5U－1

連続メディアのためのディスクスケジューリング機能を持つファイルシステムの実現

片山吉章　戸川敦之　中村素典　大久保英嗣　大野豊

立命館大学理工学部情報学科

1　はじめに

連続メディアを考慮していないファイルシステムにおいて，複数の連続メディアを再生する場合，それぞれの連続メディアが途切れないように再生するのは困難である．なぜなら，連続メディアの再生を行う際に，連続メディアの再生に間に合うようにハードディスクからメモリに連続メディアのファイルを読み込むことが可能かどうか判断することができないからである．たとえ判断することができたとしても，連続メディアのファイルを読み込むことを考慮していなければハードディスクドライブの性能を有効に発揮することはできない．

本研究では，これらの問題点を解決し，なおかつ通常のファイルシステムとほぼ同様のユーザインタフェースを持つファイルシステムを実現することを目的とする．

2　ファイルシステムの設計

2.1　連続メディアを扱うシステムの構成

連続メディアを扱うシステムは，連続メディアのファイルが格納された一つのハードディスクドライブと一つのプロセッサ（例えばワークステーション）からなるものと仮定する．ハードディスクドライブの内部に存在するディスク枚数は1枚とする．また，それぞれの連続メディアのファイルと一般のファイル(連続メディア以外のファイル)はハードディスク上に連続して格納されているものとする．

2.2　連続メディアを読み込むスケジュールについて

複数の連続メディアのファイルをハードディスクからメモリに読み込む順序として次の方法が考えられる．

- 再生レートの大きい連続メディアを優先する．
- 個々の連続メディアに優先順位を設ける．
- 主記憶上のバッファに先読みしておいたデータの残りが少ない連続メディアを優先する．

An implementation of a file system using disk scheduling for continuous media
Yoshiaki Katayama, Atsushi Togawa, Motonori Nakamura, Eiji Okubo and Yutaka Ohno
Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University
1916 Noji, Kusatsu, Shiga 525, Japan

しかし，これらの方法はそれぞれの連続メディアのファイルがハードディスク上のどの位置に格納されているかを全く無視したものであるので，より多くの連続メディアを扱うにしたがってファイル間をシークするハードディスクドライブのヘッドの移動距離が大きくなる．また，ファイル間のシークの回数も増大していくことになる．このような場合には，ハードディスクドライブの動作時間のうちシーク時間が大きな割合を占めるようになる．特に，複数のファイルを同時に操作する場合にはファイル間を頻繁にシークすることになる．この複数のファイル間のシークの回数を減らし，かつハードディスクドライブのヘッドの移動距離をできるだけ小さくすれば，全体のシーク時間を減らすことができる．

よって，本研究では以下に述べるスケジュール方法を提案する．

- ファイル毎に主記憶上にバッファを用意し，一度のシークでこのバッファを満たすまでファイルを読み込む．
- 複数のファイルをディスク配置順に読み込むことによってヘッドの無駄なファイル間の往復をできる限り避ける．

このように，本研究で実現するファイルシステムではそれぞれのファイルのハードディスク上の位置を優先的に考慮したスケジュールでファイルを操作する．

2.3　ファイルシステムの設計

ここでは，オープンした全ての連続メディアが途切れることなく再生できる条件を導く．ファイルは，File 1 ,File 2 , $\cdots$,File n の順でハードディスク上に格納されているとし，File $i(i=1,2,\cdots,n)$ のパラメータとして次のものを定める．

- データ要求周期 T_i
- 要求データ量 R_i
- バッファ量 B_i
- 読み込むデータが存在するトラックの位置 L_i

まず，トラック i からトラック j へのヘッドの移動時間 $S_{i,j}$ を求める．トラックートラック間のシーク時間を t_t とすると $S_{i,j}$ は次のようになる．

$$S_{i,j} = |i-j|t_t \qquad (1)$$

次に，x 個のセクタを読み込むために要する時間 $C(x)$ を求める．ハードディスクドライブのディスクが一回転するのに要する時間を t_r とし，1個のセクタを読み込むのに要する時間（ディスクの回転時間を含む）を t_s とすると，x 個のセクタを読み込む際のヘッドの移動時間は次のようになる．

$$t_t \left\lceil \frac{x}{n_s} \right\rceil \tag{2}$$

なお，n_s は1トラックあたりのセクタ数である．よって，x 個のセクタを読み込むために要する時間 $C(x)$ は次のようになる．

$$C(x) = t_r + xt_s + t_t \left\lceil \frac{x}{n_s} \right\rceil \tag{3}$$

次に，複数の連続メディアのファイルを File 1,File 2, $\cdots$,File n の順にハードディスクからデータを読み込む時間 r を求める．このとき，ハードディスクドライブの動作は次のようになる．まず，トラック0から File 1 の存在するトラックに移動し，File 1 をバッファを満たすまで読み込む．次に，File 1 の存在するトラックから File 2 の存在するトラックに移動し，File 2 をバッファを満たすまで読み込む．このように，File n まで読み込むと，ヘッドは再びトラック0に移動し，再び以上の動作を繰り返す．よって，r は次のようになる．

$$\begin{aligned} r &= S_{0,L_1} + C(B_1) + S_{L_1,L_2} + C(B_2) + \cdots \\ &\quad + S_{L_{n-1},L_n} + C(B_n) + S_{L_n,0} \end{aligned} \tag{4}$$

連続メディアの再生によって，File i のバッファが空になる時間は次のようになる．

$$\left(\frac{B_i}{R_i}\right) T_i \tag{5}$$

この時間内に，ハードディスクからデータを読み込み，バッファを満たせば連続メディアは途切れることなく再生できる．したがって，次に示す式を満たしていれば，全ての連続メディアの再生を途切れることなく行うことができる．

$$\min_{i=1,2,\cdots,n} \left(\frac{B_i}{R_i}\right) T_i \geq r \tag{6}$$

3 ファイルシステムの評価

ここでは，参考文献 [1] の連続メディアを考慮したファイルシステムと，本研究において実現したファイルシステムとの比較を行う．比較には [1] で評価に使用したハードディスクドライブ及び7個の連続メディアのファイルを使用する．本研究で実現したファイルシステムにおいて，[1] の7個の連続メディアのファイルをオープンした結果を表1に示す．

表1　実行結果

n	Filename	r (μ sec)	(6) 式の左辺 (μ sec)
1	sample_1_aud	85183	13000000
2	sample_1_vid	126866	2100000
3	sample_2_aud	164449	2100000
4	sample_2_vid	179732	2100000
5	cd_audio	305415	2100000
6	audio_1	361898	2100000
7	audio_2	374981	2100000

[1] のファイルシステムでは，同時にオープンできる連続メディアのファイルは4個である．しかし，本研究で実現したファイルシステムでは用意した7個全てのファイルをオープンすることが可能である．また表1より，さらに多くの連続メディアのファイルをオープンすることが可能であることがわかる．表1の連続メディアを全てオープンするために必要なバッファ量 B を表2に示す．

表2　必要なバッファ量 B

Filename	B (KByte)	[1] の B (KByte)
sample_1_aud	4	4
sample_1_vid	60	90
sample_2_aud	4	4
sample_2_vid	24	54
cd_audio	44	—
audio_1	4	—
audio_2	4	—

表2より，連続メディアのファイルのオープンを行うために必要なバッファ量を [1] のファイルシステムと比較すると，本研究で実現したファイルシステムの方が少ないことがわかる．

4 おわりに

本研究では，ユーザが連続メディアのファイルを一般のファイルと区別することなく操作することができ，かつ途切れることのない連続メディアの再生を可能とするファイルシステムのプロトタイプを実現した．本ファイルシステムを設計する際にハードディスクドライブの本来の性能を有効に引き出すように考慮したので，他の連続メディアを考慮したファイルシステムと比較してより多くの連続メディアを同時に再生することが可能となった．また，主記憶上に必要なバッファ量もより少なくすることが可能となった．

なお，本ファイルシステムは一般のファイルも同時に扱うことができるので，通常のファイルシステムと入れ替えて使用することも可能である．

参考文献

[1] K. Tindell, A. Burns: "SCHEDULING HARD REAL-TIME MULTI-MEDIA DISK TRAFFIC," YCS 204, Department of Computer Science, University of York (1993).