

## UNIX\*ファイルシステムのリアルタイムシステムへの適用

7P-2

藤間 貴久代<sup>1</sup>、水橋 由紀子<sup>1</sup>、市瀬 規善<sup>1</sup>、大塚 智司<sup>1</sup>、古城 隆<sup>2</sup>  
 1 日本電気マイコンテクノロジー(株)  
 2 日本電気(株)

## 1 はじめに

当社では32ビットマイクロプロセッサV60/V70用リアルタイムOS、RX-UXを開発している。RX-UXのファイルシステムはUNIXsystemVと互換性を持ち、UNIX処理とリアルタイム処理からの利用が可能である。一般にUNIXファイルシステムを組み込み型マイコン用のOSにそのまま適用すると、ファイルアクセス性能やファイルシステムの信頼性の問題があることが指摘されている。そこで、RX-UXではUNIXファイルシステムをリアルタイム処理の要求に応えられるようにファイルアクセス性能の改善を行った。ここではその連続ブロックファイル、及び連続ブロックアクセスの手法について述べる。

## 2 要求条件

## 2.1 ファイルアクセス性能

UNIXのファイルシステムではディスクブロックを効率的に管理するためにファイルを固定サイズ(1Kbyte)のブロックの集まりとして管理し、アドレスを管理するブロック(間接ブロック)が各ブロックをポイントする構造をとっている。そのために、ファイルアクセス時にはデータへのアクセスの前に間接ブロックのアクセスが必要になることもある(図1A<sub>1</sub>、A<sub>2</sub>)。また、UNIXではファイルのデータをブロックごとにI/O処理をしてアクセスしている。各I/O処理の間には磁気ディスクのヘッドのシーク時間やディスクの回転待ち時間などのハードウェアのオーバーヘッドがあるから、I/Oの回数が増えるほどディスクアクセスの時間は長くなってしまふ。また、I/O要求を分割するとほかのプロセスのI/O処理が間に入る可能性も出てくるのでさらに処理に時間がかかってしまふ。そこで、これらUNIXの方法よりも高速なファイルアクセスの方法を実現しなければならない。

## 2.2 ファイル管理構造の互換性

アプリケーションの生産性や移植性について考慮すると、アプリケーションからは従来と同じインターフェースでアクセスできるようにファイル構造をUNIXのもので統一することが重要である。

## 3 連続ブロックファイルと連続ブロックアクセスの実現

以上の要求からUNIXのファイルと構造互換の連続ブロックファイルを導入し、さらに連続した複数ブロックを一括してI/Oをする連続ブロックアクセス機能を実現した。

## 3.1 連続ブロックファイル

連続ブロックファイルはファイルのディスクブロックが連続的に配置されているので、アクセス時には先頭の物理ブロックアドレスとファイル内のオフセットから直接ディスクブロック番号を計算することができる(図1B)。間接ブロックへのアクセスが必要ないので、大きいファイルへも高速にアクセスできる。

## 3.2 連続ブロックアクセス

連続ブロックアクセス機能はファイルアクセス時のI/O回数を減らして高速な処理を実現する。連続ブロックファイルではディスクブロックが連続しているので、ファイル内の論理的に連続した領域はサイズに関係なく1回でアクセスできる。アプリケーションが連続ブロックファイルに対して1ブロック以上のアクセス要求を出した場合にも、1回のI/O要求にするので処理時間を短縮することができる。

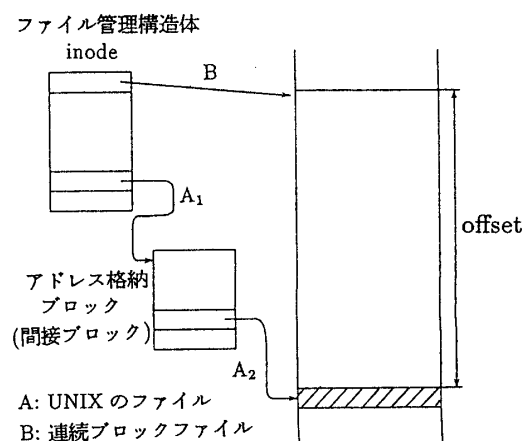


図1 UNIXのファイルと連続ブロックファイルのアクセス方法の比較

Adaptation UNIX file system to realtime processing

Kikuyo FUJIMA<sup>1</sup>, Yukiko MIZUHASHI<sup>1</sup>, Noriyoshi ICHINOSE<sup>1</sup>, Satoshi OHTSUKA<sup>1</sup>, Takashi KOJO<sup>2</sup>

1 NEC Microcomputer Technology Ltd. 2 NEC Corporation

\* UNIX オペレーティングシステムは AT&T が開発し、ライセンスしています。

### 3.3 データ構造

連続ブロックファイルは連続という属性を持つファイルとして位置づけている。この新しい属性を表わすデータをファイル管理構造体の構造を変化させずに追加することができた。

またアクセス時に参照はしないが、間接ブロックがブロックのアドレスを保持する構造も UNIX の通常ファイルと全く同じになっている。

### 3.4 連続ブロックファイルの作成方法

RX-UX では連続ブロックファイルの作成方法として、ファイルシステムの作成時に連続領域に配置する方法とシステムの稼働中に任意のサイズでダイナミックに作成する手段を提供する。ダイナミックに連続ブロックファイルを作成する場合、完全に連続なファイルとなるまでに時間はかかるので、アロケート処理中のファイルへもアクセスができるような方法を考えた。

#### 3.4.1 ブリアロケーション

ファイルシステムの作成時に連続領域にファイルを配置する方法。デバイスにファイルシステムを構築するユーティリティによって連続ブロックファイルを作成する。

#### 3.4.2 ディレイドアロケーション

すでに作成されたファイルシステムの中に連続的な領域を動的に確保する方法。

##### 1. インターフェース:

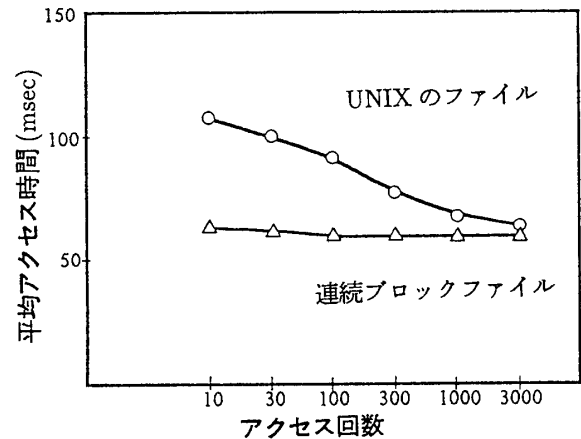
ファイルの生成時に指定する方法と既存のファイルの属性を変更するインターフェースを提供している。

```
exopen("file", mode, size);
excreat("file", mode, size);
exchmod("file", mode);
```

連続ブロックファイルを生成した後は、通常ファイルと同様に open, read, write などのシステムコールでアクセスができる。

##### 2. アロケート処理:

連続ブロックファイルを生成するシステムコールでは指定されたサイズの通常ファイルとして作成すると同時に、アロケート処理タスクを起動する。アロケート処理タスクは連続ブロックファイルのアロケートとブロック交換を1ブロックずつ行う専用のタスクであって、ファイルシステム内の連続ブロックファイルアロケート候補地を決定し、候補地内のブロックと連続ブロックファイルのブロックを交換する処理をしている。ファイルをすべて連続にするとファイルの属性を連続ブロックファイルに変更してアロケート処理は終了する。



PC上のシステムで測定

図2 連続ブロックファイルの効果

## 4 評価

30Mbyteの通常ファイルと連続ブロックファイルでランダムアクセスをしたときのアクセス時間の比較をおこなった(図2参照)。ここでは1バイトごとにアクセスをしているので連続ブロックアクセスの効果は入っていない。グラフでは、間接ブロックのアクセスがないために連続ブロックファイルの方がアクセス時間が短くなっている。しかし、頻繁にアクセスするとディスクキャッシュに間接ブロックがヒットする率が上昇するので両者の差は縮まってくる。

## 5 まとめ

以上のように連続ブロックファイルの機能を実現した結果、UNIXのファイルシステムのファイル構造の互換性を維持したまま、高速にアクセスができるファイルシステムを実現することができた。今後はディレイドアロケーションの性能の改善することが課題である。また、組み込み型のシステムの使用に耐え得る高信頼性のファイルシステムにするための機能拡張も考えていきたい。

### 参考文献

- [1] 下島他:V60リアルタイムOSのファイルシステム、情報処理第34回全国大会講演集1987 4Y-6
- [2] 下島他:V60/V70リアルタイムUNIX-概要-,情報処理第35回全国大会講演集1987 3D-3
- [3] 世良他:V60/V70リアルタイムUNIX-設計-,情報処理第35回全国大会講演集1987 3D-4