

## フラッシュメモリを用いたUNIX ファイルシステム

5U-5

川口敦生, 西岡真吾, 元田浩

(株)日立製作所基礎研究所

## 1 はじめに

UNIX ファイルシステム [1] をサポートするフラッシュメモリデバイスドライバを開発した。本ドライバは、フラッシュメモリの特性から来る様々な制約に対処するため、LFS [2] と同様な方法で、データの変化分を順にフラッシュメモリに書き込む。有効でないデータを保持しているフラッシュメモリの領域はクリーナが明示的に消去する。ベンチマークテストによると、本ドライバを使用するファイルシステムは、フラッシュメモリの特性にもかかわらず、MFS [3] に遜色ない性能を有している。

フラッシュメモリはその内容を一括して電氣的に消去できる不揮発性半導体メモリである。現在、様々なフラッシュメモリ ([4][5] 他) が入手可能であるが、その特性は表1のように整理することができる。

表1: フラッシュメモリの特性

読出し速度	80-150ns/byte
書込み速度	10 $\mu$ s/byte
消去単位	512Byte-64KByte/erase sector
速度	10ms-1s/erase sector
回数制限	10 万回
電力消費	30 - 50mA (動作時) 20 - 100 $\mu$ A (待機時)
価格	10 - 30 \$/MByte

フラッシュメモリは読み出し専用メモリ (ROM) と同様なインタフェースを持っており、CPU と直結して利用可能である。フラッシュメモリチップは消去セクタの集合から構成されており、データの書き込みは、消去セクタ毎に明示的に消去操作を施した後、データ書き込みコマンドを発行して行う。一般に書き込み速度は読み出し速度に比べて著しく遅い。

フラッシュメモリはハードディスク (HDD) に比して、現在のところ容量あたり 5~10 倍高価であるが、軽量小型である、消費電力が少ない、可動部がなく衝撃に耐え

るといった特性を有しており、携帯型情報機器の不揮発性記憶装置として有望である。

本稿では、マルチタスク環境において、上に述べたフラッシュメモリの特徴 (欠点) にどのように対処するかを探ることを目的として UNIX 用に設計、実装したフラッシュメモリデバイスドライバおよびそれにもとづくファイルシステムの性能について報告する。

## 2 設計と実装

HDD をエミュレートするデバイスドライバを開発し、その上で既存のファイルシステムモジュールを動作させることとした。フラッシュメモリは CPU から直接アクセスできるので、メモリ管理システムと強固に連係する専用のファイルシステムを開発して効率を高めるというアプローチも考えられる。本研究では作業量の観点から、まず性能の見通しを得ることを優先してデバイスドライバアプローチを採った。

フラッシュメモリはその構造上、データの上書きができない。RAM DISK のような単純な実装では、書き込み要求ごとにまず消去操作が必要となり、消去時間が長いために、システムのスループットが大幅に低下することが予想できる。

我々のドライバは、すべての書き込み要求に対して、フラッシュメモリシステムの先頭から順番にデータを書き込むログアプローチ [2] を採っている。逐次的に書き込むため、次にデータを書き込む領域も自ずとわかる。すなわち書き込み要求が来る前に必要に応じて消去操作を実施しておくことができる。ファイルシステムモジュールが指定したブロック番号と実際にデータを書き込んだ領域のアドレスを変換テーブルに保持しておき、後のデータの読み出しを可能としている。

以前書き込み要求があったブロック番号に対して、再度書き込み要求があった時は、新しいデータをログの末尾に追加し、変換テーブルを更新する。以前のデータを保持している領域は無効と記録する。システムの動作につれてログは有効な領域と無効な領域に断片化されていく。HDD を想定しているファイルシステムモジュールは 512 バイトを最小データ単位としている。フラッシュメモリの消去セクタは必ずしも 512 バイトとはなっておらず、結局消去セクタが有効なデータを保持している領

A Flash-Memory Based UNIX File System

Atsuo Kawaguchi, Shingo Nishioka, Hiroshi Motoda

Advanced Research Laboratory, Hitachi, Ltd.

Hatoyama, Saitama 350-03, Japan

域と無効領域に断片化されることになる。

我々のドライバは適当な時点でクリーナを動作させる。クリーナは消去する消去セクタを選択し、その中の有効なデータを他の消去セクタにコピーして選択した消去セクタのデータ領域をすべて無効にする。その後、明示的に消去操作を行い、選択した消去セクタを書き込み可能にする。

図1に本ドライバがフラッシュメモリ上に構築するデータ構造の概要を示す。

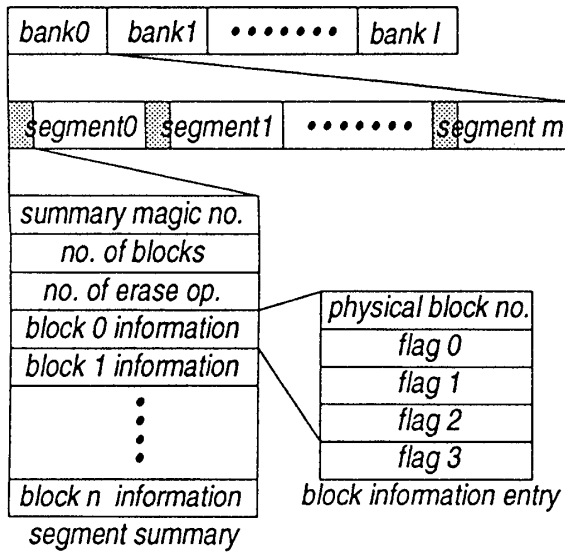


図1: フラッシュメモリ上のデータ構造

フラッシュメモリシステムは同時に書き込み、消去が可能な bank の集まりと見なされ、各 bank は消去セクタである segment の集合からなる。各 segment の先頭に segment summary が書き込まれ、その segment のブロックの数、消去の回数、各ブロックの状態 (block information) が記録される。各ブロック毎に対応する実際のブロック番号、一連のフラグを記録する。フラグ自身も逐次的に利用され、消去操作をせずにブロックの状態変化を記録できる。これらのデータはシステム起動時に読み取られ、変換テーブル等デバイスドライバが動作するのに必要な内部データを構成するのに使われる。また書き込みや消去処理に際して随時更新される。

### 3 性能評価

試作デバイスドライバの性能を Andrew ベンチマーク [6] を使って評価した。測定には 4.4BSD UNIX がシングルユーザモードで動作している 40MHz R3081, 主記憶 16MByte の試作計算機を用いた。フラッシュメモリシステムは Intel 28F008SA で構成された 16MByte のものである。主記憶、フラッシュメモリと CPU 間の最大データ転送速度は 8MByte/秒である。ファイルシス

テムモジュールには BSD FFS を用いている。

表 2: ベンチマーク結果

	Phase					Total
	1	2	3	4	5	
MFS[3]	3.1	22.0	27.0	35.2	80.6	167.9
Ram Disk	3.0	22.8	27.2	36.2	81.4	170.6
クリーナなし	4.2	23.2	27.5	37.6	86.0	178.5
クリーナあり	4.3	24.8	28.2	37.4	87.3	182.0

表2にベンチマーク結果を示す。MFS および Ram Disk の各行は試作計算機の主記憶性能を表していると考えられる。本ベンチマークではクリーナのオーバーヘッドがあまり結果に影響を及ぼしていない。なお、Phase 3 と 4 ではバッファキャッシュが有効に作用しており、実際には入出力は起こっていない。

### 4 おわりに

ログアプローチを採ることによって、実用的な UNIX ファイルシステムをフラッシュメモリを使って構成できることを示した。作業量の観点からデバイスドライバを試作してフラッシュメモリの特性に対処することを試みた。デバイスドライバレベルでは、あるブロックが無効になったことが上位のファイルシステムモジュールが実際に上書きするまでわからないため、クリーナ動作の是非判定を効率良く行うことが難しい。今後は専用ファイルシステムモジュールの開発も見据えて、消去セクタ選択ポリシーの検討や他のベンチマークを使った評価を進める予定である。

### 参考文献

- [1] M. K. McKusick, W. Joy, S. Leffler and R. S. Fabry, "A Fast File System for UNIX", ACM Transactions on Computer Systems, Vol.2, No.3, 1984.
- [2] M. Rosenblum and J. K. Ousterhout, "The Design and Implementattion of a Log-Structured File System", ACM Transactions on Computer Systems, Vol.10, No.1, 1992.
- [3] M. K. McKusick, M. J. Karels, and K. Bostic, "A Pageable Memory Based Filesystem", Proc. '90 Summer USENIX, 1990.
- [4] Advanced Micro Devices, Inc., "Am29F040 Datasheet", 1993.
- [5] Intel, Corp., "28F008SA Datasheet", 1992.
- [6] J. H. Howard, et. al., "Scale and Performance in a Distributed File System", ACM Transactions on Computer Systems, Vol.6, No.1, 1988.