

Red Hat[®] RPM Guide

Eric Foster-Johnson 著
ヴァインカーブ 監訳
中川和夫 訳



Red Hat RPM Guide

by Eric Foster-Johnson

Copyright © 2003 Red Hat, Inc.

Japanese language edition publishing by Softbank Publishing Inc.

Copyright © 2003 Softbank Publishing Inc., Tokyo.Japan.

Japanese translation rights arranged with Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256 USA through Japan UNI Agency, Inc., Tokyo.Japan.

Original Cover design by Michael J. Freeland

Cover photo © Hulton/Getty

UNIXはThe Open Groupsがライセンスする登録商標です。

LinuxはLinus Torvaldsの米国およびその他の国における登録商標あるいは商標です。

Red HatはRed Hat, Inc.の登録商標です。

X Window SystemはX Consortium, Inc.の登録商標です。

その他、本書中に記載のシステム、製品名などは一般に各社の登録商標または商標です。

なお、本文中にはTM, ®マークは明記していません。

本書の内容はすべて著作権法上の保護を受けています。著作権者および出版権者の文書による許諾を得ず、
本書の内容の一部、またはすべてを無断で複写・複製・転載することは禁じられています。

Norma , Katya , Nalanaに捧ぐ

謝辞

私が今使っているLinuxディストリビューションを開発してくれたことはもちろん、本書の刊行にあたって惜しみなく協力してくれたことに対し、Red Hat社のスタッフに心から感謝したい。私が接触したスタッフは誰もが協力的で、貴重な知恵や知識を提供してくれた。どうもありがとう。

妻のNormaにも感謝しなければならない。彼女はこの仕事の間私を助け、支えてくれた。

発刊によせて

RPMは簡単だ。「難しい」と聞かされているかもしれないし、自分でもそう思っているかもしれないが、実際には簡単だ。問題は、ちゃんと説明されていないことにある。もちろん、manページであれ、インターネットであれ、RPMに関する資料に不足はない。そのうえRPMはオープンソースのソフトウェアだ。だが、こうした資料からRPMを学ぶには時間がかかる。試行錯誤を何度も繰り返すことになり、キーボードをたたきつけ、歯ぎしりし、いらいらがつのる。

RPMをこれほどまでに詳しく解説した本は今までなかった。本書は基本的なコマンドから複雑なAPIに至るまで、RPMのすべてをカバーしている。RPMを作成するにしろ、RPMを操作するツールを使うにしろ、システム管理者にとっても開発者にとっても、本書はRPMの機能をフルに活用するうえで欠かせない。

RPMの開発元であるRed Hat社で働き始めたとき、私はすべての疑問に答えてくれるはずの噂の「RPM Guide」を手渡されるものと思っていた。しかし事態はまったく逆だった。Red Hat社の内部でさえ、RPMを巡ってさまざまな混乱が渦巻いていた。したがって、他のスタッフと同様、私も自分で1つ1つ学んでいくしかなかった。RPMをよく知っている人が周りにたくさんいるという恵まれた環境にあっても、これは時間のかかるプロセスだった。数多くの試行錯誤を必要とし、忍耐を必要とした。あの当時本書があったなら、どれほど助かっていたことか。

RPMは勉強するだけの価値がある。RPMの理解はRPMベースのLinuxシステムの管理を大いに助けてくれる。RPMの核心は、シングルシステムであろうと、何千ものマシンからなるクラスターであろうと、今日のLinuxの複雑な運用を管理するソリューションたることにあるだろう。RPMはとほうもないパワーを持っている。単一のコマンドラインで、OSの整合性を検証することもできれば、不必要なソフトウェアをまとめて削除することもできる。複雑なアクションを簡単なコマンドラインで処理するこのような能力にもかかわらず、RPMの奥は深く、学ぶことは多い。

本書の存在価値はここにある。RPMは非常に柔軟にできているが、その代償として、何をどのように実行するかを理解するのは一筋縄ではいかない。本書が例の噂の「RPM Guide」の役割をはたしてくれることを願っている。

Chip Turner
Senior Software Engineer
Red Hat Network

はじめに

Linuxのパッケージ管理では、複数のシステムを管理し、それらシステムにインストールされている各種のソフトウェアを管理しなければならない。RPM Package Management (RPM) はこうしたニーズから生まれた。

RPMが他のパッケージ管理ソリューションと異なるのは、その能力の幅広さだ。RPMを使えば、クリーンなソースから出発して、パッケージを何回でも作成できる。パッケージの内容の検証も可能であり、署名されたパッケージの身元をチェックすることができる。システムにインストールされているソフトウェアを、いろいろなやり方で調べることができる。

RPMを使えば、パッケージ管理のタスクのほとんどを自動化できる。このように「自動化」にはっきりと焦点を合わせていることも、他のパッケージ管理ソリューションにはないRPMの特長といえる。RPMはパッケージ間の依存関係も管理するから、システムにインストールされた各パッケージはスムーズに連携する。

本書はRPMの基本的な使用方法から高度なトピックまでをカバーする。RPM APIを使ったプログラミングや、Linux以外のシステムでのRPMについてもふれる。

本書が対象とする読者

本書はRed Hat Linuxやその他のLinuxシステムの管理者を対象としている。Linuxシステムの管理者はシステムにインストールされているソフトウェアの管理という日々のタスクをRPMを利用して実行できる。Linux以外のUNIXシステムの管理者の場合も、(互換性や操作性の観点から) RPMを使ってシステムを管理したいことがあるだろう。本書はこのような管理者にも適している。

ソフトウェア開発者や上級システム管理者は、RPMが備えている各種のAPIを使ってプログラムやスクリプトを書くときに本書をリファレンスとして利用できる。

ハードウェアとソフトウェアの要件

本書は最新バージョンであるRPM 4.1を使うが、説明の大半は他のバージョンにもあてはまる。プラットフォームはRed Hat Linux 8.0をターゲットとするが、他のバージョンのLinuxでも基本は変わらない。

本書の構成

本書は5つのパートからなる。各パートの内容は以下のとおり。

パートI : RPMの基礎

このパートでは、パッケージ管理ソフトウェアがなぜ必要になるかを明らかにし、背後にある考え方を始めRPMの基本を説明する。このパートの第4章からはrpmコマンドの使用法に立ち入り、パッケージのインストール、アップグレード、削除の実手順を示す。rpmコマンドはパッケージに関する情報のクエリやパッケージの検証もサポートしている。

パートII : RPMを作成する

このパートでは、RPMを作成する方法を説明する。自分で開発したアプリケーションからRPMを作成する場合もあれば、別の誰かが開発したアプリケーションをRPMとしてパッケージ化する場合もある。このパートの各章では、rpmbuildコマンドによるRPMの構築手順を定義しているspecファイルの内容を詳しく紹介し、トリガや条件オプションなどの高度なトピックにも立ち入る。第14章では、各種のLinuxシステムに対応したRPMを効率的に作成するためのヒントと推奨手法を示す。

パートIII : プログラミングRPM

RPMシステムはRPMライブラリ (rpmli**l**ib) を通じて利用できる。このパートでは、スクリプトやプログラムを通じてRPMにアクセスする方法を示す。Linuxシェルスクリプティング、Cプログラミング、Pythonスクリプティング、Perlスクリプティングについて、それぞれ章を設けて説明する。

パートIV : RPMを拡張する

このパートでは、Red Hat Linux以外のLinuxシステムや非Linux系のシステム (SolarisやWindows) でRPMを活用する方法を説明する。第21章では、RPMの設定とマクロを通じてRPM環境をカスタマイズする方法を明らかにする。

パートV : Appendix

本書のAppendixでは、追加の参考資料を提供する。各Appendixでは、rpmコマンドの構文、specファイルの内容、RPMパッケージファイルのフォーマット、開発ツールのリスト、インターネットリソースの場所を示す。

本書の表記法

本書では以下の表記法を採用している。

書体

斜体は初出の用語を示す。ユーザーがキーボードから入力するコマンドやシステムからの応答は等幅フォントで記載される。太字はユーザーがキーボードから直接入力しなければならないテキストを示す。

コード

プログラムのコードやspecファイルの内容も等幅フォントで表示される。たとえばPythonのコードは次のように表記する。

```
ts = rpm.TransactionSet()
```

本書の読み方

本書は最初から順に読むように書かれている。ただし、RPMの基本をすでに理解している場合は、最初の数章を飛ばしてもよい（必要に応じてあとで参照すればよい）。

また本文中では随所に以下のようなアイコンを使用している。これらのアイコンは特別な情報や重要な情報を示す。



起こりうる問題について注意を促す。



本文のトピックに関連して、重要な情報や技術的なデータを示す。



本文のトピックに関連して、追加情報を記載している本書の他の箇所やインターネットのサイトを示す。

追加情報

RPMに関するさらに詳しい情報が必要になった場合は、Webサイト、メーリングリスト、ニュースグループを探ってみよう。Appendixにはこれらのリソースが列挙されている。

質問はEメールを通じて筆者に直接送ってもらってもかまわない。しかし、RPMのメーリングリストに投稿したほうが、もっとすばやく回答が得られるだろう。本書に関する感想を筆者まで送っていただけるとありがたい。読者からのフィードバックは本書の今後の版に反映させたいと考えている。

本書を楽しまれることを願っている。

Eric Foster-Johnson
erc@pconline.com
2003年1月

Contents

Part1 RPMの基礎	1
Chapter1 パッケージ管理の概要	3
アプリケーションのインストール, 削除, アップグレード	3
インストールを楽にする	5
● アプリケーションレベルのユーティリティ	5
● システムに内蔵されているユーティリティ	6
Linuxのソフトウェア管理ツール: パッケージ	7
まとめ	9
Chapter2 RPMとは何か	11
Linuxパッケージ管理システムの必要性	11
RPMの設計目標	13
● 使いやすさ	14
● パッケージに焦点を合わせる	14
● パッケージをアップグレードできる	15
● パッケージの依存関係	15
● クエリを実行できる	16
● パッケージの検証	16
● 複数のアーキテクチャをサポート	17
● クリーンなソース	17
RPMの用語	18
まとめ	21
Chapter3 RPMの概要	23
パッケージファイルを理解する	23
● RPMファイルのフォーマット	23
● バイナリRPMとソースRPM	24
バイナリRPM	24
ソースRPM	25
RPMデータベースへのクエリ	26
RPMコマンドを実行する	27
● rpmコマンドの使用法	27
● その他のRPMコマンド	29
まとめ	30

Chapter4 RPMを使う	31
rpm コマンド	31
ソフトウェアのアップグレードとインストール	31
● rpm コマンドを使ってアップグレードする	32
パッケージがインストールされたかチェックする	33
インストールとアップグレード時にフィードバックを得る	33
インストール作業の詳細	38
インストールをテストする	38
一度に複数のパッケージをインストール/アップグレードする	40
別のディレクトリにインストールする	41
インストールを強行する	42
スクリプトをスキップする	43
ignore オプション	45
ドキュメント	45
● パッケージをアップグレードする	45
アップグレード時のオプション	46
スマートなアップグレード	47
● パッケージを最新化する	47
● パッケージをインストールする	48
● ネットワーク経由のインストール	48
FTPを使ってインストールする	49
HTTPを使ってインストールする	49
● ソースRPMをインストールする	50
ソフトウェアの削除	51
● パッケージが削除されたかチェックする	52
● 一度に複数のパッケージを削除する	52
● パッケージ削除のオプション	52
rpm コマンドのその他のオプション	53
まとめ	55
Chapter5 RPMデータベースを使う	57
RPMデータベースをクエリする	57
● パッケージをクエリする	58
● すべてのパッケージをクエリする	59
● クエリを工夫する	60
出力をgrepにパイプする	60
ワイルドカードを使ったクエリ	62

Contents

ファイルが属しているパッケージを見つける	62
パッケージに関する情報を取得する	64
● パッケージの説明	64
● パッケージグループ	65
● パッケージ内のファイルを一覧表示する	67
● パッケージの設定ファイルを一覧表示する	69
● パッケージのドキュメントファイルを一覧表示する	70
● パッケージ内のファイルの状態を表示する	70
● パッケージ内のスクリプトを表示する	72
● 変更を表示する	74
● クエリを組み合わせる	75
パッケージ説明とドキュメントファイルを組み合わせる	75
設定ファイルの状態を表示する	76
最近インストールしたパッケージを表示する	77
● カスタムクエリを作成する	77
クエリフォーマットのタグ	80
パッケージ情報を調べる	80
配列のフォーマット	81
特別なタグ	82
パッケージの依存関係をクエリする	83
ファイル情報をクエリする	84
クエリフォーマットのその他のタグ	85
● その他のクエリ	86
パッケージファイルに関する情報を取得する	87
インストールされたRPMパッケージを検証する	88
● システム全体を検証する	88
● 検証を制御する	89
RPM データベースのメンテナンス	90
● RPM データベースをバックアップする	91
● RPM データベースを再構築する	92
● 新しいRPM データベースを作成する	92
まとめ	93
Chapter6 パッケージの依存関係	95
依存関係の概念を理解する	95
● 機能	96
● バージョンの依存関係	97

● 競合	97
● 廃止	98
依存関係をチェックする	98
● パッケージが必要とする機能を調べる	98
● パッケージが提供している機能を知る	101
● 競合をチェックする	103
● 特定の機能を必要とするパッケージを調べる	103
● 特定の機能を提供するパッケージを調べる	107
トリガ	108
まとめ	111
Chapter7 トランザクション	113
トランザクションを理解する	113
● トランザクションはどのような場合に必要になるか	113
● トランザクションを取り消す	114
rpm コマンドによるトランザクション	114
● トランザクションID	116
トランザクションIDを確かめる	117
特定のトランザクションIDを持つパッケージを調べる	117
● トランザクションをロールバックする	118
既存のパッケージを保存する	119
まとめ	121
Chapter8 RPM管理ソフトウェア	123
RPMパッケージを見つける	123
● rpmfindとrpm2html	124
● インターネット上のRPMサイト	128
rpmfind.net	128
freshrpms.net	128
rpm.pbone.net	129
その他のサイト	129
グラフィカルなRPM管理	129
● Nautilus	130
● アプリケーションの追加 / 削除	131
● KPackage	133
● Gnome-RPM	134

Contents

RPM 管理を拡張する	136
● AutoRPM	137
● AutoUpdate	139
● Red Hat Network と up2date	140
● current	142
● urpmi と RpmDrake	142
● apt-rpm	143
● poldek	145
まとめ	146

Part2 RPM を作成する 147

Chapter9 RPM を作成する：概要	149
RPM を構築する準備	149
● 何をパッケージにするかを企画する	150
● パッケージに入れるソフトウェアを収集する	151
● 再現可能なソフトウェアの構築手順を作成する	152
ソフトウェアをアンパックする	152
README を読む	152
Linux の開発ツールを使ってソフトウェアを構築する	153
imake	154
configure スクリプト	155
Perl モジュールを構築する	155
● アップグレード方法を計画する	156
● 依存関係を判別する	156
RPM パッケージを構築する	156
● ディレクトリ構造をセットアップする	157
● ディレクトリにソースを配置する	158
● spec ファイルを作成する	159
イントロダクションのセクション	159
prep セクション	160
build セクション	160
install セクション	160
clean セクション	161
files セクション	161
● rpmbuild コマンドでRPMを構築する	162
RPM を検証する	166

まとめ	167
Chapter10 specファイルを作成する	169
specファイルを読む	169
specファイルを書く	170
● コメント	171
● specファイルをディスクに保存する	171
パッケージ情報を定義する	172
● パッケージを説明する	172
パッケージの名前を指定する	172
会社情報を設定する	174
パッケージの説明	175
プラットフォームアーキテクチャを指定する	175
● 構築に使用する場所を指定する	176
● ソースファイルを指定する	177
● パッチを指定する	178
構築手順を制御する	179
● 構築の準備をする	179
● ソフトウェアを構築する	181
● ソフトウェアをインストールする	183
● 構築のあとのクリーンアップ	184
● インストールスクリプトを定義する	184
ファイルのリストを指定する	186
● ワイルドカードを使う	186
● ファイルのディレクトリを指定する	186
● ドキュメントファイルと設定ファイルをマークする	187
● ファイル属性を設定する	189
● %filesセクションを検証する	190
● ファイルのリストを自動的に生成する	191
● 未パッケージのファイルによる構築エラーに対処する	192
変更ログのエントリを追加する	192
specファイルのマクロを定義する	193
● 内蔵マクロ	193
● specファイルに固有のマクロ	193
● 新しいマクロを定義する	194
● マクロのパラメータを指定する	195

Contents

XMLのspecファイルを作成する	196
まとめ	198
Chapter11 高度なRPMパッケージング	199
パッケージの依存関係を定義する	199
● 依存関係を指定する	200
依存関係のバージョンを指定する	200
仮想の機能を定義する	201
スクリプトエンジンとモジュールへの依存関係を指定する	201
● 前提条件の設定	202
● 構築の依存関係を指定する	202
● 依存関係を自動的に生成する	203
トリガを設定する	203
検証スクリプトを書く	207
サブパッケージを作成する	208
● サブパッケージに関する情報を記述する	209
● サブパッケージ用のスクリプトを定義する	210
● サブパッケージを構築する	211
再配置可能なパッケージを作成する	212
● prefixディレクトリを設定する	212
● filesセクションを定義する	212
● 再配置可能なパッケージの作成を巡る問題	213
● 条件付きの構築手順を定義する	214
● 条件マクロを定義する	214
● 条件ブロックを使う	215
● アーキテクチャに基づく条件	216
まとめ	217
Chapter12 rpmbuildによるパッケージの構築	219
rpmbuildコマンドを使ってパッケージを構築する	219
● 構築手順をカスタマイズする	219
● 構築手順をテストする	220
● 構築手順をデバッグする	220
● クリーンアップ	221
● 他のプラットフォーム用の構築	222

外部のspecファイルなしにRPMを構築する	224
● tarアーカイブを処理するオプション	224
● 必要なアーカイブ構造	225
ソースRPMを使う	225
● ソースRPMからバイナリRPMを再構築する	226
● ソースRPMからバイナリを再コンパイルする	229
● SRPM - ソースRPMを見つける	229
構築されたRPMに署名する	229
● GPGソフトウェアがインストールされているかチェックする	229
● 署名のための設定	230
● rpmbuildコマンドを使って署名する	232
● rpmコマンドを使って署名する	232
● 署名を検証する	232
● 公開鍵をインポートする	234
● Red Hatの公開鍵を取得する	236
まとめ	236
Chapter 13 パッケージングの補助ツール	237
パッケージングの補助ツール	237
● VIMのspecプラグインを使ってspecファイルを編集する	237
● emacsのrpm-spec-modeを利用する	241
● rpmlintを使ってspecファイルをデバッグする	244
● RUSTを使って%filesセクションを作成する	246
● setup.shとMakeRPM.pl	248
● rpm2cpioを使ってパッケージファイルを操作する	248
まとめ	254
Chapter 14 パッケージングのガイドライン	255
よくある問題を避ける	255
● メーリングリストをチェックする	255
● rpmbuildを使う	256
● システムをくつがえそうとしない	257
● 依存関係の自動的な生成をオフにする	257
● %filesにディレクトリを定義しない	258
● 依存関係の循環に対処する	258

Contents

ベストプラクティスに従う	258
● 準備	259
ソースRPMを作成する	259
クリーンなソースで始める	259
パッケージに何を入れるかを定める	259
テスト用のRPM データベースを作成する	260
● RPMを構築する	261
ツールを使う	261
rootユーザーではRPMを構築しない	262
デジタル署名を付加する	262
賢くコピーする	262
BuildRootを指定する	262
新バージョンのたびに変更ログのエントリを追加する	263
パッケージのグループを定義する	263
まとめ	263

Part3 プログラミングRPM

265

Chapter15 スクリプトを使ってRPMの管理を自動化する	267
スクリプティング	267
スクリプティング言語とプログラミング言語を区別する	268
プログラムを使う場合とスクリプトを使う場合を判別する	269
シェルスクリプティングの基本	270
● スクリプトを書く	270
● スクリプトを実行する	270
● スクリプトの実行を巡る問題	271
● スクリプトをコマンドに変換する	271
● スクリプトにコマンドラインのオプションを渡す	274
RPM ファイルをチェックする	275
RPM データベースをクエリする	277
● 同時にインストールされたすべてのパッケージをクエリする	277
● パッケージのHTMLドキュメントを読む	278
もっと先へ進むために	279
まとめ	279

Chapter16 CによるRPMプログラミング	281
Cライブラリを使ったプログラミング	282
● Cプログラミング環境をセットアップする	282
● RPMプログラミング環境をセットアップする	282
● RPMライブラリを使う	283
● RPMプログラムのコンパイルとリンク	284
インクルードファイル	284
ライブラリ	284
● RPM環境に関する情報を取得する	285
設定を出力する	286
マクロの値を展開する	288
コードの中でマクロを展開する	288
poptのパーワー	290
● popoエイリアス	290
● popoを使ったプログラミング	291
オプションテーブルを埋める	292
popoコールバック	293
オプションテーブルの特別なフラグ	294
魔法のオプション	294
コマンドラインオプションを解析する	295
コマンドラインのオプションを処理する	296
● エラー処理	297
● popoのサンプルを実行する	297
● rpmのコマンドラインオプションを処理する	301
RPMファイルを処理する	302
● RPMファイルを開く	303
● RPMのリードと署名を読む	303
● ヘッダ情報を読む	304
● ヘッダ情報へのショートカット	307
● RPMファイルを閉じる	307
RPMデータベースを扱うプログラミング	308
● データベースイテレータ	308
● 依存関係セット	310
RPMファイルをインストール済みパッケージと比較する	312
さらに先へ進むために	321
まとめ	321

Contents

Chapter17 PythonによるRPMプログラミング	323
Pythonの開発環境をセットアップする	323
● Pythonの基本パッケージをインストールする	324
● グラフィックスにPythonを使う	324
Python APIの階層	325
RPM データベースを扱うプログラミング	325
● RPMデータベースにアクセスする	326
データベースの場所を設定する	326
データベースの初期化, 再構築, 検証	327
● RPMデータベースをクエリする	327
● パッケージヘッダをチェックする	329
● rpm.hdrクラス	330
sprintfを使ってヘッダ情報を出力する	331
● 特定のパッケージをクエリする	331
● パッケージに関する情報を出力する	333
● 高度なクエリ	336
パッケージファイルを読む	338
● パッケージファイルからヘッダを読み出す	338
● 検証フラグを設定する	339
依存関係を比較する	340
パッケージのインストールとアップグレード	342
● トランザクションセットの作成	342
● トランザクション要素	343
● トランザクション要素のチェックと再配列	345
依存関係をチェックする	345
トランザクションチェックのコールバック	346
トランザクションセットの再配列	347
● トランザクションを実行する	348
トランザクション実行のコールバック	348
コールバックのサンプルコーディング	349
パッケージをアップグレードする	350
もっと先へ進むために	353
まとめ	354
Chapter18 PerlによるRPMプログラミング	355
Perl RPMモジュールを使う	356

RPM ファイルを操作する	356
● パッケージファイルを開く	356
● パッケージのタグを表示する	357
● コンビニエンスメソッド	358
● パッケージ名とバージョンを表示する	358
● ソースパッケージであるかどうか判別する	358
RPM データベースを扱うプログラミング	359
● データベースを開く	359
● パッケージを見つける	360
● パッケージを繰り返し検索する	360
● その他のクエリサブルーチン	361
● パッケージに関する情報を取得する	363
インストールの日付を表示する	365
文字列配列のタグを処理する	366
パッケージ内のファイルを一覧表示する	367
● バージョンを比較する	369
● データベースを閉じる	370
もっと先へ進むために	370
まとめ	371

Part4 RPMを拡張する

373

Chapter19 Red Hat Linux以外のシステムでRPMを使う	375
RPMのインストールに関するトラブルシューティング	375
● RPMのバージョンの相違	376
● ソフトウェアをパッケージに分割する方式の相違	376
● 依存関係の問題	377
● インストール先の相違	378
● どうしてもうまくいかないときはソースパッケージから再構築する	378
RPMの構築に関する問題	379
● ディストリビューションに固有のパッケージを作成する	379
● 依存関係の自動生成に対処する	380
● マクロの相違	381
● 再配置可能なパッケージを作成する	381
● RPM構築環境を創出する	381
ベンダを検出する	381
構築環境とマクロ	382

Contents

互換性パッケージとグルーパッケージ	384
署名の問題	385
非RPMベースのLinux ディストリビューションに対処する	385
RPMの標準化	385
● FHS	385
● RPMの採用	386
まとめ	386
Chapter20 他のOSでRPMを使う	387
Linux以外のOSでRPMを実行する	387
● RPMを入手する	388
● WindowsでRPMを実行する	389
他のOS上でRPMをブートストラップする	390
● RPMソフトウェアをダウンロードする	390
● ソフトウェアをアンパックする	390
● INSTALLファイルを読む	391
● RPMが必要とするライブラリ	391
● RPMを作成するツール	392
● RPMをコンパイルする	392
● 問題に対処する	393
RPMシステムをセットアップする	394
● RPMデータベースをセットアップする	394
空白のRPMデータベースを初期化する	394
RPMを使わずにインストールしたパッケージの依存関係を処理する	395
仮想パッケージをセットアップする	395
仮想パッケージを手動で作成する	398
● RPM環境を創出する	398
Linux以外のシステムでRPMを作成する	399
● RPMパッケージの構築環境をセットアップする	400
● パッケージをクロス構築する	400
まとめ	402
Chapter21 RPMの動作をカスタマイズする	403
RPMマクロによるカスタマイズ	403
● マクロを定義する	403
specファイルでマクロを定義する	404

マクロ初期設定ファイルでマクロを定義する	404
コマンドラインでマクロを定義する	405
● マクロをカスタマイズする	405
RPMの設定値を変更する	406
● 現在の設定を確かめる	406
● rpmrc ファイルを検出する	406
● 設定を変更する	407
● optflags を設定する	408
アーキテクチャ値を設定する	408
popt エイリアスを追加する	410
● エイリアスを定義する	410
● popo エイリアスをカスタマイズする	412
まとめ	413

Part5 Appendix

415

Appendix A RPMコマンドリファレンス	417
rpm コマンド	417
● アップグレード, 最新化, インストールのオプション	418
● 削除のオプション	419
● 署名関連のオプション	420
● 検証関連のオプション	420
● データベース関連のオプション	421
● その他のオプション	421
rpmbuild コマンド	422
● specファイルから構築する	422
● 圧縮されたtarアーカイブから構築する	422
● ソースRPMからRPMを再構築する	423
● 構築手順をカスタマイズする	423
Appendix B specファイルの構文	425
パッケージ情報ディレクティブ	425
● コメント	426
● 構築環境の設定	426
● 依存関係のディレクティブ	427
● ソースファイル	428

Contents

マクロ	428
● 変数定義マクロ	428
● 条件マクロ	429
● 内蔵マクロ	430
構築セクション	431
● 構築のための準備をする	431
● ソフトウェアのコンパイル	431
● インストール	432
● クリーンアップ	432
● インストールスクリプトとアンインストールスクリプト	432
ファイルセクション	433
● 再配置可能なパッケージを作成する	433
変更ログ	434
Appendix C RPMの進化	435
Appendix D RPMパッケージファイルの構造	441
パッケージファイル	441
● ファイル識別子	441
● 署名	442
● ヘッダ	442
ヘッダタグ	443
プライベートなヘッダタグ	444
署名タグ	444
インストールタグ	445
ファイル情報タグ	445
依存関係のタグ	446
● ペイロード	448
Appendix E RPMリソース	449
RPM関連のサイトを見つける	449
● メインのrpm.orgサイト	449
● RPMパッケージを見つけるためのサイト	450
● RPMツールのサイト	450
● プログラミング関連のサイト	451
● RPMの標準化に関するサイト	451
● RPMのライセンスに関するサイト	451

RPMのメーリングリストとニュースグループにアクセスする	452
Appendix F Linuxのテキストエディタと開発ツール	453
汎用のテキストエディタ	453
プログラミング用のテキストエディタ	454
Cプログラミング用の統合開発環境	454
Pythonプログラミング用の統合開発環境	455
INDEX	456

