リンク構造ファイルシステムで
既存アプリケーションを利用するためのツールの検討

09t267　松永　裕之(最所研究室)

ファイルを可変長ブロックの集合として扱うリンク構造ファイルシステムを，従来のアプリケーション使用できる形に変換し，編集された内容を反映するツールの試作し，試作したツールを使用して，反映方法の評価を行った．

1　はじめに

本研究室では，ファイルを可変長ブロックの集合として扱うことでファイルの履歴を保存することができるリンク構造ファイルシステムの開発を行っている [1][2]．このファイルシステムではブロックの挿入・削除によりファイルを操作しているため，既存のアプリケーションから直接使用することができない．このため，本研究では，従来のアプリケーションが操作を行ったファイルをリンク構造ファイルシステムに反映できるツールの実現を目指す．本稿では，更新を差分として取り出して，挿入・削除操作に変換し，リンク構造ファイルシステム上ファイルに反映するツールの検討および，ブロックサイズに関する評価を行う．

2　リンク構造ファイルシステム

図 1 にリンク構造ファイルシステムにおいて挿入・削除を行った場合のリンクの状況の例を示す．ここではblock1 と block2 の間に新 block5 を挿入し，block3 を削除している．リンク構造ファイルシステムでは，挿入・削除によって失われる情報を LinkHistory（図中の LHist1とLHist2）を用いて残す．block5 の挿入の際に失われるblock1 とblock2 のリンク情報をLHist1 で，削除の際に失われる block2 と block3 の間のリンク情報を LHist2 で残している．これらの LinkHistory を用いることで，過去のリンクをたどることができる．

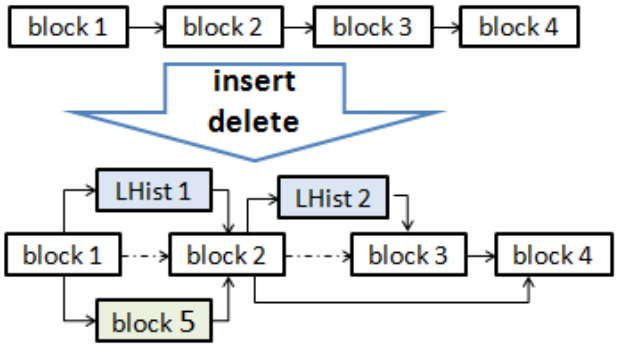

図 1.リンク構造ファイルシステムにおける履歴機能

3　従来のアプリケーションへの対応

従来のファイルシステムへ対応するためのツールの概要を図 2 に示す．ツールは元ファイルを従来のファイルシステムで編集できる形式に変換して出力する出力ツールと，編集されたファイルと編集前のファイルの差分をリンク構造ファイルシステムの操作に変換し反映する反映ツールで構成される．

本研究では，テキストファイルを対象とし，差分の取得に diffコマンドを用いる．diffコマンドはリンク構造ファイルシステム内のファイルを直接読みだすことができないため，diff コマンドで編集前のファイルを見ることができるようにするためにコピーファイル（Copy）を用意している．差分から挿入・削除に変換するパターンとしていくつか考えられる．以下に出力ツールと反映ツールでの操作について述べる．

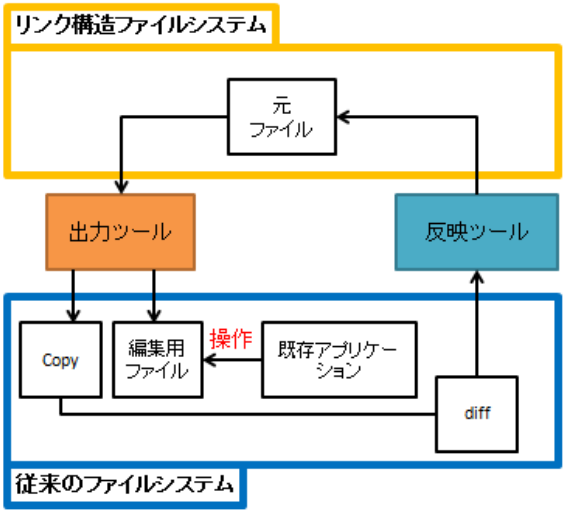

図 2.ツールの概要

出力ツール

出力ツールでは対象のファイルのブロックを 1 つずつ走査し，従来のファイルシステムで使用できる形式に変換する．同時に編集前のファイルとして Copy を生成する．この時，変換後のファイルにブロック構造を残すことができないため，行とブロックの対応関係を表すファイル LB を生成する．これにより，元ファイルのブロックと差分の中の行の対応を取ることができる．

反映ツール

図 3 に反映の代表的な方法についての例を示す．ファイルは 3 つのブロックを持ち，その中の block B，block C の部分にまたがって編集が行われたとする．変更されていない部分をそれぞれ，block B’，block C’，変更された部分を block D として，それぞれの方法でのブロックの変換について述べる．①の方法は，変更された部分とされていない部分をそれぞれ異なるブロックとして更新する．この方法では，ブロックが増加する．②の方法は編集のあった部分とない部分をまとめて 1 つのブロックとし

て更新する．この方法では，ブロック数は減少する．③の方法は編集のあった部分の終わりの部分までを 1 つのブロックとして更新する．

ブロックを更新する際，①の方法では，ブロック数が増加するので，リンク数も増えるため，リンクの占める領域および，リンクをたどる数が増えてしまうという問題がある．一方，②の方法では，ブロック数が減少することで，リンクの占める領域は減り，リンクをたどる数も減るが，ブロックが大きくなってしまい，更新の際，編集の無い部分まで，更新されたブロックに含まれ，ファイルサイズが必要以上に大きくなってしまい，記憶領域の使用効率が悪くなるという問題がある．③の方法はブロック数に変更がなく，他の方法より効率が良いのではないかと考えられるが，ブロックサイズが偏ってしまうことが考えられる．また，関数単位でブロックにしたいといったユーザの意向がある場合もあるので，それも考慮して反映方法を選択する必要がある．

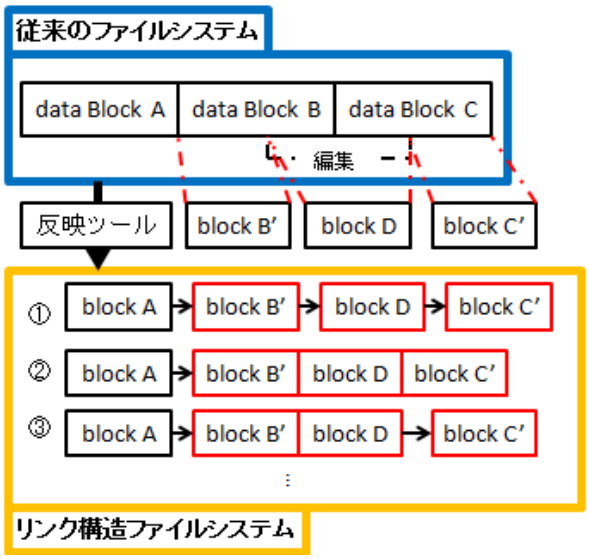

図 3．反映方法

4　評価

ブロックサイズが更新に与える影響を調べるために，提案したツールを試作し，実験を行った．評価基準として，リンク構造ファイルシステム上のファイルサイズ(リンク情報や削除されたブロックを含む)を用いる．実験では，当研究室で実際に開発バージョンごとに完全なファイルが残っているソースコードを，指定した行数でブロック構造に変換する．変換したファイルを出力ツールを用いて編集用のファイルと LB を生成する．次のバージョンと編集用のファイルの差分を取り，反映ツールを用いて更新し，リンク構造ファイルシステム上のファイルを更新する．更新されたリンク構造ファイルシステム上のファイルサイズの変化を調べた．実験に用いたファイルの諸元を表1に示す．

初期ファイルとして，指定した行数を1つのブロックとする．ブロックサイズを1～12に変化させたとき，それぞれの更新回数におけるファイルサイズを測定した．その結果を図4に示す．初期ファイル(更新回数0)はブロックを大きくするほうが保存効率が良いが，大きすぎると更新するたびにファイルサイズが必要以上に大きくなっているのが分かる．また，ブロックの分割が小さすぎる場合もリンク情報が多くなっており，ファイルサイズは大きくなっていることが分かる．今回の実験では，3～5行程度の時，ファイルサイズが小さくなった．このことから，更新するブロックサイズは大きすぎても小さすぎても保存効率が悪くなることが確認できた．

表 1.実験に使用したファイルの概要

更新回数	ファイルサイズ		更新量	
	byte 数	行数	byte 数	行数
0 (初期)	3336	141		
1	4872	194	2100	79
2	4928	196	96	4
3	5839	240	1033	50
4	5894	242	55	2

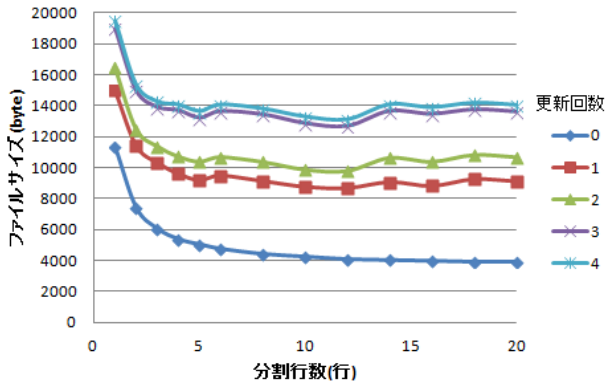

図 4.初期行数とファイルサイズの関係

5　まとめ

リンク構造ファイルシステムを既存アプリケーションで利用するためのツールを提案および，既存のアプリケーションによって行われる編集をリンク構造ファイルシステムに反映する方法について述べた．さらにブロックサイズが更新に与える影響についての評価も行った．これにより，適切なブロックサイズで更新する必要があることを確認できた．ブロックの更新は挿入，削除を行っているが，動作を 1 つにまとめた更新という動作があることで，LinkHistory を 1 つにまとめることができる．また，さらに大きいファイルに関しても実験を行う必要がある．

6　参考文献

[1] 津紀孝，最所圭三，“行指向ファイルシステムについて”平成 18 年度 電気関係学会四国支部連合大会論文集，15-37，p.242，2006

[2] 小林憲弘，大橋洸一，最所圭三，“リンク構造を用いたファイルの履歴管理について”，情報処理学会第 74 回全国大会講演論文集，3J-4，pp.1-89 - 1-90，2012.3