

リンク構造ファイルシステムへの履歴機能の追加

05T276 森俊文 (最所研究室)

本研究では、リンク構造ファイルシステムへの履歴機能の追加を行う。それまで消去されていた過去のリンク情報およびデータブロックを残すことにより、ファイルの過去のバージョンを復元する。本稿では、履歴機能を追加したリンク構造ファイルシステムの設計について述べる。

1. はじめに

従来のファイルシステムでは、ファイルの汎用性を高めるために、ファイルをデータ構造のないバイトの集合として扱ってきた。ファイルの対する変更の操作として、データの上書き、ファイルの最後尾への新しいデータの追加および指定位置より後ろのデータの破棄がある。ファイルの指定した位置での、挿入や削除などの操作は、アプリケーション側で対応してきた。

本研究室では、ファイルを可変長ブロックの集合であると考え、ブロックの挿入、削除、追加を行うことができるリンク構造ファイルシステムの研究を行っている[1]。ここでいうブロックとは、テキストファイルの行やデータベースのレコードなど意味を持ったデータの塊を意味する。

履歴機能とは、削除・挿入・追加などのファイル操作が行われたファイルの任意の過去の状態を参照できるというものである。これにより、誤ったファイル操作で削除や変更されたファイルのファイル操作前の状態を復元したり、ファイル操作がおこなわれた日時の特定ができる。

本研究では、このリンク構造ファイルシステムへの履歴機能を追加した。同様の研究として、佐藤らの NILFS がある[2]。NILFS はファイルシステム全体で履歴をとるようになってきているが、本研究ではファイル単位で履歴をとる点が異なっている。

2. リンク構造ファイルシステムの概要

リンク構造ファイルシステムの構造およびその操作を図1に示す。

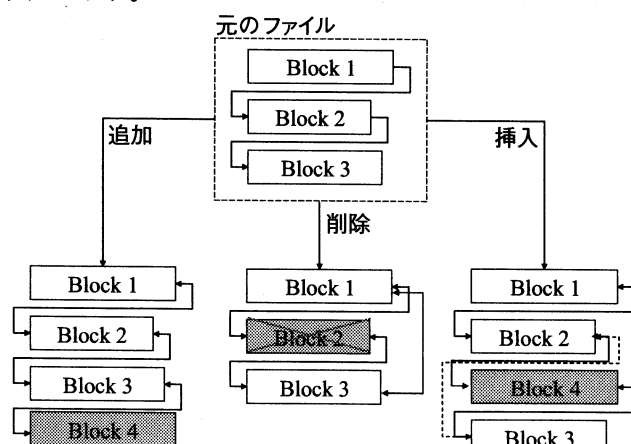


図1 リンク構造ファイルシステム

ファイルは、図1の元のファイルに示すように可変長ブロックがリンクで連結された形で表現される。このような構造にすることで、ブロックの追加、削除、挿入の3つの操作が可能となる。追加はリストの最後に新しいブロックを加え、削除は指定されたブロックをリストの中から削除し、挿入は指定されたブロックの前に新しいブロックを挿入する。

3. 履歴機能の概要

リンク構造ファイルシステムの特徴を活かし、履歴機能の追加を行う。従来のリンク構造ファイルシステムでは、削除された箇所を参照することはできなかったが、履歴機能を追加することで、任意の日時のファイルのバージョンを参照できるようにするものである。

任意の日時のファイルの状態を参照できるようにするためには、削除・挿入・追加などの操作を行う前のリンク情報およびデータブロックが必要である。これらの情報により、ファイルの過去の状態が参照できる。しかし、これまでのリンク構造ファイルシステムでは、上記の操作を行うとリンク情報は書換えられてしまうので、ファイル操作前のリンク情報を保存する仕組みが必要である。また、任意の日時を指定してファイルの状態を参照するためには、リンクの有効期間を知る必要があるため、リンクの開始時間および終了時間を保存しなければならない。そこで、挿入操作で消えていたリンク情報、削除操作で削除フラグにより削除されたとみなされたブロックがいつ削除されたなどのファイル操作に関する情報、リンクの開始時間および終了時間を保持するデータ構造などを保持する必要がある。

本研究では、新たにリンクブロックというデータ構造を導入して、これらの情報を保存することで履歴機能を実現する。また、これらのリンクブロックを管理するための情報を持つリンクデータブロックという構造も導入する。リンクデータブロックでリンクブロックを管理することで、リンクブロックの分散によるディスクアクセスのオーバーヘッドを防ぐ。

4. リンクブロック

挿入操作により書換えられて参照できなくなるリンク情報および削除操作により削除されたブロックをバイパスするために書き換えられるリンク情報を保存するた

めにリンクブロックを作成する。また、削除・挿入・追加などのファイル操作は、繰り返し行われ、その度にリンクブロックは作成されるので、その履歴を保持するために、図2および図3で示すように、リンクブロックはリスト構造を形成する。その結果、リンクブロックの情報として、リンク情報の有効期間を判別するためのリンクブロックの作成時間およびブロックの作成時間、次のリンクブロックをさすリンク情報、元のブロックをさすリンク情報の4つの情報を持たせることにした。例えば、図2の LinkBlock2には、LinkBlock2の作成時間、Block4の作成時間、LinkBlock1をさすリンク情報および Block4をさすリンク情報を保持している。一方、図3の LinkBlock2には、LinkBlock2の作成時間、Block3の作成時間、LinkBlock1をさすリンク情報および Block3をさすリンク情報を保持している。

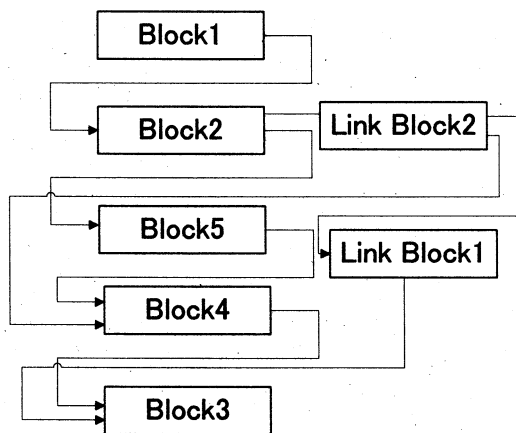


図2 リンクブロックを導入した挿入操作

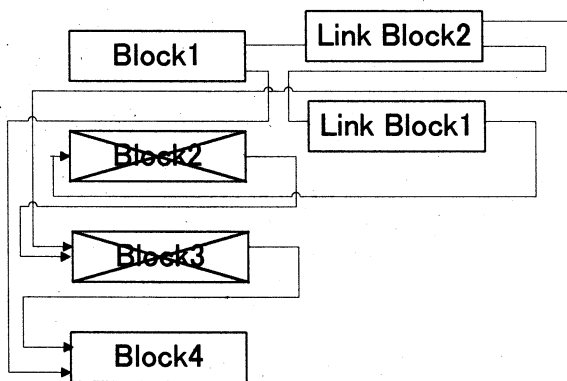


図3 リンクブロックを導入した削除操作

5. 設計

元のリンク構造ファイルシステムには、ファイルシステムの情報を保存するための File System 構造、ディレクトリの情報を保存するための Directory 構造、ファイルの情報を保存するための File 構造、ファイルを構成するためのブロックの情報を保存するための Block 構造の4つの構造を持っている[1]。

履歴機能を追加したリンク構造ファイルシステムでは、リンクブロックを追加したことにより、File System 構造に、リンクブロックを管理している

リンクデータブロックへのポインタ、Block 構造に、ファイル操作をおこなうことで書き換えられるリンク情報を保持するリンクブロックへのポインタを新たに追加し、リンクブロックの情報を保存する LinkBlock 構造、リンクブロックを管理する LinkDataBlock 構造の2つ構造を追加した。

LinkBlock 構造が保存している情報は、リンクブロックの作成時間、リンクしているブロックの作成時間、元のブロックへのポインタおよび次のリンクブロックへのポインタである。LinkDataBlock 構造が保持している情報は、LinkDataBlock 構造の作成時間、管理しているすべてのリンクブロックを使い果たした時間、管理している未使用リンクブロックの数および未使用リンクブロックの先頭へのポインタを保持している。

次にファイルシステムに対する操作について述べる。元のリンク構造ファイルシステムは、Create、Open、Close、Read、Seek、Append、Insert、Deleteの8つの機能を持っている[1]。履歴機能の追加したリンク構造ファイルシステムでは、リンクブロックを追加したことにより、Insert と Delete の2つ機能に履歴を残すための仕組みを取り入れた。

Insert は、リンクデータブロックから確保してきた未使用リンクブロックを現在位置のブロックの一つ前のブロックにリンクをつなぎ、現在位置のブロックの一つ前のブロックから現在位置のブロックへのリンク情報をリンクブロックへ保存する。そして、新しく挿入するブロックへ現在位置のブロックの一つ前のブロックからリンクをつなぎ、新しく挿入するブロックから現在位置のブロックへリンクをつなぎ、ディスクにデータを書き込む。Delete は、リンクデータブロックから確保してきた未使用リンクブロックを削除したいブロックの一つ前のブロックにリンクをつなぎ、削除したいブロックをバイパスするようにリンクをつなぎ、ディスクにデータを書き込む。

6. おわり

以上、リンク構造ファイルシステムの履歴機能の追加について述べた。現在のところ、設計まで完了している。しかし、実装まで至っていない。今後、実装を完了させて動作テストをおこない、動作確認をとりたい。また、ファイルの履歴参照方法については、検討しているところである。ファイルの履歴参照方法の詳細設計および実装していきたい。

参考文献

- [1] 津紀孝, 最所圭三, “可変長単位ブロックで操作を行うファイルシステムの開発” 香川大学工学研究科 修士論文 2007
- [2] 佐藤孝治, 小西隆介, 木原誠司, 天海良治, 盛合敏, “ログ構造化ファイルシステム NILFS の設計と実装”, 情報処理学会コンピュータシンポジウム論文集, P125-134, 2008