マルチユーザに対応したバージョン管理機能を持つファイルシステムの設計

03G464　藤本 哲郎 （最所研究室）

本研究は、マルチユーザ環境に対応したバージョン管理機能を持つファイスシステムについて、その実現の際の問題点について検討し、ファイルのデータ構造やファイル操作及びバージョン管理特有の操作等の設計を行った。

1　はじめに

近年、パソコンの急速な普及により、利用されるプログラムやデータの種類は飛躍的に増加している。そこで扱われているプログラム等は、利用者によって改善や更新を行うために日々変更され続けている。しかし、利用しやすくするための変更が、全て改善とはならない場合がある。また、人為的ミス等で現在作業中のデータが破損する場合もある。このような場合、変更前の状態や破損前の状態に戻さなければならない。このため、更新される可能性がある全てのファイルの変更前の状態を把握し、それぞれを管理する必要があるが、ファイル毎のバージョン管理を行うことは、人的なコストを莫大なものとしてしまう。さらに、プログラムの開発やデータの利用等においては、複数人でファイルの変更を行うことが多々存在する。このような作業を行う環境を、本論文ではマルチユーザ環境と呼ぶ。この場合、ファイルの管理を多人数で行うという、さらに複雑な管理が必要になる。本研究は、マルチユーザ環境でのファイルのバージョン管理を軽減することを目的として、卒業研究時に行っていたファイルシステム [1] をマルチユーザ環境で利用できるように拡張するものである。

2　バージョン管理の概要

本研究では、ファイルシステムでバージョン管理を行う。対象は通常ファイルのみで、ディレクトリ、シンボリックリンク、特殊デバイスファイル、名前つきパイプ、ソケット等は対象としない。バージョンを作成するタイミングは、主に新規ファイルを作成する場合と、ファイルを書き込み用のモードでオープンする時である。ファイルをオープンする場合には、そのファイルの中で最新のバージョンが選択され、それを基に新しいバージョンを作成する。

本システムでは、バージョンファイルと呼ぶファイルを作成し、その内でバージョン管理を行う。バージョンファイルに変更が加えられた場合、その内部でバージョンが加えられる。バージョンファイルが i-node 番号で管理されるのに対し、バージョンファイルの中のバージョンはバージョン番号で管理する。また、ユーザのバージョン管理の負担を減らすために、通常のファイル操作でもバージョンファイルを扱えるようにする。

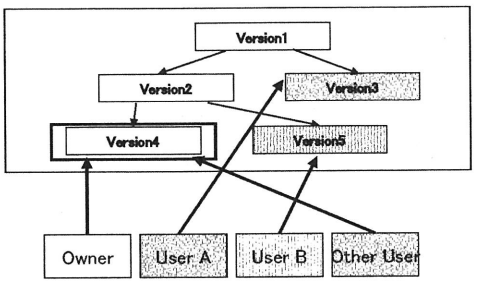

図 1: 最新バージョン

通常のファイル操作で各ユーザのバージョンを操作するためには、それぞれのユーザのバージョンを把握する必要がある。そこで、各ユーザの最新バージョンを保持することで、バージョンの指定を行わないでファイル操作を行うことができるようにする。しかし、特定のバージョンファイルを扱いたい場合には、バージョンファイル特有の操作を行う必要がある。この操作は主に、ファイルの管理者やバージョンの作成者が行う操作であり、利用するユーザはバージョン管理を行っていることを意識する必要がある。

バージョン管理を行うユーザを以下のように分類する。

- Owner – バージョン管理に対する全ての操作を行うことができる。
- Group User – バージョンファイルに編集を加えることができ、新たなバージョンを作成することができる。バージョン管理に対する操作は、自身が作成したバージョンにのみできる。
- Other – その他のユーザで、最新バージョンの読み込みのみできる。

基本的に Owner の最新バージョンをバージョンファイルの最新バージョンとする。それぞれの最新バージョンの関係を図 1 に示す。各バージョンのデータ部分をバージョンデータと呼ぶ。このデータの保存方法は、UNIX のファイル管理で使用されている i-node をモデルにしており、図 2 で示す構造を持つ。

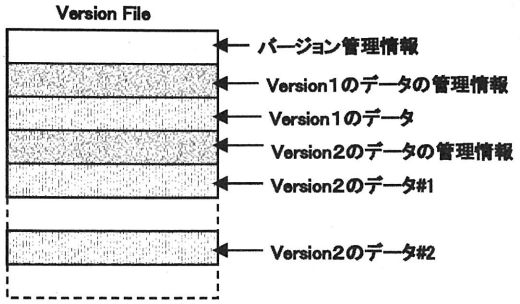

図 2: バージョンファイル内のデータ構造

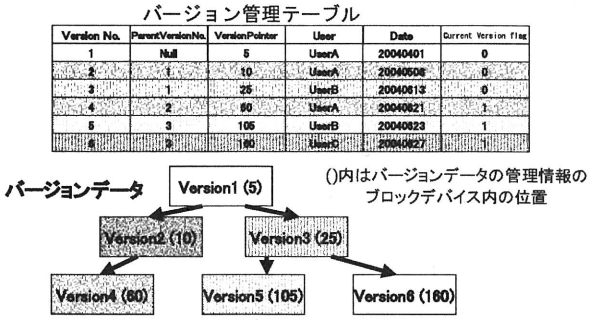

図 3: バージョン管理

次にバージョン管理の構成及び関係を図 3 に示す。バージョン管理情報には、バージョンファイル自身の情報、バージョン間の関係及びバージョンデータの管理情報の位置等の情報（バージョン管理テーブル）が含まれる。また、それぞれのバージョンを木構造で管理し、それぞれのバージョンデータをバージョンデータの管理情報で管理する。

3 設計

バージョンファイルの構造とバージョンファイルを操作するライブラリを設計した。バージョンファイルを操作するライブラリは以下の機能を含む。最初の 4 つは通常のファイル操作で、残りはバージョンファイル特有の操作である。

- ファイルオープン
- ファイルクローズ
- ファイルリード
- ファイルライト
- バージョンファイルの作成
- バージョンの削除
- バージョンデータのコピー
- ポリシーの設定

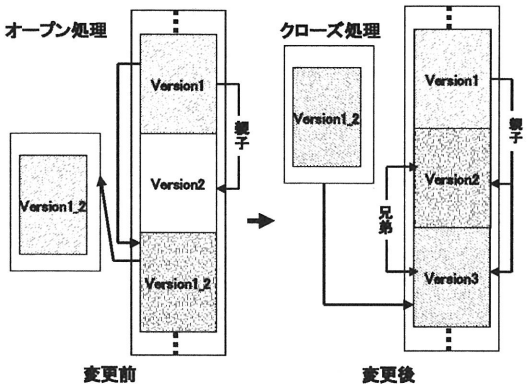

図 4: バージョン作成時のオープン

- ユーザ設定

バージョンデータを保存する場合、バージョンファイルがオープンされた時に、新たにバージョンが作られることを予測して、バージョンデータをコピーする。バージョン作成時のデータ操作を図 4 に示す。

4 まとめ

本論文ではマルチユーザ環境でバージョン管理を行う場合の問題を議論し、その問題を解決するファイルシステムの設計を行った。設計したファイルシステムの特徴は以下の通りである。

- 通常操作では、バージョン管理を意識しないでファイルにアクセスできる。
- バージョン管理を自動的に行うことができる。
- 複数ユーザのファイル変更に対応できる。
- バージョンファイルの最新状態を自動もしくは手動で決定できる。
- ユーザ毎のバージョン情報を保持する。
- バージョンの親子関係を管理する。
- バージョンの保存方法を選択できる。

今後の課題として、このシステムを OS のファイルシステムとして実装することが上げられる。また、本システムでは、バージョンを作成する場合にデータの差分管理を行っていない。差分管理を行うことで、ブロックデバイスの利用効率の向上が見込まれる。また、バージョンデータの管理部分に B 木を用いることで、データ検索の効率が向上することが考えられる。

参考文献

[1] 藤本哲郎: “ バージョン管理機能を持つファイルシステムのアプリケーションレベルでの実現に関する研究 ” 香川大学工学卒業論文, 2003.