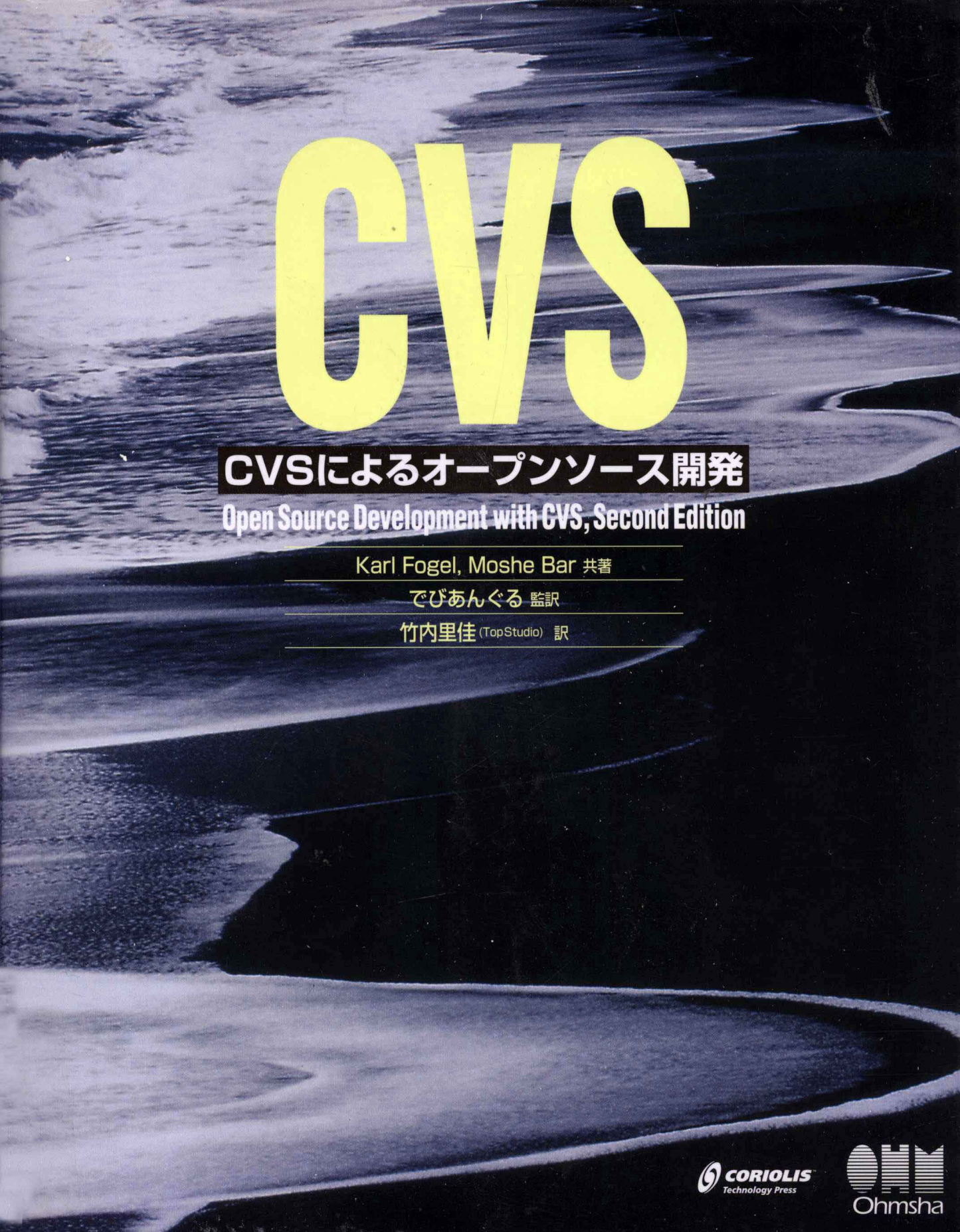
CVS
CVSによるオープンソース開発
Open Source Development with CVS, Second Edition
Karl Fogel, Moshe Bar 共著
でびあんぐる 監訳
竹内里佳 (TopStudio) 訳
CORIOLIS Technology Press
OHM Ohmsha

CVS

CVSによるオープンソース開発

Open Source Development with CVS, Second Edition

Karl Fogel, Moshe Bar 共著
でびあんぐ 監訳
竹内里佳 (TopStudio) 訳

Ohmsha

■監訳者
でびあんぐる
Debian Project（フリーのOSであるDebianをメンテナンスするボランティア団体）に所属する日本人オフィシャルメンバで構成される任意団体。詳細は監訳者紹介を参照。

■訳　者
竹内里佳（たけうち　りか）
（有）トップスタジオ所属の翻訳者。主にコンピュータ書籍の翻訳・編集・監修を手がける。プログラミング、セキュリティ、UNIX、最近ではWindows 2000/XP関連など、幅広い分野の翻訳を行っている。『Linuxプログラマーズ事典』（翔泳社刊）、『JavaプログラミングBlack Book』（インプレス刊）など、訳書多数。

CVSによるオープンソース開発

平　成 14 年 6 月 25 日　第1版第1刷発行

検印省略

著　者　Karl Fogel・Moshe Bar
監訳者　でびあんぐる
訳　者　竹内里佳
企画編集　オーム社開発局
発行者　佐藤政次
発行所　株式会社 オーム社
郵便番号　101-8460
東京都千代田区神田錦町3-1
振　替　00160-8-20018
電　話　03(3233)0641(代表)
http://www.ohmsha.co.jp/

Printed in Japan　組版　トップスタジオ　印刷・製本　エヌ・ピー・エス
落丁・乱丁本はお取替えいたします

ISBN 4-274-06473-5

愛する両親フランシスとヘンリー、彼らが与えてくれた文字どおりすべてのことに感謝して、本書をささげる

— *Karl Fogel*

❦

イスラエル — 約束の地、その人々、そしてユダヤの教えに

— *Moshe Bar*

❦

著者紹介

Karl Fogel

1971年に生まれ、パーソナルコンピュータとBBSが流行した1980年代は、コンピュータやネットワーク、電子メールなどにはいっさい触れることなく過ごした。技術的なことに関してはまったく無知な状態で(これは、彼が再び取り戻したがっている状態です)、1991年、彼はピアノを学ぶためにOberlin音楽学校に入学したが、なぜか中国語の学位を取得し、何を間違ったかコンピュータプログラミングを学んだ。

1995年にはJim Blandyとともに Cyclic Softwareを創設し、CVSの保守と商用サポートを始めた。Cyclic Softwareを譲った後は、中国南西部に赴いて英語とUNIX/Cプログラミングを1年間教えた。現在彼はシカゴに住み、CollabNetでSubversionプロジェクトのフリーソフトウェアプログラマとして働いている。Subversionは、CVSの後継として作られている新しいバージョン制御システムである。

貴重な余暇にはコンピュータとの接触を避け、人間とのコミュニケーションやピアノを楽しんでいる。

Moshe Bar

科学修士およびコンピュータサイエンス博士。欧州と南米の大学でコンピュータサイエンスを教えている。数年に渡り、Linuxカーネルや JFSファイルシステム、LinuxのNFS、Mosixクラスタリングカーネル拡張機能など、いくつかのオープンソースプロジェクトに貢献してきた。Linuxカーネルやそのファイルシステム、サイエンティフィッククラスタリングについての著書がある。現在は、世界中の企業のコンサルタントとして活躍している。

プログラミング以外にも、Byte Magazine誌(http://www.byte.com)やその他いくつかのコンピュータジャーナルのシニアエディタとしても活動中。また、余暇にはカスタム仕様のオートバイを楽しんでいる。現在のお気に入りは新車のYAMAHA Dragstar Classic 1100。

謝 辞

謝辞がなぜいつも長くなるかは、自分で本を書いてみないとわからないかもしれません。著者が全員に感謝を述べ終わる頃には、その書籍の本文のための場所が残っているのかどうか不安になるほどです。

しかし、今の私にはその理由がわかります。

本書を執筆する6ヵ月の間、私はほとんど瀕死状態であったにもかかわらず、Ben Collins Sussmanはここ13年間と同じく変わらずに友人でいてくれました。また、彼は多くの章を下書きの段階で読んでくれました。彼の賢明な批評のおかげで本書の内容を向上させることができました。Jim Blandyも同様に寛大な友人です。私も、彼が私をCVSの世界に引き込んだことに対して寛大にならなければなりません。私がどんなことに巻き込まれるかなど、Jimには知る由もなかったのですから。また、私たちが2人とも締め切りに迫られているときに、電話でCVSの「watch」機能の仕組みを辛抱強く説明してくれたことにもとても感謝しています。Noel Craggはピンチのときに辛抱強い友情と精神的な援助で支えてくれました。どうもありがとう！

Brian Fitzpatrickは友人でありマネージャでもあります。彼は忙しい仕事の時間を割き、各章の下書きを読み、有意義な意見を数多く出してくれました。彼にはいくら感謝しても足りません。また私たちの共通の雇用主であるonShore, Inc.（別名「The Mother Ship」）は、私の無謀なほどの気まぐれなスケジュールにも辛抱強く付き合ってくれました（StelとZa、Eric、Adam、どうもありがとう！）。あまりにもお世話になったので、私にはとても恩返ししきれません。友人であり同僚であるLeftyは、おそらく私が仕事を休んでいる間矢面に立っていたはずです。とても感謝しています。月曜日までに後処理をすべて終えることを約束します。

Karenのアドバイスと友情、励まし、下書きを読んでくれたこと、そして彼女の存在に感謝します。Matt Braithwaiteの協力と、BSD頭字語についての質問に答えてくれたことに感謝します。Chris Larkoshは、「並行観念形成」（parallel ideation）というすばらしく威圧的な語句を作ってくれました。KennisとRachelの初期の激励は私にとってとても大きな意味がありました。Siu Yuinの友情と強力なアイデア（特に、何かをやり遂げる際の利益という動機について）にも感謝します。Noel Taylor、特にGolosaにはすべてのことに感謝しています。Yong Qingは、時間がないと思えるようなときでも、私に楽しむことを思い出させてくれました。

Richard Stallmanの見通しと妥協することのない意志は、現在のフリーソフトウェア運動の成功の多くを導きました。彼は大きな注目を集めていますが、実際にはその倍以上の賞賛を得る資格があります。RMS、ありがとう。いつまでも変わらないでください。

Coriolisの幹部のみなさまが、本書のかなりの部分をGNU General Public License下でリリースするという危険な決断をしてくれたことに感謝します。彼らにこの決断を後悔さ

せずにすむことを祈ります。原稿取得担当エディタのJawahara Saidullah、原稿調整担当エディタのJessica Choi、プロジェクトエディタのMeredith Brittain、製作コーディネータのMarcos Uzueta、レイアウトデザイナのApril Nielsen、カバーデザイナのLaura Wellander、プロダクトマーケティングマネージャのTracy Rooneyに感謝します。彼らの助けなしでは、本書は完成し得ませんでした。

最後に、私の両親に最大の感謝を捧げます。彼らの力は大きく、最初から支持と優れたアドバイスで支えてくれました。そして、家や中華レストランでの和やかな食事を通して、私が本当に必要としているときに休息を与えてくれました。

——Karl Fogel

本書はチームワークの賜物です。私を採用してくれたKevin Weeksにはもちろんのこと、この第2版の刊行を実現するのを手伝ってくれたCoriolisの編集者たち、Meredith Brittain、Jawahara Saidullah、Jessica Choiに感謝します。彼らをはじめ、本書の製作に関わったCoriolisのスタッフは、アイデアを形にして出版する方法を熟知しているすばらしいチームです。

どの本を書くときもそうですが、本書を書くためには、多くの付き合いや家族の生活を犠牲にしなければなりませんでした。本書を優先して多くの週末や夜をつぶしましたが、Avivitは辛抱強く付き合ってくれました。ありがとう。

最後に、日々の開発とシステム管理者としての仕事の中で、CVSの使い方を私に学ばせてくれた人々にも感謝します。SAP PortalsやBaan Developmentの人々、そして学習の場となったすばらしいオープンソースの世界(学ぶべきことは今でも尽きません)の、貢献とオープンソースプロジェクト管理の活動力に感謝します。

私は非常に社交的な時間を過ごしているように思います。また、オープンソースの参加者であることを嬉しく思います。私を含め、多くのプログラマは、Richard StallmanとEric Raymondの根気強い努力に感謝しなければなりません。Linus TorvaldsやMiguel de Icaza、Eric Allman、Barak教授、Jordan Hubbard、その他多くのスタープログラマたちに感謝します。大きな星の陰になってあまり目立つことのない、その他の多くの偉大なるオープンソースプログラマたちのことも忘れることはできません。Marshall McKusickやAmnon ShilohやIngo Molnar、ここではとても全員を挙げることはできませんが、みなさんに感謝します。ありがとう。

——Moshe Bar

日本語版監訳者より

本書『CVSによるオープンソース開発』は、私たちでびあんぐるが監訳者として活動を始めた最初の書籍『CVS －バージョン管理システム－』の改訂版です。

原書の書名は前回もそして今回も一貫して『Open Source Development with CVS』でしたが、日本ではまだ「オープンソース」をうたうには時期尚早ではないかという意見から、前回は訳書の名前を別のものに変えていました。

初版のこのページで私たちは次のように述べました。

今、「オープンソース」は一時の流行語ではなく、文化として根付きつつあります。

あれから2年の月日が経ち、オープンソースは完全に定着したと言えるでしょう。コンピュータ業界でオープンソースという言葉を聞かない日はありません。各メディアはオープンソース陣営とMicrosoftの対立を盛んに煽り、政治家たちでさえも(幾分の誤解をはらみながら)オープンソースの手法と可能性に大きな関心を寄せています。

時は来たれり！ 今こそ、原書の正しき名前を名乗るときです。私たちの活動は正しかったことが証明されました。

本書はオープンソース開発の必須ツールとも言えるリビジョン制御ソフトウェア「CVS」の説明を主たる目的としていますが、同時に、オープンソースという文化を知り、その中に参加し、成功に導いていくためのガイドの役目も持っています。

原書の改訂に合わせ、本書でも各章の順番が初版から変わりました。また、CVSの最新バージョンに基づいた修正などを監訳の段階で行っています。これらの変更で、CVSとオープンソースそれぞれについての読者の理解がより容易になることを期待しています。

2002年6月

武藤 健志（でびあんぐる）

でびあんぐる、および監訳者個々人については巻末の「監訳者紹介」をご覧ください。

本書の監訳による収益は、Debian JP Project(`http://www.debian.org/`)をはじめとする各オープンソースプロジェクトに寄付する予定です。

はじめに

オープンソースソフトウェアを使わずに1日が終わることはまずありません。ときには、そうとは知らずに使っていることもあります。恋人や友人、同僚から電子メールを受け取るときには、約80%の割合で、標準的なオープンソースソフトウェアであるSendmailが使われています。

インターネット上で見ているWebページも、65%はオープンソースWebサーバーを使って公開されています。オープンソースアプリケーションそのものも、すべてではありませんが、大部分はEmacs（優れたユーザー環境およびプログラムエディタ）、公式GNU Cコンパイラgccなどのオープンソースツールの力を借りて作成されており、GNUデバッガのgdbを使ってデバッグされています。何よりも、このようなアプリケーションやその他多くのアプリケーションのソースコードを保存したり、変わり行くバージョンを管理するために、ある1つのユーティリティが使われています。それがCVSです。

つまり、オープンソースソフトウェアは、ここで述べたような市場や一部の分野で大きな役割を果たし、広く普及しています。CVSはオープンソースの発展の土台であり、開発者たちやエンドユーザーのリポジトリとして機能しています。オープンソースの世界では、エンドユーザーが品質保証をし、バグ修正をコミュニティに返すことがよくあるので、これらのユーザーと開発者との区別がないこともあります。したがって、CVSのようなソースコードのリポジトリおよびバージョン制御システムは、非常に柔軟なツールでなければならず、オープンソースコミュニティ全体に対して信頼性のある安定したフロントエンドを提供しなければなりません。

本書の目的は2つあります。ひとつは文化的なもので、もうひとつは技術的なものです。文化的な目的とは、このオープンソースの文化をある程度文書化し、オープンソースプロジェクトを管理する人、またはこれに参加する人に実用的なアドバイスをすることです。技術的な目的とは、オープンソースプロジェクトでの使用を想定して、CVSの効果的な使い方を紹介することです。

1つ目の目的に関しては、「アドバイス」という言葉を強調しておかなければなりません。実際、オープンソースという現象については、Richard StallmanやEric Raymondでさえ、独断的なことを述べたり書いたりすることはできません。オープンソースというのは非常に広大な分野であり、経済的、文化的、社会的、政治科学的にあまりにも多くの側面に影響を及ぼすため、1人の人間が完全に把握することは困難です。本書の著者たちもそれは同じことです。

CVSに関しては、本書ではオープンソースプロジェクトを想定して学習を進めていきますが、これ以外のどのような状況でも使うことができるだけの十分な知識を身につけることができます。CVSは、プログラムのソースコードを管理するだけではなく、Webサイトやテキスト文書、設定ファイルのバージョン管理にも役立ちます。

本書では、読者にはプログラミングとオンライン文書の扱い方についての知識があるものと想定しますが、CVSについては何も知らなくてもかまいません。CVSの例はUNIX環

境のものを示すので、UNIXとshシェル(bashシェル)についてはある程度の知識があると役に立ちます。

第2版を刊行した理由

本というものは、前の版がよく売れると、第2版やそれ以降の版が出るものです。再版されるというのは、何よりも明らかな成功の証です。

『CVS －バージョン管理システム－』[監注1]は、間違いなく大きな成功を収めた本です。本書の第2版を執筆するにあたって課題となったのは、本書の成功要因を壊さないことと、それと同時に新しい開発環境に合わせて内容を拡張することでした。

第1版が刊行されてから、オープンソースの世界はかなり変わりました。CVSそのものやその使い方よりも、オープンソースの世界のほうが大きく変わっています。

Linuxが人気を増し、ナスダックでオープンソースが「人気」、「魅力的」になると、オープンソースは急速に成長しました。VA Linux(現VA Software)、LinuxCare、Red Hat、その他数千もの企業がオープンソースと優秀なプログラミングマニアたちを採用しました。突如として、すべてのソフトウェアと仕様をリリースしてコミュニティに戻すというやり方が、投資家たちにとっても納得のいく(もしくは望ましい)ものとなりました。投資家たちは、ソフトウェアを売ることではなく、深い知識に対する付加価値から収入を得るようになりました。

オープンソースの人気が非常に高まったため、銀行や保険代理店、政府官庁などの大規模なITユーザーは、IT部門で「オープンソース戦略」を採用するようになりました。本書の著者の一人であるMoshe Barは、このような数多くの企業や代理店の「オープンソースコンサルタント」をしています。

どのようなソフトウェア会社も、何らかのオープンソース戦略を採用せずにやっていくことはできなくなりました。しかし、オープンソースでソフトウェアを配布することを大々的に告知しておきながら、この約束を守らなかった会社もあります。その後、2001年初期にニューエコノミーバブルがはじけると、オープンソースは突如として「不人気」に戻りました。投資家たちはとうとうハイテク企業に利益を分配するように求めました。このため、Webサイトではサービスに対して使用料を課し、サービス会社はより高い金額を請求し、ソフトウェア会社は再びソフトウェアを販売するようになりました。少なくとも、そのように努力するようになりました。

本書を執筆しているのは2001年ですが、オープンソースは依然として盛んです。バイナリのみのライセンスの下で独占環境を再生しようという動きは失敗に終わりました。オープンソースは生き続けています。したがって、オープンソースと、その土台であるCVSについて理解しておくことは意味のあることです。

【監注1】 「日本語版監訳者より」を参照。

何が変わったのか

第1版では、純粋にCVSについて説明する章と、オープンソースと開発を主題とした章とを交互に交えた構成になっていました。第2版では、読者が不必要に混乱することを避けるため、方針を変えてこの2つの主題を切り離しました。

したがって、本書ではまずCVSシステムについてすべてのことを説明し(第1章～第7章)、その後でオープンソースについて説明します(第8章～第9章)。

CVSについての章では、いくつものタイムゾーンにわたって数多くの開発者が参加する大きなプロジェクトでのCVSの複雑な仕組みについても説明します。また、プロフェッショナルな環境でのCVS管理についても深く説明し、調整やバックアップ、記憶域、クラスタ化といった機能について説明します。

オープンソースの章には、業界の変化を反映してあります。オープンソースにおいて特別に困難だった挑戦(Mozillaブラウザプロジェクトなど)や、オープンソースにおける失敗から学んだ教訓を紹介します。

用語について

現在、フリーソフトウェアの「フリー」は、ソースを自由に修正し再配布できることを示します。フリーソフトウェアが成功を収める鍵となったのは、このような自由(フリー)です。ソフトウェアの価格が無料(フリー)、または安価だったからではありません。

「オープンソース」と「フリーソフトウェア」のどちらの用語が正しいのでしょうか。初期の提案者の一人であるRichard Stallmanは、フリーソフトウェアというのが適切な用語だと主張しています(ここで言うフリーとは、無料という意味ではなく自由という意味です)。この用語については長い間討論が続いており、これからも完全に決着がつくことはないでしょう。基本的にこの2つの用語は同義であり、本書では両者を同じ意味で使います。Richard Stallmanは、エッセイ『Why 'Free Software' is better than 'Open Source'』(「フリーソフトウェア」が「オープンソース」より好ましい理由)の中で、両者を同義で使うことができない状況について優れた見解を示しています。この文書はhttp://www.gnu.org/philosophy/free-software-for-freedom.htmlで見ることができます【監注2】。フリーソフトウェアという用語は、GCCやEmacs、makeなど、その他GNUプロジェクトのソフトウェアに対して使われることが多くなっています。しかし、成長を続けるLinuxの世界では、フリーソフトウェアの定義に合うソフトウェアは、オープンソースまたはOpenSourceと呼ばれるようになっています。一般分野および経済分野の報道では、GNUソフトウェアに対しても、オープンソースという用語しか使われていません。

【監注2】 http://www1.neweb.ne.jp/wa/yamadas/column/technique/fsffj.html に日本語訳(yomoyomo訳)があります。

本書の規則

本書全体を通して、説明本文の中にコマンドラインの例が出てきます。第1のサンプルユーザーの名前はjrandom【監注3】です。彼女はyarkon.moelabs.comというコンピュータ上で作業しているので、コマンドプロンプトは次のようになります。

```
yarkon$
```

出力がある場合は、プロンプトのすぐ後に太字フォントで示します。

```
yarkon$ whoami
jrandom
yarkon$
```

場合によっては、コマンドそのものがとても長く、標準UNIX端末で2、3行に渡る場合があります。その場合、各行末のバックスラッシュ(\)は、次の行が続きであることを表します。続きの行は、読みやすいようにプロンプトの分だけインデントして示します。次に例を示します。

```
yarkon$ cvs diff -c -r prerelease-beta-2_09-19990315 -r \
        postrelease-3_0-19990325 fudgewinkle.c
```

(このコマンドの意味は、本書を読み終えるまでにわかるようになります。)

また、画面では1行に表示されるコマンドやリストを紙面の都合によって折り返した箇所には、➥記号を入れてあります。

ほかのコンピュータから実行するコマンドを紹介することもあります(2人の別々の開発者が同時に作業する例を示す場合など)。その場合、もう1人のサンプルユーザーとしてmoshebを使います。彼はpasteという名前のコンピュータ上で作業しています【監注4】。

```
paste$ whoami
mosheb
paste$
```

特に明記しない限り、すべてのコマンドは、UNIX (Bourne Shell) 環境で実行しています。UNIXについて基本的な知識があれば、本書で特に困ることはないはずです。ただし、lsコマンドは、次のように奇妙な出力を表示することがあります。

```
floss$ ls
foo.txt    bar.c    myproj/
```

【監注3】 jrandomはJ.Random、ハッカー用語で「任意の」というくらいの意味合いです。

【監注4】 paste…のり、歯みがきペーストのこと。

「myproj/」の最後のスラッシュ(/)は名前の一部ではなく、myprojがディレクトリであることを表します。スラッシュが表示されるのは、jrandomの環境ではlsコマンドがls -CFの別名(エイリアス)として設定されているからです。これは、列形式でファイルをその種類を示す記号とともに表示することを示します(ディレクトリはスラッシュ、実行可能ファイルはアスタリスク(*)、シンボリックリンクは@記号(@)など)。

出力を読む際には、ファイルとディレクトリを区別できるととても便利なので、本書の多くの例ではこの形式を使用します。本書ではlsコマンドに-CFオプションを渡していませんが、エイリアスによってこのオプションが指定され、それによる出力が表示されていると考えてください。

指導内容の実践

本書の中のCVS特有の章(第2、3、4、5、10、11章)は、GNU General Public License(GPL)によって保護されています。これらの章【監注5】はhttp://cvsbook.red-bean.com/で表示およびダウンロードすることができます。オンラインバージョン、またはツリーウェアバージョンでバグを見つけた場合は、bug-cvsbook@red-bean.comに報告してください。

【監注5】　すべて英文です。

目 次

著者紹介 v

謝 辞 vi

日本語版監訳者より ix

はじめに xvii

第1章 オープンソース開発とCVSの関係 1

1.1 フリーソフトウェアとは何か 1
1.1.1 オープンソースソフトウェア 2
1.1.2 オープンソースライセンス 3
1.1.3 オープンソースビジネスモデル 4

1.2 すべての始まり 5
1.2.1 Stallmanの考え 5
1.2.2 2種類の開発 7

1.3 CVSとの関係 8
1.3.1 diffとpatch 9
1.3.2 RCS 10
1.3.3 勝者：CVS 10

1.4 オープンソース開発の原則とそれに役立つCVSの機能 11

1.5 フリーソフトウェアの原動力 13
1.5.1 必要性 14
1.5.2 仲間 14
1.5.3 名誉 14
1.5.4 お金 15
1.5.5 派閥は強さの表れ 17

第2章 CVSの概要 19

2.1 CVSの基本 19
2.1.1 CVSは何でないか：ロック－修正－ロック解除モデル 20
2.1.2 CVSとは何か：コピー－修正－マージモデル 20

2.2 その他のバージョン制御システム 24
2.2.1 Microsoft VSS 24
2.2.2 RCSとGNU/RCS 25
2.2.3 SCCS 26
2.3 CVSでの1日の作業 27
2.3.1 CVSを起動する 29
2.3.2 リポジトリへのアクセスと作業環境 29
2.3.3 新規プロジェクトを開始する 34
2.3.4 作業用コピーをチェックアウトする 36
2.3.5 変更を加える 39
2.3.6 自分(および他人)の作業内容を調べる：updateとdiff 40
2.3.7 CVSと暗黙的な引数 45
2.3.8 コミットする 48
2.3.9 誰が何をやったか調べる(ログメッセージを表示する) 56
2.3.10 変更内容を調べて元に戻す 59
2.4 その他の便利なCVSコマンド 64
2.4.1 ファイルを追加する 64
2.4.2 ディレクトリを追加する 65
2.4.3 ファイルを削除する 65
2.4.4 ディレクトリを削除する 67
2.4.5 ファイルとディレクトリの名前を変更する 68
2.4.6 オプションの入力疲れを防ぐ 69
2.5 スナップショットを入手する(日付とタグ) 70
2.5.1 使用できる日付形式 74
2.5.2 特定の時点にマーク(タグ)を付ける 74
2.6 ブランチ 81
2.6.1 ブランチからトランクへ変更をマージする 87
2.6.2 複数のマージ 90
2.6.3 作業用コピーなしでタグまたはブランチを作成する 92

第3章 CVSリポジトリ管理 95

3.1 管理者の役割 95
3.2 CVSの入手とインストール 96
3.2.1 UNIX用のCVSを入手してビルドする 96
3.2.2 Windows用のCVSを入手してインストールする 99

3.2.3 Macintosh 用の CVS を入手してインストールする 100
3.2.4 Windows 版と Macintosh 版の制限 101
3.3 CVS ディストリビューションの中身 101
3.3.1 情報ファイル 101
3.3.2 サブディレクトリ 103
3.3.3 その他の情報源 106
3.4 リポジトリの開始 107
3.4.1 パスワード認証サーバー 109
3.4.2 リポジトリ構造の詳細 115
3.4.3 RCS 形式では @ 記号が常にクォートされる 121
3.4.4 ファイルを削除すると何が起こるか 123
3.4.5 CVSROOT/ 管理用ディレクトリ 125
3.5 その他の情報 137

第4章 CVS の高度な機能 139

4.1 基本事項を超えて 139
4.2 電話代わり 139
4.2.1 「watch」：誰がいつ何をするかを知る 140
4.2.2 ログメッセージとコミットメール 155
4.2.3 作業用コピーを削除する 157
4.3 プロジェクトヒストリの鳥瞰図 159
4.3.1 望遠鏡を使った鳥瞰図：annotate コマンド 162
4.4 キーワード展開の使用 167
4.5 ブランチの使いこなし方 169
4.5.1 トランクに何度もマージする 170
4.5.2 蟻継ぎ方式：両方向にマージする 178
4.5.3 トビウオ方式：簡単な方法 179
4.5.4 サードパーティソースを追跡する：ベンダーブランチ 182
4.6 CVS の新機能 187
4.7 専門家の領域へようこそ 188

第5章 ヒントとトラブルシューティング 189

5.1 問題が起きたら 189

5.2 一般的な原因 190

5.2.1 作業用コピー管理領域 190

5.2.2 リポジトリアクセス権 193

5.3 一般的な問題とその解決法 194

5.3.1 実際に起きた問題と解決法 195

5.3.2 変わり行く状況 206

第6章 ビルド、テスト、リリース 207

6.1 なぜリリースするのか 207

6.2 リリース作業の開始 209

6.2.1 「コードの詰め込み」効果を避ける 209

6.2.2 凍結 210

6.2.3 開発と安定したブランチ 211

6.3 テスト 213

6.3.1 テスターの採用と維持 213

6.3.2 テストの自動化 214

6.4 ビルド、インストール、パッケージ化 215

6.4.1 ビルドとインストール：makeとautoconf 215

6.4.2 CVSにパッケージ化を手伝わせる 219

6.5 リリース 222

6.5.1 変更内容を公開する 222

6.5.2 リリースをCVSに記録する：タグとリビジョン番号 222

6.6 その他の情報 224

第7章 CVSを使ったシステム管理 225

7.1 システム管理者の主な仕事 225

7.1.1 CVSの正しい使い方：バージョン管理 227

7.1.2 依存ファイル 228

7.1.3 namedツリー 229

7.1.4 後退テスト 229