

DIROSの小型化と共存形態

6M-3

— ファイル管理の小型化 —

箱守 聰 伊藤 健一 遠城 秀和 谷口 秀夫

NTTデータ通信(株)

1. はじめに

計算機による各種サービスの増加に伴い、OSは高機能化、複雑化している。しかしこれらのサービスに対して、1つのOSを高機能化するのではなく、各サービスに最も適した機能構成を持つようなOSを同一体系内で複数作成する方が、より優れた環境を提供できる^[1]。

トランザクション処理用マルチプロセッサのOSであるDIROS^[2](Distributed Real-time Operating System)は、多種の機能を提供しているが、その反面巨大化、複雑化し、システム全体のオーバーヘッドの一因ともなっている。そこで筆者らは、主にファイル管理機能及びシステム制御機能を簡易化することで小型化、高速化し、かつDIROSと同一計算機上で走行する小型DIROSを作成した。

本稿では、小型DIROSのファイル管理機能について報告する。

2. DIROSの概要

DIROSは、OS核内部が機能ごとにモジュール化されており、各処理モジュール及びそれらが制御するデバイスは、識別子によって一意に識別されている。この識別子は、装置番号、装置内プロセッサ番号、及びプロセッサ内処理モジュール番号から成り、処理モジュール間の処理要求はこの識別子を用いて行う。

DIROSのファイル管理は以下の特徴を持つ。

- (1) 全ての資源をファイルとして仮想化し、それらを木構造により階層化して管理する。ディスク上のファイルは通常ファイル、デバイスに対応したファイルは特殊ファイルと呼ぶ。
- (2) ファイルの指定は、木構造を構成するディレクトリとファイル名との文字列から成るパスにより行なう。
- (3) ユーザ、グループ毎に7つのアクセス権を設定し、資源のアクセス制御を行なう。
- (4) 各種のアクセス法、及び排他制御機能を持つ。

3. ファイル管理小型化の方針

小型DIROSのファイル管理を検討する際の方針は以下の通りである。

- (1) ファイルをアクセスするために必要最小限の機能とし、アクセス制御機能、排他制御機能は削除する。
- (2) 応用プログラム(以下、APと略す)の作成を容易にするため、システムコールのインタフェースは、DIROSと同様とする。
- (3) APの開発やデバッグ等の作業を容易にするため、DIROSとディスク上のデータが共有できるようにする。

4. ファイル管理の機能検討

4.1 木構造の削除とファイル管理方式

DIROSでは、ファイルの木構造を用いて管理しており、ユーザに対して優れた資源管理の環境を提供している。この管理方式では、指定されたファイルのパス名に沿って検索処理が行われるため、以下の問題点があり、ファイルのオープンやクローズ時のオーバーヘッドとなっている。

- ① パス名途中にあるディレクトリの情報取得のため、ディスクアクセスが多発する。
- ② 1つのディレクトリ配下に多数のディレクトリが存在する場合、次のパスの検索が複雑になる。

小型DIROSには、処理の簡素化と高速化が求められており、ファイルへアクセスする最小限の機能として個々のファイルに直接アクセスできればよく、木構造による管理は必要ない。

木構造を用いないファイル管理方式として、次の2つが考えられる。

方式1: ファイルが管理テーブルとファイル実体から成り、管理テーブルを指定してファイル名とする方式

方式2: 管理テーブルを持たず、ファイルの実体であるディスク上のデータ領域あるいはデバイスを直接指定してファイル名とする方式

The Design and the Implementation of Reduced DIROS

Satoshi HAKOMORI, Ken-ichi ITOH, Hidekazu ENJO, Hideo TANIGUCHI

NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

表1 ファイル管理方式の比較

	長所	短所
方式1	<ul style="list-style-type: none"> アクセス制御機能あり ファイルの動的拡大機能あり 	<ul style="list-style-type: none"> ディスクアクセスが増える
方式2	<ul style="list-style-type: none"> アクセスが速い 	<ul style="list-style-type: none"> アクセス制御機能なし ファイルの大きさが固定

管理テーブルは、①ファイルの属性情報、②データ領域情報（通常ファイルの時）あるいはデバイスの識別子（特殊ファイルの時）、を持ち、この管理テーブルを必要とするか否かが2つの方式の相違点である。各方式の比較を表1に示す。

以下の理由により、管理テーブルの情報は不要と判断し、方式1よりもさらに処理の簡素化、高速化が実現できる方式2を採用した。

- 小型DIROSではアクセス制御機能を用いないため、ファイルの属性情報は不要である。
- 通常ファイルの場合、ファイルの使用に際し次の制約を設ければ、データ領域情報は不要となり、またこの制約は小型DIROSでは問題とはならない。

- ① データ領域を連続にとる。
- ② 領域はあらかじめ確保し、拡張しない。

DIROSにはデータ領域をあらかじめ連続的に確保する機能が提供されている。これを用いて、小型DIROSが使用するデータ領域をDIROS上であらかじめ確保すれば、この制約の上でも2つのOSでファイルの共有を行なうことができる。

- 特殊ファイルの場合、デバイスの識別子をシステムコール発行時に陽に指定しても問題ない。

4. 2 ファイル指定法

4. 1の方式によってファイルの管理を行なうと、各ファイルは以下の方法で指定できる。

- 通常ファイルの場合は、ファイルの実体があるディスクの識別子と、ディスク内におけるファイルの先頭位置
- 特殊ファイルの場合は、そのファイルが表わすデバイスの識別子

DIROSのシステムコールではファイル名をキャラクタ列で与えており、小型DIROSでもシステムコールインタフェースの整合性を保つために、キャラクタ列により指定する。

通常ファイルの場合の指定例を図1に示す。DIROSで/usr/home/file1のパス名で表される

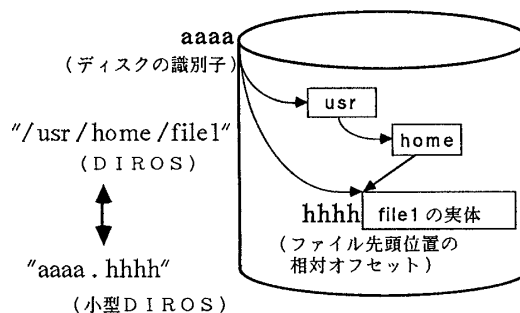


図1 DIROSと小型DIROSのファイル指定方法

ファイルは、小型DIROSではディスクのデバイス識別子aaaaと、ファイルの先頭までのオフセットhhhh（バイトで指定）を用い、“aaaa.hhhh”と表わす。

4. 3 システムコール機能

ファイル管理のシステムコールは、ファイル及びデバイスをアクセスするために最小限必要な機能として、5種のシステムコール（OPEN, CLOSE, READ, WRITE, IOCTL）を提供することとし、他はエラーリターンとする。

4. 4 他の処理モジュールの改造

ファイル管理の機能変更にともない、OS核内の他の処理モジュールでファイル管理機能を用いている部分を改造した。ファイル管理を基本機能と限定したことによる改造部分は少なく、主にファイルの指定法変更に関係した部分である。改造部分を以下に示す。

- プロセス生成時のプログラム読み込み
- プロセススワップ時のスワップ域アクセス
- プロセス、システム異常終了時の情報ファイル書き出し
- システムメッセージのログ収集

5. おわりに

小型DIROSのファイル管理機能について述べた。小型DIROSでは現在、立ち上げ処理終了後に、APを起動するためのユーザインタフェースとして簡単なコマンドモニタが動作している。

参考文献

- [1] 伊藤他：DIROSの小型化と共存形態；共存の形態，第39回情処全大(1989)
- [2] 谷口：OS機能の分散を可能にするOS構成法，信学論 Vol. J72-D-I, No. 3, pp.168-174 (1989)