

システム開発におけるファイルの履歴管理システムの実装方式の比較評価

2 F - 6

横田大輔, 早川栄一, 並木美太郎, 高橋延匡

(東京農工大学 工学部 電子情報工学科)

1. はじめに

二次記憶装置の大容量化によって、履歴を保存しておくことが容易に行えるようになってきた。プログラム開発のデバッグ時には、履歴を利用してバグを再現させデバッグを行ったり、プログラムへの間違った修正を戻すことができるので、履歴を取ることは非常に有効である。

そこで筆者は、プログラム開発での利用を目的とした履歴管理機構を持ったファイル ID 方式のファイルシステム [1] (以下、ファイル ID 方式と略記する) を設計、実現した。これは、ファイルシステムで履歴管理を行うことで、ディレクトリファイルを含むすべてのファイルに対して履歴を統一的に管理できる。これは、デバイスドライバやサーバなどの他のプログラムと密接に関係し、統一的なバージョン管理が必要となるシステム開発において非常に有効である。

本報告では、本システムでの開発の履歴を利用して、ディスク消費容量などの履歴のコストと有効性について他方式との比較評価を行う。

2. ファイル ID 方式

本システムは、ファイルシステム内に履歴管理機構を組み込んだ履歴システムで、次のような特徴がある。(1) ファイルだけでなくディレクトリの履歴も利用できる。(2) ファイルやディレクト

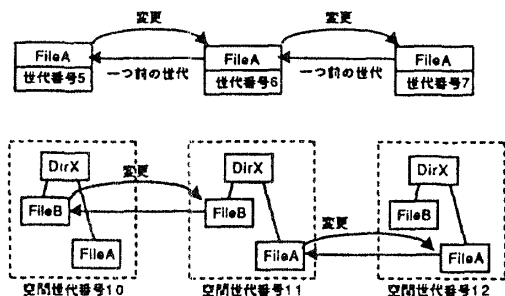


図 1 ファイルとディレクトリの履歴モデル

り単位で履歴を呼び出せる。(3) アプリケーション層で統一した履歴資源を利用できる。

履歴は、ファイルとディレクトリを図 1 のようなモデルで管理しており、ファイルのクローズのタイミングで履歴を自動的に更新している。

3. 履歴システムの比較

システム開発における、ファイル ID 方式の有効性を示すために、他の実装方式と利用できる履歴の機能を比較する。

比較する実装方式を次に示す。

(1) RCS/SCCS [2]

これは、プログラムソースの履歴管理を目的とし、テキストファイルの履歴管理を行うアプリケーションプログラムである。

(2) 仮想ディスク方式 [3]

この方式は、仮想ディスクと呼ばれる、書換えによって履歴を保存していくディスク単位の履歴管理を行うデバイスドライバである。

ここで、システム開発における履歴の利用方法を考えてみる。OS やサーバなどのシステムソフトウェアは、アプリケーション、デバイスドライバ、ライブラリなどと密接に関係しており、この間のインタフェースに起因したバグなどが発生する。したがって、過去のバグを再現したいときなど、システムを過去の状態に戻す場合には、これらの周辺のソフトウェアのバージョンを同時に戻したい場合がある。さらに、プログラムの修正時期を特定するために、ファイルごとに履歴を調査する必要がある。

このような履歴への要求に対して SCCS などのアプリケーション層で履歴管理を行う場合には、(1) オブジェクトファイルやディレクトリなどの

履歴を扱えない、(2) プログラムごとに異なる間隔で履歴を更新しているため履歴の同期を取ることが難しい、などの問題がある。

また、仮想ディスク方式など、デバイスドライバ層で履歴管理をおこなう場合には、ファイルやディレクトリなどの情報が欠落するので、履歴をファイルやディレクトリ単位で追跡することができないという問題がある。

以上の比較を表1にまとめる。

表1 実装方式の比較

実装方式	RCS/SCCS	ファイルID	仮想ディスク
実装位置	アプリケーション	ファイルシステム	デバイスドライバ
ファイル単位の利用	○	○	×
統一的な履歴管理	×	○	○

表1から、システム開発などの大規模なプログラム開発では、履歴の統一的な管理とファイル単位での履歴の管理ができるので、ファイルシステムで履歴管理を行うファイルID方式の履歴管理方式が最も有効である。

4. プログラム開発における履歴のコスト

プログラム開発での本システムの有効性を評価するために、本システムの容量面のコストを測定した。測定は、本システムの開発での履歴を、仮想ディスク方式の履歴システムで記録し、後に本システムに履歴を移すことで消費容量の評価を行った。

開発環境と規模を表2に示す。

表2 開発環境と規模

CPU	日立2050/32E
ドライブ	日立5インチ追記型光ディスクドライブ
OS	OS/omiconV2
全ソース行数	12000
開発期間	6ヶ月
全ソースサイズ	400Kbyte

履歴の記録は、①設計、②インプリメンテーション、③デバッグの開発全体の内、②③のフェーズを記録した。

これらのフェーズでは、プログラムやオブジェクトなどファイルのクローズのタイミングで履歴

を更新し、過去のすべてのテスト実行を再現できるようにプログラムソース、オブジェクトファイルの履歴を記録した。開発時間と書込みファイルサイズ、ディスク累計消費量を図2に示す。

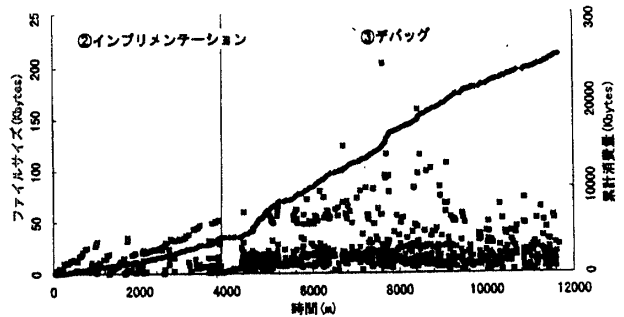


図2 履歴のディスク消費量

ディスク容量の消費は、プログラマーのスタイルや変更ファイルファイルサイズによって変化すると考えられるが、グラフをみると消費量はほぼ一定であり、開発期間に依存する。したがって、履歴のコストはプログラムの種類や規模に依存し、約1万行の本システムの場合では、ディスク消費容量は約25Mバイトであることが分かった。

5. おわりに

本報告では、ファイルID方式の履歴管理システムの有効性を評価するために、SCCS、仮想ディスク方式との比較を行い、システム開発においてはシステム全体の統一的な履歴を管理でき、特定のファイルの履歴を追跡できるファイルID方式が有効であることを定性的に述べた。さらに、システム開発における履歴のディスク容量のコストを明らかにした。今後は、いくつかのプログラム開発の履歴をとり、容量消費量の指標を示す予定である。

参考文献

- [1] 横田大輔他：OS/omiconにおける履歴管理のためのファイルシステム，情報処理学会システムソフトウェアとOS研究会報告，66-4，1994
- [2] AT & T UNIX パシフィック株式会社：UNIX System V プログラマガイドリリース 3.1，日本ソフトバンク
- [3] 横関隆他：追記型光ディスクの仮想的な書換えと世代管理機能の実現，電子情報処理学会論文誌 Vol.72 No.6，1989