

# リビジョン管理ツール —RCS



宮地元志

HDL ソース・コードをはじめとするテキスト・ファイルのリビジョン(版数)管理を支援するユーティリティ・ソフトウェア“RCS”を取り上げる。まちがえて古いファイルを編集してしまうといったミスや、必要なファイルがなくなってしまったというトラブルをなくすることができる便利なツールである。本稿では、RCSを使うことで、享受できるメリットと実際の使い方について解説する。RCSは、だれもが無償で使用可能なフリー・ソフトウェアである。Linux などには標準で付属している。 (編集部)

## はじめに

デジタルLSIの設計環境が回路図ベースから、機能記述言語によるテキスト・ベースに移行してきました。

従来、回路図などの設計データは、アプリケーション・ソフトウェア(設計ツール)に固有のフォーマット(バイナリ・ファイル)で保存していました。しかし、設計資産の再利用などを考えると、よりオープンに、アプリケーションに左右されない形であることが望まれます。

たとえば、仕様書やドキュメントも、ワープロや表計算ソフトなどの固有フォーマットで保存する方法から、SGMLやHTMLなどのマークアップ言語、つまりプレーンなテキストでの保存に移行しているところも見かけるようになってきました。こうすれば、コンピュータのプラットフォームの違い(Windows, Macintosh, UNIX など)に悩むこともなくなります。

設計の記述コードはもとより、設計仕様書、報告書、さらには各種の設定ファイルや個人的なプログラム、日誌、メモ書きにいたるまで使われているテキスト・ファイルをどう扱うかは、だれもが直面する重要なテーマの一つです。

## RCSとは

日常的に変更を加え続ける必要のあるテキスト・ファイルは、RCS(Revision Control System)やCVS(Concurrent Versions System)に代表されるリビジョン(版数)管理システムを導入して扱うことをお勧めします。

RCSは、個人的な導入に適しているといわれています。CVSはRCSのプログラム群をフロントエンドにしたリビジョン管理システムであり、比較的大規模なプロジェクトを大人数で進めるのに向いているといわれています。本稿では、筆

者が実際に活用しているRCSについて説明します。

RCSとは、Revision Control Systemという名前のとおり、ファイルのリビジョンを管理するプログラムの集まりです。複数のファイルのリビジョンを、一括で管理することができます。フリーで公開されているので、だれでも自由に利用できます。

RCSのイメージがわからないという方のために、例をあげて説明しましょう。たいせつな物を管理している場所、そうですね、たとえば銀行などを思い浮かべてください。プロジェクト専用の「仕様書/ソース・コード銀行」ができたとします。まず、みんなでもち寄ったファイルをいったんすべて預けます。内容を修正したいときには、必要なファイルを銀行から引き出します。ファイルが引き出されている間は、そのファイルをほかの人が引き出すことはできません。修正が終わったファイルは、再び預けます。このとき、どこが変わったかの記録が残ります。また、その修正に固有の改版番号が自動で付けられます。通帳記帳(履歴管理)ももちろん行ってくれます(利子は付きませんが…)

## インストール

### ●UNIX

UNIX版のRCSは、LinuxやFree-BSDなどには、はじめから付属しています。もし未導入であれば、ftpのサイトからも入手可能です。

現在のバージョンは5.7ですから、「rcs-5.7.tar.gz」で検索してください。本稿執筆時に筆者がgoo(http://www.goo.ne.jp/)で検索してみたところ、203件ほどヒットしました。

ダウンロード後、/tmpなどの適当な場所にファイルを移動して、展開します。図1の手順でインストール完了です。不明な点はINSTALL、READMEなどの各ファイルを参考にしてください。

● Windows NT

Windows NT 版のRCSは、たとえば、  
<http://www.vector.co.jp/soft/winnt/prog/se015009.html>  
 で見つけることができます。実行ファイル形式のものが、  
 gr564bnt.zip のファイル名でパッキングしてあるようです。  
 筆者は、UNIX ユーザなので実際に試してはいないのですが、  
 説明を見る限り、zip 解凍後、インストーラの指示にしたが  
 えば使えるようになるようです。

ファイル名やURL、バージョンは変更されている可能性がある  
 があるのでご注意ください。

リビジョン管理のしくみ

● テキスト・ファイル管理の問題

テキスト・ファイルを修正して保存すると、あたりまえで  
 すが古いファイルは新しいファイルで上書きされます。古い  
 ファイルを残しておきたいなら、ファイルを修正した後、  
 つねに新しい名前をつけて保存しなければなりません。

なかには、ソース・コード用のディレクトリを、バージ  
 ョンごとに管理するような構成にしている方もいるかもしれ  
 ません。たとえば、図2のような感じです。このような管理  
 も、一つの方法ではあると思います。しかし、たとえば現在  
 どのバージョンのファイルを編集しているのかは、ファイル  
 名を見るだけではわかりづらいのではないのでしょうか。こ  
 のような状況下では、修正時に誤ったバージョンのファイル  
 を上書きしてしまうなど、思わぬトラブルが起きる可能性が  
 あります。

● RCS による管理

RCS を導入すると、一つのテキスト・ファイルはRCS ファ  
 イルとワーキング・ファイルの二つに姿を変えます。

▶ RCS ファイル

前者のRCSファイルは、変更の履歴を含むファイル本体で  
 す。/RCS などのディレクトリ下に存在し、そのファイルが作  
 られてから消されるまでの履歴をもちます。このしくみによ  
 り、過去の任意の時点のファイルの姿を復元することができ  
 ます。また、保存時にどこを修正したかコメントを入れてお

```

$ tar xzvf rcs-5.7.tar.gz
rcs-5.7/
rcs-5.7/man/
rcs-5.7/man/ci.1
rcs-5.7/man/co.1

<略>

rcs-5.7/REFS
$ cd rcs-5.7
$ sh ./configure
creating cache ./config.cache
checking diff basename... diff
checking for diff... /usr/bin/diff
checking diff success status... 0

<略>

creating src/Makefile
$ make
cd man && make all
make[1]: Entering directory `/tmp/rcs-5.7/man'

<略>

make[1]: Leaving directory `/tmp/rcs-5.7/src'
$ su -

Password: typepass
# make install

<略>
    
```

【図1】 RCSのインストール手順

けば、それはRCSファイルに保存されます。後から、そのと  
 きの修正点や修正理由を明らかにすることができます。これ  
 らの内容について、ユーザは、通常、意識せずすみすみます。

▶ ワーキング・ファイル

後者のワーキング・ファイルは、そのファイルのカレント・  
 イメージです。ワーキング・ファイルは、RCS 導入前のオリ  
 ジナルのファイルとほぼ同じものと考えてよいでしょう。異  
 なる点は、ワーキング・ファイルには、現在のそのファイルの  
 状態が、図3のようなマーカとして入っていることです(もち  
 ろん入れないことも可能)。いろいろなファイル中で見かけた  
 ことがある方も多いと思います。リビジョン番号や取り出し  
 時間、修正者などがここにマーキングされています。

RCS を導入すれば、ワーキング・ファイルにはRev.2.34 な  
 どの番号が自動で付きます。2 をリリース番号、34 をパー  
 ジョン番号と呼ぶことが多いようです。

ワーキング・ファイルはその名のとおり、自由に修正がで

```

$ ls
version01/  version03.1/  version03_20000314/
version02/  version03.bak/ version03_new/
version03/  version03.org/
$
    
```

【図2】 ディレクトリによるバージョン管理

バージョンごとにディレクトリを作成して管理しているようす。  
 編集中のバージョン(ここではversion 3)は、さまざまな作業  
 用ディレクトリを作成している。

```

--- $Id: dwm_rcs.html,v 1.1 2000/04/03 04:43:05 miyaj_i Exp miyaj_i $ ---
RCSファイル名 | 登録した日時 | 状態 |
リビジョン番号 | 登録したユーザ | ロックしたユーザ
    
```

【図3】 マーカ