

## マルチメディア処理に適したリアルタイムファイルシステム

荒川 智史<sup>†</sup> 勝部 弘嗣<sup>††</sup> 御田村 晃<sup>††</sup> 瀧本 栄二<sup>††</sup> 毛利 公一<sup>†</sup> 大久保 英嗣<sup>†</sup>  
<sup>†</sup>立命館大学工学部情報学科 <sup>††</sup>立命館大学大学院理工学研究科

### 1 はじめに

動画や音声の再生を行うマルチメディアアプリケーションは、大容量の連続メディアデータを扱う。このようなデータを円滑に再生するために、カーネルは、アプリケーションに対して安定したデータ供給を行う必要がある。しかし、従来のファイルシステムでは、ファイルを分割してディスクに配置する可能性があり、ブロック単位でシークを行っているため、読出しの予測は困難である。このような背景より、本研究では、ファイル読出し時間の予測を可能とするリアルタイムファイルシステムの構築を、リアルタイムオペレーティングシステム Easel [1] 上で行っている。本論文では、以下に挙げる連続メディアデータの特徴を考慮した、ディスクへの格納手法を提案する。

- ファイルサイズが大きい。  
120 分の動画の再生を考えた場合、そのデータ容量は 4.7GB であり、通常のファイルに比べデータサイズが大きい。
- 順次アクセスによるデータの読出しが多い。  
連続メディアデータは、ファイルの先頭から順番に読み出される。
- データの変更が少ない。  
連続メディアデータの主な使用用途は再生である。そのため、プロセスは、データの変更や追加といった書込み処理は行なわない。すなわち、連続メディアデータは、多くの場合読出し専用である。

本ファイルシステムは、上記の連続メディアデータの特徴を考慮し、データをディスクの連続領域に割り当てる。これにより、ディスクアクセス時間の予測を可能とする。さらに、ディスク操作時に、デッドラインを指定可能とすることで、Easel の I/O スケジューラによって、プロセスが要求する時刻までのデータ供給を保証する。

以下、本稿では、2 章で従来のファイルシステムについて述べ、3 章で本ファイルシステムの特徴について述

A Real-Time File System for Multimedia Processings  
 Satoshi Arakawa<sup>†</sup>, Hirotsugu Katsube<sup>††</sup>, Akira Mitamura<sup>††</sup>,  
 Eiji Takimoto<sup>††</sup>, Koichi Mouri<sup>†</sup>, and Eiji Okubo<sup>†</sup>  
<sup>†</sup>Department of Computer Science, Faculty of Science and  
 Engineering, Ritsumeikan University  
<sup>††</sup>Graduate School of Science and Engineering, Ritsumeikan  
 University

べる。4 章でファイルの読出し時間の評価について述べ、5 章でまとめと今後の予定について述べる。

### 2 従来のファイルシステム

従来のファイルシステムは、ファイルをディスクへ配置する際、ディスクの先頭から空き領域を検索し、その領域を割り当てる。ファイルサイズが検索した空き領域より大きい場合、ファイルを分割して格納する。ファイルを分割して空き領域に割り当てる手法は、ディスクの空き領域を効率的に扱うことが可能となる。しかし、ファイルが分割された場合、読出し時に余分なシークが必要となり、ファイル I/O 時間の予測が困難となる。連続メディアデータは、通常のデータよりサイズが大きい場合、ファイルの分割の影響を受けやすい。

また、従来のファイルシステムは、ファイル情報をもとに、ファイルの物理的な位置を特定するが、メモリ上にファイル情報が存在しない場合、ディスクから該当のファイル情報を読み出す。このファイル情報の検索時に発生するディスクアクセスも、ファイル読出し時間の予測を困難にする。

### 3 マルチメディア処理に適したリアルタイムファイルシステム

#### 3.1 ファイルシステムの特徴

本ファイルシステムは、連続メディアデータの特徴を考慮し、次のような特徴を持つ。

- 連続メディアデータをディスクの連続領域に割り当てる。
- ファイル情報の検索によるディスクアクセスをなくすために、全てのファイル情報をメモリ上に配置する。
- ファイル読出し処理のデッドラインを指定可能なインタフェースを提供する。

#### 3.2 ファイルシステムの実現手法

##### 3.2.1 読出し処理の流れ

本ファイルシステムを用いたときの read システムコール処理の流れを以下に示す。ここでは、バッファキャッシュを用いず、すべてディスクアクセスを行う場合について述べる。

- (1) ユーザプロセスが、ファイルディスクリプタ、バッファアドレス、データサイズ、デッドラインをパラメータとして read システムコールを発行する。
- (2) ファイルシステムは、ファイルのディスク上における位置を特定し、I/O スケジューラにディスクアクセスを要求する。
- (3) I/O スケジューラは、予測読出し時間、優先度、デッドライン時刻に基づき ディスクアクセス要求の I/O スケジューリングを行う。また、ファイルシステムに対して、読出し処理の終了通知を行う。プロセスが要求するデッドラインが守れない場合、エラーを通知する。
- (4) ファイルシステムは、I/O スケジューラからディスク I/O の終了、もしくはエラー通知を受け取り、プロセスに通知する。

### 3.2.2 領域割当て

ファイルを連続領域に割り当てる手法では、データの内容を変更することによりファイルサイズが大きくなる場合、新たに連続領域を確保し直す必要がある。しかし、連続メディアデータは、データの変更は少ないため、前述の問題が発生する可能性は低い。そこで、本ファイルシステムでは、データの追加による変更を不可とする。しかし、通常のファイルは、データの変更が多発するため、連続領域に割り当てる手法を用いる場合、ディスクの利用効率が悪くなる。この問題は、VFS (Virtual File System) を用いて、従来のファイルシステムとマルチメディアの本ファイルシステムを併用することにより、解決可能である。

## 4 ファイル読出し時間の評価

### 4.1 評価方法

リアルタイムオペレーティングシステム Easel を用いて、ファイル読出し時間の測定を行った。評価は、連続領域と非連続領域において 50 個のファイルの読出し時間を測定し、平均読出し時間と標準偏差を求めた。非連続領域に配置したファイルは、同容量のファイルであっても分割の状況は異なっている。測定を行ったファイルは、100KB から 4,900KB までの 300KB ごとに用意した。実験は、Pentium III 900MHz の PC/AT 互換機を用いて行った。また、ハードディスクの性能を表 1 に示す。

### 4.2 計測結果

各領域の平均読出し時間と標準偏差の値を図 1 に示す。実験結果より、連続領域に配置したファイルの平均読出し時間は、ほぼ単調増加である。また、標準偏差の値は、読出し時間に比べ非常に小さいことから、同容量

表 1 実験に用いたハードディスクの性能

Capacity	40GB
Interface	ATA/ATAPI-5
R.P.M	7200rpm
Seek time	8.5ms
Cache	2MB

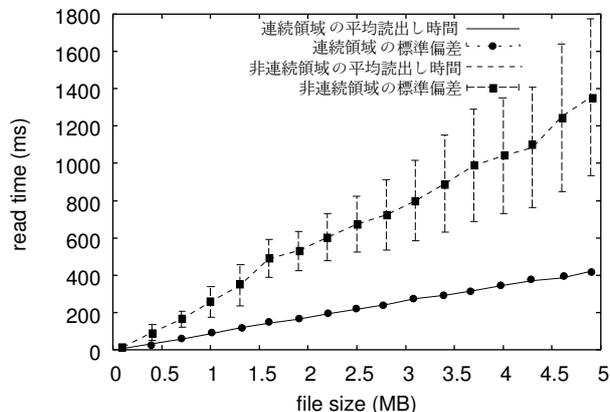


図 1 平均読出し時間と標準偏差

のファイルであれば読出し時間は一定であることがわかる。非連続領域に配置したファイルの平均読出し時間は、ほぼ単調増加であるが、連続領域に配置したファイルより読出し時間が遅い。また、標準偏差の値が大きいことから、同容量のファイルであってもファイルの分割状況によって、読出し時間に大きなばらつきがあり、容量の大きいファイルほどばらつきが大きくなっている。これらのことから、ファイルを分割する可能性のあるファイルシステムにおいて、容量の大きいファイルほど読出し時間の予測は、困難であるといえる。したがって、本ファイルシステムを用いることにより、読出し時間の短縮と、読出し時間の予測が可能になると考えられる。

## 5 おわりに

本稿では、マルチメディア処理に適したリアルタイムファイルシステムについて述べた。本ファイルシステムは、連続メディアデータに対し、ディスクの連続領域を割り当てる。これにより、ディスクアクセス時間の予測性の向上を図る。また、デッドラインが指定可能なインターフェースを提供し、本ファイルシステムと I/O スケジューラが協調して処理を行うことにより、プロセスが要求する時刻までのデータ供給を実現する。今後の予定として、本ファイルシステムに適應したバッファキャッシュの構築が挙げられる。

## 参考文献

- [1] 谷出新: “リアルタイムオペレーティングシステム Easel におけるメモリ管理機構,” 情報処理学会研究報告 2001-OS-86, Vol 2001, No.21, pp. 91-98 (2001).