?

答: 一部のディレクトリについては、スナップショットから除外することに決定しました。リストと理由については、スナップショットから除外されるディレクトリを参照してください。スナップショットからパスを除外するため、これらのパス用にサブボリュームを作成しています。

問: スナップショットはどのくらいのディスク容量を使用しますか?
また、どうすればディスク容量を解放できますか?

答: `Btrfs` ファイルシステムでは `df` が正しいディスクの使用率を表示しないため、コマンド `btrfs filesystem df MOUNT_POINT` を使用する必要があります。現時点では、`Btrfs` ツールで、スナップショットが使用するディスク容量を表示できません。

スナップショットを含む `Btrfs` パーティションの容量を空けるには、ファイルではなく、不要なスナップショットを削除する必要があります。古いスナップショットは、最近のスナップショットよりも多くの領域を使用します。詳細については、4.1.1.3項 「スナップショットのアーカイブの制御」を参照してください。

あるサービスパックから別のサービスパックにアップグレードすると、多くのデータが変更される(パッケージのアップデート)ので、スナップショットにより、システムのサブボリュームで多くのディスク容量が使用されます。これらのスナップショットが不要になった場合は、手動で削除することをお勧めします。詳細については、4.5.4項「スナップショットの削除」を参照してください。

問: ブートローダからスナップショットをブートできますか?

答: はい。詳細については、4.3項「スナップショットからのブートによるシステムロールバック」を参照してください。

問: Snapperに関する詳細情報はどこで入手できますか?

答： Snapperのホームページ(http://snapper.io/)を参照してください。

5 VNCによるリモートアクセス

VNC (Virtual Network Computing)では、グラフィカルなデスクトップを使用してリモートコンピュータを制御できます。これは、リモートシェルアクセスとは対照的です。VNCはプラットフォームに依存しないので、VNCを使用すれば、任意のオペレーティングシステムからリモートマシンにアクセスできます。

SUSE Linux Enterprise Serverでは、2種類のVNCセッションをサポートしています。1つはクライアントからのVNC接続が続く限り、「」「存続する」一時的セッションで、もう1つは明示的に終了されるまで「」「存続する」永続的セッションです。

注記: セッションタイプ

両方のタイプのセッションを1つのコンピュータの異なるポートから同時に提供ができます。ただし、オープンセッションを1つのタイプからもう一方のタイプに変換することはできません。

5.1 一時的VNCセッション

一時的セッションは、リモートクライアントによって開始されます。これにより、サーバにグラフィカルなログイン画面が開きます。この画面でセッションを開始するユーザを選択できます。さらに、ログインマネージャでサポートされている場合はデスクトップ環境も選択できます。そのようなVNCセッションへのクライアント接続を終了すると、そのセッション内で開始したアプリケーションもすべて終了します。一時的なVNCセッションは共用できませんが、1つのホストで同時に複数のセッションを実行することは可能です。

手順 5.1 一時的VNCセッションの有効化

1. まず、[YaST] › [ネットワークサービス] › [リモート管理(VNC)]の順に選択します。
2. [(リモート管理を許可する]にチェックマークを付けます。
3. 必要な場合は、[ファイアウォールでポートを開く]にもチェックマークを付けます (たとえば、ネットワークインタフェースを外部ゾーンに属するように設定する場合)。ネットワークインタフェースが複数ある場合は、[ファイアウォールの詳細]で、特定のインタフェースにだけファイアウォールポートを開くように制限します。
4. [完了]で設定を確認します。
5. 必要なパッケージの一部をまだ入手できない場合は、足りないパッケージのインストールを承認する必要があります。

注記: 使用可能な設定

SUSE Linux Enterprise Serverのデフォルト設定では、1024x768ピクセルの解像度と16ビットの色数でセッションが提供されます。セッションで使用できるポートは、「正規の」 VNCビューアの場合はポート`5901`(VNCディスプレイ`1`に相当)、Webブラウザの場合はポート`5801`です。

その他の設定は、異なるポートで使用できます。5.1.2項「一時的VNCセッションを設定する」を参照してください。

VNCディスプレイ番号とXディスプレイ番号は、一時的セッションでは互いに独立しています。VNCディスプレイ番号は、サーバがサポートするすべての設定に手動で割り当てられます(上記の例では1)。VNCセッションは、設定の1つを使用して開始されるたびに、自動的に未使用のXディスプレイ番号を取得します。

5.1.1 一時的VNCセッションを開始する

一時的VNCセッションを開始するには、VNCビューアをクライアントコンピュータにインストールしておく必要があります。SUSE Linux製品の標準ビューアは、`tigervnc`パッケージ(デフォルト)または`tightvnc`パッケージで提供される`vncviewer`です。WebブラウザとJavaアプレットを使用してVNCセッションを表示することもできます。

VNCビューアを起動し、サーバのデフォルト設定でセッションを開始するには、次のコマンドを使用します。

```
vncviewer jupiter.example.com:1
```

VNCディスプレイ番号の代わりに、2つのコロンを使用してポート番号を指定することもできます。

```
vncviewer jupiter.example.com::5901
```

または、Javaを有効にしたWebブラウザで、URLとして「`http://jupiter.example.com:5801`」を入力することにより、VNCセッションを表示できます。

5.1.2 一時的VNCセッションを設定する

デフォルト設定を変更する必要も意志もない場合は、このセクションをスキップできます。

一時的VNCセッションは、`xinetd`デーモンを介して開始されます。設定ファイルは、`/etc/xinetd.d/vnc`にあります。このファイルは、デフォルトで、6つの設定ブロックを提供します: VNCビューア用に3ブロック(`vnc1`から`vnc3`まで)、Javaアプレット用に3ブロック(`vnchttpd1`から`vnchttpd3`まで)。デフォルトでは、`vnc1` と `vnchttpd1`だけが有効です。

設定を有効にするには、`disable = yes`行の最初のカラムに`#`文字を付けて行をコメント化するか、その行を完全に削除します。設定を無効にするには、その行をコメント解除するか、追加します。

`Xvnc`サーバは、`server_args`オプションで設定できます。オプションのリストについては、`Xnvc --help`を参照してください。

カスタム設定を追加する際には、それらの設定が、同じホスト上の他の設定、他のサービス、または既存の永続的VNCセッションですでに使用中のポートを使用しないことを確認してください。

設定の変更を有効にするには、次のコマンドを入力します:

```
sudo rcxinetd reload
```

重要: ファイアウォールとVNCポート

手順5.1「一時的VNCセッションの有効化」で説明されているように、リモート管理をアクティブにすると、ファイアウォール内でポート`5801`および`5901`が開きます。VNCセッションで使用されるネットワークインタフェースがファイアウォールで保護されている場合、VNCセッションの追加ポートをアクティブにする際には各ポートを手動で開く必要があります。手順については、Book “Security Guide” 15 “Masquerading and Firewalls”を参照してください。

5.2 永続的VNCセッション

永続的VNCセッションは、サーバ上で開始されます。セッションとこのセッションで開始されたすべてのアプリケーションは、クライアント接続とは関わりなく、セッションが終了するまで実行されます。

永続的セッションは、複数のクライアントから同時にアクセスすることが可能です。この機能では、1つのクライアントがフルアクセスをもち、他のすべてのクライアントが表示専用アクセスを持つため、デモ用途に最適です。また、講師が受講生のデスクトップにアクセスする必要があるトレーニングでも使用できます。ただし、ほとんどの場合、VNCセッションの共用が必要とされることはありません。

ディスプレイマネージャを起動する一時的セッションとは対照的に、永続的セッションでは、操作準備のできたデスクトップを起動し、そのデスクトップがVNCセッションを開始したユーザとしてセッションを実行します。

永続的セッションへのアクセスは、可能な2タイプのパスワードによって保護されます:

- フルアクセスを付与する通常のパスワード。または、
- 非対話的(表示オンリー)アクセスを付与するオプションの表示オンリーパスワード。

1つのセッションに、両方の種類のクライアント接続が一度に複数存在できます。

手順 5.2 永続的VNCセッションを開始する

1. シェルを開き、VNCセッションを所有するユーザとしてログインしていることを確認します。
2. VNCセッションで使用されるネットワークインタフェースがファイアウォールで保護されている場合は、ファイアウォール内でセッションによって使用されるポートを手動で開く必要があります。複数のセッションを開始する場合は、一連のポートを開くことができます。ファイアウォールの設定方法の詳細については、Book “Security Guide” 15 “Masquerading and Firewalls”を参照してください。
   `vncserver`は、ディスプレイ`:1`にはポート`5901` 、ディスプレイ`:2`にはポート`5902`という順序でポートを使用します。永続的セッションの場合、VNCディスプレイとXディスプレイは、通常、同じ番号です。
3. 1024x769ピクセルの解像度と16ビットの色数でセッションを開始するには、次のコマンドを入力します。

   ```
   vncserver -geometry 1024x768 -depth 16
   ```

   `vncserver`コマンドは、何も指定されない場合、未使用のディスプレイ番号を選択し、その選択内容を出力します。追加オプションについては、`man 1 vncserver`を参照してください。

初めて`vncviewer`を実行すると、セッションへのフルアクセス用パスワードが要求されます。必要な場合は、セッションへの表示オンリーアクセス用パスワードも入力できます。

ここで指定するパスワードは、同じユーザによって開始される今後のセッションにも使用されます。それらのパスワードは、`vncpasswd`コマンドで変更できます。

重要: セキュリティ上の考慮事項

必ず、かなりの長さ(8文字以上)の強力なパスワードを使用してください。これらのパスワードは共用しないでください。

VNC接続は暗号化されていないので、2つのコンピュータ間のネットワークを傍受できる者たちによってセッション開始時に転送されるパスワードが読み取られる恐れがあります。

VNCセッションを終了するには、通常のローカルXセッションのシャットダウンのように、VNC環境内部で実行中のデスクトップ環境をVCNビューアからシャットダウンします。

セッションを手動で終了したい場合は、VNCサーバでシェルを開き、終了したいVNCセッションを所有するユーザとしてログインしていることを確認します。次のコマンドを実行して、ディスプレイ `:1` で実行されているセッションを終了します: `vncserver -kill :1`

5.2.1 永続的VNCセッションに接続する

永続的VNCセッションに接続するには、VCNビューアをインストールする必要があります。SUSE Linux製品の標準ビューアは、`tigervnc` パッケージ(デフォルト)または `tightvnc` パッケージで提供される `vncviewer` です。WebブラウザとJavaアプレットを使用してVNCセッションを表示することもできます。

VNCビューアを起動し、VNCサーバのディスプレイ `:1` に接続するには、次のコマンドを使用します。

```
vncviewer jupiter.example.com:1
```

VNCディスプレイ番号の代わりに、2つのコロンを使用してポート番号を指定することもできます。

```
vncviewer jupiter.example.com::5901
```

または、Javaを有効にしたWebブラウザで、URLとして「`http://jupiter.example.com:5801`」を入力することにより、VNCセッションを表示できます。

5.2.2 永続的VNCセッションを設定する

永続的VNCセッションは、`$HOME/.vnc/xstartup` を編集することによって設定できます。デフォルトでは、このシェルスクリプトは、起動元と同じGUI/ウィンドウマネージャを起動します。SUSE Linux Enterprise Serverでは、これは、GNOMEまたはIceWMのいずれかです。好みのウィンドウマネージャでセッションを開始する場合は、変数 `WINDOWMANAGER` を設定します。

```
WINDOWMANAGER=gnome vncserver -geometry 1024x768
WINDOWMANAGER=icewm vncserver -geometry 1024x768
```

注記: ユーザごとに1つの設定

永続的VNCセッションは、ユーザごとの単一設定として設定されます。同じユーザが開始する複数のセッションでは、すべて同じ起動ファイルとパスワードファイルが使用されます。

6 コマンドラインツールによるソフトウェアの管理

この章では、ソフトウェア管理の2つのコマンドラインツールとして、ZypperとRPMについて説明します。このコンテキストで使用される述語(たとえば、`repository`、`patch`、`update`など)の定義については、ブック「導入ガイド」9「ソフトウェアをインストールまたは削除する」9.1「用語の定義」を参照してください。

6.1 Zypperの使用

Zypperは、パッケージのインストール、更新、削除、およびリポジトリの管理を行うためのコマンドラインパッケージマネージャです。これは特に、リモートソフトウェア管理タスクの実行、またはシェルスクリプトからのソフトウェアの管理で役立ちます。

6.1.1 一般的な使用方法

Zypperの一般的な構文は次のとおりです。

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

ブラケットで囲まれたコンポーネントは必須ではありません。一般的なオプションおよびすべてのコマンドのリストについては、`zypper help`を参照してください。特定のコマンドのヘルプを参照するには、「`zypper help command`」と入力します。

Zypperを実行する最も簡単な方法は、その名前の後にコマンドを入力することです。たとえば、システムに必要なすべてのパッチを適用するには、次のコマンドを入力します。

```
zypper patch
```

さらに、グローバルオプションをコマンドの直前に入力することによって、1つ以上のグローバルオプションを選択することもできます。たとえば`--non-interactive`では、何も入力を求められることなく、コマンドを実行できます(自動的にデフォルトの解答が適用されます)。

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

特定のコマンドに固有のオプションを使用する場合は、コマンドの直後にそのオプションを入力します。たとえば、`--auto-agree-with-licenses`は、ライセンスの確認を求めることなく、システムで必要なすべてのパッチを適用します(自動的に受け入れられます)。

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

一部のコマンドでは、1つ以上の引数が必要です。たとえば、インストールコマンドを使用する場合、インストールするパッケージを指定する必要があります。

```
zypper install mplayer
```

また一部のオプションでは、引数が必要です。次のコマンドでは、すべての既知のパターンが表示されます。

```
zypper search -t pattern
```

上記のすべてを結合できます。たとえば、次のコマンドは、冗長モードで、`factory`リポジトリから`aspell-de`と`aspell-fr`パッケージをインストールします。

```
zypper -v install --from factory aspell-de aspell-fr
```

`--from`オプションは、指定されたリポジトリからパッケージを要求する際に、すべてのリポジトリを(依存関係の解決のため)有効に保ちます。

ほとんどのZypperコマンドには、指定のコマンドのシミュレーションを行う`dry-run`オプションがあります。このオプションは、テストの目的で使用できます。

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

Zypperは、グローバルオプション`--userdata string`をサポートします。このオプションを使用して文字列を指定することができます。指定した文字列は、Zypperのログファイルとプラグイン(Btrfsプラグインなど)に書き込まれます。これを使用して、ログファイルでトランザクションにマークを付けたり、トランザクションを特定したりできます。

```
zypper --userdata string patch
```

6.1.2 Zypperを使ったソフトウェアのインストールと削除

パッケージをインストールまたは削除するには、次のコマンドを使用します。

```
zypper install package_name
zypper remove package_name
```

Zypperでは、インストールコマンドおよび削除コマンドでパッケージを指定するために、次のようなさまざまな方法が可能です。

正確なパッケージ名(およびバージョン番号)を指定する方法

```
zypper install MozillaFirefox
```

または

```
zypper install MozillaFirefox-3.5.3
```

リポジトリエイリアスおよびパッケージ名を指定する方法

```
zypper install mozilla:MozillaFirefox
```

ここで `mozilla` は、インストールするリポジトリのエイリアスです。

ワイルドカードを使用してパッケージ名を指定する方法

次のコマンドでは、名前の先頭に「Moz」が付くすべてのパッケージがインストールされます。特にパッケージを削除する場合には、慎重に行うことが必要です。

```
zypper install 'Moz*'
```

機能によって指定する方法

たとえば、パッケージ名がわからないPerlモジュールをインストールする場合は、機能による指定が便利です。

```
zypper install firefox
```

機能、アーキテクチャ、またはバージョン、あるいはこれらすべてを指定する方法

機能とともに、アーキテクチャ(`x86_64` など)またはバージョン、あるいはその両方を指定できます。バージョンの前には、演算子として、`<` (未満)、`<=` (以下)、`=` (等しい)、`=>` (以上)、または `>` (より大きい)を付ける必要があります:

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

RPMファイルのパスで指定する方法

また、パッケージに対するローカルパスまたはリモートパスを指定できます。

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

パッケージのインストールおよび削除を同時に行うには、`+/-` 修飾子を使用します。`emacs` のインストールと `vim` の削除を同時に行うには、次のコマンドを使用します。

```
zypper install emacs -vim
```

`emacs` の削除と `vim` のインストールを同時に行うには、次のコマンドを使用します。

```
zypper remove emacs +vim
```

名前の先頭に `-` が付くパッケージ名がコマンドオプションとして解釈されないようにするには、常に第2引数としてその名前を使用します。これが可能でない場合は、名前の前に `--` を付けます。

```
zypper install -emacs +vim       # Wrong
zypper install vim -emacs        # Correct
zypper install -- -emacs +vim    # same as above
zypper remove emacs +vim         # same as above
```

指定したパッケージの削除後に、(その特定のパッケージとともに)不要になったパッケージを自動的に削除したい場合は、`--clean-deps` オプションを使用します。

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

Zypperではデフォルトで、選択したパッケージのインストールまたは削除の前に、あるいは問題が発生した際には、確認が求められます。この動作は、`--non-interactive` オプションを使用することで上書きされます。このオプションは、次のように、実際のコマンド(`install`、`remove`、`patch`)の前に指定する必要があります。

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

このオプションは、スクリプトおよびcronジョブでZypperを使用できます。

警告: 必須システムパッケージは削除しないでください。

`glibc`、`zypper`、`kernel` などのパッケージは削除しないでください。これらのパッケージはシステムで必須であり、削除するとシステムが不安定になったり、すべての動作が停止したりする場合があります。

6.1.2.1 ソースパッケージのインストールまたはダウンロード

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

```
zypper source-install package_name
```

このコマンドにより、指定したパッケージの構築依存もインストールされます。この処理が必要でない場合は、次のようにスイッチ `-D` を追加します。ビルドの依存関係のみをインストールするには、`-d` を使用します。

```
zypper source-install -D package_name # source package only
zypper source-install -d package_name # build dependencies only
```

もちろん、リポジトリリストで有効にしたソースパッケージを含むリポジトリが存在する場合にのみ動作します(ソースパッケージはデフォルトで追加されますが、有効にはなりません)。リポジトリの管理の詳細については、6.1.4項「Zypperによるリポジトリの管理」を参照してください。

リポジトリで使用可能なすべてのソースパッケージのリストは、次のコマンドで参照できます。

```
zypper search -t srcpackage
```

また、すべてのインストール済みパッケージのソースパッケージをローカルディレクトリにダウンロードすることもできます。ソースパッケージをダウンロードするには、以下を使用します。

```
zypper source-download
```

デフォルトのダウンロードディレクトリは `/var/cache/zypper/source-download` です。これは、`--directory` オプションを使って変更できます。ダウンロードや削除を行わず、不足パッケージや不要パッケージの表示のみを行う場合は、`--status` オプションを使用します。不要なソースパッケージを削除するには、`--delete` オプションを使用します。削除を無効にするには、`--no-delete` オプションを使用します。

6.1.2.2 ユーティリティ

すべての依存関係が依然として満たされていることを確認し、欠如する依存関係を修復するには、次のコマンドを使用します。

```
zypper verify
```

必要とされる依存関係に加えて、一部のパッケージでは他のパッケージが「推奨されます」。これらの推奨対象パッケージは、実際に使用可能でインストール可能な場合のみインストールされます。推奨側のパッケージがインストールされた後で、(パッケージまたはハードウェアの追加により)推奨対象パッケージが使用可能になった場合は、次のコマンドを使用します。

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

このコマンドは、WebcamまたはWi-Fiデバイスを接続した後で非常に役に立ちます。このコマンドは、デバイスのドライバと関連ソフトウェアが利用できる場合には、それらをインストールします。ドライバと関連ソフトウェアは、一定のハードウェア依存関係が満たされている場合のみインストールできます。

6.1.3 Zypperによるソフトウェアの更新

Zypperを使用してソフトウェアを更新するには3つの方法があります。パッチをインストールする、パッケージの新しいバージョンをインストールする、または配布全体を更新する方法です。最後の方法は、`zypper dist-upgrade`で行うことができます。SUSE Linux Enterprise Serverのアップグレードについては、ブック「導入ガイド」7「SUSE Linux Enterpriseのアップデート」を参照してください。

6.1.3.1 パッチのインストール

正式にリリースされたすべてのパッチをインストールしてシステムに適用するには、次のコマンドを実行します。

```
zypper patch
```

この場合、リポジトリで利用可能なすべてのパッチが関連性についてチェックされ、必要に応じてインストールされます。SUSE Linux Enterprise Serverを登録すると、このようなパッチを含む公式なアップデートリポジトリがシステムに追加されます。上記のコマンドを入力すれば、いつでも必要なときにこれらを適用できます。

Zypperでは、パッチの可用性について問い合わせるための3つの異なるコマンドが認識されます。

`zypper patch-check`

必要なパッチの数を示します(システムに適用されていてもまだインストールされていないパッチ)。

```
tux > sudo zypper patch-check
Loading repository data...
```

```
Reading installed packages...
5 patches needed (1 security patch)
```

`zypper list-patches`

必要なすべてのパッチを示します(システムに適用されていてもまだインストールされていないパッチ)。

```
tux > sudo zypper list-patches
Loading repository data...
Reading installed packages...

Repository     | Name        | Version | Category | Status  | Summary
---------------+-------------+---------+----------+---------+--------
SLES12-Updates | SUSE-2014-8 | 1       | security | needed  | openssl: Update to OpenSSL 1.0.1g
```

`zypper patches`

インストール済みかどうかや、インストール環境に適用済みかどうかに関係なく、SUSE Linux Enterprise Serverで使用可能なすべてのパッチを表示します。

また、特定の問題に関連するパッチを表示およびインストールすることもできます。特定のパッチを表示するには、次のオプションで`zypper list-patches`コマンドを使用します。

`--bugzilla[=number]`

Bugzilla発信番号で必要なすべてのパッチを表示します。オプションとして、この特定のバグのパッチを一覧するだけの場合は、バグ番号を指定できます。

`--cve[=番号]`

CVE (Common Vulnerabilities and Exposures)問題に関して必要なすべてのパッチ、または特定のCVE番号に一致するパッチだけ(番号を指定した場合)を一覧します。

特定のBugzillaまたはCVEの問題に対するパッチをインストールするには、次のコマンドを使用します。

```
zypper patch --bugzilla=number
```

または

```
zypper patch --cve=number
```

たとえば、CVE番号が CVE-2010-2713 のセキュリティ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。

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

6.1.3.2 新規パッケージバージョンのインストール

リポジトリに新しいパッケージのみが存在し、パッチが提供されていない場合は、`zypper patch` は無効です。インストールされているパッケージをすべて(システムの整合性を維持しながら)新しく入手可能なバージョンでアップデートするには、次を使用します。

```
zypper update
```

個別のパッケージをアップデートするには、updateコマンドまたはinstallコマンドのいずれかでパッケージを指定します。

```
zypper update package_name
zypper install package_name
```

インストール可能なすべての新しいパッケージのリストを、次のコマンドで取得できます。

```
zypper list-updates
```

ただし、このコマンドで表示されるのは、次の条件に一致するパッケージのみです。

- すでにインストール済みのパッケージと同じベンダである
- すでにインストール済みのパッケージと同等以上の優先度をもつリポジトリによって提供される
- インストール可能である(すべての依存関係が満たされている)

次のコマンドを使用すると、(インストール可能かどうかに関わらず)すべての新しい使用可能なパッケージのリストを取得できます。

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

新しいパッケージをインストールできない理由を見つけるには、上で説明されているように、`zypper install` コマンドまたは `zypper update` コマンドを使用します。

6.1.4 Zypperによるリポジトリの管理

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

```
zypper repos
```

結果は、次の出力のようになります。

例 6.1 ZYPPER—既知のリポジトリのリスト

```
# | Alias        | Name          | Enabled | Refresh
--+--------------+---------------+---------+--------
1 | SLEHA-12-GEO | SLEHA-12-GEO  | Yes     | No
2 | SLEHA-12     | SLEHA-12      | Yes     | No
3 | SLES12       | SLES12        | Yes     | No
```

各種コマンドのリポジトリを指定するには、エイリアス、URI、またはリポジトリ番号を`zypper repos`コマンド出力から使用できます。リポジトリの別名は、リポジトリ操作コマンド用の短いリポジトリ名です。ただし、リポジトリリストの変更後に、リポジトリ番号が変わる可能性があります。エイリアスは変更されることはありません。

デフォルトでは、URIやリポジトリの優先度など、詳細情報は表示されません。すべての詳細を表示するには、次のコマンドを使用します。

```
zypper repos -d
```

6.1.4.1 リポジトリの追加

リポジトリを追加するには、次を実行します。

```
zypper addrepo URI alias
```

URI は、インターネットリポジトリ、ネットワークリソース、ディレクトリ、CDまたはDVDのいずれかです(詳細については、http://en.opensuse.org/openSUSE:Libzypp_URIs を参照してください)。別名は、リポジトリの短い固有のIDです。このIDは、固有である必要があること以外は自由に選択できます。すでに使用されているエイリアスを指定した場合、Zypperでは警告が発行されます。

6.1.4.2 リポジトリの削除

リストからリポジトリを削除する場合は、コマンド zypper removerepo を使用し、削除するリポジトリのエイリアスまたは番号を指定します。たとえば、例6.1「Zypper—既知のリポジトリのリスト」から SLEHA-12-GEO リポジトリを削除するには、次のコマンドのいずれかを使用します。

```
zypper removerepo 1
zypper removerepo "SLEHA-12-GEO"
```

6.1.4.3 リポジトリの変更

zypper modifyrepo によりリポジトリを有効または無効にします。また、このコマンドにより、リポジトリのプロパティ(動作、名前、優先度の更新など)を変更できます。次のコマンドは、updates という名前のリポジトリを有効にし、自動更新をオンにし、リポジトリの優先度を 20に設定します。

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

リポジトリを変更する場合、1つのリポジトリだけなく、リポジトリのグループも操作できます。

-a : すべてのリポジトリ
-l : ローカルリポジトリ
-t : リモートリポジトリ
-m タイプ :特定のタイプのリポジトリ(ここで、タイプ には、次のいずれかを指定できます: http 、 https 、 ftp 、 cd 、 dvd 、 dir 、 file 、 cifs 、 smb 、 nfs 、 hd 、 iso)
リポジトリエイリアスの名前を変更するには、 renamerepo コマンドを使用します。次の例では、エイリアスを Mozilla Firefox から firefox に変更しています。

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

6.1.5 Zypperによるリポジトリおよびパッケージのクエリ

Zypperでは、リポジトリまたはパッケージをクエリするためのさまざまな方法が提供されています。使用可能なすべての製品、パターン、パッケージ、またはパッチのリストを取得するには、次のコマンドを使用します。

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

特定のパッケージについてすべてのリポジトリをクエリするには、`search`を使用します。searchは、パッケージの名前、またはパッケージの概要と説明(オプション)に関して機能します。`/`でラップされた文字列は、正規表現として解釈されます。デフォルトでは、検索で大文字と小文字は区別されません。

`fire`を含むパッケージ名の単純な検索

```
zypper search "fire"
```

正確なパッケージ`MozillaFirefox`の単純な検索

```
zypper search --match-exact "MozillaFirefox"
```

パッケージの説明とサマリも検索

```
zypper search -d fire
```

まだインストールしていないパッケージのみ表示

```
zypper search -u fire
```

文字列`fir`を含み、この後に`e`が続かないパッケージの表示

```
zypper se "/fir[^e]/"
```

特定の機能を提供するパッケージを検索するには、コマンド`what-provides`を使用します。たとえば、どのパッケージがPerlモジュール`SVN::Core`を提供するか確認したい場合は、次のコマンドを使用します。

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

単一のパッケージをクエリするには、`info`を使用し、引数として正確なパッケージ名を指定します。パッケージに関する詳細情報を表示します。パッケージの要求や推奨も表示するには、`--requires`オプションや`--recommends`オプションを使用します。

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

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

6.1.6 Zypperの設定

Zypperには、現在、設定ファイルが付属しています。この設定ファイルを使用すれば、Zypperの動作を(システム全体またはユーザ固有のでどちらかで)永続的に変更できます。システム全体に渡って変更する場合は、`/etc/zypp/zypper.conf`を編集します。ユーザ固有に変更する場合は、`~/.zypper.conf`を編集します。`~/.zypper.conf`がまだ存在していない場合は、テンプレートとして`/etc/zypp/zypper.conf`を使用できます。このテンプレートを`~/.zypper.conf`にコピーして、好みに合わせて調整してください。利用できるオプションのヘルプについては、ファイル内のコメントを参照してください。

6.1.7 トラブルシューティング

設定済みのリポジトリからのパッケージへのアクセスに問題がある場合(たとえば、一定のパッケージがリポジトリの1つに存在することを知っていても、Zypperでそのリポジトリを見つけられない場合など)は、次のコマンドでリポジトリを更新すると有効なことがあります。

```
zypper refresh
```

それも役に立たない場合は、次のコマンドを試してください。

```
zypper refresh -fdb
```

このコマンドは、生メタデータの強制ダウンロードを含むデータベースの完全な更新と再構築を強制します。

6.1.8 Btrfsファイルシステムでの Zypper ロールバック機能

ルートパーティションで Btrfs ファイルシステムが使用され、`snapper` がインストールされている場合に、ファイルシステムに対する変更をコミットして適切なファイルシステムスナップショットを作成すると、Zypper は(`snapper` によってインストールされるスクリプト経由で)自動的に `snapper` を呼び出します。これらのスナップショットは、Zypper によって行われた変更を元に戻す場合に使用できます。詳細については、第4章 Snapper を使用したシステムの回復とスナップショット管理を参照してください。

6.1.9 その他の情報

コマンドラインからのソフトウェア管理の詳細については、「`zypper help`」、「`zypper help command`」と入力するか、`zypper(8)` のマニュアルページを参照してください。最も重要なコマンドの**早見表**を含む詳しいコマンドリファレンス、およびスクリプトやアプリケーションにおける Zypper の詳しい使い方については、http://en.opensuse.org/SDB:Zypper_usage を参照してください。SUSE Linux Enterprise Server の最新バージョンにおけるソフトウェアの変更点のリストについては、http://en.opensuse.org/openSUSE:Zypper versions を参照してください。

6.2 RPM—パッケージマネージャ

RPM (RPM Package Manager)がソフトウェアパッケージを管理するのに使用されます。RPM の主要コマンドは、`rpm` と `rpmbuild` です。ユーザ、システム管理者、およびパッケージの作成者は、強力な RPM データベースでクエリーを行って、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。

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

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

ヒント: ソフトウェア開発パッケージ

多くのパッケージにおいて、ソフトウェア開発に必要なコンポーネント(ライブラリ、ヘッダ、インクルードファイルなど)は、別々のパッケージに入れられています。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、名前の拡張子`-deve`で識別できます(`alsa-devel`パッケージ、`gimp-devel`パッケージなど)。

6.2.1 パッケージの信頼性の検証

RPMパッケージにはGPG署名があります。RPMパッケージの署名を検証するには、`rpm --checksig package-1.2.3.rpm`コマンドを使用して、SUSEまたはその他の信頼できるツールから送信されたパッケージかどうか判別します。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。

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

通常RPMアーカイブのインストールはとても簡単です。`rpm -i package.rpm`の用に入力します。このコマンドで、パッケージをインストールできます。ただし、依存関係が満たされており、他のパッケージとの競合がない場合に限られます。`rpm`では、依存関係の要件を満たすためにインストールしなければならないパッケージがエラーメッセージで要求されます。バックグランドで、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。

`-U`または`--upgrade`と`-F`または`--freshen`の各オプションは、パッケージをアップデートするのに使用できます(たとえば、`rpm -F package.rpm`)。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`.」と入力します。解決されていない依存関係がない場合にパッケージのみを削除します。他のアプリケーションがTcl/Tkを必要とする限り、Tcl/Tkを削除することは理論的に不可能です。その場合でも、RPMはデータベースに援助を要求します。他の依存関係がない場合でも、また、どのような理由が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よいでしょう。

6.2.3 デルタRPMパッケージ

デルタRPMパッケージには、RPMパッケージの新旧バージョン間の違いが含まれています。デルタRPMを古いRPMに適用すると、まったく新しいRPMになります。デルタRPMは、インストールされているRPMとも互換性があるので、古いRPMのコピーを保管する必要はありません。デルタRPMパッケージは、パッチRPMよりもさらに小さく、パッケージをインターネット上で転送するのに便利です。欠点は、デルタRPMが組み込まれたアップデート操作の場合、そのままのRPMまたはパッチRPMに比べて、CPUサイクルの消費が目立って多くなることです。

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

```
makedeltarpm old.rpm new.rpm new.delta.rpm
```

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

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

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

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

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

6.2.4 RPMクエリー

`-q` オプションを使用すると、`rpm` はクエリを開始し、(`-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を行えるようにします。必要な情報の種類を指定する複数のスイッチを使用できます。詳細については、表6.1「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。

表 6.1 最も重要なRPMクエリーのオプション

`-i`	パッケージ情報
`-l`	ファイルリスト
`-f FILE`	ファイル `FILE` を含むパッケージでクエリーを行います(`FILE` にはフルパスを指定する必要があります)。
`-s`	ステータス情報を含むファイルリスト(`-l` を暗示指定)
`-d`	ドキュメントファイルだけをリストします (`-l` を暗示指定)。
`-c`	設定ファイルだけをリストします(`-l` を暗示指定)。
`--dump`	詳細情報を含むファイルリスト(`-l`、`-c`、または `-d` と共に使用します)
`--provides`	他のパッケージが `--requires` で要求できるパッケージの機能をリストします。
`--requires`, `-R`	パッケージが要求する機能
`--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6.2「rpm -q -i wget」に示された情報を表示します。

例 6.2 RPM -Q -I WGET

```
Name        : wget                         Relocations: (not relocatable)
```

```
Version     : 1.11.4                              Vendor: openSUSE
Release     : 1.70                             Build Date: Sat 01 Aug 2009 09:49:48
 CEST
Install Date: Thu 06 Aug 2009 14:53:24 CEST      Build Host: build18
Group       : Productivity/Networking/Web/Utilities   Source RPM:
 wget-1.11.4-1.70.src.rpm
Size        : 1525431                            License: GPL v3 or later
Signature   : RSA/8, Sat 01 Aug 2009 09:50:04 CEST, Key ID b88b2fd43dbdc284
Packager    : http://bugs.opensuse.org
URL         : http://www.gnu.org/software/wget/
Summary     : A Tool for Mirroring FTP and HTTP Servers
Description :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

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

ファイル名の一部分しかわからない場合は、例6.3「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。

例 6.3 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
```

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

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`または`--verify`オプションを使用して開始します。このオプションを使うと、`rpm`は、パッケージ内にあり、インストール以降変更されたことがあるすべてのファイルを表示します。`rpm`は、次の変更に関するヒントを表示するのに、8文字の記号を使用します。

表 6.2 RPM確認オプション

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

設定ファイルの場合は、文字`c`が表示されます。`/etc/wgetrc`（`wget`パッケージ)の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。 コピー数は /etc/sysconfig/backup にある変数 MAX_RPMDB_BACKUPS で制御します(デフォルト: 5)。1つのバックアップのサイズは、1GBの /usr に対しておよそ1MBです。

6.2.5 ソースパッケージのインストールとコンパイル

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

注記: インストール済みのソースパッケージ

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

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

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

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

BUILD
: すべてのソースは、このディレクトリでアンパック、パッチ、および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。

警告: システムの整合性

システムコンポーネント(`glibc` 、 `rpm` など)で実験を行わないでください。システムが正しく動作しなくなります。

次の例は、 `wget.src.rpm` パッケージを使用します。ソースパッケージをインストールすると、次のようなファイルが生成されます。

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

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

`-bp`

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

`-bc`

`-bp` と同じですが、コンパイルを実行します。

`-bi`

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

`-bb`

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

`-ba`

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

`--short-circuit`

一部のステップをスキップします。

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

6.2.6 buildによるRPMパッケージのコンパイル

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

```
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。

6.2.7 RPMアーカイブとRPMデータベース用のツール

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

フル機能のパッケージマネージャをYaSTモジュールとして使用できます詳細については、ブック 「導入ガイド」 9 「ソフトウェアをインストールまたは削除する」を参照してください。

7 BashとBashスクリプト

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

7.1 「シェル」とは何か?

従来、シェルとは、Bash(Bourne again Shell)のことでした。この章では、Bashを「シェル」と呼びます。実際にはシェルはBashの他にもあり(ash、csh、ksh、zshなど)、異なる機能と特性を持っています。他のシェルの詳細については、YaSTで「シェル」を検索してください。

7.1.1 Bash設定ファイルの知識

シェルは、次のようにして呼び出すことができます。

1. **対話型ログインシェル.** コンピュータへのログイン時に、`--login`オプションを使用してBashを呼び出す場合か、SSHを使用してリモートコンピュータへログインする場合に使用します。
2. **「通常の」対話型シェル.** xtermやkonsole、gnome-terminalなどのツールの起動時には、通常、この形式を使用します。
3. **非対話型シェル.** コマンドラインからシェルスクリプトを呼び出す場合に使用します。

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

表 7.1 ログインシェル用BASH設定ファイル

ファイル	説明
`/etc/profile`	このファイルは変更しないでください。変更しても、次の更新で変更内容が破棄される可能性があります。
`/etc/profile.local`	`/etc/profile`を拡張する場合は、このファイルを使用します。

ファイル	説明
/etc/profile.d/	特定プログラムのシステム全体に渡る設定ファイルを含みます。
~/.profile	ログインシェル用のユーザ固有の設定をここに挿入します。

表 7.2 非ログインシェル用BASH設定ファイル

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

さらに、Bashでは、次のファイルも使用します。

表 7.3 BASH用特殊ファイル

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

7.1.2 ディレクトリの構造

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

ディレクトリ	目次
`/`	ルートディレクトリ(ディレクトリツリーの開始場所)。
`/bin`	システム管理者および通常ユーザの両者が必要とするコマンドなどの必須バイナリファイル。通常、Bashなどのシェルも含みます。
`/boot`	ブートローダの静的ファイル
`/dev`	ホスト固有のデバイスのアクセスに必要なファイル
`/etc`	ホスト固有のシステム設定ファイル
`/home`	システムにアカウントを持つすべてのユーザのホームディレクトリを格納します。ただし、`root`のホームディレクトリは、`/home`でなく、`/root`にあります。
`/lib`	必須の共有ライブラリおよびカーネルモジュール
`/media`	リムーバブルメディアのマウントポイント
`/mnt`	ファイルシステムを一時的にマウントするためのマウントポイント
`/opt`	アドオンアプリケーションのソフトウェアパッケージ
`/root`	スーパーユーザ`root`のホームディレクトリ。
`/sbin`	必須のシステムバイナリ
`/srv`	システムで提供するサービスのデータ
`/tmp`	一時ファイルを格納するディレクトリ
`/usr`	読み込み専用データを含む第二階層
`/var`	ログファイルなどの可変データ
`/windows`	システムにMicrosoft Windows*とLinuxの両方がインストールされる場合のみ利用可能。Windowsデータを含みます。

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

`/bin`

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

`/boot`

ブートに必要なデータ(ブートローダやカーネルのデータなど)と、その他のデータ(カーネルがユーザモードプログラムの実行を開始する前に使用)が含まれます。

`/dev`

ハードウェアコンポーネントを記述したデバイスファイルを格納します。

`/etc`

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

`/home/username`

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

注記: ネットワーク環境でのホームディレクトリ

ネットワーク環境で作業するユーザのホームディレクトリは、`/home`以外のファイルシステム内のディレクトリにマップできます。

`/lib`

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

`/media`

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

`/mnt`

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

`/opt`

サードパーティのソフトウェアのインストール用に予約されています。オプションソフトウェアや大型アドオンプログラムのパッケージをここに格納できます。

`/root`

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

`/run`

`systemd` とさまざまなコンポーネントによって使用されるtmpfsディレクトリ。

`/sbin`

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

`/srv`

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

`/tmp`

ファイルの一時的保管を必要とするプログラムによって使用されます。

重要: ブート時の/tmpのクリーンアップ

`/tmp` に保存したデータは、システムのリブート後も残っているかは保証できません。これは、たとえば、`/etc/sysconfig/cron` 内の設定によって左右されます。

`/usr`

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

`/usr/bin`

一般ユーザがアクセスできるプログラムを含みます。

`/usr/sbin`

修復関数など、システム管理者用に予約されたプログラムを含みます。

`/usr/local`

このディレクトリには、システム管理者がディストリビューションに依存しないローカルな拡張プログラムをインストールできます。

`/usr/share/doc`

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

`packages` には、システムにインストールされたソフトウェアパッケージに含まれているドキュメントが格納されます。パッケージごとに、サブディレクトリ `/usr/share/doc/packages/packagename` が作成されます。このサブディレクトリには、多くの場合、パッケージのREADMEファイルが含まれます。例、設定ファイル、または追加スクリプトが含まれる場合もあります。

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

`/var`

`/usr` は静的な読み込み専用データを含みますが、`/var` は、システム動作時に書き込まれる可変データ(ログファイル、スプールデータなど)のディレクトリです。`/var/log/` にある重要なログファイルの概要は、表36.1「ログファイル」を参照してください。

7.2 シェルスクリプトの作成

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

例 7.1 テキストをプリントするシェルスクリプト

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

❶ 1行目は、このファイルがスクリプトであることを示すShebang文字(`#!`)で始まります。スクリプトは、Shebang文字の後に指定されたインタープリタ(ここでは、`/bin/sh`)を使用して実行されます。

❷ 2行目は、ハッシュ記号で始まるコメントです。スクリプトの動作を覚えにくい行には、コメントすることをお勧めします。

❸ 3番目の行で、組み込みコマンド`echo`を使用して、対応するテキストを出力します。

このスクリプトの実行には、次の前提条件が必要です。

1. 各スクリプトは、Shebang行を含む必要があります(この例はすでに示しました)。スクリプトにこの行がない場合は、手動でインタープリタを呼び出す必要があります。
2. スクリプトの保存場所はどこでも構いません。ただし、シェルの検索先ディレクトリを保存場所にすることをお勧めします。シェルのサーチパスは、環境変数`PATH`で設定されます。一般に、標準ユーザには`/usr/bin`への書き込みアクセスはありません。このため、スクリプトはユーザのディレクトリ`~/bin/`に保存することを推奨します。上記の例では、名前は`hello.sh`です。
3. スクリプトには、実行可能パーミッションが必要です。次のコマンドで、パーミッションを設定してください。

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

これらの前提条件をすべて満たしたら、次の方法でスクリプトを実行できます。

1. **絶対パス.** スクリプトは絶対パスで実行できます。この例では、`~/bin/hello.sh`です。
2. **任意の場所.** `PATH`環境変数にスクリプトが存在するディレクトリが含まれている場合、スクリプトを`hello.sh`で実行できます。

7.3 コマンドイベントのリダイレクト

各コマンドは、入力または出力用として、3つのチャネルを使用できます。:

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

これらのチャネルをリダイレクトするには、次の方法を使用できます。

`Command > File`

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

```
ls > listing.txt
```

`Command >> File`

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

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

`Command < File`

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

```
read a < foo
```

`Command1 | Command2`

左側のコマンドの出力を右側のコマンドの入力にします。たとえば、`cat`コマンドは`/proc/cpuinfo`ファイルの内容を出力します。この出力を`grep`で使用して、`cpu`を含む行のみをフィルタします。

```
cat /proc/cpuinfo | grep cpu
```

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

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

7.4 エイリアスの使用

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

```
alias NAME=DEFINITION
```

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

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

すべてのエイリアス定義を表示するには、`alias` を使用します。`unalias` で対応するエイリアス名を指定して、エイリアスを削除します。

7.5 Bashでの変数の使用

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

すべての環境変数を表示するには、`printenv` コマンドを使用します。変数の値を知る必要がある場合は、変数の名前を引数として挿入します。

```
printenv PATH
```

変数はグローバルでもローカルでも、`echo` で表示できます。

```
echo $PATH
```

ローカル変数を設定するには、変数名の後に等号を入れ、その後に値を指定します。

```
PROJECT="SLED"
```

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

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

変数を削除するには、`unset` を使用します。

```
unset NAME
```

次のテーブルに、シェルスクリプトで使用できる共通環境変数を示します。

表 7.5 便利な環境変数

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

7.5.1 引数変数の使用

たとえば、スクリプトfoo.shは、次のように実行できます。

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

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

次のスクリプトfoo.shは、1から4までのすべての引数をプリントします。

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

このスクリプトを既出例の引数を使用して実行すると、次の結果が出力されます。

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

7.5.2 変数置換の使用

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

`${VAR#pattern}`

左側から最も短い一致を削除します。

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

`${VAR##pattern}`

左側から最も長い一致を削除します。

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

`${VAR%pattern}`

右側から最も短い一致を削除します。

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

`${VAR%%pattern}`

右側から最も長い一致を削除します。

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

`${VAR/pattern_1/pattern_2}`

`VAR` のコンテンツを `pattern_1` から `pattern_2` に置換します。

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

7.6 コマンドのグループ化と結合

シェルでは、条件付き実行のため、コマンドを結合し、グループ化することができます。各コマンドが返す終了コードにより、コマンドの成功または失敗が判別されます。終了コードが0(ゼロ)の場合、コマンドは成功しました。それ以外はすべて、コマンド固有のエラーをマークします。

次のリストでは、コマンドをグループ化する方法を一覧します。

`Command1 ; Command2`

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

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

`Command1 && Command2`

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

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

`Command1 || Command2`

左のコマンドが失敗した場合、右のコマンドを実行します(論理OR)次の行では、`/home/tux/foo` でのディレクトリ作成に失敗した場合のみ、`/home/wilber/bar` 内にディレクトリを作成します。

```
mkdir /home/tux/foo || mkdir /home/wilber/bar
```

`funcname(){ ... }`

シェル関数を作成します。位置パラメータを使用して、関数の引数にアクセスできます。次の行では、短いメッセージをプリントする関数 `hello` を定義します。

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

この関数は、次のように呼び出せます。

```
hello Tux
```

結果は、次のようにプリントされます。

```
Hello Tux
```

7.7 よく使用されるフローコンストラクトの操作

スクリプトのフローを制御するため、シェルでは、`while`、`if`、`for`、および`case`の各構文を使用します。

7.7.1 if制御コマンド

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

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

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

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

test式は、角括弧で短縮することもできます。

```
if [ -e /tmp/foo.txt ] ; then
  echo "Found foo.txt"
fi
```

その他の役に立つ式については、http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html を参照してください。

7.7.2 Forコマンドによるループの作成

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

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

7.8 詳細情報

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

- http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html—Bash Guide for Beginners
- http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html—BASH Programming - Introduction HOW-TO
- http://tldp.org/LDP/abs/html/index.html—Advanced Bash-Scripting Guide
- http://www.grymoire.com/Unix/Sh.html—Sh - the Bourne Shell

II システム

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

64ビットプラットフォームであるPOWER、System z、およびx86_64に対応したSUSE Linux Enterprise Serverは、「出荷してすぐに」既存の32ビットアプリケーションが64ビット環境で動作するように設計されています。対応する32ビットプラットフォームには、ppc64のppcおよびx86_64のx86があります。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。現在のppc64システムは、大部分のアプリケーションを 32ビットモードで実行しますが、64ビットアプリケーションを実行することもできます。

8.1 ランタイムサポート

重要: 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。

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

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

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

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

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

8.2 ソフトウェア開発

すべての64ビットアーキテクチャで、64ビットオブジェクトの開発がサポートされています。32ビットコンパイル機能のサポートレベルは、アーキテクチャによって異なります。32ビットコンパイル機能は、GCC (GNU Compiler Collection)やbinutilsによるツールチェーンの各種実装オプションになっています。Binutilsには、アセンブラ `as` とリンカー `ld` が含まれています。

biarchコンパイラ

32ビットと64ビットのオブジェクトはどちらもbiarch開発ツールチェーンで生成できます。biarch開発ツールチェーンを使用して、32ビットと64ビットのオブジェクトを生成できます。ほぼすべてのプラットフォームにおいて、デフォルトでは64ビットオブジェクトのコンパイルが実行されます。32ビットオブジェクトは、特殊なフラグを使用すれば生成できます。この特殊なフラグは、GCCでは `-32` です。binutilsのフラグはアーキテクチャによって異なりますが、GCCは正しいフラグをリンカーやアセンブラに転送します。現在では、amd64 (x86とamd64の命令の開発をサポート)、System z、およびppc64用のbiarch開発ツールチェーンが存在します。通常、32ビットオブジェクトはppc64プラットフォームで作成されます。-m64 フラグは、64ビットオブジェクトの生成に使用する必要があります。

未サポート

SUSE Linux Enterprise Serverでは、すべてのプラットフォームで32ビットソフトウェアを直接開発できるとは限りません。ia64でx86用のアプリケーションを開発するには、対応する32ビットバージョンのSUSE Linux Enterprise Serverを使用します。

すべてのヘッダファイルは、アーキテクチャに依存しない形式で作成する必要があります。インストール済みの32ビットと64ビットのライブラリには、インストール済みのヘッダファイルに対応するAPI (アプリケーションプログラミングインタフェース)が必要です。標準のSUSE Linux Enterprise Server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。

8.3 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、対象のアーキテクチャが-32ビットアーキテクチャである場合は `rpmname- 32bit` または `rpmname-x86` (ia64の場合)と呼ばれ、対象のアーキテクチャが-64ビットアーキテクチャである場合は `rpmname- 64bit` と呼ばれます。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` または `rpmname-devel-64bit` 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。

たとえば、対象のアーキテクチャが32ビットアーキテクチャ(x86_64またはSystem z)であるシステムで `libaio` を使用するプログラムをコンパイルするには、次のRPMが必要です。

libaio-32bit
: 32ビットランタイムパッケージ

libaio-devel-32bit
: 32ビット開発用のヘッダとライブラリ

libaio
: 64ビットランタイムパッケージ

libaio-devel
: 64ビット開発用のヘッダとライブラリ

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

次の例は、対象のアーキテクチャとしてx86を採用しているx86_64システムを示しています。対象のアーキテクチャとしてppcを採用しているppc64の場合も同様です。この例は、32ビットパッケージをビルドできないia64には適用されません。

1. 32ビットコンパイラを使用します。

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

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

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

3. 32ビットオブジェクトを生成するためにアセンブラを設定します。

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

4. 次に示すような、32ビットライブラリの場所などのリンカフラグを指定します。

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

5. 32ビットオブジェクトコードライブラリの場所を指定します。

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

6. 32ビットXライブラリの場所を指定します。

```
--x-libraries=/usr/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。

x86__64、ppc64、またはSystem　zでネイティブの32ビットアプリケーションをコンパイルする場合の、configureコールの例を次に示します。

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

8.4 カーネル仕様

x86_64、ppc_64およびSystem z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すべてのAPIをサポートしていません。ただし、このサポートの有無はプラットフォームによって異なります。このため、`lspci`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非ppc 64プラットフォームでコンパイルする必要があります。IBM System zでは、32ビットカーネルABIで利用できないioctls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。

ヒント: カーネルロード可能モジュール

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。

9 Linuxシステムのブート

Linuxシステムのブートには、さまざまなコンポーネントとタスクが関係しています。ハードウェア自体がBIOSまたはUEFIにより初期化され、BIOSまたはUEFIがブートローダでカーネルを起動します。この時点以降、ブートプロセスは完全にオペレーティングシステムの制御下に入り、`systemd`によって処理されます。`systemd`は、日常的な使用、保守、または緊急時のためにセットアップをブートする一連の「ターゲット」を提供します。

9.1 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、それぞれ別のコンポーネントが実行しています。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/UEFI.** コンピュータの電源をオンにした後、BIOSまたはUEFIが画面とキーボードを初期化し、メインメモリをテストします。この段階まで、コンピュータは大容量ストレージメディアにアクセスしません。続いて、現在の日付、時刻、および最も重要な周辺機器に関する情報が、CMOS値からロードされます。最初のハードディスクとそのジオメトリが認識されると、システム制御がBIOSからブートローダに移ります。BIOSがネットワークブートをサポートしている場合は、ブートローダを提供するブートサーバを設定することもできます。x86_64システムの場合、PXEブートを利用する必要があります。他のアーキテクチャの場合は、通常、BOOTPプロトコルを使ってブートローダを取得します。

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 2 (Linuxのブートローダ)の詳細については、第12章 ブートローダGRUB 2を参照してください。ネットワークブートを行う場合、BIOSがブートローダとしての役割を果たします。BIOSは、ブートサーバからブートイメージを取得し、システムを起動します。この作業はローカルハードディスクから完全に独立した処理として行われます。

3. **カーネルと`initramfs`.** システムに制御を渡すために、ブートローダは、カーネルとRAMベースの初期ファイルシステム(`initramfs`)をメモリにロードします。カーネルは、`initramfs`のコンテンツを直接使用できます。`initramfs`には、実際のルートファイルシステムのマウント処理を行う`init`と呼ばれる小さな実行可能ファイルが含まれています。大容量ストレージにアクセスする

ために特別なハードウェアドライバが必要な場合、それらは initramfs 内になければなりません。initramfs の詳細については、9.2項「initramfs」を参照してください。システムにローカルハードディスクがない場合、initramfs がルートファイルシステムをカーネルに提供する必要があります。そのためには、iSCSIやSANなどのネットワークブロックデバイスを利用しますが、NFSをルートデバイスとして使うことも可能です。

注記: initプロセスの名前付け

一般的に「init」という名前が付くのは、次の2つの異なるプログラムです。「」

a. ルートファイルシステムをマウントする initramfs プロセス

b. システムを設定するオペレーティングシステムプロセス

そのため、この章では、それぞれを「initramfs 上の init 」および「systemd 」と呼びます。

4. initramfs 上の init . このプログラムは、適切なルートファイルシステムをマウントするために必要なすべてのアクションを実行します。必要なファイルシステムにカーネル機能を提供し、大容量ストレージコントローラ用のデバイスドライバに udev を提供します。ルートファイルシステムが見つかると、エラーをチェックしてからマウントします。これが正常に実行されれば、initramfs はクリアされ、ルートファイルシステムで systemd デーモンが実行されます。initramfs 上の init の詳細については、9.3項 「initramfs上のinit」を参照してください。udev の詳細については、第16章 udevによる動的カーネルデバイス管理を参照してください。

5. systemd . systemd は、サービスを起動し、ファイルシステムをマウントすることで、システムの実際のブートを処理します。systemd は第10章 systemdデーモンで説明されています。

9.2 initramfs

initramfs は、カーネルがRAMディスクにロードできる、小さなcpioアーカイブです。また、実際のルートファイルシステムがマウントされる前にプログラムを実行できるようにする最低限のLinux環境を提供します。この最低限のLinux環境は、BIOSまたはUEFIルーチンによってメモリにロードされ、十分なメモリがあること以外に特定のハードウェア要件はありません。initramfs には必ず、init という名前の実行可能ファイルがあります。これは、ブートプロセスの進行に伴い、ルートファイルシステム上の実際の systemd デーモンを実行します。

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

インストール済みのシステムのハードウェア(たとえば、ハードディスク)を変更する必要が生じ、このハードウェアはブート時にカーネル内に存在する他のドライバを必要とする場合には、`initramfs` ファイルを更新する必要があります。このためには、`dracut -f` を呼び出します(オプション `-f` は既存のinitramfsファイルを上書きします)。新しいハードウェア用のドライバを追加するには、`/etc/dracut.conf.d/01-dist.conf` を編集して次の行を追加します。

```
force_drivers+="driver1"
```

`driver1` はドライバのモジュール名で置き換えます。複数のドライバを追加する必要がある場合は、それぞれをスペースで区切ったリスト形式で記述します(`driver1 driver2`)。

重要: initramfsまたは initの更新

ブートローダは、カーネルと同じように `initramfs` または `init` をロードします。GRUB 2はブート時に正しいファイルのディレクトリを検索するので、`initramfs` または `init` を更新した後にGRUB 2を再インストールする必要はありません。

9.3 initramfs上のinit

`initramfs` 上の `init` の主な目的は、実際のルートファイルシステムのマウントとアクセスの準備をすることです。システム設定に応じて、`initramfs` 上の `init` は次のタスクを実行します。

カーネルモジュールのロード

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

ブロック特殊ファイルの提供

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

RAIDとLVMのセットアップの管理

RAIDまたはLVMの下でルートファイルシステムを保持するようにシステムを設定した場合、initramfs 上の init はLVMまたはRAIDを設定して、後でルートファイルシステムにアクセスできるようにします。ブック「導入ガイド」15「高度なディスクセットアップ」でRAIDとLVMに関する情報を参照してください。

ネットワーク設定の管理

ネットワークマウントしたルートファイルシステム(NFSを介してマウント)を使用するようにシステムを設定した場合、initramfs 上の init 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。

ファイルシステムがiSCSIやSANなどのネットワークブロックデバイスに常駐している場合は、ストレージサーバへの接続も initramfs 上の init によって設定されます。

初期ブート時にインストールプロセスの一環として initramfs 上の init が呼び出される場合、そのタスクは上記で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、マシンは、インストールカーネルと、YaSTインストーラを含む特殊な init をロードします。RAMファイルシステムで実行されるYaSTインストーラには、インストールメディアにアクセスしてオペレーティングシステムをインストールするために、そのメディアの場所に関する情報が必要になります。

ハードウェア認識の開始および適切なカーネルモジュールのロード

で説明しているように、ブートプロセスは、ほとんどのハードウェア設定で使用できる最小限のドライバセットで開始されます。initは、ハードウェア設定に適したドライバセットを確定する、初期ハードウェアスキャンプロセスを開始します。9.2項 「initramfs」 これらのドライバは、システムをブートするために必要なカスタム initramfs を生成するために使用されます。ブートに必要なくてもコールドプラグには必要なモジュールがある場合は、systemd を使用してロードできます。詳細については、10.6.3項「カーネルモジュールのロード」を参照してください。

インストールシステムのロード

ハードウェアが適切に認識されるとすぐに、適切なドライバがロードされます。udev プログラムが特殊なデバイスファイルを作成し、init は、YaSTインストーラを使用してインストールシステムを起動します。

YaSTの起動

最後に、`init`はYaSTを起動し、これによって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。

10 systemdデーモン

プログラム `systemd` は、プロセスID 1のプロセスであり、要求された方法でシステムを初期化します。`systemd` 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、systemd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。

SUSE Linux Enterprise Server 12から、systemdが一般的なSystem V initデーモンに取って代わりました。`systemd` は、System V initと完全な互換性があります(initスクリプトをサポートしているため)。systemdの主な利点の1つは、サービスを積極的に並行起動することで、ブート時間をかなり速くできることです。systemdは、サービスを必要なときだけ起動します。デーモンは、ブート時に無条件で起動されることはなく、最初に必要になったときに起動されます。systemdでは、カーネルのコントロールグループ(cgroup)もサポートしているほか、システムの状態をスナップショットに保存したり、その状態に復元したりすることもできます。詳細については、http://www.freedesktop.org/wiki/Software/systemd/ を参照してください。

10.1 systemdの概念

このセクションでは、systemdの背景にある概念について詳しく説明します。

10.1.1 systemdについて

systemdは、System VおよびLSBのinitスクリプトと互換性のある、Linux向けのシステム/セッションマネージャです。主な特長は次のとおりです。

- 積極的な並行機能の提供
- ソケットやD-Busアクティベーションを使用したサービスの起動
- デーモンのオンデマンド起動
- Linux cgroupsを使用したプロセスの追跡
- システム状態のスナップショットへの保存、およびその状態への復元
- マウントポイントと自動マウントポイントの保持
- 精巧なトランザクションの依存関係に基づくサービス制御ロジックの実装

10.1.2 ユニットファイル

ユニット設定ファイルには、サービス、ソケット、デバイス、マウントポイント、自動マウントポイント、スワップファイルやパーティション、起動ターゲット、監視対象のファイルシステムのパス、systemdによって制御および監視されているタイマ、一時的なシステム状態のスナップショット、リソース管理スライス、または外部で作成されたプロセスグループに関する情報がエンコーディングされます。「ユニットファイル」は、systemdの次のファイルの総称です。

- **サービス.** プロセスに関する情報(たとえば、実行中のデーモン)。サービスファイルは.serviceで終わります。
- **ターゲット.** システム起動時のユニットのグループ化に、または同期ポイントとして使用されます。ターゲットファイルは.targetで終わります。
- **ソケット.** ソケットに基づくアクティベーション(`inetd`など)でのIPC、ネットワークソケット、ファイルシステムFIFOに関する情報。ソケットファイルは.socketで終わります。
- **パス.** その他のユニットをトリガするために使用されます(たとえば、ファイル変更時のサービスの実行など)。パスファイルは.pathで終わります。
- **タイマ.** タイマ制御された、タイマに基づくアクティベーションに関する情報。タイマファイルは.timerで終わります。
- **マウントポイント.** 通常はfstabジェネレータによって自動生成されます。マウントポイントファイルは.mountで終わります。
- **自動マウントポイント.** ファイルシステムの自動マウントポイントに関する情報。自動マウントポイントファイルは.automountで終わります。
- **スワップ.** スワップデバイスに関する情報またはメモリページング用のファイル。スワップファイルは.swapで終わります。
- **デバイス.** sysfs/udev(7)デバイスツリーに公開されているデバイスユニットに関する情報。デバイスファイルは.deviceで終わります。
- **スコープ/スライス.** プロセスグループのリソースを階層管理する際の概念。スコープ/スライスファイルは.scope/.sliceで終わります。

systemd.unitの詳細については、http://www.freedesktop.org/software/systemd/man/systemd.unit.html を参照してください。

10.2 基本的な使用方法

System V initシステムでは、initスクリプト、`insserv`、`telinit`などの複数の異なるコマンドを使用してサービスを処理します。systemdでは、サービスに対する主な処理を実行する際、1 つのコマンド(`systemctl`)で済むようになっているため、サービスをより容易に管理できます。`git`や`zypper`のように、「コマンドの後ろにサブコマンド」を指定して実行します。

```
systemctl [general OPTIONS] subcommand [subcommand OPTIONS]
```

完全なマニュアルについては、`man 1 systemctl`を参照してください。

ヒント: 端末の出力とbashの補完

systemdのコマンドは、出力先が端末である場合(パイプやファイルなどではない場合)、デフォルトではページャに長い出力が送信されます。ページングモードをオフにするには、`--no-pager`オプションを使用してください。

systemdでは、bashによる補完もサポートしています。サブコマンドの最初の1文字を入力し、`<Tab>`を押すと、サブコマンドの残りを自動的に入力することができます。この機能は、`bash`シェルを利用している場合にのみ使用できるもので、`bash-completion`パッケージをインストールしておく必要があります。

10.2.1 稼働中のシステムでのサービスの管理

サービスを管理するためのサブコマンドは、System V initでのサービス管理コマンドと同じ(`start`、`stop`など)です。サービス管理コマンドの基本構文は、以下のとおりです。

systemd

```
systemctl reload|restart|start|status|stop|... <my_service(s)>.service
```

System V init

```
rc<my_service(s)> reload|restart|start|status|stop|...
```

systemdでは、複数のサービスを一括で管理できます。initスクリプトを次々と実行しなければならないSystem V initとは異なり、次のようにコマンドを実行します。

```
systemctl start <my_1st_service>.service <my_2nd_service>.service
```

システムで利用できるすべてのサービスを一覧表示するには、次のように実行します。

```
systemctl list-unit-files --type=service
```

次の表に、systemdとSystem V initの最も重要なサービス管理コマンドを示します。

表 10.1 サービス管理コマンド

タスク	systemdコマンド	System V initコマンド
起動.	`start`	`start`
停止.	`stop`	`stop`
再起動. サービスを停止し、後で起動します。サービスがまだ起動していない場合は、そのサービスを起動します。	`restart`	`restart`
条件付きの再起動. サービスが現在実行中の場合、サービスを再起動します。実行されていないサービスについては、何も行いません。	`try-restart`	`try-restart`
再ロード. サービスに対し、操作を中断せずに設定ファイルを再ロードするように指示します。Apacheに、変更後の `httpd.conf` 設定ファイルを再ロードさせる、などの使用方法をします。すべてのサービスが再ロードをサポートしているとは限らないことに注意してください。	`reload`	`reload`
再ロードまたは再起動. サービスが再ロードをサポートしていれば再ロードし、サポートしていなければ再起動します。サービスがまだ起動していない場合は、そのサービスを起動します。	`reload-or-restart`	`n/a`
条件付きの再ロードまたは再起動. サービスが再ロードをサポートしていれば再ロードし、サポートしていなければ再起動します(現在実行中の場合)。実行されていないサービスについては、何も行いません。	`reload-or-try-restart`	`n/a`

タスク	systemdコマンド	System V initコマンド
詳細なステータス情報の取得. サービスのステータスについて、情報を表示します。`systemd`コマンドでは、説明、実行ファイル、ステータス、cgroupのほか、直近でサービスが出力したメッセージ(10.6.6項「サービスのデバッグ」を参照)が表示されます。System V initで表示される詳細のレベルは、サービスごとに異なります。	`status`	`status`
簡潔なステータス情報の取得. サービスがアクティブかどうかを示します。	`is-active`	`status`

10.2.2 サービスの恒久的な有効化/無効化

上述のサービス管理コマンドでは、現在のセッションに対するサービスを操作できます。systemdでは、サービスを恒久的に有効化/無効化して、必要に応じて自動的に起動したり、常に使用不可にすることもできます。この作業は、YaSTまたはコマンドラインを使用して実行できます。

10.2.2.1 コマンドラインからのサービスの有効化/無効化

次の表に、systemdとSystem V initの有効化/無効化コマンドを示します。

重要: サービスの起動について

コマンドラインからサービスを有効化した場合、そのサービスは自動的には起動されず、次回のシステム起動またはランレベル/ターゲット変更の際に起動されます。有効化した後で、即時にサービスを起動するには、`systemctl start <my_service>.service`または`rc<my_service> start`のように、明示的にサービスを起動してください。

表 10.2 サービスの有効化/無効化コマンド

作業	`systemd`コマンド	System V initコマンド
有効化.	`systemctl enable <my_service(s)>.service`	`insserv <my_service(s)>`

作業	`systemd` コマンド	System V initコマンド
無効化.	`systemctl disable <my_service(s)>.service`	`insserv -r <my_service(s)>`
確認. サービスが有効になっているかどうかを示します。	`systemctl is-enabled <my_service>.service`	該当なし
再有効化. サービスの再起動と同様に、このコマンドはいったんサービスを無効化した後に有効化します。サービスにデフォルト値を設定して再有効化する場合に利用します。	`systemctl reenable <my_service>.service`	該当なし
マスク. サービスを「無効化」しても、手動で起動できてしまいます。サービスを完全に無効化するには、マスクを設定する必要があります。注意してご使用ください。	`systemctl mask <my_service>.service`	該当なし
マスク解除. マスクを設定したサービスは、マスクを解除しないと使用できません。	`systemctl unmask <my_service>.service`	該当なし

10.3 システムの起動とターゲットの管理

システムを起動し、シャットダウンするプロセス全体は、systemdによって管理されます。この見地から、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を保持し、CPU時間とハードウェアアクセスを調整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。

10.3.1 ターゲットとランレベル

System V initでは、システムは「ランレベル」と呼ばれる状態でブートしていました。ランレベルはシステムの起動方法および稼働中のシステムで使用可能なサービスを定義します。ランレベルは番号付けされています。よく知られているランレベルは、`0` (システムのシャットダウン)、`3` (ネットワークを使用するマルチユーザシステム)、および `5` (ネットワークとディスプレイマネージャを使用するマルチユーザシステム)です。

systemdでは、「ターゲットユニット」と呼ばれる仕組みを使用する新しい概念が導入されています。ただし、ランレベルの概念とも、完全な互換性を維持しています。ターゲットユニットには、番号ではなく名前が付けられており、特定の目的を果たします。たとえば、ターゲット `local-fs.target` や `swap.target` は、それぞれローカルファイルシステムのマウントと、スワップ領域のマウントを実行します。

ターゲット `graphical.target` は、ネットワーク機能とディスプレイマネージャ機能を使用するマルチユーザシステムで、ランレベル5に相当します。 `graphical.target` などの複合ターゲットは、他のターゲットのサブセットを組み合わせることで、「メタ」ターゲットとして機能します。systemdでは、既存のターゲットを組み合わせることで簡単にカスタムターゲットを作成できるため、非常に柔軟な運用が実現されます。

次のリストは、systemdの最も重要なターゲットユニットを示しています。すべてを網羅したリストについては、`man 7 systemd.special` を参照してください。

SYSTEMDで選択できるターゲットユニット

`default.target`
: デフォルトで起動されるターゲット。「実在する」ターゲットというよりは、別のターゲット(`graphic.target` など)に対するシンボリックリンクであるといえます。YaSTを介して恒久的に変更できます(10.4項「YaSTを使用したサービスの管理」を参照)。セッション用に変更する場合は、ブートプロンプトで、カーネルのコマンドラインオプション `systemd.unit=<my_target>.target` を使用してください。

`emergency.target`
: コンソール上で非常用のシェルを起動します。ブートプロンプトでのみ、`systemd.unit=emergency.target` と指定して使用します。

`graphical.target`
: ネットワークとマルチユーザをサポートし、ディスプレイマネージャを使用するシステムを起動します。

`halt.target`
: システムをシャットダウンします。

`mail-transfer-agent.target`
: メールの送受信に必要なすべてのサービスを起動します。

`multi-user.target`
: ネットワークに対応したマルチユーザシステムを起動します。

`reboot.target`
: システムを再起動します。

`rescue.target`
: ネットワークに対応しないシングルユーザシステムを起動します。

System V initランレベルシステムとの互換性を維持するために、systemdでは、`runlevelX.target`という名前の特別なターゲットが用意されています。それぞれ`X`の部分がランレベルの番号に対応します。

現在のターゲットを知りたい場合は、`systemctl get-default`コマンドを使用します。

表 10.3 SYSTEM Vのランレベルとsystemdのターゲットユニット

System Vランレベル	`systemd`ターゲット	用途
0	`runlevel0.target`、`halt.target`、`poweroff.target`	システムのシャットダウン
1、S	`runlevel1.target`、`rescue.target`	シングルユーザモード
2	`runlevel2.target`、`multi-user.target`	リモートネットワークなしのローカルマルチユーザ
3	`runlevel3.target`、`multi-user.target`	ネットワークを使用するフルマルチユーザ
4	`runlevel4.target`	未使用/ユーザ定義
5	`runlevel5.target`、`graphical.target`	ネットワークとディスプレイマネージャを使用するフルマルチユーザ
6	`runlevel6.target`、`reboot.target`	システムの再起動

重要: systemdで/etc/inittabが無視されることについて

System V initシステムのランレベルは、`/etc/inittab`で設定されています。systemdでは、この設定が使用されることはありません。独自のブート可能なターゲットを作成する方法については、10.5.3項「カスタムターゲットの作成」を参照してください。

10.3.1.1 ターゲット変更用のコマンド

次のコマンドを使用して、ターゲットユニットを操作します。

作業	systemdコマンド	System V initコマンド
現在のターゲット/ランレベルの変更	`systemctl isolate <my_target>.target`	`telinit X`
デフォルトのターゲット/ランレベルへの変更	`systemctl default`	該当なし

現在のターゲット/ランレベルの取得	`systemctl list-units --type=target`
	systemdでは通常、複数のアクティブターゲットを利用します。そのため、このコマンドは現在アクティブなターゲットをすべて表示します。
		`who -r`
		または
		`runlevel`

デフォルトのランレベルの恒久的な変更	サービスマネージャを使用するか、次のコマンドを実行します。 `ln -sf /usr/lib/systemd/system/<my_target>.target /etc/systemd/system/default.target`	サービスマネージャを使用するか、次の行を変更します。 `id:X:initdefault:` (`/etc/inittab`内にある)
現在のブートプロセスに対するデフォルトランレベルの変更	ブートプロンプトで次のオプションを入力します。 `systemd.unit=<my_target>.target`	ブートプロンプトで必要なランレベルの番号を入力します。

作業	systemdコマンド	System V initコマンド

ターゲットやランレベルの依存関係の表示	`systemctl show -p "Requires" <my_target.target>`
	`systemctl show -p "Wants" <my_target>.target`
	「Requires」を指定すると、ハード依存関係(必ず解決する必要がある依存関係)が表示されます。「Wants」を指定すると、ソフト依存関係(可能であれば解決される依存関係)が表示されます。
		該当なし

10.3.2 システム起動のデバッグ

systemdには、システム起動プロセスを分析できる機能が用意されています。この機能により、全サービスのリストとそのステータスを(`/varlog/`を解析する方法よりは)便利に確認することができます。systemdでは、起動手順を精査して、サービスの起動にかかっている時間を調べることもできます。

10.3.2.1 サービスの起動の確認

システムのブート後に起動された全サービスのリストを確認するには、`systemctl`と入力します。次のように、すべてのアクティブなサービスが表示されます (一部省略しています)。特定のサービスの詳細情報が必要な場合は、`systemctl status <my_service>.service`と入力してください。

例 10.1 アクティブなサービスの一覧表示

```
root # systemctl
UNIT                                LOAD   ACTIVE SUB       JOB DESCRIPTION
[...]
systemd-random-seed-load.path       loaded active waiting       Random Seed
acpid.service                       loaded active running       ACPI Event Daemon
apache2.service                     loaded failed failed        apache
avahi-daemon.service                loaded active running       Avahi mDNS/DNS-SD
 Stack
bluez-coldplug.service              loaded active exited        LSB: handles udev
 coldplug of bluetooth dongles
console-kit...-system-start.service loaded active exited        Console System
 Startup Logging
```

```
cron.service                          loaded active running       Command Scheduler
cups.service                          loaded active running       CUPS Printing
 Service
[...]
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
JOB    = Pending job for the unit.

107 units listed. Pass --all to see inactive units, too.
```

起動に失敗したサービスだけを表示する場合は、`--failed`オプションを指定してください。

例 10.2 起動に失敗したサービスの一覧表示

```
root # systemctl --failed
UNIT                    LOAD   ACTIVE SUB    JOB DESCRIPTION
apache2.service         loaded failed failed     apache
NetworkManager.service loaded failed failed     Network Manager
plymouth-start.service loaded failed failed     Show Plymouth Boot Screen

[...]
```

10.3.2.2 起動時間のデバッグ

システムの起動時間をデバッグするために、systemdでは、`systemd-analyze`コマンドが用意されています。このコマンドでは、全体の起動時間や起動時間順のサービス一覧を表示できるほか、他のサービスの起動時間と対比するために利用できる、SVG画像を生成することもできます。

システムの起動時間の一覧表示

```
root # systemd-analyze
Startup finished in 2666ms (kernel) + 21961ms (userspace) = 24628ms
```

サービスの起動時間の一覧表示

```
root # systemd-analyze blame
```

```
  6472ms systemd-modules-load.service
  5833ms remount-rootfs.service
  4597ms network.service
  4254ms systemd-vconsole-setup.service
  4096ms postfix.service
  2998ms xdm.service
  2483ms localnet.service
  2470ms SuSEfirewall2_init.service
  2189ms avahi-daemon.service
  2120ms systemd-logind.service
  1210ms xinetd.service
  1080ms ntp.service
[...]
    75ms fbset.service
    72ms purge-kernels.service
    47ms dev-vda1.swap
    38ms bluez-coldplug.service
    35ms splash_early.service
```

サービスの起動時間を表す画像

```
root # systemd-analyze plot > jupiter.example.com-startup.svg
```

10.3.2.3 起動プロセス全体の確認

上述のコマンドを利用することで、起動したサービスと起動にかかった時間を確認できます。さらに詳しい情報が必要な場合は、ブートプロンプトで次のパラメータを入力することにより、`systemd` に対して、起動手順全体の冗長ログを記録するように指示できます。

```
systemd.log_level=debug systemd.log_target=kmsg
```

`systemd` が、ログメッセージをカーネルのリングバッファに書き込むようになります。バッファを閲覧するには、`dmesg` を使用してください。

```
dmesg -T | less
```

10.3.3 System Vとの互換性

SystemdはSystem Vと互換性があるため、引き続き既存のSystem V initスクリプトを使用できます。ただし、そのままではSystemdでSystem V initスクリプトを使用できない既知の問題が少なくとも1つあります。initスクリプトで`su`または`sudo`を使用して別のユーザとしてサービスを起動すると、スクリプトエラーになり、「」「アクセス拒否」エラーが生成されます。

`su`または`sudo`を使用してユーザを変更すると、PAMセッションが開始されます。このセッションは、initスクリプトが完了すると終了します。その結果、initスクリプトで起動されたサービスも終了します。このエラーを回避するには、次の手順に従います。

1. initスクリプトと同じ名前を持ち、ファイル名拡張子`.service`が付くサービスファイルラッパーを作成します。

```
[Unit]
Description=DESCRIPTION
After=network.target

[Service]
User=USER
Type=forking ❶
PIDFile=PATH TO PID FILE ❶
ExecStart=PATH TO INIT SCRIPT start
ExecStop=PATH TO INIT SCRIPT stop
ExecStopPost=/usr/bin/rm -f PATH TO PID FILE ❶

[Install]
WantedBy=multi-user.target ❷
```

   _大文字で記述されている値はすべて適切な値に置き換えてください。

   ❶ オプション: initスクリプトでデーモンを起動する場合にのみ使用してください。

   ❷ `multi-user.target`は、`graphical.target`でブートしたときにinitスクリプトも起動します。ディスプレイマネージャでブートする場合にのみinitスクリプトを起動するときは、ここで`graphical.target`を使用します。

2. `systemctl start APPLICATION.service`でデーモンを起動します。

10.4 YaSTを使用したサービスの管理

基本的なサービス管理は、YaSTサービスマネージャモジュールで行うこともできます。このモジュールは、サービスの起動、停止、有効化、および無効化をサポートしています。サービスのステータスを表示したり、デフォルトのターゲットを変更することもできます。[YaST] › [システム] › [サービスマネージャ]の順に選択して、YaSTモジュールを起動します。

図 10.1 サービスマネージャ

[デフォルトのシステムターゲット]の変更

システムのブート先になるターゲットを変更するには、[デフォルトのシステムターゲット]ドロップダウンボックスからターゲットを選択します。最もよく使用されているターゲットは、[グラフィカルインタフェース](グラフィカルなログイン画面を起動する)と[マルチユーザ](コマンドラインモードでシステムを起動する)です。

サービスの起動または停止

テーブルからサービスを選択します。[アクティブ]列は、現在サービスが実行されているかどうかを示します([アクティブ]か、[アクティブでない]かを示します)。ステータスを切り替えるには、[起動/停止]を選択します。

サービスを起動または停止すると、現在実行されているセッションのステータスが変更されます。再起動時にステータスを変更するには、サービスを有効化または無効化する必要があります。

サービスの有効化または無効化

テーブルからサービスを選択します。[ ] 有効化列は、現在サービスが [ ]「有効化」されているか、それとも [ ]「無効化」されているかを示します。ステータスを切り替えるには、[有効/無効] を選択します。

サービスを有効化/無効化することにより、サービスがブート時に起動されるかどうか([ ]「有効化」)または [ ]「無効化」)を設定できます。この設定は、現在のセッションには影響しません。現在のセッションにおけるサービスのステータスを変更するには、サービスを起動または停止する必要があります。

ステータスメッセージの表示

サービスのステータスメッセージを表示するには、リストからサービスを選択し、[詳細の表示] を選択します。表示される内容は、コマンド `systemctl -l status <my_service>` で生成されたものと同じです。

警告: ランレベルの設定を誤るとシステムに害が及ぶことがある

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

10.5 systemdのカスタマイズ

次のセクションには、`systemd` のカスタマイズ例が示されています。

警告: カスタマイズを上書きされないようにする

systemdのカスタマイズは `/etc/systemd/` で行ってください。`/usr/lib/systemd/` では、絶対に行わないでください。そうしないと、systemdの次回の更新によって、変更内容が上書きされてしまいます。

10.5.1 サービスファイルのカスタマイズ

systemdサービスファイルは、`/usr/lib/systemd/system` にあります。サービスファイルをカスタマイズする場合は、次の手順に従います。

1. 変更対象のファイルを `/usr/lib/systemd/system` から `/etc/systemd/system` にコピーします。ファイル名は、元の名前のまま残します。

2. `/etc/systemd/system` のコピーを適宜変更します。
3. 設定変更の概要を表示するには、`systemd-delta` コマンドを使用します。このコマンドを使用すると、他の設定ファイルを上書きする設定ファイルを特定したり、複数の設定ファイルを比較対照することができます。詳細については、`systemd-delta` マニュアルページを参照してください。

ファイル名が同じ場合、`/etc/systemd` にある変更済みファイルが、`/usr/lib/systemd/system` にある元のファイルよりも優先的に使用されます。

10.5.2 「ドロップイン」ファイルの作成

設定ファイルに何行かを追加したり、設定ファイルのごく一部を変更するには、「ドロップイン」と呼ばれるファイルを使用します。ドロップインファイルを使用すると、ユニットファイルの設定を拡張できます。その際に、ユニットファイル自体は編集も上書きもされません。

たとえば、`/usr/lib/systemd/system/ foobar.service` にある `foobar` サービスの1つの値を変更するには、次の手順に従います。

1. `/etc/systemd/system/<my_service>.service.d/` というディレクトリを作成します。`.d` サフィックスが付いていることに注意してください。それ以外の点では、このディレクトリは、ドロップインファイルでパッチ適用するサービスと同じ名前になります。
2. ディレクトリ内に、`whatevermodification.conf` ファイルを作成します。このファイルには、変更する値が設定されている行のみを含めます。
3. ファイルに変更内容を保存します。このファイルは、元のファイルへの拡張として使用されます。

10.5.3 カスタムターゲットの作成

System V init SUSEシステムでは、管理者が独自のランレベル設定を作成できるように、ランレベル4は使用されていません。systemdでは、任意の数のカスタムターゲットを作成できます。ターゲットの作成は、`graphical.target` などの既存のターゲットを改変することから始めることをお勧めします。

1. 設定ファイル `/usr/lib/systemd/system/graphical.target` を `/etc/systemd/system/<my_target>.target` にコピーして、必要に応じて修正してください。
2. 前のステップでコピーした設定ファイルは、すでにターゲットの必須な(「ハード」)依存関係を構築した状態になっています。希望する(「ソフト」)依存関係も構築するには、`/etc/systemd/system/<my_target>.target.wants` ディレクトリを作成します。

3. 希望するサービスごとに、`/usr/lib/systemd/system`から`/etc/systemd/system/<my_target>.target.wants`へのシンボリックリンクを作成します。

4. ターゲットの設定が完了したら、新しいターゲットを利用できるようにするために、systemdの設定を再ロードします。

```
systemctl daemon-reload
```

10.6 高度な使用方法

次のセクションでは、システム管理者向けの高度なトピックについて説明します。さらに高度なsystemdのドキュメントについては、Lennart Pöttering氏によるsystemdの資料(http://0pointer.de/blog/projects)を参照してください。

10.6.1 システムログ

10.6.6項 「サービスのデバッグ」には、特定のサービスに対するログメッセージを閲覧する方法が説明されていますが、表示されるログメッセージは、サービスログからのものだけであるとは限りません。`systemd`が記録したすべてのログメッセージ(「」「ジャーナル」と呼ばれる)にアクセスして問い合わせることもできます。最も古いログから始めて、すべてのログを表示するには、`systemd-journalctl`コマンドを使用します。フィルタの適用や出力形式の変更については、`man 1 systemd-journalctl`を参照してください。

10.6.2 スナップショット

`systemd`の現在の状態を名前付きのスナップショットに保存し、後で`isolate`サブコマンドを使用してその状態に戻ることができます。定義した状態にいつでも戻ることができるため、サービスやカスタムターゲットをテストする際に便利です。スナップショットは現在のセッションでのみ使用可能で、システムを再起動すると自動的に削除されます。スナップショットの名前は、`.snapshot`で終わる必要があります。

スナップショットの作成

```
systemctl snapshot <my_snapshot>.snapshot
```

スナップショットの削除

```
systemctl delete <my_snapshot>.snapshot
```

スナップショットの表示

```
systemctl show <my_snapshot>.snapshot
```

スナップショットの有効化

```
systemctl isolate <my_snapshot>.snapshot
```

10.6.3 カーネルモジュールのロード

`systemd` により、次の場所にある環境設定ファイルを使用してブート時に自動的にカーネルモジュールをロードできます。

- `/usr/lib/modules-load.d` および
- `/etc/modules-load.d`

詳細については、`modules-load.d(5)` のマニュアルページを参照してください。

10.6.4 カーネルのコントロールグループ(cgroup)

従来のSystem V initシステムでは、特定のプロセスを、その生成元のサービスに対して明確に割り当てられないことがありました。Apacheなどの一部のサービスは、サードパーティのプロセス(CGIやJavaのプロセス)を多数生成し、サードパーティのプロセス自体もさらにプロセスを生成します。サービスに対する明確な割り当ては難しいことがあるだけでなく、場合によっては不可能であることもあります。一部の子プロセスを残して、サービスが正しく終了しないことも考えられます。

systemdでは、各プロセスを独自のcgroupに配置することでこの問題を解決しています。cgroupはプロセスをまとめるためのカーネルの機能で、すべての子プロセスを階層構造のグループとして管理します。systemdでは、各cgroupにそのサービスの名前が付けられています。非特権プロセスではcgroupから「離脱」できないため、サービスから生成したプロセスがどれなのかをサービス名によって判別できる効果的な仕組みです。

サービスに属するすべてのプロセスを表示するには、`systemd-cgls` コマンドを使用します。次の例のような結果になります(一部省略しています)。

例 10.3 サービスに属するすべてのプロセスの表示

```
root # systemd-cgls --no-pager
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 20
├─user.slice
│ └─user-1000.slice
│   ├─session-102.scope
│   │ ├─12426 gdm-session-worker [pam/gdm-password]
│   │ ├─15831 gdm-session-worker [pam/gdm-password]
│   │ ├─15839 gdm-session-worker [pam/gdm-password]
│   │ ├─15858 /usr/lib/gnome-terminal-server

[...]

└─system.slice
  ├─systemd-hostnamed.service
  │ └─17616 /usr/lib/systemd/systemd-hostnamed
  ├─cron.service
  │ └─1689 /usr/sbin/cron -n
  ├─ntpd.service
  │ └─1328 /usr/sbin/ntpd -p /var/run/ntp/ntpd.pid -g -u ntp:ntp -c /etc/ntp.conf
  ├─postfix.service
  │ ├─ 1676 /usr/lib/postfix/master -w
  │ ├─ 1679 qmgr -l -t fifo -u
  │ └─15590 pickup -l -t fifo -u
  ├─sshd.service
  │ └─1436 /usr/sbin/sshd -D

[...]
```

cgroupの詳細については、Book “System Analysis and Tuning Guide” 8 “Kernel Control Groups”を参照してください。

10.6.5 サービスの終了(シグナルの送信)

10.6.4項「カーネルのコントロールグループ(cgroup)」で説明したとおり、System V initのシステムでは、プロセスをその親サービスプロセスに割り当てることができないことがあります。そのため、サービスとそのすべての子プロセスを終了するのが難しくなります。終了されていない子プロセスは、ゾンビプロセスとして残ってしまいます。

各サービスをcgroupに範囲制約するという、systemdの概念を採用することで、サービスのすべての子プロセスを明確に判別し、それら各プロセスに対してシグナルを送信できます。サービスに対してシグナルを送信する場合は、`systemctl kill`コマンドを使用します。使用可能なシグナルの一覧については、`man 7 signals`を参照してください。

サービスに対する`SIGTERM`の送信

`SIGTERM`は、送信されるデフォルトのシグナルです。

```
systemctl kill <my_service>.service
```

サービスに対する`SIGNAL`の送信

`-s`オプションを使用することで、送信するシグナルを指定できます。

```
systemctl kill -s SIGNAL <my_service>.service
```

プロセスの選択

デフォルトでは、`kill`コマンドは、指定したcgroup内の`all` (すべての)プロセスに対してシグナルを送信します。`control` (制御)または`main` (メイン)のプロセスに対してだけ送信することもできます。限定されたプロセスに対する送信は、`SIGHUP`を送信して設定を再ロードさせるような場合に有効です。

```
systemctl kill -s SIGHUP --kill-who=main <my_service>.service
```

10.6.6 サービスのデバッグ

デフォルトでは、systemdは過剰に冗長な出力を行いません。サービスの起動が成功した場合は何も出力されず、失敗した場合は短いエラーメッセージが表示されます。サービスの起動や操作をデバッグする場合は、`systemctl status`コマンドを使用してください。

systemdは、独自のログ機構(「ジャーナル」)でシステムメッセージを記録します。これにより、サービスメッセージとステータスメッセージを両方とも表示できます。`status`コマンドは`tail`コマンドに似た動作をするほか、ログメッセージをさまざまな形式で表示することもできます。これにより、強力なデバッグツールとして利用できるようになっています。

サービスの起動失敗の表示

サービスの起動に失敗した場合は、`systemctl status <my_service>.service`を実行することで、詳細なエラーメッセージを表示することができます。

```
root # systemctl start apache2.service
Job failed. See system journal and 'systemctl status' for details.
root # systemctl status apache2.service
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
   Active: failed (Result: exit-code) since Mon, 04 Jun 2012 16:52:26 +0200;
 29s ago
   Process: 3088 ExecStart=/usr/sbin/start_apache2 -D SYSTEMD -k start
 (code=exited, status=1/FAILURE)
   CGroup: name=systemd:/system/apache2.service

Jun 04 16:52:26 g144 start_apache2[3088]: httpd2-prefork: Syntax error on line
205 of /etc/apache2/httpd.conf: Syntax error on li...alHost>
```

直近`n`件のサービスメッセージ

`status`サブコマンドは、デフォルトではサービスが出力した直近の10件のメッセージを表示します。表示するメッセージの件数を変更したい場合は、`--lines=n`パラメータを使用して実行してください。

```
systemctl status ntp.service
systemctl --lines=20 status ntp.service
```

追記モードによるサービスメッセージの表示

サービスメッセージを「リアルタイムに」表示するには、`--follow`オプションを使用します。このオプションは、`tail -f`に似た動作をします。

```
systemctl --follow status ntp.service
```

メッセージの出力形式

`--output=モード`パラメータを指定すると、サービスメッセージの出力形式を変更できます。最も重要なモードには次のものがあります。

`short`

デフォルトの形式。ログメッセージを、人間が読みやすいタイムスタンプと併記して表示します。

`verbose`
: すべての項目を表示する完全な出力。

`cat`
: タイムスタンプを併記しない、簡潔な出力。

10.7 詳細情報

systemdの詳細については、次のオンラインリソースを参照してください。

ホームページ
: http://www.freedesktop.org/wiki/Software/systemd

systemd (管理者向け)
: systemdの著者のうちの1人、Lennart Pötteringによるブログに、systemdに関する複数の投稿があります(本章記述時点では13個の投稿)。それらは、次のサイトに記載されています。http://0pointer.de/blog/projects

Control Centre: The `systemd` Linux init system
: http://www.h-online.com/open/features/Control-Centre-The-systemd-Linux-init-system-1565543.html

Booting up: Tools and tips for systemd, a Linux init tool
: http://www.h-online.com/open/features/Booting-up-Tools-and-tips-for-systemd-1570630.html

11 journalctl:systemdジャーナルのクエリ

SUSE Linux Enterprise 12の従来のinitスクリプトがsystemdに置き換えられた際に(第10章 systemdデーモンを参照)、ジャーナルと呼ばれる専用ログシステムが導入されました。すべてのシステムイベントがジャーナルに書き込まれるようになったため、syslogベースのサービスを実行する必要はありません。

ジャーナル自体は、systemdによって管理されるシステムサービスです。完全な名前はsystemd-journald.serviceです。カーネル、ユーザプロセス、およびシステムサービスの標準入力とエラーから受信したログ情報に基づいて、構造化されたインデックスジャーナルを維持することで、ログデータを収集して保存します。systemd-journald.serviceサービスはデフォルトでオンになっています。

```
# systemctl status systemd-journald
systemd-journald.service - Journal Service
   Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static)
   Active: active (running) since Mon 2014-05-26 08:36:59 EDT; 3 days ago
     Docs: man:systemd-journald.service(8)
           man:journald.conf(5)
 Main PID: 413 (systemd-journal)
   Status: "Processing requests..."
   CGroup: /system.slice/systemd-journald.service
           └─413 /usr/lib/systemd/systemd-journald
[...]
```

11.1 ジャーナルの永続化

ジャーナルは、デフォルトでは/run/log/journal/にログデータを保存します。/run/ディレクトリは本質的に揮発性であるため、再起動するとログデータは失われます。ログデータを永続化するには、systemd-journaldサービスがそのデータを保存できる、適切な所有権と許可のある/var/log/journal/ディレクトリが存在している必要があります。systemdは自動的にディレクトリを作成します。永続的なログに切り替えるには、次の手順を実行します。

1. rootとして、/etc/systemd/journald.confを開き編集します。

   ```
   # vi /etc/systemd/journald.conf
   ```

2. Storage=を含む行をコメント解除し、次のように変更します。

```
[...]
[Journal]
Storage=persistent
#Compress=yes
[...]
```

3. ファイルを保存して、systemd-journaldを再起動します。

```
systemctl restart systemd-journald.service
```

11.2 journalctlの便利なスイッチ

このセクションでは、デフォルトの`journalctl`の動作を拡張する一般的な便利なオプションをいくつか紹介します。スイッチはすべて、`journalctl`のマニュアルページの`man 1 journalctl`で説明されています。

ヒント

特定の実行可能ファイルに関連するすべてのジャーナルメッセージを表示するには、実行可能ファイルのフルパスを指定します。

```
# journalctl /usr/lib/systemd/systemd
```

-f

最新のジャーナルメッセージのみを表示し、新しいログエントリがジャーナルに追加されるとそれらを出力します。

-e

メッセージを出力してジャーナルの最後に移動します。これにより、最新のエントリをページャ内に表示できます。

-r

ジャーナルのメッセージを逆順に出力します。これにより、最新のエントリが最初に一覧にされます。

-k

カーネルメッセージのみを表示します。これは、フィールド照合機能 `_TRANSPORT=kernel` と同等です(11.3.3項「フィールドに基づくフィルタ」を参照)。

-u

指定した `systemd` ユニットのメッセージのみを表示します。これは、フィールド照合機能 `_SYSTEMD_UNIT=UNIT` と同等です(11.3.3項「フィールドに基づくフィルタ」を参照)。

```
# journalctl -u apache2
[...]
Jun 03 10:07:11 pinkiepie systemd[1]: Starting The Apache Webserver...
Jun 03 10:07:12 pinkiepie systemd[1]: Started The Apache Webserver.
```

11.3 ジャーナル出力のフィルタ

スイッチなしで `journalctl` を呼び出すと、最も古いエントリを先頭にジャーナルのすべてのコンテンツが表示されます。出力は、特定のスイッチとフィールドによってフィルタできます。

11.3.1 ブート番号に基づくフィルタ

`journalctl` は特定のシステムブートに基づいてメッセージをフィルタできます。利用可能なブートを一覧もするには、次を実行します。

```
# journalctl --list-boots
-1 097ed2cd99124a2391d2cffab1b566f0 Mon 2014-05-26 08:36:56 EDT—Fri 2014-05-30
 05:33:44 EDT
 0 156019a44a774a0bb0148a92df4af81b Fri 2014-05-30 05:34:09 EDT—Fri 2014-05-30
 06:15:01 EDT
```

1番目の列にはブートオフセットが一覧にされます。現在のブートの場合は `0`、直前のブートの場合は `-1`、2つ前のブートの場合は `-2` といった具合になります。2番目の列には、ブートIDが含まれ、特定のブートに限定するためのタイムスタンプが続きます。

現在のブートのすべてのメッセージを表示します。

```
# journalctl -b
```

直前のブートのジャーナルメッセージを表示する必要がある場合は、オフセットパラメータを追加します。次の例は、直前のブートメッセージを出力します。

```
# journalctl -b -1
```

もう1つの方法は、ブートIDに基づいてブートメッセージを一覧にする方法です。このためには、_BOOT_IDフィールドを使用します。

```
# journalctl _BOOT_ID=156019a44a774a0bb0148a92df4af81b
```

11.3.2 時間間隔に基づくフィルタ

開始日または終了日、あるいはその両方を指定して、`journalctl`の出力をフィルタできます。日付指定は、「2014-06-30 9:17:16」の形式にする必要があります。時間の部分を省略すると、夜中の12:00と想定されます。秒を省略すると、「:00」と想定されます。日付の部分を省略すると、当日と想定されます。数式の代わりに、「yesterday」、「today」、「tomorrow」などのキーワードを使用できます。これらはそれぞれ、前日の夜中12:00、当日、翌日を示します。「now」を指定すると、当日を示します。また、`-`または`+`をプレフィックスとして付けて、現在時刻の前後を示す相対時間を指定することもできます。

現在時刻以降の新しいメッセージのみを表示し、出力を継続的に更新します。

```
# journalctl --since "now" -f
```

直前の夜12:00から午前3:20までのすべてのメッセージを表示します。

```
# journalctl --since "today" --until "3:20"
```

11.3.3 フィールドに基づくフィルタ

特定のフィールドによってジャーナルの出力をフィルタできます。照合するフィールドの構文は、`FIELD_NAME=MATCHED_VALUE`です(`_SYSTEMD_UNIT=httpd.service`など)。1つのクエリに複数の照合を指定することで、出力メッセージイベントをさらにフィルタすることができます。デフォルトフィールドのリストについては、`man 7 systemd.journal-fields`を参照してください。

特定のプロセスIDによって生成されたメッセージを表示します。

```
# journalctl _PID=1039
```

特定のユーザIDに属するメッセージを表示します。

```
# journalctl _UID=1000
```

カーネルリングバッファのメッセージを表示します(`dmesg`が生成するものと同じ)。

```
# journalctl _TRANSPORT=kernel
```

サービスの標準出力またはエラー出力のメッセージを表示します。

```
# journalctl _TRANSPORT=stdout
```

指定されたサービスによって生成されたメッセージのみを表示します。

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service
```

2つの異なるフィールドを指定すると、同時に両方の式に一致するエントリのみが表示されます。

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1488
```

2つの照合が同じフィールドを示している場合は、いずれかの式に一致するすべてのエントリが表示されます。

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service _SYSTEMD_UNIT=dbus.service
```

「+」セパレータを使用して、2つの式を論理「OR」で組み合わせることができます。次の例は、プロセスIDが1480のAvahiサービスプロセスのすべてのメッセージと、D-Busサービスのすべてのメッセージを表示します。

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1480 +
 _SYSTEMD_UNIT=dbus.service
```

11.4 systemdエラーの調査

このセクションでは、`apache2`の起動時に`systemd`によってレポートされたエラーを検出および修復する方法を示す簡単な例を紹介します。

1. apache2サービスの起動を試みます。

```
# systemctl start apache2.service
Job for apache2.service failed. See 'systemctl status apache2.service' and
 'journalctl -xn' for details.
```

2. サービスの状態に関する記述を確認します。

```
# systemctl status apache2.service
apache2.service - The Apache Webserver
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
   Active: failed (Result: exit-code) since Tue 2014-06-03 11:08:13 CEST; 7min
 ago
  Process: 11026 ExecStop=/usr/sbin/start_apache2 -D SYSTEMD -DFOREGROUND \
           -k graceful-stop (code=exited, status=1/FAILURE)
```

障害の原因となっているプロセスのIDは、11026です。

3. プロセスID11026に関連するメッセージの詳細バージョンを表示します。

```
# journalctl -o verbose _PID=11026
[...]
MESSAGE=AH00526: Syntax error on line 6 of /etc/apache2/default-server.conf:
[...]
MESSAGE=Invalid command 'DocumenttRoot', perhaps misspelled or defined by a
 module
[...]
```

4. `/etc/apache2/default-server.conf` 内のタイプミスを修復し、apache2サービスを起動して、そのステータスを出力します。

```
# systemctl start apache2 && systemctl status apache2
apache2.service - The Apache Webserver
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
   Active: active (running) since Tue 2014-06-03 11:26:24 CEST; 4ms ago
  Process: 11026 ExecStop=/usr/sbin/start_apache2 -D SYSTEMD -DFOREGROUND
           -k graceful-stop (code=exited, status=1/FAILURE)
 Main PID: 11263 (httpd2-prefork)
   Status: "Processing requests..."
   CGroup: /system.slice/apache2.service
           ├─11263 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
```

```
           ├─11280 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11281 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11282 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11283 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           └─11285 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
```

11.5 Journaldの設定

systemd-journaldサービスの動作を調整するには、`/etc/systemd/journald.conf`を変更します。このセクションでは、基本的なオプションの設定のみを取り上げます。ファイルの詳細な説明については、`man 5 journald.conf`を参照してください。変更を有効にするために、次のコマンドでジャーナルを再起動する必要がある点に注意してください。

```
# systemctl restart systemd-journald.service
```

11.5.1 ジャーナルサイズ制限の変更

ジャーナルログデータを永続的な場所に保存する場合(11.1項 「ジャーナルの永続化」を参照)、ジャーナルログデータは`/var/log/journal`が存在するファイルシステムの最大10%を使用します。たとえば、`/var/log/journal`を30GBの`/var`パーティションに配置すると、ジャーナルは最大3GBのディスク容量を使用します。この制限を変更するには、`SystemMaxUse`オプションを変更(およびコメント解除)します。

```
SystemMaxUse=50M
```

11.5.2 ジャーナルの/dev/ttyXへの転送

ジャーナルを端末デバイスに転送し、好みの端末画面(たとえば、`/dev/tty12`)でシステムメッセージに関する通知を受信できます。journaldオプションを次のように変更します。

```
ForwardToConsole=yes
TTYPath=/dev/tty12
```

11.5.3 ジャーナルのSyslog機能への転送

Journaldは、`rsyslog`などの従来のsyslog実装との下位互換性があります。以下が正しいことを確認します。

- rsyslogがインストールされている。

```
# rpm -q rsyslog
rsyslog-7.4.8-2.16.x86_64
```

- rsyslogサービスが有効である。

```
# systemctl is-enabled rsyslog.service
enabled
```

- `/etc/systemd/journald.conf`でsyslogへの転送が有効になっている。

```
ForwardToSyslog=yes
```

12 ブートローダGRUB 2

この章では、SUSE® Linux Enterprise Serverで使用されているブートローダGRUB 2の設定方法について説明します。これは、現在「GRUB 2 Legacy」と呼ばれる従来のGRUBブートローダの後継バージョンです。「」GRUB 2は、SUSE® Linux Enterprise Serverのバージョン12以降でデフォルトのブートローダになっています。YaSTモジュールは、最も重要な設定を行うために使用できます。ブート手順は、総じて第9章 Linuxシステムのブートで説明しています。UEFIマシンでのセキュアブートのサポートの詳細については、第13章 UEFI (Unified Extensible Firmware Interface)を参照してください。

12.1 GRUB LegacyとGRUB 2の主な相違点

- 設定が異なるファイルに保存されます。
- より多くのファイルシステム(Btrfsなど)がサポートされています。
- LVMまたはRAIDデバイスに保存されたファイルを直接読み込めます。
- テーマによってユーザインタフェースを翻訳および変更できます。
- ファイルシステムなどの追加機能をサポートするモジュールをロードするためのメカニズムが組み込まれています。
- 他のカーネルとオペレーティングシステム(Windowsなど)のブートエントリを自動的に検索して生成します。
- Bashに似た最小限のコンソールが組み込まれています。

12.2 設定ファイルの構造

GRUB 2の設定は、次のファイルに基づいています。

`/boot/grub2/grub.cfg`

このファイルには、GRUB 2メニュー項目の設定が含まれます。これは、GRUB Legacyで使用されていた`menu.lst`に代わるものです。`grub.cfg`は`grub2-mkconfig`コマンドによって自動的に生成されます。編集しないでください。

`/boot/grub2/custom.cfg`

このオプションファイルは、ブート時に`grub.cfg`によって直接調達され、ブートメニューにカスタム項目を追加するために使用できます。

`/etc/default/grub`

このファイルは、GRUB 2のユーザ設定を制御し、通常は背景やテーマなどの追加の環境設定を含みます。

`/etc/grub.d/` にあるスクリプト

このディレクトリ内のスクリプトは、`grub2-mkconfig` コマンドの実行時に読み込まれます。スクリプトの命令はメインの設定ファイル `/boot/grub/grub.cfg` に統合されます。

`/etc/sysconfig/bootloader`

この設定ファイルは、ブートローダをYaSTで設定するときと、新しいカーネルがインストールされる際に使用されます。perl-bootloaderで評価され、それに従ってブートローダ設定ファイル(GRUB 2の `/boot/grub2/grub.cfg` など)が変更されます。`/etc/sysconfig/bootloader` は、GRUB 2固有の設定ファイルではありません。その値は、SUSE Linux Enterprise Serverにインストールされているすべてのブートローダに適用されます。

`/boot/grub2/x86_64-efi`、`/boot/grub2/power-ieee1275`、`/boot/grub2/s390x`

これらの設定ファイルにはアーキテクチャ固有のオプションが含まれます。

GRUB 2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、他の設定ファイルからコンパイルされた `/boot/grub2/grub.cfg` ファイルからロードされます(以下を参照)。GRUB 2設定ファイルはすべてシステムファイルとみなされ、編集するには `root` 特権が必要です。

注記: 設定の変更の有効化

GRUB 2設定ファイルを手動で編集した後、`grub2-mkconfig` を実行して変更を有効化する必要があります。ただし、YaSTを使用して設定を変更した場合、`grub2-mkconfig` は自動的に実行されるため、この作業は必要ありません。

12.2.1 /boot/grub2/grub.cfgファイル

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 2の設定ファイル `/boot/grub2/grub.cfg` に基づいており、このファイルにはメニューを使用してブートできるすべての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含まれています。

システムをブートするたびに、GRUB 2はファイルシステムから直接メニューファイルを読み込みます。このため、設定ファイルを変更するたびにGRUB 2を再インストールする必要がありません。`grub.cfg` は、カーネルをインストールまたは削除すると自動的に再構築されます。

grub.cfgは、grub2-mkconfigによって、/etc/default/grubファイルと、/etc/grub.d/ディレクトリにあるスクリプトからコンパイルされます。そのため、このファイルは手動で編集しないでください。代わりに、関連するソースファイルを編集するか、12.3項 「YaSTによるブートローダの設定」で説明されているようにYaST[ブートローダ]モジュールを使用して設定を変更します。

12.2.2 /etc/default/grubファイル

ここには、メニューを表示するタイミングやブートするデフォルトのOSなど、GRUB 2のより一般的なオプションが含まれます。すべての使用可能なオプションについては、次のコマンドの出力を参照してください。

```
grep "export GRUB_DEFAULT" -A50 /usr/sbin/grub2-mkconfig | grep GRUB_
```

定義済みの変数以外にユーザ独自の変数を導入して、後から/etc/grub.dディレクトリにあるスクリプト内でそれらの変数を使用できます。

/etc/default/grubを編集した後は、grub2-mkconfig を実行して、メインの構成ファイルを更新します。

注記: スコープ

このファイルに設定されているオプションはすべて、全ブートエントリに影響する汎用オプションです。Xenカーネルまたは Xenハイパーバイザに固有のオプションは、GRUB_*_XEN_*設定オプションを介して設定できます。詳細については、以下を参照してください。

GRUB_DEFAULT

デフォルトでブートされるブートメニューエントリを設定します。値は、数値、メニューエントリの完全な名前、または「saved」になります。「」

GRUB_DEFAULT=2は、3番目(0から数える)のブートメニューエントリをブートします。

GRUB_DEFAULT="2>0"は、3番目の最上位レベルのメニューエントリの1番目にあるサブメニューエントリをブートします。

GRUB_DEFAULT="Example boot menu entry"は、「Example boot menu entry」というタイトルのメニューエントリをブートします。「」

GRUB_DEFAULT=savedは、grub2-rebootコマンドまたはgrub2-set-defaultコマンドによって指定されたエントリをブートします。grub2-rebootは次回の再起動時にのみ有効なデフォルトブートエントリを設定するのに対し、grub2-set-defaultは変更しない限りデフォルトとして使用されるブートエントリを設定します。

GRUB_HIDDEN_TIMEOUT

ユーザがキーを押すまで、指定された秒数待機します。この間は、ユーザがキーを押さない限りメニューは表示されません。指定された時間内にキーが押されなかった場合、制御は GRUB_TIMEOUT に渡されます。GRUB_HIDDEN_TIMEOUT=0 は、まず< Shift >キーが押されているかどうかを確認し、押されている場合はブートメニューを表示し、押されていない場合は即座にデフォルトのメニューエントリをブートします。これは、GRUB　2によって識別されるブート可能なOSが1つだけの場合のデフォルトです。

GRUB_HIDDEN_TIMEOUT_QUIET

false が指定されていて、GRUB_HIDDEN_TIMEOUT 機能が有効な場合は、空の画面にカウントダウンタイマが表示されます。

GRUB_TIMEOUT

自動的にデフォルトのブートエントリをブートする前に、ブートメニューを表示する時間(秒数)。キーを押すとタイムアウトはキャンセルされ、GRUB　2はユーザが手動で選択するまで待機します。GRUB_TIMEOUT=-1 は、ユーザがブートエントリを手動で選択するまでメニューを表示します。

GRUB_CMDLINE_LINUX

この行のエントリは、標準モードおよび回復モード用のブートエントリの最後に追加されます。この行を使用して、カーネルパラメータをブートエントリに追加します。

GRUB_CMDLINE_LINUX_DEFAULT

GRUB_CMDLINE_LINUX と同じですが、標準モードでのみエントリが追加されます。

GRUB_CMDLINE_LINUX_RECOVERY

GRUB_CMDLINE_LINUX と同じですが、回復モードでのみエントリが追加されます。

GRUB_CMDLINE_LINUX_XEN_REPLACE

このエントリは、すべてのXenブートエントリの GRUB_CMDLINE_LINUX パラメータを完全に置き換えます。

GRUB_CMDLINE_LINUX_XEN_REPLACE_DEFAULT

GRUB_CMDLINE_LINUX_XEN_REPLACE と同じですが、GRUB_CMDLINE_LINUX_DEFAULT のパラメータのみを置き換えます。

GRUB_CMDLINE_XEN

このエントリは、Xenゲストカーネルのカーネルパラメータのみを指定します。基本原則は、GRUB_CMDLINE_LINUX と同じです。

`GRUB_CMDLINE_XEN_DEFAULT`

`GRUB_CMDLINE_XEN`と同じです。基本原則は、`GRUB_CMDLINE_LINUX_DEFAULT`と同じです。

`GRUB_TERMINAL`

入出力端末デバイスを有効化および指定します。`console` (PC BIOSおよびEFIコンソール)、`serial` (シリアル端末)、`ofconsole` (Open Firmwareコンソール)、またはデフォルトの`gfxterm` (グラフィックモード出力)のいずれかになります。また、必要なオプションを引用符で囲むことで、2つ以上のデバイスを有効にすることもできます(たとえば、`GRUB_TERMINAL="console serial"`)。

`GRUB_GFXMODE`

`gfxterm`グラフィカル端末で使用される解像度。使用できるモードはグラフィックカード(VBE)でサポートされているモードのみである点に注意してください。デフォルトは「auto」で、優先解像度の選択を試みます。GRUB 2コマンドラインで「`vbeinfo`」と入力すると、GRUB 2で使用可能な画面解像度が表示されます。コマンドラインにアクセスするには、GRUB 2ブートメニュー画面が表示されているときに「`c`」と入力します。
また、色数を解像度設定に追加することで色数も指定できます(たとえば、`GRUB_GFXMODE=1280x1024x24`)。

`GRUB_BACKGROUND`

`gfxterm`グラフィカル端末の背景イメージを設定します。イメージはブート時にGRUB 2によって読み込み可能なファイルで、拡張子`.png`、`.tga`、`.jpg`、または`.jpeg`で終わる必要があります。必要であれば、イメージは画面に合わせて拡大されます。

`GRUB_DISABLE_OS_PROBER`

このオプションを`true`に設定すると、他のオペレーティングシステムの自動検索は無効になります。`/boot/`内のカーネルイメージと、`/etc/grub.d/`内にあるユーザ独意のスクリプトのオプションのみが検出されます。

`SUSE_BTRFS_SNAPSHOT_BOOTING`

このオプションを`true`に設定すると、GRUB 2を直接Snapperスナップショットでブートできます。詳細については、4.3項 「スナップショットからのブートによるシステムロールバック」を参照してください。

注記

すべての`*_DEFAULT`パラメータは、手動またはYaSTで処理できます。

すべてのオプションのリストについては、GNU GRUBのマニュアル (http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration) を参照してください。すべての設定可能なパラメータのリストについては、http://en.opensuse.org/Linuxrc を参照してください。

12.2.3 /etc/grub.d内のスクリプト

このディレクトリ内のスクリプトは `grub2-mkconfig` コマンドの実行時に読み込まれ、スクリプトの命令は `/boot/grub2/grub.cfg` に統合されます。`grub.cfg` 内のメニュー項目の順序は、このディレクトリ内のファイルの実行順序によって決まります。まず、名前が数字で始まるファイルが、最も小さい数字が付いたものから順番に実行されます。`00_header` は `10_linux` の前に実行され、10_linux は `40_custom` の前に実行されます。アルファベットの名前が付いたファイルが存在する場合は、名前が数字で始まるファイルの後に実行されます。`grub2-mkconfig` の実行中に `grub.cfg` へ出力を生成するのは実行可能ファイルのみです。デフォルトでは、`/etc/grub.d` ディレクトリ内のファイルはすべて実行可能ファイルです。最も重要なスクリプトは次のとおりです。

`00_header`

システムファイルの場所、表示設定、テーマ、以前に保存したエントリなどの環境変数を設定します。また、`/etc/default/grub` に保存されている初期設定をインポートします。通常、このファイルを変更する必要はありません。

`10_linux`

ルートデバイス上のLinuxカーネルを識別し、関連するメニューエントリを作成します。これには、関連する回復モードオプション(有効な場合)が含まれます。最新のカーネルのみがメインメニューページに表示され、その他のカーネルはサブメニューに含まれます。

`30_os-prober`

このスクリプトは、`OS-prober` を使用してLinuxやその他のオペレーティングシステムを検索し、結果をGRUB 2メニューに示します。他の特定のオペレーティングシステム(WindowsやOS Xなど)を識別するためのセクションがあります。

`40_custom`

このファイルを使用すると、`grub.cfg` に簡単にカスタムブートエントリを組み込むことができます。最初は、`exec tail -n +3 $0` の部分を変更しないようにしてください。

`90_persistent`

これは、`grub.cfg` ファイルの対応部分をコピーし、それを未変更のまま出力する特殊なスクリプトです。このようにすることで、`grub.cfg` の該当部分を直接変更し、`grub2-mkconfig` を実行時しても変更内容を維持できます。

処理シーケンスは、名前の先頭の数値によって設定され、最も小さい数値が最初に実行されます。スクリプトの名前が同じ数値で始まる場合は、名前全体のアルファベット順で順序が決まります。

12.2.4 BIOSドライブとLinuxドライブのマッピング

GRUB Legacyでは、`device.map`設定ファイルを使用して、BIOSドライブ番号からLinuxデバイス名を派生させていました。BIOSドライブとLinuxデバイスのマッピングは常に正しく推測できるとは限りません。たとえば、BIOS設定でIDEとSCSIのブートシーケンスが入れ替わると、GRUB Legacyは誤った順序を取得します。

GRUB 2では、`grub.cfg`の生成時にデバイスID文字列(UUID)またはファイルシステムラベルを使用することで、この問題を回避しています。GRUB 2ユーティリティは一時デバイスマップをオンザフライで作成します。通常、特に単一ディスクのシステムの場合は、この処理で十分です。

ただし、GRUB 2の自動デバイスマッピングメカニズムを無効にする必要がある場合は、カスタムマッピングファイル`/boot/grub2/device.map`を作成します。次の例では、マッピングを変更して、`DISK 3`をブートディスクにしています。パーティション番号は、GRUB Legacyでは`0`から始まっていましたが、GRUB 2では`1`から始まる点に注意してください。

```
(hd1)  /dev/disk-by-id/DISK3 ID
(hd2)  /dev/disk-by-id/DISK1 ID
(hd3)  /dev/disk-by-id/DISK2 ID
```

12.2.5 ブート手順実行中のメニューエントリの編集

メニューエントリを直接編集できると、誤設定が原因でシステムがブートしなくなった場合に役立ちます。また、システム設定を変更せずに新しい設定をテストする場合にも使用できます。

1. グラフィカルブートメニューで、編集するエントリを矢印キーで選択します。
2. E を押して、テキストベースのエディタを開きます。
3. 矢印キーを使用して、編集する行へ移動します。

図 12.1 GRUB 2ブートエディタ

ここでは2つのオプションがあります。

a. スペース区切りのパラメータを、`linux` または `linuxefi` で始まる行の終わりに追加して、カーネルパラメータを編集します。すべてのパラメータのリストはhttp://en.opensuse.org/Linuxrc から入手できます。

b. または、一般オプションを編集して、カーネルバージョンなどを変更します。< `<Tab>` >キーを押すと、考えられる完了結果がすべて提示されます。

4. `F10` キーを押して変更内容が反映されたシステムをブートするか、< `Esc` >キーを押して編集内容は破棄し、GRUB 2メニューに戻ります。

この方法で加えた変更は、現在のブートプロセスだけに適用され、永続的に保存されることはありません。

重要: 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。詳細については、図36.2「USキーボードレイアウト」を参照してください。

注記: インストールメディアのブートローダ

従来のBIOSが搭載されたシステム上にあるインストールメディアのブートローダは、引き続きGRUB Legacyになります。ブートオプションを追加するには、エントリを選択し、入力を開始します。インストールブートエントリに追加した内容は、インストール済みシステムに永続的に保存されます。

注記: System zでのGRUB 2メニューエントリの編集

IBM System zでのカーソルの移動と編集コマンドは異なります。詳細については、12.4項「System zにおける端末の使用上の相違点」を参照してください。

12.2.6 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 2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メニューエントリのブートを防止するために、ブートパスワードを設定できます。

重要:

設定すると、ブートのたびにブートパスワードが必要になります。つまり、システムは自動的にはブートしません。

ブートパスワードを設定するには、次の手順に従います。または、YaSTを使用してください([パスワードでブートローダを保護する] を参照してください)。

1. `grub2-mkpasswd-pbkdf2` を使用してパスワードを暗号化します。

```
tux > sudo grub2-mkpasswd-pbkdf2
Password: ****
Reenter password: ****
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

2. `set superusers` コマンドを使用して、結果の文字列をまとめて `/etc/grub.d/40_custom` ファイルに貼り付けます。

```
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

3. `grub2-mkconfig` を実行して、メインの設定ファイルに変更をインポートします。
再起動後、メニューエントリのブートを試みると、ユーザ名とパスワードの入力が求められます。「`root`」と入力し、`grub2-mkpasswd-pbkdf2` コマンドの実行時に入力したパスワードを入力します。資格情報が正しい場合、システムは選択したブートエントリをブートします。

12.3 YaSTによるブートローダの設定

SUSE Linux Enterprise Serverシステムでブートローダの汎用オプション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]の順に選択します。モジュールにシステムの現在のブートローダ設定が示され、変更を加えられます。

[ブートコードオプション]タブで、タイプ、場所、および高度なローダ設定に関する設定を表示および変更できます。GRUB 2ブートローダを使用するには、それが使用可能なブートローダのリストから選択されていることを確認します。

図 12.2 カーネルパラメータ

12.3.1 ブートローダタイプの変更

[ブートコードオプション]タブでブートローダタイプを設定します。SUSE Linux Enterprise Serverのデフォルトのブートローダは、GRUB 2です。GRUBまたはGRUB2-EFIを使用するには、次の手順に従います。

手順 12.1 ブートローダのタイプの変更

1. [ブートローダ]で[GRUB2]、[GRUB2-EFI]、または他のエントリのいずれかを選択します。

   重要: EFIシステムではGRUB2-EFIが必須

   EFIシステムがある場合は、GRUB2-EFIのみをインストールできます。それ以外をインストールすると、システムはブート不能になります。

2. 開かれたダイアログで、次のアクションのいずれかを選択します。

   [新しい設定を提案する]

   YaSTは新しい設定を提案します。

   [Convert Current Configuration (現在の設定の変換)]

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

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

   カスタム設定を書き込みます。このアクションは、SUSE Linux Enterprise Serverのインストール時には使用できません。

   [ディスクに保存された環境設定の読み込み]

   専用のブートローダ設定ファイルをロードします。このアクションは、SUSE Linux Enterprise Serverのインストール時には使用できません。

3. [OK]を2回クリックして、変更内容を保存します。

変換中に、古いGRUB 2はディスクに保存されます。これを使うには、ブートローダのタイプをGRUB 2に戻し、[変換前に保存した環境設定に戻す]を選択します。この操作は、インストール済みのシステムでのみ実行可能です。

注記: カスタムのブートローダ

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

12.3.2 ブートローダの場所の変更

ブートローダの場所を変更するには、次の手順に従います。

手順 12.2 ブートローダの場所の変更

1. [ブートコードオプション]タブを選択し、[ブートローダの場所]で、次のいずれかのオプションを選択します。

   [マスタブートレコードからブート]
   最初のディスクのMBRにブートローダをインストールします(BIOS 中のブートシーケンスプリセットによる)。

   [ルートパーティションからブート]
   `/` パーティションのブートセクタにブートローダがインストールされます(デフォルト)。

   [ブートパーティションからブート]
   `/boot` パーティションのブートセクタにブートローダがインストールされます。

   [拡張パーティションからブート]
   拡張パーティションコンテナにブートローダがインストールされます。

   [カスタムブートパーティション]
   このオプションを選択すると、手動でブートローダの場所を指定できます。

2. [OK]をクリックして、変更を適用します。

12.3.3 ディスクの順序の変更

コンピュータに複数のハードディスクがある場合、ディスクのブートシーケンスを指定できます。詳細については、12.2.4項「BIOSドライブとLinuxドライブのマッピング」を参照してください。

手順 12.3 ディスクの順序の設定

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

12.3.4 詳細オプションの設定

詳細なブートオプションは、[ブートローダのインストール] › [ブートローダのオプション]の順に選択して、設定できます。

12.3.4.1 タブ1: [ブートローダのオプション]

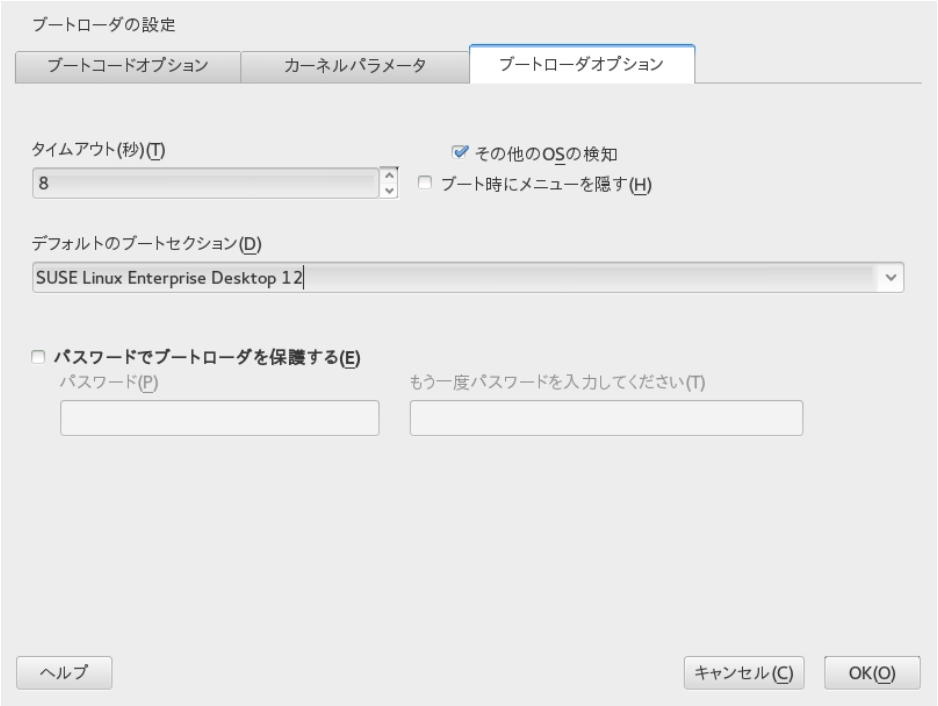


図 12.3 ブートローダのオプション:

[ブートローダのタイムアウト]

新しい値を入力するか、マウスで適切な矢印キーをクリックして、[タイムアウト(秒)]の値を変更します。

［その他のOSの検知］

選択すると、ブートローダはWindowsや他のLinuxインストールなど、インストール済みの他のシステムを検索します。

［ブート時にメニューを隠す］

ブートメニューを隠し、デフォルトエントリをブートします。

［デフォルトブートエントリの調整］

［デフォルトのブートセクション］リストから目的のエントリを選択します。「」ブートエントリ名内の「>」記号は、ブートセクションとそのサブセクションを区切っている点に注意してください。「」

［パスワードでブートローダを保護する］

ブートローダとシステムを追加のパスワードで保護します。詳細については、12.2.6項「ブートパスワードの設定」を参照してください。

12.3.4.2 タブ2: ［カーネルパラメータ］

図 12.4 カーネルパラメータ

［VGAモード］

［VGAモード］オプションは、ブートプロセス時のデフォルトの画面解像度を指定します。

[カーネルコマンドラインパラメータ]

カーネルパラメータは、デフォルトパラメータの最後に追加されます。オプションパラメータは標準のカーネルにのみ追加され、フェールセーフパラメータはフェールセーフカーネルまたは回復カーネルにのみ追加されます。使用できるすべてのパラメータのリストについては、http://en.opensuse.org/Linuxrc を参照してください。

[グラフィカルコンソールを使用する]

オンにすると、テキストモードではなくグラフィカルなスプラッシュスクリーンにブートメニューが表示されます。ブート画面の解像度は、[コンソールの解像度]リストから設定できます。グラフィカルテーマ定義ファイルは、[コンソールのテーマ]ファイル選択機能で指定できます。

[シリアルコンソールの使用]

コンピュータがシリアルコンソールで制御されている場合は、このオプションを有効にして、どのCOMポートをどの速度で使用するか指定します。`info grub`またはhttp://www.gnu.org/software/grub/manual/grub.html#Serial-terminal を参照してください。

12.3.4.3 タブ3: [ブートコードオプション]

図 12.5 カーネルパラメータ

[ブートパーティション用パーティションテーブルにアクティブフラグを設定]

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

[MBRに汎用ブートコードを書き込む]

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

12.4 System zにおける端末の使用上の相違点

3215および3270端末では、カーソルの移動方法と、GRUB 2内での編集コマンドの実行方法にいくつかの相違点と制限事項があります。

12.4.1 制限

対話処理

対話処理は大幅に制限されています。多くの場合、入力しても結果は視覚的なフィードバックとして表示されません。カーソルの位置を確認するには、下線(_)を入力します。

注記

3270端末では、画面の表示と更新は3215端末より優れています。

カーソルの移動

「従来」の方法でカーソルを移動することはできません。「」< Alt >、< Meta >、< Ctrl >、およびカーソルキーは動作しません。カーソルを移動するには、12.4.2項 「キーの組み合わせ」に一覧にされたキーの組み合わせを使用します。

キャレット

キャレット(^)は制御文字として使用されます。文字として ^ を入力し他の文字を続けるには、 ^ 、 ^ 、「 LETTER 」と入力します。

<Enter>

< Enter >キーは動作しません。代わりに、< ^ – j >を使用します。

12.4.2 キーの組み合わせ

共通の代用キー:	<^-J>	決定する(「Enter」「」)
	<^-L>	中止して、直前の「状態」に戻る「」
	<^-I>	タブ補完機能(編集およびシェルモード)
メニューモードで使用可能なキー:	<^-A>	最初のエントリ
	<^-E>	最後のエントリ
	<^-P>	前のエントリ
	<^-N>	次のエントリ
	<^-G>	前のページ
	<^-C>	次のページ
	<^-F>	選択したエントリをブートする、またはサブメニューに切り替える(<^-J>と同じ)
	<E>	選択したエントリを編集する
	<c>	GRUBシェルを起動する
編集モードで使用可能なキー:	<^-P>	前の行に戻る
	<^-N>	次の行に進む
	<^-B>	1文字戻る
	<^-F>	1文字進む
	<^-A>	行の先頭に移動する
	<^-E>	行の末尾

	<^>–<H>	バックスペース
	<^>–<D>	削除
	<^>–<K>	行を削除する
	<^>–<Y>	ヤンク(コピー)
	<^>–<O>	行を開く
	<^>–<L>	画面を更新する
	<^>–<X>	エントリをブートする
	<^>–<C>	GRUBシェルを起動する
コマンドラインモードで使用可能なキー:	<^>–<P>	前のコマンド
	<^>–<N>	履歴の次のコマンド
	<^>–<A>	行の先頭に移動する
	<^>–<E>	行の末尾
	<^>–<B>	1文字戻る
	<^>–<F>	1文字進む
	<^>–<H>	バックスペース
	<^>–<D>	削除
	<^>–<K>	行を削除する
	<^>–<U>	行を破棄する
	<^>–<Y>	ヤンク(コピー)

12.5 役立つGRUB 2コマンド

`grub2-mkconfig`

`/etc/default/grub`および`/etc/grub.d/`のスクリプトに基づいて、新しい`/boot/grub2/grub.cfg`を生成します。

例 12.1 GRUB2-MKCONFIGの使用法

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

ヒント: 構文チェック

パラメータを付けずに`grub2-mkconfig`を実行すると、設定がSTDOUTに出力され、そこで設定を確認できます。構文をチェックするには、`/boot/grub2/grub.cfg`が書き込まれた後に`grub2-script-check`を使用します。

`grub2-mkrescue`

インストールされたGRUB 2設定の、ブート可能なレスキューイメージを作成します。

例 12.2 GRUB2-MKRESCUEの使用法

```
grub2-mkrescue -o save_path/name.iso iso
```

`grub2-script-check`

指定したファイルの構文エラーをチェックします。

例 12.3 GRUB2-SCRIPT-CHECKの使用

```
grub2-check-config /boot/grub2/grub.cfg
```

`grub2-once`

次のブート時にのみ使用されるデフォルトブートエントリを設定します。使用可能なブートエントリのリストを取得するには、`--list`オプションを使用します。

例 12.4 GRUB2-ONCEの使用法

```
grub2-once number_of_the_boot_entry
```

ヒント

オプションを付けずにプログラムを呼び出すと、使用可能なすべてのオプションのリストを取得できます。

12.6 詳細情報

GRUB 2の詳細情報は、http://www.gnu.org/software/grub/ で入手できます。また、`grub` 情報ページも参照してください。http://www.suse.com/support にあるTechnical Information Search (技術情報検索)で、キーワード「GRUB 2」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。「」

13 UEFI (Unified Extensible Firmware Interface)

UEFI (Unified Extensible Firmware Interface) は、システムハードウェアに付属のファームウェア、システムのすべてのハードウェアコンポーネント、およびオペレーティングシステムの間のインタフェースです。

UEFIは、従来のPC-BIOSに代わって、PCで幅広く利用されるようになっています。例えば、UEFIは64ビットシステムを適切にサポートし、最も重要な機能の1つである安全なブート(「セキュアブート」、ファームウェアバージョン2.3.1c以降が必要)を提供します。最後に、UEFIを使用すると、すべてのx86プラットフォームで標準のファームウェアが利用可能になります。

さらに、UEFIには以下の利点があります。

- GUIDパーティションテーブル(GPT)を使う大きなディスク(2 TiB以上)からのブート。
- CPUに依存しないアーキテクチャおよびドライバ。
- ネットワーク機能を持つ柔軟なプレOS環境。
- PC-BIOSライクなエミュレーション経由でレガシーオペレーティングシステムのブートをサポートするCSM(Compatibiity Support Module)。

詳細については、http://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface を参照してください。以降のセクションは、UEFIの一般的な概要を示すものではなく、特定の機能がSUSE Linux Enterpriseにどのように実装されているかを示すヒントです。

13.1 セキュアブート

UEFIの世界では、ブートストラッププロセスの保護とは、信頼チェーンの確立を意味します。SUSE Linux Enterpriseとの関連では、「」「プラットフォーム」はこの信頼チェーンのルートであり、マザーボードおよびオンボードファームウェアが「」「プラットフォーム」とみなされます。また、別の言い方をすれば、ハードウェアベンダー、およびそのハードウェアベンダーからコンポーネントの製造元やOSベンダーなどにつながる信頼チェーンです。

信頼は公開鍵の暗号で表されます。ハードウェアベンダーは、ファームウェアにいわゆるプラットフォームキー(PK)を設定し、信頼のルートを表します。オペレーティングシステムベンダーなどとの信頼関係は、このプラットフォームキーを使ってキーに署名することによって文書化されます。

最後に、これらの「」「信頼された」キーのいずれかで署名されていない限りファームウェアがコード(OSブートローダも、PCI Expressカードやディスクのフラッシュメモリに保存されたドライバも、ファームウェアのアップデートも)を実行できないようにすることによって、セキュリティが確立されます。

基本的に、セキュアブートを使用するには、ファームウェアによって信頼されたキーで署名されたOSローダが必要であり、読み込むカーネルが信頼できることを検証するためにOSローダが必要です。

キー交換キー(KEK)をUEFIキーデータベースに追加できます。この方法で、PKのプライベート部分で署名されている限り、他の証明書を使用できます。

13.1.1 SUSE Linux Enterpriseへの実装

Microsoftのキー交換キー(KEK)がデフォルトでインストールされます。

注記: GUIDパーティションテーブル(GPT)が必要

セキュアブート機能を使用するには、マスタブートレコード(MBR)を使用した古いパーティションをGUIDパーティションテーブル(GPT)に置換する必要があります。

YaSTは、インストール時にEFIモードを検出すると、GPTパーティションの作成を試みます。UEFIでは、FATフォーマットのEFIシステムパーティション(ESP)上でEFIプログラムが見つかるものと想定されます。

UEFIセキュアブートに対応するには、基本的に、ブートローダがデジタル署名されており、ファームウェアがそのデジタル署名を信頼されたキーとして認識することが必要です。SUSE Linux Enterpriseのお客様の利便性を考え、このキーはファームウェアによってあらかじめ信頼されているので、手動での操作は不要です。

これには2つの方法があります。1つは、ハードウェアベンダーにSUSEキーを署名してもらい、SUSEがその署名を使ってブートローダに署名する方法です。もう1つは、MicrosoftのWindows Logo Certificationプログラムを利用してブートローダの認定を受け、MicrosoftにSUSE署名キーを認識してもらう(つまり、MicrosoftのKEKを使って署名してもらう)方法です。これで、SUSEは、UEFI署名サービス(この場合はMicrosoft)によって署名されたローダを入手できます。

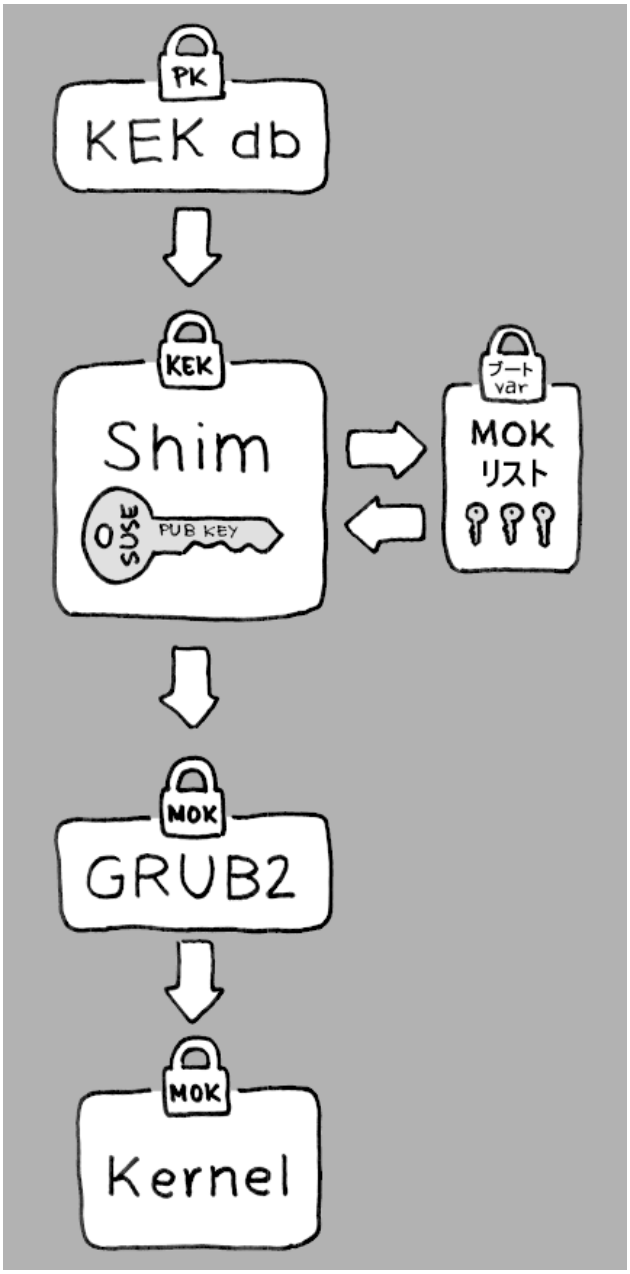


図 13.1 UEFIのセキュアブートプロセス

実装階層においてSUSEは`shim`ローダを使用します。これは法的問題を回避するスマートなソリューションで、証明書および署名の手順が大幅に簡素化されます。`shim`ローダの処理は、ELILOやGRUB 2などのブートローダをロードすることです。次にこのブートローダが、SUSEキーのみで署名されたカーネルをロードします。SUSEは、UEFIセキュアブートが有効化されたSLE11 SP3の新規インストールで、この機能を提供します。

信頼ユーザには2種類あります。

- 1つ目は、キーを保持するユーザです。プラットフォームキー(PK)によって、ほとんどすべてのことが許可されます。キー交換キー(KEK)では、PKの変更を除き、PKに可能なすべてのことが許可されます。

- 2つ目は、マシンに物理的にアクセスできる任意のユーザです。物理的にアクセスできるユーザは、マシンを再起動したりUEFIを設定したりできます。

UEFIには、これらのユーザのニーズを満たすため、2種類の変数があります。

- 1つ目は「認証変数」と呼ばれるもので、ブートプロセス(いわゆるブートサービス環境)および実行中のOSの両方から更新できますが、更新できるのは、古い変数値の署名に使用されたものと同じキーを使って新しい変数値が署名されている場合のみです。また、この変数は、より大きなシリアル番号を持つ値にのみ追加または変更できます。

- 2つ目は、「ブートサービス専用変数」と呼ばれるものです。この変数は、ブートプロセス中に動作する任意のコードにアクセスできます。ブートプロセスの終了後、OSが起動する前に、ブートローダは `ExitBootServices` コールを呼び出す必要があります。その後、これらの変数にはアクセスできなくなり、OSはこれらに触れられません。

さまざまなUEFIキーリストは1つ目のタイプなので、オンラインでの更新、追加、および、キー/ドライバ/ファームウェアの指紋のブラックリスト登録ができます。セキュアブートの実装に役立つのは、2つ目の「Boot Service Only Variable(ブートサービス専用変数)」です。これは、安全かつオープンソースで使いやすくなっており、GPL v3と互換性があるためです。

SUSEは、最初にFedoraによった開発された `shim` (小さくシンプルなEFIブートローダ)で起動します。システム上のUEFIキーデータベースで利用可能なKEKにもとづいて、SUSE KEKで署名された証明書およびMicrosoft発行の証明書によって署名されます。

これによって `shim` のロードおよび実行が可能になります。

`shim` は、続いて、ロードしようとしているブートローダが信頼されていることを確認します。デフォルトで、`shim` は、本体に組み込まれている独自のSUSE証明書を使用します。また、`shim` は、追加のキーを「登録」してデフォルトのSUSEキーを上書きできます。以下、これらを「マシン所有者キー」、または省略してMOKと呼びます。

次に、ブートローダはカーネルを検証および起動し、カーネルがモジュールで同じことを実行します。

13.1.2 Machine Owner Key(マシン所有者キー、MOK)

ユーザ(「マシンの所有者」)がブートプロセスの任意のコンポーネントを置換する場合は、Machine Owner Key(マシン所有者キー、MOK)を使用します。`mokutils` ツールがコンポーネントの署名およびMOKの管理を支援します。

登録プロセスでは、まずマシンを起動し、shimのロード中に(キーを押すなどして)ブートプロセスを中断します。これによってshimが登録モードに移行するので、ユーザは、デフォルトのSUSEキーをブートパーティションのファイルに含まれるキーに置換できます。ユーザがこの処理を選択すると、shimはそのファイルのハッシュを計算し、結果を「Boot Service Only(ブートサービス専用)」変数にします。これによってshimは、ブートサービス以外でファイルが変更された場合にその変更を検出でき、ユーザ承認済みのMOKリストの改ざんを回避できます。

これらすべてがブート時に行われ、検証済みのコードのみが実行されます。このため、コンソールにいるユーザのみがマシン所有者のキーセットを使用できます。OSにリモートアクセスするマルウェアやハッカーではあり得ません。ハッカーやマルウェアはファイルの変更しかできず、「Boot Service Only(ブートサービス専用)」変数に保存されたハッシュを変更できないためです。

いったんロードされshimによって検証されたブートローダは、カーネルを検証する場合はshimにコールバックします(検証コードの複製を避けるため)。shimはMOKと同じリストを使用し、カーネルをロードできるかどうかをブートローダに知らせます。

このようにして、独自のカーネルまたはブートローダをインストールできます。物理的にそこにいることによって新しいキーセットをインストールしそれを認証する必要があるのは、最初の再起動時のみです。MOKは単一のMOKではなくリストなので、shimに複数のベンダーのキーを信頼させることができ、ブートローダからのデュアルブートやマルチブートが可能です。

13.1.3 カスタムカーネルのブート

以下はhttp://en.opensuse.org/openSUSE:UEFI#Booting_a_custom_kernelにもとづいています。

セキュアブートでは、セルフコンパイルカーネルを使用できます。ただし、独自の証明書を使って署名し、その証明書をファームウェアまたはMOKに知らせる必要があります。

1. カスタムのX.509キー、および署名に使用される証明書を作成します。

   ```
   openssl req -new -x509 -newkey rsa:2048 -keyout key.asc \
     -out cert.pem -nodes -days 666 -subj "/CN=$USER/"
   ```

   証明書の作成の詳細については、http://en.opensuse.org/openSUSE:UEFI_Image_File_Sign_Tools#Create_Your_Own_Certificateを参照してください。

2. PKCS#12形式でキーと証明書をパッケージ化します。

   ```
   openssl pkcs12 -export -inkey key.asc -in cert.pem \
   ```

```
  -name kernel_cert -out cert.p12
```

3. `pesign`とともに使用するNSSデータベースを生成します。

```
certutil -d . -N
```

4. PKCS#12に含まれるキーおよび証明書をNSSデータベースにインポートします。

```
pk12util -d . -i cert.p12
```

5. `pesign`を使用して、新しい署名でカーネルを「bless」します。

```
pesign -n . -c kernel_cert -i arch/x86/boot/bzImage \
  -o vmlinuz.signed -s
```

6. カーネルイメージの署名をリスト表示します。

```
pesign -n . -S -i vmlinuz.signed
```

その時点で、通常通り`/boot`にカーネルをインストールできます。カーネルにはカスタム署名があるため、署名に使用された証明書をUEFIファームウェアまたはMOKにインポートする必要があります。

7. ファームウェアまたはMOKにインポートするため、証明書をDERフォーマットに変換します。

```
openssl x509 -in cert.pem -outform der -out cert.der
```

8. よりアクセスしやすくするため、証明書をESPにコピーします。

```
sudo cp cert.der /boot/efi/
```

9. `mokutil`を使用して自動的にMOKリストを起動します。

   a. 証明書をMOKにインポートします。

   ```
   mokutil --root-pw --import cert.der
   ```

   `--root-pw`オプションにより、`root`ユーザを直接使用できます。

   b. これから登録する証明書のリストを確認します。

   ```
   mokutil --list-new
   ```

c. システムを再起動します。`shim`によってMokManagerが起動されるはずです。`root`パスワードを入力して、MOKリストに証明書をインポートすることを確認してください。

d. 新しくインポートしたキーが登録されたかどうかを確認します。

```
mokutil --list-enrolled
```

a. また、MOKを手動で起動する場合は以下の手順を実行します。
再起動

b. GRUB 2メニューで<`c`>キーを押します。

c. 以下のコマンドをタイプします。

```
chainloader $efibootdir/MokManager.efi
boot
```

d. [Enroll key from disk]を選択します。

e. `cert.der`ファイルに移動して<`Enter`>キーを押します。

f. 指示に従ってキーを登録します。通常、「`0`」を押してから「`y`」を押して確認します。
また、ファームウェアメニューに、署名データベースに新しいキーを追加する方法が用意されている場合があります。

13.1.4 機能と制限

セキュアブートモードでブートする場合、次の機能が適用されます。

- UEFIのデフォルトのブートローダがある場所へのインストール。これは、EFIブートエントリを維持または復元するメカニズムです。
- UEFIを介して再起動する。
- フォールバック先のレガシーBIOSがない場合、XenハイパーバイザはUEFIを使用してブートする。
- UEFI IPv6 PXEブートのサポート。
- UEFIはビデオモードのサポートを利用できる。カーネルはUEFIからビデオモードを取得して、同じパラメータでKMSモードを設定できます。
- USBデバイスからのUEFIブートがサポートされる。

セキュアブートモードでブートする場合、次の制限が適用されます。

- セキュアブートを簡単に回避できないようにするため、セキュアブートで実行する場合は一部のカーネル機能が無効になっています。
- ブートローダ、カーネル、およびカーネルモジュールが署名されている必要があります。
- KexecおよびKdumpは無効になっています。
- ハイバネーション(ディスクの休止)は無効になっています。
- ルートユーザであっても、`/dev/kmem` および `/dev/mem` にアクセスできません。
- ルートユーザであっても、I/Oポートにアクセスできません。すべてのX11グラフィカルドライバはカーネルドライバを使用する必要があります。
- sysfs経由でPCI BARにアクセスすることはできません。
- ACPIの `custom_method` は使用できません。
- asus-vmiモジュールに対してdebufgsを使用できません。
- `acpi_rsdp` パラメータはカーネルに影響を及ぼしません。

13.2 その他の情報

- http://www.uefi.org —UEFIのホームページです。現在のUEFI仕様が掲載されています。
- Olaf Kirch氏およびVojtěch Pavlík氏によるブログ記事(上の章の内容はこれらの記事に基づいています)。
  - http://www.suse.com/blogs/uefi-secure-boot-plan/
  - http://www.suse.com/blogs/uefi-secure-boot-overview/
  - http://www.suse.com/blogs/uefi-secure-boot-details/
- http://en.opensuse.org/openSUSE:UEFI —UEFIとopenSUSEに関するページです。

14 特別なシステム機能

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

14.1 特殊ソフトウェアパッケージ

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

14.1.1 bashパッケージと/etc/profile

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

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

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

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

```
cp /etc/skel/.profile ~/.profile
```

それから、個人的な調整点を、*.old ファイルから書き戻します。

14.1.2 cronパッケージ

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

cronテーブルは、/var/cron/tabs にあります。/etc/crontab はシステム全体のcronテーブルとして機能します。ユーザ名を入力して、タイムテーブルの後、コマンドの前に直接コマンドを実行するようにします。例14.1「/etc/crontab内のエントリ」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です。cron のマニュアルページを参照してください(man cron 使用)。

例 14.1 /ETC/CRONTAB内のエントリ

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

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

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

hourly 、daily 、または他の特定の周期の管理スクリプトをカスタム時間で実行するには、/etc/crontab のエントリを使用して、定期的にタイムスタンプファイルを削除します(例14.2「/etc/crontab:タイムスタンプファイルの削除」を参照してください。そこでは、hourly という名前の付いているファイルが毎時59分に、daily という名前の付いているファイルが毎日午前2時14分に削除されるようになっています)。

例 14.2 /ETC/CRONTAB:タイムスタンプファイルの削除

```
59 *  * * *     root  rm -f /var/spool/cron/lastrun/cron.hourly
14 2  * * *     root  rm -f /var/spool/cron/lastrun/cron.daily
29 2  * * 6     root  rm -f /var/spool/cron/lastrun/cron.weekly
```

```
44 2  1 * *     root  rm -f /var/spool/cron/lastrun/cron.monthly
```

または、/etc/sysconfig/cronのDAILY_TIMEをcron.dailyを起動する時刻に設定します。MAX_NOT_RUNの設定では、ユーザが長期間、指定したDAILY_TIMEにコンピュータを起動しなくても、毎日のタスクの実行がトリガされるようにします。MAX_NOT_RUNの最大値は14日です。

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

14.1.3 Cronステータスメッセージの停止

cronステータスメッセージによって大量の電子メールが生成されるのを避けるため、新しいインストールでは、/etc/sysconfig/cronのデフォルト値SEND_MAIL_ON_NO_ERRORが「no」に設定されています。cronのマニュアルページで説明されているように、この設定が「no」になっていても、cronのデータ出力は引き続きMAILTOアドレスに送信されます。

アップデートの場合は、ニーズに合わせてこれらの値を設定することをお勧めします。

14.1.4 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ス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、一定間隔で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で実行できます。通常、これらのログファイルは、FHSで指定されるように/var/log内に格納され、毎日記録が追加されるためにサイズが増大します。logrotateパッケージを使用して、これらのファイルが増大するのを制御できます。

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

例 14.3 /ETC/LOGROTATE.CONFの例

```
# see "man logrotate" for details
```

```
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

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

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

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

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

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

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

重要: パーミッション

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

14.1.5 locateコマンド

ファイルをすばやく検索するためのコマンド`locate`は、標準のインストール済みソフトウェアには含まれていません。必要に応じて、`findutils-locate`の後継パッケージである`mlocate`パッケージを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。

14.1.6 ulimitコマンド

`ulimit`(user limits)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定して、それを表示できます。`ulimit`はアプリケーションが使用できるメモリの制限に特に役立ちます。これを使用して、アプリケーションがシステムリソースを過剰に使用して速度が低下したり、オペレーティングシステムをハングさせたりすることを防止できます。

`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14.1「`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。

表 14.1 `ulimit`:ユーザのためのリソースの設定

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

システム全体のデフォルトエントリは、`/etc/profile`で設定されます。このファイルを直接編集することはお勧めしません。システムをアップグレードすると変更内容が上書きされるためです。システム全体のプロファイル設定をカスタマイズするには、`/etc/profile.local`を使用します。ユーザごとの設定は、`~USER/.bashrc`で行う必要があります。

例 14.4 ULIMIT:~/.BASHRC中の設定

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

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

メモリ割り当ては、KB単位で指定する必要があります。詳細については、`man bash`コマンドでマニュアルページを参照してください。

重要: `ulimit`のサポート

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。PAM(`pam_limits`など)は、`ulimit`の代わりに使用できる包括的な調整手段を提供しています。

14.1.7 freeコマンド

`free`コマンドは、空いている物理メモリ、使用済み物理メモリ、システム内のスワップ領域のほか、カーネルによって消費されたバッファとキャッシュの合計量を表示します。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。

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

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

ただし、目的が現在のRAM使用量である場合は、`/proc/meminfo`で情報を見つけてください。

14.1.8 manページとinfoページ

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

14.1.9 manコマンドを使用したマニュアルページの選択

マニュアルページを読み込むには、`man マニュアルページ`を入力します。同じ名前でさまざまなセクションに存在するマニュアルページは、対応するセクション番号とともに一覧表示されます。表示するマニュアルページを選択します。セクション番号を数秒内に入力しないと、最初のマニュアルページが表示されます。

これをデフォルトのシステム動作に戻すには、`~/.bashrc`などのシェル初期化ファイルで`MAN_POSIXLY_CORRECT=1`を設定します。

14.1.10 GNU Emacs用の設定

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

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

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

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

これらのファイルの詳細については、`info:/emacs/InitFile` の「Init File」にあるEmacs情報ファイルを参照してください。これらのファイルを無効にする(必要な場合)方法についても記載されています。

Emacsのコンポーネントは、いくつかのパッケージに分かれています。

- 基本パッケージの `emacs`。
- `emacs-x11` (通常インストールされている): X11をサポートしているプログラム。
- `emacs-nox`: X11をサポートしていないプログラム。
- `emacs-info`: info形式のオンラインマニュアル。

- `emacs-el`：Emacs Lisp内のコンパイルされていないライブラリファイル。これらは、実行時には必要ありません。
- 必要に応じて`emacs-auctex`(LaTeX)、`psgml`(SGMLおよびXML)、`gnuserv`(クライアント/サーバ操作)など、さまざまなアドオンパッケージをインストールできます。

14.2 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。<Alt–F1>～<Alt–F6>を使用して切り替えます。7番目のコンソールはX用に予約されており、10番目のコンソールにはカーネルメッセージが表示されます。

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

14.3 キーボードマッピング

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

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.emacs
/etc/skel/.gnu-emacs
/etc/skel/.vimrc
/etc/csh.cshrc
/etc/termcap
/usr/share/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
```

```
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

これらの変更は、terminfoエントリを使用するアプリケーション、または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、emacsなど)にのみ影響します。システムに付随しないアプリケーションは、これらのデフォルト値に合わせる必要があります。

Xの下では、<compose>キー(マルチキー)を/etc/X11/Xmodmapで説明されているように有効化できます。

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

ヒント: その他の情報

XKBに関する情報は、/usr/share/doc/packages/xkeyboard-config (xkeyboard-configパッケージの一部)に記載されている文書を参照してください。

14.4 言語および国固有の設定

本システムは、非常に広い範囲で国際化されており、現地の状況に合わせて柔軟に変更できます。言い換えれば、国際化(I18N)が特定のローカライズ(L10N)を可能にします。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_LC_で定義します。これは、単なる現地語サポートだけでなく、Messages(メッセージ) (言語)、Character Set(文字セット)、Sort Order(ソート順)、Time and Date(時刻と日付)、Numbers(数字)およびMoney(通貨)の各カテゴリも指します。これらのカテゴリはそれぞれ、独自の変数を使用して直接定義することも、ファイルlanguageにあるマスタ変数を使用して間接的に定義することも可能です(man localeコマンドでmanページを参照)。

RC_LC_MESSAGES, RC_LC_CTYPE, RC_LC_COLLATE, RC_LC_TIME, RC_LC_NUMERIC, RC_LC_MONETARY

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

RC_LC_ALL

この変数は、すでに参照された変数の値を上書きします。

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

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

変数は、YaSTのsysconfigエディタで設定できます。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>]]
```

14.4.1 例

言語コードと国コードは必ず一緒に設定する必要があります。言語の設定は、http://www.evertype.com/standards/iso639/iso639-en.html および http://www.loc.gov/standards/iso639-2/ で入手できる、ISO 639規格に従います。国コードはISO 3166に一覧にされています(http://en.wikipedia.org/wiki/ISO_3166 を参照)。

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

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

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

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

LANG=en_IE@euro

上記の例では、ユーロ記号が言語設定に明示的に組み込まれています。この設定は今では廃止され、UTF-8もユーロ記号を表現します。アプリケーションがISO-8859-15をサポートし、UTF-8をサポートしない場合にのみ役に立ちます。

/etc/sysconfig/language への変更は、次のプロセスチェーンで有効になります。

- Bashの場合は、/etc/profile によって読み込まれた /etc/profile.d/lang.sh が、/etc/sysconfig/language を解析します。
- tcshの場合は、ログイン時に /etc/csh.login によって読み込まれた /etc/profile.d/lang.csh が、/etc/sysconfig/language を解析します。

これによって、/etc/sysconfig/language に加えられたすべての変更が、これらを手動で有効にしなくても、各シェルへの次回ログイン時に使用可能になります。

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。

14.4.2 ~/.i18nでのロケール設定

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

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

14.4.3 言語サポートの設定

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

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

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

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

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

```
LANG="nn_NO"
```

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

または

```
LANG="nb_NO"
```

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

ノルウェー語では、`LC_TIME`の扱いも違うので注意してください。

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

14.4.4 詳細情報

- 『The GNU C Library Reference Manual』の「Locales and Internationalization」の章。glibc-info **パッケージに格納されています。** パッケージは、SUSE Linux Enterprise SDKから入手できます。このSDKは、SUSE Linux Enterprise向けのアドオン製品であり、http://download.suse.com/からダウンロードできます。「**SUSE Linux Enterpriseソフトウェア開発キット**」で検索してください。
- 『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(http://tldp.org/HOWTO/Unicode-HOWTO-1.html)

15 プリンタの運用

SUSE® Linux Enterprise Serverは、リモートネットワークプリンタも含め、さまざまな種類のプリンタを使った印刷をサポートしています。プリンタは手動で設定することも、YaSTを使用して設定することもできます。設定の詳細については、ブック「導入ガイド」8「YaSTによるハードウェアコンポーネントの設定」8.3「プリンタの設定」を参照してください。プリントジョブの開始、管理には、グラフィカルインタフェースまたはコマンドラインユーティリティの両方を利用できます。 プリンタが正常に動作しない場合は、15.8項「トラブルシューティング」を参照してください。

CUPS (Common Unix Printing System)は、SUSE Linux Enterprise Serverの標準印刷システムです。

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、プリンタがサポートされているインタフェース(USB、Ethernet、またはWi-Fi)を備えていること、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。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。

現在では、標準的な印刷ジョブフォーマットとしてPDFがPostScriptに取って代わりつつあります。PostScriptに加え、PDFも直接印刷できるPostScript+PDFプリンタは、すでに存在しています。従来のPostScriptプリンタでは、印刷ワークフローでPDFをPostScriptに変換する必要があります。

標準的なプリンタ(PCLおよびESC/Pなどの言語)

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」と呼びます。非常によく知られている言語としては、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があります。これらのプリンタ言語は、通常、Linuxによってサポートされており、十分な印刷結果が得られています。Linuxは、一部の特殊な印刷機能に対応できない場合があります。HPとEpson以外には、現時点で、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。

独自規格のプリンタ(GDIプリンタ)

これらのプリンタは、共通のプリンタ言語をサポートしていません。これらのプリンタは独自のプリンタ言語を使用しており、新しいエディション/モデルがリリースされると、プリンタ言語も変更される可能性があります。一般的にこのようなプリンタでは、Windowsドライバしか利用できません。詳細については、15.8.1項 「標準的なプリンタ言語をサポートしないプリンタ」を参照してください。

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

http://www.linuxfoundation.org/OpenPrinting/

プリンタデータベースのあるOpenPrintingホームページです。このデータベースは、最新のLinuxサポートステータスを示します。しかし、Linuxのディストリビューションが統合できるのは、製造の時点で使用可能だったドライバだけです。したがって、現時点で「」「完全にサポート済み」と評価されているプリンタであっても、最新バージョンのSUSE Linux Enterprise Server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。

http://pages.cs.wisc.edu/~ghost/

GhostscriptのWebページ。

`/usr/share/doc/packages/ghostscript/catalog.devices`

組み込みのGhostscriptドライバのリスト。

15.1 CUPS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の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トキューの名前だけでなく、必要に応じて、プリンタ固有のオプションなど、フィルタに関する情報も含まれます。

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

このフィルタは、印刷中のアプリケーションが生成したデータ(通常的はPostScriptやPDFですが、ASCII、JPEGなどの場合もあります)を、プリンタ固有のデータ(PostScript、PCL、ESC/Pなど)に変換します。プリンタの機能については、PPDファイルに記述されています。PPDファイルには、プリン

タ固有のオプションが記述されています。各オプションに対しては、プリンタでそのオプションを有効にするために必要なパラメータが指定されています。フィルタシステムは、ユーザが有効として選択したオプションを確認します。

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

15.2 プリンタに接続するための方法とプロトコル

プリンタをシステムに接続するには、さまざまな方法があります。CUPS印刷システムの設定は、ローカルプリンタと、ネットワーク経由でシステムに接続されているプリンタを区別しません。プリンタ接続の詳細については、http://en.opensuse.org/SDB:CUPS_in_a_Nutshell にアクセスして「CUPS in a Nutshell」という記事を参照してください。

System z IBM System zのメインフレームにローカルで接続するz/VMによって提供されるプリンタおよび類似デバイスは、CUPSでサポートされていません。これらのプラットフォーム上では、ネットワーク経由の印刷だけを利用できます。ネットワークプリンタのケーブリング(ケーブル接続)は、プリンタメーカの指示にしたがって設置する必要があります。

警告: 稼働中システムのケーブル接続の変更

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

15.3 ソフトウェアのインストール

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

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。パッケージ `manufacturer-PPDs` および `OpenPrintingPPDs-postscript` で、多くのPPDファイルが提供されています。15.7.3項 「各種パッケージ内のPPDファイル」および15.8.2項 「特定のPostScriptプリンタに適したPPDファイルが入手できない」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(ブック 「導入ガイド」 8 「YaSTによるハードウェアコンポーネントの設定」8.3.1.1 「YaSTによるドライバの追加」を参照)。その後は、プリンタのセットアップ時にPPDファイルを選択できるようになります。

プリンタメーカーがソフトウェアパッケージ全体をインストールさせようとする場合には注意してください。第一に、このタイプのインストールを行うと、SUSE Linux Enterprise Serverによって提供されているサポートが失われる場合があります。第二に、印刷コマンドが異なる動作をする可能性があり、システムが他のメーカーのデバイスに対応できなくなる場合があります。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。

15.4 ネットワークプリンタ

ネットワークプリンタは、さまざまなプロトコルをサポートでき、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が標準化されているので、一部のメーカーは標準を変更します。そして、メーカーは、2、3のオペレーティングシステムにのみ対応するドライバを提供します。残念なことに、Linuxドライバはめったに提供されません。現在の状況では、あらゆるプロトコルがLinux環境で円滑に動作するという仮定に基づいて行動することはできません。したがって、機能する設定を実現するために、さまざまなオプションを実験する必要があります。

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

socket
: ソケットは、プレインプリントデータのTCPソケットへの直接送信に使用される接続です。一般的に使用されるsocketのポート番号のいくつかは、`9100` または `35` です。デバイスURI (Uniform Resource Identifier)の構文は、socket://`IP.of.the.printer`:`port` です(たとえば、`socket://192.168.2.202:9100/`)。

LPD (line printer daemon、ラインプリンタデーモン)
: LPDプロトコルについては、RFC 1179で説明されています。このプロトコルの下では、印刷キューのIDなど、一部のジョブ関連データが送信されてから、実際の印刷データが送信されます。したがって、LPDプロトコルの設定時には印刷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、プリント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。LPDサービスが使用するポート番号は`515`です。デバイスURIの例は、`lpd://192.168.2.202/LPT1`です。

IPP (Internet Printing Protocol、インターネット印刷プロトコル)
: IPPは、HTTPプロトコルに基づいた比較的新しい(1999年)プロトコルです。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。IPPを正しく設定するには、印刷キューの名前は必須です。IPPのポート番号は`631`です。デバイスURIの例は、`ipp://192.168.2.202/ps`および`ipp://192.168.2.202/printers/ps`です。

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

設定を行う前に、プリンタがサポートしているプロトコルを決定する必要があります。メーカーから必要な情報が提供されていない場合は、コマンド`nmap`(`nmap`パッケージに付属)を使用して、プロトコルを推定します。`nmap`はホストのオープンポートを確認します。例:

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

15.5 コマンドラインツールによるCUPS設定

CUPSは、`lpinfo`、`lpadmin`、`lpoptions`などのコマンドラインツールで設定できます。バックエンド(USBなど)とパラメータで構成されるデバイスURIが必要です。システム上の有効なデバイスURIを判断するには、`lpinfo -v | grep ":/"`コマンドを使用します。

```
# lpinfo -v | grep ":/"
direct usb://ACME/FunPrinter%20XL
network socket://192.168.2.253
```

`lpadmin`を使用すると、CUPSサーバ管理者は、印刷キューの追加、削除、または管理を実行できます。プリントキューを追加するには、次の構文を使用します。

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

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

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

```
lpadmin -p ps -v usb://ACME/FunPrinter%20XL -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

ネットワークプリンタの設定例:

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

`lpadmin`のオプションの詳細は、`lpadmin(8)`のマニュアルページを参照してください。

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

1. 最初に、すべてのオプションを列挙します。

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

   例:

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

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

2. 次のように`lpadmin`を使用してオプションを変更します。

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

3. 新しい設定値の確認:

   ```
   lpoptions -p queue -l

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

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

15.6 コマンドラインからの印刷

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

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

15.7 SUSE Linux Enterprise Serverの特別な機能

CUPSの複数の機能は、SUSE Linux Enterprise Server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。

15.7.1 CUPSとファイアウォール

デフォルトのSUSE Linux Enterprise Serverインストールを実行した後、SuSEfirewall2はアクティブになり、ネットワークインタフェースは着信トラフィックをブロックする**外部ゾーン**に設定されます。SuSEFirewall2の設定の詳細については、Book “Security Guide” 15 “Masquerading and Firewalls” 15.4 “SuSEFirewall2”および http://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings を参照してください。

15.7.1.1 CUPSクライアント

通常、CUPSクライアントはファイアウォール内部の信頼されるネットワーク環境の通常のワークステーションで実行されます。この場合、ネットワークインタフェースを**内部ゾーン**に設定し、ワークステーションにネットワーク内部から到達できるようにすることを推奨します。

15.7.1.2 CUPSサーバ

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

15.7.2 ネットワークプリンタの参照

CUPSサーバは、共有プリンタが利用可能かどうか、およびそのステータスをネットワーク上で定期的にアナウンスします。クライアントは、この情報にアクセスすることで、印刷ダイアログなどに利用可能なプリンタのリストを表示できます。これを「参照」と呼びます。

CUPSサーバでは、ネットワークを介して印刷キューをアナウンスする際に、従来のCUPS参照プロトコルまたはBonjour/DND-SDが使用されます。ネットワーク印刷キューを参照するには、`cups-browsed`サービスを、CUPSサーバを介して印刷するすべてのクライアントで実行する必要があります。`cups-browsed`は、デフォルトでは起動されません。アクティブなセッションでこのサービスを起動するには、`sudo systemctl start cups-browsed.service`を使用します。ブート後にこのサービスが自動的に起動されるようにするには、すべてのクライアントで`sudo systemctl enable cups-browsed.service`を実行してサービスを有効にします。

`cups-browsed`を起動しても参照できない場合は、CUPSサーバがBonjour/DND-SDを介してネットワーク印刷キューをアナウンスしている可能性があります。この場合、`avahi`パッケージを追加インストールし、すべてのクライアントに対して`sudo systemctl start avahi-daemon.service`を実行することで、関連するサービスを起動する必要があります。

15.7.3 各種パッケージ内のPPDファイル

YaSTのプリンタ環境設定では、`/usr/share/cups/model`にインストールされたPPDファイルを使用して、CUPSのキューがセットアップされます。プリンタモデルに適合するPPDファイルを見つけるため、YaSTはハードウェア検出時に判別されたベンダおよびモデルを、すべてのPPDファイル内のベンダおよびモデルと比較します。このために、YaSTのプリンタ環境設定機能は、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。

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。たとえば、PostScriptプリンタを使用している場合、その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`または`OpenPrintingPPDs-postscript`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。

追加のPPDファイルは次のパッケージで提供されています。

- gutenprint: Gutenprintドライバとそれに一致するPPD
- splix: SpliXドライバとそれに一致するPPD
- OpenPrintingPPDs-ghostscript: Ghostscriptの組み込みドライバ用PPD
- OpenPrintingPPDs-hpijs: HP以外のプリンタ向けのHPIJSドライバ用PPD

15.8 トラブルシューティング

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

15.8.1 標準的なプリンタ言語をサポートしないプリンタ

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

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

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

専有のLinuxドライバを機能させようと時間を費やす代わりに、標準プリンタ言語(PostScript推奨)をサポートするプリンタを購入する方が費用効率が高い場合があります。この方法により、ドライバの問題を一度で完全に解決できます。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。

15.8.2 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`パッケージまたは`OpenPrintingPPDs-postscript`パッケージに、PostScriptプリンタに適したPPDファイルが含まれていない場合は、プリンタメーカのドライバCDにあるPPDファイルを使用したり、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したりすることができます。

PPDファイルがzipアーカイブ(.zip)または自己展開zipアーカイブ(`.exe`)の形で提供されている場合、`unzip`を使用してそのファイルを展開します。最初に、PPDファイルのライセンス(許諾契約)条項を読みます。次に`cupstestppd`ユーティリティを使って、PPDファイルが「Adobe PostScript Printer Description File Format Specification, version 4.3」に準拠しているかどうかを確認します。「FAIL」ユーティリティから失敗が返された場合は、PPDファイル中のエラーは深刻なもので、問題を引き起こす可能性があります。`cupstestppd`によって報告された問題点は、取り除く必要があります。必要に応じて、適切なPPDファイルが入手できるかどうかをプリンタメーカに問い合わせることも考えられます。

15.8.3 ネットワークプリンタ接続

ネットワークの問題の識別

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

TCP/IPネットワークの確認

TCP/IPネットワークと名前解決が正しく機能していることが必要です。

リモート`lpd`の確認

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

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

`lpd`への接続を確立できない場合、`lpd`がアクティブになっていないか、ネットワークの基本的な問題があります。
`root`ユーザで次のコマンドを使用し、リモート`host`上の`queue`に関するステータスレポート(おそらく、非常に長い)を照会することもできます。これは、該当の`lpd`がアクティブで、そのホストが照会を受け付けることを前提にしています。

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

`lpd`が応答しない場合、それがアクティブになっていないか、ネットワークの基本的な問題が発生している可能性があります。`lpd`が応答する場合、その応答は、`host`上にある`queue`を介して印刷ができない理由を示すはずです。例15.1「`lpd`からのエラーメッセージ」で示す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。

例 15.1 `lpd`からのエラーメッセージ

```
lpd: your host does not have line printer access
lpd: queue does not exist
printer: spooling disabled
printer: printing disabled
```

リモート`cupsd`の確認

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

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

ブロードキャストを行っているCUPSネットワークサーバが存在している場合、出力は例15.2「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。

例 15.2 CUPSネットワークサーバからのブロードキャスト

```
ipp://192.168.2.202:631/printers/queue
```

System z IBM System zのEthernetデバイスは、デフォルトではブロードキャストを受信しないことを考慮してください。
次のコマンドを使用して、`host`上の`cupsd` (ポート`631`)に対するTCP接続を確立できるかどうかをテストすることができます。

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

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

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

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

ネットワークプリンタまたは印刷サーバボックスのトラブルシューティング

印刷サーバボックス上のスプーラは時々、複数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。これはプリントサーバボックスのスプーラで発生するため、この問題を解決する方法はありません。回避策として、TCPソケットを使用して、プリントサーバボックスに接続されているプリンタに直接送信することで、プリントサーバボックス内のスプーラを使用しないようにします。詳細については、15.4項「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。プリンタがプリント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、プリント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` 内に記述されている複数の一般的な既知のポートだけを確認します。可能性のある

すべてのポートをチェックするには、`nmap -p from_port-to_portIP-address`コマンドを使用します。これは、ある程度の時間を要することがあります。詳細な情報については、`nmap`のマニュアルページを参照してください。
次のようなコマンドを入力します。

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

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

15.8.4 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)、印刷システムはこれを検出しません。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PPDファイルを選択します。

15.8.5 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と報告される前に、バックエンドは何回の試行の失敗が妥当であるかを判断します。それ以上の試行は無駄に終わる可能性があるので、`cupsd`はそれぞれのキューの印刷を無効にします。問題の原因を取り除いた後、システム管理者は`cupsenable`コマンドを使用して、印刷を再度有効にする必要があります。

15.8.6 CUPS参照:印刷ジョブの削除

CUPSネットワークサーバが参照機能を使用して自らのキューをクライアントホストへブロードキャストし、クライアントホスト側で適切なローカル`cupsd`がアクティブになっている場合、クライアント側の`cupsd`はアプリケーションから印刷ジョブを受け付け、サーバ側の`cupsd`へそれらを転送します。サーバ上で`cupsd`が印刷ジョブを受け付けると、そのジョブには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で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時点で、その印刷ジョブは完了したと考えるからです。

サーバ上にあるプリントジョブを削除したい場合、`lpstat -h cups.example.com -o`などのコマンドを使用してサーバ上のジョブ番号を判断します。サーバがまだそのプリントジョブを完了していない(つまり、プリンタへ完全に送信していない)ことが前提条件です。このジョブ番号を使用して、サーバ上にある印刷ジョブを削除できます。

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

15.8.7 異常な印刷ジョブとデータ転送エラー

印刷プロセス中にプリンタの電源を切ったり、コンピュータをシャットダウンすると、印刷ジョブはキュー内に残ります。コンピュータ(またはプリンタ)の電源を再度投入すると、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が印刷されることがあります。この状況を修正するには、次の手順に従います。

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 cups.example.com -o`を使用して、どのキューが現在印刷に使用されているかを確認します。`cancel queue-jobnumber`または`cancel -h cups.example.com queue-jobnumber`を使用して、該当のプリント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ているにもかかわらず、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。
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。

15.8.8 CUPSのデバッグ

CUPSの問題を特定するために、次の一般的な手順を実行します。

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

15.8.9 詳細情報

SUSE Linuxでの印刷の詳細については、openSUSE Support Database (http://en.opensuse.org/Portal:Printing)にアクセスしてください。SUSE Knowledgebase (http://www.suse.com/support/)では、さまざまな個別の問題のソリューションが紹介されています。`CUPS` のテキスト検索機能により関連する記事を見つけてください。

16 udevによる動的カーネルデバイス管理

カーネルは、実行中のシステムのほぼすべてのデバイスを追加または削除できます。デバイス状態の変更(デバイスが接続されているか、または取り外されたか)をユーザスペースに反映させる必要があります。デバイスは、接続後、検出されるとすぐに設定されなければなりません。特定のデバイスのユーザは、このデバイスの認識された状態が変更された場合は通知される必要があります。udevは、/dev ディレクトリのデバイスノートファイルおよびシンボリックリンクを動的に維持するために必要なインフラストラクチャを提供します。udev規則は、外部ツールをカーネルデバイスイベント処理に接続する方法を提供します。これにより、カーネルデバイス処理の一部として実行する特定のスクリプトを追加するなど、udev デバイス処理をカスタマイズしたり、デバイス処理中に評価する追加データを要求およびインポートしたりできます。

16.1 /devディレクトリ

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

/dev ディレクトリのコンテンツは一時的なファイルシステム内で管理され、すべてのファイルはシステムの起動時にレンダリングされます。意図的に、手動で作成または変更されたファイルはリブート時に復元されません。対応するカーネルデバイスの状態にかかわらず、/dev ディレクトリ内に常駐する静的ファイルおよびディレクトリは、systemd-tmpfilesで作成できます。環境設定ファイルは、/usr/lib/tmpfiles.d/ および /etc/tmpfiles.d/ にあります。詳細については、systemd-tmpfiles(8) のマニュアルページを参照してください。

16.2 カーネルのueventとudev

必要なデバイス情報は、sysfs ファイルシステムによってエクスポートされます。カーネルが検出および初期化するすべてのデバイスについて、そのデバイス名を含んだディレクトリが作成されます。このディレクトリには、デバイス固有のプロパティのある属性ファイルが含まれます。

デバイスが追加または削除されるたびに、カーネルはueventを送信して、udev に変更を通知します。udev デーモンは、起動時に1回、 /etc/udev/rules.d/*.rules ファイルから提示されたすべてのルールを読み込んで解析し、メモリ内に保存します。ルールファイルを変更、追加、または削除した

場合、udevadm control reload_rulesコマンドを実行することで、このデーモンは、メモリ内のすべてのルールを再ロードできます。udevのルールとそれらの構文の詳細については、16.6項「udevルールによるカーネルデバイスイベント処理への影響」を参照してください。

受信したすべてイベントは、提供されている一連のルールに照らして照合されます。ルールによって、イベント環境キーを追加または変更したり、作成するデバイスノードに特定の名前を要求したり、ノードを指すシンボリックリンクを追加したり、またはデバイスノードの作成後に実行するプログラムを追加したりできます。ドライバのコアueventは、カーネルのネットリンクソケットから受信されます。

16.3 ドライバ、カーネルモジュールおよびデバイス

カーネルバスドライバは、デバイスを検出します。検出されたデバイスごとに、カーネルは内部デバイス構造を作成し、ドライバコアは、ueventをudevデーモンに送信します。バスデバイスは、デバイスの種類を示す特別な形式のIDを識別します。通常、これらのIDは、ベンダー、製品IDおよびサブシステム固有の値で構成されています。各バスには、これらのIDに対してMODALIASという独自のスキームを持ちます。カーネルは、デバイス情報を読み取り、この情報からMODALIAS ID文字列を作成し、イベントとともに文字列を送信します。USBマウスの場合、次のようになります。

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

各デバイスドライバは、既知の処理可能デバイスのエイリアスのリストを持ちます。このリストは、カーネルモジュールファイル自体にも含まれています。depmodプログラムは、IDリストを読み取り、現在使用可能なすべてのモジュールについて、カーネルの/lib/modulesディレクトリ内にmodules.aliasを作成します。このインフラストラクチャにより、MODALIASキーを持つイベントごとにmodprobeを呼び出すだけで簡単にモジュールをロードできます。modprobe $MODALIASが呼び出されると、そのデバイスに付けられたデバイスエイリアスとモジュールによって提示されるエイリアスとが一致します。一致したエントリが見つかると、そのモジュールがロードされます。これはすべてudevによって自動的にトリガされます。

16.4 ブートおよび初期デバイスセットアップ

udevデーモンが実行される前のブートプロセスで発生するすべてのデバイスイベントは失われます。これは、これらのイベントを処理するインフラストラクチャがルートファイルシステムに常駐し、その時点で使用できないからです。その消失の埋め合せに、カーネルは、sysfsファイルシステム内の各デバイスのデバイスディレクトリにueventファイルを生成します。そのファイルにaddと書き込むことにより、

カーネルは、ブート時に消失したものと同じイベントを再送します。/sys 内のすべての uevent ファイルを含む単純なループにより、すべてのイベントが再びデバイスノードを作成し、デバイスセットアップを実行します。

たとえば、ブート時に存在するUSBマウスは、ドライバがその時点で使用できないため、初期のブートロジックでは初期化されない場合があります。デバイス検出イベントは、消失し、そのデバイスのカーネルモジュールは検出されません。接続されている可能性のあるデバイスを手動で検索する代わりに、ルートファイルシステムが使用可能になった後で、udev がカーネルにすべてのデバイスイベントを要求します。これにより、USBマウスデバイスのイベントが再び実行されます。これで、マウントされたrootファイルシステム上のカーネルモジュールが検出され、USBマウスを初期化できます。

ユーザスペースでは、実行時のデバイスのcoldplugシーケンスとデバイス検出との間に明らかな違いはありません。どちらの場合も、同じルールを使用して一致検出が行われ、設定された同じプログラムが実行されます。

16.5 実行中のudevデーモンの監視

udevadm monitor プログラムを使用すると、ドライバのコアイベントとudevイベントプロセスのタイミングをビジュアル化できます。

```
UEVENT[1185238505.276660] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1 (usb)
UDEV  [1185238505.279198] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1 (usb)
UEVENT[1185238505.279527] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0
 (usb)
UDEV  [1185238505.285573] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0
 (usb)
UEVENT[1185238505.298878] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/
input/input10 (input)
UDEV  [1185238505.305026] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/
input/input10 (input)
UEVENT[1185238505.305442] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/
input/input10/mouse2 (input)
UEVENT[1185238505.306440] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/
input/input10/event4 (input)
UDEV  [1185238505.325384] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/
input/input10/event4 (input)
UDEV  [1185238505.342257] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/
input/input10/mouse2 (input)
```

UEVENT 行は、カーネルがnetlinkで送信したイベントを示します。UDEV 行は、完了したudevイベントハンドラを示します。タイミングは、マイクロ秒で出力されます。UEVENT および UDEV 間の時間は、udevがこのイベントの処理に要した時間、またはudevデーモンがこのイベントと関連する実行中のイベントとの同期の実行に遅れた時間です。たとえば、パーティションイベントは、メインディスクイベントがハードウェアに問い合わせたデータに依存する可能性があるため、ハードディスクパーティションのイベントは常に、メインデバイスイベントが完了するのを待ちます。

udevadm monitor --env は、完全なイベント環境を表示します。

```
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw
```

udevは、syslogにもメッセージを送信します。 どのメッセージをsyslogに送信するかを左右するデフォルトのsyslog優先度は、udev 設定ファイル /etc/udev/udev.conf で指定されています。実行中のデーモンのログ優先度は、udevadm control log_priority=level/number で変更できます。

16.6 udevルールによるカーネルデバイスイベント処理への影響

udev ルールは、カーネルがイベント自体に追加する任意のプロパティや、カーネルが sysfs にエクスポートする任意の情報と一致することができます。また、この規則で、外部プログラムからの追加情報を要求することもできます。各イベントは、指定されたすべての規則と一致します。すべての規則は、/etc/udev/rules.d ディレクトリにあります。

規則ファイル内の各行には、少なくとも1つのキー値ペアが含まれています。これらは、一致と割り当てキーという2種類のキーです。すべての一致キーが各値と一致する場合、その規則が適用され、割り当てキーに指定された値が割り当てられます。一致するルールがある場合、デバイスノードの名前を指定、ノードを指すシンボリックリンクを追加、またはイベント処理の一部として指定されたプログラムを実

行できます。一致するルールがない場合、デフォルトのデバイスノード名を使用して、デバイスノードが作成されます。ルールの構文とデータの一致またはインポート用に提供されているキーの詳細については、udev のマニュアルページで説明されています。以下に示すルール例では、udev ルール構文の基本を紹介します。これらのルール例は、すべて、/etc/udev/rules.d/50-udev-default.rules の下にある udev デフォルトルールセットに含まれています。

例 16.1 udevルールの例

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb/%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

console ルールは、3つのキーで構成されています。その内訳は、一致キーが1つ(KERNEL)、割り当てキーが2つ(MODE 、 OPTIONS)です。KERNEL 一致ルールは console タイプのアイテムをデバイスリストから検索します。正確な一致だけが有効であり、このルールの実行をトリガします。MODE キーは、特別パーミッションをデバイスノードに割り当てます。この例では、読み取り/書き込みパーミッションをこのデバイスの所有者にのみ割り当てます。OPTIONS キーは、この規則をこのタイプのデバイスに適用される最後の規則にします。以降の規則は、この特定デバイスタイプとマッチしても、どのような結果も生じません。

serial devices ルールは、50-udev-default.rules には存在しなくなりましたが、依然その知識は重要です。この規則は、2つの一致キー(KERNEL 、 ATTRS)および1つの割り当てキー(SYMLINK)で構成されます。KERNEL キーは、ttyUSB タイプのすべてのデバイスを検索します。このキーで * ワイルドカードを使用すると、これらのデバイスのいくつかとマッチします。2つ目の一致キー ATTRS は、ttyUSB デバイスの sysfs にある product 属性ファイルに一定の文字列が含まれているかどうかをチェックします。割り当てキー(SYMLINK)は、/dev/pilot の下に、このデバイスへのシンボリックリンクを追加します。このキーで演算子(+=)を使用すると、前/後の規則が他のシンボリックリンクを追加した場合でも、udevはこの操作を追加実行します。この規則は、2つの一致キーを含むので、両方の条件が満たされる場合のみ適用されます。

printer ルールは、USBプリンタを対象とし、2つの一致キー(SUBSYSTEM 、 KERNEL)を含みます。規則全体を適用するには、これらのキーを両方とも適用する必要があります。3つの割り当てキーは、このデバイスタイプの名前付け(NAME)、シンボリックデバイスリンクの作成、(SYMLINK)、およびこのデバイスタイプのグループメンバーシップ(GROUP)を処理します。 KERNEL キーで * ワイルドカードを使用すると、いくつかの lp プリンタデバイスとマッチします。 NAME および SYMLINK の両キーで置き換えを使用すると、これらの文字列を内部デバイス名で拡張できます。たとえば、最初の lp USBプリンタへのシンボリックリンクは /dev/usblp0 になります。

kernel firmware loader ルールでは、ランタイム時の外部ヘルパースクリプトで、udevが追加ファームウェアをロードします。 SUBSYSTEM 一致キーは、 firmware サブシステムを検索します。 ACTION キーは、 firmware サブシステムに属するデバイスが追加されているかどうかをチェックします。 RUN+= キーは、 firmware.sh スクリプトの実行をトリガして、ファームウェアを見つけます。

すべての規則に共通する一般的特性は次のとおりです。

- 各規則は、カンマで区切られた1つ以上のキー値ペアで構成されます。
- キーの動作は、演算子で決定されます。udevルールは、いくつかの異なる演算子をサポートします。
- 指定する各値は、引用符で囲む必要があります。
- 規則ファイルの各行が1つの規則に相当します。規則が1行を超える場合は、shell構文のように、 \ を使用して異なる行を結合してください。
- udev ルールは、shell型のパターンをサポートします。このパターンは、 * 、 ? 、および [] の各パターンとマッチします。
- udevルールは、置換をサポートします。

16.6.1 udevルールでの演算子の使用

キーを作成する場合は、作成するキーのタイプによって、いくつかの異なる演算子から選択できます。一致キーは、通常、検索値に一致するか、明示的に一致しない値を見つけるために使用されます。一致キーは、次の演算子のいずれかを含みます。

==

等価の比較。キーに検索パターンが含まれている場合は、そのパターンと一致するすべての結果が有効です。

`!=`
: 非等価の比較。キーに検索パターンが含まれている場合は、そのパターンと一致するすべての結果が有効です。

割り当てキーでは、次のどの演算子でも使用できます。

`=`
: 値をキーに割り当てます。すでに値のリストで構成されているキーはリセットされ、指定した1つの値だけが割り当てられます。

`+=`
: エントリのリストを含むキーに値を追加します。

`:=`
: 最終値を割り当てます。以降の規則による変更は許可されません。

16.6.2 udevルールでの置換の使用

udevルールは、プレースホルダと置換の使用をサポートします。それらは、他のスクリプトでの使用と同様な方法で使用します。`udev` ルールでは、次の置換を使用できます。

`%r`、`$root`
: デフォルトのデバイスディレクトリ `/dev`。

`%p`、`$devpath`
: `DEVPATH` の値。

`%k`、`$kernel`
: `KERNEL` の値または内部デバイス名。

`%n`、`$number`
: デバイス番号。

`%N`、`$tempnode`
: デバイスファイルの一時名。

`%M`、`$major`
: デバイスのメジャー番号。

`%m`、`$minor`
: デバイスのマイナー番号。

`%s{attribute}`、`$attr{attribute}`
: `sysfs`属性の値(`attribute`で指定)。

`%E{variable}`、`$attr{variable}`
: 環境変数の値(`variable`で指定)。

`%c`、`$result`
: `PROGRAM`の出力。

`%%`
: `%`文字。

`$$`
: `$`文字。

16.6.3 udev一致キーの使用

一致キーは、`udev`ルールの適用前に満たす必要のある条件を記述します。次の一致キーが使用可能です。

`ACTION`
: イベント動作の名前。たとえば、`add`または`remove`(デバイスの追加または削除の場合)。

`DEVPATH`
: イベントデバイスのデバイスパス。たとえば、`DEVPATH=/bus/pci/drivers/ipw3945`(ipw3945ドライバに関連するすべてのイベントを検索する場合)。

`KERNEL`
: イベントデバイスの内部(カーネル)名。

`SUBSYSTEM`
: イベントデバイスのサブシステム。たとえば、`SUBSYSTEM=usb`(USBデバイスに関連するすべてのイベント用)。

`ATTR{filename}`
: イベントデバイスのsysfs属性。 `vendor`属性ファイル名に含まれた文字列とマッチするには、たとえば、`ATTR{vendor}=="On[sS]tream"`を使用できます。

`KERNELS`
: `udev`にデバイスパスを上方に検索させ、一致するデバイス名を見つけます。

SUBSYSTEMS

udevにデバイスパスを上方に検索させ、一致するデバイスサブシステム名を見つけます。

DRIVERS

udevにデバイスパスを上方に検索させ、一致するデバイスドライバ名を見つけます。

ATTRS{filename}

udevにデバイスパスを上方に検索させ、一致するsysfs属性値を持つデバイスを見つけます。

ENV{key}

環境変数の値。たとえば、ENV{ID_BUS}="ieee1394でFireWire bus IDに関連するすべてのイベントを検索します。

PROGRAM

udevに外部プログラムを実行させます。成功の場合は、プログラムが終了コードとしてゼロを返します。プログラムの出力はSTDOUTに送られ、RESULTキーで使用できます。

RESULT

最後のPROGRAM呼び出しの出力文字列とマッチします。このキーは、PROGRAMキーと同じ規則に含めるか、それ以降のキーに含めてください。

16.6.4 udev割り当てキーの使用

上記で説明した一致キーに対し、割り当てキーでは満たすべき条件を記述しません。値、名前、アクションをudevが保守するデバイスノードに割り当てます。

NAME

作成するデバイスノードの名前。いったんルールでノード名が設定されると、このノードのNAMEキーを持つ他のルールはすべて無視されます。

SYMLINK

作成するノードに関連するシンボリックリンクの名前。複数の一致ルールで、デバイスノードとともに作成するシンボリックリンクを追加できます。1つのルール内で、スペース文字でシンボリックリンク名を区切ることで、1つのノードに複数のシンボリックリンクを指定することもできます。

OWNER、GROUP、MODE

新しいデバイスノードのパーミッションここで指定する値は、すでにコンパイルされている値を上書きします。

ATTR{key}

イベントデバイスの sysfs 属性に書き込む値を指定します。== 演算子を使用すると、このキーは、sysfs属性の値とのマッチングにも使用されます。

ENV{key}

環境への変数のエクスポートを udev に指示します。== 演算子を指定すると、このキーは、環境変数とのマッチングにも使用されます。

RUN

このデバイスに対して実行されるプログラムのリストにプログラムを追加するように、udev に指示します。このデバイスのイベントをブロックしないようにするため、これは非常に短いタスクに限定してください。

LABEL

GOTO のジャンプ先にするラベルを追加します。

GOTO

いくつかのルールをスキップし、GOTO キーで参照されるラベルを含むルールから続行するように、udev に指示します。

IMPORT{type}

変数をイベント環境(外部プログラムの出力など)にロードします。udevは、いくつかの異なるタイプの変数をインポートします。 タイプが指定されていない場合、udev は、ファイルパーミションの実行可能ビットに基づいてタイプを決定しようとします。

- program - 外部プログラムを実行し、その出力をインポートします。
- file - テキストファイルをインポートします。
- parent - 親デバイスから保存されたキーをインポートします。

WAIT_FOR_SYSFS

一定のデバイスに指定された sysfs ファイルが作成されるまで、udev を待機させます。たとえば、WAIT_FOR_SYSFS="ioerr_cnt" では、ioerr_cnt ファイルが作成されるまで、udev を待機させます。

オプション

OPTION キーには、次の可能な値があります。

- last_rule - 以降のすべての規則を無視します。
- ignore_device - このイベントを完全に無視します。

- `ignore_remove` - このデバイスの以降のすべての削除イベントを無視します。
- `all_partitions` - ブロックデバイス上のすべての使用可能なパーティションにデバイスノードを作成します。

16.7 永続的なデバイス名の使用

動的デバイスディレクトリおよび `udev` ルールインフラストラクチャによって、認識順序やデバイスの接続手段にかかわらず、すべてのディスクデバイスに一定の名前を指定できるようになりました。カーネルが作成する適切なブロックデバイスはすべて、特定のバス、ドライブタイプまたはファイルシステムに関する特別な知識を備えたツールによって診断されます。動的カーネルによって指定されるデバイスノード名とともに、`udev` は、デバイスをポイントする永続的なシンボリックリンクのクラスを維持します。

```
/dev/disk
|-- by-id
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B -> ../../sda
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part1 -> ../../sda1
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part6 -> ../../sda6
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part7 -> ../../sda7
|   |-- usb-Generic_STORAGE_DEVICE_02773 -> ../../sdd
|   `-- usb-Generic_STORAGE_DEVICE_02773-part1 -> ../../sdd1
|-- by-label
|   |-- Photos -> ../../sdd1
|   |-- SUSE10 -> ../../sda7
|   `-- devel -> ../../sda6
|-- by-path
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part6 -> ../../sda6
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part7 -> ../../sda7
|   |-- pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sr0
|   |-- usb-02773:0:0:2 -> ../../sdd
|   |-- usb-02773:0:0:2-part1 -> ../../sdd1
`-- by-uuid
    |-- 159a47a4-e6e6-40be-a757-a629991479ae -> ../../sda7
    |-- 3e999973-00c9-4917-9442-b7633bd95b9e -> ../../sda6
    `-- 4210-8F8C -> ../../sdd1
```

16.8 udevで使用するファイル

`/sys/*`

Linuxカーネルによって提供される仮想ファイルシステム。現在知られているデバイスをすべてエクスポートします。この情報は、`udev` が使用して `/dev` 内にデバイスノードを作成します。

`/dev/*`

動的に作成されたデバイスノード、およびsystemd-tmpfilesで作成された静的コンテンツ。詳細については、`systemd-tmpfiles(8)` のマニュアルページを参照してください。

以下のファイルおよびディレクトリには、`udev` インフラストラクチャの重要な要素が含まれています。

`/etc/udev/udev.conf`

メイン `udev` 設定ファイル

`/etc/udev/rules.d/*`

規則と一致するudevイベント.

`/usr/lib/tmpfiles.d/` および `/etc/tmpfiles.d/`

静的 `/dev` コンテンツを管理します。

`/usr/lib/udev/*`

`udev` ルールから呼び出されるヘルパープログラム

16.9 詳細情報

`udev` インフラストラクチャの詳細については、以下のマニュアルページを参照してください。

`udev`

`udev` 、キー、ルールなどの重要な設定課題に関する一般情報

`udevadm`

`udevadm` は、`udev` のランタイム動作を制御し、カーネルイベントを要求し、イベントキューを管理し、簡単なデバッグメカニズムを提供します。

`udevd`

`udev` イベント管理デーモンに関する情報

17 X Windowシステム

Xウィンドウシステム(X11)は、UNIX系のグラフィカルユーザインタフェースで、事実上の標準となっています。Xはネットワークベースであり、あるホスト上で起動されたアプリケーションを、任意のネットワーク(LANやインターネット)を介して接続されている他のホスト上で表示できるようにします。この章では、X設定の基本情報と、SUSE® Linux Enterprise Serverでのフォント使用の背景情報を提供します。

ほとんどの場合、X Windowシステムでは設定をする必要がまったくありません。ハードウェアは、Xの起動時に動的に検出されるため、`xorg.conf`の使用はお勧めしません。それでも、Xの動作を変更するためにカスタムオプションを指定する必要がある場合は、`/etc/X11/xorg.conf.d/`にある設定ファイルを変更できます。

ヒント: IBM System z: グラフィカルユーザインタフェースの設定

IBM System zには、X.Orgがサポートする入出力デバイスはありません。そのため、このセクションで取り上げられている設定手順は使用できません。IBM System zの詳細については、ブック「導入ガイド」4「IBM System zでのインストール」を参照してください。

17.1 フォントのインストールと設定

Linuxのフォントは次の2つに分類できます。

アウトラインフォントまたはベクトルフォント

グリフの形状に関する描画命令として数学的記述が含まれています。このため、品質を損なうことなく各グリフを任意のサイズに拡大縮小できます。このようなフォント(グリフ)を使用するには、数学的記述をラスタ(グリッド)に変換する必要があります。このプロセスを「フォントのラスタライズ」と呼びます。「フォントヒンティング」(フォント内に組み込まれている)は、特定のサイズのレンダリング結果を向上および最適化します。ラスタライズとヒンティングは、FreeTypeライブラリによって行われます。

Linuxで一般的な形式は、PostScript Type 1とType 2、TrueType、およびOpenTypeです。

ビットマップフォントまたはラスタフォント

特定のフォントサイズ用にデザインされたピクセルの配列で構成されます。ビットマップフォントは非常に高速でレンダリングも容易です。ただし、ベクトルフォントと比較した場合、ビットマップフォントは品質を損なわずに拡大縮小することはできません。そのため、これらのフォントは通常、複数のサイズで配布されます。現在でも、Linuxコンソールや一部の端末ではビットマップフォントが使用されています。

Linuxでは、PCF (Portable Compiled Format)またはBDF (Glyph Bitmap Distribution Format)が最も一般的な形式です。

これらのフォントの外観は、主に次の2つの側面による影響を受けます。

- 適切なフォントファミリを選択する
- ユーザが読みやすい結果を実現するアルゴリズムでフォントをレンダリングする

最後の点は、ベクトルフォントにのみ関係があります。上述の2つの点は主観に大きく左右されますが、何らかのデフォルト値を作成する必要があります。

Linuxのフォントレンダリングシステムは、異なる関係を持つ複数のライブラリで構成されます。基本のフォントレンダリングライブラリはFreeType (http://www.freetype.org/)で、サポートされている形式のフォントグリフを最適化されたビットマップグリフに変換します。レンダリングプロセスはアルゴリズムとそのパラメータによって制御されます(特許の問題が絡む場合があります)。

FreeTypeを使用するすべてのプログラムまたはライブラリは、Fontconfig (http://www.fontconfig.org/)ライブラリを参照する必要があります。このライブラリは、ユーザとシステムからフォント設定を収集します。ユーザが自分のFontconfig設定を修正した場合、このような変更によってアプリケーションはFontconfig対応になります。

アラビア語、ハン語、パスパ文字などのスクリプトに必要なより高度なOpenTypeシェーピング、およびその他のより高レベルのテキスト処理は、Harfbuzz (http://www.harfbuzz.org/)やPango (http://www.pango.org/)などが行います。

17.1.1 インストール済みフォントの表示

システムにインストールされているフォントの概要を表示するには、`rpm`コマンドまたは`fc-list`コマンドを使用します。どちらのコマンドでも適切な回答が得られますが、システムおよびユーザの設定によっては異なるリストが返されることがあります。

`rpm`

システムにインストールされている、フォントが格納されたソフトウェアパッケージを参照するには、`rpm`を起動します。

```
rpm -qa '*fonts*'
```

すべてのフォントパッケージがこの式を満たす必要があります。ただし、このコマンドは、`fonts-config`のような誤検知を返す場合があります(これはフォントではなく、フォントも含みません)。

`fc-list`

アクセスできるフォントファミリ、およびそれらのフォントがシステムまたはホームのどちらにインストールされているかに関する概要を参照するには、`fc-list`を起動します。

```
fc-list ':' family
```

注記: `fc-list`コマンド

`fc-list`コマンドは、Fontconfigライブラリのラッパーです。Fontconfig (正確にはそのキャッシュ)に対して、多くの有用な情報を問い合わせることができます。詳細については、`man 1 fc-list`を参照してください。

17.1.2 フォントの表示

インストールされているフォントファミリのデザインを知りたい場合は、`ftview`コマンド(`ft2demos`パッケージ)を使用するか、http://fontinfo.opensuse.org/ にアクセスします。たとえば、FreeMonoフォントを14ポイントで表示するには、`ftview`を次のように使用します。

```
ftview 14 /usr/share/fonts/truetype/FreeMono.ttf
```

さらに詳しい情報が必要な場合は、http://fontinfo.opensuse.org/ にアクセスして、サポートされているスタイル(通常のフォント、太字、斜体など)と言語を参照します。

17.1.3 フォントの問い合わせ

パターンを指定した場合にどのフォントが使用されるかを問い合わせるには、`fc-match`コマンドを使用します。

たとえば、インストール済みのフォントをパターンに含めると、`fc-match`は、ファイル名、フォントファミリ、およびスタイルを返します。

```
tux > fc-match 'Liberation Serif'
LiberationSerif-Regular.ttf: "Liberation Serif" "Regular"
```

目的のフォントがシステムに存在しない場合は、Fontconfigの照合ルールが実行され、利用可能なフォントの中で最もそのフォントに似ているフォントを見つけようとします。つまり、要求は次のように置換されます。

```
tux > fc-match 'Foo Family'
DejaVuSans.ttf: "DejaVu Sans" "Book"
```

Fontconfigは「エイリアス」をサポートしており、名前は別のファミリ名に置換されます。代表的な例は、「sans-serif」、「serif」、「monospace」などの汎用名です。これらのエイリアスは、実際のファミリ名で置換することも、ファミリ名の優先リストで置換することもできます。

```
tux > for font in serif sans mono; do fc-match "$font" ; done
DejaVuSerif.ttf: "DejaVu Serif" "Book"
DejaVuSans.ttf: "DejaVu Sans" "Book"
DejaVuSansMono.ttf: "DejaVu Sans Mono" "Book"
```

現在インストールされているフォントによっては、使用中のシステムでの結果は異なる場合があります。

注記: Fontconfigに従った類似性ルール

Fontconfigは、指定された要求に従って「常に」、できる限り類似性の高い実際のファミリを返します(少なくともファミリが1つインストールされている場合)。「類似性」は、Fontconfigの内部メトリクスと、ユーザまたは管理者のFontconfig設定に依存します。

17.1.4 フォントのインストール

新しいフォントをインストールする主な方法は次のとおりです。

1. `*.ttf`や`*.otf`などのフォントファイルを既知のフォントディレクトリに手動でインストールする。システム全体で使用できるようにする場合は、標準のディレクトリ`/usr/share/fonts`を使用します。自分のホームディレクトリにインストールする場合は、`~/.config/fonts`を使用します。標準のディレクトリ以外を使用する場合は、Fontconfigで別のディレクトリを選択できます。Fontconfigにディレクトリを認識させるには、`<dir>`要素を使用します。詳細については、17.1.5.2項「Fontconfig XMLの詳細」を参照してください。

2. `zypper`を使用してフォントをインストールする。SUSEディストリビューションであっても、M17N:fonts (http://download.opensuse.org/repositories/M17N:/fonts/)リポジトリであっても、多くのフォントはすでにパッケージとして利用可能です。次のコマンドを使用して、リポジトリをリストに追加します。たとえば、SLE 12にリポジトリを追加するには、次の手順に従います。

   ```
   sudo zypper ar
   ```

```
    http://download.opensuse.org/repositories/M17N:/fonts/SLE_12/
M17N:fonts.repo
```

`FONT_FAMILY_NAME`を検索するには、次のコマンドを使用します。

```
sudo zypper se 'FONT_FAMILY_NAME*fonts'
```

17.1.5 フォントの外観の設定

レンダリングメディアおよびフォントサイズによっては、満足できる結果が得られないことがあります。たとえば、近年の平均的なモニタの解像度は100dpiであるため、ピクセルが大きくなりすぎ、グリフが綺麗に表示されません。

アンチエイリアス(グレースケールスムージング)、ヒンティング(グリッドフィッティング)、またはサブピクセルレンダリング(1方向の解像度を3倍にする)など、低解像度に対応するアルゴリズムはいくつもあります。これらのアルゴリズムはフォントの形式によっても異なることがあります。

重要: サブピクセルレンダリングの特許の問題

サブピクセルレンダリングはSUSEディストリビューションでは使用されていません。FreeType2はこのアルゴリズムをサポートしていますが、このアルゴリズムは、2019年末に有効期限が切れる複数の特許で保護されています。したがって、サブピクセルレンダリングがコンパイルされたFreeType2ライブラリがシステムにない限り、Fontconfigのサブピクセルレンダリングオプションを設定しても効果はありません。

Fontconfigでは、レンダリングアルゴリズムをすべてのフォントに対して個別に選択することも、フォントのセットに対して選択することもできます。

17.1.5.1 sysconfigによるフォントの設定

SUSE Linux Enterprise Serverには、Fontconfig上に`sysconfig`層があります。これは、フォント設定を試してみる場合の開始点として便利です。デフォルト設定を変更するには、設定ファイル`/etc/sysconfig/fonts-config`を編集します(またはYaST sysconfigモジュールを使用します)。ファイルの編集後、`fonts-config`を実行します。

```
sudo /usr/sbin/fonts-config
```

アプリケーションを再起動して結果を表示します。次の点に注意してください。

- 一部のアプリケーションでは再起動は必要ありません。たとえば、Firefoxは随時Fontconfig設定を再読み込みします。新たに作成したタブや再ロードしたタブには、新しいフォント設定が後で適用されます。
- パッケージをインストールまたは削除するたびに`fonts-config`スクリプトが自動的に呼び出されます(呼び出されない場合は、フォントソフトウェアパッケージのバグです)。
- `fonts-config`コマンドラインオプションで、すべてのsysconfig変数を一時的に上書きできます。詳細については、`fonts-config --help`を参照してください。

いくつかのsysconfig変数は変更することができます。`man 1 fonts-config`またはYaST sysconfigモジュールのヘルプを参照してください。次に、変数の例を示します。

レンダリングアルゴリズムの使用方法

検討対象: `FORCE_HINTSTYLE`、`FORCE_AUTOHINT`、`FORCE_BW`、`FORCE_BW_MONOSPACE`、`USE_EMBEDDED_BITMAPS`、および`EMBEDDED_BITMAP_LANGAGES`

汎用エイリアスの優先リスト

使用対象: `PREFER_SANS_FAMILIES`、`PREFER_SERIF_FAMILIES`、`PREFER_MONO_FAMILIES`、および`SEARCH_METRIC_COMPATIBLE`

次のリストは設定例を示しています。これは「最も読みやすい」フォント(コントラストが高い)から「最も美しい」フォント(スムージングが強い)の順にソートされています。

ビットマップフォント

ビットマップフォントを優先させる場合は、`PREFER_*_FAMILIES`変数を使用します。これらの変数については、ヘルプセクションの例に従ってください。これらのフォントは白黒でレンダリングされスムージングされない点、およびビットマップフォントはいくつかのサイズしか用意されていない点に注意してください。次の設定

```
SEARCH_METRIC_COMPATIBLE="no"
```

を使用して、メトリック互換性主導型のファミリ名の置換を無効にすることを検討します。

白黒にレンダリングされるスケーラブルフォント

アンチエイリアスなしでレンダリングされるスケーラブルフォントは、ビットマップフォントと同様の結果になる可能性がありますが、フォントの拡大縮小機能は維持されます。Liberationファミリのような適切にヒンティングされたフォントを使用します。ただし、残念ながら、適切にヒンティングされたフォントは多くありません。この方法を強制するには、次の変数を設定します。

```
FORCE_BW="yes"
```

白黒にレンダリングされる等幅フォント

等幅フォントは、アンチエイリアスのみを使用せずにレンダリングします。そうでない場合は、デフォルト設定を使用します。

```
FORCE_BW_MONOSPACE="yes"
```

デフォルト設定

すべてのフォントはアンチエイリアスを使用してレンダリングされます。適切にヒンティングされたフォントは「バイトコードインタープリタ」 (BCI)でレンダリングされ、それ以外はautohinter (`hintstyle=hintslight`)でレンダリングされます。関連するsysconfig変数はすべてデフォルト設定のままにします。

CFFフォント

CFF形式のフォントを使用します。現在、FreeType2には数々の点で改良が重ねられており、このフォントは、デフォルトのTrueTypeフォントよりも可読性が高いと考えることができます。`PREFER_*_FAMILIES`の例に従って、このフォントを試してみてください。場合によっては、次の設定を使用して、より濃く太いフォントにできます。

```
SEARCH_METRIC_COMPATIBLE="no"
```

その理由は、このフォントは、デフォルトでは`hintstyle=hintslight`でレンダリングされているためです。次の設定の使用も検討してください。

```
SEARCH_METRIC_COMPATIBLE="no"
```

Autohinterの排他的使用

適切にヒンティングされたフォントに対しても、FreeType2のautohinterを使用します。これにより、太さが増してコントラストが下がるため、不鮮明になる場合があります。これを有効にするには、次の変数を設定します。

```
FORCE_AUTOHINTER="yes"
```

ヒンティングのレベルを制御するには、`FORCE_HINTSTYLE`を使用します。

17.1.5.2 Fontconfig XMLの詳細

Fontconfigの環境設定のフォーマットは、eXtensible Markup Language (XML)です。ここで取り上げるいくつかの例は、完全なリファレンスではなく概要です。詳しい情報とその他の例については、`man 5 fonts-conf`または`/etc/fonts/conf.d/`を参照してください。

中央のFontconfig設定ファイルは `/etc/fonts/fonts.conf` で、他の例と `/etc/fonts/conf.d/` ディレクトリ全体が含まれます。Fontconfigをカスタマイズする場合、変更を挿入できる場所は2つあります。

FONTCONFIG設定ファイル

1. **システム全体の変更.** `/etc/fonts/local.conf` ファイルを編集します(デフォルトで空の `fontconfig` 要素が含まれています)。
2. **ユーザ固有の変更.** `~/.config/fontconfig/fonts.conf` ファイルを編集します。Fontconfig設定ファイルは、`~/.config/fontconfig/conf.d/` ディレクトリに保存します。

ユーザ固有の変更は、システム全体の設定よりも優先されます。

注記: 非推奨のユーザ設定ファイル

`~/.fonts.conf` ファイルには非推奨のマークが付いているため、今後は使用しないことをお勧めします。代わりに `~/.config/fontconfig/fonts.conf` を使用してください。

すべての設定ファイルには `fontconfig` 要素が必要です。そのため、最小限のファイルは次のようになります。

```
<?xml version="1.0"?>
   <!DOCTYPE fontconfig SYSTEM "fonts.dtd">
   <fontconfig>
   <!-- Insert your changes here -->
   </fontconfig>
```

デフォルトのディレクトリでは不十分な場合は、各ディレクトリを指定した `dir` 要素を挿入します。

```
<dir>/usr/share/fonts2</dir>
```

Fontconfigは、「再帰的」にフォントを検索します。

次のFontconfigスニペットでフォントレンダリングアルゴリズムを選択できます(例17.1「レンダリングアルゴリズムを指定する」を参照)。

例 17.1 レンダリングアルゴリズムを指定する

```
<match target="font">
```

```
 <test name="family">
  <string>FAMILY_NAME</string>
 </test>
 <edit name="antialias" mode="assign">
  <bool>true</bool>
 </edit>
 <edit name="hinting" mode="assign">
  <bool>true</bool>
 </edit>
 <edit name="autohint" mode="assign">
  <bool>false</bool>
 </edit>
 <edit name="hintstyle" mode="assign">
  <const>hintfull</const>
 </edit>
</match>
```

さまざまなフォントプロパティをテストできます。たとえば、フォントファミリ(例を参照)、サイズの間隔、スペーシング、フォント形式などについて、`<test>`要素をテストできます。`<test>`を完全に破棄した場合、すべての`<edit>`要素が各フォントに適用されます(グローバルな変更)。

例 17.2 エイリアスとファミリ名の置換

ルール1

```
<alias>
 <family>Alegreya SC</family>
 <default>
  <family>serif</family>
 </default>
</alias>
```

ルール2

```
<alias>
 <family>serif</family>
 <prefer>
  <family>Droid Serif</family>
 </prefer>
```

```
</alias>
```

ルール3

```
<alias>
 <family>serif</family>
 <accept>
  <family>STIXGeneral</family>
 </accept>
</alias>
```

例17.2「エイリアスとファミリ名の置換」のルールは、「優先ファミリリスト」 (PFL)を作成します。要素に応じて異なるアクションが実行されます。

ルール1の `<default>` の場合

このルールは、`serif` ファミリ名をPFLの「末尾」に追加します。

ルール2の `<prefer>` の場合

このルールは、PFLに `Alegreya SC` が存在する場合、PFLで `serif` が最初に出現する箇所の「直前」に「」「Droid Serif」を追加します。

ルール3の `<accept>` の場合

このルールは、PFLで `serif` ファミリ名が最初に出現する箇所の「直後」に「」「STIXGeneral」ファミリ名を追加します。

まとめると、スニペットがルール1 - ルール2 - ルール3という順序で記述されている場合、ユーザが「Alegreya SC」を要求すると、表17.1「Fontconfigルールからの PFLの作成」で説明されているようにPFLが作成されます。

表 17.1 FONTCONFIGルールからのPFLの作成

順序	現在のPFL
要求	`Alegreya SC`
ルール1	`Alegreya SC`、`serif`
ルール2	`Alegreya SC`、`Droid Serif`、`serif`
ルール3	`Alegreya SC`、`Droid Serif`、`serif`、`STIXGeneral`

Fontconfigのメトリクスでは、ファミリ名は、他のパターン(スタイルやサイズなど)に比べて最も高い優先度を持ちます。Fontconfigは、システムに現在インストールされているファミリを確認します。「Alegreya　SC」がインストールされている場合、Fontconfigはそれを返します。インストールされていない場合、「Droid Serif」などを要求します。

注意してください。Fontconfigスニペットの順序を変更すると、Fontconfigが異なる結果を返す可能性があります。表17.2「順序を変更したFontconfigルールからのPFL生成結果」を参照してください。

表 17.2 順序を変更したFONTCONFIGルールからのPFL生成結果

順序	現在のPFL	注
要求	`Alegreya SC`	同じ要求が実行されます。
ルール2	`Alegreya SC`	`serif`がPFLに存在しないため、何も置換されません。
ルール3	`Alegreya SC`	`serif`がPFLに存在しないため、何も置換されません。
ルール1	`Alegreya SC`、`serif`	`Alegreya SC`がFPLに存在するため、置換が実行されます。

注記: 意味

`<default>`のエイリアスは、このグループ(インストールされていない場合)の分類または組み込みであると考えてください。この例が示すように、`<default>`は常にこのグループの`<prefer>`および`<accept>`のエイリアスより前に配置する必要があります。

`<default>`の分類は、汎用のエイリアスのserif、sans-serif、および等幅に限定されません。複雑な例については、`/usr/share/fontconfig/conf.avail/30-metric-aliases.conf`を参照してください。

例17.3「エイリアスとファミリ名の置換」に示す次のFontconfigスニペットは、`serif`グループを作成します。このグループのすべてのファミリは、前のフォントがインストールされていない場合、他のフォントを置換できます。

例 17.3 エイリアスとファミリ名の置換

```
<alias>
```

```
 <family>Alegreya SC</family>
 <default>
  <family>serif</family>
 </default>
</alias>
<alias>
 <family>Droid Serif</family>
 <default>
  <family>serif</family>
 </default>
</alias>
<alias>
 <family>STIXGeneral</family>
 <default>
  <family>serif</family>
 </default>
</alias>
<alias>
 <family>serif</family>
 <accept>
  <family>Droid Serif</family>
  <family>STIXGeneral</family>
  <family>Alegreya SC</family>
 </accept>
</alias>
```

優先度は、<accept> エイリアス内の順序によって決まります。同様に、それよりも強い <prefer> エイリアスを使用できます。

例17.2「エイリアスとファミリ名の置換」を例17.4「エイリアスとファミリ名の置換」で拡張します。

例 17.4 エイリアスとファミリ名の置換

ルール4

```
<alias>
 <family>serif</family>
 <accept>
  <family>Liberation Serif</family>
```

```
 </accept>
</alias>
```

ルール5

```
<alias>
 <family>serif</family>
 <prefer>
  <family>DejaVu Serif</family>
 </prefer>
</alias>
```

例17.4「エイリアスとファミリ名の置換」の拡張された設定では、PFLは次のように展開されます。

表 17.3 FONTCONFIGルールからのPFL生成結果

順序	現在のPFL
要求	`Alegreya SC`
ルール1	`Alegreya SC`、`serif`
ルール2	`Alegreya SC`、`Droid Serif`、`serif`
ルール3	`Alegreya SC`、`Droid Serif`、`serif`、`STIXGeneral`
ルール4	`Alegreya SC`、`Droid Serif`、`serif`、`Liberation Serif`、`STIXGeneral`
ルール5	`Alegreya SC`、`Droid Serif`、`DejaVu Serif`、`serif`、`Liberation Serif`、`STIXGeneral`

注記: 意味

- 同じ汎用名に対して複数の `<accept>` 宣言が存在する場合、最後に解析された宣言が「優先」されます。システム全体の設定を作成する場合、可能であれば、ユーザ(`/etc/fonts/conf.d/*-user.conf`)の「後」に `<accept>` を使用しないでください。
- 同じ汎用名に対して複数の `<prefer>` 宣言が存在する場合、最後に解析された宣言が「優先」されます。可能であれば、システム全体の設定では、ユーザの「前」に `<prefer>` を使用しないでください。
- 同じ汎用名に対しては、すべての `<prefer>` 宣言が `<accept>` 宣言よりも優先されます。ユーザが `<prefer>` だけでなく `<accept>` も自由に使用できるようにする場合、管理者はシステム全体の設定で `<prefer>` を使用しないようにする必要があります。一方、ユーザは通常 `<prefer>` を使用するため、これが悪影響を及ぼさないようにする必要があります。また、システム全体の設定の `<prefer>` の使用も確認します。

17.2 その他の情報

X11に関する詳細情報を入手するには、`xorg-docs` パッケージをインストールしてください。`man 5 xorg.conf` には、手動設定の形式に関する詳細情報が記載されています(必要な場合)。X11開発の詳細情報は、プロジェクトのホームページhttp://www.x.org で参照できます。

ドライバは、`xf86-video-*` パッケージにあります(たとえば、`xf86-video-nv`)。パッケージで配布されるドライバの大半については、関連するマニュアルページに詳細が記載されてます。たとえば、`nv` ドライバを使用する場合は、`man 4 nv` でドライバの詳細を参照できます。

サードパーティのドライバ情報は、`/usr/share/doc/packages/<package_name>` に記載されています。たとえば、`x11-video-nvidiaG03` の場合、パッケージのインストール後は、`/usr/share/doc/packages/x11-video-nvidiaG03` でマニュアルを参照できます。

18 FUSEによるファイルシステムへのアクセス

FUSEは、file system in userspaceの頭字語です。これは、特権のないユーザとしてファイルシステムを設定およびマウントできることを意味します。通常、このタスクを行うためには、`root`である必要があります。FUSE自体は、カーネルモジュールです。FUSEは、プラグインと組み合わせることで、ほとんどすべてのファイルシステムにアクセスするように拡張できます(リモートSSH接続、ISOイメージなど)。.

18.1 FUSEの設定

FUSEを使用するには、まず、`fuse`パッケージをインストールする必要があります。使用するファイルシステムによって、別々のパッケージとして使用できるプラグインを追加する必要があります。FUSEプラグインはSUSE Linux Enterpriseには付属していません。

一般的には、FUSEは設定の必要がなく、そのまま使用します。ただし、すべてのマウントポイントを結合するディレクトリの作成をお勧めします。たとえば、ディレクトリ`~/mounts`を作成し、そこに、各種のファイルシステムのサブディレクトリを挿入します。

18.2 利用可能なFUSEプラグイン

FUSEはプラグインに依存します。次のテーブルに、よく利用されるプラグインを一覧します。FUSEプラグインはSUSE Linux Enterpriseには付属していません。

表 18.1 利用可能なFUSEプラグイン

`fuseiso`	ISO9660ファイルシステムを含むCD-ROMをマウントします。
`ntfs-3g`	NTFSボリュームをマウントします(読み込み/書き込みサポート付き)。
`sshfs`	SSHファイル転送プロトコルに基づくファイルシステムクライアント。
`wdfs`	WebDAVファイルシステムをマウントします。

18.3 詳細情報

詳細については、FUSEのホームページhttp://fuse.sourceforge.netを参照してください。

III サービス

19 ネットワークの基礎

Linuxには、あらゆるタイプのネットワークストラクチャに統合するために必要なネットワークツールと機能が用意されています。ネットワークカードを使用したネットワークアクセスは、YaSTによって設定できます。手動による環境設定も可能です。この章では、基本的メカニズムと関連のネットワーク設定ファイルのみを解説します。

Linuxおよび他のUnix系オペレーティングシステムは、TCP/IPプロトコルを使用します。これは1つのネットワークプロトコルではなく、さまざまなサービスを提供する複数のネットワークプロトコルのファミリです。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、TCP/IPプロトコルファミリ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。RFCの詳細については、http://www.ietf.org/rfc.html を参照してください。

TCP/IPプロトコルファミリを構成する主要なプロトコル

TCP
: TCP(Transmission Control Protocol): 接続指向型の安全なプロトコルです。転送データは、まず、アプリケーションによってデータストリームとして送信され、オペレーティングシステム.によって適切なフォーマットに変換されます。データは、送信当初のデータストリーム形式で、宛先ホストのアプリケーションに着信します。TCPは転送中に損失したデータや順序が正しくないデータがないか、判定します。データの順序が意味を持つ場合は常にTCP/IPが実装されます。

UDP
: UDP(User Datagram Protocol): コネクションレスで安全でないプロトコルです。転送されるデータは、アプリケーションで生成されたパケットの形で送信されます。データが受信側に到着する順序は保証されず、データの損失の可能性があります。UDPはレコード指向のアプリケーションに適しています。TCPよりも遅延時間が小さいことが特徴です。

ICMP
: ICMP (Internet Control Message Protocol):基本的にはエンドユーザ向けのプロトコルではありませんが、エラーレポートを発行し、TCP/IPデータ転送にかかわるマシンの動作を制御できる特別な制御プロトコルです。またICMPには特別なエコーモードがあります。エコーモードは、pingで使用されています。

IGMP

IGMP (Internet Group Management Protocol): このプロトコルは、IPマルチキャストを実装した場合のマシンの動作を制御します。

に示したように、データのやり取りはさまざまなレイヤで実行されます。図19.1「TCP/IPの簡易階層モデル」実際のネットワークレイヤは、IP (インターネットプロトコル)によって実現される確実性のないデータ転送です。IPの上で動作するTCP (転送制御プロトコル)によって、ある程度の確実性のあるデータ転送が保証されます。IP層の下層には、Ethernetなどのハードウェア依存プロトコルがあります。

図 19.1 TCP/IPの簡易階層モデル

図では、各レイヤに対応する例を1つまたは2つ示しています。レイヤは抽象化レベルに従って並べられています。最下位レイヤは最もハードウェアに近い部分です。一方、最上位レイヤは、ハードウェアがまったく見えないほぼ完全な抽象化になります。各レイヤにはそれぞれの固有の機能があります。各レイヤ固有の機能は、上記の主要プロトコルの説明を読めば大体わかります。データリンク層と物理層は、Ethernetなどの使用される物理ネットワークを表します。

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、パケットにまとめられます(一度に全部を送信できません)。TCP/IPパケットの最大サイズは約64KBです。パケットサイズは通常、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。Ethernetの最大パケットサイズは、約1500バイトです。Ethernet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19.2「TCP/IPイーサネットパケット」に、Ethernetケーブル上に送出されるTCP/IPデータパケットの例を示します。誤り検出のためのチェックサムは、パケットの先頭ではなく最後に付加されます。これによりネットワークハードウェアの処理が簡素化されます。

図 19.2 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、データは各層を通過します。これらの層は、物理層を除き、すべてLinuxカーネルに実装されています。各レイヤは、隣接する下位レイヤに渡せるようにデータを処理します。最下位レイヤは、最終的にデータを送信する責任を負います。データを受信したときには、この手順全体が逆の順序で実行されます。重なり合ったたまねぎの皮のように、各レイヤで伝送データからプロトコルヘッダが除去されていきます。最後に、トランスポートレイヤが、着信側のアプリケーションがデータを利用できるように処理します。この方法では、1つのレイヤが直接やり取りを行うのは隣接する上下のレイヤのみです。データが伝送される物理的なネットワークは、100MBit/sのFDDIかもしれませんし、56Kbit/sのモデム回線かもしれませんが、アプリケーションがその違いを意識することはありません。同様に、物理ネットワークは、パケットの形式さえ正しければよく、伝送されるデータの種類を意識することはありません。

19.1 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、19.2項「IPv6—次世代インターネット」を参照してください。

19.1.1 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、固有の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19.1「IPアドレスの表記」の2行目に示すような形式で表記されます。

例 19.1 IPアドレスの表記

```
IP Address (binary):  11000000 10101000 00000000 00010100
IP Address (decimal):      192.     168.       0.      20
```

10進表記では、4つの各バイトが10進数で表記され、ピリオドで区切られます。IPアドレスは、ホストまたはネットワークインタフェースに割り当てられます。使用できるのは1回のみです。このルールには例外もありますが、次の説明には直接関係していません。

IPアドレスにあるピリオドは、階層構造を表しています。1990年代まで、IPアドレスは、各クラスに固定的に分類されていました。しかし、このシステムがあまりに柔軟性に乏しいことがわかったので、今日、そのような分類は行われていません。現在採用されているのは、クラスレスルーティング(CIDR: classless inter domain routing)です。

19.1.2 ネットマスクとルーティング

ネットマスクは、サブネットのアドレス範囲を定義するために用いられます。2台のホストが同じサブネットに存在する場合、相互に直接アクセスできます。同じサブネットにない場合は、サブネットのすべてのトラフィックを処理するゲートウェイのアドレスが必要です。2つのIPアドレスが同じサブネットワークに属しているかどうかを確認するには、両方のアドレスとネットマスクの「AND」を求めます。結果が同一であれば、両方のIPアドレスは同じローカルネットワークに属しています。相違があれば、それらのIPアドレス、そしてそれらに対応するインタフェースが連絡するには、ゲートウェイを通過する必要があります。

ネットマスクの役割を理解するには、例19.2「IPアドレスとネットマスクの論理積(AND)」を参照してください。ネットマスクは、そのネットワークにいくつのIPアドレスが属しているかを示す、32ビットの値から成っています。`1`になっているビットは、IPアドレスのうち、特定のネットワークに属することを示すビットに対応します。`0`になっているビットは、サブネット内での識別に使われるビットに対応します。これは、`1`になっているビット数が多いほど、サブネットが小さいことを意味します。ネットマスクは常に連続

する1のビットから構成されているので、その数だけでネットマスクを指定することができます。例19.2「IPアドレスとネットマスクの論理積(AND)」の、24ビットからなる第1のネットワークは、192.168.0.0/24と書くこともできます。

例 19.2 IPアドレスとネットマスクの論理積(AND)

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
----------------------------------------------------------------
Result of the link:         11000000 10101000 00000000 00000000
In the decimal system:           192.     168.       0.       0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask    (255.255.255.0): 11111111 11111111 11111111 00000000
----------------------------------------------------------------
Result of the link:         11010101 10111111 00001111 00000000
In the decimal system:           213.      95.      15.       0
```

また、たとえば同じEthernetケーブルに接続しているすべてのマシンは、普通同じサブネットに属し、直接アクセスできます。サブネットがスイッチまたはブリッジで物理的に分割されていても、これらのホストは直接アクセス可能です。

ローカルサブネットの外部のIPアドレスには、ターゲットネットワーク用のゲートウェイが設定されている場合にのみ、連絡できます。最も一般的には、外部からのすべてのトラフィックを扱うゲートウェイを1台だけ設置します。ただし、異なるサブネット用に、複数のゲートウェイを設定することも可能です。

ゲートウェイを設定すると、外部からのすべてのIPパケットは適切なゲートウェイに送信されます。このゲートウェイは、パケットを複数のホストを経由して転送し、それは最終的に宛先ホストに到着します。ただし、途中でTTL (存続期間)に達した場合は破棄されます。

特殊なアドレス

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19.2「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

これは、「このサブネット上のすべてのホストにアクセスする」と言い換えることができます。「」このアドレスを生成するには、2進数形式のネットマスクを反転させ、基本ネットワークアドレスと論理和をとります。そのため上記の例では、192.168.0.255になります。このアドレスをホストに割り当てることはできません。

ローカルホスト

アドレス `127.0.0.1` は、各ホストの「ループバックデバイス」に割り当てられます。「」このアドレスと、IPv4で定義された完全な `127.0.0.0/8` ループバックネットワークからのすべてのアドレスで、自分のマシンへの接続を設定できます。IPv6では、ループバックアドレスは1つだけです (`::1`)。

IPアドレスは、世界中で固有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をセットアップする場合のために、3つのアドレスドメインが用意されています。これらは、外部のインターネットに直接接続することはできません。インターネット上で転送されることがないからです。このようなアドレスドメインは、RFC 1597で、表19.1「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。

表 19.1 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`

19.2 IPv6—次世代インターネット

重要: IBM System z: IPv6のサポート

IPv6は、IBM System zハードウェアのCTCおよびIUCVネットワーク接続ではサポートされていません。

WWW (ワールドワイドウェブ)の出現により、ここ15年間でTCP/IP経由で通信を行うコンピュータの数が増大し、インターネットは爆発的に拡大しました。CERN (http://public.web.cern.ch)のTim Berners-Leeが1990年にWWWを発明して以来、インターネットホストは、数千から約1億まで増加しました。

前述のように、IPv4のアドレスはわずか32ビットで構成されています。しかも、多くのIPアドレスが失われています。というのは、ネットワークの編成方法のせいで、使われないIPアドレスが無駄に割り当てられてしまうからです。サブネットで利用できるアドレスの数は、(2のビット数乗 - 2)で与えられます。たとえば、1つのサブネットでは、2、6、または14個のアドレスが使用可能です。たとえば128台のホストをイ

ンターネットに接続するには、256個のIPアドレスを持つサブネットが必要ですが、そのうち2つのIPアドレスは、サブネット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。

19.2.1 長所

この新しいプロトコルがもたらした最大かつ最もわかりやすい進歩は、利用可能なアドレス空間の飛躍的な増加です。IPv6アドレスは、従来の32ビットではなく、128ビットで構成されています。これにより、2の128乗、つまり、約3.4×1038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、19.2.2項「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、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。

それでもルータがスイッチに接続されていれば、ルータは、ネットワークのホストに相互に通信する方法を通知するフラグ付きの通知を定期的に送信します。詳細については、RFC 2462、`radvd.conf(5)` のマニュアルページ、およびRFC 3315を参照してください。

モバイル性

IPv6を使用すると、複数のアドレスを1つのネットワークインタフェースに同時に割り当てることができます。これにより、ユーザは複数ネットワークに簡単にアクセスできます。このことは、携帯電話会社が提供する国際ローミングサービスにたとえられます。携帯電話を海外に持って行った場合、現地会社のサービス提供エリアに入ると自動的に携帯電話はそのサービスにログインし、同じ番号で普段と同じように電話をかけることができます。

安全な通信

IPv4では、ネットワークセキュリティは追加機能です。IPv6にはIPSecが中核的機能の1つとして含まれているので、システムが安全なトンネル経由で通信でき、インターネット上での部外者による通信傍受を防止します。

後方互換性

現実的に考えて、インターネット全体を一気にIPv4からIPv6に切り替えるのは不可能です。したがって、両方のプロトコルが、インターネット上だけでなく1つのシステム上でも共存できることが不可欠です。これは、一方ではアドレスの互換性によって(IPv4アドレスは容易にIPv6アドレスに変換できます)、他方ではトンネルの使用によって保証されています。参照先 19.2.3項「IPv4とIPv6の共存」. 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。

19.2.2 アドレスのタイプと構造

これまでに述べたように、現在のIPプロトコルには、IPアドレス数が急激に不足し始めているということと、ネットワーク設定とルーティングテーブルの管理がより複雑で煩雑な作業になっているという、2つの重要な問題があります。IPv6では、1つ目の問題を、アドレス空間を128ビットに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桁の英数字が入った8つのフィールドで構成され、それぞれのフィールドが16進数表記の16ビットを表します。各フィールドは、コロン(`:`)で区切られます。各フィールドで先頭の0は省略できますが、数字の間にある0や末尾の0は省略できません。もう1つの規則として、0のバイトが5つ以上連続する場合は、まとめて2つのコロン(::)で表すことができます。ただし、アドレスごとに`::`は1回しか使用できません。この省略表記の例については、例19.3「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。

例 19.3 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。に示すアドレスには、最初の 64ビットがアドレスのネットワー

ク部分を構成する情報、最後の　64ビットにホスト部分を構成する情報が入っています。例19.4「プレフィクスの長さを指定したIPv6アドレス」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にあるかそうでないかを判定します。

例 19.4 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。に、一部のプレフィクスタイプを示します。IPv6のプレフィクス

IPV6のプレフィクス

`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(19.2.3項「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

このタイプのアドレスは、ローカルのサブネットでのみ使用できます。このタイプのソース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、およびゼロ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IDを指定する16ビットのフィールドからなります。ここでも、残りはゼロバイトで埋められます。

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。

19.2.3　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(19.2.2項「アドレスのタイプと構造」を参照)。

ワールドワイドなIPv4ネットワークと隔離されているIPv6ホストは、トンネルを使って通信を行うことができます。IPv6パケットをIPv4パケットにカプセル化すれば、それをIPv4ネットワークに送ることができます。2つのIPv4ホスト間のこのような接続をトンネルと呼びます。これを行うには、パケットにIPv6の宛先アドレス(または対応するプレフィクス)とともに、トンネルの受信側にあるリモートホストのIPv4アドレスも含める必要があります。基本的なトンネルは、ホストの管理者間が合意すれば、手動で設定が可能です。これは、静的トンネリングとも呼ばれます。

ただし、静的トンネルの環境設定とメンテナンスは、あまりに手間がかかるので、多くの場合、日常の通信には向きません。そこで、IPv6は、動的トンネリングを実現する3つの異なる方法を提供しています。

6over4
: IPv6パケットが自動的にIPv4パケットとしてカプセル化され、マルチキャスト対応のIPv4ネットワークによって送信されます。IPv6は、ネットワーク全体(インターネット)を巨大なLAN (local area network)だと思い込んで動作することになります。これにより、IPv4トンネルの着信側の端を自動的に判定できます。ただし、この方法は拡張性に欠けているだけではなく、IPマルチキャストがインターネット上で広く普及しているとはいえないという事実も障害となります。したがってこの解決方法を採用できるのは、マルチキャストが利用できる小規模な企業内ネットワークだけです。この方式の仕様は、RFC 2529に規定されています。

6to4
: この方式では、IPv6アドレスからIPv4アドレスを自動的に生成することで、隔離されたIPv6ホストがIPv4ネットワーク経由で通信できるようにします。しかし、隔離されたIPv6ホストとインターネットの間の通信に関して、多くの問題が報告されています。この方式は、RFC 3056で規定されています。

IPv6トンネルブローカ
: 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。

19.2.4 IPv6の設定

IPv6を設定するには、通常、個々のワークステーションの設定を変更する必要はありません。IPv6は、デフォルトで有効になっています。インストール済みシステムでIPv6を有効または無効にするには、YaSTの［ネットワーク設定］モジュールを使用します。［グローバルオプション］タブで、必要に応じて［IPv6を有効にする］オプションをオン/オフします。次回の再起動時まで一時的に有効にするには、`root`として、「`modprobe -i ipv6`」と入力します。IPv6モジュールはロード後にアンロードすることはできません。

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/quaggaを使用してアドレスとルーティングの両方を自動設定することもできます。

`/etc/sysconfig/network` ファイルを使用してさまざまなタイプのトンネルをセットアップする方法の詳細については、`ifcfg-tunnel` のマニュアルページ(`man ifcfg-tunnel`)を参照してください。

19.2.5 詳細情報

ここでの概要は、IPv6に関する情報を網羅しているわけではありません。IPv6の詳細については、次のオンラインドキュメントや書籍を参照してください。

http://www.ipv6.org/
: IPv6のあらゆる情報にここからリンクできます。

http://www.ipv6day.org
: 独自のIPv6ネットワークを開始するには、すべての情報が必要です。

http://www.ipv6-to-standard.org/
: IPv6対応製品のリスト。

http://www.bieringer.de/linux/IPv6/
: Linux IPv6-HOWTOと多くの関連トピックへのリンクが用意されています。

RFC2640
: IPv6に関する基本的なRFCです。

IPv6 Essentials
: Silvia HagenによるIPv6 Essentials (ISBN 0-596-00125-8)は、このトピックに関するあらゆる重要な面を扱っている本です。

19.3 ネームレゾリューション

DNSはIPアドレスに1つまたは複数のホスト名を割り当てるとともに、ホスト名をIPアドレスに割り当てます。Linuxでは、この変換は通常、bindという特別な種類のソフトウェアによって行われます。また、この変換を行うマシンをネームサーバと呼びます。ホスト名は、その名前構成要素がピリオド(.)で区切られた階層システムを構成しています。しかしながら名前の階層構造は、先に述べたIPアドレスの階層構造とは無関係です。

`hostname.domain`という形式で書かれた完全な名前、たとえば、`jupiter.example.com`を考えてみましょう。「完全修飾ドメイン名」(FQDN: Fully Qualified Domain Name)と呼ばれるフルネームは、ホスト名とドメイン名(`example.com`)で構成されます。ドメイン名には最上位ドメイン(TLD)(`com`)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 Enterprise Serverでのネームサーバアクセスの設定については、19.4.1.4項「ホスト名とDNSの設定」に記載されています。独自のネームサーバの設定については、第22章 ドメインネームシステムに説明があります。

`whois`プロトコルは、DNSと密接な関係があります。このプログラムを使用すると、特定のドメインの登録者名をすぐに検索できます。

注記: MDNSおよび.localドメイン名

`.local`トップレベルドメインは、リゾルバではリンクローカルドメインとして処理されます。DNS要求は通常のDNS要求ではなく、マルチキャスト要求として送信されます。ネームサーバ設定で`.local`ドメインをすでに使用している場合は、このオプションを`/etc/host.conf`でオフに変更する必要があります。詳細については、`host.conf`のマニュアルページを参照してください。

インストール中にMDNSをオフにするには、`nomdns=1`をブートパラメータとして使用してください。

マルチキャストDNSの詳細は、http://www.multicastdns.orgを参照してください。

19.4 YaSTによるネットワーク接続の設定

Linuxでは多くのタイプのネットワーク接続がサポートされています。その多くは、異なるデバイス名と、ファイルシステム内の複数の場所に分散した設定ファイルを使用しています。手動によるネットワーク設定のさまざまな面についての詳細は、19.5項「ネットワークの手動環境設定」を参照してください。

ネットワークケーブルと接続され、リンクアップしているネットワークインタフェースはすべて自動的に設定されます。インストール済みのシステムには、いつでも付加的なハードウェアを設定することができます。以降のセクションでは、SUSE Linux Enterprise Serverがサポートするすべてのタイプのネットワーク接続について、その設定方法を説明します。

ヒント: IBM System z: ホットプラグ対応ネットワークカード

IBM System zプラットフォームでは、ホットプラグ可能なネットワークカードがサポートされていますが、DHCPを介したネットワークの自動統合は(PCの場合とは異なり)サポートされていません。検出後はインタフェースを手動で設定してください。

19.4.1 YaSTでのネットワークカードの設定

YaSTでEthernetカードまたはWi-Fi/Bluetoothカードを設定するには、[ネットワークデバイス] › [ネットワーク設定]の順に選択します。モジュールの開始後に、YaSTは[ネットワーク設定]ダイアログを表示します。ダイアログには[グローバルオプション]、[概要]、[ホスト名/DNS]、および[ルーティング]の4つのタブがあります。

[グローバルオプション]タブでは、ネットワークのセットアップ方法、IPv6、一般的なDHCPオプションの使用など、一般的なネットワークオプションを設定できます。詳細については、19.4.1.1項「グローバルネットワークオプションの設定」を参照してください。

[概要]タブには、インストールされたネットワークインタフェースと環境設定に関する情報が含まれています。正しく検出されたネットワークカードの名前が表示されます。このダイアログでは、手動で新しいカードを設定し、それらの設定内容を削除または変更できます。自動検出されなかったカードを手動で設定する場合は、19.4.1.3項　「検出されないネットワークカードの設定」を参照してください。すでに設定済みのカードの設定を変更する場合については、19.4.1.2項　「ネットワークカードの設定の変更」を参照してください。

[ホスト名/DNS]タブでは、マシンのホスト名を設定し、使用サーバに名前を付けることができます。詳細については、19.4.1.4項「ホスト名とDNSの設定」を参照してください。

[ルーティング]タブは、ルーティングの設定で使用します。詳細については、19.4.1.5項　「ルーティングの設定」を参照してください。

図 19.3 ネットワーク設定の実行

19.4.1.1　グローバルネットワークオプションの設定

YaSTの[ネットワーク設定]モジュールの[グローバルオプション]タブを使用して、NetworkManager、IPv6およびDHCPのクライアントオプションの使用など、重要なグローバルネットワークオプションを設定できます。この設定は、すべてのネットワークインタフェースに適用されます。

注記: Workstation ExtensionでのNetworkManagerの提供

NetworkManagerはWorkstation Extensionで提供されるようになりました。NetworkManagerをインストールするには、Workstation Extensionソフトウェアアドオンを有効にして、NetworkManagerパッケージを選択します。

[ネットワークのセットアップ方法]では、ネットワーク接続を管理する方法を選択します。NetworkManagerデスクトップアプレットですべてのインタフェースの接続を管理する場合は、[NetworkManagerサービス]を選択します。NetworkManagerは、複数の有線ネットワークおよび無線ネットワーク間の切り替えに適しています。デスクトップ環境を実行しない場合、またはコンピュータがXenサーバ(仮想システム)であるか、ネットワーク内でDHCPやDNSなどのネットワークサービスを提供する場合は、[Wickedサービス]の方法を使用します。NetworkManagerを使用する場合は、`nm-applet`を使用して、ネットワークオプションを設定する必要があります。[ネットワーク設定]モジュールのタブである[概要]、[ホスト名/DNS]、および[ルーティング]は無効になります。NetworkManagerの詳細については、SUSE Linux Enterprise Desktopのマニュアルを参照してください。

[IPv6プロトコル設定]で、IPv6プロトコルを使用するかどうかを選択します。IPv4とともにIPv6を使用できます。デフォルトでは、IPv6は有効です。ただし、IPv6プロトコルを使用しないネットワークでは、IPv6プロトコルを無効にした方が応答時間がより短くなる場合があります。IPv6を無効にするには、[IPv6を有効にする]を無効にします。IPv6が無効な場合、カーネルはIPv6モジュールを自動的にロードしません。この設定は、再起動後に適用されます。

[DHCPクライアントオプション]では、DHCPクライアントのオプションを設定します。[DHCPクライアントID]は、単一ネットワーク上の各DHCPクライアントで異なる必要があります。空白のままにした場合は、デフォルトでネットワークインタフェースのハードウェアアドレスになります。ただし、同じネットワークインタフェース、したがって同じハードウェアアドレスを使用して複数の仮想マシンを実行している場合は、ここで自由形式の固有識別子を指定します。

[送信するホスト名]では、DHCPクライアントがDHCPサーバにメッセージを送信するときに、ホスト名オプションフィールドで使用される文字列を指定します。一部のDHCPサーバでは、このホスト名(ダイナミックDNS)に応じて、ネームサーバゾーン(順レコードおよび逆レコード)を更新します。また一部のDHCPサーバでは、クライアントからのDHCPメッセージで、[送信するホスト名]オプションフィールドに特定の文字列が含まれていることが必要です。現在のホスト名(`/etc/HOSTNAME`で定義されたホスト名)を送信する場合は、[**自動**]のままにします。ホスト名を送信しない場合は、このオプションフィールドを空のままにします。

DHCPからの情報に従ったデフォルトのルートを変更しない場合は、[DHCPで既定のルートを変更する]をオフにします。

19.4.1.2 ネットワークカードの設定の変更

ネットワークカードの設定を変更するには、YaSTの[ネットワーク設定] > [概要]で検出されたカードのリストから目的のカードを選択し、[編集]をクリックします。[ネットワークカードの設定]ダイアログが表示されます。このダイアログの[一般]、[アドレス]、および[ハードウェア]タブを使用してカードの設定を変更します。

19.4.1.2.1 IPアドレスの設定

[Network Card Setup]ダイアログの[アドレス]タブで、ネットワークカードのIPアドレス、またはそのIPアドレスの決定方法を設定できます。IPv4およびIPv6の両アドレスがサポートされます。ネットワークカードは、[IPアドレスなし](ボンドデバイスで有用)の場合や、[静的に割り当てられたIPアドレス](IPv4またはIPv6)、あるいは[DHCP]または[Zeroconf]のいずれかまたは両方を経由して割り当てられる[動的アドレス]を持つ場合もあります。

[Dynamic Address]を使用する場合は、[DHCP Version 4 Only](IPv4の場合)、[DHCP Version 6 Only](IPv6の場合)、または[DHCP Both Version 4 and 6]のいずれを使用するかを選択します。

可能であれば、インストール時に利用可能なリンクを持つ最初のネットワークカードがDHCPによる自動アドレス設定を使用するように自動的に設定されます。

注記: IBM System zとDHCP

IBM System zプラットフォームでは、DHCPベースのアドレス設定はMACアドレスを持つネットワークカードの場合にのみサポートされます。これに該当するのは、OSAカードおよびOSA Expressカードだけです。

DSL回線を使用していてISP(Internet Service Provider)からスタティックIPが割り当てられていない場合も、DHCPを使用する必要があります。DHCPを使用することを選択する場合は、YaSTネットワークカード設定モジュールの[ネットワーク設定]ダイアログにある[グローバルオプション]タブの[DHCPクライアントオプション]で詳細を設定します。さまざまなホストが同じインタフェースを介して通信するようにバーチャルホストがセットアップされている場合は、各ホストの識別に[DHCPクライアントID]が必要になります。

DHCPは、クライアント設定には適していますが、サーバ設定には適していません。静的なIPアドレスを設定するには、以下の手順に従ってください。

1. YaSTネットワークカード設定モジュールの[概要]タブの検出されたカードのリストから目的のカードを選択し、[編集]をクリックします。

2. [アドレス]タブで、[Statically Assigned IP Address]を選択します。

3. [IPアドレス]を入力します。IPv4およびIPv6の両アドレスを使用できます。[サブネットマスク]にネットワークマスクを入力します。IPv6アドレスが使用されている場合は、フォーマット `/64` のプレフィックス長に対する[サブネットマスク]を使用します。
オプションで、このアドレスの完全修飾[ホスト名]を入力できます。このホスト名は、`/etc/hosts` 設定ファイルに書き込まれます。

4. [次へ]をクリックします。

5. 環境設定を有効にするには、[OK]をクリックします。

静的アドレスを使用する場合、ネームサーバとデフォルトゲートウェイは、自動的には設定されません。ネームサーバを設定するには、19.4.1.4項 「ホスト名とDNSの設定」に従って手順を進めます。ゲートウェイを設定するには、19.4.1.5項 「ルーティングの設定」に従って手順を進めます。

19.4.1.2.2 複数のアドレスの設定

1台のネットワークデバイスに、複数のIPアドレスを割り当てることができます。

注記: エイリアスは互換機能

これらのエイリアスまたはラベルはそれぞれIPv4でのみ動作し、IPv6では、無視されます。`iproute2` ネットワークインタフェースを使用する場合、1つ以上のアドレスを持つことができます。

YaSTを使用してネットワークカードに追加のアドレスを設定するには、次の手順に従います。

1. YaSTの[ネットワーク設定]モジュールの[概要]タブの検出されたカードのリストから目的のカードを選択し、[編集]をクリックします。

2. [アドレス] › [追加アドレス]タブで、[追加]をクリックします。

3. [IPアドレスラベル]、[IPアドレス]、および[ネットマスク]に適切な値を入力します。エイリアス名にはインタフェース名を含めないでください。

4. 設定内容を有効にするために、設定を確認します。

19.4.1.2.3 デバイス名およびUdevルールの変更

ネットワークカードのデバイス名が使用されている場合、ネットワークカードのデバイス名を変更できます。また、ハードウェア(MAC)アドレスまたはバスIDを介してudevによりネットワークカードを識別するかどうかを選択できます。大型のサーバでは、カードのホットスワッピングを容易にするために後者のオプションが適しています。YaSTを使ってこうしたオプションを設定するには、次の手順に従います。

1. YaSTの［ネットワーク設定］モジュールの［概要］タブの検出されたカードのリストから目的のカードを選択し、［編集］をクリックします。
2. ［ハードウェア］タブを開きます。現在のデバイス名が［Udevルール］に表示されます。［変更］をクリックします。
3. udevで［MACアドレス］または［バスID］によりカードを識別するかどうかを選択します。カードの現在のMACアドレスおよびバスIDがダイアログに表示されます。
4. デバイス名を変更するには、［Change Device Name］オプションをオンにし、名前を編集します。
5. 設定内容を有効にするために、設定を確認します。

19.4.1.2.4 ネットワークカードカーネルドライバの変更

一部のネットワークカードには、複数のカーネルドライバを使用できます。カードがすでに設定されている場合は、YaSTで利用可能で適切なドライバのリストから、使用するカーネルドライバを選択できます。また、カーネルドライバのオプションを指定することもできます。YaSTを使ってこうしたオプションを設定するには、次の手順に従います。

1. YaSTのネットワーク設定モジュールの［概要］タブの検出されたカードのリストから目的のカードを選択し、［編集］をクリックします。
2. ［ハードウェア］タブを開きます。
3. ［モジュール名］で、使用するカーネルドライバを選択します。選択したドライバのオプションを、［オプション］に「`option=value`」の形式で入力します。他にもオプションを使用する場合は、スペースで区切る必要があります。
4. 設定内容を有効にするために、設定を確認します。

19.4.1.2.5 ネットワークデバイスの有効化

`wicked`を使った方法を使用している場合、デバイスをブート時、ケーブル接続時、カード検出時、または手動で起動するように設定したり、起動しないように設定したりすることができます。デバイスの起動方法を変更するには、次の手順に従います。

1. YaSTで、[ネットワークデバイス] › [ネットワーク設定]で検出されたカードの一覧からカードを選択し、[編集]をクリックします。
2. [一般]タブの[デバイスの起動]から、適切な項目を選択します。
   システムブート中にデバイスを起動するには、[ブート時]を選択します。[ケーブル接続時]では、インタフェースで物理接続が存在するかどうかが監視されます。[ホットプラグ時]では、インタフェースは可能な限り早急に設定されます。これは、[ブート時]オプションに似ていますが、インタフェースがブート時に存在しない場合にエラーが発生しない点のみが異なります。[ifup]でインタフェースを手動で制御する場合は、[`手動`]を選択します。デバイスを全く起動しない場合は、[起動しない]を選択します。[NFSrootオン]は[ブート時]に似ていますが、インタフェースは`systemctl stop wicked.service`コマンドではシャットダウンしません。このオプションは、NFSまたはiSCSIのルートファイルシステムを使用する場合に選択します。
3. 設定内容を有効にするために、設定を確認します。

19.4.1.2.6 最大転送単位サイズの設定

インタフェースの最大転送単位(MTU)を設定できます。MTUでは、最大許容パケットサイズ(バイト)を参照します。MTUが大きいと、帯域幅の効率が高くなります。ただし、パケットが大きくなると、低速なインタフェースの処理がしばらく阻止され、以降のパケットの遅延が増加する場合があります。

1. YaSTで、[ネットワークデバイス] › [ネットワーク設定]で検出されたカードの一覧からカードを選択し、[編集]をクリックします。
2. [一般]タブの[MTUを設定]リストから、適切な項目を選択します。
3. 設定内容を有効にするために、設定を確認します。

19.4.1.2.7 IPoIB (IP-over-InfiniBand)用のインフィニバンドの設定

1. YaSTで、[ネットワークデバイス] › [ネットワーク設定]でインフィニバンドデバイスを選択し、[編集]をクリックします。

2. [一般]タブの[IP-over-InfiniBand](IPoIB)モードで[接続済み](デフォルト)または[データグラム]を選択します。

3. 設定内容を有効にするために、設定を確認します。

インフィニバンドの詳細については、`/usr/src/linux/Documentation/infiniband/ipoib.txt`を参照してください。

19.4.1.2.8 ファイアウォールの設定

Book “Security Guide” 15 “Masquerading and Firewalls”15.4.1 “Configuring the Firewall with YaST”で説明しているような詳細なファイアウォール設定を行わずに、デバイスに基本的なファイアウォールを設定することができます。次の手順に従います。

1. YaSTの[ネットワークデバイス] › [ネットワーク設定]モジュールを開きます。[概要]タブで、検出されたカードの一覧からカードを選択し、[編集]をクリックします。

2. [ネットワーク設定]ダイアログの[一般]タブを表示します。

3. インタフェースを割り当てる[ファイアウォールゾーン]を指定します。次のオプションを指定できます。

   Firewall Disabled
   : このオプションは、ファイアウォールが無効であり、ファイアウォールがまったく実行しない場合にのみ利用可能です。コンピュータが、外部ファイアウォールにより保護されている、より規模の大きいネットワークに接続している場合にのみ、このオプションを使用してください。

   自動割り当てゾーン
   : このオプションは、ファイアウォールが有効になっている場合のみ、利用できます。ファイアウォールが実行中であり、インタフェースがファイアウォールゾーンに自動的に割り当てられます。こうしたインタフェースには、`any`キーワードを含むゾーンまたは外部ゾーンが使用されます。

   内部ゾーン(未保護)
   : ファイアウォールを実行しますが、このインタフェースを保護するルールは使いません。コンピュータが、外部ファイアウォールにより保護されている、より規模の大きいネットワークに接続している場合に、このオプションを使用してください。また、マシンに追加ネットワークインタフェースが存在する場合、内部ネットワークに接続するインタフェースで使用できます。

非武装地帯(DMZ)

非武装地帯ゾーンは、内部ネットワークと(悪意のある)インターネットとの中間にあたるゾーンです。このゾーンに割り当てられたホストは、内部ネットワークおよびインターネットからアクセスされますが、ホストから内部ネットワークにアクセスすることはできません。

外部ゾーン

このインタフェースでファイアウォールを実行し、(危険な可能性のある)他のネットワークトラフィックからインタフェースを保護します。これがデフォルトのオプションです。

4. 設定内容を有効にするために、設定を確認します。

19.4.1.3 検出されないネットワークカードの設定

ネットワークカードが正しく検出されなかった場合、そのカードは検出されたカードのリストに含まれません。システムにそのカード用のドライバが間違いなく含まれている場合は、そのようなカードを手動で設定することができます。特殊なネットワークデバイスタイプ(ブリッジ、ボンド、TUN、TAPなど)も設定できます。未検出のネットワークカードまたは特殊なデバイスを設定するには、次の手順に従います。

1. YaSTの[ネットワークデバイス] › [ネットワーク設定] › [概要]ダイアログで[追加]をクリックします。
2. [ハードウェア]ダイアログで、使用可能なオプションからインタフェースの[デバイスの型]と[環境設定名]を設定します。ネットワークカードが、PCMCIAデバイスかUSBデバイスの場合、それぞれのチェックボックスを選択して、[次へ]をクリックしダイアログを終了します。それ以外の方法では、必要に応じて、カードとその[オプション]で使用されるカーネルの[モジュール名]を定義できます。
   [Ethtoolオプション]では、インタフェースの`ifup`により使用される`ethtool`オプションを設定できます。使用可能なオプションについては、`ethtool`マニュアルページを参照してください。オプション文字列が`-`で始まる場合(たとえば`-K interface_name rx on`)、文字列内の2番目の単語が現在のインタフェースの名前に置換されます。それ以外の場合(たとえば`autoneg off speed 10`)、`-s interface_name`の前に`ifup`が追加されます。
3. [次へ]をクリックします。
4. [一般]、[アドレス]、および[ハードウェア]タブで、インタフェースのIPアドレス、デバイス起動方法、ファイアウォールゾーンなどの必要なオプションを設定します。環境設定オプションの詳細については、19.4.1.2項「ネットワークカードの設定の変更」を参照してください。
5. インタフェースのデバイスタイプとして、[ワイヤレス]を選択した場合は、次のダイアログでワイヤレス接続の設定を行います。

6. 新しいネットワーク設定を有効にするために、設定を確認します。

19.4.1.4 ホスト名とDNSの設定

Ethernetカードがすでに利用できる状態で、インストール時にネットワーク設定を変更しなかった場合、コンピュータのホスト名が自動的に生成され、DHCPが有効になります。また、ホストがネットワークに参加するために必要なネームサービス情報も自動的に生成されます。ネットワークアドレス設定にDHCPを使用している場合は、ドメインネームサーバのリストは自動的に記入されます。静的設定を利用する場合は、これらの項目を手動で設定してください。

コンピュータ名を変更し、ネームサーバの検索リストを修正するには、以下の手順に従ってください。

1. YaST内の[ネットワークデバイス]モジュールの[ネットワーク設定] › [ホスト名/DNS]タブに移動します。

2. [ホスト名]にホスト名を入力し、必要に応じて[ドメイン名]にドメイン名を入力します。マシンがメールサーバである場合、ドメインは特に重要です。ホスト名はグローバルであり、すべての設定ネットワークインタフェースに適用されることに注意してください。
   IPアドレスを取得するためにDHCPを使用している場合、DHCPによりコンピュータのホスト名が自動的に設定されます。異なるネットワークに接続する場合は、異なるホスト名が割り当てられることがあり、ランタイムにホスト名が変更されるとグラフィックデスクトップが混同される可能性があるので、この機能を無効にした方が良い場合もあります。DHCPを使用したIPアドレスの取得を無効にするには、[DHCPでホスト名を変更する]をオフにします。
   [ホスト名をループバックIPに割り当てる]では、ホスト名を`/etc/hosts`内の`127.0.0.2`(ループバック) IPアドレスに関連付けます。アクティブネットワークが存在しないときでも常に解決可能なホスト名を必要とする場合に有用なオプションです。

3. [DNS環境設定の変更]では、DNS設定(ネームサーバ、検索リスト、`/etc/resolv.conf`ファイルのコンテンツ)を変更する方法を選択します。
   [既定のポリシーを使用する]オプションを選択した場合、(DHCPクライアントまたはNetworkManagerから)動的に取得されたデータと、(YaSTまたは設定ファイルで)静的に定義されたデータをマージする`netconfig`スクリプトにより設定が処理されます。ほとんどの場合、デフォルトのポリシーで十分です。
   [手動でのみ]オプションを選択した場合、`netconfig`では`/etc/resolv.conf`ファイルを変更できません。ただし、このファイルは手動で編集できます。
   [Custom Policy]オプションを選択した場合、マージポリシーを定義する[Custom Policy Rule]文字列を指定する必要があります。この文字列は、設定の有効なソースとみなされるインタフェース名のカンマで区切られたリストから構成されます。完全なインタフェース名以外に、複

数のインタフェースに一致する基本的なワイルドカードを使用することもできます。たとえば `eth* ppp?` は、先頭がethであり、以降にppp0-ppp9を含むすべてのインタフェースが対象になります。`/etc/sysconfig/network/config` ファイルで定義された静的な設定を適用する方法を示す次の2つの特別なポリシー値が存在します。

`STATIC`
: 静的な設定は、動的な設定とマージされる必要があります。

`STATIC_FALLBACK`
: 静的な設定は、動的設定が利用できない場合のみ使用されます。

詳細については、`netconfig`(8)のマニュアルページ(`man 8 netconfig`)を参照してください。

4. [ネームサーバ]および[ドメイン検索]リストに入力します。ネームサーバは、ホスト名ではなく、192.168.1.116などのIPアドレスにより指定する必要があります。[ドメイン検索]タブで指定した名前は、ドメインが指定されていないホスト名の解決のために使用されるドメイン名です。複数の[ドメイン検索]を使用する場合は、カンマまたは空白でドメインを区切ります。

5. 設定内容を有効にするために、設定を確認します。

コマンドラインからYaSTを使用してホスト名を編集することもできます。YaSTによる変更はすぐに有効になります(`/etc/HOSTNAME` ファイルを手動で編集する場合はすぐに有効にはなりません)。ホスト名を変更するには、次のコマンドを実行します。

```
yast dns edit hostname=hostname
```

ネームサーバを変更するには、次のコマンドを実行します。

```
yast dns edit nameserver1=192.168.1.116
yast dns edit nameserver2=192.168.1.117
yast dns edit nameserver3=192.168.1.118
```

19.4.1.5 ルーティングの設定

コンピュータを他のコンピュータやネットワークと通信させるには、ネットワークトラフィックが正しい経路を通過するように、ルーティング情報を設定する必要があります。DHCPを使用している場合、この情報は自動的に設定されます。静的アドレスを使用する場合は、このデータを手作業で追加する必要があります。

1. YaSTで、[ネットワーク設定] › [ルーティング]の順に移動します。
2. [デフォルトゲートウェイ]のIPアドレス(IPv4および必要に応じてIPv6)を入力します。デフォルトゲートウェイは、可能性のあるすべての宛先に一致しますが、必要なアドレスに一致するルーティングテーブルエントリが存在する場合は、デフォルトゲートウェイ経由のデフォルトルートの代わりにそのエントリが使用されます。
3. [ルーティングテーブル]には、さらに追加エントリを入力できます。[宛先]のネットワークIPアドレス、[ゲートウェイ]のIPアドレス、および[ネットマスク]を入力します。定義されたネットワークにトラフィックがルーティングされる[デバイス]を選択します(マイナス記号はデバイスを表わします)。このいずれかの値を省略する場合は、マイナス記号(`-`)を使用します。デフォルトゲートウェイをテーブルに入力するには、[宛先]フィールドを `default` のままにします。

   注記

   追加のデフォルトルートが使用されている場合、より高い優先度を持つルートを決定するためのメトリックオプションを指定できます。メトリックオプションを指定するには、[オプション]に `- metric番号` を入力します。最も高いメトリックを持つルートがデフォルトとして使用されます。ネットワークデバイスが切断している場合は、そのルートが削除され、次のルートが使用されます。ただし、現在のカーネルは静的なルーティングでメトリックを使用せず、`multipathd` などのルーティングデーモンのみがメトリックを使用します。
4. システムがルータの場合、必要に応じて、[ネットワーク設定]で[IPv4転送]および[IPv6転送]を有効にします。
5. 設定内容を有効にするために、設定を確認します。

19.4.2 IBM System z: ネットワークデバイスの設定

IBM System z用のSUSE Linux Enterprise Serverは、さまざまな種類のネットワークインタフェースをサポートしています。これらのインタフェースは、YaSTを使って設定することができます。

19.4.2.1 qeth-hsiデバイス

`qeth-hsi` (Hipersocket)インタフェースをインストール済みのシステムに追加するには、YaSTで[ネットワークデバイス] › [ネットワーク設定]モジュールを起動します。READデバイスアドレスとして使用するため、[Hipersocket]とマークされたデバイスの1つを選択して、[編集]をクリックします。読み込みチャネル、書き込みチャネル、および制御チャネルのデバイス番号を入力します(デバイス番号

形式の例: `0.0.0800`)。[次へ]をクリックします。[ネットワークアドレスの設定]ダイアログで、新しいインタフェースのIPアドレスとネットマスクを指定し、[次へ]と[OK]をクリックしてネットワークの設定を終了します。

19.4.2.2 qeth-ethernetデバイス

`qeth-ethernet` (IBM OSA Expressイーサネットカード)インタフェースをインストール済みのシステムに追加するには、YaSTで[ネットワークデバイス] › [ネットワーク設定]モジュールを起動します。READデバイスアドレスとして使用するため、[IBM OSA Expressイーサネットカード]とマークされたデバイスの1つを選択して[編集]をクリックします。読み込みチャネル、書き込みチャネル、および制御チャネルのデバイス番号を入力します(デバイス番号形式の例: `0.0.0700`)。必要なポート名、ポート番号(該当する場合)、および追加オプション(『Linux for IBM IBM System z: Device Drivers, Features, and Commands』リファレンスマニュアル、http://www.ibm.com/developerworks/linux/linux390/documentation_suse.html を参照)のほか、IPアドレスおよび適切なネットマスクを入力します。[次へ]と[OK]をクリックして、ネットワークの設定を終了します。

19.4.2.3 ctcデバイス

`ctc` (IBMパラレルCTCアダプタ)インタフェースをインストール済みのシステムに追加するには、YaSTで[ネットワークデバイス] › [ネットワーク設定]モジュールを起動します。READデバイスアドレスとして使用する[IBMパラレルCTCアダプタ]というマークの付いたデバイスの1つを選択して、[設定]をクリックします。お使いのデバイスに合わせて[デバイス設定]を選択します(通常は、[互換モード])。自分のIPアドレスとリモートのIPアドレスを指定します。必要に応じて、[詳細] › [詳細設定]の順に選択してMTUサイズを調整します。[次へ]と[OK]をクリックして、ネットワークの設定を終了します。

警告: CTCは、サポートされなくなりました

このインタフェースを使用することはお勧めしません。SUSE Linux Enterprise Serverの今後のバージョンでは、このインタフェースはサポートされません。

19.4.2.4 lcsデバイス

`lcs` (IBM OSA-2アダプタ)インタフェースをインストール済みのシステムに追加するには、YaSTで[ネットワークデバイス] › [ネットワーク設定]モジュールを起動します。[IBM OSA-2アダプタ]というマークの付いたデバイスの1つの選択して、[設定]をクリックします。ポート番号や他のオプション

(『Linux for IBM System z: Device Drivers, Features, and Commands 』リファレンスマニュアル、http://www.ibm.com/developerworks/linux/linux390/documentation_suse.html を参照)、IPアドレス、およびネットマスクを入力します。[次へ]と[OK]をクリックして、ネットワークの設定を終了します。

19.4.2.5 IUCVデバイス

`iucv` (IUCV)インタフェースをインストール済みのシステムに追加するには、YaSTで[ネットワークデバイス] › [ネットワーク設定]モジュールを起動します。[IUCV]とマークされたデバイスを選択し、[編集]をクリックします。IUCVパートナーの名前を入力するように要求されます([ピア])。パートナー名(大文字と小文字が区別されます)を入力して、[次へ]をクリックします。自分の[IPアドレス]と、パートナーの[リモートIPアドレス]の両方を指定します。必要な場合は、[Set MTU]サイズを[一般]タブで設定します。[次へ]と[OK]をクリックして、ネットワークの設定を終了します。

警告: IUCVはサポートされなくなりました

このインタフェースを使用することはお勧めしません。SUSE Linux Enterprise Serverの今後のバージョンでは、このインタフェースはサポートされません。

19.5 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、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、ここで説明するネットワーク環境設定の背景知識がYaSTでの設定作業に役立つことがあります。

19.5.1 `wicked`ネットワーク環境設定

`wicked`と呼ばれるツールとライブラリは、ネットワーク環境設定用の新しいフレームワークを提供します。

ネットワーク管理は相互に影響を持つ複数の層で構成されていますが、その相互作用は明確とはいえず、悪影響があっても把握しづらかったり、あいまいな制約や慣習があったりします。このようなネットワーク管理のさまざまな層を1つのスクリプト(多くても2つ程度のスクリプト)に寄せ集めている点が、従来のネットワークインタフェース管理における問題の1つです。異なるシナリオに対応するために特別なハックを使った層がいくつもあると、保守負担が増加します。現状では、dhcpcdなどのデーモンによって実装されるアドレス設定プロトコルが使用されていますが、他のインフラストラクチャとの相互作用は

十分ではありません。そこで、インタフェースを永続的に識別できるようにするため、多くのudevサポートを必要とするインタフェース命名スキームが導入されたものの、これは洗練されているとはいいがたい手段です。

wickedというアイデアが生まれたのは、この問題をさまざまな方法で分解するためです。どの方法もまったく新しいものではありませんが、異なるプロジェクトから得たアイデアをまとめようとする試みから、総合的により優れた解決策が生まれることが期待できます。

アプローチの1つは、クライアント/サーバモデルを使用することです。これにより、wickedは、アドレス設定のような作業について、フレームワーク全体と効果的に統合された標準化機能を定義できます。たとえば、アドレス設定では、管理者はDHCPまたはIPv4 zeroconf経由でインタフェースを設定するよう要求できます。アドレス設定サービスが実行するのは、サーバからリースを取得してwickedサーバプロセスに渡すことだけで、その後はwickedサーバプロセスが要求されたアドレスおよびルートをインストールします。

問題を分解するもう1つのアプローチは、階層化を強制的に導入することです。すべてのタイプのネットワークインタフェースに対して、ネットワークインタフェースのデバイス層(VLAN、ブリッジ、ボンド、または準仮想化されたデバイス)を設定するdbusサービスを定義できます。アドレス設定といった共通の機能は、こうしたデバイス固有のサービスの上に階層化した結合サービスによって実装します。これにより、サービスを個別に実装する必要がなくなります。

wickedフレームワークは、そのタイプに応じてネットワークインタフェースにアタッチされるさまざまなdbusサービスを使用して、これら2つの側面を実装します。ここでは、wickedにおける現在のオブジェクト階層をおおまかに説明します。

各ネットワークインタフェースは、`/org/opensuse/Network/Interfaces`の子オブジェクトを介して表されます。子オブジェクトの名前は、そのifindexで指定されます。たとえば、ループバックインタフェースは通常、ifindex 1を取り、`/org/opensuse/Network/Interfaces/1`です。登録されている最初のEthernetインタフェースは、`/org/opensuse/Network/Interfaces/2`です。

各ネットワークインタフェースには「クラス」が関連付けられており、そのクラスを使用して、サポートするdbusインタフェースが選択されます。「」デフォルトでは、各ネットワークインタフェースは、クラス`netif`に属し、wickeddはこのクラスと互換性のあるすべてのインタフェースを自動的にアタッチします。現在の実装では、これには次のインタフェースが含まれます。

org.opensuse.Network.Interface
: リンクアップとリンクダウンの取得、MTUの割り当てなどの、一般的なネットワークインタフェース機能。

org.opensuse.Network.Addrconf.ipv4.dhcp ,
org.opensuse.Network.Addrconf.ipv6.dhcp,
org.opensuse.Network.Addrconf.ipv4.auto,
org.opensuse.Network.Addrconf.ipv6.auto
: DHCP、IPv6 autoconf、IPv4 zeroconfなどのアドレス設定サービス。

これ以外に、ネットワークインタフェースで特別な設定メカニズムが必要な場合や、ネットワークインタフェースがこのようなメカニズムを備えている場合もあります。たとえば、Ethernetデバイスの場合、リンク速度、チェックサム計算のオフロードなどを制御可能にしたい場合があります。これを実現するために、Ethernetデバイスには、`netif`のサブクラスである、`netif-ethernet`という独自のクラスがあります。このため、Ethernetインタフェースに割り当てられたdbusインタフェースには、上記に一覧にされているすべてのサービス、および`netif-ethernet`クラスに属するオブジェクトでのみ使用可能なサービスである`org.opensuse.Network.Ethernet`が含まれています。

同様に、ブリッジ、VLAN、ボンド、インフィニバンドなどのインタフェースタイプのクラスも存在します。Ethernetデバイスの上に位置し、実際には仮想ネットワークインタフェースであるVLANなど、最初に作成する必要があるインタフェースとはどのように相互作用すればよいのでしょうか。このような場合、wickedは、`org.opensuse.Network.VLAN.Factory`などのファクトリインタフェースを定義します。このようなファクトリインタフェースは、要求されたタイプのインタフェースを作成できる単一の機能を提供します。これらのファクトリインタフェースは、`/org/opensuse/Network/Interfaces`リストノードにアタッチされます。

19.5.1.1 現在サポートされている内容

`wicked`は現在、以下をサポートしています。

- SUSEおよびRedHatスタイルの`/etc/sysconfig/network`ファイルを解析する環境設定ファイルバックエンド。開発はSUSEのインストール済み環境上で行われるため、多くの場合、前者の方がRedHat用ファイルよりも安定しています。
- ネットワークインタフェース設定をXMLで表す環境設定ファイルバックエンド。netcfが使用する設定から発展した構文。
- Ethernetやインフィニバンド、VLAN、ブリッジ、ボンドデバイスなどの、「標準」のネットワークインタフェースの起動とシャットダウン。「」ブリッジとボンドには、依然としていくつかの問題があります。
- ワイヤレス。まだ完全ではなく、1つのネットワークに制限されています。
- 内蔵DHCPv4クライアントおよび内蔵DHCPv6クライアント。

- リンクが検出されるとすぐに、自動的にインタフェースを起動することができるいくつかの実験コードがあります。
- XMLリーダ/ライタの実装。これは規格に完全準拠しているとはいえないものの、コンパクトなフットプリントを持ち、適度に高速であると思われます。これにはXPath 1.0の部分的な実装も付属していて、XMLによるインタフェースの記述から情報を抽出できるため、XMLを自分自身で解析する必要はありません。

19.5.1.2 wickedの使用

SUSE Linux Enterpriseでは、NetworkManagerを選択しなかった場合、`wicked`がデフォルトで実行されています。これを有効化する必要がある場合には、次を呼び出します。

```
systemctl enable --force wicked.service
```

これにより、wickedサービスが有効になり、`wicked.service`エイリアスリンクに対して`network.service`が作成され、次回ブート時にネットワークを起動します。

サーバプロセスを起動します。

```
systemctl start wickedd.service
```

これにより、`wickedd` (メインサーバ)および関連するサプリカントがデバッグモードで起動し、トレース情報をsyslogに出力します。

```
/usr/sbin/wickedd --foreground
/usr/lib/wicked/bin/wickedd-dhcp4 --foreground
/usr/lib/wicked/bin/wickedd-auto4 --foreground
/usr/lib/wicked/bin/wickedd-dhcp6 --foreground
```

ネットワークを起動します。

```
systemctl start wicked.service
```

または、`network.service`エイリアスを使用します。

```
systemctl start network.service
```

これらのコマンドは、デフォルト、または`/etc/wicked/client.xml`で定義されるシステム設定ソースを使用しています。

デバッグを有効にするには、/etc/sysconfig/network/configでWICKED_DEBUG_PARAMを設定します(これは将来変更される場合があります)。次に例を示します。

```
WICKED_DEBUG_PARAM="--debug most"
```

クライアントユーティリティを使用して、すべてのインタフェース、またはifnameで指定したインタフェースに関するインタフェース情報を表示します。

```
wicked show all
wicked show ifname
```

XML出力の場合は、以下を実行します。

```
wicked show-xml all
wicked show-xml ifname
```

1つのインタフェースを起動します。

```
wicked ifup eth0
wicked ifup wlan0
...
```

設定ソースが指定されていないため、wickedクライアントは、/etc/wicked/client.xmlで定義されている設定のデフォルトソースを確認します。

1. firmware: iBFT (iSCSI Boot Firmware Table)
2. compat: ifcfgファイル—互換性のため実装
3. wicked:PATH (デフォルト: /etc/wicked/ifconfig)に保存されるPATHネイティブのwicked XML設定フォーマット

特定のインタフェースに対してwickedがこれらのソースから取得した設定がすべて適用されます。重要度の順序は、firmware、compat、wickedの順です。これは将来、ifcfgの互換性要件が緩和されれば変更される可能性があります。

続いて、サンプルVLANインタフェースなど、関心の高いものを起動します。

```
wicked ifup --ifconfig ./samples/wicked/vlan-static.xml eth0.42
```

これは、「eth0.42」という名前のVLANインタフェースを起動します。VLANタグ42と数個のIPアドレスが静的に割り当てられています。動作しているかどうかを確認するため、以下を実行してみます。

```
ip addr show
ip route show
```

上のコマンドは、指定されたファイルからすべてのインタフェースの記述を取得し、「eth0.42」というインタフェースを起動します。このファイルに含まれているインタフェース1つだけであるため、インタフェース名の代わりに `all` を使用することもできます。名前が示すように、これは、このファイルに一覧にされているすべてのインタフェースを起動します。

単一のインタフェースを起動するため、クライアントはXML要素から、複数のサーバメソッドと引数を実行し、目的のインタフェースの状態を「up」に切り替えるようサーバに命令します。この操作により、まだVLANインタフェースがない場合は、ただちに作成されます。

同様の方法でインタフェースを停止します。

```
wicked ifdown eth0.42
```

インタフェースを停止して削除するには、以下を使用します。

```
wicked ifdown --delete --ifconfig ./samples/wicked/vlan-static.xml eth0.42
```

詳細については、`wicked` のマニュアルページを参照してください。

19.5.1.3 複数のインタフェースの起動

ボンドおよびブリッジの場合、1つのファイルにデバイストポロジ全体を定義し、それをまとめて起動します。これは、特にボンドにとって重要です。ボンドの場合、最初にスレーブデバイスを作成する必要があるためです(スレーブデバイスがVLANなどの仮想デバイスである場合)。

このようなシナリオの場合、1つのファイルでデバイストポロジを定義し、wickedを呼び出して、設定全体を起動します。例については、パッケージのマニュアル(`/usr/share/doc/packages/wicked`)の `samples/wicked/bridge-static.xml` を参照してください。この設定は、2つのVLANインタフェースから構築されたEthenetブリッジを定義します。これを起動するには、次を呼び出します。

```
wicked ifup --ifconfig ./samples/wicked/bridge-static.xml all
```

クライアントは適切な順序でデバイスを起動します。最初に2つのVLANインタフェースを作成してからブリッジを作成し、最後にVLANインタフェースをポートとしてブリッジに追加します。

19.5.1.4 増分変更の処理

`wicked`では、再設定のためにインタフェースを実際に停止する必要はありません(カーネルによって要求される場合を除く)。たとえば、静的に設定されたネットワークインタフェースに別のIPアドレスまたはルートを追加するには、インタフェース定義にIPアドレスを追加して、もう一度「ifup」操作を実行します。サーバは変更された設定のみを更新しようとします。これは、リンクレベルのオプション(デバイスMTUやMACアドレスなど)に加え、静的設定からDHCPに切り替える場合などはネットワークレベルの設定(アドレス、ルート、アドレス設定モードなど)にも適用されます。

もちろん、ブリッジやボンドなど複数の実デバイスを組み合わせる仮想インタフェースでは、処理は複雑になります。ボンドデバイスの場合、デバイスの稼働中に特定のパラメータを変更することはできません。これを行うと、エラーが発生します。

ただし、この状態でも、ボンドまたはブリッジの子デバイスを追加または削除したり、ボンドのプライマリインタフェースを選択したりする操作は有効です。

19.5.1.5 Wicked拡張機能: アドレス設定

`wicked`は、シェルスクリプトによって拡張可能な設計になっています。これらの拡張機能は、`config.xml`ファイルで定義できます。

現状では、複数の異なるクラスの拡張機能がサポートされています。

- リンク設定: クライアントによって提供される環境設定に従ってデバイスのリンク層を設定し、それを再び終了するスクリプトです。
- アドレス設定: デバイスのアドレス設定を管理するスクリプトです。通常、アドレス設定およびDHCPは、`wicked`自体で管理されますが、拡張機能によって実装できます。
- ファイアウォール拡張機能: これらのスクリプトでファイアウォールルールを適用できます。

通常、拡張機能には、開始および終了コマンド、オプションの「pid file」、およびスクリプトに渡される一連の環境変数があります。「」

これがどのように機能するかを説明するために、`etc/server.xml`で定義されているファイウォール拡張機能を取り上げます。

```
<dbus-service interface="org.opensuse.Network.Firewall">
 <action name="firewallUp"   command="/etc/wicked/extensions/firewall up"/>
 <action name="firewallDown" command="/etc/wicked/extensions/firewall down"/>

 <!-- default environment for all calls to this extension script -->
```

```
 <putenv name="WICKED_OBJECT_PATH" value="$object-path"/>
 <putenv name="WICKED_INTERFACE_NAME" value="$property:name"/>
 <putenv name="WICKED_INTERFACE_INDEX" value="$property:index"/>
</dbus-service>
```

拡張機能は、dbusサービスインタフェースにアタッチされ、このインタフェースのアクションに対して実行するコマンドを定義します。さらに、宣言によって、アクションに渡される環境変数を定義および初期化できます。

19.5.1.6 Wicked拡張機能: 環境設定ファイル

スクリプトを使用して環境設定ファイルの処理を拡張することもできます。たとえば、DNSのリースの更新は、最終的には、`server.xml` で動作が設定された `extensions/resolver` スクリプトで処理されます。

```
<system-updater name="resolver">
 <action name="backup" command="/etc/wicked/extensions/resolver backup"/>
 <action name="restore" command="/etc/wicked/extensions/resolver restore"/>
 <action name="install" command="/etc/wicked/extensions/resolver install"/>
 <action name="remove" command="/etc/wicked/extensions/resolver remove"/>
</system-updater>
```

更新内容が `wickedd` に届くと、システムアップデータルーチンがリースを解析して、適切なコマンド(`backup`、`install` など)をリゾルバスクリプトで呼び出します。これにより、`/sbin/netconfig` を使用してDNSを設定するか、フォールバックとして手動で `/etc/resolv.conf` を作成してDNSを設定します。

19.5.2 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。

19.5.2.1 /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従来の環境設定が含まれています。

注記: wickedおよびifcfg-*ファイル

wickedは、--ifconfigオプションの使用時にcompat:プレフィックスで互換性モードを指定した場合に、これらのファイルを読み込みます。/etc/wicked/client.xmlにあるSUSE Linux Enterprise Server 12のデフォルト設定に応じて、wickedは、/etc/wicked/ifconfig内のXML設定ファイルの前にこれらのファイルを読み込みます。

ifcfg-*ファイルには、起動モードやIPアドレスなどの情報が含まれています。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およびwirelessファイルのほとんどの変数をifcfg-*ファイルで使用できます。ただし、/etc/sysconfig/network/configの変数の大半はグローバル変数であり、ifcfgファイル内で上書きすることはできません。たとえば、NETWORKMANAGERやNETCONFIG_*は、グローバル変数です。

macvlanおよびmacvtabインタフェースの設定方法については、ifcfg-macvlanおよびifcfg-macvtapのマニュアルページを参照してください。たとえば、macvlanインタフェースでは、ifcfg-macvlan0を次のように設定します。

```
STARTMODE='auto'
MACVLAN_DEVICE='eth0'
#MACVLAN_MODE='vepa'
#LLADDR=02:03:04:05:06:aa
```

ifcfg.templateについては、19.5.2.2項 「/etc/sysconfig/network/config、/etc/sysconfig/network/dhcp、および/etc/sysconfig/network/wireless」を参照してください。

System z IBM System zは、USBをサポートしていません。インタフェースファイル名とネットワークエイリアスには、qethのようなSystem z固有の要素が含まれます。

19.5.2.2 /etc/sysconfig/network/config、/etc/sysconfig/network/dhcp、および/etc/sysconfig/network/wireless

configファイルには、ifup、ifdown、およびifstatusの動作に関する汎用的な設定が記述されています。また、dhcpにはDHCPの設定が、wirelessには無線LANカードの設定が記述されています。3つの環境設定ファイル内の変数にはコメントが付きます。/etc/sysconfig/network/config内の一部の変数は、ifcfg-*ファイルでも使用できます。このファイルでは、高い優先度が設定されます。/etc/sysconfig/network/ifcfg.templateファイルは、インタフェースごとに指定で

きる変数を一覧表示します。ただし、/etc/sysconfig/network/configの変数の大半はグローバル変数であり、ifcfgファイル内で上書きすることはできません。たとえば、NETWORKMANAGERやNETCONFIG_*は、グローバル変数です。

19.5.2.3 /etc/sysconfig/network/routesと/etc/sysconfig/network/ifroute-*

TCP/IPパケットのスタティックルーティングは、/etc/sysconfig/network/routesおよび/etc/sysconfig/network/ifroute-*ファイルによって決定されます。ホストへのルート、ゲートウェイ経由のホストへのルート、およびネットワークへのルートなど、さまざまなシステムタスクが必要とするすべてのスタティックルートは、/etc/sysconfig/network/routesファイルに指定できます。個別のルーティングが必要な各インタフェースに対して、付加環境設定ファイル/etc/sysconfig/network/ifroute-*を定義します。ワイルドカード(*)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  Options
```

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。ネットワークは、IPv4ルートでは10.10.0.0/16、IPv6ルートではfc00::/7のように、CIDR表記(関連付けられたルーティングプレフィックス長付きのアドレス)で記述する必要があります。キーワードのdefaultは、そのルートがゲートウェイと同じアドレスファミリ内のデフォルトゲートウェイであることを示しています。ゲートウェイのないデバイスの場合は、明示的な宛先0.0.0.0/0または::/0を使用します。

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列は非推奨になりました。これは、宛先のIPv4ネットマスクを示すために使用されていました。デフォルトルートであるIPv6ルートの場合、または第1列でプレフィックス長を使用する場合(CIDR表記)は、ここにダッシュ記号(-)を入力します。

第4列は、インタフェースの名前です。ダッシュ記号(-)を使用して空のままにすると、/etc/sysconfig/network/routesで意図しない動作を引き起こす場合があります。詳細については、routesのマニュアルページを参照してください。

第5列(オプション)では、特殊なオプションを指定することができます。詳細については、routesのマニュアルページを参照してください。

例 19.5 一般的なネットワークインタフェースとスタティックルートの例

```
# --- IPv4 routes in CIDR prefix notation:
```

```
# Destination     [Gateway]         -                   Interface
127.0.0.0/8       -                 -                   lo
204.127.235.0/24  -                 -                   eth0
default           204.127.235.41    -                   eth0
207.68.156.51/32  207.68.145.45     -                   eth1
192.168.0.0/16    207.68.156.51     -                   eth1

# --- IPv4 routes in deprecated netmask notation"
# Destination     [Dummy/Gateway]   Netmask             Interface
#
127.0.0.0         0.0.0.0           255.255.255.0       lo
204.127.235.0     0.0.0.0           255.255.255.0       eth0
default           204.127.235.41    0.0.0.0             eth0
207.68.156.51     207.68.145.45     255.255.255.255     eth1
192.168.0.0       207.68.156.51     255.255.0.0         eth1

# --- IPv6 routes are always using CIDR notation:
# Destination     [Gateway]                 -           Interface
2001:DB8:100::/64 -                         -           eth0
2001:DB8:100::/32 fe80::216:3eff:fe6d:c042 -            eth0
```

19.5.2.4 /etc/resolv.conf

/etc/resolv.conf 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)。search オプションでは、最大256文字で最大6つのドメインを指定できます。完全修飾でない名前を解決する場合は、search の各エントリを付加して完全修飾名の生成が試みられます。nameserver オプションでは、1行に1つずつ、最大3つのネームサーバを指定できます。コメントの先頭には、ハッシュマークまたはセミコロン記号(# または ;)を付加します。例については、例19.6「/etc/resolv.conf」を参照してください。

ただし、/etc/resolv.conf は、手動では編集しないでください。このファイルは、netconfig スクリプトで生成されます。YaSTを使用せずに静的DNS設定を定義するには、/etc/sysconfig/network/config ファイルの該当する変数を手動で編集します。

NETCONFIG_DNS_STATIC_SEARCHLIST
: ホスト名の検索に使用されるDNSドメイン名のリスト

NETCONFIG_DNS_STATIC_SERVERS
: ホスト名の検索に使用されるネームサーバのIPアドレスのリスト

NETCONFIG_DNS_FORWARDER
: 設定する必要のあるDNSフォワーダの名前。たとえば、bindまたはresolver

NETCONFIG_DNS_RESOLVER_OPTIONS
: /etc/resolv.confに記述される任意のオプション。例:

```
debug attempts:1 timeout:10
```

詳細については、resolv.confのマニュアルページを参照してください。

NETCONFIG_DNS_RESOLVER_SORTLIST
: 最大10項目のリスト。例:

```
130.155.160.0/255.255.240.0 130.155.0.0
```

詳細については、resolv.confのマニュアルページを参照してください。

netconfigでDNS環境設定を無効にするには、NETCONFIG_DNS_POLICY=''を設定します。netconfigの詳細については、netconfig(8)のマニュアルページ(man 8 netconfig)を参照してください。

例 19.6 /etc/resolv.conf

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

19.5.2.5 /sbin/netconfig

netconfigは、追加のネットワーク環境設定を管理するモジュール式ツールです。このツールは、事前定義されたポリシーに従って、 DHCPまたはPPPなどの自動設定メカニズムにより提供される設定と、静的に定義された設定をマージします。要求された変更は、netconfigモジュールの呼び出しによって適用されます。このモジュールは、環境設定ファイルの変更と、サービスまたは同様のアクションの再起動を行います。

`netconfig`は、3つの主要なアクションを認識します。`netconfig modify`コマンドと`netconfig remove`コマンドは、 DHCPやPPPなどのデーモンによって使用され、netconfigの設定値を提供したり、削除します。ユーザが使用できるのは、`netconfig update`コマンドだけです。

変更

`netconfig modify`コマンドは、現在のインタフェースとサービス固有の動的設定を変更し、ネットワーク設定を更新します。netconfigは、標準入力からか、または`--lease-file filename`オプションで指定されたファイルから設定を読み込み、システムのリブートまたは次の変更/削除アクションまで、それらの設定を内部的に保存します。同じインタフェースとサービスの組み合わせに関する既存設定は、上書きされます。インタフェースは、`-i interface_name`パラメータで指定されます。サービスは、`-s service_name`パラメータで指定されます。

削除

`netconfig remove`コマンドは、特定のインタフェースとコマンドの組み合わせに対する変更アクションによる動的設定を削除し、ネットワーク設定を更新します。インタフェースは、`-i interface_name`パラメータで指定されます。サービスは、`-s service_name`パラメータで指定されます。

`update`

`netconfig update`コマンドは、現在の設定で、ネットワーク設定を更新します。これは、ポリシーや静的環境設定が変更された場合に便利です。指定したサービスのみ(`dns`、`nis`、または`ntp`)を更新するには、`-m module_type`パラメータを使用します。

netconfigポリシーおよび静的環境設定は、手動またはYaSTで、`/etc/sysconfig/network/config`ファイル内で定義します。DHCPやPPPなどの自動設定ツールで提供された動的設定は、`netconfig modify`および`netconfig remove`のアクションで、これらのツールによって直接配信されます。NetworkManagerが有効な場合、netconfig (ポリシーモードが`auto`)は、NetworkManagerの設定のみを使用し、従来のifup方式で設定された他のインタフェースからの設定を無視します。NetworkManagerが設定を提供しない場合は、静的設定がフォールバックとして使用されます。NetworkManagerと`wicked`方式の混合使用はサポートされません。

`netconfig`の詳細については、`man 8 netconfig`を参照してください。

19.5.2.6 /etc/hosts

このファイル(例19.7「/etc/hosts」を参照)では、IP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一覧に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。

例 19.7 /etc/hosts

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

19.5.2.7 /etc/networks

このファイルには、ネットワーク名とネットワークアドレスの対応が記述されています。形式は、ネットワーク名をアドレスの前に指定すること以外は、hosts ファイルと同様です。詳細については、例19.8「/etc/networks」を参照してください。

例 19.8 /etc/networks

```
loopback    127.0.0.0
localnet    192.168.0.0
```

19.5.2.8 /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19.2「/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19.9「/etc/host.conf」を参照してください。

表 19.2 /ETC/HOST.CONFファイルのパラメータ

order hosts,bind	名前の解決の際、サービスがアクセスされる順序を指定します。有効な引数は次のとおりです(空白またはカンマで区切ります)。

	hosts: `/etc/hosts` ファイルを検索します。
	bind: ネームサーバにアクセスします。
	nis: NISを使用します。
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、ネットワークの環境設定にはまったく影響を与えません。
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)。ローカルドメインにある名前は `/etc/hosts` ファイルにありますが、付加されるドメイン名でも認識する必要がある場合には便利なオプションです。

例 19.9 `/etc/host.conf`

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

19.5.2.9 `/etc/nsswitch.conf`

GNU C Library 2.0を導入すると、Name Service Switch (NSS)も合わせて導入されます。詳細については、`nsswitch.conf(5)` manページおよび『The GNU C Library Reference Manual』を参照してください。

クエリの順序は、ファイル `/etc/nsswitch.conf` で定義します。`nsswitch.conf` の例については、例 19.10「`/etc/nsswitch.conf`」を参照してください。コメントの先頭には `#` 記号が付きます。この例では、`hosts` データベースの下のエントリは、要求がDNSを介して、`/etc/hosts` (`files`)に送信されることを意味しています(第22章 ドメインネームシステム参照)。

例 19.10 `/etc/nsswitch.conf`

```
passwd:     compat
```

```
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files
rpc:        files
ethers:     files
netmasks:   files
netgroup:   files nis
publickey:  files

bootparams: files
automount:  files nis
aliases:    files nis
shadow:     compat
```

NSSで利用できる「データベース」については、表19.3「/etc/nsswitch.confで利用できるデータベース」を参照してください。
NSSデータベースの環境設定オプションについては、表19.4「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。

表 19.3 /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。`man 5 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。
`netmasks`	ネットワークとそのサブネットマスクのリスト。サブネットを使用する場合のみ必要です。
`group`	`getgrent`によって使用されるユーザグループ。`group`のマニュアルページも参照してください。
`hosts`	`gethostbyname`および同類の関数によって使用されるホスト名とIPアドレス。

netgroup	アクセス許可を制御するための、ネットワーク内にある有効なホストとユーザのリスト。netgroup(5) manページを参照してください。
networks	ネットワーク名とアドレス。getnetentによって使用されます。
publickey	NFSとNIS+によって使用されるSecure_RPCの公開鍵と秘密鍵。
passwd	ユーザパスワード。getpwentによって使用されます。passwd(5) manページを参照してください。
protocols	ネットワークプロトコル。getprotoentによって使用されます。protocols(5) manページを参照してください。
rpc	リモートプロシージャコール名とアドレス。getrpcbynameおよび同様の関数によって使用されます。
services	ネットワークサービス。getserventによって使用されます。
shadow	ユーザのシャドウパスワード。getspnamによって使用されます。shadow(5) manページを参照してください。

表 19.4 NSS「データベース」の環境設定オプション

ファイル	直接アクセスファイル。たとえば/etc/aliases。
db	データベース経由のアクセス。
nis、nisplus	NIS。Book “Security Guide” 3 “Using NIS”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。

19.5.2.10 /etc/nscd.conf

このファイルは、nscd (name service cache daemon)の環境設定に使用します。`nscd(8)` および `nscd.conf(5)` マニュアルページを参照してください。デフォルトでは、nscdによって `passwd` と `groups` のシステムエントリがキャッシュされます。キャッシュが行われないと名前やグループにアクセスするたびにネットワーク接続が必要になるため、このキャッシュ処理は NIS や LDAP といったディレクトリサービスのパフォーマンスに関して重要な意味を持ちます。`hosts` はデフォルトではキャッシュされません。これは、nscd でホストをキャッシュすると、ローカルシステムで正引き参照と逆引き参照のルックアップチェックを信頼できなくなるからです。したがって、nscdを使用して名前をキャッシュするのではなく、キャッシュDNSサーバをセットアップします。

`passwd` オプションのキャッシュを有効にすると、新しく追加したローカルユーザが認識されるまで、通常、約15秒かかります。この待ち時間を短縮するには、次のコマンドを使用してnscdを再起動します。

```
systemctl restart nscd.service
```

19.5.2.11 /etc/HOSTNAME

`/etc/HOSTNAME` には、完全修飾ホスト名(FQHN)が含まれています。完全修飾ホスト名は、ドメイン名が付加されたホスト名です。このファイルに指定できるのは、ホスト名が設定されている1行のみです。このファイルはマシンのブート時に読み込まれます。

19.5.3 設定のテスト

設定内容を設定ファイルに書き込む前に、それをテストすることができます。テスト環境を設定するには、`ip` コマンドを使用します。接続をテストするには、`ping` コマンドを使用します。

`ip` コマンドは、ネットワーク設定を直接変更します。ただし、変更内容は環境設定ファイルに保存されません。正しい環境設定ファイルに変更内容を保存しない限り、変更したネットワーク設定は再起動時に失われてしまいます。

注記: `ifconfig`および`route`は廃止

`ifconfig`および`route`ツールは廃止されました。代わりに、`ip`を使用してください。たとえば、`ifconfig`では、インタフェース名は9文字に制限されます。

19.5.3.1 `ip`によるネットワークインタフェースの設定

`ip`は、ネットワークデバイス、ルーティング、ポリシールーティング、およびトンネルの表示と設定を行うツールです。

`ip`は非常に複雑なツールです。一般的には、`ip optionsobjectcommand`の形式で指定します。objectの部分には、次のオブジェクトを指定することができます。

リンク
: ネットワークデバイスを表します。

アドレス
: デバイスのIPアドレスを表します。

隣接
: このオブジェクトは、ARPまたはNDISCのキャッシュエントリを表します。

route
: ルーティングテーブルエントリを表します。

ルール
: ルーティングポリシーデータベース中のルールを表します。

maddress
: マルチキャストアドレスを表します。

mroute
: マルチキャストルーティングキャッシュエントリを表します。

tunnel
: IPトンネルを表します。

commandを指定しないと、デフォルトのコマンド(通常は`list`)が使用されます。

デバイスの状態を変更するには、`ip link set device_name command`コマンドを使用します。たとえば、デバイスeth0を無効にするには、`ip link set eth0 down`を実行します。このデバイスを再び有効にする場合は、`ip link set eth0 up`を実行します。

デバイスを有効にしたら、そのデバイスを設定することができます。デバイスのIPアドレスを使用する場合は、`ip addr add ip_address + dev device_name`を使用します。たとえば、インタフェースeth0にアドレス「 192.168.12.154/30」を設定し、標準のブロードキャスト(`brd`オプション)を使用する場合は、「`ip addr add 192.168.12.154/30 brd + dev eth0`」と入力します。

接続を実際に利用可能にするには、デフォルトゲートウェイの設定も必要です。システムのゲートウェイを設定するには、「`ip route add gateway_ip_address`」を入力します。あるIPアドレスを別のIPアドレスに変換するには、`nat`: `ip route add nat ip_address via other_ip_address`を使用します。

すべてのデバイスを表示する場合は、`ip link ls`を使用します。動作しているインタフェースだけを表示する場合は、`ip link ls up`を使用します。デバイスのインタフェース統計情報を印刷する場合は、「`ip -s link ls device_name`」と入力します。デバイスのアドレスを表示する場合は、「`ip addr`」と入力します。`ip addr`の出力には、デバイスのMACアドレスに関する情報も表示されます。すべてのルートを表示する場合は、`ip route show`を使用します。

`ip`の使用方法の詳細については、`ip help`を入力するか、または`ip(8)`マニュアルページを参照してください。`help`オプションは、すべての`ip`サブコマンドに関して利用できます。たとえば、`ip addr`のヘルプが必要な場合は、`ip addr help`と入力します。`ip`マニュアルについては、`/usr/share/doc/packages/iproute2/ip-cref.pdf`を参照してください。

19.5.3.2 pingを使った接続のテスト

`ping`コマンドは、TCP/IP接続が正常に動作しているかどうかを調べるための、標準ツールです。pingコマンドはICMPプロトコルを使って、小さなデータパケットECHO_REQUESTデータグラムを、宛先ホストに送信し、即時応答を要求します。これが機能した場合、`ping`はそのことを示すメッセージを表示します。これは、ネットワークリンクが機能していることを示します。

`ping`は、2台のコンピュータ間の接続機能をテストするだけでなく、接続品質に関する基本的な情報も提供します。ping例19.11「pingコマンドの出力」コマンドの実行結果例は、を参照してください。 最後から2番目の行に、転送パケット数、失われたパケット数、および`ping`の実行時間の合計が記載されています。

pingの宛先には、ホスト名またはIPアドレスを指定することができます。たとえば、`ping example.com`や`ping 192.168.3.100`のように指定します。pingコマンドを実行すると、<`Ctrl`–`C`>を押すまでの間、継続的にパケットが送信されます。

接続されているかどうかを確認するだけで良い場合は、`-c`オプションを使って送信するパケット数を指定することができます。たとえば、PINGを3パケットに制限する場合は、「`ping -c 3 example.com`」を入力します。

例 19.11 PINGコマンドの出力

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

デフォルトでは、pingは1秒ごとにパケットを送信します。間隔を変更するには、`-i` オプションを指定します。たとえば、pingの間隔を10秒に増やす場合は、「`ping -i 10 example.com`」と入力します。

複数のネットワークデバイスを持つシステムの場合、特定のインタフェースアドレスを指定してpingを実行することができます。その場合は、`-I` オプションを、選択したデバイスの名前とともに使用します。たとえば、`ping -I wlan1 example.com` と指定します。

pingのオプションと使用方法の詳細については、「`ping -h`」と入力するか、または `ping(8)` のマニュアルページを参照してください。

ヒント: IPv6アドレスのping

IPv6の場合は、`ping6` コマンドを使用します。ただし、リンクローカルアドレスをpingするには、`-I` でインタフェースを指定する必要があります。アドレスが `eth1` を介して到達可能な場合は、次のコマンドが有効です。

```
ping6 -I eth1 fe80::117:21ff:feda:a425
```

19.5.4 ユニットファイルと起動スクリプト

上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サービス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 `multi-user.target` のいずれかに切り替わったときに起動します。これらのユニットファイルの一部は、ネットワークプログラム用のユニットファイルと起動スクリプトで説明されています。`systemd` の詳細については、第10章 systemdデーモンを参照してください。`systemd` ターゲットの詳細については、`systemd.special` のマニュアルページ(`man systemd.special`)を参照してください。

`network.target`

`network.target` は、ネットワークのsystemdターゲットですが、その意味はシステム管理者が指定した設定により異なります。
詳細については、http://www.freedesktop.org/wiki/Software/systemd/NetworkTarget/ を参照してください。

`multi-user.target`

`multi-user.target` は、必要なすべてのネットワークサービスを持つ、マルチユーザシステムのsystemdターゲットです。

`xinetd.service`

xinetdを開始します。xinetdを使用すると、サーバサービスがシステム上で利用できるようになります。たとえば、FTP接続の開始時に必ずvsftpdを起動することができます。

`rpcbind.service`

RPCプログラム番号をユニバーサルアドレスに変換するrpcbindユーティリティを起動します。NFSサーバなどのRPCサービスで必要です。

`ypserv.service`

NISサーバを起動します。

`ypbind.service`

NISクライアントを起動します。

`/etc/init.d/nfsserver`

NFSサーバを起動します。

`/etc/init.d/postfix`

postfixプロセスを制御します。

19.6 ボンディングデバイスの設定

システムによって、通常のEthernetデバイスの規格のデータセキュリティ/可用性の要件を超えるネットワーク接続の実装が望ましいことがあります。その場合、数台のEthernetデバイスを集めて1つのボンディングデバイスを設定できます。

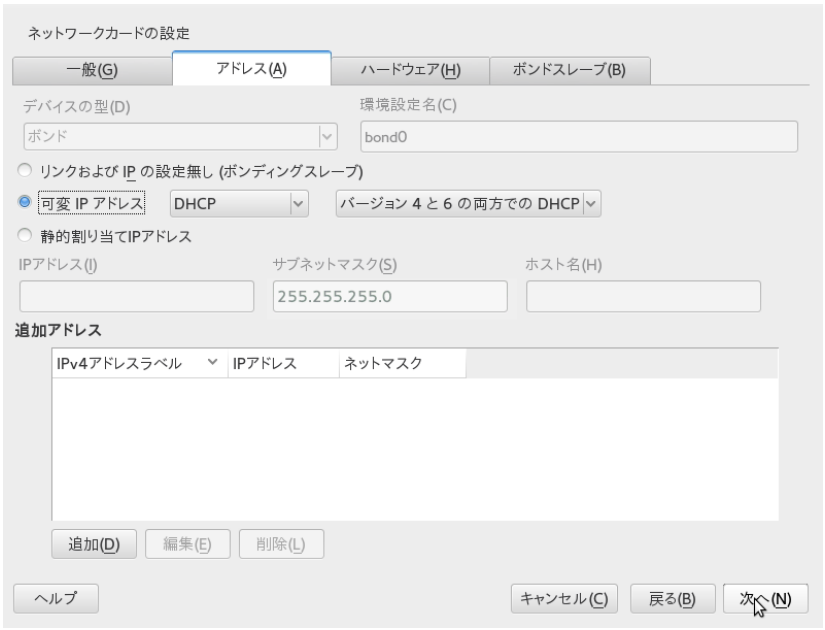
ボンディングデバイスの設定には、ボンディングモジュールオプションを使用します。ボンディングデバイスの振る舞いは、主にボンディングデバイスのモードによって影響されます。デフォルトの動作は、`mode=active-backup`であり、アクティブなスレーブに障害が発生すると、別のスレーブデバイスがアクティブになります。

ヒント: ボンディングとXen

ボンディングデバイスの使用が有用なのは、利用可能なネットワークカードが複数あるマシンの場合のみです。大半の設定では、Dom0でのみボンディング設定を使用する必要があることになります。VMゲストシステムに複数のネットワークカードが割り当てられている場合のみ、VMゲストでのボンド設定が役立つことがあります。

ボンディングデバイスを設定するには、次の手順に従います。

1. [YaST] › [ネットワークデバイス] › [ネットワーク設定]の順に選択します。
2. [追加]を使用し、[デバイスの型]を[ボンド]に変更します。[次へ]で続行します。



3. IPアドレスをボンディングデバイスに割り当てる方法を選択します。3つの方法から選択できます。

- IPアドレスなし
- 可変IPアドレス(DHCPまたはZeroconf)
- 固定IPアドレス

ご使用の環境に適合する方法を使用します。

4. [ボンドスレーブ]タブで該当するチェックボックスをオンにして、ボンドに含めるEthernetデバイスを選択します。
5. [ボンドドライバオプション]を編集します。設定には次のモードを使用できます。

   - balance-rr
   - active-backup
   - balance-xor
   - ブロードキャスト
   - 802.3ad
     `802.3ad`は、標準化されたLACP「IEEE 802.3adダイナミックリンク集約」モードです。「」
   - balance-tlb
   - balance-alb

6. パラメータ`miimon=100`が[ボンドドライバオプション]に追加されていることを確認します。このパラメータがないと、データの整合性が定期的にチェックされません。
7. [次へ]をクリックし、[OK]でYaSTを終了して、デバイスを作成します。

すべてのモードと他の多数のオプションの詳細は、「[Linux Ethernet Bonding Driver HOWTO]」に記載されています。このドキュメントは、kernel-sourceをインストールすると、`/usr/src/linux/Documentation/networking/bonding.txt`で読むことができます。

19.6.1 ボンディングスレーブのホットプラグ

特定のネットワーク環境(高可用性など)では、ボンディングスレーブインタフェースを別のものに置換しなければならないことがあります。ネットワークデバイスで頻繁に障害が発生するなどの理由があります。解決方法として、ボンディングスレーブのホットプラグを設定します。

ボンドは以下のように(man 5 ifcfg-bondingに従って)通常通りに設定されます。たとえば、

```
ifcfg-bond0
          STARTMODE='auto' # or 'onboot'
          BOOTPROTO='static'
          IPADDR='192.168.0.1/24'
          BONDING_MASTER='yes'
          BONDING_SLAVE_0='eth0'
          BONDING_SLAVE_1='eth1'
          BONDING_MODULE_OPTS='mode=active-backup miimon=100'
```

スレーブはSTARTMODE=hotplugおよびBOOTPROTO=noneで指定されます。

```
ifcfg-eth0
          STARTMODE='hotplug'
          BOOTPROTO='none'

ifcfg-eth1
          STARTMODE='hotplug'
          BOOTPROTO='none'
```

BOOTPROTO=noneはethtoolオプション(指定した場合)を使用しますが、ifup eth0にはリンクアップを設定しません。これは、スレーブインタフェースがボンドマスターによって制御されるためです。

STARTMODE=hotplugにより、スレーブインタフェースが利用可能になるとすぐに、ボンドに自動的に追加されます。

MACアドレスではなく、バスIDでデバイスを照合するように、/etc/udev/rules.d/70-persistent-net.rulesのudevルールを変更する必要があります(hwinfo --netcardに表示されるudev KERNELSキーワードを「SysFS BusID」とします)。これによって障害のあるハードウェアを置換(同じスロットで、MACが異なるネットワークカードを使用)できるようになり、ボンドがすべてのスレーブのMACアドレスを変更するので混乱を避けられます。

次に例を示します。

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
KERNELS=="0000:00:19.0", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"
```

ブート時に network.service はホットプラグスレーブを待機しませんが、ボンドの準備が整うのを待機します。これには少なくとも1つのスレーブが利用可能であることが必要です。スレーブインタフェースの1つがシステムから削除されると(NICドライバからアンバインド、NICドライバの rmmod 、または実際のPCIホットプラグ取り外し)、カーネルによってボンドから自動的に削除されます。システムに新しいカードが追加されると(スロットのハードウェアが置換されると)、udev は、バスベースの永続名規則を使って名前をスレーブ名に変更し、ifup を呼び出します。ifup 呼び出しによって、ボンドに自動的に追加されます。

20 SLP

ネットワーククライアントを設定するには、ネットワーク上で提供されるサービス(印刷やLDAPなど)に関する詳しい知識が必要です。ネットワーククライアントでこのようなサービスを容易に設定できるようにするため、「サービスロケーションプロトコル」(SLP)が開発されました。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、および設定データを通知します。このような通知情報を利用して、SLPをサポートする各種アプリケーションを自動的に設定することができます。

SUSE® Linux Enterprise Serverは、SLPによって提供されるインストールソースを使用するインストールをサポートしています。また、多くのシステムサービスは、統合SLPをサポートしています。ご利用のシステムでインストールサーバ、ファイルサーバ、プリントサーバなどの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。SLPサポートを提供するサービスには、cupsd、login、ntp、openldap2、postfix、rpasswd、rsyncd、saned、sshd (fish経由)、vnc、およびypservがあります。

ネットワーククライアントでSLPサービスを使用するのに必要なすべてのパッケージは、デフォルトでインストールされます。ただし、SLPによりサービスを「提供する」場合は、`openslp-server` パッケージがインストールされていることを確認します。

20.1 SLPフロントエンド`slptool`

`slptool` は、SLPサービスを問い合わせて登録するコマンドラインツールです。このクエリ機能は、診断を行う場合に便利です。次に、`slptool` で最も重要なサブコマンドを示します。`slptool --help` を実行すると、使用可能なすべてのオプションと機能のリストが表示されます。

findsrvtypes
: ネットワーク上で利用可能なすべてのサービスタイプのリストを表示します。

```
tux >  slptool findsrvtypes
service:install.suse:nfs
service:install.suse:ftp
service:install.suse:http
service:install.suse:smb
service:ssh
service:fish
service:YaST.installation.suse:vnc
service:smtp
```

```
service:domain
service:management-software.IBM:hardware-management-console
service:rsync
service:ntp
service:ypserv
```

findsrvs `service type`

`service type`を提供しているすべてのサーバのリストを表示します。

```
tux >  slptool findsrvs service:ntp
service:ntp://ntp.example.com:123,57810
service:ntp://ntp2.example.com:123,57810
```

findattrs `service type`//`host`

`host`上の`service type`の属性のリストを表示します。

```
tux >  slptool findattrs service:ntp://ntp.example.com
(owner=tux),(email=tux@example.com)
```

register `service type`//`host`:`port` "(`attribute=value`),(`attribute=value`)"

オプションの属性リストを使用して`host`上の`service type`を登録します。

```
slptool register service:ntp://ntp.example.com:57810 \
"(owner=tux),(email=tux@example.com)"
```

deregister `service type`//`host`

`host`上の`service type`を登録解除します。

```
slptool deregister service:ntp://ntp.example.com
```

詳細については、`slptool --help`を実行してください。

20.2 SLPによるサービスの提供

SLPサービスを提供するには、SLPデーモン(`slpd`)が動作している必要があります。SUSE Linux Enterprise Serverのほとんどのシステムサービスと同様に、`slpd`は別の起動スクリプトを使用して制御されます。インストール後に、このデーモンはデフォルトで非アクティブになります。現在のセッショ

ンでこのデーモンを有効にするには、`sudo systemctl start slpd.service`を実行します。システムの起動時に`slpd`を有効にする必要がある場合は、`sudo systemctl enable slpd.service`を実行します。

SUSE Linux Enterprise Serverのアプリケーションの多くは`libslp`ライブラリを使用することで、最初から統合SLPをサポートしています。サービスがSLPサポートでコンパイルされていない場合は、SLPを介して利用できるように次の方法のいずれかを使用してください。

/etc/slp.reg.d による静的登録

新規サービスに個別の登録ファイルを作成します。次の例は、スキャナサービスを登録します。

```
## 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TービスURLと分けるようにします。0 から 65535 で登録期間の値を設定します。 `0`の場合は登録する必要がありません。`65535`はすべての制限を削除します。
登録ファイルにはまた、2つの変数`watch-port-tcp`および`description`が含まれます。`watch-port-tcp`は、SLPサービスアナウンスとリンクして、`slpd`にサービスのステータスをチェックさせることにより、関連サービスがアクティブかどうかを確認します。2つ目の変数には、サービスに関するさらに詳細な説明が含まれており、正しいブラウザを使用している場合に表示されます。

ヒント: YaSTとSLP

インストールサーバ、YOUサーバなどのようにYaSTが処理を行うサービスの一部では、モジュールダイアログでSLPがアクティブになった時点で自動的にこの登録が実行されます。続いてYaSTはこれらのサービスの登録ファイルを作成します。

`/etc/slp.reg`による静的登録

この方法と、`/etc/slp.reg.d`による手続きの唯一の違いは、すべてのサービスが中央のファイルにグループ化されることです。

`slptool`による動的登録

設定ファイルなしでサービスを動的に登録する必要がある場合は、slptoolコマンドラインユーティリティを使用します。このユーティリティを使用すると、`slpd`を再起動せずに既存の提供サービスを登録解除することもできます。詳細については、20.1項 「SLPフロントエンド`slptool`」を参照してください。

20.2.1 SLPインストールサーバのセットアップ

ネットワーク内でSLP経由でインストールデータをアナウンスすると、サーバのIPアドレスやインストールメディアのパスといったインストールデータがSLPクエリによって自動的に要求されるため、ネットワークインストールが大幅に容易になります。手順については、ブック「導入ガイド」14「リモートインストール」14.2「インストールソースを保持するサーバのセットアップ」を参照してください。

20.3 詳細情報

RFC 2608、2609、2610

一般的にRFC 2608はSLPの定義を取り扱います。RFC 2609は、使用されるサービスURLの構文を詳細に扱います。またRFC 2610ではSLPを使用したDHCPについて説明しています。

http://www.openslp.org

OpenSLPプロジェクトのホームページです。

`/usr/share/doc/packages/openslp`

このディレクトリには、SUSE Linux Enterprise Serverの詳細を含む`README.SuSE`、RFC、および2つの入門的なHTMLドキュメントなど、`openslp-server`パッケージ付属のSLPのドキュメントが格納されています。SLP機能を使用するプログラマに役立つより詳しい情報については、SUSEソフトウェア開発キットに付属の`openslp-devel`パッケージに含まれる『プログラマガイド』を参照してください。

21 NTPによる時刻の同期

NTP (network time protocol)メカニズムは、システムの時刻をネットワーク上で同期させるためのプロトコルです。最初に、マシンは信頼できる時刻を持つサーバに時刻を照会できます。次に、ネットワーク上の他のコンピュータがこのマシン自体に対し、時刻を照会できます。目的は2つあり、絶対的な時間を維持することと、ネットワーク内のすべてのマシンのシステム時刻を同期させることです。

正確なシステム時刻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クロックがデータベースやクラスタなどのアプリケーション要件に合致しないことがよくあります。システムタイムを手動で修正することは時に問題を発生させる可能性があります。たとえば、時間を逆廻りに戻すことで重要なアプリケーションの誤動作を誘発することもあります。ネットワーク内では、すべてのマシンのシステムタイムを同期させることが通常必要とされますが、手動での時刻調整はよい方法ではありません。NTPには、これらの問題を解決するメカニズムがあります。NTPサービスは、ネットワーク内の信頼できるタイムサーバのヘルプによって、システム時間を継続的に調整します。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。

21.1 YaSTでのNTPクライアントの設定

`ntp`パッケージ付属のNTPデーモン(`ntpd`)は、ローカルコンピュータを時間の参照に使用するように事前設定されています。ただし、ハードウェアクロックは、より正確な時間ソースが利用できない場合の予備としてのみ使用されます。YaSTを利用すれば、NTPクライアントを簡単に設定することができます。

21.1.1 基本的な設定

YaST NTPクライアントの設定([ネットワークサービス] › [NTP環境設定])は、タブで構成されています。`ntpd`の起動モードと照会先のサーバは、[一般的な設定]タブで設定します。

[手動でのみ]
: [手動でのみ]は、`ntpd`デーモンを手動で開始する場合に選択します。

[今すぐ開始し、システム起動時に開始するよう設定]
: システムのブート時に自動的に`ntpd`を起動するには、[今すぐ開始し、システム起動時に開始するよう設定]を選択します。この設定をお勧めします。

21.1.2 基本的な設定の変更

[一般の設定]タブの下部には、クライアントに対するサーバおよび時刻情報のその他の情報源が表示されます。必要に応じて、[追加]、[削除]、および[編集]を使用してこのリストを変更します。[]Display Log]では、クライアントのログファイルを表示できます。

時刻情報の情報源を追加するには、[追加]をクリックします。表示されるダイアログで、時刻同期に使用する情報源のタイプを選択します。次のオプションを指定できます。

図 21.1 YAST: NTPサーバ

サーバ

[選択]プルダウンリスト(図21.1「YaST: NTPサーバ」参照)で、ローカルネットワーク上のタイムサーバ([ローカルNTPサーバ])または目的のタイムゾーンを担当する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設定するか決定します。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、[受諾]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、[受諾]をクリックしてダイアログを閉じます。メインダイアログの[テスト]を使用して、選択されているサーバの可用性をテストします。[オプション]では、`ntpd`の追加オプションを指定できます。[Access Control Options]を使用すると、コンピュータ上で実行するデーモンによりリモートコンピュータが実行可能なアクションを制限できます。このフィールドは、[セキュリティの設定]タブで[NTP サービスを設定したサーバに制限する]にチェックマークを入れた後でのみ有効にな

ります(図21.2「高度なNTP設定:セキュリティの設定」参照)。このオプションは、/etc/ntp.conf 内の restrict 節に対応します。たとえば nomodify notrap noquery は、サーバがコンピュータのNTP設定を変更し、NTPデーモンのトラップ機能(リモートイベントのログ記録機能)を使用することを拒否します。自身の管理下にないサーバについては(たとえばインターネット上のサーバなど)、こうした制限を適用することをお勧めします。
詳細については、/usr/share/doc/packages/ntp-doc (ntp-doc パッケージの一部)を参照してください。

ピア

ピアは、対称的な関係が確立されたコンピュータで、タイムサーバとクライアントの両方の役割を果たします。サーバの代わりに、同じネットワーク内のピアを使用するには、そのピアシステムのアドレスを入力します。ダイアログのそれ以外の内容は[サーバ]ダイアログと同じです。

ラジオクロック

時刻同期にシステムのラジオクロックを使用するには、クロックタイプ、ユニット番号、デバイス名、およびその他のオプションをこのダイアログで指定します。ドライバを微調整するには、[ドライバの調整]をクリックします。ローカルラジオクロックの動作の詳細については、/usr/share/doc/packages/ntp-doc/refclock.html を参照してください。

ブロードキャストの発信

時刻情報とクエリは、ネットワーク上にブロードキャストすることができます。このダイアログでは、このブロードキャストの送信先を指定します。電波時計のような信頼できる時刻ソースがない限りブロードキャストをアクティブにしないでください。

ブロードキャストの着信

クライアントで情報をブロードキャスト経由で受け取る場合は、どのアドレスからのパケットを受け入れるかをこのフィールドに指定します。

図 21.2 高度なNTP設定:セキュリティの設定

[セキュリティの設定]タブで(図21.2「高度なNTP設定:セキュリティの設定」参照)、`ntpd`をchroot jailで起動するかどうか指定します。デフォルトでは、[Run NTP Daemon in Chroot Jail]が選択されています。このオプションは、攻撃によってシステム全体が危険な状態に陥ることを防ぐので、`ntpd`が攻撃された場合のセキュリティを強化します。

[NTPサービスを設定したサーバに制限する]は、リモートコンピュータがユーザのコンピュータのNTP設定を表示および変更すること、およびリモートイベントログのトラップ機能を使用することを拒否し、それによってシステムのセキュリティを向上させます。[一般の設定]タブの時間ソースのリストで、個別のコンピュータに対するアクセス制御オプションを上書きしない限り、こうした制限は有効になるとすべてのリモートコンピュータに適用されます。他のすべてのリモートコンピュータでは、ローカルタイムのクエリのみが許可されます。

SuSEfirewall2がアクティブな場合、[ファイアウォールでポートを開く]を有効にします(デフォルト)。ポートを閉じたままにすると、タイムサーバと接続を確立することはできません。

21.2 ネットワークでのntpの手動設定

ネットワーク内のタイムサーバを使用するには、serverパラメータを設定するのが最も簡単です。たとえば、タイムサーバ`ntp.example.com`がネットワークから接続可能な場合、その名前をファイル`/etc/ntp.conf`に行として追加します。

```
server ntp.example.com
```

別のタイムサーバを追加するには、別の行にキーワードの「server」を挿入します。systemctl start ntp.serviceコマンドでntpdを初期化後、時間が安定し、ローカルコンピュータのクロックを修正するドリフトファイルが作成されるまで、約1時間かかります。ドリフトファイルを用いることで、バードウェアクロックの定誤差はコンピュータの電源が入った時点で、すぐに算出されます。修正はすぐに反映されるため、システム時刻がより安定します。

NTP機構をクライアントとして使用するには、2種類の方法があります。まず、クライアントは既知のサーバに定期的に時間を照会することができます。クライアント数が多い場合、この方法はサーバの過負荷を引き起こす可能性があります。2つ目は、ネットワークでブロードキャストを行う時刻サーバから送信されるNTPブロードキャストを、クライアントが待機する方法です。この方法には不利な面があります。サーバの精度が不明なこと、そしてサーバから送信される情報が誤っていた場合、深刻な問題が発生する可能性があることです。

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。

21.3 ランタイム時の動的時刻同期

ネットワークに接続せずにシステムが起動すると、ntpdは起動しますが、設定ファイルで設定されたタイムサーバのDNS名を解決できません。これは、暗号化されたWi-Fiでネットワークマネージャを使用するときに発生します。

ランタイム時にntpdでDNS名を解決するには、dynamicオプションを設定する必要があります。ネットワークが起動後に確立されると、ntpdは再度名前を検索し、時刻を取得するタイムサーバに到達します。

/etc/ntp.confを手動で編集して、dynamicを1つ以上のserverエントリに追加します。

```
server ntp.example.com dynamic
```

または、YaSTを使用して、次の手順に従います。

1. YaSTで、[ネットワークサービス] › [NTP環境設定]の順にクリックします。
2. 設定するサーバを選択します。[編集]をクリックします。
3. [オプション]フィールドを有効にして、[dynamic]を追加します。他のオプションが入力されている場合は、スペースで区切ります。

4. [OK]をクリックして、編集ダイアログを閉じます。前の手順を繰り返して、必要に応じてすべてのサーバを変更します。

5. 最後に、[OK]をクリックして設定を保存します。

21.4 ローカルリファレンスクロックの設定

`ntpd`ソフトウェアパッケージには、ローカルリファレンスクロックに接続するためのドライバが含まれています。サポートされているクロックのリストは、`ntp-doc`パッケージの`/usr/share/doc/packages/ntp-doc/refclock.htm`ファイルに記載されています。各ドライバには、番号が関連付けられています。NTPでは、実際の設定は疑似IPアドレスを使用して行われます。クロックは、ネットワークに存在しているものとして`/etc/ntp.conf`ファイルに入力されます。このため、これらのクロックには`127.127.t.u`という形式の特別なIPアドレスが割り当てられます。ここで、`t`はクロックのタイプを示し、使用されているドライバを決定します。`u`はユニットのタイプを示し、使用されているインタフェースを決定します。

通常、各ドライバは設定をより詳細に記述する特別なパラメータを持っています。`/usr/share/doc/packages/ntp-doc/driverNN.html`(ここで`NN`はドライバの番号)ファイルは、特定のクロックタイプの情報を提供します。たとえば、「タイプ 8」クロック(シリアルインタフェース経由のラジオクロック)はクロックをさらに細かく指定する追加モードを必要とします。また、Conrad DCF77レシーバモジュールはモード 5です。このクロックを優先参照として使用するには、キーワード`prefer`を指定します。Conrad DCF77レシーバモジュールの完全な`server`行は次のようになります。

```
server 127.127.8.0 mode 5 prefer
```

他のクロックも同じパターンで記述されます。`ntp-doc`パッケージのインストール後は、ntpのマニュアルを`/usr/share/doc/packages/ntp-doc`ディレクトリで参照できます。ドライバパラメータについて説明するドライバページへのリンクは、ファイル`/usr/share/doc/packages/ntp-doc/refclock.htm`に記述されています。

21.5 ETR(External Time Reference)とのクロックの同期

ETR(External Time Reference)とのクロック同期のサポートを利用できます。ETRは、2**20(2の20乗)マイクロ秒ごとに、発振器信号と同期信号を送信して、すべての接続先サーバのTODクロックの同期を保ちます。

可用性のため、2ユニットのETRをコンピュータに接続できます。クロックが同期チェックの許容値を超えた場合は、すべてのCPUがマシンをチェックし、クロックが同期していないことを示します。この事態が発生した場合は、XRC対応デバイスへのすべてのDASD I/Oがクロックの再同期まで停止します。

ETRサポートは2つの `sysfs` 属性を介して有効化されます。`root` として次のコマンドを実行します。

```
echo 1 > /sys/devices/system/etr/etr0/online
echo 1 > /sys/devices/system/etr/etr1/online
```

22 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192.168.2.100がホスト名 `jupiter` 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、19.3項「ネームレゾリューション」でDNSに関する一般的な説明を参照してください。次の設定例は、デフォルトのDNSサーバであるBINDについて示しています。

22.1 DNS用語

ゾーン

ドメインのネームスペースは、ゾーンと呼ばれる領域に分割されます。たとえば、`example.com` の場合は、`com` ドメインの `example` セクション(つまりゾーン)を表します。

DNSサーバ

DNSサーバは、ドメインの名前とIP情報を管理するサーバです。マスタゾーン用にプライマリDNSサーバ、スレーブゾーン用にセカンダリサーバ、またはキャッシュ用にいずれのゾーンも持たないスレーブサーバを持つことできます。

マスタゾーンのDNSサーバ

マスタゾーンにはネットワークからのすべてのホストが含まれ、DNSサーバのマスタゾーンにはドメイン内のすべてのホストに関する最新のレコードが格納されます。

スレーブゾーンのDNSサーバ

スレーブゾーンはマスタゾーンのコピーです。スレーブゾーンのDNSサーバは、ゾーン転送操作によりマスタサーバからゾーンデータを取得します。スレーブゾーンのDNSサーバは、有効なゾーンデータである(期限切れでない)限り、ゾーンに適切に応答します。スレーブがゾーンデータの新規コピーを取得できない場合、ゾーンへの応答を停止します。

フォワーダ

フォワーダは、DNSサーバがクエリに回答できない場合に、そのクエリの転送先になるDNSサーバです。1つの環境設定内で複数の設定ソースを有効にするには、`netconfig` を使用します(`man 8 netconfig` も参照)。

レコード

レコードは、名前とIPアドレスに関する情報です。サポートされているレコードおよびその構文は、BINDのドキュメントで説明されています。次は、特別なレコードの一部です。

NSレコード

NSレコードは、指定のドメインゾーンの担当マシンをネームサーバに指定します。

MXレコード

MX(メール交換)レコードは、インターネット上でメールを転送する際に通知するマシンを説明します。

SOAレコード

SOA (Start of Authority)レコードは、ゾーンファイル内で最初のレコードです。SOAレコードは、DNSを使用して複数のコンピュータ間でデータを同期化する際に使用されます。

22.2 インストール

DNSサーバをインストールするには、YaSTを起動してから、[ソフトウェア] › [ソフトウェア管理]の順に選択します。[表示] › [パターン]の順に選択して、[DHCPおよびDNSサーバ]を選択します。依存関係のあるパッケージのインストールを確認して、インストールプロセスを完了します。

22.3 YaSTでの設定

YaST DNSモジュールを使用して、ローカルネットワーク用にDNSサーバを設定します。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2、3の決定を行う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、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(ACLのセットアップ、ロギング、TSIGキーなどのオプション)を処理できます。

22.3.1 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環境設定モードに入ることができます。

1. モジュールを初めて起動すると、図22.1「DNSサーバのインストール:フォワーダの設定」のような[フォワーダの設定]ダイアログが表示されます。[ローカルDNS解決ポリシー]を使用して、次のオプションを設定できます。

   - [フォワーダのマージは無効です]
   - [自動マージ]

- [フォワーダのマージは有効です]
- [カスタム設定]—[カスタム設定]をオンにした場合は、[カスタムポリシー]を指定できます。デフォルトでは([自動マージ]が選択されている場合)、[カスタムポリシー]は[`auto (自動)`]に設定されますが、ここで、インタフェース名を設定したり、2つの特殊なポリシー名`STATIC`および`STATIC_FALLBACK`の一方を選択したりできます。

[ローカルDNS解決フォワーダ]で、使用するサービスとして、[システムネームサーバを使用しています]、[このネームサーバ(バインド)]、または[ローカルdnsmasqサーバ]のいずれかを指定します。
これらのすべての設定の詳細については、`man 8 netconfig`を参照してください。

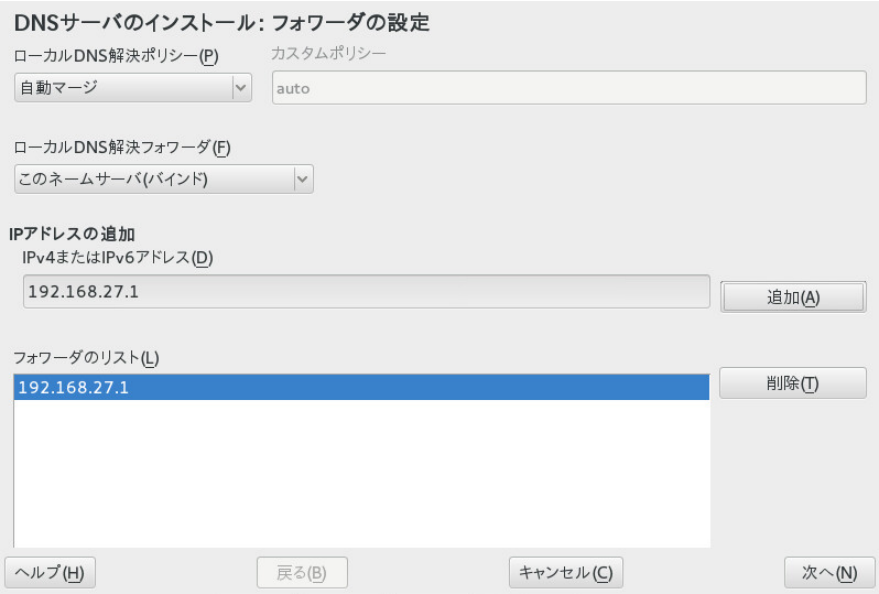


図 22.1 DNSサーバのインストール:フォワーダの設定

フォワーダは、ご使用のDNSサーバが回答できないクエリの送信先とするDNSサーバです。フォワーダのIPアドレスを入力して、[追加]をクリックします。

2. [DNSゾーン]ダイアログは、複数の部分で構成されており、22.6項「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。新しいゾーンの場合は、[名前]にゾーン名を入力します。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、[タイプ](マスタ、スレーブ、または転送)を選択します。図22.2「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、[Edit]をクリックします。ゾーンを削除するには、[Delete]をクリックします。

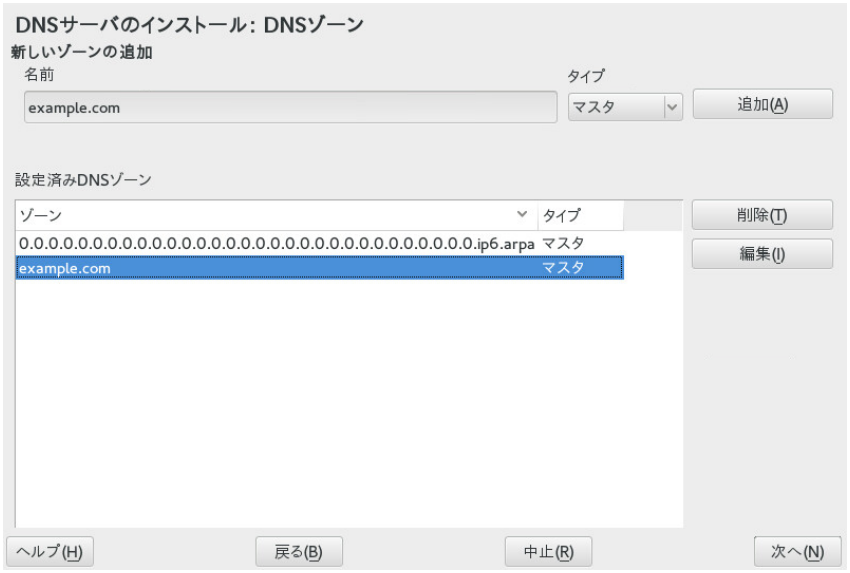


図 22.2 DNSサーバのインストール:DNSゾーン

3. 最後のダイアログでは、[ファイアウォールで開いているポート]をクリックして、ファイアウォールのDNSポートを開くことができます。次に、ブート時にDNSサーバ　を起動するかどうか([オン]か、[オフ]か)を決定します。LDAPサポートを有効にすることもできます。詳細については、図 22.3「DNSサーバのインストール:完了ウィザード」を参照してください。

図 22.3 DNSサーバのインストール:完了ウィザード

22.3.2 エキスパート設定

YaSTの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。

22.3.2.1 起動

[起動]では、DNSサーバをシステムのブート中に起動するか、それとも手動で起動するか指定します。DNSサーバをすぐに起動するには、[[今すぐDNSサーバを起動する]を選択します。DNSサーバを停止するには、[[今すぐDNSサーバを停止する]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、今すぐDNSサーバをリロードする]を選択します。ファイアウォールのDNSポートを開くには[ファイアウォール内でポートを開く]を、ファイアウォールの設定を変更するには[Firewall Details]をクリックします。

[LDAPサポートを有効にする]を選択すると、ゾーンファイルがLDAPデータベースによって管理されるようになります。ゾーンデータを変更してそれがLDAPデータベースに書き込まれると、設定を再ロードするように要求されます。DNSサーバを再起動すると、変更がすぐに反映されます。

22.3.2.2 フォワーダ

ローカルDNSサーバは、要求に応答できない場合、要求を[フォワーダ]に転送しようとします(そのように設定されている場合)。このフォワーダは、手動で、[Forwarder List]に追加できます。フォワーダが、ダイアルアップ接続でのように静的でない場合は、[netconfig]が設定を処理します。netconfigの詳細については、`man 8 netconfig`を参照してください。

22.3.2.3 基本的なオプション

このセクションでは、基本的なサーバオプションを設定します。[オプション]メニューから目的の項目を選択して、対応するテキストボックスに値を指定します。新しいエントリを追加するには、[追加]を選択してください。

22.3.2.4 ログ

DNSサーバがログに記録する内容とログの方法を設定するには、[ログ記録]を選択します。[Log Type]に、DNSサーバがログデータを書き込む場所を指定します。システム全体のログを使用する場合は[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]を選択します。別のファイルを指定する場合は、ログファイルの名前、最大サイズ(メガバイト(MB))、および保管するバージョンの数も指定します。

[追加ログ]には、さらに詳細なオプションが用意されています。[すべてのDNS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。詳細については、図22.4「DNSサーバ:ログの記録」を参照してください。

図 22.4 DNSサーバ:ログの記録

22.3.2.5 ACL

このダイアログでは、アクセス制限を強制するACL(アクセス制御リスト)を定義します。[名前]に個別名を入力したら、次の形式で、[値]にIPアドレス(ネットマスクは省略可)を指定します。

```
{ 192.168.1/24; }
```

設定ファイルの構文に従って、アドレスの末尾にはセミコロンを付け、中カッコで囲む必要があります。

22.3.2.6 TSIGキー

TSIG (トランザクションシグネチャー)の主な目的は、DHCPおよびDNSサーバ間で安全な通信を行うことです。22.8項「安全なトランザクション」を参照してください。

TSIGキーを生成するには、[キーID]フィールドに個別名を入力し、キーを格納するファイルを[ファイル名]フィールドに入力します。[生成]をクリックすると、選択内容が確定されます。

作成済みのキーを使用するには、[キーID]フィールドを空白のままにして、[ファイル名]で、そのキーが保存されているファイルを選択します。その後、[追加]をクリックすると、入力内容が確定されます。

22.3.2.7 DNSゾーン(スレーブゾーンの追加)

スレーブゾーンを追加するには、[DNSゾーン]を選択し、ゾーンタイプに[スレーブ]を選択し、新規ゾーンの名前を書き込み、[追加]をクリックします。

[マスタDNSサーバのIP]の下の[ゾーンエディタ]サブダイアログで、スレーブがデータをプル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。

22.3.2.8 DNSゾーン(マスタゾーンの追加)

マスタゾーンを追加するには、[DNSゾーン]を選択し、ゾーンタイプに[マスタ]を選択し、新規ゾーンの名前を書き込み、[追加]をクリックします。マスタゾーンの追加時には、逆引きゾーンも必要です。たとえば、ゾーン `example.com` (サブネット `192.168.1.0/24` 内のホストをポイントするゾーン)を追加する際には、カバーされるIPアドレス範囲の逆引きゾーンも追加する必要があります。定義上、このゾーンの名前は、`1.168.192.in-addr.arpa` となります。

22.3.2.9 DNSゾーン(マスタゾーンの編集)

マスタゾーンを編集するには、[DNSゾーン]を選択し、テーブルからマスタゾーンを選択し、[編集]をクリックします。このダイアログには、[基本](最初に表示される)、[NSレコード]、[MXレコード]、[SOA]、および[レコード]のページがあります。

に示す基本ダイアログを使用すると、ダイナミックDNSの設定と、クライアントおよびスレーブネームサーバへのゾーン転送に関するアクセスオプションを定義できます。図22.5「DNSサーバ:　ゾーンエディタ(基本)」ゾーンの動的更新を許可するには、[動的アップデートの許可]および対応するTSIGキーを選択します。このキーは、更新アクションの開始前に定義しておく必要があります。ゾーン転送を有効にするには、対応するACLを選択します。ACLは事前に定義しておく必要があります。

[基本]ダイアログで、ゾーン転送を有効にするかどうか選択します。リストされたACLを使用して、ゾーンをダウンロードできるユーザを定義します。

図 22.5 DNSサーバ: ゾーンエディタ(基本)

ゾーンエディタ(NSレコード)

[レコード]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。詳細については、図22.6「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。

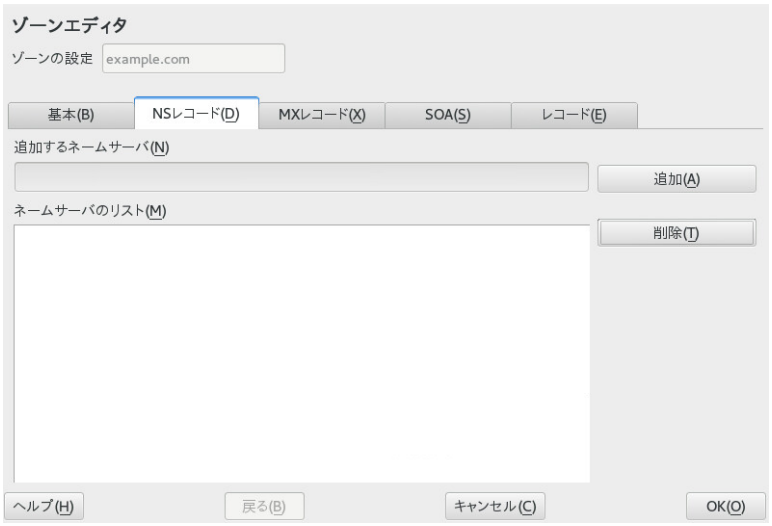


図 22.6 DNSサーバ:ゾーンエディタ(NSレコード)

ゾーンエディタ(MXレコード)

現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。詳細については、図22.7「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。

図 22.7 DNSサーバ:ゾーンエディタ(MXレコード)

ゾーンエディタ(SOA)

このページでは、SOA (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22.6「/var/lib/named/example.com.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。

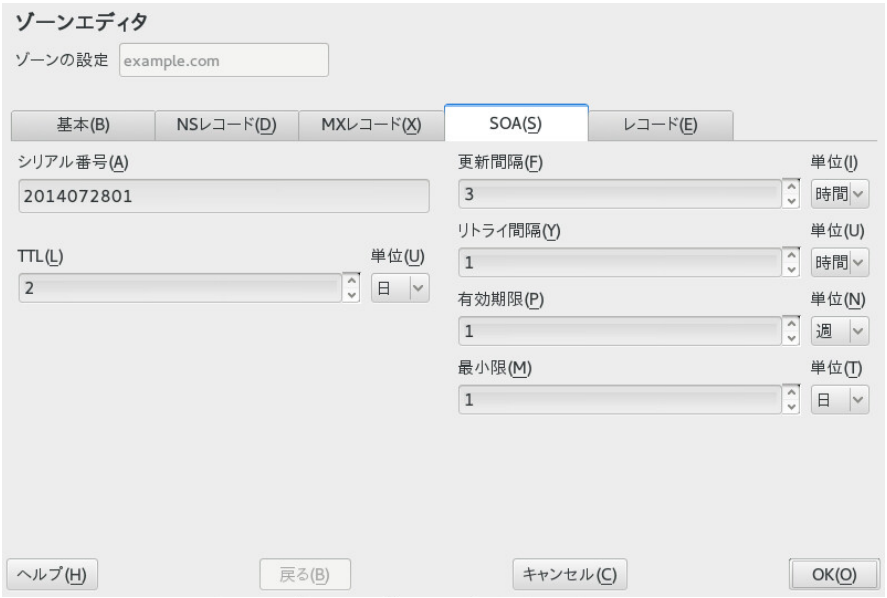


図 22.8 DNSサーバ:ゾーンエディタ(SOA)

ゾーンエディタ(レコード)

このダイアログでは、名前解決を管理します。[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]タイプは、メインエントリを表します。この値はIPアドレス(IPv4)でなければなりません。IPv6アドレスの場合は、[AAAA]を使用します。[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、次の例のように、`A`レコードとは反対です。

```
hostname.example.com. IN A 192.168.0.1
1.0.168.192.in-addr.arpa IN PTR hostname.example.com.
```

注記: 逆引きゾーンの編集

正引きゾーンの追加後、メインメニューに戻って、編集用の逆引きゾーンを選択します。次に、タブ[基本]で、チェックボックス[Automatically Generate Records From]にチェック印を入れ、正引きゾーンを選択します。これにより、正引きゾーンでのすべての変更が、逆引きゾーンで自動的に更新されます。

22.4 BINDネームサーバの起動

SUSE® Linux Enterprise Serverシステムでは、ネームサーバBIND (Berkeley Internet Name Domain)は、事前設定されて提供されるので、インストールが正常に完了すればただちに起動できます。すで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、すで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、そのゾーンを設定してはじめて、正しいDNSにすることができます。簡単な例については、`/usr/share/doc/packages/bind/config`のドキュメントを参照してください。

ヒント: 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`ファイル内の`NETCONFIG_DNS_POLICY`変数を`auto`に設定します。

ただし、公式のドメインは、その1つが責任のある機関によって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、`systemctl start named.service`コマンドを入力します。`systemctl status named.service`を使用して、namedが(ネームサーバプロセスが呼び出されたときに)正常に起動したかどうかを確認します。サーバが正常に起動したらすぐに、`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、「`host 127.0.0.1`」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、`systemctl status named.service`コマンドを使用して、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か、予期しない動作をする場合は、`journalctl -e`の出力を確認します。

プロバイダのネームサーバ(またはすでにネットワーク上で動作しているネームサーバ)をフォワーダとして使用する場合は、`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22.1「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。各自サイトの設定に合わせて変更してください。

例 22.1 NAMED.CONFファイルの転送オプション

```
options {
        directory "/var/lib/named";
        forwarders { 10.11.12.13; 10.11.12.14; };
        listen-on { 127.0.0.1; 192.168.1.116; };
        allow-query { 127/8; 192.168/16 };
        notify no;
};
```

options 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。環境設定ファイル /etc/named.conf またはゾーンファイルを変更したら、systemctl reload named.service を使用して、BINDにそれらを再ロードさせます。または、systemctl restart named.service を使用してネームサーバを停止してから再起動しても同じ結果が得られます。サーバは systemctl stop named.service を入力していつでも停止することができます。

22.5 The /etc/named.conf環境設定ファイル

BINDネームサーバ自体のすべての設定は、/etc/named.conf ファイルに格納されます。ただし、処理するドメインのゾーンデータ(ホスト名、IPアドレスなどで構成されている)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22.2「基本的な/etc/named.confファイル」に示します。

例 22.2 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";
};
```

22.5.1 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 192.168.1.116 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 192.168.2.0/24 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システムのジャーナルに保存します。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、システムのジャーナル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;

BIND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。

すべての利用可能なオプションのリストについては、マニュアルページ man 5 named.conf を参照してください。

22.5.2 ロギング

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22.3「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。

例 22.3 ログを無効にするエントリ

```
logging {
        category default { null; };
};
```

22.5.3 ゾーンエントリ

例 22.4 EXAMPLE.COMのゾーンエントリ

```
zone "example.com" in {
      type master;
      file "example.com.zone";
      notify no;
};
```

`zone`の後、管理対象のドメイン名(`example.com`)を指定し、その後に`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22.4「example.com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22.5「example.net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。

例 22.5 EXAMPLE.NETのゾーンエントリ

```
zone "example.net" in {
      type slave;
      file "slave/example.net.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、ゾーン`.`(`hint`タイプ)を使用します。このゾーン定義はそのまま使用できます。

`example.com.zone` ファイルまたは「slave/example.net.zone」ファイル

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。スレーブの場合は、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、外部の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。! * によってそのような操作が禁止されるため、前述のエントリは同じものをアーカイブします。

22.6 ゾーンファイル

ゾーンファイルは2種類必要です。一方はIPアドレスをホスト名に割り当て、もう一方は逆にIPアドレスのホスト名を提供します。

ヒント: ゾーンファイルでのドット(ピリオド、フルストップ)の使用

フィルタフィールドの右側にある `"."` はゾーンファイル内で重要な意味を持ちます。ホスト名の末尾にドット(`.`)がない場合は、ゾーンが追加されます。フルドメイン名が付いた完全なホスト名には、末尾にドット(`.`)を付けて、ドメインが再度追加されるないようにします。ネームサーバ設定エラーの原因として最も頻繁に挙げられるのは、おそらく「.」の打ち忘れや位置の間違いです。

最初に、ドメイン `example.com` に責任を負うゾーンファイル `example.com.zone` について示します(例22.6「/var/lib/named/example.com.zoneファイル」を参照してください)。

例 22.6 /VAR/LIB/NAMED/EXAMPLE.COM.ZONEファイル

```
1.  $TTL 2D
2.  example.com. IN SOA      dns  root.example.com. (
3.                2003072441  ; serial
4.                1D          ; refresh
```

```
5.              2H          ; retry
6.              1W          ; expiry
7.              2D )        ; minimum
8.
9.              IN NS       dns
10.             IN MX       10 mail
11.
12. gate        IN A        192.168.5.1
13.             IN A        10.0.0.1
14. dns         IN A        192.168.1.116
15. mail        IN A        192.168.3.108
16. jupiter     IN A        192.168.2.100
17. venus       IN A        192.168.2.101
18. saturn      IN A        192.168.2.102
19. mercury     IN A        192.168.2.103
20. ntp         IN CNAME    dns
21. dns6        IN A6  0    2002:c0a8:174::
```

1行目:

`$TTL`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、先頭の`example.com`です。この末尾には、「`.`」(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。この名前は、末尾に「`.`」が付いていないので、`dns`から`dns.example.com`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。`@`記号はすで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@example.com`の場合、エントリは`root.example.com`を読み込む必要があります。フィルタフィールドの右側にある`"."`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。ここでは、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`では、このドメインを担当するネームサーバを指定します。`dns`は、`dns.example.com`に拡張されます。これは、末尾に「`.`」が付いていないためです。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`example.com`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`mail.example.com`が指定されています。ホスト名の前の数字は、初期設定値です。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。

12〜19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。ここでは、名前が「`.`」なしで一覧にされています。これは、これらの名前にはドメインが含まれていないためです。したがって、これらの名前にはすべて、`example.com`が追加されます。ホス

トgateには、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AAAA が付きます。

注記: IPv6の構文

IPv6レコードの構文は、IPv4と少し異なっています。断片化の可能性があるため、アドレスの前に消失したビットに関する情報を入力する必要があります。IPv6アドレスを必要な数の「0」で埋めるには、アドレス内の正しい位置に2つコロンを追加します。

```
pluto     AAAA 2345:00C1:CA11::1234:5678:9ABC:DEF0
pluto     AAAA 2345:00D2:DA11::1234:5678:9ABC:DEF0
```

20行目:

エイリアスntpをdnsの別名として使用できます(CNAMEは一般名という意味)。

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は、168.192.in-addr.arpaに解決されます。参照先 例22.7「逆引き」。

例 22.7 逆引き

```
1.  $TTL 2D
2.  168.192.in-addr.arpa.   IN SOA dns.example.com. root.example.com. (
3.                          2003072441      ; serial
4.                          1D              ; refresh
5.                          2H              ; retry
6.                          1W              ; expiry
7.                          2D )            ; minimum
8.
9.                          IN NS           dns.example.com.
10.
11. 1.5                     IN PTR          gate.example.com.
12. 100.3                   IN PTR          www.example.com.
13. 253.2                   IN PTR          cups.example.com.
```

1行目:

$TTL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` の逆引きを有効にします。ゾーン名は `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「`.`」が付いて指定されます。残りのエントリは、前出の `example.com` の例の記述と同じです。

3〜7行目:

前出の例の `example.com` 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アドレスの最後の部分のみが、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、ゾーン転送は、異なるバージョンのBIND間でも問題なく行えるはずです。

22.7 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(22.8項 「安全なトランザクション」参照)。

22.8 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、`man dnssec-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`/etc/named.conf`ファイルに含める必要があります。

```
key host1-host2 {
 algorithm hmac-md5;
 secret "ejIkuCyyGJwwuN3xAteKgg==";
};
```

警告: /etc/named.confのファイルパーミッション

/etc/named.conf**の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。外部ファイルをインクルードするには、次のようにします。

```
include  "filename"
```

ここで、`filename`には、キーを持つファイルへの絶対パスを指定します。

サーバ`host1`が`host2`(この例では、アドレス`10.1.2.3`)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 10.1.2.3 {
  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。

22.9 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。

`dnssec-signzone`コマンドを使用すると、生成されたキーのセット(`keyset-`ファイル)を作成し、それらを安全な方法で親ゾーンに転送し、署名することができます。これによって、`/etc/named.conf`内のゾーンごとにインクルードするファイルが生成されます。

22.10 その他の情報

ここで扱ったトピックの詳細については、/usr/share/doc/packages/bind/armディレクトリにインストールされる`bind-doc`パッケージ内の『`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 Enterprise ServerのBINDに関する最新情報が含まれています。

23 DHCP

DHCP(Dynamic Host Configuration Protocol)の目的は、ネットワーク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。クライアント側でNetworkManagerを使用する場合は、クライアントを設定する必要はありません。これは、環境を変更し、一度に1つのインタフェースしかない場合に便利です。DHCPサーバが実行しているマシン上ではNetworkManagerを使用しないでください。

ヒント: IBM System z: DHCPのサポート

IBM System zプラットフォーム上では、OSAおよびOSA Expressネットワークカードを使用しているインタフェースに対してのみDHCPを使用できます。DHCPの自動環境設定機能に必要なMACアドレスを持つのは、これらのカードだけです。

DHCPサーバの設定方法の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はサーバが用意したアドレスプールから、アドレスを各関連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。これは、ネットワークにアドレス以上のクライアントが存在しない場合にのみ機能します。

DHCPは、システム管理者の負担を軽減します。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てはるかに簡単です。また、特に新しいコンピュータ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。

この章では、`192.168.2.1`をゲートウェイとし、DHCPサーバをワークステーション`192.168.2.0/24`と同じサブネットで実行します。このサーバは、固定IPアドレス`192.168.2.254`を持ち、2つのアドレス範囲(`192.168.2.10`～`192.168.2.20`および`192.168.2.100`～`192.168.2.200`)を操作対象とします。

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。

23.1 YaSTによるDHCPサーバの設定

DHCPサーバをインストールするには、YaSTを起動して、[ソフトウェア] › [ソフトウェア管理]の順に選択します。[フィルタ] › [パターン]の順に選択してから、[DHCPおよびDNSサーバ]を選択します。依存関係のあるパッケージのインストールを確認して、インストールプロセスを完了します。

重要: LDAPのサポート

YaST DHCPモジュールは、サーバ設定をローカルに(DHCPサーバを実行するホスト上に)保存するか、その設定データをLDAPサーバに管理させるように、セットアップできます。LDAPを使用するには、LDAP環境を設定してからDHCPサーバを設定してください。

LDAPの詳細については、Book “Security Guide” 5 “LDAP—A Directory Service”を参照してください。

YaST

DHCPモジュール(`yast2-dhcp-server`)を使用すると、ローカルネットワーク用に独自のDHCPサーバをセットアップできます。このモジュールは、ウィザードモードまたはエキスパート設定モードで実行できます。

23.1.1 初期設定(ウィザード)

このモジュールを初めて起動すると、ウィザードが開始して、サーバ管理に関していくつか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は、さらに高度な設定タスクを行う場合に使用できます。次の手順に従います。

1. そのリストから、DHCPサーバがリスンするインタフェースを選択し、[選択]をクリックします。この後、[選択したインタフェースのファイアウォールを開く]を選択して、このインタフェース用のファイアウォールを開き、[次へ]をクリックします。詳細については、図23.1「DHCPサーバ:カードの選択」を参照してください。

図 23.1 DHCPサーバ:カードの選択

2. チェックボックスを使って、LDAPサーバがDHCP設定を自動的に格納する必要があるかどうかを指定します。テキストボックスに、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。詳細については、図23.2「DHCPサーバ:グローバル設定」を参照してください。

図 23.2 DHCPサーバ:グローバル設定

3. 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。詳細については、図23.3「DHCPサーバ:ダイナミックDHCP」を参照してください。

図 23.3 DHCPサーバ:ダイナミックDHCP

4. DHCPサーバ開始方法を定義します。システムのブート時にDHCPサーバを自動的に起動するか、必要に応じて(たとえば、テスト目的で)手動で起動するか指定します。[完了]をクリックして、サーバの環境設定を完了します。詳細については、図23.4「DHCPサーバ:起動」を参照してください。

図 23.4 DHCPサーバ:起動

5. 前のステップで説明した方法で動的DHCPを使用するかわりに、アドレスを疑似静的方式で割り当てるようにサーバを設定することもできます。下部のテキストボックスを使用して、この方法で管理するホストのリストを指定します。具体的には、[[名前]と[[IPアドレス]に、この種のクライアントに与える名前とIPアドレスを指定し、さらに[[ハードウェアアドレス]と[[ネットワークタイプ](トークンリングまたはイーサネット)を指定します。上部に表示されるクライアントリストを修正するには、[[追加]、[[編集]、および[[削除]を使用します。詳細については、図 23.5「DHCPサーバ:ホスト管理」を参照してください。

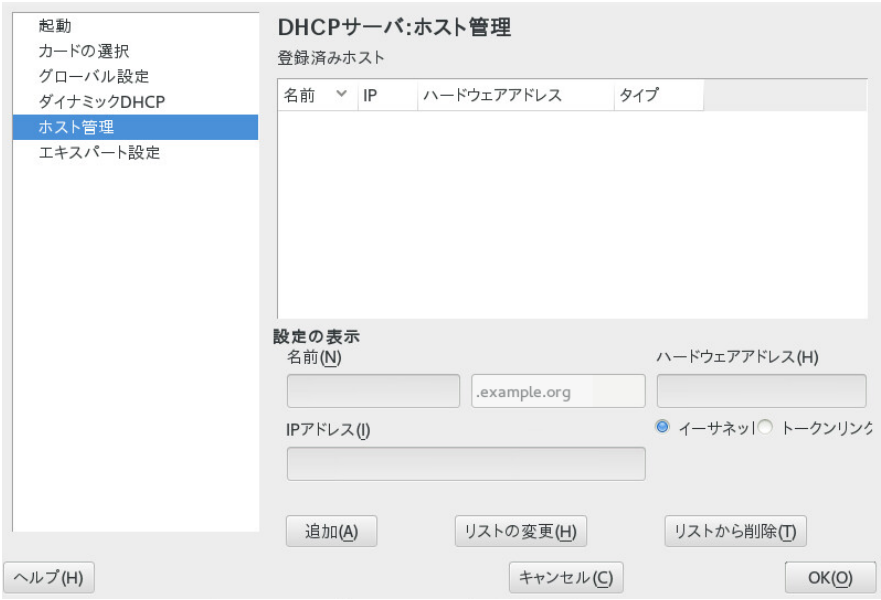


図 23.5 DHCPサーバ:ホスト管理

23.1.2 DHCPサーバ設定(エキスパート)

前述の環境設定方法に加えて、DHCPサーバのセットアップを詳細に変更できるようにエキスパート設定モードが用意されています。エキスパート環境設定を開始するには、[スタートアップ]ダイアログの[DHCPサーバエキスパート環境設定]をクリックします(図23.4「DHCPサーバ:起動」を参照)。

chroot環境と宣言

この最初のダイアログで[DHCPサーバの起動]を選択し、既存の環境設定を編集可能にします。DHCPサーバの動作のうち、重要なのはchroot環境またはchroot jailで動作してサーバホストを保護する機能です。DHCPサーバが外部からの攻撃にさらされるとしても、攻撃者はchroot jailの中にとどまるためシステムの残りの部分には進入できません。ダイアログの下部には、定義済みの宣言を示すツリービューが表示されます。これらの修正には、[追加]、[削除]、および[編集]を使用します。[詳細]を選択すると、上級者用のダイアログが追加表示されます。図23.6「DHCPサーバ:Chroot Jailと宣言」を参照してください。[追加]を選択後、追加する宣言の種類を定義します。[詳細]から、サーバのログファイルの表示、TSIGキー管理の設定、およびDHCPサーバのセットアップに応じたファイアウォール設定の調整を行うことができます。

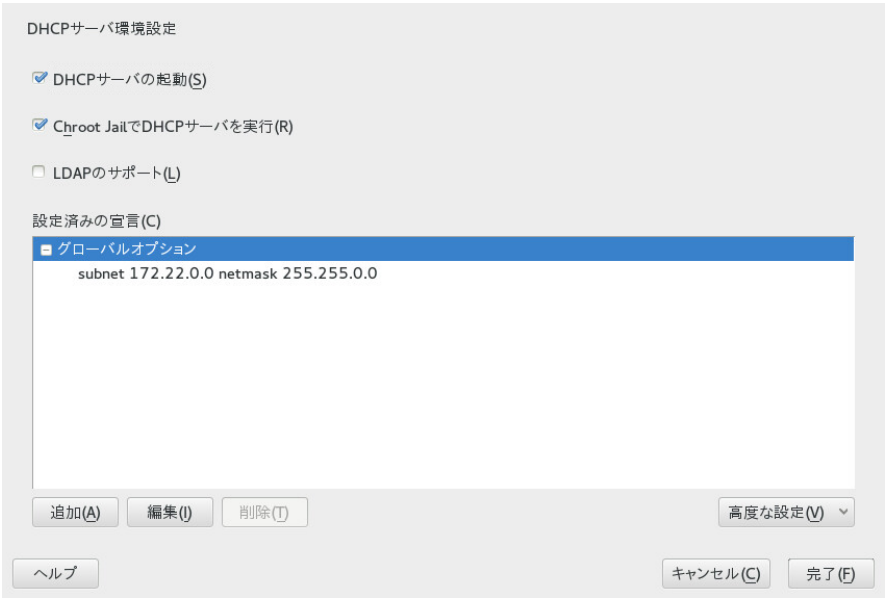


図 23.6 DHCPサーバ:CHROOT JAILと宣言

宣言タイプの選択

DHCPサーバの[グローバルオプション]は、多数の宣言で構成されています。このダイアログでは、宣言タイプ[サブネット]、[ホスト]、[共有ネットワーク]、[グループ]、[アドレスプール]、および[クラス]を設定できます。この例は、新しいサブネットの選択を示しています(図23.7「DHCPサーバ:宣言タイプの選択」を参照)。

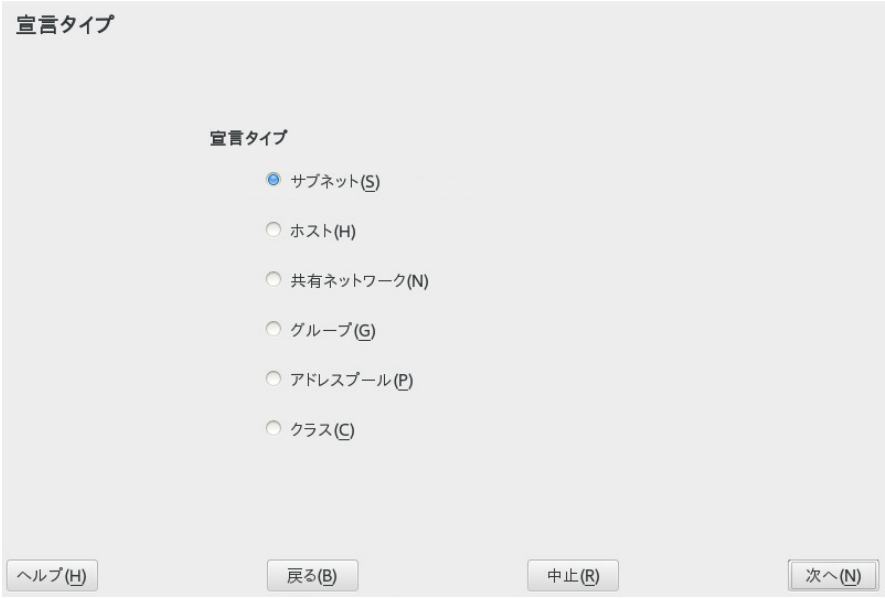


図 23.7 DHCPサーバ:宣言タイプの選択

サブネットの設定

このダイアログでは、IPアドレスとネットマスクを使用して新しいサブネットを指定できます。ダイアログの中央部分で[追加]、[編集]、および[削除]を使用して、選択したサブネットのDHCPサーバ起動オプションを変更します。サブネットのダイナミックDNSを設定するには、[ダイナミックDNS]を選択します。

図 23.8 DHCPサーバ:サブネットの設定

TSIGキー管理

前のダイアログでダイナミックDNSを設定するように選択した場合は、セキュアゾーン転送用のキー管理を設定できます。[OK]を選択すると別のダイアログが表示され、ダイナミックDNSのインタフェースを設定できます(図23.10「DHCPサーバ:ダイナミックDNS用のインタフェースの設定」を参照)。

図 23.9 DHCPサーバ:TSIGの設定

ダイナミックDNS:インタフェースの設定

ここでは、[このサブネットでダイナミックDNSを有効にする]を選択して、サブネットのダイナミックDNSを有効化できます。その後、ドロップダウンリストを使用して正引きゾーンと逆引きゾーン両方のTSIGキーを選択し、そのキーがDNSとDHCPサーバに共通であることを確認します。[グローバルダイナミックDNS設定の更新]を使用すると、ダイナミックDNS環境に従ってグローバルDHCPサーバ設定を自動的に更新および調整できます。最後に、ダイナミックDNSに従って更新する正引きゾーンと逆引きゾーンについて、プライマリネームサーバの名前を個別に指定し、この2つのゾーンを定義します。[OK]を選択すると、サブネットの設定ダイアログに戻ります(図23.8「DHCPサーバ:サブネットの設定」を参照)。[[OK]を選択すると、エキスパート設定ダイアログに戻ります

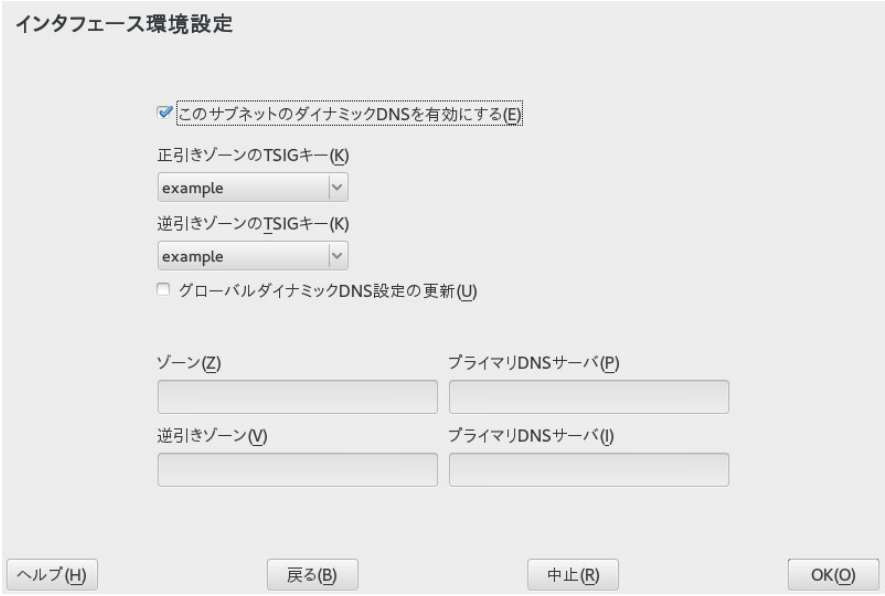


図 23.10 DHCPサーバ:ダイナミックDNS用のインタフェースの設定

ネットワークインタフェースの環境設定

DHCPサーバがリスンするインタフェースを定義し、ファイアウォール設定を調整するには、[エキスパート環境設定]ダイアログで[詳細] › [インタフェースの設定]の順に選択します。表示されるインタフェースリストから、DHCPサーバがリスンするインタフェースを1つ以上選択します。すべてのサブネット内のクライアントがサーバと通信できるようにする必要があり、サーバホストでもファイアウォールを実行する場合は、ファイアウォールを適宜調整してください。調整するには、[Adapt Firewall Settings (ファイアウォール設定の調整)]を選択します。設定を完了した後、[OK]をクリックして元のダイアログに戻ると、YaSTがSuSEfirewall2のルールを新しい条件に合わせて調整します(図23.11「DHCPサーバ:ネットワークインタフェースとファイアウォール」を参照)。

図 23.11 DHCPサーバ:ネットワークインタフェースとファイアウォール

設定ステップをすべて完了した後、[OK]を選択してダイアログを閉じます。これでサーバは新規環境設定に従って起動します。

23.2 DHCPソフトウェアパッケージ

SUSE Linux Enterprise Serverでは、DHCPサーバとDHCPクライアントのどちらも利用できます。使用可能なDHCPサーバは、Internet Systems Consortiumによって公開された`dhcpd`です。クライアント側には、`dhcp-client` (同じくISCが公開)および`wicked`パッケージに付属のツールがあります。

デフォルトでは、`wicked`ツールとともに、`wickedd-dhcp4.service`と`wickedd-dhcp6.service`がインストールされます。どちらもシステムをブートするたびに自動的に起動され、DHCPサーバを検出します。環境設定ファイルは必要ありません。標準的なセットアップであれば

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、環境設定ファイル `/etc/dhclient.conf` および `/etc/dhclient6.conf` によって制御されるISC `dhcp-client` を使用します。

23.3 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23.1「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。

例 23.1 環境設定ファイル/ETC/DHCPD.CONF

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "example.com";
option domain-name-servers 192.168.1.116;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
option subnet-mask 255.255.255.0;

subnet 192.168.2.0 netmask 255.255.255.0
 {
  range 192.168.2.10 192.168.2.20;
  range 192.168.2.100 192.168.2.200;
 }
```

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。

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サーバがコンピュータにIPアドレスを割り当てた場合に、コンピュータ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サーバを最大3つ指定します。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。また、ネームサーバでは、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、第22章　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、要求しているクライアントで使用され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)が指定されます。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。例23.1「環境設定ファイル/etc/dhcpd.conf」では、クライアントは `192.168.2.10` ～ `192.168.2.20` および `192.168.2.100` ～ `192.168.2.200` 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、`systemctl start dhcpd.service` 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。`rcdhcpd check-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、`journalctl` コマンドで問い合わせることができるメインシステムログで情報を探せば、原因が突き止められます(第11章 `journalctl`:systemdジャーナルのクエリを参照してください)。

デフォルトのSUSE Linux Enterprise Serverシステムでは、セキュリティ上の理由から、chroot環境で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。通常は、`systemctl start dhcpd.service` 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、手動でコピーする必要はありません。

23.3.1 固定IPアドレスを持つクライアント

DHCPは、事前定義の静的アドレスを特定のクライアントに割り当てる場合にも使用できます。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。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クライアン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、`00:30:6E:08:EC:80`)。たとえば、例23.2「環境設定ファイルへの追加」のような数行を例23.1「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、対応するホストに同じデータのセットを割り当てます。

例 23.2 環境設定ファイルへの追加

```
host jupiter {
hardware ethernet 00:30:6E:08:EC:80;
fixed-address 192.168.2.100;
}
```

クライアントの名前を1行目に(`host hostname` (ここでは`jupiter`に置き換わる))、MACアドレスを2行目に入力します。LinuxホストでMACアドレスを確認するには、`ip link show`コマンドの後にネットワークデバイス(たとえば、`eth0`)を指定して実行します。出力例を次に示します。

```
link/ether 00:30:6E:08:EC:80
```

上の例では、MACアドレス`00:30:6E:08:EC:80`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`192.168.2.100`とホスト名`jupiter`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。

23.3.2 SUSE Linux Enterprise Serverのバージョン

セキュリティ向上のため、SUSE Linux Enterprise Server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、 `etc/sysconfig/dhcpd` ファイルの `DHCPD_CONF_INCLUDE_FILES` 変数で、これらのファイルを設定します。syslogデーモンの再起動後もDHCPログが継続して動作するようにするには、 `/etc/sysconfig/syslog` ファイル内の `SYSLOGD_ADDITIONAL_SOCKET_DHCP` エントリを指定します。

23.4 その他の情報

DHCPの詳細については、Internet Systems 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` のマニュアルページにも詳細が記載されています。

24 NetworkManagerの使用

NetworkManagerは、ラップトップなどの携帯用コンピュータのための理想的なソリューションです。NetworkManagerは、802.1x保護ネットワークへの接続など、ネットワーク接続のための最新の暗号化タイプおよび標準をサポートしています。802.1Xは、「IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control」(ポートごとにネットワークアクセスの制御を行う、ローカル/メトロポリタンエリアネットワーク向け IEEE 標準)です。「」NetworkManagerを使用すると、ネットワークインタフェースの設定および移動時の有線/ワイヤレスネットワーク間の切り替えについて心配する必要がなくなります。NetworkManagerでは、既知のワイヤレスネットワークに自動的に接続するか、または複数のネットワーク接続を並行して管理できます。後者の場合、最も高速な接続がデフォルトとして使用されます。さらに、利用可能なネットワーク間を手動で切り換えたり、システムトレイのアプレットを使用してネットワーク接続を管理できます。

単一の接続をアクティブにする代わりに、複数の接続を一度にアクティブにできます。これにより、Ethernetからラップトップの接続プラグを抜いても、無線接続により接続が維持されます。

24.1 NetworkManagerの使用

NetworkManagerは、高度で直感的なユーザインタフェースを提供します。このインタフェースを使用すると、ネットワーク環境を簡単に切り換えることができます。ただし、NetworkManagerは、次の場合には適しません。

- コンピュータが、DHCPまたはDNSサーバなど、ネットワーク内で他のコンピュータにネットワークサービスを提供している場合。
- コンピュータがXenサーバの場合、またはシステムがXen内の仮想システムの場合。

24.2 NetworkManagerの有効化/無効化

ラップトップコンピュータでは、NetworkManagerがデフォルトで有効です。ただし、YaSTネットワーク設定モジュールでいつでも有効または無効にできます。

1. YaSTを実行し、[ネットワークデバイス] › [ネットワーク設定]の順に選択します。
2. [Network Settings]ダイアログが開きます。[グローバルオプション]タブを開きます。
3. NetworkManagerを使用してネットワーク接続を設定および管理する

a. [ネットワークのセットアップ方法]フィールドで、[NetworkManagerを使ってユーザが制御]を選択します。

b. [OK]をクリックしてYaSTを閉じます。

c. 24.3項 「ネットワーク接続の設定」に従って、NetworkManagerを使用してネットワーク接続を設定します。

4. NetworkManagerを無効にし、ネットワークをユーザ自身の設定で制御する

a. [ネットワークのセットアップ方法]フィールドで、[Controlled by wicked (wickedによる制御)]を選択します。

b. [OK]をクリックします。

c. DHCP経由の自動環境設定または静的IPアドレスによる手動設定で、YaSTでネットワークカードを設定します。
YaSTを使用したネットワーク設定の詳細については、19.4項 「YaSTによるネットワーク接続の設定」を参照してください。

24.3 ネットワーク接続の設定

YaSTでNetworkManagerを有効にした後、GNOMEで使用可能なNetworkManagerフロントエンドでネットワーク接続を設定します。有線、ワイヤレス、モバイルブロードバンド、DSL、VPN接続など、あらゆるタイプのネットワーク接続に対応するタブが表示されます。

GNOMEでネットワーク設定ダイアログを開くには、[Status (状態)]メニューから[設定]メニューを開き、[ネットワーク]エントリをクリックします。

注記: オプションの利用可否

システムセットアップによっては、接続を設定できない場合があります。保護された環境では、一部のオプションがロックされているか、または `root` パーミッションを必要とする場合があります。詳細については、システム管理者にお問い合わせください。

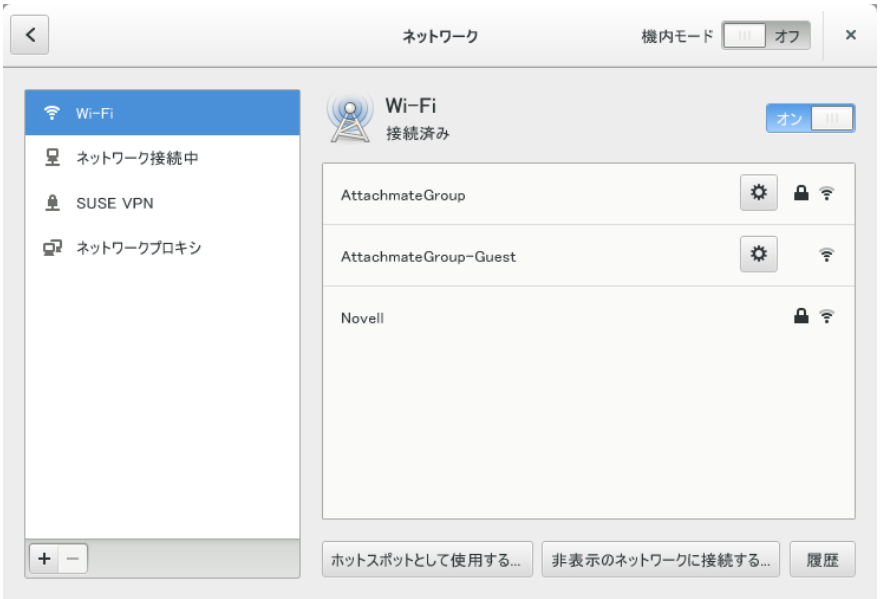


図 24.1 GNOMEネットワーク接続のダイアログ

手順 24.1 接続の追加と編集

1. NetworkManagerの設定ダイアログを開きます。
2. 接続を追加する
    a. 左下隅の[+]アイコンをクリックします。
    b. 目的の接続タイプを選択して、画面の指示に従います。
    c. 終了したら、[追加]をクリックします。
    d. 変更を確定した後、[Status (状態)]メニューを開くと、新たに設定されたネットワーク接続が使用可能なネットワークのリストに表示されます。
3. 接続を編集する
    a. 編集するエントリを選択します。
    b. 歯車アイコンをクリックして[Connection Settings (接続の設定)]ダイアログを開きます。
    c. 変更を行ったら、[適用]をクリックして変更を保存します。

d. 使用している接続をシステム接続として利用できるようにするには、[識別情報]タブを開き、[Make available to other users (他のユーザが利用できるようにする)]チェックボックスをオンにします。ユーザ接続とシステム接続の詳細については、24.4.1項「ユーザおよびシステムの接続」を参照してください。

24.3.1 有線ネットワーク接続の管理

コンピュータが有線ネットワークに接続している場合、NetworkManagerアプレットを使用して接続を管理します。

1. 接続の詳細を変更したり、接続をオフにしたりするには、[Status (状態)]メニューを開き、[Wired (有線)]をクリックします。
2. 設定を変更するには、[Wired Settings (有線の設定)]をクリックし、歯車アイコンをクリックします。
3. すべてのネットワーク接続をオフにするには、[Airplane Mode (機内モード)]設定を有効にします。

24.3.2 ワイヤレスネットワーク接続の管理

可視のワイヤレスネットワークは、[Wireless Networks (ワイヤレスネットワーク)]の下のGNOME NetworkManagerアプレットメニューに一覧にされます。各ネットワークの信号強度もメニューに表示されます。暗号化されたワイヤレスネットワークには、シールドアイコンが付きます。

手順 24.2 可視のワイヤレスネットワークへの接続

1. 可視のワイヤレスネットワークに接続するには、[Status (状態)]メニューを開いて[Wi-Fi]をクリックします。
2. [Turn On (オンにする)]をクリックして、ネットワークを有効にします。
3. [Select Network (ネットワークの選択)]をクリックしてWi-Fiネットワークを選択し、[接続]をクリックします。
4. ネットワークが暗号化されている場合は、環境設定ダイアログが開きます。このダイアログには、ネットワークで使用されている暗号化のタイプと、ログインアカウント情報を入力するためのテキストボックスが表示されます。

手順 24.3 不可視のワイヤレスネットワークへの接続

1. サービスセット識別子(SSIDまたはESSID)をブロードキャストしないため自動的に検出されないネットワークに接続するには、[Status (状態)]メニューを開き、[Wi-Fi]をクリックします。
2. [Wi-Fi Settings (Wi-Fi設定)]をクリックして[Detailed Settings (詳細設定)]メニューを開きます。
3. 使用するWi-Fiが有効になっていることを確認し、[Connect to Hidden Network (非公開のネットワークに接続)]をクリックします。
4. 表示されるダイアログの[ネットワーク名]に、SSIDまたはESSIDを入力し、必要に応じて暗号化パラメータを設定します。

明示的に選択された無線ネットワークは、可能な限り接続が維持されます。その時点でネットワークケーブルが接続されていれば、無線接続の稼働中に、[Stay connected when possible]に設定したすべての接続が確立されます。

24.3.3 Wi-Fi/Bluetoothカードのアクセスポイントとしての設定

お使いのWi-Fi/Bluetoothカードでアクセスポイントモードがサポートされている場合、NetworkManagerを使用して設定できます。

1. [Status (状態)]メニューを開き、[Wi-Fi]をクリックします。
2. [Wi-Fi Settings (Wi-Fi設定)]をクリックして[Detailed Settings (詳細設定)]メニューを開きます。
3. [Use as Hotspot (ホットスポットとして使用)]をクリックして、画面の指示に従います。
4. 結果のダイアログに表示される資格情報を使用して、リモートマシンからホットスポットに接続します。

24.3.4 NetworkManagerとVPN

NetworkManagerは、数種類のVPN (仮想私設網)技術をサポートしています.各技術について、SUSE Linux Enterprise ServerにはNetworkManagerの一般的なサポートを提供する基本パッケージが付属しています。加えて、アプレットに対応するデスクトップ固有のパッケージをインストールすることも必要です。

OpenVPN

このVPN技術を使用するには、次のアイテムをインストールします:

- `NetworkManager-openvpn` および
- `NetworkManager-openvpn-gnome`

vpnc (Cisco)

このVPN技術を使用するには、次のアイテムをインストールします:

- `NetworkManager-vpnc` および
- `NetworkManager-vpnc-gnome`

PPTP(ポイントツーポイントトンネリングプロトコル)

このVPN技術を使用するには、次のアイテムをインストールします:

- `NetworkManager-pptp` および
- `NetworkManager-pptp-gnome`

パッケージのインストールを完了したら、VPN接続を設定します(手順24.1「接続の追加と編集」を参照してください)。

24.4 NetworkManagerとセキュリティ

NetworkManagerは、ワイヤレス接続を「信頼された」と「信頼なし」という2種類で区別します。 「信頼された」接続とは、過去に明示的に選択したネットワークです。 その他は「信頼なし」です。信頼された接続は、アクセスポイントのMACアドレスと名前で識別されます。MACアドレスを使用して、信頼された接続が同じ名前でも、異なるアクセスポイントを使用できないようにすることができます。

NetworkManagerにより、定期的に、使用可能なネットワークがスキャンされます。信頼されたネットワークが複数検出された場合、最近使用されたものが自動的に選択されます。すべてのネットワークが信頼されないネットワークの場合は、NetworkManagerはユーザがネットワークを選択するまで待機します。

暗号化設定が変更されても、名前とMACアドレスが同じままの場合は、NetworkManagerは接続を試みますが、まず、新しい暗号化設定の確認とアップデート(新しいキーなど)の提供を求めるプロンプトが表示されます。

無線接続を使用している状態からオフラインモードに切り替えると、NetworkManagerでSSIDまたはESSIDが空白になります。これにより、カードの接続解除が確保されます。

24.4.1 ユーザおよびシステムの接続

NetworkManagerは、**ユーザ**および**システム**という2種類の接続を認識します。ユーザ接続は、最初のユーザがログインしたとき、NetworkManagerで利用可能になる接続です。ユーザは、必要な資格情報を要求されます。ユーザがログアウトすると、接続は切断され、NetworkManagerから削除されます。システム接続として定義された接続は、すべてのユーザが共有でき、NetworkManagerの起動直後で、どのユーザもまだログインしていないとき、利用可能になります。システム接続の場合、すべての資格情報を接続作成時に提供する必要があります。そのようなシステム接続は、認証を要求するネットワークへの自動接続に使用することができます。NetworkManagerでユーザ接続またはシステム接続を設定する方法については、24.3項「ネットワーク接続の設定」を参照してください。

24.4.2 パスワードと資格情報の保存

暗号化ネットワークに接続するたびに資格情報を再入力しないようにするには、GNOMEキーリングマネージャを使用して資格情報を暗号化し、マスタパスワードを使用して安全にディスク上に保存できます。

NetworkManagerは、安全な接続(暗号化された有線、ワイヤレス、またはVPNの接続など)のための証明書を証明書ストアから取得することもできます。詳細については、Book “Security Guide” 12 “Certificate Store”を参照してください。

24.5 よくある質問とその回答

NetworkManagerによる特別なネットワークオプションの設定に関するFAQ (よくある質問と答え)は、次のとおりです。

24.5.1 特定のデバイスには、どのようにして接続しますか?

デフォルトでは、NetworkManager内の接続は、デバイスタイプ固有の接続であり、同じタイプのすべての物理デバイスに適用されます。1つの接続タイプについて複数の物理デバイスが使用可能である場合(たとえば、マシンに2枚のEthernetカードが取り付けられている場合)、特定のデバイスに接続を関連付けることができます。

GNOMEでこれを行うには、まずデバイスのMACアドレスを調べます。このために、アプレットから利用できる[接続情報]か、またはコマンドラインツール(`nm-tool`や`wicked show all`など)の出力を使用します。次に、ネットワーク接続を設定するためのダイアログを起動し、変更する接続を選択します。[有線]タブまたは[無線]タブで、デバイスの[MACアドレス]を入力し、変更を確定します。

24.5.2 同じESSIDを持つ複数のアクセスポイントが検出された場合、どのようにして特定のアクセスポイントを指定しますか?

異なる無線帯域(a/b/g/n)を持つ複数のアクセスポイントが利用可能な場合、デフォルトでは、最も強い信号を持つアクセスポイントが自動的に選択されます。このデフォルトを無効にするには、ワイヤレス接続の設定時に[BSSID]フィールドを使用します。

BBSID (Basic Service Set Identifier)は、各Basic Service Setを固有に識別します。インフラストラクチャBasic Service Setでは、BSSIDは、ワイヤレスアクセスポイントのMACアドレスです。独立型(アドホック)Basic Service Setでは、BSSIDは、46ビットの乱数から生成されローカルに管理されるMACアドレスです。

24.3項「ネットワーク接続の設定」に説明されているように、ネットワーク接続を設定するダイアログを開始します。変更したいワイヤレス接続を選択し、[編集]をクリックします。[ワイヤレス]タブで、BSSIDを入力します。

24.5.3 どのようにして、ネットワーク接続を他のコンピュータと共用しますか?

プライマリデバイス(インターネットに接続するデバイス)には、特別な設定は必要ありません。ただし、ローカルハブまたはローカルコンピュータに接続するデバイスは、次の手順で設定する必要があります。

1. 24.3項 「ネットワーク接続の設定」に説明されているように、ネットワーク接続を設定するダイアログを開始します。変更したい接続を選択し、[編集]をクリックします。[IPv4設定]タブに切り替えて、[方法]ドロップダウンボックスから[他のコンピュータと共有]を選択します。これで、IPトラフィックの転送が有効になり、デバイス上でDHCPサーバが実行されます。NetworkManagerで変更内容を確認します。

2. DCHPサーバは、ポート67を使用するので、そのポートがファイアウォールによってブロックされていないことを確認してください。そのためには、接続を共有するマシンで、YaSTを起動して、[セキュリティとユーザ] › [ファイアウォール]の順に選択します。[許可されるサービス]カテゴリに切り替えます。[DCHP Server]が[許可されるサービス]として表示されていない場合は、[Services to Allow]から[DCHP Server]を選択し、[追加]をクリックします。YaSTで変更内容を確認します。

24.5.4 静的DNSアドレスに、どのようにして自動(DHCP, PPP, VPN)アドレスを提供しますか?

DHCPサーバが無効なDNS情報(および/またはルート)を提供する場合は、次の手順でそれを無効にできます。24.3項 「ネットワーク接続の設定」に説明されているように、ネットワーク接続を設定するダイアログを開始します。変更したい接続を選択し、[編集]をクリックします。[IPv4設定]タブに切り替えて、[方法]ドロップダウンボックスから[自動(DHCP)アドレスのみ]を選択します。[DNS Servers (DNSサーバ)]および[Search Domains (検索ドメイン)]フィール

ドにDNS情報を入力します。自動的に取得されたルートを無視するには、[自動的に取得されたルートを無視する]で[ルート]をクリックし、各チェックボックスをオンにします。変更内容を確認します。

24.5.5 どのようにしたら、ユーザがログインする前に、パスワード保護されたネットワークにNetworkManagerを接続できますか?

そのような目的に使用できる`system connection`を定義します。詳細については、24.4.1項「ユーザおよびシステムの接続」を参照してください。

24.6 トラブルシューティング

場合によっては、接続に関する問題が発生することがあります。NetworkManagerに関してよく発生する問題としては、アプレットが起動しない、VPNオプションがないなどがあります。これらの問題の解決、防止方法は、使用ツールによって異なります。

NetworkManagerデスクトップアプレットが起動しない

ネットワークがNetworkManager制御に設定されている場合、アプレットは自動的に起動します。アプレットが起動しない場合は、24.2項 「NetworkManagerの有効化/無効化」の説明に従ってYaST内でNetworkManagerが有効になっているかどうかを確認します。その後、NetworkManager-gnomeパッケージもインストールされていることを確認します。デスクトップアプレットがインストールされているのに何らかの理由で実行されていない場合は、手動でアプレットを起動してください。デスクトップアプレットがインストールされているのに何らかの理由で実行されていないときは、コマンド`nm-applet`で手動で起動します。

NetworkManagerアプレットにVPNオプションが表示されない

NetworkManager、アプレット、およびNetworkManager用VPNのサポートは、個別のパッケージで配布されます。NetworkManagerアプレットにVPNオプションが表示されない場合は、使用しているVPNテクノロジのNetworkManagerサポートが含まれたパッケージがインストールされているかどうかを確認します。詳細については、24.3.4項 「NetworkManagerとVPN」を参照してください。

ネットワーク接続を使用できない

ネットワーク接続が正しく設定され、ネットワーク接続の他のすべてのコンポーネントも(ルータなど)、正常に機能している場合は、コンピュータ上でネットワークインタフェースを再起動すると、問題が解決する場合があります。そのためには、コマンドラインに`root`としてログインし、`systemctl restart wicked.services`を実行します。

24.7 その他の情報

NetworkManagerの詳細については、次のWebサイトおよびディレクトリから入手可能です。

NetworkManagerプロジェクトページ

http://projects.gnome.org/NetworkManager/

パッケージのマニュアル

NetworkManagerおよびGNOMEアプレットの最新情報については、次のディレクトリにある情報も参照してください。

- `/usr/share/doc/packages/NetworkManager/`
- `/usr/share/doc/packages/NetworkManager-gnome/`

25 Samba

Sambaを使用すると、Mac OS X、Windows、OS/2マシンに対するファイルサーバおよびプリントサーバをUnixマシン上に構築できます。Sambaは、今や成熟の域に達したかなり複雑な製品です。YaSTで、または環境設定ファイルを手動で編集することで、Sambaを設定します。

25.1 用語集

ここでは、Sambaのマニュアルや YaSTモジュールで使用される用語について説明します。

SMBプロトコル

SambaはSMB(サーバメッセージブロック)プロトコルを使用します。SMBはNetBIOSサービスを基にしています。Microsoftがこのプロトコルをリリースしたので、他のソフトウェアメーカは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。

ヒント: IBM System z: NetBIOSのサポート

IBM System zではSMB over TCP/IPのみがサポートされています。これら2つのシステムではNetBIOSをサポートしていません。

CIFSプロトコル

CIFS (common Internet file system)プロトコルは、Sambaがサポートしているプロトコルです。CIFSは、ネットワーク上で使用する標準のリモートファイルシステムで、ユーザグループによる共同作業およびネットワーク間でのドキュメントの共有ができるようにします。

NetBIOS

NetBIOSは、マシン間通信用に設計された、ネームサービスを提供するソフトウェアインタフェース(API)です。これにより、ネットワークに接続されたマシンが、それ自体の名前を維持できます。予約を行えば、これらのマシンを名前によって指定できます。名前を確認する一元的なプロセスはありません。ネットワーク上のマシンでは、すでに使用済みの名前でない限り、名前をいくつでも予約できます。NetBIOSインタフェースは、異なるネットワークアーキテクチャに実装できるようになっています。ネットワークハードウェアと比較的密接に機能する実装はNetBEUIと呼ばれますが、これはよくNetBIOSとも呼ばれます。NetBIOSとともに実装されるネットワークプロトコルは、Novell IPX (TCP/IP経由の NetBIOS)とTCP/IPです。

TCP/IP経由で送信されたNetBIOS名は、`/etc/hosts`で使用されている名前、またはDNSで定義された名前とまったく共通点がありません。NetBIOSは独自の、完全に独立した名前付け規則を使用しています。しかし、管理を容易にするために、DNSホスト名に対応する名前を使用するか、DNSをネイティブで使用することをお勧めします。これはSambaが使用するデフォルトでもあります。

Sambaサーバ

Sambaサーバは、SMB/CIFSサービスおよびNetBIOS over IPネーミングサービスをクライアントに提供します。Linuxの場合、3種類のSambaサーバデーモン(SMB/CIFSサービス用smnd、ネーミングサービス用nmbd、認証用winbind)が用意されています。

Sambaクライアント

Sambaクライアントは、SMBプロトコルを介してSambaサーバからSambaサービスを使用するシステムです。Mac OS X、Windows、OS/2などの一般的なオペレーティングシステムは、すべてSMBプロトコルをサポートしています。TCP/IPプロトコルは、すべてのコンピュータにインストールする必要があります。Sambaは、異なるUNIXフレーバーに対してクライアントを提供します。Linuxでは、SMB用のカーネルモジュールがあり、Linuxシステムレベルでの SMBリソースの統合が可能です。Sambaクライアントに対していずれのデーモンも実行する必要はありません。

共有

SMBサーバは、そのクライアントに対し、共有によってリソースを提供します。共有は、サーバ上のサブディレクトリのあるディレクトリおよびプリンタです。これは名前によってエクスポートされ、名前によってアクセスされます。共有名にはどのような名前も設定できます。エクスポートディレクトリの名前である必要はありません。プリンタにも名前が割り当てられます。クライアントはプリンタに名前でアクセスできます。

DC

ドメインコントローラ(DC)は、ドメインのアカウントを処理するサーバです。データレプリケーションには、1つのドメインの中で追加のドメインコントローラが使用できます。

25.2 Sambaサーバのインストール

Sambaサーバをインストールするには、YaSTを起動して、[ソフトウェア] › [ソフトウェア管理]の順に選択します。[表示 []] › [パターン]の順に選択し、[ファイルサーバ]を選択します。必要なパッケージのインストールを確認して、インストールプロセスを完了します。

25.3 Sambaの起動および停止

Sambaサーバは、自動(ブート中)か手動で起動または停止できます。ポリシーの開始および停止は、25.4.1項「YaSTによるSambaサーバの設定」で説明しているように、YaST Sambaサーバ設定の一部です。

コマンドラインで、「`systemctl stop smb.service nmb.service`」と入力して、Sambaに必要なサービスを停止し、「`systemctl start nmb.service smb.service`」と入力して起動します。`smb.service`は、必要に応じて`winbind`を処理します。

ヒント

`winbind`は、独立したサービスであり、個別の`samba-winbind`パッケージとしても提供されます。

25.4 Sambaサーバの設定

SUSE® Linux Enterprise ServerのSambaサーバは、YaSTを使って、または手動で設定することができます。手動で設定を行えば細かい点まで調整できますが、YaSTのGUIほど便利ではありません。

25.4.1 YaSTによるSambaサーバの設定

Sambaサーバを設定するには、YaSTを起動して、[ネットワークサービス] › [Sambaサーバ]の順に選択します。

25.4.1.1 初期Samba設定

このモジュールを初めて起動すると、[Sambaインストール]ダイアログが起動して、サーバ管理に関していくつかの基本的な事項を決定するように要求されます。設定の最後に、Samba管理者パスワードを要求されます([Sambaルートパスワード])。次回起動時には、[Samba Configuration]ダイアログが表示されます。

[Sambaインストール]ダイアログは、次の2つのステップとオプションの詳細設定で構成されています。

ワークグループまたはドメイン名

［Workgroup or Domain Name］から既存の名前を選択するか、新しい名前を入力し、［次へ］を入力します。

Sambaサーバのタイプ

次のステップでは、サーバをPDC(プライマリドメインコントローラ)として機能させるか、BDC(バックアップドメインコントローラとして機能させるか、またはドメインコントローラとしては機能させないかを指定します。［次へ］で続行します。

詳細なサーバ設定に進まない場合は、［OK］を選択して確認します。次に、最後のポップアップボックスで、［Sambaルートパスワード］を設定します。

この設定はすべて、後から［Sambaの設定］ダイアログで［起動］、［共有］、［識別情報］、［信頼されたドメイン］、［LDAP設定］の各タブを使用して変更することができます。

25.4.1.2 Sambaの詳細設定

Sambaサーバモジュールの初回起動中、2つの初期化ステップ(25.4.1.1項 「初期Samba設定」参照)の直後に［Sambaの設定］ダイアログが表示されます。ここでは、Sambaサーバの設定を編集することができます。

設定を編集し終わったら、［OK］をクリックして設定を保存します。

25.4.1.2.1 サーバを起動する

［Start Up］［タブで、Sambaサーバの起動に関する設定を行います。］システムのブート時に毎回サービスが起動されるようにするには、［During Boot］を選択します。手動起動を有効化するには、［Manually］を選択します。Sambaサーバの起動の詳細については、25.3項 「Sambaの起動および停止」を参照してください。

このタブで、ファイアウォールのポートを開くこともできます。そのためには、［Open Port in Firewall］を選択します。複数のネットワークインタフェースがある場合は、［Firewall Details］をクリックし、インタフェースを選択した後、［OK］をクリックして、Sambaサービス用のネットワークインタフェースを選択します。

25.4.1.2.2 共有

［共有］［タブで、有効にするSambaの共有を指定します。］homesおよびプリンタなど、事前定義済みの共有がいくつかあります。［状態の変更］を使用して、［有効］と［無効］の間で切り替えます。新規の共有を追加するには［追加］、共有を削除するには［削除］をクリックします。

[ユーザにディレクトリの共有を許可する]を選択すると、[許可するグループ]中のグループメンバーに、各自のディレクトリを他のユーザと共有させることができます。たとえば、ローカルの範囲の`users`、あるいはドメインの範囲では`DOMAIN\Users`を設定します。また、ユーザにはファイルシステムへのアクセスを許可するパーミッションがあることを確認してください。[最大共有数]で、共有の最大数を制限することができます。認証なしでユーザ共用へのアクセスを許可するには、[ゲストアクセスを許可]を有効にします。

25.4.1.2.3 ID

[識別情報]タブで、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。名前解決にMicrosoft Windows Internet Name Service(WINS)を使用することもできます。この場合、[Use WINS for Hostname Resolution]を有効にし、DHCP経由でWINSサーバを取得([Retrieve WINS server via DHCP]を使用)するかどうか決定します。TDBデータベースではなくLDAPなど、エキスパートグローバル設定またはユーザ認証ソースを設定するには、[詳細設定]をクリックします。

25.4.1.2.4 信頼されたドメイン

他のドメインのユーザを、自分のドメインにアクセスさせるには、[Trusted Domains]タブで適切な設定を行います。新しいドメインを追加するには、[追加]をクリックします。選択したドメインを削除するには、[削除]をクリックします。

25.4.1.2.5 LDAP設定

[LDAP Settings][タブでは、認証に使用するLDAPサーバを設定することができます。]LDAPサーバへの接続をテストするには、[Test Connection]をクリックします。エキスパートLDAP設定を設定するか、デフォルト値を使用する場合、[詳細な設定]をクリックします。

LDAP設定に関する詳細については、Book “Security Guide” 5 “LDAP—A Directory Service”を参照してください。

25.4.2 サーバの手動設定

Sambaをサーバとして使用する場合は、`samba`をインストールします。Sambaの主要設定ファイルは、`/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。次のデフォルトのセクションには、個別のファイルとプリンタ共有が入っています。

- [homes]
- [プロファイル]
- [ユーザ]
- [グループ]
- [プリンタ]
- [印刷$]

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。

25.4.2.1 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 = WORKGROUP`

この行は、Sambaサーバをワークグループに割り当てます。`WORKGROUP` 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。このパラメータに関する詳細については、`smb.conf` のマニュアルページを参照してください。

`os level = 20`

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(たとえば `2`)を選択します。この重要なトピックの詳細については、『Samba 3 Howto』のネット「ワークブラウジング」の章を参照してください。『Samba 3 Howto』の詳細については、25.9項「その他の情報」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 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。はじめに、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、`wins support = Yes`オプションを設定します。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。

25.4.2.2 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。

例 25.1 CD-ROMの共有

```
[cdrom]
       comment = Linux CD-ROM
       path = /media/cdrom
       locking = No
```

`[cdrom]`および`コメント`

`[cdrom]`セクションエントリは、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]`

`[homes]` 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。

例 25.2 [HOMES]共有

```
[homes]
 comment = Home Directories
 valid users = %S
 browseable = No
 read only = No
 inherit acls = Yes
```

[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。生成される共有の名前は、ユーザ名になります。

`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、`read only = No` 値を設定します。これは `writable = 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パーミッションを割り当てることができません。`create mask` パラメータは、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。

25.4.2.3 セキュリティレベル

セキュリティを向上させるため、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では、次の方法で権限を確認できます。

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。

ADSレベルのセキュリティ(`セキュリティ=ADS`)

このモードでは、Sambaはアクティブディレクトリ環境のドメインメンバーとして動作します。このモードで操作するには、Sambaを実行しているコンピュータにKerberosがインストールされ設定済みであることが必要です。Sambaを使用してコンピュータをADSレルムに結合させる必要があります。これは、YaSTの［Windowsドメインメンバーシップ］モジュールを使用して行います。

ドメインレベルのセキュリティ(`セキュリティ=ドメイン`)

このモードは、マシンがWindows NTドメインに参加している場合にのみ正しく動作します。Sambaはユーザ名とパスワードをWindows NT プライマリドメインコントローラまたはバックアップドメインコントローラに渡すことによって、これらを検証しようとします。Windows NT Serverが行うのと同じ方法です。暗号化されたパスワードパラメータが `yes` に設定されている必要があります。

共有、ユーザ、サーバ、またはドメインレベルのセキュリティの設定は、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 3 HOWTO』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。

25.5 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。

25.5.1 YaSTによるSambaクライアントの設定

SambaクライアントをSambaサーバまたはWindowsサーバ上のリソース(ファイルまたはプリンタ)にアクセスするように設定します。NTまたはActive Directoryのドメインまたはワークグループを、[ネットワークサービス] › [Windowsドメインメンバーシップ]の順に選択して表示したダイアログに入力します。[Linuxの認証にもSMBの情報を使用する]を有効にした場合、ユーザ認証は、Samba、NT、またはKerberosのサーバ上で実行されます。

[エキスパート設定]をクリックして、高度な設定オプションを設定します。たとえば、認証による自動的なサーバホームディレクトリのマウントを有効化するには、[サーバディレクトリのマウント]のテーブルを使用します。これにより、CIFS上でホストされると、ホームディレクトリにアクセスできるようになります。詳細については、`pam_mount` のマニュアルページを参照してください。

すべての設定を完了したら、ダイアログを確認して設定を終了します。

25.6 ログインサーバとしてのSamba

Windowsクライアントが大部分を占めるネットワークでは、ユーザが有効なアカウントとパスワードを持つ場合のみ登録できることが求められるのが普通です。Windowsベースのネットワークでは、このタスクはPDC (プライマリドメインコントローラ)によって処理されます。Windows NTサーバをPDCとして使用することもできますが、Sambaサーバを使用しても処理できます。例25.3「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。

例 25.3 SMB.CONFファイルのグローバルセクション

```
[global]
    workgroup = WORKGROUP
    domain logons = Yes
    domain master = Yes
```

ユーザアカウントとパスワードをWindowsに準拠した暗号化形式で作成する必要があります。そのためにはコマンド `smbpasswd -a name` を実行します。さらに次のコマンドを使用して、Windows ドメイン概念で必要になるコンピュータのドメインアカウントを作成します。

```
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。

```
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正常に実行できるようにするため、必要な管理者権限を持つSamba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
```

25.7 Active Directoryネットワーク内のSambaサーバ

LinuxサーバとWindowsサーバの両方を利用する場合、2つの独立した認証システムまたはネットワークを作成するか、または単一の中央認証システムを持つ単一のネットワークに両方のサーバを接続します。SambaはActive Directoryドメインと連携できるため、お使いのSUSE Linux Enterprise ServerをActive Directory (AD)に参加させることができます。

Active Directoryドメインに参加させるには、次の手順に従います。

1. rootとしてログインし、YaSTを起動します。
2. [ネットワークサービス] › [Windowsドメインメンバーシップ]の順に選択します。
3. [Windows Domain Membership]画面の[Domain or Workgroup]に、参加するドメインを入力します。

図 25.1 WINDOWSドメインメンバーシップの決定

4. SUSE Linux Enterprise ServerでLinux認証にSMBソースを使用する場合は、[Linuxの認証にもSMBの情報を用いる]を選択します。

5. ドメインへの参加を確認するメッセージが表示されたら、[OK]をクリックします。

6. Active DirectoryサーバのWindows管理者用パスワードを入力し、[OK]をクリックします。Active Directoryドメインコントローラから、すべての認証データを取得できるようになりました。

25.8 詳細トピック

このセクションでは、Sambaスイートのクライアントとサーバの両方の部分を管理するためのより高度なテクニックを紹介します。

25.8.1 Btrfsでの透過的なファイル圧縮

Sambaでは、クライアントは、Btrfsファイルシステムに配置されている共有のファイルおよびディレクトリの圧縮フラグをリモートで操作できます。Windowsエクスプローラでは、[ファイル] › [プロパティ] › [詳細]ダイアログを使用することで、ファイル/ディレクトリに透過的な圧縮対象のフラグを付けることができます。

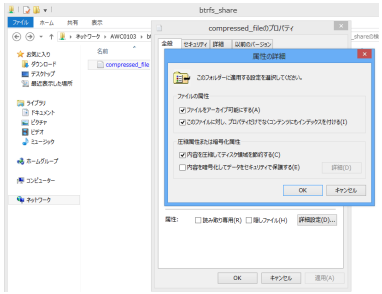


図 25.2 WINDOWSエクスプローラの［属性の詳細］ダイアログ

圧縮対象フラグが付いたファイルは、アクセスまたは変更があると、基礎となるファイルシステムによって透過的に圧縮および圧縮解除されます。通常、これによってファイルアクセス時に余分なCPUオーバーヘッドが生じますが、ストレージ容量の節約になります。新しいファイルとディレクトリは、FILE_NO_COMPRESSIONオプションを指定して作成しない限り、親ディレクトリの圧縮フラグを継承します。

Windowsエクスプローラでは、圧縮ファイルとディレクトリは、未圧縮のファイル/ディレクトリとは視覚的に見分けが付くように表示されます。

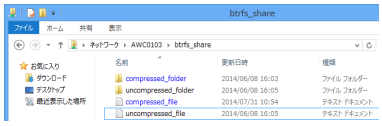


図 25.3 WINDOWSエクスプローラでの圧縮ファイルのディレクトリリスト

Samba共有の圧縮を有効にするには、手動で、

```
vfs objects = btrfs
```

`/etc/samba/smb.conf`に共有設定を追加して実行するか、YaSTを使用して［ネットワークサービス］›［Sambaサーバ］›［追加］の順に選択して［btrfs機能を利用する］をオンにします。

25.8.2 スナップショット

スナップショット(シャドウコピーとも呼ばれる)は、特定の時点におけるファイルシステムサブボリュームの状態のコピーです。Snapperは、Linuxでこれらのスナップショットを管理するためのツールです。スナップショットは、Btrfsファイルシステムまたはシンプロビジョニングされ LVMボリュームでサポートされています。Sambaスイートは、サーバ側とクライアント側の両方で、FSRVPプロトコルを介したリモートスナップショットの管理をサポートしています。

25.8.2.1 以前のバージョン

Sambaサーバのスナップショットは、ファイルまたはディレクトリの以前のバージョンとしてリモートWindowsクライアントにエクスポートできます。

Sambaサーバでスナップショットを有効にするには、次の条件を満たしている必要があります。

- SMBネットワーク共有がBtrfsサブボリューム上に存在している。
- SMBネットワーク共有のパスに、関連するSnapper環境設定ファイルが含まれている。次のコマンドを使用して、Snapperファイルを作成できます。

  ```
  snapper -c <cfg_name> create-config /path/to/share
  ```

  Snapperの詳細については、第4章　Snapperを使用したシステムの回復とスナップショット管理を参照してください。
- スナップショットディレクトリツリーでは、関連するユーザにアクセスを許可する必要があります。詳細については、vfs_snapperマニュアルページの「PERMISSIONS」のセクション(`man 8 vfs_snapper`)を参照してください。

リモートスナップショットをサポートするには、`/etc/samba/smb.conf`ファイルを変更する必要があります。変更するには、[YaST] › [ネットワークサービス] › [Sambaサーバ]の順に選択するか、または次のコマンドを使用して関連する共有セクションを手動で拡張します。

```
vfs objects = snapper
```

手動での`smb.conf`への変更を有効にするために、Sambaサービスを再起動する必要がある点に注意してください。

```
systemctl restart nmb.service smb.service
```

図 25.4 スナップショットが有効な新しいSAMBA共有の追加

設定後、Samba共有パスでSnapperによって作成されたスナップショットには、Windowsエクスプローラのファイルまたはディレクトリの［以前のバージョン］タブからアクセスできます。

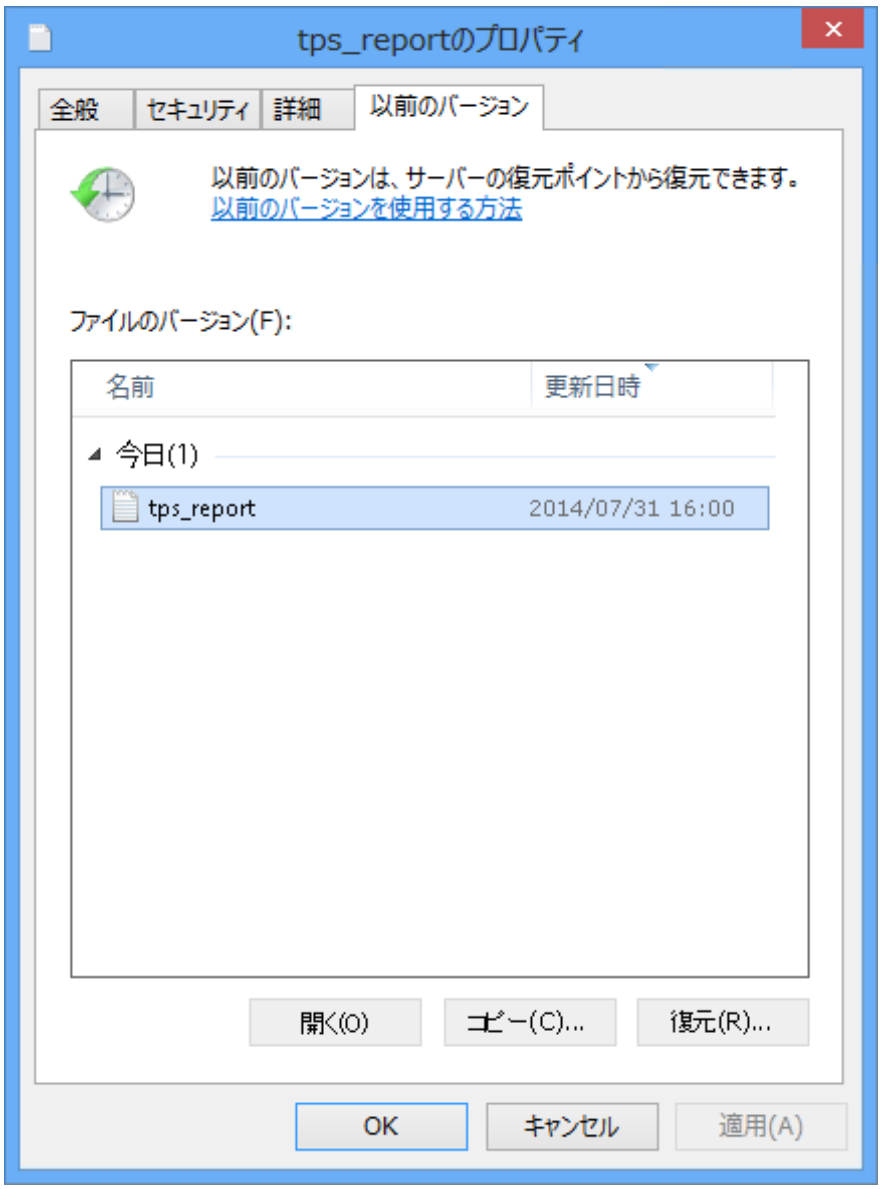


図 25.5 WINDOWSエクスプローラの［以前のバージョン］タブ

25.8.2.2 リモート共有スナップショット

デフォルトでは、スナップショットは、Snapperコマンドラインユーティリティまたは Snapperのタイムライン機能を使用して、Sambaサーバ上でローカルでのみ作成および削除できます。

Sambaは、リモートホストからの共有スナップショット作成および削除要求をFSRVP (File Server Remote VSS Protocol)を使用して処理するように設定できます。

25.8.2.1項 「以前のバージョン」で説明されている環境設定と前提条件に加え、`/etc/samba/smb.conf` に次のグローバル設定が必要です。

```
[global]
rpc_daemon:fssd = fork
registry shares = yes
include = registry
```

その後、FSRVPクライアント(Sambaの`rpcclient`およびWindows Server 2012 `DiskShadow.exe`を含む)は、特定の共有のスナップショットを作成または削除したり、スナップショットを新しい共有として公開したりするようSambaに命令できます。

25.8.2.3 `rpcclient`によるLinuxからのスナップショットのリモート管理

`samba-client`パッケージには、特定の共有の作成と公開をWindows/Sambaサーバにリモートで要求できるFSRVPクライアントが含まれています。SUSE Linux Enterprise Serverの既存のツールを使用して、公開された共有をマウントし、そのファイルをバックアップできます。サーバへの要求は、`rpcclient`バイナリを使用して送信されます。

例 25.4 `rpcclient`を使用したWINDOWS SERVER 2012共有スナップショットの要求

`win-server.example.com`サーバに`EXAMPLE`ドメインの管理者として接続します。

```
# rpcclient -U 'EXAMPLE\Administrator' ncacn_np:win-
server.example.com[ndr64,sign]
Enter EXAMPLE/Administrator's password:
```

`rpcclient`にSMB共有が表示されることを確認します。

```
rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)
```

SMB共有がスナップショットの作成をサポートしていることを確認します。

```
rpcclient $> fss_is_path_sup windows_server_2012_share \
UNC \\WIN-SERVER\windows_server_2012_share\ supports shadow copy requests
```

共有スナップショットの作成を要求します。

```
rpcclient $> fss_create_expose backup ro windows_server_2012_share
13fe880e-e232-493d-87e9-402f21019fb6: shadow-copy set created
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777): \
\\WIN-SERVER\windows_server_2012_share\ shadow-copy added to set
13fe880e-e232-493d-87e9-402f21019fb6: prepare completed in 0 secs
13fe880e-e232-493d-87e9-402f21019fb6: commit completed in 1 secs
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777): \
share windows_server_2012_share@{1C26544E-8251-445F-BE89-D1E0A3938777} \
exposed as a snapshot of \\WIN-SERVER\windows_server_2012_share\
```

スナップショット共有がサーバによって公開されたことを確認します。

```
rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)

netname: windows_server_2012_share@{1C26544E-8251-445F-BE89-D1E0A3938777}
remark: (null)
path:   \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy{F6E6507E-F537-11E3-9404-
B8AC6F927453}\Shares\windows_server_2012_share\
password:       (null)
```

スナップショット共有の削除を試みます。

```
rpcclient $> fss_delete windows_server_2012_share \
13fe880e-e232-493d-87e9-402f21019fb6 1c26544e-8251-445f-be89-d1e0a3938777
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777): \
\\WIN-SERVER\windows_server_2012_share\ shadow-copy deleted
```

スナップショット共有がサーバによって削除されたことを確認します。

```
rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)
```

25.8.2.4 `DiskShadow.exe`によるWindowsからのスナップショットのリモート管理

同様に、Linux　　Sambaサーバ上のSMB共有のスナップショットを、クライアントとして動作するWindows環境から管理できます。Windows　Server　2012には、25.8.2.3項　「`rpcclient`によるLinuxからのスナップショットのリモート管理」で説明した`rpcclient`と同様にリモート共有を管理できる`DiskShadow.exe`ユーティリティが含まれています。最初にSambaサーバを慎重に設定する必要がある点に注意してください。

以下は、Windows　Serverクライアントが共有のスナップショットを管理できるようにSambaサーバを設定する手順の例です。EXAMPLEはテスト環境で使用されるActive　Directoryドメイン、fsrvp-server.example.comはSambaサーバのホスト名、`/srv/smb`はSMB共有のパスである点に注意してください。

手順 25.1 SAMBAサーバの詳細な設定

1. YaSTを介してActive　Directoryドメインに参加します。詳細については、25.7項　「Active Directoryネットワーク内のSambaサーバ」を参照してください。
2. Active DirectoryのDNSエントリが正しいことを確認します。

```
fsrvp-server:~ # net -U 'Administrator' ads dns register \
fsrvp-server.example.com <IP address>
Successfully registered hostname with DNS
```

3. Btrfsサブボリュームを`/srv/smb`に作成します。

```
fsrvp-server:~ # btrfs subvolume create /srv/smb
```

4. パス`/srv/smb`にSnapper環境設定ファイルを作成します。

```
fsrvp-server:~ # snapper -c <snapper_config> create-config /srv/smb
```

5. パス`/srv/smb`に新しい共有を作成し、YaSTの［スナップショットを公開する］チェックボックスをオンにします。25.8.2.2項　「リモート共有スナップショット」に説明されているように、次のスニペットを`/etc/samba/smb.conf`のグローバルセクションに追加します。

```
[global]
 rpc_daemon:fssd = fork
 registry shares = yes
 include = registry
```

6. `systemctl restart nmb.service smb.service`を実行して、Sambaを再起動します。

7. Snapperのパーミッションを設定します。

```
fsrvp-server:~ # snapper -c <snapper_config> set-config \
ALLOW_USERS="EXAMPLE\\\\Administrator EXAMPLE\\\\win-client$"
```

ALLOW_USERSに`.snapshots`サブディレクトリのトラバースも許可されていることを確認します。

```
fsrvp-server:~ # snapper -c <snapper_config> set-config SYNC_ACL=yes
```

重要: パスのエスケープ

「¥」エスケープには注意してください。`/etc/snapper/configs/<snapper_config>`に保存された値を確実に1回エスケープするには、2回エスケープします。

「EXAMPLE¥win-client$」はWindowsクライアントのコンピュータアカウントに対応します。Windowsは、このアカウントが認証されている間に初期FSRVP要求を発行します。

8. Windowsクライアントアカウントに必要な特権を付与します。

```
fsrvp-server:~ # net -U 'Administrator' rpc rights grant \
"EXAMPLE\\win-client$" SeBackupPrivilege
Successfully granted rights.
```

「EXAMPLE¥Administrator」ユーザの場合、すでに特権が付与されているため、上のコマンドは必要ありません。

手順 25.2 WINDOWSクライアントのセットアップと`DiskShadow.exe`の実行

1. Windows Server 2012 (ホスト名の例:WIN-CLIENT)をブートします。

2. Active DirectoryドメインEXAMPLEに、SUSE Linux Enterprise Serverで参加します。

3. 再起動します。

4. Powershellを開きます。

5. `DiskShadow.exe`を起動し、バックアップ手順を開始します。

```
PS C:\Users\Administrator.EXAMPLE> diskshadow.exe
Microsoft DiskShadow version 1.0
Copyright (C) 2012 Microsoft Corporation
On computer:  WIN-CLIENT,  6/17/2014 3:53:54 PM

DISKSHADOW> begin backup
```

6. プログラムを終了、リセット、または再起動しても、シャドウコピーが継続動作するように指定します。

```
DISKSHADOW> set context PERSISTENT
```

7. 指定した共有がスナップショットをサポートしているかどうかを確認し、スナップショットを作成します。

```
DISKSHADOW> add volume \\fsrvp-server\sles_snapper

DISKSHADOW> create
Alias VSS_SHADOW_1 for shadow ID {de4ddca4-4978-4805-8776-cdf82d190a4a} set as
 \
 environment variable.
Alias VSS_SHADOW_SET for shadow set ID {c58e1452-c554-400e-a266-d11d5c837cb1} \
 set as environment variable.

Querying all shadow copies with the shadow copy set ID \
 {c58e1452-c554-400e-a266-d11d5c837cb1}

 * Shadow copy ID = {de4ddca4-4978-4805-8776-cdf82d190a4a}     %VSS_SHADOW_1%
    - Shadow copy set: {c58e1452-c554-400e-a266-d11d5c837cb1}  %VSS_SHADOW_SET%
    - Original count of shadow copies = 1
    - Original volume name: \\FSRVP-SERVER\SLES_SNAPPER\ \
      [volume not on this machine]
    - Creation time: 6/17/2014 3:54:43 PM
    - Shadow copy device name:
      \\FSRVP-SERVER\SLES_SNAPPER@{31afd84a-44a7-41be-b9b0-751898756faa}
    - Originating machine: FSRVP-SERVER
    - Service machine: win-client.example.com
    - Not exposed
```

```
    - Provider ID: {89300202-3cec-4981-9171-19f59559e0f2}
    - Attributes:  No_Auto_Release Persistent FileShare

Number of shadow copies listed: 1
```

8. バックアップ手順を終了します。

```
DISKSHADOW> end backup
```

9. スナップショットが作成された後、その削除を試み、削除されたことを確認します。

```
DISKSHADOW> delete shadows volume \\FSRVP-SERVER\SLES_SNAPPER\
Deleting shadow copy {de4ddca4-4978-4805-8776-cdf82d190a4a} on volume \
 \\FSRVP-SERVER\SLES_SNAPPER\ from provider \
{89300202-3cec-4981-9171-19f59559e0f2} [Attributes: 0x04000009]...

Number of shadow copies deleted: 1

DISKSHADOW> list shadows all

Querying all shadow copies on the computer ...
No shadow copies found in system.
```

25.9 その他の情報

Sambaのマニュアルは`samba-doc`パッケージに付属していますが、デフォルトではインストールされません。インストールするには、`zypper install samba-doc`を実行します。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、`/usr/share/doc/packages/samba`ディレクトリに格納されているその他のオンラインマニュアルと例を参照できます。また、コメント付きの設定例(`smb.conf.SUSE`)が`examples`サブディレクトリに用意されています。Samba関連の情報を参照できるもう1つのファイルは、`/usr/share/doc/packages/samba/README.SUSE`です。

Sambaチームが作成した『Samba- HOWTO』(https://wiki.samba.org を参照)では、トラブルシューティングについても説明されています。またマニュアルのPart Vでは、手順を追って設定を確認するためのガイドが用意されています。

26 NFS共有ファイルシステム

ネットワーク上でファイルシステムを分散して共有することは、企業環境では一般的なタスクです。十分に実績のあるネットワークファイルシステム (NFS)は、NIS (Yellow Pagesプロトコル)と連携して機能します。LDAPと連携して機能し、Kerberosも使用できるより安全なプロトコルについては、NFSv4をチェックしてください。pNFSとの組み合わせで、パフォーマンスのボトルネックをなくすことができます。

NFSをNISと連携して使用すると、ネットワークをユーザに対し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任意のファイルシステムを分散できます。適切なセットアップを行えば、現在どの端末を使用しているかに係わりなく、常に同じ環境で作業できます。

重要: 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。タイムアウトを回避するには、機能するDNSシステムが必要です。mountdデーモンは逆引きを行うので、少なくともログ目的でDNSは必要です。

26.1 用語集

以下の用語は、YaSTモジュールで使用されています。

エクスポート

NFSサーバによってエクスポートされ、クライアントがシステムに統合できるディレクトリ。

NFSクライアント

NFSクライアントは、ネットワークファイルシステムプロトコルを介してNFSサーバからのNFSサービスを使用するシステムです。TCP/IPプロトコルはLinuxカーネルにすでに統合されており、追加ソフトウェアをインストールする必要はありません。

NFSサーバ

NFSサーバは、NFSサービスをクライアントに提供します。`nfsd` (ワーカー)、`idmapd` (IDへのユーザおよびグループ名のマッピングと、その逆のマッピング)、`statd` (ファイルのロック)、および `mountd` (マウント要求)。

NFSv3

NFSv3はバージョン3の実装で、クライアント認証をサポートする「古い」ステートレスなNFSです。

NFSv4

NFSv4は、Kerberosによるセキュアなユーザ認証をサポートする新しいバージョン4の実装です。NFSv4で必要なポートは1つのみであるため、NFSv3よりもファイアウォール環境に適しています。
プロトコルはhttp://tools.ietf.org/html/rfc3530で指定されています。

pNFS

パラレル NFS。NFSv4のプロトコル拡張。任意のpNFSクライアントは、NFSサーバ上のデータに直接アクセスできます。

26.2 NFSサーバのインストール

NFSサーバソフトウェアは、デフォルトインストールの一部ではありません。26.3項 「NFSサーバの設定」に従ってNFSサーバを設定すると、必要なパッケージのインストールを自動的に求められます。別の方法として、YaSTまたはZypperでパッケージ `nfs-kernel-server` をインストールします。

NIS同様、NFSはクライアント/サーバシステムです。ただし、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ことができます(インポート)。

注記: NFSボリュームをエクスポート元サーバにローカルでマウントする

エンタープライズクラスのすべてのLinuxシステムの場合と同様に、NFSボリュームのエクスポート元サーバへのローカルマウントは、SUSE Linux Enterpriseシステムではサポートされていません。

26.3 NFSサーバの設定

NFSサーバの設定は、YaSTを使用するか、または手動で完了できます。認証のため、NFSをKerberosと組み合わせることもできます。

26.3.1 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サーバにすることができます。NFSサーバとは、アクセスを許可されたすべてのホスト、またはグループのすべてのメンバーに、ディレクトリやファイルをエクスポートするサーバのことです。これにより、サーバは、ホストごとにアプリケーションをローカルインストールせずにアプリケーションを提供することもできます。

そのようなサーバをセットアップするには、次の手順に従います。

手順 26.1 NFSサーバをセットアップする

1. YaSTを起動し、[ネットワークサービス] › [NFSサーバ]の順に選択します(図26.1「NFSサーバ設定ツール」を参照してください)。追加のソフトウェアをインストールするよう求められることがあります。

図 26.1 NFSサーバ設定ツール

2. [開始]ラジオボタンをオンにします。

3. システム(SuSEfirewall2)でファイアウォールが有効になっている場合は、[ファイアウォールでポートを開く]をオンにします。YaSTは、`nfs` サービスを有効にすることによってNFSサーバの設定を適用します。

4. [NFSv4を有効にする]を選択するかどうかを決定します。NFSv4を無効にした場合、YaSTでサポートされるのはNFSv3およびNFSv2のみになります。

   - NFSv4を選択した場合は、追加で適切なNFSv4ドメイン名を入力します。ここで指定する名前は、このサーバにアクセスするNFSv4クライアントの `/etc/idmapd.conf` ファイルで指定された名前にする必要があります。このパラメータは、NFSv4サポートに必要な `idmapd` デーモンが使用します(サーバとクライアントの両方で)。特に必要のない限り、そのまま `localdomain` (デフォルト)を使用してください。

5. サーバに安全にアクセスするには、[GSSセキュリティを有効にする]をクリックします。この手順の前提条件として、ドメインにKerberosをインストールし、サーバとクライアントの両方でKerberosを有効にしておく必要があります。[次へ]をクリックして、次の設定ダイアログに進みます。

6. ディレクトリをエクスポートするには、ダイアログの上半分にある［ディレクトリの追加］をクリックします。
7. 許可されるホストをまだ設定していない場合は、自動的に別のダイアログが表示されるので、クライアント情報およびオプションを入力します。ホストを示すワイルドカードを入力します(通常はデフォルト設定のまま使用できます)。
   4種類の方法でホストを指定することができます。1台のホスト(名前またはIPアドレス)(single host)、ネットグループ(netgroups)、ワイルドカード(すべてのコンピュータがサーバにアクセスできることを示す `*` など)(wild cards)、およびIPネットワーク(IP networks)です。
   これらのオプションの詳細については、`exports` のマニュアルページを参照してください。
8. ［完了］をクリックして設定を完了します。

26.3.2 ファイルシステムの手動エクスポート

NFSエクスポートサービスの環境設定ファイルは、`/etc/exports` と `/etc/sysconfig/nfs` です。NFSv4サーバ環境設定には、これらのファイルに加えて `/etc/idmapd.conf` も必要です。サービスを起動または再起動するには、`systemctl restart nfsserver.service` コマンドを実行します。これにより、NFSv4が `/etc/sysconfig/nfs` で設定されている場合は、`rpc.idmapd` も起動します。NFSサーバは、RPCポートマッパーに依存しています。したがって、ポートマッパーサービスも起動または再起動してください。

注記

NFSv4は、SUSE Linux Enterprise Serverで利用できる最新版のNFSプロトコルです。NFSv3と同じ方法で、NFSv4でのエクスポート用にディレクトリを設定できるようになりました。

以前のSUSE Linux Enterprise Server 11バージョンでは、`/etc/exports` のバインドマウントが必須でした。これは引き続きサポートされていますが、非推奨になりました。

/etc/exports

`/etc/exports` ファイルには、エントリのリストが含まれています。各エントリはそれぞれ共有するディレクトリと共有方法を示します。`/etc/exports` 中の一般的なエントリは、次の項目から成り立っています。

```
/shared/directory   host(option_list)
```

たとえば、次のような指定内容です。

```
/export/data   192.168.1.2(rw,sync)
```

ここでは、許可されたクライアントを識別するためにIPアドレス `192.168.1.2` が使われています。ホスト名、ホストを表すワイルドカード、または(`*.abc.com` や `*` など)ネットグループ(`@my-hosts`)を使用できます。
すべてのオプションとそれらの意味の詳細については、`exports` のマニュアルページを参照してください(`man exports`)。

/etc/sysconfig/nfs

`/etc/sysconfig/nfs` ファイルには、NFSv4サーバデーモンの動作を決定する小数のパラメータが含まれています。`NFS4_SUPPORT` パラメータを `yes` に設定することが重要です。`NFS4_SUPPORT` は、NFSサーバがNFSv4エクスポートとクライアントをサポートするかどうかを決定します。

/etc/idmapd.conf

Linuxコンピュータ上の各ユーザには、ユーザ名とIDがあります。idmapdは、サーバへのNFSv4要求やクライアントへのNFSv4応答用に、名前とID間のマッピングサービスを提供しています。NFSv4はその通信に名前だけを使用するので、idmapdは、NFSv4のサーバとクライアントの両方で実行されている必要があります。
NFSを使ってファイルシステムを共有するマシン間では、ユーザへのユーザ名とID (uid)の割り当てには同じ方法を使用してください。そのためには、NIS、LDAP、または他の同一ドメイン認証機構を利用することができます。
`/etc/idmapd.conf` ファイルの `Domain` パラメータは、クライアントとサーバの両方に対して同じ値に設定する必要があります。確信のない場合には、クライアントとサーバの両方のファイルで、`localdomain` をそのまま使用してください。環境設定ファイルの例を次に示します。

```
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
```

詳細については、`idmapd` および `idmapd.conf` のマニュアルページを参照してください(`man idmapd` および `man idmapd.conf`)。

/etc/exports または /etc/sysconfig/nfs を変更したら、NFSサーバサービスを起動/再起動します。

```
systemctl restart nfsserver.service
```

/etc/idmapd.conf を変更したら、設定ファイルを再ロードします。

```
killall -HUP rpc.idmapd
```

NFSサービスをブート時に起動する必要がある場合は、次のコマンドを実行します。

```
systemctl enable nfsserver.service
```

26.3.3 NFSでのKerberosの使用

NFSでKerberos認証を使用するには、GSSセキュリティを有効にする必要があります。最初のYaST NFSサーバのダイアログで、[GSSセキュリティを有効にする]を選択します。ただし、この機能を使用するには、機能するKerberosサーバが必要です。YaSTはKerberosサーバの設定は行いません。その提供機能を使用するだけです。YaST環境設定に加えて、Kerberos認証も使用する場合は、NFS設定を実行する前に、少なくとも次の手順を完了してください。

1. サーバとクライアントの両方が、同じKerberosドメインにあることを確認します。つまり、クライアントとサーバが同じKDC(Key Distribution Center)サーバにアクセスし、krb5.keytab ファイル(the default location on any machine is /etc/krb5.keytab)を共有していなければなりません。Kerberosの詳細については、Book “Security Guide” 7 “Network Authentication with Kerberos”を参照してください。
2. クライアントで systemctl start gssd.service コマンドを実行して、gssdサービスを起動します。
3. サーバで systemctl start svcgssd.service コマンドを実行して、svcgssdサービスを起動します。

Kerberos化されたNFSの設定の詳細については、26.5項「詳細情報」のリンクを参照してください。

26.4 クライアントの設定

ホストをNFSクライアントとして設定する場合、他のソフトウェアをインストールする必要はありません。必要なすべてのパッケージは、デフォルトでインストールされます。

26.4.1 YaSTによるファイルシステムのインポート

認証されたユーザは、YaST NFSクライアントモジュールを使用して、NFSディレクトリをNFSサーバからローカルファイルツリーにマウントできます。次の手順に従います。

手順 26.2 NFSディレクトリのインポート

1. YaST NFSクライアントモジュールを起動します。
2. [NFS共有]タブで[追加]をクリックします。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ローカルでマウントするマウントポイントを入力します。
3. NFSv4を使用する場合は、[NFS設定]タブで[NFSv4を有効にする]を選択します。また、[NFSv4ドメイン名]に、NFSv4サーバが使用する値と同じ値が入力されている必要があります。デフォルトドメインは、`localdomain`です。
4. NFSでKerberos認証を使用するには、GSSセキュリティを有効にする必要があります。[GSSセキュリティを有効にする]を選択します。
5. ファイアウォールを使用しており、リモートコンピュータのサービスにアクセスを許可する場合は、[NFS設定]タブで[ファイアウォールでポートを開く]をオンにします。チェックボックスの下には、ファイアウォールのステータスが表示されます。
6. [OK]をクリックして変更内容を保存します。

設定は`/etc/fstab`に書かれ、指定されたファイルシステムがマウントされます。後でYaST設定クライアントを起動した時に、このファイルから既存の設定が取得されます。

26.4.2 ファイルシステムの手動インポート

NFSサーバからファイルシステムを手動でインポートするには、RPCポートマッパーが実行していることが前提条件です。RPCポートマッパーを適切に起動するのは`nfs.service`です。そのため、`root`として「`systemctl start nfs.service`」を入力し、RPCポートマッパーを起動します。次に、`mount`を使用して、ローカルパーティションと同様に、リモートファイルシステムをファイルシステムにマウントできます。

```
mount host:remote-pathlocal-path
```

たとえば、`nfs.example.com`マシンからユーザディレクトリをインポートするには、次の構文を使用します。

```
mount nfs.example.com:/home /home
```

26.4.2.1 自動マウントサービスの使用

autofsデーモンを使用して、リモートファイルシステムを自動的にマウントすることができます。`/etc/auto.master` ファイルに次のエントリを追加します。

```
/nfsmounts /etc/auto.nfs
```

これで、`/nfsmounts` ディレクトリがクライアント上のすべてのNFSマウントのルートディレクトリの役割を果たすようになります(`auto.nfs` ファイルが正しく設定されている場合)。ここでは、`auto.nfs` という名前を使用しましたが、任意の名前を選択することができます。`auto.nfs` で、次のようにしてすべてのNFSマウントのエントリを追加します。

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

`root` として `systemctl start autofs.service` を実行して設定を有効にします。この例で、`server1` の `/data` ディレクトリの `/nfsmounts/localdata` はNFSでマウントされ、`server2` の `/nfsmounts/nfs4mount` はNFSv4でマウントされます。

autofsの実行中に `/etc/auto.master` ファイルを編集した場合、変更を反映するには、`systemctl restart autofs.service` で自動マウント機能を再起動する必要があります。

26.4.2.2 /etc/fstabの手動編集

`/etc/fstab` 内の典型的なNFSv3マウントエントリは、次のようになります:

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

NFSv4マウントの場合は、3番目の列で `nfs` の代わりに `nfs4` を使用します。

```
nfs.example.com:/data /local/pathv4 nfs4 rw,noauto 0 0
```

`noauto` オプションを使用すると、起動時にファイルシステムが自動マウントされません。対応するファイルシステムを手動でマウントする場合は、マウントポイントのみを指定してmountコマンドを短くできます。

```
mount /local/path
```

注記

ただし、`noauto` オプションを入力しないと、起動時に、システムのinitスクリプトによって、それらのファイルシステムがマウントされます。

26.4.3 パラレルNFS(pNFS)

NFSは、1980年代に開発された、もっとも古いプロトコルの1つです。そのため、小さなファイルを共有したい場合は、通常、NFSで十分です。しかし、大きなファイルを送信したい場合や多数のクライアントがデータにアクセスしたい場合は、NFSサーバがボトルネックとなり、システムのパフォーマンスに重大な影響を及ぼします。これは、ファイルのサイズが急速に大きくなっているのに対し、Ethernetの相対速度が追い付いていないためです。

「通常の」NFSサーバにファイルを要求すると、サーバはファイルのメタデータを検索し、すべてのデータを収集して、ネットワークを介してクライアントに送信します。しかし、ファイルが小さくても大きくてもパフォーマンスのボトルネックが問題になります。

- 小さいファイルでは、メタデータの収集に時間がかかる.
- 大きいファイルでは、サーバからクライアントへのデータ送信に時間がかかる

pNFS(パラレルNFS)は、ファイルシステムメタデータをデータの場所から分離することによって、この制限を克服します。このため、pNFSには2種類のサーバが必要です。

- データ以外のすべてのトラフィックを扱う「メタデータ」または「制御サーバ」
- データを保持する1つ以上の「ストレージサーバ」

メタデータサーバとストレージサーバによって、単一の論理NFSサーバが構成されます。クライアントが読み込みまたは書き出しを行う場合、メタデータサーバがNFSv4クライアントに対して、ファイルのチャンクにアクセスするにはどのストレージサーバを使用すればよいかを指示します。クライアントはサーバのデータに直接アクセスできます。

SUSE Linux Enterpriseはクライアント側でのみpNFSをサポートします。

26.4.3.1 YaSTを使用したpNFSクライアントの設定

手順26.2「NFSディレクトリのインポート」に従って進めます。ただし、[pNFS (v4.1)]チェックボックスをクリックし、オプションで[NFSv4共有]をクリックします。YaSTが必要な手順をすべて実行し、必要なすべてのオプションを`/etc/exports`ファイルに書き込みます。

26.4.3.2 pNFSクライアントの手動設定

26.4.2項 「ファイルシステムの手動インポート」を参照して開始します。ほとんどの設定はNFSv4サーバによって行われます。pNFSを使用する場合に異なるのは、`minorversion` オプションおよびメタデータサーバ `MDS_SERVER` を `mount` コマンドに追加することだけです。

```
mount -t nfs4 -o minorversion=1 MDS_SERVER MOUNTPOINT
```

デバッグを支援するために、`/proc` ファイルシステムの値を変更します。

```
echo 32767 > /proc/sys/sunrpc/nfsd_debug
echo 32767 > /proc/sys/sunrpc/nfs_debug
```

26.5 詳細情報

NFSサーバとクライアントの設定情報は、`exports`、`nfs`、および `mount` のマニュアルページのほか、`/usr/share/doc/packages/nfsidmap/README` からも入手できます。オンラインドキュメンテーションについては、次のWebサイトを参照してください。

- 詳細な技術ヘルプについては、SourceForge (http://nfs.sourceforge.net/) を参照してください。
- NFSでのKerberosの設定方法は、NFS Version 4 Open Source Reference Implementation (http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html) を参照してください。
- Linux NFSv4 (http://www.citi.umich.edu/projects/nfsv4/linux/faq/) には、NFSv4に関するFAQが用意されています。

27 Autofsによるオンデマンドマウント

autofsは、指定したディレクトリをオンデマンドベースで自動的にマウントするプログラムです。これは高い効率を実現するためにカーネルモジュールに基づいており、ローカルディレクトリとネットワーク共有の両方を管理できます。これらの自動的なマウントポイントは、アクセスがあった場合にのみマウントされ、非アクティブな状態が一定時間続くとアンマウントされます。このオンデマンドの動作によって帯域幅が節約され、/etc/fstabで管理する静的マウントよりも高いパフォーマンスが得られます。autofsは制御スクリプトですが、automountは実際の自動マウントを実行するコマンド(デーモン)です。

27.1 インストール

デフォルトでは、autofsはSUSE Linux Enterprise Serverにインストールされません。その自動マウント機能を利用するには、最初に、次のコマンドを使用してインストールします。

```
sudo zypper install autofs
```

27.2 環境設定

vimなどのテキストエディタで設定ファイルを編集して、autofsを手動で設定する必要があります。autofsの基本的な設定手順は2つあります。「マスタ」マップファイルを使用する手順と、特定のマップファイルを使用する手順です。

27.2.1 マスタマップファイル

autofsのデフォルトのマスタ設定ファイルは/etc/auto.masterです。その場所を変更するには、/etc/sysconfig/autofs内のDEFAULT_MASTER_MAP_NAMEオプションの値を変更します。次に、SUSE Linux Enterprise Serverのデフォルトのマスタ設定ファイルの内容を示します。

```
#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5). ❶
```

```
#
#/misc  /etc/auto.misc❷
#/net -hosts
#
# Include /etc/auto.master.d/*.autofs❸
#
#+dir:/etc/auto.master.d
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master❹
```

❶ 自動マウント機能のマップの形式については、`autofs` のマニュアルページ(`man 5 autofs`)で多くの貴重な情報が提供されています。

❷ デフォルトではコメント化(#)されていますが、これは単純な自動マウント機能のマッピング構文の例です。

❸ マスタマップファイルを複数のファイルに分割する必要がある場合、この行のコメント化を解除し、マッピング(サフィックスは `.autofs`)を `/etc/auto.master.d/` ディレクトリに配置します。

❹ `+auto.master` により、NIS (NISの詳細については、Book “Security Guide” 3 “Using NIS” 3.1 “Configuring NIS Servers”を参照)を使用していてもそのマスタマップが確実に見つかるようになります。

`auto.master` のエントリには3つのフィールドがあり、構文は次のとおりです。

```
mount point      map name      options
```

mount point

`autofs` ファイルシステムをマウントする基本の場所(`/home` など)。

map name

マウントに使用するマップソースの名前。マップファイルの構文については、27.2.2項「マップファイル」を参照してください。

options

これらのオプションを指定した場合、指定したマップ内のすべてのエントリにデフォルトとして適用されます。

ヒント

オプションの `map-type`、`format`、および `options` の特定の値の詳細については、[auto.master] のマニュアルページ(`man 5 auto.master`)を参照してください。

`auto.master` の次のエントリは、`autofs` に対し、`/etc/auto.smb` 内を検索して `/smb` ディレクトリにマウントポイントを作成するよう指示します。

```
/smb    /etc/auto.smb
```

27.2.1.1 直接マウント

直接マウントは、関連するマップファイル内で指定されたパスにマウントポイントを作成します。`auto.master` でマウントポイントを指定するのではなく、マウントポイントフィールドを `/-` に置き換えます。たとえば、次の行は、`autofs` に対し、`auto.smb` で指定された場所にマウントポイントを作成するよう指示します。

```
/-        /etc/auto.smb
```

ヒント: フルパスを使用しないマップ

ローカルまたはネットワークのフルパスでマップファイルを指定していない場合、マップファイルはネームサービススイッチ(NSS)設定を使用して検索されます。

```
/-        auto.smb
```

27.2.2 マップファイル

重要: 他のタイプのマップ

`autofs`による自動マウントのマップタイプとしては「ファイル」が最も一般的ですが、他のタイプもあります。マップは、コマンドの出力や、LDAPまたはデータベースのクエリ結果で指定することもできます。マップタイプの詳細については、`man 5 auto.master`マニュアルページを参照してください。

マップファイルは、ソースの場所(ローカルまたはネットワーク)と、ソースをローカルにマウントするためのマウントポイントを指定します。マップの全般的な形式はマスタマップと同様です。異なるのは、optionsをエントリの最後ではなくmount pointとlocationの間に記述する点です。

```
mount point      options      location
```

mount point
: ソースの場所をどこにマウントするかを指定します。ここには、`auto.master`で指定されたベースマウントポイントに追加する1つのディレクトリ名(「間接」マウント)、またはマウントポイントのフルパス(直接マウント、27.2.1.1項「直接マウント」を参照)のいずれかを指定できます。

options
: 関連するエントリのマウントオプションを、カンマで区切ったオプションのリストで指定します。このマップファイルのオプションも`auto.master`に含まれている場合、これらが追加されます。

location
: ファイルシステムのマウント元の場所を指定します。通常は、標準の表記方法`host_name:path_name`によるNFSまたはSMBボリュームです。マウントするファイルシステムが「/」で始まる場合(ローカルの`/dev`エントリやsmbfs共有など)、`:/dev/sda1`のように、コロン記号「:」のプレフィックスを付ける必要があります。

27.3 操作とデバッグ

このセクションでは、`autofs`サービスの操作を制御する方法と、自動マウント機能の操作を調整する際に詳細なデバッグ情報を表示する方法の概要について説明します。

27.3.1 autofsサービスの制御

autofsサービスの動作は、systemdによって制御されます。autofs用のsystemctlコマンドの一般的な構文は、次のとおりです。

```
sudo systemctl sub-command autofs.service
```

ここで、sub-commandは次のいずれかです。

enable
: ブート時に自動マウント機能のデーモンを起動します。

start
: 自動マウント機能のデーモンを起動します。

stop
: 自動マウント機能のデーモンを停止します。自動マウントポイントにはアクセスできません。

status
: autofsサービスの現在のステータスと、関連するログファイルの一部を出力します。

restart
: 自動マウント機能を停止して起動します。実行中のデーモンをすべて終了し、新しいデーモンを起動します。

reload
: 現在のauto.masterマップを確認して、エントリに変更があるデーモンを再起動し、新しいエントリがある場合は新しいデーモンを起動します。

27.3.2 自動マウント機能の問題のデバッグ

autofsでディレクトリをマウントする際に問題が発生する場合は、automountデーモンを手動で実行して出力メッセージを確認してください。

1. autofsを停止します。

   ```
   sudo systemctl stop autofs.service
   ```

2. 1つの端末から、フォアグラウンドでautomountを手動で実行し、詳細な出力を生成します。

   ```
   sudo automount -f -v
   ```

3. 別の端末から、マウントポイントにアクセスして(たとえば、`cd`または`ls`を使用して)、自動マウントファイルシステムをマウントしてみます。

4. 1番目の端末から、`automount`の出力で、マウントに失敗した理由またはマウントが試行されていない理由についての詳細情報がないかどうかを確認します。

27.4 NFS共有の自動マウント

次の手順は、ネットワーク上で利用可能なNFS共有を自動マウントするよう`autofs`を設定する方法を示しています。この方法は上で説明した情報を利用しています。また、NFSのエクスポートを熟知していることが前提です。NFSの詳細については、第26章 NFS共有ファイルシステムを参照してください。

1. マスタマップファイル`/etc/auto.master`を編集します。

```
sudo vim /etc/auto.master
```

`/etc/auto.master`の最後に新しいNFSマウント用の新しいエントリを追加します。

```
/nfs      /etc/auto.nfs      --timeout=10
```

これは、ベースマウントポイントは`/nfs`で、NFS共有は`/etc/auto.nfs`マップで指定されていることを`autofs`に伝え、非アクティブな状態が10秒間続いたらこのマップ内のすべての共有を自動的にアンマウントするよう指示します。

2. NFS共有用の新しいマップファイルを作成します。

```
sudo vim /etc/auto.nfs
```

通常、`/etc/auto.nfs`には、各NFS共有に対して別個の行が含まれます。形式については、27.2.2項 「マップファイル」を参照してください。マウントポイントおよびNFS共有のネットワークアドレスを記述する行を追加します。

```
export      jupiter.com:/home/geeko/doc/export
```

上述の行は、要求があると、`jupiter.com`ホスト上の`/home/geeko/doc/export`ディレクトリがローカルホスト上の`/nfs/export`ディレクトリ(`/nfs`は`auto.master`マップから取得)に自動マウントされることを意味します。`/nfs/export`ディレクトリは、`autofs`によって自動的に作成されます。

3. 以前に同じNFS共有を静的にマウントしていた場合、必要に応じて`/etc/fstab`の関連する行をコメント化します。行は次のようになります。

```
#jupiter.com:/home/geeko/doc/export /nfs/export nfs defaults 0 0
```

4. `autofs`を再ロードし、動作しているかどうかを確認します。

```
sudo systemctl restart autofs.service
```

```
# ls -l /nfs/export
total 20
drwxr-xr-x  6 1001 users 4096 Oct 25 08:56 ./
drwxr-xr-x  3 root root     0 Apr  1 09:47 ../
drwxr-xr-x  5 1001 users 4096 Jan 14  2013 .images/
drwxr-xr-x 10 1001 users 4096 Aug 16  2013 .profiled/
drwxr-xr-x  3 1001 users 4096 Aug 30  2013 .tmp/
drwxr-xr-x  4 1001 users 4096 Oct 25 08:56 SLE-12-manual/
```

リモート共有上にあるファイルのリストを参照できる場合、`autofs`は機能しています。

27.5 詳細トピック

このセクションでは、`autofs`の基本的な説明よりも詳しいトピックについて説明します。ここで説明するのは、ネットワーク上で利用可能なNFS共有の自動マウント、マップファイルでのワイルドカードの使用、およびCIFSファイルシステムに固有の情報です。

27.5.1 /netマウントポイント

このヘルパーマウントポイントは、大量のNFS共有を使用する場合に便利です。`/net`には、ローカルネットワーク上にあるすべてのNFS共有がオンデマンドで自動マウントされます。このエントリはすでに`auto.master`ファイルに存在しているため、エントリのコメント化を解除して`autofs`を再起動するだけで済みます。

```
/net      -hosts
```

```
systemctl restart autofs.service
```

たとえば、jupiterという名前のサーバと/exportという名前のNFS共有がある場合、

```
# cd /net/jupiter/export
```

コマンドラインで次のように入力してマウントできます。

27.5.2 ワイルドカードを使用したサブディレクトリの自動マウント

個別に自動マウントする必要があるサブディレクトリが含まれるディレクトリがある場合(代表的なケースは、個々のユーザのホームディレクトリが内部にある/homeディレクトリ)、autofsには便利な解決方法が備わっています。

ホームディレクトリの場合は、auto.masterに次の行を追加します。

```
/home      /etc/auto.home
```

続いて、/etc/auto.homeファイルに正しいマッピングを追加し、ユーザのホームディレクトリが自動的にマウントされるようにする必要があります。1つの解決方法は、各ディレクトリに対して個別のエントリを作成することです。

```
wilber      jupiter.com:/home/wilber
penguin      jupiter.com:/home/penguin
tux      jupiter.com:/home/tux
[...]
```

これは、auto.home内にあるユーザのリストを管理する必要があるため、効率的とは言えません。マウントポイントの代わりにアスタリスク「*」を使用し、マウントするディレクトリの代わりにアンパサンド「&」を使用します。

```
*      jupiter:/home/&
```

27.5.3 CIFSファイルシステムの自動マウント

SMB/CIFS共有を自動マウントする場合(SMB/CIFSプロトコルの詳細については、第25章 Sambaを参照)、マップファイルの構文を変更する必要があります。オプションフィールドに-fstype=cifsを追加し、共有の場所にコロン「:」のプレフィックスを付けます。

```
mount point      -fstype=cifs      ://jupiter.com/export
```

28 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、タブレット、またはスマートフォン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。すべてのコンピュータで最新バージョンのデータを使用できるように、どのコンピュータでも作業ができて、ファイルの変更ができればと考えるでしょう。

28.1 使用可能なデータ同期ソフトウェア

データの同期は、高速ネットワークで固定接続されているコンピュータ間ではまったく問題なく実現できます。この場合、NFSなどのネットワークファイルシステムを使用し、ファイルをサーバに保存して、すべてのホストがネットワーク経由で同じデータにアクセスすればよいわけです。ところがこの方法は、ネットワーク接続が低速な場合、または固定でない場合には不可能です。ラップトップをもって外出しているとき、必要なファイルをローカルハードディスクにコピーする必要があります。しかし、そうすると今度は、変更したファイルを同期させる必要があります。1台のコンピュータでファイルを変更したときは、必ず他のすべてのコンピュータでファイルを更新しなければなりません。たまにコピーする程度なら、手動でscpまたはrsyncを使用してコピーすればよいでしょう。しかし、ファイルが多い場合、手順が複雑になるだけでなく、新しいファイルを古いファイルで上書きしてしまうといった間違いを防ぐために細心の注意が必要になります。

警告: データ損失の危険

データを同期システムで管理する前に、使用するプログラムをよく理解し、機能をテストしておく必要があります。重要なファイルのバックアップは不可欠です。

このように手動によるデータの同期は、時間がかかる上に間違いが起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。

28.1.1 CVS

CVSは、多くの場合プログラムソースのバージョン管理に使用されるプログラムで、複数のコンピュータでファイルのコピーを保存する機能を持っています。したがって、データ同期にも適しています。CVSはサーバ上に一元的なリポジトリを設定し、ファイルおよびファイルの変更内容を保存します。ローカルに実行された変更はリポジトリにコミットされ、更新によって他のコンピュータに取得されます。両方の処理はユーザによって実行される必要があります。

CVSは、複数のコンピュータで変更が行われた場合、非常に優れたエラー回復力を発揮します。変更内容がマージされ、同じ行が変更された場合は、競合がレポートされます。競合が生じても、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。

28.1.2 rsync

バージョン管理は不要であっても、低速ネットワーク接続を使用して大きなディレクトリ構造を同期させる必要がある場合は、ツールrsyncの適切に開発されたメカニズムを使用して、ファイル内の変更箇所のみを送信できます。この処理では、テキストファイルのみでなくバイナリファイルも対象となります。ファイル間の差分を検出するために、rsyncはファイルをブロック単位で分割してチェックサムを計算します。

変更内容の検出処理は高コストを伴います。rsyncの使用量に合わせて、同期対象となるシステムの規模を調整する必要があります。特に、RAMが重要です。

28.2 プログラムを選択する場合の決定要因

使用するプログラムを決定する際に重要な要因がいくつかあります。

28.2.1 クライアント/サーバ対ピアツーピア

一般に、データの配信には2種類のモデルが使用されます。1つは、すべてのクライアントが、そのファイルを一元的なサーバによって同期させるモデルです。サーバはすべてのクライアントから、少なくともいずれかの時点でアクセスできる必要があります。このモデルは、CVSが使用します。

もう1つは、すべてのネットワークホストがそれぞれのデータをピアとして相互に同期させるモデルです。rsyncは、実際にクライアントモードで動作しますが、任意のクライアントがサーバとして動作できます。

28.2.2 移植性

CVS、およびrsyncは、各種のUNIXおよびWindowsシステムなど、他の多くのオペレーティングシステムでも使用できます。

28.2.3 インタラクティブと自動制御

CVSでは、ユーザが手動によってデータの同期を開始します。これにより、データの同期を詳細に制御でき、競合の処理も容易です。ただし、同期の間隔が長すぎると、競合が起こりやすくなります。

28.2.4 競合:問題と解決策

複数のユーザが大きなプログラミングプロジェクトにかかわっている場合も、CVSでは、競合はまれにしか発生しません。これはドキュメントが個別の行単位でマージされるためです。競合が起こると、影響を受けるのは1台のクライアントだけです。CVSでは、通常、競合が容易に解決できます。

rsyncには、競合処理の機能はありません。ユーザは、意図せずにファイルを上書きしないように注意し、考えられる競合はすべて手動で解決する必要があります。安全のために、RCSなどのバージョン管理システムを追加採用できます。

28.2.5 ファイルの選択と追加

CVSでは、新しいディレクトリやファイルは、コマンド`cvs add`を使って明示的に追加する必要があります。これにより、同期の対象となるファイルについて、ユーザがより詳細に制御できます。しかし他方で、新しいファイルが見過ごされることが多く、特に`cvs update`の出力に表示される疑問符は、ファイルの数が多いためにたびたび無視されます。

28.2.6 履歴

CVSは追加機能として、古いバージョンのファイルが再構成できます。変更を行うたびに簡単な編集コメントを挿入しておくと、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。

28.2.7 データ量と必要なハードディスク容量

同期の対象となるすべてのホストには、分散されたデータを処理できるだけの十分なハードディスクの空き容量が必要です。CVSでは、サーバ上のリポジトリデータベースに余分な容量が必要となります。ファイルの履歴もサーバに保存されるため、このための容量も別に必要です。テキスト形式のファイルが変更されたときには、変更された行だけを保存すれば足ります。バイナリファイルは、ファイルが変更されるたびに、ファイルのサイズと同じだけの容量が必要なため、テキストより必要な容量が多くなります。

28.2.8 GUI

CVSを使い慣れたユーザは、通常、コマンドラインでプログラムを制御します。しかし、cervisiaのようなLinux用のグラフィカルユーザインタフェースがあり、また他のオペレーティングシステム用にwincvsなども用意されています。開発ツールやEmacsなどのテキストエディタの多くが、CVSをサポートしています。競合の解決は、これらのフロントエンドの方が、はるかに容易です。

28.2.9 使いやすさ

rsyncは、より使いやすく初心者向けです。CVSは、より操作が難しくなっています。ユーザはレポジトリとローカルデータの間のインタラクションを理解する必要があります。データを変更すると、最初にローカルでリポジトリとマージする必要があります。これはコマンド `cvs` または `update` で実行します。次にコマンド `cvs` または `commit` でデータをリポジトリに送信する必要があります。この手順をいったん理解すれば、初心者の方でもCVSを簡単に利用できるようになります。

28.2.10 攻撃に備えるセキュリティ

伝送中、データは妨害や改ざんから保護される必要があります。CVSやrsyncはいずれもSSH (セキュアシェル)経由で容易に使用できるため、この種の攻撃に対するセキュリティを備えています。CVSをrsh(リモートシェル)経由で実行するのは避けるべきです。また、安全でないネットワークでpserverメカニズムを使用してCVSにアクセスすることもお勧めできません。

28.2.11 データ損失からの保護

CVSは、プログラミングプロジェクト管理のため長期間にわたって開発者に使用されてきたため、きわめて安定しています。CVSでは開発履歴が保存されるため、誤ってファイルを削除するといったユーザの誤操作にも対応できます。

表 28.1 ファイル同期化ツールの機能: -- = とても悪い、- = 悪い、または利用不可、O = 普通、+ = 良好、++ = とても良好、X = 利用可能

	CVS	rsync
クライアント/サーバ	C-S	C-S
移植性	Lin、Un*x、Win	Lin、Un*x、Win
対話処理	x	x
Speed	o	+
競合	++	o
ファイル選択	Sel./file, dir.	ディレクトリ
履歴	x	-
ハードディスクスペース	--	o
GUI	o	-
難度	o	+
攻撃	+ (SSH)	+ (SSH)
データ損失	++	+

28.3 CVSの概要

CVSは、個々のファイルが頻繁に編集され、ASCIIテキストやプログラムソーステキストのようなファイル形式で保存される場合の同期に適しています。CVSを使用して他の形式、たとえばJPEGファイルのデータを同期させることは可能ですが、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ため、結果としてデータ量が膨大になります。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。

28.3.1 CVSサーバの設定

サーバとは、すべてのファイルの最新バージョンを含め、有効なファイルが配置されるホストです。固定のワークステーションであれば、どれでもサーバとして使用できます。可能であれば、CVSレポジトリのデータを定期バックアップに含めます。

CVSサーバを設定するとき、できればユーザアクセスをSSH経由で許可します。ユーザがサーバに `tux` として認識され、CVSソフトウェアがサーバとクライアントにインストールされている場合、次の環境変数をクライアント側に設定する必要があります。

```
CVS_RSH=ssh CVSROOT=tux@server:/serverdir
```

コマンド `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
```

28.3.2 CVSの使用

これで、すべてのホストが `cvs co synchome` を使用して同期リポジトリからチェックアウトできます。これにより、クライアントに新しいサブディレクトリ `synchome` が作成されます。変更内容をサーバにコミットするには、ディレクトリ `synchome` (またはそのサブディレクトリ)に移動し、「`cvs commit`」と入力します。

デフォルトでは、すべてのファイル(サブディレクトリを含め)がサーバにコミットされます。個別のファイルまたはディレクトリだけをコミットするには、`cvs commit file1 directory1` のように指定します。新しいファイルとディレクトリは、サーバにコミットする前に、`cvs add file1 directory1` のようなコマンドを使用してレポジトリに追加する必要があります。この後、`cvs commit file1 directory1` を実行して、新しく追加したファイルとディレクトリをコミットします。

他のワークステーションに移動する場合、同じワークステーションの以前のセッションで同期リポジトリからチェックアウトしていない場合は、ここでチェックアウトします。

サーバとの同期は、`cvs update` を使用して起動します。`cvs update file1 directory1` を使用すると、ファイルやディレクトリを個別に更新できます。現行のファイルとサーバに格納されているバージョンとの違いを確認するには、コマンド `cvs diff` または `cvs diff file1 directory1` を使用します。更新によって変更されたファイルを確認する場合は、`cvs -nq update` を使用します。

更新時に表示されるステータス記号の例を次に示します。

U
: ローカルバージョンが更新されました。この更新はサーバが提供しているすべてのファイル、およびローカルにシステムに存在しないすべてのファイルに影響します。

M
: ローカルバージョンが変更されました。サーバ上で変更があれば、その差分がローカルコピーに取り込まれていることがあります。

P
: ローカルバージョンに対し、サーバ上のバージョンからパッチが適用されました。

C
: ローカルファイルが、レポジトリの現在のバージョンと競合しています。

?
: このファイルがCVSに存在しません。

ステータス `M` は、ローカルで変更されたファイルを示します。ローカルコピーをサーバにコミットするか、ローカルファイルを削除して更新を再実行します。この場合、不足しているファイルは、サーバから取得されます。ローカルに変更したファイルをコミットしたが、そのファイルで同じ行に変更があり以前にコミットされている場合は、競合が `C` で示されて表示されることがあります。

この場合、ファイル内の競合マーク(「>>」および「<<」)を確認し、2つのバージョンのどちらを採用するか決定します。これは厄介な作業のため、変更を破棄し、ローカルファイルを削除して「cvs up」と入力し、現在のバージョンをサーバから取得することもできます。

28.4 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。

28.4.1 設定と操作

rsyncには2つの操作モードがあります。このプログラムを使用してデータをアーカイブまたはコピーできます。そのためには、ターゲットシステム上にSSHなどのリモートシェルがあれば十分です。ただし、rsyncをdaemonとして使用し、ネットワークにディレクトリを提供することもできます。

rsync の基本操作モードの場合、特別な設定は不要です。rsync では、ディレクトリ全体を別のシステムに直接ミラー化できます。たとえば、次のコマンドでは、tuxのホームディレクトリのバックアップがバックアップサーバsun上に作成されます。

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

次のコマンドは、ディレクトリを復元する場合に使用します。

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

ここまでの操作は、scpのような通常のコピーツールの場合とほぼ同じです。

rsyncのすべての機能を完全に使用可能にするには、「rsync」モードで操作する必要があります。そのためには、いずれかのシステムでrsyncdデーモンを起動します。設定はファイル /etc/rsyncd.conf 内で行います。たとえば、rsyncでディレクトリ /srv/ftp を使用可能にするには、次の設定を使用します。

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
```

```
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log

[FTP]
        path = /srv/ftp
        comment = An Example
```

続いて、`systemctl start rsyncd.service`を実行して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［サービスマネージャ］で有効にするか、またはコマンドを手動で入力します。

```
root #  systemctl enable rsyncd.service
```

代わりに、syncdをxinetdによって起動することもできます。ただし、この方法はrsyncdの使用頻度が低いサーバの場合にのみ使用してください。

この例では、すべての接続を示すログファイルも作成されます。このファイルは`/var/log/rsyncd.log`に格納されます。

これで、クライアントシステムからの転送をテストできます。そのためには次のコマンドを使用します。

```
rsync -avz sun::FTP
```

このコマンドを入力すると、サーバのディレクトリ`/srv/ftp`にあるファイルがすべてリストされます。このリクエストはログファイル`/var/log/rsyncd.log`にも記録されます。実際の転送を開始するには、ターゲットディレクトリを指定します。現在のディレクトリには`.`.を使用してください。たとえば、次のようにします。

```
rsync -avz sun::FTP .
```

デフォルトでは、rsyncでの同期中にファイルは削除されません。ファイルを削除する必要がある場合は、オプション「`--delete`」を追加してください。新しい方のファイルが削除されないように、代わりにオプション`--update`を使用することもできます。競合が発生した場合は、手動で解決する必要があります。

28.5 詳細情報

CVS

CVSの重要情報については、ホームページ「http://www.cvshome.org」を参照してください。

rsync

rsyncに関する重要な情報は、マニュアルページ `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。

Subversion

サブバージョンは、SUSE Linux Enterprise SDKから入手できます。このSDKは、SUSE Linux Enterprise向けのアドオン製品であり、http://download.suse.com/からダウンロードできます。「`SUSE Linux Enterpriseソフトウェア開発キット`」で検索してください。

29 Apache HTTPサーバ

http://www.netcraft.com/の調査によると、Apache HTTP Server (Apache)は世界で最も広く利用されているWebサーバです。ApacheはApache Software Foundation (http://www.apache.org/)により開発され、ほとんどのオペレーティングシステムに対応しています。SUSE® Linux Enterprise Serverには、Apache version 2.4が付属しています。この章では、Webサーバのインストール、環境設定、設定方法、SSL、CGI、その他のモジュールの使用方法、およびApacheのトラブルシューティング方法について説明します。

29.1 クイックスタート

ここでは、Apacheをすばやく簡単に設定、起動する方法について説明します。Apacheをインストールおよび設定するには、`root`である必要があります。

29.1.1 要件

Apache Webサーバをセットアップする前に、次の要件が満たされていることを確認してください。

1. マシンのネットワークが適切に設定されているか。この項目の詳細については、第19章 ネットワークの基礎を参照してください。
2. マシンの正確なシステム時間は、タイムサーバとの同期により維持されます。これは、HTTPプロトコルの一部が正確な時間に依存するために必要です。この項目の詳細については、第21章 NTPによる時刻の同期を参照してください。
3. 最新のセキュリティアップデートがインストールされています。不明な場合は、YaSTオンラインアップデートを実行します。
4. ファイアウォールで、デフォルトのWebサーバポート(`80`)を開きます。ポートを開くには、SuSEFirewall2を設定して外部ゾーンで[HTTPサーバ]サービスを実行できるようにします。これには、YaSTを使用します。詳細については、Book “Security Guide” 15 “Masquerading and Firewalls” 15.4.1 “Configuring the Firewall with YaST”を参照してください。

29.1.2 インストール

SUSE Linux Enterprise ServerのApacheは、デフォルトではインストールされません。「そのまますぐに」実行できる標準の事前定義された設定を使用してインストールするには、次の手順を使用します。

手順 29.1 デフォルト設定でAPACHEをインストールする

1. YaSTを起動して、[ソフトウェア] › [ソフトウェア管理]の順に選択します。
2. [フィルタ] › [パターン]の順に選択し、[サーバ機能]から[WebおよびLAMPサーバ]を選択します。
3. 依存関係のあるパッケージのインストールを確認して、インストールプロセスを完了します。

このインストールには、`apache2-prefork` マルチプロセシングモジュール とPHP5モジュールが含まれています。モジュールの詳細については、29.4項「モジュールのインストール、有効化、および設定」を参照してください。

29.1.3 開始

Apacheは、ブート時に自動的に起動することも、手動で起動することもできます。

Apacheをターゲット`multi-user.target`および`graphical.target`でブート時に自動的に起動するには、次のコマンドを実行します。

```
root # systemctl enable apache2.service
```

SUSE Linux Enterprise Serverのsystemdターゲットの詳細、およびYaST[サービスマネージャ]の詳細については、10.4項「YaSTを使用したサービスの管理」を参照してください。

シェルを使用してApacheを手動で起動するには、`systemctl start apache2.service`を実行します。

手順 29.2 APACHEが実行中かどうかチェックする

Apacheの起動時にエラーメッセージが表示されなければ、通常、このWeb serverが実行されています。これをテストするには:

1. ブラウザを起動し、http://localhost/ を開きます。
Apacheが立ち上がって稼動している場合は、「It works!」で始まるテストページが表示されます。

2. このページが表示されない場合は、29.8項「トラブルシューティング」を参照してください。

Webサーバの起動後は、ドキュメントを追加、必要に応じて設定を調整、およびモジュールをインストールして機能を追加することができます。

29.2 Apacheの設定

SUSE Linux Enterprise Serverには、2つの設定オプションがあります。

- Apacheを手動で設定する
- ApacheをYaSTで設定する

手動で設定を行えば細かい点まで調整できますが、YaSTのGUIほど便利ではありません。

重要: 設定変更後のApacheのリロードまたは再起動

設定の変更は、ほとんどの場合、Apacheをリロード(または再起動)しないと有効になりません。`systemctl reload apache2.service`を使用してApacheを手動で再ロードするか、29.3項「Apacheの起動および停止」に示されている再起動オプションの1つを使用します。

YaSTでApatcheを設定する場合、これを自動化するには、29.2.3.2項「HTTPサーバの設定」で説明されているように、[HTTPサービス]を[有効]に設定します。

29.2.1 Apache設定ファイル

このセクションでは、Apache設定ファイルの概要を示します。環境設定にYaSTを使用する場合は、これらのファイルを操作する必要はありません。ただし、後で手動設定に切り替える場合に、この情報が役立つことがあります。

Apache設定ファイルは、次の2つの場所にあります。

- `/etc/sysconfig/apache2`
- `/etc/apache2/`

29.2.1.1 /etc/sysconfig/apache2

/etc/sysconfig/apache2 は、ロードするモジュール、インクルードする付加的な設定ファイル、サーバを起動するときのフラグ、コマンドラインに追加するべきフラグなど、Apacheのいくつかのグローバル設定を制御します。このファイルの各設定オプションについては、詳細なドキュメントが存在するので、ここでは説明しません。一般的な目的のWebサーバの場合には、/etc/sysconfig/apache2 の内容を設定するだけで十分でしょう。

29.2.1.2 /etc/apache2/

/etc/apache2/ には、Apacheのすべての設定ファイルが含まれます。ここでは、各ファイルの目的について説明します。各ファイルには、複数の設定オプション(ディレクティブ)が含まれています。これらのファイルの各設定オプションについては、詳細なドキュメントがあるので、ここでは説明しません。

Apache設定ファイルは、次のように編成されます。

```
/etc/apache2/
     |
     |- charset.conv
     |- conf.d/
     |   |
     |   |- *.conf
     |
     |- default-server.conf
     |- errors.conf
     |- httpd.conf
     |- listen.conf
     |- magic
     |- mime.types
     |- mod_*.conf
     |- server-tuning.conf
     |- ssl.*
     |- ssl-global.conf
     |- sysconfig.d
     |   |
     |   |- global.conf
     |   |- include.conf
     |   |- loadmodule.conf . .
     |
```

```
|- uid.conf
|- vhosts.d
|   |- *.conf
```

「ETC/APACHE2」内のAPACHE設定ファイル

`charset.conv`

各言語に使用する文字セットを指定します。このファイルは、編集しないでください。

`conf.d/*.conf`

他のモジュールによって追加される設定ファイル。これらの設定ファイルは、必要に応じて仮想ホスト設定に含めることができます。その例として、`vhosts.d/vhost.template`を参照してください。設定ファイルを仮想ホスト設定に含めることにより、仮想ホストごとに別のモジュールセットを指定できます。

`default-server.conf`

すべての仮想ホストに対応するグローバル設定で、それぞれ適切なデフォルト値が指定されています。デフォルト値を変更する代わりに、仮想ホスト設定で上書きします。

`errors.conf`

Apacheによるエラーの対処方法を定義します。すべての仮想ホストに対してこれらのメッセージをカスタマイズするには、このファイルを編集します。カスタマイズしない場合は、仮想ホスト設定内のこれらのディレクティブを上書きします。

`httpd.conf`

メインのApacheサーバ設定ファイル。このファイルは変更しません。インクルード文およびグローバル設定が含まれています。ここに記載されている設定ファイルのグローバル設定を上書きします。仮想ホスト設定内のホスト固有の設定(ドキュメントルートなど)を変更します。

`listen.conf`

Apacheを特定のIPアドレスおよびポートにバインドします。名前ベースの仮想ホスティングもこのファイルで設定します。詳細については、29.2.2.1.1項「名前ベースの仮想ホスト」を参照してください。

`magic`

Apacheが自動的に不明なファイルのMIMEタイプを判別できるようにするmime_magicモジュール用のデータ。このファイルは、変更しないでください。

`mime.types`

システムで認識されるMIMEタイプ(実際には`/etc/mime.types`へのリンク)。このファイルは、編集しないでください。このリスト以外にMIMEタイプを追加する必要がある場合は、`mod_mime-defaults.conf`に追加します。

`mod_*.conf`

デフォルトでインストールされるモジュール用の設定ファイル。詳細については、29.4項 「モジュールのインストール、有効化、および設定」を参照してください。オプションのモジュール用の設定ファイルは、`conf.d` ディレクトリ内にあります。

`server-tuning.conf`

各MPMの設定ディレクティブ(29.4.4項 「マルチプロセシングモジュール」を参照)、およびApacheのパフォーマンスを制御する一般的な設定オプションが含まれています。このファイルを変更する場合は、Webサーバを適切にテストしてください。

`ssl-global.conf` and `ssl.*`

グローバルSSL設定およびSSL証明書データ。詳細については、29.6項 「SSLをサポートするセキュアWebサーバのセットアップ」を参照してください。

`sysconfig.d/*.conf`

`/etc/sysconfig/apache2` から自動的に生成される設定ファイル。これらのファイルは、いずれも変更しません。その代わりに、`/etc/sysconfig/apache2` を編集します。このディレクトリに、他の設定ファイルを格納しないでください。

`uid.conf`

Apacheを実行する際に使用するユーザおよびグループIDを指定します。このファイルは、変更しないでください。

`vhosts.d/*.conf`

仮想ホストの設定はこのファイルにあるはずです。このディレクトリには、SSLの有無に関わらず、仮想ホストのテンプレートファイルが格納されます。このディレクトリ内の `.conf` で終わるファイルは、すべて自動的にApache設定に含まれます。詳細については、29.2.2.1項 「仮想ホスト設定」を参照してください。

29.2.2 Apacheを手動で設定する

Apacheを手動設定するには、`root` ユーザとしてプレーンテキストの設定ファイルを編集する必要があります。

29.2.2.1 仮想ホスト設定

仮想ホスト という用語は、同じ物理マシンから複数のURI (universal resource identifiers)のサービスを行えるApacheの機能を指しています。これは、www.example.comとwww.example.netのような複数のドメインを、1台の物理マシン上の単一のWebサーバで保持できることを意味しています。

管理の手間(1つのWebサーバを維持すればよい)とハードウェアの費用(ドメインごとの専用のサーバを必要としない)を省くために仮想ホストを使うことは、よく行われています。仮想ホストは名前ベース、IPベース、またはポートベースのいずれかになります。

すべての既存仮想ホストをリストするには、コマンド `httpd2 -S` を使用します。デフォルトサーバおよびすべての仮想ホストが、それらのIPアドレスおよびリスニングポートとともにリストに表示されます。リストには、各仮想ホストの設定ファイル内での位置を示すエントリも含まれています。

仮想ホストを設定するには、YaSTを使用するか(29.2.3.1.4項「仮想ホスト」で説明)、または設定ファイルを手動で編集します。SUSE Linux Enterprise ServerのApacheは、デフォルトでは、`/etc/apache2/vhosts.d/` の仮想ホストごとに1つの設定ファイルを使用するようになっています。このディレクトリ内で、拡張子が `.conf` のファイルは、すべて自動的に設定に含まれます。仮想ホストの基本的なテンプレートはこのディレクトリ内に用意されています(`vhost.template` 、またはSSLサポートのある仮想ホストの場合は `vhost-ssl.template`)。

ヒント: 常に仮想ホスト設定を作成する

Webサーバに1つのドメインしか存在しない場合でも、常に仮想ホストの設定ファイルを作成することをお勧めします。そうすることによって、ドメイン固有の設定が1つのファイルにまとまるだけでなく、仮想ホストの設定ファイルを移動、削除、または名前変更することによって使用可能な基本設定に常時フォールバックできます。同じ理由で、仮想ホストごとに個別の設定ファイルも作成します。

名前ベースの仮想ホストを使用する際、ドメイン名が仮想ホスト設定と一致しない場合に使用されるデフォルト設定を設定することを推奨します。デフォルト仮想ホストは、その設定が最初にロードされるホストです。設定ファイルの順序は、ファイル名で決定されるので、デフォルト仮想ホスト設定のファイル名は、下線文字(`_`)で始めて(たとえば、 `_default_vhost.conf`)、そのファイルが最初にロードされるようにします。

`<VirtualHost> </VirtualHost>` ブロックには、特定のドメインに適用される情報を記述します。Apacheは、クライアントから定義済みの仮想ホストへの要求を受け取ると、このセクションに記述されているディレクティブを使用します。仮想ホストでは、ほぼすべてのディレクティブを使用できます。Apacheの設定ディレクティブの詳細については、http://httpd.apache.org/docs/2.4/mod/quickreference.html を参照してください。

29.2.2.1.1 名前ベースの仮想ホスト

名前ベースの仮想ホストでは、1つのIPアドレスで複数のWebサイトを運用することができます。Apacheは、クライアントから送られたHTTPヘッダのホストフィールドを使用して、仮想ホスト宣言の1つの、一致する`ServerName`エントリに要求を接続します。一致する`ServerName`が見つからない場合には、指定されている最初の仮想ホストがデフォルトとして用いられます。

最初のステップは、サービスを提供する、名前ベースの異なるホストそれぞれに対して`<VirtualHost>`ブロックを作成することです。各`<VirtualHost>`ブロック内には、少なくとも、サービスの提供対象ホストを指定する`ServerName`ディレクティブと、ファイルシステム内でそのホストのコンテンツが存在する場所を示す`DocumentRoot`ディレクティブが必要です。

例 29.1 名前ベースのVirtualHostエントリの基本例

```
<VirtualHost *:80>
# This first-listed virtual host is also the default for *:80
ServerName www.example.com
ServerAlias example.com
DocumentRoot /srv/www/htdocs/domain
</VirtualHost>

<VirtualHost *:80>
ServerName other.example.com
DocumentRoot /srv/www/htdocs/otherdomain
</VirtualHost>
```

`VirtualHost`開始タグには、名前ベースの仮想ホスト設定の引数としてIPアドレス(または完全修飾ドメイン名)が採用されます。ポート番号ディレクティブはオプションです。

ワイルドカード*をIPアドレスの代わりに使用することもできます。IPv6アドレスを使用する場合には、アドレスを角カッコの中に記述することが必要です。

例 29.2 名前ベースのVirtualHostディレクティブ

```
<VirtualHost 192.168.3.100:80>
  ...
</VirtualHost>

<VirtualHost 192.168.3.100>
  ...
</VirtualHost>

<VirtualHost *:80>
```

```
  ...
</VirtualHost>

<VirtualHost *>
  ...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
  ...
</VirtualHost>
```

29.2.2.1.2 IPベースの仮想ホスト

この仮想ホスト設定では、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。Apacheの1つのインスタンスが、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。

物理サーバは、IPベースの仮想ホストごとに、1つのIPアドレスを持つ必要があります。マシンに複数のネットワークカードがない場合には、仮想ネットワークインタフェース(IPエイリアス)を使用することもできます。

次の例では、IP `192.168.3.100` のマシンでApacheが実行されており、付加的なIP `192.168.3.101` および `192.168.3.102` で2つのドメインをホストしています。すべての仮想サーバについて、`VirtualHost` ブロックが個別に必要です。

例 29.3 IPベースのVirtualHostディレクティブ

```
<VirtualHost 192.168.3.101>
  ...
</VirtualHost>

<VirtualHost 192.168.3.102>
  ...
</VirtualHost>
```

ここでは、`VirtualHost` ディレクティブは、`192.168.3.100` 以外のインタフェースに対してのみ指定されています。`Listen` ディレクティブが `192.168.3.100` に対しても設定される場合、このインタフェースへのHTTP要求に応答するために別のIPベースの仮想ホストを作成する必要があります。作成しない場合、デフォルトのサーバ設定(`/etc/apache2/default-server.conf`)内のディレクティブが適用されます。

29.2.2.1.3 基本的な仮想ホスト設定

仮想ホストをセットアップするには、少なくとも次のディレクティブが各仮想ホスト設定に含まれている必要があります。オプションについては、`/etc/apache2/vhosts.d/vhost.template`を参照してください。

`ServerName`
: ホストに割り当てられている完全修飾ドメイン名。

`DocumentRoot`
: Apacheがこのホストにファイルをサービスする際に使用されるディレクトリパス。セキュリティ上の理由から、ファイルシステム全体へのアクセスはデフォルトで禁じられているため、`Directory`コンテナ内でこのディレクトリを明示的にロック解除する必要があります。

`ServerAdmin`
: サーバ管理者の電子メールアドレス。このアドレスは、Apacheが作成するエラーページなどに表示されます。

`ErrorLog`
: この仮想ホストに関するエラーログファイル。仮想ホストごとに個別のエラーログファイルを作成する必要はありませんが、エラーのデバッグが簡単にできるため、作成されるのが一般的です。`/var/log/apache2/`はApacheのログファイルのデフォルトディレクトリです。

`CustomLog`
: この仮想ホストに関するアクセスログファイル。仮想ホストごとに個別のアクセスログファイルを作成する必要はありませんが、ホストごとのアクセス統計を個別に分析できるため、作成されるのが一般的です。`/var/log/apache2/`はApacheのログファイルのデフォルトディレクトリです。

セキュリティ上の理由から、ファイルシステム全体へのアクセスはデフォルトで禁じられています。したがって、`DocumentRoot`など、Apacheによりサービスされるファイルを保管したディレクトリを明示的にロック解除する必要があります。

```
<Directory "/srv/www/www.example.com/htdocs">
  Require all granted
</Directory>
```

注記

`Require all granted` ステートメントは、旧バージョンのApacheでは

```
Order allow,deny
Allow from all
```

として表われていました。この古い構文は、現在も `mod_access_compat` モジュールでサポートされています。

完全な設定ファイルは次のようになります。

例 29.4 基本的な仮想ホスト設定

```
<VirtualHost 192.168.3.100>
  ServerName www.example.com
  DocumentRoot /srv/www/www.example.com/htdocs
  ServerAdmin webmaster@example.com
  ErrorLog /var/log/apache2/www.example.com_log
  CustomLog /var/log/apache2/www.example.com-access_log common
  <Directory "/srv/www/www.example.com/htdocs">
  Require all granted
  </Directory>
</VirtualHost>
```

29.2.3 ApacheをYaSTで設定する

YaSTを使用してWebサーバを設定するには、YaSTを起動して、[ネットワークサービス] › [HTTPサーバ] の順に選択します。このモジュールを初めて起動するときに、[HTTPサーバウィザード] が起動して、サーバ管理に関していくつかの基本的な事項を決定するように要求されます。このウィザードの完了後、[HTTPサーバ] のモジュールを呼び出すたびに、[HTTPサーバの環境設定] ダイアログが起動します。詳細については、29.2.3.2項「HTTPサーバの設定」を参照してください。

29.2.3.1 HTTP Server Wizard

HTTP Server Wizardには、5つのステップがあります。ダイアログの最後のステップでは、上級者用の設定モードに入って、さらに詳細な設定を行うかどうか選択できます。

29.2.3.1.1 Network Device Selection (ネットワークデバイスの選択)

ここでは、Apacheが着信リクエストをリスンするために使用する、ネットワークインタフェースとポートを指定します。既存のネットワークインタフェースとそれらに対応するIPアドレスから、任意のものを組み合わせて選択できます。他のサービスによって予約されていないものであれば、3つの範囲(ウェルノウンポート、レジスタードポート、ダイナミックまたはプライベートポート)のうちのどのポートでも使用できます。デフォルトの設定では、ポート`80`ですべてのネットワークインタフェース(IPアドレス)をリスンします。

ファイアウォールでWebサーバがリスンするポートを開くには、[ファイアウォールでポートを開く]をクリックします。これは、LAN、WAN、または公共のインターネットなど、ネットワーク上でWebサーバを利用可能にする場合には必須です。外部からのWebサーバへのアクセスが不要なテスト段階でのみ、ポートを閉じておくことは有用です。複数のネットワークインタフェースが存在する場合は、[ファイアウォールの詳細...]をクリックして、ポートを開くインタフェースを指定します。

[次へ] をクリックして設定を続けます。

29.2.3.1.2 モジュール

[モジュール]設定オプションによって、Webサーバでサポートされるスクリプト言語の有効化または無効化を設定できます。他のモジュールの有効化または無効化の詳細については、29.2.3.2.2項「サーバモジュール」を参照してください。[次へ]をクリックして次のダイアログに進みます。

29.2.3.1.3 Default Host (デフォルトのホスト)

このオプションは、デフォルトのWebサーバに関連しています。29.2.2.1項 「仮想ホスト設定」で説明されているように、Apacheは、1つの物理的マシンで複数の仮想ホストに使用することができます。設定ファイルで最初に宣言された仮想ホストは通常、デフォルトのホストと呼ばれます。各仮想ホストは、デフォルトホストの設定を継承します。

ホストの設定(「ディレクティブ」)を編集するには、テーブル内の適切なエントリを選択して、[編集]をクリックします。新しいディレクティブを追加するには、[追加]をクリックします。ディレクティブを削除するには、そのアカウントを選択し、[削除]をクリックします。

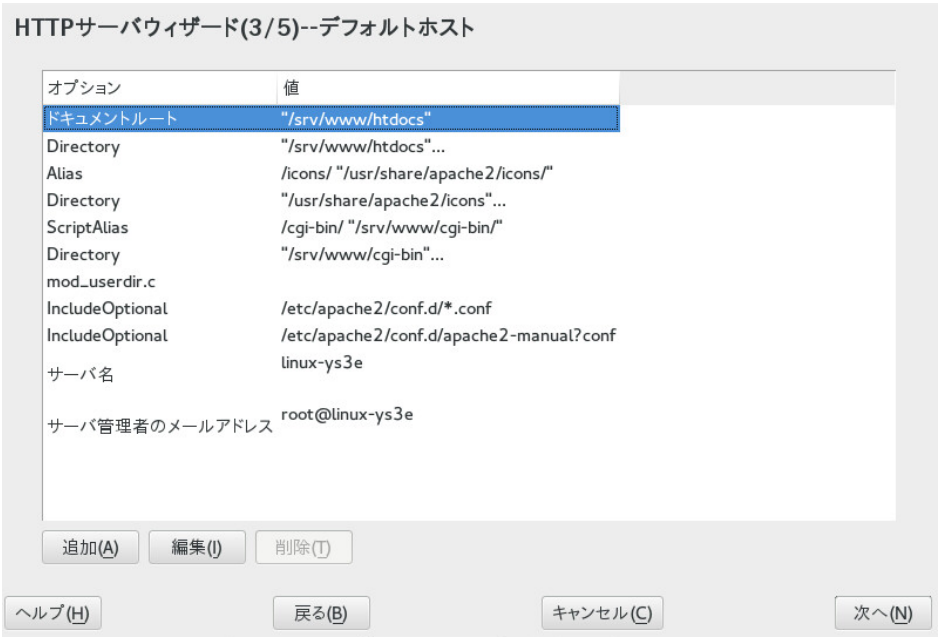


図 29.1 HTTP SERVER WIZARD:デフォルトホスト

これはサーバのデフォルト設定のリストです。

ドキュメントルート

Apacheがこのホストにファイルを送るときに使用されるディレクトリパス。`/srv/www/htdocs`はデフォルトの場所です。

別名

`Alias`ディレクティブを使えば、URLを物理的なファイルシステムの場所にマップすることができます。このことは、パスのURLエイリアスを行えば、ファイルシステムの`Document Root`の外にあるパスでもアクセスできることを意味しています。

デフォルトのSUSE Linux Enterprise Serverでは、`Alias /icons`が`/usr/share/apache2/icons`を指しています。ここには、ディレクトリのインデックスビューで使用されるApacheのアイコンがあります。

ScriptAlias

Alias ディレクティブと同様に、 ScriptAlias ディレクティブはURLをシステム内の場所にマップします。 相違点は、`ScriptAlias`はターゲットディレクトリをCGIの場所として指定するということです。つまり、その場所にあるCGIスクリプトが実行されます。

ディレクトリ

[ディレクトリ]設定を使用して、指定したディレクトリにのみ適用される設定オプションのグループを含めることができます。

`/srv/www/htdocs`、`/usr/share/apache2/icons`、`/srv/www/cgi-bin`ディレクトリのアクセスおよび表示オプションをここで設定します。デフォルトを変更する必要はありません。

対象項目

インクルードにより、他の設定ファイルを指定できます。2つの**インクルード**ディレクティブが設定済みです。`/etc/apache2/conf.d/`は外部モジュールに付属する設定ファイルを保持するディレクトリです。このディレクティブにより、このディレクトリ内の`.conf`で終わるすべてのファイルが対象となります。もう1つのディレクティブでは、`/etc/apache2/conf.d/apache2-manual.conf`という`apache2-manual`設定ファイルが対象となります。

サーバ名

クライアントがWebサーバとコンタクトするために使うデフォルトのURLを指定します。http://**FQDN/にあるWebサーバへの接続用FQDN(完全修飾ドメイン名)か、またはそのIPアドレスを使用します。**ここでは任意の名前は選択できません。サーバはこの名前で「認識」されなければなりません。

`Server Administrator E-Mail`

サーバ管理者の電子メールアドレス。このアドレスは、Apacheが作成するエラーページなどに表示されます。

[デフォルトホスト]のステップを完了したら、[次へ]をクリックして、設定を続けます。

29.2.3.1.4 仮想ホスト

このステップでは、ウィザードはすでに設定されている仮想ホストのリストを表示します(29.2.2.1項「仮想ホスト設定」を参照)。YaST　HTTPウィザードを起動する前に手動で変更を行っていなければ、仮想ホストは表示されません。

ホストを追加するには、[追加]をクリックし、[サーバ名]、[サーバのコンテンツルート](`DocumentRoot`)、[管理者電子メール]などホストに関する基本情報を入力するためのダイアログを開きます。[サーバ解像度]は、ホストの識別方法を決めるために使用されます(名前ベースまたはIPベース)。[仮想ホストIDの変更]で名前またはIPアドレスを指定します。

[次へ]をクリックして、仮想ホスト設定ダイアログの2番目の部分に進みます。

仮想ホスト設定のパート2では、CGIスクリプトを有効にするかどうか、およびこれらのスクリプトを使用するディレクトリを指定できます。また、SSLも有効にできます。SSLを有効化する場合は、証明書のパスも指定する必要があります。SSLおよび証明書の詳細については、29.6.2項　「SSLサポートのあるApacheの設定」を参照してください。[ディレクトリインデックス]オプションを使用して、クライアントがディレクトリを要求するときに表示するファイルを指定できます(デフォルトでは`index.html`)。ファイ

ルを変更する場合は、1つ以上のファイル名(スペースで区切る)を追加します。[公開HTMLを有効にする]で、ユーザのパブリックディレクトリ(`~user/public_html/`)のコンテンツを、サーバの`http://www.example.com/~user`からアクセスできるようにします。

重要: 仮想ホストの作成

仮想ホストを自由に追加することはできません。名前ベースの仮想ホストを使用する場合は、各ホスト名がネットワーク内で解決されている必要があります。IPベースの仮想ホストを使用する場合は、使用可能な各IPアドレスに対し1つのホストのみを割り当てることができます。

29.2.3.1.5 概要

これはウィザードの最後のステップです。ここでは、Apacheサーバをいつ、どのようにして起動するか(ブート時に起動するか、手動で起動するか)を指定します。また、ここまで行った設定の簡単な要約を確認します。この設定でよければ、[完了]をクリックして、設定を完了します。変更する場合は、希望のダイアログまで[戻る]をクリックして戻ります。[HTTPサーバのエキスパート環境設定]をクリックして、29.2.3.2項「HTTPサーバの設定」で説明しているダイアログを開きます。

図 29.2 HTTP SERVER WIZARD:概要

29.2.3.2 HTTPサーバの設定

[HTTPサーバの設定]ダイアログでは、ウィザード(Webサーバを最初に設定する場合にのみ実行)よりも詳細に設定を調整できます。このダイアログは、次で説明する4つのタブで構成されています。ここで変更する設定オプションは、すぐには適用されません。変更を適用するには、常に[完了]をクリックして変更を確認する必要があります。[中止]をクリックすると、設定モジュールを終了し、変更が破棄されます。

29.2.3.2.1 待ち受けポートおよびアドレス

[HTTPサービス]で、Apacheを実行するか([有効にする])、または停止するか([無効])を選択します。[Listen on Ports]で、サーバが使用可能なアドレスおよびポートについて[追加]、[編集]、または[削除]を選択します。デフォルトでは、ポート`80`ですべてのインタフェースをリスンします。常に[ファイアウォールでポートを開く]にチェックマークを入れておく必要があります。そうしないと、外部からWebサーバにアクセスできなくなります。外部からのWebサーバへのアクセスが不要なテスト段階でのみ、ポートを閉じておくことは有用です。複数のネットワークインタフェースが存在する場合は、[ファイアウォールの詳細...]をクリックして、ポートを開くインタフェースを指定します。

[ログファイル]で、アクセスログファイルまたはエラーログファイルのいずれかを確認します。これは、設定をテストする場合に便利です。ログファイルは別個のウィンドウに表示されますが、そこから、Webサーバを再起動または再ロードすることも可能です。詳細については、29.3項「Apacheの起動および停止」を参照してください。これらのコマンドはすぐに有効になり、ログメッセージもすぐに表示されます。

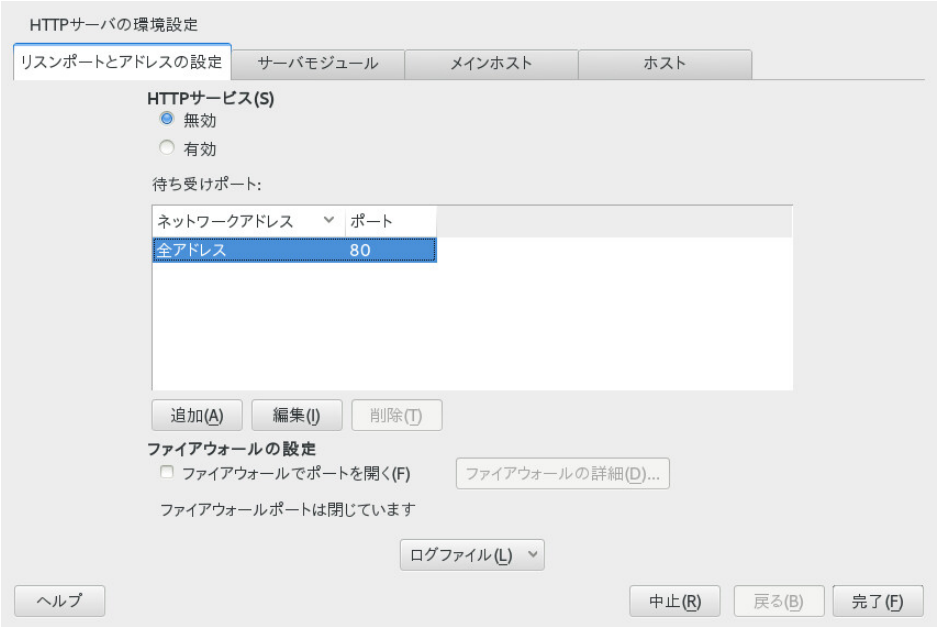


図 29.3 HTTP SERVER CONFIGURATION:設定:リッスンポートとアドレス

29.2.3.2.2 サーバモジュール

[状態の変更]をクリックして、Apache2モジュールのステータス(有効または無効)を変更できます。すでにインストールされているがリストに含まれていない新規モジュールを追加するには、[Add Module]をクリックします。モジュールの詳細については、29.4項 「モジュールのインストール、有効化、および設定」を参照してください。

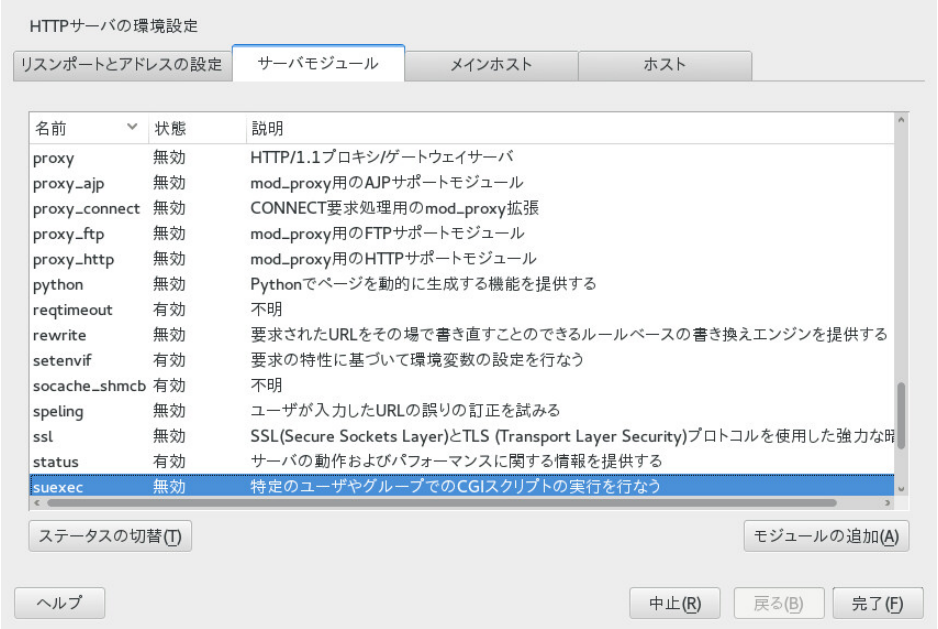


図 29.4 HTTP SERVER CONFIGURATION:サーバモジュール

29.2.3.2.3 メインホストまたはホスト

これらのダイアログは、すでに説明したものと同じです。詳細については、29.2.3.1.3項 「Default Host (デフォルトのホスト)」および29.2.3.1.4項 「仮想ホスト」を参照してください。

29.3 Apacheの起動および停止

29.2.3項 「ApacheをYaSTで設定する」の説明のようにYaSTを設定すると、Apacheは、ブート時に`multi-user.target`および`graphical.target`で起動されます。YaSTの[サービスマネージャ]、あるいは`systemctl`コマンドラインツール(`systemctl enable`または`systemctl disable`)を使用して、この動作を変更できます。

稼働中のシステムでApacheを起動、停止、または操作するには、次の説明に従って`systemctl`または`apachectl`コマンドを使用します。

`systemctl`コマンドの一般的な情報については、10.2.1項　「稼働中のシステムでのサービスの管理」を参照してください。

`systemctl status apache2.service`

Apacheが起動したかどうかをチェックします。

`systemctl start apache2.service`

Apacheが実行中でない場合に起動します。

`systemctl stop apache2.service`

親プロセスを終了して、Apacheを終了します。

`systemctl restart apache2.service`

Apacheをいったん停止し、再起動します。Apacheが実行中でなかった場合は、新規に起動します。

`systemctl try-restart apache2.service`

Apacheがすでに実行中の場合にのみ、停止して再起動します。

`systemctl reload apache2.service`

フォークしたすべてのApacheプロセスに、シャットダウンする前に要求を完了させて、それからWebサーバを停止します。1つのプロセスが終了するたびに、新たに開始したプロセスで置き換えられるので、最終的にはApacheの完全な「再起動」になります。

`apachectl -k graceful`

すべての着信要求をただちに処理する2つ目のウェブサーバを起動します。ウェブサーバの以前のインスタンスは`GracefulShutdownTimeout`で設定された一定時間、引き続きすべの既存要求を処理します。

このコマンドは、新しいバージョンへのアップグレード時、または再起動が必要な設定オプションの変更時に便利です。このオプションを使用すると、サーバのダウンタイムが最小限になります。`GracefulShutdownTimeout`の設定が必要です。これを設定しないと、このコマンドを実行しても通常どおり再起動されます。ゼロに設定した場合、残っている要求がすべて完全に処理されるまで、サーバが無制限に待機します。

最初のApacheインスタンスが必要なリソースをすべてクリアできなかった場合、graceful restartは失敗します。この場合、コマンドの結果はgraceful stopとなります。

`systemctl stop apache2.service`

既存の要求を完了できるように、`GracefulShutdownTimeout`で設定された一定時間の経過後にWebサーバを停止します。

`GracefulShutdownTimeout`の設定が必要です。これを設定しないと、このコマンドを実行しても通常どおり再起動されます。ゼロに設定した場合、残っている要求がすべて完全に処理されるまで、サーバが無制限に待機します。

`apachectl configtest`

実行中のWebサーバに影響することなく、設定ファイルの構文をチェックします。　このチェックは、サーバが起動、再ロードまたは再起動するたびに行われるため、通常は明示的にテストを実行する必要はありません(設定エラーが検出された場合、Webサーバは起動、再ロードまたは再起動されません)。

ヒント: その他のフラグ

コマンドにその他のフラグを指定した場合、これらはWebサーバに渡されます。

29.4　モジュールのインストール、有効化、および設定

Apacheソフトウェアは、モジュール形式で構築されており、一部の主要タスクを除いてはモジュールごとに処理されます。この方法で、HTTPさえもモジュールによって処理されています(http_core)。

Apacheのモジュールは、ビルド時にApacheのバイナリに組み込むことも、実行時に動的にロードすることもできます。動的なモジュールのロード方法の詳細については、29.4.2項 「有効化と無効化」を参照してください。

Apacheモジュールは、次の4つのカテゴリに分類されます。

基本モジュール

基本モジュールは、デフォルトでApacheにコンパイルされています。SUSE Linux Enterprise ServerのApacheでは、`mod_so`　(他のモジュールのロードに必要)および`http_core`のみがコンパイルされています。他のモジュールは、サーバのバイナリに入れる代わりに、ランタイム時に入れるように共有オブジェクトとして利用できます。

拡張モジュール

一般に、拡張とされているモジュールは、Apache　ソフトウェアパッケージに含まれてはいますが、通常、サーバに静的にはコンパイルされていません。SUSE Linux Enterprise Serverでは、これらはApacheにランタイムでロードすることができる共有オブジェクトとして利用可能になっています。

外部モジュール

外部とラベルされているモジュールは、公式のApacheのディストリビューションには含まれていません。ただし、SUSE Linux Enterprise Serverはそれらのいくつかを提供しています。

MPM(マルチプロセシングモジュール)

MPMは、Webサーバへのリクエストを受け取って処理する役割を果たすもので、Webサーバソフトウェアの中核となっています。

29.4.1 モジュールのインストール

29.1.2項 「インストール」で説明されているデフォルトインストールを行った場合は、すべての基本モジュールと拡張モジュール、マルチプロセシングモジュール、プリフォークMPM、および外部モジュールの `mod_php5` と `mod_python` がすでにインストールされています。

YaSTを起動し、[ソフトウェア] › [ソフトウェア管理] の順に選択して、その他の外部モジュールをインストールできます。[フィルタ] › [検索] の順に選択し、[apache] を検索します。他のパッケージの中で、使用可能な外部Apacheモジュールがすべて検索結果のリストに表示されます。

29.4.2 有効化と無効化

特定モジュールの有効化/無効化は、手動で行うか、YaSTを使用します。YaSTでは、29.2.3.1項 「HTTP Server Wizard」で説明されているモジュール設定を使用して、スクリプト言語モジュール(PHP5、Perl、およびPython)を有効または無効にする必要があります。その他のすべてのモジュールは、29.2.3.2.2項 「サーバモジュール」で説明しているように有効化または無効化できます。

手動でモジュールを有効化または無効化する場合は、`a2enmod mod_foo` または `a2dismod mod_foo` コマンドをそれぞれ使用します。`a2enmod -l` は、すべての現在アクティブなモジュールのリストを出力します。

重要: 外部モジュール用の設定ファイルを含める

手動で外部モジュールを有効化した場合は、各設定ファイルがすべての仮想ホスト設定にロードされていることを確認します。外部モジュール用の設定ファイルは、`/etc/apache2/conf.d/` 内に位置し、デフォルトではロードされません。各仮想ホスト上に同じモジュールが必要な場合は、このディレクトリ内の `*.conf` を含めることができます。必要でない場合は、個々のファイルを含めます。その例として、「`/etc/apache2/vhost.d/vhost.template`」を参照してください。

29.4.3 基本および拡張モジュール

すべての基本および拡張モジュールは、Apacheのマニュアルに詳しく説明されています。ここでは、主要なモジュールについて簡単に説明します。各モジュールの詳細については、http://httpd.apache.org/docs/2.4/mod/ を参照してください。

`mod_actions`

特定のMIMEタイプ(`application/pdf` など)、特定の拡張子を持つファイル(`.rpm` など)、または特定の要求方法(`GET` など)が要求された場合に、常にスクリプトを実行する方法を提供します。このモジュールは、デフォルトで有効です。

`mod_alias`

`Alias` および `Redirect` ディレクティブを提供します。これにより、特定のディレクトリにURIをマップ(`Alias`)、または要求されたURLを別の場所にリダイレクトできます。このモジュールは、デフォルトで有効です。

`mod_auth*`

認証モジュールは、`mod_auth_basic` による基本認証や `mod_auth_digest` によるダイジェスト認証など、さまざまな認証方法を提供します。

`mod_auth_basic` および `mod_auth_digest` は、認証プロバイダモジュールの `mod_authn_*` (たとえば、テキストファイルベースの認証用の `mod_authn_file`)および認証モジュールの `mod_authz_*` (たとえば、ユーザ認証用の `mod_authz_user`)と組み合わせる必要があります。

この項目の詳細は、http://httpd.apache.org/docs/2.4/howto/auth.html の「Authentication HOWTO」で説明されています。

`mod_autoindex`

Autoindexは、インデックスファイル(`index.html` など)が存在しない場合にディレクトリリストを生成します。これらのインデックスのルックアンドフィールは設定可能です。このモジュールは、デフォルトで有効です。ただし、ディレクトリリストは、デフォルトで `Options` ディレクティブを経由して無効化されています。仮想ホスト設定でこの設定を上書きします。このモジュール用のデフォルト設定は、`/etc/apache2/mod_autoindex-defaults.conf` に存在します。

`mod_cgi`

`mod_cgiは、CGIスクリプトを実行するのに必要です。` このモジュールは、デフォルトで有効です。

`mod_deflate`

このモジュールを使用して、配信前にファイルタイプを圧縮するようにApacheを設定できます。

`mod_dir`

mod_dirは、`DirectoryIndex`ディレクティブを提供します。これを使用して、ディレクトリが要求されたときに(デフォルトでは`index.html`)自動的に配信されるファイルを設定できます。ディレクトリ要求に末尾のスラッシュが含まれていない場合は、正しいURLへの自動リダイレクトも提供します。このモジュールは、デフォルトで有効です。

`mod_env`

CGIスクリプトやSSIページに渡す環境を制御します。環境変数を設定、設定解除したり、httpdプロセスを起動したシェルから渡すことができます。このモジュールは、デフォルトで有効です。

`mod_expires`

`mod_expires`を使用すると、`Expires`ヘッダの送信によって、プロキシとブラウザのキャッシュがドキュメントを更新する頻度を制御できます。このモジュールは、デフォルトで有効です。

`mod_include`

`mod_includeは、動的にHTMLページを生成するための基本機能を提供するSSI (Server-Side Includes)を使用できるようにします。` このモジュールは、デフォルトで有効です。

`mod_info`

http://localhost/server-info/にサーバ設定の包括的な概要を表示します。セキュリティ上の理由から、このURLへのアクセスは常に制限されます。デフォルトでは、`localhost`にのみ、このURLへのアクセスが許可されます。`mod_info`は、`/etc/apache2/mod_info.conf`で設定されます。

`mod_log_config`

このモジュールを使用して、Apacheログファイルの書式を設定できます。このモジュールは、デフォルトで有効です。

`mod_mime`

mimeモジュールは、ファイル名の拡張子(HTMLドキュメント用の`text/html`など)に基づいた、適切なMIMEヘッダを使用してファイルが配信されるようにします。このモジュールは、デフォルトで有効です。

`mod_negotiation`

コンテンツネゴシエーションに必要です。詳細については、http://httpd.apache.org/docs/2.4/content-negotiation.html を参照してください。このモジュールは、デフォルトで有効です。

`mod_rewrite`

`mod_alias`の機能を提供しますが、それ以外の機能と柔軟性も提供します。`mod_rewrite`を使用すると、複数の規則、要求ヘッダなどに基づいてURLをリダイレクトできます。

`mod_setenvif`

クライアントから送信されたブラウザ文字列やIPアドレスなどの、クライアントからのリクエスト詳細に基づいて環境変数を設定します。このモジュールは、デフォルトで有効です。

`mod_spelling`

`mod_spelling`は、大文字小文字の違いなど、URLの表記エラーの訂正を自動的に試みます。

`mod_ssl`

Webサーバとクライアント間の暗号化接続を有効化します。詳細については、29.6項 「SSLをサポートするセキュアWebサーバのセットアップ」を参照してください。このモジュールは、デフォルトで有効です。

`mod_status`

サーバの動作およびパフォーマンスに関する情報をhttp://localhost/server-status/に表示します。セキュリティ上の理由から、このURLへのアクセスは常に制限する必要があります。デフォルトでは、`localhost`にのみ、このURLへのアクセスが許可されます。`mod_status`は、`/etc/apache2/mod_status.conf`で設定されます。

`mod_suexec`

`mod_suexecは、CGIスクリプトを別のユーザとグループで実行できるようにします。`このモジュールは、デフォルトで有効です。

`mod_userdir`

`~user/`の下に、ユーザ固有のディレクトリを用意します。`UserDir`ディレクティブを設定で指定する必要があります。このモジュールは、デフォルトで有効です。

29.4.4 マルチプロセシングモジュール

SUSE Linux Enterprise Serverには、Apacheで使用するための2つの異なるマルチプロセッシングモジュール(MPM)が用意されています。

- プリフォークMPM
- 29.4.4.2項「ワーカーMPM」

29.4.4.1 プリフォークMPM

プリフォークMPMは、スレッド対応でない、プリフォークWebサーバを実装します。プリフォークMPMは、各要求を分離し、個々の子プロセスの分岐で処理するApacheバージョン 1.xと同じように、このWebサーバを動作させます。これにより、問題のあるリクエストが他のものに影響することがなくなるので、Webサーバのロックアップを避けられます。

プロセスベースのアプローチによって安定性がもたらされますが、プリフォークMPMは、もう一方のワーカーMPMよりも多くのシステムリソースを消費します。プリフォークMPMは、Unixベースのオペレーティングシステムでのデフォルトの MPMとみなされています。

重要: このドキュメントでのMPM

このドキュメントでは、Apacheがプリフォーク MPMで使用されていることを仮定しています。

29.4.4.2 ワーカーMPM

ワーカーMPMは、マルチスレッド対応のWebサーバを提供します。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。ワーカーMPMは、子プロセスを分岐する代わりに、サーバプロセスでスレッドを使用することによってリクエストを処理します。プリフォークした子プロセスはマルチスレッドになります。このアプローチでは、プリフォークMPMの場合よりもシステムリソースの消費が少なくなるので、Apacheの性能が良くなります。

主な欠点としては、ワーカーMPMの安定性の問題が挙げられます。スレッドが壊れた場合、プロセスのすべてのスレッドに影響してしまいます。最悪の場合には、サーバがクラッシュすることがあります。特に、ApacheでCGI (Common Gateway Interface)を使用している場合、負荷が大きくなると、スレッドがシステムリソースと通信できなくなり、内部サーバエラーが生じることがあります。ワーカーMPMを使用すべきでないという意見の別の根拠は、利用できるApacheのモジュールのすべてがスレッドセーフになっているわけではなく、そのためワーカーMPMと組み合わせて使用することはできないという点です。

警告: MPMと組み合わせてPHPモジュールを使用する

利用可能なPHPモジュールのすべてがスレッドセーフになっているわけではありません。ワーカーMPMと `mod_php` は併用しないでください。

29.4.5 外部モジュール

ここでは、SUSE Linux Enterprise Serverに付属しているすべての外部モジュールを記載しています。モジュールのドキュメントは、記載のディレクトリ内に存在します。

mod-apparmor

`mod_php5`や`mod_perl`などのモジュールが処理する個々のCGIスクリプトに対して、AppArmor制限を提供するために、Apacheにサポートを追加します。

パッケージ名: `apache2-mod_apparmor`
詳細: Book “Security Guide”

`mod_perl`

`mod_perlは、埋め込まれているインタプリタでPerlスクリプトを実行できるようにします。`サーバに埋め込まれている永続的なインタプリタにより、外部インタプリタの起動のオーバーヘッド、およびPerlの起動時間のペナルティを回避できます。

パッケージ名: `apache2-mod_perl`
環境設定ファイル: `/etc/apache2/conf.d/mod_perl.conf`
詳細: `/usr/share/doc/packages/apache2-mod_perl`

`mod_php5`

PHPは、サーバ側クロスプラットフォームのHTML埋込みスクリプト言語です。

パッケージ名: `apache2-mod_php5`
環境設定ファイル: `/etc/apache2/conf.d/php5.conf`
詳細: `/usr/share/doc/packages/apache2-mod_php5`

`mod_python`

`mod_pythonは、Apache HTTPサーバへのPythonの埋込みができるようにし、Webベースのアプリケーションの設計で、さらに柔軟性を持たせ、パフォーマンスを向上させます。`

パッケージ名: `apache2-mod_python`
詳細: `/usr/share/doc/packages/apache2-mod_python`

`mod_security`

`mod_security`により、さまざまな範囲の攻撃からWebアプリケーションを保護するためのファイアウォールがWebアプリケーションに提供されます。さらに、HTTPトラフィックモニタリングおよびリアルタイム分析も可能です。

パッケージ名: `apache2-mod_security2`

環境設定ファイル: `/etc/apache2/conf.d/mod_security2.conf`
詳細: `/usr/share/doc/packages/apache2-mod_security2`
マニュアル: http://modsecurity.org/documentation/

29.4.6 コンパイル

上級ユーザは、カスタムのモジュールを記述してApacheを拡張することができます。Apache用のモジュールを開発したり、サードパーティのモジュールをコンパイルしたりするには、`apache2-devel` パッケージ、および対応する開発ツールが必要です。`apache2-devel` には、Apache用の追加モジュールのコンパイルに必要な `apxs2` ツールも含まれています。

`apxs2` は、ソースコードからモジュールをコンパイルし、インストールすることを可能にします(設定ファイルへの必要な変更も含みます)。これは、実行時にApacheにロードされる、ダイナミック共有オブジェクト (DSO)を作成します。

`apxs2` バイナリは、`/usr/sbin` の下層にあります

- `/usr/sbin/apxs2` —MPMと共に動作する拡張モジュールを構築するのに適しています。インストール場所は `/usr/lib/apache2` です。
- `/usr/sbin/apxs2-prefork` —プリフォークMPMモジュールに適しています。インストール場所は `/usr/lib/apache2-prefork` です。
- `/usr/sbin/apxs2-worker` —ワーカーMPMモジュールに適しています。インストール場所は `/usr/lib/apache2-worker` です。

次のコマンドで、ソースコードからモジュールをインストールして、アクティブにします。

```
cd /path/to/module/source; apxs2 -cia
    mod_foo.c
```

ここで、`-c` はモジュールをコンパイルし、`-i` はモジュールをインストールし、`-a` はモジュールをアクティブにします。`apxs2` のその他のオプションについては、`apxs2(1)` manページを参照してください。

29.5 CGIスクリプトの実行

ApacheのCGI (Common Gateway Interface)により、通常CGIスクリプトとして呼ばれるスクリプトまたはプログラムを含んだ動的コンテンツを作成できます。CGIスクリプトは、どのプログラム言語でも作成できます。通常、PerlまたはPHPなどのスクリプト言語が使用されます。

ApacheがCGIスクリプトで作成されたコンテンツを配信できるようにするには、`mod_cgi`を有効にする必要があります。`mod_alias`も必要です。デフォルトでは、両モジュールとも有効化されています。モジュールの有効化の詳細については、29.4.2項「有効化と無効化」を参照してください。

警告: CGIセキュリティ

サーバがCGIスクリプトを実行できるようになると、潜在的なセキュリティホールが発生します。詳細については、29.7項「セキュリティ問題の回避」を参照してください。

29.5.1 Apacheの設定

SUSE Linux Enterprise Serverでは、CGIスクリプトの実行は、`/srv/www/cgi-bin/`ディレクトリ内でのみ許可されています。この場所は、すでにCGIスクリプトを実行するように設定されています。仮想ホスト設定を作成しておらず(29.2.2.1項「仮想ホスト設定」を参照してください)、ホスト固有のディレクトリにスクリプトを配置する場合は、このディレクトリのロックを解除し、設定する必要があります。

例 29.5 VIRTUALHOST CGIの設定

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/" ❶

<Directory "/srv/www/www.example.com/cgi-bin/">
 Options +ExecCGI ❷
 AddHandler cgi-script .cgi .pl ❸
 Require all granted ❹
</Directory>
```

❶ このディレクトリ内のすべてのファイルをCGIスクリプトとして処理するようにApacheに指示します。

❷ CGIスクリプトの実行を有効化します。

❸ .plおよび.cgiの拡張子が付いたファイルをCGIスクリプトとして処理するようにサーバに指示します。必要に応じて調整します。

❹ `Require`ディレクティブは、デフォルトのアクセス状態を制御します。この場合、指定したディレクトリへのアクセスが無制限に許可されます。認証および権限の詳細については、http://httpd.apache.org/docs/2.4/howto/auth.html を参照してください。

29.5.2 テストスクリプトの実行

CGIプログラミングは通常のプログラミングとは異なり、CGIプログラムとスクリプトの前に`Content-type: text/html`などのMIMEタイプヘッダを記述する必要があります。このヘッダはクライアントに送信されるので、クライアントは、受信したコンテンツによってコンテンツの種類を識別します。次に、このスクリプトの出力は、クライアント(通常はWebブラウザ)が認識できる形式(たいていの場合はHTML、プレーンテキスト、画像など)でなければなりません。

Apacheパッケージの一部として、`/usr/share/doc/packages/apache2/test-cgi`内に簡単なテストスクリプトが含まれています。このスクリプトは、いくつかの環境変数の内容をプレーンテキストとして出力します。このスクリプトを`/srv/www/cgi-bin/`か、仮想ホストのスクリプトディレクトリ`/srv/www/www.example.com/cgi-bin/`のいずれかにコピーし、「`test.cgi`」という名前を付けます。

Webサーバがアクセスできるファイルは、`root`ユーザが所有している必要があります。詳細については、29.7項 「セキュリティ問題の回避」を参照してください。Webサーバは別のユーザ名で実行しているので、CGIスクリプトはworld-executableおよびworld-readableである必要があります。CGIディレクトリに移動し、`chmod 755 test.cgi`コマンドを使用して適切なパーミッションを適用します。

次に、`http://localhost/cgi-bin/test.cgi`または`http://www.example.com/cgi-bin/test.cgi`を呼び出します。「CGI/1.0 test script report」を参照してください。

29.5.3 CGIトラブルシューティング

テストプログラムの出力の代わりにエラーメッセージが表示される場合は、次を確認します。

CGIトラブルシューティング

- カスタムCGIディレクトリを設定した場合、適切に設定されていますか?不明な場合は、デフォルトのCGIディレクトリの`/srv/www/cgi-bin/`内にあるスクリプトを実行し、`http://localhost/cgi-bin/test.cgi`を呼び出します。
- ファイルのパーミッションは正しいですか?CGIディレクトリに移動して、`ls -l test.cgi`を実行します。その出力が次で始まっているかどうかを確認します。

```
-rwxr-xr-x  1 root root
```

- そのスクリプトにプログラミングエラーがないかどうか確認します。`test.cgi`を変更しなかった場合は該当しませんが、独自のプログラムを使用する場合は、必ず、プログラミングエラーがないかどうか確認してください。

29.6 SSLをサポートするセキュアWebサーバのセットアップ

クレジットカード情報などの機密データをWebサーバやクライアント間で送信する場合は必ず、認証を使用して、安全で、暗号化された接続の確立を推奨します。**mod_sslは、クライアントとWebサーバ間のHTTP通信にセキュアソケットレイヤ(SSL)プロトコルとトランスポートレイヤセキュリティ(TLS)プロトコルを使用して、強力な暗号化を行います。** SSL/TSLを使用することにより、Webサーバとクライアント間でプライベートな接続が確立されます。データの整合性が保証され、クライアントとサーバ間で相互認証ができるようになります。

この目的で、サーバは、URLに対するリクエストに応答する前に、サーバの有効な識別情報を含むSSL証明書を送ります。これにより、サーバが唯一の正当な通信相手であることが保証されます。加えて、この証明書は、クライアントとサーバの間の暗号化された通信が、重要な内容がプレーンテキストとして見られる危険なしに、情報を転送できることを保証します。

mod_sslは、SSL/TSLプロトコル自体は実装しませんが、ApacheとSSLライブラリ間のインタフェースとして機能します。 SUSE Linux Enterprise Serverでは、OpenSSLライブラリが使用されます。OpenSSLは、Apacheとともに自動的にインストールされます。

Apacheで`mod_ssl`を使用した場合の最も明白な効果は、URLのプレフィックスが`http://`ではなく`https://`となることです。

29.6.1 SSL証明書の作成

SSL/TSLをWebサーバで使用するには、SSL証明書を作成する必要があります。この証明書は、両者が互いに相手を識別できるように、Webサーバとクライアント間の認証に必要です。証明書の整合性を確認するには、すべてのユーザが信用する者によって署名される必要があります。

3種類の証明書を作成することができます。テストの目的のみの「ダミー証明書」、あらかじめ定義されている信用する一部のユーザグループ用の自己署名付き証明書、および公的な独立団体のCA (Certificate Authority)によって署名される証明書です。

証明書の作成は、2つのステップで行うことができます。はじめに、CAの秘密鍵が生成され、次に、この鍵を使用してサーバ証明書が署名されます。

ヒント: 詳細情報

SSL/TSLの概念および定義の詳細については、http://httpd.apache.org/docs/2.4/ssl/ssl_intro.html を参照してください。

29.6.1.1 「ダミー」証明書の作成

ダミー証明書を生成するには、スクリプト `/usr/bin/gensslcert` を呼び出します。次のファイルを作成または上書きします。`gensslcert` のオプションのスイッチを使用して、証明書を微調整します。詳細は、`/usr/bin/gensslcert -h` を呼び出してください。

- `/etc/apache2/ssl.crt/ca.crt`
- `/etc/apache2/ssl.crt/server.crt`
- `/etc/apache2/ssl.key/server.key`
- `/etc/apache2/ssl.csr/server.csr`

`ca.crt` のコピーは、ダウンロード用に `/srv/www/htdocs/CA.crt` にも配置されます。

重要: テスト専用

ダミー証明書は、実働システム上では使用しないでください。テストの目的のみで使用してください。

29.6.1.2 自己署名付き証明書の作成

イントラネットまたは定義されている一部のユーザグループ用にセキュアWebサーバをセットアップするときは、多くの場合、独自の認証局(CA)を通じて証明書に署名すれば十分です。Webブラウザは自己署名付き証明書を認識できないため、このようなサイトの訪問者には「これは信頼できないサイトです」という警告が表示されます。

重要: 自己署名付き証明書

自己署名付き証明書は、CA (Certificate Authority)として認識および信用するユーザによってアクセスされるWebサーバ上でのみ使用します。自己署名付き証明書をパブリックショップなどで使用することはお勧めしません。

まず、証明書署名要求(CSR)を生成する必要があります。`openssl` と、証明書の書式として `PEM` を使用します。このステップでは、パスフレーズを入力し、いくつかの質問に回答するよう求められます。入力したパスフレーズは後で必要になるため、覚えておいてください。

```
sudo openssl req -new > new.cert.csr
Generating a 1024 bit RSA private key
..++++++
```

```
.........++++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:❶
Verifying - Enter PEM pass phrase:❷
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:❸
State or Province Name (full name) [Some-State]:❹
Locality Name (eg, city) []:❺
Organization Name (eg, company) [Internet Widgits Pty Ltd]:❻
Organizational Unit Name (eg, section) []:❼
Common Name (e.g. server FQDN or YOUR name) []:❽
Email Address []:❾

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:❿
An optional company name []:⓫
```

❶ パスフレーズを入力し、

❷ ...もう一度入力します(パスフレーズを覚えてください)。

❸ 2文字の国コードを入力します(`GB`や`CZ`など)。

❹ 住所のある都道府県の名前を入力します。

❺ 都市名を入力します(`Prague`など)。

❻ 勤務先の組織の名前を入力します。

❼ 組織部門を入力します。組織部門がない場合は空白のままにします。

❽ サーバのドメイン名または自分の氏名を入力します。

❾ 勤務先の電子メールアドレスを入力します。

❿ チャレンジパスワードは空白のままにします。入力した場合は、Apache　Webサーバを再起動するたびにチャレンジパスワードを入力する必要があります。

11 オプションの会社名を入力するか、空白のままにします。

これで証明書を生成できます。もう一度 `openssl` を使用します。証明書の形式はデフォルトの `PEM` です。

手順 29.3 証明書を生成する

1. 鍵の秘密部分を `new.cert.key` にエクスポートします。証明書署名要求(CSR)の作成時に入力したパスフレーズを入力するようプロンプトが表示されます。

```
sudo openssl rsa -in privkey.pem -out new.cert.key
```

2. 署名要求に入力した情報に従って、証明書の公開部分を生成します。`-days` オプションで、証明書が期限切れになるまでの期間を指定します。証明書を取り消すことも、期限切れになる前に置き換えることもできます。

```
sudo openssl x509 -in new.cert.csr -out new.cert.cert -req \
-signkey new.cert.key -days 365
```

3. 関連するディレクトリに証明書ファイルをコピーし、Apacheサーバが読み込めるようにします。秘密鍵 `/etc/apache2/ssl.key/server.key` を全ユーザに対して読み込み可能にせずに、公開PEM証明書を `/etc/apache2/ssl.crt/server.crt` 全ユーザに対して読み込み可能にします。

```
sudo cp new.cert.cert /etc/apache2/ssl.crt/server.crt
sudo cp new.cert.key /etc/apache2/ssl.key/server.key
```

ヒント

最後のステップとして、Webブラウザ内の認識および信用されたCAのリストに組み込めるように、`/etc/apache2/ssl.crt/server.crt` からユーザがアクセスできる場所に公開証明書ファイルをコピーします。コピーしない場合、ブラウザは、この証明書が不明な認証局から発行されたものであると見なします。

29.6.1.3 公式に署名された証明書の取得

証明書に署名する公式なCA (Certificate Authority)は、多数存在します。証明書は、信用のあるサードパーティによって署名されるため、完全に信用できます。通常、一般に運営されているセキュアWebサーバでは、証明書が公式に署名されます。

最も良く知られている公式なCAには、Thawte (http://www.thawte.com/)またはVerisign (http://www.verisign.com)があります。これらや、その他のCAは、すべてのブラウザにすでにコンパイルされているため、これらのCAによって署名された証明書は、ブラウザによって自動的に許可されます。

公式に署名された証明書を要求するとき、CAに証明書を送信しません。代わりに、CSR (Certificate Signing Request)を発行します。CSRを作成するには、`/usr/share/ssl/misc/CA.sh -newreq`スクリプトを呼び出します。

はじめに、スクリプトは、CSRの暗号化に使用されているパスワードを問い合わせてきます。その後、識別名を入力するよう求められます。このとき、国名または組織名など、いくつかの質問に答える必要があります。ここで入力した内容が証明書に含まれ、確認されるため、有効なデータを入力します。すべての質問に答える必要はありません。該当しない、または空白のままにする場合は、「.」を使用します。一般名は、CA自体の名前です。`My company` CAなど、意味のある名前を選択します。最後に、チャレンジパスワードおよび代替の企業名を入力する必要があります。

スクリプトを呼び出したディレクトリでCSRを検索します。ファイルには、`newreq.pem`という名前が付きます。

29.6.2 SSLサポートのあるApacheの設定

Webサーバ側のSSLとTLS要求用のデフォルトのポートは 443です。ポート 80をリスンする「通常」のApacheと、ポート 443をリスンするSSL/TL対応のApacheとの間に競合は生じません。通常、ポート80とポート443への要求はそれぞれ別の仮想ホストが処理し、別の仮想サーバに送られます。

重要: ファイアウォール設定

ポート443でSSL対応のApache用のファイアウォールを開くことを忘れないでください。ファイアウォールは、Book “Security Guide” 15 “Masquerading and Firewalls” 15.4.1 “Configuring the Firewall with YaST”で説明されているように、YaSTを使用して設定できます。

SSLモジュールはグローバルサーバ設定でデフォルトで有効になっています。ホストで無効にされている場合は、コマンド`a2enmod ssl`で有効にします。最終的にSSLを有効にするには、サーバをフラグ「SSL」で起動する必要があります。このためには、`a2enflag SSL`を呼び出します。サーバ証明書をパスワードで暗号化している場合は、`/etc/sysconfig/apache2`で`APACHE_TIMEOUT`の値を増やし、Apacheの起動時にパスフレーズを入力するのに十分な時間が与えられるようにします。これらの変更を適用するため、サーバを再起動します。再ロードでは不十分です。

仮想ホスト設定ディレクトリには、SSL固有ディレクティブが詳細に記述されている `/etc/apache2/vhosts.d/vhost-ssl.template` テンプレートが含まれています。一般的な仮想ホスト設定については、29.2.2.1項「仮想ホスト設定」を参照してください。

始めるには、テンプレートを `/etc/apache2/vhosts.d/mySSL-host.conf` にコピーして編集します。次のディレクティブの値を調整するだけです。

- `DocumentRoot`
- `ServerName`
- `ServerAdmin`
- `ErrorLog`
- `TransferLog`

29.6.2.1 名前ベースの仮想ホストとSSL

IPアドレスが1つだけのサーバで、複数のSSL対応の仮想ホストを実行することはできません。名前ベースの仮想ホスティングでは、要求されたサーバ名をApacheが知っている必要があります。SSL接続の問題は、SSL接続が(デフォルトの仮想ホストの使用により)確立された後でのみ、そのような要求の読み込みが可能なことです。その結果、証明書がサーバ名に一致しないという警告メッセージが表示されます。

SUSE Linux Enterprise Serverは、SNI (Server Name Indication)と呼ばれるSSLプロトコルの拡張を組み込んでおり、仮想ドメインの名前をSSLネゴシエーションの一部として送信することで、この問題を解決します。これにより、サーバが正しい仮想ドメインに早く「切り替わり」、ブラウザに正しい証明書を提示することが可能になります。

SUSE Linux Enterprise Serverでは、SNIはデフォルトで有効になっています。名前ベースの仮想ホストをSSLで使用可能にするには、29.2.2.1.1項「名前ベースの仮想ホスト」で説明されているようにサーバを設定します (ただし、SSLでは、ポート `80` ではなく、ポート `443` を使用)。

重要: SNIブラウザのサポート

SNIは、クライアント側でもサポートされる必要があります。SNIは、ほとんどのブラウザでサポートされていますが、モバイルハードウェアの一部のブラウザやWindows* XP上のInternet ExplorerとSafariにはSNIのサポートがありません。詳細については、http://en.wikipedia.org/wiki/Server_Name_Indicationを参照してください。

ディレクティブ `SSLStrictSNIVHostCheck` を使用して、SNIに非対応のブラウザを処理する方法を設定しますSNI非対応ブラウザは、サーバ設定で `on` に設定されると、すべての仮想ホストに関して拒否されます。`VirtualHost` ディレクティブ内で `on` に設定されると、この特定のホストへのアクセスが拒否されます。

サーバ設定で `off` に設定されると、サーバはSNIサポートがないかのように動作します。SSL要求は、(ポート443に対して)定義された最初の仮想ホストによって処理されます。

29.7 セキュリティ問題の回避

公共のインターネットに公開しているWebサーバについては、管理面での不断の努力が求められます。ソフトウェアと、偶然の設定ミスの両方に関連したセキュリティの問題が発生することは避けられません。それらに対処するためのいくつかのヒントを紹介します。

29.7.1 最新版のソフトウェア

Apacheソフトウェアに脆弱性が見つかると、SUSEからセキュリティ上の勧告が出されます。これには、脆弱性を修正するための指示が含まれているので、可能な限り早期の適用が必要です。SUSEセキュリティ通知は、次の場所から入手できます。

- Webページ. http://www.suse.com/support/security/
- メーリングリストのアーカイブ. http://lists.opensuse.org/opensuse-security-announce/
- セキュリティアナウンスメントのリスト. http://www.suse.com/support/update/

29.7.2 DocumentRootの許可

SUSE Linux Enterprise Serverのデフォルトでは、`DocumentRoot`ディレクトリの`/srv/www/htdocs`およびCGIディレクトリの`/srv/www/cgi-bin`の所有者はユーザおよびグループの`root`になっています。これらのパーミッションは変更しないでください。ディレクトリにすべてのユーザが書き込み可能な場合、どのユーザもそれらのディレクトリにファイルを格納できます。その後これらのファイルは、Apacheにより`wwwrun`のパーミッションで実行されます。その結果、意図しない仕方で、ユーザがファイルシステムのリソースにアクセスできるようになる可能性があります。`/srv/www`のサブディレクトリを使用して仮想ホストの`DocumentRoot`およびCGIディレクトリを配置し、このユーザおよびグループの`root`がディレクトリとファイルの所有者であることを確認します。

29.7.3 ファイルシステムアクセス

デフォルトでは、ファイルシステム全体へのアクセスは、`/etc/apache2/httpd.conf`で定義されています。これらのディレクティブは決して上書きしないでください。ただし、Apacheが読み込む必要のあるすべてのディレクトリに対するアクセスは有効にしてください。詳細については、29.2.2.1.3項 「基本的な仮想ホスト設定」を参照してください。このためには、パスワードまたはシステム設定ファイルなど重要なファイルは外部から読み取ることができないことを確認します。

29.7.4 CGIスクリプト

Perl、PHP、SSIまたは他のプログラミング言語によるインタラクティブなスクリプトは、事実上、任意のコマンドを実行できるため、一般的なセキュリティの問題が存在します。サーバから実行されるスクリプトは、サーバの管理者が信用するソースからのみインストールされる必要があります。一般的には、ユーザが独自のスクリプトを実行できる環境は適切ではありません。また、すべてのスクリプトに対してセキュリティ監査を行うこともお勧めします。

スクリプトの管理をできるだけ簡単にするため、CGIスクリプトの実行をグローバルに許可するのではなく、通常、特定のディレクトリに制限されています。設定には、ディレクティブの`ScriptAlias`および`Option ExecCGI`が使用されます。SUSE Linux Enterprise Serverのデフォルト設定では、任意の場所からのCGIスクリプトの実行は許可されていません。

すべてのCGIスクリプトは同一のユーザとして実行するため、異なるスクリプトが互いに競合する可能性があります。suEXECモジュールは、CGIスクリプトを別のユーザとグループで実行できるようにします。

29.7.5 ユーザディレクトリ

ユーザディレクトリを(`mod_userdir`または`mod_rewrite`を使用して)有効化する場合は、`.htaccess`ファイルを許可しないことをお勧めします。これらのファイルは、ユーザによるセキュリティ設定の上書きを可能にするからです。`AllowOverRide`ディレクティブを使用して、少なくとも、ユーザの操作を制限する必要があります。SUSE Linux Enterprise Serverでは、`.htaccess`ファイルはデフォルトで有効化されていますが、ユーザは`mod_userdir`を使用するときにいずれの`Option`ディレクティブも上書きすることは許可されていません(`/etc/apache2/mod_userdir.conf`設定ファイルを参照してください)。

29.8 トラブルシューティング

Apacheが起動しないと、Webページにアクセスすることはできず、ユーザがWebサーバに接続することもできないので、問題の原因を見つけ出すことは重要です。次に、エラーが説明されている場所とチェックすべき重要事項について説明します。

`apache2.service`サブコマンドの出力:

Webサーバをバイナリの`/usr/sbin/httpd2`で起動/停止する代わりに、`systemctl`コマンドを使用します(29.3項 「Apacheの起動および停止」を参照)。このスクリプトは、エラーを詳細に説明し、設定エラーを修正するコツやヒントも提供します。

ログファイルと冗長性レベル

致命的エラーと致命的でないエラーの両方について、Apacheログファイル(主に、デフォルトで`/var/log/apache2/error_log`にあるエラーログファイル)をチェックしてください。さらに、ログファイルにさらに詳細な情報を記録することが必要な場合には、`LogLevel`ディレクティブで、記録されるメッセージの詳細を制御することができます。

ヒント: 簡単なテスト

`tail -F /var/log/apache2/my_error_log`コマンドで、Apacheのログメッセージを確認します。その後、`systemctl restart apache2.service`を実行します。そして、ブラウザでの接続をもう一度試みて、出力を確認してください。

ファイアウォールとポート

よくある間違いで、サーバのファイアウォール設定でApache用のポートを開けていないことがあります。YaSTでApacheを設定する場合には、この点を扱うための別のオプションが存在します(29.2.3項 「ApacheをYaSTで設定する」を参照してください)。Apacheを手動で設定する場合は、YaSTのファイアウォールモジュールを使用してHTTPとHTTPS用のファイアウォールポートを開きます。

このようにしても、エラーを特定できない場合には、http://httpd.apache.org/bug_report.html の、オンラインのApacheバグデータベースをチェックしてください。加えて、http://httpd.apache.org/userslist.html のメーリングリストで、Apacheのユーザコミュニティに参加することができます。お勧めできるニュースグループは、comp.infosystems.www.servers.unix です。

29.9 詳細情報

apache2-doc **パッケージには、ローカルインストールおよび参照用にそれぞれローカライズされている完全なApacheマニュアルが含まれています。** これは、デフォルトではインストールされません。このマニュアルを最も素早くインストールするには、`zypper in apache2-doc` コマンドを使用します。Apacheマニュアルは、インストールされると、http://localhost/manual/ から表示できるようになります。また、Webのhttp://httpd.apache.org/docs-2.4/ からもアクセスできます。SUSE固有の設定に関するヒントについては、`/usr/share/doc/packages/apache2/README.*` ディレクトリを参照してください。

29.9.1 Apache 2.4

Apache 2.4の新機能のリストについては、http://httpd.apache.org/docs/2.4/new_features_2_4.html を参照してください。バージョン2.2から2.4へのアップグレード情報もhttp://httpd.apache.org/docs-2.4/upgrading.html で参照できます。

29.9.2 Apacheモジュール

29.4.5項「外部モジュール」で簡単に説明されている外部Apacheモジュールの詳細は、次の場所で入手できます。

mod-apparmor

http://en.opensuse.org/SDB:AppArmor

mod-auth_kerb

http://modauthkerb.sourceforge.net/

`mod_perl`

http://perl.apache.org/

`mod_php5`

http://www.php.net/manual/en/install.unix.apache2.php

`mod_python`

http://www.modpython.org/

`mod_security`

http://modsecurity.org/

29.9.3 開発

Apacheモジュールの開発、またはApache Webサーバプロジェクトへの参加に関する情報については、次を参照してください。

Apache開発情報

http://httpd.apache.org/dev/

Apache開発者ドキュメント

http://httpd.apache.org/docs/2.4/developer/

PerlおよびCを使用したApacheモジュールの作成

http://www.modperl.com/

29.9.4 その他の情報源

SUSE Linux Enterprise ServerのApacheに固有な問題が発生した場合は、Technical Information Search (http://www.novell.com/support)を参照してください。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。

30 YaSTを使用したFTPサーバの設定

YaST [FTPサーバ]モジュールを使用すると、コンピュータをFTP (File Transfer Protocol)サーバとして機能するように設定できます。匿名および/または認証されたユーザがコンピュータに接続し、FTPプロトコルを使用してファイルをダウンロードできます。設定によっては、それらのユーザがFTPサーバにファイルをアップロードすることも可能です。YaSTはvsftpd (Very Secure FTP Daemon)を使用します。

YaST FTPサーバモジュールがシステム内にない場合は、`yast2-ftp-server`パッケージをインストールしてください。

YaSTで、FTPサーバを設定するには、次の手順に従います。

1. YaSTコントロールセンターを開き、[ネットワークサービス] › [FTPサーバ]の順に選択するか、`root`として`yast2 ftp-server`コマンドを実行します。
2. システムにFTPサーバがインストールされていない場合は、YaST FTPサーバモジュールの起動時に、インストールするサーバをどれにするか質問されます。vsftpdサーバを選択してダイアログを確認します。
3. [起動]ダイアログで、FTPサーバの起動に関するオプションを設定します。詳細については、30.1項「FTPサーバの起動」を参照してください。
   [一般]ダイアログで、FTPディレクトリ、歓迎メッセージ、ファイル作成マスクなどの各種パラメータを設定します。詳細については、30.2項「FTP一般設定」を参照してください。
   [Performance]ダイアログで、FTPサーバの負荷に影響するパラメータを設定します。詳細については、30.3項「FTPパフォーマンス設定」を参照してください。
   [認証]ダイアログで、匿名および/または認証されたユーザに対してFTPサーバを使用可能にするかどうか設定します。詳細については、30.4項「認証」を参照してください。
   [エキスパート設定]ダイアログで、FTPサーバの操作モード、SSL接続、およびファイアウォール設定を設定します。詳細については、30.5項「エキスパート設定」を参照してください。
4. [完了]を押して設定を保存します。

30.1 FTPサーバの起動

[FTP Start-Up]ダイアログの[サービス開始]フレームで、FTPサーバを起動する方法を設定します。システムブート時の自動的なサーバ起動とサーバの手動起動のどちらかを選択できます。FTP接続要求後にのみFTPサーバを起動する場合は、[xinetd経由]を選択します。

FTPサーバの現在のステータスが、[FTP Start-Up]ダイアログの[開始/停止]フレームに表示されます。[FTPを開始する]をクリックして、FTPサーバを起動します。サーバを停止するには、[FTPを停止する]をクリックします。サーバの設定を変更したら、[設定を保存してFTPを再起動する]をクリックします。[完了]を押して設定モジュールを終了すると、設定が保存されます。

[FTP起動]ダイアログの[選択されたサービス]フレームに、使用されるFTPサーバ(vsftpdまたはpure-ftpd)が表示されます。両方のサーバがインストールされている場合、それらを切り替えることができます。現在の設定は自動的に変換されます。

図 30.1 FTPサーバの設定 — 起動

30.2 FTP一般設定

[FTP General Settings]ダイアログの[一般の設定]フレームで、FTPサーバへの接続後に表示される[Welcome message]を設定できます。

[Chroot Everyone]オプションをオンにした場合は、すべてのローカルユーザが、ログイン後、ホームディレクトリのchroot jailに配置されます。このオプションは、セキュリティに影響します(特に、ユーザがアップロードパーミッションまたはシェルアクセスを持つ場合)。したがって、このオプションの有効化には、注意が必要です。

[Verbose Logging]オプションをオンにすると、すべてのFTP要求と応答がログされます。

匿名および/または認証されたユーザが作成するファイルのパーミッションは、umaskで制限できます。[Umask for Anonymous])]で匿名ユーザ用のファイル作成マスクを設定し、[Umask for Authenticated Users]で認証されたユーザ用のファイル作成マスクを設定します。マスクは、必ず0で始まる8進数として入力してください。umaskの詳細については、umaskマニュアルページ(`man 1p umask`)を参照してください。

[FTP Directories]フレームで、匿名/認証されたユーザ用のディレクトリを設定します。[参照]をクリックすると、ローカルファイルシステムから使用できるディレクトリを選択できます。匿名ユーザのデフォルトFTPディレクトリは、`/srv/ftp`です。ただし、vsftpdでは、このディレクトリにすべてのユーザが書き込むことはできません。代わりに、書き込みパーミッション付きのサブディレクトリ`upload`が匿名ユーザ用に作成されます。

注記: FTPディレクトリの書込みパーミッション

pure-ftpdサーバでは、匿名ユーザ用のFTPディレクトリを書き込み可能にできます。サーバ間で切り換えを行う場合は、vsftpdサーバに戻す前に、pure-ftpdで使用したディレクトリから書き込みパーミッションを削除したことを確認してください。

30.3 FTPパフォーマンス設定

[パフォーマンス]ダイアログで、FTPサーバの負荷に影響するパラメータを設定します。[Max Idle Time]は、リモートクライアントがFTPのコマンド間で待機できる最大時間(分)です。これよりアクティビティのない時間が長くなると、リモートクライアントの接続は切断されます。[Max Clients for One IP]では、1つのIPアドレスから接続できるクライアントの最大数を決定します。[最大クライアント]では、接続できるクライアントの最大数を決定します。クライアントをさらに追加すると、エラーメッセージが表示されます。

最大データ転送速度(KB/秒)の設定は、ローカルの認証されたユーザについては[Local Max Rate]、匿名クライアントについては[Anonymous Max Rate]で行います。速度設定のデフォルト値は、`0`であり、無制限のデータ転送速度を意味します。

30.4 認証

[認証]ダイアログの[Enable/Disable Anonymous and Local Users]フレームでは、どのユーザにFTPサーバへのアクセスを許可するか設定できます。次のオプションのいずれかを選択できます: 匿名ユーザのみ、(システムにアカウントのある)認証されたユーザのみ、またはその両方のタイプのユーザにアクセスを付与します。

FTPサーバへのファイルのアップロードを許可するには、[認証]ダイアログの[Uploading]フレームにある[Enable Upload]をオンにします。ここでは、各ボックスにチェック印を入れることで、匿名ユーザにも、アップロードまたはディレクトリの作成を許可できます。

注記: vsftp—匿名ユーザのファイルのアップロードを許可する

vsftpdサーバを使用し、匿名ユーザにファイルをアップロードさせたり、ディレクトリを作成させる場合は、すべてのユーザ用書き込みパーミッション付きのサブディレクトリを、匿名FTPディレクトリ内に作成する必要があります。

30.5 エキスパート設定

FTPサーバは、アクティブモードまたはパッシブモードで実行できます。デフォルトでは、サーバはパッシブモードで実行されます。アクティブモードに切り換えるには、[エキスパート設定]ダイアログの[パッシブモードを許可する]オプションをオフにします。データストリーム用に使用するサーバのポート範囲を変更することもできます。このためには、[Min Port for Pas. Mode]と[Max Port for Pas. Mode]のオプションを微調整します。

クライアントとサーバ間で暗号化された通信が必要な場合は、[SSLを有効に]できます。サポートされるプロトコルのバージョンをチェックし、SSL暗号化接続で使用されるDSA証明書を指定します。

システムがファイアウォールで保護されている場合は、[ファイアウォール内でポートを開く]をオンにして、FTPサーバへの接続を有効にします。

30.6 さらに詳細な説明が必要な場合は

FTPサーバの詳細については、`vsftpd`および`vsftpd.conf`のマニュアルページを参照してください。

31 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。これは、WebまたはFTPサーバなど、要求されたインターネットオブジェクトを、サーバよりも要求しているワークステーションに近いマシン上に格納することを意味します。Squidは、応答時間や低帯域幅の使用を最適化するために複数の階層上でセットアップされます。エンドユーザにとって透過的なモードである場合さえあります。squidGuardを利用すれば、Webコンテンツをフィルタリングすることができます。

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とともに、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。

31.1 プロキシキャッシュに関する注意事項

プロキシキャッシュとして、Squidは複数の方法で使用されます。ファイアウォールと組み合わせると、セキュリティに役立ちます。複数のプロキシを一緒に使用できます。また、キャッシュされるオブジェクトのタイプ、およびその期間も決定できます。

31.1.1 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。この設定では、SquidはWebアクセスを完全に制御します。

ファイアウォール設定にDMZが含まれている場合、プロキシはこのゾーン内で動作しなければなりません。「透過的な」プロキシの実装方法については、31.5項 「透過型プロキシの設定」を参照してください。この場合、プロキシに関する情報が必要とされないので、クライアントの設定が簡略化されます。

31.1.2 複数のキャッシュ

複数のSquidインスタンスを設定して、これらの間でオブジェクトを交換できます。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。

ヒント

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。

31.1.3 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。

31.2 システム要件

最も重要なのは、システムにかかる最大ネットワーク負荷を判断することです。ピーク時の負荷は1日の平均負荷の4倍を超えることもあるため、負荷のピークに注意する必要があります。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。

31.2.1 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。

31.2.2 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になります。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。

31.2.3 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、31.6項「cachemgr.cgi」を参照してください。

31.2.4 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。

31.3 Squidの起動

まだインストールしていない場合は、`squid`パッケージをインストールします。`squid`はデフォルトのSUSE® Linux Enterprise Serverインストールスコープに含まれていません。

SquidはSUSE® Linux Enterprise Server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。

31.3.1 Squidの起動コマンドと停止コマンド

Squidを起動するには、`root`としてコマンドラインで「`systemctl start squid.service`」と入力します。初期起動時には、最初に`/var/cache/squid`内でキャッシュの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、起動スクリプトにより自動的に実行され、完了までに数秒または数分かかります。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および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、31.4.2項「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、`systemctl reload squid.service`を使用します。または、`systemctl restart squid.service`を使用して、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、`systemctl status squid.service`コマンドを使用します。Squidをシャットダウンするには、`systemctl stop squid.service`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。

警告: Squidの終了

`kill`または`killall`を使ってSquidを終了すると、キャッシュが破損してしまう可能性があります。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。

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確認してください。起動エラーの原因は、Squidにより`/var/log/squid/cache.log`ファイルに記録されます。Squidをシステムのブート時に自動的にロードする必要がある場合は、`systemctl enable squid.service`を実行してサービスを有効にします。

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。

31.3.2 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(22.4項 「BINDネームサーバの起動」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の確立時に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調整されます。この動作は`/etc/sysconfig/network/config`ファイルの`NETCONFIG_DNS_POLICY` sysconfig変数で制御されます。YaST sysconfigエディタで、`NETCONFIG_DNS_POLICY`を`""`に設定します。次に、`/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`設定ファイル内の`forwarders`にサーバ名とそのIPアドレスを入力します。ダイナミックDNSを使用すると、sysconfig変数の`NETCONFIG_DNS_POLICY`を「`auto`」に設定することによって、この動作を接続の確立時に自動的に実行することができます。

スタティックDNS

スタティックDNSを使用する場合は、接続の確立時にいずれの自動DNS調整も行われないため、sysconfig変数を変更する必要はありません。ただし、`/etc/resolv.conf` ファイルにローカルのDNSサーバを入力する必要があります。また、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` 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。

ヒント: 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。

31.4 etc/squid/squid.conf設定ファイル

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` 設定ファイルには、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。この方法では、簡単にデフォルト値に戻し、変更と比較することができます。

ヒント: 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` ファイルを使用すると、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。

31.4.1 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または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、ブラウザで使用する親の演算子でも指定されているポート番号(通常は `8080`)を入力します。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ため、ポート番号に続けて `default` および `no-query` を指定することもできます。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で頻繁に求められる応答に対して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。これは、Squidのメモリ使用量を指定せず、メモリ使用量を超えても構いません。

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%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,

cache_log /var/log/squid/cache.log,

cache_store_log /var/log/squid/store.log

これらの3つのエントリは、Squidによるすべてのアクションのログファイルへのパスを指定します。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

このエントリを使用して、ログファイルでクライアントのIPアドレスをマスクします。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。このようにして、クライアントのプライバシーを保護できます。

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確認されるため、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。SUSE Linux Enterprise Server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`内に設定されたcronジョブ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。設定しない場合は、次のような行がヘッダに追加されます。

```
X-Forwarded-For: 192.168.0.1
```

negative_ttl 5 minutes; negative_dns_ttl 5 minutes

通常、これらのl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に変更します。次にブラウザの更新をクリックすると、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。

31.4.2 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。この場合、ACLを設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
```

および最後の

```
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。プロキシ認証と適切なACLを利用すれば、さまざまなユーザグループ個別にインターネットアクセスを制御することができます。squidGuardを使用する場合は、個別にインストール、設定する必要があります。

auth_param basic program /usr/sbin/pam_auth

プロキシ上でユーザを認証する必要がある場合は、pam_authなどの対応するプログラムを設定します。最初にpam_authにアクセスすると、ユーザ名とパスワードを入力するためのログインウィンドウが表示されます。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。Microsoft　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。

31.5　透過型プロキシの設定

一般的なプロキシサーバの作業では、Webブラウザがプロキシサーバの特定のポートに要求を送信し、プロキシが要求に応じて必要なオブジェクトを提供します。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。

31.5.1 /etc/squid/squid.conf内の設定オプション

squidを透過的なプロキシとして動作させるには、メインの設定ファイル `/etc/squid/squid.conf` 内で `http_port` タグの `transparent` オプションを使用します。squidの再起動後に必要なことは、HTTPポートを `http_port` で指定されたポートにリダイレクトするようファイアウォールを再設定することだけです。次のsquid設定行では、これはポート3128になっています。

```
http_port 3128 transparent
```

31.5.2 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、Book “Security Guide” 15 “Masquerading and Firewalls” 15.4.1 “Configuring the Firewall with YaST”で説明しているように、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。透過型プロキシを設定するには、次に示すようにいくつかのファイアウォールオプションを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"
- インターネットを指すデバイス: `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)。この例では、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。

例 31.1 ファイアウォールの設定:オプション15

```
# 15.)
# Which accesses to services should be redirected to a local port on
# the firewall machine?
#
# This option can be used to force all internal users to surf via
# your squid proxy, or transparently redirect incoming webtraffic to
# a secure webserver.
#
# Format:
# list of <source network>[,<destination network>,<protocol>[,dport[:lport]]
# Where protocol is either tcp or udp. dport is the original
# destination port and lport the port on the local machine to
# redirect the traffic to
#
# An exclamation mark in front of source or destination network
# means everything EXCEPT the specified network
#
# Example: "10.0.0.0/8,0/0,tcp,80,3128 0/0,172.20.1.1,tcp,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以外のプロトコルをサポートしているため、要求は他のポートからFTP (ポート21)、HTTPSまたはSSL (ポート443)などのプロキシに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="192.168.0.0/16,0/0,tcp,80,3128"
```

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。

31.3項 「Squidの起動」のように、Squidを起動します。すべてが正常に機能していることを確認するには、`/var/log/squid/access.log`のSquidログを確認します。
すべてのポートが正常に設定されていることを確認するには、ネットワーク外部の任意のコンピュータから、マシンのポートスキャンを実行します。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。

31.6 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。

31.6.1 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。で説明しているように、Apacheを設定します。第29章 Apache HTTPサーバApacheがすでに稼働しているかどうかを確認するには、`root`として`systemctl status apache2.service`コマンドを入力します。稼働していない場合は、「`systemctl start apache2.service`」を入力して、SUSE Linux Enterprise Server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。32ビットの場合は次のようになります。

```
cp /usr/lib/squid/cachemgr.cgi /srv/www/cgi-bin/
```

64ビット環境では、`cachemgr.cgi` ファイルは `/usr/lib64/squid/` の下に位置しており、これをApacheディレクトリにコピーするコマンドは次のとおりです。

```
cp /usr/lib64/squid/cachemgr.cgi /srv/www/cgi-bin/
```

31.6.2 /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。最初に、2つのACLを定義し、http_accessオプションがこれらのACLを使用して、CGIスクリプトからSquidへのアクセスを付与するようにします。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
```

次の規則によって、ApacheにSquidへのアクセス権が付与されます。

```
http_access allow manager localhost
http_access deny manager
```

これらの規則は、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31.2「アクセスルール」に示すACLを追加します。

例 31.2 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31.3「アクセスルール」に規則を追加して、Webサーバからのアクセスを許可します。

例 31.3 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、`systemctl reload squid.service` を使用します。

31.6.3 統計情報の表示

対応するWebサイトのhttp://webserver.example.org/cgi-bin/cachemgr.cgi にアクセスします。[続行]をクリックして様々な統計情報をブラウズします。

31.7 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。squidGuardを利用すれば、Squidキャッシュ上にある異なるユーザグループに対して、異なる制限を持つ複数のアクセスルールを定義することができます。squidGuardは、Squidの標準リダイレクタインタフェースを使用しています。squidGuardの機能を以下に示します。

- 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 を使用するにははじめに、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。に設定例が用意されています。http://www.squidguard.org/Doc/examples.html 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
```

他の `redirect_children` と呼ばれるオプションには、コンピュータ上で動作するリダイレクト(この場合はsquidGuard)プロセス数を設定します。「」 プロセスをより多く設定すると、RMMもそれだけ多く必要になります。最初は、`4` などの少ない数で試します。

```
redirect_children 4
```

最後に、`systemctl reload squid.service` 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。

31.8 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。このツールはSUSE Linux Enterprise Serverのデフォルトインストールスコープには含まれていません。これを使用するには、`calamaris` パッケージをインストールしてください。

rootとしてログインし、「cat access.log | 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。

ヒント: シェルとファイルの順序

access.log.1、access.log.2 などのような類似ファイルが複数ある場合、 デフォルトのシェルbashはこれらのファイルを番号以外の順序でソートして、access.log.を一覧表示します。*. この問題を解決するには、次の構文を使用できます。access.log.{1..42}。これによって1～42の数字拡張子の付いたファイルのリストが生成されます。

-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.{10..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。

31.9 詳細情報

にあるSquidのホームページにアクセスしてください。http://www.squid-cache.org/ ここにはS「quid 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 `howtoenh` 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。

32 SFCBを使用したWebベースの企業管理

32.1 概要および基本概念

SUSE® Linux Enterprise Server (SLES)は、異種コンピューティングシステムおよび環境を統合管理するためのオープンスタンダードベースのツールのコレクションを提供しています。弊社の企業ソリューションでは、Distributed Management Task Forceが提案する標準を実装しています。ここでは、基本コンポーネントについて説明します。

Distributed Management Task Force, Inc (DMTF)は、企業およびインターネットの環境に対する管理標準の開発を推進する業界団体です。DMTFは、管理の標準とイニシアチブを統合し、管理ソリューションを、より高い統合性とコスト効果を持つ、より相互運用可能なものにすることを目的としています。DMTF標準は、制御および通信のための共通システム管理コンポーネントを提供します。こうしたソリューションは、プラットフォームや技術に依存しません。Webベースの企業管理および共通情報モデルは重要な技術の1つです。

Webベースの企業管理(WBEM)は、管理およびインターネット標準技術群です。WBEMは、企業のコンピューティング環境の管理を統合するために開発されました。Webテクノロジを使用した統一管理ツールコレクションを作成する機能を業界に提供するものです。WBEMは、次の標準で構成されます。

- データモデル: CIM(Common Information Model)標準
- 符号化規格: CIM-XML符号化規格
- 伝送メカニズム: CIM operations over HTTP

共通情報モデルは、システム管理について記述した概念的な情報モデルです。特別な実装は必要なく、管理システム、ネットワーク、サービス、およびアプリケーション間で管理情報を交換できます。CIMには、2つのパート(CIM仕様とCIMスキーマ)があります。

- CIM仕様は、言語、ネーミング、およびメタスキーマを記述します。メタスキーマは、モデルの公式な定義です。メタスキーマは、モデルの内容、使用方法、および意味の説明に使う用語を定義します。メタスキーマの要素は、クラス、プロパティ、およびメソッドです。また、メタスキーマは、指示と関連付けをクラスのタイプとして、参照をプロパティとしてサポートします。
- CIMスキーマは、実際のモデルを記述します。このスキーマは、管理対象環境について利用可能な情報を編成できる汎用の概念的なフレームを提供する、プロパティと関連を持つ一連の名前が付けられたクラスです。

Common Information Model Object Manager (CIMOM)は、CIM標準に基づいてオブジェクトを管理するアプリケーションです(CIM Object Manager)。CIMOMは、CIMOMプロバイダと、管理者がシステムを管理するCIMクライアントの間の通信を管理します。

CIMOMプロバイダは、クライアントアプリケーションから要求された特定のタスクをCIMOM内で実行するソフトウェアです。各プロバイダは、CIMOMのスキーマの1つまたは複数の機能や役割を果たします。これらのプロバイダは、ハードウェアを直接操作します。

SBLIM (Standards Based Linux Instrumentation for Manageability)は、Webベースの企業管理(WBEM)をサポートするために設計されたツールのコレクションです。SUSE® Linux Enterprise Serverは、Small Footprint CIM Brokerと呼ばれるSBLIMプロジェクトのオープンソースCIMOM(またはCIMサーバ)を使用します。

コンパクトなフットプリントのCIMブローカは、リソースに制限のある環境または埋め込み環境での使用を対象としたCIMサーバです。このサーバは、モジュール性と軽量性を同時に備えた設計になっています。このサーバはオープンスタンダードをベースとし、CMPIプロバイダ、CIM-XMLエンコーディング、および管理オブジェクトフォーマット(MOF)をサポートします。これは高度に設定可能なサーバであり、プロバイダがクラッシュしても動作は安定しています。また、HTTP、HTTPS、Unixドメインソケット、サービスロケーションプロトコル(SLP)、Javaデータベース接続(JDBC)など、さまざまなトランスポートプロトコルがサポートされるために、簡単にアクセスできます。

32.2 SFCBの設定

SFCB (Small Footprint CIM Broker)環境を設定するには、SUSE Linux Enterprise Serverのインストール時にYaSTの［Webベースの企業管理パターン］が選択されていることを確認します。また、すでに実行中のサーバにインストールするコンポーネントとしてこれを選択します。次のパッケージがシステムにインストールされていることを確認します。

cim-schema、CIM (Common Information Model)スキーマ
: 共通情報モデル(CIM)が含まれます。CIMは、ネットワーク/企業環境内の総合的な管理情報を記述するモデルです。CIMは仕様とスキーマで構成されます。仕様は、他の管理モデルとの統合に関する詳細を定義しています。スキーマは、実際のモデルを記述しています。

cmpi-bindings-pywbem
: CMPIタイプのCIMプロバイダをPythonで記述および実行するためのアダプタが含まれます。

cmpi-pywbem-base
: 基本システムのCIMプロバイダが含まれます。

cmpi-pywbem-power-management
: DSP1027に基づく電源管理プロバイダが含まれます。

python-pywbem

管理対象オブジェクトをクエリおよび更新するために、WBEMプロトコルを使用してCIM操作呼び出しを行うためのPythonモジュールが含まれます。

cmpi-provider-register、CIMOM中立プロバイダ登録ユーティリティ

システム上に存在するすべてのCIMOMをCMPIプロバイダパッケージに登録できるユーティリティが含まれます。

sblim-sfcb、コンパクトなフットプリントのCIMブローカ

コンパクトなフットプリントのCIMブローカが含まれます。これは、CIM Operations over HTTPプロトコルに準拠するCIMサーバです。堅牢でリソース消費が抑制されているために、埋め込み環境およびリソースが制約された環境に特に適しています。SFCBでは、Common Manageability Programming Interface (CMPI)に対して記述されたプロバイダがサポートされます。

sblim-sfcc

コンパクトなフットプリントのCIMクライアントライブラリのランタイムライブラリが含まれます。

sblim-wbemcli

WBEMコマンドラインインタフェースが含まれます。これは、特に基本的なシステム管理タスクに適したスタンドアロンコマンドラインWBEMクライアントです。

smis-providers

Linuxファイルシステム上のボリュームおよびスナップショットを計測するためのプロバイダが含まれます。これらのプロバイダはそれぞれ、SNIAのSMI-Sボリューム管理プロファイルおよびコピーサービスプロファイルに基づきます。

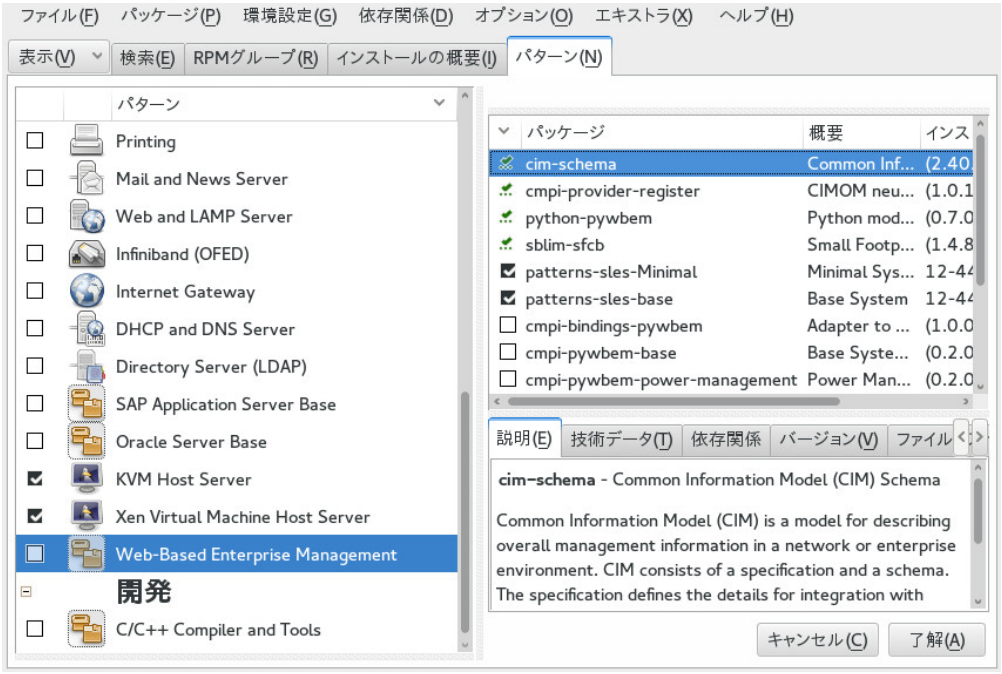


図 32.1 WEBベースの企業管理パターンのパッケージ選択

32.2.1 追加プロバイダのインストール

SUSE® Linux Enterprise Serverソフトウェアリポジトリには、Webベースの企業管理インストールパターンにない追加CIMプロバイダが含まれます。YaSTソフトウェアインストールモジュールでパターン `sblim-cmpi-` を検索することにより、プロバイダのリストやインストールの状態を簡単に参照できます。これらのプロバイダは、DHCP、NFS、カーネルパラメータ設定など、システム管理のさまざまなタスクに対応します。SFCBとともに使用するプロバイダをインストールしておくと役立ちます。

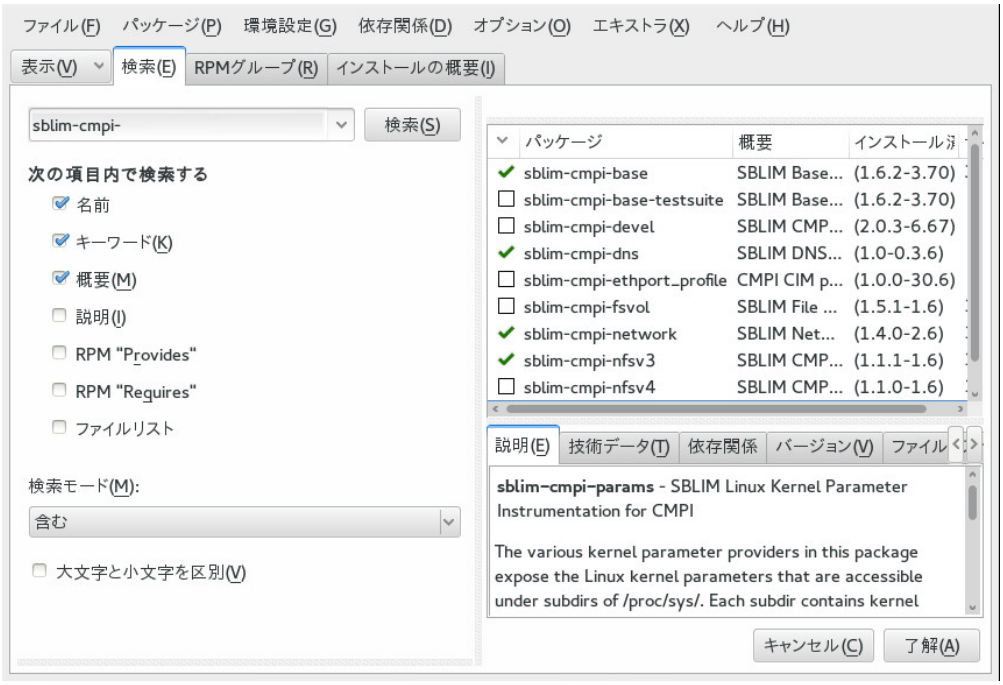


図 32.2 追加CIMプロバイダのパッケージ選択

32.2.2 SFCBの起動、終了、およびステータスの確認

CIMサーバのsfcbdデーモンは、Webベースの企業管理ソフトウェアとともにインストールされ、システム起動時にデフォルトで開始されます。次の表で、sfcbdの起動、停止、および確認ステータスを説明します。

表 32.1 SFCBDの管理用コマンド

タスク	Linux コマンド
Start sfcbd	コマンドラインで `root` として「`systemctl start sfcb.service`」と入力します。
sfcbdの停止	コマンドラインで `root` として「`systemctl stop sfcb.service`」と入力します。
sfcbdの状態の確認	コマンドラインで `root` として「`systemctl status sfcb.service`」と入力します。

32.2.3 セキュアアクセスの確保

SFCBのデフォルトのセットアップは、比較的安全(セキュア)です。ただし、SFCBコンポーネントに対するアクセスが組織で必要とされる安全性を満たしていることを確認します。

32.2.3.1 証明書

安全にSSL (Secure Socket Layers)通信を行うには、証明書が必要になります。SFCBがインストールされている場合、自己署名付き証明書が生成されています。

`/etc/sfcb/sfcb.cfg` の `sslCertificateFilePath:path_filename` 設定を変更することで、デフォルトの証明書のパスを商用証明書または自己署名付きの証明書のパスに置き換えることができます。ファイルは、PEMフォーマットであることが必要です。

デフォルトで生成されたサーバ証明書は、次の場所に置かれています。

`/etc/sfcb/server.pem`

注記: SSL証明書のパス

デフォルトで生成される証明書ファイル `servercert.pem` および `serverkey.pem` は、`/etc/ssl/servercerts` ディレクトリに保存されています。ファイル `/etc/sfcb/client.pem`、`/etc/sfcb/file.pem`、および `/etc/sfcb/server.pem` は、これらのファイルへのシンボリックリンクです。

新しい証明書を生成する場合は、`root` としてコマンドラインに次のコマンドを入力します。

```
tux@mercury:~> sh /usr/share/sfcb/genSslCert.sh
Generating SSL certificates in .
Generating a 2048 bit RSA private key
.....................................................................+++
.+++
writing new private key to '/var/tmp/sfcb.0Bjt69/key.pem'
-----
```

デフォルトでは、このスクリプトにより現在の作業ディレクトリに証明書 `client.pem`、`file.pem`、および `server.pem` が生成されます。スクリプトにより `/etc/sfcb` ディレクトリに証明書を生成する場合は、コマンドにこのディレクトリを追加する必要があります。これらのファイルがすでに存在する場合、警告メッセージが表示されます。古い証明書は上書きされません。

```
tux@mercury:~> sh /usr/share/sfcb/genSslCert.sh /etc/sfcb
```

```
Generating SSL certificates in .
WARNING: server.pem SSL Certificate file already exists.
         old file will be kept intact.
WARNING: client.pem SSL Certificate trust store already exists.
         old file will be kept intact.
```

ファイルシステムから古い証明書を削除し、このコマンドを再実行する必要があります。

SFCBで証明書を使用する方法を変更する場合は、32.2.3.3項「認証」を参照してください。

32.2.3.2 ポート

デフォルトでは、SFCBはセキュアなポート5989を使用するすべての通信を受け入れるように設定されます。ここでは、通信ポートのセットアップと推奨される設定について説明します。

ポート5989(セキュア)

SFCB通信がHTTPSサービスを介して使用するセキュアなポート。デフォルトの設定です。この設定で、CIMOMとクライアントアプリケーション間のすべての通信は、サーバとワークステーション間でインターネットを通じて送信されるときに暗号化されます。ユーザは、SFCBサーバにアクセスするためにクライアントアプリケーションで認証を受ける必要があります。この設定を記録しておくことをお勧めします。ルータやファイアウォールがクライアントアプリケーションとモニタリングされるノードとの間に存在する場合に、SFCB　CIMOMが必要なアプリケーションと通信できるようにするには、このポートを開いておく必要があります。

ポート5988(非セキュア)

SFCB通信がHTTPSサービスを介して使用する非セキュアなポート。デフォルトでは、この設定は無効にされています。この設定では、CIMOMとクライアントアプリケーション間のすべての通信は、サーバとワークステーション間でインターネットを通じて送信されるときに、誰でも認証なしで開き、レビューできます。この設定は、CIMOMの問題をデバッグするときのみに使用することをお勧めします。問題が解決されたら、すぐにセキュアでないポートオプションを無効にしてください。SFCB　CIMOMがセキュアでないアクセスを要求する必要なアプリケーションと通信できるようにするには、クライアントアプリケーションとモニタリングされるノードとの間に存在するルータやファイアウォールでこのポートを開いておく必要があります。

デフォルトのポートの割り当てを変更する場合は、32.2.3.2項「ポート」を参照してください。

32.2.3.3 認証

SFCBでは、HTTP基本認証、およびクライアント証明書に基づく認証がサポートされます(HTTP over SSL接続)。基本HTTP認証は、SFCB環境設定ファイル(デフォルトでは `/etc/sfcb/sfcb.cfg`)で、`doBasicAuth` = `true` を指定することにより有効になります。SFCBのSUSE® Linux Enterprise Serverインストールでは、プラグ可能認証モジュール(PAM)アプローチがサポートされます。したがって、ローカルルートユーザは、ローカルルートユーザの資格情報によりSFCB CIMOMに対して認証を行うことができます。

`sslClientCertificate` 設定プロパティが `accept` または `require` に設定されている場合、SFCB HTTPアダプタは、HTTP over SSL (HTTPS)で接続した時にクライアントに証明書を要求します。`require` が指定された場合、(`sslClientTrustStore` を介して指定されたクライアント信頼ストアに従って)クライアントは有効な証明書を提供する必要がありますクライアントが証明書を提供しない場合、接続はCIMサーバにより拒否されます。

`sslClientCertificate` = `accept` という設定は、明確でないことがあります。基本認証およびクライアント証明書認証が両方許可されている場合に、この設定は非常に役立ちます。クライアントが有効な証明書を提供できれば、HTTPS接続が確立され、基本認証手順は実行されません。この機能で証明書を検証できない場合、HTTP基本認証が代わりに実行されます。

32.3 SFCB CIMOM設定

SFCBは、CIMサーバの軽量な実装ですが、高度に設定可能です。複数のオプションによりその動作を制御できます。SFCBサーバは次の3つの方法で制御できます。

- 適切な環境変数を設定する
- コマンドラインオプションを使用する
- 環境設定ファイルを変更する

32.3.1 環境変数

いくつかの環境変数は、SFCBの動作に直接影響します。これらの環境変数の変更を有効にするには、`systemctl restart sfcb.service` でSFCBデーモンを再起動する必要があります。

`PATH`

`sfcbd` デーモンおよびユーティリティへのパスを指定します。

`LD_LIBRARY_PATH`

sfcbランタイムライブラリへのパスを指定します。また、このパスをシステムワイドの動的ローダ設定ファイル `/etc/ld.so.conf` に追加できます。

`SFCB_PAUSE_PROVIDER`

プロバイダ名を指定します。SFCBサーバは、プロバイダが最初にロードされた後に一時停止します。その後、デバッグの目的でプロバイダのプロセスにランタイムデバッガを接続できます。

`SFCB_PAUSE_CODEC`

SFCBコーデックの名前を指定します(現在、`http` のみサポートしています)。SFCBサーバは、コーデックが最初にロードされた後に一時停止します。その後、プロセスにランタイムデバッガを接続できます。

`SFCB_TRACE`

SFCBのデバッグメッセージレベルを指定します。有効な値は、0(デバッグメッセージなし)、または1(重要なデバッグメッセージ)～4(すべてのデバッグメッセージ)です。デフォルトは1です。

`SFCB_TRACE_FILE`

有効な値は、0 (デバッグメッセージなし)または1 (主要なデバッグメッセージ)～4 (すべてのデバッグメッセージ)です。この変数を設定すると、指定のファイルにデバッグメッセージが代わりに書き込まれます。

`SBLIM_TRACE`

SBLIMプロバイダのデバッグメッセージレベルを指定します。有効な値は、0(デバッグメッセージなし)、または1(重要なデバッグメッセージ)～4(すべてのデバッグメッセージ)です。

`SBLIM_TRACE_FILE`

デフォルトでは、SBLIMプロバイダはトレースメッセージをSTDERRに出力します。この変数を設定すると、指定のファイルにトレースメッセージが代わりに書き込まれます。

32.3.2 コマンドラインオプション

SFCBデーモン `sfcbd` には、特定のランタイム機能をオン/オフするためのコマンドラインオプションがあります。SFCBデーモンの開始時に、これらのオプションを入力します。

`-c, --config-file=FILE`

SFCBデーモンの開始時に、デフォルトで `/etc/sfcb/sfcb.cfg` から設定が読み込まれます。このオプションでは、代替の環境設定ファイルを指定できます。

`-d, --daemon`

バックグラウンドで実行するようにsfcbdとその子プロセスを強制します。

`-s, --collect-stats`

ランタイム統計情報の収集をオンにします。現在の作業ディレクトリの `sfcbStat` ファイルに、さまざまなsfcbdランタイム統計情報が書き込まれます。デフォルトでは、統計情報は収集されません。

`-l, --syslog-level=LOGLEVEL`

システムログ機能の冗長性レベルを指定します。`LOGLEVEL` は、LOG_INFO、LOG_DEBUG、またはLOG_ERR (デフォルト)のいずれかになります。

`-k, --color-trace=ログレベル`

プロセスごとに異なる色でトレース出力を印刷して、デバッグを容易にします。

`-t, --trace-components=NUM`

`NUM` がトレースするコンポーネントを定義するORビットマスク整数である場合に、コンポーネントレベルのトレースメッセージをアクティブにします。`-t ?を指定した後` すべてのコンポーネントおよび関連する整数ビットマスクが表示されます。

```
tux@mercury:~> sfcbd -t ?
---   Traceable Components:     Int       Hex
---         providerMgr:          1 0x0000001
---         providerDrv:          2 0x0000002
---          cimxmlProc:          4 0x0000004
---          httpDaemon:          8 0x0000008
---             upCalls:         16 0x0000010
---            encCalls:         32 0x0000020
---     ProviderInstMgr:         64 0x0000040
---    providerAssocMgr:        128 0x0000080
---           providers:        256 0x0000100
---         indProvider:        512 0x0000200
---    internalProvider:       1024 0x0000400
---          objectImpl:       2048 0x0000800
---               xmlIn:       4096 0x0001000
---              xmlOut:       8192 0x0002000
---             sockets:      16384 0x0004000
---          memoryMgr:       32768 0x0008000
---            msgQueue:      65536 0x0010000
```

```
---           xmlParsing:     131072 0x0020000
---      responseTiming:     262144 0x0040000
---           dbpdaemon:     524288 0x0080000
---                 slp:    1048576 0x0100000
```

sfcbdの内部機能を表示し、メッセージを生成しすぎない有用な値は `-t 2019` です。

32.3.3 SFCB環境設定ファイル

SFCBは、起動後に環境設定ファイル `/etc/sfcb/sfcb.cfg` からランタイム設定を読み込みます。この動作は、起動時に `-c` オプションを使用して上書きできます。

環境設定ファイルには、**オプション** : 値 のペアが1行に1つずつ含まれています。このファイルに変更を加える場合は、使用している環境にネイティブな形式でファイルを保存するどのテキストエディタでも使用できます。

オプションがシャープ記号(#)でコメントアウトされている設定では、デフォルト設定が使用されます。

次のオプションリストは、完全でない可能性があります。完全なリストについては、`/etc/sfcb/sfcb.cfg` と `/usr/share/doc/packages/sblim-sfcb/README` を参照してください。

32.3.3.1 httpPort

目的

CIMクライアントからのHTTP(非セキュア)要求を受信するためにsfcbdがリスンするローカルポート値を指定します。デフォルトは `5988` です。

構文

`httpPort: port_number`

32.3.3.2 enableHttp

目的

SFCBがHTTPクライアント接続を受け入れるかどうかを指定します。デフォルトは `false` です。

構文

`enableHttp: option`

オプション	説明
true	HTTP接続を有効にします。
false	HTTP接続を無効にします。

32.3.3.3 httpProcs

目的

新しい着信HTTP要求を拒否するまでの同時HTTPクライアント接続の最大数を指定します。デフォルトは`8`です。

構文

`httpProcs: max_number_of_connections`

32.3.3.4 httpUserSFCB、httpUser

目的

これらのオプションは、HTTPサーバを実行するユーザを管理します。`httpUserSFCB`が`true`の場合、HTTPは、SFCBメインプロセスと同じユーザで実行されます。`false`の場合は、`httpUser`で指定されたユーザ名が使用されます。この設定は、HTTPとHTTPSの両方のサーバに使用されます。`httpUserSFCB`を`false`に設定する場合は、`httpUser`を指定する必要があります。デフォルトは、`true`です。

構文

`httpUserSFCB: true`

32.3.3.5 httpLocalOnly

目的

HTTP要求をローカルホストだけに制限するかどうか指定します。デフォルトは `false` です。

構文

```
httpLocalOnly: false
```

32.3.3.6 httpsPort

目的

sfcbdがCIMクライアントからのHTTPS要求をリスンするローカルポート値を指定します。デフォルトは `5989` です。

構文

```
httpsPort: port_number
```

32.3.3.7 enableHttps

目的

SFCBがHTTPSクライアント接続を受け入れるかどうかを指定します。デフォルトは `true` です。

構文

```
enableHttps: option
```

オプション	説明
true	HTTPS接続を有効にします。

オプション	説明
false	HTTPS接続を無効にします。

32.3.3.8 httpsProcs

目的

新しい着信HTTPS要求を拒否するまでの同時HTTPSクライアント接続の最大数を指定します。デフォルトは`8`です。

構文

```
httpsProcs: max_number_of_connections
```

32.3.3.9 enableInterOp

目的

SFCBで表示サポートにinterop名前空間を提供するかどうかを指定します。デフォルトは`true`です。

構文

```
enableInterOp: option
```

オプション	説明
true	interop名前空間を有効にします。
false	interop名前空間を無効にします。

32.3.3.10 provProcs

目的

同時プロバイダプロセスの最大数を指定します。この時点以降、新しい着信要求により新しいプロバイダのロードが必要になった場合は、既存のプロバイダのいずれかが最初に自動的にアンロードされます。デフォルトは `32` です。

構文

```
provProcs: max_number_of_procs
```

32.3.3.11 doBasicAuth

目的

要求を受け入れる前に、クライアントユーザーIDに基づいて基本認証のオンまたはオフを切り替えます。デフォルト値は `true` で、基本的なクライアント認証が実行されます。

構文

```
doBasicAuth: option
```

オプション	説明
true	基本認証を有効にします。
false	基本認証を無効にします。

32.3.3.12 basicAuthLib

目的

ローカルライブラリ名を指定します。SFCBサーバは、クライアントユーザIDを認証するためにライブラリをロードします。デフォルトは `sfcBasicPAMAuthentication` です。

構文

`provProcs:` `max_number_of_procs`

32.3.3.13 useChunking

目的

このオプションは、HTTP/HTTPSの「チャンク」使用のオンまたはオフを切り替えます。オンに切り替えた場合、サーバは大量の応答データを、バッファして1つの「チャンク」ですべてを返信するのではなく、小さなチャンクでクライアントに返信します。デフォルトは `true` です。

構文

`useChunking:` `option`

オプション	説明
true	HTTP/HTTPSデータチャンクを有効にします。
false	HTTP/HTTPSデータチャンクを無効にします。

32.3.3.14 keepaliveTimeout

目的

1つの接続で、2つの要求の間、要求がなされてから接続を閉じるまでSFCB　HTTPプロセスが待機する最大時間を秒数で指定します。`0` に設定すると、HTTP　keep-aliveが無効になります。デフォルトは `0` です。

構文

`keepaliveTimeout:` `secs`

32.3.3.15 keepaliveMaxRequest

目的

1つの接続で連続して受け付ける要求の最大数を指定します。`0`に設定すると、HTTP keep-aliveが無効になります。デフォルト値は`10`です。

構文

```
keepaliveMaxRequest: number_of_connections
```

32.3.3.16 registrationDir

目的

プロバイダの登録データ、ステージング領域、および静的リポジトリを含む登録ディレクトリを指定します。デフォルトは`/var/lib/sfcb/registration`です。

構文

```
registrationDir: dir
```

32.3.3.17 providerDirs

目的

SFCBがプロバイダライブラリを検索するディレクトリのリストをスペースで区切って指定します。デフォルトは`/usr/lib64 /usr/lib64 /usr/lib64/cmpi`です。

構文

```
providerDirs: dir
```

32.3.3.18 providerSampleInterval

目的

プロバイダマネージャが待機中のプロバイダをチェックする間隔を秒で指定します。デフォルトは `30` です。

構文

```
providerSampleInterval: secs
```

32.3.3.19 providerTimeoutInterval

目的

待機中のプロバイダがプロバイダマネージャによりアンロードされるまでの間隔を秒で指定します。デフォルトは `60` です。

構文

```
providerTimeoutInterval: secs
```

32.3.3.20 providerAutoGroup

目的

プロバイダ登録ファイルで他のグループを指定しておらず、このオプションを `true` に設定されている場合、同じ共有ライブラリのすべてのプロバイダが同じプロセス内で実行されます。

構文

```
providerAutoGroup: option
```

オプション	説明
true	プロバイダのグループを有効にします。
false	プロバイダのグループを無効にします。

32.3.3.21 sslCertificateFilePath

目的

サーバ証明書を含むファイルの名前を指定します。ファイルは、PEM (Privacy Enhanced Mail、RFC 1421、およびRFC 1424)フォーマットであることが必要です。このファイルは、`enableHttps`が`true`に設定されている場合にのみ必要です。デフォルトは`/etc/sfcb/server.pem`です。

構文

```
sslCertificateFilePath: path
```

32.3.3.22 sslKeyFilePath

目的

サーバ証明書の秘密鍵が含まれるファイルの名前を指定します。このファイルはPEMフォーマットであることが必要であり、パスフレーズによって保護できない場合があります。このファイルは、`enableHttps`が`true`に設定されている場合にのみ必要です。デフォルトは`/etc/sfcb/file.pem`です。

構文

```
sslKeyFilePath: path
```

32.3.3.23 sslClientTrustStore

目的

CAまたはクライアントの自己署名付きの証明書のいずれかを含むファイルの名前を指定します。このファイルはPEMフォーマットであることが必要であり、`sslClientCertificate`が`accept`または`require`に設定されている場合にのみ必要になります。デフォルトは`/etc/sfcb/client.pem`です。

構文

`sslClientTrustStore:` `path`

32.3.3.24 sslClientCertificate

目的

SFCBがクライアント証明書に基づく認証を処理する方法を指定します。`ignore`に設定した場合、クライアントに証明書は要求されません。`accept`に設定した場合、クライアントに証明書が要求されますが、クライアントが証明書を提示しなくとも失敗しません。`require`に設定した場合、クライアントが証明書を提示しないときは、クライアント接続が拒否されます。デフォルト値は`ignore`です。

構文

`sslClientCertificate:` `option`

オプション	説明
ignore	クライアント証明書の要求を無効にします。
承諾	クライアント証明書の要求を無効にします。 証明書が存在しなくとも失敗しません。
必要	有効な証明書を持たないクライアント接続を拒否します。

32.3.3.25 certificateAuthLib

目的

クライアント証明書に基づいてユーザ認証を要求するローカルライブラリの名前を指定します。この設定は、`sslClientCertificate`が`ignore`に設定されていない場合にのみ必要です。デフォルト値は`sfcCertificateAuthentication`です。

構文

```
certificateAuthLib: file
```

32.3.3.26 traceLevel

目的

SFCBのトレースレベルを指定します。この設定は、環境変数`SFCB_TRACE_LEVEL`を設定することにより上書きできます。デフォルト値は`0`です。

構文

```
traceLevel: num_level
```

32.3.3.27 traceMask

目的

SFCBのトレースマスクを指定します。この設定は、コマンドラインオプション`--trace-components`で上書きできます。デフォルト値は`0`です。

構文

```
traceMask: mask
```

32.3.3.28 traceFile

目的

SFCBのトレースファイルを指定します。この設定は、環境変数`SFCB_TRACE_LEVEL`を設定することにより上書きできます。デフォルト値は、`stderr`(標準エラー出力)です。

構文

```
traceFile: output
```

32.4 高度なSFCBタスク

この章では、SFCBの使用方法に関連するより高度なトピックを取り上げます。このトピックを理解するには、Linuxファイルシステムの基礎知識とLinuxコマンドラインの使用経験が必要です。この章には、次のタスクが含まれています。

- CMPIプロバイダのインストール
- SFCBのテスト
- `wbemcli` CIMクライアントの使用

32.4.1 CMPIプロバイダのインストール

CMPIプロバイダをインストールするには、`providerDirs`設定オプションにより指定されたいずれかのディレクトリに共有ライブラリがコピーされていることを確認する必要があります。32.3.3.17項「providerDirs」を参照してください。プロバイダはまた、`sfcbstage`コマンドおよび`sfcbrepos`コマンドを使用して適切に登録されていることが必要です。

プロバイダパッケージは通常、SFCB用に準備されます。したがって、インストールにより適切な登録が行われます。大半のSBLIMプロバイダは、SFCB用に準備されています。

32.4.1.1 クラスリポジトリ

クラスリポジトリは、SFCBがCIMクラスに関する情報を保存する場所です。通常これは、名前空間コンポーネントから成るディレクトリツリーから構成されます。一般的なCIM名前空間は `root/cimv2` または `root/interop` であり、ファイルシステム上のクラスリポジトリディレクトリパスにそれぞれ変換されます。

`/var/lib/sfcb/registration/repository/root/cimv2`

および

`/var/lib/sfcb/registration/repository/root/interop`

各名前空間ディレクトリには、ファイル `classSchemas` が含まれます。ファイルには、その名前空間の下に登録されたすべてのCIMクラスのコンパイル済みバイナリ表現があります。また、CIMスーパークラスに関して必要な情報も含まれます。

さらに各名前空間ディレクトリには、名前空間のすべての修飾子を含むファイル `修飾子` が含まれます。sfcbdの再起動時に、クラスプロバイダはディレクトリ `/var/lib/sfcb/registration/repository/` およびそのすべてのサブディレクトリをスキャンして、登録済みの名前空間を決定します。次に、`classSchemas` ファイルがデコードされ、各名前空間のクラス階層が構築されます。

32.4.1.2 新しいクラスの追加

SFCBは、ライブCIMクラス操作を生成できません。クラスをオフラインで追加、変更、または削除し、`systemctl restart sfcb.service` でSFCBサービスを再起動して変更内容を登録します。

SFCBは、プロバイダクラスおよび登録情報を保存するために、「ステージング領域」と呼ばれる場所を使用します。SUSE® Linux Enterprise Serverシステムでは、これは `/var/lib/sfcb/stage/` の下にあるディレクトリ構造です。

新しいプロバイダを追加するには、次の操作が必要です。

- プロバイダクラス定義ファイルを、ステージング領域ディレクトリの `/mofs` サブディレクトリ(`/var/lib/sfcb/stage/mofs`)にコピーします。
- クラス(複数可)の名前およびプロバイダタイプを含む登録ファイル、および実行可能なライブラリファイルの名前を `/regs` サブディレクトリにコピーします。

ステージングディレクトリには、2つのデフォルト「mof」(クラス定義)ファイル(`indication.mof` と `interop.mof`)があります。ルートステージングディレクトリ `/var/lib/sfcb/stage/mofs` の下にあるMOFのファイルは、`sfcbrepos` コマンドの実行後に各名前空間にコピーされます。`interop.mof` は、interop名前空間に対してのみコンパイルされます。

ディレクトリレイアウトは、次の例のようになります。

```
tux@mercury:~> ls /var/lib/sfcb/stage
default.reg  mofs  regs
```

```
tux@mercury:~> ls /var/lib/sfcb/stage/mofs
indication.mof  root
```

```
tux@mercury:~> ls /var/lib/sfcb/stage/mofs/root
cimv2  interop  suse  virt
```

```
tux@mercury:~> ls -1 /var/lib/sfcb/stage/mofs/root/cimv2 | less
Linux_ABIParameter.mof
Linux_BaseIndication.mof
Linux_Base.mof
Linux_DHCPElementConformsToProfile.mof
Linux_DHCPEntity.mof
[..]
OMC_StorageSettingWithHints.mof
OMC_StorageVolumeDevice.mof
OMC_StorageVolume.mof
OMC_StorageVolumeStorageSynchronized.mof
OMC_SystemStorageCapabilities.mof
```

```
tux@mercury:~> ls -1 /var/lib/sfcb/stage/mofs/root/interop
ComputerSystem.mof
ElementConformsToProfile.mof
HostSystem.mof
interop.mof
Linux_DHCPElementConformsToProfile.mof
[..]
OMC_SMIElementSoftwareIdentity.mof
OMC_SMISubProfileRequiresProfile.mof
OMC_SMIVolumeManagementSoftware.mof
ReferencedProfile.mof
RegisteredProfile.mof
```

```
tux@mercury:~> ls -1 /var/lib/sfcb/stage/regs
AllocationCapabilities.reg
```

```
Linux_ABIParameter.reg
Linux_BaseIndication.reg
Linux_DHCPGlobal.reg
Linux_DHCPRegisteredProfile.reg
[..]
OMC_Base.sfcb.reg
OMC_CopyServices.sfcb.reg
OMC_PowerManagement.sfcb.reg
OMC_Server.sfcb.reg
RegisteredProfile.reg
```

```
tux@mercury:~>  cat /var/lib/sfcb/stage/regs/Linux_DHCPRegisteredProfile.reg
[Linux_DHCPRegisteredProfile]
   provider: Linux_DHCPRegisteredProfileProvider
   location: cmpiLinux_DHCPRegisteredProfile
   type: instance
   namespace: root/interop
#
[Linux_DHCPElementConformsToProfile]
   provider: Linux_DHCPElementConformsToProfileProvider
   location: cmpiLinux_DHCPElementConformsToProfile
   type: instance association
   namespace: root/cimv2
#
[Linux_DHCPElementConformsToProfile]
   provider: Linux_DHCPElementConformsToProfileProvider
   location: cmpiLinux_DHCPElementConformsToProfile
   type: instance association
   namespace: root/interop
```

SFCBは、各プロバイダについてカスタムプロバイダ登録ファイルを使用します。

注記: SBLIMプロバイダ登録ファイル

SBLIM Webサイト上のすべてのSBLIMプロバイダには、すでに、SFCB用の.regファイルを生成するための登録ファイルが含まれています。

SFCB登録ファイルのフォーマットは次のとおりです。

```
[<class-name>]
   provider: <provide-name>
   location: <library-name>
   type: [instance] [association] [method] [indication]
   group: <group-name>
   unload: never
   namespace: <namespace-for-class> ...
```

ここで:

`<class-name>`
: CIMクラス名(必須)

`<provider-name>`
: CMPIプロバイダ名(必須)

`<location-name>`
: プロバイダライブラリ名(必須)

`type`
: プロバイダのタイプ(必須)。これは、`instance`、`association`、`method`、または`indication`の任意の組み合わせです。

`<group-name>`
: 複数のプロバイダをグループ化し、単一のプロセスの下で実行することで、さらにランタイムリソースを最小化できます。同じ<group-name>の下で登録されたすべてのプロバイダは、同じプロセスの下で実行します。デフォルトでは、各プロバイダは別個のプロセスとして実行します。

`unload`
: プロバイダのアンロードポリシーを指定します。現在サポートされている唯一のオプションは`never`であり、これはプロバイダが待機時間について監視されず、決してアンロードされないことを指定します。デフォルトでは、待機時間が環境設定ファイルで指定された値を超えたときに各プロバイダがアンロードされます。

`namespace（ネームスペース）`
: このプロバイダが実行できる名前空間のリストです。この設定は必須ですが、大半のプロバイダで`root/cimv2`になります。

すべてのクラス定義およびプロバイダ登録ファイルがステージング領域に保存されたら、コマンド`sfcbrepos -f`でSFCBクラスリポジトリを再構築する必要があります。

このようにしてクラスの追加、変更、または削除を行うことができます。クラスリポジトリを再構築した後、コマンド`systemctl restart sfcb.service`でSFCBを再起動します。

またSFCBパッケージには、プロバイダクラスmofファイルおよび登録ファイルを、ステージング領域の適切な場所にコピーするユーティリティが含まれています。

```
sfcbstage -r [provider.reg] [class1.mof] [class2.mof] ...
```

このコマンドを実行した後、さらにクラスリポジトリを再構築し、SFCBサービスを再起動する必要があります。

32.4.2 SFCBのテスト

SFCBパッケージには、2つのテストスクリプト(`wbemcat`と`xmltest`)が含まれます。

`wbemcat`は、未加工のCIM-XMLデータをHTTPプロトコル経由で、ポート5988上でリスンする指定されたSFCBホスト(デフォルトではlocalhost)に送信します。次に、返された結果を表示します。次のファイルには、標準的なEnumerateClasses要求のCIM-XML表現が含まれます。

```
<?xml version="1.0" encoding="utf-8"?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
  <MESSAGE ID="4711" PROTOCOLVERSION="1.0">
    <SIMPLEREQ>
      <IMETHODCALL NAME="EnumerateClasses">
        <LOCALNAMESPACEPATH>
          <NAMESPACE NAME="root"/>
          <NAMESPACE NAME="cimv2"/>
        </LOCALNAMESPACEPATH>
        <IPARAMVALUE NAME="ClassName">
          <CLASSNAME NAME=""/>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="DeepInheritance">
          <VALUE>TRUE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="LocalOnly">
          <VALUE>FALSE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="IncludeQualifiers">
          <VALUE>FALSE</VALUE>
        </IPARAMVALUE>
```

```
        <IPARAMVALUE NAME="IncludeClassOrigin">
          <VALUE>TRUE</VALUE>
        </IPARAMVALUE>
      </IMETHODCALL>
    </SIMPLEREQ>
  </MESSAGE>
</CIM>
```

SFCB　CIMOMにこの要求を送信すると、登録済みのプロバイダが存在するすべてのサポートクラスのリストが返されます。ファイルを`cim_xml_test.xml`として保存した場合を考えます。

```
tux@mercury:~> wbemcat cim_xml_test.xml | less
HTTP/1.1 200 OK
Content-Type: application/xml; charset="utf-8"
Content-Length: 337565
Cache-Control: no-cache
CIMOperation: MethodResponse

<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateClasses">
[..]
<CLASS NAME="Linux_DHCPParamsForEntity" SUPERCLASS="CIM_Component">
<PROPERTY.REFERENCE NAME="GroupComponent" REFERENCECLASS="Linux_DHCPEntity">
</PROPERTY.REFERENCE>
<PROPERTY.REFERENCE NAME="PartComponent" REFERENCECLASS="Linux_DHCPParams">
</PROPERTY.REFERENCE>
</CLASS>
</IRETURNVALUE>
</IMETHODRESPONSE>
</SIMPLERSP>
</MESSAGE>
</CIM>
```

表示されるクラスは、システムにインストールされているプロバイダに応じて異なります。

2番目のスクリプト`xmltest`もまた、未加工のCIM-XMLテストファイルをSFCB CIMOMに送信するために使用されます。次に、以前に保存された「良好な」結果ファイルに対して、返された結果を比較します。対応する「良好」なファイルがまだ存在しない場合は、後から使用できるように作成されます。

```
tux@mercury:~> xmltest cim_xml_test.xml
Running test cim_xml_test.xml ... OK
 Saving response as cim_xml_test.OK
tux@mercury:~> xmltest cim_xml_test.xml
Running test cim_xml_test.xml ... Passed
```

32.4.3 コマンドラインCIMクライアント: wbemcli

SBLIMプロジェクトには、`wbemcat`および`xmltest`に加えて、より高度なコマンドラインCIMクライアントである`wbemcli`が含まれます。このクライアントは、SFCBサーバにCIM要求を送信し、返された結果を表示するために使用されます。これはCIMOMライブラリに依存せず、WBEMに準拠するすべての実装で使用できます。

たとえば、SFCBに登録済みのSBLIMプロバイダにより実装されたすべてのクラスを表示する必要がある場合は、「EnumerateClasses」(ec)要求をSFCBに送信します。

```
tux@mercury:~> wbemcli -dx ec http://localhost/root/cimv2
To server: <?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0"><SIMPLEREQ><IMETHODCALL \
    NAME="EnumerateClasses"><LOCALNAMESPACEPATH><NAMESPACE NAME="root"> \
    </NAMESPACE><NAMESPACE NAME="cimv2"></NAMESPACE> \
    </LOCALNAMESPACEPATH>
<IPARAMVALUE NAME="DeepInheritance"><VALUE>TRUE</VALUE> \
    </IPARAMVALUE>
<IPARAMVALUE NAME="LocalOnly"><VALUE>FALSE</VALUE></IPARAMVALUE>
<IPARAMVALUE NAME="IncludeQualifiers"><VALUE>FALSE</VALUE> \
    </IPARAMVALUE>
<IPARAMVALUE NAME="IncludeClassOrigin"><VALUE>TRUE</VALUE> \
    </IPARAMVALUE>
</IMETHODCALL></SIMPLEREQ>
</MESSAGE></CIM>
From server: Content-Type: application/xml; charset="utf-8"
From server: Content-Length: 337565
```

```
From server: Cache-Control: no-cache
From server: CIMOperation: MethodResponse
From server: <?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateClasses">
<IRETURNVALUE>
<CLASS NAME="CIM_ResourcePool" SUPERCLASS="CIM_LogicalElement">
<PROPERTY NAME="Generation" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="ElementName" TYPE="string">
</PROPERTY>
<PROPERTY NAME="Description" TYPE="string">
</PROPERTY>
<PROPERTY NAME="Caption" TYPE="string">
</PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime">
</PROPERTY>
[..]
<CLASS NAME="Linux_ReiserFileSystem" SUPERCLASS="CIM_UnixLocalFileSystem">
<PROPERTY NAME="FSReservedCapacity" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="TotalInodes" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="FreeInodes" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="ResizeIncrement" TYPE="uint64">
<VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="IsFixedSize" TYPE="uint16">
<VALUE>0</VALUE>
</PROPERTY>
[..]
```

`-dx` オプションでは、`wbemcli` でSFCBに送信された実際のXMLも、受信した実際のXMLも表示されます。上記の例では、多数返されるクラスのうちの第1のクラスが `CIM_ResourcePool`、第2のクラスが `Linux_ReiserFileSystem`.です。他の登録済みの全クラスでも、同様のエントリが表示されます。

`-dx` オプションを省略した場合、`wbemcli` は返却されたデータのコンパクト表現のみを表示します。

```
tux@mercury:~> wbemcli ec http://localhost/root/cimv2
localhost:5988/root/cimv2:CIM_ResourcePool Generation=,ElementName=, \
    Description=,Caption=,InstallDate=,Name=,OperationalStatus=, \
    StatusDescriptions=,Status=,HealthState=,PrimaryStatus=, \
    DetailedStatus=,OperatingStatus=,CommunicationStatus=,InstanceID=, \
    PoolID=,Primordial=,Capacity=,Reserved=,ResourceType=, \
    OtherResourceType=,ResourceSubType=, \AllocationUnits=
localhost:5988/root/cimv2:Linux_ReiserFileSystem FSReservedCapacity=, \
    TotalInodes=,FreeInodes=,ResizeIncrement=,IsFixedSize=,NumberOfFiles=, \
    OtherPersistenceType=,PersistenceType=,FileSystemType=,ClusterSize=, \
    MaxFileNameLength=,CodeSet=,CasePreserved=,CaseSensitive=, \
    CompressionMethod=,EncryptionMethod=,ReadOnly=,AvailableSpace=, \
    FileSystemSize=,BlockSize=,Root=,Name=,CreationClassName=,CSName=, \
    CSCreationClassName=,Generation=,ElementName=,Description=,Caption=, \
    InstanceID=,InstallDate=,OperationalStatus=,StatusDescriptions=, \
    Status=,HealthState=,PrimaryStatus=,DetailedStatus=,OperatingStatus= \
    ,CommunicationStatus=,EnabledState=,OtherEnabledState=,RequestedState= \
    ,EnabledDefault=,TimeOfLastStateChange=,AvailableRequestedStates=, \
    TransitioningToState=,PercentageSpaceUse=
    [..]
```

32.5 詳細情報

WBEMおよびSFCBの詳細については、次の資料を参照してください。

http://www.dmtf.org
: Distributed Management Task Force Webサイト

http://www.dmtf.org/standards/wbem/
: Webベースの企業管理(WBEM) Webサイト

http://www.dmtf.org/standards/cim/
: 共通情報モデル(CIM) Webサイト

http://sblim.wiki.sourceforge.net/
: Standards Based Linux Instrumentation (SBLIM) Webサイト

http://sblim.wiki.sourceforge.net/Sfcb
: Small Footprint CIM Broker (SFCB) Webサイト

http://sblim.wiki.sourceforge.net/Providers
: SBLIMプロバイダパッケージ

IV モバイルコンピュータ

33 Linuxでのモバイルコンピューティング

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、そしてこれらを使ったデータ交換ではないでしょうか。外付けハードディスク、フラッシュディスク、デジタルカメラなどのモバイルハードウェアコンポーネントは、ラップトップやデスクトップシステムに接続できます。多くのソフトウェアコンポーネントで、モバイルコンピューティングを想定しており、一部のアプリケーションは、モバイル使用に合わせて特別に作成されています。

33.1 ラップトップ

ラップトップのハードウェアは通常のデスクトップシステムとは異なります。これは交換可能性、空間要件、電力消費などの基準を考慮する必要があるためです。モバイルハードウェアメーカーは、ラップトップハードウェアを拡張するために使用可能なPCMCIA(Personal Computer Memory Card International Association)、Mini PCI、Mini PCIeなどの標準インタフェースを開発してきました。この標準はメモリカード、ネットワークインタフェースカード、および外部ハードディスクをカバーします。

33.1.1 電源消費量

ラップトップの製造時、消費電力を最適化したシステムコンポーネントを組み込むことで、電源に接続しなくてもシステムを快適に使用できるようにしています。電源の管理に関するこうした貢献は少なくともオペレーティングシステムの貢献度と同じくらい重要です。SUSE® Linux Enterprise Serverはラップトップの電源消費量に影響するさまざまな方法をサポートすることで、バッテリー使用時の動作時間に数々の効果をあげています。次のリストでは電源消費量節約への貢献度の高い順に各項目を示します。

- CPUの速度を落とす。
- 休止中にディスプレイの照明を切る。
- ディスプレイの明るさを手動で調節する。
- ホットプラグ対応の使用していないアクセサリを切断する(USB CD-ROM、外付けマウス、使用していないPCMCIAカード、Wi-Fi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 Enterprise Serverでの電源管理の詳細な背景情報は、第34章 電源管理に示されています。

33.1.2 操作環境の変化の統合

モバイルコンピューティングに使用する場合、ご使用のシステムを操作環境の変化に順応させる必要があります。多くのサービスは環境に依存するので、環境を構成するクライアントの再設定が必要です。SUSE Linux Enterprise Serverがこのタスクを処理します。

図 33.1 既存環境でのモバイルコンピュータの統合

スモールホームネットワークとオフィスネットワーク間でラップトップを持ち運びする場合に影響のあるサービスは次のとおりです。

ネットワーク

IPアドレスの割り当て、名前解決、インターネット接続、およびその他のネットワークへの接続が含まれます。

印刷
: 使用可能なプリンタの現在のデータベース、および使用可能なプリントサーバが、ネットワークに応じて表示されなければなりません。

電子メールとプロキシ
: 印刷と同様、現在の環境に対応するサーバが表示されなければなりません。

X(グラフィック環境)
: ご使用のラップトップがプロジェクタまたは外付けモニタに一時的に接続されている場合、別のディスプレイ設定が使用可能になっている必要があります。

SUSE Linux Enterprise Serverではラップトップを既存の操作環境に統合させる複数の方法を提供しています。

NetworkManager
: NetworkManagerは、特にラップトップでのモバイルネットワーキング用に調整されています。NetworkManagerは、ネットワーク環境間、またはモバイルブロードバンド(GPRS、EDGE、または3G)、ワイヤレスLAN、Ethernetなどのさまざまなタイプのネットワーク間を容易に、自動的に切り替える方法を提供します。NetworkManagerは、ワイヤレスLANでのWEPおよびWPA-PSKの暗号化をサポートします。また、ダイアルアップ接続もサポートします。GNOMEデスクトップには、NetworkManagerのフロントエンドが含まれています。詳細については、24.3項「ネットワーク接続の設定」を参照してください。

表 33.1 NETWORKMANAGERの使用

コンピュータの条件	NetworkManagerを使用する
ラップトップである	対応
別のネットワークに接続される場合がある	対応
ネットワークサービスを提供する(DNSまたはDHCP)	非対応
スタティックIPアドレスのみを使用する	非対応

NetworkManagerがネットワーク設定を扱うのが適切でない場合、YaSTツールを使用してネットワークを設定します。

ヒント: DNS設定と、各種ネットワーク接続

ラップトップを持って移動し、ネットワーク接続の種類を頻繁に変更する場合、すべてのDNSアドレスがDHCPによって正しく割り当てられていれば、NetworkManagerは正常に機能します。一部の接続で静的DNSアドレスを使用する場合は、そのアドレスを`/etc/sysconfig/network/config`内の`NETCONFIG_DNS_STATIC_SERVERS`オプションに追加します。

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も使用でき、適切なインストールソースの検索作業が最小化されます。SLPの詳細については、第20章 SLPを参照してください。

33.1.3 ソフトウェアオプション

モバイル用途には、専用ソフトウェアにより対応されるシステムモニタリング(特にバッテリの充電)、データ同期、周辺機器との無線通信、インターネットなど、さまざまな特別タスク領域が存在します。以降のセクションでは、SUSE Linux Enterprise Server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。

33.1.3.1 システムモニタリング

SUSE Linux Enterprise Serverでは2種類のシステムモニタリングツールを提供しています。

電源管理

[電源管理]は、GNOMEデスクトップの省エネルギー関係の動作を調整できるアプリケーションです。通常は、[コンピュータ] › [コントロールセンター] › [システム] › [電源管理]を介してアクセスします。

システムモニタ

[システムモニタ]は、測定可能なシステムパラメータを1つのモニタリング環境に集めます。このモニタは、デフォルトでは、3つのタブに出力情報を表示します。[プロセス]は、CPUロード、メモリ使用量、プロセスのID番号と優先度など、現在実行中のプロセスの詳細情報を提供します。収

集されたデータの表示とフィルタリングをカスタマイズできます。新しいタイプのプロセス情報を追加するには、プロセステーブルのヘッダを左クリックして、隠したい列やビューに追加したい列を選択します。さまざまなデータページで各種のシステムパラメータを監視したり、ネットワーク上でさまざまなマシンにあるデータを並行して収集したりすることも可能です。[リソース]タブには、CPU、メモリ、およびネットワークの履歴のグラフが表示され、[ファイルシステム]タブにはすべてのパーティションとその使用量が一覧にされます。

33.1.3.2 データの同期化

ネットワークから切断されたモバイルマシンと、オフィスのネットワーク上にあるワークステーションの両方で作業を行う場合、すべての場合で処理したデータを同期しておくことが必要になります。これには電子メールフォルダ、ディレクトリ、個別の各ファイルなど、オフィスでの作業時と同様、オフィス外で作業する場合にも必要となるものが含まれます。両方の場合のソリューションを次に示します。

電子メールの同期化

オフィスネットワークで電子メールを保存するためにIMAPアカウントを使用します。これで電子メールは、Mozilla Thunderbird MailやEvolutionなどの切断型IMAP対応電子メールクライアントを使用するワークステーションからアクセスできるようになります。**送信メッセージ**で常に同じフォルダを使用するには、電子メールクライアントでの設定が必要になります。また、この機能により、同期プロセスが完了した時点でステータス情報とともにすべてのメッセージが使用可能になります。未送信メールについての信頼できるフィードバックを受信するためには、システム全体で使用されるMTA　postfixまたはsendmailの代わりに、メッセージ送信用のメールクライアントに実装されたSMTPサーバーを使用します。

ファイルとディレクトリの同期

ラップトップとワークステーション間のデータの同期に対応するユーティリティが複数あります。最もよく使用されるものには、`rsync`というコマンドラインツールがあります。詳細については、そのマニュアルページを参照してください(`man 1 rsync`)。

33.1.3.3 ワイヤレス通信: Wi-Fi

Wi-Fiは、これらのワイヤレステクノロジの中では最大規模で、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。個々のマシンを相互に接続して、独立したワイヤレスネットワークを構築することも、インターネットにアクセスすることも可能です。「アクセスポイント」と呼ばれるデバイスがWi-Fi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アクセスの中継点としての役目を果たします。モバイルユーザは、場所や、どのアクセスポイントが最適な接続を提供するかに応じてさまざまなアクセスポイントを切り替えることができます。Wi-Fi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。

Wi-Fiカードは、IEEEが策定した802.11標準を使用して通信します。当初、この規格は最大伝送速度　2MBit/sについて提供されましたが、その後、データ伝送速度を高めるために複数の補足事項が追加されています。これらの補足事項では、モジュレーション、伝送出力、および伝送速度などの詳細が定義されています(表33.2「各種Wi-Fi規格の概要」参照)。さらに、多数の企業が専有権またはドラフト機能を持つハードウェアを実装しています。

表 33.2 各種WI-FI規格の概要

名前(802.11)	周波数(GHz)	最大伝送速度(MBit/s)	メモ
a	5	54	干渉が少ない
b	2.4	11	あまり普及せず
g	2.4	54	広く普及、11bと後方互換
n	2.4および/または5	300	Common(通常のネットワーキング)
ac	5	最大865	2015年には一般的になると予測される
ad	60	最大約7000	2012年にリリースされ、現時点ではあまり一般的でない。SUSE Linux Enterprise Serverでは未サポート

802.11レガシカードは、SUSE® Linux Enterprise Serverではサポートされていません。802.11a/b/g/nを使用する大半のカードはサポートされています。通常、新しいカードは　802.11n規格に準拠していますが、802.11gを使用するカードもまだあります。

33.1.3.3.1　動作モード

ワイヤレスネットワークでは、高速で高品質、そして安全な接続を確保するために、さまざまなテクニックや設定が使用されています。通常、Wi-Fiカードは「管理モード」で動作します。ただし、動作タイプごとに異なる設定が必要です。基本的に、ワイヤレスネットワークは次の4つのネットワークモードに分類できます。

アクセスポイントを経由する管理モード(インフラストラクチャモード) (デフォルトモード)

管理ネットワークには、管理要素として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またはデフォルトモードとも呼ばれます)では、ネットワーク内のWi-Fi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。権限のある局だけが接続できるようにするため、さまざまな認証メカニズム(WPAなど)が使用されます。これは、消費エネルギー量が最小のメインモードでもあります。

アドホックモード(ピアツーピアネットワーク)

アドホックネットワークには、アクセスポイントはありません。アドホックネットワークでは、局同士が直接に通信するので、通常、アドホックネットワークは管理ネットワークより低速です。ただし、アドホックネットワークでは、参加局の伝送範囲と数が大幅に制限されます。それらのネットワークでは、WPA認証もサポートしません。WPAセキュリティを使用する場合は、アドホックモードを使用しないでください。すべてのカードがアドホックモードを確実にサポートするとは限らない点に注意してください。

マスタモード

マスタモードでは、使用中のWi-Fiカードがマスタモードをサポートしていることを前提に、Wi-Fiカードをアクセスポイントとして使用します。Wi-Fiカードの詳細については、http://linux-wless.passys.nl を参照してください。

メッシュモード

ワイヤレスメッシュネットワークは、「メッシュ型トポロジ」で編成されます。ワイヤレスメッシュネットワークの接続はすべてのワイヤレスメッシュ「ノード」に分散されます。このネットワークに属する各ノードは他のノードに接続して接続を共有します。これは広域に渡って行われる可能性があります(SLE12ではサポートされていません)。

33.1.3.3.2 認証

有線ネットワークよりもワイヤレス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。

旧式のWi-FiカードはWEP (Wired Equivalent Privacy)のみをサポートしています。ただし、WEPは安全でないことが判明したので、Wi-Fi業界はWPAという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。WPA (WPA2と同義の場合もあります)をデフォルトの認証方式にする必要があります。

通常、ユーザは認証方式を一切選択できません。たとえば、カードが管理モードで動作している場合、認証はアクセスポイントによって設定されます。認証方法はNetworkManagerに表示されます。

33.1.3.3.3 暗号化

権限のないユーザが無線ネットワークで交換されるデータパケットを読み込んだりネットワークにアクセスしたりできないように、さまざまな暗号化方式が存在しています。

WEP (IEEE 802.11で定義)

この規格では、RC4暗号化アルゴリズムを使用します。当初のキー長は40ビットでしたが、その後104ビットも使用されています。通常、初期化ベクタの24ビットを含めるものとして、長さは64ビットまたは128ビットとして宣言されます。ただし、この規格には一部弱点があります。このシステムで生成されたキーに対する攻撃が成功する場合があります。それでも、ネットワークをまったく暗号化しないよりはWEPを使用する方が適切です。

非標準の「ダイナミックWEP」を実装しているベンダーもいます。これは、WEPとまったく同様に機能し、同じ弱点を共有しますが、キーがキー管理サービスによって定期的に変更されます。

TKIP (WPA/IEEE 802.11iで定義)

このキー管理プロトコルはWPA規格で定義されており、WEPと同じ暗号化アルゴリズムを使用しますが、弱点は排除されています。データパケットごとに新しいキーが生成されるので、これらのキーに対する攻撃は無駄になります。TKIPはWPA-PSKと併用されます。

CCMP (IEEE 802.11iで定義)

CCMPは、キー管理を記述したものです。通常は、WPA-EAPに関連して使用されますが、WPA-PSKとも併用できます。暗号化はAESに従って行われ、WEP規格のRC4暗号化よりも厳密です。

33.1.3.4 ワイヤレス通信: 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。また範囲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、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Wi-Fiテクノロジが適しています。

33.1.3.5 ワイヤレス通信: 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。受信者へのファイルの長距離送信はモバイルネットワークで処理します。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。

33.1.4 データのセキュリティ

無認証のアクセスに対し、複数の方法でラップトップ上のデータを保護するのが理想的です。実行可能なセキュリティ対策は次の領域になります。

盗難からの保護

常にシステムを物理的な盗難から守ることを心がけます。チェーンなど、さまざまな防犯ツールが小売店で販売されています。

強力な認証

ログインとパスワードによる標準の認証に加えて、生体認証を使用します。SUSE Linux Enterprise Serverは、指紋認証をサポートしています。

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 Enterprise Serverでの暗号化パーティションの作成については、Book “Security Guide” 11 “Encrypting Partitions and Files”に記載されています。また、YaSTによりユーザを追加するときに暗号化されたホームディレクトリを作成できます。

重要: データのセキュリティとディスクへのサスペンド

暗号化パーティションは、ディスクへのサスペンドのイベントの際にもアンマウントされません。それで、これらのパーティション上のデータは、ハードウェアが盗まれた場合、ハードディスクのレジュームを行うことで、誰にでも入手できるようになります。

ネットワークセキュリティ

データの転送には、転送方法に関わらず、セキュリティ保護が必要です。Linuxおよびネットワークに関する一般的なセキュリティ問題については、Book “Security Guide” 1 “Security and Confidentiality”を参照してください。

33.2 モバイルハードウェア

SUSE Linux Enterprise ServerはFireWire (IEEE 1394)またはUSB経由のモバイルストレージデバイスを自動検出します。モバイルストレージデバイスという用語は、FireWire、USBハードディスク、USBフラッシュドライブ、デジタルカメラのいずれにも適用されます。これらのデバイスは、対応するインタフェースを介してシステムに接続されるとすぐに自動的に検出されて設定されます。GNOMEのファイルマネージャは、モバイルハードウェアアイテムを柔軟に処理します。これらのメディアを安全にアンマウントするには、ファイルマネージャの［ボリュームのマウント解除］(GNOME)機能を使用します。

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ファイルマネージャに表示されます。アイコンをクリックすると、ドライブの内容が表示されます。ここでディレクトリやファイルの作成および編集、削除を実行できます。システムによって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、対応するメニュー項目を選択します。この名前変更はファイルマネージャでの表示に限られています。`/media`にマウントされているデバイスのデスクリプタは、これには影響されません。

USBフラッシュディスク

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。

33.3 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33.1.3.3項「ワイヤレス通信: Wi-Fi」ですでに説明しました。携帯電話側のこれらのプロトコルの設定はそれぞれのマニュアルに記載されています。

33.4 詳細情報

モバイルデバイスおよびLinuxに関連するすべてのお問い合わせはhttp://tuxmobil.org/を参照してください。このWebサイトでは、ラップトップのハードウェア、ソフトウェア、PDA、携帯電話、その他のモバイルハードウェアについて複数のセクションで取り扱います。

http://tuxmobil.org/ではhttp://www.linux-on-laptops.com/、と同様の内容について参照できます。ラップトップおよびハンドヘルドデバイスについての情報はここを参照してください。

SUSEはラップトップを主題としたドイツ語の専用メーリングリストを運営しています。http://lists.opensuse.org/opensuse-mobile-de/を参照してください。このリストではユーザと開発者がSUSE Linux Enterprise Serverでのモバイルコンピューティングに関するあらゆるテーマを話題にしています。英語での投稿には回答されますが、アーカイブされた情報のほとんどはドイツ語です。英語の投稿ではhttp://lists.opensuse.org/opensuse-mobile/を使用します。

34 電源管理

System z この章で説明する機能とハードウェアは、IBM System zには存在しないため、この章はこれらのプラットフォームに関係ありません。

電源管理はラップトップコンピュータで特に重要ですが、他のシステムでも役に立ちます。ACPI(Advanced Configuration and Power Interface)は、最近のすべてのコンピュータ(ラップトップ、デスクトップ、サーバ)で使用できます。電源管理テクノロジでは、適切なハードウェアとBIOSルーチンを必要とします。ほとんどのラップトップと多くの新型デスクトップおよびサーバは、これらの必要条件を満たしています。電源の節約や騒音の低減のために、CPU周波数を制御することもできます。

34.1 省電力機能

省電力機能はラップトップをモバイル使用する場合に限らず、デスクトップシステムでも重要です。ACPIの主要な機能と、その使用目的は、以下のとおりです。

スタンバイ
: サポートされていない.

サスペンド(メモリに保存)
: このモードでは、システム状態をすべてRAMに書き込みます。その後、RAMを除くシステム全体がスリープします。この状態では、コンピュータの消費電力が非常に小さくなります。この状態の利点は、ブートやアプリケーションの再起動をせずに、数秒でスリープ前の作業をスリープの時点から再開できることです。この機能は、ACPI状態 S3に対応します。

ハイバーネーション(ディスクに保存)
: この動作モードでは、システム状態がすべてハードディスクに書き込まれ、システムの電源がオフになります。すべてのアクティブデータを書き込むには、少なくともRAMの大きさのスワップパーティションが必要です。この状態から再開するには、30〜90秒かかります。サスペンド前の状態が復元されます。メーカの中には、このモードを便利なハイブリッド仕様にして提供するものもあります(たとえば、IBM ThinkpadのRediSafe)。対応するACPI状態は、S4です。Linux環境では、suspend to diskはACPIから独立したカーネルルーチンにより実行されます。

バッテリモニタ
: ACPI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、システムは、重要な充電ステータスに達した時点で実行するようにアクションを調整します。

自動電源オフ

シャットダウンの後、コンピュータの電源が切れます。これは、バッテリが空になる直前に自動シャットダウンが行われる場合に特に重要です。

プロセッサ速度の制御

CPUとの接続では、次の3つの方法で省エネできます: 周波数と電圧の調節(PowerNow!またはSpeedstep)、スロットリング、およびプロセッサをスリープ状態(C-states)にすること。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。

34.2 ACIP(詳細設定と電源インタフェース)

ACPI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し、制御できるように設計されています。ACPIは、PnP(Power Management Plug and Play)とAPM(Advanced Power Management)の両方に優先します。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、journaldにレポートされます。ジャーナルログメッセージの表示の詳細については、第11章 `journalctl`:systemdジャーナルのクエリを参照してください。ACPIに生じた問題のトラブルシューティングについては、34.2.2項「トラブルシューティング」を参照してください。

34.2.1 CPUパフォーマンスの制御

CPUには、3つの省エネ方法があります。

- 周波数と電圧の調節
- クロック周波数のスロットリング(T-states)
- プロセッサのスリープ状態への切り替え(C-states)

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。

周波数調節とスロットリングに意味があるのは、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、常に、最も経済的なC-stateが適用されるからです。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。通常、カーネルのオンデマンドガバナによって動的に制御される動的な周波数調節が最良のアプローチです。

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。

詳細については、Book “System Analysis and Tuning Guide” 10 “Power Management”を参照してください。

34.2.2 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=ht
: 単純なリソース設定のみを実行します。ACPIを他の目的には使用しません。

acpi=off
: ACPIを無効にします。

警告: ACPIなしに起動できない場合

一部の新型のコンピュータは(特に、SMPシステムとAMD64システム)、ハードウェアを正しく設定するためにACPIが必要です。これらのコンピュータでACPIを無効にすると、問題が生じます。

コンピュータは時折、USBまたはFireWireを介して接続されたハードウェアと混同されることがあります。コンピュータが起動を拒否した場合、必要のないハードウェアのプラグをすべてはずして再試行してください。

システムのブートメッセージを調べてみましょう。そのためには、ブート後にコマンド`dmesg -T | grep -2i acpi`を使用します(または、問題の原因がACPIだとは限らないので、すべてのメッセージを調べます)。ACPIテーブルの解析時にエラーが発生した場合は、最も重要なテーブルDSDT (Differentiated System Description Table)を改善されたバージョンと置き換えることができます。この場合、BIOSで障害のあるDSDTが無視されます。具体的な手順については34.4項「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およびインストールすると、詳細情報が発行されます。

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。

34.2.2.1 詳細情報

- http://tldp.org/HOWTO/ACPI-HOWTO/ (詳細なACPI HOWTO、DSDTパッチが含まれています)
- http://www.acpi.info (Advanced Configuration and Power Interface: 詳細設定と電源インタフェース)
- http://acpi.sourceforge.net/dsdt/index.php (Bruno DucrotによるDSDTパッチ)

34.3 ハードディスクの休止

Linux環境では、不要な場合にハードディスクを完全にスリープ状態にしたり、より経済的な静止モードで動作さることができます。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、最大限に省電力したい場合は、次の方法のいくつかを`hdparm`コマンドでテストしてください。

このコマンドを使用すると、各種のハードディスク設定を変更できます。`-y`オプションは、簡単にハードディスクをスタンバイモードに切り替えます。`-Y`を指定すると、スリープ状態になります。`hdparm -S x`を使用すると、一定時間アクティビティがなければハードディスクが回転を停止します。`x`は、次の

ように置換します。0を指定するとこの機構が無効になり、ハードディスクは常時稼働します。1から240までの値を指定すると、指定した値x　5秒が設定値になります。241から251は、30分の1倍から11倍(30分から5.5時間)に相当します。

ハードディスクの内部省電力オプションは、オプション-Bで制御できます。0　(最大限の省電力)～255　(最大限のスループット)の値を選択します。結果は使用するハードディスクに応じて異なり、査定するのは困難です。ハードディスクを静止状態に近づけるにはオプション-Mを使用します。128　(静止)～254　(高速)の値を選択します。

ハードディスクをスリープにするのは、多くの場合簡単ではありません。Linuxでは、多数のプロセスがハードディスクに書き込むので、ウェイクアップが常に繰り返されています。したがって、ハードディスクに書き込むデータを、Linuxがどのように処理するかを理解することは重要です。はじめに、すべてのデータがRAMにバッファされます。このバッファは、pdflushデーモンによって監視されます。データが一定の寿命に達するか、バッファがある程度一杯になると、バッファの内容がハードディスクにフラッシュされます。バッファサイズはダイナミックであり、メモリサイズとシステム負荷に対応して変化します。デフォルトでは、データの完全性を最大まで高めるように、pdflushの間隔が短く設定されています。pdflushデーモンはバッファを5秒おきにチェックし、データをハードディスクに書き込みます。次の変数が使用できます。

`/proc/sys/vm/dirty_writeback_centisecs`
: pdflushスレッドが起動するまでの遅延(100分の1秒台)を含みます。

`/proc/sys/vm/dirty_expire_centisecs`
: ダーティページが次に最新の変更を書き込まれるまでの時間枠を定義します。デフォルト値は3000(つまり 30秒)です。

`/proc/sys/vm/dirty_background_ratio`
: pdflushが書き込みを始めるまでのダーティページの最大割合。デフォルトは5パーセントです。

`/proc/sys/vm/dirty_ratio`
: メモリ全体の中でダーティページの割合がこの値を超えると、プロセスは書き込みを続けずに、短時間でダーティバッファを書き込むように強制されます。

警告: データの完全性に関する障害

pdflushデーモンの設定を変更すると、データの完全性が損なわれる可能性があります。

これらのプロセスとは別に、Btrfs、Ext3、Ext4などのジャーナリングファイルシステムは、それらが持つメタデータをpdflushとは無関係に書き込むので、ハードディスクがスピンダウンしなくなります。

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。

34.4 トラブルシューティング

すべてのエラーメッセージとアラートは、`journalctl` コマンドで問い合わせ可能なシステムジャーナルに記録されます(詳細については、第11章 `journalctl`:systemdジャーナルのクエリを参照してください)。以下のセクションでは、最も頻繁に起こる問題について解説します。

34.4.1 CPU周波数調節が機能しません。

カーネルのソースを参照して、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。`kernel-source` パッケージがインストールされている場合は、この情報を `/usr/src/linux/Documentation/cpu-freq/*` で入手できます。

34.5 その他の情報

- http://en.opensuse.org/SDB:Suspend_to_RAM ↗—「How to get Suspend to RAM working」
- http://old-en.opensuse.org/Pm-utils ↗—「How to modify the general suspend framework」

V トラブルシューティング

35 ヘルプとドキュメント

SUSE® Linux Enterprise Serverではさまざまな情報源とドキュメントが提供されており、その多くはインストール済みのシステムに統合されています。

`/usr/share/doc` 内のドキュメント
: この従来のヘルプディレクトリには、システムのさまざまなドキュメントファイルやリリースノートが格納されます。このディレクトリの `packages` サブディレクトリには、インストール済みパッケージの情報も含まれています。詳細については35.1項「ドキュメントディレクトリ」を参照 てください。

シェルコマンドのマニュアルページと情報ページ
: シェルを使用する場合は、コマンドのオプションを記憶しておく必要はありません。シェルは以前からマニュアルページおよび情報ページによって統合ヘルプを提供しています。詳細については35.2項「manページ」および35.3項「情報ページ」を参照してください。

デスクトップヘルプセンター
: GNOMEデスクトップのヘルプセンター([Help (ヘルプ)])では、システムの最も重要なドキュメントリソースに検索可能な形式で一元的にアクセスできます。これらのリソースにはインストール済みのアプリケーションのオンラインヘルプ、マニュアルページ、情報ページ、および製品に付属しているSUSEマニュアルが含まれます。

一部のアプリケーション用の別なヘルプパッケージ
: YaSTで新しいソフトウェアをインストールすると、ほとんどの場合、ソフトウェアのマニュアルも自動的にインストールされ、通常、デスクトップのヘルプセンターに表示されます。ただし、GIMPなどの一部のアプリケーションは、YaSTとは別個にインストールされる独自のオンラインヘルプパッケージを利用しており、ヘルプセンターには表示されない場合があります。

35.1 ドキュメントディレクトリ

インストールされたLinuxシステム上のドキュメント検索用の従来のディレクトリは、`/usr/share/doc` です。このディレクトリには通常、リリースノート、マニュアルなどに加えて、システムにインストールされたパッケージに関する情報が含まれます。

注記: インストール済みパッケージに依存する内容

Linuxの世界では、ソフトウェアのように、多くのマニュアルとその他の文書はパッケージ形式で用意されています。`/usr/share/docs` 内の情報の種類および内容は、インストールされている(文書)パッケージに応じて異なります。ここに記載されているサブディレクトリが見つからない場合は、対応するパッケージがシステムにインストールされているかどうかを確認し、必要に応じてYaSTに追加してください。

35.1.1 SUSEマニュアル

これらのガイドブックは、HTMLおよびPDFの各バージョンが複数の言語で提供されています。`manual` サブディレクトリには、製品で使用可能な大半のSUSEマニュアルのHTMLバージョンがあります。製品で使用可能なすべての文書の概要については、マニュアルの序文を参照してください。

複数の言語がインストールされている場合、`/usr/share/doc/manual` には異なる言語版のマニュアルが含まれる場合があります。SUSEマニュアルのHTMLバージョンは、両デスクトップのヘルプセンターでも利用可能です。インストールメディアでの文書のPDF版およびHTML版の検索場所については、SUSE Linux Enterprise Serverのリリースノートを参照してください。これらの文書は、インストールされたシステムの `/usr/share/doc/release-notes/` 、またはオンラインの製品固有のWebページ(http://www.suse.com/doc/)で参照できます。

35.1.2 パッケージのドキュメント

`packages` の下で、システムにインストールしたソフトウェアパッケージに含まれているドキュメントを見つけてください。各パッケージについて、サブディレクトリ `/usr/share/doc/packages/packagename` が作成されます。このサブディレクトリには、パッケージのREADMEファイルが含まれます。さらにサンプル、環境設定ファイル、または追加スクリプトが含まれることがあります。次のリストに、`/usr/share/doc/packages` の下にある一般的なファイルを示します。これらのエントリはいずれも必須ではなく、多くのパッケージがその一部のみを含みます。

`AUTHORS`
: 主な開発者のリスト。

`BUGS`
: 既知のバグまたは誤動作。また、Bugzilla Webページへのリンクがあり、そこでバグを検索できる場合があります。

`CHANGES`,
`ChangeLog`
: バージョン間の変更点の概要です。非常に詳細なものなので、通常は、開発者にとって興味あるものです。

`COPYING`,
`LICENSE`
: ライセンス情報。

`FAQ`
: メーリングリストやニュースグループから集められた質問と答えが含まれています。

`INSTALL`
: システムにこのパッケージをインストールする方法。このファイルに目を通している時点でパッケージがすでにインストールされており、このファイルの内容を無視しても問題はありません。

`README`, `README.*`
: ソフトウェアに関する一般的な情報。たとえば、ソフトウェアの目的および使用方法などです。

今後の課題
: まだ実装されていないものの、今後実装される予定の機能についての説明です。

`MANIFEST`
: ファイルのリストと、それぞれの簡単な概要です。

`NEWS`
: このバージョンでの新しい点が記されています。

35.2 manページ

マニュアルページは、どのLinuxシステムにおいても重要な役割を担っています。マニュアルページでは、コマンドと利用可能なオプションおよびパラメータについての使用法が説明されています。マニュアルページは、`man`の後にコマンド名(たとえば「`man ls`」)を入力して開くことができます。

マニュアルページは、シェルに直接表示されます。ナビゲートするには、`Page ↑`および`Page ↓`を使用して上下に移動します。<`Home`>キーと<`End`>キーを使用すると、それぞれドキュメントの最初と最後に移動できます。 `Q`キーを押すと、この表示モードが終了します。 `man`コマンド自体の詳細については、`man man`と入力します。マニュアルページは、表35.1「マニュアルページ—カテゴリと説明」(マニュアルページ自身から抽出)に示すように、カテゴリ別にソートされています。

表 35.1 マニュアルページーカテゴリと説明

数値	説明
1	実行可能プログラムまたはシェルコマンド
2	システムコール(カーネルによって提供される機能)
3	ライブラリコール(プログラムライブラリ内での機能)
4	特別なファイル(通常は `/dev` 内にあります)
5	ファイル形式と命名規則(`/etc/fstab`)
6	ゲーム
7	その他(マクロパッケージおよび規則)、例: man(7)、groff(7)
8	システム管理コマンド (通常は、 `root` の場合のみ)
9	カーネルルーチン(非標準)

各マニュアルページは、NAME、SYNOPSIS、DESCRIPTION、SEE ALSO、LICENSINGおよびAUTHORといういくつかのパートで構成されています。 コマンドのタイプによっては、他のセクションが追加されている場合があります。

35.3 情報ページ

情報ページは、システム上にあるもう1つの重要な情報ソースです。通常、情報ページの内容はマニュアルページよりも詳細です。これらはコマンドラインオプションよりも詳細な情報で構成され、チュートリアルやリファレンスマニュアル全体が含まれている場合もあります。特定のコマンドの情報ページを表示するには、 `info` の後にコマンド名(たとえば「 `info ls` 」)を入力します。シェルで直接ビューアを使用してinfoページを参照し、「ノード」と呼ばれるさまざまなセクションを表示できます。と呼ばれるさまざまなセクションを表示できます。 Space を使用して前に移動し、 <- を使用して後ろに移動します。

ノード内で、Page ↑ および Page ↓ を使用して参照することもできますが、前および後ろのノードにも移動できるのは Space および <– のみです。Q を押すと、表示モードを終了します。すべてのコマンドに情報ページが付属するわけではありません。逆も同様です。

35.4 リソースのオンライン化

オンラインバージョンのSUSEマニュアル(/usr/share/doc にインストールされます)に加えて、Webで製品固有のマニュアルやドキュメントにアクセスすることもできます。SUSE Linux Enterprise Server用に提供されているすべてのマニュアルの概要については、http://www.suse.com/doc/ にある製品ごとのマニュアルに関するWebページをご覧してください。

製品ごとの追加情報を検索する場合は、次のWebサイトも参照してください。

SUSEテクニカルサポート
: 質問がある場合や、技術的な問題について解決策が必要な場合、http://www.suse.com/support/ でSUSEテクニカルサポートを利用できます。

SUSEフォーラム
: SUSE製品に関して議論できるいくつかのフォーラムがあります。リストについては、http://forums.suse.com/ を参照してください。

SUSEに関する意見交換
: 記事、ヒント、質疑応答、およびダウンロードできる無料ツールを提供するオンラインコミュニティ(http://www.suse.com/communities/conversations/)

GNOMEマニュアル
: GNOMEユーザ、管理者、および開発者向けのマニュアル(http://library.gnome.org/)

Linux Documentation Project
: TLDP(Linux Documentation Project)は、Linux関係のマニュアルを作成するボランティアチームによって運営されています(http://www.tldp.org 参照)。これは、おそらく、Linuxに関する最も総合的なドキュメントリソースです。マニュアルのセットには初心者向けのチュートリアルも含まれますが、主にシステム管理者などの経験者向けの内容になっています。TLDPは、HOWTO(操作方法)、FAQ(よくある質問)、ガイド(ハンドブック)を無償で提供しています。TLDPからのマニュアルの一部は、SUSE Linux Enterprise Server上でも利用できます。

汎用の検索エンジンも使用できます。たとえば、CDへの書き込みやLibreOfficeファイルの変換でトラブルがある場合は、検索する語句として「Linux CD-RW help(Linux CD-RWヘルプ)」または「OpenOffice file conversion problem (OpenOfficeファイルの変換の問題)」を使用します。

36 最も頻繁に起こる問題およびその解決方法

この章では、一連の潜在的な問題とその解決法について説明します。ここで状況が正確に記載されていなくても、問題解決のヒントになる類似した状況が見つかる場合があります。

36.1 情報の検索と収集

Linuxでは、非常に詳細なレポートが提供されます。システムの使用中に問題が発生した場合、調べる必要のあるところが何箇所かあります。それらのほとんどは、Linuxシステム一般で標準とされるもので、残りのいくつかはSUSE Linux Enterprise Serverシステムに関連するものです。大半のログファイルはYaSTを使って表示することができます([その他] › [起動ログを表示])。

YaSTでは、サポートチームが必要なシステム情報のすべてを収集することができます。[その他] › [サポート]の順に選択し、問題のカテゴリを選択します。すべての情報が収集されたら、それをサポートリクエストに添付します。

最も頻繁にチェックされるログファイルのリストの後には、一般的な目的に関する説明があります。~を含むパスは、現在のユーザのホームディレクトリを参照します。

表 36.1 ログファイル

ログファイル	説明
`~/.xsession-errors`	現在実行中のデスクトップアプリケーションからのメッセージです。
`/var/log/apparmor/`	AppArmorからのログファイル。詳細については、Book “Security Guide” を参照してください。
`/var/log/audit/audit.log`	システムのファイル、ディレクトリ、またはリソースに対するすべてのアクセスを追跡し、システムコールをトレースする監査からのログファイル。詳細については、Book “Security Guide” を参照してください。
`/var/log/mail.*`	メールシステムから受け取るメッセージです。
`/var/log/NetworkManager`	NetworkManagerからのログファイルで、ネットワーク接続についての問題を収集します。

ログファイル	説明
/var/log/samba/	Sambaサーバおよびクライアントのログメッセージを含んでいるディレクトリです。
/var/log/warn	カーネルおよびシステムのログデーモンから受け取る、「警告」レベル以上のすべてのメッセージ。
/var/log/wtmp	現在のコンピュータセッションのユーザのログインレコードを含むバイナリファイルです。last コマンドを使用して表示させます。
/var/log/Xorg.*.log	Xウィンドウシステムから受け取る、起動時および実行時のさまざまなログファイルです。Xの失敗した起動をデバッグするのに役に立ちます。
/var/log/YaST2/	YaSTのアクションおよびその結果を含んでいるディレクトリです。
/var/log/zypper.log	Zypperのログファイル。

ログファイルとは別に、稼働中のシステムの情報も提供されます。詳細については、「表 36.2: /procファイルシステムによるシステム情報」を参照してください。

表 36.2 /procファイルシステムによるシステム情報

ファイル	説明
/proc/cpuinfo	プロセッサのタイプ、製造元、モデル、およびパフォーマンスなどを含む情報を表示します。
/proc/dma	どのDMAチャネルが現在使用されているかを表示します。
/proc/interrupts	どの割り込みが使用されているか、各割り込みの使用回数を表示します。
/proc/iomem	I/Oメモリの状態を表示します。
/proc/ioports	その時点でどのI/Oポートが使用されているかを表示します。

ファイル	説明
/proc/meminfo	メモリステータスを表示します。
/proc/modules	個々のモジュールを表示します。
/proc/mounts	現在マウントされているデバイスを表示します。
/proc/partitions	すべてのハードディスクのパーティション設定を表示します。
/proc/version	現在のLinuxバージョンを表示します。

Linuxカーネルは、/proc ファイルシステムの場合を除いて、メモリ内ファイルシステムである sysfs モジュールで情報をエクスポートします。このモジュールは、カーネルオブジェクトとその属性および関係を表します。sysfs の詳細については、第16章 udevによる動的カーネルデバイス管理でudevのコンテキストを参照してください。表 36.3には、/sys の下にある最も一般的なディレクトリの概要が含まれています。

表 36.3 /sysファイルシステムによるシステム情報

ファイル	説明
/sys/block	システム内で検出された各ブロックデバイスのサブディレクトリが含まれています。一般に、これらの大半はディスクタイプのデバイスです。
/sys/bus	各物理バスタイプにのサブディレクトリが含まれます。
/sys/class	デバイスの機能タイプとしてグループ化されたサブディレクトリが含まれます(graphics、net、printerなど)。
/sys/device	グローバルなデバイス階層が含まれます。

Linuxには、システム解析とモニタリング用のさまざまなツールが含まれています。システム診断で使用される最も重要なツールの選択については、Book “System Analysis and Tuning Guide” 2 “System Monitoring Utilities”を参照してください。

次の各シナリオは、問題を説明するヘッダに続いて、推奨される解決方法、より詳細な解決方法への利用可能な参照、および関連する他のシナリオへの相互参照が書かれた、1つまたは2つの段落から構成されています。

36.2 インストールの問題

インストールの問題とは、コンピュータがインストールに失敗した状態のことを指します。インストールが全体において失敗する、またはグラフィカルインストーラが起動できないという可能性があります。ここでは、通常経験するような問題のいくつかに集中して説明し、そのような場合に考えられる解決方法または回避方法を示します。

36.2.1 メディアの確認

SUSE Linux Enterprise Serverのインストールメディアの使用中に問題が発生した場合、インストールメディアの整合性をチェックします。メディアからブートし、ブートメニューから[Check Installation Media (インストールメディアのチェック)]を選択します。実行中のシステムで、YaSTを起動して、[ソフトウェア] › [メディアチェック]の順に選択します。SUSE Linux Enterprise Serverのメディアをチェックするには、メディアをドライブに挿入し、YaSTの[メディアチェック]画面で、[チェック開始]をクリックします。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。メディアを自分で書き込んだ場合、メディアの問題が発生する場合があります。メディアを低速(4x)で書き込むと、問題を回避できます。

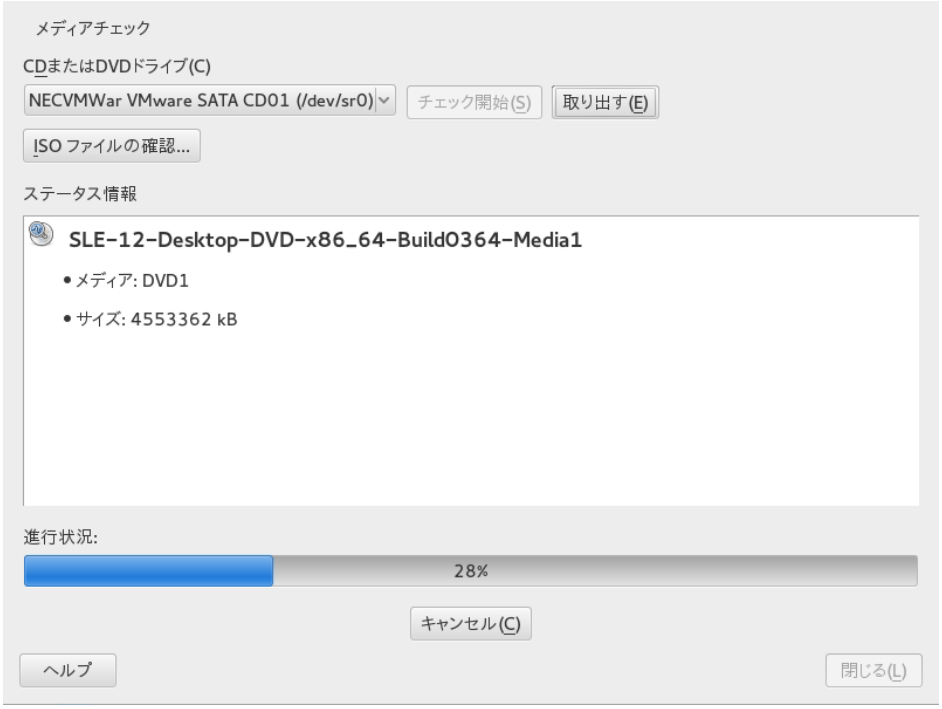


図 36.1 メディアの確認

36.2.2 ブート可能なDVDドライブが利用不可

お使いのコンピュータにブート可能なDVD-ROMドライブがない場合、または使用しているドライブがLinuxでサポートされていない場合、内蔵DVD-ROMドライブを使用しないでコンピュータをインストールするオプションがいくつかあります。

外付けブートデバイスの使用

BIOSおよびインストールカーネルによりサポートされている場合、外部DVDドライブまたはUSBストレージからブートします。ブート可能なUSBストレージデバイスの作成方法については、を参照してください。

PXE経由のネットワークブート

マシンにDVDドライブがない場合でも、使用可能なEthernet接続がある場合は、完全にネットワークベースのインストールを実行します。詳細については、ブック「導入ガイド」14「リモートインストール」14.1.3「VNCによるリモートインストール—PXEブートとWake on LAN」とブック「導入ガイド」14「リモートインストール」14.1.6「SSHによるリモートインストール—PXEブートとWake on LAN」を参照してください。

36.2.2.1 外付けブートデバイス

Linuxでは、既存のDVDドライブはほとんどサポートされます。システムにDVDドライブが存在しない場合でも、USB、FireWire、またはSCSIを通じて接続する外部DVDドライブ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。

ライブCDからインストールする場合は、「」「ライブUSBフラッシュディスク」を作成して、このディスクからブートすることもできます。

36.2.3 インストールメディアからのブートに失敗する

コンピュータでインストールメディアが起動しない理由の1つとして、BIOS内のブートシーケンスの設定が誤っている場合があります。BIOSブートシーケンスでは、ブート用の最初のエントリとしてDVDドライブがセットされている必要があります。そうでない場合、コンピュータは他のメディア(通常ハードディスク)からブートを試みます。BIOSのブートシーケンスを変更するための説明は、マザーボードに付属するマニュアルまたは次の段落に記載されています。

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とき(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。通常は、< Del >、 F1 、または< Esc >のいずれかのキーを押します。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。

手順 36.1 BIOSのブートシーケンスの変更

1. ブートルーチンによって宣言されたように、適切なキーを使用してBIOSを入力します。その後、BIOS画面が表示されるのを待ちます。
2. 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ら、そのエントリを選択して、< Enter >キーを押して確定します。
3. 開いた画面で、[BOOT SEQUENCE]または[BOOT ORDER]というサブエントリを探します。DVDドライブが最初に表示されるまで PgUp キーまたは PgDown キーを押して、設定を変更します。

4. <Esc>キーを押して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>キーを押します。設定が保存されていることを確認するには、<Y>キーを押します。

手順 36.2 SCSI BIOS (ADAPTECホストアダプタ)内でのブートシーケンスの変更

1. <Ctrl>–<A>を押してセットアップを開きます。
2. [ディスクユーティリティ]を選択します。これで、接続したハードウェアコンポーネントが表示されるようになります。
   ご使用のDVDドライブに割り当てられているSCSI IDの記録をとります。
3. <Esc>キーを押して、メニューを閉じます。
4. [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>キーを押します。
5. DVDドライブのIDを入力して、再度<Enter>キーを押します。
6. <Esc>キーを2回押して、SCSI BIOSの起動画面に戻ります。
7. [はい]を押して、この画面を終了しコンピュータを起動します。

最終的なインストールが使用する言語やキーボードレイアウトに関係なく、BIOS設定では、通常以下の図に示されているようなUSキーボードレイアウトが使用されます。

図 36.2 USキーボードレイアウト

36.2.4 ブートできない

ハードウェアのタイプ(主にかなり旧式かごく最近のタイプ)では、インストールが失敗するものもあります。多くの場合、インストールカーネル内でこのタイプのハードウェアのサポートが欠けているか、または、ある種のハードウェアに問題を引き起こすACPIのような、カーネルに含まれている特定の機能が原因の可能性があります。

最初のインストールブート画面から、標準の[インストール]モードを使用してインストールするのに失敗した場合、以下のことを試してみてください。

1. DVDがドライブにまだ入った状態であれば、<Ctrl>-<Alt>-<Del>を押すか、ハードウェアリセットボタンを使用して、コンピュータを再起動します。
2. ブート画面が表示されたら、<F5>キーを押すか、キーボードの矢印キーを使用して、[ACPIなし]を探し、<Enter>キーを押してブートおよびインストールプロセスを開始します。このオプションはACPIの電源管理技術を無効にします。
3. ブック「導入ガイド」6「YaSTによるインストール」の中での説明に従って、インストールを進めます。

これが失敗する場合、以上で述べた手順の代わりに[セーフ設定]を選択してインストール処理を続行します。このオプションはACPIおよびDMAサポートを無効化します。このオプションを使うと、ほとんどのハードウェアが起動します。

両方のオプションともに失敗する場合、ブートオプションプロンプトを使用して、ハードウェアタイプをサポートするのに必要な追加のパラメータをインストールカーネルに渡します。ブートオプションとして使用可能なパラメータの詳細については、`/usr/src/linux/Documentation/kernel-parameters.txt`にあるカーネルマニュアルを参照してください。

ヒント: カーネルマニュアルの取得

kernel-source **パッケージをインストールして、カーネルマニュアルを表示します。**

他にさまざまなACPI関連のカーネルパラメータがあります。それらのパラメータは、インストールのために起動する前のブートプロンプトで入力できます。

`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=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であっても、常にACPIを有効にします。このパラメータは、`acpi=off` に加えて設定された場合、ACPIも有効にします。

`acpi=noirq`

ACPIはIRQルーティングには使用しません。

`acpi=ht`

hyper-threadingを有効化するのに十分なACPIのみ実行します。

`acpi=strict`

厳密にはACPI仕様互換ではないプラットフォームに対する耐性が弱くなります。

`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。

`pnpacpi=off`

このオプションは、BIOSセットアップに誤った割り込みまたはポートがある場合のシリアルまたはパラレルの問題向けです。

`notsc`

タイムスタンプカウンタを無効にします。このオプションを使用して、システムのタイミングについての問題に対処できます。これは最近の機能で、コンピュータに特に時間や全面的なハングなどの遅れが見られる場合に、このオプションを試す価値があります。

`nohz=off`

nohz機能を無効にします。マシンがハングした場合、このオプションが役に立ちます。それ以外の場合は、使用しません。

パラメータの正しい組み合わせを決定したら、システムが次回適切に起動することを確実にするために、YaSTは自動的にそれらのパラメータをブートローダの設定に書き込みます。

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。［メモリテスト］がエラーを返す場合、それは通常はハードウェアのエラーです。

36.2.5 グラフィカルインストーラを起動できない

メディアをドライブに挿入しコンピュータを再起動した後に、インストール画面が表示されますが、［インストール］を選択すると、グラフィカルインストーラは起動しません。

この問題に対処する方法はいくつかあります。

- インストールダイアローグ用に、他の画面解像度を選択してみます。
- インストール用に[テキストモード]を選択します。
- VNCを介して、グラフィカルインストーラを使ってリモートインストールをします。

手順 36.3 インストール時の画面解像度の変更

1. インストールのために起動します。
2. F3 キーを押して、インストール用に低解像度を選択するメニューを開きます。
3. [インストール]を選択し、ブック「導入ガイド」6「YaSTによるインストール」の中の説明に従ってインストールを続行します。

手順 36.4 テキストモードのインストール

1. インストールのために起動します。
2. F3 キーを押して、[テキストモード]を選択します。
3. [インストール]を選択し、ブック「導入ガイド」6「YaSTによるインストール」の中の説明に従ってインストールを続行します。

手順 36.5 VNCによるインストール

1. インストールのために起動します。
2. ブートオプションプロンプトに以下のテキストを入力します。

   ```
   vnc=1 vncpassword=some_password
   ```

   some_password の部分はVNCインストール用に使用するパスワードに置き換えます。
3. [インストール]を選択し、キーを押してインストールを開始します。Enter
   グラフィカルインストールルーチンに入る代わりに、システムはテキストモードで実行され、その後停止します。その際、IPアドレスおよびポート番号が含まれるメッセージが表示されますが、これらは、ブラウザインタフェースまたはVNCビューアアプリケーションを使用してインストーラにアクセスできるようにするために必要です。
4. ブラウザを使用してインストーラにアクセスする場合、ブラウザを起動して将来SUSE Linux Enterprise Serverが起動するマシン上のインストール手順で与えられたアドレス情報を入力し、< Enter キーを押します。

```
http://ip_address_of_machine:5801
```

ブラウザウィンドウでは、VNCのパスワードを入力するように要求するダイアログが開かれます。パスワードを入力し、ブック 「導入ガイド」 6 「YaSTによるインストール」の説明に従ってインストールを続行します。

重要: クロスプラットフォームのサポート

VNC経由のインストールでは、Javaサポートが有効化されていれば、オペレーションシステムやブラウザの種類を問いません。

プロンプトが表示されたら、VNCビューアにIPアドレスとパスワードを入力します。インストールダイアログを表示するウィンドウが開きます。通常のようにインストールを続行します。

36.2.6 最低限のブート画面だけが起動する

メディアをドライブに挿入して、BIOSルーチンは終了しますが、システム上でグラフィカルブート画面が開始しません。その代わりに、最小限のテキストベースのインタフェースが起動されます。これは、グラフィカルブート画面を表示するのに十分なグラフィックメモリを持っていないコンピュータを使用する場合に起こる可能性がります。

テキストのブート画面は最小限にに見えますが、グラフィカルブート画面が提供する機能とほぼ同じものを提供します。

ブートオプション

グラフィカルインタフェースとは違い、キーボードのカーソルキーを使って異なるブートオプションを選択することはできません。テキストモードのブート画面のブートメニューでは、ブートプロンプトで入力するキーワードが表示されます。これらのキーワードはグラフィカルバージョンで提供されているオプションにマップしています。任意の選択を入力し< Enter キーを押して、ブートプロセスを起動します。

カスタムブートオプション

ブートオプションを選択したあと、ブートプロンプトで適切なキーワードを入力するか、36.2.4項「ブートできない」の中で説明されているカスタムブートオプションを入力します。インストールプロセスを起動するには、< Enter キーを押します。

画面解像度

Fキーを使用して、インストール用の画面解像度を判別します。テキストモードで起動する必要がある場合は、 ーを選択しま

36.3 ブートの問題

ブートの問題とは、システムが適切にブートしないような場合を指します(意図したターゲットおよびログイン画面までブートしない場合)。

36.3.1 GRUB 2ブートローダのロードに失敗する

ハードウェアが問題なく機能している場合、ブートローダが壊れてしまってLinuxがコンピュータ上で起動できない可能性があります。このような場合、ブートローダを修復する必要があります。そのためには、36.6.2項 「レスキューシステムの使用」の説明に従ってレスキューシステムを起動し、36.6.2.4項 「ブートローダの変更と再インストール」の手順に従う必要があります。

コンピュータが起動しない理由は他にBIOS関連のものが考えられます。

BIOS設定
: ハードディスクの参照情報については、BIOSを確認してください。ハードディスク自体が現在のBIOS設定に見つからない場合、GRUB 2が単に開始されない可能性があります。

BIOSブートオーダー
: お使いのシステムのブートオーダーがハードディスクを含んでいるか確認します。ハードディスクオプションが有効になっていない場合、システムは適切にインストールされていますが、ハードディスクへのアクセスが要求される際に起動に失敗する可能性があります。

36.3.2 グラフィカルログインがない

マシンは起動するものの、グラフィカルログインマネージャがブートしない場合は、デフォルトのsystemdターゲットの選択、またはXウィンドウシステムの設定のいずれかに問題があると考えられます。現在のデフォルトのsystemdターゲットを確認するには、`sudo systemctl get-default`コマンドを実行します。返された値が`graphical.target`で「ない」場合、`sudo systemctl isolate graphical.target`コマンドを実行します。グラフィカルログイン画面が起動する場合は、ログインして、[YaST] › [システム] › [サービスマネージャ]を起動し、[デフォルトのシステムターゲット]を[Graphical Interface (グラフィカルインタフェース)]に設定します。今後、システムはグラフィカルログイン画面でブートするようになります。

ブートするかグラフィカルターゲットに切り替わっても、グラフィカルログイン画面が起動しない場合は、ご使用のデスクトップかXウィンドウソフトウェアの設定が間違っているか、破損している可能性があります。`/var/log/Xorg.*.log`のログファイルで、Xサーバが起動を試みた際にXサーバによって記録された詳細メッセージを調べます。デスクトップの起動に失敗する場合は、`journalctl`コマンド(詳細は第11章 `journalctl`:systemdジャーナルのクエリを参照)で問い合わせ可能なシステム

ジャーナルにエラーメッセージが記録されている場合があります。これらのエラーメッセージがXサーバの設定の問題を示唆している場合は、これを直すようにしてください。それでもグラフィカルシステムが起動しない場合は、グラフィカルデスクトップを再インストールすることを考えてください。

36.3.3 ルートBtrfsパーティションをマウントできない

`btrfs` ルートパーティションが壊れた場合は、次のオプションを試してみてください。

- `-o recovery` オプションを使用してパーティションをマウントする。
- これが失敗する場合は、ルートパーティション上で `btrfs-zero-log` を実行する。

36.4 Loginの問題

ログインの問題とは、お使いのマシンが予期されるようこそ画面またはログインプロンプトまで実際に起動するが、ユーザ名およびパスワードを受け付けない、または受け付けるが、その後適切な動きをしない場合です(グラフィックデスクトップ開始の失敗、エラーの発生、コマンドラインに落ちる、など)。

36.4.1 有効なユーザ名とパスワードを使っても失敗する

この問題は、一般的にシステムがネットワーク認証またはディレクトリサービスを使用するように設定されており、何らかの理由で、設定されたサーバから結果を取得できない場合に発生します。このような場合でも、`root` ユーザは唯一のローカルユーザとしてこれらのコンピュータにログインできます。次に、コンピュータが一見機能しているように見えるのにログインを正しく処理できない一般的な理由をいくつか挙げます。

- ネットワークが機能していません。この場合の更なる対処方法については、36.5項 「ネットワークの問題」を参照してください
- 現在、DNSが機能していません(このためGNOMEが動作せず、システムはセキュアサーバに検証済みの要求を送信できません)。すべてのアクションに対して、コンピュータに極端に長い時間かかる場合は、この問題の可能性があります。このトピックの詳細は、36.5項 「ネットワークの問題」を参照してください。
- システムがKerberosを使用するように設定されている場合、システムのローカルタイムは、Kerberosサーバのタイムとの間で許容される相違を超えてしまっている可能性があります(通常 300秒)。NTP (network time protocol)が適切に動いていない、またはローカルのNTPサーバが動いていない場合、Kerberos　の認証は機能しなくなります。その理由は、この認証はネットワーク間の一般的なクロック同期に依存しているからです。

- システムの認証設定が間違って設定されています。関連するPAM設定ファイルの中に誤字や命令の順序違いがないか確認します。PAMおよび関連する設定ファイルの構文に関する背景情報の詳細については、Book “Security Guide” 2 “Authentication with PAM”を参照してください。

- ホームパーティションが暗号化されています。このトピックの詳細は、36.4.3項 「暗号化されたホームパーティションへのログインが失敗します」を参照してください。

外部のネットワーク問題を含まない他のすべての問題については、解決方法としてシステムをシングルユーザモードに再起動して、動作モードに再び起動してログインし直す前に、設定を修復します。シングルユーザモードで起動するには、次の手順に従います。

1. システムを再起動します。ブート画面の表示に続き、プロンプトが表示されます。
2. < Esc >キーを押して、スプラッシュスクリーンを終了し、GRUB 2テキストベースメニューに移動します。
3. B を押して、GRUB 2エディタを起動します。
4. カーネルパラメータを含む行に次のパラメータを追加します。

   ```
   systemd.unit=rescue.target
   ```

5. F10 キーを押します。
6. `root` のユーザ名とパスワードを入力します。
7. 必要な変更をすべて加えます。
8. コマンドラインに「`systemctl isolate graphical.target`」と入力して、完全なマルチユーザおよびネットワークモードでブートします。

36.4.2 有効なユーザ名とパスワードが受け付けられない

これは、今のところユーザが経験する問題のうち、最も一般的なものです。その理由は、この問題が起こる原因がたくさんあるからです。ローカルのユーザ管理および認証を使用するか、ネットワーク認証を使用するかによって、異なる原因によりログイン失敗が発生します。

ローカルユーザ管理は、次の原因により失敗する可能性があります。

- 間違ったパスワードを入力した可能性があります。
- ユーザのホームディレクトリが、破損または書き込み保護されたデスクトップ設定ファイルを含んでいます。
- この特定のユーザを認証するのに、X Window Systemに何らかの問題があります。特に、ユーザのホームディレクトリが、現在のLinuxをインストールする以前の他のLinuxディストリビューションによって使用されている場合です。

ローカルログイン失敗の原因を発見するには、次の手順に従います。

1. 認証方式全体をデバッグする前に、ユーザがパスワードを正しく覚えているか確認します。ユーザが正しいパスワードを覚えていない場合は、YaSTユーザ管理モジュールを使用してそのユーザのパスワードを変更します。Caps Lock キーに注意し、必要に応じてそのロックを解除します。
2. `root` ユーザでログインし、ログインプロセスおよびPAMのエラーメッセージがないかどうか `journalctl -e` でシステムジャーナルを確認します。
3. コンソールからログインしてみます(Ctrl – Alt – F1 キーを使用)。これが成功する場合、PAMには問題はありません。その理由は、そのユーザをそのコンピュータ上で認証可能だからです。Xウィンドウシステムまたは GNOMEデスクトップに問題がないか探してみてください。詳細については、36.4.4項 「ログインは成功したがGNOMEデスクトップが失敗する」を参照してください。
4. ユーザのホームディレクトリが他のLinuxディストリビューションによって使用されている場合、ユーザのホームにある `Xauthority` ファイルを削除します。Ctrl – Alt – F1 キーを押してコンソールログインを使用し、`rm .Xauthority` をこのユーザとして実行します。これにより、X認証の問題はこのユーザに関してはなくなるはずです。グラフィカルログインを再試行します。
5. 設定ファイルが壊れていて、デスクトップが開始できなかった場合、36.4.4項 「ログインは成功したがGNOMEデスクトップが失敗する」に進みます。

以下では、特定のユーザのネットワーク認証が、特定のコンピュータ上で失敗するのかの一般的な理由のいくつかを挙げます。

- 間違ったパスワードを入力した可能性があります。
- コンピュータのローカル認証ファイルの中に存在し、ネットワーク認証システムからも提供されるユーザ名が競合しています。
- ホームディレクトリは存在しますが、それが壊れている、または利用不可能です。書き込み保護がされているか、その時点でアクセスできないサーバ上にディレクトリが存在するかのどちらかの可能性があります。

- 認証システム内で、ユーザがその特定のサーバにログインする権限がありません。
- コンピュータのホスト名が何らかの理由で変更されていて、そのホストにユーザがログインするパーミッションがありません。
- コンピュータが、認証サーバまたはそのユーザの情報を含んでいるディレクトリサーバに接続できません。
- この特定のユーザを認証するのに、X Window Systemに何らかの問題があります。特に、ユーザのホームが、現在のLinuxをインストールする以前に他のLinuxディストリビューションによって使用されている場合です。

ネットワーク認証におけるログイン失敗の原因を突き止めるには、次の手順に従います。

1. 認証方式全体をデバッグする前に、ユーザがパスワードを正しく覚えているか確認します。
2. 認証用にマシンが利用するディレクトリサーバを判別し、それがきちんと動作しており、他のマシンと適切に通信していることを確認します。
3. ユーザのユーザ名およびパスワードが他のマシン上でも使用できるかを判別し、そのユーザの認証データが存在し、適切に配布されていることを確認します。
4. 別のユーザが、問題のある動きをしているマシンにログインできるかどうかを確認します。別のユーザで問題なくログインできる場合、または `root` でログインできる場合、ログイン後、`journalctl -e` >ファイルでシステムジャーナルを調べます。ログインの試行に対応するタイムスタンプを見つけ出し、PAMによって、エラーメッセージが生成されていないか判別します。
5. コンソールからログインしてみます(Ctrl – Alt – F1 キーを使用)。これが成功する場合、PAMやユーザのホームがあるディレクトリサーバには問題はありません。その理由は、そのユーザをそのコンピュータ上で認証可能だからです。Xウィンドウシステムまたは GNOMEデスクトップに問題がないか探してみてください。詳細については、36.4.4項 「ログインは成功したがGNOMEデスクトップが失敗する」を参照してください。
6. ユーザのホームディレクトリが他のLinuxディストリビューションによって使用されている場合、ユーザのホームにある `Xauthority` ファイルを削除します。 Ctrl – Alt – F1 キーを押してコンソールログインを使用し、`rm .Xauthority` をこのユーザとして実行します。これにより、X認証の問題はこのユーザに関してはなくなるはずです。グラフィカルログインを再試行します。
7. 設定ファイルが壊れていて、デスクトップが開始できなかった場合、36.4.4項 「ログインは成功したがGNOMEデスクトップが失敗する」に進みます。

36.4.3 暗号化されたホームパーティションへのログインが失敗します

ラップトップでは暗号化されたホームパーティションの使用が推奨されます。ラップトップにログインできない場合、通常その理由は簡単です。パーティションのロックを解除できなかったためです。

ブート時に、暗号化パーティションのロックを解除するためにパスフレーズを入力する必要があります。パスフレーズを入力しない場合、パーティションがロックしたまま起動プロセスが続行します。

暗号化されたパーティションのロックを解除するには、次の手順に従います。

1. <Ctrl>-<Alt>-<F1>でテキストコンソールに切り替えます。
2. `root`になります。
3. 次のコマンドにより、ロックを解除するプロセスを再開します。

```
systemctl restart home.mount
```

4. 暗号化されたパーティションのロックを解除するためのパスフレーズを入力します。
5. テキストコンソールを終了し、<Alt>-<F7>でログイン画面に切り替えます。
6. 通常通りログインします。

36.4.4 ログインは成功したがGNOMEデスクトップが失敗する

この場合に、GNOME環境設定ファイルが破損している可能性があります。兆候としては、キーボードがうまく動かない、画面のジオメトリが歪んでいる、または画面が空の灰色領域として表示されるなどがあります。この問題の重要な特徴は、他のユーザがログインする場合は、コンピュータは普通に機能するという点です。このような場合、問題のユーザのGNOME設定ディレクトリを単に新しい場所に移すことで、が新しいデスクトップを初期化するので、比較的簡単にこの問題を解決できます。ユーザはGNOMEの再設定を強いられますが、データが失われません。

1. <Ctrl>-<Alt>-<F1>を押して、テキストコンソールを切り替えます。
2. ユーザ名でログインします。
3. ユーザのGNOME設定ディレクトリを、一時的な場所に移動します。

```
mv .gconf  .gconf-ORIG-RECOVER
mv .gnome2 .gnome2-ORIG-RECOVER
```

4. ログアウトします。
5. もう一度ログインします。ただし、アプリケーションは何も実行しないでください。
6. 次のようにして、`~/.gconf-ORIG-RECOVER/apps/` ディレクトリを、新しい `~/.gconf` ディレクトリにコピーすることで個々のアプリケーション設定データ(Evolutionの電子メールクライアントデータを含む)を回復します。

   ```
   cp -a .gconf-ORIG-RECOVER/apps .gconf/
   ```

   これによってログインの問題が生じる場合は、重要なアプリケーションデータのみの回復を試み、アプリケーションの残りを再設定します。

36.5 ネットワークの問題

システム上の問題は、最初はそうは見えないのですが、ネットワークに関する問題であることが多いです。例えば、システムにユーザがログインできない理由は、ある種のネットワークの問題であったりします。ここでは、ネットワークの問題に直面した場合の簡単なチェックリストを紹介します。

手順 36.6 ネットワークの問題を識別する方法

コンピュータとネットワークの接続の確認をする場合、以下の手順に従ってください。

1. Ethernet接続を使用する場合、はじめにハードウェアを確認します。ネットワークケーブルがコンピュータおよびルータ(またはハブなど)にしっかり差し込まれていることを確認してください。Ethernetコネクタの横に制御ランプがある場合、通常はその両方がアクティブになります。接続に失敗する場合、お使いのネットワークケーブルが他のコンピュータでは使用可能かどうか確認します。使用可能な場合、ネットワークカードに問題の原因があります。ネットワークのセットアップにハブやスイッチを使用している場合は、それらが誤っている可能性もあります。
2. 無線接続を使用7る場合、他のコンピュータからワイヤレスリンクが確立できるかどうか確認します。そうでない場合は、無線ネットワークの管理者にお問い合わせください。
3. 基本的なネットワーク接続を確認し終わったら、どのサービスが応答していないかを探します。お使いの構成上のすべてのネットワークサーバのアドレス情報を集めます。適切なYaSTモジュール内で探すか、システム管理者に問い合わせてください。以下のリストには、ある構成内に含まれる一般的なネットワークサーバを、それらの故障の兆候とともに表わしています。

DNS (ネームサービス)

壊れた、あるいは誤作動しているネームサービスは、ネットワークの機能にさまざまな形で影響を与えます。ローカルマシンの認証をネットワークサーバで行っている場合、名前解決に問題があるためにそれらのサーバが見つからないと、ユーザはログインすることもできません。壊れたネームサーバが管理するネットワーク内のマシンは、お互いを「認識」「」できないため通信できません。

NTP (タイムサービス)

誤作動している、または完全に壊れたNTPサービスは、Kerberosの認証およびXサーバの機能に影響を与えます。

NFS (ファイルサービス)

NFSによってマウントされたディレクトリ内のデータを必要とするアプリケーションがあった場合、このNFSサービスがダウンしてるか、間違って設定されていると、そのアプリケーションは起動できないか、または正しく機能しません。最悪のケースとしては、`.gconf` サブディレクトリを含んでいる、あるユーザのホームディレクトリが、NFSサーバの故障のために検出されなかった場合、そのユーザ個人のデスクトップ設定が起動しません。

Samba (ファイルサービス)

アプリケーションが、故障したSambaサーバ上のディレクトリに保存されたデータを必要とする場合、アプリケーションは起動できないか、または正しく機能しません。

NIS (ユーザ管理)

SUSE Linux Enterprise Serverシステムがユーザデータを提供するために故障したNISサーバを使用している場合、ユーザはマシンにログインできません。

LDAP (ユーザ管理)

SUSE Linux Enterprise Serverシステムがユーザデータを提供するために故障したLDAPサーバを使用している場合、ユーザはこのマシンにログインできません。

Kerberos (認証)

認証が機能せず、すべてのコンピュータへのログインが失敗します。

CUPS (ネットワーク印刷)

ユーザが印刷できません。

4. ネットワークサーバが起動しているか、ネットワーク上で接続を確立できる設定になっているか、を確認します。

重要: 制限

次で説明するデバッグの手順は、内部ルーティングを必要としない、簡単なネットワークサーバ/クライアント設定にのみ適用されます。サーバとクライアントの両方が、追加でルーティングする必要のない同じサブネットのメンバーであることが前提です。

a. `ping IP address`または`hostname` (`host name`はサーバのホスト名で置き換えます)を使って、サーバが稼働中で、ネットワークに応答しているかどうかを確認します。このコマンドが成功する場合は、目的のホストは起動しており、ネットワークのネームサービスは正しく設定されていることがわかります。
pingが「`destination host unreachable`」というメッセージで失敗する場合、お使いのシステムまたは宛先のサーバが正しく設定されていないか、ダウンしています。その場合、他のマシンから`ping IP address`または`your_hostname`を実行して、お使いのシステムに到達可能か確認してください。他のマシンからお使いのマシンへ到達可能な場合、宛先のサーバがまったく動作していないか、正しく設定されていません。
pingが「`unknown host`」というメッセージで失敗する場合、ネームサービスが正しく設定されていないか、使用したホスト名が正しくありません。この問題を詳細に調べるには、ステップ 4.bを参照してください。それでもpingが失敗する場合は、ネットワークカードが正しく設定されていないか、ネットワークのハードウェアに障害があります。

b. `host hostname`を使用して、接続しようとしているサーバのホスト名が適切なIPアドレスに変換され、またその逆も問題ないか確認します。このコマンドによって、このホストのIPアドレスが返される場合、ネームサービスは起動中です。この`host`コマンドが失敗する場合、お使いのホスト上の名前とアドレス解決に関係するすべてのネットワーク設定ファイルを確認します。

`/etc/resolv.conf`

このファイルは、ネームサーバおよび現在使用中のドメインを管理するために使用されます。このファイルは手動で変更するか、YaSTまたはDHCPによる自動調整が可能ですが、自動調整のほうをお勧めします。ただし、このファイルが以下のような構造およびネットワークアドレスを含んでいること、さらにドメイン名が正しいことを確認してください。

```
search fully_qualified_domain_name
nameserver ipaddress_of_nameserver
```

このファイルには1つ以上のネームサーバのアドレスを含むことができますが、その中の少なくとも1つは、お使いのホストの名前解決が正しくできる必要があります。必要に応じて、YaSTネットワーク設定モジュール([ホスト名/DNS]タブ)を使用してこのファイルを修正します。
ネットワーク接続がDHCP経由の場合、YaSTのDNSおよびHostnameモジュール内で、[DHCP経由でのホスト名の変更]および[DHCP経由でのネームサービスおよび検索リストの更新]を選択し、DHCPを有効化してホスト名およびネームサービス情報を変更します。

`/etc/nsswitch.conf`

このファイルは、Linuxがネームサービス情報を探す場所を示します。このようになります。

```
 ...
hosts: files dns
networks: files dns
...
```

`dns`エントリは必須です。これにより、Linuxは外部のネームサーバを使用するようになります。通常、これらのエントリはYaSTにより自動的に管理されますが、慎重にチェックする必要があります。
ホスト上で、すべての関連エントリが正しい場合は、システム管理者に依頼して、正しいゾーン情報に関するDNSサーバの設定を確認してもらいます。DNSの詳細については、第22章　ドメインネームシステムを参照してください。お使いのホストのDNS設定およびDNSサーバが正しいことが確認できた場合、ネットワークおよびネットワークデバイス設定の確認に進みます。

c. お使いのシステムがネットワークサーバに接続できない状況で、ネームサービスの問題を障害原因の可能性リストから除外した場合は、ネットワークカードの設定を確認します。`ip addr show network_device`コマンドを使用して、このデバイスが適切に設定されているか確認します。`inet address`がネットマスク(`/mask`)を使用して正しく設定されていることを確認します。IPアドレス内に間違いがある場合、またはネットワークマスク内で不明のビットがある場合は、ネットワーク設定が使用不可能になります。必要であれば、サーバ上でもこの確認をしてください。

d. ネームサービスおよびネットワークサービスが正しく設定され起動している場合でも、外部のネットワーク接続がタイムアウトするのに時間がかかったり、完全に失敗する場合は、`traceroute fully_qualified_domain_name`(`root`ユーザで実行)コマンドを使用して、リクエストがネットワーク上でどのルートを使用するか追跡します。このコマンドは、

お使いのコンピュータのリクエストが宛先に到達するまでに経由するゲートウェイ(ホップ)をリストします。各ホップの応答時間およびこのホップにそもそも到達可能か否かをリストします。tracerouteおよびpingコマンドを組み合わせて原因を追究し、管理者に知らせてください。

ネットワーク障害の原因を突き止めたら、自身でそれを解決するか(自分のコンピュータ上に問題がある場合)、お使いのネットワークのシステム管理者に原因について報告し、サービスを再設定するか、必要なシステムを修理してもらってください。

36.5.1 NetworkManagerの問題

ネットワーク接続に問題がある場合は、手順36.6「ネットワークの問題を識別する方法」の説明に従って原因を絞り込んでください。NetworkManagerが原因と考えられる場合は、以降の説明に従ってNetworkManager障害の理由を調べるために役立つログを取得してください。

1. シェルを開いて、`root`としてログインします。
2. NetworkManagerを再起動します。

```
systemctl restart Network.Manager.service
```

3. 一般ユーザとしてhttp://www.opensuse.orgなどのWebページを開いて、正常に接続できているかどうかを確認します。
4. `/var/log/NetworkManager`にある、NetworkManagerに関する情報を収集します。

NetworkManagerについての詳細は、第24章 NetworkManagerの使用を参照してください。

36.6 データの問題

データの問題とは、コンピュータが正常に起動するかしないかに関係なく、システム上でデータが壊れており、システムの修復が必要な場合を言います。このような状況では、システムに障害が発生する前の状態にシステムを復元するために、重要なデータをバックアップする必要があります。SUSE Linux Enterprise Serverには、システムのバックアップ/復元や、レスキューシステム(壊れたシステムを外部から復元するのに使用できる)用に、専用のYaSTモジュールが用意されています。

36.6.1 パーティションイメージの管理

パーティション全体、さらにはハードディスク全体からバックアップを実行することが必要になる場合があります。Linuxには、ディスクの正確なコピーを作成できる`dd`ツールが付属しています。`gzip`と組み合わせることで、若干の領域の節約になります。

手順 36.7 ハードディスクのバックアップと復元

1. `root`ユーザとしてシェルを起動します。
2. ソースデバイスを選択します。これは、`/dev/sda`などが一般的です(`SOURCE`というラベルが付きます)。
3. イメージを保存する場所を決めます(`BACKUP_PATH`というラベルが付きます)。これは、ソースデバイスとは異なる場所にする必要があります。つまり、`/dev/sda`からバックアップを作成する場合、イメージファイルは`/dev/sda`に保存しないでください。
4. コマンドを実行して圧縮イメージファイルを作成します。

   ```
   dd if=/dev/SOURCE | gzip > /BACKUP_PATH/image.gz
   ```

5. 次のコマンドによりハードディスクを復元します。

   ```
   gzip -dc /BACKUP_PATH/image.gz | dd of=/dev/SOURCE
   ```

バックアップするパーティションのみが必要な場合は、`SOURCE`プレースホルダーを対応するパーティションに置き換えます。この場合、イメージファイルを同じハードディスクにおくことができます。ただし、パーティションは異なります。

36.6.2 レスキューシステムの使用

システムが起動し正常に稼動するのに失敗する理由はいくつか考えられます。最も一般的な理由としては、システムクラッシュによるファイルシステムの破損や、ブートローダ設定の破損があります。

このような状況の解決を支援するため、SUSE Linux Enterprise Serverには、ブート可能なレスキューシステムが含まれています。レスキューシステムは、RAMディスクにロードして、ルートファイルシステムとしてマウントできる小さなLinuxシステムで、これを利用して外部からLinuxパーティションにアクセスすることができます。レスキューシステムを使用して、システムの重要な部分を復元したり、適切な変更を行ったりできます。

- 任意の種類の設定ファイルを操作できます。
- ファイルシステムの欠陥をチェックして、自動修復プロセスを開始することができます。

- インストールされているシステムを、「他のルート」環境内からアクセスすることができます。
- ブートローダの設定を確認、変更、および再インストールできます。
- 正常にインストールされていないデバイスドライバや使用不能なカーネルを修復できます。
- partedコマンドを使って、パーティションサイズを変更できます。このツールの詳細については、GNU PartedのWebサイト(http://www.gnu.org/software/parted/parted.html)を参照してください。

レスキューシステムは、さまざまなソースや場所からロードすることができます。一番簡単な方法は、オリジナルのインストールメディアからレスキューシステムをブートすることです。

注記: IBM System zでのレスキューシステムの起動

IBM System zでは、レスキュー目的でインストールシステムを使用することができます。レスキューシステムを起動するには、36.7項「IBM System z: initrdのレスキューシステムとしての使用」の指示に従ってください。

1. インストールメディアをDVDドライブに挿入します。
2. システムを再起動します。
3. ブート画面で、F4を押し、[DVD-ROM]を選択します。次に、メインメニューから[レスキューシステム]を選択します。
4. `Rescue:` プロンプトに「`root`」と入力します。パスワードは必要ありません。

ハードウェア設定にDVDドライブが含まれていない場合は、ネットワークソースからレスキューシステムをブートできます。次の例は、リモートブートの場合のシナリオです。DVDなど、他のブートメディアを使用する場合は、`info` ファイルを適宜変更し、通常のインストールと同様にブートします。

1. PXEブートセットアップの設定を入力し、`install=protocol://instsource` 行と `rescue=1` 行を追加します。修復システムを起動する必要がある場合は、代わりに `repair=1` を使用します。通常のインストールと同様に、`protocol` はサポートする任意のネットワークプロトコル(NFS、HTTP、FTPなど)を表しています。また、`instsource` は、ネットワークインストールソースへのパスを表します。
2. ブック「導入ガイド」14「リモートインストール」14.3.7「Wake on LAN」に説明したように、「Wake on LAN」を使用してシステムをブートします。
3. `Rescue:` プロンプトに「`root`」と入力します。パスワードは必要ありません。

レスキューシステムが起動したら、< Alt – F1 >～< Alt – F6 >を使って、仮想コンソールを使用することができます。

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確認して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。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。たとえばシステム保守用には fdisk 、mkfs 、mkswap 、mount 、および shutdown があり、ネットワーク保守用には ip および ss があります。/usr/bin ディレクトリには、vi editor、find、less、およびsshがあります。

システムメッセージを表示するには、dmesg コマンドを使用するか、または journalctl を使用してシステムログを参照してください。

36.6.2.1 環境設定ファイルの確認と修正

レスキューシステムを使った環境設定情報の修正例として、環境設定ファイルが壊れたためシステムが正常にブートできなくなった場合を考えてみましょう。このような場合は、レスキューシステムを使って設定ファイルを修復します。

環境設定ファイルを修正するには、以下の手順に従ってください。

1. 前述のいずれかの方法を使って、レスキューシステムを起動します。
2. /dev/sda6 下にあるルートファイルシステムをレスキューシステムにマウントするには、以下のコマンドを使用します。

   ```
   mount /dev/sda6 /mnt
   ```

   システム中のすべてのディレクトリが、/mnt 下に配置されます。
3. マウントしたルートファイルシステムのディレクトリに移動します。

   ```
   cd /mnt
   ```
4. 問題の発生している設定ファイルを、viエディタで開きます。次に、設定内容を修正して、ファイルを保存します。
5. レスキューシステムから、ルートファイルシステムをアンマウントします。

   ```
   umount /mnt
   ```
6. コンピュータを再起動します。

36.6.2.2 ファイルシステムの修復と確認

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しか方法はありません。レスキューシステムには、`btrfs` 、`ext2`、`ext3`、`ext4`、`reiserfs`、`xfs`、`dosfs`、および `vfat` の各ファイルシステムを確認し、修復するユーティリティが用意されています。コマンド `fsck.FILESYSTEM` を探します。たとえば、`btrfs` のファイルシステムを確認する必要がある場合、`fsck.btrfs` を使用します。

36.6.2.3 インストール済みシステムへのアクセス

レスキューシステムからインストール済みのシステムにアクセスする必要がある場合は、それをchange root(ルート変更)環境で行う必要があります。これは、たとえば、ブートローダの設定を変更したり、ハードウェア設定ユーティリティを実行するために行います。

インストール済みシステムに基づいたchange root(ルート変更)環境を設定するには、以下の手順に従ってください。

1. まず、インストールしたシステムからのルートパーティションとデバイスファイルシステムをマウントします(デバイス名を現在の設定に変更します)。

   ```
   mount /dev/sda6 /mnt
   mount --bind /dev /mnt/dev
   ```

2. 新しい環境に「change root」(ルート変更)します。

   ```
   chroot /mnt
   ```

3. `/proc` および `/sys` をマウントします。

   ```
   mount /proc
   mount /sys
   ```

4. 最後に、インストール済みシステムから、残りのパーティションをマウントします。

   ```
   mount -a
   ```

5. これで、インストール済みシステムにアクセスできるようになります。システムを再起動する前に、`umount -a` を使ってパーティションをアンマウントし、`exit` コマンドを実行して「change root」(ルート変更)環境を終了してください。

警告: 制限

インストール済みシステムのファイルやアプリケーションにフルアクセスできますが、いくつかの制限事項もあります。実行中のカーネルは、レスキューシステムでブートされたカーネルであり、ルート変更環境でブートされたカーネルではありません。このカーネルは、必要最低限のハードウェアしかサポートしておらず、カーネルのバージョンが完全に一致しない限り、インストール済みシステムからカーネルモジュールを追加することはできません。常に、現在実行中の(レスキュー)カーネルのバージョンを`uname -r`でチェックし、次に、一致するサブディレクトリがchange root環境の`/lib/modules`ディレクトリに存在するかどうか調べてください。存在する場合は、インストールされたモジュールを使用できます。そうでない場合は、フラッシュディスクなど、他のメディアにある正しいバージョンを提供する必要があります。多くの場合、レスキューカーネルのバージョンは、インストールされているバージョンと異なります。その場合は、たとえば、サウンドカードなどに簡単にアクセスすることはできません。また、GUIも利用できません。

また、<`Alt`–`F1`>から<`Alt`–`F6`>を使ってコンソールを切り替えると、「change root」(ルート変更)環境は終了することに注意してください。

36.6.2.4 ブートローダの変更と再インストール

場合によっては、ブートローダが壊れてしまい、システムをブートできなくなることもあります。たとえば、ブートローダが正常に機能しないと、起動ルーチンは物理ドライブとそのLinuxファイルシステム中の場所とを関連付けられず、正常な処理を行うことができません。

ブートローダの設定を確認し、ブートローダを再インストールするには、次の手順に従います。

1. 36.6.2.3項「インストール済みシステムへのアクセス」の説明に従って、インストール済みシステムにアクセスするために必要な作業を行います。
2. 次のファイルが第12章 ブートローダGRUB 2に示されているGRUB 2の設定ルールに従って正しく設定されているかどうかチェックし、必要に応じて修正します。
   - `/etc/default/grub`
   - `/boot/grub2/device.map` (オプションファイルで、手動で作成した場合にのみ存在します。)
   - `/boot/grub2/grub.cfg` (このファイルが生成されます。編集しないでください。)
   - `/etc/sysconfig/bootloader`
3. 次のコマンドシーケンスを使って、ブートローダを再インストールします。

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. パーティションをアンマウントして、「」「change root」(ルート変更)環境からログアウトします。次に、システムを再起動します。

```
umount -a
exit
reboot
```

36.6.2.5 カーネルインストールの修復

カーネルアップデートによって、システムの操作に影響する可能性のある新しいバグが導入される場合があります。たとえば、一部のシステムハードウェアのドライバに障害が発生し、そのハードウェアのアクセスや使用ができなくなることがあります。その場合は、機能した最後のカーネルに戻すか(システムで使用可能な場合)、インストールメディアから元のカーネルをインストールします。

ヒント: 更新後も最後のカーネルを保持する方法

正常でないカーネルアップデート後にブートできなくなることを防ぐには、カーネルの複数バージョン機能を使用して、更新後にどのカーネルを保持するか `libzypp` に指示します。

たとえば、最後の2つのカーネルと現在実行中のカーネルを常に保持するには、次のコードを、

```
multiversion.kernels = latest,latest-1,running
```

`/etc/zypp/zypp.conf` ファイルに追加します。詳細については、ブック「導入ガイド」11「複数バージョンのカーネルのインストール」を参照してください。

また、SUSE Linux Enterprise Serverでサポートされていないデバイスのドライバが破損し、その再インストールまたはアップデートが必要な場合があります。たとえば、ハードウェアベンダが、ハードウェアRAIDコントローラなどの特定のデバイスを使用している場合は、オペレーティングシステムによって認識されるバイナリドライバが必要です。ベンダは、通常、要求されたドライバの修正または更新バージョンを含むドライバアップデートディスク(DUD)をリリースします。

両方のケースで、レスキューモードでインストールされているシステムにアクセスし、カーネル関係の問題を修正する必要があります。さもないと、システムが正しくブートしないことがあります。

1. SUSE Linux Enterprise Serverメディアからのブート

2. 正常でないカーネルアップデート後に修復を行っている場合、次のステップはスキップしてください。DUD(ドライバアップデートディスク)を使用する必要がある場合は、F6 を押して、ブートメニューの表示後にドライバアップデートをロードし、 ドライバアップデートへのパスまたはURLを選択して、[はい]をクリックして確認します。

3. ブートメニューから[レスキューシステム]を選択し、< Enter >を押します。DUDの使用を選択した場合は、ドライバアップデートの保存先を指定するように要求されます。

4. `Rescue:` プロンプトに「`root`」と入力します。パスワードは必要ありません。

5. ターゲットシステムを手動でマウントし、新しい環境に「change root」(ルート変更)します。詳細については、36.6.2.3項「インストール済みシステムへのアクセス」を参照してください。

6. DUDを使用する場合は、障害のあるデバイスドライバパッケージのインストール/再インストール/アップデートを行います。インストールされたカーネルバージョンがインストールするドライバのバージョンと正確に一致することを常に確認してください。
   障害のあるカーネルアップデートのインストールを修復する場合は、次の手順で、インストールメディアから元のカーネルをインストールできます。

   a. DVDデバイスを`hwinfo --cdrom`で識別し、識別したデバイスを`mount /dev/sr0 /mnt`でマウントします。

   b. DVD上のカーネルファイルが保存されているディレクトリにナビゲートします(たとえば、`cd /mnt/suse/x86_64/`)。

   c. 必要なパッケージ`kernel-*`、`kernel-*-base`、および`kernel-*-extra`のカスタマイズしたバージョンを、`rpm -i`コマンドでインストールします。

7. 設定ファイルを更新し、必要に応じてブートローダを再初期化します。詳細については、36.6.2.4項「ブートローダの変更と再インストール」を参照してください。

8. システムドライブからブート可能なメディアをすべて除去し、再起動します。

36.7 IBM System z: initrdのレスキューシステムとしての使用

SUSE® Linux Enterprise Server for IBM System zのカーネルをアップグレードまたは変更した場合、何らかの原因でシステムが不整合な状態で再起動されると、インストールされているシステムのIPL標準処理が失敗する可能性があります。このような場合は、インストールシステムをレスキューのために使用できます。

SUSE Linux Enterprise Server for IBM System zのインストールシステムをIPL (再起動)します(ブック「導入ガイド」4「IBM System zでのインストール」4.2「インストールの準備」を参照)。[Start Installation (インストールの開始)]を選択し、必要なパラメータをすべて入力します。インストールシステムがロードされて、インストールの制御にどの表示タイプを使用するか尋ねられたら、[SSH]を選択します。これで、パスワードを使用せずに、root としてSSHを使用してシステムにログインできるようになります。

この状態では、設定されているディスクはありません。作業を続行する前に、ディスクを設定する必要があります。

手順 36.8 DASDの設定

1. DASDを設定するには、以下のコマンドを使用します。

```
dasd_configure 0.0.0150 1 0
```

ここで、「0.0.0150」は、DASDが接続されているチャネルを表します。1 は、ディスクをアクティブにすることを表しています(ここに 0 を指定すると、ディスクが無効になる)。0 は、ディスクに「DIAGモード」でアクセスしないことを表します(ここに 1 を指定すると、ディスクへのDAIGアクセスが有効になります)。

2. DASDがオンラインになり(cat /proc/partitions で確認)、コマンドを使用できるようになります。

手順 36.9 ZFCPディスクの設定

1. zFCPディスクを設定するには、まずzFCPアダプタを設定する必要があります。そのためには次のコマンドを使用します。

```
zfcp_host_configure 0.0.4000 1
```

0.0.4000 はアダプタが接続されているチャネルを、1 (ここに 0 を指定するとアダプタが無効になる)はアクティブにすることを示します。

2. アダプタをアクティブにしたら、ディスクを設定することができます。そのためには次のコマンドを使用します。

```
zfcp_disk_configure 0.0.4000 1234567887654321 8765432100000000  1
```

`0.0.4000`は前に使われていたチャネルIDを、`1234567887654321`はWWPN(World wide Port Number)を、そして`8765432100000000`はLUN(論理ユニット番号)を表しています。`1`(ここに`0`を指定するとディスクが無効になる)は、ディスクをアクティブにすることを表しています。

3. zFCPディスクがオンラインになり(`cat /proc/partitions`で確認)、コマンドを使用できるようになります。

これで、レスキューシステムが完全に設定され、インストールされたシステムの修復を開始できます。最も一般的な問題の修復方法については、36.6.2項「レスキューシステムの使用」を参照してください。

A　マニュアルの更新

この章では、SUSE® Linux Enterprise Server 11 SP3のリリース以降に、このドキュメントのコンテンツに加えられた変更点を一覧します。

このマニュアルは次の日付に更新されました。

- A.1項「2014年10月(SUSE Linux Enterprise Server 12の初期リリース)」

A.1　2014年10月(SUSE Linux Enterprise Server 12の初期リリース)

全般

- KDEは現在では出荷されていないため、KDEに関するマニュアルと参照をすべて削除しました。
- SuSEconfigはサポートされなくなったため、SuSEconfigの参照をすべて削除しました(Fate#100011)。
- System V initからsystemdに移行しました(Fate#310421)。マニュアルの関連部分を更新しました。
- YaST Runlevel Editorをサービスマネージャに変更しました(Fate#312568)。マニュアルの関連部分を更新しました。
- ISDNはサポートされなくなったため、ISDNのサポートに関する参照をすべて削除しました(Fate#314594)。
- YaST DSLモジュールは現在では出荷されていないため、このモジュールに関する参照をすべて削除しました(Fate#316264)。
- YaSTモデムモジュールは現在では出荷されていないため、このモジュールに関する参照をすべて削除しました(Fate#316264)。
- Btrfsがrootパーティションのデフォルトのファイルシステムになりました(Fate#315901)。マニュアルの関連部分を更新しました。
- `dmesg`で、`ctime()`と同様の形式で、ユーザが判読可能なタイムスタンプが提供されるようになりました(Fate#316056)。マニュアルの関連部分を更新しました。

- syslogおよびsyslog-ngがrsyslogに置き換えられました(Fate#316175)。マニュアルの関連部分を更新しました。
- MySQLの代わりにMariaDBがリレーショナルデータベースとして付属するようになりました(Fate#313595)。マニュアルの関連部分を更新しました。
- SUSE関連製品は、http://download.novell.com ではなくhttp://download.suse.com から利用できるようになりました。それに応じてリンクを調整しました。
- Novellカスタマーセンターは SUSEカスタマーセンターに置き換えられました。マニュアルの関連部分を更新しました。
- `/var/run` はtmpfsとしてマウントされます(Fate#303793)。マニュアルの関連部分を更新しました。
- ItaniumおよびX86アーキテクチャはサポートされなくなりました。マニュアルの関連部分を更新しました。
- `ifconfig` を使用した従来のネットワークセットアップ方法が `wicked` に置き換えられました。マニュアルの関連部分を更新しました。
- 多くのネットワーキングコマンドが非推奨になり、新しいコマンド(ほとんどの場合は `ip`)に置き換えられました。マニュアルの関連部分を更新しました。

  `arp`: `ip neighbor`
  `ifconfig`: `ip addr`, `ip link`
  `iptunnel`: `ip tunnel`
  `iwconfig`: `iw`
  `nameif`: `ip link`, `ifrename`
  `netstat`: `ss`, `ip route`, `ip -s link`, `ip maddr`
  `route`: `ip route`

- テクニカルフィードバックに基づいて、マニュアルに対してさまざまな細かい修正と追加を行いました。

第1章 YaSTオンラインアップデート

- YaSTで、デルタRPMの使用を有効または無効にするオプションが提供されるようになりました(Fate#314867)。
- 再起動が必要なパッチをインストールする前に、YaSTから通知を受信し、処理方法を選択できます。

第2章 サポート用システム情報の収集

- 2.1項「現在のシステム情報の表示」のセクションを追加しました(Fate#315869)。
- Supportconfig Analysis (SCA)ツールとアプライアンスに関するセクションが追加されました:2.4項「システム情報の分析」(Fate#315699)。
- 2.5項 「カーネルモジュールのサポート」のセクションを追加しました(http://bugzilla.novell.com/show_bug.cgi?id=869159)。
- 章を更新して再構成しました。

第3章 テキストモードのYaST

- ソフトウェアインストールモジュールでパッケージをフィルタおよび選択する方法に関する情報を追加しました。

第4章 Snapperを使用したシステムの回復とスナップショット管理

- 章を更新し、新機能を追加しました(Fate#312751、Fate#316238、Fate#316233、Fate#316232、Fate#316222、Fate#316203、Fa
- 4.3項 「スナップショットからのブートによるシステムロールバック」のセクションを追加しました(Fate#316231、Fate#316221、Fate#316541、Fate#316522)。

第5章 VNCによるリモートアクセス

- デフォルトのVNCビューアが `tigervnc` になりました。
- 永続的なVNCセッションにおけるウィンドウマネージャの起動に関する修正を追加しました。

第6章 コマンドラインツールによるソフトウェアの管理

- Zypperのrug互換性モードに関するマニュアルを削除しました(Fate#317708)。
- 6.1.5項「Zypperによるリポジトリおよびパッケージのクエリ」が書き直されました。

第9章 Linuxシステムのブート

- System V initがsystemdに置き換えられたため、章が大幅に短くなりました。systemdは別の章で説明されています: 第10章 systemdデーモン。

第10章 systemdデーモン

- systemdとYaSTサービスマネージャに関する新しい章を追加しました(Fate#316631、Fate#312568)。
- カーネルモジュールのロードに関する新しいセクション(http://bugzilla.novell.com/show_bug.cgi?id=892349)。

第11章 journalctl:systemdジャーナルのクエリ

新しい章を追加しました(http://bugzilla.novell.com/show_bug.cgi?id=878352)。

第12章 ブートローダGRUB 2

- GRUB LegacyのマニュアルをGRUB 2に関する新しい章に入れ替えました。
- LILOのサポートは廃止されました。
- 新たに12.4項「System zにおける端末の使用上の相違点」のセクションを追加しました。

第13章 UEFI (Unified Extensible Firmware Interface)

- 章を更新し、新機能を追加しました(Fate#314510、Fate#316365)。
- http://doccomments.provo.novell.com/comments/25080を修正しました。

第15章 プリンタの運用

新しいCUPSバージョンに従って章とセクションを更新し、共通の印刷データ形式としてPDF化しました(Fate#314630)。

第17章 X Windowシステム

- 各起動時のダイナミックコンフィギュレーションを反映するため章を更新しました。
- 17.1項「フォントのインストールと設定」を更新しました。

第19章 ネットワークの基礎

- NetworkManagerがWorkstation Extensionの一部になりました: 19.4.1.1項「グローバルネットワークオプションの設定」(Fate#316888)。
- ネットワーク設定用の新しい`wicked`フレームワークに関するセクションを追加しました: 19.5項「ネットワークの手動環境設定」(Fate#316649)。
- `/etc/resolv.conf`に追加できる他のオプションの説明を追加しました: 19.5.2項「環境設定ファイル」(Fate#316048)。

第20章 SLP

- 章を書き直し、`slptool`コマンドに関する情報が大幅に増えました。

第22章 ドメインネームシステム

- YaST DNSモジュールがフォワーダの設定をサポートするようになりました(Fate#309036)。

第23章 DHCP

- dhcpcdは出荷停止になったため削除されました(Fate#316111)。

第25章 Samba

- 25.8項「詳細トピック」のセクションを追加しました。
- 25.8.1項「Btrfsでの透過的なファイル圧縮」のセクションを追加しました。
- 25.8.2項「スナップショット」のセクションを追加しました。

第26章 NFS共有ファイルシステム

- NFSv4共有の設定がNFSv3とほぼ同様になりました。特に、以前は必須だったバインドマウント設定は非推奨になりました(Fate#315589)。
- エクスポート元サーバでNFSボリュームをローカルにマウントすることはサポートされていません。

第27章 Autofsによるオンデマンドマウント

- `autofs`に関する章を追加しました(Fate#316185)。

第29章 Apache HTTPサーバ

- monoおよびmod_monoはディストリビューションから削除されたため、これらの参照を削除しました。
- 章をApacheバージョン2.4に更新しました(Fate#316067)。
- 非推奨になったディレクティブ`NameVirtualHost`を削除し、それに応じて29.2.2.1項「仮想ホスト設定」を更新しました。
- `Order`、`Allow`、および`Deny`ディレクティブを標準の`Require`に更新しました。
- 29.6項「SSLをサポートするセキュアWebサーバのセットアップ」から仮想の「怪しげ」な会社を削除しました。

第30章 YaSTを使用したFTPサーバの設定

- `pure-ftpd`は廃止されました(Fate #315176、Fate#316308)。

第34章 電源管理

- `pm-utils`パッケージへの古い参照を削除しました。

第36章 最も頻繁に起こる問題およびその解決方法

- 新たに36.3.3項「ルートBtrfsパーティションをマウントできない」のセクションを追加しました(Fate#308679、Fate#315126)。
- 非推奨になったYaST修復モジュールに関するセクションを削除しました(Fate#308679)。

Wi-Fi設定

- Wi-Fi設定はNetworkManagerで実行できるため、YaSTを使用したWi-Fi設定に関する章を削除しました: 第24章 NetworkManagerの使用。

タブレットPC

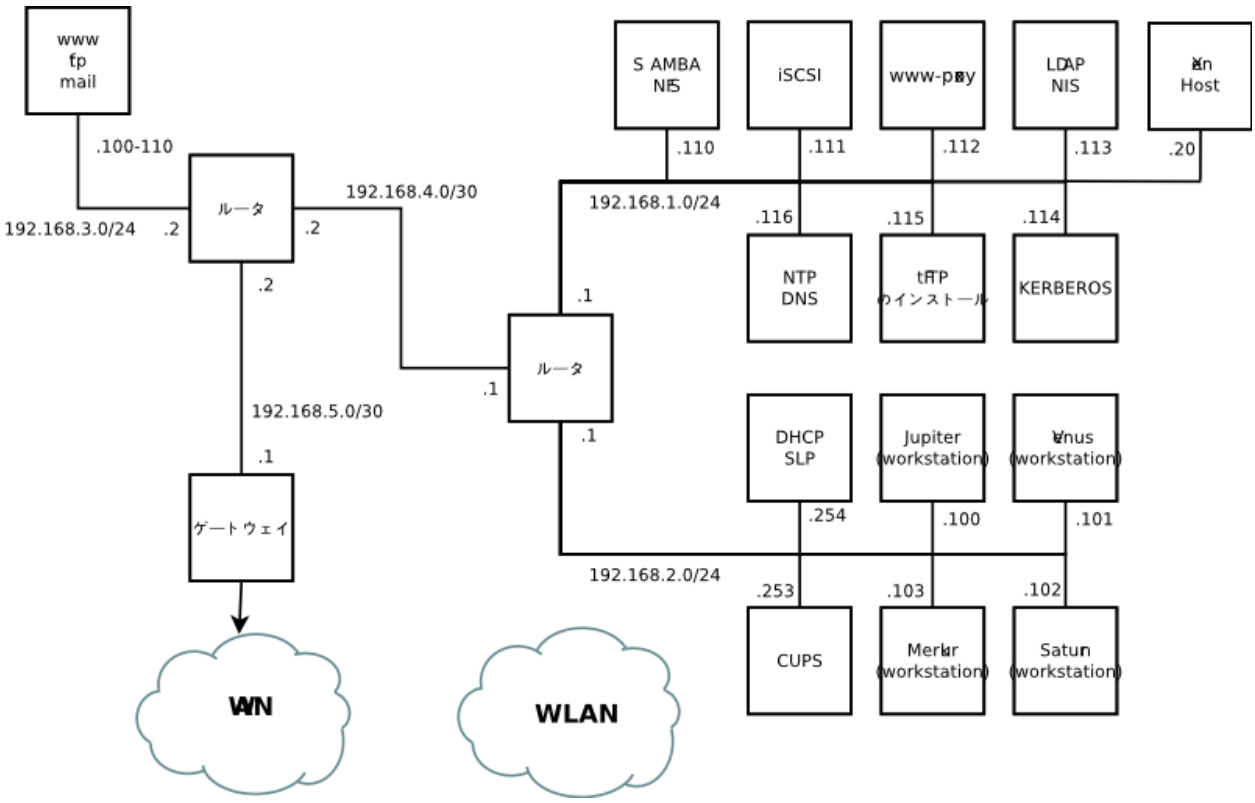
- タブレットPCに関する非推奨になった章を削除しました。

バグ修正

- 2.5項 「カーネルモジュールのサポート」のセクションを追加しました(http://bugzilla.novell.com/show_bug.cgi?id=869159)。
- 新たに第11章 `journalctl`:systemdジャーナルのクエリ章を追加しました(http://bugzilla.novell.com/show_bug.cgi?id=878352)。

B サンプルネットワーク

このサンプルネットワークは、SUSE® Linux Enterprise Serverマニュアルのすべてのネットワーク関連の章で使用されます。



C GNU Licenses

This appendix contains the GNU Free Documentation License version 1.2.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy

of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/ ↗.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
  Copyright (c) YEAR YOUR NAME.
  Permission is granted to copy, distribute and/or modify this document
  under the terms of the GNU Free Documentation License, Version 1.2
  or any later version published by the Free Software Foundation;
  with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.
  A copy of the license is included in the section entitled “GNU
  Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
  with the Invariant Sections being LIST THEIR TITLES, with the
  Front-Cover Texts being LIST, and with the Back-Cover Texts being
LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.