



SUSE Linux Enterprise Desktop 15 SP5

管理ガイド

管理ガイド

SUSE Linux Enterprise Desktop 15 SP5


このガイドでは、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: 2025 年 4 月 03 日

<https://documentation.suse.com> 

Copyright © 2006–2025 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、配布、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の商標については、<http://www.suse.com/company/legal/> を参照してください。サードパーティ各社とその製品の商標は、所有者であるそれぞれの会社に所属します。商標記号(®、™など)は、SUSEおよび関連会社の商標を示します。アスタリスク(*)は、第三者の商標を示します。

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

序文 xx

- 1 利用可能なマニュアル xx
- 2 ドキュメントの改善 xx
- 3 マニュアルの表記規則 xxi
- 4 サポート xxiii
 - SUSE Linux Enterprise Desktopのサポートステートメント xxiii • 技術プレビュー xxiv

I 共通のタスク 1

1 BashとBashスクリプト 2

- 1.1 「シェル」とは何か? 2
 - Bashの環境設定ファイル 2 • ディレクトリ構造 5
- 1.2 シェルスクリプトの作成 9
- 1.3 コマンドイベントのリダイレクト 11
- 1.4 エイリアスの使用 12
- 1.5 Bashでの変数の使用 12
 - 引数変数の使用 13 • 変数置換の使用 14
- 1.6 コマンドのグループ化と結合 15
- 1.7 よく使用されるフローコンストラクトの操作 16
 - if制御コマンド 16 • forコマンドによるループの作成 16
- 1.8 詳細情報 17

2 sudoの基本 18

- 2.1 sudoの基本的な使用方法 18
 - 単一コマンドの実行 18 • シェルの起動 19

- 2.2 **sudoの構成 20**
設定ファイルの編集 20 • 基本的なsudoersの設定構文 21 • 基本的なsudoersのルール 22
- 2.3 **sudoの使用例 24**
rootパスワードなしの**sudo**の使用 24 • X.Orgアプリケーションでの**sudo**の使用 26
- 2.4 詳細情報 26
- 3 YaSTの使用 27**
- 3.1 YaSTインタフェースの概要 27
- 3.2 便利なキーの組み合わせ 27
- 4 テキストモードのYaST 29**
- 4.1 モジュールでのナビゲーション 30
- 4.2 高度なキーの組み合わせ 31
- 4.3 キーの組み合わせの制約 32
- 4.4 YaSTコマンドラインオプション 33
コマンドラインからのパッケージのインストール 33 • 個々のモジュールの使用 33 • YaSTモジュールのコマンドラインパラメータ 34
- 5 YaSTによる言語および国の設定の変更 59**
- 5.1 システム言語の変更 59
YaSTでシステムの言語を変更する 60 • デフォルトシステム言語を切り替える 62 • 標準のXアプリケーションとGNOMEアプリケーションでの言語切り替え 63
- 5.2 国および時間の設定の変更 63
- 6 YaSTによるユーザの管理 66**
- 6.1 [ユーザとグループの管理] ダイアログ 66
- 6.2 ユーザアカウントの管理 68

- 6.3 ユーザアカウントの追加オプション 70
自動ログインおよびパスワードレスログイン 70 • パスワードポリシーの強制 70 • クォータの管理 71
- 6.4 ローカルユーザのデフォルト設定の変更 74
- 6.5 グループへのユーザの割り当て 74
- 6.6 グループを管理する 75
- 6.7 ユーザ認証方法を変更する 76
- 6.8 デフォルトのシステムユーザ 78
- 7 YaSTオンラインアップデート 81**
 - 7.1 オンライン更新ダイアログ 82
 - 7.2 パッチのインストール 83
 - 7.3 撤回されたパッチの表示 84
 - 7.4 自動オンラインアップデート 85
- 8 ソフトウェアをインストールまたは削除する 88**
 - 8.1 用語の定義 88
 - 8.2 インストール済みシステムの登録 90
YaSTでの登録 90 • SUSEConnectを使用した登録 90
 - 8.3 YaSTソフトウェアマネージャの使用 90
ソフトウェアの検索 91 • パッケージまたはパターンのインストールと削除 93 • パッケージの更新 95 • パッケージの依存関係 97 • 推奨パッケージの取り扱い 98
 - 8.4 ソフトウェアリポジトリおよびサービスの管理 99
ソフトウェアリポジトリの追加 99 • リポジトリプロパティの管理 101 • リポジトリキーの管理 102
 - 8.5 GNOMEパッケージアップデータ 102
 - 8.6 GNOMEソフトウェアを使用したパッケージの更新 105

9 コマンドラインツールによるソフトウェアの管理 107

9.1 Zypperの使用 107

一般的な使用方法 107 • Zypperサブコマンドの使用 109 • Zypperを使ったソフトウェアのインストールと削除 109 • Zypperによるソフトウェアの更新 114 • 削除されたファイルを使用しているプロセスとサービスの特定 120 • Zypperによるリポジトリの管理 122 • Zypperによるリポジトリおよびパッケージのクエリ 124 • ライフサイクル情報の表示 126 • Zypperの設定 127 • トラブルシューティング 127 • BtrfsファイルシステムでのZypperロールバック機能 127 • 詳細情報 128

9.2 RPM - パッケージマネージャ 128

パッケージの信頼性の検証 129 • 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 129 • デルタRMPパッケージ 131 • RPM クエリー 131 • ソースパッケージのインストールとコンパイル 134 • buildによるRPMパッケージのコンパイル 136 • RPMアーカイブとRPMデータベース用のツール 137

10 Snapperを使用したシステムの回復とスナップショット管理 138

10.1 デフォルト設定 139

デフォルト設定 140 • スナップショットのタイプ 140 • スナップショットから除外されるディレクトリ 141 • 設定のカスタマイズ 142

10.2 Snapperを使用した変更の取り消し 145

YaSTおよびZypperによる変更の取り消し 146 • Snapperを使用したファイルの復元 151

10.3 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 153

ロールバック後のスナップショット 155 • スナップショットブートエントリのアクセスと識別 156 • 制限 158

10.4 ユーザホームディレクトリでのSnapperの有効化 159

pam_snapperのインストールとユーザの作成 160 • ユーザを削除する 161 • ホームディレクトリでのスナップショットの手動有効化 161

10.5 Snapper設定の作成と変更 161

既存の設定の管理 163

- 10.6 スナップショットの手動での作成と管理 166
スナップショットのメタデータ 167 • スナップショットの作成 169 • スナップショットのメタデータ修正 170 • スナップショットの削除 171
- 10.7 スナップショットの自動クリーンアップ 172
番号付きスナップショットのクリーンアップ 172 • タイムラインスナップショットのクリーンアップ 174 • 違いがないスナップショットのペアのクリーンアップ 176 • 手動で作成されたスナップショットのクリーンアップ 176 • ディスククォータサポートの追加 177
- 10.8 スナップショットが使用する排他的なディスク容量の表示 178
- 10.9 ホットスポットに関する一般的な質問とその回答 (FAQ) 180
- 11 KLPによるライブカーネルパッチ 182**
 - 11.1 カーネルライブパッチの利点 182
 - 11.2 カーネルライブパッチの概要 182
カーネルライブパッチのスコープ 184 • カーネルライブパッチの制限 184
 - 11.3 YaSTを使用したカーネルライブパッチの有効化 184
 - 11.4 コマンドラインからのカーネルライブパッチの有効化 185
 - 11.5 カーネルライブパッチの適用 186
ライブパッチの有効期限を確認する 187
 - 11.6 カーネルライブパッチの問題のトラブルシューティング 187
手動によるパッチのダウングレード 187
- 12 ユーザ空間ライブパッチ 189**
 - 12.1 ユーザ空間ライブパッチについて 189
前提条件 189 • サポートされているライブラリ 190 • libpulpの使用 190
 - 12.2 詳細情報 192
- 13 トランザクション更新 193**
 - 13.1 テクノロジープレビューの制限 193
 - 13.2 transactional-update有効化 195

- 13.3 自動更新の管理 195
- 13.4 **transactional-update** コマンド 196
- 13.5 トラブルシューティング 199
- 14 VNCによるリモートグラフィカルセッション 200**
 - 14.1 **vncviewer** クライアント 200
 - vncviewer CLIを使用した接続 200
 - vncviewer GUIを使用した接続 201
 - 暗号化されていない接続の通知 201
 - 14.2 Remmina: リモートデスクトップクライアント 202
 - インストール 202
 - メインウィンドウ 202
 - リモートセッションの追加 202
 - リモートセッションの開始 204
 - 保存されたセッションの編集、コピー、および削除 205
 - コマンドラインからのリモートセッションの実行 205
 - 14.3 VNCサーバでのワンタイムセッションの設定 206
 - 使用可能な設定 207
 - 一時的VNCセッションを開始する 208
 - 一時的VNCセッションを設定する 208
 - 14.4 永続的VNCサーバセッションを設定する 209
 - vncserverを使用して開始されたVNCセッション 210
 - vncmanagerを使用して開始されたVNCセッション 211
 - 14.5 VNCサーバでの暗号化の設定 215
 - 14.6 Waylandとの互換性 216
- 15 rsyncによるファイルのコピー 217**
 - 15.1 概念の概要 217
 - 15.2 基本的な構文 217
 - 15.3 ファイルとディレクトリのローカルでのコピー 218
 - 15.4 ファイルとディレクトリのリモートでのコピー 219
 - 15.5 rsyncサーバの設定と使用 219
 - 15.6 詳細情報 222

II LINUXシステムのブート 223

16 ブートプロセスの概要 224

16.1 用語集 224

16.2 Linuxのブートプロセス 225

初期化とブートローダの段階 225 • カーネルの段階 227 • initramfs上のinit段階 229 • systemd段階 232

17 UEFI (Unified Extensible Firmware Interface) 233

17.1 セキュアブート 233

SUSE Linux Enterprise Desktop上の実装 234 • Machine Owner Key(マシン所有者キー、MOK) 236 • カスタムカーネルのブート 237 • Inbox以外のドライバの使用 239 • 機能と制限 240

17.2 詳細情報 241

18 ブートローダGRUB 2 242

18.1 GRUB LegacyとGRUB 2の主な相違点 242

18.2 設定ファイルの構造 242

/boot/grub2/grub.cfgファイル 243 • /etc/default/grubファイル 244 • /etc/grub.d内のスクリプト 247 • BIOSドライブとLinuxデバイスのマッピング 248 • ブート手順実行中のメニューエントリの編集 249 • ブートパスワードの設定 250 • ブートメニューエントリへの許可されたアクセス 251

18.3 YaSTによるブートローダの設定 252

ブートローダの場所とブートコードオプション 253 • ディスクの順序の変更 255 • 詳細オプションの設定 256

18.4 役立つGRUB 2コマンド 259

18.5 詳細情報 260

19 systemdデーモン 261

19.1 systemdの概念 261

ユニットファイル 262

- 19.2 基本的な使用方法 263
 - 稼働中のシステムでのサービスの管理 263 • サービスの恒久的な有効化/無効化 265
- 19.3 システムの起動とターゲットの管理 266
 - ターゲットとランレベルの比較 267 • システム起動のデバッグ 270 • System Vとの互換性 273
- 19.4 YaSTを使用したサービスの管理 274
- 19.5 カスタマイズsystemd 275
 - ユニットファイルのカスタマイズ 275 • ドロップインファイルの作成 276 • xinetdサービスをsystemdに変換する 277 • カスタムターゲットの作成 278
- 19.6 高度な使用方法 279
 - 一時ディレクトリの消去 279 • システムログ 280 • スナップショット 280 • カーネルモジュールのロード 281 • サービスのロード前にアクションを実行 281 • カーネルのコントロールグループ(cgroup) 282 • サービスの終了(シグナルの送信) 283 • D-Busサービスに関する重要な注意事項 284 • サービスのデバッグ 284
- 19.7 systemdタイマユニット 285
 - systemdタイマタイプ 286 • systemdタイマとサービスユニット 286 • 実例 286 • systemdタイマの管理 288
- 19.8 詳細情報 288

III システム 290

20 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 291

- 20.1 ランタイムサポート 291

- 20.2 カーネル仕様 292

21 journalctl: systemdジャーナルのクエリ 293

- 21.1 ジャーナルの永続化 293
- 21.2 journalctl: 便利なスイッチ 294

- 21.3 ジャーナル出力のフィルタ 295
 - ブート番号に基づくフィルタ 295 • 時間間隔に基づくフィルタ 295 • フィールドに基づくフィルタ 296
- 21.4 systemdエラーの調査 297
- 21.5 Journaldの設定 298
 - ジャーナルサイズ制限の変更 298 • ジャーナルの/dev/ttyXへの転送 298 • ジャーナルのsyslog機能への転送 299
- 21.6 YaSTを使用したsystemdジャーナルのフィルタ 299
- 21.7 GNOMEでのログの表示 300
- 22 update-alternatives: 複数のバージョンのコマンドとファイルの管理 301**
 - 22.1 概要 301
 - 22.2 使用例 303
 - 22.3 alternativesの概要の取得 303
 - 22.4 特定のalternativesに関する詳細の表示 303
 - 22.5 alternativesのデフォルトバージョンの設定 304
 - 22.6 カスタムalternativesのインストール 305
 - 22.7 依存するalternativesの定義 306
- 23 ネットワークの基礎 309**
 - 23.1 IPアドレスとルーティング 312
 - IPアドレス 312 • ネットマスクとルーティング 312
 - 23.2 IPv6 - 次世代インターネット 314
 - 長所 315 • アドレスのタイプと構造 317 • IPv4とIPv6の共存 321 • IPv6の設定 322 • 詳細情報 322
 - 23.3 ネームレゾリューション 323
 - 23.4 YaSTによるネットワーク接続の設定 324
 - YaSTでのネットワークカードの設定 325

- 23.5 NetworkManager 337
 - NetworkManagerとwicked 337 • NetworkManagerの機能および環境設定ファイル 338 • NetworkManager機能の制御およびロックダウン 339
- 23.6 ネットワーク接続の手動環境設定 339
 - wickedネットワーク環境設定 339 • 環境設定ファイル 346 • 設定のテスト 358 • ユニットファイルと起動スクリプト 362
- 23.7 ボンディングデバイスの設定 363
 - Bondポートのホットプラグ 365
- 23.8 ネットワークチーミング用チームデバイスの設定 367
 - 使用事例: ネットワークチーミングによる負荷分散 370 • 使用事例: ネットワークチーミングによるフェールオーバー 371 • 使用事例: チームデバイス上でのVLAN 373
- 24 プリンタの運用 375**
 - 24.1 CUPSのワークフロー 376
 - 24.2 プリンタに接続するための方法とプロトコル 377
 - 24.3 ソフトウェアのインストール 377
 - 24.4 ネットワークプリンタ 378
 - 24.5 コマンドラインツールによるCUPSの設定 379
 - 24.6 コマンドラインからの印刷 381
 - 24.7 SUSE Linux Enterprise Desktopの特別な機能 381
 - CUPSとファイアウォール 381 • ネットワークプリンタの参照 382 • 各種パッケージ内のPPDファイル 382
 - 24.8 トラブルシューティング 383
 - 標準的なプリンタ言語をサポートしないプリンタ 383 • 特定のPostScriptプリンタに適したPPDファイルが入手できない 384 • ネットワークプリンタ接続 384 • エラーメッセージを生成しない異常なプリントアウト 387 • 無効にされたキュー 387 • CUPS参照: 印刷ジョブの削除 387 • 異常な印刷ジョブとデータ転送エラー 388 • CUPSのデバッグ 388 • 詳細情報 389

25 グラフィカルユーザインタフェース 390

- 25.1 Xウィンドウシステム 390
- 25.2 フォントのインストールと設定 390
 - インストール済みフォントの表示 392 • フォントの表示 392 • フォントの問い合わせ 393 • フォントのインストール 393 • フォントの外観の設定 394
- 25.3 管理者用のGNOME設定 403
 - dconfシステム 403 • システム全体の設定 403 • Waylandでのタイリングウィンドウの管理 404 • 詳細情報 406
- 25.4 SUSE Primeを使用したIntelとNVIDIA Optimus GPUの切り替え 406
 - 前提条件 407 • SUSE Primeのインストールと使用 407 • NVIDIAドライバのインストール 408

26 FUSEによるファイルシステムへのアクセス 409

- 26.1 FUSEの設定 409
- 26.2 NTFSパーティションのマウント 409
- 26.3 詳細情報 410

27 複数バージョンのカーネルのインストール 411

- 27.1 マルチバージョンサポートの有効化と設定 411
 - 使用していないカーネルの自動削除 412 • 使用事例: 再起動後にのみ古いカーネルを削除する 413 • 使用事例: 古いカーネルをフォールバックとして保持する 414 • 使用事例: 特定のバージョンのカーネルの保持 414
- 27.2 YaSTによる複数のカーネルバージョンのインストールと削除 414
- 27.3 Zypperによる複数のカーネルバージョンのインストールと削除 415

28 カーネルモジュールの管理 418

- 28.1 lsmodおよびmodinfoによるロード済みモジュールの一覧作成 418
- 28.2 カーネルモジュールの追加と削除 419
 - ブート時のカーネルモジュールの自動ロード 419 • modprobeによるカーネルモジュールのブラックリスト化 420

29 udevによる動的カーネルデバイス管理 422

- 29.1 /devディレクトリ 422
- 29.2 カーネルueventsとudev 422
- 29.3 ドライバ、カーネルモジュールおよびデバイス 423
- 29.4 ブートおよび初期デバイスセットアップ 424
- 29.5 実行中のudevデーモンの監視 424
- 29.6 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 425
 - udevルールでの演算子の使用 427
 - udevルールでの置換の使用 428
 - udev一致キーの使用 429
 - udev割り当てキーの使用 430
- 29.7 永続的なデバイス名の使用 432
- 29.8 udevで使用するファイル 433
- 29.9 詳細情報 433

30 特別なシステム機能 435

- 30.1 特殊ソフトウェアパッケージに関する情報 435
 - bashパッケージと/etc/profile 435
 - cronパッケージ 436
 - cronステータスメッセージの停止 437
 - ログファイル: パッケージ logrotate 437
 - locateコマンド 437
 - ulimitコマンド 438
 - freeコマンド 439
 - manページとinfoページ 439
 - manコマンドを使用したマニュアルページの選択 439
 - GNU Emacs用の設定 440
- 30.2 バーチャルコンソール 441
- 30.3 キーボード割り当て 441
- 30.4 言語および国固有の設定 442
 - システム全体のロケール設定 443
 - 例 444
 - ~/i18nでのロケール設定 445
 - 言語サポートの設定 445
 - 詳細情報 446

31 NetworkManagerの使用 447

- 31.1 NetworkManagerの使用事例 447
- 31.2 NetworkManagerの有効化/無効化 448

- 31.3 ネットワーク接続の設定 448
 - 有線ネットワーク接続の管理 450 • ワイヤレスネットワーク接続の管理 451 • 無線でのキャプティブポータル検出の有効化 452 • Wi-Fi/Bluetoothカードのアクセスポイントとしての設定 452 • NetworkManagerとVPN 452
- 31.4 NetworkManagerとセキュリティ 454
 - ユーザおよびシステムの接続 455 • パスワードと資格情報の保存 455 • ファイアウォールゾーン 455
- 31.5 ホットスポットに関する一般的な質問とその回答 (FAQ) 456
- 31.6 トラブルシューティング 458
- 31.7 詳細情報 458

IV ハードウェア設定 460

32 システムのキーボードレイアウト設定 461

33 サウンドカードの設定 462

34 プリンタの設定 465

- 34.1 プリンタの環境設定 465
 - YaSTによるドライバの追加 466 • ローカルプリンタ設定の編集 468
- 34.2 YaSTによるネットワーク印刷の設定 468
 - CUPSの使用 468 • CUPS以外の印刷サーバの使用 470
- 34.3 ネットワーク上のプリンタ共有 470

35 スキャナの設定 471

- 35.1 HPのオールインワンデバイスの設定 471
- 35.2 ネットワーク経由のスキャナの共有 472
- 35.3 ネットワーク経由のスキャン 472

36 電源管理 473

- 36.1 省電力機能 473

36.2 ACIP(詳細設定と電源インタフェース) 474
CPUパフォーマンスの制御 475 • トラブルシューティング 475

36.3 ハードディスクの休止 477

36.4 トラブルシューティング 478
CPU周波数調節が機能しない 479

37 永続的なメモリ 480

37.1 概要 480

37.2 用語 482

37.3 使用例 484
DAXを使用したPMEM 484 • BTTを使用したPMEM 485

37.4 永続的なメモリを管理するためのツール 485

37.5 永続的なメモリのセットアップ 487
使用可能なNVDIMMストレージの表示 487 • DAXを使用した単一のPMEM
ネームスペースとしてストレージを設定する 488 • BTTを使用したPMEM
ネームスペースの作成 490 • PMEM/BTTにファイルシステムジャーナルを
配置する 491

37.6 詳細情報 492

V サービス 493

38 YaSTによるサービス管理 494

39 NTPによる時刻の同期 496

39.1 YaSTでのNTPクライアントの設定 497
NTPデーモン開始 497 • 設定元のタイプ 498 • タイムサーバの設定
498

39.2 ネットワークでのNTPの手動設定 499

39.3 **chronyc**を使用した実行時のchronydの設定 500

39.4 ランタイム時の動的時刻同期 501

39.5 ローカルリファレンスクロックの設定 501

39.6 ETR (External Time Reference)とのクロックの同期 502

VI トラブルシューティング 503

40 ヘルプとドキュメント 504

40.1 ドキュメントディレクトリ 504

SUSEマニュアル 505 • パッケージのマニュアル 505

40.2 マニュアルページ 506

40.3 情報ページ 508

40.4 リソースのオンライン化 508

41 サポート用システム情報の収集 510

41.1 現在のシステム情報の表示 510

41.2 supportconfigによるシステム情報の収集 511

サービス要求番号の作成 511 • アップロード先 512 • YaSTでのSupportconfigアーカイブの作成 512 • コマンドラインからのsupportconfigアーカイブの作成 515 • **supportconfig**の出力の理解 515 • supportconfigの一般的なオプション 517 • アーカイブコンテンツの概要 518

41.3 グローバルテクニカルサポートへの情報の送信 521

41.4 システム情報の分析 523

SCAコマンドラインツール 524 • SCAアプライアンス 525 • カスタム分析パターンの開発 537

41.5 インストール時の情報収集 537

41.6 カーネルモジュールのサポート 537

技術的背景 538 • サポート対象外のモジュールの使用 539

41.7 詳細情報 540

42 最も頻繁に起こる問題およびその解決方法 541

42.1 情報の検索と収集 541

42.2 ブートの問題 544

GRUB 2ブートローダをロードできない 544 • ログインなし、またはプロンプトが表示される 545 • グラフィカルログインがない 546 • ルートBtrfsパーティションをマウントできない 546 • ルートパーティションを強制的に確認する 546 • スワップを無効にしてブートを有効にする 547 • デュアルブートシステムでの再起動中にGRUB 2が失敗する 547

42.3 ログインの問題 547

有効なユーザ名とパスワードを使っても失敗する 547 • 有効なユーザ名とパスワードが受け付けられない 548 • 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 551 • GNOMEデスクトップに問題があります 551

42.4 ネットワークの問題 552

NetworkManagerの問題 556

42.5 データの問題 557

パーティションイメージの管理 557 • レスキューシステムの使用 558

A サンプルネットワーク 565

B GNU licenses 566

序文

1 利用可能なマニュアル

オンラインマニュアル

オンラインマニュアルは<https://documentation.suse.com> にあります。さまざまな形式のマニュアルをブラウズまたはダウンロードできます。



注記: 最新のアップデート

最新のアップデートは、通常、英語版マニュアルで入手できます。

リリースノート

リリースノートは<https://www.suse.com/releasesnotes/> を参照してください。


ご使用のシステムで


オフラインで利用するには、システムの `/usr/share/doc` にインストールされたマニュアルを確認してください。「マニュアルページ」には、多くのコマンドについても詳しく説明されています。説明を表示するには、`man` コマンドに確認したいコマンドの名前を付加して実行してください。システムに `man` コマンドがインストールされていない場合は、`sudo zypper install man` コマンドでインストールします。

2 ドキュメントの改善


このドキュメントに対するフィードバックや貢献を歓迎します。フィードバックを提供するための次のチャンネルが利用可能です。

サービス要求およびサポート

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。

サービス要求を提出するには、SUSE Customer Centerに登録済みのSUSEサブスクリプションが必要です。<https://scc.suse.com/support/requests> からログインして新規作成をクリックしてください。

バグレポート

<https://bugzilla.suse.com/> から入手できるドキュメントを使用して、問題を報告してください。

このプロセスを容易にするには、このドキュメントのHTMLバージョンの見出しの横にあるReport an issue (問題を報告する)アイコンをクリックしてください。これにより、Bugzillaで適切な製品とカテゴリが事前に選択され、現在のセクションへのリンクが追加されます。バグレポートの入力を直ちに開始できます。

Bugzillaアカウントが必要です。

ドキュメントの編集に貢献

このドキュメントに貢献するには、このドキュメントのHTMLバージョンの見出しの横にあるEdit source document (ソースドキュメントの編集)アイコンをクリックしてください。GitHubのソースコードに移動し、そこからプルリクエストをオープンできます。GitHubアカウントが必要です。



注記: Edit source document (ソースドキュメントの編集)は英語でのみ利用可能

Edit source document (ソースドキュメントの編集)アイコンは、各ドキュメントの英語版でのみ使用できます。その他の言語では、代わりにReport an issue (問題を報告する)アイコンを使用してください。

このドキュメントの編集に使用する環境の詳細は、<https://github.com/SUSE/doc-sle>にあるリポジトリのREADMEを参照してください。

メール

ドキュメントに関するエラーの報告やフィードバックはdoc-team@suse.com宛に送信してもかまいません。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記載してください。また、関連するセクション番号とタイトル(またはURL)、問題の簡潔な説明も記載してください。

3 マニュアルの表記規則

このマニュアルでは、次の通知と表記規則が使用されています。

- /etc/passwd: ディレクトリ名およびファイル名
- PLACEHOLDER: PLACEHOLDERは、実際の値で置き換えられます。
- PATH: 環境変数
- 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の名前

- `package_name`: ソフトウェアパッケージの名前
- `Alt`、`Alt + F1`: 押すキーまたはキーの組み合わせ。キーはキーボードのように大文字で表示されます。
- ファイル、ファイル > 名前を付けて保存: メニュー項目、ボタン
- Chapter 1, 「Example chapter」: このガイドの別の章への相互参照。
- `root`特権で実行する必要があるコマンド。多くの場合、これらのコマンドの先頭に `sudo` コマンドを置いて、特権のないユーザとしてコマンドを実行することもできます。

```
# command
> sudo command
```

- 特権のないユーザでも実行できるコマンド。

```
> command
```

- 通知



警告: 警告の通知

続行する前に知っておくべき、無視できない情報。セキュリティ上の問題、データ損失の可能性、ハードウェアの損傷、または物理的な危険について警告します。



重要: 重要な通知

続行する前に知っておくべき重要な情報です。



注記: メモの通知

追加情報。たとえば、ソフトウェアバージョンの違いに関する情報です。



ヒント: ヒントの通知

ガイドラインや実地的なアドバイスなどの役に立つ情報です。

- コンパクトな通知





追加情報。たとえば、ソフトウェアバージョンの違いに関する情報です。



ガイドラインや実地的なアドバイスなどの役に立つ情報です。

4 サポート

SUSE Linux Enterprise Desktopのサポートステートメントと、技術レビューに関する概要を以下に示します。製品ライフサイクルの詳細については、<https://www.suse.com/lifecycle>  を参照してください。

サポート資格をお持ちの場合、<https://documentation.suse.com/sles-15/html/SLES-all/cha-adm-support.html>  を参照して、サポートチケットの情報を収集する方法の詳細を確認してください。

4.1 SUSE Linux Enterprise Desktopのサポートステートメント

サポートを受けるには、SUSEの適切な購読が必要です。利用可能なサポートサービスを具体的に確認するには、<https://www.suse.com/support/>  にアクセスして製品を選択してください。

サポートレベルは次のように定義されます。

L1

問題の判別。互換性情報、使用サポート、継続的な保守、情報収集、および利用可能なドキュメントを使用した基本的なトラブルシューティングを提供するように設計されたテクニカルサポートを意味します。

L2

問題の切り分け。データの分析、お客様の問題の再現、問題領域の特定、レベル1で解決できない問題の解決、またはレベル3の準備を行うように設計されたテクニカルサポートを意味します。

L3

問題解決。レベル2サポートで特定された製品の欠陥を解決するようにエンジニアリングに依頼して問題を解決するように設計されたテクニカルサポートを意味します。

契約されているお客様およびパートナーの場合、SUSE Linux Enterprise Desktopでは、次のものを除くすべてのパッケージに対してL3サポートを提供します。

- 技術レビュー。
- サウンド、グラフィック、フォント、およびアートワーク。
- 追加の顧客契約が必要なパッケージ。
- メインのパッケージとともにのみサポートが提供される、名前が `-devel` で終わるパッケージ(ヘッダファイルや同様の開発者用のリソースを含む)。


SUSEは、元のパッケージの使用のみをサポートします。つまり、変更も、再コンパイルもされないパッケージをサポートします。

4.2 技術レビュー

技術レビューとは、今後のイノベーションを垣間見ていただくための、SUSEによって提供されるパッケージ、スタック、または機能を意味します。技術レビューは、ご利用中の環境で新しい技術をテストする機会を参考までに提供する目的で収録されています。私たちはフィードバックを歓迎しています。技術レビューをテストする場合は、SUSEの担当者に連絡して、経験や使用例をお知らせください。ご入力いただいた内容は今後の開発のために役立たせていただきます。

技術レビューには、次の制限があります。

- 技術レビューはまだ開発中です。したがって、機能が不完全であったり、不安定であったり、運用環境での使用には適していなかったりする場合があります。
- 技術レビューにはサポートが「提供されません」。
- 技術レビューは、特定のハードウェアアーキテクチャでしか利用できないことがあります。
- 技術レビューの詳細および機能は、変更される場合があります。そのため、今後リリースされる技術レビューへのアップグレードができない場合や、再インストールが必要となる場合があります。
- SUSEで、レビューがお客様や市場のニーズを満たしていない、またはエンタープライズ標準に準拠していないことを発見する場合があります。技術レビューは製品から予告なく削除される可能性があります。SUSEでは、このようなテクノロジーのサポートされるバージョンを将来的に提供できない場合があります。

ご使用の製品に付属している技術レビューの概要については、<https://www.suse.com/releasesnotes> にあるリリースノートを参照してください。

| 共通のタスク

- 1 BashとBashスクリプト 2
- 2 **sudo**の基本 18
- 3 YaSTの使用 27
- 4 テキストモードのYaST 29
- 5 YaSTによる言語および国の設定の変更 59
- 6 YaSTによるユーザの管理 66
- 7 YaSTオンラインアップデート 81
- 8 ソフトウェアをインストールまたは削除する 88
- 9 コマンドラインツールによるソフトウェアの管理 107
- 10 Snapperを使用したシステムの回復とスナップショット管理 138
- 11 KLPによるライブカーネルパッチ 182
- 12 ユーザ空間ライブパッチ 189
- 13 トランザクション更新 193
- 14 VNCによるリモートグラフィカルセッション 200
- 15 rsyncによるファイルのコピー 217

1 BashとBashスクリプト

今日、多数のユーザが、GNOMEなどのGUI(グラフィカルユーザインターフェース)を介してコンピュータを使用しています。GUIは多くの機能を提供していますが、自動タスクを実行する際には用途は限定されています。シェルはGUIを適切に補完するものです。この章では、シェル(ここではBashシェル)のいくつかの側面について概要を説明します。

1.1 「シェル」とは何か?

従来、Linux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。シェルはBash以外にもあり(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。

1.1.1 Bashの環境設定ファイル

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などのコマンドラインインタフェース(CLI)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呼び出されます。

シェルによって読み込まれる設定ファイルは異なります。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。



ヒント

Bashは、実行されているシェルのタイプに応じて特定の順序で設定ファイルを検索します。詳細については、Bashのマニュアルページ([`man 1 bash`](#))を参照してください。見出し`INVOCATION`を検索します。

表 1.1: ログインシェル用BASH設定ファイル

ファイル	説明
<u>/etc/profile</u>	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。
<u>/etc/profile.local</u>	<u>/etc/profile</u> を拡張する場合は、このファイルを使用します。
<u>/etc/profile.d/</u>	特定プログラムのシステム全体にわたる設定ファイルを含みます。
<u>~/.profile</u>	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。

ログインシェルは、表1.2「非ログインシェル用Bash設定ファイル」に示す設定ファイルも参照します。

表 1.2: 非ログインシェル用BASH設定ファイル

<u>/etc/bash.bashrc</u>	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。
<u>/etc/bash.bashrc.local</u>	Bashのシステム全体にわたる変更を挿入する場合のみ、このファイルを使用します。
<u>~/.bashrc</u>	ユーザ固有の設定をここに挿入します。

また、Bashは複数のファイルを使用します。

表 1.3: BASH用特殊ファイル

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

ファイル	説明
<u>~/.alias</u>	よく使用されるコマンドのユーザ定義エイリアス。エイリアスの定義の詳細については、 <u>man 1 alias</u> を参照してください。

非ログインシェル

ユーザがシステムにログインするのをブロックする特殊なシェル([/bin/false](#)と[/sbin/nologin](#))があります。ユーザがシステムにログインしようとする、両方ともサイレントに失敗します。これはシステムユーザのセキュリティ対策として意図されたものですが、最新のLinuxオペレーティングシステムには、PAMやAppArmorなど、システムアクセスを制御するためのより効果的なツールがあります。

SUSE Linux Enterprise Desktopのデフォルトでは、人間のユーザに[/bin/bash](#)を、システムユーザに[/bin/false](#)または[/sbin/nologin](#)を割り当てます。歴史的な理由のため、[nobody](#)ユーザは[/bin/bash](#)を使用します。これは、システムユーザのデフォルトとして使用されていた最小限の特権を持つユーザであるためです。ただし、[nobody](#)を使用することにより得られたわずかなセキュリティも、複数のシステムユーザが使用すると失われます。これを[/sbin/nologin](#)に変更できるはずです。それをテストする最も早い方法は、変更を行い、サービスやアプリケーションを中断させていないかどうかを確認することです。

次のコマンドを使用して、[/etc/passwd](#)で、すべてのユーザ、システム、および人間のユーザに割り当てられているシェルを一覧にします。出力は、システムのサービスおよびユーザによって異なります。

```
> sort -t: -k 7 /etc/passwd | awk -F: '{print $1"\t" $7}' | column -t
tux                /bin/bash
nobody             /bin/bash
root               /bin/bash
avahi              /bin/false
chrony             /bin/false
dhcpd              /bin/false
dnsmasq            /bin/false
ftppsecure         /bin/false
lightdm            /bin/false
mysql              /bin/false
postfix            /bin/false
rtkit              /bin/false
sshd               /bin/false
tftp               /bin/false
unbound            /bin/false
bin                /sbin/nologin
```



```

daemon      /sbin/nologin
ftp          /sbin/nologin
lp           /sbin/nologin
mail         /sbin/nologin
man          /sbin/nologin
nscd         /sbin/nologin
polkitd      /sbin/nologin
pulse        /sbin/nologin
qemu         /sbin/nologin
radvd        /sbin/nologin
rpc          /sbin/nologin
statd        /sbin/nologin
svn          /sbin/nologin
systemd-coredump /sbin/nologin
systemd-network /sbin/nologin
systemd-timesync /sbin/nologin
usbmux       /sbin/nologin
vnc          /sbin/nologin
wwwrun       /sbin/nologin
messagebus   /usr/bin/false
scard        /usr/sbin/nologin

```

1.1.2 ディレクトリ構造

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。

表 1.4: 標準的なディレクトリツリーの概要

ディレクトリ	目次
<u>/</u>	ルートディレクトリ(ディレクトリツリーの開始場所)。
<u>/bin</u>	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などのシェルも含まれます。
<u>/boot</u>	ブートローダの静的ファイル
<u>/dev</u>	ホスト固有のデバイスのアクセスに必要なファイル
<u>/etc</u>	ホスト固有のシステム設定ファイル

ディレクトリ	目次
<u>/home</u>	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、 <u>root</u> のホームディレクトリは、 <u>/home</u> でなく、 <u>/root</u> にあります。
<u>/lib</u>	必須の共有ライブラリおよびカーネルモジュール
<u>/media</u>	リムーバブルメディアのマウントポイント
<u>/mnt</u>	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
<u>/opt</u>	アドオンアプリケーションのソフトウェアパッケージ
<u>/root</u>	スーパーユーザ <u>root</u> のホームディレクトリ。
<u>/sbin</u>	必須のシステムバイナリ
<u>/srv</u>	システムで提供するサービスのデータ
<u>/tmp</u>	一時ファイルを格納するディレクトリ
<u>/usr</u>	読み込み専用データを含む第二階層
<u>/var</u>	ログファイルなどの可変データ
<u>/windows</u>	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。これらのコマンドには、ls、mkdir、cp、mv、rmおよびrmdirが含まれています。また、/binには、SUSE Linux Enterprise DesktopのデフォルトシェルであるBashも含まれます。

/boot

ブートに必要なデータ(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、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。/var/runは、/runへのシンボリックリンクです。

/sbin

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

/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。データが残っているかどうかは、たとえば/etc/tmpfiles.d/tmp.confの設定によって異なります。

/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などいくつか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42.1「ログファイル」](#) を参照してください。

/windows

システムにMicrosoft WindowsとLinuxの両方がインストールされている場合のみ使用可能。システムのWindowsパーティションで使用可能なWindowsデータを含みます。このディレクトリのデータを編集できるかどうかは、Windowsパーティションが使用するファイルシステムによって異なります。FAT32の場合、このディレクトリのファイルを開いて編集できます。SUSE Linux Enterprise Desktopは、NTFSに対して書き込みアクセスもサポートしています。ただし、NTFS-3gファイルシステムのドライバでは機能が制限されています。

1.2 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、さまざまな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 1.1: 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 最初の行は、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。「Shebang」の後に指定されたインタプリタによってスクリプトが実行されます。この場合、指定されたインタプリタは/bin/shです。
- ❷ 2行目は、ハッシュ記号で始まるコメントです。意図を理解しにくい行にはコメントすることをお勧めします。適切にコメントすると、行の目的および機能を覚えることができます。また、他の読み手もスクリプトをより良く理解できます。コメントは開発コミュニティにおいてグッドプラクティスとみなされます。
- 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。

このスクリプトを実行するには、その前にいくつかの前提条件があります。

1. すべてのスクリプトには(上記の例のように) Shebang行が含まれている必要があります。この行がない場合は、インタプリタを手動で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。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。

1.3 コマンドイベントのリダイレクト

各コマンドは、入力または出力用として、3つのチャンネルを使用できます。:

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

これらのチャンネルをリダイレクトするには、次の方法を使用できます。

Command > File

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

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

Command >> File

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

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

Command < File

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

```
> 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
```

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

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

1.4 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、次のとおりです。

```
alias NAME=DEFINITION
```

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

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

すべてのエイリアス定義を表示するには、`alias`を使用します。`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。

1.5 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
```

次の表には、シェルスクリプトで使用できる一般的な環境変数が含まれています。

表 1.5: 便利な環境変数

<u>HOME</u>	現在のユーザのホームディレクトリ
<u>HOST</u>	現在のホスト名
<u>LANG</u>	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。
<u>PATH</u>	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
<u>PS1</u>	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。
<u>PS2</u>	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。
<u>PWD</u>	現在の作業ディレクトリ
<u>USER</u>	現在のユーザ

1.5.1 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" "" ""
```

1.5.2 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}
```

1.6 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。

次に、コマンドをグループ化する方法を示します。

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
```

1.7 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。

1.7.1 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、<https://bash.cyberciti.biz/guide/If..else..fi> を参照してください。





1.7.2 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ファイルの特定の情報をプリントします。

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

1.8 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

2 sudoの基本

特定のコマンドを実行するには、root特権が必要です。ただし、セキュリティ上の理由のため、また間違いを避けるため、rootとしてログインすることは推奨されません。より安全な方法は、通常のユーザとしてログインしてから、**sudo**を使用して昇格された特権でコマンドを実行することです。

SUSE Linux Enterprise Desktopでは、**sudo**は**su**と同様に機能するように設定されています。ただし、**sudo**には、ユーザが他のユーザの特権でコマンドを実行できるようにする柔軟なメカニズムがあります。このコマンドを使用して、指定の特権を持つ役割を特定のユーザとグループに割り当てることができます。たとえば、**users**グループのメンバーが、**wilber**ユーザの特権でコマンドを実行できるようにすることができます。コマンドへのアクセスは、コマンドオプションを禁止することにより、さらに制限できます。suでは、PAMを使用した認証で常にrootパスワードを必要としますが、**sudo**では、ユーザの資格情報を使用して認証するように設定できます。すなわち、ユーザはrootパスワードを共有する必要がなく、セキュリティが向上します。

2.1 sudoの基本的な使用方法

次の章では、**sudo**の基本的な使用方法の概要について説明します。

2.1.1 単一コマンドの実行

標準ユーザは、コマンドの前に**sudo**を追加することで、任意のコマンドをrootとして実行できます。これにより、rootパスワードを入力するように求められます。正常に認証されたら、**root**としてコマンドが実行されます。

```
> id -un ❶
tux
> sudo id -un
root's password: ❷
root
> id -un
tux ❸
> sudo id -un
❹
root
```

❶ **id -un**コマンドは、現在のユーザのログイン名を出力します。

- ② 入力時には、パスワードは表示されません(クリアテキストとしてだけでなく、マスク文字としても表示されません)。
- ③ **sudo**で始まるコマンドのみが、昇格された特権で実行されます。
- ④ 昇格された特権は特定の期間保持されるため、再びrootパスワードを入力する必要はありません。



ヒント: I/Oリダイレクト

sudoの使用時に、I/Oリダイレクトは機能しません。

```
> sudo echo s > /proc/sysrq-trigger
bash: /proc/sysrq-trigger: Permission denied
> sudo cat < /proc/l/maps
bash: /proc/l/maps: Permission denied
```

上記の例では、**echo**および**cat**コマンドのみが昇格された特権で実行されます。リダイレクトはユーザの特権を使用してユーザのシェルで実行されます。昇格された権限でリダイレクトを実行するには、[2.1.2項「シェルの起動」](#)に記載されているようにシェルを起動するか、**dd**ユーティリティを使用します。

```
echo s | sudo dd of=/proc/sysrq-trigger
sudo dd if=/proc/l/maps | cat
```

2.1.2 シェルの起動

昇格された特権でコマンドを実行するたびに**sudo**を使用することは、必ずしも実用的ではありません。**sudo bash**コマンドを使用できますが、組み込みメカニズムのいずれかを使用してシェルを起動することをお勧めします。

sudo -s (<command>)

SHELL環境変数で指定したシェル、またはターゲットユーザのデフォルトのシェルを起動します。コマンドが指定される場合は、シェルに渡されます(**-c**オプションを使用)。そうでない場合、シェルは対話的モードで実行されます。

```
tux:~ > sudo -s
root's password:
root:/home/tux # exit
tux:~ >
```

`sudo -i (<command>)`

`-s`と同様ですが、シェルはログインシェルとして起動します。つまり、シェルの起動ファイル(`.profile`など)は処理され、現在の作業ディレクトリはターゲットユーザのホームディレクトリに設定されます。

```
tux:~ > sudo -i
root's password:
root:~ # exit
tux:~ >
```



ヒント: 環境変数

デフォルトでは、`sudo`は環境変数を伝達しません。この動作は、`env_reset`オプションを使用して変更できます(有用なフラグとオプションを参照してください)。

2.2 `sudo`の構成

`sudo`は、設定可能なオプションの幅広い範囲を提供します。



注記: `sudo`からのロックアウト

誤って`sudo`からロックアウトした場合は、`su -`と`root`パスワードを使用してルートシェルを起動してください。エラーを修正するには、`visudo`を実行します。

2.2.1 設定ファイルの編集

`sudo`向けの主なポリシー設定ファイルは、`/etc/sudoers`です。ファイルの形式が正しくない場合、システムからロックアウトされてしまう可能性があるため、編集に`visudo`を使用することを強くお勧めします。このコマンドは、編集の競合を防ぎ、変更を保存する前に構文エラーをチェックします。

`EDITOR`環境変数を次のように設定して、`vi`の代わりに別のエディタを使用することもできます。

```
sudo EDITOR=/usr/bin/nano visudo
```


`/etc/sudoers` ファイルはシステムパッケージによって提供されているため、ファイル内で直接行われた変更によって更新が破損する可能性があります。そのため、カスタム設定は、`/etc/sudoers.d/` ディレクトリ内のファイルに対して行うことをお勧めします。ファイルを作成または編集するには、次のコマンドを使用します。

```
sudo visudo -f /etc/sudoers.d/NAME
```

次のコマンドは、別のエディタ(この場合は、**nano**)を使用してファイルを開きます。

```
sudo EDITOR=/usr/bin/nano visudo -f /etc/sudoers.d/NAME
```



注記: `/etc/sudoers.d` 内の無視されるファイル

`/etc/sudoers` の `#includedir` ディレクティブは、`~` (チルダ) で終わるファイル、または `.` (ドット) を含むファイルは無視します。

visudo コマンドの詳細については、`man 8 visudo` を実行してください。

2.2.2 基本的な `sudoers` の設定構文

`sudoers` の設定ファイルには、2種類のオプション(文字列とフラグ)があります。文字列には任意の値を含めることができますが、フラグはONかOFFのいずれかのみです。`sudoers` の設定ファイルの最も重要な構文構造は次のとおりです。

```
# Everything on a line after # is ignored ❶
Defaults !insults # Disable the insults flag ❷
Defaults env_keep += "DISPLAY HOME" # Add DISPLAY and HOME to env_keep
tux ALL = NOPASSWD: /usr/bin/frobnicate, PASSWD: /usr/bin/journalctl ❸
```

- ❶ 例外が2つあります。`#include` と `#includedir` は通常のコマンドです。
- ❷ `!` 文字を削除して、目的のフラグをONに設定します。
- ❸ 2.2.3項「基本的な `sudoers` のルール」を参照してください。

有用なフラグとオプション

`targetpw`

このフラグは、呼び出し元のユーザが、ターゲットユーザのパスワード(ON)(`root` など)と、呼び出し元のユーザのパスワード(OFF)のいずれを要求されるかを決定します。

```
Defaults targetpw # Turn targetpw flag ON
```

rootpw

設定すると、**sudo**は、rootパスワードを要求します。デフォルトは [オフ] です。

```
Defaults !rootpw # Turn rootpw flag OFF
```

env_reset

設定されている場

合、**sudo**は、TERM、PATH、HOME、MAIL、SHELL、LOGNAME、USER、USERNAME、およびSUDO_*を使用して最低限の環境を構築します。また、env_keepに列挙されている変数は、呼び出し元の環境からインポートされます。デフォルトは [ON] です。

```
Defaults env_reset # Turn env_reset flag ON
```

env_keep

env_resetフラグがONの場合に保持する環境変数の一覧。

```
# Set env_keep to contain EDITOR and PROMPT
Defaults env_keep = "EDITOR PROMPT"
Defaults env_keep += "JRE_HOME" # Add JRE_HOME
Defaults env_keep -= "JRE_HOME" # Remove JRE_HOME
```

env_delete

env_resetフラグがOFFの場合に削除する環境変数の一覧。

```
# Set env_delete to contain EDITOR and PROMPT
Defaults env_delete = "EDITOR PROMPT"
Defaults env_delete += "JRE_HOME" # Add JRE_HOME
Defaults env_delete -= "JRE_HOME" # Remove JRE_HOME
```

Defaultsトークンを使用することで、ユーザ、ホスト、およびコマンドのコレクションのエイリアスを作成することもできます。さらに、一連のユーザのみを対象としてオプションを適用することができます。

/etc/sudoers設定ファイルの詳細については、man 5 sudoersを参照してください。

2.2.3 基本的なsudoersのルール

各ルールは、次のスキームに従います([]はオプション部分を示しています)。

```
#Who      Where      As whom    Tag          What
User_List Host_List = [(User_List)] [NOPASSWD:|PASSWD:] Cmdnd_List
```

SUDOERSルール構文

User_List

1つ以上の(コンマで区切られた)識別子。ユーザ名、%GROUPNAME形式のグループ、または#UID形式のユーザIDを指定します。否定は!プレフィックスで指定できます。

Host_List

1つ以上の(コンマで区切られた)識別子。(完全修飾された)ホスト名またはIPアドレスのいずれかを指定します。否定は!プレフィックスで指定できます。通常、Host_ListにはALLを選択します。

NOPASSWD: | PASSWD:

NOPASSWD:の後に、Cmd_Listと一致するコマンドを実行すると、パスワードは要求されません。

PASSWDはデフォルトです。PASSWDとNOPASSWDの両方が同じ行に存在する場合にのみ指定する必要があります。

```
tux ALL = PASSWD: /usr/bin/foo, NOPASSWD: /usr/bin/bar
```

Cmd_List

1つまたは複数の(コンマで区切られた)指定子。実行可能ファイルへのパスの後に、オプションで使用可能な引数を指定します。

```
/usr/bin/foo      # Anything allowed
/usr/bin/foo bar  # Only "/usr/bin/foo bar" allowed
/usr/bin/foo ""   # No arguments allowed
```

ALLは、User_List、Host_List、およびCmd_Listとして使用できます。

パスワードを入力しなくても、tuxがすべてのコマンドをrootとして実行できるようにするルールは次のとおりです。

```
tux ALL = NOPASSWD: ALL
```

tuxがsystemctl restart apache2を実行できるようにするルールは次のとおりです。

```
tux ALL = /usr/bin/systemctl restart apache2
```

tuxがwallをadminとして引数なしで実行できるようにするルールは次のとおりです。

```
tux ALL = (admin) /usr/bin/wall ""
```



警告: 危険なルール

`Defaults targetpw`なしで、`ALL ALL = ALL`のようなルールを使用「「しないでください」」。そうしないと、だれでも`root`としてコマンドを実行できるようになります。



重要: Winbindとsudo

`sudoers` ファイルでグループ名を指定する場合は、レルムの代わりにNetBIOSドメイン名を使用するようにしてください。次に例を示します。

```
%DOMAIN\\GROUP_NAME ALL = (ALL) ALL
```

`winbindd`を使用する場合、この形式は、`smb.conf`ファイルの`winbind separator`オプションによっても異なることに注意してください。デフォルトでは、`\`です。たとえば、`+`に変更された場合、`sudoers`ファイルのアカウント形式は`DOMAIN+GROUP_NAME`である必要があります。

2.3 sudoの使用例

デフォルト設定は標準的な使用シナリオで機能しますが、特定のニーズに合わせてデフォルト設定をカスタマイズできます。

2.3.1 rootパスワードなしのsudoの使用

設計では、`wheel`グループのメンバーは、すべてのコマンドを`root`として`sudo`で実行できます。次の手順では、`wheel`グループにユーザアカウントを追加する方法について説明します。

1. `wheel`グループが存在することを確認します。

```
> getent group wheel
```

前のコマンドで結果が返されない場合、`wheel`グループを作成する`system-group-wheel`パッケージをインストールします。

```
> sudo zypper install system-group-wheel
```

2. ユーザアカウントを`wheel`グループに追加します。

ユーザアカウントがまだwheelグループのメンバーではない場合、`sudo usermod -a -G wheel USERNAME`コマンドを使用して追加します。ログアウトしてからもう一度ログインして、変更を有効にします。`groups USERNAME`コマンドを実行して、変更が成功したことを確認します。

3. ユーザアカウントの通常のパスワードで認証します。

`visudo`コマンド(2.2.1項「設定ファイルの編集」を参照)を使用して、`/etc/sudoers.d/userpw`ファイルを作成し、次を追加します。

```
Defaults !targetpw
```

4. 新しいデフォルトのルールを選択します。

ユーザにパスワードの再入力を求めるかどうかによって、`/etc/sudoers`の適切な行のコメントを解除し、デフォルトのルールをコメントアウトします。

```
## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL
```

5. デフォルトのルールにより厳しい制約を設けます。

`/etc/sudoers`の、すべてを許可するルールをコメントアウトするか、削除します。

```
ALL ALL=(ALL) ALL # WARNING! Only use this together with 'Defaults targetpw'!
```



警告: sudoersの危険なルール

このステップはスキップしないでください。そうしないと、「すべての」ユーザが「すべての」コマンドをrootとして実行できるようになってしまいます。

6. 設定をテストします。

`sudo`を、wheelのメンバーとして、またはメンバー以外として実行します。

```
tux:~ > groups
users wheel
tux:~ > sudo id -un
tux's password:
root
wilber:~ > groups
users
wilber:~ > sudo id -un
wilber is not in the sudoers file. This incident will be reported.
```

2.3.2 X.Orgアプリケーションでのsudoの使用

グラフィカルアプリケーションをsudoで起動すると、通常、次のエラーが発生します。

```
> sudo xterm
xterm: Xt error: Can't open display: %s
xterm: DISPLAY is not set
```

簡単な回避策は、xhostを使用して、ルートユーザがローカルユーザのXセッションに一時的にアクセスできるようにすることです。これは、次のコマンドを使用して実行されます。

```
xhost si:localuser:root
```

次のコマンドは許可されたアクセスを削除します。

```
xhost -si:localuser:root
```



警告: 潜在的なセキュリティの問題

ルート特権でグラフィカルアプリケーションを実行すると、セキュリティに影響を与えます。例外としてのみグラフィカルアプリケーションのルートアクセスを有効にすることをお勧めします。グラフィカルアプリケーションが閉じられたらすぐに、許可されたルートアクセスを取り消すことも推奨されます。

2.4 詳細情報

sudo --help コマンドは、使用可能なコマンドラインオプションの簡単な概要を提供し、man sudoers コマンドは、sudoers とその設定に関する詳細情報を提供します。

3 YaSTの使用

YaSTは、すべての必須インストールおよびシステム設定タスクにグラフィカルインタフェースを提供するSUSE Linux Enterprise Desktopツールです。パッケージの更新、プリンタの設定、ファイアウォール設定の変更、FTPサーバの設定、ハードディスクのパーティション作成が必要な場合、YaSTを使用して実行できます。Rubyで記述されたYaSTは、モジュールを介して新しい機能を追加できる拡張可能なアーキテクチャを備えています。

YaSTに関する追加情報については、<https://yast.opensuse.org/>にあるプロジェクトの公式Webサイトを参照してください。

3.1 YaSTインタフェースの概要

YaSTには2つのグラフィカルインタフェースがあります。1つはKDEやGNOMEなどのグラフィカルデスクトップ環境で使用するためのもので、もう1つはXサーバを使用しないシステムで使用するためのncursesベースの疑似グラフィカルインタフェースです(第4章「テキストモードのYaST」を参照してください)。

YaSTのグラフィカルバージョンでは、YaSTのすべてのモジュールはカテゴリ別にグループ化されており、ナビゲーションサイドバーによって、目的のカテゴリのモジュールに素早くアクセスできます。上部の検索フィールドでは、名前でモジュールを検索することができます。特定のモジュールを検索するには、検索フィールドにその名前を入力します。入力すると、入力した文字列に一致するモジュールが表示されます。

3.2 便利なキーの組み合わせ

YaSTのグラフィカルバージョンはキーボードショートカットをサポートしています

Print Screen

スクリーンショットを作成して保存します。特定のデスクトップ環境では機能しない場合があります。

Shift + F4

視覚障がいのあるユーザ向けに最適化されたカラーパレットを有効および無効にします。

Shift + F7

デバッグメッセージのログを有効/無効にします。

Shift – F8

ファイルダイアログを開いて、ログファイルをユーザ定義の場所に保存します。

Ctrl – Shift – Alt – D

DebugEventを送信します。YaSTモジュールは特殊なデバッグアクションを実行してこれに応答できます。結果は特定のYaSTモジュールによって異なります。

Ctrl – Shift – Alt – M

マクロレコーダを開始および停止します。

Ctrl – Shift – Alt – P

マクロを再生します。

Ctrl – Shift – Alt – S

スタイルシートエディタを表示します。

Ctrl – Shift – Alt – T

ウィジェットツリーをログファイルにダンプします。

Ctrl – Shift – Alt – X

ターミナルウィンドウ(xterm)を開きます。VNCを使用したインストールプロセスで便利です。

Ctrl – Shift – Alt – Y

ウィジェットツリーブラウザを表示します。

4 テキストモードのYaST

ncursesベースの疑似グラフィカルYaSTインタフェースは、主にシステム管理者がXサーバを使用せずにシステムを管理できるように設計されています。このインタフェースには、従来のGUIと比較していくつかの利点があります。キーボードを使用してncursesインタフェースをナビゲートすることができ、事実上すべてのインタフェース要素に対するキーボードショートカットがあります。ncursesインタフェースは、リソースが少なく、中程度のハードウェアでも高速に実行されます。SSH接続を介してncursesベースのバージョンのYaSTを実行することができるため、リモートシステムを管理できます。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文字であることに注意してください。

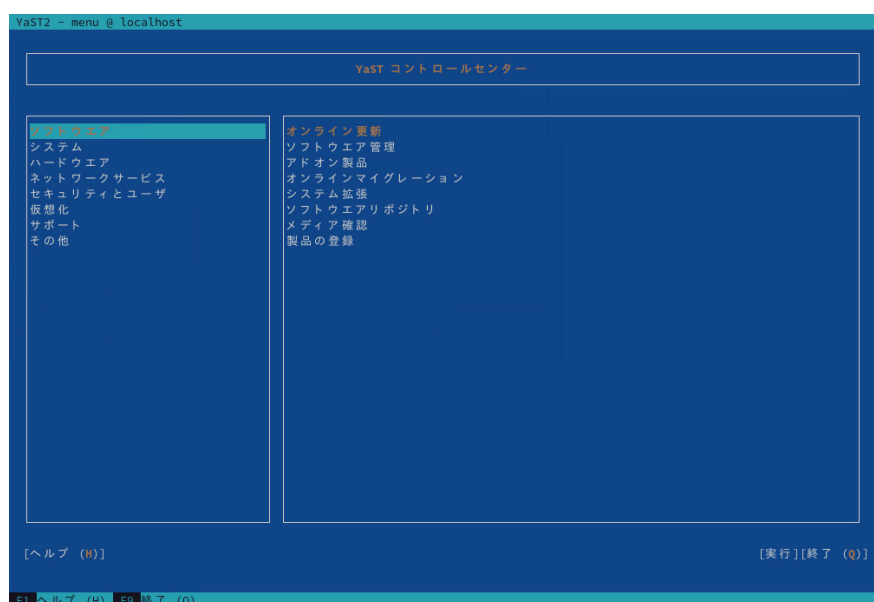


図 4.1: テキストモードのYASTのメインウィンドウ

ncursesベースのバージョンのYaSTを起動するには、ターミナルを開いて、**sudo yast2** コマンドを実行します。<Tab> または矢印キーを使用して、メニュー項目、フィールド、ボタンなどのインタフェース要素間をナビゲートします。YaSTのすべてのメニュー項目およびボタンには、適切なファンクションキーまたはキーボードショートカットを使用してアクセスできます。たとえば、**F9** を押して現在の操作をキャンセルし、**F10** キーを使用して変更を受け入れることができます。YaSTのncursesベースのインタフェースの各メニュー項目およびボタンには、そのラベルにハイライトされた文字があります。この文字は、インタフェース要素に割り当てられたキーボードショートカットの一部です。たとえば、文字**Q**は終了ボタンでハイライトされます。これは、**Alt + Alt+Q** を押してボタンを有効にできることを意味します。



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。

4.1 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、4.3項「[キーの組み合わせの制約](#)」を参照してください。

ボタンと選択リスト間の移動

選択リストを含むボタンおよびフレーム間を移動するには、**<Tab>** を使用します。反対方向に移動するには、**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ト

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移動するには、矢印キー(**↑** と **↓**)を使用します。個々のエントリがフレームの幅よりも長い場合は、**Shift-→** キーまたは **Shift-←** キーを使用して水平にスクロールします。矢印キーで選択範囲が別のフレームに移動する場合は、代わりに **Ctrl-E** または **Ctrl-A** を使用してください。

ボタン、ラジオボタン、チェックボックスの操作

空の角括弧(チェックボックス)または空の丸括弧(ラジオボタン)が付いている項目を選択するには、**Space** または **Enter** 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。

特殊キー

ファンクションキー(**F1** から **F12**)を使用すると、特定のボタンの機能を素早く利用できます。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。

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

左側のフィルタを使用して、指定された文字列に一致するパッケージを一覧にします。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。

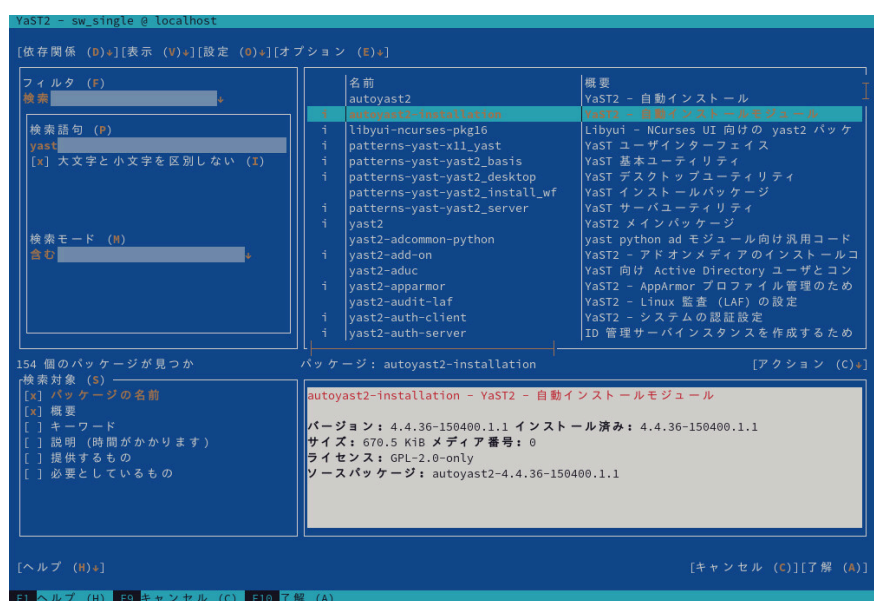


図 4.2: ソフトウェアインストールモジュール

4.2 高度なキーの組み合わせ

ncursesベースのバージョンのYaSTは、高度なキーの組み合わせを提供します。

Shift + F1

高度なホットキーを一覧表示します。

Shift - F4

配色を変更します。

Ctrl - Q

アプリケーションを終了します。

Ctrl - L

画面を更新します。

Ctrl - D F1

高度なホットキーを一覧表示します。

Ctrl - D Shift - D

ダイアログをスクリーンショットとしてログファイルにダンプします。

Ctrl - D Shift - Y

YDialogSpyを開いてウィジェット階層を表示します。

4.3 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。

Esc の代わりに **Alt** を使用

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)の組み合わせを代わりに使用できます。

ファンクションキーの制約

ファンクションキー(**F1** ... **F12**)は各種機能にも使用されます。一部のファンクションキーは、端末に専有され、YaSTで使え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、プリアテキストコンソールでは常に完全に使用できます。

4.4 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コマンドオプションのリストを取得するには、次のコマンドを使用します。

```
> sudo yast -h
```

4.4.1 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。

```
> sudo yast -i package_name
```

あるいは、

```
> sudo yast --install -i 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ではコマンドラインからソフトウェアを管理するための基本的な機能を提供しますが、高度なパッケージ管理タスクにはZypperを使用することを検討してください。Zypperの使用に関する詳細については、[9.1項「Zypperの使用」](#)を参照してください。

4.4.2 個々のモジュールの使用

時間を節約するために、次のコマンドを使用して個々のYaSTモジュールを起動することができます。

```
> sudo yast module_name
```

yast -lまたは**yast --list**を使用してシステムで使用可能なすべてのモジュールのリストを表示します。

4.4.3 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。モジュールの使用可能なオプションを表示するには、次のコマンドを使用します。

```
> sudo 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.
```

以下のセクションでは、コマンドラインサポートがあるすべてのYaSTモジュールと、それらのすべてのコマンドおよび利用可能なオプションについて簡単に説明します。

4.4.3.1 一般的なYaSTモジュールコマンド

すべてのYaSTモジュールは、以下のコマンドをサポートしています。

help

モジュールのサポートされているすべてのコマンドを、その説明と共に一覧表示します。

```
> sudo yast lan help
```

longhelp

helpと同じですが、すべてのコマンドのオプションの詳細なリストとその説明が追加されています。

```
> sudo yast lan longhelp
```

xmlhelp

longhelpと同じですが、出力はXML文書として構成され、ファイルにリダイレクトされます。

```
> sudo yast lan xmlhelp xmlfile=/tmp/yast_lan.xml
```

interactive

「対話的」モードにします。これにより、**sudo yast**で事前に修正せずにモジュールのコマンドを実行できます。対話的モードを終了するには、exitを使用します。

4.4.3.2 yast add-on

指定されたパスから新しいアドオン製品を追加します。

```
> sudo yast add-on http://server.name/directory/Lang-AddOn-CD1/
```

次のプロトコルを使用して、ソースパスを指定できます:。 http:// ftp:// nfs:// disk:// cd:// または dvd://。

4.4.3.3 yast audit-laf

Linux監査フレームワークを表示および設定します。詳細については、『Security and Hardening Guide』を参照してください。yast audit-lafは次のコマンドを受け付けます。

set

オプションを設定します。

```
> sudo yast audit-laf set log_file=/tmp/audit.log
```

すべてのオプションのリストについては、yast audit-laf set helpを実行してください。

show

オプションの設定を表示します。

```
> sudo yast audit-laf show diskpace
space_left: 75
space_left_action: SYSLOG
admin_space_left: 50
admin_space_left_action: SUSPEND
action_mail_acct: root
disk_full_action: SUSPEND
disk_error_action: SUSPEND
```

すべてのオプションのリストについては、yast audit-laf show helpを実行してください。

4.4.3.4 yast dhcp-server

DHCPサーバを管理し、その設定を行います。yast dhcp-serverは次のコマンドを受け付けます。

無効

DHCPサーバサービスを無効にします。

enable

DHCPサーバサービスを有効にします。

ホスト

個々のホストの設定を行います。

interface

リスンするネットワークインタフェースを指定します。

```
> sudo yast dhcp-server interface current
Selected Interfaces: eth0
Other Interfaces: bond0, pbu, eth1
```

すべてのオプションのリストについては、**yast dhcp-server interface help**を実行してください。

options

グローバルDHCPオプションを管理します。すべてのオプションのリストについては、**yast dhcp-server options help**を実行してください。

status

DHCPサービスのステータスを出力します。

サブネット

DHCPサブネットオプションを管理します。すべてのオプションのリストについては、**yast dhcp-server subnet help**を実行してください。

4.4.3.5 yast dns-server

DNSサーバの設定を管理します。**yast dns-server**は次のコマンドを受け付けます。

acls

アクセス制御リストの設定を表示します。

```
> sudo yast dns-server acls show
ACLs:
-----
Name          Type          Value
-----
any           Predefined
localips      Predefined
localnets    Predefined
none          Predefined
```


dnsrecord

ゾーンリソースレコードを設定します。

```
> sudo yast dnsrecord add zone=example.org query=office.example.org type=NS  
value=ns3
```

すべてのオプションのリストについては、**yast dns-server dnsrecord help**を実行してください。

forwarders

DNSフォワーダを設定します。

```
> sudo yast dns-server forwarders add ip=10.0.0.100  
> sudo yast dns-server forwarders show  
[...]  
Forwarder IP  
-----  
10.0.0.100
```

すべてのオプションのリストについては、**yast dns-server forwarders help**を実行してください。

ホスト

「A」とそれに関連する「PTR」レコードを一度に処理します。

```
> sudo yast dns-server host show zone=example.org
```

すべてのオプションのリストについては、**yast dns-server host help**を実行してください。

ログ

ログ設定を行います。

```
> sudo yast dns-server logging set updates=no transfers=yes
```

すべてのオプションのリストについては、**yast dns-server logging help**を実行してください。

mailserver

ゾーンメールサーバを設定します。

```
> sudo yast dns-server mailserver add zone=example.org mx=mx1 priority=100
```

すべてのオプションのリストについては、**yast dns-server mailserver help**を実行してください。

nameserver

ゾーンネームサーバを設定します。

```
> sudo yast dns-server nameserver add zone=example.com ns=ns1
```

すべてのオプションのリストについては、**yast dns-server nameserver help**を実行してください。

soa

権威の開始点(SOA)レコードを設定します。

```
> sudo yast dns-server soa set zone=example.org serial=2006081623 ttl=2D3H20S
```

すべてのオプションのリストについては、**yast dns-server soa help**を実行してください。

起動

DNSサーバサービスを管理します。

```
> sudo yast dns-server startup atboot
```

すべてのオプションのリストについては、**yast dns-server startup help**を実行してください。

transport

ゾーン転送ルールを設定します。すべてのオプションのリストについては、**yast dns-server transport help**を実行してください。

zones

DNSゾーンを管理します。

```
> sudo yast dns-server zones add name=example.org zonetype=master
```

すべてのオプションのリストについては、**yast dns-server zones help**を実行してください。

4.4.3.6 yast disk

すべてのディスクまたはパーティションに関する情報を出力します。唯一サポートされているコマンドは**list**であり、その後に次のいずれかのオプションが続きます。

disks

システム内のすべての設定されているディスクを一覧表示します。

```
> sudo yast disk list disks
Device    | Size      | FS Type | Mount Point | Label | Model
-----+-----+-----+-----+-----+-----
/dev/sda  | 119.24 GiB |         |             |       | SSD 840
/dev/sdb  | 60.84 GiB  |         |             |       | WD1003FBYX-0
```

パーティション

システム内のすべてのパーティションを一覧表示します。

```
> sudo yast disk list partitions
Device      | Size      | FS Type | Mount Point | Label | Model
-----+-----+-----+-----+-----+-----
/dev/sda1   | 1.00 GiB  | Ext2    | /boot       |       |
/dev/sdb1   | 1.00 GiB  | Swap    | swap        |       |
/dev/sdc1   | 698.64 GiB | XFS     | /mnt/extra  |       |
/dev/vg00/home | 580.50 GiB | Ext3    | /home       |       |
/dev/vg00/root | 100.00 GiB | Ext3    | /           |       |
[...]
```

4.4.3.7 yast ftp-server

FTPサーバ設定を行います。**yast ftp-server**は次のオプションを受け付けます。

SSL、TLS

SSLおよびTLSを介して安全な接続を制御します。SSLオプションは、**vsftpd**のみに対して有効です。

```
> sudo yast ftp-server SSL enable
> sudo yast ftp-server TLS disable
```

アクセス

アクセス許可を設定します。

```
> sudo yast ftp-server access authen_only
```

すべてのオプションのリストについては、**yast ftp-server access help**を実行してください。

anon_access

匿名ユーザのアクセス許可を設定します。

```
> sudo yast ftp-server anon_access can_upload
```

すべてのオプションのリストについては、**yast ftp-server anon_access help**を実行してください。

anon_dir

匿名ユーザのディレクトリを指定します。ディレクトリはサーバ上にあらかじめ存在している必要があります。

```
> sudo yast ftp-server anon_dir set_anon_dir=/srv/ftp
```

すべてのオプションのリストについては、**yast ftp-server anon_dir help**を実行してください。

chroot

「change root」(ルート変更)環境(chroot)を制御します。

```
> sudo yast ftp-server chroot enable  
> sudo yast ftp-server chroot disable
```

idle-time

FTPサーバが現在の接続を終了するまでの最大アイドル時間を分単位で設定します。

```
> sudo yast ftp-server idle-time set_idle_time=15
```

ログ

ログメッセージをログファイルに保存するかどうかを判断します。

```
> sudo yast ftp-server logging enable  
> sudo yast ftp-server logging disable
```

max_clients

同時に接続されるクライアントの最大数を指定します。

```
> sudo yast ftp-server max_clients set_max_clients=1500
```

max_clients_ip

IPを介して同時に接続されるクライアントの最大数を指定します。

```
> sudo yast ftp-server max_clients_ip set_max_clients=20
```

max_rate_anon

匿名クライアントに許可される最大データ転送速度(KB/秒)を指定します。

```
> sudo yast ftp-server max_rate_anon set_max_rate=10000
```

max_rate_authen

ローカルに認証されたユーザに許可される最大データ転送速度(KB/秒)を指定します。

```
> sudo yast ftp-server max_rate_authen set_max_rate=10000
```

port_range

パッシブ接続応答のポート範囲を指定します。

```
> sudo yast ftp-server port_range set_min_port=20000 set_max_port=30000
```

すべてのオプションのリストについては、**yast ftp-server port_range help**を実行してください。

show

FTPサーバ設定を表示します。

startup

FTPの起動方法を制御します。

```
> sudo yast ftp-server startup atboot
```

すべてのオプションのリストについては、**yast ftp-server startup help**を実行してください。

umask

authenticated:anonymousユーザのファイルumaskを指定します。

```
> sudo yast ftp-server umask set_umask=177:077
```

welcome_message

FTPサーバに接続された際に表示するテキストを指定します。

```
> sudo yast ftp-server welcome_message set_message="hello everybody"
```

4.4.3.8 yast http-server

HTTPサーバ(Apache2)を設定します。**yast http-server**は次のコマンドを受け付けます。

configure

HTTPサーバのホスト設定を行います。

```
> sudo yast http-server configure host=main servername=www.example.com \  
serveradmin=admin@example.com
```

すべてのオプションのリストについては、**yast http-server configure help**を実行してください。

hosts

仮想ホストを設定します。

```
> sudo yast http-server hosts create servername=www.example.com \  
serveradmin=admin@example.com documentroot=/var/www
```

すべてのオプションのリストについては、**yast http-server hosts help**を実行してください。

listen

HTTPサーバがリスンするポートとネットワークアドレスを指定します。

```
> sudo yast http-server listen add=81  
> sudo yast http-server listen list  
Listen Statements:  
=====  
:80  
:81  
> sudo yast http-server delete=80
```

すべてのオプションのリストについては、**yast http-server listen help**を実行してください。

mode

ウィザードモードを有効または無効にします。

```
> sudo yast http-server mode wizard=on
```

modules

Apache2サーバモジュールを制御します。

```
> sudo yast http-server modules enable=php5,rewrite  
> sudo yast http-server modules disable=ssl  
> sudo http-server modules list  
[...]  
Enabled rewrite  
Disabled ssl  
Enabled php5  
[...]
```

4.4.3.9 yast kdump

kdumpを設定します。**kdump**の詳細については、『System Analysis and Tuning Guide』、第18章「Kexec and Kdump」、18.7項「Basic Kdump configuration」を参照してください。**yast kdump**は次のコマンドを受け付けます。

copykernel

カーネルをダンプディレクトリにコピーします。

customkernel

カスタムカーネルの名前の`kernel_string`部分を指定します。この命名規則は`/boot/vmlinu[zx]-kernel_string[.gz]`です。

```
> sudo yast kdump customkernel kernel=kdump
```

すべてのオプションのリストについては、`yast kdump customkernel help`を実行してください。

dumpformat

ダンプカーネルイメージの(圧縮)形式を指定します。使用可能な形式は、「none」、「ELF」、「compressed」または「lzo」です。

```
> sudo yast kdump dumpformat dump_format=ELF
```

dumplevel

ダンプレベル番号を0～31の範囲で指定します。

```
> sudo yast kdump dumplevel dump_level=24
```

dumptarget

ダンプイメージの保存先を指定します。

```
> sudo kdump dumptarget target=ssh server=name_server port=22 \
dir=/var/log/dump user=user_name
```

すべてのオプションのリストについては、`yast kdump dumptarget help`を実行してください。

immediatereboot

Kdumpカーネルにコアを保存した直後に、システムを再起動するかどうかを制御します。

```
> sudo yast kdump immediatereboot enable
> sudo yast kdump immediatereboot disable
```

keepolddumps

保持する古いダンプイメージの数を指定します。ダンプイメージをすべて保持するには、ゼロを指定します。

```
> sudo yast kdump keepolddumps no=5
```

kernelcommandline

kdumpカーネルに渡す必要があるコマンドラインを指定します。

```
> sudo yast kdump kernelcommandline command="ro root=LABEL=/"
```

kernelcommandlineappend

デフォルトのコマンドライン文字列に「追加」する必要があるコマンドラインを指定します。

```
> sudo yast kdump kernelcommandlineappend command="ro root=LABEL=/"
```

notificationcc

通知メッセージのコピーを送信するための電子メールアドレスを指定します。

```
> sudo yast kdump notificationcc email="user1@example.com user2@example.com"
```

notificationto

通知メッセージを送信するための電子メールアドレスを指定します。

```
> sudo yast kdump notificationto email="user1@example.com user2@example.com"
```

show

kdump設定を表示します。

```
> sudo yast kdump show
Kdump is disabled
Dump Level: 31
Dump Format: compressed
Dump Target Settings
target: file
file directory: /var/crash
Kdump immediate reboots: Enabled
Numbers of old dumps: 5
```

smtppass

通知メッセージの送信に使用されるプレーンテキストのSMTPパスワードを含むファイルを指定します。

```
> sudo yast kdump smtppass pass=/path/to/file
```

smtpserver

通知メッセージの送信に使用されるSMTPサーバのホスト名を指定します。

```
> sudo yast kdump smtpserver server=smtp.server.com
```


smtpuser

通知メッセージの送信に使用されるSMTPユーザ名を指定します。

```
> sudo yast kdump smtpuser user=smtp_user
```

起動

起動オプションを有効または無効にします。

```
> sudo yast kdump startup enable alloc_mem=128,256
> sudo yast kdump startup disable
```

4.4.3.10 yast keyboard

仮想コンソールのシステムキーボードを設定します。GNOMEやKDEなどのグラフィカルデスクトップ環境のキーボード設定には影響しません。yast keyboardは次のコマンドを受け付けます。

リスト

使用可能なすべてのキーボードレイアウトを一覧表示します。

set

新しいキーボードレイアウト設定を有効にします。

```
> sudo yast keyboard set layout=czech
```

概要

現在のキーボードの設定を表示します。

4.4.3.11 yast lan

ネットワークカードを設定します。yast lanは次のコマンドを受け付けます。

追加

新しいネットワークカードを設定します。

```
> sudo yast lan add name=vlan50 ethdevice=eth0 bootproto=dhcp
```

すべてのオプションのリストについては、yast lan add helpを実行してください。

delete

既存のネットワークカードを削除します。

```
> sudo yast lan delete id=0
```

編集

既存のネットワークカードの設定を変更します。

```
> sudo yast lan edit id=0 bootproto=dhcp
```

リスト

ネットワークカードの設定の概要を表示します。

```
> sudo yast lan list
id name,                bootproto
0 Ethernet Card 0, NONE
1 Network Bridge,  DHCP
```

4.4.3.12 **yast language**

システム言語を設定します。**yast language**は次のコマンドを受け付けます。

リスト

使用可能なすべての言語を一覧表示します。

set

メインのシステム言語と第2言語を指定します。

```
> sudo yast language set lang=cs_CZ languages=en_US,es_ES no_packages
```

4.4.3.13 **yast mail**

メールシステムの設定を表示します。

```
> sudo yast mail summary
```

4.4.3.14 **yast nfs**

NFSクライアントを制御します。**yast nfs**は次のコマンドを受け付けます。

追加

新しいNFSマウントを追加します。

```
> sudo yast nfs add spec=remote_host:/path/to/nfs/share file=/local/mount/point
```

すべてのオプションのリストについては、**yast nfs add help**を実行してください。

delete

既存のNFSマウントを削除します。

```
> sudo yast nfs delete spec=remote_host:/path/to/nfs/share file=/local/mount/point
```

すべてのオプションのリストについては、**yast nfs delete help**を実行してください。

編集

既存のNFSマウントを変更します。

```
> sudo yast nfs edit spec=remote_host:/path/to/nfs/share \
file=/local/mount/point type=nfs4
```

すべてのオプションのリストについては、**yast nfs edit help**を実行してください。

リスト

既存のNFSマウントを一覧表示します。

```
> sudo yast nfs list
Server          Remote File System  Mount Point  Options
-----
nfs.example.com /mnt                /nfs/mnt     nfs
nfs.example.com /home/tux/nfs_share /nfs/tux     nfs
```

4.4.3.15 yast nfs-server

NFSサーバを設定します。**yast nfs-server**は次のコマンドを受け付けます。

追加

エクスポートするディレクトリを追加します。

```
> sudo yast nfs-server add mountpoint=/nfs/export hosts=*.allowed_hosts.com
```

すべてのオプションのリストについては、**yast nfs-server add help**を実行してください。

delete

NFSエクスポートからディレクトリを削除します。

```
> sudo yast nfs-server delete mountpoint=/nfs/export
```

set

NFSサーバの追加パラメータを指定します。

```
> sudo yast nfs-server set enablev4=yes security=yes
```

すべてのオプションのリストについては、**yast nfs-server set help**を実行してください。

start

NFSサーバサービスを起動します。

```
> sudo yast nfs-server start
```

stop

NFSサーバサービスを停止します。

```
> sudo yast nfs-server stop
```

概要

NFSサーバ設定の概要を表示します。

```
> sudo yast nfs-server summary
NFS server is enabled
NFS Exports
* /mnt
* /home

NFSv4 support is enabled.
The NFSv4 domain for idmapping is localdomain.
NFS Security using GSS is enabled.
```

4.4.3.16 yast nis

NISクライアントを設定します。**yast nis**は次のコマンドを受け付けます。

構成

NISクライアントのグローバル設定を変更します。

```
> sudo yast nis configure server=nis.example.com broadcast=yes
```

すべてのオプションのリストについては、**yast nis configure help**を実行してください。

無効

NISクライアントを無効にします。

```
> sudo yast nis disable
```

enable

マシンをNISクライアントとして有効にします。

```
> sudo yast nis enable server=nis.example.com broadcast=yes automounter=yes
```

すべてのオプションのリストについては、**yast nis enable help**を実行してください。

検索

特定のドメインで使用可能なNISサーバを表示します。

```
> sudo yast nis find domain=nisdomain.com
```

概要

NISクライアントの設定の概要を表示します。

4.4.3.17 yast nis-server

NISサーバを設定します。**yast nis-server**は次のコマンドを受け付けます。

master (マスタ)

NISマスタサーバを設定します。

```
> sudo yast nis-server master domain=nisdomain.com yppasswd=yes
```

すべてのオプションのリストについては、**yast nis-server master help**を実行してください。

slave (スレーブ)

NISワーカーサーバを設定します。

```
> sudo yast nis-server slave domain=nisdomain.com master_ip=10.100.51.65
```

すべてのオプションのリストについては、**yast nis-server slave help**を実行してください。

stop

NISサーバを停止します。

```
> sudo yast nis-server stop
```

概要

NISサーバの設定の概要を表示します。

```
> sudo yast nis-server summary
```

4.4.3.18 **yast proxy**

プロキシ設定を行います。**yast proxy**は次のコマンドを受け付けます。

認証

プロキシの認証オプションを指定します。

```
> sudo yast proxy authentication username=tux password=secret
```

すべてのオプションのリストについては、**yast proxy authentication help**を実行してください。

enable、disable

プロキシ設定を有効または無効にします。

set

現在のプロキシ設定を変更します。

```
> sudo yast proxy set https=proxy.example.com
```

すべてのオプションのリストについては、**yast proxy set help**を実行してください。

概要

プロキシ設定を表示します。

4.4.3.19 **yast rdp**

リモートデスクトップの設定を制御します。**yast rdp**は次のコマンドを受け付けます。

allow

サーバのデスクトップへのリモートアクセスを許可します。

```
> sudo yast rdp allow set=yes
```

リスト

リモートデスクトップ設定の概要を表示します。

4.4.3.20 **yast samba-client**

Sambaクライアントの設定を行います。**yast samba-client**は次のコマンドを受け付けます。

構成

Sambaのグローバル設定を変更します。

```
> sudo yast samba-client configure workgroup=FAMILY
```

isdomainmember

マシンがドメインのメンバーであるかどうかを確認します。

```
> sudo yast samba-client isdomainmember domain=SMB_DOMAIN
```

joindomain

マシンをドメインのメンバーにします。

```
> sudo yast samba-client joindomain domain=SMB_DOMAIN user=username password=pwd
```

winbind

Winbindサービス(winbinddデーモン)を有効または無効にします。

```
> sudo yast samba-client winbind enable  
> sudo yast samba-client winbind disable
```

4.4.3.21 **yast samba-server**

Sambaサーバ設定を行います。**yast samba-server**は次のコマンドを受け付けます。

backend

ユーザ情報を格納するバックエンドを指定します。

```
> sudo yast samba-server backend smbpasswd
```

すべてのオプションのリストについては、**yast samba-server backend help**を実行してください。

構成

Sambaサーバのグローバル設定を行います。

```
> sudo yast samba-server configure workgroup=FAMILY description='Home server'
```

すべてのオプションのリストについては、**yast samba-server configure help**を実行してください。

リスト

使用できる共有のリストを表示します。

```
> sudo yast samba-server list
Status      Type Name
=====
Disabled    Disk profiles
Enabled     Disk print$
Enabled     Disk homes
Disabled    Disk groups
Enabled     Disk movies
Enabled     Printer printers
```

role

Sambaサーバの役割を指定します。

```
> sudo yast samba-server role standalone
```

すべてのオプションのリストについては、**yast samba-server role help**を実行してください。

service

Sambaサービス(smbとnmb)を有効または無効にします。

```
> sudo yast samba-server service enable
> sudo yast samba-server service disable
```

share

単一のSamba共有を操作します。

```
> sudo yast samba-server share name=movies browseable=yes guest_ok=yes
```

すべてのオプションのリストについては、**yast samba-server share help**を実行してください。

4.4.3.22 yast security

ホストのセキュリティレベルを制御します。**yast security**は次のコマンドを受け付けます。

level

ホストのセキュリティレベルを指定します。


```
> sudo yast security level server
```

すべてのオプションのリストについては、**yast security level help**を実行してください。

set

特定のオプションの値を設定します。

```
> sudo yast security set passwd=sha512 crack=yes
```

すべてのオプションのリストについては、**yast security set help**を実行してください。

summary

現在のセキュリティ設定の概要を表示します。

```
sudoyast security summary
```

4.4.3.23 yast sound

サウンドカードの設定を行います。**yast sound**は次のコマンドを受け付けます。

追加

新しいサウンドカードを設定します。パラメータを指定しない場合、最初に検出されたカードが追加されます。

```
> sudo yast sound add card=0 volume=75
```

すべてのオプションのリストについては、**yast sound add help**を実行してください。

channels

サウンドカードの使用可能なボリュームチャンネルを一覧表示します。

```
> sudo yast sound channels card=0  
Master 75  
PCM 100
```

modules

使用可能なすべてのサウンドカーネルモジュールを一覧表示します。

```
> sudo yast sound modules  
snd-atiixp ATI IXP AC97 controller (snd-atiixp)  
snd-atiixp-modem ATI IXP MC97 controller (snd-atiixp-modem)
```

```
snd-virtuoso Asus Virtuoso driver (snd-virtuoso)
[...]
```

playtest

サウンドカードでテストサウンドを再生します。

```
> sudo yast sound playtest card=0
```

remove

設定されたサウンドカードを削除します。

```
> sudo yast sound remove card=0
> sudo yast sound remove all
```

set

サウンドカードの新しい値を指定します。

```
> sudo yast sound set card=0 volume=80
```

show

サウンドカードに関する詳細情報を表示します。

```
> sudo yast sound show card=0
Parameters of card 'ThinkPad X240' (using module snd-hda-intel):

align_buffer_size
  Force buffer and period sizes to be multiple of 128 bytes.
bdl_pos_adj
  BDL position adjustment offset.
beep_mode
  Select HDA Beep registration mode (0=off, 1=on) (default=1).
  Default Value: 0
enable_msi
  Enable Message Signaled Interrupt (MSI)
[...]
```

summary

システム上のすべてのサウンドカードの設定概要を出力します。

```
> sudo yast sound summary
```

volume

サウンドカードの音量レベルを指定します。

```
sudoyast sound volume card=0 play
```

4.4.3.24 yast sysconfig

/etc/sysconfigのファイル内の変数を制御します。**yast sysconfig**は次のコマンドを受け付けます。

クリア

空の値を変数に設定します。

```
> sudo yast sysconfig clear=POSTFIX_LISTEN
```



ヒント: 複数ファイルの変数

変数が複数のファイルで使用可能な場合は、VARIABLE_NAME\$FILE_NAME構文を使用します。

```
> sudo yast sysconfig clear=CONFIG_TYPE$/etc/sysconfig/mail
```

詳細

変数に関する詳細情報を表示します。

```
> sudo yast sysconfig details variable=POSTFIX_LISTEN
Description:
Value:
File: /etc/sysconfig/postfix
Possible Values: Any value
Default Value:
Configuration Script: postfix
Description:
  Comma separated list of IP's
  NOTE: If not set, LISTEN on all interfaces
```

リスト

変更された変数の概要を表示します。**all**を使用して、すべての変数とその値を一覧表示します。

```
> sudo yast sysconfig list all
AOU_AUTO_AGREE_WITH_LICENSES="false"
AOU_ENABLE_CRONJOB="true"
AOU_INCLUDE_RECOMMENDS="false"
[...]
```

set

変数の値を設定します。

```
> sudo yast sysconfig set DISPLAYMANAGER=gdm
```



ヒント: 複数ファイルの変数

変数が複数のファイルで使用可能な場合は、VARIABLE_NAME\$FILE_NAME構文を使用します。

```
> sudo yast sysconfig set CONFIG_TYPE$/etc/sysconfig/mail=advanced
```

4.4.3.25 yast tftp-server

TFTPサーバを設定します。yast tftp-serverは次のコマンドを受け付けます。

ディレクトリ

TFTPサーバのディレクトリを指定します。

```
> sudo yast tftp-server directory path=/srv/tftp
> sudo yast tftp-server directory list
Directory Path: /srv/tftp
```

status

TFTPサーバサービスのステータスを制御します。

```
> sudo yast tftp-server status disable
> sudo yast tftp-server status show
Service Status: false
> sudo yast tftp-server status enable
```

4.4.3.26 yast timezone

タイムゾーンを設定します。yast timezoneは次のコマンドを受け付けます。

リスト

使用可能なすべてのタイムゾーンを地域別にグループ化して一覧表示します。

```
> sudo yast timezone list
Region: Africa
Africa/Abidjan (Abidjan)
Africa/Accra (Accra)
Africa/Addis_Ababa (Addis Ababa)
[...]
```

set

タイムゾーン設定の新しい値を指定します。

```
> sudo yast timezone set timezone=Europe/Prague hwclock=local
```

概要

タイムゾーンの設定の概要を表示します。

```
> sudo yast timezone summary
Current Time Zone: Europe/Prague
Hardware Clock Set To: Local time
Current Time and Date: Mon 12. March 2018, 11:36:21 CET
```

4.4.3.27 **yast users**

ユーザアカウントを管理します。**yast users**は次のコマンドを受け付けます。

追加

新しいユーザを追加します。

```
> sudo yast users add username=user1 password=secret home=/home/user1
```

すべてのオプションのリストについては、**yast users add help**を実行してください。

delete

既存のユーザアカウントを削除します。

```
> sudo yast users delete username=user1 delete_home
```

すべてのオプションのリストについては、**yast users delete help**を実行してください。

編集

既存のユーザアカウントを変更します。

```
> sudo yast users edit username=user1 password=new_secret
```

すべてのオプションのリストについては、**yast users edit help**を実行してください。

リスト

ユーザタイプによってフィルタされた既存のユーザを一覧表示します。

```
> sudo yast users list system
```

すべてのオプションのリストについては、**yast users list help**を実行してください。

show

ユーザに関する詳細を表示します。

```
> sudo yast users show username=wwwrun
Full Name: WWW daemon apache
List of Groups: www
Default Group: wwwrun
Home Directory: /var/lib/wwwrun
Login Shell: /sbin/nologin
Login Name: wwwrun
UID: 456
```

すべてのオプションのリストについては、**yast users show help**を実行してください。

5 YaSTによる言語および国の設定の変更

この章では、言語と国の設定方法について説明します。言語は、システム全体でグローバルに、特定のユーザまたはデスクトップで個別に、または単一のアプリケーションで一時的に変更できます。また、第二言語を設定し、日付と国の設定を調整することもできます。

別の国または多言語環境で作業している場合、その状況に応じてシステムを設定する必要があります。SUSE® Linux Enterprise Desktopは、複数の`locales`を並行して扱うことができます。ロケールは、ユーザインタフェースに反映される言語と国を定義するパラメータのセットです。

主要言語はインストール時に選択され、キーボードとタイムゾーンの設定は適切に調整されています。ただし、追加言語をインストールしたり、インストールした言語のどれをデフォルトにするか決定することができます。

それらのタスクでは、[5.1項「システム言語の変更」](#)に説明があるYaSTの言語モジュールを使用します。第一言語以外でアプリケーションまたはデスクトップを起動する必要がある場合は、二次言語をインストールしてオプションのローカライズを適用します。

YaSTのタイムゾーンモジュールを使用すると、国およびタイムゾーンの設定を適切に調整できます。タイムゾーンモジュールでは、タイムサーバとシステムクロックを同期することもできます。詳細については、[5.2項「国および時間の設定の変更」](#)を参照してください。

5.1 システム言語の変更

デスクトップを使用する方法や、システム全体を別の言語に切り替えるかデスクトップ環境のみを切り替えるかの指定などに応じて、いくつかのオプションがあります。

システム言語をグローバルに変更する

[5.1.1項「YaSTでシステムの言語を変更する」](#)および[5.1.2項「デフォルトシステム言語を切り替える」](#)の説明に従って、YaSTで別のローカライズパッケージをインストールし、そのデフォルト言語を設定します。変更は次回ログイン後に有効になります。システム全体で変更を反映するには、システムを再起動するか、またはすべての実行サービス、アプリケーション、およびプログラムを終了して再起動します。

デスクトップの言語だけを変更する

以下の説明に従ってYaSTを使用してデスクトップ環境に目的の言語パッケージをインストール済みであれば、デスクトップのコントロールセンターを使用してデスクトップの言語を切り替えることができます。詳細については、『GNOMEユーザガイド』、第3

章「設定のカスタマイズ」、3.2項「言語の設定」を参照してください。Xサーバの再起動後、デスクトップ全体に新たに選択した言語が反映されます。デスクトップフレームワークに属さないアプリケーションでは、この変更が適用されず、YaSTで設定した言語で引き続き表示されることがあります。

1つのアプリケーションの言語だけを一時的に切り替える

1つのアプリケーションのみを、YaSTでインストール済みの別の言語で実行することもできます。そのためには、言語コードを指定して、コマンドラインからそのアプリケーションを起動します(5.1.3項「標準のXアプリケーションとGNOMEアプリケーションでの言語切り替え」参照)。

5.1.1 YaSTでシステムの言語を変更する

YaSTは、次の2つの異なる言語カテゴリをサポートしています。

第一言語

YaSTに設定された第一言語は、YaSTおよびデスクトップ環境を含んだ、システム全体に適用されます。この言語は、別の言語を手動で指定しない限り、利用可能な場合に常に使用されます。

第二言語

第二言語をインストールして、システムを多言語にします。第二言語としてインストールされている言語は、必要に応じて手動で選択できます。たとえば、一定の言語でワープロを行うため、その言語でアプリケーションを起動する場合は、第二言語を使用します。

追加の言語をインストールする前に、インストール後にそれらの中からデフォルトのシステム言語(第一言語)とするものを決めておく必要があります。

YaSTの言語モジュールにアクセスするには、YaSTを起動し、システム > 言語の順にクリックします。コマンドラインで `sudo yast2 language &` を実行して、言語ダイアログを直接起動することもできます。



手順 5.1: 追加言語をインストールする

追加言語をインストールするときに、YaSTを使用してrootユーザにいくつかのロケールを設定できます。ステップ 4を参照してください。/etc/sysconfig/languageファイルのロケール変数(LC_*)をrootに対してどのように設定するかは、rootユーザのロケール設定オプションで指定します。それらを通常ユーザ用と同じロケールに設定するか、言語の変更によってまったく影響されないようにするか、または変数RC_LC_CTYPEだけを通常ユーザ用と同じ値に設定することができます。RC_LC_CTYPE変数は、言語固有の関数呼び出しのローカライゼーションを設定します。

1. YaSTの言語モジュールで言語を追加するには、第二言語でインストールする言語を選択します。
2. 言語をデフォルト言語にするには、その言語を第一言語として設定します。
3. さらに、新しい第一言語に合わせてキーボードを変更し、必要に応じてタイムゾーンを調整します。



ヒント: 詳細設定

高度なキーボード設定を指定するには、YaSTでハードウェア > System Keyboard Layout (システムのキーボードレイアウト)の順に選択します。高度なタイムゾーン設定を指定するには、YaSTでシステム > 日付と時刻の順に選択します。詳細については、[第32章「システムのキーボードレイアウト設定」](#)および[5.2項「国および時間の設定の変更」](#)を参照してください。

4. rootユーザに固有の言語設定を変更するには、詳細情報をクリックします。
 - a. rootユーザのロケール設定を目的の値に設定します。詳細については、ヘルプをクリックします。
 - b. rootにUTF-8 エンコーディングを使用を使用するかどうかを決めます。
5. ロケールが利用可能な第一言語のリストに含まれていない場合は、詳細なロケール設定で、そのロケールを指定してください。ただし、特定のロケールが不完全になる場合があります。
6. OKをクリックして、ダイアログで行った変更を確認します。第二言語を選択していれば、その追加言語にローカライズされたソフトウェアパッケージがYaSTによってインストールされます。

これで、システムが多言語になります。ただし、第一言語以外の言語でアプリケーションを起動するには、該当する言語を[5.1.3項「標準のXアプリケーションとGNOMEアプリケーションでの言語切り替え」](#)の説明どおりに明示的に設定する必要があります。

5.1.2 デフォルトシステム言語を切り替える

デフォルトのシステム言語をグローバルに切り替えるには、次の手順に従います。

1. YaSTの言語モジュールを起動します。
2. 第一言語で新しいシステム言語を選択します。

！ 重要: 以前のシステム言語の削除

別の第一言語に切り替えると、以前の第一言語用にローカライズされたソフトウェアパッケージがシステムから削除されます。デフォルトシステム言語を切り替えても、以前の第一言語は追加言語として保持するには、該当するチェックボックスをオンにすることで、以前の第一言語を第二言語として追加できます。

3. キーボードとタイムゾーンのオプションを適宜調整します。
4. OKをクリックして、変更を確認します。
5. YaSTによって変更内容が適用された後、現在のXセッションを再起動して(たとえば、いったんログアウトして再度ログインします)、YaSTとデスクトップアプリケーションに新しい言語設定が反映されるようにします。

5.1.3 標準のXアプリケーションとGNOMEアプリケーションでの言語切り替え

YaSTで各言語をインストールした後、1つのアプリケーションのみを他のアプリケーションとは別の言語で実行できます。

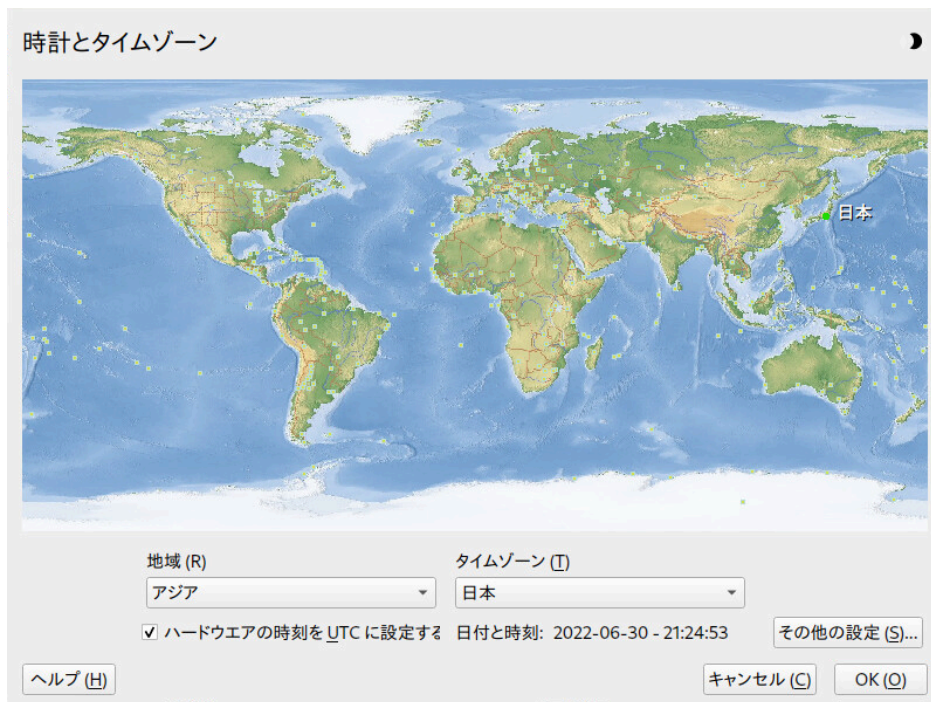
次のコマンドで、アプリケーションをコマンドラインから起動します。

```
LANG=LANGUAGE application
```

たとえば、f-spotをドイツ語で起動するには、**LANG=de_DE f-spot**を実行します。他の言語については、適切な言語コードを使用します。利用可能なすべての言語コードのリストは、**locale -av**コマンドで取得します。

5.2 国および時間の設定の変更

YaSTの日付と時刻モジュールを使用して、システムの日付、時計、およびタイムゾーンの情報を作業地域に合わせて調整します。YaSTのモジュールにアクセスするには、YaSTを起動してシステム > 日付と時刻の順にクリックします。コマンドラインで**sudo yast2 timezone &**を実行して、時計とタイムゾーンダイアログを直接起動することもできます。



まず、ヨーロッパなどの一般的な地域を選択します。作業地と一致する国(たとえば、ドイツ)を選択します。

ワークステーションで実行されるオペレーティングシステムに応じて、ハードウェアクロックの設定を調整します。

- マシン上でMicrosoft Windows*などの別のオペレーティングシステムを実行している場合、システムでは、UTCではなく、ローカルタイムを使用している可能性があります。この場合は、ハードウェアの時刻をUTCに設定するをオフにします。
- コンピュータでLinuxだけを実行する場合は、ハードウェアクロックをUTCに設定し、標準時間から夏時間への切り換えを自動的に実行させます。

！ 重要: UTCへのハードウェアクロックの設定

標準時間からサマータイムへの転換(およびその逆)は、ハードウェアクロック(CMOSクロック)がUTCに設定されている場合にのみ、自動的に行われます。この処理は、NTPとの時間の自動同期機能を使用している場合にも実行されます。これは、ハードウェアとシステムクロックの時間差が15分未満であれば、時間の自動同期が機能するからです。

誤ったシステム時間は、深刻な問題の原因になる場合があります(バックアップの失敗、メールメッセージの削除、リモートファイルシステムでの障害の発生など)。ハードウェアのクロックを「常に」UTCに設定することを強くお勧めします。

日付と時刻を手動で変更できるほか、NTPサーバにマシンが永続的に同期できるようにするか、ハードウェアの時刻を調整する目的でのみ同期するかを選択できます。

手順 5.2: 日付と時刻を手動で調整する

1. YaSTのタイムゾーンモジュールで、その他の設定をクリックして日付と時刻を設定します。
2. 手動を選択し、日時の値を入力します。
3. 変更内容を確認します。

手順 5.3: NTPサーバにより日付と時刻を設定する

1. 日付と時刻を変更するには、その他の設定をクリックします。
2. NTPサーバと同期を選択します。
3. まだ入力されていない場合は、NTPサーバのアドレスを入力します。

4. 設定ボタンをクリックすると、「高度なNTP設定」を開くことができます。詳細については、[39.1項「YaSTでのNTPクライアントの設定」](#)を参照してください。
5. 変更内容を確認します。

6 YaSTによるユーザの管理

インストール時にシステム用のローカルユーザを作成できました。YaSTのユーザとグループの管理モジュールを使用して、ユーザの追加や既存ユーザの編集が可能です。また、ネットワークサーバでユーザを認証するようにシステムを設定できます。

6.1 [ユーザとグループの管理] ダイアログ

ユーザまたはグループを管理するには、YaSTを起動し、セキュリティとユーザ > ユーザとグループの管理の順にクリックします。コマンドラインで `sudo yast2 users &` を実行して、ユーザとグループの管理ダイアログを直接起動することもできます。

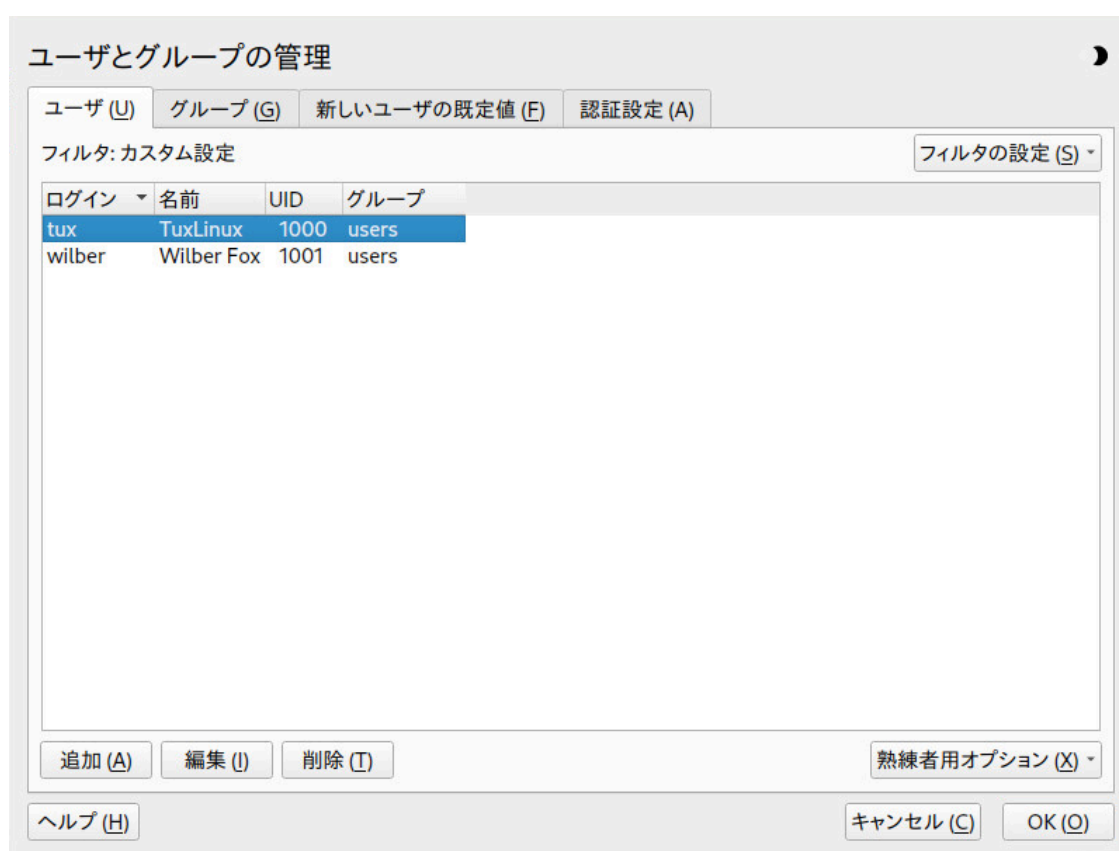


図 6.1: YASTのユーザとグループの管理

各ユーザには、システム全体で利用できるユーザID (UID)が割り当てられます。マシンにログインできるユーザ以外にも、内部での使用のみが目的のさまざまな「システムユーザ」が存在します。各ユーザは、1つ以上のグループに割り当てられます。「システムユーザ」と同様に、内部用途の「システムグループ」も存在します。

メインウィンドウには、表示および変更するために選択したユーザセット(ローカルユーザ、ネットワークユーザ、システムユーザ)に応じていくつかのタブが表示されます。タブでは、次のタスクを実行できます。

ユーザアカウントの管理

ユーザタブから、6.2項「[ユーザアカウントの管理](#)」の説明に従って、ユーザアカウントを作成、変更、削除、または一時的に無効にします。6.3項「[ユーザアカウントの追加オプション](#)」では、パスワードポリシーの強制、暗号化したホームディレクトリの使用、ディスククォータの管理などの高度なオプションについて説明しています。

デフォルト設定の変更

新しいユーザのデフォルト設定タブで定義された設定に応じて、ローカルユーザアカウントが作成されます。6.4項「[ローカルユーザのデフォルト設定の変更](#)」では、デフォルトのグループ割り当て、またはホームディレクトリのデフォルトパスおよびアクセス許可を変更する方法を説明します。

グループへのユーザの割り当て

6.5項「[グループへのユーザの割り当て](#)」では、個別ユーザのグループの割り当てを変更する方法を説明します。

グループを管理する

グループタブから、既存のグループの追加、変更、または削除を行うことができます。この方法については、6.6項「[グループを管理する](#)」を参照してください。

ユーザ認証方法を変更する

コンピュータがNISやLDAPなどのユーザ認証方法を提供するネットワークに接続されている場合は、認証設定タブで、認証方法を選択できます。詳細については、6.7項「[ユーザ認証方法を変更する](#)」を参照してください。

ユーザとグループの管理用に、このダイアログでは同様の機能が提供されます。ダイアログ上部にある適切なタブを選択することにより、ユーザとグループの管理ビューを簡単に切り替えることができます。

[フィルタ] オプションで、変更するユーザまたはグループを定義できます。ユーザまたはグループタブで、フィルタの設定をクリックすると、ユーザまたはグループを表示および編集できます。該当する場合、ローカルユーザやLDAPユーザなどの特定のカテゴリに応じて一覧表示されます。フィルタの設定 > フィルタのカスタマイズで、カスタムフィルタをセットアップおよび使用できます。

選択したフィルタに応じて、このダイアログから次のオプションおよび機能がすべて利用できる限りません。

6.2 ユーザアカウントの管理

YaSTでは、ユーザアカウントの作成、変更、削除、または一時的な無効化が可能です。熟練したユーザか管理者でない限り、ユーザアカウントを変更しないでください。



注記: 既存ユーザのユーザIDを変更する

ファイル所有権はユーザ名ではなくユーザIDにバインドされます。ユーザIDの変更後、この変更に合わせてユーザのホームディレクトリのファイルが自動的に調整されます。ただし、ユーザは、IDの変更後、ファイルシステム内の他の場所で作成したファイルのファイル所有権を失います(それらのファイルの所有権が手動で変更されない限り)。

次の手順は、デフォルトのユーザアカウントの設定方法を示しています。さらに詳細なオプションについては、[6.3項「ユーザアカウントの追加オプション」](#)を参照してください。

手順 6.1: ユーザアカウントを追加または変更する

1. YaSTのユーザとグループの管理ダイアログを開き、ユーザタブをクリックします。
2. フィルタを設定では、管理するユーザセットを定義します。このダイアログには、システムのユーザ、およびユーザが属するグループが一覧にされます。
3. 既存のユーザに対するオプションを変更するには、エントリを選択し、編集をクリックします。
新しいユーザアカウントを作成するには、追加をクリックします。
4. (ログインで使用される)ユーザ名およびパスワードなど、最初のタブで適切なユーザデータを入力します。このデータは、新しいユーザを作成するために十分なものです。ここでOKをクリックすると、システムは、ユーザIDを自動的に割り当て、その他すべての値をデフォルトに設定します。
5. このユーザのメールボックスにシステム通知が配信されるようにする場合は、システムメールの受信を有効にします。これによってrootのメールエイリアスが作成され、このユーザは最初にrootとしてログインしなくてもシステムメールが読めるようになります。
システムサービスにより送信されたメールは、ローカルメールボックス/var/spool/mail/USERNAMEに保存されます(USERNAMEは選択されたユーザのログイン名)。電子メールを読むには、**mail**コマンドを使用できます。
6. ユーザIDまたはユーザのホームディレクトリへのパスなど、さらに詳細な情報を調整するには、詳細タブを使用します。

既存のユーザのホームディレクトリを再配置する必要がある場合は、新しいホームディレクトリへのパスを入力し、新しい場所に移動により現在のホームディレクトリの内容を移動します。ホームディレクトリを再配置する必要がない場合は、既存データが存在しなくても新しいホームディレクトリが作成されます。

7. パスワードを定期的に変更することをユーザに強制するか、他のパスワードオプションを設定するには、パスワードの設定に切り替え、オプションを調整します。詳細については、6.3.2項「パスワードポリシーの強制」を参照してください。
8. すべてのオプションが希望どおりに設定されたら、OKをクリックします。
9. OKをクリックして、管理ダイアログを閉じ、変更内容を保存します。新たに追加されたユーザは、作成済みのログイン名とパスワードを使用してシステムにログインできるようになります。
また、ユーザとグループの管理ダイアログを閉じずにすべての変更を保存するには、熟練者向けオプション、変更を今すぐ書き込む順にクリックします。



ヒント: ユーザIDの一致

ネットワーク内でのIDに(ローカル)ユーザIDを一致させると便利です。たとえば、ラップトップの新しい(ローカル)ユーザは、ネットワーク環境に統合する際に同じユーザIDを割り当てる必要があります。これにより、ユーザが「オフライン」で作成するファイルのファイル所有権は、直接ネットワーク上で作成した場合のファイル所有権と同じになります。

手順 6.2: ユーザアカウントを無効化または削除する

1. YaSTのユーザとグループの管理ダイアログを開き、ユーザタブをクリックします。
2. ユーザアカウントを削除しないで一時的に無効にするには、リストからユーザを選択し、編集をクリックします。ユーザログインを禁止するを有効にします。ユーザは、アカウントを再び有効にするまで、マシンにログインできません。
3. ユーザアカウントを削除するには、リストからユーザを選択して、削除をクリックします。ユーザのホームディレクトリを削除するか、またはこのデータを保持するかを選択します。

6.3 ユーザアカウントの追加オプション

SUSE® Linux Enterprise Desktopには、デフォルトユーザアカウントの設定のほか、さまざまなオプションも用意されています。たとえば、パスワードポリシーの強制、暗号化したホームディレクトリの使用、ユーザとグループのディスククォータの定義のためのオプションがあります。

6.3.1 自動ログインおよびパスワードレスログイン

GNOMEデスクトップ環境を使用している場合、特定のユーザには「自動ログイン」を設定し、すべてのユーザに「パスワードなしのログイン」を設定できます。自動ログインでは、ユーザがブート時にデスクトップ環境に自動的にログインします。この機能は、一度に1人のユーザについてのみ有効にできます。パスワードなしのログインでは、どのユーザも、ログインマネージャにユーザ名を入力するだけでシステムにログインできます。



警告: セキュリティリスク

複数のユーザがアクセスできるマシンで「自動ログイン」または「パスワードレスログイン」を有効にすることはセキュリティ上のリスクを伴います。どのユーザでもシステムおよびデータにアクセスでき、認証の必要もありません。システムに機密情報などの重要なデータを保管している場合は、この機能は使用しないでください。

自動ログインまたはパスワードなしのログインを有効にするには、**熟練者向けオプション** > ログイン設定の順に選択し、YaSTのユーザとグループの管理でこれらの機能にアクセスします。

6.3.2 パスワードポリシーの強制

複数のユーザが使用するシステムでは、最低限のパスワードセキュリティポリシーを強制することをお勧めします。ユーザに定期的にパスワードを変更させたり、推測しにくいような複雑なパスワードを使用させることができます。ローカルユーザの場合は、次の手順に従います。

手順 6.3: パスワードを設定する

1. YaSTのユーザとグループの管理ダイアログを開き、ユーザタブを選択します。
2. ユーザを選択し、編集をクリックします。

3. パスワードの設定タブに切り替えます。ユーザの最後のパスワード変更がタブに表示されています。
4. 次回のログインでパスワードを変更するようにユーザに強制するには、次のログイン時にパスワード変更を強制するを有効にします。
5. パスワードのローテーションを強制するには、同じパスワードを使用できる最長日数および同じパスワードを使用する最短日数を設定します。
6. 期限切れになる前にパスワードを変更するようにユーザに通知するには、パスワード失効予告日数に日数を設定します。
7. パスワードが期限切れになった後ユーザがログインできる期間を制限するには、パスワードの有効期限切れログインを使用できる日数の値を変更します。
8. また、アカウント全体の特定の有効期限を指定できます。有効期限をYYYY-MM-DD形式で入力します。これはパスワード関連の設定ではなく、アカウント自体に適用されます。
9. オプションおよびデフォルト値の詳細については、ヘルプをクリックしてください。
10. 変更内容を反映するには、OKをクリックします。

6.3.3 クォータの管理

システム容量が通知なく枯渇することのないように、システム管理者はユーザまたはグループに対するクォータを設定できます。クォータは、1つ以上のファイルシステムに対して定義されるもので、これにより使用可能なディスク容量および作成可能なiノード(インデックスノード)の数を制限できます。iノードは、通常のファイル、ディレクトリ、または他のファイルシステムオブジェクトに関する基本的な情報を保存するファイルシステム上のデータ構造です。また、ファイル名とコンテンツを除いて、ファイルシステムオブジェクト(ユーザおよびグループの所有権、読み取り、書き込み、または実行のパーミッションなど)のすべての属性を保存します。

SUSE Linux Enterprise Desktopでは、softクォータとhardクォータを使用できます。さらに、ユーザまたはグループが特定量まで一時的にクォータを超過できる猶予間隔を定義できます。

ソフトクォータ

限界に近づいたときにユーザに通知する警告レベルを定義します。管理者は、パーティションのデータのクリーンアップと削減を行うようにユーザに促すことができます。通常、ソフトクォータの限界値は、ハードクォータの限界値よりも低くなります。

ハードクォータ

書き込み要求が拒否される限界を定義します。ハードクォータに達すると、それ以上データを格納することができなくなり、アプリケーションがクラッシュする可能性が高くなります。

Grace period (猶予期間)

ソフトクォータに達してから警告の発行までの時間を定義します。通常、1時間、数時間など小さな値を設定します。

手順 6.4: パーティションのクォータサポートの有効化

特定のユーザおよびグループにクォータを設定するには、YaSTのエキスパートパーティショナで、対応するパーティションのクォータサポートを有効にしておく必要があります。

1. YaSTでシステム > ディスクの分割の順に選択し、はいをクリックして続行します。
2. エキスパートパーティショナで、クォータを有効にするパーティションを選択して、編集をクリックします。
3. Fstabオプションをクリックし、Enable Quota Supportを有効にします。quotaパッケージがまだインストールされていない場合は、はいのクリックで各メッセージを確認することにより、クォータパッケージがインストールされます。
4. 変更を確認し、エキスパートパーティショナを終了します。
5. 次のコマンドを入力して、quotaonサービスが実行されていることを確認してください。

```
> sudo systemctl status quotaon.service
```

サービスは、activeとしてマークされている必要があります。そうでない場合は、**systemctl start quotaon.service**コマンドを使用して開始する必要があります。

手順 6.5: ユーザまたはグループのクォータを設定する

これで、特定のユーザまたはグループに対するソフトクォータまたはハードクォータを定義し、猶予間隔を指定できます。

1. YaSTのユーザとグループの管理で、クォータの設定対象とするユーザまたはグループを選択し、編集をクリックします。
2. プラグインタブで、ユーザクォータの管理のエントリを選択してから、起動をクリックしてクォータの設定ダイアログを開きます。

3. ファイルシステムから、クォータを適用するパーティションを選択します。

クォータの設定

ファイルシステム (F)
/dev/sda4

サイズ制限

ソフト制限 (S)
5000

ハード制限 (H)
75000

日 時間 分 秒
0 0 0 0

inode 制限

ソフト制限 (O)
0

ハード制限 (D)
0

日 時間 分 秒
0 0 0 0

ヘルプ (H) キャンセル (C) OK (O)

4. サイズ制限では、ディスクスペースの容量を制限します。ユーザまたはグループがこのパーティションで持つことができる1KBブロックの数を入力します。ソフト制限およびハード制限の値を指定します。
5. さらに、ユーザまたはグループがこのパーティションで持つことができるiノードの数を制限できます。inode制限で、ソフト制限およびハード制限を入力します。
6. サイズまたはiノードに対して指定されたソフト制限をユーザまたはグループが既に超過している場合にのみ猶予間隔を定義できます。このソフト制限を超過していない場合、時間に関連するテキストボックスは有効になりません。ユーザまたはグループが上記の制限セットを超過できる期間を指定します。
7. 入力した設定を確認して、OKをクリックします。
8. OKをクリックして、管理ダイアログを閉じ、変更内容を保存します。
また、ユーザとグループの管理ダイアログを閉じずにすべての変更を保存するには、熟練者向けオプション「変更を今すぐ書き込む」順をクリックします。

SUSE Linux Enterprise Desktopには、`repquota`や`warnquota`などのコマンドラインツールも付属しています。システム管理者はこれらのツールを使用してディスク使用量を制限したり、所定のクォータを超過しているユーザに電子メール通知を送信したりすることができます。管

理者はまた、[quota_nld](#)を使用することにより、超過したクォータに関するカーネルメッセージをD-BUSに転送できます。詳細については、[repquota](#)、[warnquota](#)、および[quota_nld](#)のマニュアルページを参照してください。

6.4 ローカルユーザのデフォルト設定の変更

新しくローカルユーザを作成する際には、いくつかのデフォルト設定がYaSTで使用されます。これらには、たとえば、ユーザが属するプライマリグループとセカンダリグループ、ユーザのホームディレクトリのアクセスパーミッションなどが含まれます。これらのデフォルト設定値は、必要に応じて変更することができます。

1. YaSTのユーザとグループの管理ダイアログを開き、新しいユーザの既定値タブを選択します。
2. 新しいユーザが自動的に属するプライマリグループを変更するには、デフォルトグループから別のグループを選択します。
3. 新しいユーザのセカンダリグループを変更するには、セカンダリグループでグループを追加するか変更します。グループ名はカンマで区切る必要があります。
4. 新しいユーザのホームディレクトリのデフォルトパスとして`/home/USERNAME`を使用しない場合は、ホームディレクトリのパスプレフィックスを変更します。
5. 新たに作成したホームディレクトリのデフォルトのパーミッションモードを変更するには、ホームディレクトリ用のUmaskのumask値を調整します。の詳細については、『Security and Hardening Guide』、第19章「Access control lists in Linux」および[umaskumask](#)のマニュアルページを参照してください。
6. それぞれのオプションの詳細については、ヘルプをクリックしてください。
7. 変更内容を反映するには、OKをクリックします。

6.5 グループへのユーザの割り当て

ユーザとグループの管理ダイアログの新しいユーザの既定値タブからアクセス可能なデフォルト設定に従って、さまざまなグループにローカルユーザが割り当てられます。次に、個別ユーザのグループ割り当てを変更する方法を説明します。新しいユーザに対するデフォルトのグループの割り当てを変更する必要がある場合については、[6.4項「ローカルユーザのデフォルト設定の変更」](#)を参照してください。

手順 6.6: ユーザのグループ割り当てを変更する

1. YaSTのユーザとグループの管理ダイアログを開き、ユーザタブをクリックします。ユーザ、およびユーザが属するグループが一覧にされます。
2. 編集をクリックし、詳細タブに切り替えます。
3. ユーザが属するプライマリグループを変更するには、デフォルトグループをクリックし、リストからグループを選択します。
4. 追加のセカンダリグループをユーザに割り当てるには、追加のグループのリストで対応するチェックボックスをオンにします。
5. OKをクリックして、変更を適用します。
6. OKをクリックして、管理ダイアログを閉じ、変更内容を保存します。
また、ユーザとグループの管理ダイアログを閉じずにすべての変更を保存するには、熟練者向けオプション「変更を今すぐ書き込む」順をクリックします。

6.6 グループを管理する

YaSTでは、グループの追加、変更、または削除も容易に実行できます。

手順 6.7: グループを作成および変更する

1. YaSTのユーザとグループの管理ダイアログを開き、グループタブをクリックします。
2. フィルタを設定では、管理するグループセットを定義します。ダイアログに、システム内のグループが一覧にされます。
3. 新しいグループを追加するには、追加をクリックします。
4. 既存のグループを変更するには、グループを選択して編集をクリックします。
5. 次のダイアログで、データを入力または変更します。右のリストでは、グループのメンバになることができる利用可能なすべてのユーザおよびシステムユーザの概要が表示されます。



6. 新しいグループに既存のユーザを追加するには、選択可能なグループのメンバーのリストで、該当するボックスをオンにして選択します。既存のユーザをグループから削除するには、このボックスをオフにします。
7. OKをクリックして、変更を適用します。
8. OKをクリックして、管理ダイアログを閉じ、変更内容を保存します。
また、ユーザとグループの管理ダイアログを閉じずにすべての変更を保存するには、熟練者向けオプション › 変更を今すぐ書き込むの順にクリックします。

グループを削除するには、すべてのグループメンバーを削除する必要があります。グループを削除するには、リストからグループを選択し、削除をクリックします。OKをクリックして、管理ダイアログを閉じ、変更内容を保存します。また、ユーザとグループの管理ダイアログを閉じずにすべての変更を保存するには、熟練者向けオプション › 変更を今すぐ書き込むの順にクリックします。

6.7 ユーザ認証方法を変更する

マシンがネットワークに接続されている場合は認証方法を変更できます。次のオプションを指定できます。

NIS

ユーザはネットワーク上のすべてのシステムに対し、1台のNISサーバ上で集中的に管理されます。詳細については、『Security and Hardening Guide』、第3章「Using NIS」を参照してください。

SSSD

システムセキュリティサービスデーモン(SSSD)は、ユーザデータをローカルにキャッシュすることにより、実際のディレクトリサービスが(一時的に)アクセス不能な場合でもユーザがデータを利用できるようにします。詳細については、『Security and Hardening Guide』、第4章「Setting up authentication clients using YaST」、4.2項「SSSD」を参照してください。

Samba

SMB認証は、通常、LinuxとWindowsが混在するネットワークで使用されます。詳細については、『Security and Hardening Guide』、第7章「Active Directory support」を参照してください。

認証方法を変更するには、以下の手順に従ってください。

1. YaSTのユーザとグループの管理ダイアログを開きます。
2. 認証設定タブをクリックすると、利用可能な認証方法と現在の設定の概要が表示されます。
3. 認証方法を変更するには、設定をクリックし、変更する認証方法を選択します。これにより、YaSTのクライアント設定モジュールに直接切り替わります。適切なクライアントの設定について詳細は、次のセクションを参照してください。

NIS: 『Security and Hardening Guide』、第3章「Using NIS」、3.2項「Configuring NIS clients」

LDAP: 『Security and Hardening Guide』、第4章「Setting up authentication clients using YaST」、4.1項「Configuring an authentication client with YaST」

SSSD: 『Security and Hardening Guide』、第4章「Setting up authentication clients using YaST」、4.2項「SSSD」

4. この設定を確認した後、ユーザとグループの管理の概要に戻ります。
5. OKをクリックして、管理ダイアログを閉じます。

6.8 デフォルトのシステムユーザ

デフォルトでは、SUSE Linux Enterprise Desktopに削除できないユーザ名が作成されます。これらのユーザは通常、Linux Standard Baseで定義されます。次のリストに、一般的なユーザ名とその目的を示します。

デフォルトでインストールされる一般的なユーザ名

bin,

daemon

レガシアプリケーションとの互換性を維持するために用意されているレガシユーザ。新しいアプリケーションではこのユーザ名を使用しないでください。

gdm

グラフィカルログインを提供したり、ローカル表示とリモート表示を管理したりするために、GNOMEディスプレイマネージャ(GDM)によって使用されています。

lp

CUPS (Common Unix Printing System)用にプリンタデーモンによって使用されています。

mail

sendmailやpostfixなどの、ユーザ用に予約されたメーラープログラム。

man

manページへのアクセスに使用されています。

messagebus

プロセス間通信用のソフトウェアバスであるD-Bus (デスクトップバス)へのアクセスに使用されています。デーモンはdbus-daemonです。

nobody

ファイルを所有せず権限付きグループに属していないユーザ。Linux Standard Baseは、デーモンごとに別個のユーザアカウントを設定することを推奨しているため、現在ではあまり使用されていません。

nscd

ネームサービスキャッシュデーモンによって使用されています。このデーモンは、NISとLDAPのパフォーマンスを向上させる参照サービスです。デーモンはnscdです。

polkitd

PolicyKit認可フレームワークによって使用されています。このフレームワークは、権限のないプロセスの認可要求を処理します。デーモンはpolkitdです。

postfix

Postfixメーラーによって使用されています。

pulse

Pulseaudioサウンドサーバによって使用されています。

root

適切なすべての権限を付与するシステム管理者が使用しています。

rpc

RPCポートマッパーであるrpcbindコマンドによって使用されています。

rtkit

リアルタイムスケジューリングモード用のD-Busシステムサービスを提供するrtkitパッケージによって使用されています。

salt

Saltが提供しているパラレルリモート実行用のユーザ。デーモンの名前はsalt-masterです。

scard

スマートカードとそのリーダーとの通信を行うユーザ。デーモンの名前はpcscdです。

srvGeoClue

位置情報を提供するためにGeoClue D-Busサービスによって使用されています。

sshd

セキュアでないネットワーク上で暗号化されたセキュアな通信を確保するためにSecure Shellデーモン(SSH)によって使用されています。

statd

ネットワークステータスマニタ(NSM)プロトコルによって使用されています。再起動通知をリッスンするためにrpc.statdデーモンで実装されます。

systemd-coredump

コアダンプの取得、保存、処理を行うために/usr/lib/systemd/systemd-coredumpコマンドによって使用されています。

systemd-timesync

リモートのネットワークタイムプロトコル(NTP)サーバとローカルシステムクロックを同期させるために/usr/lib/systemd/systemd-timesyncdコマンドによって使用されています。

7 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、[8.5項「GNOMEパッケージアップデート」](#)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 Desktopの現在のパッチは、アップデートソフトウェアリポジトリから入手できます。インストール時に製品を登録した場合、アップデートリポジトリはすでに設定されています。SUSE Linux Enterprise Desktopを登録していない場合は、YaSTで製品登録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、[8.4項「ソフトウェアリポジトリおよびサービスの管理」](#)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 Desktop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 Customer Center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。



注記: 更新を受信するためのファイアウォール設定

デフォルトでは、SUSE Linux Enterprise Desktopのファイアウォールは着信接続のみをブロックします。システムが発信トラフィックをブロックする別のファイアウォールの背後にある場合は、更新を受信するために、ポート80および443の<https://scc.suse.com/>および<https://updates.suse.com/>への接続を許可する必要があります。

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。

7.1 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。コマンドラインで `yast2 online_update` を実行して、このモジュールを開始することもできます。

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 7.1: 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 Desktopの使用可能なパッチの一覧が示されます。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。

必要な修正(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。

必要のない修正

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 Desktop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。

7.2 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。

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、適用または受諾をクリックします。
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、7.1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には、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、システムは最新の状態になりました。

7.3 撤回されたパッチの表示

バグが発生するリスクを最小限に抑えるために、保守アップデートは慎重にテストされます。パッチにバグが含まれていることが判明した場合、パッチは自動的に撤回されます。バグのあるパッチを元に戻すために、新しいアップデート(バージョン番号が大きい)が発行され、再インストールがブロックされます。撤回されたパッチとその履歴は、パッケージの分類タブで確認できます。

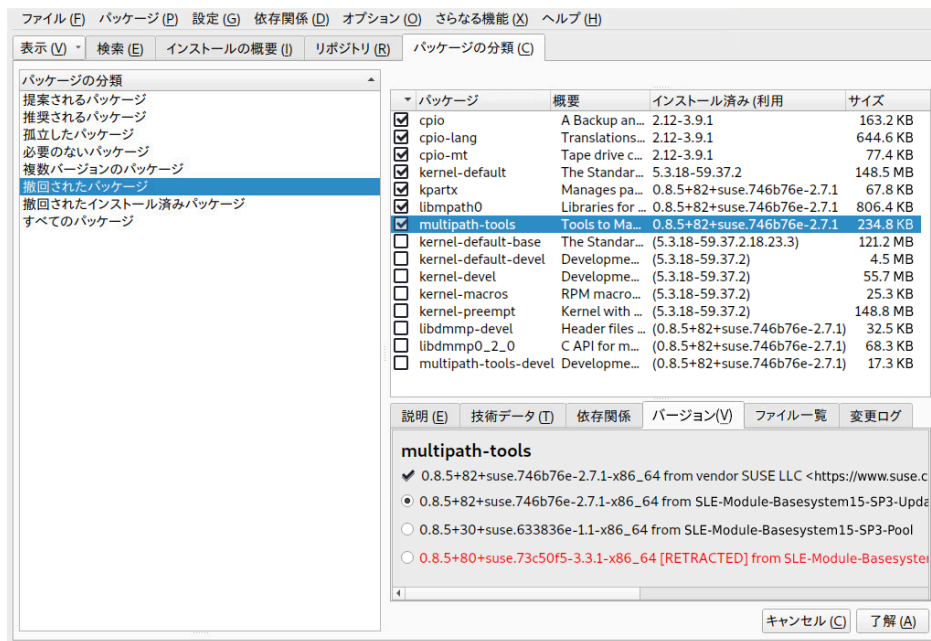


図 7.2: 撤回されたパッチと履歴の表示

7.4 自動オンラインアップデート

YaSTを使用して、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できます。[yast2-online-update-configuration](#)パッケージをインストールします。

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集中的に使用するので、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。

YaST ソフトウェアモジュールのパッチタブを使用して、バグレポートやCVE揭示への参照を含む、使用可能なパッチ、インストール済みのパッチを確認できます。

手順 7.2: 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 > オンライン更新の順に選択します。環境設定 > オンライン更新の順に選択します。[yast2-online-update-configuration](#)がインストールされていない場合、インストールするように求められます。

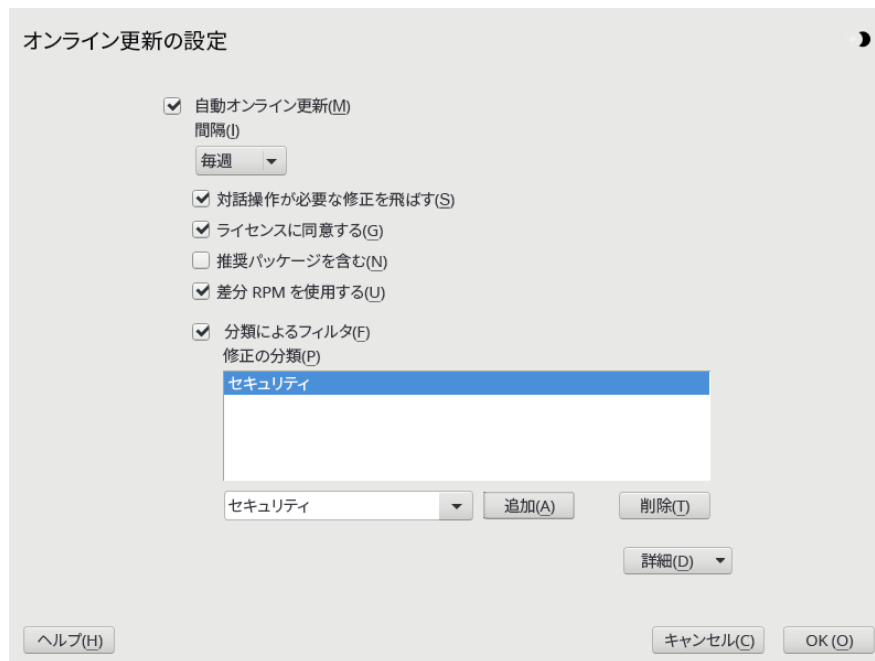


図 7.3: YASTオンライン更新設定

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. 更新間隔として毎日、毎週、または毎月を選択します。
3. パッチで重要なサービスを再起動する際などに、管理者の注意が必要な場合があります。たとえば、すべてのコンテナの再起動が必要なDocker Open Source Engineのアップデートの場合などです。これらのパッチをインストールする前に、ユーザはその結果について知らされ、パッチのインストールの確認を求められます。このようなパッチは「対話操作が必要な修正」と呼ばれます。
パッチを自動的にインストールする場合は、対話操作が必要な修正のインストールを了承しているとみなされます。インストールする前にこれらのパッチを確認する場合は、対話操作が必要な修正を飛ばすを選択します。この場合、自動的なパッチ適用中に対話操作が必要な修正は飛ばされます。手動オンラインアップデートを定期的に行って、対話操作が必要な修正がインストールを待機しているかどうかを確認します。
4. ライセンス契約を自動的に受諾するには、ライセンスに同意するを有効にします。
5. 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、Include Recommended Packagesを有効にします。
6. 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。

7. 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、分類によるフィルタ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。自動的にセキュリティアップデートのみを有効にして、その他すべてを手動で確認することをお勧めします。パッチの適用は通常信頼できますが、セキュリティ以外のパッチをテストし、問題がある場合はロールバックすることもできます。

- パッケージマネージャとYaSTでは、パッケージ管理および、YaST機能とモジュール用のパッチを提供します。
- セキュリティパッチでは、重要なアップデートとバグ修正を提供します。
- 推奨パッチは、オプションのバグ修正と拡張です。
- オプションは新しいパッケージです。
- その他はその他に相当します。
- ドキュメントは使用されていません。

8. OKをクリックして設定を確認します。

自動オンラインアップデートでは、システムは後で自動的に再起動されません。システムの再起動が必要なシステムの更新がある場合は、手動で再起動する必要があります。

8 ソフトウェアをインストールまたは削除する

YaSTのソフトウェア管理モジュールを使用すると、ソフトウェアパッケージの検索、インストール、削除を実行できます。パッケージをインストールするとき、YaSTは、すべての依存関係を自動的に解決します。インストールメディアにないパッケージをインストールするために、ソフトウェアリポジトリおよびYaSTを追加し、それらを管理できます。アップデートアプレットを使用してソフトウェアアップデートを管理することによって、システムを最新に保つこともできます。

YaSTソフトウェアマネージャを使用すると、システム上でソフトウェアソースを管理できます。このYaSTモジュールには2つのバージョンがあります。X Windowのグラフィックバージョンと、コマンドラインで使用するテキストベースバージョンです。以下ではグラフィックバージョンについて説明します。テキストベースのYaSTについては第4章「テキストモードのYaST」を参照してください。



注記: 変更の確認とレビュー

パッケージのインストール、更新、または削除を行う場合、ソフトウェアマネージャでの変更は、了解または適用で確認後にだけ適用されます。YaSTでは、すべてのアクションを記録したリストが保持されているので、変更内容をシステムに適用する前に、それらを確認し、必要に応じて変更できます。

8.1 用語の定義

SUSE Linux Enterprise Desktopでのソフトウェアのインストールと削除に精通するには、次の用語を理解しておく必要があります。

リポジトリ

パッケージとそのパッケージに関する追加情報(パッケージメタデータ)を保存しているローカルディレクトリまたはリモートディレクトリ。

(リポジトリの)エイリアス/リポジトリ名

リポジトリの短い名前(ZypperではAliasと呼び、YaSTではリポジトリ名と呼びます)。これは、リポジトリを追加するときにユーザが選択できますが、固有の名前とする必要があります。

リポジトリ記述ファイル

各リポジトリは、リポジトリのコンテンツ(パッケージ名、バージョンなど)を説明したファイルを提供します。これらのリポジトリ記述ファイルは、YaSTで使用するローカルキャッシュにダウンロードされます。

製品

SUSE® Linux Enterprise Desktopなどの製品全体を指します。

パターン

パターンは、特定の用途専用に設計されたパッケージのインストール可能なグループです。たとえば、Laptopパターンには、モバイルコンピューティング環境に必要なすべてのパッケージが含まれています。パターンは、パッケージ依存関係を定義し(必須パッケージや推奨パッケージなど)、インストール用としてマークされたパッケージが事前選択されている状態で提供されます。これによって、特定の用途に必要な最も重要なパッケージが、パターンのインストール後にシステムで使用可能になります。パターン内のパッケージは、必要に応じて手動で選択または選択解除できます。

Package

パッケージは、rpm形式の圧縮ファイルであり、特定のプログラムのファイルを含んでいます。

パッチ

パッチは、1つ以上のパッケージから成り、デルタRPMで適用できます。また、まだインストールされていないパッケージへの依存関係を導入することもあります。

解決可能

製品、パターン、パッケージ、またはパッチに関する一般的な用語。最も一般に使用される解決可能なタイプは、パッケージまたはパッチです。

デルタRPM

デルタRPMは、パッケージに定義された2つのバージョンどうしのバイナリ差分のみで構成されているので、ダウンロードサイズが最小限ですみます。インストールの前に、RPMのフルパッケージがローカルコンピュータ上で再構築されます。

パッケージの依存関係

一定のパッケージは、共有ライブラリなどの他のパッケージに依存しています。言い換えれば、パッケージの中には、他のパッケージをrequireとしているものがあります。このようなパッケージは、必須パッケージがないとインストールできません。満たす必要のある依存関係(パッケージ要件)に加えて、特定のパッケージは、他のパッケージ

を recommend にします。これらの推奨されているパッケージは、実際に利用できる場合にのみインストールされます。利用できない場合は無視されますが、それらを推奨パッケージとしているパッケージはインストールできます。

8.2 インストール済みシステムの登録

インストール時に登録を飛ばした場合やシステムの再登録が必要な場合、いつでもシステム登録を行えます。その際には、YaSTモジュール [製品の登録] を使用するか、コマンドラインツール SUSEConnect を使用します。

8.2.1 YaSTでの登録

システムを登録するには、YaSTを起動してソフトウェアに切り替え、製品の登録を選択します。

デフォルトでは、SUSE Customer Centerにシステムを登録します。組織でローカル登録サーバが提供されている場合は、自動検出されたサーバのリストからいずれかのサーバを選択できます。または、手動でURLを指定してください。

8.2.2 SUSEConnectを使用した登録

コマンドラインから登録するには、次のコマンドを使用します。

```
> sudo SUSEConnect -r REGISTRATION_CODE -e EMAIL_ADDRESS
```

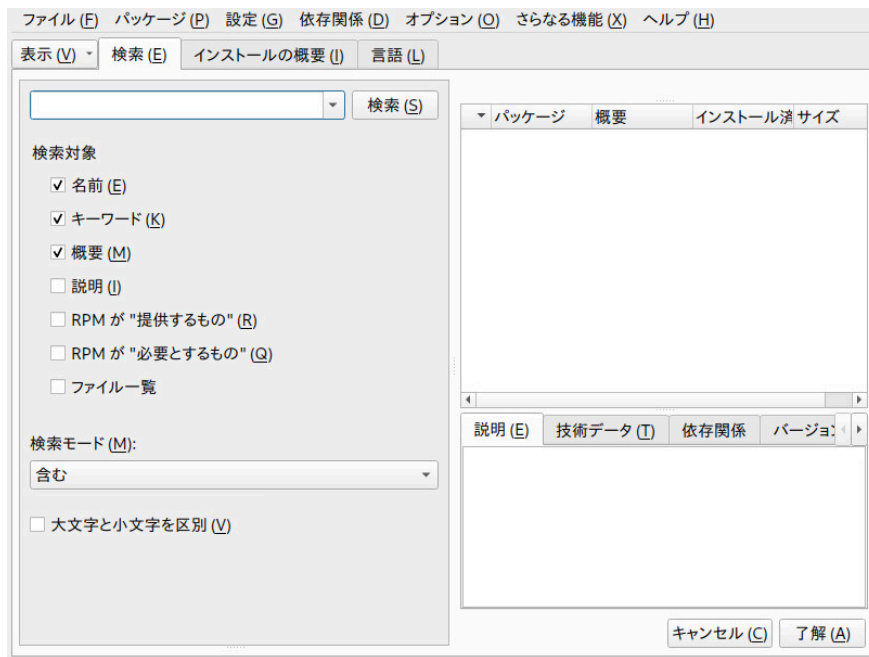
REGISTRATION_CODEは、SUSE Linux Enterprise Desktopと一緒に受け取った登録コードで置き換えます。EMAIL_ADDRESSは、各自または各自の組織が登録の管理に使用しているSUSEアカウントに関連付けられたEメールアドレスで置き換えます。

ローカル登録サーバで登録するには、次のようにサーバへのURLも入力します。

```
> sudo SUSEConnect -r REGISTRATION_CODE -e EMAIL_ADDRESS --url "URL"
```

8.3 YaSTソフトウェアマネージャの使用

ソフトウェア > ソフトウェアマネージャの順に選択して、YaSTコントロールセンターからソフトウェアマネージャを起動します。



8.3.1 ソフトウェアの検索

YaSTソフトウェアマネージャでは、現在有効になっているすべてのリポジトリからパッケージやパターンをインストールできます。このソフトウェアマネージャは、検索対象のソフトウェアの検出を容易にするさまざまな表示とフィルタを提供します。検索ビューは、ウィンドウのデフォルト表示です。ビューを変更するには、表示をクリックし、以下のいずれかのエントリをドロップダウンボックスで選択します。選択した表示が新しいタブで開きます。

パッケージまたはパターンの検索用ビュー

パターン

システム上のインストールに使用できるすべてのパターンを一覧します。

Package Groups (パッケージグループ)

グループ別にソートしたすべてのパッケージを一覧します(グラフィックス、プログラミング、セキュリティなど)。

言語

新しいシステム言語の追加に必要なすべてのパッケージを抽出するフィルタ。

リポジトリ

リポジトリ別にパッケージを抽出するフィルタ。複数のリポジトリを選択するには、**Ctrl** キーを押しながらリポジトリ名をクリックします。「擬似リポジトリ」**@System**を選択すると、現在インストールされているすべてのパッケージが一覧されます。

サービス

特定のモジュールまたは拡張機能に属するパッケージを表示します。エントリ(たとえば、**Basesystem**または**High Availability**)を選択して、このモジュールまたは拡張機能に属するパッケージのリストを表示します。

検索

特定の基準に従って、パッケージを検索できます。検索する用語を入力し、**Enter** を押します。次の項目内で検索する場所を指定し、検索モードを変更することにより、検索を絞り込みます。たとえば、パッケージ名は知らないが、検索するアプリケーションの名前だけは知っている場合は、検索プロセスにパッケージの説明を含めるようにします。

インストールの概要

インストール、更新、または削除するパッケージをすでに選択している場合は、このビューに、了解をクリックするとシステムに適用される変更が表示されます。このビューで特定の状態にあるパッケージをフィルタするには、各チェックボックスを選択または選択解除します。ステータスフラグの詳細を表示するには **Shift + F1** を押します。



ヒント: アクティブリポジトリに属さないパッケージの検出

アクティブリポジトリに属さないすべてのパッケージを一覧するには、表示 > リポジトリ > **@System**の順に選択し、次に、セカンダリフィルタ > メンテナンスされていないパッケージの順に選択します。たとえば、リポジトリを削除した後で、そのリポジトリから取得したパッケージがインストールされたまま残っていないことを確認する場合に、このオプションが役立ちます。



ヒント: オンラインでソフトウェアを検索する

オンライン検索機能によって、あらゆる登録済みおよび未登録のモジュールと拡張機能を対象として、パッケージを検索できます。

手順 8.1: オンラインでソフトウェアを検索する

オンラインでソフトウェアパッケージを検索するには、次の手順に従ってください。

1. オプション > オンライン検索から、オンライン検索ウィンドウを開きます。
2. パッケージ名を入力し、**Enter** キーを押すか、検索をクリックします。YaSTがSUSEカスタマーセンターに問い合わせし、テーブルに結果を表示します。このテーブルには各パッケージのモジュールまたは拡張機能が含まれます。詳細を確認するには、パッケージを選択します。
3. 対応するテーブル行と選択の切り替えをクリックして、インストールするパッケージを1つ以上選択します。または、テーブル行をダブルクリックすることもできます。パッケージが未登録のモジュールまたは拡張機能に属する場合は、YaSTが登録するかどうか確認します。
4. 次へをクリックして変更内容を確認し、パッケージをインストールします。

8.3.2 パッケージまたはパターンのインストールと削除

一定のパッケージは、共有ライブラリなどの他のパッケージに依存しています。いくつかのパッケージは、システム上で他のパッケージと共存できません。これらの依存関係や競合の解決が可能な場合は、YaSTによって自動的に解決されます。選択によって、自動的に解決できない依存関係の競合が発生した場合は、8.3.4項「[パッケージの依存関係](#)」の説明に従って、競合を手動で解決する必要があります。



注記: パッケージの削除

パッケージを削除する場合、デフォルトでは、選択したパッケージのみが削除されます。指定したパッケージの削除に伴って不要になる他のすべてのパッケージもYaSTで削除できるようにするには、メインメニューでオプション > パッケージの削除時にクリーンアップするの順に選択します。

1. パッケージを検索します(8.3.1項「[ソフトウェアの検索](#)」参照)。

2. 検出されたパッケージは、右側のペインに一覧されます。パッケージをインストールまたは削除するには、パッケージを右クリックして、インストールまたは削除を選択します。該当するオプションがない場合は、パッケージ名の先頭に表示された記号で示されているパッケージステータスを確認し、**Shift + F1** を押してヘルプを表示します。



ヒント: 一覧表示されたすべてのパッケージにアクションを適用する方法

右ペインに一覧表示されたすべてのパッケージにアクションを適用するには、メインメニューに移動し、**パッケージ > このリスト内のすべての順**に選択してアクションを選択します。

3. パターンをインストールするには、パターン名を右クリックして、インストールを選択します。
4. パターンを削除することはできません。代わりに、削除したいパターンのパッケージを選択し、それらを削除用にマークします。
5. さらにパッケージを選択するには、上記の手順を繰り返します。
6. 変更を適用する前に、**表示 > インストールの概要**の順にクリックすると、変更内容をレビューまたは変更できます。デフォルトでは、ステータスを変更するすべてのパッケージが一覧にされます。
7. パッケージの状態を元に戻すには、パッケージを右クリックし、次のエントリの1つを選択します。つまり、パッケージの削除または更新が予定されている場合は**保持**を選択し、パッケージのインストールが予定されている場合は**インストールしない**を選択します。すべての変更を破棄し、ソフトウェアマネージャを終了するには、**キャンセル**と**中止**をクリックします。
8. 完了したら、**受諾**をクリックして、変更を適用します。
9. YaSTが追加の依存関係を検出すると、インストール、更新または削除する関連パッケージのリストが表示されます。**続行**をクリックして、それらを受け入れます。選択されているすべてのパッケージのインストール、更新、または削除が完了すると、YaSTソフトウェアマネージャが自動的に終了します。



注記: ソースパッケージのインストール

現時点では、YaSTソフトウェアマネージャを使用してソースパッケージをインストールすることはできません。このためには、コマンドラインツール`zypper`を使用します。詳細については、[9.1.3.5項「ソースパッケージのインストールまたはダウンロード」](#)を参照してください。

8.3.3 パッケージの更新

個々のパッケージを更新する代わりに、インストールされているすべてのパッケージまたは特定リポジトリのすべてのパッケージを更新することもできます。パッケージの大量更新時には、一般に、次の側面を考慮します:

- パッケージを提供するリポジトリの優先順位、
- パッケージのアーキテクチャ(たとえば、AMD64/Intel 64)、
- パッケージのバージョン番号、
- パッケージのベンダ。

更新の候補を選択する上でどの側面が最も重要であるかは、選択する個々の更新オプションに依存します。

1. インストール済みのすべてのパッケージを最新バージョンに更新するには、メインメニューから、**パッケージ > すべてのパッケージ > 新しいバージョンがあれば更新するの順**に選択します。

一定のポリシーを使用して、使用できる更新候補がないかどうか、すべてのリポジトリがチェックされます。このポリシーでは、まずYaSTによる検索範囲を、インストール済みのパッケージと同じアーキテクチャおよびベンダのパッケージに限定します。検索でパッケージが見つかったら、以下のプロセスに従って、見つかったパッケージから「最良の」更新候補が選択されます。ただし、同じベンダの同類のパッケージが見つからない場合は、同じアーキテクチャのすべてのパッケージに検索が拡大されます。それでも同類のパッケージが見つからない場合は、すべてのパッケージが対象となり、次の基準に従って、「最良の」更新候補が選択されます。

1. リポジトリの優先度: 最高の優先度をもつリポジトリからのパッケージを優先します。
2. この基準で複数のパッケージが選択された場合は、アーキテクチャが「最良」であるパッケージが選択されます。(最良のアーキテクチャとは、インストール済みパッケージのアーキテクチャと同一のアーキテクチャです)。

選択したパッケージのバージョンがインストール済みパッケージのバージョン番号より高い場合は、インストール済みパッケージが選択した更新候補で更新および置換されます。

このオプションでは、インストール済みパッケージのアーキテクチャとベンダを変更しないようにしていますが、特定の条件下では、それらは許容されます。



注記: 強制的に更新する

代わりに、パッケージ > すべてのパッケージ > 強制的に更新するの順に選択すると、適用される基準は同じですが、検出された候補パッケージは無条件でインストールされます。したがって、このオプションを選択すると、特定のパッケージがダウングレードする場合があります。

2. 大量更新用パッケージを特定のリポジトリからのパッケージにするには:
 - a. 8.3.1項「ソフトウェアの検索」の説明に従って、更新に使用するリポジトリを選択します。
 - b. ウィンドウの右側で、システムパッケージをこのリポジトリ内のバージョンに切り替えるをクリックします。この設定は、パッケージを入れ替えるときにパッケージベンダを変更することをYaSTに対して明示的に許可します。
了解で続行すると、すべてのインストール済みパッケージが、このリポジトリ(使用可能な場合)からのパッケージで置換されます。これによって、ベンダとアーキテクチャが変更され、特定のパッケージがダウングレードすることがあります。
 - c. これを回避するには、システムパッケージをこのリポジトリ内のバージョンに切り替えることをキャンセルするをクリックします。このキャンセルは、了解ボタンをクリックする前にしかできません。
3. 変更を適用する前に、表示 > インストールの概要 の順にクリックすると、変更内容をレビューまたは変更できます。デフォルトでは、ステータスを変更するすべてのパッケージが一覧されます。

4. すべてのオプションを好みどおりに設定したら、受諾で変更内容を確認して大量更新を開始します。

8.3.4 パッケージの依存関係

ほとんどのパッケージは、他のパッケージに依存しています。たとえば、共有ライブラリを使用するパッケージは、そのライブラリを提供するパッケージに依存します。共存できないパッケージがあり、競合を引き起こします。たとえば、メール転送エージェント(sendmailまたはpostfix)は、1つしかインストールできません。ソフトウェアのインストールまたは削除時には、ソフトウェアマネージャが未解決のままの依存関係や競合が残っていないことを確認してシステムの整合性を確保します。

依存関係や競合の解決に1つのソリューションしかない場合は、その依存関係や競合は自動的に解決されます。複数のソリューションがあると必ず、手動で解決する必要のある競合が発生します。競合の解決にベンダやアーキテクチャの変更が必要な場合も、手動で解決する必要があります。受諾をクリックして、ソフトウェアマネージャで変更を適用すると、自動リゾルバでトリガされたすべてのアクションの概要が表示され、確認を要求されます。

依存関係は、デフォルトで、自動的にチェックされます。パッケージのステータスを変更するたびに(たとえば、パッケージをインストールまたは削除用にマークする)、チェックが実行されます。これは、一般的には便利ですが、依存関係の競合を手動で解決する際にはわずらわしくなることがあります。この機能を無効にするには、メインメニューに移動して依存関係 > 自動確認の選択を無効にします。依存関係の確認は、依存関係 > 今すぐ確認の順に選択して手動で実行します。整合性の確認は、受諾をクリックして選択を確定すると、必ず実行されます。

パッケージの依存関係をレビューするには、パッケージを右クリックし、解決器の情報表示を選択します。依存関係を示すマップが開きます。すでにインストールされているパッケージは、緑の枠内に表示されます。



注記: パッケージ競合の手動解決

競合の処理に精通していない限り、パッケージの競合を処理する場合は、YaSTによる指示に従うようにします。そうしないと、競合を解決できないことがあります。行った変更はいずれも他の競合をトリガする可能性があり、結局、競合の数は確実に増加することに留意してください。このようになった場合は、キャンセルでソフトウェアマネージャをキャンセルし、すべての変更を中止で破棄して、やり直します。



図 8.1: ソフトウェアマネージャの競合管理

8.3.5 推奨パッケージの取り扱い

パッケージには、プログラムの実行に必須の、強い依存関係(特定のライブラリなど)だけでなく、新しい機能や変換の追加など、弱い依存関係もあります。このような弱い依存関係を推奨パッケージと呼びます。

新しいパッケージをインストールする場合、推奨されるパッケージはデフォルトでインストールされます。既存のパッケージを更新する場合、不足している推奨事項は自動的にインストールされません。これを変更するには、`/etc/sysconfig/yast2`で`PKGMGR_RECOMMENDED="yes"`を設定します。インストール済みのパッケージに関する推奨パッケージが欠落している場合、それらすべてをインストールするには、YaST > Software Managerを開始し、エクストラ > Install All Matching Recommended Packages(該当する推奨パッケージをすべてインストールする)を選択します。

新規パッケージのインストール時の推奨パッケージのインストールを無効にするには、YaST Software Managerで、依存関係 > Install Recommended Packages(推奨パッケージのインストールを無効にします。Zypperコマンドラインツールを使用してパッケージをインストールしている場合、`--no-recommends`オプションを使用します。

8.4 ソフトウェアリポジトリおよびサービスの管理

サードパーティソフトウェアをインストールするには、ソフトウェアリポジトリをシステムに追加します。デフォルトでは、システムを登録すると、SUSE Linux Enterprise Desktop-DVD 15 SP5や一致するアップデートリポジトリなどの製品リポジトリが自動的に設定されます。登録の詳細については、『Deployment Guide』、第4章「Installation steps」、4.6項「Registration」または『Upgrade Guide』、第4章「Upgrading offline」、4.7項「Registering your system」を参照してください。最初を選択した製品によっては、変換、辞書などを含んだ追加リポジトリも設定できます。

リポジトリを管理するには、YaSTを起動し、ソフトウェア>ソフトウェアリポジトリの順に選択します。設定されたソフトウェアリポジトリダイアログが開きます。ここで、ダイアログの右隅にある表示をすべてのサービスに変更することによって、サービスの購読を管理することもできます。このコンテキストではサービスは、1つまたは複数のソフトウェアを提供できるRepository Index Service (RIS) です。この種のサービスは管理者またはベンダから動的に変更できます。

各リポジトリは、リポジトリコンテンツ(パッケージ名、バージョンなど)を説明したファイルを提供します。YaSTは、これらのリポジトリ説明ファイルをローカルキャッシュにダウンロードします。ソフトウェアリポジトリには、その整合性確認のため、リポジトリメンテナのGPGキーで署名することができます。新しいリポジトリを追加するたびに、YaSTでリポジトリのキーをインポートできます。



警告: 外部ソフトウェアソースの信用

外部ソフトウェアのリポジトリをリポジトリリストに追加する場合は、その前に、リポジトリを信頼できるかどうか確認してください。SUSEは、サードパーティのソフトウェアリポジトリからインストールされたソフトウェアによって発生するどのような問題についても、責任を負いません。

8.4.1 ソフトウェアリポジトリの追加

DVD/CD、USBフラッシュドライブ、ローカルディレクトリ、ISOイメージ、またはネットワークソースからリポジトリを追加できます。

YaSTの設定されたソフトウェアリポジトリダイアログでリポジトリを追加するには、次の手順に従います。

1. Add (追加)をクリックします。
2. ダイアログに表示されているオプションのいずれかを選択します。



図 8.2: ソフトウェアリポジトリの追加

- ネットワークをスキャンして、SLP経由でサービスをアナウンスしているインストールサーバを検索するには、SLPを使用したスキャンを選択して次へをクリックします。
- リムーバブルメディアからリポジトリを追加するには、該当するオプションを選択して、メディアを挿入するか、またはUSBデバイスをコンピュータに接続します。次へをクリックして、インストールを開始します。
- 大半のリポジトリでは、該当のオプションを選択して次へをクリックすると、メディアへのパス(またはURL)を指定するように求められます。Repository Name (リポジトリ名)の指定は任意です。何も指定しない場合は、製品名またはURLがリポジトリ名として使用されます。

リポジトリの説明をダウンロードはデフォルトで有効になっています。このオプションを有効にしていない場合は、後で必要になったときにYaSTによって自動的にダウンロードされます。

- 追加したリポジトリによっては、リポジトリのGPGキーのインポートを求められたり、ライセンスへの合意を求められたりします。
確認すると、メタデータがダウンロードされ、解析されます。これにより、Configured Repositories (設定されたリポジトリ)のリストにリポジトリが追加されます。
- 必要に応じて、[8.4.2項「リポジトリプロパティの管理」](#)の説明に従い、リポジトリのプロパティを調整します。

5. OKをクリックして変更内容を確認し、設定ダイアログを閉じます。
6. リポジトリを正常に追加できると、ソフトウェアマネージャが起動し、そのリポジトリからパッケージをインストールできるようになります。詳細については、[第8章「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。

8.4.2 リポジトリプロパティの管理

ソフトウェアリポジトリの設定されたソフトウェアリポジトリの概要では、次のリポジトリプロパティを変更できます。

Status

リポジトリのステータスは、有効または無効のどちらかです。有効なリポジトリからのパッケージだけをインストールできます。一時的にリポジトリを無効にするにはEnable (有効化)を無効にします。リポジトリ名をダブルクリックして、その状態を切り替えることもできます。リポジトリを完全に削除するには、削除をクリックします。

更新

リポジトリを更新すると、そのコンテンツの説明(パッケージ名、バージョンなど)は、YaSTで使用するローカルキャッシュにダウンロードされます。これは、CDやDVDなどの静的リポジトリでは1回で十分ですが、内容が頻繁に変更されるリポジトリでは頻繁な更新が必要です。リポジトリのキャッシュを最新の状態に保つ最も簡単な方法は、自動的に更新するの選択です。手動更新を行うには、更新をクリックして、オプションの1つを選択します。

ダウンロードしたパッケージを維持します

インストールの前に、リモートリポジトリからのパッケージをダウンロードします。これらのパッケージは、デフォルトでは、インストールが正常に完了すると削除されます。ダウンロードしたパッケージを維持しますを有効にすると、ダウンロードしたパッケージが削除されません。ダウンロードの場所は、`/etc/zypp/zypp.conf`に設定されます。これは、デフォルトでは、`/var/cache/zypp/packages`です。

優先度

リポジトリの優先度は、1~200の値です。ここで、1は最高の優先度、200は最低の優先度です。YaSTで追加した新しいリポジトリの優先度は、デフォルトで99です。特定のリポジトリに関して優先度値が何であってもよい場合は、値を0に設定しても、そのリポジトリにデフォルト優先度(99)を適用できます。パッケージが2つ以上のリポジトリにある場合は、優先度の最も高いリポジトリが優先して使用されます。これは、ローカルリポジトリ(たとえば、DVD)に高い優先度を与えることによって、インターネットから不必要にパッケージをダウンロードしないようにする場合に有用です。

❗ 重要: 優先度とバージョンの比較

優先度の最も高いリポジトリが、常に、優先されます。したがって、更新リポジトリには必ず最高の優先度が割り当てられるようにします。そのようにしないと、次のオンラインアップデートまで更新されない古いバージョンがインストールされる可能性があります。

名前とURL

リポジトリ名またはリポジトリのURLを変更するには、それをシングルクリックでリストから選択し、次に、**編集**をクリックします。

8.4.3 リポジトリキーの管理

ソフトウェアリポジトリには、その整合性確認のため、リポジトリメンテナのGPGキーで署名することができます。新しいリポジトリを追加するたびに、そのキーをYaSTでインポートできます。そのキーを他の任意のGPGキーのように検証し、キーが変更されていないことを確認してください。キーの変更を見つけた場合は、リポジトリに何らかの問題がある可能性があります。キーの変更の原因を突き止めるまで、リポジトリをインストールソースとして無効にしてください。

インポートしたすべてのキーを管理するには、設定されたソフトウェアリポジトリダイアログでGPG鍵をクリックします。マウスでエントリを選択して、ウィンドウ下部にキーのプロパティを表示します。追加、編集、または削除をクリックすることで、該当する操作をキーに対して実行します。

8.5 GNOMEパッケージアップデート

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パッチおよびアップデートを提供します。デスクトップで利用可能なツールを使用して、またはYaSTオンラインアップデートモジュールを実行することにより、インストールできます。このセクションでは、パッケージアップデートを使用してGNOMEデスクトップからシステムをアップデートする方法について説明します。

YaSTオンラインアップデートモジュールとは異なり、GNOMEパッケージアップデートは、アップデートリポジトリからパッチのインストールを提供するだけでなく、すでにインストールされているパッケージの新しいバージョンも提供します(パッチ修正のセキュリティの

問題または誤動作、機能およびバージョン番号は通常、変わりません。新しいバージョンのパッケージによりバージョン番号が増え、通常は機能が追加されるか主な変更が導入されます)。

新しいパッチまたはパッケージのアップデートが利用可能な場合は常に、GNOMEでは、通知領域またはロック画面で通知が表示されます。

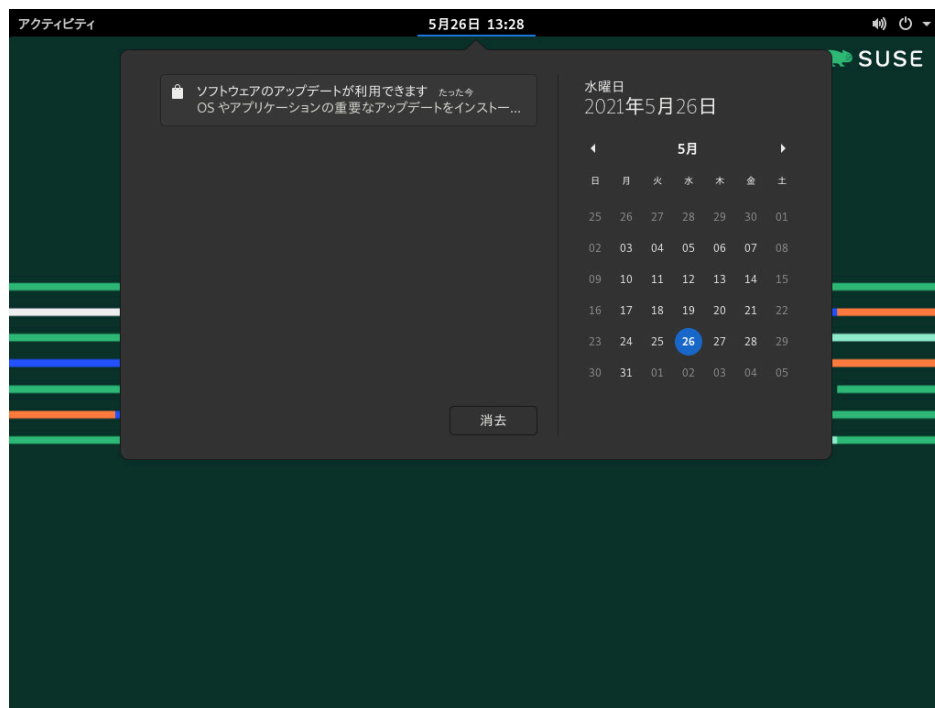
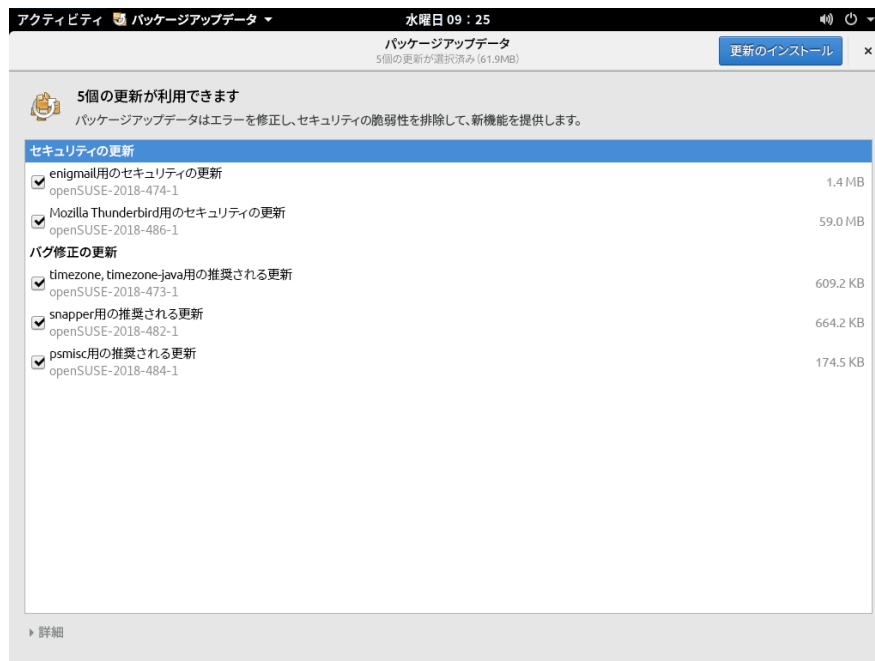


図 8.3: GNOMEのデスクトップに表示されたアップデート通知

パッケージアップデートの通知設定を行うには、GNOMEで設定を起動し、通知 > パッケージアップデートの順に選択します。

手順 8.2: GNOMEパッケージアップデートを使用したパッチおよびアップデートのインストール

1. パッチおよびアップデートをインストールするには、通知メッセージをクリックします。これにより、GNOMEでパッケージアップデートが開きます。または、`package U`と入力し、パッケージアップデートを選択して、アクティビティからアップデートを開きます。



2. アップデートは次の4つのカテゴリに分類されます。

セキュリティアップデート(パッチ)

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。

推奨されるアップデート(パッチ)

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。これらのアップデートをインストールすることを強くお勧めします。

オプションのアップデート(パッチ)

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。

その他のアップデート

インストールされるパッケージの新しいバージョン。

すべての使用可能なアップデートはインストール用に事前に選択されています。すべてのアップデートをインストールしない場合は、まず、不要なアップデートを選択解除します。すべてのセキュリティアップデートおよび推奨されるアップデートは必ずインストールすることを強くお勧めします。

アップデートに関する詳細情報を取得するには、そのタイトルと詳細をクリックします。情報はパッケージリストの下ボックスに表示されます。

3. 更新をインストールするをクリックしてインストールを開始します。

4. 一部のアップデートでは、マシンを再起動するか、ログアウトする必要がある場合があります。インストール後に表示される、手順に関するメッセージを確認します。

8.6 GNOMEソフトウェアを使用したパッケージの更新

GNOMEパッケージアップデートに加えて、GNOMEでは次の機能を持つGNOMEソフトウェアを提供します。

- PackageKitを介してRPMとして配布されたソフトウェアをインストール、更新、削除する
- Flatpakとして配布されたソフトウェアをインストール、更新、削除する
- GNOMEシェル拡張機能(<https://extensions.gnome.org>)をインストール、更新、削除する
- 「Linux Vendor Firmware Service」(LVFS、<https://fwupd.org>)を使用してハードウェアデバイスのファームウェアを更新する

このほか、GNOMEソフトウェアでは、ソフトウェアのスクリーンショット、レーティング、およびレビューを提供します。

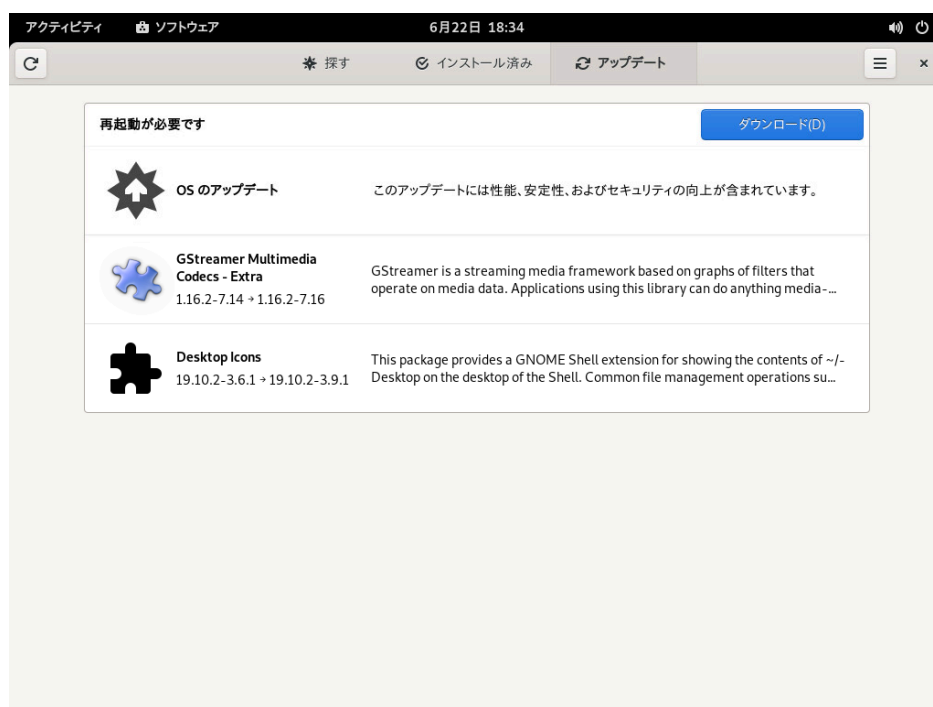


図 8.4: GNOMEソフトウェア—更新ビュー

GNOMEソフトウェアには、SUSE Linux Enterprise Desktopで提供される他のツールと次のような違いがあります。

- YaSTまたはZypperとは異なり、RPMとしてパッケージされたソフトウェアをインストールする場合、GNOMEソフトウェアはAppStreamメタデータを提供するソフトウェアに制限されます。これには、ほとんどのデスクトップアプリケーションが含まれます。
- GNOMEパッケージアップデートは実行中のシステム内のパッケージをアップデートし(それぞれのアプリケーションを再起動する必要があります)、GNOMEソフトウェアはアップデートをダウンロードし、再起動後に適用します。

9 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用語(repository、patch、updateなど)の定義については、[8.1項「用語の定義」](#)を参照してください。

9.1 Zypperの使用

Zypperは、パッケージのインストール、更新、および削除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。リポジトリの管理も行います。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。

9.1.1 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のコマンド

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、次のコマンドを使用します。

```
> sudo zypper patch
```

グローバルオプション

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。

```
> sudo zypper --non-interactive patch
```

上の例では、オプション--non-interactiveは、確認を一切表示せずにコマンドを実行することを意味します(自動的にデフォルトの回答を適用します)。

コマンド固有のオプション

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。

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

上の例では、`--auto-agree-with-licenses`を使用して、ライセンスの確認を表示せずに、必要なすべてのパッチをシステムに適用します。その代わりに、自動的にライセンスに同意します。

引数

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、コマンド `install` を使用する場合、「インストール」するパッケージを1つまたは複数指定する必要があります。

```
> sudo zypper install mplayer
```

一部のオプションでは、1つの引数が必要です。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 `mc` と `vim` パッケージをインストールします。

```
> sudo zypper -v install --from factory mc vim
```

`--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。`--repo` は、`--from` のエイリアスで、いずれかのものを使用できます。

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。

```
> sudo 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。

```
> sudo zypper --userdata STRING patch
```


9.1.2 Zypperサブコマンドの使用

Zypperサブコマンドは、`zypper_execdir`設定オプションで指定されたディレクトリに保存される実行可能ファイルです。デフォルトでは`/usr/lib/zypper/commands`です。サブコマンドがそこで見つからない場合、Zypperはその`$PATH`の残りの部分を自動的に検索します。これにより、独自のローカル拡張機能を作成し、ユーザ空間に保存できます。

Zypperシェルでのサブコマンドの実行、グローバルZypperオプションの使用はサポートされていません。

使用可能なサブコマンドの一覧表示:

```
> zypper help subcommand
[...]
Available zypper subcommands in '/usr/lib/zypper/commands'

  appstream-cache
  lifecycle
  migration
  search-packages

Zypper subcommands available from elsewhere on your $PATH

  log                Zypper logfile reader
                     (/usr/sbin/zypper-log)
```

サブコマンドのヘルプ画面の表示

```
> zypper help appstream-cache
```

9.1.3 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。

```
> sudo zypper install PACKAGE_NAME
> sudo zypper remove PACKAGE_NAME
```



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`などの必須システムパッケージは削除しないでください。これらを削除すると、システムが不安定になったり、まったく動作しなくなったりする可能性があります。

9.1.3.1 インストールまたは削除するパッケージの選択

コマンド `zypper install` および `zypper remove` でパッケージを指定するには、さまざまな方法があります。

正確なパッケージ名を指定する

```
> sudo zypper install MozillaFirefox
```

正確なパッケージ名およびバージョン番号を指定する

```
> sudo zypper install MozillaFirefox-52.2
```

リポジトリエイリアスおよびパッケージ名を指定する

```
> sudo 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

名前が特定の文字列で始まるか、特定の文字列で終わるパッケージをすべて選択できます。特にパッケージを削除する場合は、ワイルドカードの使用には注意が必要です。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。

```
> sudo zypper install 'Moz*'
```



ヒント: すべての `-debuginfo` パッケージを削除

問題をデバッグする際、実行中のプロセスに関する情報を多く得るために、一時的に多数の `-debuginfo` パッケージをインストールする場合があります。デバッグセッションが終了したら、この環境を消去する必要があります。それには以下を実行します。

```
> sudo zypper remove '*-debuginfo'
```

機能によって指定する

たとえば、名前がわからないパッケージをインストールする場合は、機能による指定が便利です。次のコマンドは、パッケージ `MozillaFirefox` をインストールします。

```
> sudo zypper install firefox
```

機能、ハードウェアアーキテクチャ、またはバージョンによって指定する

機能とともに、ハードウェアアーキテクチャとバージョンを指定できます。

- 機能の後にピリオドを付けて、その後に目的のハードウェアアーキテクチャの名前を追加します。たとえば、AMD/Intel 64アーキテクチャ(Zypperでの名前はx86_64_64)を指定するには、次のコマンドを使用します。

```
> sudo zypper install 'firefox.x86_64'
```

- バージョンは文字列の最後に追加し、バージョンの前に演算子を付ける必要があります。使用できる演算子は、<(より小さい)、<=(以下)、=(等しい)、>=(以上)、>(より大きい)です。

```
> sudo zypper install 'firefox>=74.2'
```

- 必要なハードウェアアーキテクチャとバージョンを組み合わせることもできます。

```
> sudo zypper install 'firefox.x86_64>=74.2'
```

RPMファイルのパスによって指定する

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。

```
> sudo zypper install /tmp/install/MozillaFirefox.rpm
> sudo zypper install http://download.example.com/MozillaFirefox.rpm
```

9.1.3.2 パッケージのインストールと削除の結合

パッケージのインストールと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。

```
> sudo 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。

```
> sudo 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。

```
> sudo zypper install -emacs +vim      # Wrong
```

```
> sudo zypper install vim -emacs      # Correct
> sudo zypper install -- -emacs +vim  # Correct
> sudo zypper remove emacs +vim      # Correct
```

9.1.3.3 削除されたパッケージの依存関係のクリーンアップ

指定したパッケージの削除後に、不要になったパッケージも自動的に削除されるようにしたい場合は、`--clean-deps`オプションを使用します。

```
> sudo zypper rm --clean-deps PACKAGE_NAME
```

9.1.3.4 スクリプトでのZypperの使用

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。

```
> sudo 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。

9.1.3.5 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には、次のコマンドを使用します。

```
> zypper source-install PACKAGE_NAME
```

ソースパッケージをインストールするデフォルトの場所は、`root`として実行する場合は`/usr/src/packages/`、ユーザとして実行する場合は`~/rpmbuild`になります。これらの値はローカルの`rpm`設定で変更できます。

このコマンドにより、指定したパッケージのビルド依存関係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。

```
> sudo zypper source-install -D PACKAGE_NAME
```

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。

```
> sudo zypper source-install -d PACKAGE_NAME
```

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、[9.1.6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。

9.1.3.6 無効にされたリポジトリからのパッケージのインストール

通常、パッケージのインストールや更新は、有効化されたリポジトリからしかできません。`--plus-content TAG`オプションを使用すると、リポジトリをリフレッシュし、現在のZypperセッション中のみ一時的に有効にして、終了したら無効にすることができます。

たとえば、追加の`-debuginfo`パッケージまたは`-debugsource`パッケージを提供するリポジトリを有効にするには、`--plus-content debug`を使用します。このオプションは複数回指定できます。

そうした「デバッグ」リポジトリを一時的に有効にして、特定の`-debuginfo`パッケージをインストールするには、次のオプションを使用します。

```
> sudo zypper --plus-content debug \
install "debuginfo(build-id)=eb844a5c20c70a59fc693cd1061f851fb7d046f4"
```

`debuginfo`パッケージがないと、`build-id`文字列が、`gdb`によって報告されます。



注記: 無効なインストールメディア

SUSE Linux Enterprise Desktopインストールメディアのリポジトリは引き続き設定されていますが、インストールが正常に完了すると無効になります。`--plus-content`オプションを使用して、オンラインリポジトリの代わりにインストールメディアからパッケージをインストールできます。**zypper**を呼び出す前に、DVDをコンピュータのドライブに挿入するなどして、メディアが使用可能であることを確認してください。

9.1.3.7 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。

```
> sudo zypper install-new-recommends
```

このコマンドは、Webカメラまたは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。

9.1.4 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 Desktopのアップグレードについては、『Upgrade Guide』、第2章「Upgrade paths and methods」を参照してください。

9.1.4.1 必要なすべてのパッチのインストール

SUSE Linux Enterpriseへの「パッチ適用」は、インストールされたパッケージの新しいバージョンをインストールする最も信頼できる方法です。これにより、正しいバージョンを持つ必要なすべてのパッケージがインストールされ、「競合している」とみなされるパッケージバージョンが確実に省略されます。

システムに適用される、正式にリリースされたすべての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。

```
> sudo zypper patch
```

コンピュータに設定されているリポジトリから使用可能なすべてのパッチが、インストール環境に関係があるかどうかを確認されます。関係がある場合(および`optional`または`feature`として分類されていない場合)、パッチはただちにインストールされます。**zypper patch**が成功すると、例外を確認しない限り、脆弱なバージョンパッケージがインストールされないことが保証されます。正式な更新リポジトリはSUSE Linux Enterprise Desktopのインストールを登録した後でのみ使用可能であることに注意してください。

インストールするパッチにシステムの再起動が必要な変更が含まれる場合、事前に警告が表示されます。

プレーンの**zypper patch**コマンドでは、サードパーティリポジトリからのパッチは適用されません。サードパーティリポジトリも更新するには、次のように`with-update`コマンドオプションを使用します。

```
> sudo zypper patch --with-update
```

オプションのパッチもインストールするには、次のコマンドを使用します。

```
> sudo zypper patch --with-optional
```

Bugzillaの特定の問題に関連するすべての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。

```
> sudo zypper patch --bugzilla=NUMBER
```

特定のCVEデータベースエントリに関連するすべての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。

```
> sudo zypper patch --cve=NUMBER
```

たとえば、番号がCVE-2010-2713 CVE-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。

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


Zypperおよびパッケージ管理自体に影響するパッチのみをインストールするには、次のコマンドを使用します。

```
> sudo zypper patch --updatestack-only
```

`updatestack-only`コマンドオプションを使用する場合、ほかのリポジトリも更新しようとしてそれ以外のコマンドオプションを指定すると、そのコマンドオプションは削除されます。

9.1.4.2 パッチのリストの表示

パッチが使用可能かどうかを確認するため、Zypperでは次の情報を参照できます。

必要なパッチの数

必要なパッチ(システムに適用されるパッチであってもまだインストールされていないもの)の数のリストを表示するには、`patch-check`を使用します。

```
> zypper patch-check
Loading repository data...
Reading installed packages...
5 patches needed (1 security patch)
```

このコマンドを`--updatestack-only`オプションと組み合わせて使用すると、Zypperおよびパッケージ管理自体に影響するパッチのみのリストを表示できます。

必要なパッチのリスト

必要なすべてのパッチ(システムに適用されるパッチであってもまだインストールされていないもの)のリストを表示するには、`zypper list-patches`を使用します。

すべてのパッチのリスト

インストール済みかどうか、およびインストール環境に適用されるかどうかに関係なく、SUSE Linux Enterprise Desktopで使用可能なすべてのパッチのリストを表示するには、`zypper patches`を使用します。

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の問題によって指定する

Bugzillaの問題に関連する、必要なすべてのパッチのリストを表示するには、オプション`--bugzilla`を使用します。

特定のバグに対応するパッチのリストを表示するには、`--bugzilla=NUMBER`のようにバグ番号を指定することもできます。Bugzillaの複数の問題に関連するパッチを検索するには、次の例のように、バグ番号の間にカンマを追加します。


```
> zypper list-patches --bugzilla=972197,956917
```

CVE番号によって指定する

CVE (Common Vulnerabilities and Exposures)データベースのエントリに関連する、必要なすべてのパッチのリストを表示するには、オプション `--cve` を使用します。

特定のCVEデータベースエントリに対応するパッチのリストを表示するには、`--cve=NUMBER` のようにCVE番号を指定することもできます。複数のCVEデータベースエントリに関連するパッチを検索するには、次の例のように、CVE番号の間にカンマを追加します。

```
> zypper list-patches --cve=CVE-2016-2315,CVE-2016-2324
```

撤回されたパッチを一覧表示する

SUSE Linux Enterprise 15のコードストリームでは、一部のパッチが自動的に撤回されます。アップデートに新しいバグが含まれるリスクがあるため、保守アップデートは慎重にテストされます。アップデートにバグが含まれていることが判明した場合、バグのあるアップデートを元に戻すために新しいアップデート(バージョン番号が大きい)が発行され、バグのあるアップデートは再度インストールされないようにブロックされます。撤回されたパッチを `zypper` で一覧表示できます。

```
> zypper lp --all |grep retracted
SLE-Module-Basesystem15-SP3-Updates | SUSE-SLE-Module-Basesystem-15-SP3-2021-1965
| recommended | important | --- | retracted | Recommended update for multipath-
tools
SLE-Module-Basesystem15-SP3-Updates | SUSE-SLE-Module-Basesystem-15-SP3-2021-2689
| security | important | --- | retracted | Security update for cpio
SLE-Module-Basesystem15-SP3-Updates | SUSE-SLE-Module-Basesystem-15-SP3-2021-3655
| security | important | reboot | retracted | Security update for the Linux
Kernel
```

撤回された(または任意の)パッチに関する詳細情報を参照してください。

```
> zypper patch-info SUSE-SLE-Product-SLES-15-2021-2689
Loading repository data...
Reading installed packages...

Information for patch SUSE-SLE-Product-SLES-15-2021-2689:
-----
Repository   : SLE-Product-SLES15-LTSS-Updates
Name          : SUSE-SLE-Product-SLES-15-2021-2689
Version       : 1
Arch          : noarch
Vendor        : maint-coord@suse.de
Status        : retracted
Category      : security
Severity      : important
```

```

Created On   : Mon 16 Aug 2021 03:44:00 AM PDT
Interactive  : ---
Summary     : Security update for cpio
Description :
    This update for cpio fixes the following issues:

    It was possible to trigger Remote code execution due to a integer overflow
    (CVE-2021-38185, bsc#1189206)

    UPDATE:
    This update was buggy and could lead to hangs, so it has been retracted.
    There will be a follow up update.
    [...]

```

競合するパッケージのパッチ

```

Information for patch openSUSE-SLE-15.3-2022-333:
-----
Repository  : Update repository with updates from SUSE Linux Enterprise 15
Name        : openSUSE-SLE-15.3-2022-333
Version     : 1
Arch       : noarch
Vendor     : maint-coord@suse.de
Status      : needed
Category    : security
Severity    : important
Created On  : Fri Feb  4 09:30:32 2022
Interactive  : reboot
Summary     : Security update for xen
Description :
    This update for xen fixes the following issues:

    - CVE-2022-23033: Fixed guest_physmap_remove_page not removing the p2m mappings.
    (XSA-393) (bsc#1194576)
    - CVE-2022-23034: Fixed possible DoS by a PV guest Xen while unmapping a grant.
    (XSA-394) (bsc#1194581)
    - CVE-2022-23035: Fixed insufficient cleanup of passed-through device IRQs.
    (XSA-395) (bsc#1194588)
Provides    : patch:openSUSE-SLE-15.3-2022-333 = 1
Conflicts   : [22]
    xen.src < 4.14.3_06-150300.3.18.2
    xen.noarch < 4.14.3_06-150300.3.18.2
    xen.x86_64 < 4.14.3_06-150300.3.18.2
    xen-devel.x86_64 < 4.14.3_06-150300.3.18.2
    xen-devel.noarch < 4.14.3_06-150300.3.18.2
    [...]

```

上記のパッチは、影響を受けるバージョンまたは脆弱なバージョンの22パッケージと競合します。これらの影響を受けるパッケージまたは脆弱なパッケージのいずれかがインストールされると、競合が発生し、パッチは「必要」に応じて分類されます。zypper

patchは、利用可能なすべてのパッチをインストールしようとします。問題が発生した場合は、問題が報告され、すべてのアップデートがインストールされるわけではないことが通知されます。この競合は、影響を受けるパッケージまたは脆弱なパッケージを更新するか、それらを削除することで解決できます。SUSEアップデートリポジトリには固定パッケージも付属しているため、アップデートは競合を解決するための標準的な方法です。依存関係の問題やパッケージのロックなどにより、パッケージを更新できない場合は、ユーザの承認後に削除されます。

必要かどうかに関係なくすべてのパッチのリストを表示するには、追加でオプション**--all**を使用します。たとえば、CVE番号が割り当てられたすべてのパッチのリストを表示するには、次のコマンドを使用します。

```
> zypper list-patches --all --cve
Issue | No. | Patch | Category | Severity | Status
-----+-----+-----+-----+-----+-----
cve | CVE-2019-0287 | SUSE-SLE-Module.. | recommended | moderate | needed
cve | CVE-2019-3566 | SUSE-SLE-SERVER.. | recommended | moderate | not needed
[...]
```

9.1.4.3 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新しく入手可能なバージョンで更新するには、次のコマンドを使用します。

```
> sudo zypper update
```

！ 重要

zypper updateは問題のあるパッケージを無視します。たとえば、パッケージがロックされている場合、**zypper update**は、より新しいバージョンのパッケージが利用可能であってもそのパッケージを省略します。逆に、パッケージが脆弱であるとみなされた場合、**zypper patch**は競合を報告します。

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。

```
> sudo zypper update PACKAGE_NAME
> sudo 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。

```
> sudo 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。

9.1.4.4 孤立パッケージの特定

Zypperからリポジトリを削除する場合や、システムをアップグレードする場合には、いくつかのパッケージが「孤立」状態になる可能性があります。これらの「孤立」パッケージは、どのアクティブなリポジトリにも属していません。次のコマンドで、これらのリストを表示できます。

```
> sudo zypper packages --orphaned
```

このリストを使用して、パッケージが引き続き必要か、それとも削除しても安全かを判断できます。

9.1.5 削除されたファイルを使用しているプロセスとサービスの特定

パッケージにパッチを適用したり、パッケージを更新または削除したりした場合、更新または削除によって削除されたファイルを引き続き使用している実行中のプロセスがシステムに存在することがあります。削除されたファイルを使用しているプロセスのリストを表示するには、**zypper ps**を使用します。プロセスが既知のサービスに属している場合は、サービス名のリストが表示され、そのサービスを容易に再起動できます。デフォルトでは、**zypper ps**は次のような表を表示します。

```
> zypper ps
```

PID	PPID	UID	User	Command	Service	Files
814	1	481	avahi	avahi-daemon	avahi-daemon	/lib64/ld-2.19.s-> /lib64/libdl-2.1-> /lib64/libpthrea-> /lib64/libc-2.19->
[...]						

PID: プロセスのID

PPID: 親プロセスのID

UID: プロセスを実行しているユーザのID

UID: プロセスを実行しているユーザのログイン名

Command: プロセスを実行するために使用されるコマンド

Service: サービス名(コマンドがシステムサービスに関連付けられている場合にのみ)

Files: 削除されたファイルのリスト

次のように指定することで、**zypper ps**の出力フォーマットを制御できます。

zypper ps -s

削除されたファイルを表示しない短い表を作成します。

```
> zypper ps -s
PID | PPID | UID | User | Command | Service
-----+-----+-----+-----+-----+-----
814 | 1 | 481 | avahi | avahi-daemon | avahi-daemon
817 | 1 | 0 | root | irqbalance | irqbalance
1567 | 1 | 0 | root | sshd | sshd
1761 | 1 | 0 | root | master | postfix
1764 | 1761 | 51 | postfix | pickup | postfix
1765 | 1761 | 51 | postfix | qmgr | postfix
2031 | 2027 | 1000 | tux | bash |
```

zypper ps -ss

システムサービスに関連付けられているプロセスのみを表示します。

PID	PPID	UID	User	Command	Service
814	1	481	avahi	avahi-daemon	avahi-daemon
817	1	0	root	irqbalance	irqbalance
1567	1	0	root	sshd	sshd
1761	1	0	root	master	postfix
1764	1761	51	postfix	pickup	postfix
1765	1761	51	postfix	qmgr	postfix

zypper ps -sss

削除されたファイルを使用しているシステムサービスのみを表示します。

```
avahi-daemon
```

```
irqbalance
postfix
sshd
```

zypper ps --print "systemctl status %s"

再起動が必要な可能性があるサービスのステータス情報を取得するコマンドを表示します。

```
systemctl status avahi-daemon
systemctl status irqbalance
systemctl status postfix
systemctl status sshd
```

サービスの処理の詳細については、[第19章「systemdデーモン」](#)を参照してください。

9.1.6 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 9.1: ZYPPER - 既知のリポジトリのリスト

```
> zypper repos
# | Alias          | Name          | Enabled | Refresh
---+-----+-----+-----+-----
1 | SLEHA-15-GE0   | SLEHA-15-GE0 | Yes     | No
2 | SLEHA-15       | SLEHA-15     | Yes     | No
3 | SLES15         | SLES15       | 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
```

9.1.6.1 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。

```
> sudo 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、https://en.opensuse.org/openSUSE:Libzypp_URIsを参照してください)。ALIAS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。

9.1.6.2 リポジトリの更新

zypperは、設定されているリポジトリからパッケージの変更点をフェッチできます。変更点をフェッチするには、次のコマンドを実行します。

```
> sudo zypper refresh
```



注記: **zypper**のデフォルトの動作

一部のコマンドではデフォルトで**refresh**が自動的に実行されるため、ユーザがこのコマンドを明示的に実行する必要はありません。

refreshコマンドと**--plus-content**オプションを使用すると、無効になっているリポジトリ内の変更点も表示できます。

```
> sudo zypper --plus-content refresh
```

このオプションは、リポジトリ内の変更点をフェッチしますが、無効になっているリポジトリは同じ状態(無効)のままにします。

9.1.6.3 リポジトリの削除

リストからリポジトリを削除するには、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、SLEHA-12-GE0から例9.1「**Zypper - 既知のリポジトリのリスト**」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。

```
> sudo zypper removerepo 1
> sudo zypper removerepo "SLEHA-12-GE0"
```

9.1.6.4 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。

```
> sudo 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でなく、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 TYPE: 特定のタイプのリポジトリ(ここで、TYPE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。

```
> sudo zypper renamerepo 'Mozilla Firefox' firefox
```

9.1.7 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。特定のパッケージに関する情報を取得するには、infoコマンドを使用します。

9.1.7.1 ソフトウェアの検索

zypper searchコマンドは、パッケージ名に対して機能し、オプションでパッケージの概要と説明に対しても機能します。/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]/"
```

9.1.7.2 すべてのSLEモジュール間のパッケージの検索

現在有効なSLEモジュールの内部または外部の両方のパッケージを検索するには、**search-packages**サブコマンドを使用します。このコマンドは、SUSEカスタマーセンターに連絡し、次のように一致するパッケージのすべてのモジュールを検索します。

```
> zypper search-packages package1 package2
```

zypper search-packagesでは、次のオプションを提供します。

- 検索文字列の完全一致を検索: -x, --match-exact
- モジュール別に結果をグループ化(デフォルト: パッケージ別にグループ化): -g, --group-by-module
- パッケージに関する詳細を表示: -d, --details
- 検索結果をXMLで表示: --xmlout

9.1.7.3 特定の機能の検索

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)'
```

`what-provides PACKAGE_NAME`は`rpm -q --whatprovides PACKAGE_NAME`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。

9.1.7.4 パッケージ情報の表示

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。パッケージ名がリポジトリのどのパッケージ名にも一致しない場合は、パッケージ以外に一致するものの詳細情報を出力します。特定のタイプを要求して(`-t`オプションを使用)、そのタイプが存在しない場合は、使用可能なほかの一致を出力しますが、詳細な情報は出力しません。

ソースパッケージを指定した場合、そのソースパッケージからビルドされたバイナリパッケージを表示します。バイナリパッケージを指定した場合、そのバイナリパッケージをビルドするために使用されたソースパッケージを出力します。

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
```

9.1.8 ライフサイクル情報の表示

SUSE製品は一般的に10年間サポートされています。多くの場合、3年間のサポートを追加するSUSEの拡張サポート提供を利用して、この標準的なライフサイクルを延長することができます。製品によっては、正確なサポートライフサイクルを<https://www.suse.com/lifecycle>で見つけることができます。

製品とサポートされているパッケージのライフサイクルを確認するには、以下のよう**zypper lifecycle**コマンドを使用します。

```
# zypper lifecycle
Product end of support
Codestream: SUSE Linux Enterprise Server 15          2028-07-31
Product: SUSE Linux Enterprise Server 15 SP3         n/a*

Module end of support
Basesystem Module          n/a*
Desktop Applications Module n/a*
```

Package end of support if different from product:

autofs

Now, installed 5.1.3-7.3.1, update available

5.1.3-7.6.1

9.1.9 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。

9.1.10 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、特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でそのパッケージを見つけれない場合など)は、次のコマンドでリポジトリを更新すると有効ことがあります。

```
> sudo 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。

```
> sudo 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。

9.1.11 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10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。

9.1.12 詳細情報

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。詳しいコマンドリファレンス、最も重要なコマンドのcheat sheets、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、https://en.opensuse.org/SDB:Zypper_usageを参照してください。SUSE Linux Enterprise Desktop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、https://en.opensuse.org/openSUSE:Zypper_versionsを参照してください。

9.2 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。

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、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**-devel**で識別できます(**alsa-devel**パッケージ、**gimp-devel**パッケージなど)。

9.2.1 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。

オペレーティングシステムの問題を修復する場合、暫定修正(PTF)を実動システムにインストールしなければならない場合があります。SUSEから提供されるパッケージは、特別なPTFキーに照らして署名されています。ただし、SUSE Linux Enterprise 11と異なり、SUSE Linux Enterprise 12システムでは、このキーはデフォルトでインポートされません。キーを手動でインポートするには、次のコマンドを使用します。

```
> sudo rpm --import \  
/usr/share/doc/packages/suse-build-key/suse_ptf_key.asc
```

キーをインポートしたら、PTFパッケージをシステムにインストールできます。

9.2.2 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`」のように入れます。このコマンドで、パッケージをインストールできます。ただし、依存関係が満たされており、他のパッケージとの競合がない場合に限られます。`rpm`では、依存関係の要件を満たすためにインストールしなければならないパッケージがエラーメッセージで要求されます。バックグラウンドで、RPMデータベースは競合が起きないようにします。ある特定のファイルは、1つのパッケージだけにしか属せません。別のオプションを選択すると、`rpm`にこれらのデフォルト値を無視させることができますが、この処置を行うのは専門知識のある人に限られます。それ以外の人が行うと、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、以下の戦略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。
- アップデート前にシステム管理者が環境設定ファイルを変更した場合、rpmは拡張子.rpmorigまたは.rpmsave(バックアップファイル)で変更されたファイルを保存し、新しいパッケージからバージョンをインストールします。これは、最初にインストールされたファイルと新しいバージョンが異なる場合にのみ実行されます。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、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。」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。

9.2.3 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、そのまゝの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。

makedeltarpm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。

```
> sudo 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。

```
> sudo 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。

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

テクニカル詳細については、</usr/share/doc/packages/deltarpm/README>を参照してください。

9.2.4 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。表9.1「基本的なRPMクエリオプション」を参照してください。

表 9.1: 基本的なRPMクエリオプション

-i	パッケージ情報
-----------	---------

<u>-l</u>	ファイルリスト
<u>-f FILE</u>	ファイル <code>FILE</code> を含むパッケージでクエリーを行います(<code>FILE</code> にはフルパスを指定する必要があります)。
<u>-s</u>	ステータス情報を含むファイルリスト(<u>-l</u> を暗示指定)
<u>-d</u>	ドキュメントファイルだけをリストします (<u>-l</u> を暗示指定)。
<u>-c</u>	設定ファイルだけをリストします(<u>-l</u> を暗示指定)。
<u>--dump</u>	詳細情報を含むファイルリスト(<u>-l</u> 、 <u>-c</u> 、または <u>-d</u> と共に使用します)
<u>--provides</u>	他のパッケージが <u>--requires</u> で要求できるパッケージの機能をリストします。
<u>--requires</u> 、 <u>-R</u>	パッケージが要求する機能
<u>--scripts</u>	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9.2 「`rpm -q -i wget`」 に示された情報を表示します。

例 9.2: `rpm -q -i wget`

```

Name       : wget
Version    : 1.14
Release    : 17.1
Architecture: x86_64
Install Date: Mon 30 Jan 2017 14:01:29 CET
Group      : Productivity/Networking/Web/Utilities
Size       : 2046483
License    : GPL-3.0+
Signature  : RSA/SHA256, Thu 08 Dec 2016 07:48:44 CET, Key ID 70af9e8139db7c82
Source RPM : wget-1.14-17.1.src.rpm
Build Date : Thu 08 Dec 2016 07:48:34 CET
Build Host : sheep09
Relocations : (not relocatable)
Packager   : https://www.suse.com/

```



```
Vendor      : SUSE LLC <https://www.suse.com/>
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.
Distribution: SUSE Linux Enterprise 15
```

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。例:

```
> rpm -q -f /bin/rpm /usr/bin/wget
rpm-4.14.1-lp151.13.10.x86_64
wget-1.19.5-lp151.4.1.x86_64
```

ファイル名の一部分しかわからない場合は、例9.3「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 9.3: 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 PACKAGE` 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用すると、`rpm` には、インストールした後で変更されたパッケージ内のすべてのファイルが表示されます。`rpm` では、8文字の記号を使用して、次の変更に関するいくつかのヒントを提供します。

表 9.2: RPM 確認オプション

<u>5</u>	MD5チェックサム
<u>S</u>	ファイルサイズ
<u>L</u>	シンボリックリンク
<u>T</u>	変更時間

<u>D</u>	メジャーデバイス番号とマイナーデバイス番号
<u>U</u>	所有者
<u>G</u>	グループ
<u>M</u>	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_RPMDDB_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、1 GBの/usrに対しておよそ1 MBです。

9.2.5 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

元のソース(.tar.bz2ファイルや.tar.gzファイルなど)用、およびディストリビューション固有の調整(ほとんどの場合.diffまたは.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を使ってソースパッケージをインストールすると、必要なすべてのコンポーネントが/usr/src/packagesにインストールされます。ソースと調整はSOURCES、関連する.specファイルはSPECSに格納されます。



警告: システムの整合性

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-1.19.5.tar.bz2
/usr/src/packages/SOURCES/wgetrc.patch
/usr/src/packages/SPECS/wget.spec
```

rpmbuild -bX /usr/src/packages/SPECS/wget.spec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。

specファイルのBuildRootディレクティブは非推奨です。この機能がまだ必要な場合は、回避方法として--buildrootオプションを使用してください。

9.2.6 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ー独自の処理をするようにするには、ビルド環境の初期化を省略するか、**rpmrpm**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**build**のマニュアルページで、詳細な情報が得られます。

9.2.7 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、[第8章「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。

10 Snapperを使用したシステムの回復とスナップショット管理

Snapperにより、ファイルシステムスナップショットを作成および管理できます。ファイルシステムスナップショットは特定の時点でのファイルシステムの状態のコピーを保持できます。Snapperの標準セットアップは、システムの変更のロールバックを許可するように設計されています。ただし、これを使用して、ユーザーデータのオンディスクバックアップを作成することもできます。この機能のベースとして、SnapperはBtrfsファイルシステム、またはXFSあるいはExt4ファイルシステムでシンプロビジョニングされたLVMボリュームを使用します。

SnapperにはコマンドラインインタフェースとYaSTインタフェースがあります。Snapperでは、次のタイプのファイルシステムでファイルシステムスナップショットを作成および管理できます。

- Btrfs。サブボリュームのファイルシステムスナップショットをネイティブにサポートするLinux用のコピーオンライトファイルシステム。(サブボリュームは物理パーティション内で別個にマウント可能なファイルシステムです。)
Btrfsスナップショットから起動することもできます。詳細については、[10.3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。
- XFSまたはExt4でフォーマットされたシンプロビジョニングLVMボリューム。

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[10.2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[10.2.2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[10.3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。
- 実行中のシステム内で、スナップショットを手動で作成および管理する。詳細については[10.6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。

10.1 デフォルト設定

SUSE Linux Enterprise Desktop上のSnapperは、システム変更の「取り消しおよび回復ツール」として設定されています。デフォルトでは、SUSE Linux Enterprise Desktop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16 GB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、スナップショットは/以外のパーティションでは無効になっています。



ヒント: インストール済みシステムでのSnapperの有効化

インストール中にSnapperを無効にした場合、後からいつでも有効にできます。そのためには、次のコマンドを実行して、ルートファイルシステムのデフォルトのSnapper設定を作成します。

```
> sudo snapper -c root create-config /
```

その後、10.1.4.1項「スナップショットの有効化/無効化」の説明に従って、別のスナップショットタイプを有効にします。

Btrfsルートファイルシステムでは、スナップショットには、インストーラが提案するように設定されたサブボリュームと、少なくとも16GBのパーティションサイズを持つファイルシステムが必要です。

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。

結果として、スナップショットを含むパーティションは、スナップショットを含まない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大きく異なります。経験則として、パーティションには通常の2倍の容量を割り当てます。ディスク容量が不足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クリーンアップされます。詳細については、[10.1.4.4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。

10.1.1 デフォルト設定

16 GBより大きいディスク

- 環境設定ファイル: [/etc/snapper/configs/root](#)
- [USE_SNAPPER=yes](#)
- [TIMELINE_CREATE=no](#)

16 GB未満のディスク

- 設定ファイル: 作成されない
- [USE_SNAPPER=no](#)
- [TIMELINE_CREATE=yes](#)

10.1.2 スナップショットのタイプ

スナップショット自体は技術的な意味では同じですが、トリガするイベントに基づいて、次の3種類のスナップショットを区別しています。

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。YaST OSのインストール方法(デフォルト)を使用すると、ルートファイルシステムを除いてタイムラインスナップショットが有効になります。

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。

10.1.3 スナップショットから除外されるディレクトリ

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。

/boot/grub2/i386-pc、/boot/grub2/x86_64-efi、/boot/grub2/powerpc-ieee1275、/boot/grub2/s390x-emu

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。これらのディレクトリは、アーキテクチャ固有です。最初の2つのディレクトリはAMD64/Intel 64マシン上に存在し、その後の2つのディレクトリはそれぞれIBM POWERとIBM Z上に存在します。

/home

/homeが独立したパーティションに存在していない場合、ロールバック時のデータ損失を避けるために除外されます。

/opt

サードパーティ製品は通常、/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

スナップショットから除外される一時ファイルとキャッシュを含むすべてのディレクトリ。

/usr/local

このディレクトリは、ソフトウェアの手動インストール時に使用します。ロールバック時にこれらのインストール済みソフトウェア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。

/var

このディレクトリには、ログ、一時キャッシュ、/var/optのサードパーティ製品など、多くのバリアブルファイルが含まれており、仮想マシンのイメージとデータベースのデフォルトの場所です。したがって、このサブボリュームはスナップショットからすべてのこのバリアブルデータを除外するように作成され、コピーオンライトが無効になっています。

10.1.4 設定のカスタマイズ

SUSE Linux Enterprise Desktopには、適切なデフォルト設定が用意されていて、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。

10.1.4.1 スナップショットの有効化/無効化

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"`

YaST OSのインストール方法(デフォルト)を使用すると、ルートファイルシステムを除いてタイムラインスナップショットが有効になります。

インストールスナップショットの有効化/無効化

有効化: パッケージ `snapper-zypp-plugin`

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`に設定します。

無効化. `/etc/sysconfig/yast2`で`USE_SNAPPER`を`no`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。

10.1.4.2 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な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なマークが付きます。
- 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)。重要なマークが付いた`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なマークが付きます(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>

```

10.1.4.3 新規サブボリュームの作成とマウント

/階層下に新規のサブボリュームを作成し、永続的にマウントすることができます。このようなサブボリュームはスナップショットから除外されます。既存のスナップショット内に作成してはいけません。ロールバック後にスナップショットを削除できなくなるためです。

SUSE Linux Enterprise Desktopは、/@サブボリュームが設定されており、このサブボリュームは、/opt、/srv、/home、その他の永続サブボリュームの独立したルートとしての役割を果たします。作成し、永続的にマウントする新規のサブボリュームは、この初期のルートファイルシステムに作成しなければなりません。

そのように設定するには、次のコマンドを実行します。この例では、新規サブボリューム、/usr/importantは/dev/sda2から作成されます。

```

> sudo mount /dev/sda2 -o subvol=@ /mnt
> sudo btrfs subvolume create /mnt/usr/important
> sudo umount /mnt

```

/etc/fstabの該当エントリは、次のようになります。

```
/dev/sda2 /usr/important btrfs subvol=@/usr/important 0 0
```



ヒント: コピーオンライト(cow)の無効化

サブボリュームには、仮想化ディスクイメージ、データベースファイル、ログファイルなど、頻繁に変更されるファイルが含まれている場合があります。その場合、ディスクブロックの重複を避けるため、このボリュームのコピーオンライト機能を無効にすることを検討します。そのためには、/etc/fstabでnodatacowマウントオプションを使用します。

```
/dev/sda2 /usr/important btrfs nodatacow,subvol=@/usr/important 0 0
```

1つのファイルまたはディレクトリに対してコピーオンライトを無効にするには、コマンド **chattr +C PATH**を使用します。

10.1.4.4 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。ディスク容量が不足してシステムが停止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、最大10個の重要なインストールスナップショットと管理スナップショット、および最大10個の標準のインストールスナップショットと管理スナップショットが保持されます。これらのスナップショットがルートファイルシステムのサイズの50%超を占有する場合、追加のスナップショットは削除されます。最低でも、4つの重要なスナップショットと2つの標準スナップショットは常に保持されます。

これらの値の変更方法については、[10.5.1項「既存の設定の管理」](#)を参照してください。

10.1.4.5 シンプロビジョニングされた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ニング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、『Deployment Guide』、第6章「Expert Partitioner」、6.2項「LVM configuration」を参照してください。

シンプロビジョニングLVMボリュームでSnapperを使用するには、そのようにSnapperの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`、`ext4`または`xfs`は、`FILESYSTEM`の有効な値です。例:

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

[10.5.1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。

10.2 Snapperを使用した変更の取り消し

SUSE Linux Enterprise Desktop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、[10.1.4.1項「スナップショットの有効化/無効化」](#)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使用できます。

！ 重要: 変更の取り消しとロールバックの比較

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

10.3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、10.3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`などの特定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。

10.2.1 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のモジュールまたは**snappersnapper**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 10.1: 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。

スナップショット

現在の設定 root

ID	種類	開始日	終了日	説明	ユーザーデータ
1	単一	2022-06-22 04:43:49		first root filesystem	
2	単一	2022-06-22 04:55:48		after installation	important=yes
3 & 4	事前および事後	2022-06-22 05:00:07	2022-06-22 05:00:20	yast repositories	
6 & 7	事前および事後	2022-06-22 05:00:41	2022-06-22 05:00:43	zypp(ruby.ruby2.5)	important=no
5 & 8	事前および事後	2022-06-22 05:00:24	2022-06-22 05:00:45	yast sw_single	
10 & 11	事前および事後	2022-06-22 05:01:02	2022-06-22 05:01:04	zypp(ruby.ruby2.5)	important=no
9 & 12	事前および事後	2022-06-22 05:00:48	2022-06-22 05:01:08	yast sw_single	
13 & 14	事前および事後	2022-06-22 05:04:07	2022-06-22 05:05:55	yast lan	
15 & 16	事前および事後	2022-06-22 05:06:00	2022-06-22 05:28:25	yast ntp-client	
17 & 18	事前および事後	2022-06-22 05:28:30	2022-06-22 05:37:59	yast lan	
19 & 20	事前および事後	2022-06-22 05:38:06	2022-06-22 05:47:30	yast ntp-client	
21	事前	2022-06-22 05:47:51		yast snapper	

変更点の表示 作成 (I) 修正 (M) 削除 (T) 更新 (R)

ヘルプ (H) 閉じる (L)

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内容に差異のあるファイルのリストが表示されます。



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。

ファイルを復元中です

これらのファイルはスナップショット「33」から復元されます:

/var/lib/samba/private/msg.sock/9228
/var/lib/samba/private/msg.sock/9239
/var/lib/samba/usershares

元のスナップショットに存在するファイルは、現在のシステムにコピーされます。

スナップショットに存在しなかったファイルは削除されます。

よろしいですか?

いいえ(N)

はい(Y)

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 10.2: **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 MODULE_NAME**」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。

```
> sudo snapper list -t pre-post
```

Pre #	Post #	Pre Date	Post Date	Description
311	312	Tue 06 May 2018 14:05:46 CEST	Tue 06 May 2018 14:05:52 CEST	zypp(y2base)
340	341	Wed 07 May 2018 16:15:10 CEST	Wed 07 May 2018 16:15:16 CEST	zypp(zypper)
342	343	Wed 07 May 2018 16:20:38 CEST	Wed 07 May 2018 16:20:42 CEST	zypp(y2base)
344	345	Wed 07 May 2018 16:21:23 CEST	Wed 07 May 2018 16:21:24 CEST	zypp(zypper)
346	347	Wed 07 May 2018 16:41:06 CEST	Wed 07 May 2018 16:41:10 CEST	zypp(y2base)
348	349	Wed 07 May 2018 16:44:50 CEST	Wed 07 May 2018 16:44:53 CEST	zypp(y2base)
350	351	Wed 07 May 2018 16:46:27 CEST	Wed 07 May 2018 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**をクリックします。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。

```
> sudo 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/Shalheader
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 FILENAME。 FILENAME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。

```
> sudo 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 FILENAMES。 FILENAMES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。

```
> sudo 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/Shalheader
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。

10.2.2 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、特定の時点でどのような変更が加えられたのかを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、[10.5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の比較

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[10.3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 10.3: 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から、または「`yast2 snapper`」と入力してSnapperモジュールを起動します。
2. スナップショットを選択するための現在の設定を選択します。
3. ファイルを復元するための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 10.4: `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。

```
> sudo 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。

```
> sudo 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。

```
> sudo 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

```
> sudo 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。

10.3 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 Desktop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。デフォルトのSnapper設定(root)で作成されたスナップショットのみがブート可能です。

！ 重要: サポートされる設定

SUSE Linux Enterprise Desktop 15 SP5の時点では、システムのロールバックは、ルートパーティションのデフォルトのサブボリューム設定が変更されていない場合にのみサポートされます。

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の比較

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

10.2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、[10.1.3項「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書き込むことはできないので注意してください。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、ロールバックを実行したくない状態になっている場合は、再起動して現在のシステム状態にブートします。その後、別のスナップショットを選択するか、レスキューシステムを開始することができます。
- b. ロールバックを実行するには、次のコマンドを実行し

```
> 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。ロールバック前のファイルシステムの状態のスナップショットが作成されます。rootのデフォルトのサブボリュームは、新しい読み書きスナップショットに置き換えられます。詳細については、[10.3.1項「ロールバック後のスナップショット」](#)を参照してください。

-dオプションを使用してスナップショットの説明を追加すると役に立ちます。例:

```
New file system root since rollback on DATE TIME
```



ヒント: 特定のインストール状態へのロールバック

スナップショットがインストール時に無効になっていない場合、最初のシステムインストールの最後に初期のブート可能スナップショットが作成されます。このスナップショットをブートすることで、いつでもその状態に戻ることができます。スナップショットは、after installationという記述で識別できます。

ブート可能スナップショットは、サービスパックや新しいメジャーリリースへのシステムアップグレードの開始時にも作成されます(スナップショットが無効になっていない場合のみ)。

10.3.1 ロールバック後のスナップショット

ロールバックの実行前に、動作中のファイルシステムのスナップショットが作成されます。この説明では、ロールバックで復元されたスナップショットのIDを参照します。

ロールバックで作成されたスナップショットは、Cleanup属性に値numberが付きます。したがって、設定されているスナップショット数に達すると、ロールバックスナップショットは自動的に削除されます。詳細については、[10.7項「スナップショットの自動クリーンアップ」](#)を参照してください。スナップショットに重要なデータが含まれている場合は、スナップショットが削除される前にデータを抽出してください。

10.3.1.1 ロールバックスナップショットの例

たとえば、新規インストール後に、システムで次のスナップショットが使用可能であるとして

```
# snapper --iso list
Type | # | | Cleanup | Description | Userdata
-----+---+---+-----+-----+-----
single | 0 | | | current | 
single | 1 | | | first root filesystem | 
single | 2 | | number | after installation | important=yes
```

sudo snapper rollbackを実行すると、スナップショット3が作成され、ロールバック実行前のシステムの状態が格納されます。スナップショット4は新しいデフォルトのBtrfsサブボリュームであるため、再起動後にシステムになります。

```
# snapper --iso list
Type | # | | Cleanup | Description | Userdata
-----+---+---+-----+-----+-----
single | 0 | | | current | 
single | 1 | | number | first root filesystem | 
single | 2 | | number | after installation | important=yes
single | 3 | | number | rollback backup of #1 | important=yes
single | 4 | | | |
```

10.3.2 スナップショットブートエントリのアクセスと識別

スナップショットからブートするには、マシンを再起動して、Start Bootloader from a read-only snapshot(読み取り専用スナップショットからBootloaderを始動)を選択します。ブート可能なすべてのスナップショットをリストした画面が開きます。最も新しいスナップショットが先頭に表示され、最も古いものは最後に表示されます。↓ キーおよび↑ キーを使用して移動し、Enter キーを押して、選択したスナップショットを有効にします。ブートメニューからスナップショットを有効にしても、マシンは即座に再起動されません。選択したスナップショットのブートローダが開くだけです。



```
SLES15-SP4 (5.14.21-150400.19,2022-05-03T11:53,post,ya
SLES15-SP4 (5.14.21-150400.19,2022-05-03T11:50,pre,ya
SLES15-SP4 (5.14.21-150400.19,2022-05-03T11:50,post,zy
SLES15-SP4 (5.14.21-150400.19,2022-05-03T11:50,pre,zy
SLES15-SP4 (5.14.21-150400.19,2022-05-03T11:49,post,ya
```

図 10.1: ブートローダ: スナップショット



警告: 現在、UEFIを使用してBtrfsスナップショットからXenをブートすると失敗します

詳細については、<https://www.suse.com/support/kb/doc/?id=000020602> を参照してください。

ブートローダの各スナップショットエントリの名前は、命名規則に従っているため、容易に識別できます。

[*] ① OS ② (KERNEL ③ , DATE ④ TIME ⑤ , DESCRIPTION ⑥)

- ① スナップショットにimportantとマークが付いている場合、そのエントリには*が付きます。
- ② オペレーティングシステムラベル。
- ④ 日付フォーマット(YYYY-MM-DD)。
- ⑤ 時刻フォーマット(HH:MM)。
- ⑥ このフィールドには、スナップショットの説明が入ります。手動で作成されたスナップショットの場合、この説明は--descriptionオプションで作成された文字列、またはカスタム文字列です(ヒント: [ブートローダスナップショットエ](#)

ントリのカスタム説明の設定を参照)。自動で作成されたスナップショットの場合、`zypp(zypper)`や`yast_sw_single`など、呼びだされたツールです。長い説明は、ブート画面のサイズに応じて、切り捨てて表示されます。



ヒント: ブートローダスナップショットエントリのカスタム説明の設定

スナップショットの説明フィールドのデフォルトの文字列をカスタム文字列に置き換えることができます。この機能は、自動的に作成された説明が不十分な場合や、ユーザが入力した説明が長すぎる場合などに役立ちます。カスタム文字列、`STRING`をスナップショット、`NUMBER`に設定するには、次のコマンドを使用します。

```
> sudo snapper modify --userdata "bootloader=STRING" NUMBER
```

説明は25文字未満にしてください。このサイズを超える部分はブート画面では一切読めません。

10.3.3 制限

システム全体をスナップショット作成時と同一の状態に復元する、システムの「完全な」ロールバックは不可能です。

10.3.3.1 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、[10.1.3項「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。

ユーザデータ量

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。

10.3.3.2 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、`/boot`のロールバックを実行する際には保証できません。

10.4 ユーザホームディレクトリでのSnapperの有効化

いくつかのユースケースをサポートする、ユーザの`/home`ディレクトリのスナップショットを有効にできます。

- 個々のユーザは独自のスナップショットおよびロールバックを管理できます。
- システムユーザ、たとえば、設定ファイル、ドキュメントなどのコピーを追跡したいデータベース、システム、およびネットワーク管理者。
- SambaはホームディレクトリおよびBtrfsバックエンドと共有します。

各ユーザのディレクトリは/homeのBtrfsサブボリュームです。これを手動で設定できます(10.4.3項「ホームディレクトリでのスナップショットの手動有効化」を参照)。ただし、`pam_snapper`を使用する方がより便利です。`pam_snapper`パッケージでは、`pam_snapper.so`モジュールおよびヘルパースクリプトがインストールされ、ユーザの作成およびSnapper設定が自動化されます。

`pam_snapper`では、`useradd`コマンド、プラグ可能認証モジュール(PAM)、およびSnapperとの統合が提供されます。デフォルトでは、ユーザログイン時およびログアウト時にスナップショットが作成され、特定のユーザは延長された期間にログインしたままであるため、タイムベースのスナップショットも作成されます。通常のSnapperコマンドと設定ファイルを使用して、デフォルトを変更できます。

10.4.1 `pam_snapper`のインストールとユーザの作成

最も簡単な方法は、Btrfsでフォーマットされた新しい/homeディレクトリで既存のユーザなしで開始する方法です。両方のノードに `pam_snapper`:

```
# zypper in pam_snapper
```

`/etc/pam.d/common-session`に次の行を追加します。

```
session optional pam_snapper.so
```

`/usr/lib/pam_snapper/pam_snapper_useradd.sh`スクリプトを使用して、新しいユーザとホームディレクトリを作成します。デフォルトで、スクリプトはドライランを実行します。スクリプトを編集し、`DRYRUN=1`を`DRYRUN=0`に変更します。これで、新しいユーザを作成できます。

```
# /usr/lib/pam_snapper/pam_snapper_useradd.sh \
username group passwd=password
Create subvolume '/home/username'
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
```

`/etc/skel`からのファイルは最初のログイン時にユーザのホームディレクトリにコピーされます。ユーザの設定がSnapper設定を一覧表示して作成されていることを確認します。

```
# snapper list --all
```

```
Config: home_username, subvolume: /home/username
Type   | # | Pre # | Date | User | Cleanup | Description | Userdata
-----+---+-----+-----+-----+-----+-----+-----
single | 0 |       |      | root |         | current     |
```

時間が経過するにつれて、この出力にはスナップショットのリストが取り込まれ、ユーザは標準のSnapperコマンドを管理できます。

10.4.2 ユーザを削除する

`/usr/lib/pam_snapper/pam_snapper_userdel.sh`スクリプトを使用してユーザを削除します。デフォルトでは、ドライランを実行し、それを編集して、`DRYRUN=1`を`DRYRUN=0`に変更します。ユーザ、ユーザのホームサブボリューム、Snapper設定が削除され、すべてのスナップショットが削除されます。

```
# /usr/lib/pam_snapper/pam_snapper_userdel.sh username
```

10.4.3 ホームディレクトリでのスナップショットの手動有効化

Snapperを使用してユーザのホームディレクトリを手動で設定するためのステップがあります。`/home`は、Btrfsでフォーマットされる必要があり、ユーザはまだ作成されていません。

```
# btrfs subvol create /home/username
# snapper -c home_username create-config /home/username
# sed -i -e "s/ALLOW_USERS=\"\"/ALLOW_USERS=\"username\"/g" \
/etc/snapper/configs/home_username
# yast users add username=username home=/home/username password=password
# chown username.group /home/username
# chmod 755 /home/username/.snapshots
```

10.5 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。

ルートファイルシステムに十分な容量(約12GB)がある場合、ルートファイルシステム(`/`)のスナップショットはインストール時に自動的に有効になります。対応するデフォルト設定は`root`という名前です。これにより、YaSTおよびZypperのスナップショットが作成および管理されます。デフォルト値のリストについては、[10.5.1.1項「環境設定データ」](#)を参照してください。



注記: スナップショットを有効にするための最小ルートファイルシステム

10.1項「デフォルト設定」で説明されているように、スナップショットを有効にするには、ルートファイルシステムに追加の空き容量が必要です。この容量は、インストールされているパッケージの量と、スナップショットに含まれるボリュームに加えられた変更の量によって異なります。スナップショットの頻度と、アーカイブされるスナップショットの数も重要です。

インストール時にスナップショットを自動的に有効にするには、最小サイズのルートファイルシステムが必要です。現在、このサイズは約12GBです。この値は今後、基本システムのアーキテクチャとサイズに応じて変わる可能性があります。これは、インストールメディアにあるファイル/control.xmlの次のタグの値に依存します。

```
<root_base_size>  
<btrfs_increase_percentage>
```

これは、 $\text{ROOT_BASE_SIZE} * (1 + \text{BTRFS_INCREASE_PERCENTAGE}/100)$ という式で計算されます。

この値は最小サイズであることに注意してください。ルートファイルシステム用に、これよりも多くの容量を使用することを検討します。一般的には、スナップショットが有効でない場合に使用するサイズの2倍にします。

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。

```
> sudo snapper -c www-data ❶ create-config /srv/www ❷
```

- 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、[10.5.1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。

```
> sudo snapper -c www-data create-config -t MY_DEFAULTS /srv/www
```

10.5.1 既存の設定の管理

snapperコマンドは、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。

設定の一覧表示

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

```
> sudo snapper list-configs
Config | Subvolume
-----+-----
root   | /
usr    | /usr
local  | /local
```

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`を**snapper list-configs**で示される設定名のいずれかに置き換えます。環境設定オプションの詳細については、[10.5.1.1項「環境設定データ」](#)を参照してください。

デフォルトの設定を表示するには、次のコマンドを実行します。

```
> sudo snapper -c root get-config
```


設定の変更

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`を`snapper list-configs`で示される
設定名のいずれかに置き換えます。`OPTION`および`VALUE`に指定可能な値は、[10.5.1.1項](#)
「環境設定データ」に一覧にされています。

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`を`snapper list-configs`で示される設定名のいずれかに置き換えます。

10.5.1.1 環境設定データ

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
に、各オプションの詳細を示します。値を変更するには、`snapper -c CONFIG set-config`
"`KEY=VALUE`"を実行します。

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、[10.5.1.2項](#)「通常ユーザとしてのSnapperの使用」を参照してください。
デフォルトの設定は""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。
デフォルトの設定は"`yes`"です。

EMPTY_*

同一の事前および事後スナップショットを持つスナップショットペアのクリーンアップ
アルゴリズムを定義します。詳細については[10.7.3項](#)「違いがないスナップショットの
ペアのクリーンアップ」を参照してください。

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。
デフォルトの設定は"`btrfs`"です。

NUMBER_*

インストールおよび管理スナップショットのクリーンアップアルゴリズムを定義しま
す。詳細については[10.7.1項](#)「番号付きスナップショットのクリーンアップ」を参照し
てください。

QGROUP / SPACE_LIMIT

クリーンアップアルゴリズムにクォータサポートを追加します。詳細については[10.7.5 項「ディスククォータサポートの追加」](#)を参照してください。

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。

デフォルトの設定は"/"です。

SYNC_ACL

Snapperが通常ユーザによって使用される場合([10.5.1.2 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。デフォルトの設定は"no"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。有効値:

yes, no。

デフォルトの設定は"no"です。

TIMELINE_CLEANUP / TIMELINE_LIMIT_*

タイムラインスナップショットのクリーンアップアルゴリズムを定義します。詳細については[10.7.2 項「タイムラインスナップショットのクリーンアップ」](#)を参照してください。

10.5.1.2 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

このような目的のために、ユーザまたは/およびグループに許可を付与する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 10.5: 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. Snapper設定がまだ存在しない場合は、ユーザがSnapperを使用できるようにする必要のあるパーティションまたはサブボリュームに対してSnapper設定を作成します。手順については、10.5項「Snapper設定の作成と変更」を参照してください。例:

```
> sudo 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)。必要に応じて調整します。詳細については、10.5.1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。

```
> sudo 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listコマンドを使ってテストできます。

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

10.6 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10.5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には、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`**)。

10.6.1 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特定の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。既存のスナップショットとそのメタデータを表示するには、**`snapper list`**を使用します。

`snapper --config home list`

設定 `home` のスナップショットの一覧を示します。デフォルト設定(`root`)のスナップショットの一覧が示されるようにするには、**`snapper -c root list`**または**`snapper list`**を使用します。

`snapper list -a`

すべての既存設定の一覧を示します。

`snapper list -t pre-post`

デフォルト(`root`)設定の事前および事後スナップショットの全ペアの一覧を示します。

`snapper list -t single`

デフォルト(`root`)設定について、`single`タイプの全スナップショットの一覧を示します。

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10.6.1.1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な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10.7項「スナップショットの自動クリーンアップ」を参照してください。

10.6.1.1 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

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スナップショットはpostスナップショットに対応しています。たとえば、これはYaST/Zypperの自動スナップショットに使用されます。

post

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スナップショットはpreスナップショットに対応しています。たとえば、これはYaST/Zypperの自動スナップショットに使用されます。

single

スタンドアロンのスナップショットです。たとえば、これは1時間ごとの自動スナップショットに使用されます。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。

10.6.1.2 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。保持するさまざまなタイプのスナップショットの数を、Snapper設定で定義することができます(詳細は10.5.1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古いスナップショットは削除しますが、毎時、毎日、毎月、および毎年のスナップショットを複数保持します。

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。

10.6.2 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モジュールSnapperの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。Snapper用のYaSTインタフェースここでは明示的に説明されていませんが、等価な機能を提供しています。



ヒント: スナップショットの説明

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。オプション `--userdata` を介して追加情報を指定することもできます。

snapper create --from 17 --description "with package2"

既存のスナップショットからスタンドアロンスナップショット(シングルタイプ)を作成します。これは、**snapper list**からのスナップショットの番号で指定されます。(これはSnapperバージョン0.8.4以降に適用されます。)

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`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、スナップショット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なマークが付きます。

```
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なマークが付きます。

```
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。

10.6.3 スナップショットのメタデータ修正

Snapperでは、スナップショットの説明、クリーンアップアルゴリズム、および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。**snapper list**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。

10.6.4 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**コマンドを実行します。

現在のデフォルトのサブボリュームスナップショットの削除は許可されません。

Snapperでスナップショットを削除すると、空いたスペースはバックグラウンドで実行されているBtrfsプロセスによって要求されます。つまり、空きスペースが見えるように、あるいは使用できるようになるまでに遅れが生じます。スナップショットの削除で空いたスペースをすぐに使用したい場合は、deleteコマンドで`--sync`オプションを指定します。



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。

snapper delete --sync 23

デフォルト設定(root)のスナップショット23を削除し、空いたスペースをすぐに使用できるようにします。



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するのに、Snapper用のメタデータを含むXMLファイルが欠落している場合があります。この場合、スナップショットがSnapperには見えないため、手動で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するようにします。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、[10.6.1.2項「クリーンアップアルゴリズム」](#)を参照してください。

10.7 スナップショットの自動クリーンアップ

スナップショットはディスク容量を占有し、時間が経つにつれて、スナップショットが占有するディスク容量が増える可能性があります。ディスク容量が不足しないようにするために、Snapperは、古いスナップショットを自動的に削除するためのアルゴリズムを備えています。これらのアルゴリズムは、タイムラインスナップショットと番号付きスナップショット(管理スナップショットとインストールスナップショットのペア)を区別します。各タイプに対して、保持するスナップショットの数を指定できます。

そのほかに、オプションでディスク容量クォータを指定し、スナップショットが占有可能な最大ディスク容量を定義することもできます。また、事前スナップショットと事後スナップショットのペアに違いがない場合、それらのペアを自動的に削除することもできます。

クリーンアップアルゴリズムは常に1つのSnapper設定にバインドされるため、各設定に対してアルゴリズムを設定する必要があります。特定のスナップショットが自動的に削除されないようにするには、[問：](#)を参照してください。

デフォルトのセットアップ(root)は、番号付きスナップショットと、空の事前/事後スナップショットのペアのクリーンアップを実行するように設定されています。クォータサポートが有効になっている場合、スナップショットは、ルートパーティションの使用可能なディスク容量を50%までしか占有できません。タイムラインスナップショットはデフォルトで無効になっているため、タイムラインのクリーンアップアルゴリズムも無効になっています。

10.7.1 番号付きスナップショットのクリーンアップ

番号付きスナップショット(管理スナップショットとインストールスナップショットのペア)のクリーンアップは、Snapper設定の次のパラメータで制御します。

NUMBER_CLEANUP

インストールスナップショットと管理スナップショットのペアのクリーンアップを有効または無効にします。有効にすると、スナップショットのペアは、スナップショットの合計数がNUMBER_LIMITまたはNUMBER_LIMIT_IMPORTANT、あるいはその両方で指定された数を超え、「かつ」NUMBER_MIN_AGEで指定された保存期間を超えた場合に削除されます。有効な値: yes(はい) (有効)、no(いいえ) (無効)。

デフォルトの設定は"yes"です。

変更または設定を行うコマンドの例:

```
> sudo snapper -c CONFIG set-config "NUMBER_CLEANUP=no"
```

NUMBER_LIMIT / NUMBER_LIMIT_IMPORTANT

保持する通常のインストール/管理スナップショットのペア、または重要なインストール/管理スナップショットのペア、あるいはその両方の数を定義します。NUMBER_CLEANUPが"no"に設定されている場合、無視されます。

デフォルト値は、NUMBER_LIMITでは"2-10"、NUMBER_LIMIT_IMPORTANTでは"4-10"です。クリーニングアルゴリズムは、スナップショットとファイルシステムのスペースを考慮せずに、指定された最大値を超えるスナップショットを削除します。また、スナップショットとファイルシステムの制限に達するまで、最小値を超えるスナップショットも削除します。

変更または設定を行うコマンドの例:

```
> sudo snapper -c CONFIG set-config "NUMBER_LIMIT=10"
```



重要: 範囲値と定数値の比較

クォータサポートが有効な場合は([10.7.5項「ディスククォータサポートの追加」](#)を参照)、制限を「最小値-最大値」の範囲として指定する必要があります。たとえば、2-10のように指定します。クォータサポートが無効な場合は、定数値(たとえば10)を指定する必要があります。そうしないと、エラーが発生してクリーンアップに失敗します。

NUMBER_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。ここで指定した期間に達していなければ、スナップショットはいくつ存在していても削除されません。

デフォルトの設定は"1800"です。

変更または設定を行うコマンドの例:

```
> sudo snapper -c CONFIG set-config "NUMBER_MIN_AGE=864000"
```



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、NUMBER_LIMIT*で定義された数のスナップショットを常に保持する場合は、NUMBER_MIN_AGEを0に設定します。

次の例は、保存期間に関係なく最新の10個の重要なスナップショットと通常のスナップショットを保持するための設定を示しています。

```
NUMBER_CLEANUP=yes  
NUMBER_LIMIT_IMPORTANT=10  
NUMBER_LIMIT=10  
NUMBER_MIN_AGE=0
```

特定の期間を超えてスナップショットを保存しない場合、NUMBER_LIMIT*を0に設定し、期間をNUMBER_MIN_AGEに指定します。

次の例は、10日経っていないスナップショットのみを保持するための設定を示しています。

```
NUMBER_CLEANUP=yes  
NUMBER_LIMIT_IMPORTANT=0  
NUMBER_LIMIT=0  
NUMBER_MIN_AGE=864000
```

10.7.2 タイムラインスナップショットのクリーンアップ

タイムラインスナップショットのクリーンアップは、Snapper設定の次のパラメータで制御します。

TIMELINE_CLEANUP

タイムラインスナップショットのクリーンアップを有効または無効にします。有効にすると、スナップショットは、スナップショットの合計数がTIMELINE_LIMIT_*で指定された数を超え、「かつ」TIMELINE_MIN_AGEで指定された保存期間を超える場合に削除されます。有効値: yes, no。

デフォルトの設定は"yes"です。

変更または設定を行うコマンドの例:

```
> sudo snapper -c CONFIG set-config "TIMELINE_CLEANUP=yes"
```

TIMELINE_LIMIT_DAILY, TIMELINE_LIMIT_HOURLY, TIMELINE_LIMIT_MONTHLY,

TIMELINE_LIMIT_WEEKLY, TIMELINE_LIMIT_YEARLY

1時間、1日、1週間、1カ月間、および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。ただし、TIMELINE_LIMIT_WEEKLYは例外であり、デフォルトで「"0"」に設定されています。

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。

デフォルトの設定は"1800"です。

例 10.1: タイムライン設定の例

```
TIMELINE_CLEANUP="yes"
TIMELINE_CREATE="yes"
TIMELINE_LIMIT_DAILY="7"
TIMELINE_LIMIT_HOURLY="24"
TIMELINE_LIMIT_MONTHLY="12"
TIMELINE_LIMIT_WEEKLY="4"
TIMELINE_LIMIT_YEARLY="2"
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保存期間が30分(1800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24個のスナップショットが保持されます。
- 1日ごと: 各日の最初に作成されたスナップショットが、直近の7日分保持されます。
- 1カ月ごと: 各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12カ月分保持されます。

- 1週ごと: 各週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4週分保持されます。
- 1年ごと: 各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2年分保持されます。

10.7.3 違いがないスナップショットのペアのクリーンアップ

10.1.2項「スナップショットのタイプ」で説明したように、YaSTモジュールまたはZypperを実行すると、起動時に事前スナップショットが作成され、終了時に事後スナップショットが作成されます。変更を何も加えていない場合、事前スナップショットと事後スナップショットには違いがありません。Snapper設定で次のパラメータを設定することで、そのような「空の」スナップショットのペアを自動的に削除できます。

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。

デフォルトの設定は"yes"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。

デフォルトの設定は"1800"です。

10.7.4 手動で作成されたスナップショットのクリーンアップ

Snapperは、手動で作成されたスナップショットに対するカスタムクリーンアップアルゴリズムを備えていません。ただし、手動で作成されたスナップショットに、numberクリーンアップアルゴリズムまたはtimelineクリーンアップアルゴリズムを割り当てることができます。その場合、スナップショットは、指定されたアルゴリズムの「クリーンアップキュー」に入ります。クリーンアップアルゴリズムは、スナップショットの作成時に指定することも、既存のスナップショットを変更して指定することもできます。

snapper create --description "Test" --cleanup-algorithm number

デフォルト(ルート)設定のスタンドアロンスナップショット(singleタイプ)を作成して、numberクリーンアップアルゴリズムを割り当てます。

`snapper modify --cleanup-algorithm "timeline" 25`

番号が25のスナップショットを変更して、クリーンアップアルゴリズム`timeline`を割り当てます。

10.7.5 ディスククォータサポートの追加

上で説明したnumberクリーンアップアルゴリズムまたはtimelineクリーンアップアルゴリズム、あるいはその両方のほかに、Snapperはクォータもサポートします。スナップショットが占有できる使用可能な容量の割合を定義できます。この割合の値は常に、各Snapper設定で定義されたBtrfsサブボリュームに適用されます。

Btrfsクォータはユーザにではなく、サブボリュームに適用されます。Btrfsクォータを使用するだけでなく、ディスクスペースクォータをユーザやグループに適用することもできます(たとえば、`quota`コマンドを使用)。

インストール時にSnapperを有効にした場合、クォータサポートは自動的に有効になっています。後から手動でSnapperを有効にする場合は、`snapper setup-quota`を実行することでクォータサポートを有効にできます。そのためには有効な設定が必要です(詳細については、10.5項「Snapper設定の作成と変更」を参照してください)。

クォータサポートは、Snapper設定の次のパラメータで制御します。

QGROUP

Snapperによって使用されるBtrfsクォータグループです。設定されていない場合は、`snapper setup-quota`を実行します。すでに設定されている場合は、`man 8 btrfs-qgroup`について十分理解している場合にのみ変更してください。この値は`snapper setup-quota`で設定されます。値を変更しないでください。

SPACE_LIMIT

スナップショットが使用できる容量の制限を、1を100%とする小数で指定します。値の範囲は0~1(0.1 = 10%、0.2 = 20%、...)です。

次の制限とガイドラインが適用されます。

- クォータは、既存のnumberクリーンアップアルゴリズムまたはtimelineクリーンアップアルゴリズム、あるいはその両方に「追加」する形でのみアクティブ化されます。クリーンアップアルゴリズムがアクティブになっていない場合、クォータ制約は適用されません。
- クォータサポートが有効な場合、Snapperは必要に応じてクリーンアップを2回実行します。最初の実行では、numberスナップショットおよびtimelineスナップショットに対して指定されているルールを適用します。この実行後にクォータを超えた場合にのみ、2回目の実行でクォータ固有のルールが適用されます。
- クォータサポートが有効になっていて、クォータを超えた場合でも、Snapperは常に、`NUMBER_LIMIT*`および`TIMELINE_LIMIT*`の値で指定された数のスナップショットを保持します。したがって、`NUMBER_LIMIT*`および`TIMELINE_LIMIT*`に対する値の範囲(MIN-MAX)を指定して、クォータを確実に適用できるようにすることをお勧めします。たとえば、`NUMBER_LIMIT=5-20`が設定されている場合、Snapperは、最初のクリーンアップを実行して、標準の番号付きスナップショットの数を20に減らします。これら20個のスナップショットがクォータを超えると、Snapperは、2回目の実行時に、クォータが満たされるまで最も古いスナップショットから順番に削除します。スナップショットが占有する容量にかかわらず、少なくとも5つのスナップショットは常に保持されます。

10.8 スナップショットが使用する排他的なディスク容量の表示

スナップショットはデータを共有してストレージ容量を効率的に使用できるため、`du`や`df`などの通常のコマンドを使用しても、使用済みディスク容量は正確に測定されません。クォータを有効にしてBtrfsのディスク容量を解放する場合は、共有領域ではなく、各スナップショットで使用されている排他的なディスク容量を把握する必要があります。Snapper 0.6以降では、各スナップショットの使用済みディスク容量がUsed Space列に報告されます。

```
# snapper--iso list
# | Type | Pre # | Date | User | Used Space | Cleanup | Description
| Userdata
-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
0 | single | | | root | | | current
|
1* | single | | 2019-07-22 13:08:38 | root | 16.00 KiB | | first root
filesystem |
2 | single | | 2019-07-22 14:21:05 | root | 14.23 MiB | number | after
installation | important=yes
```

```

3 | pre | | 2019-07-22 14:26:03 | root | 144.00 KiB | number | zypp(zypper)
  | important=no
4 | post | 3 | 2019-07-22 14:26:04 | root | 112.00 KiB | number |
  | important=no
5 | pre | | 2019-07-23 08:19:36 | root | 128.00 KiB | number | zypp(zypper)
  | important=no
6 | post | 5 | 2019-07-23 08:19:43 | root | 80.00 KiB | number |
  | important=no
7 | pre | | 2019-07-23 08:20:50 | root | 256.00 KiB | number | yast sw_single
  |
8 | pre | | 2019-07-23 08:23:22 | root | 112.00 KiB | number |
  | zypp(ruby.ruby2.5) | important=no
9 | post | 8 | 2019-07-23 08:23:35 | root | 64.00 KiB | number |
  | important=no
10 | post | 7 | 2019-07-23 08:24:05 | root | 16.00 KiB | number |
    |

```

btrfs コマンドは、スナップショットが使用するスペースの別のビューを提供します。

```

# btrfs qgroup show -p /
qgroupid          rfer          excl parent
-----
0/5              16.00KiB      16.00KiB ---
[...]
0/272            3.09GiB      14.23MiB 1/0
0/273            3.11GiB     144.00KiB 1/0
0/274            3.11GiB     112.00KiB 1/0
0/275            3.11GiB     128.00KiB 1/0
0/276            3.11GiB      80.00KiB 1/0
0/277            3.11GiB     256.00KiB 1/0
0/278            3.11GiB     112.00KiB 1/0
0/279            3.12GiB      64.00KiB 1/0
0/280            3.12GiB      16.00KiB 1/0
1/0              3.33GiB     222.95MiB ---

```

qgroupid列には、各サブボリュームの識別番号が表示され、qgroupレベルとIDの組み合わせが割り当てられます。

rfer列には、サブボリュームに参照されるデータ合計数が表示されます。

excl列には、各サブボリュームの排他的なデータが表示されます。

parent列には、サブボリュームの親qgroupが表示されます。

最後の項目、1/0、は親qgroupの合計が表示されます。上記の例では、サブボリュームのすべてが削除される場合、222.95MiBが解放されます。次のコマンドを実行して、各サブボリュームに関連するスナップショットが確認されます。

```

# btrfs subvolume list -st /
ID gen top level path
--
267 298 266 @/.snapshots/1/snapshot

```



```
272 159 266 @/.snapshots/2/snapshot
273 170 266 @/.snapshots/3/snapshot
274 171 266 @/.snapshots/4/snapshot
275 287 266 @/.snapshots/5/snapshot
276 288 266 @/.snapshots/6/snapshot
277 292 266 @/.snapshots/7/snapshot
278 296 266 @/.snapshots/8/snapshot
279 297 266 @/.snapshots/9/snapshot
280 298 266 @/.snapshots/10/snapshot
```

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、スナップショットがシステムサブボリュームの多くのディスク領域を占有することになります。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[10.6.4 項「スナップショットの削除」](#)を参照してください。

10.9 ホットスポットに関する一般的な質問とその回答 (FAQ)

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： 特定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、[10.1.3 項「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： 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、[10.3 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： スナップショットを削除から保護することができますか？

答： 現在のところ、Snapperでは、スナップショットが手動で削除されるのを防ぐ方法はありません。ただし、スナップショットがクリーンアップアルゴリズムによって自動的に削除されるのを防ぐことはできます。手動で作成されたスナップショット([10.6.2 項「スナップショットの作成」](#)を参照)には、`--cleanup-algorithm`でクリーンアップアルゴリズムを指定しない限り、クリーンアップアルゴリズムは割り当てられていません。自動的に作成されたスナップショットには、常にnumberアルゴリズムまたはtimelineアルゴリズムが割り当てられています。1つ以上のスナップショットからそのような割り当てを削除するには、次の手順に従います。

1. 使用可能なすべてのスナップショットの一覧を表示します。


```
> sudo snapper list -a
```

2. 削除されないようにするスナップショットの数を記憶します。
3. 次のコマンドを実行します。数字のプレースホルダを、記憶した数に置き換えます。

```
> sudo snapper modify --cleanup-algorithm "" #1 #2 #n
```

4. もう一度**snapper list -a**を実行して、結果を確認します。変更したスナップショットの列**Cleanup**のエントリが空になります。

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。

11 KLPによるライブカーネルパッチ

このドキュメントでは、KLP (カーネルライブパッチ)適用テクノロジーの基本原理について説明するとともに、SLE Live Patchingサービスの使用ガイドラインを提供します。

KLPでは、再起動せずにLinuxカーネルに最新のセキュリティアップデートを適用できます。これにより、ミッションクリティカルなシステムにとって特に重要なシステムのアップタイムと可用性が最大化されます。

このドキュメントで提供される情報は、AMD64/Intel 64、POWER、およびIBM Zアーキテクチャに関連しています。

11.1 カーネルライブパッチの利点

KLPにはいくつかの利点があります。

- 多数のサーバを自動的に最新状態に保つことは、組織が特定のコンプライアンス認定を取得または維持するために不可欠です。KLPは、コストのかかる保守ウィンドウの必要性を削減しながら、コンプライアンスの達成に役立ちます。
- サービスレベル契約を締結している企業は、特定レベルのシステムのアクセシビリティとアップタイムを保証する必要があります。ライブパッチにより、ダウンタイムを発生させることなくシステムにパッチを適用できます。
- KLPは標準のシステムアップデートメカニズムの一部であるため、専門的なトレーニングや複雑な保守ルーチンの導入の必要はありません。

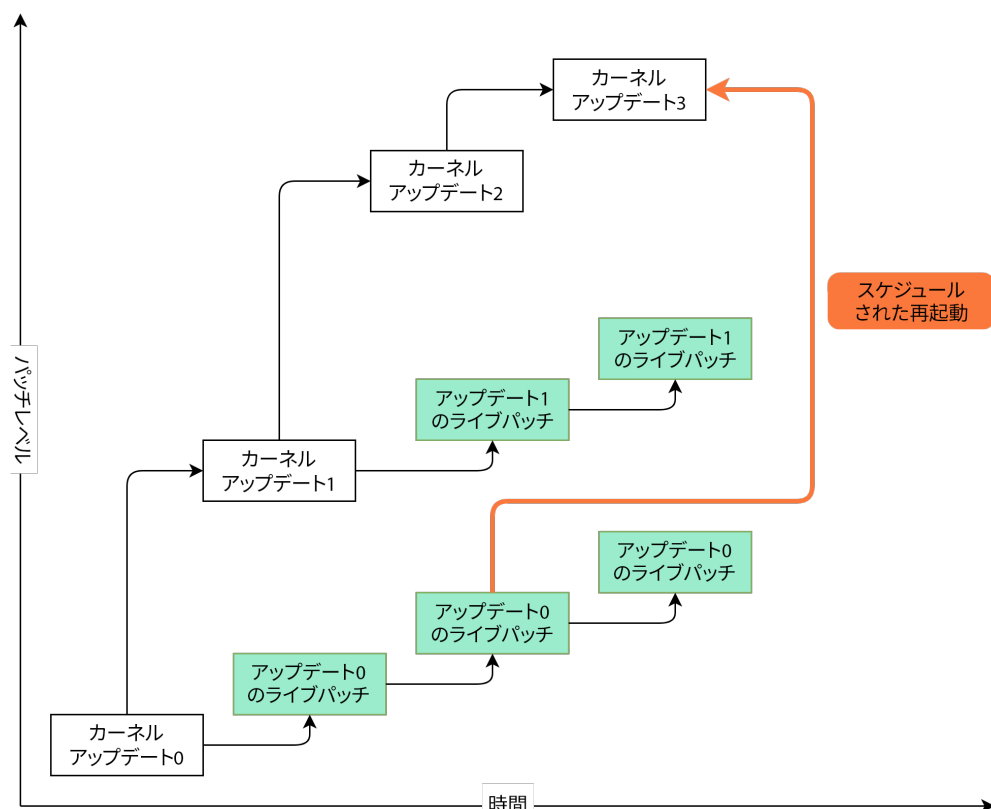
11.2 カーネルライブパッチの概要

カーネルライブパッチは、メインのカーネルパッケージとは別のコードが変更されたパッケージとして提供されます。ライブパッチは累積的であるため、最新のパッチには、カーネルパッケージの以前のものからのすべての修正が含まれています。各カーネルライブパッケージは発行された正確なカーネルリビジョンに関連付けられています。ライブパッチパッケージバージョン番号は修正が追加されるたびに増加します。

！ 重要: カーネルアップデートと比較したライブパッチ

ライブパッチには重要な修正のみが含まれ、再起動が必要な通常のカーネルアップデートに置き換わるものではありません。ライブパッチは、適切なカーネルアップデートおよび再起動が実行されるまでカーネルを保護する一時的な対策として検討してください。

次の図は、ライブパッチとカーネルアップデートの全体的な関係を示しています。現在アクティブなライブパッチによって対処されるCVEと不具合レポートのリストは、**`klp -v patches`** コマンドを使用して表示できます。



ライブパッチとともに複数のバージョンのカーネルパッケージをインストールすることができます。これらのパッケージは競合しません。実行中のカーネル用にライブパッチとともにアップデートされたカーネルパッケージをインストールできます。この場合、システムを再起動するように求められる場合があります。SLE Live Patchingサブスクリプションを持つユーザは、実行中のカーネルのライブパッチアップデートがある限り、テクニカルサポートを受ける資格があります(11.5.1項「ライブパッチの有効期限を確認する」を参照)。

KLPを有効にすると、すべてのカーネルアップデートにライブパッチパッケージが付属します。このライブパッチには修正は含まれておらず、対応するカーネルの将来のライブパッチのシードとして機能します。これらの空のシードパッチは `initial patches` と呼ばれます。

11.2.1 カーネルライブパッチのスコープ

SLE Live Patchingのスコープには、SUSE Common Vulnerability Scoring System (CVSS、SUSE CVSSはCVSS v3.1システムに基づく)レベル7以上の脆弱性の修正と、システムの安定性またはデータ破損に関連するバグ修正が含まれます。ただし、指定されたカテゴリに該当するすべての修正のライブパッチを作成することは技術的に不可能な場合があります。したがって、SUSEではカーネルライブパッチの作成が技術的な理由で不可能な状況で修正をスキップする権利を留保します。現在、対象となる修正の95%以上がライブパッチとしてリリースされています。CVSS (SUSE CVSSレーティングのベース)の詳細については、「[Common Vulnerability Scoring System SIG \(https://www.first.org/cvss/\)](https://www.first.org/cvss/)」を参照してください。

11.2.2 カーネルライブパッチの制限

KLPには、関数の置換と、相互に依存する関数セットの置換の適切な処理が含まれます。これは、古いコードへの呼び出しを別のメモリ位置にある更新されたコードにリダイレクトすることによって行われます。データ構造の変更は、データがそのまま残り、拡張または再解釈できないため、状況をより複雑にします。データ構造の間接的な変更を許可する技術はありますが、特定の修正はライブパッチに変換できません。この状況では、システムの再起動が、修正を適用する唯一の方法です。

11.3 YaSTを使用したカーネルライブパッチの有効化


ご使用のシステムでKLPを有効にするには、アクティブなSLESおよびSLE Live Patchingサブスクリプションが必要です。サブスクリプションのステータスを確認し、SLE Live Patchingサブスクリプションの登録コードを取得するには、[SUSE Customer Center \(https://scc.suse.com/\)](https://scc.suse.com/)を参照してください。

カーネルライブパッチをシステムで有効にするには、次の手順に従います。

1. **`yast2 registration`** コマンドを実行して、拡張機能の選択をクリックします。

2. 入手可能な拡張機能のリストでSUSE Linux Enterprise Live Patching 15を選択し、次へをクリックします。
3. ライセンス条項を確認し、次へをクリックします。
4. SLE Live Patchingの登録コードを入力し、次へをクリックします。
5. Installation Summary (インストールの概要)と選択されているPatterns (パターン)を確認します。Live PatchingとSLE Live Patching Lifecycle Dataパターンが、依存関係を満たすための追加のパッケージとともにインストール用に自動的に選択される必要があります。
6. Accept (承諾)をクリックしてインストールを完了します。これにより、システムに基本のカーネルライブパッチコンポーネント、初期ライブパッチ、および必要な依存関係がインストールされます。

11.4 コマンドラインからのカーネルライブパッチの有効化

カーネルライブパッチを有効にするには、アクティブなSLESおよびSLES Live Patchingサブスクリプションが必要です。サブスクリプションのステータスを確認し、SLES Live Patchingサブスクリプションの登録コードを取得するには、[SUSE Customer Center \(https://scc.suse.com/\)](https://scc.suse.com/) を参照してください。

1. **`sudo SUSEConnect --list-extensions`**を実行します。SLES Live Patchingの正確なアクティベーションコマンドに注意します。コマンド出力の例(省略形):

```
$ SUSEConnect --list-extensions
...
SUSE Linux Enterprise Live Patching 15 SP5 x86_64
Activate with: SUSEConnect -p sle-module-live-patching/15.5/x86_64 \
-r ADDITIONAL REGCODE
```

2. 取得したコマンドに続いて**`-r LIVE_PATCHING_REGISTRATION_CODE`**を使用してSLES Live Patchingを有効にします。例:

```
SUSEConnect -p sle-module-live-patching/15.5/x86_64 \
-r LIVE_PATCHING_REGISTRATION_CODE
```


3. コマンド**`zypper install -t pattern lp_sles`**を使用して必要なパッケージと依存関係をインストールします。

この時点で、システムはすでにライブパッチが適用されています。

プロセスがバックグラウンドでどのように機能するかを次に示します。パッケージインストーラシステムが、ライブパッチを適用できるカーネルがインストールされていること、およびソフトウェアチャンネルにそのカーネルのライブパッチがあることを検出すると、システムはインストール用のライブパッチを選択します。カーネルはパッケージインストールの一部としてライブパッチ修復を受信します。製品インストールが完了する前でも、カーネルにライブパッチが適用されます。

11.5 カーネルライブパッチの適用

カーネルライブパッチは、定期的なシステムアップデートの一部としてインストールされます。ただし、認識すべきいくつかのことがあります。

- `kernel-livepatch-*`パッケージが実行中のカーネル用にインストールされている場合、カーネルにライブパッチが適用されます。コマンド `zypper se --details kernel-livepatch-*` を使用して、システムにインストールされているカーネルライブパッチパッケージを確認できます。
- `kernel-default`パッケージがインストールされる場合、アップデートマネージャはシステムを再起動するように求めます。このメッセージが表示されないようにするには、パッチ適用操作からカーネルアップデートを除外できます。これは、Zypperを使用してパッケージロックを追加することで実行できます。SUSE Managerはチャンネルコンテンツをフィルタすることもできます([Live Patching with SUSE Manager \(https://documentation.suse.com/external-tree/en-us/suma/4.1/suse-manager/administration/live-patching.html\)](https://documentation.suse.com/external-tree/en-us/suma/4.1/suse-manager/administration/live-patching.html)  (SUSE Managerによるライブパッチ適用)を参照)。
- `k1p status`コマンドを使用してパッチ適用ステータスを確認できます。インストール済みのパッチを調べるには、`k1p -v patches`コマンドを実行します。
- システムに複数のカーネルパッケージがインストールされている可能性があります、任意の時点で実行されるのはそれらの1つのみであることに注意してください。同様に、複数のライブパッチパッケージがインストールされる場合がありますが、カーネルにロードされるライブパッチは1つのみです。
- アクティブなライブパッチは、`initrd`に含まれています。これは、予期しない再起動が発生した場合に、ライブパッチ修正が適用された状態でシステムが起動するため、再びパッチを適用する必要がないことを意味します。

11.5.1 ライブパッチの有効期限を確認する

`lifecycle-data-sle-module-live-patching`がインストールされていることを確認し、`zypper lifecycle`コマンドを実行します。出力の `Package end of support if different from product` セクションにライブパッチの有効期限が表示されるはずです。

すべてのライブパッチは基盤となるカーネルパッケージのリリースから1年間、アップデートを受け取ります。「[Maintained kernels, patch updates and lifecycle \(https://www.suse.com/products/live-patching/current-patches/\)](https://www.suse.com/products/live-patching/current-patches/)」ページでは、製品の拡張機能をインストールすることなく、実行中のカーネルバージョンに基づいて有効期限を確認できます。

11.6 カーネルライブパッチの問題のトラブルシューティング

11.6.1 手動によるパッチのダウングレード

最新のライブパッチに問題がある場合は、現在インストールされているライブパッチを以前のバージョンにダウングレードできます。システムで問題が発生する前に、パッチのダウングレードを実行することをお勧めします。システムログにカーネル警告またはカーネルエラーレースが含まれているシステムは、パッチのダウングレード手順に適していない場合があることに注意してください。システムがパッチダウングレードの要件を満たしているかどうか不明な場合は、SUSEテクニカルサポートに連絡してください。

手順 11.1: 手動によるパッチのダウングレード

1. `klp -v patches` コマンドを使用して実行中のライブパッチを特定します。`RPM:` で開始される行に現在実行中のパッチが表示されます。例:

```
RPM: kernel-livepatch-5_3_18-24_29-default-2-2.1.x86_64
```

前の例の `5_3_18-24_29-default` は、実行中のカーネルの正確なバージョンを示しています。

2. コマンド `zypper search -s kernel-livepatch-RUNNING_KERNEL_VERSION-default` を使用して、以前のバージョンのパッチを検索します。このコマンドは、使用可能なパッケージバージョンのリストを返します。新しいライブパッチパッケージがリリースされるたびに、バージョン番号が1つずつ増加することに注意してください。現在のリリースより1つ前のバージョン番号を選択してください。

3. `zypper in --oldpackage kernel-livepatch-RUNNING_KERNEL_VERSION-default=DESIRED_VERSION` コマンドを使用して、目的のバージョンをインストールします。

12 ユーザ空間ライブパッチ

この章では、ユーザ空間ライブパッチの基本的な原則と使用法について説明します。

12.1 ユーザ空間ライブパッチについて

ユーザ空間ライブパッチ(ULP)とは、実行中のプロセスで使用されるライブラリに、中断することなくパッチを適用するプロセスのことです。ライブパッチとしてセキュリティ修正が利用できるようになるたびに、ライブパッチを適用した後にプロセスを再起動することなくカスタマサービスが保護されます。

ライブパッチ操作は、`libpulp`の一部である`ulp`ツールを使用して実行されます。`libpulp`は、`libpulp.so`ライブラリおよび、ライブラリをライブパッチ可能にし、ライブパッチを適用する`ulp`バイナリから構成されるフレームワークです。



ヒント

標準ユーザまたは特権を持つユーザのいずれかとして`sudo`メカニズムを使用して`ulp`コマンドを実行できます。両者の違いは、`sudo`を使用して`ulp`を実行すると、`root`によって実行されているプロセスまたはパッチプロセスの情報を表示できることです。

12.1.1 前提条件

ULPを機能させるには、2つの要件を満たす必要があります。

- 次を実行してシステムにULPをインストールしている。

```
> sudo zypper in libpulp0 libpulp-tools
```

- 目的のライブパッチサポートを含むアプリケーションを`libpulp.so.0`ライブラリをプリロードすることによって起動する必要がある。詳しくは12.1.3項「`libpulp`の使用」を参照してください。

12.1.2 サポートされているライブラリ

現時点では、glibcおよびopenssl (openssl1_1)のみがサポートされています。追加のパッケージは、ライブパッチ用に準備できてから使用できるようになります。glibcおよびopensslのライブパッチを受け取るには、glibc-livepatchesパッケージとopenssl-livepatchesパッケージの両方をインストールします。

```
> zypper install glibc-livepatches openssl-livepatches
```

12.1.3 libpulpの使用

アプリケーションでライブパッチを有効にするには、アプリケーション起動時にlibpulp.so.0ライブラリをプリロードする必要があります。

```
> LD_PRELOAD=/usr/lib64/libpulp.so.0 APPLICATION_CMD
```

12.1.3.1 ライブラリがライブパッチ可能かどうかの確認

ライブラリがライブパッチ可能かどうかを確認するには、次のコマンドを使用します。

```
> ulp livepatchable PATH_TO_LIBRARY
```

12.1.3.2 .soファイルがライブパッチコンテナかどうかの確認

共有オブジェクト(.so)は、ULPパッチ記述が埋め込まれている場合にはライブパッチコンテナです。次のコマンドを使用して確認できます。

```
> readelf -S SHARED_OBJECT | grep .ulp
```

共有オブジェクトに.ulpセクションと.ulp.revセクションの両方があることを出力が示している場合、ライブパッチコンテナです。

12.1.3.3 ライブパッチの適用

ライブパッチは、次のような、ulp triggerコマンドを使用して適用されます。

```
> ulp trigger -p PID LIVEPATCH.so
```

PIDを、パッチ対象のライブラリを使用する実行中のプロセスのプロセスIDに置き換え、LIVEPATCH.soを実際のライブパッチファイルに置き換えます。このコマンドは、次のステータスメッセージのいずれかを返します。

SUCCESS

ライブパッチ操作は成功しました。

SKIPPED

パッチは、プロセスでロードされたどのライブラリ用にも設計されていなかったためスキップされました。

ERROR

エラーが発生しました。libpulp内部メッセージバッファを調べて詳細情報を取得できます。詳細については、[12.1.3.6項「内部メッセージキューの表示」](#)を参照してください。

ワイルドカードを使用して複数のライブパッチを適用することもできます。次に例を示します。

```
> ulp trigger '*.so'
```

このコマンドは、現在のフォルダ内のすべてのパッチをlibpulpライブラリがロードされているすべてのプロセスに適用しようとします。パッチは、プロセスに不適である場合、自動的にスキップされます。最後に、適用に成功したパッチ数および適用されたプロセス数がツールに表示されます。

12.1.3.4 ライブパッチを元に戻す

ulp triggerコマンドを使用してライブパッチを元に戻します。ライブパッチを元に戻すには、2つの方法があります。--revertスイッチを使用し、ライブパッチコンテナを渡すことによってライブパッチを元に戻すことができます。

```
> ulp trigger -p PID --revert LIVEPATCH.so
```

または、特定のライブラリに関連付けられているすべてのパッチを削除することもできます。次に例を示します。

```
> ulp trigger -p PID --revert-all=LIBRARY
```

この例では、LIBRARYは、libcrypto.so.1.1などの実際のライブラリを表します。

後者の方法は、元のライブパッチのソースコードを利用できない場合に便利です。または、特定の古いパッチを削除し、新しいパッチを適用するが、同時にターゲットアプリケーションがセキュアコードの実行を継続する場合です。次に例を示します。

```
> ulp trigger -p PID --revert-all=libcrypto.so.1.1 new_livepatch2.so
```

12.1.3.5 適用パッチの表示

次を実行することによって、ライブパッチが適用されているアプリケーションを確認できます。

```
> ulp patches
```

ライブパッチ可能なライブラリ、プログラムにロードされているパッチ、およびパッチが対処しているバグが出力に示されます。

```
PID: 10636, name: test
Livepatchable libraries:
  in /lib64/libc.so.6:
    livepatch: libc_livepatch1.so
    bug labels: jsc#SLE-0000
  in /usr/lib64/libpulp.so.0:
```

ライブパッチによってパッチされている機能を確認することも可能です。


```
> ulp dump LIVEPATCH.so
```

12.1.3.6 内部メッセージキューの表示

`libpulp.so`からのログメッセージは、ライブラリ内のバッファに格納され、ユーザが要求しない限り表示されません。これらのメッセージを表示するには、次を実行します。

```
> ulp messages -p PID
```

12.2 詳細情報

`libpulp`に関する詳細については、プロジェクトの[Git repository \(https://github.com/SUSE/libpulp\)](https://github.com/SUSE/libpulp) を参照してください。

13 トランザクション更新

トランザクション更新は、ルートファイルシステムが読み込み専用のときにSLESを更新するためのテクノロジープレビューとして、SUSE Linux Enterprise Desktopで利用できます。トランザクション更新はアトミックであり(すべての更新が成功した場合にのみすべての更新が適用されます)、ロールバックをサポートします。システムが再起動されるまで変更は有効にならないため、実行中のシステムには影響しません。再起動は中断を伴うので、管理者は、再起動の方が実行中のサービスを妨害するよりもコストがかかるかどうかを判断する必要があります。再起動でコストがかかりすぎる場合は、トランザクション更新を使用しないでください。

トランザクションの更新は、**transactional-update**スクリプトによって、毎日実行されます。スクリプトは使用可能な更新をチェックします。更新がある場合は、ルートファイルシステムの新しいスナップショットをバックグラウンドで作成し、リリースチャネルから更新をフェッチします。新しいスナップショットが完全に更新された後で、アクティブとマークされ、システムの次の再起動後に新しいデフォルトのルートファイルシステムになります。**transactional-update**が自動的に実行するように設定されている場合(これはデフォルトの動作です)、システムも再起動します。更新を実行する時刻と再起動の保守時間枠の両方が設定可能です。

ルートファイルシステムのスナップショットの一部であるパッケージのみを更新できます。パッケージにスナップショットの一部ではないファイルが含まれている場合、更新は失敗するか、システムが破損する可能性があります。

ライセンスを受諾する必要があるRPMは更新できません。

13.1 テクノロジープレビューの制限

テクノロジープレビューとして、機能には特定の制限があります。次のパッケージは、**transactional-update**では機能しません。

- nginxのデフォルトのindex.htmlページは使用できない場合があります
- tomcat-webapps および tomcat-admin-webapps
- phpMyAdmin
- sca-appliance-*

- [mpi-selector](#)
- [emacs](#)はEmacsゲームを除いて機能します
- [bind](#) および [bind-chrootenv](#)
- [docbook*](#)
- [sblim-sfcb*](#)
- [texlive*](#)
- [iso_ent](#)
- [openjade](#)
- [opensp](#)
- [pcp](#)
- [plymouth](#)
- [postgresql-server-10](#)
- [pulseaudio-gdm-hooks](#)
- [smartmontools](#)

システムインストーラのアップデータコンポーネントは、トランザクションの更新をサポートしていないため、読み込み専用ファイルシステムでは動作しません。

その他の考慮事項:

- システムを更新してから、マシンを再起動するまでの時間を最小限に抑えることをお勧めします。
- 1つの更新のみを一度に適用できます。更新後、および次の更新が適用される前に必ず再起動してください。
- [update-alternatives](#)は、トランザクション更新後、マシンが再起動されるまで、実行しないでください。
- トランザクション更新後、再起動後まで新しいシステムユーザまたはシステムグループを作成しないでください。通常のユーザおよびグループを作成することは許容されます (UID > 1000、GID > 1000)。

- YaSTはまだトランザクション更新を認識していません。YaSTモジュールで追加のパッケージをインストールする必要がある場合、これは機能しません。/etcの設定ファイルのみを変更する通常のシステム操作は機能します。
- php7-fastcgiの場合、/srv/www/cgi-bin/phpを示すシンボリックリンク/usr/bin/php-cgiを手動で作成する必要があります。
- ntpは、古いSLESバージョンのレガシモジュールの一部です。ntpは新しいSUSE Linux Enterprise Desktopインストールでサポートされておらず、chronyに置き換えられています。ntpを引き続き使用する場合は、トランザクション更新で正しく機能させるために、新規インストールが必要です。
- sblim-sfcb: エコシステム全体がトランザクション更新と互換性がありません。
- btrfsmaintenanceパッケージの**btrfs-defrag**は、読み込み専用ルートファイルシステムでは機能しません。
- **btrfs-balance**の場合、/etc/sysconfig/btrfsmaintenanceの変数BTRFS_BALANCE_MOUNTPOINTSを/から/.snapshotsに変更する必要があります。
- **btrfs-scrub**の場合、/etc/sysconfig/btrfsmaintenanceの変数BTRFS_SCRUB_MOUNTPOINTSを/から/.snapshotsに変更する必要があります。

13.2 transactional-update有効化

システムのインストール時にトランザクショナルサーバモジュールを有効にし、トランザクショナルサーバシステム役割を選択する必要があります。実行しているシステムで後でトランザクショナルサーバモジュールからパッケージをインストールすることはサポートされておらず、システムが破損する可能性があります。

ルートパーティションのサブボリュームレイアウトの変更または独自のパーティションへのルートパーティションのサブディレクトリまたはサブボリュームの配置(/home、/var、/srv、および/optを除く)はサポートされておらず、システムが破損する可能性が高いです。

13.3 自動更新の管理

自動更新は、1日1回実行するsystemd.timerによって制御されます。これは、すべての更新に適用され、マシンが再起動する必要があることをrebootmgrdに知らせます。更新が実行される時刻を調整できます。systemd.timer(5)を参照してください。rebootmgrdがシステムを再起動するときのメンテナンス期間を調整するには、rebootmgrd(8)を参照してください。

次のコマンドで自動トランザクション更新を無効にできます。

```
# systemctl --now disable transactional-update.timer
```

13.4 transactional-update コマンド

transactional-update コマンドは、更新のアトミックインストールまたは削除を有効にします。更新は、そのすべてが正常にインストールされた場合にのみ適用されます。**transactional-update** は、更新が適用される前にシステムのスナップショットを作成し、このスナップショットを復元できます。再起動後にのみ、すべての変更がアクティブになります。

--continue

--continue オプションは、再起動せずに既存のスナップショットに複数の変更を行うためのものです。

デフォルトの **transactional-update** 動作は、現在のルートファイルシステムから新しいスナップショットを作成することです。新しいパッケージのインストールなど、何かを忘れた場合は、再起動して以前の変更を適用し、**transactional-update** を再度実行して忘れたパッケージをインストールしてから再起動する必要があります。再起動せずに、スナップショットにさらに変更を追加するために **transactional-update** コマンドを複数回実行できません。これは、以前のスナップショットからの変更を含まない独立したスナップショットが作成されるためです。

--continue オプションを使用して、再起動せずに必要なだけ変更を行います。別個のスナップショットが毎回作成され、各スナップショットには、以前のスナップショットで作成されたすべての変更、および新しい変更が含まれます。このプロセスを必要な回数だけ繰り返します。最終スナップショットに必要なものがすべて含まれている場合は、システムを再起動すると、最終スナップショットが新しいルートファイルシステムになります。

--continue オプションのもう1つの便利な機能は、既存のスナップショットを新しいスナップショットのベースとして選択できることです。次の例では、**transactional-update** を実行して、スナップショット13に基づいてスナップショットに新しいパッケージをインストールしてから、もう一度実行して別のパッケージをインストールする方法を示しています。

```
# transactional-update pkg install package_1
```

```
# transactional-update --continue 13 pkg install package_2
```


--continue [num] オプションは **snapper create --from** を呼び出します。10.6.2項「スナップショットの作成」を参照してください。

cleanup

現在のルートファイルシステムがアクティブなルートファイルシステムと同一である場合(再起動後、**transactional-update**が更新を含む新しいスナップショットを作成する前)、クリーンアップアルゴリズムのないすべての古いスナップショットはクリーンアップアルゴリズムセットを取得します。これにより、古いスナップショットがSnapperによって確実に削除されます。(snapper(8)のクリーンアップアルゴリズムに関するセクションを参照してください。)これにより、`/var/lib/overlay`内の参照されていない(したがって未使用の)`/etc`オーバーレイディレクトリもすべて削除されます。

```
# transactional-update cleanup
```

pkg in/install

zypper install コマンドを使用して、使用可能なチャネルから個々のパッケージをインストールします。このコマンドは、Program Temporary Fix (PTF) RPMファイルをインストールするために使用することもできます。

```
# transactional-update pkg install package_name
```

あるいは、

```
# transactional-update pkg install rpm1 rpm2
```

pkg rm/remove

zypper remove コマンドを使用してアクティブなスナップショットから個々のパッケージを削除します。このコマンドは、PTF RPMファイルを削除するために使用することもできます。

```
# transactional-update pkg remove package_name
```

pkg up/update

zypper update コマンドを使用してアクティブなスナップショットから個々のパッケージを更新します。ベースファイルシステムのスナップショットの一部であるパッケージのみを更新できます。

```
# transactional-update pkg update package_name
```

up/update

使用可能な新しい更新がある場合、新しいスナップショットが作成され、**zypper up/update** がスナップショットを更新します。

```
# transactional-update up
```

dup

使用可能な新しい更新がある場合、新しいスナップショットが作成され、**zypper dup --no-allow-vendor-change**がスナップショットを更新します。スナップショットは後で有効にされ、再起動後に新しいルートファイルシステムになります。

```
# transactional-update dup
```

patch

使用可能な新しい更新がある場合、新しいスナップショットが作成され、**zypper patch**がスナップショットを更新します。

```
# transactional-update patch
```

rollback

これはデフォルトのサブボリュームを設定します。読み書き可能なファイルシステムを備えたシステムでは、**snapper rollback**が呼び出されます。読み込み専用ファイルシステムで引数を指定しない場合、現在のシステムは新しいデフォルトのルートファイルシステムに設定されます。数値を指定する場合、スナップショットがデフォルトのルートファイルシステムとして使用されます。読み込み専用ファイルシステムでは、追加のスナップショットは作成されません。

```
# transactional-update rollback snapshot_number
```

grub.cfg

これは、新しいGRUB2設定を作成します。カーネルパラメータの追加など、ブート設定の調整が必要になる場合があります。**/etc/default/grub**を編集し、**transactional-update grub.cfg**を実行して、再起動して変更を有効にします。直接再起動する必要があります。そうしないと、新しいGRUB2設定が次のトランザクション更新によってデフォルトで上書きされます。

```
# transactional-update grub.cfg
```

reboot

このパラメータは、アクションが完了した後に再起動をトリガーします。

```
# transactional-update dup reboot
```

--help

これにより、オプションとサブコマンドを含むヘルプ画面が印刷されます。

```
# transactional-update --help
```

13.5 トラブルシューティング

アップグレードが失敗する場合は、**supportconfig**を実行して、ログデータを収集します。/var/log/transactional-update.logを含む結果のファイルをSUSEサポートに提供します。

14 VNCによるリモートグラフィカルセッション

仮想ネットワークコンピューティング (VNC)では、グラフィカルデスクトップを介してリモートコンピュータにアクセスしたり、リモートグラフィカルアプリケーションを実行したりできます。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。この章では、デスクトップクライアントvncviewerおよびRemminaを使用してVNCサーバに接続する方法、およびVNCサーバを操作する方法について説明します。

SUSE Linux Enterprise Desktop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。

VNCサーバでは両方のタイプのセッションを異なるポートで同時に提供できます。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。

14.1 vncviewerクライアント

サーバによって提供されるVNCサービスに接続するには、クライアントが必要です。SUSE Linux Enterprise Desktopのデフォルトはvncviewerで、これはtigervncパッケージで提供されます。

14.1.1 vncviewer CLIを使用した接続

VNCビューアを起動し、サーバとの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
```



注記: ディスプレイ番号とポート番号

VNCクライアントで実際に指定するディスプレイ番号またはポート番号は、ターゲットマシンで**vncserver**によって選択されるディスプレイ番号またはポート番号と同じである必要があります。詳細については、[14.4項「永続的VNCサーバセッションを設定する」](#)を参照してください。

14.1.2 vncviewer GUIを使用した接続

--listenまたは接続先ホストを指定せずに**vncviewer**を実行すると、接続の詳細を入力するよう求めるウィンドウが表示されます。[14.1.1項「vncviewer CLIを使用した接続」](#)のようにVNC server (VNCサーバ)フィールドにホストを入力し、接続をクリックします。

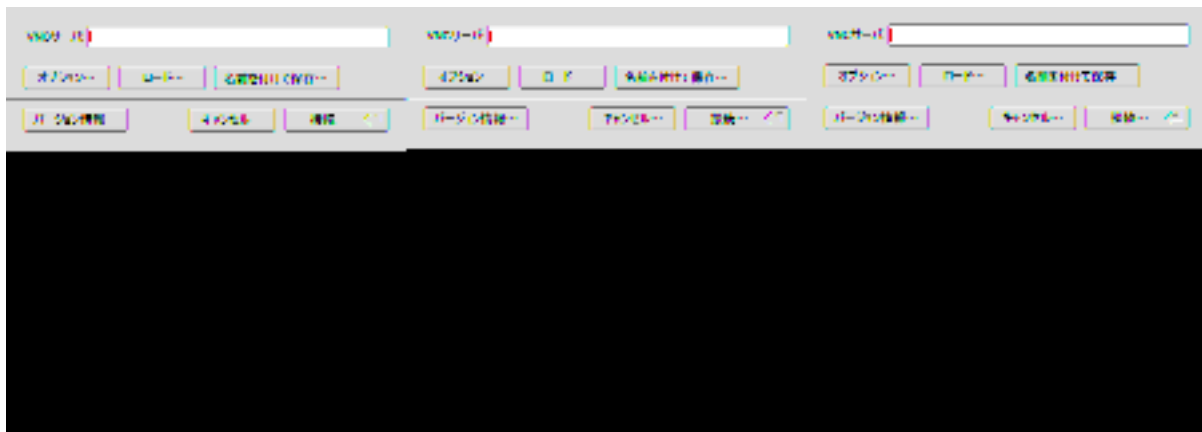


図 14.1: VNCVIEWER

14.1.3 暗号化されていない接続の通知

VNCプロトコルは、さまざまな種類の暗号化接続をサポートしています。これをパスワード認証と混同しないでください。接続がTLSを使用していない場合、「(Connection not encrypted!) (接続が暗号化されていません!)」というテキストがVNCビューアのウィンドウタイトルに表示されることがあります。

14.2 Remmina: リモートデスクトップクライアント

Remminaは、最新の機能豊富なリモートデスクトップクライアントです。VNC、SSH、RDP、Spiceなど、複数のアクセス方法をサポートしています。

14.2.1 インストール

Remminaを使用するには、`remmina`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、インストールされていない場合はインストールします。Remmina用のVNCプラグインもインストールすることを忘れないでください。

```
# zypper in remmina remmina-plugin-vnc
```

14.2.2 メインウィンドウ

`remmina`コマンドを入力してRemminaを実行します。



図 14.2: REMMINAのメインウィンドウ

メインアプリケーションウィンドウには、保存されているリモートセッションのリストが表示されます。ここでは、新しいリモートセッションを追加および保存したり、保存せずに新しいセッションをクイックスタートしたり、以前に保存したセッションを開始したり、Remminaのグローバル設定を行うことができます。

14.2.3 リモートセッションの追加


新しいリモートセッションを追加して保存するには、メインウィンドウの左上で、をクリックします。リモートデスクトップ初期設定ウィンドウが開きます。

図 14.3: リモートデスクトップ初期設定

新しく追加したリモートセッションプロファイルを指定するフィールドに入力します。最も重要な設定には次のものがあります。

名前

プロファイルの名前。この名前は、メインウィンドウにリストされます。

プロトコル

リモートセッションに接続するときに使用するプロトコル(VNCなど)。

または8単位

リモートサーバのIPアドレスまたはDNSアドレスとディスプレイ番号。

ユーザ名、パスワード

リモート認証に使用する資格情報。認証しない場合は空のままにします。

色数、品質

接続速度と品質に応じて最適なオプションを選択します。

より詳細な設定を入力するには、**詳細**タブを選択します。



ヒント: 暗号の無効化

クライアントとリモートサーバ間の通信が暗号化されていない場合は、Disable encryption (暗号の無効化)を有効にします。そうしないと接続が失敗します。

高度なSSHトンネリングと認証オプションについては、SSHタブを選択してください。

「保存」をクリックして確定します。新しいプロファイルがメインウィンドウに表示されます。

14.2.4 リモートセッションの開始

以前に保存したセッションを開始するか、または接続の詳細を保存せずにリモートセッションをクイックスタートすることができます。

14.2.4.1 リモートセッションのクイックスタート

接続の詳細を追加および保存することなく、リモートセッションをすばやく開始するには、メインウィンドウの上部にあるドロップダウンボックスとテキストボックスを使用します。

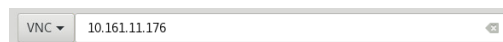


図 14.4: クイックスタート

ドロップダウンボックスから通信プロトコル(「VNC」など)を選択して、次にVNCサーバのDNSまたはIPアドレスを入力し、それに続けてコロンとディスプレイ番号を入力して、**Enter** で確定します。

14.2.4.2 保存されたリモートセッションを開く

特定のリモートセッションを開くには、セッションのリストからダブルクリックします。

14.2.4.3 リモートセッションウィンドウ

リモートセッションは別のウィンドウのタブで開きます。タブごとに1つのセッションをホストします。ウィンドウの左側にあるツールバーは、ウィンドウ/セッションの管理(フルスクリーンモードの切り替え、セッションの表示サイズに合わせたウィンドウのサイズ変更、セッションへの特定のキーストロークの送信、セッションのスクリーンショット撮影、画質の設定など)に役立ちます。

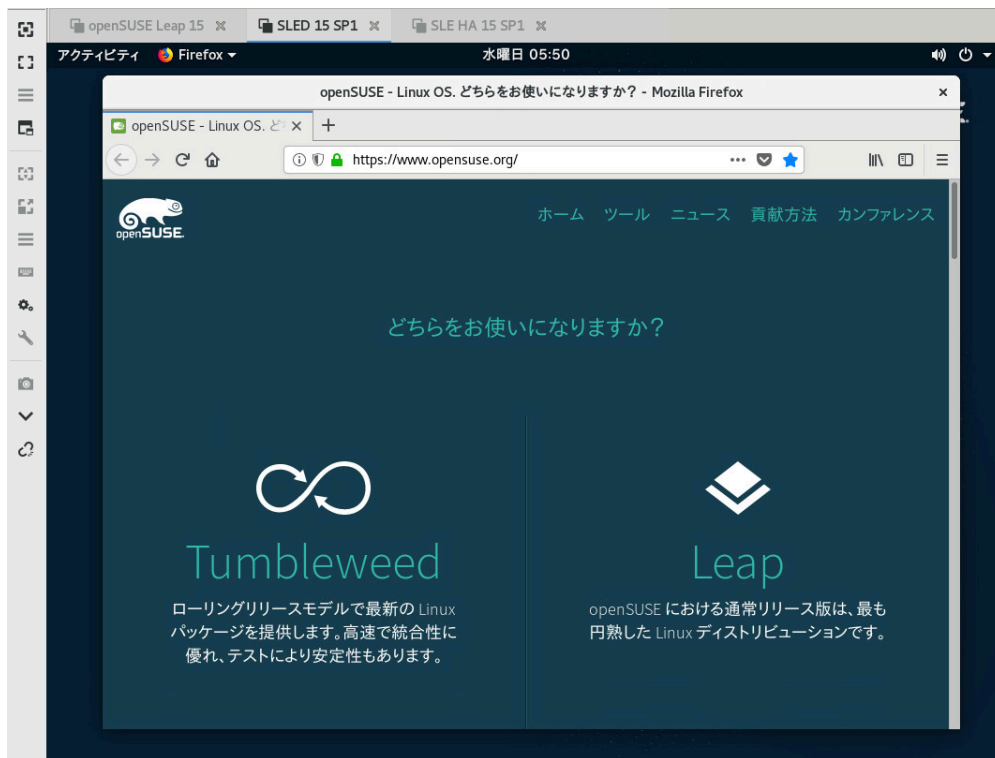


図 14.5: REMMINAのリモートセッションの表示

14.2.5 保存されたセッションの編集、コピー、および削除

保存されたリモートセッションを「編集」するには、Remminaのメインウィンドウでその名前を右クリックし、編集を選択します。関連するフィールドの説明については、[14.2.3項「リモートセッションの追加」](#)を参照してください。

保存されたリモートセッションを「コピー」するには、Remminaのメインウィンドウでその名前を右クリックし、コピーを選択します。リモートデスクトップ初期設定ウィンドウで、プロファイルの名前を変更し、関連するオプションを必要なら調整し、保存をクリックして確定します。

保存されたリモートセッションを「削除」するには、Remminaのメインウィンドウでその名前を右クリックし、削除を選択します。次のダイアログでははいをクリックして確定します。

14.2.6 コマンドラインからのリモートセッションの実行

最初にメインのアプリケーションウィンドウを開くことなく、コマンドラインまたはバッチファイルからリモートセッションを開く必要がある場合は、次の構文を使用します。

```
> remmina -c profile_name.remmina
```

Remminaのプロファイルファイルは、ホームディレクトリの`.local/share/remmina/`ディレクトリに保存されます。開きたいセッションに属しているプロファイルファイルを決定するには、Remminaを実行し、メインウィンドウでセッション名をクリックし、下部のウィンドウのステータス行でプロファイルファイルへのパスを読み込みます。

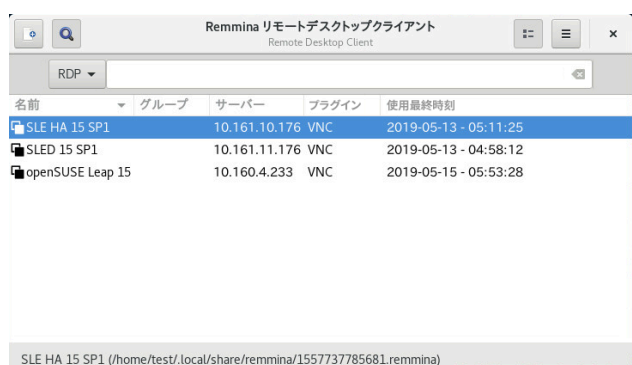


図 14.6: プロファイルファイルへのパスの読み込み

が実行されていないときに、プロファイルファイルの名前を `sle15.remmina.remmina` などのより合理的なファイル名に変更することができます。プロファイルファイルをカスタムディレクトリにコピーして、そこから `remmina -c` コマンドを使用して実行することもできます。

14.3 VNCサーバでのワンタイムセッションの設定

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キャンセルすると、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 14.1: 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. セッション管理機能無しのリモート管理を許可するをオンにします。
3. WebブラウザウィンドウでVNCセッションにアクセスする場合は、Web ブラウザを利用したアクセスを有効にするをアクティブにします。
4. 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。

5. 次へをクリックすると設定が確定し、
6. 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。



ヒント: ディスプレイマネージャの再起動

YaSTはディスプレイマネージャの設定を変更します。現在のグラフィカルセッションからログアウトし、ディスプレイマネージャを再起動して変更を有効にする必要があります。



図 14.7: リモート管理

14.3.1 使用可能な設定

SUSE Linux Enterprise Desktop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。設定を変更する必要がある場合は、システム管理者に詳細を問い合わせてください。

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。

デフォルトでは、VNCクライアントとサーバの両方が、インストール後に生成される自己署名SSL証明書を使用してセキュアな通信を試みます。デフォルトの証明書を使用することも、独自の証明書に置き換えることもできます。自己署名証明書を使用する際、初回の接続前に、VNCビューアおよびWebブラウザの両方で署名を確認する必要があります。



ヒント

一部のVNCクライアントは、デフォルトの自己署名証明書を使用した安全な接続の確立を拒否します。たとえば、Vinagreクライアントは、GnuTLSグローバル信頼ストアに対して証明書を検証し、証明書が自己署名である場合には検証に失敗します。この場合、x509以外の暗号化方式を使用するか、VNCサーバに対して正しく署名された証明書を生成し、これをクライアントのシステム信頼ストアにインポートします。

14.3.2 一時的VNCセッションを開始する

一時的VNCセッションに接続するには、VNCビューアをインストールする必要があります。[14.1項「vncviewerクライアント」](#)も参照してください。または、JavaScript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。

14.3.3 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、systemdソケットxvnc.socketを介して開始されます。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:。ビューア用にブロック(vnc1からvnc3vncまで)、JavaScriptクライアント用に3ブロック(vnchttpd1からvnchttpd33まで)。デフォルトでは、vnc1とvnchttpd1だけが有効です。

ブート時にVNCサーバソケットをアクティブにするには、次のコマンドを実行します。

```
> sudo systemctl enable xvnc.socket
```

すぐにソケットを起動するには、次のコマンドを実行します。

```
> sudo systemctl start xvnc.socket
```

Xvncサーバは、`server_args`オプションを介して設定できます。オプションのリストについては、`Xvnc --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 systemctl reload xvnc.socket
```

！ 重要: ファイアウォールとVNCポート

手順14.1「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、『Security and Hardening Guide』、第23章「Masquerading and firewalls」を参照してください。

14.4 永続的VNCサーバセッションを設定する

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セッションでも使用できます。

💡 ヒント: 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、VNCビューアをインストールする必要があります。詳細については、14.1項「`vncviewer`クライアント」を参照してください。または、JavaScript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。

永続的VNCセッションには次の2つのタイプがあります。

- `vncserver`を使用して開始されたVNCセッション
- `vncmanager`を使用して開始されたVNCセッション

14.4.1 vncserverを使用して開始されたVNCセッション

このタイプの永続的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 14.2: vncserverを使用した永続的VNCセッションの開始

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、『Security and Hardening Guide』、第23章「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. 1024x768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 -alwaysshared -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。その他の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vncserver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セッションをキャンセルするには、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VNC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キャンセルしたい場合は、VNCサーバでシェルを開き、キャンセルしたい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キャンセルします: **vncserver -kill :1**

14.4.1.1 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 Desktop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には、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。

14.4.2 vncmanagerを使用して開始されたVNCセッション

手順 14.3: 永続的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. セッション管理機能付きのリモート管理を許可するをアクティブにします。
3. WebブラウザウィンドウでVNCセッションにアクセスする場合は、Web ブラウザを利用したアクセスを有効にするをアクティブにします。

4. 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。
5. 次へをクリックすると設定が確定し、
6. 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。



ヒント: ディスプレイマネージャの再起動

YaSTはディスプレイマネージャの設定を変更します。現在のグラフィカルセッションからログアウトし、ディスプレイマネージャを再起動して変更を有効にする必要があります。

14.4.2.1 永続的VNCセッションを設定する

手順14.3「永続的VNCセッションの有効化」で説明したVNCセッション管理を有効にすると、通常、**vncviewer**やRemminaなどの好みのVNCビューアでリモートセッションに接続できます。ログイン画面が表示されます。ログインすると、デスクトップ環境のシステムトレイに「VNC」アイコンが表示されます。アイコンをクリックすると、VNCセッションウィンドウが開きます。表示されない場合、またはデスクトップ環境がシステムトレイのアイコンをサポートしていない場合は、手動で**vncmanager-controller**を実行してください。



図 14.8: VNCセッション設定

VNCセッションの動作に影響するいくつかの設定があります。

Non-persistent, private (非永続的、プライベート)

これは一時的セッションに相当します。このセッションは他のユーザに表示されず、セッションを切断すると終了します。詳細については、[14.3項「VNCサーバでのワンタイムセッションの設定」](#)を参照してください。

Persistent, visible (永続的、可視)

このセッションは他のユーザに表示され、セッションを切断しても実行され続けます。

セッション名

ここで、永続的セッションの名前を指定して、再接続時に簡単に識別できるようにすることができます。

No password required (パスワード不要)

セッションに、ユーザの資格情報でログインすることなく、自由にアクセス可能です。

Require user login (ユーザログインが必要)

セッションにアクセスするには、有効なユーザ名とパスワードでログインする必要があります。Allowed users (許可されたユーザ) テキストボックスに有効なユーザ名を一覧表示します。

Allow one client at time (同時に1つのクライアントを許可)

同時に複数のユーザがセッションに参加しないようにします。

Allow multiple clients at time (同時に複数のクライアントを許可)

複数のユーザが永続的セッションに同時に参加できるようにします。リモートプレゼンテーションやトレーニングセッションに便利です。

OKをクリックして、確定します。

14.4.2.2 永続的VNCセッションへの参加

14.4.2.1項「永続的VNCセッションを設定する」で説明した永続的VNCセッションを設定した後、VNCビューアでそのセッションに参加することができます。VNCクライアントからサーバに接続すると、新しいセッションを作成するか、既存のセッションに参加するかを選択するよう求めるプロンプトが表示されます。

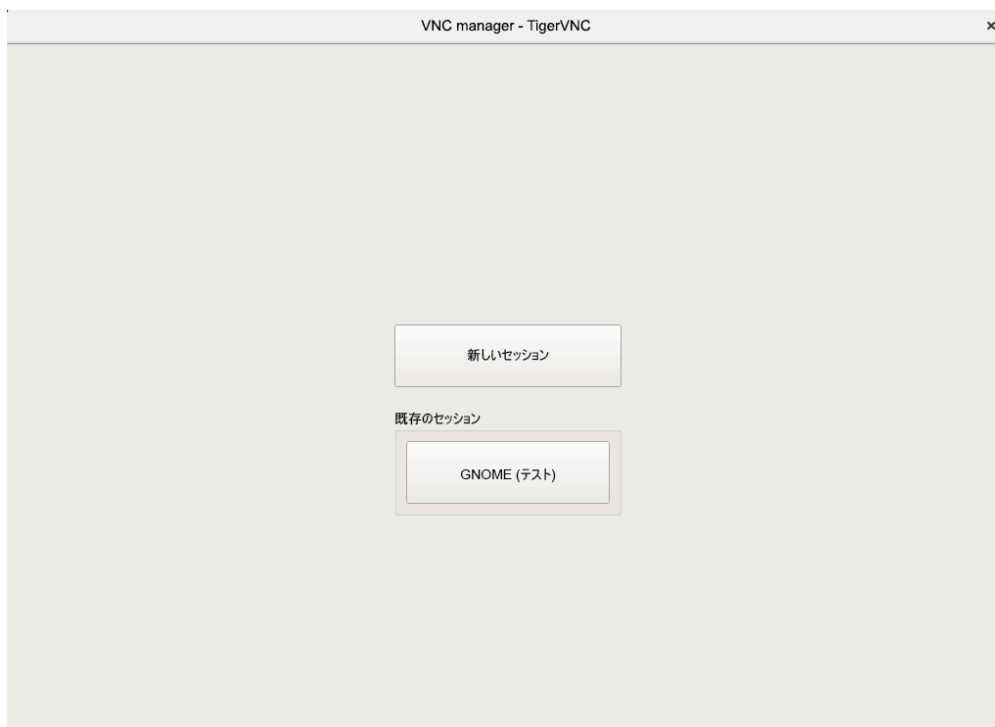


図 14.9: 永続的VNCセッションへの参加

既存のセッションの名前をクリックすると、永続的セッションの設定に応じて、ログインアカウント情報の入力を求められることがあります。

14.5 VNCサーバでの暗号化の設定

VNCサーバが正しく設定されている場合、VNCサーバとクライアント間の通信はすべて暗号化されます。セッションの開始時に認証が行われ、実際のデータ転送はその後に開始されます。一時的VNCセッションか永続的VNCセッションにかかわらず、セキュリティオプションは、`server_args`行にある`/usr/bin/Xvnc`コマンドの`-securitytypes`パラメータを介して設定されます。`-securitytypes`パラメータでは、認証方法と暗号化の両方を選択します。次のオプションがあります。

認証

None、TLSNone、x509None

認証なし。

VncAuth、TLSVnc、x509Vnc

カスタムパスワードを使用する認証。

Plain、TLSPlain、x509Plain

PAMを使用してユーザのパスワードを検証する認証。

暗号化

None、vncAuth、plain

暗号化なし。

TLSNone、TLSVnc、TLSPlain

匿名のTLS暗号化。すべてが暗号化されますが、リモートホストの検証は行われません。したがって、受動的攻撃からは保護されますが、中間者攻撃からは保護されません。

x509None、x509Vnc、x509Plain

証明書によるTLS暗号化。自己署名証明書を使用する場合、初回接続時に証明書を検証するよう要求されます。以降の接続では、証明書が変更された場合にのみ警告が表示されます。したがって、初回接続時には中間者攻撃以外のすべての攻撃から保護されます (SSHの一般的な使用方法と同様)。マシン名に一致する認証局によって署名された証明書を使用すると、完全なセキュリティを実現できます (HTTPSの一般的な使用方法と同様)。



ヒント

一部のVNCクライアントは、デフォルトの自己署名証明書を使用した安全な接続の確立を拒否します。たとえば、Vinagreクライアントは、GnuTLSグローバル信頼ストアに対して証明書を検証し、証明書が自己署名である場合には検証に失敗

します。この場合、x509以外の暗号化方式を使用するか、VNCサーバに対して正しく署名された証明書を生成し、これをクライアントのシステム信頼ストアにインポートします。



ヒント: 署名とキーのパス

X509ベースの暗号化では、-X509Certオプションと-X509Keyオプションで、X509証明書とキーのパスを指定する必要があります。

複数のセキュリティタイプをカンマで区切って選択した場合、クライアントとサーバの両方でサポートおよび許可されているセキュリティタイプが使用されます。この方法により、サーバ上で日和見暗号化を設定できます。これは、暗号化をサポートしないVNCクライアントをサポートする必要がある場合に便利です。

暗号化が有効であることがわかっているサーバに接続する場合、クライアント側で、許可されているセキュリティタイプを指定してダウングレード攻撃を防止することもできます(ただし、この場合、vncviewerに「Connection not encrypted! (接続が暗号化されていません)」という警告メッセージが表示されます)。

14.6 Waylandとの互換性

リモート管理(VNC)機能は、X11を使用し、Waylandが有効になっていると画面が空白になる場合があります。ディスプレイマネージャは、WaylandではなくX11を使用するように設定する必要があります。gdmの場合、/etc/gdm/custom.confを編集します。[daemon]セクションで、WaylandEnable=falseを設定ファイルに追加します。ログインするとき、ユーザは、X11互換セッションも選択する必要があります。GNOMEのWaylandオプションを削除したい場合、gnome-session-waylandパッケージを削除し、ロックすることができます。

15 rsyncによるファイルのコピー

現在の通常のユーザは、複数のコンピュータ(家庭用および職場用のマシン、ラップトップ、スマートフォン、またはタブレット)を持っています。このため、複数のデバイス間でファイルとドキュメントを同期させることがますます重要になっています。



警告: データ損失の危険

同期ツールの使用を開始する前に、その特徴や機能を十分に理解しておく必要があります。重要なファイルは必ずバックアップしてください。

15.1 概念の概要

低速なネットワーク接続で大量のデータを同期するために、rsyncは、ファイル内の変更のみを転送して信頼性を高めています。この処理は、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。

変更の検出には特定の処理能力が要求されます。そのため、両側のマシンにRAMなどのリソースが十分あることを確認してください。

rsyncが特に役立つのは、わずかな変更しかない大量のデータを定期的に転送する必要がある場合です。多くの場合、バックアップの操作がこれに該当します。また、rsyncは、Webサーバのディレクトリツリー全体を格納するステージングサーバをDMZ内のWebサーバにミラーリングする場合にも便利です。

その名前に反して、rsyncは同期ツールではありません。データを一度に一方向にのみコピーするツールです。その逆にはコピーせず、コピーすることもできません。コピー元とコピー先の両方を同期できる双方向ツールが必要な場合は、Csyncを使用してください。

15.2 基本的な構文

rsyncは、次の基本的な構文を持つコマンドラインツールです。

```
rsync [OPTION] SOURCE [SOURCE]... DEST
```

アクセスパーミッションと書き込みパーミッションがあれば、ローカルマシンでもリモートマシンでも使用できます。複数のSOURCEエントリを指定できます。SOURCEおよびDESTのブレースホルダには、パス、URL、またはその両方を指定できます。

rsyncで最もよく使われるオプションは次のとおりです。

-v

より詳細なテキストを出力します。

-a

アーカイブモード。ファイルを再帰的にコピーし、タイムスタンプ、ユーザ/グループの所有権、ファイルパーミッション、およびシンボリックリンクを保持します。

-z

転送データを圧縮します。



注記: 末尾のスラッシュの数

rsyncを操作する場合は、特に末尾のスラッシュに注意する必要があります。ディレクトリの後に末尾のスラッシュがある場合、そのスラッシュはディレクトリの「内容」を示します。末尾のスラッシュがない場合は、「ディレクトリそのもの」を表します。

15.3 ファイルとディレクトリのローカルでのコピー

次の説明は、現在のユーザがディレクトリ/var/backupに対する書き込みパーミッションを持っていることを想定しています。1つのファイルをマシン上のディレクトリから別のパスにコピーするには、次のコマンドを使用します。

```
> rsync -avz backup.tar.xz /var/backup/
```

ファイルbackup.tar.xzが/var/backup/にコピーされ、絶対パスは/var/backup/backup.tar.xzになります。

/var/backup/ディレクトリの後に「末尾のスラッシュ」を追加するのを忘れないください。スラッシュを挿入しない場合、ファイルbackup.tar.xzは、ディレクトリ/var/backup/の「中ではなく」、/var/backup(ファイル)にコピーされます。

ディレクトリをコピーする場合も、1つのファイルをコピーする場合と同様です。次の例では、ディレクトリtux/とその内容をディレクトリ/var/backup/にコピーします。

```
> rsync -avz tux /var/backup/
```

コピーは絶対パス/var/backup/tux/にあります。

15.4 ファイルとディレクトリのリモートでのコピー

両方のマシンにrsyncツールが必要です。リモートディレクトリ間でファイルをコピーするには、IPアドレスまたはドメイン名が必要です。ローカルマシンとリモートマシンの現在のユーザ名が同じ場合、ユーザ名は省略できます。

ファイルfile.tar.xzをローカルホストからリモートホスト192.168.1.1に同じユーザ(ローカルとリモート)でコピーするには、次のコマンドを使用します。

```
> rsync -avz file.tar.xz tux@192.168.1.1:
```

好みに応じて、次のコマンドを使用することもできます。処理結果は同じです。

```
> rsync -avz file.tar.xz 192.168.1.1:~  
> rsync -avz file.tar.xz 192.168.1.1:/home/tux
```

標準設定では、すべての場合に、リモートユーザのパスフレーズの入力を求めるプロンプトが表示されます。このコマンドは、file.tar.xzをユーザtuxのホームディレクトリ(通常は/home/tux)にコピーします。

ディレクトリをリモートでコピーする場合も、ローカルでコピーする場合と同様です。次の例では、ディレクトリtux/とその内容をホスト192.168.1.1のリモートディレクトリ/var/backup/にコピーします。

```
> rsync -avz tux 192.168.1.1:/var/backup/
```

ホスト192.168.1.1で書き込みパーミッションを持っていると想定すると、コピーは絶対パス/var/backup/tuxにあります。

15.5 rsyncサーバの設定と使用

rsyncは、デフォルトポート873で着信接続をリスンするデーモン(rsyncd)として実行できます。このデーモンは「コピーターゲット」を受信できます。

次に、jupiter上に「バックアップ」ターゲットを持つrsyncサーバを作成する方法を説明します。このターゲットを使用してバックアップを保存できます。rsyncサーバを作成するには、以下の手順を実行します。

手順 15.1: RSYNCサーバの設定

1. jupiterで、すべてのバックアップファイルを保存するディレクトリを作成します。この例では、`/var/backup`を使用します。

```
# mkdir /var/backup
```

2. 所有権を指定します。この場合、ディレクトリはグループusersのユーザtuxによって所有されます。

```
# chown tux.users /var/backup
```

3. rsyncdデーモンを設定します。

設定ファイルを、メインファイルと、バックアップターゲットを格納する特定の「モジュール」に分割します。こうすることで、後で他のターゲットを簡単に追加できます。グローバル値は`/etc/rsyncd.d/*.inc`ファイルに保存できます。一方、モジュールは`/etc/rsyncd.d/*.conf`ファイルに配置します。

- a. ディレクトリ`/etc/rsyncd.d/`を作成します。

```
# mkdir /etc/rsyncd.d/
```

- b. メイン設定ファイル`/etc/rsyncd.conf`に、次の行を追加します。

```
# rsyncd.conf main configuration file
log file = /var/log/rsync.log
pid file = /var/lock/rsync.lock

&merge /etc/rsyncd.d ①
&include /etc/rsyncd.d ②
```

- ① グローバル値を`/etc/rsyncd.d/*.inc`ファイルからメイン設定ファイルにマージします。
 - ② モジュール(またはターゲット)を`/etc/rsyncd.d/*.conf`ファイルからロードします。これらのファイルにはグローバル値への参照を含めないでください。
- c. 次の行を使用して、ファイル`/etc/rsyncd.d/backup.conf`内にモジュール(バックアップターゲット)を作成します。

```
# backup.conf: backup module
[backup] ①
  uid = tux ②
  gid = users ②
  path = /var/backup ③
```



```
auth users = tux ④
secrets file = /etc/rsyncd.secrets ⑤
comment = Our backup target
```

- ① 「バックアップ」ターゲット。好きな名前を使用できます。ただし、ターゲットには用途に応じた名前を付け、*.confファイルと同じ名前を使用することをお勧めします。
 - ② ファイル転送の実行時に使用されるユーザ名またはグループ名を指定します。
 - ③ バックアップを保存するパスを定義します(ステップ 1から)。
 - ④ 許可されているユーザのカンマ区切りリストを指定します。最も単純な形式では、このモジュールへの接続を許可されているユーザ名が含まれます。この例では、ユーザ tux のみが許可されています。
 - ⑤ ユーザ名とプレーンパスワードが記述された行が含まれるファイルのパスを指定します。
- d. 次の内容で /etc/rsyncd.secrets ファイルを作成し、PASSPHRASE を置き換えます。

```
# user:passwd
tux:PASSPHRASE
```

- e. root のみがこのファイルを読み込めるようにします。

```
# chmod 0600 /etc/rsyncd.secrets
```

4. 次のコマンドを使用して、rsyncd デーモンを起動して有効にします。

```
# systemctl enable rsyncd
# systemctl start rsyncd
```

5. rsync サーバにアクセスできるかどうかをテストします。

```
> rsync jupiter::
```

次のような応答が表示されます。

```
backup          Our backup target
```

異なる場合は、設定ファイル、ファイアウォール設定、およびネットワーク設定を確認してください。

上記の手順でrsyncサーバが作成されたので、このサーバを使用してバックアップを保存できます。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。これは、転送のデバッグに便利です。

バックアップターゲットの内容を一覧にするには、次のコマンドを使用します。

```
> rsync -avz jupiter::backup
```

このコマンドを入力すると、サーバのディレクトリ/var/backupにあるファイルがすべて一覧表示されます。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、ソース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、次のコマンドは、現在のディレクトリをrsyncバックアップサーバにコピーします。

```
> rsync -avz . jupiter::backup
```

デフォルトでは、rsyncは実行時にファイルとディレクトリを削除しません。削除を有効にするには、追加オプション--deleteを記述する必要があります。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。

15.6 詳細情報

Csync

双方向ファイル同期ツール。 <https://csync.org/> を参照してください。

RSnapshot

増分バックアップを作成します。 <https://rsnapshot.org/> を参照してください。

Unison

CSyncに似たファイル同期ツールですが、グラフィカルインタフェースを備えています。 <https://www.seas.upenn.edu/~bcpierce/unison/> を参照してください。

Rear

ディザスタリカバリフレームワーク。Administration Guide of the SUSE Linux Enterprise High Availability Extension, chapter Disaster Recovery with Rear (Relax-and-Recover) (<https://documentation.suse.com/sle-ha-15/html/SLE-HA-all/cha-ha-rear.html>) を参照してください。

II Linuxシステムのブート

- 16 ブートプロセスの概要 224
- 17 UEFI (Unified Extensible Firmware Interface) 233
- 18 ブートローダGRUB 2 242
- 19 systemdデーモン 261

16 ブートプロセスの概要

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。マシンのアーキテクチャに依存する、ファームウェアとハードウェアの初期化プロセスの後、ブートローダGRUB 2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設定をブートする一連の「ターゲット」を提供します。

16.1 用語集

この章ではあいまいに解釈される可能性のある用語を使用します。ここでの使用方法を理解するには、以下の定義を読んでください。

init

一般的に「init」という名前が付くのは、次の2つの異なるプロセスです:

- ルートファイルシステムをマウントするinitramfsプロセス
- 実際のルートファイルシステムから実行される他のすべてのプロセスを開始するオペレーティングシステムプロセス

両方のケースで、systemdプログラムがこのタスクを担当します。ルートファイルシステムをマウントするために、まずinitramfsから実行されます。成功したら、最初のプロセスとしてルートファイルシステムから再実行されます。これら2つのsystemdプロセスの混同を避けるため、まず「init on initramfs」として最初のプロセスを実行し、「systemd」として2番目のプロセスを実行します。

initrd/initramfs

initrd (最初のRAMディスク)は、カーネルによってロードされ、一時ルートファイルシステムとして/dev/ramからマウントされるルートファイルシステムイメージを含むイメージファイルです。ファイルシステムのマウントには、ファイルシステムドライバが必要です。

カーネル2.6.13以降、initrdは、ファイルシステムドライバのマウントが必要ない、initramfs (最初のRAMファイルシステム)で置き換えられました。SUSE Linux Enterprise Desktopは、initramfsを排他的に使用します。ただし、initramfsは/boot/initrdとして格納されるため、多くの場合「initrd」と呼ばれます。この章では、initramfsという名前を排他的に使用します。

16.2 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。

1. 16.2.1項「初期化とブートローダの段階」
2. 16.2.2項「カーネルの段階」
3. 16.2.3項「initramfs上のinit段階」
4. 16.2.4項「systemd段階」

16.2.1 初期化とブートローダの段階

初期化段階中に、マシンのハードウェアが設定され、デバイスが準備されます。このプロセスはハードウェアアーキテクチャ間で大きく異なります。

SUSE Linux Enterprise Desktopは、すべてのアーキテクチャでブートローダGRUB 2を使用します。アーキテクチャおよびファームウェアによって、GRUB 2ブートローダの起動は、マルチステップのプロセスとなる可能性があります。ブートローダの目的は、カーネルおよび、RAMベースの初期ファイルシステム(initramfs)をロードすることです。GRUB 2についての詳細については、[第18章「ブートローダGRUB 2」](#)を参照してください。

16.2.1.1 AArch64およびAMD64/Intel 64での初期化とブートローダ段階

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。ブートメディアとそのジオメトリが認識されると、システム制御がBIOS/UEFIからブートローダに移ります。

従来のBIOSが備わっているマシンでは、ブートディスクの先頭の512バイト物理データセクタ(マスタブートレコード、MBR)のコードのみをロードできます。最小のGRUB 2のみがMBRに適合します。その唯一の目的は、MBRと最初のパーティション(MBRパーティションテーブル)の間のギャップから、またはBIOSブートパーティション(GPTパーティションテーブル)からファイルシステムドライバを含むGRUB 2コアイメージをロードすることです。このイメージにはファイルシステムドライバが含まれるため、ルートファイルシステム上にある/bootにアクセスできます。/bootには、カーネルとinitramfsイメージとともに、GRUB 2コアの追加のモジュールも含まれます。このパーティションにアクセスすると、GRUB 2はカーネルをロードし、initramfsはメモリにイメージを作成し、カーネルに制御を移します。

BIOSシステムが、暗号化された`/boot`パーティションを含む暗号化されたファイルシステムからブートする場合、復号化のパスワードを2度入力する必要があります。最初にGRUB 2によって`/boot`を復号化した後で、`systemd`用に暗号化されたボリュームをマウントする必要があります。

UEFIを搭載したマシンでは、従来のBIOSを搭載するマシンよりも、ブートプロセスははるかに簡単です。ファームウェアは、GPTパーティションテーブルを備えたディスクのFATでフォーマットされたシステムパーティションから読み取ることができます。このEFIシステムパーティション(`/boot/efi`としてマウントされる実行中のシステム)は、ファームウェアによって直接ロードされ実行される完全に装備されたGRUB 2をホストする十分なスペースを保持します。

BIOS/UEFI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。その後、システムはPXEを介してブートできます。BIOS/UEFIはブートローダとして動作します。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。

16.2.1.2 IBM Zでの初期化とブートローダ段階

IBM Zでは、ブートプロセスは、**zipl** (zイニシャルプログラムロード)と呼ばれるブートローダによって初期化される必要があります。**zipl**はさまざまなファイルシステムからの読み込みをサポートしますが、SLEデフォルトファイルシステム(Btrfs)またはスナップショットからのブートはサポートしません。したがって、SUSE Linux Enterprise Desktopはブート時に完全なBtrfsサポートを保証する2段階のブートプロセスを使用します。

1. **zipl**は、Ext2、Ext3、Ext4、またはXFSファイルシステムでフォーマットできるパーティション`/boot/zipl`からブートします。このパーティションには、メモリにロードされる最小のカーネルとinitramfsが含まれます。initramfsには、Btrfsドライバ(その他の間)およびブートローダGRUB 2が含まれます。カーネルは`initgrub`パラメータで開始され、GRUB 2を開始するように指示されます。
2. カーネルはルートファイルシステムをマウントするため、`/boot`にアクセス可能になります。これでGRUB 2がinitramfsから開始されます。GRUBは`/boot/grub2/grub.cfg`からその設定を読み込み、`/boot`から最後のカーネルとinitramfsをロードします。これで新しいカーネルがKexecを介してロードされます。

16.2.2 カーネルの段階

ブートローダがシステム制御に渡されると、ブートプロセスはすべてのアーキテクチャで同じになります。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、カーネルが引き継ぎます。

カーネルはメモリ管理を設定し、CPUタイプとその機能を検出した後で、ハードウェアを初期化し、initramfsでロードされたメモリから一時ルートファイルシステムをマウントします。

16.2.2.1 initramfsファイル

initramfs(初期RAMファイルシステム)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。/boot/initrdにあります。dracutというツールで作成することもできます。詳細については、man 8 dracutを参照してください。

initramfsは、実際のルートファイルシステムがマウントされる前に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。これは、systemd unit systemd-udev-trigger.serviceで実行されます。

16.2.2.1.1 initramfsの再生成

initramfsには、ドライバが含まれるため、そのドライバのいずれかの新しいバージョンが利用可能になるとすぐにinitramfsをアップデートする必要があります。これは、ドライバアップデートを含むパッケージをインストールするときに自動的に実行されます。YaSTまたはzypperは、initramfsを生成するコマンドの出力を表示することで、これについて通知します。ただし、initramfsを手動で再生成する必要がある場合があります。

- ハードウェアの変更によるドライバの追加
- RAIDまたはLVMへのシステムディレクトリの移動
- ルートファイルシステムを含むLVMグループまたはBtrfs RAIDへのディスクの追加
- カーネル変数の変更

ハードウェアの変更によるドライバの追加

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、ブート時にそのハードウェア用の他のドライバがカーネル内に必須の場合には、initramfsファイルを更新する必要があります。

/etc/dracut.conf.d/10-DRIVER.confを開くか作成し、次の行を追加してください(行頭の空白に注意)。

```
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て指定します。

```
force_drivers+=" DRIVER1 DRIVER2 "
```

手順16.1「initramfsの生成」に従って手順を進めます。

RAIDまたはLVMへのシステムディレクトリの移動

スワップファイル、または実行中のシステムの/usrなどのシステムディレクトリをRAIDまたは論理ボリュームに移動するときには常に、ソフトウェアRAIDまたはLVMドライバのサポートを含むinitramfsを作成する必要があります。

これを作成するには、/etc/fstabで各エントリを作成し、新しいエントリ(たとえばmount -aおよび/またはswapon -a)をマウントします。

手順16.1「initramfsの生成」に従って手順を進めます。

ルートファイルシステムを含むLVMグループまたはBtrfs RAIDへのディスクの追加

ルートファイルシステムを含む論理ボリュームグループまたはBtrfs RAIDにディスクを追加(または削除)する際には常に、大きくなったボリュームのサポートを含むinitramfsを作成する必要があります。手順16.1「initramfsの生成」の指示に従います。

手順16.1「initramfsの生成」に従って手順を進めます。

カーネル変数の変更

関連するファイル(`/etc/sysctl.conf`または`/etc/sysctl.d/*.conf`)を編集して、**sysctl**インタフェースでカーネル変数の値を変更した場合、次にシステムを再起動したときに変更内容が失われます。実行時に**sysctl --system**を使用して値をロードしても、変更内容は `initramfs` ファイルに保存されません。[手順16.1「initramfsの生成」](#)の説明に従って手順を進め、アップデートする必要があります。

手順 16.1: INITRAMFSの生成

次の手順のすべてのコマンドを root ユーザとして実行する必要があることに注意してください。

1. `/boot`ディレクトリを入力します。

```
# cd /boot
```

2. **dracut**を使用して新しい`initramfs`ファイルを生成し、`MY_INITRAMFS`を任意のファイル名に置き換えます。

```
# dracut MY_INITRAMFS
```

または、**dracut -f** `FILENAME`を実行して、既存のinitファイルを置き換えます。

3. (以前のステップで**dracut -f**を実行した場合は、このステップはスキップします)。以前のステップで作成した`initramfs`ファイルから`initrd`へのシンボリックリンクを作成します。

```
# ln -sf MY_INITRAMFS initrd
```

4. IBM Zアーキテクチャで、**grub2-install**を補足的に実行します。

16.2.3 initramfs上のinit段階

`initramfs`からカーネルによってマウントされた一時ルートファイルシステムには、(以下の`initramfs`上の`init`と呼ばれる)実行可能な`systemd`が含まれます。[16.1項「用語集」](#)も参照してください。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。

`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

カーネルはロードされたモジュールに応じて、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。

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、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。SUSE Linux Enterprise Desktopは、プライマリターゲットを使用できない場合の、セカンダリiSCSIターゲットからのブートをサポートしています。



注記: マウントできなかった場合の処理

ルートファイルシステムをブート環境内からマウントできなかった場合は、ブートを続行する前にルートファイルシステムを確認して修復しておく必要があります。Ext3ファイルシステムおよびExt4ファイルシステムでは、ファイルシステムチェッカが自動的に起動されます。XFSファイルシステムおよびBtrfsファイルシステムでは修復プロセスが自動化されていないため、ファイルシステムを修復するために使用できるオプションに関する情報が表示されます。ファイルシステムが正常に修復された場合、ブート環境を終了すると、システムはルートファイルシステムのマウントを再試行します。成功した場合、ブートは通常どおり続行されます。

16.2.3.1 インストールプロセスのinitramfs上のinit段階

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。インストールシステムはinitramfsからsystemdを起動せず、これらのタスクがlinuxrcで実行されることにも注意してください。

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。YaSTインストーラは、RAMファイルシステムで実行され、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を必要とします。

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

16.2.2.1項「initramfsファイル」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構成で利用できる最小限のドライバセットで開始されます。AArch64、POWER、およびAMD64/Intel 64マシンでは、linuxrcは、ハードウェア構成に適したドライバセットを判断する、初期ハードウェアスキャンプロセスを開始します。IBM Zでは、ドライバのリストおよびそのパラメータは、linuxrcまたはparmfileなどを介して提供される必要があります。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、19.6.4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、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、linuxrc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、linuxrc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。

16.2.4 systemd段階

「実際の」ルートファイルシステムが見つかり、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。systemdはLinuxのシステムおよびサービスマネージャです。PID 1として起動する親プロセスで、ユーザスペースサービスを起動して維持するinitシステムとして機能します。詳細については第19章「systemdデーモン」を参照してください。

17 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との間のインターフェースです。

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した。たとえば、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ブレイズ環境。
- 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(Compatibility 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 Desktopにどのように実装されているかを示すヒントです。

17.1 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 Desktop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。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、特定の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。

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場合、他の証明書を使用できます。

17.1.1 SUSE Linux Enterprise Desktop上の実装

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は、UEFI/x86_64インストール環境ではデフォルトで有効になっています。Enable Secure Boot Support(セキュアブートサポートの有効化)オプションは、ブートローダの設定ダイアログのBoot Code Options(ブートコードオプション)タブにあります。ファームウェアでセキュアブートが有効になっている場合のブート、および無効になっている場合のブートもサポートします。

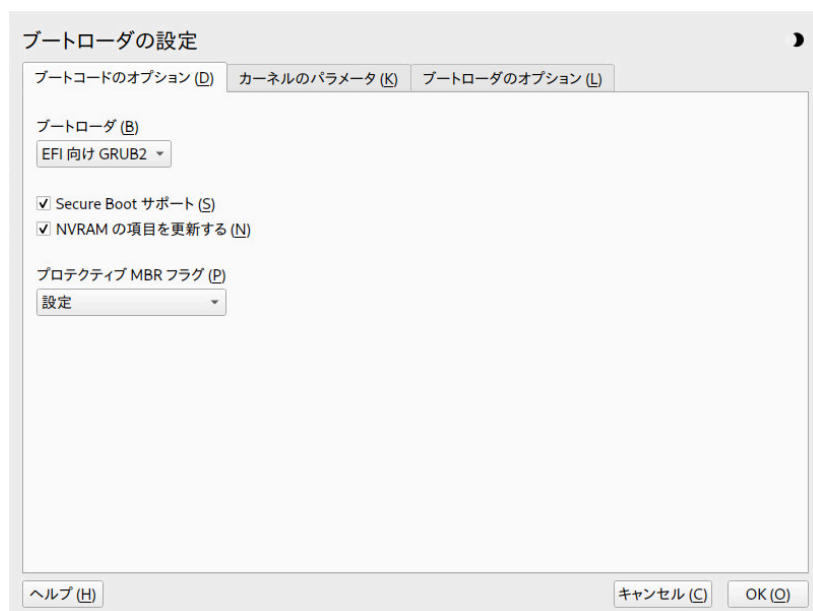


図 17.1: セキュアブートサポート

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、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。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。

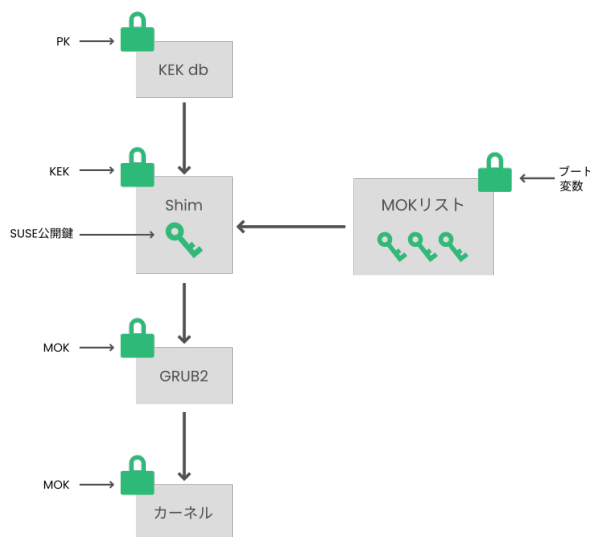


図 17.2: UEFIのセキュアブートプロセス

実装層で、SUSEは、デフォルトでインストールされている`shim`ローダを使用します。法的な問題を回避するスマートなソリューションであり、証明書と署名に関する手順を大きく簡素化します。`shim`ローダの処理は、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つ目はいわゆる「認証された変数」で、ブートプロセス(いわゆるブートサービス環境)と実行中のOSの両方からアップデートできます。これは、変数の新しい値が、その変数の古い値が署名されたときと同じキーで署名されている場合にのみ実行できます。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。

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shim (SUSEとMicrosoftが署名した小型でシンプルなEFIブートローダ)から始まります。

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。

17.1.2 Machine Owner Key(マシン所有者キー、MOK)

ブートプロセスの一部である特定のカーネル、ドライバ、または他のコンポーネントを置き換えるには、マシン所有者キー(MOK)を使用する必要があります。mokutilツールはMOKを管理するのに役立ちます。

`mokutil`を使用してMOK登録要求を作成できます。要求は、`MokNew`と呼ばれるUEFIラシタイム(RT)変数に保存されます。次のブート時に、`shim`ブートローダは`MokNew`を検出し、`MokManager`をロードします。これにより、いくつかのオプションが表示されます。Enroll key from disk (ディスクからキーを登録)およびEnroll hash from disk (ディスクからハッシュを登録)オプションを使用して、`MoList`にキーを追加することができます。Enroll MOK (MOKを登録)オプションを使用して、`MokNew`変数からキーをコピーします。

ディスクからのキーの登録は通常、シムが`grub2`のロードに失敗し、`MokManager`のロードにフォールバックしたときに行われます。`MokNew`はまだ存在しないため、UEFIパーティションでキーを検索するオプションがあります。

17.1.3 カスタムカーネルのブート

以下はhttps://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、https://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、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を手動で起動するに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. タイプ:

```
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。

17.1.4 Inbox以外のドライバの使用

セキュアブートを有効にしたインストールでは、Inbox以外のドライバ(SUSE Linux Enterprise Desktopに付属していないドライバ)の追加がサポートされません。SolidDriver/PLDPで使用する署名キーは、デフォルトでは信頼されていません。

セキュアブートを有効にしたインストールでは、サードパーティドライバを2つの方法でインストールできます。いずれの方法でも以下を行います。

- インストール前にファームウェア/システム管理ツールを使用して、必要なキーをファームウェアデータベースに追加します。このオプションは、使用している特定のハードウェアによって異なります。詳細については、ハードウェアベンダーに問い合わせてください。
- <https://drivers.suse.com/> またはハードウェアベンダーから入手したブート可能なドライバISOを使用して、初回ブート時に必要なキーをMOKリストに登録します。

ブート可能なドライバISOを使用してドライバキーをMOKリストに登録するには、次の手順に従います。

1. 空のCD/DVDメディアに上記のISOイメージを書き込みます。
2. この新しいCD/DVDメディアを使用してインストールを開始します。その際には、標準のインストールメディア、またはネットワークインストールサーバへのURLを用意しておきます。
ネットワークインストールを行う場合、ブートコマンドラインでinstall=オプションを使用して、ネットワークインストールソースのURLを入力します。

光学メディアからインストールする場合、インストーラが最初にドライバキットからブートされた後、製品の最初のインストールディスクを挿入するように要求されます。

3. アップデートされたドライバを含むinitrdが、インストールに使用されます。

詳細については、https://drivers.suse.com/doc/Usage/Secure_Boot_Certificate.html を参照してください。

17.1.5 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ビデオモードのサポート。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。
- SUSE Linux Enterprise Server 15 SP3以降、KexecおよびKdumpはセキュアブートモードでサポートされる。

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特定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。
- 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`debugfs`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。

17.2 詳細情報

- <https://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)。
 - <https://www.suse.com/c/uefi-secure-boot-plan/> 
 - <https://www.suse.com/c/uefi-secure-boot-overview/> 
 - <https://www.suse.com/c/uefi-secure-boot-details/> 
- <https://en.opensuse.org/openSUSE:UEFI>  —UEFIとopenSUSEに関するページです。

18 ブートローダGRUB 2

この章では、SUSE® Linux Enterprise Desktop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 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 Desktop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16章「ブートプロセスの概要」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17章「UEFI (Unified Extensible Firmware Interface)」を参照してください。

18.1 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みます。
- 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。

18.2 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は編集しないでください。コマンドgrub2-mkconfig -o /boot/grub2/grub.cfgによって自動的に生成されます。

/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。SUSE Linux Enterprise Desktop 12 SP2からは、**grub-once**を使用する場合も、これらのエントリが解析されます。

/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

このディレクトリのスクリプトは、コマンド**grub2-mkconfig -o /boot/grub2/grub.cfg**の実行中に読み込まれます。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、ブートローダタイプや、UEFIセキュアブートサポートを有効にするかどうかなどの特定の基本的な設定を保持します。

/boot/grub2/x86_64-efi、/boot/grub2/power-ieee1275

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、複数の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 -o /boot/grub2/grub.cfg**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、YaSTはこのコマンドを自動的に実行するため、この作業は必要ありません。

18.2.1 /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ロードします。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す。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、ファイル/etc/default/grub、およびコマンド**grub2-mkconfig -o /boot/grub2/grub.cfg**の実行中に/etc/grub.d/ディレクトリで見つかった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、[18.3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。

18.2.2 /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を編集した後で、**grub2-mkconfig -o /boot/grub2/grub.cfg**を使用してメインの設定ファイルをアップデートします。



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-once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。grub2-editenv listは、次のブートエントリをリストします。

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す。デフォルトは「auto」で、優先解像度の選択を試みます。GRUB 2のコマンドラインで「videoinfo」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なければならず、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/内にあるユーザ独自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Snapperのスナップショットの状態に直接ブートできます。詳細については、[10.3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。

すべてのオプションのリストについては、[GNU GRUB manual \(http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration\)](http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration) を参照してください。

18.2.3 /etc/grub.d内のスクリプト

このディレクトリのスクリプトは、コマンド **grub2-mkconfig -o /boot/grub2/grub.cfg** の実行中に読み込まれます。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。



ヒント: grub.cfgの永続的なカスタムコンテンツ

grub2-mkconfig を実行するたびに `/boot/grub2/grub.cfg` が再コンパイルされるため、カスタムコンテンツはすべて失われます。**grub2-mkconfig** の実行後にカスタムコンテンツを失わずに行を `/boot/grub2/grub.cfg` に直接挿入するには、次の行の間に挿入します。

```
### BEGIN /etc/grub.d/90_persistent ###
```

および

```
### END /etc/grub.d/90_persistent ###
```

`90_persistent` スクリプトにより、このようなコンテンツが確実に保存されます。最も重要なスクリプトのリストを以下に示します。

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macOS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。最初の`exec tail -n +3 $0`の部分は変更しないようにしてください。

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。



ヒント: `/boot/grub2/custom.cfg`

`/boot/grub2/custom.cfg`を作成してコンテンツを入力すると、ブート時に40_customの直後に自動的に`/boot/grub2/grub.cfg`に組み込まれます。

18.2.4 BIOSドライブとLinuxデバイスのマッピング

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では、この処理で十分です。

ただし、GRUB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1から始まっていましたが、GRUB 2では0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
```

18.2.5 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。



```
GNU GRUB version 2.04

set root='hd0,gpt2'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint='hd0,gpt2' 3c2\
51c37-7ebb-4aaa-a658-eca1e810198d
else
  search --no-floppy --fs-uuid --set=root 3c251c37-7ebb-4aaa-a65\
8-eca1e810198d
fi
echo      'Loading Linux 5.3.18-8-default ...'
linux     /boot/vmlinuz-5.3.18-8-default root=UUID=3c251c37-7\
ebb-4aaa-a658-eca1e810198d ${extra_cmdline} splash=silent resume=/dev/v\
da4 mitigations=auto quiet crashkernel=195M,high crashkernel=72M,low
echo      'Loading initial ramdisk ...'
initrd    /boot/initrd-5.3.18-8-default

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB
menu.
```

図 18.1: 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<https://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。『Deployment Guide』、第8章「Troubleshooting」、8.3項「Booting from installation media fails」、US keyboard layout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。ブートパラメータを追加するには、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。

18.2.6 ブートパスワードの設定

GRUB 2は、オペレーティングシステムのブート前でも、ファイルシステムにアクセスできるようにします。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: ブート時にパスワードが必要です

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。

```
> 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. メインの設定ファイルに変更をインポートするには、次を実行します。

```
> sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

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。詳細については、<https://www.gnu.org/software/grub/manual/grub.html#Security> を参照してください。

18.2.7 ブートメニューエントリへの許可されたアクセス

認可のレベルに応じてブートメニューエントリへのアクセスを許可するように GRUB 2 を設定できます。パスワードで保護された複数のユーザアカウントを設定し、それらのユーザアカウントに異なるメニューエントリへのアクセス権を割り当てることができます。GRUB 2 で認可を設定するには、次の手順に従います。

1. GRUB 2 で使用するユーザアカウントごとに 1 つのパスワードを作成して暗号化します。18.2.6 項「ブートパスワードの設定」で説明されているように、**grub2-mkpasswd-pbkdf2** コマンドを使用します。
2. `/etc/grub.d/10_linux` ファイルを削除します。これにより、デフォルトの GRUB 2 メニューエントリを出力できなくなります。
3. `/boot/grub2/custom.cfg` ファイルを編集して、カスタムメニューエントリを手動で追加します。次のテンプレートは、使用例に合わせて調整します。

```
set superusers=admin
password admin ADMIN_PASSWORD
password maintainer MAINTAINER_PASSWORD

menuentry 'Operational mode' {
    insmod ext2
    set root=hd0,1
```



```

echo 'Loading Linux ...'
linux /boot/vmlinuz root=/dev/vda1 $GRUB_CMDLINE_LINUX_DEFAULT $GRUB_CMDLINE_LINUX
mode=operation
echo 'Loading Initrd ...'
initrd /boot/initrd
}

menuentry 'Maintenance mode' --users maintainer {
    insmod ext2
    set root=hd0,1
    echo 'Loading Linux ...'
    linux /boot/vmlinuz root=/dev/vda1 $GRUB_CMDLINE_LINUX_DEFAULT $GRUB_CMDLINE_LINUX
    mode=maintenance
    echo 'Loading Initrd ...'
    initrd /boot/initrd
}

```

4. メインの設定ファイルに変更をインポートします。

```
> sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

これまでの例で:

- GRUB 2メニューにはOperational mode (動作モード)と保守モードの2つのエントリがあります。
- ユーザを指定しない場合は、両方のブートメニューエントリにアクセスできますが、GRUB 2コマンドラインにアクセスしたり、既存のメニューエントリを編集したりすることはできません。
- adminユーザは、GRUB 2コマンドラインにアクセスし、既存のメニューエントリを編集できます。
- maintenanceユーザは、リカバリメニュー項目を選択できます。

18.3 YaSTによるブートローダの設定

SUSE Linux Enterprise Desktopシステムでブートローダ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を標準モードとEFIモードのどちらで使用するかを選択することができます。

！ 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。

！ 重要: ブートローダの再インストール

ブートローダを再インストールするには、必ずYaSTで設定を変更して、その後で元に戻すという操作を実行します。たとえば、GRUB2-EFIを再インストールするには、一度GRUB2を選択して、すぐにGRUB2-EFIに戻します。

元に戻さない場合、ブートローダが完全には再インストールされない可能性があります。

📄 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。

18.3.1 ブートローダの場所とブートコードオプション

ブートローダのデフォルトの場所はパーティション設定によって異なり、マスタブートレコード(MBR)か、/パーティションのブートセクタです。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 18.1: 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

ディレクトリ/bootが含まれるディスクのMBRにブートローダをインストールします。通常は/にマウントされるディスクになりますが、/bootが異なるディスクの別個のパーティションにマウントされる場合は、そのディスクのMBRが使用されます。

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。

Custom Root Partition (カスタムル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、変更内容を適用します。



図 18.2: ブートコードオプション

ブートコードオプションタブには、以下の追加のオプションがあります。

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

/bootディレクトリを含むパーティションをアクティブにします。POWERシステムの場合、PRePパーティションをアクティブにします。古いBIOSおよび/またはレガシーオペレーティングシステムを使用しているシステムでは、非アクティブなパーティションからのブートに失敗する可能性があるため、このオプションを使用してください。このオプションをアクティブのままにしておくのが安全です。

MBRに汎用ブートコードを書き込む

MBRにカスタムの「非GRUB」コードが含まれている場合、このコードは、このオプションにより、汎用の、オペレーティングシステムに依存しないコードに置き換えられます。このオプションを無効にすると、システムが起動できなくなる可能性があります。

Enable Trusted Boot Support

信頼されたコンピューティング機能(TPM(Trusted Platform Module))をサポートするTrustedGRUB2を開始します。詳細については、<https://github.com/Sirrix-AG/TrustedGRUB2>を参照してください。

プロテクトティブMBRフラグセクションには、次のオプションが含まれています。

set

これは、従来のレガシBIOSブートに適しています。

remove

これは、UEFIブートに適しています。

変更禁止

これは通常、すでに機能しているシステムがある場合の最適な選択肢です。

多くの場合、YaSTがデフォルトで最適な選択肢となります。

18.3.2 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。リストの最初のディスクは、MBRからブートする場合にGRUB 2がインストールされる場所です。これは、デフォルトでSUSE Linux Enterprise Desktopがインストールされるディスクです。リストの残りは、GRUB 2のデバイスマップのヒントです(18.2.4項「BIOSドライブとLinuxデバイスのマッピング」を参照)。



警告: ブートできないシステム

デフォルト値は、通常、ほぼすべての展開で有効です。ディスクのブート順序を正しく変更しないと、次の再起動時にシステムをブートできなくなる可能性があります。たとえば、リスト内の最初のディスクがBIOSのブート順序の一部ではなく、リスト内の他のディスクに空のMBRがある場合などです。

手順 18.2: 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. ディスク順序の編集をクリックします。
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。

18.3.3 詳細オプションの設定

詳細なブートパラメータを設定するには、Boot Loader Options (ブートローダオプション) タブを使用します。

18.3.3.1 ブートローダオプションタブ

ブートローダの設定

ブートコードのオプション (D) カーネルのパラメータ (K) **ブートローダのオプション (L)**

タイムアウト (秒) (T)
8 ☐ ブート時にメニューを隠す (H)

既定のブートセクション (D)
SLES 15-SP4

☒ ブートローダをパスワードで保護する (E)
☒ 項目の修正のみを保護する (R)

GRUB2 ユーザ 'root' のパスワード (P) パスワードの再入力 (T):
.....

ヘルプ (H) キャンセル (C) OK (O)

図 18.3: ブートローダオプション

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。手動による環境設定の詳細は、18.2.6項「ブートパスワードの設定」を参照してください。このオプションを有効にすると、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ブートしません。ただし、GRUB 1の動作を希望する場合は、さらに項目の修正のみを保護するを有効にします。この設定では、誰でもブートエントリを選択してシステムをブートできますが、GRUB 2 rootユーザのパスワードは、ブートエントリを変更する場合にのみ必要です。

18.3.3.2 カーネルパラメータタブ



図 18.4: カーネルパラメータ

オプションのカーネルコマンドラインパラメータ

システム機能の有効化/無効化、ドライブの追加などを実行するには、ここでオプションのカーネルパラメータを指定します。

CPU緩和策

SUSEでは、CPUサイドチャネル攻撃を防ぐために導入されているすべてのソフトウェア緩和策に対する1つ以上のカーネルブートコマンドラインパラメータをリリースしました。これらの一部により、性能の低下を招く場合があります。設定に応じて、セキュリティと性能とのバランスをとるために次のオプションのいずれかを選択してください。

自動。 お使いのCPUモデルで必要なすべての緩和策を有効化しますが、CPUスレッドを跨いだ攻撃は保護できません。この設定による性能面への影響は、負荷内容によって異なります。

自動 + SMT無し。 利用可能なセキュリティ面の緩和策をすべて実施することになります。お使いのCPUモデルで必要なすべての緩和策を有効化します。さらに、複数のCPUスレッドを跨いだサイドチャネル攻撃を防ぐため、同時マルチスレッディング(SMT)の機能も無効化します。これにより、負荷内容にもよりますが、[自動] よりも性能面への影響が増すことになります。

オフ。 全ての緩和策を無効化します。CPUのモデルによってさまざまなサイドチャネル攻撃の可能性が高まることになります。この設定により性能面への影響はなくなります。

手動。 緩和レベルを設定しません。カーネルのコマンドラインオプションを使用してCPU緩和策を手動で指定します。

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デフォルトで自動的に設定されますが、コンソールの解像度を介して手動で設定することもできます。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チューザーで指定できます。これを変更するだけで、独自のカスタムメイドのテーマを適用できます。

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](http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal)または<http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal>を参照してください。

18.4 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 18.1: 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。



重要: grub2-mkconfigはUEFIセキュアブートテーブルを修復できません

UEFIセキュアブートを使用していて、システムがGRUB 2に正常にアクセスできなくなった場合、Shimを再インストールして、UEFIブートテーブルを再生成する必要がある場合があります。次のようにします。

```
# shim-install --config-file=/boot/grub2/grub.cfg
```

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 18.2: 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 18.3: GRUB2-SCRIPT-CHECKの使用

```
grub2-script-check /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 18.4: GRUB2-ONCEの使用法

```
grub2-once number_of_the_boot_entry
```



ヒント: **grub2-once help**

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。

18.5 詳細情報

GRUB 2の詳細情報は、<https://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。「にあるTechnical Information Search(技術情報検索)で、キーワード」 GRUB 2<https://www.suse.com/support>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。

19 systemdデーモン

systemdはシステムの初期化を担当し、プロセスID 1を持ちます。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。systemdは、System V initデーモンに代わるものであり、(initスクリプトをサポートすることにより)System V initと完全に互換性があります。

systemdの主な利点は、サービスの開始を並列化することによりブート時間を大幅に短縮できることです。さらに、systemdは、サービスが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は、Kernel Control Groups (cgroups)、スナップショットの作成、システムの状態の復元もサポートしています。詳細については、<http://www.freedesktop.org/wiki/Software/systemd/>を参照してください。

19.1 systemdの概念

次のセクションでは、systemdの背後にある概念について説明します。

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。systemdの主な特徴は次のとおりです。

- 並列化機能
- 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の作成と復元
- 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

19.1.1 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含まれます。

「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ユニットファイルの詳細については、<http://www.freedesktop.org/software/systemd/man/systemd.unit.html>を参照してください。

19.2 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。「や」のように、**git**コマンドの後ろにサブコマンド**zypper**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。

19.2.1 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)
```

System V init

```
rcMY_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。

```
> sudo systemctl start MY_1ST_SERVICE MY_2ND_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。

```
> sudo 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 19.1: サービス管理コマンド

タスク	<u>systemd</u> 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。一部のサービスは再ロードをサポートしていません。	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
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。 <u>systemd</u> 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	status	status

タスク	systemd コマンド	System V init コマンド
でサービスが出力したメッセージ(19.6.9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。		
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

19.2.2 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进行操作できます。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。

19.2.2.1 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起動されず、次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**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 19.2: サービスの有効化/無効化コマンド

タスク	systemd コマンド	System V init コマンド
有効化.	systemctl enable <u>MY_SERVICE(S)</u>	insserv <u>MY_SERVICE(S)</u> 、 chkconfig -a MY_SERVICE(S)

タスク	systemd コマンド	System V initコマンド
無効化.	systemctl disable <u>MY_SERVICE(S).service</u>	insserv -r <u>MY_SERVICE(S)</u> 、 chkconfig -d 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 <u>MY_SERVICE</u>	chkconfig MY_SERVICE
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 <u>MY_SERVICE</u>	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 <u>MY_SERVICE</u>	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 <u>MY_SERVICE</u>	該当なし

19.3 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。この点から見ると、カーネルは、他のプログラムからの要求に従って、他のすべてのプロセスを管理し、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。

19.3.1 ターゲットとランレベルの比較

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(19.4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、カーネルパラメータ `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 19.3: **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	<u>runlevel0.target</u> , <u>halt.target</u> , <u>poweroff.target</u>	システムのシャットダウン
1、S	<u>runlevel1.target</u> , <u>rescue.target</u> ,	シングルユーザモード
2	<u>runlevel2.target</u> , <u>multi-user.target</u> ,	リモートネットワークなしのローカルマルチユーザ
3	<u>runlevel3.target</u> , <u>multi-user.target</u> ,	ネットワークを使用するフルマルチユーザ
4	<u>runlevel4.target</u>	未使用/ユーザ定義
5	<u>runlevel5.target</u> , <u>graphical.target</u> ,	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	<u>runlevel6.target</u> , <u>reboot.target</u> ,	システムの再起動

！ 重要: systemdは/etc/inittabを無視する

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、[19.5.4項「カスタムターゲットの作成」](#)を参照してください。

19.3.1.1 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。

タスク	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 <u>MY_TARGET</u> .ターゲット	telinit <u>X</u>
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 あるいは、 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/<u>MY_TARGET</u>.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: での <u>/etc/inittab</u>
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= <u>MY_TARGET</u> .ターゲット	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。

タスク	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" <u>MY_TARGET</u> .ターゲット systemctl show -p "Wants" <u>MY_TARGET</u>. ターゲット 「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

19.3.2 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(/var/log/を解析することなく)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。

19.3.2.1 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**を使用してください。

例 19.1: アクティブなサービスの一覧表示

```
# systemctl
```

UNIT	LOAD	ACTIVE	SUB	JOB DESCRIPTION
[...]				
iscsi.service	loaded	active	exited	Login and scanning of iSC+
kmod-static-nodes.service	loaded	active	exited	Create list of required s+
libvirtd.service	loaded	active	running	Virtualization daemon
nscd.service	loaded	active	running	Name Service Cache Daemon
chronyd.service	loaded	active	running	NTP Server Daemon
polkit.service	loaded	active	running	Authorization Manager
postfix.service	loaded	active	running	Postfix Mail Transport Ag+
rc-local.service	loaded	active	exited	/etc/init.d/boot.local Co+

```
rsyslog.service          loaded active running   System Logging 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.

161 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

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 19.2: 起動に失敗したサービスの一覧表示

```
# 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

[...]
```

19.3.2.2 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

```
# systemd-analyze
Startup finished in 2666ms (kernel) + 21961ms (userspace) = 24628ms
```

サービスの起動時間の一覧表示

```
# systemd-analyze blame
15.000s backup-rpmdb.service
14.879s mandb.service
7.646s backup-sysconfig.service
4.940s postfix.service
4.921s logrotate.service
4.640s libvirtd.service
4.519s display-manager.service
3.921s btrfsmaintenance-refresh.service
3.466s lvm2-monitor.service
```

```
2.774s plymouth-quit-wait.service
2.591s firewalld.service
2.137s initrd-switch-root.service
1.954s ModemManager.service
1.528s rsyslog.service
1.378s apparmor.service
[...]
```

サービスの起動時間を表す画像

```
# systemd-analyze plot > jupiter.example.com-startup.svg
```



19.3.2.3 起動プロセス全体の確認

先に示したコマンドは、開始されるサービスとその起動時間を一覧表示しています。より詳細な概要については、ブートプロンプトで次のパラメータを指定して、完全な起動手順の詳細なログを作成するように `systemd` に指示します。

```
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
```

19.3.3 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
```

`UPPERCASE LETTERS`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`コマンドで、デーモンを起動します。

19.4 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。

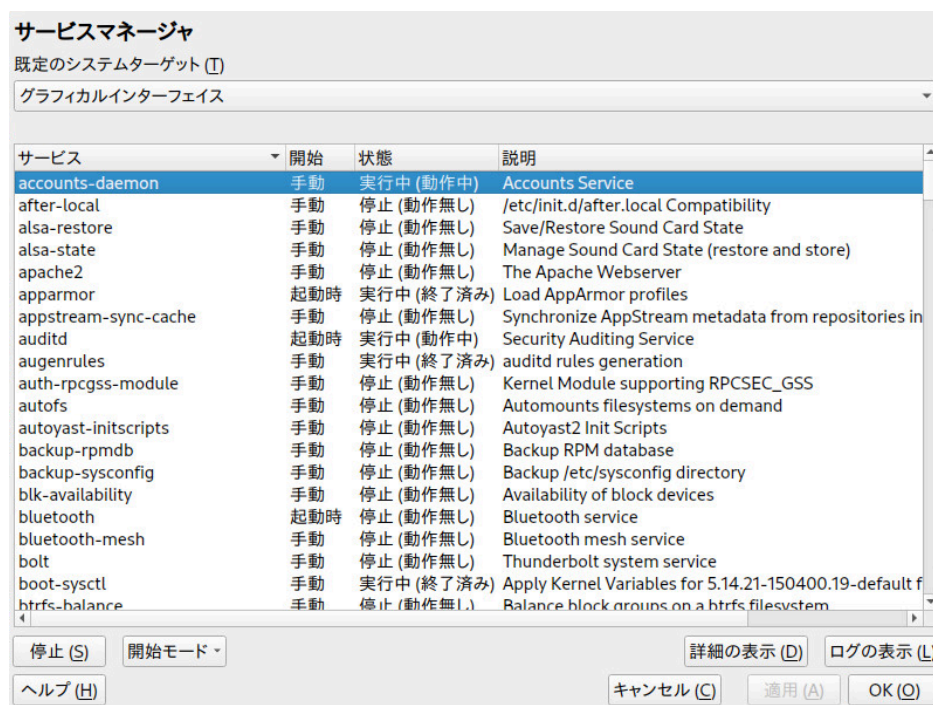


図 19.1: サービスマネージャ

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。状態列は、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、起動または停止を選択します。

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。

サービスの起動動作の定義

サービスは起動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。テーブルからサービスを選択します。起動列には、現在手動で起動されているか、起動時に起動されているかが表示されます。ステータスを切り替えるには、起動モードを選択します。

現在のセッションのサービスステータスを変更するには、先に記載されているように、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。

19.5 カスタマイズsystemd

以降の各項には、`systemd`のカスタマイズ例が示されています。



警告: カスタマイズが上書きされないようにする

`systemd`をカスタマイズする場合は、常にディレクトリ `/etc/systemd/` を使用し、`/usr/lib/systemd/` は使用しないでください。そうしないと、`systemd`の次の更新によって、変更内容が上書きされてしまいます。

19.5.1 ユニットファイルのカスタマイズ

ユニットファイルをカスタマイズするための推奨される方法は、`systemctl edit SERVICE` コマンドを使用する方法です。このコマンドはデフォルトのテキストエディタを起動し、`/etc/systemd/system/NAME.service.d/`にある`override.conf`ファイルを使用してディレクトリを作成します。また、このコマンドは、実行中の`systemd`プロセスに変更について通知するようにします。

または、`systemctl edit --full SERVICE`を実行して、空のファイルの代わりに元のファイルのコピーを開いて編集することもできます。ファイルを編集する際には、既存のセクションを削除しないようにしてください。

演習として、システムがMariaDBの起動を待機する時間を変更します。rootとして`systemctl edit --full mariadb.service`を実行します。開かれたファイルは次のように表示されます。


```
[Unit]
Description=MySQL server
Wants=basic.target
Conflicts=mariadb.target
After=basic.target network.target

[Install]
WantedBy=multi-user.target
Alias=mysql.service

[Service]
Restart=on-abort
Type=notify
ExecStartPre=/usr/lib/mysql/mysql-systemd-helper install
ExecStartPre=/usr/lib/mysql/mysql-systemd-helper upgrade
ExecStart=/usr/lib/mysql/mysql-systemd-helper start

# Configures the time to wait for start-up/stop
TimeoutSec=300

# Prevent writes to /usr, /boot, and /etc
ProtectSystem=full

# Prevent accessing /home, /root and /run/user
ProtectHome=true

UMask=007
```

`TimeoutSec`値を調整して、変更を保存します。変更を有効にするには、`root`として、**`systemctl daemon-reload`**を実行します。

詳細については、**`man 1 systemctl`**コマンドで呼び出すことが可能なマニュアルページを参照してください。

19.5.2 ドロップインファイルの作成

環境設定ファイルのマイナーな変更については、いわゆるドロップイン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/FOOBAR.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、`your_modification.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。



注記: 名前の競合を回避する

ドロップインファイルとSUSEによって提供されるファイルとの名前の競合を回避するには、すべてのドロップインファイル名の前に2桁の数字とダッシュを付けることをお勧めします(たとえば、`80-override.conf`)。

次の範囲が予約されています。

- 0-19は、`systemd`アップストリーム用に予約されています。
- 20-25は、SUSEによって提供される`systemd`用に予約されています。
- 26-29は、(`systemd`以外の)SUSEパッケージ用に予約されています。
- 50は、`systemctl set-property`を使用して作成されたドロップインファイル用に予約されています。

この範囲を超える2桁の数字を使用して、SUSEによって提供されるドロップインファイルが独自のドロップインファイルを上書きしないようにします。

`systemctl cat $UNIT`を使用して、ユニット設定で考慮されるファイルを一覧表示し、確認することができます。

19.5.3 `xinetd`サービスを`systemd`に変換する

SUSE Linux Enterprise Desktop 15のリリース以降、`xinetd`インフラストラクチャは削除されました。このセクションでは、既存のカスタム`xinetd`サービスファイルを、`systemd`ソケットに変換する方法の概要を説明します。

`xinetd`サービスファイルごとに、少なくとも2つの`systemd`ユニットファイル(ソケットファイル(`*.socket`)および関連するサービスファイル(`*.service`))が必要です。ソケットファイルはどのソケットを作成するかを`systemd`に指示し、サービスファイルはどの実行可能ファイルを起動するかを`systemd`に指示します。

次のような`xinetd`サービスファイルの例を考えてみます。

```
# cat /etc/xinetd.d/example
service example
```

```
{
  socket_type = stream
  protocol = tcp
  port = 10085
  wait = no
  user = user
  group = users
  groups = yes
  server = /usr/libexec/example/exampleled
  server_args = -auth=bsdtcp exampledump
  disable = no
}
```

systemdに変換するには、以下の2つの一致するファイルが必要です。

```
# cat /usr/lib/systemd/system/example.socket
[Socket]
ListenStream=0.0.0.0:10085
Accept=false

[Install]
WantedBy=sockets.target
```

```
# cat /usr/lib/systemd/system/example.service
[Unit]
Description=example

[Service]
ExecStart=/usr/libexec/example/exampleled -auth=bsdtcp exampledump
User=user
Group=users
StandardInput=socket
```

systemd 「socket」 および 「service」 ファイルオプションの完全なリストについては、systemd.socketとsystemd.serviceのマニュアルページ([man 5 systemd.socket](#)、[man 5 systemd.service](#))を参照してください。

19.5.4 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。

```
> sudo systemctl daemon-reload
```

19.6 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。

19.6.1 一時ディレクトリの消去

`systemd`によって、定期的に一時ディレクトリを消去できます。前バージョンのシステムの設定は、自動的に移行されアクティブになります。一時ファイルを管理する`tmpfiles.d`は、`/etc/tmpfiles.d/*.conf`、`/run/tmpfiles.d/*.conf`、および`/usr/lib/tmpfiles.d/*.conf`の各ファイルから設定を読み取ります。`/etc/tmpfiles.d/*.conf`にある設定は、他の2つのディレクトリにある関連設定より優先します(`/usr/lib/tmpfiles.d/*.conf`には、パッケージの設定ファイルが保存されています)。

設定のフォーマットは、パスごとに1行で、アクション、パス、およびオプションでモード、所有権、経過時間、引数のフィールドが含まれています(アクションによって変わります)。次の例は、X11ロックファイルのリンクを解除します。

Type	Path	Mode	UID	GID	Age	Argument
r	/tmp/.X[0-9]*-lock					

`tmpfile timer`のステータスを取得するには、以下のようにします。

```
> sudo systemctl status systemd-tmpfiles-clean.timer
systemd-tmpfiles-clean.timer - Daily Cleanup of Temporary Directories
Loaded: loaded (/usr/lib/systemd/system/systemd-tmpfiles-clean.timer; static)
Active: active (waiting) since Tue 2018-04-09 15:30:36 CEST; 1 weeks 6 days ago
```

```
Docs: man:tmpfiles.d(5)
      man:systemd-tmpfiles(8)
```

```
Apr 09 15:30:36 jupiter systemd[1]: Starting Daily Cleanup of Temporary Directories.
Apr 09 15:30:36 jupiter systemd[1]: Started Daily Cleanup of Temporary Directories.
```

一時ファイルの処理について詳しくは、[man 5 tmpfiles.d](#)を参照してください。

19.6.2 システムログ

19.6.9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、すべてのログメッセージを表示するには、`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 journalctl](#)を参照してください。

19.6.3 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

```
> sudo systemctl snapshot MY_SNAPSHOT.snapshot
```

スナップショットの削除

```
> sudo systemctl delete MY_SNAPSHOT.snapshot
```

スナップショットの表示

```
> sudo systemctl show MY_SNAPSHOT.snapshot
```

スナップショットの有効化

```
> sudo systemctl isolate MY_SNAPSHOT.snapshot
```

19.6.4 カーネルモジュールのロード

systemdにより、/etc/modules-load.d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。このファイルはMODULE.confという名前で、次のような内容です。

```
# load module MODULE at boot time
MODULE
```

カーネルモジュールをロードするための設定ファイルがパッケージによってインストールされる場合、そのファイルは/usr/lib/modules-load.dにインストールされます。同じ名前の環境設定ファイルが2つ存在する場合、/etc/modules-load.dにあるファイルが優先されます。

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。

19.6.5 サービスのロード前にアクションを実行

System Vでは、サービスをロードする前に実行する必要があるinitアクションは、/etc/init.d/before.local に指定する必要がありました。この手順は、systemdではサポートされません。サービスの起動前にアクションを実行する必要がある場合、以下のようにしてください。

カーネルモジュールのロード

ドロップインファイルを/etc/modules-load.dディレクトリに作成します(構文は、man modules-load.dを参照)。

ファイルまたはディレクトリの作成、ディレクトリの消去、所有権の変更

ドロップインファイルを/etc/tmpfiles.dに作成します(構文は、man tmpfiles.dを参照)。

その他のタスク

システムサービスファイル(/etc/systemd/system/before.serviceなど)を、次のテンプレートから作成します。

```
[Unit]
Before=NAME OF THE SERVICE YOU WANT THIS SERVICE TO BE STARTED BEFORE
[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=YOUR_COMMAND
# beware, executable is run directly, not through a shell, check the man pages
# systemd.service and systemd.unit for full syntax
[Install]
# target in which to start the service
```

```
WantedBy=multi-user.target
#WantedBy=graphical.target
```

サービスファイルを作成したら、次のコマンドを実行する必要があります(rootユーザとして実行)。

```
> sudo systemctl daemon-reload
> sudo systemctl enable before
```

サービスファイルを変更するたびに、以下を実行する必要があります。

```
> sudo systemctl daemon-reload
```

19.6.6 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。特定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 19.3: サービスに属するすべてのプロセスの表示

```
# 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
└─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、『System Analysis and Tuning Guide』、第10章「Kernel control groups」を参照してください。

19.6.7 サービスの終了(シグナルの送信)

19.6.6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停止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。

```
> sudo systemctl kill MY_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。

```
> sudo systemctl kill -s SIGNAL MY_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。


```
> sudo systemctl kill -s SIGHUP --kill-who=main MY_SERVICE
```

19.6.8 D-Busサービスに関する重要な注意事項

D-Busサービスは、`systemd`クライアントと、pid 1として実行される`systemd`マネージャ間の通信用のメッセージバスです。`dbus`はスタンドアロンのデーモンですが、初期化インフラストラクチャの不可欠な要素です。

動作中のシステムで`dbus`を終了または再起動することは、pid 1の終了または再起動と同様の結果をもたらします。`systemd`のクライアント/サーバ通信が切断され、`systemd`のほとんどの機能が使用できなくなります。

したがって、`dbus`の終了または再起動は推奨されず、サポートもされません。

`dbus`または`dbus`に関連するパッケージを更新するには、再起動する必要があります。再起動が必要かどうか疑問に思う場合は、`sudo zypper ps -s`コマンドを実行します。`dbus`が一覧表示されているサービスに表示される場合は、システムを再起動する必要があります。

自動更新が再起動が必要なパッケージをスキップするように設定されている場合でも、`dbus`は更新されることに留意してください。

19.6.9 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`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。

```
# systemctl start apache2
Job failed. See system journal and 'systemctl status' for details.
# systemctl status apache2
    Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
    Active: failed (Result: exit-code) since Mon, 04 Apr 2018 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
```

```
Apr 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の表示

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。

```
> sudo systemctl status chronyd  
> sudo systemctl --lines=20 status chronyd
```

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」を表示するには、`--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。

```
> sudo systemctl --follow status chronyd
```

メッセージの出力形式

`--output=MODE`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。

19.7 systemdタイマユニット

cronと同様、systemdタイマユニットは、Linuxでジョブをスケジュールするためのメカニズムを提供します。systemdタイマユニットは、cronと同じ目的を果たしますが、いくつかの利点があります。

- タイマユニットを使用してスケジュールされたジョブは、他のsystemdサービスに依存する可能性があります。
- タイマユニットは通常のsystemdサービスとして扱われるため、systemctlで管理することができます。
- タイマはリアルタイムおよび単調にすることができます。
- タイマユニットはsystemdジャーナルに記録されるため、監視とトラブルシューティングが容易になります。

systemdタイマユニットは、.timerファイル名拡張子で識別されます。

19.7.1 systemdタイマタイプ

タイマユニットは単調タイマとリアルタイムタイマを使用できます。

- cronjobsと同様に、リアルタイムタイマはカレンダーのイベントにトリガされます。リアルタイムタイマはオプションOnCalendarを使用して定義されます。
- 単調タイマは特定の開始時点から指定の時間が経過するとトリガされます。後者はシステム起動イベントまたはまたはシステムユニットアクティベーションイベントのいずれかです。OnBootSec、OnUnitActiveSec、OnTypeSecなどの単調タイマを定義するためのいくつかのオプションがあります。単調タイマは永続的ではなく、各再起動後にリセットされます。

19.7.2 systemdタイマとサービスユニット

すべてのタイマユニットには、それが制御する対応するsystemdユニットファイルが必要です。言い換えると、.timerファイルは対応する.serviceファイルを有効にし、管理します。タイマとともに使用する場合、.serviceファイルには[Install]セクションは必要ありません。サービスがタイマによって管理されるためです。

19.7.3 実例

systemdタイマユニットの基本を理解するために、foo.shシェルスクリプトをトリガするタイマを設定します。

最初のステップは、シェルスクリプトを制御するsystemdサービスユニットを作成することです。これを行うには、編集用に新しいテキストファイルを開いて、次のサービスユニット定義を追加します。

```
[Unit]
Description="Foo shell script"

[Service]
ExecStart=/usr/local/bin/foo.sh
```

ファイルをfoo.serviceという名前で/etc/systemd/system/ディレクトリに保存します。

次に、編集用に新しいテキストファイルを開いて、次のタイマ定義を追加します。

```
[Unit]
Description="Run foo shell script"

[Timer]
OnBootSec=5min
OnUnitActiveSec=24h
Unit=foo.service

[Install]
WantedBy=multi-user.target
```

先の例で[Timer]セクションはトリガするサービス(foo.service)とトリガするタイミングを指定します。この場合、オプションOnBootSecはシステム起動の5分後にサービスをトリガする単調タイマを指定し、オプションOnUnitActiveSecはサービスが有効になってから24時間後にサービスをトリガします(つまり、タイマは1日に1回サービスをトリガします)。最後に、オプションWantedByはシステムがマルチユーザターゲットに到達したときに、タイマが開始されるように指定します。

単調タイマの代わりに、オプションOnCalendarを使用してリアルタイムタイマを指定できます。次のリアルタイムタイマ定義は週に1回、月曜日の12時に関連するサービスをトリガします。

```
[Timer]
OnCalendar=weekly
Persistent=true
```

オプションPersistent=trueは、タイマが最後の開始時刻を逃した場合(たとえば、システムの電源がオフになっているため)、タイマが有効化された直後にサービスがトリガされることを示します。

オプションOnCalendarを使用して、次の形式で、サービスをトリガするための特定の日時を定義することもできます。DayOfWeek Year-Month-Day Hour:Minute:Second。次の例は、毎日午前5時にサービスをトリガします。

```
OnCalendar=*-*-* 5:00:00
```

アスタリスクを使用して任意の値を指定し、コンマを使用して可能な値をリストできます。..によって区切られた2つの値を使用して、連続した範囲を示します。次の例は毎月金曜日の午後6時にサービスをトリガします。

```
OnCalendar=Fri *-*-1..7 18:00:00
```

異なる時間にサービスをトリガするために、複数の`OnCalendar`エントリを指定できます。

```
OnCalendar=Mon..Fri 10:00  
OnCalendar=Sat,Sun 22:00
```

次の例では、平日の午前10時および週末の午後10時にサービスがトリガされます。

タイマユニットファイルの編集が終了したら、`foo.timer`という名前で`/etc/systemd/system/`ディレクトリに保存します。作成したユニットファイルが正しいかどうかを確認するには、次のコマンドを実行します。

```
> sudo systemd-analyze verify /etc/systemd/system/foo.*
```

コマンドが出力を返さない場合、ファイルは検証に成功しています。

タイマを開始するには、コマンド**`sudo systemctl start foo.timer`**を使用します。起動時にタイマを有効にするには、コマンド**`sudo systemctl enable foo.timer`**を使用します。

19.7.4 systemdタイマの管理

タイマは通常のsystemdユニットとして扱われるため、**`systemctl`**を使用して管理できます。たとえば、**`systemctl start`**でタイマを開始し、**`systemctl enable`**でタイマを有効にすることができます。また、コマンド**`systemctl list-timers`**を使用して、すべてのアクティブタイマを一覧表示できます。非アクティブなタイマを含むすべてのタイマを一覧表示するには、コマンド**`systemctl list-timers --all`**を実行します。

19.8 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> 

管理者向け systemd

systemdの著者のうちの1人、Lennart Pöttering氏によるブログに、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> 

III システム

- 20 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 291
- 21 **journalctl**: systemdジャーナルのクエリ 293
- 22 **update-alternatives**: 複数のバージョンのコマンドとファイルの管理 301
- 23 ネットワークの基礎 309
- 24 プリンタの運用 375
- 25 グラフィカルユーザインタフェース 390
- 26 FUSEによるファイルシステムへのアクセス 409
- 27 複数バージョンのカーネルのインストール 411
- 28 カーネルモジュールの管理 418
- 29 udevによる動的カーネルデバイス管理 422
- 30 特別なシステム機能 435
- 31 NetworkManagerの使用 447

20 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 Desktopは64ビットプラットフォームで利用できます。ただし、開発者はすべての32ビットアプリケーションを64ビットシステムに移植しているわけではありません。この章では、32ビットサポートを64ビットのSUSE Linux Enterprise Desktopプラットフォームで実装する方法について簡潔に説明します。

64ビットプラットフォームのAMD64およびIntel 64に対応したSUSE Linux Enterprise Desktopは、既存の32ビットアプリケーションが64ビット環境で「すぐに」実行できるように設計されています。このサポートにより、対応する64ビット移植版が使用可能になるのを待たなくても、使用したい32ビットアプリケーションを引き続き使用できます。



注記: 32ビットアプリケーションを構築するためのサポートなし

SUSE Linux Enterprise Desktopでは32ビットアプリケーションのコンパイルをサポートしていません。32ビットバイナリのランタイムサポートのみ提供します。

20.1 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利用可能な場合は、両方のバージョンをインストール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、ランタイムエラーになるのを回避するために、インストールする一方のバージョンを決めてください。

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 Desktopは、ユーザとアプリケーションを仲介するレイヤとしての認証プロセスでPAMを使用します。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、常に両方のPAMバージョンをインストールしてください。

正しく実行するには、すべてのアプリケーションに一連の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これら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保持するため、64ビットライブラリと32ビットライブラリは同じ場所に保存されます。`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。

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ビットライブラリで使用可能な領域は`/lib`および`/usr/lib`の下になっています。

`/lib`の下にある32ビットサブディレクトリのデータコンテンツがワードサイズに依存しない場合、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。

20.2 カーネル仕様

AMD 64またはIntel 64向けの64ビットカーネルには、64ビットと32ビットのカーネルABI (アプリケーションバイナリインタフェース)が用意されています。32ビットのカーネルABIは、該当する32ビットカーネルのABIと同じものです。つまり、32ビットアプリケーションと64ビットアプリケーションが64ビットカーネルで通信できることを意味します。

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はシステムプログラムによって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`などのいくつかのアプリケーションは、コンパイルする必要があります。

64ビットカーネルは64ビットカーネルモジュールのみロードすることができます。そのため、64ビットカーネル用に特別に64ビットモジュールをコンパイルする必要があります。64ビットカーネルでは、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。32ビットアプリケーションを64ビットシステム環境で使いたい場合は、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ください。」カーネルロード可能モジュールの64ビットバージョンとカーネルAPIの32ビットコンパイルバージョンがこのモジュール用に入手可能であることを確認してください。

21 journalctl: systemdジャーナルのクエリ

systemdは、「ジャーナル」と呼ばれる独自のロギングシステムを備えています。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、標準入力、およびシステムサービス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サービスはデフォルトでオンになっています。

```
> sudo 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
[...]
```

21.1 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。ログデータを永続的にするには、/var/log/journal/ディレクトリを作成し、これが正しいアクセスモードであり、所有権も正しいことを確認し、systemd-journaldサービスがそのデータを保存できるようにします。永続的なログ記録に切り替えるには、次のコマンドを実行します。

```
> sudo mkdir /var/log/journal
> sudo systemd-tmpfiles --create --prefix=/var/log/journal
> sudo journalctl --flush
```

/run/log/journal/に保存されるログデータは/var/log/journal/にフラッシュされます。

21.2 journalctl: 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メッセージ

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。

```
> sudo 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。

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(21.3.3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(21.3.3項「フィールドに基づくフィルタ」を参照)。

```
> sudo 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.
```

21.3 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。

21.3.1 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。

```
> sudo 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、その1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。

```
> sudo 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。

```
> sudo 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。

```
> sudo journalctl _BOOT_ID=156019a44a774a0bb0148a92df4af81b
```

21.3.2 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。数値式ではなく、キーワード「yesterday」、

「today」、または「tomorrow」を指定することができます。これらは、当日の前日の夜中の12:00、当日の夜中の12:00、または当日の翌日の夜中の12:00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。

```
> sudo 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。

```
> sudo journalctl --since "today" --until "3:20"
```

21.3.3 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。

```
> sudo 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)。

```
> sudo 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。

```
> sudo 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。

```
> sudo 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。

```
> sudo 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。

```
> sudo 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。

```
> sudo journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1480 +  
_SYSTEMD_UNIT=dbus.service
```

21.4 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  
Job for apache2.service failed. See 'systemctl status apache2' 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。

```
> sudo systemctl status apache2  
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。

```
> sudo 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。

```
> sudo 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 [...]
```

21.5 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ります。

```
> sudo systemctl restart systemd-journald
```

21.5.1 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(21.1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
```

21.5.2 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
```

21.5.3 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。

```
> sudo rpm -q rsyslog
rsyslog-7.4.8-2.16.x86_64
```

- rsyslogサービスが有効である。

```
> sudo systemctl is-enabled rsyslog
enabled
```

- /etc/systemd/journald.confでsyslogへの転送が有効になっている。

```
ForwardToSyslog=yes
```

21.6 YaSTを使用したsystemdジャーナルのフィルタ

(journalctl構文を処理することなく)systemdジャーナルを簡単にフィルタするには、YaSTのジャーナルモジュールを使用します。**sudo zypper in yast2-journal**を使用してモジュールをインストールした後、YaSTでSystem (システム) > Systemd Journal (systemdジャーナル)の順に選択して起動します。または、コマンドラインで「**sudo yast2 journal**」と入力して起動します。

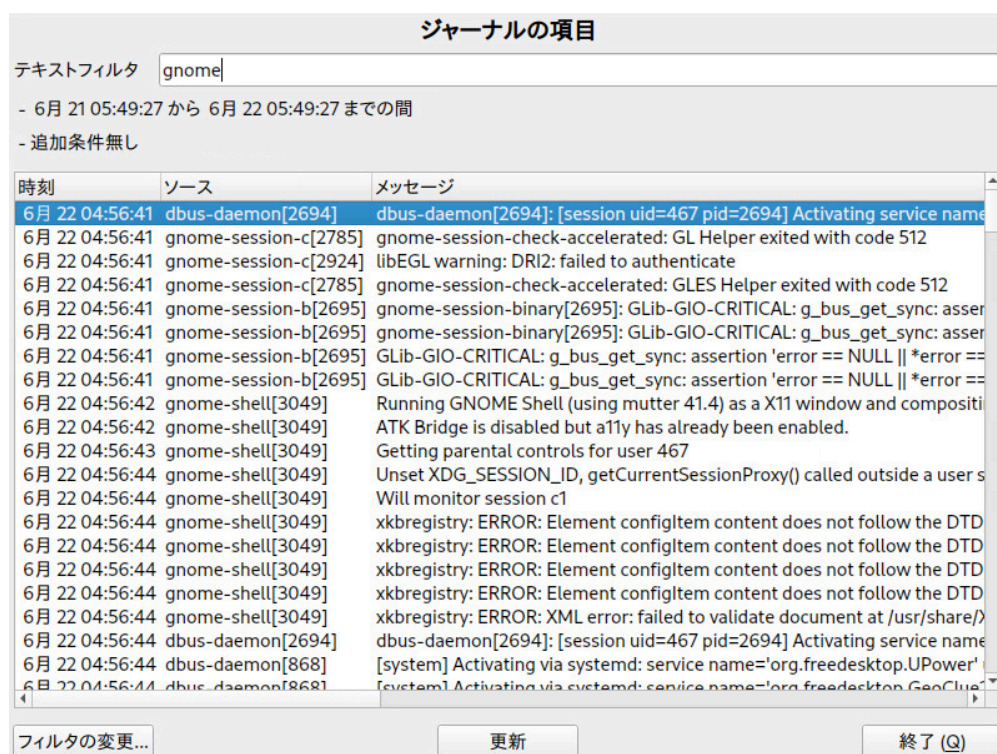


図 21.1: YAST SYSTEMDジャーナル

このモジュールでは、ログエントリが表に表示されます。上部にある検索ボックスを使用すると、**grep**を使用する場合と同様に、特定の文字を含むエントリを検索することができます。日時、ユニット、ファイル、または優先度でエントリをフィルタするには、Change filters (フィルタの変更)をクリックし、個々のオプションを設定します。

21.7 GNOMEでのログの表示

「GNOMEログ」でジャーナルを表示することができます。アプリケーションメニューから開始します。システムログメッセージを表示するには、rootとして実行する必要があります(たとえば、**xdg-su gnome-logs**を使用)。このコマンドは、**Alt + F2**を押すと実行できます。

22 **update-alternatives**: 複数のバージョンのコマンドとファイルの管理

システムに同じツールの複数のバージョンがインストールされていることがよくあります。管理者に選択肢を提供し、異なる複数のバージョンを並行してインストールして使用できるようにするために、alternativesシステムでは、このような複数のバージョンを一貫性を持って管理することができます。

22.1 概要

SUSE Linux Enterprise Desktopでは、いくつかのプログラムが同じまたは類似のタスクを実行します。たとえば、Java 1.7とJava 1.8が両方ともシステムにインストールされている場合、alternativesシステムスクリプト(**update-alternatives**)がRPMパッケージ内から呼び出されます。デフォルトでは、alternativesシステムはバージョン1.8を指し示します。上位バージョンの優先度が高くなります。ただし、管理者はデフォルトを変更することができ、汎用名としてバージョン1.7を指すことができます。

この章では、以下の用語を使用します。

用語集

管理ディレクトリ

デフォルトの/var/lib/rpm/alternativesディレクトリには、alternativesの現在の状態に関する情報が含まれています。

代わりとなる製品

ファイルシステム内の特定のファイルの名前。alternativesシステムを使用して汎用名でアクセス可能にすることができます。

alternativesディレクトリ

シンボリックリンクを含むデフォルトの/etc/alternativesディレクトリ。

汎用名

alternativesシステムを使用して使用可能な複数のファイルのうちの1つを指し示す名前 (たとえば、/usr/bin/edit)。

リンクグループ

グループとして更新できる関連するシンボリックリンクのセット。

マスタリンク

グループ内の他のリンクの設定方法を決定するリンクグループ内のリンク。

スレーブリンク

マスタリンクによって制御されるリンクグループ内のリンク。

シンボリックリンク(Symlink)

同じファイルシステム内の別のファイルへの参照となるファイル。alternativesシステムは、alternativesディレクトリ内のシンボリックリンクを使用して、ファイルのバージョンを切り替えます。

alternativesディレクトリ内のシンボリックリンクは、update-alternatives コマンドによって管理者が変更できます。

alternativesシステムは、シンボリックリンクに関する情報を作成、削除、維持、および表示するための update-alternatives コマンドを提供します。これらのシンボリックリンクは、通常、コマンドを指しますが、JARアーカイブ、マニュアルページ、およびその他のファイルを指すこともできます。この章の例では、コマンドとマニュアルページを使用しますが、他のファイルタイプにも適用できます。

alternativesシステムはalternativesディレクトリを使用して、使用可能なalternativesへのリンクを収集します。alternativeを含む新しいパッケージがインストールされると、新しいalternativeがシステムに追加されます。新しいパッケージのalternativeがデフォルトとして選択されるかどうかは、その優先順位と設定されたモードに依存します。上位バージョンのパッケージの優先度が高くなります。alternativesシステムは、次の2つのモードで動作することができます。

- **自動モード**。このモードでは、alternativesシステムによって、グループ内のリンクが、グループに適した最優先のalternativesを確実に指し示すようになります。
- **手動モード**。このモードでは、alternativesシステムによって、システム管理者の設定は変更されません。

たとえば、java コマンドのalternativesシステムでのリンク階層は次のようになっています。

例 22.1: javaコマンドのALTERNATIVESシステム

```
/usr/bin/java ❶  
-> /etc/alternatives/java ❷  
-> /usr/lib64/jvm/jre-10-openjdk/bin/java ❸
```

- ❶ 汎用名。
- ❷ alternativesディレクトリ内のシンボリックリンク。
- ❸ alternativesの1つ。

22.2 使用例

デフォルトでは、**update-alternatives** スクリプトはRPMパッケージ内から呼び出されます。あるパッケージがインストールまたは削除されると、このスクリプトはそのすべてのシンボリックリンクに対処します。ただし、以下の操作についてはコマンドラインから手動で実行することができます。

- 汎用名に対する現在のalternativesを表示する。
- alternativeのデフォルトを変更する。
- alternativeの関連ファイルのセットを作成する。

22.3 alternativesの概要の取得

すべての設定済みのalternativesの名前を取得するには、次を使用します。

```
> ls /var/lib/alternatives
```

すべての設定済みのalternativesの概要およびその値を取得するには、次を使用します

```
> sudo update-alternatives --get-selections
asadmin                auto      /usr/bin/asadmin-2.7
awk                    auto      /usr/bin/gawk
chardetect              auto      /usr/bin/chardetect-3.6
dbus-launch             auto      /usr/bin/dbus-launch.x11
default-displaymanager auto      /usr/lib/X11/displaymanagers/gdm
[...]
```

22.4 特定のalternativesに関する詳細の表示

alternativesを確認する最も簡単な方法は、コマンドのシンボリックリンクをたどることです。たとえば、**java** コマンドが参照しているリンクを見つけるには、次のコマンドを使用します。

```
> readlink --canonicalize /usr/bin/java
/usr/lib64/jvm/jre-10-openjdk/bin/java
```

同じパスが表示されている場合(この例では、/usr/bin/java)、このコマンドに対して使用可能なalternativesはありません。

すべてのalternatives(スレーブを含む)を表示するには、**--display** オプションを使用します。

```
> sudo update-alternatives --display java
```

```

java - auto mode
link best version is /usr/lib64/jvm/jre-1.8.0-openjdk/bin/java
link currently points to /usr/lib64/jvm/jre-1.8.0-openjdk/bin/java
link java is /usr/bin/java
slave java.1.gz is /usr/share/man/man1/java.1.gz
slave jre is /usr/lib64/jvm/jre
slave jre_exports is /usr/lib64/jvm-exports/jre
slave keytool is /usr/bin/keytool
slave keytool.1.gz is /usr/share/man/man1/keytool.1.gz
slave orbd is /usr/bin/orbd
slave orbd.1.gz is /usr/share/man/man1/orbd.1.gz
[...]

```

22.5 alternativesのデフォルトバージョンの設定

デフォルトでは、/usr/binのコマンドは、優先順位が最も高いalternativesディレクトリを参照します。たとえば、デフォルトでは、コマンド java に対して次のバージョン番号が表示されます。

```

> java -version
openjdk version "10.0.1" 2018-04-17
OpenJDK Runtime Environment (build 10.0.1+10-suse-lp150.1.11-x8664)
OpenJDK 64-Bit Server VM (build 10.0.1+10-suse-lp150.1.11-x8664, mixed mode)

```

デフォルトの java コマンドを変更して前のバージョンを参照するには、次のコマンドを実行します。

```

> sudo update-alternatives --config java
root's password:
There are 2 choices for the alternative java (providing /usr/bin/java).

```

Selection	Path	Priority	Status
* 0	/usr/lib64/jvm/jre-10-openjdk/bin/java	2005	auto mode
1	/usr/lib64/jvm/jre-1.8.0-openjdk/bin/java	1805	manual mode
2	/usr/lib64/jvm/jre-10-openjdk/bin/java	2005	manual mode
3	/usr/lib64/jvm/jre-11-openjdk/bin/java	0	manual mode

Press <enter> to keep the current choice[*], or type selection number:

システムおよびインストールされているバージョンに応じて、正確なJavaバージョン番号は変わります。1を選択すると、java に対して次のバージョン番号が表示されます。

```

> java -version
java version "1.8.0_171"
OpenJDK Runtime Environment (IcedTea 3.8.0) (build 1.8.0_171-b11 suse-lp150.2.3.1-x86_64)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)

```

また、次の点に注意してください。

- 手動モードを使用して、別のJavaバージョンをインストールする場合、alternativesシステムはリンクに触れず、汎用名も変更しません。
- 自動モードを使用して、別のJavaバージョンをインストールする場合、alternativesシステムはJavaマスタリンクとすべてのスレーブリンクを変更します(22.4項「特定のalternativesに関する詳細の表示」を参照)。マスタ-スレーブの関係を確認するには、次のコマンドを使用します。

```
> sudo update-alternatives --display java
```

22.6 カスタムalternativesのインストール

このセクションでは、システムでカスタムalternativesを設定する方法について説明します。この例には、以下の前提があります。

- 同様の機能を備えた`foo-2`と`foo-3`という2つのスクリプトがあります。
- これらのスクリプトは、`/usr/bin`内のシステムツールとの競合を避けるために、`/usr/local/bin`ディレクトリに保存されています。
- `foo-2`または`foo-3`のいずれかを指し示すマスタリンク`foo`があります。

使用しているシステムにalternativesを用意するには、次の手順を実行します。

1. スクリプトを`/usr/local/bin`ディレクトリにコピーします。
2. スクリプトを実行可能にします。

```
> sudo chmod +x /usr/local/bin/foo-{2,3}
```

3. 両方のスクリプトに対して`update-alternatives`を実行します。

```
> sudo update-alternatives --install \  
  /usr/local/bin/foo ① \  
  foo ② \  
  /usr/local/bin/foo-2 ③ \  
  200 ④  
> sudo update-alternatives --install \  
  /usr/local/bin/foo ① \  
  foo ② \  
  /usr/local/bin/foo-3 ③ \  
  300 ④
```

`--install`の後のオプションには、次の意味があります。

- ① 汎用名。混乱を避けるため、これは通常、バージョン番号のないスクリプト名です。
- ② マスタリンクの名前。同じである必要があります。
- ③ `/usr/local/bin`にある元のスクリプトへのパス。
- ④ 優先度。foo-2に、foo-3よりも低い優先度を与えます。意味のある数値の増加を使用して、優先度を分けることをお勧めします。たとえば、foo-2の優先度を200、foo-3の優先度を300にします。

4. マスタリンクを確認します。

```
> sudo update-alternatives --display foo
foo - auto mode
  link best version is /usr/local/bin/foo-3
  link currently points to /usr/local/bin/foo-3
  link foo is /usr/local/bin/foo
/usr/local/bin/foo-2 - priority 200
/usr/local/bin/foo-3 - priority 300
```

上記の手順を完了したら、マスタリンク`/usr/local/bin/foo`を使用できます。

必要に応じて、追加のalternativesをインストールすることもできます。alternativeを削除するには、次のコマンドを使用します。

```
> sudo update-alternatives --remove foo /usr/local/bin/foo-2
```

このスクリプトが削除されると、fooグループのalternativesシステムは次のようになります。

```
> sudo update-alternatives --display foo
foo - auto mode
  link best version is /usr/local/bin/foo-3
  link currently points to /usr/local/bin/foo-3
  link foo is /usr/local/bin/foo
/usr/local/bin/foo-3 - priority 300
```

22.7 依存するalternativesの定義

alternativesがあっても、スクリプト自体は十分なものではありません。ほとんどのコマンドは、スタンドアロンではありません。これらのコマンドには、拡張機能、設定、マニュアルページなどの追加ファイルが付属しています。マスタリンクに依存するalternativesを作成するには、「スレーブalternatives」を使用します。

22.6項「カスタムalternativesのインストール」の例を拡張し、次のマニュアルページと環境設定ファイルを用意すると仮定します。

- `/usr/local/man/man1`ディレクトリに保存されている2つのマニュアルページ(`foo-2.1.gz`および`foo-3.1.gz`)。
- `/etc`に格納された2つの環境設定ファイル(`foo-2.conf`および`foo-3.conf`)。

以下の手順に従って、alternativesに追加ファイルを追加します。

1. 環境設定ファイルを`/etc`にコピーします。

```
> sudo cp foo-{2,3}.conf /etc
```

2. マニュアルページを`/usr/local/man/man1`ディレクトリにコピーします。

```
> sudo cp foo-{2,3}.1.gz /usr/local/man/man1/
```

3. `--slave`オプションを使用して、メインスクリプトにスレーブリンクを追加します。

```
> sudo update-alternatives --install \  
  /usr/local/bin/foo foo /usr/local/bin/foo-2 200 \  
  --slave /usr/local/man/man1/foo.1.gz \  
  foo.1.gz \  
  /usr/local/man/man1/foo-2.1.gz \  
  --slave /etc/foo.conf \  
  foo.conf \  
  /etc/foo-2.conf  
> sudo update-alternatives --install \  
  /usr/local/bin/foo foo /usr/local/bin/foo-3 300 \  
  --slave /usr/local/man/man1/foo.1.gz \  
  foo.1.gz \  
  /usr/local/man/man1/foo-3.1.gz \  
  --slave /etc/foo.conf \  
  foo.conf \  
  /etc/foo-3.conf
```

4. マスタリンクを確認します。

```
foo - auto mode  
link best version is /usr/local/bin/foo-3  
link currently points to /usr/local/bin/foo-3  
link foo is /usr/local/bin/foo  
slave foo.1.gz is /usr/local/man/man1/foo.1.gz  
slave foo.conf is /etc/foo.conf  
/usr/local/bin/foo-2 - priority 200  
slave foo.1.gz: /usr/local/man/man1/foo-2.1.gz  
slave foo.conf: /etc/foo-2.conf
```

```
/usr/local/bin/foo-3 - priority 300
  slave foo.1.gz: /usr/local/man/man1/foo-3.1.gz
  slave foo.conf: /etc/foo-3.conf
```

`update-alternatives --config foo`を使用してリンクをfoo-2に変更すると、スレーブリンクもすべて変更されます。

23 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、複数の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台のマシン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によって結び付けられた複数のネットワークから成る世界規模のネットワークは、「インターネット」とも呼ばれます。

RFCは、「Request for Comments」の略です。RFCは、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、<https://datatracker.ietf.org/>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): これ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。

図23.1「TCP/IPの簡易階層モデル」に示したように、データのやり取りはさまざまなレイヤで実行されます。実際の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。

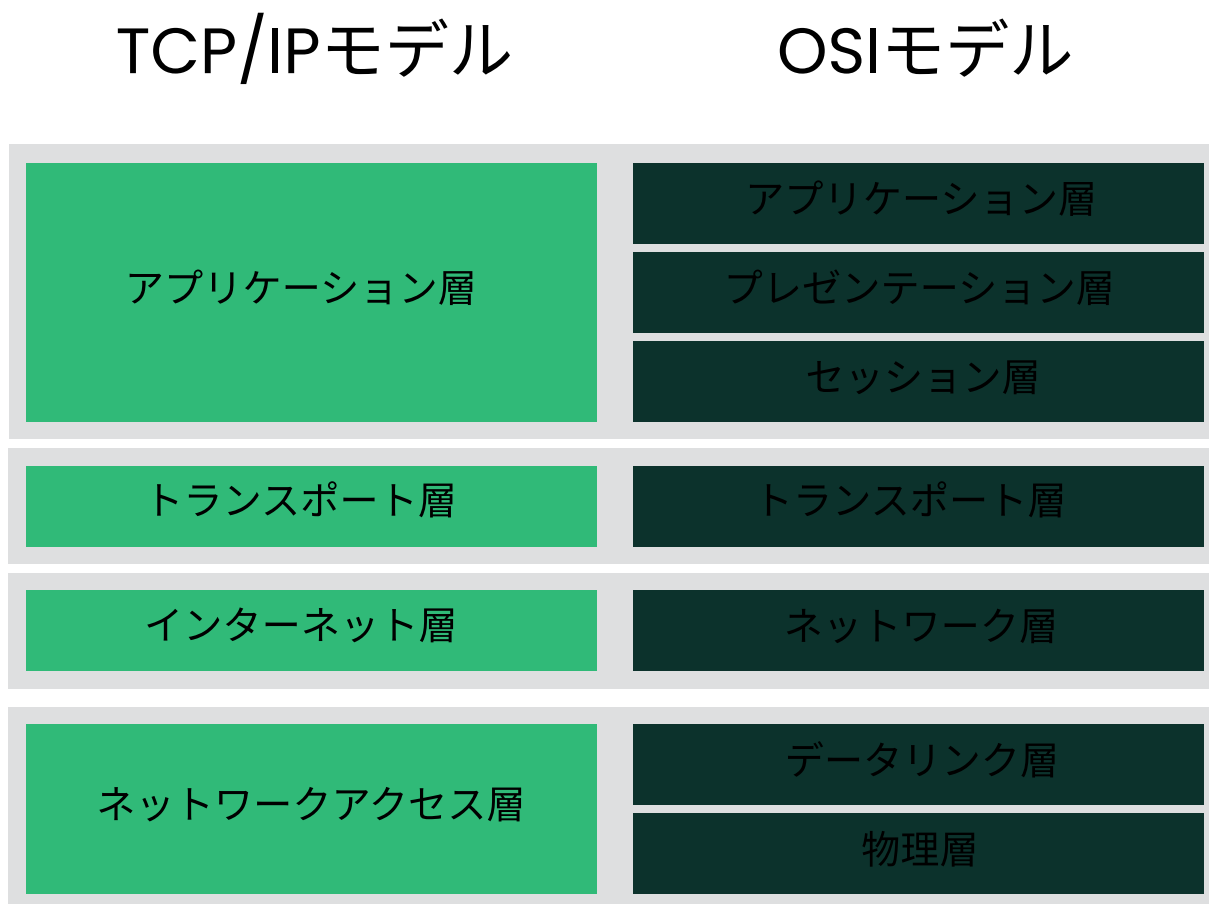


図 23.1: TCP/IPの簡易階層モデル

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、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23.2「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 23.2: TCP/IPイーサネットパケット

アプリケーションがデータをネットワーク経由で送信すると、データは各レイヤを通過します。これらのレイヤは、物理レイヤを除き、すべてLinuxカーネルに実装されています。各レイヤは、隣接する下位レイヤに渡せるようにデータを処理します。最下位レイヤは、最終的にデータを送信する責任を負います。データを受信したときには、この手順全体が逆の順序で実行されます。重なり合ったたまねぎの皮のように、各レイヤで伝送データからプロトコルヘッダが除去されていきます。最後に、トランスポートレイヤが、着信側のアプリケーションがデータを利用できるように処理します。この方法では、1つのレイヤが直接やり取りを行うのは隣接する上下のレイヤのみです。アプリケーションの場合、データが無線接続と有線接続のどちらで送信されるかは関係ありません。同様に、物理ネットワークは、パケットの形式が正しければよく、伝送されるデータの種類を意識することはありません。

23.1 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、23.2項「IPv6 - 次世代インターネット」を参照してください。

23.1.1 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23.1「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 23.1: 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ー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。

23.1.2 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ー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23.2「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 23.2「IPアドレスとネットマスクの論理積(AND)」の、ビットからなる第1のネットワークは、192.168.0.0/24/24と書くこともできます。

例 23.2: 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23.2「IPアドレスとネットマスクの論理積(AND)」のResult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てはできません。

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23.1「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 23.1: 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	Domain
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

23.2 IPv6 - 次世代インターネット

ワールドワイドウェブ(WWW)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、いくつかの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。IPv4ネットワークでホストを設定するには、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、複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。

23.2.1 長所

IPv6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 3.4×10^{38} 個のIPアドレスが得られます。

しかしながら、IPv6アドレスがその先行プロトコルと異なるのはアドレス長だけではありません。IPv6アドレスは内部構造も異なり、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、[23.2.2項「アドレスのタイプと構造」](#)を参照してください。

次に、IPv6プロトコルの他の利点を紹介します。

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、新しくシステムを設定すると、手動で環境設定しなくても、(ローカル)ネットワークに統合されます。新しいホストは自動環境設定メカニズムを使用して、「ネイバーディスカバリ」(ND)と呼ばれるプロトコルにより、近隣のルータから得られる情報を元に自身のアドレスを生成します。この方法は、管理者の介入が不要だけでなく、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を送信します。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ーフェースに同時に割り当てることができます。これにより、ユーザは複数のネットワークに簡単にアクセスできます。これは携帯電話会社が提供する国際ローミングサービスに似ています。国際ローミングサービスとは、携帯電話を国外に持ち出し、現地サービスのサービス地域に入ると、電話が自動的に現地サービスにログインするというサービスで、これによりどこにいても同じ番号で電話を受けられ、また自国にいるのと同様に電話をかけることができます。

セキュリティで保護された通信

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。このことは、互換アドレスであること(IPv4アドレスは簡単にIPv6アドレスに変換可能)により、および複数のトンネルを使用することにより、保証されます。[23.2.3項「IPv4とIPv6の共存」](#)を参照してください。また、システムは「デュアルスタックIP」テクニックによって、両方のプロトコルを同時にサポートできるので、2つのプロトコルバージョン間に相互干渉のない、完全に分離された2つのネットワークスタックが作成されます。

マルチキャストによるサービスの詳細なカスタマイズ

IPv4では、特定のサービス(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、サーバが、「マルチキャスト」によってホストのアドレス指定を行う、つまり、複数のホストを1つのグループの部分としてアドレス指定することで、より細かいアプローチが可能になります。これは、「ブロードキャスト」によるすべてのホストのアドレス指定や、「ユニキャスト」による各ホストの個別のアドレス指定とは異なります。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。特定の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。

23.2.2 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つの大きな制限があります。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番目の制限は、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)を組み合わせることで軽減されます。

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에만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ーに配信されます。エニーキャストアドレスは、特定のネットワーク領域で特定のサービスを提供するサーバについて、ホストが情報を得られるようにするために使用します。同じタイプのすべてのサーバは、エニーキャストアドレスが同じになります。ホストがサービスを要求すると、ルーティングプロトコルによって最も近い場所にあるサーバが判断され、そのサーバが応答します。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23.3「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 23.3: IPv6アドレスの例

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```

IPv6アドレスの各部の機能は個別に定められています。最初の4バイトはプレフィクスを形成し、アドレスのタイプを指定します。中間部分はアドレスのネットワーク部分ですが、使用しなくてもかまいません。アドレスの最後の4桁はホスト部分です。IPv6でのネットマスクは、アドレスの末尾のスラッシュの後にプレフィクスの長さを指定して定義します。例23.4「プレフィクスの長さを指定したIPv6アドレス」に示すアドレスには、最初の64ビットがアドレスのネットワーク部分を構成する情報、最後の64ビットにホスト部分を構成する情報が入っています。言い換えると、64は、ネットマスクに64個の1ビット値が左から埋められていることを意味します。IPv4と同様、IPアドレスとネットマスクのANDをとることにより、ホストが同じサブネットにあるかそうでないかを判定します。

例 23.4: 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。一部をIPv6プレフィクスに示します。

IPv6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ーが必要です。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、環境設定手順が簡素化されます。実際には、最初のアドレスビットが統合されてEUI-64-64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 (point-to-point protocol)のインタフェースのようにMACを持たないインタフェースにEUI-64トークンを割り当てられるようになります。

IPv6は、この基本構造の上で、以下の5種類のユニキャストアドレスを区別します。

:: (未指定)

このアドレスは、インタフェースが初めて初期化される時(すなわち、アドレスが他の方法で判定できないとき)に、ホストがそのソースアドレスとして使用します。

:::1 (ループバック)

ループバックデバイスのアドレス。

IPv4互換アドレス

IPv6アドレスが、IPv4アドレスおよび96個の0ビットからなるプレフィ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(23.2.3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せんが、それより広いインターネットにはルーティングしてはなりません。つまり、組織自体のネットワークの内側だけで使用するよう制限する必要があります。このようなアドレスはイントラネット用に使用され、IPv4によって定義されているプライベート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。これらのいずれかのネットワークを、MACと既知のプレフィクスを使用して自動設定できるので、IPv6が有効になると、(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。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。

23.2.3 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(23.2.2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。そのためには、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、この方法では、拡張性に欠けることになるだけでなく、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ことが障害にもなります。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。

23.2.4 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、「ルータアダプタイズプロトコル」を使用して、実装する必要のあるプレフィクスとゲートウェイをネットワークルータに問い合わせます。IPv6ルータは、`radvd`プログラムを使用して設定できます。このプログラムは、IPv6アドレスに使用するプレフィクスとルータをワークステーションに通知します。または、`zebra/quagga`を使用してアドレスとルーティングの両方を自動設定することもできます。

`/etc/sysconfig/network`ファイルを使用して複数の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。

23.2.5 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。より新しいプロトコルの詳細については、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。

RFC2460

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。

23.3 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)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、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。さらに、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。そのようなネームサーバの指定は、YaSTを使用すれば簡単です。

whois プロトコルは、DNSと密接な関係があります。このプログラムを使用すると、特定のドメインの登録者情報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DNS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。

23.4 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、[23.6項「ネットワーク接続の手動環境設定」](#)を参照してください。

SUSE Linux Enterprise Desktopでは、デフォルトでNetworkManagerがアクティブになっており、すべてのネットワークカードが設定されます。NetworkManagerがアクティブでない場合は、リンクアップしている(つまり、ネットワークケーブルが接続されている)最初のインタフェースだけが自動的に設定されます。インストール済みのシステムには、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 Desktopがサポートするすべてのタイプのネットワーク接続について、ネットワークの設定方法を説明します。

23.4.1 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、システム > 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、[23.4.1.1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、[23.4.1.3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、[23.4.1.2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、[23.4.1.4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われます。詳細については、[23.4.1.5項「ルーティングの設定」](#)を参照してください。



図 23.3: ネットワーク設定の実行

23.4.1.1 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、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。

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、[第31章「NetworkManagerの使用」](#)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、AUTO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。

23.4.1.2 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。

23.4.1.2.1 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。SUSE Linux Enterprise Desktopでは、デフォルトでNetworkManagerがアクティブになっており、すべてのネットワークカードが設定されます。

DSL回線を使用していてISP(Internet Service Provider)からスタティックIPが割り当てられていない場合も、DHCPを使用する必要があります。DHCPを使用することを選択する場合は、YaSTネットワークカード設定モジュールのネットワーク設定ダイアログにあるグローバルオプションタブのDHCPクライアントオプションで詳細を設定します。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。



注記: インタフェースのアクティブ化とリンク検出

ネットワークインタフェースのアクティブ化中に、**wicked**はキャリアを確認して、リンクが検出された場合にのみIP設定を適用します。リンク状態に関係なく設定を適用する必要がある場合(たとえば、特定のアドレスをリスンしているサービスをテストする場合など)、変数`LINK_REQUIRED=no`を`/etc/sysconfig/network/ifcfg`にあるインタフェースの設定ファイルに追加することで、リンク検出をスキップできます。

また、変数`LINK_READY_WAIT=5`を使用して、リンクを待機するタイムアウトを秒単位で指定できます。

設定ファイル`ifcfg-*`の詳細については、[23.6.2.5項「/etc/sysconfig/network/ifcfg-*」](#) および [man 5 ifcfg](#)を参照してください。

静的アドレスを使用する場合、ネームサーバとデフォルトゲートウェイは、自動的に設定されません。ネームサーバを設定するには、[23.4.1.4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、[23.4.1.5項「ルーティングの設定」](#)に従って手順を進めます。

23.4.1.2.2 複数のアドレスの設定

単一のネットワークデバイスは、エイリアスまたはラベルと呼ばれる複数のIPアドレスを持つことができます。



注記: エイリアスは互換機能

エイリアスまたはラベルはIPv4でのみ機能します。**iproute2**ネットワークインタフェースを使用すると、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の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。IPエイリアスは/32ネットマスクを使用して追加する必要があります。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。

23.4.1.2.3 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. 一般タブを開きます。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。

23.4.1.2.4 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「`=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。

23.4.1.2.5 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、システム > ネットワーク設定で検出されたカードの一覧からカードを選択し、編集をクリックします。
2. 一般タブのデバイスの起動から、適切な項目を選択します。
システムブート中にデバイスを起動するには、ブート時を選択します。ケーブル接続時では、インタフェースで物理接続が存在するかどうか監視されます。ホットプラグ時を選択した場合、インタフェースは利用可能になったときに設定されます。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 network**コマンドを使用してシャットダウンしません。また、**network**サービスは、**wicked**がアクティブになっている場合は、**wicked**サービスも処理します。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。



ヒント: ルートファイルシステムとしてのNFS

ルートパーティションがネットワーク経由でNFS共有としてマウントされている(ディスクレス)システムでは、NFS共有にアクセス可能なネットワークデバイスの設定を慎重に行う必要があります。

システムの停止、システムの再起動時のデフォルトの処理順序は、ネットワーク接続を切断してから、ルートパーティションをアンマウントするという順序になります。NFSルートの場合、この順序では問題が発生します。NFS共有とのネットワーク接続が先に無効にされているため、ルートパーティションを正常にアンマウントできないためです。システムが該当するネットワークデバイスを無効にしないようにするには、[network device configuration(ネットワークデバイスの設定)]タブ(23.4.1.2.5項「ネットワークデバイスの有効化」を参照)を開いて、デバイスの起動ペインのNFSrootオンを選択します。

23.4.1.2.6 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、システム > 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。

23.4.1.2.7 PCIe多機能デバイス

LAN、iSCSI、およびFCoEをサポートする多機能デバイスがサポートされています。FCoEクライアント(`yast2 fcoe-clientyast`)は、追加の列にプライベートフラグを表示して、ユーザがFCoE用のデバイスを選択できるようにします。ネットワークモジュール(`yast2 lanyast`)は、ネットワーク設定の「ストレージ専用デバイス」を除外します。

23.4.1.2.8 IPoIB (IP-over-InfiniBand)用のインフィニバンドの設定

1. YaSTで、システム > 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。

23.4.1.2.9 ファイアウォールの設定

『Security and Hardening Guide』、第23章「Masquerading and firewalls」、23.4項「firewalld」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。以下に手順を示します。

1. YaSTで、システム > 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。

ファイアウォール無効

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動作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。

23.4.1.3 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のシステム > 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、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す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の詳細については、**ethtool**の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)、**ifup**は-s INTERFACE_NAMEを先頭に追加します。
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、23.4.1.2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。

23.4.1.4 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のシステム > 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. ホスト名を入力します。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サーバ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ため、この機能を無効にする必要があります。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。

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、`/run/netconfig/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。通常は、このデフォルト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`/run/netconfig/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()`のマニュアルページ(`man 8 netconfig 8`)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
```

23.4.1.5 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、`default`宛先フィールドをそのままにします。



注記: ルートの優先度付け

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断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 NUMBER` を入力します。可能な最小メ

リックは0です。メトリックが最小のルートは、優先度が最も高く、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。

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。

23.5 NetworkManager

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を使用すると、移動時のネットワーク間の切り替えおよびネットワークインタフェースの設定について心配する必要がなくなります。

！ 重要

NetworkManagerは、SLEDまたはWorkstation Extensionを備えたデスクトップワークロードに対してのみSUSEでサポートされます。すべてのサーバ証明書はネットワーク設定ツールとしてwickedを使用して実行され、NetworkManagerを使用すると無効になる可能性があります。NetworkManagerは、サーバワークロードに関してSUSEでサポートされていません。

23.5.1 NetworkManagerとwicked

ただし、NetworkManagerはすべての場合に適合するソリューションではありません。したがって、ネットワーク接続を管理する場合、引き続きwickedによる制御方法またはNetworkManagerを選択できます。NetworkManagerでネットワーク接続を管理する場合は、[31.2項「NetworkManagerの有効化/無効化」](#)に従ってYaSTネットワーク設定モジュールでNetworkManagerを有効にし、NetworkManagerでネットワーク接続を設定します。ユースケースのリスト、およびNetworkManagerを設定および使用する方法の詳細については、[第31章「NetworkManagerの使用」](#)を参照してください。

次に、wickedとNetworkManagerの相違をいくつか示します。

root特権

ネットワークセットアップにNetworkManagerを使用する場合、アプレットを使用するデスクトップ環境内からいつでも簡単にネットワーク接続を切り替え、停止または開始できます。NetworkManagerでは、root特権を要求されることなく、ワイヤレスカード接続の変更および設定もできます。この理由から、NetworkManagerは、モバイルワークステーションに理想的なソリューションと言えます。

wickedには、ユーザ管理デバイスのように、ユーザによる操作の有無に関係なく接続を切り替え、停止、または開始する方法がいくつか用意されています。ただし、この場合は常に、ネットワークデバイスを変更または設定するためのroot特権が必要です。このことは、多くの場合、考えられるすべての接続を事前に設定することができないモバイルコンピューティングでは問題になります。

ネットワーク接続のタイプ

wickedとNetworkManagerの両方で、無線ネットワーク(WEP、WPA-PSK、およびWPA-Enterpriseアクセスを使用)および有線ネットワーク(DHCPと静的設定を使用)とのネットワーク接続を操作できます。これらの設定では、ダイヤルアップおよびVPNによる接続もサポートします。NetworkManagerでは、モバイルブロードバンド(3G)モデムを接続したり、DSL接続をセットアップしたりすることもできますが、これは従来の設定では不可能です。

NetworkManagerは、コンピュータが常に最適な接続を使用して接続されるようにします。ネットワークケーブルの接続が誤って切断された場合は、再接続しようとします。また、ワイヤレス接続のリストから信号強度が最高のネットワークを検出し、自動的にそれを使用して接続します。wickedと同じ機能を利用するには、より多くの設定作業が必要です。

23.5.2 NetworkManagerの機能および環境設定ファイル

NetworkManagerで作成された個別のネットワーク接続設定は、設定プロファイルに保存されます。NetworkManagerまたはYaSTで設定された「システム」接続は、/etc/NetworkManager/system-connections/*または/etc/sysconfig/network/ifcfg-*に保存されます。GNOMEの場合、すべてのユーザ定義接続はGConfに保存されます。

プロファイルが設定されていない場合は、NetworkManagerにより自動的にプロファイルが作成され、Auto \$INTERFACE-NAMEという名前が付けられます。これは、(安全性を確保しながら)可能な限り多くの場合に、設定なしで動作することを目的として作成されます。自動的に作成されたプロファイルが適切でない場合は、GNOMEにより提供されるネットワーク接続設定ダイアログを使用して必要に応じてプロファイルを変更します。詳細については、[31.3項「ネットワーク接続の設定」](#)を参照してください。

23.5.3 NetworkManager機能の制御およびロックダウン

中央管理されたマシンでは、たとえばユーザが管理者の定義した接続の変更を許可されている場合、またはユーザが独自のネットワーク設定を定義することが許可されている場合に、Polkitにより特定のNetworkManager機能を制御するか、または無効にできます。対応するNetworkManagerポリシーを表示または変更するには、Polkitのグラフィカルな認証ツールを起動します。このポリシーは、左側のツリーで、`network-manager-settings`エントリの下にあります。Polkitの概要、およびその使用方法の詳細については、『Security and Hardening Guide』、第18章「The Polkit authentication framework」を参照してください。

23.6 ネットワーク接続の手動環境設定

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。

23.6.1 **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。

従来のネットワークインタフェース管理の課題の1つは、ネットワーク管理のさまざまな層が1つのスクリプト、または最大2つの異なるスクリプトにごちゃ混ぜになってしまうことです。これらのスクリプトは、あまりはっきりしない形で互いに作用し合います。このため、予期しない問題や、不明瞭な制約、慣習などが発生します。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えない手段です。

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、特定のアドレス設定を使用して、管理者は、DHCPまたは

IPv4 zeroconfを介してインタフェースを設定するように要求することができます。この場合、アドレス設定サービスは、単にそのサーバからリースを取得し、要求されたアドレスとルートをインストールするwickedサーバプロセスに渡すだけです。

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

DHCP、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できる必要があります。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。

23.6.1.1 wickedアーキテクチャと機能

wickedサービスは、図23.4「wicked アーキテクチャ」に示されている複数の要素で構成されます。

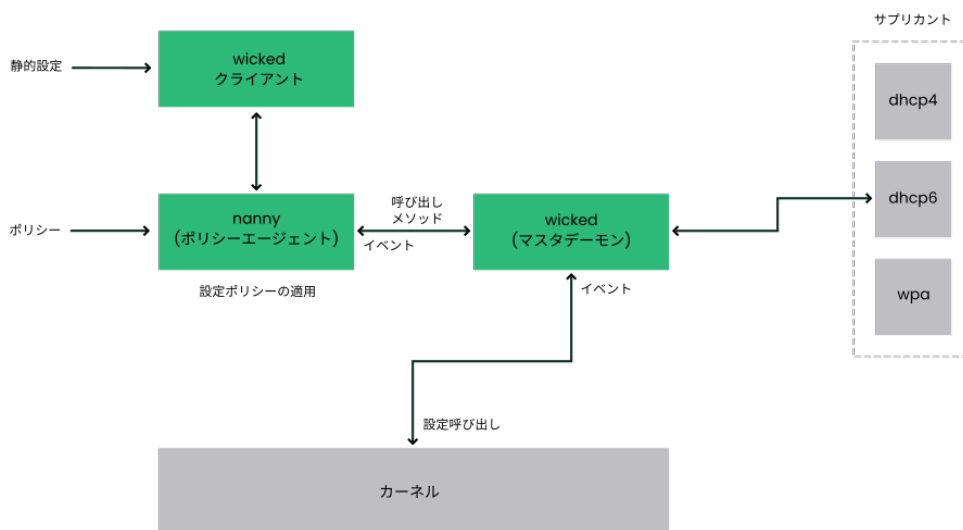


図 23.4: wicked アーキテクチャ

wickedは、現在次の要素をサポートしています。

- SUSE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。
- ネットワークインタフェース設定をXMLで表す内部環境設定バックエンド。

- 「通常の」ネットワークインタフェース(EthernetまたはInfiniBandなど)、VLAN、ブリッジ、ボンド、tun、tap、dummy、macvlan、macvtap、hsi、qeth、iucv、およびワイヤレス(現在はwpa-psk/eapネットワークに限定)デバイスの起動と停止。
- 内蔵DHCPv4クライアントおよび内蔵DHCPv6クライアント。
- nannyデーモン(デフォルトで有効)によって、デバイスが使用可能になると設定済みインタフェースが自動的に起動され(インタフェースのホットプラグ)、リンク(キャリア)が検出されるとIP設定が設定されます。詳細については、[23.6.1.3項「nanny」](#)を参照してください。
- wickedは、systemdに統合されているDBusサービスのグループとして実装されました。したがって、通常のsystemctlコマンドがwickedに適用されます。

23.6.1.2 使用 wicked

SUSE Linux Enterpriseでは、デフォルトでwickedが稼働しています。現在何が有効になっているか、稼働しているかどうかを確認するには、以下を呼び出します。

```
systemctl status network
```

wickedが有効になっている場合、以下の行に表示されます。

```
wicked.service - wicked managed network interfaces
  Loaded: loaded (/usr/lib/systemd/system/wicked.service; enabled)
  ...
```

以外が稼働している場合(NetworkManagerなど)で、wickedに切り替えたい場合、稼働中のサービスを停止してからwickedを有効にします。

```
systemctl is-active network && \
systemctl stop      network
systemctl enable --force wicked
```

これにより、wickedサービスが有効になり、network.serviceエイリアスリンクに対してwicked.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
```

wickedd(メインサーバ)と関連サブリカントが起動されます。

```
/usr/lib/wicked/bin/wickedd-auto4 --systemd --foreground
```

```
/usr/lib/wicked/bin/wickedd-dhcp4 --systemd --foreground
/usr/lib/wicked/bin/wickedd-dhcp6 --systemd --foreground
/usr/sbin/wickedd --systemd --foreground
/usr/sbin/wickedd-nanny --systemd --foreground
```

次にネットワークを起動します

```
systemctl start wicked
```

または、network.serviceエイリアスを使用します。

```
systemctl start network
```

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、次の例のように、/etc/sysconfig/network/configにWICKED_DEBUGを設定します。

```
WICKED_DEBUG="all"
```

または、いくつかを省略して、以下のようにします。

```
WICKED_DEBUG="all,-dbus,-objectmodel,-xpath,-xml"
```

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

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。firmware、次にcompatの順に重要です。これは将来変わる場合があります。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。

23.6.1.3 nanny

nannyは、イベントドリブンおよびポリシードリブンのデーモンで、デバイスのホットプラグなど、非同期や非要求のシナリオを担当します。nannyデーモンは、遅延したデバイスや、一時的に停止したデバイスの始動、再始動に役立ちます。nannyは、デバイスやリンクの変更を監視し、現行ポリシーセットで定義されている新規デバイスを統合します。Nannyは、指定されているタイムアウト制約によりifupがすでに終了していたとしても、引き続き設定されます。

nannyデーモンは、デフォルトで、システム上有効になっています。/etc/wicked/common.xml環境設定ファイルで有効に設定されています。

```
<config>
...
<use-nanny>true</use-nanny>
</config>
```

この設定によって、ifupおよびifreloadは、有効な設定を持つポリシーをnannyデーモンに適用します。nannyはwickeddを設定して、ホットプラグがサポートされます。nannyデーモンは、バックグラウンドでイベントや変更の発生まで待機します(新規デバイスやキャリアの追加など)。

23.6.1.4 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(ifcfg-bondX)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。これにより、wickedは、最上位のインタフェース名(ブリッジまたはボンドの)が指定されれば、設定全体を起動できます。

```
wicked ifup br0
```

このコマンドは、ブリッジとその依存関係を適切な順序で自動的に設定するため、依存関係(ポートなど)を別途リストする必要がありません。

1つのコマンドで複数のインタフェースを起動するには、以下のようにします。

```
wicked ifup bond0 br0 br1 br2
```

また、すべてのインタフェースを起動するには、以下のようにします。

23.6.1.5 Wickedによるトンネルの使用

Wickedでトンネルを使用する必要がある場合は、`TUNNEL_DEVICE`を使用します。これにより、オプションデバイス名を指定して、トンネルをデバイスにバインドできます。トンネル化パケットは、このデバイス経由でのみルーティングされます。

詳細については、`man 5 ifcfg-tunnel`を参照してください。

23.6.1.6 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、デバイスMTUやMACアドレスなどのリンクレベルのオプションに適用されるほか、(静的設定からDHCPに切り替える場合などは)アドレス、ルート、さらにはアドレス設定モードなどのネットワークレベルの設定にも適用されます。

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。

23.6.1.7 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ア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-service>タグにアタッチされ、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。

23.6.1.8 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ー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/run/netconfig/resolv.confを作成してDNSを設定します。

23.6.2 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。

23.6.2.1 `/etc/wicked/common.xml`

`/etc/wicked/common.xml` ファイルには、すべてのアプリケーションが使用する共通定義が含まれます。このディレクトリにある他の設定ファイルにより読み込まれ、インクルードされます。このファイルを使用して、すべての `wicked` コンポーネントのデバッグを有効にすることはできますが、その場合はファイル `/etc/wicked/local.xml` を使用することをお勧めします。保守アップデートを適用すると、`/etc/wicked/common.xml` が上書きされて、変更内容が失われる可能性があります。デフォルトインストールでは、`/etc/wicked/common.xml` ファイルに `/etc/wicked/local.xml` がインクルードされるので、通常は `/etc/wicked/common.xml` を変更する必要はありません。

`<use-nanny>` を `false` に設定して `nanny` を無効にする場合は、`wickedd.service` を再起動してから、次のコマンドを実行してすべての構成とポリシーを適用します。

```
> sudo wicked ifup all
```



注記: 環境設定ファイル

`wickedd`、`wicked`、または `nanny` の各プログラムは、それぞれの固有の設定ファイルが存在しない場合に、`/etc/wicked/common.xml` の読み込みを試みます。

23.6.2.2 `/etc/wicked/server.xml`

ファイル `/etc/wicked/server.xml` は、起動時に `wickedd` サーバプロセスによって読み込まれます。このファイルには、`/etc/wicked/common.xml` の拡張機能が保存されます。さらに、リゾルバの処理および `addrconf` サブリカント (DHCP など) からの情報の受信を設定します。

このファイルに必要な変更は、`/etc/wicked/server.xml` にインクルードされる、別ファイルの `/etc/wicked/server-local.xml` に追加することをお勧めします。別ファイルを使用することによって、保守更新中に変更内容が上書きされることはなくなります。

23.6.2.3 `/etc/wicked/client.xml`

`/etc/wicked/client.xml` は、`wicked` コマンドによって使用されます。このファイルでは、`ibft` により管理されるデバイスを検出するときに使用されるスクリプトの場所を指定し、ネットワークインタフェース設定の場所を設定します。

このファイルに必要な変更は、`/etc/wicked/server.xml` にインクルードされる、別ファイルの `/etc/wicked/client-local.xml` に追加することをお勧めします。別ファイルを使用することによって、保守更新中に変更内容が上書きされることはなくなります。

23.6.2.4 [/etc/wicked/nanny.xml](#)

[/etc/wicked/nanny.xml](#)は、リンク層の種類を設定します。設定に独自の変更を加えた場合は、保守更新時に変更内容が失われることがないように、別ファイルの[/etc/wicked/nanny-local.xml](#)にそれらの設定を追加しておくことをお勧めします。

23.6.2.5 [/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**ファイルおよびifcfg-*ファイル

wickedは、[compat:](#)プレフィクスを指定した場合にのみ、これらのファイルを読み取ります。[/etc/wicked/client.xml](#)にあるSUSE Linux Enterprise Desktop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もうとします。

[--ifconfig](#)スイッチは、多くの場合テストでのみ指定します。指定した場合、[/etc/wicked/ifconfig](#)に定義されたデフォルトの環境設定ソースは適用されません。

[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、[NETCONFIG_*](#)は、グローバル変数です。

[macvlan](#)および[macvtap](#)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、[23.6.2.6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。

23.6.2.6 `/etc/sysconfig/network/config`、`/etc/sysconfig/network/dhcp`、および`/etc/sysconfig/network/wireless`

`config`ファイルは、`ifup`、`ifdown`および`ifstatus`の動作の一般設定を含みます。`dhcp`は、DHCPの設定、およびワイヤレスLANカードの`wireless`を含みます。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`の一部の変数は、`ifcfg-*`ファイルでも使用できます。このファイルでは、それらの変数がより高い優先順位で処理されます。`/etc/sysconfig/network/ifcfg.template`ファイルは、インタフェースごとに指定できる変数を一覧表示します。ただし、`/etc/sysconfig/network/config`の変数の大半はグローバル変数であり、`ifcfg`ファイル内で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。



注記: DHCPv6の使用

SUSE Linux Enterprise 11では、IPv6 Router Advertisements (RA)が適切に設定されていないネットワークでもDHCPv6は動作しました。SUSE Linux Enterprise 12から、DHCPv6が動作するには、ネットワーク上の少なくとも1つのルータが、ネットワークがDHCPv6で管理されていることを示すRAを送出することが求められるようになりました。

ルータを正しく設定できないネットワーク用に、ユーザは`ifcfg`オプションを使用して、`DHCLIENT6_MODE='managed'`を`ifcfg`ファイルに指定することによって、この動作を無効にできます。インストールシステムで`boot`パラメータを使用することによって、この回避策を有効にできます。

```
ifcfg=eth0=dhcp6,DHCLIENT6_MODE=managed
```

23.6.2.7 `/etc/sysconfig/network/routes` および `/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 23.5: 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
```

23.6.2.8 `/var/run/netconfig/resolv.conf`

`/var/run/netconfig/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23.6「`/var/run/netconfig/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。これは `netconfig` スクリプトによって生成され、`/run/netconfig/resolv.conf`へのシンボリックリンクです。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`

設定する必要があるDNSフォワーダの名前。たとえば、`bind`または`resolver`

`NETCONFIG_DNS_RESOLVER_OPTIONS`

`/var/run/netconfig/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 23.6: `/var/run/netconfig/resolv.conf`

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

23.6.2.9 `/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。

modify

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、上書きされません。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。

remove

netconfig removeコマンドは、特定のインタフェースとサービスの組み合わせに対する編集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。

23.6.2.10 /etc/hosts

このファイル(例23.7「`/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、各エントリは空白とタブで区切ります。コメントは常に `#` 記号の後に記入します。

例 23.7: `/etc/hosts`

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

23.6.2.11 /etc/networks

このファイルには、ネットワーク名とネットワークアドレスの対応が記述されています。形式は、ネットワーク名をアドレスの前に指定すること以外は、`hosts` ファイルと同様です。例23.8「`/etc/networks`」を参照してください。

例 23.8: `/etc/networks`

```
loopback    127.0.0.0
localnet    192.168.0.0
```

23.6.2.12 /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。パラメータは常に、1行に1つずつ入力する必要があります。コメントの先頭には#記号が付きます。[表23.2「/etc/host.confファイルのパラメータ」](#)には、利用可能なパラメータが表示されます。[/etc/host.conf](#)の例については、[例23.9「/etc/host.conf」](#)を参照してください。

表 23.2: /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。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 23.9: [/etc/host.conf](#)

```
# We have named running
```

```
order hosts bind
# Allow multiple address
multi on
```

23.6.2.13 /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\(5\)](#)のマニュアルページおよび『「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23.10「/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](#)データベースのエントリによると、要求がDNS経由で[/etc/hosts \(files\)](#)に送信されています。

例 23.10: [/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23.3「/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23.4「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 23.3: [/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。
-------------------------	----------------------------------------------------------------------------------------------------

<u>ethers</u>	イーサネットアドレス。
<u>netmasks</u>	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。
<u>group</u>	<u>getgrent</u> 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。
<u>hosts</u>	<u>gethostbyname</u> 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。
<u>netgroup</u>	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。 <u>netgroup(5)</u> のマニュアルページを参照してください。
<u>networks</u>	ネットワーク名とアドレス。 <u>getnetent</u> によって使用されます。
<u>publickey</u>	NFSとNIS+によって使用されるSecure_RPCの公開鍵と秘密鍵。
<u>passwd</u>	ユーザパスワード。 <u>getpwent</u> によって使用されます。 <u>passwd(5)</u> のマニュアルページを参照してください。
<u>protocols</u>	ネットワークプロトコル。 <u>getprotoent</u> によって使用されます。 <u>protocols(5)</u> のマニュアルページを参照してください。
<u>rpc</u>	リモートプロシージャコール名とアドレス。 <u>getrpcbyname</u> および同様の関数によって使用されます。
<u>services</u>	ネットワークサービス。 <u>getservent</u> によって使用されます。

<u>shadow</u>	ユーザのシャドウパスワード。getspnamによって使用されます。 <u>shadow(5)</u> のマニュアルページを参照してください。
---------------	------------------------------------------------------------------------

表 23.4: NSS「データベース」の環境設定オプション

<u>files</u>	直接アクセスファイル。たとえば/etc/ <u>aliases</u> 。
<u>db</u>	データベース経由のアクセス。
<u>nis</u> 、 <u>nisplus</u>	NIS。『Security and Hardening Guide』、第3章「Using NIS」を参照。
<u>dns</u>	<u>hosts</u> および <u>networks</u> の拡張としてのみ使用できます。
<u>compat</u>	<u>passwd</u> 、 <u>shadow</u> および <u>group</u> の拡張としてのみ使用できます。

23.6.2.14 /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)の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、groupsおよびhostsのシステムエントリがキャッシュされます。これは、NISやLDAPのようにディレクトリサービスのパフォーマンスにとって重要です。このようになっていないと、names、groupsまたはhostsにアクセスするたびにネットワーク接続を使用する必要があります。

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。

```
> sudo systemctl restart nscd
```

23.6.2.15 /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。

23.6.3 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**の廃止

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。

23.6.3.1 **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。その一般的な構文は**ip** **OPTIONS** **OBJECT** **COMMAND**です。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。

link

ネットワークデバイスを表します。

address

デバイスのIPアドレスを表します。

neighbor

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。

rule

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。

コマンドを使用してデバイスの状態を変更します。

```
> sudo ip link set DEV_NAME
```

たとえば、デバイスeth0を無効にするには、次のコマンドを入力します

```
> sudo ip link set eth0 down
```

再度有効にするには、次のコマンドを使用します

```
> sudo ip link set eth0 up
```



ヒント: NICデバイスの接続解除

次のコマンドを使用してデバイスを無効にする場合

```
> sudo ip link set DEV_NAME down
```

ソフトウェアレベルでネットワークインタフェースが無効になります。

Ethernetケーブルが接続されていないか、接続されているスイッチがオフになっているかのように、リンクの喪失をシミュレートする場合は、次のコマンドを実行します

```
> sudo ip link set DEV_NAME carrier off
```

たとえば、`ip link set DEV_NAME down`は`DEV_NAME`を使用してすべてのルートを破棄しますが、`ip link set DEV carrier off`は破棄しません。`carrier off`にはネットワークデバイスドライバからのサポートが必要であることに注意してください。

デバイスを物理ネットワークに接続するには、次のコマンドを実行します

```
> sudo ip link set DEV_NAME carrier on
```

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。IPアドレスを設定するには、次のコマンドを使用します

```
> sudo ip addr add IP_ADDRESS + dev DEV_NAME
```

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、次のコマンドを入力します

```
> sudo ip addr add 192.168.12.154/30 brd + dev eth0
```

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、次のコマンドを入力します

```
> sudo ip route add default via gateway_ip_address
```

すべてのデバイスを表示するには、次のコマンドを使用します

```
> sudo ip link ls
```

動作しているインタフェースだけを表示する場合は、次のコマンドを使用します

```
> sudo ip link ls up
```

デバイスのインタフェース統計情報を印刷する場合は、次のコマンドを入力します

```
> sudo ip -s link ls DEV_NAME
```

特に仮想ネットワークデバイスに関する追加の役立つ情報を表示するには、次のコマンドを入力します

```
> sudo ip -d link ls DEV_NAME
```

さらに、デバイスのネットワークレイヤ(IPv4、IPv6)アドレスを表示するには、次のコマンドを入力します

```
> sudo ip addr
```

出力では、デバイスのMACアドレスに関する情報を参照することができます。すべてのルートを表示する場合は、次のコマンドを使用します

```
> sudo ip route show
```

ipの使用法の詳細については、**ip help**を入力するか、または**man 8 ip**マニュアルページを参照してください。helpオプションは、次のように、すべての**ip**サブコマンドに関して利用できます。

```
> sudo ip addr help
```

</usr/share/doc/packages/iproute2/ip-cref.pdf>で**ip**マニュアルを検索します。

23.6.3.2 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。例23.11「**ping**コマンドの出力」コマンドの実行結果例は、**ping**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。

宛先として、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、パケットを3つだけ送信する場合は、**ping -c 3 example.com**と入力します。

例 23.11: 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。そのためには、**-I**オプションに選択デバイス名を指定します。たとえば、**ping -I wlan1 example.com**のように指定します。

pingのオプションと使用方法の詳細は、ping -hとを入力するか、またはping (8)マニュアルページを参照してください。



ヒント: IPv6アドレスのping

IPv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
```

23.6.4 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のターゲット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19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。

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

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備えた、マルチユーザシステムのsystemdターゲットです。

rpcbind

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

NISサーバを起動します。

ypbind

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。

23.7 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、active-backupであり、アクティブなポートに障害が発生すると、別のBondポートがアクティブになります。以下のボンディングモードが使用可能です。

0 (balance-rr)

パケットは、ラウンドロビン方式で、最初の使用可能なインタフェースから最後の使用可能なインタフェースに送信されます。耐障害性と負荷分散を提供します。

1 (active-backup)

1つのネットワークインタフェースのみがアクティブです。失敗すると、別のインタフェースがアクティブになります。この設定は、SUSE Linux Enterprise Desktopのデフォルトです。耐障害性を提供します。

2 (balance-xor)

トラフィックは、ボンディングに含まれるデバイスの数に基づいて、使用可能なすべてのインタフェース間で分割されます。スイッチのサポートが必要です。耐障害性と負荷分散を提供します。

3 (broadcast)

すべてのトラフィックはすべてのインタフェースに対してブロードキャストされます。スイッチのサポートが必要です。耐障害性を提供します。

4 (802.3ad)

同じ速度と両面設定を共有するグループにインタフェースを集約します。インタフェースドライバでの`ethtool`のサポート、およびIEEE 802.3adダイナミックリンク集約をサポートし、それ用に設定されているスイッチが必要です。耐障害性と負荷分散を提供します。

5 (balance-tlb)

アダプティブ送信負荷分散。インタフェースドライバでの`ethtool`のサポートが必要ですが、スイッチのサポートは必要ありません。耐障害性と負荷分散を提供します。

6 (balance-alb)

アダプティブ負荷分散。インタフェースドライバでの`ethtool`のサポートが必要ですが、スイッチのサポートは必要ありません。耐障害性と負荷分散を提供します。

モードの詳細については、<https://www.kernel.org/doc/Documentation/networking/bonding.txt> を参照してください。



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。



注記: IBM POWER: ボンディングモード5および6 (balance-tlb / balance-alb)がibmvethでサポートされない

tlb/albボンディング設定と電源ファームウェアで競合が発生しています。つまり、tlb/albモードのボンディングドライバが仮想Ethernet MACアドレスとして一覧表示されているソースおよび宛先MACアドレスの両方を使用してEthernet Loopbackパケットを送信します。これらのパケットは電源ファームウェアによってサポートされていません。したがって、ボンディングモード5および6はibmvethによってサポートされません。

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 > システム > ネットワーク設定を実行します。
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. Bondポート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、ボンディングモードを選択します。
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。

23.7.1 Bondポートのホットプラグ

特定のネットワーク環境(高可用性など)では、Bondポート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、Bondポート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'
```

Bondポートは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、Bondポートインタフェースがボンドデバイスによって制御されるためです。

STARTMODE=hotplugにより、Bondポートインタフェースが利用可能になると、ボンドに自動的に追加されます。

/etc/udev/rules.d/70-persistent-net.rulesのudevルールは、MACアドレスではなく、バスID(hwinfo --netcardで表示される"SysFS BusID"に等しいudev KERNELSキーワード)によってデバイスを一致させるために変更する必要があります。これにより、異常なハードウェア(同じスロットにあるがMACが異なるネットワークカード)の交換が可能になり、ボンドがすべてのBondポートのMACアドレスを変更するときに混乱を避けることができます。

例:

```
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systemd network.serviceはホットプラグBondポートを待機しませんが、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Bondポートが利用可能であることが必要です。Bondポートインタフェースの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Bondポート名に変更し、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。

23.8 ネットワークチーミング用チームデバイスの設定

「リンク集約」という用語は、論理層を提供するためにネットワーク接続を結合(または集約)することを表す一般用語です。「チャンネルチーミング」、「Ethernetボンディング」、「ポートトランケーティング」などの用語が使用されることもありますが、これらは同義語であり、同じ概念を表しています。

これは、「ボンディング」として広く知られている概念であり、当初はLinuxカーネルに統合されていました(当初の実装については、[23.7項「ボンディングデバイスの設定」](#)を参照)。

「ネットワークチーミング」という用語は、この概念の新しい実装を表すために使用されます。

ボンディングとネットワークチーミングの主な違いは、チーミングはteamdインスタンスのインタフェースを提供する一連の小さなカーネルモジュールを供給するという点です。それ以外はすべてユーザ空間で処理されます。すべての機能がカーネル内に排他的に組み込まれている当初のボンディングの実装とは、この点が異なります。比較については、[表23.5「ボンディングとチームの機能比較」](#)を参照してください。

表 23.5: ボンディングとチームの機能比較

機能	ボンディング	チーム
ブロードキャスト、ラウンドロビンTXポリシー	yes	yes
アクティブバックアップTXポリシー	yes	yes
LACP (802.3ad)のサポート	yes	yes
ハッシュベースのTXポリシー	yes	yes
ユーザがハッシュ関数を設定可能	no	yes
TX負荷分散サポート(TLB)	yes	yes
LACPのTX負荷分散サポート	no	yes

機能	ボンディング	チーム
Ethtoolリンク監視	yes	yes
ARPリンク監視	yes	yes
NS/NA (IPV6)リンク監視	no	yes
TX/RXパスに対するRCUロック	no	yes
ポートの優先順位とスティッキネス	no	yes
ポートごとに別個のリンク監視設定	no	yes
複数のリンク監視設定	limited	yes
VLANのサポート	yes	yes
複数デバイスのスタック	yes	yes
ソース: http://libteam.org/files/teamdev.pp.pdf		

ボンディングとネットワークチーミングの両方の実装は、並行して使用できます。ネットワークチーミングは、既存のボンディング実装の代替手段です。ボンディングがネットワークチーミングに置き換わるわけではありません。

ネットワークチーミングは、さまざまな事例で使用できます。次の技術に関連する最も重要な2つの事例について、後で説明します。

- 複数のネットワークデバイス間での負荷分散
- ネットワークデバイスの1つに障害が発生した場合の、別のデバイスへのフェールオーバー

現在は、チーミングデバイスの作成をサポートするYaSTモジュールは存在しません。ネットワークチーミングは手動で設定する必要があります。一般的な手順を次に示します。この手順は、あらゆるネットワークチーミング設定に適用できます。

手順 23.1: 一般的な手順

1. 必要なパッケージがすべてインストールされていることを確認します。パッケージ `libteam-tools`、`libteamdctl0`、および `python-libteam` をインストールします。

2. `/etc/sysconfig/network/`に設定ファイルを作成します。通常は、`ifcfg-team0`という名前を付けます。複数のネットワークチーミングデバイスが必要な場合は、昇順に番号を付けます。

この設定ファイルで使用するさまざまな変数については、マニュアルページ(`man ifcfg`および`man ifcfg-team`)を参照してください。設定例は、システム内にあるファイル`/etc/sysconfig/network/ifcfg.template`で参照できます。

3. チーミングデバイスに使用するインタフェースの設定ファイル(通常は`ifcfg-eth0`および`ifcfg-eth1`)を削除します。
どちらのファイルも、バックアップを作成してから削除することを推奨します。Wickedが、チーミングに必要なパラメータを含む設定ファイルを再作成します。
4. 必要に応じて、Wickedの設定ファイルにすべてのパラメータが含まれているかどうかを確認します。

```
> sudo wicked show-config
```

5. ネットワークチーミングデバイス`team0`を起動します。

```
> sudo wicked ifup all team0
```

詳しいデバッグ情報が必要な場合は、`all`サブコマンドの後に`--debug all`オプションを指定します。

6. ネットワークチーミングデバイスのステータスを確認します。それには、次のコマンドを実行します。

- Wickedから`teamd`インスタンスの状態を取得します。

```
> sudo wicked ifstatus --verbose team0
```

- インスタンス全体の状態を取得します。

```
> sudo teamdctl team0 state
```

- `teamd`インスタンスの`systemd`状態を取得します。

```
> sudo systemctl status teamd@team0
```

これらは必要に応じて少しずつ異なる情報を表示します。

7. 後で`ifcfg-team0`ファイルの内容を一部変更する必要がある場合は、次のコマンドでその設定を再ロードします。

```
> sudo wicked ifreload team0
```

チームングデバイスを起動または停止する場合、**systemctl**を使用「しない」でください。代わりに、上記の**wicked**コマンドを使用します。

チームデバイスを完全に削除するには、次の手順を実行します。

手順 23.2: チームデバイスの削除

1. ネットワークチームングデバイス team0 を停止します。

```
> sudo wicked ifdown team0
```

2. ファイル `/etc/sysconfig/network/ifcfg-team0` の名前を `/etc/sysconfig/network/.ifcfg-team0` に変更します。ファイル名の先頭にドットを挿入することにより、wicked でファイルが「非表示」になります。設定が本当に必要ない場合は、ファイルを削除することもできます。

3. 設定を再ロードします。

```
> sudo wicked ifreload all
```

23.8.1 使用事例: ネットワークチームングによる負荷分散

負荷分散は帯域幅を改善するために使用されます。次の設定ファイルを使用して、負荷分散機能を備えたネットワークチームングデバイスを作成します。手順23.1「一般的な手順」に従ってデバイスを設定します。**teamdctl**の出力を確認します。

例 23.12: ネットワークチームングによる負荷分散の設定

```
STARTMODE=auto ❶  
BOOTPROTO=static ❷  
IPADDRESS="192.168.1.1/24" ❷  
IPADDR6="fd00:deca:fbad:50::1/64" ❷  
  
TEAM_RUNNER="loadbalance" ❸  
TEAM_LB_TX_HASH="ipv4,ipv6,eth,vlan"  
TEAM_LB_TX_BALANCER_NAME="basic"  
TEAM_LB_TX_BALANCER_INTERVAL="100"  
  
TEAM_PORT_DEVICE_0="eth0" ❹  
TEAM_PORT_DEVICE_1="eth1" ❹  
  
TEAM_LW_NAME="ethtool" ❺  
TEAM_LW_ETHTOOL_DELAY_UP="10" ❻
```

- ① チーミングデバイスの起動を制御します。値`auto`は、インタフェースが、ネットワークサービスを使用可能な場合に設定され、再起動時に毎回自動的に起動されることを意味します。
デバイスを手動で制御する(自動的に起動しないようにする)必要がある場合は、`STARTMODE`を`manual`に設定します。
- ② 静的IPアドレス(この場合、IPv4では`192.168.1.1`、IPv6では`fd00:deca:fbad:50::1`)を設定します。
ネットワークチーミングデバイスが動的IPアドレスを使用する必要がある場合は、`BOOTPROTO="dhcp"`を設定し、`IPADDRESS`と`IPADDR6`の行を削除(またはコメント)します。
- ③ `TEAM_RUNNER`を`loadbalance`に設定して、負荷分散モードを有効にします。
- ④ ネットワークチーミングデバイスを作成するために集約する必要がある1つまたは複数のデバイスを指定します。
- ⑤ 従属デバイスの状態を監視するリンクウォッチャを定義します。デフォルト値`ethtool`は、デバイスが起動していてアクセス可能かどうかのみを確認します。その場合、この確認にはほとんど時間はかかりません。ただし、デバイスが実際にパケットを送受信できるかどうかの確認は行われません。
接続でさらに高い信頼性が必要な場合は、`arp_ping`オプションを使用します。これにより、任意のホスト(`TEAM_LW_ARP_PING_TARGET_HOST`変数で設定されている)にpingが送信されます。返信を受信した場合のみ、このネットワークチーミングデバイスが起動しているものとみなされます。
- ⑥ リンクが起動(または停止)してからランナに通知されるまでの遅延(ミリ秒)を定義します。

23.8.2 使用事例: ネットワークチーミングによるフェールオーバー

フェールオーバーは、並行して動作するバックアップネットワークデバイスを使用することにより、重要なネットワークチーミングデバイスの高可用性を確保するために使用します。バックアップネットワークデバイスは常時実行され、メインデバイスに障害が発生すると処理を引き継ぎます。

次の設定ファイルを使用して、フェールオーバー機能を備えたネットワークチーミングデバイスを作成します。手順23.1「一般的な手順」に従ってデバイスを設定します。`teamdctl`の出力を確認します。

例 23.13: DHCPネットワークチーミングデバイスの設定

```
STARTMODE=auto ❶  
BOOTPROTO=static ❷  
IPADDR="192.168.1.2/24" ❷  
IPADDR6="fd00:deca:fbad:50::2/64" ❷  
  
TEAM_RUNNER=activebackup ❸  
TEAM_PORT_DEVICE_0="eth0" ❹  
TEAM_PORT_DEVICE_1="eth1" ❹  
  
TEAM_LW_NAME=ethtool ❺  
TEAM_LW_ETHTOOL_DELAY_UP="10" ❻  
TEAM_LW_ETHTOOL_DELAY_DOWN="10" ❻
```

- ❶ チーミングデバイスの起動を制御します。値`auto`は、インタフェースが、ネットワークサービスを使用可能な場合に設定され、再起動時に毎回自動的に起動されることを意味します。
デバイスを手動で制御する(自動的に起動しないようにする)必要がある場合は、`STARTMODE`を`manual`に設定します。
- ❷ 静的IPアドレス(この場合、IPv4では`192.168.1.2`、IPv6では`fd00:deca:fbad:50::2`)を設定します。
ネットワークチーミングデバイスが動的IPアドレスを使用する必要がある場合は、`BOOTPROTO="dhcp"`を設定し、`IPADDRESS`と`IPADDR6`の行を削除(またはコメント)します。
- ❸ `TEAM_RUNNER`を`activebackup`に設定して、フェールオーバーモードを有効にします。
- ❹ ネットワークチーミングデバイスを作成するために集約する必要がある1つまたは複数のデバイスを指定します。
- ❺ 従属デバイスの状態を監視するリンクウォッチャを定義します。デフォルト値`ethtool`は、デバイスが起動していてアクセス可能かどうかのみを確認します。その場合、この確認にはほとんど時間はかかりません。ただし、デバイスが実際にパケットを送受信できるかどうかの確認は行われません。
接続でさらに高い信頼性が必要な場合は、`arp_ping`オプションを使用します。これにより、任意のホスト(`TEAM_LW_ARP_PING_TARGET_HOST`変数で設定されている)にpingが送信されます。返信を受信した場合のみ、このネットワークチーミングデバイスが起動しているものとみなされます。
- ❻ リンクが起動(または停止)してからランナに通知されるまでの遅延(ミリ秒)を定義します。

23.8.3 使用事例: チームデバイス上でのVLAN

VLANは「Virtual Local Area Network」(仮想ローカルエリアネットワーク)の略です。複数の「論理」(仮想)Ethernetを1つの物理Ethernet上で実行できます。ネットワークを論理的に複数のブロードキャストドメインに分割し、パケットが同じVLANに指定されたポート間でのみ切り替えられるようにします。

次の使用例では、チームデバイス上に静的なVLANを2つ作成します。

- vlan0。IPアドレス192.168.10.1にバインドされます。
- vlan1。IPアドレス192.168.20.1にバインドされます。

以下に手順を示します。

1. スイッチでVLANタグを有効にします。チームデバイスに負荷分散を使用するには、スイッチが「Link Aggregation Control Protocol」(LACP) (802.3ad)に対応している必要があります。詳細については、ハードウェアマニュアルを参照してください。
2. チームデバイスで負荷分散またはフェールオーバーのどちらを使用するかを決定します。23.8.1項「使用事例: ネットワークチーミングによる負荷分散」または23.8.2項「使用事例: ネットワークチーミングによるフェールオーバー」の説明に従ってチームデバイスを設定します。
3. /etc/sysconfig/network内に、次の内容が含まれるファイルifcfg-vlan0を作成します。

```
STARTMODE="auto"  
BOOTPROTO="static" ①  
IPADDR='192.168.10.1/24' ②  
ETHERDEVICE="team0" ③  
VLAN_ID="0" ④  
VLAN='yes'
```

- ① IPADDRで指定された固定IPアドレスを定義します。
 - ② IPアドレスを定義します。ここではネットマスクを一緒に定義しています。
 - ③ VLANインタフェースに使用する実際のインタフェースが含まれます。ここでは、チームデバイス(team0)です。
 - ④ VLANの固有IDを指定します。できれば、ファイル名とVLAN_IDは、名前ifcfg-vlanVLAN_IDに対応させます。この場合、VLAN_IDは0で、ファイル名はifcfg-vlan0になります。
4. ファイル/etc/sysconfig/network/ifcfg-vlan0を/etc/sysconfig/network/ifcfg-vlan1にコピーして、次の値を変更します。

- IPADDRは、192.168.10.1/24から192.168.20.1/24に変更。
- VLAN_IDは、0から1に変更。

5. 2つのVLANを起動します。

```
# wicked ifup vlan0 vlan1
```

6. ifconfigの出力を確認します。

```
# ifconfig -a
[...]
vlan0      Link encap:Ethernet  HWaddr 08:00:27:DC:43:98
            inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
            inet6 addr: fe80::a00:27ff:fedc:4398/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 b)  TX bytes:816 (816.0 b)

vlan1      Link encap:Ethernet  HWaddr 08:00:27:DC:43:98
            inet addr:192.168.20.1 Bcast:192.168.20.255 Mask:255.255.255.0
            inet6 addr: fe80::a00:27ff:fedc:4398/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 b)  TX bytes:816 (816.0 b)
```

24 プリンタの運用

SUSE® Linux Enterprise Desktopは、リモートネットワークプリンタなどの、さまざまな種類のプリンタを使用し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、[第34章「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、[24.8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 Desktop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を使用して、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、特定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ーは存在しません。

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。通常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、[24.8.1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.openprinting.org/printers> 

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 Desktopがリリースされた時点では、そのステータスに達していなかった可能性があります。そのため、これらのデータベースは必ずしも正しいステータスを表しているとは限らず、おおよその状況を提示するだけにとどまっています。

<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。

24.1 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で構成されます。これには、プリンタの名前やプリントキューの名前のほか、オプションでフィルタに関する情報(プリンタ固有のオプションなど)が含まれます。

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。

24.2 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、複数の方法があります。CUPS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、https://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。



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を注意してください。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。

24.3 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。24.7.3項「各種パッケージ内のPPDファイル」および24.8.2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(34.1.1項「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 Desktop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、メーカーのソフトウェアをインストールすることをお勧めしません。

24.4 ネットワークプリンタ

ネットワークプリンタは、複数の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、特定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、いくつかの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、プレー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ーによる実装は、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ています。IPPを使用する場合、他のプロトコルより、ジョブとの関連性が高いデータが送信されます。CUPSは、IPPを使用して内部のデータ送信を行います。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。この目的で使用するプロトコルは、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 IP.OF.THE.PRINTER
```

24.5 コマンドラインツールによるCUPSの設定

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。

```
> sudo 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。

```
> sudo 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。

```
> sudo 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:

```
> sudo 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。

```
> sudo 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。

```
> sudo lpadmin -p QUEUE -o Resolution=600dpi
```

3. 新しい設定値の確認:

```
> sudo 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。

24.6 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、「`lp -d QUEUENAME FILENAME`」を入力し、QUEUENAMEおよびFILENAMEを対応する名前で置き換えます。

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。通常はFILENAMEを指定しません。たとえば、「`lp -d QUEUENAME`」と入力します。

24.7 SUSE Linux Enterprise Desktopの特別な機能

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

24.7.1 CUPSとファイアウォール

デフォルトのSUSE Linux Enterprise Desktopインストールを完了した後、`firewalld`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publicゾーンに設定されます。

`firewalld`がアクティブな場合は、内部ネットワークゾーンを介してmdnsおよびippを許可することにより、クライアントがネットワークプリンタをブラウズできるように設定する必要があります。パブリックゾーンでは、プリンタキューを公開しないでください。

(`firewalld`の設定の詳細については、『Security and Hardening Guide』、第23章

「Masquerading and firewalls」、23.4項「`firewalld`」およびhttps://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settingsを参照してください。)

24.7.1.1 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Internal Zone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。

24.7.1.2 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Internal Zone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。

24.7.2 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/DNS-SDが使用されます。ネットワーク印刷キューの参照を有効にするには、`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`**コマンドを使用します。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`**コマンドを実行してサービスを有効にします。

`cups-browsed`を起動してもブラウズできない場合は、CUPSサーバがBonjour/DNS-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`**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。

`firewalld`を介してプリンタの参照を許可する方法の詳細については、[24.7.1項「CUPSとファイアウォール」](#)を参照してください。

24.7.3 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

24.8 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。

24.8.1 標準的なプリンタ言語をサポートしないプリンタ

これらのプリンタは、共通のプリンタ言語をサポートしておらず、独自のコントロールシーケンスを使用しないと対処できません。そのため、これらのプリンタは、メーカーがドライバを添付した特定のバージョンのオペレーティングシステムでのみ動作します。GDIは、Microsoft*がグラフィックデバイス用に開発したプログラミングインタフェースです。通常、メーカーはWindows用のドライバだけを提供しており、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ことです。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。

一部のメーカーは、プリンタに独自規格のドライバを提供しています。独自規格のプリンタドライバの欠点は、インストール済みの印刷システムとそのドライバを組み合わせたときに動作するという保証も、複数の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。

24.8.2 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、プリンタメーカーのドライバCDにあるPPDファイルを使用したり、プリンタメーカーの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ーに問い合わせることも考えられます。

24.8.3 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、lpd上の515 (ポートHOST)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実行し、リモートHOST上のQUEUEに関するステータスレポート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、例24.1「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 24.1: 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24.2「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 24.2: 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
```

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、**cupsd**がアクティブになっていないか、ネットワークの基本的な問題があります。**lpstat -h HOST -l-t**は、それぞれの**cupsd**がアクティブであり、ホストがクエリを受け入れる場合、**HOST**上のすべてのキューのステータスレポートを返します。

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。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マシンのトラブルシューティング

印刷サーバマシン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ある場合、問題を引き起こすことがあります。これは印刷サーバマシン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、印刷サーバマシンに接続されているプリンタに直接送信することで、印刷サーバマシン内のスプーラを使用しないようにします。[24.4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、印刷サーバマシン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、印刷サーバマシン内にある、該当するTCPポートについて把握する必要があります。プリンタが印刷サーバマシンに接続されていて、電源がオンになっている場合、印刷サーバマシン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、印刷サーバマシン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、印刷サーバマシン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_PORT IP_ADDRESS**コマンドを使用します。これは、ある程度の時間を要することがあります。詳細な情報については、**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。

24.8.4 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。

24.8.5 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ため、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。

24.8.6 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。

サーバ上の印刷ジョブを削除するには、`lpstat -h cups.example.com -o`などのコマンドを使用して、サーバ上のジョブ番号を判別します。これは、サーバが印刷ジョブを完了(つまり、プリンタに完全に送信)していないことが前提となります。取得したジョブ番号を使用して、次のようにサーバ上の印刷ジョブを削除します。

```
> cancel -h cups.example.com QUEUE-JOBNUMBER
```

24.8.7 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。

印刷ジョブが破損しているか、ホストとプリンタ間の通信に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ず、判読不能な文字を含む多数の用紙を印刷します。この問題を解決するには、次の手順を実行します。

1. プリンタの動作を停止するために、インクジェットプリンタの場合、すべての用紙を取り除きます。レーザープリンタの場合、用紙トレイを開けます。上位機種のプリンタでは、現在のプリントアウトをキャンセルするボタンを用意していることもあります。
2. この時点で、印刷ジョブはキューに残っている可能性があります。ジョブがキューから削除されるのは、ジョブ全体をプリンタへ送信した後に限られるからです。`lpstat -o`または`lpstat -h 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`を使用して印刷ジョブを削除します。
3. 印刷ジョブがすでにキューから削除され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。

24.8.8 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で**LogLevel debug**を設定します。
2. cupsd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を開始します。
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。

24.8.9 詳細情報

SUSE Linux Enterprise Desktopでの印刷の詳細については、openSUSE Support Database (<https://en.opensuse.org/Portal:Printing>)にアクセスしてください。SUSE Knowledgebase (<https://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。

25 グラフィカルユーザインタフェース

SUSE Linux Enterprise Desktopには、X.orgサーバ、Wayland、およびGNOMEデスクトップが含まれています。この章では、すべてのユーザのグラフィカルユーザインタフェースの環境設定について説明します。

25.1 Xウィンドウシステム

X.orgサーバは、X11プロトコルを実装するための事実上の標準です。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。

X Window Systemは、ほとんどの場合設定不要です。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。

SUSE Linux Enterprise Desktop 15 SP5では、X.orgサーバの代替機能としてWaylandが含まれています。この機能はインストール時に選択可能です。

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-ati`)。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います。たとえば、`ati`ドライバを使用する場合は、`man 4 ati`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-nvidiaG04`でマニュアルを参照できます。

サーバにパッケージ`xrdp`をインストールし、RDPクライアントソフトウェアを使用して、リモートデスクトッププロトコルを介してサーバにアクセスします。

25.2 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ーを選択する
- 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、またはパスパなどのスクリプトおよびその他の高レベルのテキスト処理に必要な、より洗練されたOpenType形成は、Harfbuzz (<http://www.harfbuzz.org/>)  またはPango (<http://www.pango.org/>)  を使用して行われます。

25.2.1 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ー、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。詳しくは「**man 1 fc-list**」を参照してください。

25.2.2 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ー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。

25.2.3 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。

```
> 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。

```
> 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。

```
> 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。

25.2.4 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。システム全体で使え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、[25.2.5.2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/) (<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、SUSE Linux Enterprise Desktop 15 SP5のリポジトリを追加するには次のようにします。

```
> sudo zypper ar  
https://download.opensuse.org/repositories/M17N:/fonts/SLE_15/
```

FONT_FAMILY_NAMEを検索するには、次のコマンドを使用します。

```
> zypper se 'FONT_FAMILY_NAME*fonts'
```

25.2.5 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。

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。

25.2.5.1 sysconfigによるフォントの設定

SUSE Linux Enterprise Desktop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

PREFER_*_FAMILIES変数を介してビットマップフォントを優先します。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。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、太く(場合によっては不鮮明な)、コントラストの低い文字形状になります。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。

25.2.5.2 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25.1「レンダリングアルゴリズムを指定する」を参照)。

例 25.1: 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 25.2: 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25.2「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。

<default> from **ルール1**

このルールは、serifファミリ名をPFLの「末尾」に追加します。

<prefer> from **ルール2**

このルールは、PFLに「が存在する場合、PFLで」「が最初に出現する箇所の「直前」に」Droid SerifserifAlegreya SCを追加します。

<accept> from **ルール3**

このルールは、PFLで「ファミリ名が最初に出現する箇所の」直後serif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25.1「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 25.1: FONTCONFIGルールからのPFLの作成

順序	現在のPFL
要求	<u>Alegreya SC</u>
ルール1	<u>Alegreya SC</u> 、 <u>serif</u>
ルール2	<u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u>
ルール3	<u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u> 、 <u>STIXGeneral</u>

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25.2「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 25.2: 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	注意
要求	<u>Alegreya SC</u>	同じ要求が実行されます。
ルール2	<u>Alegreya SC</u>	<u>serif</u> がPFLに存在しないため、何も置換されません。
ルール3	<u>Alegreya SC</u>	<u>serif</u> がPFLに存在しないため、何も置換されません。
ルール1	<u>Alegreya SC</u> 、 <u>serif</u>	<u>Alegreya SC</u> がPFL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25.3「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 25.3: 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25.2「エイリアスとファミリ名の置換」を例25.4「エイリアスとファミリ名の置換」で拡張します。

例 25.4: 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>

```



```
</prefer>
</alias>
```

例25.4「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 25.3: FONTCONFIGルールからのPFL生成結果

順序	現在のPFL
要求	<u>Alegreya SC</u>
ルール1	<u>Alegreya SC</u> 、 <u>serif</u>
ルール2	<u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u>
ルール3	<u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u> 、 <u>STIXGeneral</u>
ルール4	<u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u> 、 <u>Liberation Serif</u> 、 <u>STIXGeneral</u>
ルール5	<u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>DejaVu Serif</u> 、 <u>serif</u> 、 <u>Liberation Serif</u> 、 <u>STIXGeneral</u>



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。

25.3 管理者用のGNOME設定

25.3.1 dconfシステム

GNOMEデスクトップの設定は、`dconf`で管理されます。ユーザが個人用設定を変更し、システム管理者がすべてのユーザ用のデフォルト値または必須値を設定することが可能な階層構造のデータベースまたはレジストリです。GNOME 2の`dconf`システムの代わりに`gconf`が使用されます。

グラフィカルユーザインタフェースで`dconf`オプションを表示するには、`dconf-editor`を使用します。コマンドラインで設定オプションにアクセスして変更を行うには、`dconf`を使用します。

GNOME Tweaksツールは、通常のGNOME設定以外の追加の設定オプション用の使いやすいユーザインタフェースを提供します。このツールは、GNOMEアプリケーションメニューから、またはコマンドラインから`gnome-tweak-tool`を使用して起動できます。

25.3.2 システム全体の設定

グローバル`dconf`設定パラメータは、`/etc/dconf/db/`ディレクトリで設定できます。これには、GDMの設定やユーザ用の特定の設定オプションのロックが含まれます。

例として次の手順を使用し、システム全体の設定を作成します。

1. `/etc/dconf/db/`内に`.d`で終わる新しいディレクトリを作成します。このディレクトリには、設定オプションを含む任意の量のテキストファイルを格納することができます。この例では、次の内容のファイル`/etc/dconf/db/network.d/00-proxy`を作成します。

```
# This is a comment
[system/proxy/http]
host='10.0.0.1'
enabled=true
```

2. 新しい設定ディレクティブを`dconf`データベース形式に解析します。

```
> sudo dconf update
```

3. ファイル`/etc/dconf/profiles/user`を作成して、デフォルトユーザプロファイルに新しい`network`設定データベースを追加します。次に、以下の内容を追加します。

```
system-db:network
```

ファイル`/etc/dconf/profiles/user`はGNOME(デフォルト)です。他のプロファイルは、環境変数`DCONF_PROFILE`で定義できます。

4. オプション: ユーザのプロキシ設定をロックするには、ファイル`/etc/dconf/db/network/locks/proxy`を作成します。次に、変更できないキーと共にこのファイルに行を追加します。

```
/system/proxy/http/host  
/system/proxy/http/enabled
```

グラフィカルな`dconf-editor`を使用して1人のユーザのプロファイルを作成し、次に`dconf dump /`を使用してすべての設定オプションを一覧表示することができます。この場合、設定オプションは、グローバルプロファイルに格納することができます。

グローバル設定の詳細については、<https://wiki.gnome.org/Projects/dconf/SystemAdministrators>  を参照してください。

25.3.3 Waylandでのタイリングウィンドウの管理

WaylandでGNOMEデスクトップを実行していて、ウィンドウを並べて配置したい場合は、「gTile」 GNOME拡張機能を使用することをお勧めします。gTileを使用すると、設定可能なグリッドスキームに基づいてウィンドウを再配置できます。たとえば、6つの端末を起動して、デスクトップ領域全体を使用するように3行と2列に配置することができます。

25.3.3.1 gTileのインストール

重要

gTile GNOME拡張機能を使用するには、GNOMEデスクトップを実行していて、`chrome-gnome-shell` (SLED 15 SP4以前の場合)または`gnome-browser-connector` (SLED 15 SP5以降の場合)パッケージがインストールされている必要があります。


1. ブラウザで、gTile拡張機能のWebページ(<https://extensions.gnome.org/extension/28/gtile/> )を指定します。ブラウザ拡張機能がないことに関する警告が表示された場合は、リンクをクリックして、GNOME Shell拡張機能のインストールと有効化方法に関するWebブラウザの指示に従ってください。



図 25.1: GNOME SHELL拡張機能がないことに関する警告

- GNOME Shell拡張機能がWebブラウザにインストールされると、gTile Webページの右上にOFFボタンが表示されます。gTile拡張機能を有効にするには、それをクリックしてONに切り替えます。



図 25.2: GNOME GTILE拡張機能が有効化されました

- GNOMEパネルの右側にあるGNOMEトレイにgTileトレイアイコンが表示されます。これを使用して、gTileウィンドウレイアウトを有効にできます。



図 25.3: GTILEトレイアイコン

25.3.3.2 gTile拡張機能の使用

手順 25.1: ウィンドウの配置

- 現在フォーカスされているウィンドウのgTileを開くには、**Meta** + **Enter** キーを押します。
- ウィンドウを配置する列数を設定するには、番号キーを押します。

3. Autotile Main window sizes設定に従ってウィンドウを自動的にタイル化するには、**M** キーを押します。デフォルトでは、初めて **M** キーを押すと、現在のウィンドウが画面の半分を使用し、他のすべてのウィンドウが画面の残りの半分の列にタイル化されて表示されます。**M** キーを繰り返し押して、バリエーションを切り替えることができます。これは、大きなメインウィンドウといくつかの小さなセカンダリウィンドウを保持するのに役立ちます。
4. 現在のウィンドウを画面の事前定義されたタイル内に移動するには、矢印キーを使用します。
ウィンドウが占めるタイル数を減らす、または増やすには、**Shift** キーと矢印キーを使用します。
5. ウィンドウを隣接するウィンドウとデスクトップの端にスナップするには、**Ctrl + Alt + S** キーを押します。
6. gTileウィンドウを閉じるには、**Esc** キーを押します。

手順 25.2: 詳細設定

1. gTile設定を開くには、次のコマンドを実行します：**`gnome-extensions-app`**。
2. gTileを選択し、設定をクリックします。
3. プリセットとキーの組み合わせを定義して、頻繁に使用されるウィンドウレイアウトを簡単に切り替えます。
gTile拡張機能の設定方法の詳細については、<https://github.com/gTile/gTile> を参照してください。

25.3.4 詳細情報

詳細については、<http://help.gnome.org/admin/> を参照してください。

25.4 SUSE Primeを使用したIntelとNVIDIA Optimus GPUの切り替え

SUSE Primeは、オンボードのIntelグラフィックプロセッシングユニット(GPU)と、NVIDIAの「切り替え可能なグラフィックス」Optimusテクノロジーを搭載したNVIDIA GPUを切り替えるためのツールです。OptimusはオンボードのIntel GPUとディスクリートNVIDIA GPUを簡単に

切り替えるためのメカニズムを提供します。これは、ラップトップを省電力モードまたは最大のパフォーマンスで実行するように設計されています。電力を節約するためにIntel GPUを使用し、3DアプリケーションにはNVIDIA GPUを使用します。

SUSE Primeは、SUSE Linux Enterprise Workstation Extension for SUSE Linux Enterprise 15 SP5に含まれています。

SUSE PrimeはWaylandではなく、X11を実行するシステムでのみ動作します。ご使用のシステムでWaylandを実行している場合に、SUSE Primeを使用するには、Waylandを無効にして、X11にフォールバックしてください(25.4.1項「前提条件」を参照)。

25.4.1 前提条件

SUSE Linux Enterprise 15 SP5 (25.4.3項「NVIDIAドライバのインストール」を参照)に含まれているNVIDIAドライバを使用した設定済みで動作するNVIDIA Optimus GPUと、オンボードIntel GPUが必要です。NVIDIA Optimus用の古いスイッチングツールであるBumblebeeはインストールしないでください。

`/etc/X11/xorg.conf`ファイル、および`/etc/X11/xorg.conf.d`ディレクトリにアクティブな「ServerLayout」、「Device」、または「Screen」セクションを持つ設定ファイルがあってはなりません。

SUSE PrimeはX11でのみ動作します。**loginctl**コマンドを使用して、システムがX11またはWaylandを使用しているかどうかを確認します。

```
> loginctl
  SESSION      UID USER      SEAT      TTY
      2        1000 tux        seat0
> loginctl show-session 2|grep Type
Type=x11
```

システムでWaylandを使用している場合は、`/etc/gdm/custom.conf`を編集して、`WaylandEnable=false`のコメントを解除して無効にしてください。次に再起動します。

25.4.2 SUSE Primeのインストールと使用

NVIDIAグラフィックスカードがすでに取り付けられ、機能しているはずです。このダイアログボックスが開いていない場合は、25.4.3項「NVIDIAドライバのインストール」を参照してください。

`suse-prime`パッケージをインストールします。

```
> sudo zypper install suse-prime
```

GPUを切り替えるには、次のコマンドのいずれかを実行して、ログアウトしてからログインします。

```
> sudo prime-select intel
> sudo prime-select intel2
> sudo prime-select nvidia
```

モード設定ドライバの場合は**intel**ドライバを使用してください。**intel2**は、`xf86-video-intel`ドライバを使用するシステム用です。この情報は、`inxi`をインストールして実行することで:

```
> inxi -G
Graphics: Device-1: Intel Xeon E3-1200 v3/4th Gen Core Processor Integrated Graphics Controller
          Display Server: x11(X.org 1.20.1 ) drivers: modesetting (unloaded: fbdev, vesa)
          Resolution: 1920x1080@60.00hz
          OpenGL: renderer: Mesa DRI Intel Haswell Desktop version: 4.5 Mesa 18.2.8
```

どのGPUが現在アクティブですか?

```
> sudo /usr/sbin/prime-select get-current
Driver configured: intel
```

25.4.3 NVIDIAドライバのインストール

使用するドライバがわかるようにNVIDIAカードを識別する必要がある場合は、次のコマンドを実行します。

```
> /sbin/lspci | grep VGA
```

次の手順に従って、Zypperでドライバをインストールします。

使用可能なドライバパッケージを一覧表示します。

```
> sudo zypper se nvidia
```

次に、NVIDIAグラフィックカード用ドライバをインストールします。

```
> sudo zypper se packagename
```


26 FUSEによるファイルシステムへのアクセス

FUSEは、「file system in user space」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)。

26.1 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。

通常は、FUSEを設定する必要はありません。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。

26.2 NTFSパーティションのマウント

NTFS(「New Technology File System」)は、Windowsのデフォルトのファイルシステムです。通常の場合では、特権のないユーザは外部のFUSEライブラリを使用してNTFSブロックデバイスをマウントできません。そのため、次に説明する方法でWindowsパーティションをマウントするには、ルート特権が必要です。NTFSパーティションのマウントは、SUSE Linux Enterprise ServerおよびSUSE Linux Enterprise Desktop with SUSE Linux Enterprise Workstation Extensionでのみサポートされています。

1. `root`になって、パッケージ`ntfs-3g`をインストールします。これはSUSE Linux Enterprise Workstation Extensionで提供されています。
2. マウントポイントとして使用するディレクトリ(`~/mounts/windows`など)を作成します。
3. 必要なWindowsパーティションを見つけます。YaSTを使用し、パーティシヨナモジュールを起動して、Windowsに属するパーティションを確認します(ただし、何も変更しないでください)。代替として、`root`になって、`/sbin/fdisk -l`を実行することもできます。パーティションタイプ`HPFS/NTFS`のパーティションを捜します。

4. 読み書きモードでパーティションをマウントします。プレースホルダ`DEVICE`を各Windowsパーティションで置き換えます。

```
> ntfs-3g /dev/DEVICE MOUNT POINT
```

Windowsパーティションを読み込み専用モードを使用するには、`-o ro`を追加します。

```
> ntfs-3g /dev/DEVICE MOUNT POINT -o ro
```

コマンド`ntfs-3g`は、現在のユーザ(UID)とグループ(GID)を使用して、所定のデバイスをマウントします。書き込みパーミッションを別のユーザに設定するには、コマンド`id USER`を使用して、UID値とGID値の出力を取得します。次のコードで設定してください。

```
# id tux
uid=1000(tux) gid=100(users) groups=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE MOUNT POINT -o uid=1000,gid=100
```

その他のオプションについては、マニュアルページを参照してください。

リソースをアンマウントするには、`fusermount -u MOUNT POINT`を実行します。

26.3 詳細情報

詳細については、<https://github.com/libfuse/libfuse>にあるFUSEのホームページを参照してください。

27 複数バージョンのカーネルのインストール

SUSE Linux Enterprise Serverでは、複数バージョンのカーネルを並行でインストールできます。2番目のカーネルをインストールすると、ブートエントリとinitrdfが自動的に作成されるので、手動での設定が別途必要になることはありません。マシンを再起動すると、新しく追加したカーネルが追加のブートパラメータとして利用できるようになります。

この機能を使用すると、カーネルのアップデートを安全な状態でテストでき、実績のある以前のカーネルにいつでもフォールバックできます。そのためには、YaSTのオンラインアップデートやアップデートアプレットなどのアップデートツールを使用せず、この章で説明するプロセスに従います。



警告: サポートエンタイトルメント

独自にコンパイルしたカーネルやサードパーティのカーネルをインストールすると、マシンのサポートエンタイトルメントが全面的に無効になります。SUSE Linux Enterprise Desktopに付属するカーネルおよびSUSE Linux Enterprise Desktopの正式なアップデートチャンネルで配布されるカーネルのみがサポートされています。



ヒント: ブートローダ設定カーネルの確認

別のカーネルをインストールした後は、デフォルトのブートエントリを目的に合わせて設定するために、ブートローダ設定を確認することをお勧めします。詳細については、[18.3項「YaSTによるブートローダの設定」](#)を参照してください。

27.1 マルチバージョンサポートの有効化と設定

SUSE Linux Enterprise Server 12以降、複数バージョンのソフトウェアパッケージのインストール(マルチバージョンサポート)は、デフォルトで有効になっています。この設定を確認するには、次の手順に従います。

1. 任意のエディタで、rootとして`/etc/zypp/zypp.conf`を開きます。
2. 文字列`multiversion`を検索します。この機能に対応しているすべてのカーネルパッケージでマルチバージョンサポートが有効になっている場合、次の行はコメント解除された状態で表示されます。

```
multiversion = provides:multiversion(kernel)
```

3. マルチバージョンサポートを特定のカーネルに限定するには、それらのパッケージの名前をカンマ区切りリストとして `/etc/zypp/zypp.conf` の `multiversion` オプションに追記します。たとえば、次のような記述とします。

```
multiversion = kernel-default, kernel-default-base, kernel-source
```

4. 変更を保存します。



警告: KMP (カーネルモジュールパッケージ)

アップデートした新しいカーネル用に、ベンダーが提供する必須のカーネルモジュール (カーネルモジュールパッケージ) もインストールされていることを確認してください。最終的にカーネルモジュールが見つからなくても、カーネルアップデートプロセスで警告は表示されません。これは、システム上に保持されている古いカーネルによって依然としてパッケージ要件が満たされているためです。

27.1.1 使用していないカーネルの自動削除

マルチバージョンサポートを有効にして新しいカーネルを頻繁にテストしていると、ブートメニューが急速に複雑になります。通常 `/boot` パーティションの容量は限られているため、`/boot` オーバーフローによるトラブルが発生する場合があります。YaSTや以下で説明する Zypper などを使用して、使用されていないバージョンのカーネルを手動で削除できますが、このようなカーネルを自動的に削除するように `libzypp` を設定することもできます。デフォルトでは、どのカーネルも削除されません。

1. 任意のエディタで、`root` として `/etc/zypp/zypp.conf` を開きます。
2. 文字列 `multiversion.kernels` を検索し、その行のコメント指定を解除することで、このオプションを有効にします。このオプションは、以下の各値のカンマ区切りリストをとります。

`5.3.18-53.3`: 指定されたバージョン番号のカーネルを保持します。

`latest`: 最新のバージョン番号のカーネルを保持します。

`latest-N`: N 番目に新しいバージョン番号のカーネルを保持します。

`running`: 実行しているカーネルを保持します。

oldest: 最も古いバージョン番号のカーネルを保持します(これはSUSE Linux Enterprise Desktopに元から付属しているバージョンのカーネルです)。

oldest+N. N番目に古いバージョン番号のカーネルを保持します。
次の例を示します

```
multiversion.kernels = latest,running
```

最新バージョンのカーネルおよび現在実行しているカーネルを保持します。これはマルチバージョン機能を有効にしない場合に似ていますが、古いバージョンのカーネルが削除される時期が、インストールの直後ではなく、「次のリブートの後」である点が異なります。

```
multiversion.kernels = latest,latest-1,running
```

最新とその次に新しいバージョンのカーネルおよび現在実行しているカーネルを保持します。

```
multiversion.kernels = latest,running,5.3.18-53.3
```

最新バージョンのカーネル、現在実行しているカーネル、および5.3.18-53.3バージョンのカーネルを保持します。



ヒント: 実行しているカーネルの保持

特殊な設定を使用している場合を除き、runningというマークの付いたカーネルは常に保持してください。

実行しているカーネルを保持しない場合、カーネルは更新時に削除されます。つまり、実行しているカーネルのモジュールも削除され、ロードできなくなります。

実行しているカーネルを保持しない場合は、必ずカーネルのアップグレード後すぐに再起動して、モジュールの問題が発生しないようにしてください。

27.1.2 使用事例: 再起動後にのみ古いカーネルを削除する

新規カーネルを使用して正常にシステムを再起動できてから、古いカーネルを削除するようにします。

/etc/zypp/zypp.conf内の以下の行を変更します。

```
multiversion.kernels = latest,running
```

前のパラメータは、最新のカーネル、および実行中のカーネル(最新でない場合のみ)を保持するようにシステムに指示します。

27.1.3 使用事例: 古いカーネルをフォールバックとして保持する

1つ以上のバージョンのカーネルを保持し、「スペア」として保有しておきます。

これは、テスト用のカーネルが必要な場合に役立ちます。何らかの問題が生じたとき(マシンが起動しないなど)、正常に動作していた1つ以上のバージョンのカーネルをまだ使用することができるためです。

/etc/zypp/zypp.conf内の以下の行を変更します。

```
multiversion.kernels = latest,latest-1,latest-2,running
```

新しいカーネルのインストール後にシステムを再起動する際、システムは3つのカーネルを保持します。現在のカーネル(latest,runningとして設定)と、その直前のカーネルつ(latest-1およびlatest-2として設定)です。

27.1.4 使用事例: 特定のバージョンのカーネルの保持

通常、定期的にシステムアップデートを行い、新規バージョンのカーネルをインストールします。しかし、独自のバージョンのカーネルをコンパイルして、システムに保持させたい場合があります。

/etc/zypp/zypp.conf内の以下の行を変更します。

```
multiversion.kernels = latest,5.3.18-53.3,running
```

新規カーネルのインストール後にシステムを再起動するときに、システムは2つのカーネルを保存します。新規の実行中のカーネル(latest,runningと設定)および独自でコンパイルしたカーネル(5.3.18-53.3と設定)の2つです。

27.2 YaSTによる複数のカーネルバージョンのインストールと削除

次の手順で、YaSTを使用して複数のカーネルをインストールまたは削除できます。

1. YaSTを起動し、ソフトウェア > Software Management (ソフトウェア管理)を選択してソフトウェアマネージャを開きます。

2. 表示 > パッケージの分類 > 複数バージョンのパッケージを選択して、複数のバージョンを提供できるすべてのパッケージを一覧表示します。

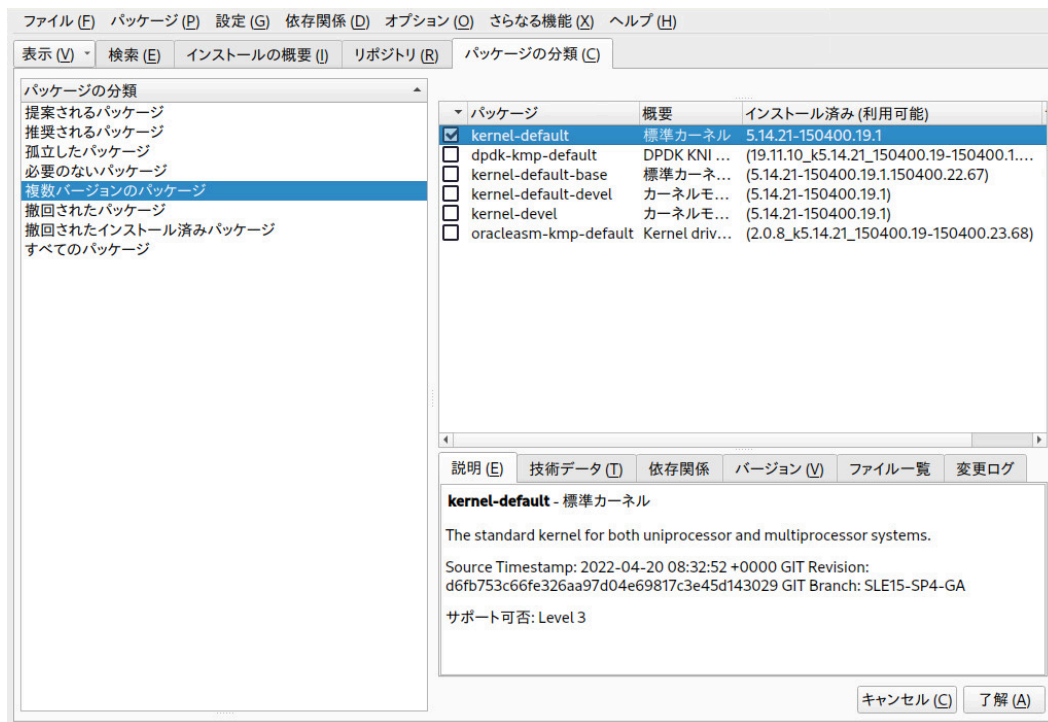


図 27.1: YASTソフトウェアマネージャ: マルチバージョン表示

3. パッケージを選択し、そのパッケージのバージョンタブを下部ペインの左側で開きます。
4. パッケージをインストールするには、そのパッケージの横のチェックボックスをクリックします。インストールの対象として選択されていることを示す緑色のチェックマークが表示されます。
すでにインストール済みのパッケージ(白いチェックマークで表示)を削除するには、削除の対象として選択されていることを示す赤色のXが表示されるまで、そのパッケージの横のチェックボックスをクリックします。
5. 了解をクリックしてインストールを開始します。

27.3 Zypperによる複数のカーネルバージョンのインストールと削除

次の手順で、**zypper**を使用して複数のカーネルをインストールまたは削除できます。

1. コマンド `zypper se -s 'kernel*'` を使用して、存在するすべてのカーネルパッケージを一覧表示します。

S	Name	Type	Version	Arch	Repository
-----+-----+-----+-----+-----					
i+	kernel-default	package	5.14.21-150400.6.3	x86_64	SLE-Module-Basesystem15-SP4-Pool
	kernel-default-base	package	5.14.21-150400.6.3.150400.22.27	x86_64	SLE-Module-Basesystem15-SP4-Pool
	kernel-default-devel	package	5.14.21-150400.6.3	x86_64	SLE-Module-Basesystem15-SP4-Pool
	kernel-devel	package	5.14.21-150400.6.4	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-all	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-amdgpu	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-ath10k	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-ath11k	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-atheros	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-bluetooth	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-bnx2	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-brcm	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-chelsio	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-dpaa2	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-i915	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-intel	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-iwlwifi	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-liquidio	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-marvell	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-media	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-mediatek	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-mellanox	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-mwifiex	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-network	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-nfp	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-nvidia	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool
i	kernel-firmware-platform	package	20220119-150400.1.1	noarch	SLE-Module-Basesystem15-SP4-Pool

i	kernel-firmware-prestera	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-qcom	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-qlogic	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-radeon	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-realtek	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-serial	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-sound	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-ti	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-ueagle	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
i	kernel-firmware-usb-network	package	20220119-150400.1.1	noarch	SLE-Module-
	Basesystem15-SP4-Pool				
	kernel-macros	package	5.14.21-150400.6.4	noarch	SLE-Module-
	Basesystem15-SP4-Pool				

2. インストールする場合は、次のように正確なバージョンを指定します。

```
> sudo zypper in kernel-default-5.3.18-53.3
```

3. カーネルをアンインストールする場合は、コマンド `zypper se -si 'kernel*'` を使用して、インストールされているすべてのカーネルを一覧表示し、`zypper rm` PACKAGENAME-VERSION を使用して、目的のパッケージを削除します。

28 カーネルモジュールの管理

Linuxはモノリシックカーネルですが、カーネルモジュールを使用して拡張することができます。カーネルモジュールは、オンデマンドでカーネルに挿入したり、カーネルから削除したりできる特別なオブジェクトです。実際面では、カーネルモジュール自体に含まれないドライバやインタフェースを追加および削除できます。Linuxは、カーネルモジュールを管理するためのコマンドをいくつか備えています。

28.1 lsmodおよびmodinfoによるロード済みモジュールの一覧作成

lsmodコマンドを使用すると、現在ロードされているカーネルモジュールを表示できます。コマンドの出力は次のようになります。

```
> lsmod
Module                Size  Used by
snd_usb_audio         188416  2
snd_usbmidi_lib       36864   1 snd_usb_audio
hid_plantronics       16384   0
snd_rawmidi           36864   1 snd_usbmidi_lib
snd_seq_device        16384   1 snd_rawmidi
fuse                  106496   3
nfs_v3                 45056   1
nfs_acl               16384   1 nfs_v3
```

出力は3つの列に分かれています。Module列には、ロード済みモジュールの名前が一覧にされ、Size列には各モジュールのサイズが表示されます。Used by列には、参照モジュールの数と名前が表示されます。このリストは不完全な場合があります。

特定のカーネルモジュールに関する詳細情報を表示するには、**modinfo** MODULE_NAME コマンドを使用します。MODULE_NAMEには、目的のカーネルモジュールの名前を指定します。**modinfo**バイナリは、ユーザのPATH環境変数に存在しない/sbinディレクトリにあります。つまり、**modinfo**コマンドを標準ユーザとして実行する場合、バイナリのフルパスを指定する必要があります。

```
> /sbin/modinfo kvm
filename:      /lib/modules/5.3.18-57-default/kernel/arch/x86/kvm/kvm.ko.xz
license:      GPL
author:       Qumranet
suserelease:  SLE15-SP3
srcversion:   3D8FBA9060D4537359A06FC
depends:       irqbypass
```

```
supported:    yes
retpoline:    Y
intree:       Y
name:         kvm
vermagic:     5.3.18-57-default SMP mod_unload modversions
```

28.2 カーネルモジュールの追加と削除

`insmod`と`rmmod`を使用してカーネルモジュールを追加および削除できますが、これらの代わりに`modprobe`ツールを使用することをお勧めします。`modprobe`には、依存関係の自動解決やブラックリスト化など、重要な利点があります。

パラメータを指定せずに使用すると、`modprobe`コマンドは、指定したカーネルモジュールをインストールします。`modprobe`はルート特権で実行する必要があります。

```
> sudo modprobe acpi
```

カーネルモジュールを削除するには、`-r`パラメータを使用します。

```
> sudo modprobe -r acpi
```

28.2.1 ブート時のカーネルモジュールの自動ロード

カーネルモジュールを手動でロードする代わりに、`systemd-modules-load.service`サービスを使用してブートプロセス中に自動的にロードできます。カーネルモジュールを有効にするには、`.conf`ファイルを`/etc/modules-load.d/`ディレクトリに追加します。次の例のように、設定ファイルにモジュールと同じ名前を付けることをお勧めします。

```
/etc/modules-load.d/rt2800usb.conf
```

設定ファイルには目的のカーネルモジュールの名前を記述する必要があります(例: `rt2800usb`)。

ここで説明する方法を使用すると、パラメータなしでカーネルモジュールをロードできます。特定のオプションを指定してカーネルモジュールをロードする必要がある場合は、代わりに`/etc/modprobe.d/`ディレクトリに設定ファイルを追加します。ファイルには拡張子`.conf`を付ける必要があります。ファイルの名前は、`priority-modulename.conf`という命名規則に従う必要があります。たとえば、`50-thinkfan.conf`のようにします。設定ファイルには、カーネルモジュールの名前と目的のパラメータを記述する必要があります。次のコマンド例を使用すると、カーネルモジュールの名前とそのパラメータが記述された設定ファイルを作成できます。

```
> echo "options thinkpad_acpi fan_control=1" | sudo tee /etc/modprobe.d/thinkfan.conf
```



注記: カーネルモジュールのロード

ほとんどのカーネルモジュールは、デバイスが検出されたときか、ユーザ空間によって特定の機能が要求されたときに、システムによって自動的にロードされます。したがって、ほとんどの場合、モジュールを手動で `/etc/modules-load.d/` に追加する必要はありません。

28.2.2 modprobeによるカーネルモジュールのブラックリスト化

カーネルモジュールをブラックリスト化すると、そのカーネルモジュールはブートプロセス中にロードされなくなります。これは、システムで問題を引き起こす疑いがあるモジュールを無効にする場合に便利です。なお、カーネルモジュールをブラックリスト化しても、`insmod` ツールまたは `modprobe` ツールを使用してそのカーネルモジュールを手動でロードできます。

モジュールをブラックリストに登録するには、次の内容を含むファイル `/etc/modprobe.d/60-blacklist-MODULE_NAME.conf` を作成します。

```
blacklist MODULE_NAME
```

rootとして`dracut`コマンドを実行して新しいinitrdイメージを生成し、マシンを再起動します(`NAME`を現在のinitrdの名前に置き換え、`KERNELVERSION`を現在実行中のカーネルに置き換えます)。

```
> su
echo "blacklist nouveau" >> /etc/modprobe.d/60-blacklist-nouveau.conf
/usr/bin/dracut --logfile /var/log/YaST2/mkinitrd.log --force /boot/${initrd-NAME}
${KERNELVERSION}
reboot
```

カーネルモジュールを一時的にのみ無効にするには、ブート時にオンザフライでブラックリスト化します。そのためには、ブート画面が表示されたら **E** キーを押します。最小限のエディタが表示され、そこでブートパラメータを変更できます。次のような行を見つけます。

```
linux /boot/vmlinuz...splash= silent quiet showopts
```

行の最後に `modprobe.blacklist=MODULE_NAME` コマンドを追加します。例:

```
linux /boot/vmlinuz...splash= silent quiet showopts modprobe.blacklist=nouveau
```

F10 キーまたは **Ctrl + X** キーを押し、指定した設定でブートします。

GRUBを使用してカーネルモジュールを完全にブラックリスト化するには、`/etc/default/grub` ファイルを編集用を開き、**GRUB_CMD_LINUX** コマンドに **modprobe.blacklist=MODULE_NAME** オプションを追加します。次に **sudo grub2-mkconfig -o /boot/grub2/grub.cfg** コマンドを実行して、変更を有効にします。

29 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、検出されたら、設定し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して、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。

29.1 /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ディレクトリ内に存在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。

29.2 カーネルuevents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/usr/lib/udev/rules.d/*.rulesおよび/etc/udev/rules.d/*.rulesファイルからすべての規則を読み込んで解析し、メモリ内に保持します。

規則ファイルが変更、追加、または削除されると、このデーモンは、`udevadm control --reload` コマンドで、メモリに再ロードできます。`udev`のルールとそれらの構文の詳細については、29.6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 `uevents` は、カーネルのネットリンクソケットから受信されます。

29.3 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。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ったら、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。

29.4 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一イベントを再送します。`/sys`内のすべての`uevent`ファイルを含む単純なループにより、すべてのイベントが再びデバイスノードを作成し、デバイスセットアップを実行します。

たとえば、ブート時に存在するUSBマウスは、ドライバがその時点で使用できないため、初期のブートロジックでは初期化されない場合があります。デバイス検出イベントは、消失し、そのデバイスのカーネルモジュールは検出されません。接続されている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。

29.5 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。

29.6 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。イベントは、ディレクトリ/usr/lib/udev/rules.d/(デフォルト規則用)および/etc/udev/rules.d(システム固有の設定)で提供されるすべての規則と照合されます。

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udevデフォルトルールセット/usr/lib/udev/rules.d/50-udev-default.rulesに含まれています。

例 29.1: udevルール例

```
# 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/usb/lp0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、*、?、[]パターンと一致します。
- udevルールは、置換をサポートします。

29.6.1 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。

29.6.2 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

値 DEVPATH。

%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}、\$env{VARIABLE}

環境変数の値(VARIABLEで指定)。

%c、\$result

PROGRAMの出力。

%%

%文字。

\$\$

\$文字。

29.6.3 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}=="0n[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。

29.6.4 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、いくつかのタイプの変数をインポートします。タイプが指定されていない場合、udevは、ファイルパーミッションの実行可能ビットに基づいてタイプを決定しようとします。

- program - 外部プログラムを実行しその出力をインポートするように、udevに指示します。
- file - テキストファイルをインポートするように、udevに指示します。
- parent - 親デバイスから保存されたキーをインポートするように、udevに指示します。

WAIT_FOR_SYSFS

udevに、指定されたsysfsファイルが特定のデバイス用に作成されるのを待機するように指示します。たとえば、WAIT_FOR_SYSFS="ioerr_cnt"は、udevに、ioerr_cntファイルが作成されるまで待機するように通知します。

OPTIONS

OPTIONキーには、いくつかの値を指定できます。

- last_rule - 以降のすべての規則を無視するように、udevに指示します。
- ignore_deviceは、このイベントを無視するようにudevに指示します。

- `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するように、`udev`に指示します。
- `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するように、`udev`に指示します。

29.7 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
```

29.8 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/*

規則と一致するシステム固有のudevイベント。/usr/lib/udev/rules.d/*からデフォルトの規則を変更するか、上書きするには、ここでカスタム規則を追加できます。ファイルはアルファベット順に解析されます。優先度の高いファイルの規則は優先度の低い規則を変更または上書きします。数が小さくなればなるほど、優先度が高くなります。

/usr/lib/udev/rules.d/*

規則と一致するデフォルトudevイベント。このディレクトリのファイルはパッケージにより所有され、更新で上書きされます。ここでファイルを追加、削除、または編集しないでください。代わりに、/etc/udev/rules.dを使用してください。

/usr/lib/udev/*

udevルールから呼び出されるヘルパープログラム

/usr/lib/tmpfiles.d/ および /etc/tmpfiles.d/

静的/devコンテンツを管理します。

29.9 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。

udev

udevイベント管理デーモンに関する情報

30 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することをお勧めします。この章の最後では、言語および国固有設定(l18nおよびl10n)について説明します。

30.1 特殊ソフトウェアパッケージに関する情報

次の章では、次のツールに関する基本情報を提供します。bash、cron、logrotate、locate、ulimit、およびfree。

30.1.1 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。

30.1.2 cronパッケージ

cronを使用すると、事前に定義された時間にバックグラウンドでコマンドを自動的に実行できます。cronは特別な形式のタイムテーブルを使用し、ツールには複数のデフォルトのタイムテーブルが付属しています。必要に応じて、ユーザはカスタムテーブルを指定することもできます。

cronテーブルは/var/spool/cron/tabsに置かれます。/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30.1「/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マニュアルページ(man cron)を参照してください。

例 30.1: /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30.2「/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 30.2: /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、ユーザが長時間、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。たとえば、`/etc/cron.daily`には、コンポーネント`suse.de-backup-rpmdb`、`suse.de-clean-tmp`または`suse.de-cron-local`があります。

30.1.3 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。

30.1.4 ログファイル: パッケージlogrotate

カーネルそのものと一緒に、定期的にシステム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」)が複数あります。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。詳細については、『*System Analysis and Tuning Guide*』、第3章「System log files」、3.3項「Managing log files with **logrotate**」を参照してください。

30.1.5 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。

30.1.6 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30.1「**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 30.1: **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/.profile` で行う必要があります。

例 30.3: **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` を参照してください。



重要: **ulimit** 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。

30.1.7 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。

30.1.8 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。

30.1.9 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 MAN_PAGE**」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、最初の手動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。

30.1.10 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オプションを使用して設定を行う場合、この設定は、`~/.gnu-emacs-custom`に保存されます。

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

これらのファイルの詳細については、`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、次のいくつかのパッケージに分かれています。

- 基本パッケージの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。

30.2 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。

30.3 キーボード割り当て

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。



ヒント: 詳細情報

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。

30.4 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。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で定義されたLC_変数で行います。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(**locale**コマンドでmanページを参照)。

変数のリスト

RC_LC_MESSAGES, RC_LC_CTYPE, RC_LC_COLLATE, RC_LC_TIME, RC_LC_NUMERIC,
RC_LC_MONETARY

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

RC_LC_ALL

この変数は、すでに参照された変数の値を上書きします。

RC_LANG

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

ROOT_USES_LANG

この変数はyesまたはctype (デフォルト)に設定できます。yesに設定される場合、rootは言語および国固有の設定を使用します。設定されていない場合、システム管理者は常にPOSIX環境で作業します。

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結合されます。

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

30.4.1 システム全体のロケール設定

systemdは初期起動時に/etc/locale.confを読み込みます。このファイルで設定されたロケール設定は、個別の設定がない限り、すべてのサービスまたはユーザによって継承されます。



注記: SUSE Linux Enterprise Desktopでの古い設定ファイルの動作

以前のバージョンのSUSE Linux Enterprise Desktopはロケール設定を/etc/sysconfig/language、/etc/sysconfig/keyboard、および/etc/sysconfig/consoleから読み込んでいました。SUSE Linux Enterprise Desktop 15 GA以降、これらのファイルは廃止されたものとみなされています。systemdでは、これらのファイルから設定を読み込まなくなりました。代わりに、systemdでは、/etc/locale.confを読み込みます。

ただし、/etc/sysconfig/languageで定義されている変数は引き続き使用されます。これらの変数はシステム全体のロケールを上書きし、ユーザシェルの異なるロケール設定を定義するために使用することができます(30.4.2項「例」を参照)。

システム全体のロケールを設定するには、次のいずれかを実行できます。

- /etc/locale.confに設定を書き込む。各行は環境に似た変数割り当てです(変数のリストについては、[man 5 locale.conf](#)を参照してください):

```
LANG=de_DE.UTF-8
```

設定を微調整するには、1行に1つ変数を追加することができます。

- **localectl**コマンドを使用する:

```
# localectl set-locale LANG=de_DE.UTF-8
```

ここでも、**localectl set-locale**コマンドの後に追加の変数を指定することもできます。

systemdパッケージの更新中に古いシステムとの後方互換性を維持するために、言及されているすべての変数は、まだ定義されていない場合は、sysconfigから最終的な宛先にマイグレートされます。

30.4.2 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せん。

`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。

30.4.3 ~/.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クスは付けません。たとえば、RC_LANGの代わりにLANGを使用します。

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

30.4.4 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_N0"

LANGUAGE="nn_N0:nb_N0:no"

あるいは、

LANG="nb_N0"

LANGUAGE="nb_N0:nn_N0:no"

ノルウェー語では、LC_TIMEの扱いも違います。

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。

30.4.5 詳細情報

- 「『The GNU C Library Reference Manual』」の「Locales and Internationalization」の章。パッケージ glibc-info。
- 『「UTF-8 and Unicode FAQ for Unix/Linux」』、Markus Kuhn著。Webページ<https://www.cl.cam.ac.uk/~mgk25/unicode.html>  (現在のアドレス)を参照してください。

31 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。

！ 重要

NetworkManagerは、SLEDまたはWorkstation Extensionを備えたデスクトップワークロードに対してのみSUSEでサポートされます。すべてのサーバ証明書はネットワーク設定ツールとしてwickedを使用して実行され、NetworkManagerを使用すると無効になる可能性があります。NetworkManagerは、サーバワークロードに関してSUSEでサポートされていません。

31.1 NetworkManagerの使用事例

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。

31.2 NetworkManagerの有効化/無効化

デスクトップおよび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。YaSTのネットワーク設定モジュールを使用して、いつでも無効にしたり有効にしたりできます。

1. YaSTを実行し、システム > 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. 31.3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外部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、23.4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。

31.3 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、無線、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。



ヒント: NetworkManager接続エディタ

以前のSUSE Linux Enterprise Desktopリリースでは、ネットワーク接続は「NetworkManager接続エディタ」と呼ばれるアプリケーションを使用して設定されていました。「GNOMEコントロールセンター」がその設定機能を完全に置き代えたため、これはデフォルトではインストールされなくなりました。

NetworkManager接続エディタを使用してネットワーク接続を設定する必要がある場合は、`NetworkManager-connection-editor`パッケージを手動でインストールしてください。

```
> sudo zypper install NetworkManager-connection-editor
```

GNOMEで「Network Configuration (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。詳細は、システム管理者にお問い合わせください。

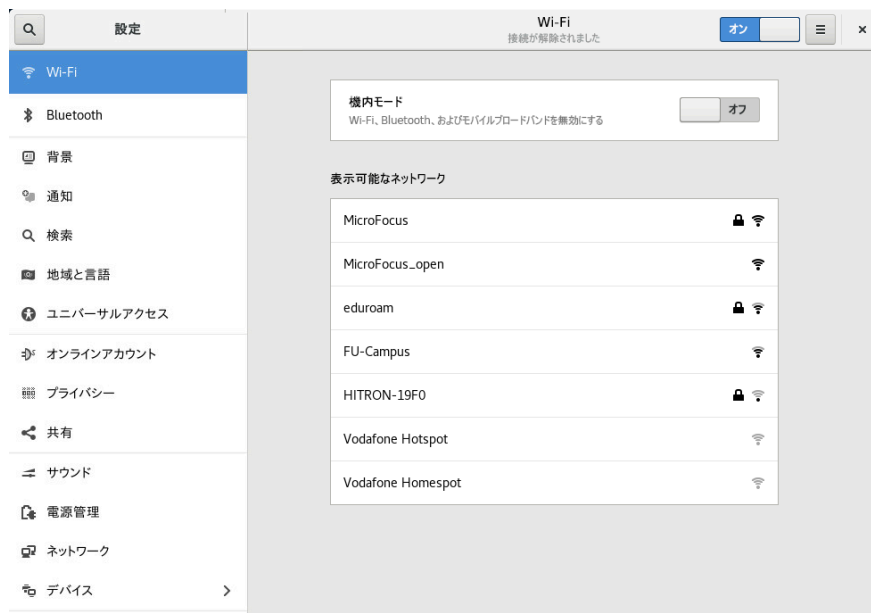


図 31.1: GNOMEネットワーク接続のダイアログ

手順 31.1: 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. 変更を確認した後で、新しく設定されたネットワーク接続が、[ステータス]メニューの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、31.4.1項「ユーザおよびシステムの接続」を参照してください。

31.3.1 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、[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。

31.3.2 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。暗号化された無線ネットワークには、シールドアイコンが付きます。

手順 31.2: 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 31.3: 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せず、自動的に検出されないネットワークに接続するには、[状態] メニューを開き、Wi-Fi (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。

31.3.3 無線でのキャプティブポータル検出の有効化

初回接続時には、公共の無線ホットスポットの多くでランディングページ (キャプティブポータル) へのアクセスが必要になります。ユーザがログインする前に、またはライセンス条項に同意する前に、すべてのHTTPリクエストはプロバイダのキャプティブポータルにリダイレクトされます。

キャプティブポータルから無線ネットワークに接続する場合、NetworkManagerおよびGNOMEでは、接続プロセスの一部としてログインページが自動的に表示されます。これにより、ユーザはいつ接続したかを常に把握できます。また、ブラウザを使用してログインせずに、できるだけ早く設定することができます。

この機能を有効にするには、パッケージ `NetworkManager-branding-SLE` をインストールし、次のコマンドでNetworkManagerを再起動します。

```
> sudo systemctl restart network
```

キャプティブポータルからネットワークに接続すると、NetworkManager (またはGNOME) によってキャプティブポータル用のログインページが表示されます。ユーザの資格情報を使用してログインすると、インターネットにアクセスできるようになります。

31.3.4 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。

31.3.5 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 Desktopには、NetworkManagerに対する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、次のパッケージをインストールします。

- [NetworkManager-openvpn](#)
- [NetworkManager-openvpn-gnome](#)

OpenConnect

このVPN技術を使用するには、次のパッケージをインストールします。

- [NetworkManager-openconnect](#)
- [NetworkManager-openconnect-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、次のパッケージをインストールします。

- [NetworkManager-pptp](#)
- [NetworkManager-pptp-gnome](#)

次の手順は、NetworkManagerを使用してコンピュータをOpenVPNクライアントとして設定する方法を示しています。他のタイプのVPNも同様の手順で設定します。

最初に、パッケージ[NetworkManager-openvpn-gnome](#)がインストールされ、すべての依存関係が解決されていることを確認します。

手順 31.4: NETWORKMANAGERによるOPENVPNの設定

1. パネル右端のステータスアイコンをクリックしてwrench and screwdriver (レンチとスクリュードライバ)アイコンをクリックし、アプリケーションの設定を開きます。All Settings (すべての設定)ウィンドウで、ネットワークを選択します。
2. +アイコンをクリックします。
3. VPN、OpenVPNの順に選択します。
4. 認証タイプを選択します。OpenVPNサーバのセットアップに応じて、Certificates (TLS) (証明書(TLS))またはPassword with Certificates (TLS) (パスワードと証明書(TLS))を選択します。
5. 各テキストボックスに必要な値を入力します。設定例では、次のようになります。

ゲートウェイ	VPNサーバのリモートエンドポイント
--------	--------------------

User name (ユーザ名)	ユーザ(Password with Certificates (TLS) (パスワードと証明書(TLS))が選択されている場合のみ)
パスワード	ユーザのパスワード(Password with Certificates (TLS) (パスワードと証明書(TLS))が選択されている場合のみ)
User Certificate (ユーザ証明書)	<u>/etc/openvpn/client1.crt</u>
CA Certificate (CA証明書)	<u>/etc/openvpn/ca.crt</u>
Private Key (秘密鍵)	<u>/etc/openvpn/client1.key</u>

- 追加をクリックして、設定を完了します。
- 接続を有効にするには、設定アプリケーションのネットワークパネルでスイッチボタンをクリックします。または、パネル右端のステータスアイコンをクリックし、使用するVPNの名前をクリックして接続をクリックします。

31.4 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。

31.4.1 ユーザおよびシステムの接続

NetworkManagerは、userおよびsystemという2種類の接続を認識します。

ユーザ接続では、すべてのユーザがNetworkManagerで認証を受ける必要があります。NetworkManagerは、ユーザの資格情報をローカルGNOMEキーリングに保存するため、接続するたびに再入力する必要はありません。

システム接続はすべてのユーザが自動的に利用できます。接続を作成する最初のユーザが必要な資格情報を入力すると、その後、他のすべてのユーザは資格情報を知らなくてもアクセスできます。ユーザとシステムの接続設定の違いは、単一のチェックボックスMake available to other users (他のユーザが利用できるようにする)です。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、31.3項「ネットワーク接続の設定」を参照してください。

31.4.2 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。

31.4.3 ファイアウォールゾーン

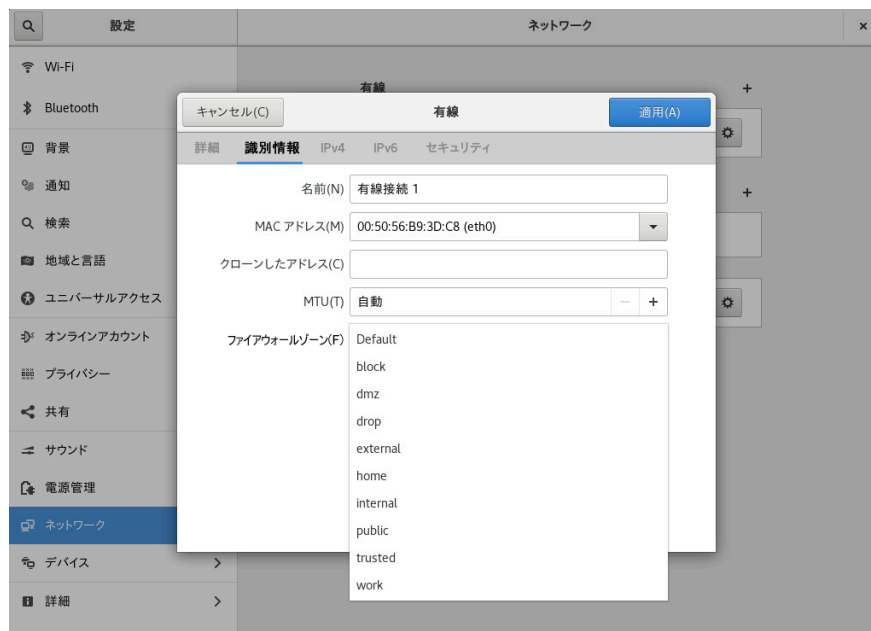


図 31.2: NETWORKMANAGERのfirewalldゾーン

ファイアウォールゾーンは、許可されるネットワーク接続に関する一般的なルールを設定します。有線接続の「firewalld」のゾーンを設定するには、接続設定の「個人情報」タブに移動します。Wi-Fi接続の「firewalld」のゾーンを設定するには、接続設定の「セキュリティ」タブに移動します。

ホームネットワークの場合は、ゾーンhomeを使用します。公衆無線ネットワークの場合は、publicに切り替えます。セキュアな環境で、すべての接続を許可したい場合は、ゾーンtrustedを使用します。

firewalldの詳細については、『Security and Hardening Guide』、第23章「Masquerading and firewalls」、23.4項「firewalld」を参照してください。

31.5 ホットスポットに関する一般的な質問とその回答 (FAQ)

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。

5. 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。

6. 同じESSIDを持つ複数のアクセスポイントが検出された場合、どのようにして特定のアクセスポイントを指定しますか？

異なる無線帯域(a/b/g/n)を持つ複数のアクセスポイントが利用可能な場合、デフォルトでは、最も強い信号を持つアクセスポイントが自動的に選択されます。このデフォルトを無効にするには、ワイヤレス接続の設定時にBSSIDフィールドを使用します。

BSSID (Basic Service Set Identifier)は、各Basic Service Setを固有に識別します。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。

31.3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。

7. どのようにして、ネットワーク接続を他のコンピュータと共有しますか？

プライマリデバイス(インターネットに接続するデバイス)には、特別な設定は必要ありません。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. 31.3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 Settings (IPv4設定)タブに切り替えて、Method (方法)ドロップダウンボックスからShared to other computers (他のコンピュータと共有)を有効にします。これで、IPトラフィックの転送が有効になり、デバイス上でDHCPサーバが実行されます。NetworkManagerで変更内容を確認します。
2. DHCPサーバは、ポート67を使用するので、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。DHCP Serverが許可されるサービスとして表示されていない場合は、Services to AllowからDHCP Serverを選択し、追加をクリックします。YaSTで変更内容を確認します。

8. 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。31.3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)アドレスのみを有効にします。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。

9. 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、31.4.1項「ユーザおよびシステムの接続」を参照してください。

31.6 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、[31.2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テクノロジーの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、[31.3.5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 wickeds` コマンドを実行します。

31.7 詳細情報

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

<https://gitlab.freedesktop.org/NetworkManager/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/](#)ボタンをクリックします。

IV ハードウェア設定

- 32 システムのキーボードレイアウト設定 **461**
- 33 サウンドカードの設定 **462**
- 34 プリンタの設定 **465**
- 35 スキャナの設定 **471**
- 36 電源管理 **473**
- 37 永続的なメモリ **480**

32 システムのキーボードレイアウト設定

YaSTのSystem Keyboard Layout (システムのキーボードレイアウト)モジュールでは、システムで使用するデフォルトのキーボードレイアウトを定義できます(このレイアウトはコンソールでも使用します)。ユーザのレベルでは、それぞれのXセッションで、デスクトップの各種ツールを使用してキーボードレイアウトを変更できます。

1. YaSTでハードウェア > System Keyboard Layout (システムのキーボードレイアウト)の順にクリックして、YaSTのSystem Keyboard Configuration (システムのキーボード設定)ダイアログを開きます。コマンドラインで`sudo yast2 keyboard`を実行して、このモジュールを起動することもできます。
2. リストから目的のキーボードレイアウトを選択します。
3. テストテキストボックスで、選択したキーボードレイアウトを試します。
4. 想定どおりの結果が得られれば、変更を確定してダイアログを閉じます。
5. 結果はファイル`/etc/vconsole.conf`(テキストコンソール用)および`/etc/X11/xorg.conf.d/00-keyboard.conf`(X11用)に保存されます。
6. システム > Sysconfigエディタ > ハードウェア > キーボードから、高度なキーボード設定を行えます。ここでは、キーボードのリピート速度と遅延設定の指定や、NumLock、CapsLock、ScrollLockの有効化/無効化を行えます。これらの設定は`/etc/sysconfig/keyboard`に保存されます。

33 サウンドカードの設定

YaSTでは、ほとんどのサウンドカードが自動的に検出され、適切な値で設定されます。デフォルト設定を変更する場合や自動設定できなかったサウンドカードを設定する場合は、YaSTのサウンドモジュールを使用します。このモジュールでは、追加のサウンドカードを設定したり、サウンドカードの順序を切り替えることもできます。

サウンドモジュールを起動するには、YaSTを起動し、ハードウェア>サウンドの順にクリックします。rootユーザとしてコマンドラインで**yast2 sound &**を実行して、サウンド設定ダイアログを直接起動することもできます。サウンドモジュールが使用できない場合は、**sudo zypper install yast2-sound**コマンドを使用してインストールします。

手順 33.1: サウンドカードの設定

新しいサウンドカードを追加した場合または既存のサウンドカードをYaSTで自動設定できなかった場合は、以下の手順を実行します。新しいサウンドカードを設定するには、サウンドカードのベンダとモデルを知っている必要があります。不確かな場合は、サウンドカードのマニュアルを参照して、必要な情報を取得してください。ALSAでサポートされているサウンドカードとその対応サウンドモジュールの参照リストについては、<http://www.alsa-project.org/main/index.php/Matrix:Main>を参照してください。設定時には、次のセットアップオプションから選択できます。

簡易自動設定

設定手順をこれ以上実行する必要はありません。サウンドカードは自動的に設定されます。後で変更できるボリュームなどのオプションを設定できます。

標準の設定

設定中に、出力音量を調整し、テストサウンドを再生できます。

詳細設定

熟練者専用。サウンドカードのすべてのパラメータをカスタマイズできます。

! 重要: 詳細設定

やるべきことを正確に知っている場合のみ、このオプションを使用してください。そうでない場合は、ここのパラメータをいじらず、標準設定か簡易設定のオプションを使用します。

1. YaSTのサウンドモジュールを起動します。
2. 検出済みだが、まだ設定されていませんが選択されているサウンドカードを設定するには、リストから各エントリを選択して編集をクリックします。

新しいサウンドカードを設定するには、追加をクリックします。サウンドカードのベンダとモデルを選択し、次へをクリックします。

3. 設定オプションの1つを選択し、次へをクリックします。
4. 標準の設定を選択した場合は、テストでサウンド設定をテストし、音量を調整できます。耳やスピーカーを傷めないよう、ボリュームの調整は10%程度から開始します。
5. すべてのオプションが希望どおりに設定されたら、次へをクリックします。
サウンド設定ダイアログに、新しく設定されたか、または変更されたサウンドカードが表示されます。
6. 必要なくなったサウンドカードの設定を削除するには、そのエントリを選択して、削除をクリックします。
7. OKをクリックして、変更内容を保存し、YaSTのサウンドモジュールを終了します。

手順 33.2: サウンドカードの設定を変更する

1. 個々のサウンドカードの設定を変更する場合は(熟練者専用)、サウンド設定ダイアログでサウンドカードのエントリを選択し、編集をクリックします。
これによって、サウンドカードの高度なオプション画面が開き、そこで、さまざまなパラメータを微調整できます。詳細については、ヘルプをクリックします。
2. 設定済みのサウンドカードの音量を調節したり、サウンドカードをテストするには、サウンド設定ダイアログでサウンドカードのエントリを選択し、その他をクリックします。各メニュー項目を選択します。



注記: YaSTのミキサー

YaSTのミキサーの設定では、基本オプションのみを提供します。それらのオプションは、トラブルシューティング用です(たとえば、テストサウンドが聞こえるかどうかなど)。その他、音量の順にクリックしてYaSTのミキサーの設定にアクセスします。サウンドオプションの日常的な使用と微調整には、デスクトップからミキサーアプレットを使用するか、または **alsasound** コマンドラインツールを使用します。

3. MIDIファイルを再生する場合は、その他、シーケンサの実行の順に選択します。
4. サポートされているサウンドカードが検出された場合は、MIDIファイルの再生用にサウンドフォントをインストールできます。
 - a. 元のドライバCD-ROMをCDまたはDVDのドライブに挿入します。

- b.** その他 › サウンドフォントのインストールの順に選択して、SF2 SoundFonts™ をハードディスクにコピーします。このサウンドフォントは、/usr/share/sfbank/creative/ディレクトリに保存されます。
- 5.** システムに複数のサウンドカードを設定した場合は、サウンドカードの順序を調節できます。サウンドカードをプライマリデバイスとして設定するには、サウンド設定でサウンドカードを選択し、その他 › プライマリカードとして設定するの順にクリックします。インデックス0のサウンドデバイスがデフォルトデバイスになり、システムとアプリケーションによって使用されます。
- 6.** SUSE Linux Enterprise Desktopでは、デフォルトでPulseAudioサウンドシステムが使用されます。このシステムは、複数のオーディオストリームのミックスを容易にする抽象レイヤであり、ハードウェアのどのような制限もバイパスします。PulseAudioサウンドシステムを有効または無効にするには、その他 › PulseAudio設定の順にクリックします。有効にした場合は、PulseAudioデーモンがサウンドの再生に使用されます。システム全体で別の設定を使用するには、PulseAudio Support (PulseAudioサポート)を無効にします。

OKをクリックしてYaSTのサウンドモジュールを終了すると、すべてのサウンドカードの音量と設定が保存されます。ミキサーの設定は、/etc/asound.stateファイルに保存されます。ALSA設定データは、/etc/modprobe.d/soundファイルの末尾に付加され、/etc/sysconfig/soundに書き込まれます。

34 プリンタの設定

YaSTを使用して、ローカルプリンタおよびネットワークプリンタを設定できます。印刷に関する詳細(一般情報、技術詳細、トラブルシューティング)は、[第24章「プリンタの運用」](#)に記載されています。

YaSTでハードウェア>プリンタの順に選択してプリンタモジュールを起動します。デフォルトでは、モジュールがプリンタ設定画面で開き、使用可能な設定済みのプリンタをすべて一覧します。これは、ネットワークを介して多数のプリンタにアクセスできる環境で、特に役に立ちます。ここから、テストページの印刷を実行し、プリンタを設定することもできます。



注記: CUPSの起動

マシンに接続されているプリンタを使用するには、CUPSがインストール済みでシステムで実行中である必要があります。CUPSが実行されていない場合、起動するように求められます。ブート時にCUPSが起動していない場合、これを有効にすることも求められます(推奨)。

34.1 プリンタの環境設定

通常、USBプリンタは自動検出されます。自動検出されない場合、プリンタの電源がオンになっていて、マシンに接続されているかを確認します。

プリンタの設定プロセスは3つのステップで構成されます。つまり、接続タイプを指定し、ドライバを選択し、このセットアップ用の印刷キューに名前を付けます。

多くのプリンタモデルは複数のドライバを利用できます。プリンタを設定するとき、YaSTはデフォルトでこれらにrecommendedのマークを付けます。通常は、ドライバを変更する必要はありません。ただし、カラープリンタでモノクロ印刷だけをしたい場合は、カラー印刷をサポートしないドライバを使用できます。画像印刷時にPostScriptプリンタでパフォーマンス上の問題が発生する場合は、PostScriptからPCLドライバに変更してみてください(ただし、使用するプリンタがPCLを理解できる場合)。

プリンタ用ドライバがリストされていない場合は、該当する標準言語を使用する汎用ドライバをリストから選択してみてください。プリンタのマニュアルを参照して、プリンタがサポートしている言語(プリンタを制御するコマンドのセット)を見つけてください。これでうまくいかない場合は、[34.1.1項「YaSTによるドライバの追加」](#)を参照して他の解決方法を試してください。

プリンタは、常に印刷キューを介して使用されます。これにより、同時実行ジョブをキューに入れ、次々に処理することができます。各印刷キューは、特定のドライバに割り当てられ、プリンタは、複数のキューを持つことができます。例として、これにより、カラープリンタ上に、モノクロでだけ印刷する2つ目のキューを設定することができます。印刷キューの詳細については、[24.1項「CUPSのワークフロー」](#)を参照してください。

手順 34.1: 新しいプリンタを追加する

1. ハードウェア > プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. プリンタ設定画面で追加をクリックします。
3. プリンタがすでにSpecify the Connectionの下に一覧されている場合は、次のステップに進みます。そうでない場合は、さらに検出を試すか、または接続ウィザードを起動します。
4. Find and Assign a Driverにあるテキストボックスにベンダ名とモデル名を入力し、検索をクリックします。
5. プリンタに適したドライバを選択します。最初に表示されるドライバを選択することをお勧めします。適切なドライバが表示されない場合、次を試してみます。
 - a. 検索条件をチェックします。
 - b. 検出を再度実行する (&F)をクリックして、検索を拡張します。
 - c. [34.1.1項「YaSTによるドライバの追加」](#)の説明のとおり、ドライバを追加します。
6. Default paper sizeを指定します。
7. 任意の名前の設定フィールドに、印刷キューの固有の名前を入力します。
8. これで、プリンタはデフォルト値で設定され、使用可能な状態になりました。OKをクリックして、プリンタ設定画面に戻ります。新しく設定されたプリンタがプリンター一覧に表示されます。

34.1.1 YaSTによるドライバの追加

SUSE Linux Enterprise Desktopで使用可能なすべてのプリンタドライバがデフォルトでインストールされるわけではありません。プリンタを追加するとき、適切なドライバがドライバの検出と割り当てダイアログにない場合、そのプリンタのドライバを含むドライバパッケージをインストールします。

手順 34.2: 追加のドライバパッケージをインストールする

1. ハードウェア > プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. プリンタ設定画面で追加をクリックします。
3. Find and Assign a Driverセクションでドライバパッケージをクリックします。
4. リストから1つまたは複数の適切なドライバパッケージを選択します。プリンタ記述ファイルへのパスは指定「しないでください」。
5. OKを選択し、パッケージのインストールを確認します。
6. これらのドライバを直接使用するには、[手順34.1「新しいプリンタを追加する」](#)の説明に従って続行します。

PostScriptプリンタでは、プリンタドライバソフトウェアが不要です。PostScriptプリンタは、特定のモデルに適したPPD (PostScript Printer Description) ファイルのみを必要としています。PPDファイルは、プリンタの製造元から提供されます。

PostScriptプリンタを追加するときに、ドライバの検出と割り当てダイアログに適切なPPDファイルが表示されない場合は、ご使用のプリンタに適したPPDファイルをインストールしてください。

PPDファイルには複数のソースがあります。まず、SUSE Linux Enterprise Desktopに付属している追加のドライバパッケージのうち、デフォルトでインストールされていないものを試してみることをお勧めします(インストール手順については後で説明します)。これらのパッケージに、ご使用のプリンタに適したドライバが含まれていない場合、PPDファイルは、プリンタベンダから直接か、またはPostScriptプリンタのドライバCDから入手します。詳細については、[24.8.2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。PPDファイルは、<http://www.linuxfoundation.org/collaborate/workgroups/openprinting/database/databaseintro> の「OpenPrinting.org printer database」で見つけることもできます。OpenPrintingからPPDファイルを使用する場合、これらは、SUSE Linux Enterprise Desktopでサポートされていない場合があることに注意してください。

手順 34.3: POSTSCRIPTプリンタのPPDファイルを追加する

1. ハードウェア > プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. プリンタ設定画面で追加をクリックします。
3. Find and Assign a Driverセクションでドライバパッケージをクリックします。
4. PPDファイルへの完全パスをMake a Printer Description File Availableの下テキストボックスに入力します。
5. OKをクリックして、Add New Printer Configuration画面に戻ります。

6. このPPDファイルを直接使用するには、[手順34.1「新しいプリンタを追加する」](#)の説明に従って続行します。

34.1.2 ローカルプリンタ設定の編集

プリンタの既存の設定を編集することで、接続タイプやドライバなどの基本設定を変更できるだけでなく、用紙サイズ、解像度、メディアソースなどのデフォルト設定を調整することもできます。プリンタの説明や場所を変更することで、プリンタの識別子を変更できます。

1. ハードウェア > プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. プリンタ設定画面で、リストからローカルプリンタ設定を選択し、編集をクリックします。
3. 接続タイプまたはドライバを変更します([手順34.1「新しいプリンタを追加する」](#)参照)。この手順は、現在の設定で問題がある場合にのみ必要です。
4. 必要に応じて既定のプリンタを選択して、このプリンタをデフォルトにします。
5. 現在のドライバに対する全てのオプションをクリックして、デフォルト設定を調整します。設定を変更するには、+記号をクリックしてオプションのリストを展開します。オプションをクリックして、デフォルトを変更します。変更内容を反映するには、OKをクリックします。

34.2 YaSTによるネットワーク印刷の設定

ネットワークプリンタは、自動的に検出されません。ネットワークプリンタは、YaSTのプリンタモジュールを使用して手動で設定する必要があります。ネットワークの設定内容に応じて、印刷サーバ(CUPS、LPD、SMB、またはIPX)に印刷したり、ネットワークプリンタに直接印刷(TCP経由を推奨)することができます。YaSTのプリンタモジュールの左ペインでネットワーク経由で印刷するを選択してネットワーク印刷の設定画面にアクセスします。

34.2.1 CUPSの使用

Linux環境では、ネットワークを介した印刷に、CUPSが使用されます。最も簡単なセットアップは、すべてのクライアントが直接アクセスできる1つのCUPSサーバだけを使用する印刷です。複数のCUPSサーバによる印刷では、リモートCUPSサーバと通信するCUPSデーモンが稼動中であることが必要です。

！ 重要: ネットワーク印刷キューの参照

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/DNS-SDが使用されます。ユーザが自分の印刷ジョブをどのプリンタに送信するかを選択できるようにクライアントでこれらのリストを参照する必要があります。ネットワーク印刷キューを参照するには、`cups-filters-cups-browsed`パッケージに付属する`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、YaSTでネットワーク印刷を設定すると自動的に起動します。

`cups-browsed`を起動してもブラウズできない場合は、CUPSサーバがBonjour/DNS-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`**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。

手順 34.4: 1つのCUPSサーバを介して印刷する

1. ハードウェア > プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. 左側のペインで、ネットワーク経由で印刷するを選択します。
3. すべての印刷を1つのCUPSサーバ経由で直接行うをオンにして、サーバの名前またはIPアドレスを指定します。
4. テストサーバをクリックして、正しい名前またはIPアドレスが選択されていることを確認します。
5. OKをクリックして、プリンタ設定画面に戻ります。これで、CUPSサーバを介して利用できるすべてのプリンタが一覧されます。

手順 34.5: 複数のCUPSサーバを介して印刷する

1. ハードウェア > プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. 左側のペインで、ネットワーク経由で印刷するを選択します。
3. CUPSサーバからのプリンタアナウンスメントを受け入れるを選択します。
4. `General Settings`の下で、使用するサーバを指定します。すべての使用可能なネットワーク、または特定のホストからの接続を受け入れることができます。最後のオプションを選択する場合は、ホスト名またはIPアドレスを指定する必要があります。

5. ローカルCUPSサーバの起動を要求されたら、OKをクリックし、次にはいをクリックします。サーバが起動すると、YaSTの画面はプリンタ設定画面に戻ります。一覧を更新をクリックして、これまでに検出されたプリンタを参照します。

34.2.2 CUPS以外の印刷サーバの使用

CUPS以外の印刷サーバから印刷サービスを提供しているネットワークでは、ハードウェア>プリンタの順に選択してYaSTのプリンタモジュールを起動し、左ペインからネットワーク経由で印刷するを選択します。接続ウィザードを起動し、該当する接続タイプを選択します。ご利用の環境でのネットワークプリンタの設定については、ネットワーク管理者にお問い合わせください。

34.3 ネットワーク上のプリンタ共有

ローカルCUPSデーモンが管理するプリンタは、ネットワーク上で共有できます。その場合、実質的にローカルコンピュータがCUPSサーバになります。プリンタを共有するには、CUPSのいわゆる「ブラウジングモード」を有効にします。ブラウジングを有効にした場合は、リモートCUPSデーモンをリッスンするために、ローカル印刷キューがネットワーク上で利用可能になります。また、専用のCUPSサーバをセットアップして、そのサーバですべての印刷キューを管理し、リモートクライアントをそのサーバに直接アクセスさせることも可能です。この場合、ブラウジングを有効にする必要はありません。

手順 34.6: プリンタを共有する

1. ハードウェア>プリンタの順に選択して、YaSTのプリンタモジュールを起動します。
2. 左側のペインからプリンタの共有を選択します。
3. リモートアクセスを許可するを選択します。ローカルネットワーク内のコンピュータ向けを選択します。さらに、既定でローカルネットワーク内にプリンタを公開も選択して参照モードを有効にします。
4. OKをクリックしてCUPSサーバを再起動し、プリンタ設定画面に戻ります。
5. CUPSとファイアウォールの設定については、https://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings を参照してください。

35 スキャナの設定

YaSTを使用して、スキャナまたはSCSIスキャナを設定できます。[sane-backends](#)パッケージには、ハードウェアドライバなどの、スキャナの使用に必要な必須コンポーネントが含まれています。HPのオールインワンデバイスを所有している場合は、[35.1項「HPのオールインワンデバイスの設定」](#)を参照してください。ネットワークスキャナの設定方法については、[35.3項「ネットワーク経由のスキャン」](#)を参照してください。

手順 35.1: USBまたはSCSIのスキャナを設定する

1. USBまたはSCSIスキャナをコンピュータに接続し、オンにします。
2. YaSTを起動し、ハードウェア > スキャナの順に選択します。YaSTが、自動的に、スキャナのデータベースをビルドし、スキャナモデルの検出を試行します。
USBまたはSCSIのスキャナが正しく検出されない場合は、まず、その他 > 検出の再開の順に選択して、再試行してください。
3. スキャナをアクティブにするには、検出されたスキャナのリストからそれを選択して、編集をクリックします。
4. リストからご使用のモデルを選択して、次へと完了をクリックします。
5. その他 > テストの順に選択して、正しいドライバを選択しているかどうか確認します。
6. OKをクリックして、設定画面を閉じます。

35.1 HPのオールインワンデバイスの設定

HPのオールインワンデバイスは、ネットワーク経由で使用される場合でも、YaSTで設定できます。USB用HPオールインワンデバイスを所有している場合は、[手順35.1「USBまたはSCSIのスキャナを設定する」](#)の説明に従って設定を開始します。デバイスが正しく検出され、テストが成功した場合は、デバイスを使用できます。

USBデバイスが正しく検出されないか、HPのオールインワンデバイスがネットワークに接続している場合は、HPデバイスマネージャを実行します。

1. YaSTを起動し、ハードウェア > スキャナの順に選択します。YaSTによってスキャナデータベースがロードされます。
2. その他 > hp-setupの起動の順に選択し、画面上の指示に従ってHPデバイスマネージャを起動します。HPデバイスマネージャを完了すると、YaSTスキャナモジュールによって自動検出が再開されます。

3. その他 > テスト の順に選択して、デバイスをテストします。
4. OKをクリックして、設定画面を閉じます。

35.2 ネットワーク経由のスキヤナの共有

SUSE Linux Enterprise Desktopを使用すると、ネットワークでスキヤナを共有できます。そのためには、次のようにスキヤナを設定します。

1. スキヤナを設定します(第35章「スキヤナの設定」参照)。
2. その他 > ネットワーク経由のスキヤンの順に選択します。
3. サーバの設定 > sanedで許可するクライアントの順に選択して、スキヤナの使用を許可するクライアントのホスト名(カンマで区切る)を入力し、OKをクリックして設定のダイアログを終了します。

35.3 ネットワーク経由のスキヤン

ネットワーク上で共有されるスキヤナを使用するには、次の手順に従います。

1. YaSTを起動し、ハードウェア > スキヤナの順に選択します。
2. その他 > ネットワーク経由のスキヤンの順に選択して、ネットワークスキヤナの設定メニューを開きます。
3. クライアントの設定 > netメタドライバで使用するサーバの順に選択して、スキヤナの接続先にするコンピュータのホスト名を入力します。
4. OKをクリックして、終了します。これで、ネットワークスキヤナが [スキヤナの設定] ウィンドウに表示され、使用できる状態になりました。

36 電源管理

電源管理はラップトップコンピュータで特に重要ですが、他のシステムでも役に立ちます。ACPI(Advanced Configuration and Power Interface)は、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ーでは、適切なハードウェアとBIOSルーチンを必要とします。ほとんどのラップトップと多くの新型デスクトップおよびサーバは、これらの必要条件を満たしています。電源の節約や騒音の低減のために、CPU周波数を制御することもできます。

36.1 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

サポートされていません。

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。メーカーの中には、このモードを便利なハイブリッド仕様にして提供するものもあります(たとえば、IBM ThinkpadのRediSafe)。対応するACPI状態は、S4です。Linux環境では、suspend to diskはACPIから独立したカーネルルーチンにより実行されます。



注記: **mkswap**でフォーマットするとスワップパーティションのUUIDが変更される

可能であれば、**mkswap**で既存のスワップパーティションを再フォーマットしないでください。**mkswap**で再フォーマットすると、スワップパーティションのUUIDの値が変更されます。YaSTで再フォーマットするか(/etc/fstabが更新されます)、/etc/fstabを手動で調整します。

バッテリーモニタ

ACPIは、バッテリーをチェックして、充電ステータスに関する情報を提供します。また、システムは、重要な充電ステータスに達した時点で実行するようにアクションを調整します。

自動電源オフ

シャットダウンの後、コンピュータの電源が切れます。これは、バッテリーが空になる直前に自動シャットダウンが行われる場合に特に重要です。

プロセッサ速度の制御

CPU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。

36.2 ACPI(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はバッテリー、ACアダプタ、温度、ファン、および「close lid」や「battery low」などのシステムイベントに関する情報も提供します。

BIOSには個々のコンポーネントとハードウェアアクセス方法についての情報が入ったテーブルがあります。オペレーティングシステムは、この情報を使用して、割り込みまたはコンポーネントの有効化と無効化などのタスクを実行します。BIOSに格納されているコマンドを、オペレーティングシステムが実行するとき、機能はBIOSの実装方法に依存します。ACPIが検出可能で、ロードできるテーブルは、`journalctl`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21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、36.2.2項「**トラブルシューティング**」を参照してください。

36.2.1 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。カーネルのオンデマンドガバナによって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ー使用時間を延長する場合など、最後の手段として使用する必要があります。ただし、スロットリングの割合が高すぎると、スムーズに動作しない特定の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、『System Analysis and Tuning Guide』、第12章「Power management」を参照してください。

36.2.2 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に、BIOSにおけるACPIの指定を故意に変えていることがあります。ACPIを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外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36.4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、常にメーカーに連絡することをお勧めします。特に、Linuxに関するサポートを常に提供していないメーカーには、問題を通知する必要があります。なぜなら、メーカーは、自社の顧客の無視できない数がLinuxを使用しているとわかってやっと、問題を真剣に受け止めるからです。

36.2.2.1 詳細情報

- <https://tldp.org/HOWTO/ACPI-HOWTO/>  (詳細なACPI HOWTO、DSDTパッチが含まれています)
- <https://uefi.org/specifications>  (Advanced Configuration and Power Interface: 詳細設定と電源インタフェース)

36.3 ハードディスクの休止

Linux環境では、不要な場合にハードディスクを完全にスリープ状態にしたり、より経済的な静止モードで動作させることができます。最近のラップトップの場合、ハードディスクを手動でオフに切り替える必要はありません。不要な場合は自動的に経済的な動作モードになります。ただし、最大限に省電力したい場合は、次の方法を**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、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: データの完全性リスク

`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ドウンしなくなります。モバイル機器では、これを避けるために特別なカーネル拡張が開発されています。その拡張を使用するには、`laptop-mode-tools`パッケージをインストールし、詳細について、`/usr/src/linux/Documentation/laptops/laptop-mode.txt`を参照してください。

もう1つの重要な要因は、アクティブプログラムが動作する方法です。たとえば、優れたエディタは、変更中のファイルを定期的にハードディスクに自動バックアップし、これによってディスクがウェイクアップされます。データの完全性を犠牲にすれば、このような機能を無効にできます。

この接続では、メールデーモン`postfix`が変数 `POSTFIX_LAPTOP` を使用します。この変数を`yes`に設定すると、`postfix`がハードディスクにアクセスする頻度は大幅に減少します。

SUSE Linux Enterprise Desktopでは、これらの技術は`laptop-mode-tools`により制御されます。

36.4 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21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。

36.4.1 CPU周波数調節が機能しない

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。

37 永続的なメモリ

この章では、1つ以上のNVDIMMで構成される「永続的なメモリ」とも呼ばれる不揮発性メインメモリとSUSE Linux Enterprise の使用に関する追加情報を記載します。

37.1 概要

永続的なメモリとは、新しいタイプのコンピュータストレージで、動的RAM (DRAM)に近い速度を発揮し、RAMのバイト単位のアドレス指定、およびソリッドステートドライブ(SSD)のパフォーマンスを併せ持ちます。

SUSEは現在、AMD64/Intel 64およびPOWERアーキテクチャを搭載したマシン上のSUSE Linux Enterprise Serverでの永続的なメモリの使用をサポートしています。

従来のRAMと同様に、永続的なメモリはメインボードのメモリスロットに直接設置されます。そのため、RAM、DIMMと同じ物理フォームファクタで提供されます。これらは、NVDIMM、不揮発性デュアルインラインメモリモジュールとして知られています。

ただし、永続的なメモリはいくつかの点においてRAMとは異なり、フラッシュベースのSSDに類似しています。これらは両方ともソリッドステートメモリ回路の形態に基づいていますが、それにもかかわらず、両方とも非揮発性ストレージを提供し、システムの電源がオフにされたり、再起動されてもそのコンテンツは保持されます。両方の形態のメディアについて、データの書き込みは読み取りよりも低速で、両方とも限定された回数のリライトサイクルをサポートしています。また、SSDと同様に、特定の用途でより適している場合には、永続的なメモリへのセクタレベルのアクセスが可能です。

モデルごとに、Intel 3D XPointや、NANDフラッシュとDRAMを組み合わせるなど、さまざまな形態の電子ストレージメディアを使用します。新たな形態の不揮発性RAMも開発中です。つまり、NVDIMMのさまざまなベンダーおよびモデルで、さまざまなパフォーマンスや耐久性特性が提供されることを意味しています。

含まれるストレージテクノロジーは開発の初期段階であるため、さまざまなベンダーのハードウェアに異なった制限が与えられる場合があります。この一般的な内容は次のとおりです。永続的なメモリはDRAMより最大10倍低速ですが、フラッシュストレージより約1000倍高速です。フラッシュメモリの全セクタの消去およびリライトプロセスではなく、バイト単位でリライト可能です。つまり、リライトサイクルは限定されていますが、ほとんどの形態の永続的なメモリが、フラッシュストレージの数千サイクルと比較すると、何百万サイクルのリライトを処理することができます。

ただし、この結果次の2つの制約を受けます。

- 現在のテクノロジーでは、永続的なメモリのみを使用してシステムを実行し、不揮発性メインメモリを得ることはできません。従来のRAMとNVDIMM両方の混在したものを使用する必要があります。オペレーティングシステムおよびアプリケーションは、高速な追加のストレージを提供するNVDIMMとともに、従来のRAMで実行されます。
- さまざまなベンダーの永続的なメモリのパフォーマンス特性は、使われているNVDIMM数、および装着に適したメモリスロットなど、特定のサーバのNVDIMMのハードウェア仕様をプログラマが認識している必要があることを示しています。これは、ハイパーバイザーの使用、異なるホストマシン間のソフトウェアのマイグレーションなどに影響します。


この新しいストレージサブシステムはACPI標準のバージョン6で定義されています。ただし、`libnvdimm`はブレ標準のNVDIMMをサポートし、同様に使用することができます。



ヒント: Intel Optane DC永続メモリ

Intel Optane DIMMメモリは、次の特定のモードで使用できます。

- 「App Direct Mode」。Intel Optaneメモリは、高速の永続的なストレージとして使用されます。SSDデバイスおよびNVMeデバイスの代替です。このモードのデータは、システムの電源をオフにしても保持されます。
App Direct Modeは、SLE 12 SP4以降サポートされています。
- 「Memory Mode」。Intel Optaneメモリは、コスト効果の高い、大容量のDRAMの代替品として機能します。このモードでは、個々のDRAM DIMMは最も頻繁にアクセスされるデータ用のキャッシュとして動作し、Optane DIMMメモリは大容量メモリを提供します。ただし、DRAMのみのシステムと比較すると、このモードは、ランダムアクセス負荷の状況では低速です。このモードを利用するOptane固有の拡張機能を使用せずにアプリケーションを実行すると、メモリのパフォーマンスが低下する場合があります。このモードのデータは、システムの電源をオフにすると失われます。
Memory Modeは、SLE 15 SP1以降サポートされています。
- 「Mixed Mode」。Intel Optaneメモリは、パーティション化されているため、両方のモードで同時に動作できます。
Mixed Modeは、SLE 15 SP1以降サポートされています。

Intel Optane DC永続メモリの詳細については、<https://www.intel.com/content/dam/support/us/en/documents/memory-and-storage/data-center-persistent-mem/Intel-Optane-DC-Persistent-Memory-Quick-Start-Guide.pdf> を参照してください。

37.2 用語

地域

「領域」とは1つ以上の「ネームスペース」に分けることが可能な永続的なメモリのブロックです。領域の永続的なメモリにアクセスするには、まず、その領域をネームスペースに割り当てる必要があります。

ネームスペース

NVM Express SSDネームスペース、またはSCSI論理ユニット番号(LUN)と比較可能な、非揮発性ストレージの単一の連続アドレス指定範囲。ネームスペースはサーバの/devディレクトリに個別のブロックデバイスとして表示されます。要求されるアクセス方法に従って、ネームスペースは複数のNVDIMMから大きなボリュームにストレージを混合したり、より小さなボリュームにパーティショニングしたりできます。

モード

各ネームスペースには、そのネームスペースに対して有効化されるNVDIMM機能を定義する「モード」もあります。同じ親領域の兄弟ネームスペースは常に同じタイプですが、異なるモードに設定することもできます。ネームスペースのモードは次のとおりです。

devdax

Device-DAXモード。単一文字のデバイスファイルを作成します(/dev/daxX.Y)。ファイルシステムの作成は必要「ありません」。

fsdax

File system-DAXモード。他のモードが指定されない場合のデフォルトです。ext4またはXFSのDAXをサポートするブロックデバイス(/dev/pmemX [.Y])を作成します。

sector

メタデータのチェックサムを実行しないレガシーファイルシステム用。小さなブートボリュームに適しています。他のオペレーティングシステムと互換性があります。

raw

ラベルまたはメタデータのないメモリディスク。DAXをサポートしません。他のオペレーティングシステムと互換性があります。



注記

rawモードはSUSEによってサポートされていません。rawネームスペースにファイルシステムをマウントすることはできません。

タイプ

各ネームスペースおよび領域には、そのネームスペースまたは領域に関連付けられた永続的なメモリへのアクセス方法を定義する「タイプ」があります。ネームスペースは常に親領域と同じタイプを持ちます。2つの異なる方法で設定可能な永続的なメモリと、非推奨のブロックモードの2つのタイプがあります。

永続的なメモリ(PMEM)

PMEMストレージはRAMのようにバイトレベルのアクセスを提供します。PMEMを使用すると、単一ネームスペースに複数のインターリーブされたNVDIMMを含めることができ、すべてを単一デバイスとして使用できます。

PMEMネームスペースを設定するには2つの方法があります。

DAXを使用したPMEM

直接アクセス(DAX)用に設定されるPMEMネームスペースとは、メモリへのアクセスがカーネルのページキャッシュをバイパスして、メディアに直接行われることを意味します。ソフトウェアはネームスペースのすべてのバイトを直接、個別に読み書きできます。

ブロック変換テーブル(BTT)を使用したPMEM

BTTモードで動作するように設定されたPMEMネームスペースは、どちらかというRAMのようなバイトアドレス指定可能なモデルではなく、従来のディスクドライブのようにセクタベースでアクセスされます。変換テーブルメカニズムバッチはセクタサイズのユニットにアクセスします。

BTTの利点はデータ保護です。ストレージサブシステムは、各セクタが基盤となるメディアに完全に書き込まれるようにします。セクタが完全に書き込まれない場合(すなわち、何らかの理由により、書き込み操作に失敗する場合)、セクタ全体が以前の状態にロールバックされます。したがって、指定されたセクタは部分的に書き込まれることはありません。

また、BTTネームスペースへのアクセスはカーネルによってキャッシュされます。

この欠点は、BTTネームスペースにはDAXができない点です。

ブロックモード(BLK)

ブロックモードストレージは、各NVDIMMを個別のデバイスとしてアドレス指定します。この使用は非推奨で、サポートされなくなりました。

devdaxネームスペースを除き、他のすべてのタイプは従来のドライブのように、ファイルシステムでフォーマットされる必要があります。SUSE Linux Enterprise Desktopでは、これに対してext2、ext4およびXFSファイルシステムをサポートしています。

直接アクセス(DAX)

DAXでは、たとえばmmapシステムコールを使用して、永続的なメモリをプロセスのアドレススペースに直接マップすることができます。

DIMM物理アドレス(DPA)

単一DIMMメモリへのオフセットとしてのメモリアドレス。つまりそのDIMM上で最も小さいアドレス指定可能なバイトとして0から開始します。

ラベル

ネームスペース定義など、NVDIMMに保存されるメタデータ。これにはDSMを使用してアクセスできます。

デバイス固有のメソッド(DSM)

NVDIMM上のファームウェアにアクセスするためのACPIメソッド。

37.3 使用例

37.3.1 DAXを使用したPMEM

このメモリアクセスの形式はトランザクションではありません。電源異常などのシステム障害が発生する場合には、データがストレージに完全に書き込まれない場合があります。PMEMストレージはアプリケーションが部分的に書き込まれたデータの状態を処理できる場合にのみ適しています。

37.3.1.1 バイトアドレス指定可能なストレージの大容量を活用するアプリケーション

サーバがバイト単位の大容量高速ストレージを直接使用可能なアプリケーションをホストする場合、プログラマはmmapシステムコールを使用して、追加のシステムRAMを使用せずに、永続的なメモリのブロックをアプリケーションのアドレススペースに直接配置することができます。

37.3.1.2 カーネルページキャッシュの使用を避ける

ページキャッシュ用のRAMの使用を節約し、代わりにそれを使用するアプリケーションに指定するには、カーネルページキャッシュの使用を避けます。たとえば、非揮発性メモリを仮想マシン(VM)イメージの保持専用にすることができます。これらはキャッシュされませんが、ホスト上のキャッシュ使用率を削減し、ホストごとのVMを増やすことができます。

37.3.2 BTTを使用したPMEM

高速ストレージのディスクのようなプールとしてNVDIMMのセットに永続的なメモリを使用したい場合に役立ちます。たとえば、BTTを使用してファイルシステムジャーナルをPMEMに配置すると、停電やその他の突然の中断後のファイルシステムの回復の信頼性が向上します(37.5.3項「BTTを使用したPMEMネームスペースの作成」を参照)。

アプリケーションに対して、このようなデバイスは高速SSDとして認識され、他のストレージデバイスのように使用できます。たとえば、LVMは永続的なメモリの上部に階層化することができ、通常のように動作します。

BTTの利点は、セクタ書き込みの原子性が保証されるため、データ整合性に依存する高度なアプリケーションでも機能し続けるという点です。メディアエラーレポートは標準のエラーレポーティングチャンネルを介して機能します。

37.4 永続的なメモリを管理するためのツール

永続的なメモリを管理するには、`ndctl`パッケージをインストールする必要があります。これをインストールすることにより、NVDIMMを設定するためのユーザスペースライブラリのセットを提供する`libndctl`パッケージもインストールされます。

これらのツールは、3タイプのNVDIMMをサポートする、`libnvdimm`ライブラリを介して機能します。

- PMEM
- BLK
- 同時のPMEMとBLK

`ndctl`ユーティリティには、次のコマンドを使用してアクセス可能な、便利な`man`ページセットがあります。

```
> ndctl help subcommand
```


使用可能なサブコマンドのリストを表示するには、次を使用します。

```
> ndctl --list-cmds
```

使用可能なサブコマンドには次のものがあります。

バージョン

NVDIMMサポートツールの現在のバージョンを表示します。

enable-namespace

指定されたネームスペースを使用できるようにします。

disable-namespace

指定されたネームスペースが使用されないようにします。

create-namespace

指定されたストレージデバイスから新しいネームスペースを作成します。

destroy-namespace

指定されたネームスペースを削除します。

enable-region

指定された領域を使用できるようにします。

disable-region

指定された領域が使用されないようにします。

zero-labels

デバイスからメタデータを消去します。

read-labels

指定されたデバイスのメタデータを取得します。

リスト

使用可能なデバイスを表示します。

help

ツールの使用に関する情報を表示します。

37.5 永続的なメモリのセットアップ

37.5.1 使用可能なNVDIMMストレージの表示

`ndctl list` コマンドを使用して、システム内で使用可能なすべてのNVDIMMを一覧表示できます。

次の例では、システムにトリプルチャネルでインターリーブされた単一セットの3個のNVDIMMがあります。

```
# ndctl list --dimms

[
  {
    "dev": "nmem2",
    "id": "8089-00-0000-12325476"
  },
  {
    "dev": "nmem1",
    "id": "8089-00-0000-11325476"
  },
  {
    "dev": "nmem0",
    "id": "8089-00-0000-10325476"
  }
]
```

別のパラメータ、`ndctl list`を使用して、使用可能な領域を一覧表示することもできます。



注記

領域は番号順に表示されない場合があります。

3つのNVDIMMしかありませんが、4つの領域として表示されることに注意してください。

```
# ndctl list --regions

[
  {
    "dev": "region1",
    "size": 68182605824,
    "available_size": 68182605824,
    "type": "blk"
  },
  {
```

```

"dev": "region3",
"size": 202937204736,
"available_size": 202937204736,
"type": "pmem",
"iset_id": 5903239628671731251
},
{
  "dev": "region0",
  "size": 68182605824,
  "available_size": 68182605824,
  "type": "blk"
},
{
  "dev": "region2",
  "size": 68182605824,
  "available_size": 68182605824,
  "type": "blk"
}
]

```

スペースは次の2つの異なる形態で利用できます:。BLKタイプの3つの個別の64GB領域として、または3つがインターリーブされたNVDIMM上にすべてのスペースを提供するPMEMタイプの1つに結合された189GB領域を単一のボリュームとして。

`available_size`に表示される値は、`size`の値と同じであることを注意してください。これは、スペースのどれもまだ割り当てられていないということを意味します。

37.5.2 DAXを使用した単一のPMEMネームスペースとしてストレージを設定する

最初の例として、直接アクセス(DAX)を使用した単一のPMEMネームスペースに3つのNVDIMMを設定します。

最初のステップは、新しいネームスペースを作成することです。

```

# ndctl create-namespace --type=pmem --mode=fsdax --map=memory
{
  "dev": "namespace3.0",
  "mode": "memory",
  "size": 199764213760,
  "uuid": "dc8ebb84-c564-4248-9e8d-e18543c39b69",
  "blockdev": "pmem3"
}

```

これにより、DAXをサポートする、ブロックデバイス `/dev/pmem3` が作成されます。デバイス名の `3` (この場合は `region3`) は、親地域番号から継承されます。

--map=memoryオプションにより、NVDIMM上にPMEMストレージスペースの一部が置かれ、これはstruct pagesと呼ばれる内部カーネルデータ構造を割り当てるために使用できます。これにより、新しいPMEMネームスペースが0_DIRECT I/OやRDMAなどの機能で使えるようになります。

カーネルデータ構造用に一部の永続的なメモリを予約するのは、生成されるPMEMネームスペースの容量がPMEM親領域よりも小さいためです。

次に、新しいブロックデバイスがオペレーティングシステムで利用可能であることを確認します。

```
# fdisk -l /dev/pmem3
Disk /dev/pmem3: 186 GiB, 199764213760 bytes, 390164480 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

使用する前に、他のドライブのように、フォーマットする必要があります。この例では、XFSを使用してフォーマットします。

```
# mkfs.xfs /dev/pmem3
meta-data=/dev/pmem3      isize=256    agcount=4, agsize=12192640 blks
               =          sectsz=4096   attr=2, projid32bit=1
               =          crc=0        finobt=0, sparse=0
data        =            bsize=4096    blocks=48770560, imaxpct=25
               =            sunit=0     swidth=0 blks
naming      =version 2     bsize=4096    ascii-ci=0 ftype=1
log         =internal log  bsize=4096    blocks=23813, version=2
               =            sectsz=4096  sunit=1 blks, lazy-count=1
realtime    =none         extsz=4096    blocks=0, rtextents=0
```

次に、新しいドライブを特定のディレクトリにマウントできます。

```
# mount -o dax /dev/pmem3 /mnt/pmem3
```

ここで、DAX対応デバイスがあることを確認できます。

```
# mount | grep dax
/dev/pmem3 on /mnt/pmem3 type xfs (rw,relatime,attr2,dax,inode64,noquota)
```

これで、XFSファイルシステムでフォーマットされ、DAXでマウントされたPMEMネームスペースが設定されます。

mmap()は、ファイルに呼び出しを行い、ファイルシステムはNVDIMM上の永続的なメモリに直接マップする仮想アドレスを返し、ページキャッシュをバイパスします。

そのファイルシステム内のファイルに対するfsyncまたはmsync呼び出しが行われても、変更されたデータは完全にNVDIMMに書き込まれています。これらの呼び出しはmmapマッピングを介してユーザスペースで変更されているページに関連付けられているプロセッサキャッシュラインをフラッシュします。

37.5.2.1 ネームスペースの削除

同じストレージを使用する他のボリュームタイプを作成する前に、このPMEMボリュームをアンマウントしてから削除する必要があります。

まず、このボリュームをアンマウントします。

```
# umount /mnt/pmem3
```

次にネームスペースを無効にします。

```
# ndctl disable-namespace namespace3.0
disabled 1 namespace
```

そして削除します。

```
# ndctl destroy-namespace namespace3.0
destroyed 1 namespace
```

37.5.3 BTTを使用したPMEMネームスペースの作成

BTTはセクタ書き込みの原子性を提供します。これは、Ext4ジャーナルやXFSジャーナルなどのデータ保護が必要な場合に適しています。電源障害が発生した場合、ジャーナルは保護され、回復可能である必要があります。次の例は、BTTを使用したPMEMネームスペースをセクタモードで作成する方法と、このネームスペースにファイルシステムジャーナルを配置する方法を示しています。

```
# ndctl create-namespace --type=pmem --mode=sector
{
  "dev": "namespace3.0",
  "mode": "sector",
  "uuid": "51ab652d-7f20-44ea-b51d-5670454f8b9b",
  "sector_size": 4096,
  "blockdev": "pmem3s"
}
```

次に、新しいデバイスが存在することを確認します。

```
# fdisk -l /dev/pmem3s
Disk /dev/pmem3s: 188.8 GiB, 202738135040 bytes, 49496615 sectors
Units: sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

以前に設定したDAX対応PMEMネームスペースと同様に、このBTT対応PMEMネームスペースはNVDIMM上で使用可能なすべてのストレージを消費します。



注記

デバイス名の最後のs (`/dev/pmem3s`)は、`sector`を表し、BTTを使用するように設定されるネームスペースを簡単に区別するために使用されます。

ボリュームは前の例と同様に、フォーマットし、マウントできます。

ここに表示されるPMEMネームスペースはDAXを使用することはできません。その代わりに、BTTを使用して、「セクタ書き込みの原子性」を提供します。PMEMブロックドライバからのセクタ書き込みが行われるたびに、BTTは新しいセクタを割り当てて新しいデータを受け取ります。BTT原子性により、新しいデータが完全に書き込まれた後で、その内部マッピング構造をアップデートし、新しく書き込まれたデータがアプリケーションで利用できるようにします。このプロセス中に任意のポイントで電源障害が発生した場合、書き込みは消失しますが、アプリケーションはまだ存在する古いデータにアクセスできます。これにより、「tornセクタ」と呼ばれる状況が回避されます。

このBTT対応PMEMネームスペースは他の標準ブロックデバイスのようにファイルシステムでフォーマットして使用できます。DAXと併用することはできません。ただし、このブロックデバイス上のファイルの**mmap**マッピングはページキャッシュを使用します。

37.5.4 PMEM/BTTにファイルシステムジャーナルを配置する

別のデバイスにファイルシステムジャーナルを配置する場合は、ファイルシステムと同じファイルシステムブロックサイズを使用する必要があります。これは4096である可能性が高く、次のコマンドでブロックサイズを確認できます。

```
# blockdev --getbsz /dev/sda3
```

次の例では、別のNVDIMMデバイスに新しいExt4ジャーナルを作成し、SATAデバイスにファイルシステムを作成してから、新しいファイルシステムをジャーナルに接続します。

```
# mke2fs -b 4096 -O journal_dev /dev/pmem3s
# mkfs.ext4 -J device=/dev/pmem3s /dev/sda3
```





次の例では、SATAドライブに新しいXFSファイルシステムを作成し、別のNVDIMMデバイスにジャーナルを作成します。

```
# mkfs.xfs -l logdev=/dev/pmem3s /dev/sda3
```

オプションに関する詳細については、[man 8 mkfs.ext4](#)および[man 8 mkfs.xfs](#)を参照してください。

37.6 詳細情報

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。

- [Persistent Memory Wiki \(https://nvdimm.wiki.kernel.org/\)](https://nvdimm.wiki.kernel.org/) 
NVDIMMシステムを構成するための手順、テストに関する情報、およびNVDIMMの有効化に関連する仕様へのリンクが記載されています。LinuxのNVDIMMサポートは現在開発中のため、このサイトも構築中です。
- [Persistent Memory Programming \(http://pmem.io/\)](http://pmem.io/) 
Linuxおよび他のオペレーティングシステムの下で、非揮発性メモリを搭載したシステムを設定、使用、およびプログラミングする方法に関する情報。ユーザスペースの永続的なメモリをプログラミングするために役立つAPIを提供することを目的とした、NVMライブラリ(NVML)について説明しています。
- [LIBNVDIMM: Non-Volatile Devices \(https://www.kernel.org/doc/Documentation/nvdimm/nvdimm.txt\)](https://www.kernel.org/doc/Documentation/nvdimm/nvdimm.txt) 
これは、カーネル開発者を対象としており、現在のLinuxカーネルツリーのドキュメントディレクトリの一部です。NVDIMM有効化に含まれている異なるカーネルモジュール、カーネル実装のテクニカル詳細、`sysfs`ツールによって使用されるカーネルへの`ndctl`インタフェースについて説明しています。
- [GitHub: pmem/ndctl \(https://github.com/pmem/ndctl\)](https://github.com/pmem/ndctl) 
Linuxカーネルの`libnvdimm`サブシステムを管理するためのユーティリティライブラリ。ユーザスペースライブラリ、ユニットテスト、およびマニュアルも含まれます。

V サービス

38 YaSTによるサービス管理 **494**

39 NTPによる時刻の同期 **496**

38 YaSTによるサービス管理

YaSTでは、デフォルトのシステムターゲット、サービスを制御し、サービスステータスを表示し、ログファイルを読み込むためのサービスマネージャを提供します。SUSE Linux Enterprise Desktop 15 SP5の新機能は、オンデマンドでサービスが開始されるように設定する、systemdソケットベースのサービスアクティベーションのためのYaSTサポートです。

systemdはオンデマンドでサービスを開始するために、ソケットベースのアクティベーションでサービスを開始することをサポートします。これらのサービスには、2つのユニットタイプ(サービスとソケット)があります。たとえば、CUPSはcups.serviceおよびcups.socketによって制御されます。YaSTでは、使用するサービススタートアップのタイプを選択できます。

図38.1「YaSTサービスマネージャ」では、起動モードのドロップダウンボックスのオプションが表示されます。起動時、オンデマンド、手動。ソケットベースのアクティベーションにはオンデマンドを選択します。これにより、リスニングネットワークソケットが開き、要求があるとサービスが開始されます。

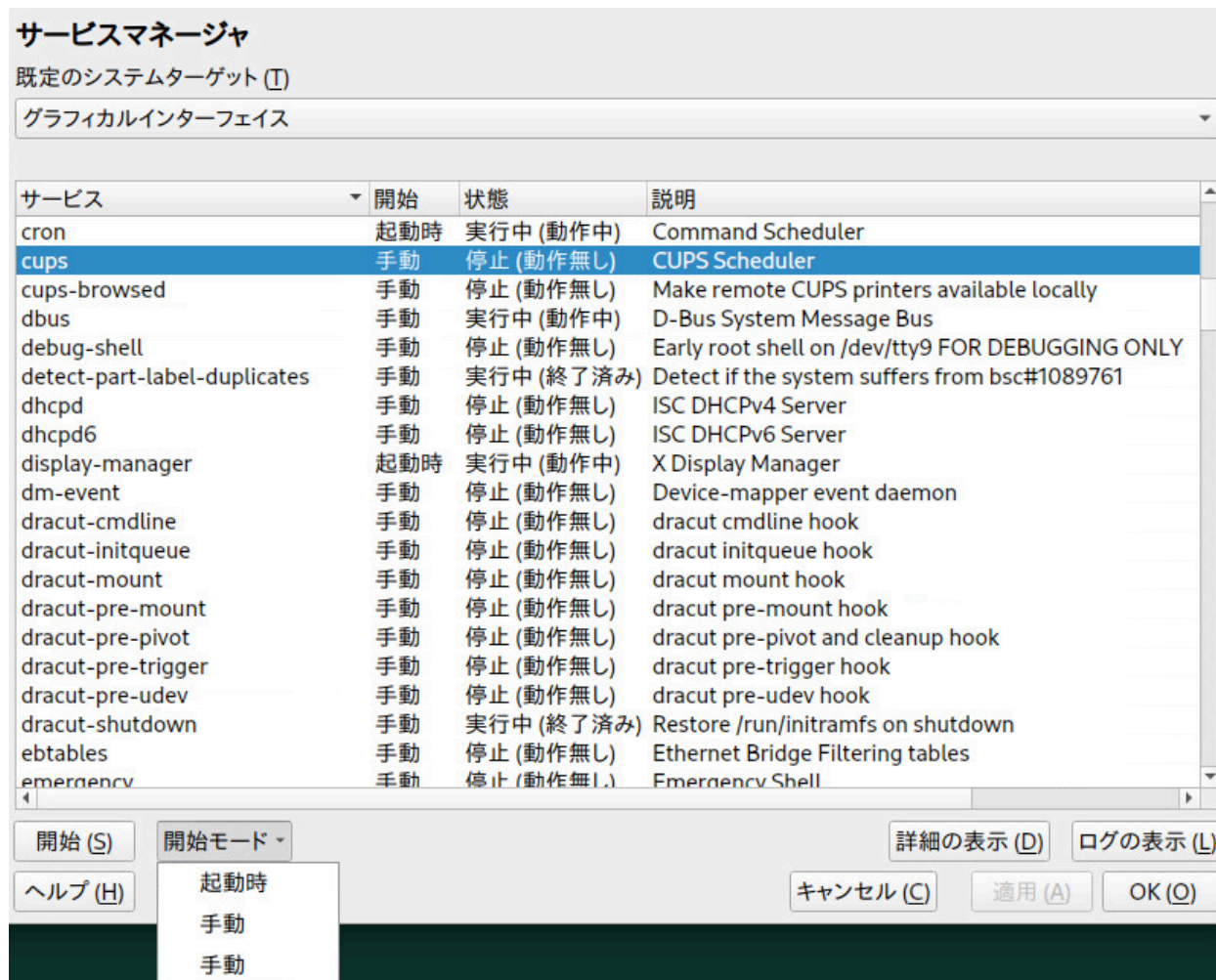


図 38.1: YASTサービスマネージャ

オンデマンドオプションは、それをサポートするサービスでのみ表示されます。現在、これは CUPS、dbus、iscsid、iscsiuio、multipathd、pcscd、rpcbind、TFTP、virtlockd、virtlogd など、サービスの小さなサブセットです。ソケットアクティベーションの動作方法に関する詳細については、[man 5 systemd.socket](#)を参照してください。

39 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を使用して、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。

SUSE Linux Enterprise Desktop 15以降、chronyは、NTPでデフォルトで実装されるようになりました。chronyは2つの部分で構成されています。chronydは、ブート時に起動可能なデーモンであり、chronycは、chronydのパフォーマンスを監視し、実行時にさまざまな動作パラメータを変更するためのコマンドラインインタフェースプログラムです。

SUSE Linux Enterprise Desktop 15.2以降、NTPクライアント設定用のYaSTモジュールは、デーモンとして実行するように設定されていない場合に、cronデーモンではなくsystemd-timerを設定し、chronyを実行します。



注記

アクティブディレクトリによる時間同期を有効にするには、『Security and Hardening Guide』、第7章「Active Directory support」、7.3.3項「Joining Active Directory using Windows domain membership」、Joining an Active Directory domain using Windows domain membershipにある指示に従います。

39.1 YaSTでのNTPクライアントの設定

`chrony`パッケージ付属のNTPデーモン(`chronyd`)は、ローカルコンピュータハードウェアクロックを時間の参照に使用するように事前設定されています。ハードウェアクロックの精度は、その時間ソースに大きく依存します。たとえば、原子時計やGPS受信機は非常に正確な時間ソースですが、一般的なRTCチップは信頼できる時間ソースではありません。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。

YaST NTPクライアント設定(ネットワークサービス > NTP設定)ウィンドウでは、NTPデーモンの開始時期や設定ソースの種類を指定したり、カスタムタイムサーバを追加することができます。



図 39.1: [NTP設定] ウィンドウ

39.1.1 NTPデーモン開始

NTPデーモンを開始する時期は、次の3つのオプションから選択できます。

手動でのみ

手動でのみを選択すると、`chrony`デーモンが手動で起動します。

デーモンを使用せずに同期する

デーモンを使用せずに同期するを選択すると、永続的に動作する`chrony`を使用せずに、定期的にシステム時間を設定します。同期間隔(分)を設定できます。

今すぐおよびブート時

システムのブート時に自動的にchronyd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。

39.1.2 設定元のタイプ

設定元ドロップダウンボックスで、動的または静的のいずれかを選択します。お使いのサーバが(パブリック) NTPサーバの固定セットのみを使用している場合は、静的を設定し、内部ネットワークがDHCP経由でNTPサーバを提供している場合は、動的が適しています。

39.1.3 タイムサーバの設定

クライアントが問い合わせるタイムサーバは、NTP設定ウィンドウの下部に表示されます。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。

追加をクリックして、新しいタイムサーバを追加します。

図 39.2: タイムサーバの追加

1. アドレスフィールドに、マシン時刻を同期させるタイムサーバまたはタイムサーバのプールのURLを入力します。URLを入力したら、テストをクリックして、有効な時間ソースを指していることを確認します。

2. chronydデーモンの開始時により多くの要求を送信することによって時刻同期を高速化するには、初期同期の高速化を有効にします。
3. ブート時にchronydデーモンを自動的に開始しインターネットに接続されていない可能性のあるシステムでブート時間を短縮するには、オフライン起動を有効にします。このオプションは、ネットワーク接続がNetworkManagerによって管理されるラップトップの場合などに役立ちます。
4. OKをクリックして、確定します。

39.2 ネットワークでのNTPの手動設定

chronyは、その設定を/etc/chrony.confファイルから読み込みます。コンピュータのクロックを同期させるには、使用するタイムサーバをchronyに指示する必要があります。特定のサーバ名またはIPアドレスを使用できます。以下に例を示します。

```
0.suse.pool.ntp.org
1.suse.pool.ntp.org
2.suse.pool.ntp.org
3.suse.pool.ntp.org
```

「プール」名を指定することもできます。プール名は複数のIPアドレスに解決されます。

```
pool pool.ntp.org
```



ヒント: 同じネットワーク上のコンピュータ

同じネットワーク上の複数のコンピュータで時刻を同期させる場合、それらのコンピュータをすべて外部サーバと同期させることはお勧めしません。1つのコンピュータを、外部のタイムサーバと同期させるタイムサーバとし、他のコンピュータを、クライアントとして機能させることをお勧めします。信頼性のあるタイムサーバと区別するには、サーバの/etc/chrony.confにlocalディレクティブを追加します。

```
local stratum 10
```

chronyを起動するには、次のコマンドを実行します。

```
systemctl start chronyd.service
```


`chronyd`を初期化した後、時間が安定するまでにある程度時間がかかり、ローカルコンピュータクロックを修正するためのドリフトファイルが作成されます。ドリフトファイルを用いることで、ハ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。

ブート時に`chrony`が自動的に起動するようにサービスを有効にするには、次のコマンドを実行します。

```
systemctl enable chronyd.service
```



警告: `yast-timesync.service`サービスの競合

`chronyd.service`サービスに加えて、SLEDには`yast-timesync.service`が含まれています。`yast-timesync.service`は、タイマによって5分おきにトリガされ、`-q`オプションを指定して`chronyd`を実行し、システム時刻を設定して終了します。任意の時点で実行できる`chronyd`のインスタンスは1つのみであるため、`chronyd`関連のサービスを両方同時に有効化または起動しないでください。

39.3 `chronyc`を使用した実行時の`chronyd`の設定

実行時に`chronyc`を使用して`chronyd`の動作を変更することができます。`chronyd`の操作に関するステータスレポートも生成します。

`chronyc`は、対話的または非対話的モードで実行できます。`chronyc`を対話形式で実行するには、コマンドラインで`chronyc`を入力します。プロンプトを表示し、コマンド入力を待ちます。たとえば、オンラインまたはオフラインのNTPソースの数を確認するには、次のコマンドを実行します。

```
# chronyc
chronyc> activity
200 OK
4 sources online
2 sources offline
1 sources doing burst (return to online)
1 sources doing burst (return to offline)
0 sources with unknown address
```

`chronyc`のプロンプトを終了するには、`quit`または`exit`を入力します。

対話型プロンプトを使用する必要がない場合は、次のようにコマンドを直接入力します。

```
# chronyc activity
```



注記: 一時的な変更

chronycを使用して行われた変更は、永続的ではありません。変更内容は、次の**chronyd**の再起動後に失われます。永続的な変更を行う場合は、`/etc/chrony.conf`を変更してください。

chronycコマンドの完全なリストについては、そのマニュアルページ(**man 1 chronyc**)を参照してください。

39.4 ランタイム時の動的時刻同期

chronydは、ネットワーク接続なしでブートするシステムでは正常に起動しますが、ツールは設定ファイルで指定されたタイムサーバのDNS名を解決できません。

chronydは、`server`、`pool`、および**peer**ディレクティブによって指定されたタイムサーバ名を、時間間隔を増やして成功するまで解決しようとします。

chronydの起動時にタイムサーバにアクセスできない場合は、`offline`オプションを指定することができます。

```
server server_address offline
```

この場合、**chronyd**は、次のコマンドを使用して有効にするまで、サーバをポーリングしようとしません。

```
# chronyc online server_address
```

`auto_offline`オプションが設定されている場合、タイムサーバに2つの要求を送信して応答を受信しなかったときに、**chronyd**はそのタイムサーバがオフラインになったとみなします。このオプションにより、ネットワークリンクを切断するときに**chronyc**から'`offline`'コマンドを実行する必要がなくなります。

39.5 ローカルリファレンスクロックの設定

ソフトウェアパッケージ**chrony**は、SHMまたはSOCKドライバを介してタイミングデータにアクセスするために、他のプログラム(`gpsd`など)を利用しています。`/etc/chrony.conf`の**refclock**ディレクティブを使用して、時間ソースとして使用するハードウェア基準クロックを指定します。これには、2つの必須パラメータ(ドライバ名とドライバ固有のパラメータ)があります。2つのパラメータの後には、ゼロ以上の**refclock**オプションが続きます。**chronyd**には、次のドライバが含まれています。

- PPS - カーネル pulse per second APIのドライバ。例:

```
refclock PPS /dev/pps0 lock NMEA refid GPS
```

- SHM - NTP共有メモリドライバ。例:

```
refclock SHM 0 poll 3 refid GPS1  
refclock SHM 1:perm=0644 refid GPS2
```

- SOCK - Unixドメインソケットドライバ。例:

```
refclock SOCK /var/run/chrony.ttyS0.sock
```

- PHC - PTPハードウェアクロックドライバ。例:

```
refclock PHC /dev/ptp0 poll 0 dpoll -2 offset -37  
refclock PHC /dev/ptp1:nocrossts poll 3 pps
```

個々のドライバのオプションの詳細については、man 8 chrony.confを参照してください。

39.6 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
```

VI **トラブルシューティング**

- 40 ヘルプとドキュメント **504**
- 41 サポート用システム情報の収集 **510**
- 42 最も頻繁に起こる問題およびその解決方法 **541**

40 ヘルプとドキュメント

SUSE® Linux Enterprise Desktop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40.1項「**ドキュメントディレクトリ**」を参照してください。

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40.2項「**マニュアルページ**」および40.3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。

特定のアプリケーション用の別のヘルプパッケージ

YaSTを使って新しくソフトウェアをインストールした場合、通常はそのソフトウェアのドキュメントも自動的にインストールされ、デスクトップのHelp Centerに表示されます。ただし、GIMPなどの特定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。

40.1 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、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。

40.1.1 SUSEマニュアル

これらのガイドブックは、HTMLおよびPDFの各バージョンを複数の言語で提供しています。`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PDF版およびHTML版の文書の検索場所については、SUSE Linux Enterprise Desktop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(<https://www.suse.com/releasesnotes/>)で参照できます。

40.1.2 パッケージのマニュアル

`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るドキュメントがあります。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。これには詳細情報が含まれているため、開発者にとって有用です。

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。

TODO

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。

40.2 マニュアルページ

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 40.1 「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 40.1: マニュアルページ - 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 <u>/dev</u> 内にあります)
5	ファイル形式と命名規則(<u>/etc/fstab</u>)
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 <u>root</u> に関するもののみ)
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。

40.3 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。**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。

40.4 リソースのオンライン化

SUSE Linux Enterprise Desktop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、<https://documentation.suse.com/> にある製品ごとのマニュアルに関するWebページをご覧ください。

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、<https://www.suse.com/support/> でSUSEテクニカルサポートを利用できます。

SUSE Linux Enterprise Desktopユーザコミュニティ

SUSE and Rancher Community (<https://community.suse.com/>) 

SUSEブログ

SUSEブログでは、記事、ヒント、質疑応答を提供しています。<https://www.suse.com/c/blog/> 

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(<https://help.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(<https://www.tldp.org> を参照)。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 Desktop上でも利用できます。

汎用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。

41 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 Desktop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、パーティションセットアップ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 Desktop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。

41.1 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すばや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 41.1: `root`としてログインしたときの`hostinfo`の出力

```
Welcome to SUSE Linux Enterprise Server 15 SP2 Snapshot8 (x86_64) - Kernel \r (\l).

Distribution:      SUSE Linux Enterprise Server 15 SP2
Current As Of:    Wed 25 Mar 2020 12:09:20 PM PDT
Hostname:         localhost
Kernel Version:   5.3.18-8-default
Architecture:     x86_64
Installed:        Thu 19 Mar 2020 11:25:13 AM PDT
Status:           Not Tainted
```

```
Last Installed Package: Wed 25 Mar 2020 11:42:24 AM PDT
Patches Needed: 0
Security: 0
3rd Party Packages: 219
Network Interfaces
eth0: 192.168.2/24 2002:c0a8:20a::/64
Memory
Total/Free/Avail: 7.4Gi/6.4Gi/6.8Gi (91% Avail)
CPU Load Average: 7 (3%) with 2 CPUs
```

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、[41.6項「カーネルモジュールのサポート」](#)を参照してください。

41.2 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引き渡すことができる詳細なシステム情報を含むTARアーカイブを作成するには、次のいずれかを使用します。

- コマンド `supportconfig` または、
- YaST サポートモジュール。

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。

システムにインストールされているパッケージに応じて、これらのパッケージの一部に Supportconfig プラグインが組み込まれています。Supportconfig を実行すると、すべてのプラグインも同様に実行され、アーカイブ用の1つ以上の結果ファイルが作成されます。これには、チェックされるトピックが特定のプラグインを含むものだけになるという利点があります。Supportconfig プラグインは、ディレクトリ `/usr/lib/supportconfig/plugins/` に格納されています。


41.2.1 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、<https://scc.suse.com/support/requests> にアクセスして、画面の指示に従います。サービス要求番号を書き留めます。







注記: プライバシーポリシー

SUSE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、<https://www.suse.com/company/policies/privacy/> を参照してください。

41.2.2 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41.1「YaSTを使用したサポートへの情報の送信」または手順41.2「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。

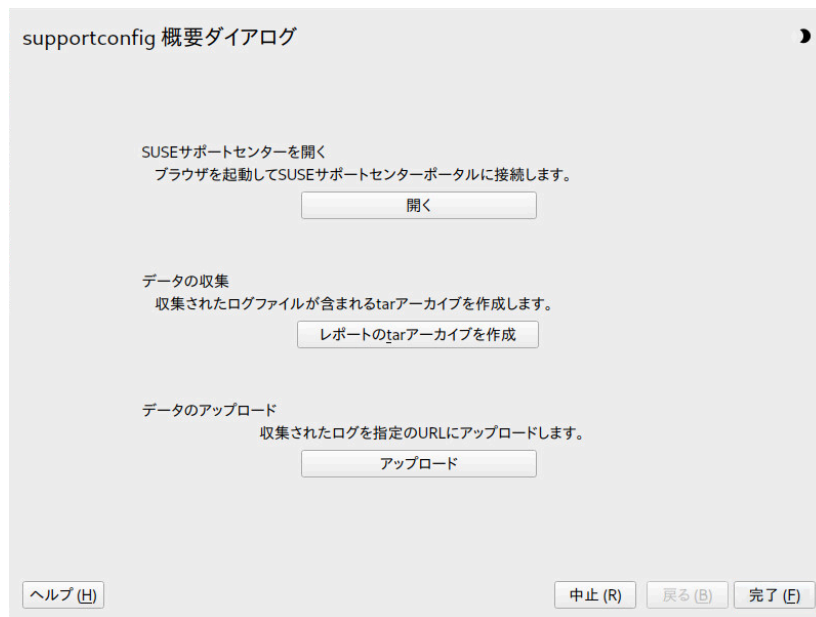
- 北米: FTP <ftp://support-ftp.us.suse.com/incoming/> 、FTPS <ftps://support-ftp.us.suse.com/incoming/> 
- EMEA (ヨーロッパ、中東、およびアフリカ): FTP <ftp://support-ftp.emea.suse.com/incoming> 、FTPS <ftps://support-ftp.emea.suse.com/incoming> 

または、サービス要求URL <https://scc.suse.com/support/requests> 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。

41.2.3 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)があらかじめ選択されています。最初にレポート機能をテストするには、Only gather a minimum amount of info (最小限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詳細については、[supportconfig](#)のマニュアルページを参照してください。
次へをクリックします。
4. 連絡先情報を入力します。この情報はbasic-environment.txtファイルに保存され、作成されるアーカイブに組み込まれます。
5. 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は、必須の情報のアップロードに入力します。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、[41.2.2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には、情報のアップロードを空のままにします。
6. 次へをクリックして、情報収集プロセスを開始します。



プロセスが完了したら、次へをクリックします。

7. 収集されたデータを確認するため、ファイル名から目的のファイルを選択して、そのコンテンツをYaSTで表示します。サポートへの送信前にTARアーカイブからファイルを削除するには、データから削除を使用します。次へをクリックします。
8. 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、アーカイブを/var/logに保存するよう求められます(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、scc_HOST_DATE_TIME.tbzです。
9. アーカイブをサポートに直接アップロードする場合は、ログファイルのtarアーカイブをURLへアップロードが有効になっていることを確認します。ここに表示されるアップロード先は、ステップ5でYaSTによって提案されたものです。アップロード先を変更するには、41.2.2項「アップロード先」で使用可能なアップロードサーバを確認します。
10. アップロードをスキップするには、ログファイルのtarアーカイブをURLへアップロードを無効にします。
11. 変更内容を確認し、YaSTモジュールを閉じます。

41.2.4 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41.2「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. **supportconfig**を実行します。このツールをオプションなしで実行するだけで十分です。ただし、最も一般的なオプションが次のリストに表示されます。

```
-E MAIL,  
-N NAME,  
-O COMPANY,  
-P PHONE
```

次の連絡先データを設定します。電子メールアドレス(-E)、会社名(-O)、名前(-N)、および電話番号(-P)。

```
-i KEYWORDS,  
-F
```

チェックする機能を制限します。プレースホルダKEYWORDSは、大文字と小文字を区別したキーワードのカンマ区切りのリストです。supportconfig -Fですべてのキーワードのリストを取得します。

```
-r SRNUMBER
```

生成されたTARアーカイブをアップロードする際のサービス要求番号を定義します。

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scc_HOST_DATE_TIME.tbzです。

41.2.5 supportconfigの出力の理解

supportconfigをYaSTを介して実行するか直接実行するかにかかわらず、このスクリプトは実行した内容の概要を示します。

```
Support Utilities - Supportconfig
```

```

Script Version: 3.0-98
Script Date: 2017 06 01

[...]
Gathering system information
  Data Directory:    /var/log/scc_d251_180201_1525 ❶

  Basic Server Health Check...           Done ❷
  RPM Database...           Done ❷
  Basic Environment...       Done ❷
  System Modules...         Done ❷
[...]
  File System List...        Skipped ❸
[...]
  Command History...        Excluded ❹
[...]
  Supportconfig Plugins:      1 ❺
    Plugin: pstree...        Done
[...]
Creating Tar Ball

==[ DONE ]=====
Log file tar ball: /var/log/scc_d251_180201_1525.txz ❻
Log file size:      732K
Log file md5sum:    bf23e0e15e9382c49f92cbce46000d8b
=====

```

- ❶ 結果を格納する一時データディレクトリ。このディレクトリはtarファイルとしてアーカイブされています。❻を参照してください。
- ❷ この機能は(デフォルトで、または手動で選択されて)有効化され、正常に実行されました。結果はファイルに保存されます(表41.1「[TARアーカイブの機能とファイル名の比較](#)」参照)。
- ❸ 1つ以上のRPMパッケージのファイルが変更されたため、この機能はスキップされました。
- ❹ この機能は-xオプションで選択解除されたため、除外されました。
- ❺ スクリプトは1つのプラグインを見つけました。プラグイン**pstree**を実行します。このプラグインは、ディレクトリ/usr/lib/supportconfig/plugins/にありました。詳細については、マニュアルページを参照してください。
- ❻ アーカイブのtarファイル名。デフォルトではxzで圧縮されています。

41.2.6 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。

```
> sudo supportconfig -m
```

情報を特定のトピックに限定する

特定の領域または機能セットにのみ関係する問題をすでに特定している場合は、**supportconfig**の次回実行時に収集する情報を特定の領域に限定する必要があります。たとえば、LVMに関する問題を検出し、LVM設定に対して行った最近の変更をテストしたい場合などです。この場合、LVMのみに関する最低限のSupportconfig情報を収集することが理にかなっています。

```
> sudo supportconfig -i LVM
```

追加のキーワードはカンマで区切ることができます。たとえば、追加のディスクテストの場合、次のようになります。

```
> sudo supportconfig -i LVM,DISK
```

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。

```
> sudo supportconfig -F
```

追加の連絡先情報を出力に含める

```
> sudo 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

```
> sudo 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。

41.2.7 アーカイブコンテンツの概要

TARアーカイブには、各種機能のすべての結果が含まれています。選択した内容(すべてまたは小さなセットのみ)に応じて、アーカイブに含まれるファイルの数は変わります。機能のセットは、`-i`オプションによって制限できます(41.2.6項「`supportconfig`の一般的なオプション」を参照)。

アーカイブの内容を一覧表示するには、次の`tar`コマンドを使用します。

```
# tar xf /var/log/scc_earth_180131_1545.tbz
```

TARアーカイブ内には、次のファイル名のファイルが常にあります。

アーカイブ内にある最低限のファイル

basic-environment.txt

このスクリプトが実行された日付と、ディストリビューションのバージョン、ハイパーバイザ情報などのシステム情報が記載されています。

basic-health-check.txt

アップタイム、仮想メモリ統計、空きメモリとハードディスク、ゾンビプロセスのチェックなどの基本的なヘルスチェックが記載されています。

hardware.txt

CPUアーキテクチャに関する情報、接続されているすべてのハードウェアのリスト、割り込み、I/Oポート、カーネルブートメッセージなどの基本的なハードウェアチェックが記載されています。

messages.txt

システムジャーナルからのログメッセージが記載されています。

rpm.txt

インストールされているすべてのRPMパッケージのリスト、名前、発信元、およびバージョンが記載されています。

summary.xml

ディストリビューション、バージョン、製品固有のフラグメントなどのXML形式の情報が記載されています。

supportconfig.txt

supportconfigスクリプト自体に関する情報が記載されています。

y2log.txt

特定のパッケージ、設定ファイル、ログファイルなどのYaST固有の情報が記載されています。

表41.1 「TARアーカイブの機能とファイル名の比較」には、利用可能なすべての機能とそのファイル名が一覧表示されています。プラグインと同様に、追加のサービスパックでリストを拡張することができます。

表 41.1: TARアーカイブの機能とファイル名の比較

機能	ファイル名
<u>APPARMOR</u>	<u>security-apparmor.txt</u>
<u>AUDIT</u>	<u>security-audit.txt</u>
<u>AUTOFS</u>	<u>fs-autofs.txt</u>
<u>BOOT</u>	<u>boot.txt</u>
<u>BTRFS</u>	<u>fs-btrfs.txt</u>
<u>DAEMONS</u>	<u>systemd.txt</u>
<u>CIMOM</u>	<u>cimom.txt</u>
<u>CRASH</u>	<u>crash.txt</u>
<u>CRON</u>	<u>cron.txt</u>
<u>DHCP</u>	<u>dhcp.txt</u>
<u>DISK</u>	<u>fs-diskio.txt</u>
<u>DNS</u>	<u>dns.txt</u>
<u>DOCKER</u>	<u>docker.txt</u>
<u>DRBD</u>	<u>drbd.txt</u>
<u>ENV</u>	<u>env.txt</u>
<u>ETC</u>	<u>etctxt</u>
<u>HA</u>	<u>ha.txt</u>
<u>HAPROXY</u>	<u>haproxy.txt</u>
<u>HISTORY</u>	<u>shell_history.txt</u>
<u>IB</u>	<u>ib.txt</u>

機能	ファイル名
IMAN	novell-iman.txt
ISCSI	fs-iscsi.txt
LDAP	ldap.txt
LIVEPATCH	kernel-livepatch.txt
LVM	lvm.txt
MEM	memory.txt
MOD	modules.txt
MPIO	mpio.txt
NET	network-*.txt
NFS	nfs.txt
NTP	ntp.txt
NVME	nvme.txt
OCFS2	ocfs2.txt
OFILES	open-files.txt
PRINT	print.txt
PROC	proc.txt
SAR	sar.txt
SLERT	slert.txt
SLP	slp.txt
SMT	smt.txt
SMART	fs-smartmon.txt
SMB	samba.txt
SRAID	fs-softraid.txt

機能	ファイル名
<u>SSH</u>	<u>ssh.txt</u>
<u>SSSD</u>	<u>sssd.txt</u>
<u>SYSCONFIG</u>	<u>sysconfig.txt</u>
<u>SYSFS</u>	<u>sysfs.txt</u>
<u>TRANSACTIONAL</u>	<u>transactional-update.txt</u>
<u>TUNED</u>	<u>tuned.txt</u>
<u>UDEV</u>	<u>udev.txt</u>
<u>UFILES</u>	<u>fs-files-additional.txt</u>
<u>UP</u>	<u>updates.txt</u>
<u>WEB</u>	<u>web.txt</u>
<u>X</u>	<u>x.txt</u>

41.3 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、[41.2.1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、[41.2.1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 41.1: 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。[41.2.3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、[41.2.3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、[41.2.2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 41.2: 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、[41.2.3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。

```
> sudo 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。

```
> sudo 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。

```
> sudo supportconfig -r 12345678901
```

b. `/var/log/scc_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、[41.2.2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。

41.4 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 Desktop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(パッケージ `sca-server-report`によって提供される **scatool**)は、クライアント側で実行され、コマンドラインから実行されます。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。

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、[41.4.3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。

41.4.1 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41.1「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 41.1: 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います。このツールには、タブ補完機能を備えた対話型コンソールも用意されています。外部マシン上で **supportconfig** を実行し、ローカルマシン上でその後の分析を実行することが可能です。

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、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、**<Tab>** 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。

41.4.2 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します。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

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、[41.4.2.5.4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。

41.4.2.1 インストールのクイックスタート

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、この手順に従います。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、[41.4.2.2項「前提条件」](#)～[41.4.2.3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。

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 41.3: 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て、次のコマンドを実行します(推奨パッケージを受け入れてください)。

```
> sudo zypper install sca-appliance-* sca-patterns-* \
vsftpd yast2 yast2-ftp-server
> sudo systemctl enable apache2
> sudo systemctl start apache2
> sudo systemctl enable vsftpd
> sudo systemctl start vsftpd
> sudo 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。

```
> sudo systemctl enable mysql
> sudo systemctl start mysql
> sudo mysql_secure_installation
> sudo 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 41.4: 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。

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

41.4.2.2 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)

41.4.2.3 インストールと基本セットアップ

41.4.2.2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。
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 mysql
```

4. 両方のサービスを開始します。

```
> sudo systemctl start apache2 mysql
```

これで、[手順41.5「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 41.5: 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 12のサーバとSUSE Linux Enterprise Server 15のサーバがある場合、sca-patterns-sle12パッケージとsca-patterns-sle15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

```
> sudo 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ます。画面の指示に従います。

```
> sudo 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には、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ます。画面の指示に従います。

```
> sudo 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。41.4.2.5.1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。41.4.2.4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のオプションを変更する必要があります。詳細については、41.4.2.5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のレポートには、セキュリティ関連の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が不正なアクセスから確実に保護されるようにしてください。

41.4.2.4 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が分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 41.5 「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。

41.4.2.4.1 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には、既存のアーカイブファイル(`/var/log/scc_*.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。

41.4.2.4.2 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。

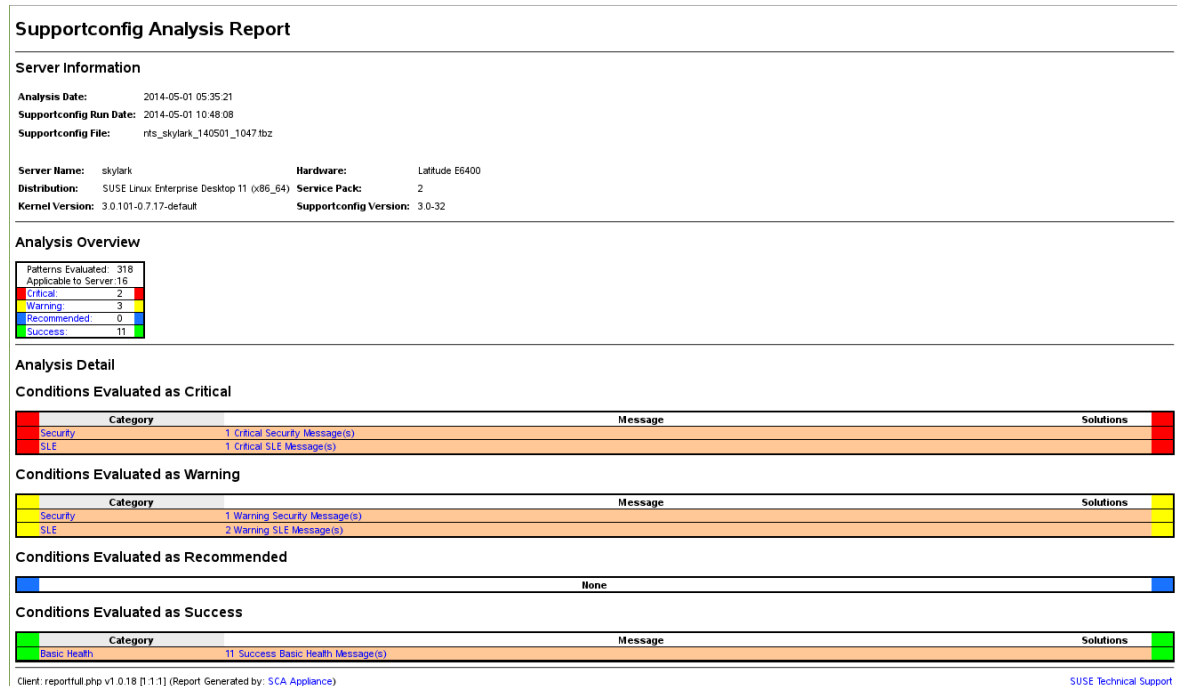


図 41.2: SCAアプライアンスによって生成されるHTMLレポート

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 (<https://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。

41.4.2.5 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。

41.4.2.5.1 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41.5「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 41.6: 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. `/srv/www/htdocs/sca/web-config.php`をエディタで開きます。
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。

41.4.2.5.2 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 15 SP5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RMT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ンネルからアップデートを取得します。このチャンネルがRMT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 41.7: 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. `/etc/sca/sdagent-patterns.conf`をエディタで開きます。
3. 次のエントリを変更します。

```
UPDATE_FROM_PATTERN_REPO=1
```

次のように変更してください。

```
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。

41.4.2.5.3 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 41.8: 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. `/etc/sca/sdagent.conf`をエディタで開きます。
3. 次のエントリを変更します。

```
ARCHIVE_MODE=0
```

次のように変更してください。

```
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。

41.4.2.5.4 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することができます。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 41.9: 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. `/etc/sca/sdagent.conf`をエディタで開きます。
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。例:

```
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。

41.4.2.6 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。**`scadb`**は、パッケージ**`sca-appliance-broker`**によって提供されます。

手順 41.10: 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(41.4.3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 41.11: 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
```

41.4.3 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、<https://www.suse.com/c/blog/sca-pattern-development/> を参照してください。

41.5 インストール時の情報収集

インストール時には、**supportconfig**を使用できません。ただし、**save_y2logs**を使用してYaSTからログファイルを収集することができます。このコマンドにより、`/tmp`ディレクトリに`.tar.xz`アーカイブが作成されます。

インストール開始後に問題が発生したときは、**linuxrc**で作成したログファイルから情報を収集できる場合があります。**linuxrc**は、YaSTが起動する前に実行される小さなコマンドです。このログファイルは、`/var/log/linuxrc.log`にあります。



重要: インストールログファイルがインストールしたシステムにない

インストール時に使用可能だったログファイルが、インストールしたシステムでは使用できなくなっていました。インストーラの実行中に、インストールログファイルを正しく保存してください。

41.6 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 Program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。サポートされていないカーネルモジュールは、別のRPMパッケージ(kernel-FLAVOR-extra)に含まれています。このパッケージは、SUSE Linux Enterprise DesktopおよびSUSE Linux Enterprise Workstation Extensionでのみ使用できます。デフォルト(FLAVOR=default|xen|...)では、これらのカーネルはロードされません。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。

41.6.1 技術的背景

- **Linuxカーネル:** SUSE Linux Enterprise 15 SP5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)。このデフォルト値は、インストーラと、インストールしたシステムで使用されます。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、41.6.2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: サポート

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。

41.6.2 サポート対象外のモジュールの使用

一般的なサポート性は重要ですが、サポート対象外のモジュールのロードが必要な状況が発生する可能性があります。たとえば、テストまたはデバッグ目的の場合や、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です。

- デフォルトを上書きするには、`/lib/modprobe.d/10-unsupported-modules.conf`を`/etc/modprobe.d/10-unsupported-modules.conf`にコピーし、変数`allow_unsupported_modules`の値を0から1に変更します。`/lib/modprobe.d/10-unsupported-modules.conf`を直接編集しないでください。`suse-module-tools`パッケージが更新されるたびに、変更内容が上書きされます。
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、`/lib/modprobe.d/10-unsupported-modules.conf`および**modprobe**マニュアルページのコメント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、もうつの特殊なカーネルコマンドラインオプション(`oem-modules=1=1`)を使用していない場合は、引き続きデフォルトで、サポート対象外のモジュールは許可されません。

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。

41.7 詳細情報

- `man supportconfig`—`supportconfig` 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` マニュアルページ。
- `man scadb`—`scadb` マニュアルページ。
- `man setup-sca`—`setup-sca` マニュアルページ。
- <https://mariadb.com/kb/en/> —MariaDB のマニュアル
- <https://www.suse.com/c/blog/sca-pattern-development/> —独自の SCA パターンを作成(およびテスト)する方法
- <https://www.suse.com/c/blog/basic-server-health-check-supportconfig/> —「A Basic Server Health Check with Supportconfig」
- <https://community.microfocus.com/img/gw/groupwise/w/groupwise/34308/create-your-own-supportconfig-plugin> —「Create Your Own Supportconfig Plugin」
- <https://www.suse.com/c/blog/creating-a-central-supportconfig-repository/> —「Creating a Central Supportconfig Repository」

42 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。

42.1 情報の検索と収集

Linuxでは、詳細なレポートが提供されます。システムで問題が発生したときに参照すべき場所がいくつかあります。そのほとんどは、一般的にLinuxシステム標準です。一部は、SUSE Linux Enterprise Desktopシステムと関係があります。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は、サポートチームが必要とするすべてのシステム情報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 42.1: ログファイル

ログファイル	説明
<u>~/.xsession-errors</u>	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。
<u>/var/log/apparmor/</u>	AppArmorからのログファイル。詳細については、『Security and Hardening Guide』を参照してください。
<u>/var/log/audit/audit.log</u>	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、『Security and Hardening Guide』を参照してください。
<u>/var/log/mail.*</u>	メールシステムから受け取るメッセージです。

ログファイル	説明
<u>/var/log/NetworkManager</u>	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。
<u>/var/log/samba/</u>	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。
<u>/var/log/warn</u>	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。
<u>/var/log/wtmp</u>	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。
<u>/var/log/Xorg.*.log</u>	Xウィンドウシステムからの、起動時およびランタイム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。
<u>/var/log/YaST2/</u>	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。
<u>/var/log/zypper.log</u>	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。[表 42.2: /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 42.2: /procファイルシステムによるシステム情報

ファイル	説明
<u>/proc/cpuinfo</u>	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。
<u>/proc/dma</u>	どのDMAチャネルが現在使用されているかを表示します。

ファイル	説明
<u>/proc/interrupts</u>	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。
<u>/proc/iomem</u>	I/Oメモリの状態を表示します。
<u>/proc/ioports</u>	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。
<u>/proc/meminfo</u>	メモリステータスを表示します。
<u>/proc/modules</u>	個々のモジュールを表示します。
<u>/proc/mounts</u>	現在マウントされているデバイスを表示します。
<u>/proc/partitions</u>	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。
<u>/proc/version</u>	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29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 42.3](#)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 42.3: /sysファイルシステムによるシステム情報

ファイル	説明
<u>/sys/block</u>	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。
<u>/sys/bus</u>	各物理バスタイプのサブディレクトリが含まれます。

ファイル	説明
<u>/sys/class</u>	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)。
<u>/sys/device</u>	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用意されています。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、『System Analysis and Tuning Guide』、第2章「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。

42.2 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)。

42.2.1 GRUB 2ブートローダをロードできない

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、[42.5.2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、[42.5.2.4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。

または、次の手順でレスキューシステムを使用してブートローダを修復できます。インストールメディアからマシンをブートします。ブート画面で、[詳細 > Linuxシステムのブート](#)を選択します。インストール済みシステムとカーネルが含まれるディスク、およびデフォルトのカーネルオプションを選択します。

システムがブートしたら、YaSTを起動してシステム > ブートローダに切り替えます。MBRに汎用ブートコードを書き込むオプションが有効になっていることを確認して、OKをクリックします。これにより、ブートローダが壊れている場合は上書きして修復し、ブートローダが見つからない場合はインストールします。

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が単に開始されてい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。

42.2.2 ログインなし、またはプロンプトが表示される

通常、これはカーネルのアップグレードに失敗した後に発生します。プロセスの最終段階にときどき発生するシステムコンソールのエラーであるということから、「カーネルパニック」として知られます。」ソフトウェアのアップグレードに続いてコンピュータを再起動した直後である場合は、実証済みの古いバージョンのLinuxカーネルおよび関連ファイルを使用して再起動します。この操作は、ブートプロセス中にGRUB 2ブートローダ画面で行います。次の手順に従ってください。

1. コンピュータをリセットボタンで再起動するか、いったんオフにしてからオンにします。
2. GRUB 2のブート画面が表示されたら、詳細オプション項目を選択して、以前のカーネルをメニューから選択します。コンピュータは、旧バージョンのカーネルとその関連ファイルを使用して起動します。
3. ブートプロセスが完了してから、新たにインストールされたカーネルを削除します。必要に応じて、デフォルトのブート項目を、YaSTブートローダモジュールを使用する古いカーネルに設定します。詳細については、[18.3項「YaSTによるブートローダの設定」](#)を参照してください。ただし、多くの場合、このファイルの更新は必要ありません。このファイルは、通常、ロールバック処理時に自動更新ツールにより変更されるからです。
4. 再起動します。

それでも問題が解決されない場合は、インストールメディアを使用してコンピュータをブートしてください。コンピュータが起動したら、「[ステップ 3](#)」を続行します。

42.2.3 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、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。エラーメッセージは**journalctl**コマンド(詳細は第21章「**journalctl: systemdジャーナルのクエリ**」を参照)で問い合わせることができます。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。

42.2.4 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。

42.2.5 ルートパーティションを強制的に確認する

ルートパーティションが壊れた場合、パラメータ **forcefsck** をブートプロンプトで使用します。これにより、オプション **-f** (強制) が **fsck** コマンドに渡されます。

42.2.6 スワップを無効にしてブートを有効にする

スワップデバイスが使用できず、システムがブート中に有効にできない場合、ブートは失敗する場合があります。次のオプションをカーネルコマンドラインに追加して、すべてのスワップデバイスを無効にしてみます。

```
systemd.device_wants_unit=off systemd.mask=swap.target
```

特定のスワップデバイスを無効にしてみることもできます。

```
systemd.mask=dev-sda1.swap
```

42.2.7 デュアルブートシステムでの再起動中にGRUB 2が失敗する

再起動中にGRUB 2が失敗した場合は、BIOSのFast Boot設定を無効にします。

42.3 ログインの問題

ログインの問題は、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に発生します(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)。

42.3.1 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、[42.4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。このトピックの詳細は、[42.4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い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、『Security and Hardening Guide』、第2章「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、[42.3.3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** 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。

42.3.2 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、[42.3.4項「GNOMEデスクトップに問題があります」](#)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、[42.3.4項「GNOMEデスクトップに問題があります」](#)に進みます。

次に、特定のマシンで特定のユーザのネットワーク認証が失敗する一般的な理由を示します。

- 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**で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、[42.3.4 項「GNOMEデスクトップに問題があります」](#)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、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、42.3.4項「GNOMEデスクトップに問題があります」に進みます。

42.3.3 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、その理由はパーティションのロックを解除できなかったためである場合があります。

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。

42.3.4 GNOMEデスクトップに問題があります

GNOMEデスクトップで問題が発生している場合は、グラフィカルなデスクトップ環境の動作不良をトラブルシューティングするためのいくつかの方法があります。次に説明する推奨手順は、壊れたGNOMEデスクトップを修復するための最も安全なオプションを提供します。

手順 42.1: GNOMEのトラブルシューティング

1. YaSTを起動し、セキュリティとユーザに切り替えます。
2. ユーザおよびグループの管理ダイアログを開いて、追加をクリックします。

3. 必須フィールドに入力して、OKをクリックし、新しいユーザを作成します。
4. ログアウトして、新しいユーザとしてログインします。これにより、新しいGNOME環境が提供されます。
5. 古いユーザアカウントの`~/.local/`と`~/.config/`ディレクトリから新しいユーザアカウントの各ディレクトリに個々のサブディレクトリをコピーします。
コピー操作を行うたびにログアウトして、新しいユーザとして再度ログインし、GNOMEが引き続き正常に動作するかどうかを確認します。
6. GNOMEを壊す設定ファイルが見つかるまで、前の手順を繰り返します。
7. 古いユーザとしてログインし、問題のある設定ファイルを別の場所に移動します。ログアウトして、古いユーザとして再度ログインします。
8. 以前に作成したユーザを削除します。

42.4 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。たとえば、システムにユーザがログインできない理由は、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 42.2: 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. 無線接続を使用する場合、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。次のリストには、セットアップにかかわ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マウントされたディレクトリに保存されたデータを必要とするアプリケーションがあった場合、このサービスがダウンしてるか、間違ってい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、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)

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アプリケーションがある場合、そのアプリケーションは起動できないか、正しく機能しません。

NIS (ユーザ管理)

SUSE Linux Enterprise Desktop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 Desktop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/HOSTNAME(サーバのホスト名またはIPアドレスで置き換えます)を使って、サーバが起動中で、ネットワークに反応するかどうか確認します。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、他のコンピュータから**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コンピュータに接続可能な場合には、宛先のサーバが動作していないか、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。

/var/run/netconfig/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。これは /run/netconfig/resolv.conf へのシンボリックリンクであり、通常、YaSTまたはDHCPによって自動的に調整されます。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。

ネットワーク接続をDHCPで処理している場合は、DHCPでホスト名とネームサービスの情報を変更できるようにします。このためには、YaSTネットワーク設定モジュール([ホスト名/DNS] タブ)で、DHCPでホスト名を設定(すべてのインタフェースに対してグローバルに設定することも、インタフェースごとに設定することもできます)、およびUpdate Name Servers and Search List via DHCP (DHCPでネームサーバと検索リストを更新)を選択します。

/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。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到達可能かどうかをリストします。tracerouteおよびpingコマンドを組み合わせることで原因を追究し、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。

42.4.1 NetworkManagerの問題

ネットワーク接続に問題がある場合は、[手順42.2「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。

```
> sudo systemctl restart NetworkManager
```

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の状態に関する情報を収集します。

NetworkManagerについての詳細は、[第31章「NetworkManagerの使用」](#)を参照してください。

42.5 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。

42.5.1 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、領域の節約になります。

手順 42.3: 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
```

パーティションをバックアップするだけでよい場合は、**SOURCE**プレースホルダを各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。

42.5.2 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 Desktop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。

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: プロンプトで「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. 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。『Deployment Guide』、第13章「Preparing network boot environment」、13.5項「Using wake-on-LAN for remote wakeups」
3. `Rescue`: プロンプトで「`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` キーを使って、仮想コンソールを使用することができます。

シェルおよび他の便利なユーティリティ(マウントプログラムなど)は、`/bin`ディレクトリにあります。`/sbin`ディレクトリには、ファイルシステムを検討し、修復するための重要なファイルおよびネットワークユーティリティが入っています。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。

42.5.2.1 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。

```
> sudo 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。

```
> sudo 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。

```
> sudo umount /mnt
```

6. マシンを再起動します。

42.5.2.2 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。このシステムには、[btrfs](#)、[ext2](#)、[ext3](#)、[ext4](#)、[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は、**fsck.btrfs** を使用します。

42.5.2.3 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. ヒント: LVMボリュームグループのインポート

LVMセットアップを使用している場合は(詳細については、『Deployment Guide』、第6章「Expert Partitioner」、6.2項「LVM configuration」を参照)、既存のボリュームグループをすべてインポートし、デバイスを検索してマウントできます。

```
rootvgimport -a
```

lsblkを実行して、ルートパーティションに対応するノードを確認します。この例では/dev/sda2です。

```
> lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda          8:0    0 149,1G  0 disk
├─sda1       8:1    0    2G  0 part  [SWAP]
├─sda2       8:2    0   20G  0 part  /
└─sda3       8:3    0  127G  0 part
   └─cr_home 254:0    0  127G  0 crypt /home
```

2. インストール済みシステムからルートパーティションをマウントします。

```
> sudo mount /dev/sda2 /mnt
```

3. /proc、/dev、および/sysの各パーティションをマウントします。

```
> sudo mount -t proc none /mnt/proc
> sudo mount --rbind /dev /mnt/dev
> sudo mount --rbind /sys /mnt/sys
```

4. これで、**bash**シェルを維持したまま、新規の環境に「ルートを変更」できます。

```
> chroot /mnt /bin/bash
```

5. 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
```

6. 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同一でない限り、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。

42.5.2.4 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. 42.5.2.3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. GRUB 2ブートローダ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。インストールされていない場合、`grub2`パッケージをインストールして実行します。

```
> sudo grub2-install /dev/sda
```

3. 次のファイルが第18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`
- `/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`

4. 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。

```
> sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 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
```

42.5.2.5 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
```

この内容は、`/etc/zypp/zypp.conf` ファイルに保存されます。詳細については、[第27章「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 Desktop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になる場合があります。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。

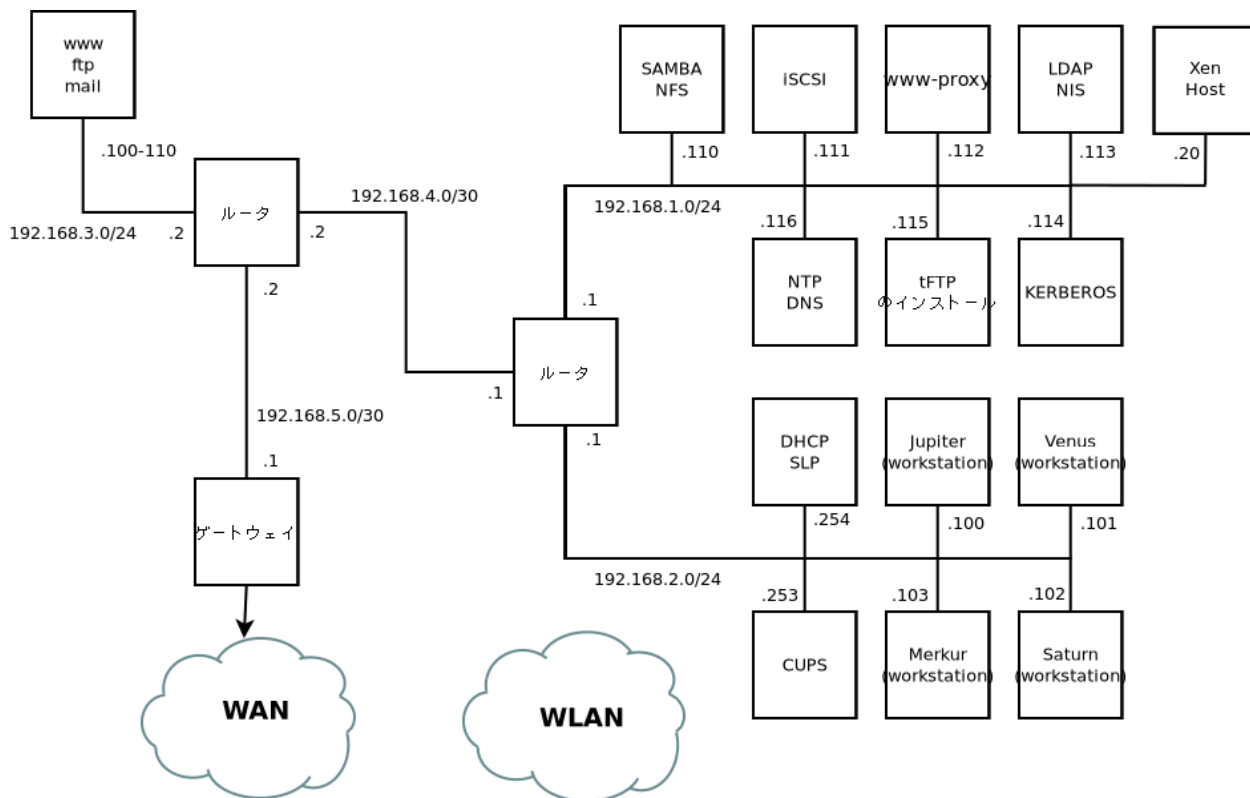
両方のケースで、レスキューモードでインストールされているシステムにアクセスし、カーネル関係の問題を修正する必要があります。さもないと、システムが正しくブートしないことがあります。

1. SUSE Linux Enterprise Desktopインストール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: プロンプトで「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、[42.5.2.3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、[42.5.2.4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。

A 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 Desktopのドキュメントの、すべてのネットワーク関連の章で使用されます。



B 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 non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License 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 non-commercially, 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 <https://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.