

異OS間における分散ファイルシステムの構築

7N-6

谷口 秀夫 遠城 秀和 井村 佳弘

NTTデータ通信(株)

1. はじめに

ファイル管理機能をネットワーク化した分散処理OSは、UNIX-likeなOS間をサポートするNFS^[1]やRFS^[2]などのように、同種OS間をサポートした例が多い。

本稿では、リモートファイルアクセス機能^[3]を持つUNIXを搭載したマシン(以降、WS:Work Stationと略す)とトランザクション処理用の分散型リアルタイムオペレーティングシステム^[4](Distributed Real-time Operating System: DIROS)を搭載したマシン(以降、CS:Control Stationと略す)をLANで結合し、分散ファイルシステムを構築する方式について、CSの機能・メカニズムを報告する。

2. DIROSの概要

DIROSは、マルチマイクロプロセッサをサポートするトランザクション処理用OSである。異OS間の分散ファイルシステムを構築する際に関連するDIROSの特徴を以下に示す。

- (1) 磁気ディスクのファイルは文字列で識別し、通常ファイルと呼ぶ。
- (2) 入出力装置や通信回線は通常ファイルと同じように文字列で識別し、特殊ファイルと呼ぶ。
- (3) 通常/特殊ファイルは、ディレクトリにより木構造で管理する。
- (4) ファイル管理機能には、オープン・クローズ・データ入出力の基本アクセス機能以外に、アクセス権変更などの拡張制御機能がある。
- (5) トランザクション処理用に、処理の要求と処理の結果取得を別システムコールとする非完了システムコール機能を持つ。
- (6) UNIXのようなユーザフレンドリなインタフェースは持たない。

3. 実現する機能範囲3.1 UNIXマシンからの被アクセス

WSは、UNIXシステムシステムコールのうち、ファイル操作に関するものの大半(18個)をリモートファイルア

クセス機能として実現している。

DIROSは、以下のメリットにより、WSのリモートファイルアクセス機能と同じ機能を被アクセス機能として実現する。

- (1) WS側は、アクセスするファイルがWS上かCS上かにかかわらず同じ形式でアクセスできる。つまり、同じ応用プログラムでWSやCSにアクセスできる。
- (2) ファイル操作関連システムコールの大半をサポートすれば、UNIXのファイル操作コマンドのほとんどをCS上のファイルに対して利用できる。

3.2 UNIXマシンへのアクセス

UNIXファイル管理機能に比べ、DIROSファイル管理機能は高機能であるため、すべての機能をリモート化することは難しい。そこで、以下の基本アクセス機能をリモート化した。

- ・CONNECT: ファイルとアクセス識別子の連結
- ・DISCONNECT: ファイルとアクセス識別子の切離し
- ・OPEN: ファイルのオープン
- ・CLOSE: ファイルのクローズ
- ・READ: ファイルからのデータ読み込み
- ・WRITE: ファイルへのデータ書き出し
- ・CONTROL: ファイルの制御
- ・SEEK: ファイルのアクセス位置変更

UNIXは、入出力装置なども磁気ディスクのファイルと同様に木構造で管理しているため、上記機能だけでもかなりのリモート処理が可能である。

4. メカニズム

3章で述べた機能を実現するには、①DIROSシステムコールとUNIXシステムコールの相互変換法、②リモート/ローカル解析法を明確にする必要がある。各々について次節以降に述べるとともに、その処理メカニズムを記述する。

4.1 UNIXからDIROSへのシステムコール変換

UNIXシステムコールからDIROSシステムコールへの変換は、①リモートファイルアクセス機能提供とともに、②CS上でUNIXコマンドが使えることによるDIROSのマンマシンインタフェース向上を目的に、DIROS上のプロセスがUNIXシステムコールを利用できる形で実現す

Distributed File System on Heterogeneous Computing Environment

Hideo TANIGUCHI, Hidekazu ENJO, Yoshihiro IMURA

NTT DATA COMMUNICATIONS SYSTEMS CORPORATION

*: UNIXはAT&Tのベル研究所が開発したOSです。

る。

UNIXのシステムコールインタフェースを実現する方式^[5]として、以下の3案がある。

- {案1} ライブラリとして実現し、同一プロセス上でシミュレートする方式
- {案2} 別プロセスでシミュレートする方式
- {案3} OS核でシミュレートする方式

- ①UNIXコマンドをロードモジュールのまま利用でき、
- ②シミュレート速度が速い{案3}を採用する。

4.2 DIROSからUNIXへのシステムコール変換

DIROSシステムコールからUNIXシステムコールへの変換は、基本アクセス機能のみでよいため、①CONNECT&OPENをopen、②CLOSE&DISCONNECTをclose、③READをread、WRITEをwrite、④CONTROLをioctl、⑤SEEKをlseekに各々対応づけると、簡単なルーチンで変換できる。

4.3 リモート/ローカル解析

アクセス先がリモートかローカルかの判断は、ローカル処理とリモート処理のファイルパス名形式統一のため以下のようになる。

- (1) リモートに関する情報をネットワークディレクトリが持つ。
- (2) ファイルのパス名途中にネットワークディレクトリがあるとリモート処理と判断する。

4.4 通信メカニズム

本機能は、システムコール変換処理の以外に、①代行処理 ②LAN制御とのデータ送受信処理、③リモート/ローカル解析処理からなり、それらをどこで実現するかにより、表1に示す方式案がある。

表1. 通信メカニズムの方式案

方式案	1	2	3	4	5	6	7	8
代行処理	外	内	外	外	内	内	外	内
データ送受信処理	外	外	外	内	内	外	内	内
リモート/ローカル解析処理	外	外	内	外	外	内	内	内

内: OS核内で実現する
外: OS核外で実現する

以下の理由により、{案3}を採用する。概要を図1に示す。

- (1) ローカル処理とリモート処理の応用プログラムについてロードモジュール互換を保障するため、リモート/ローカル解析処理はOS核内で実現したい。
- (2) LAN以外のいろいろな通信路サポートを容易に実現できるようにするため、LAN制御とのデータ送受信処理はOS核外で実現したい。

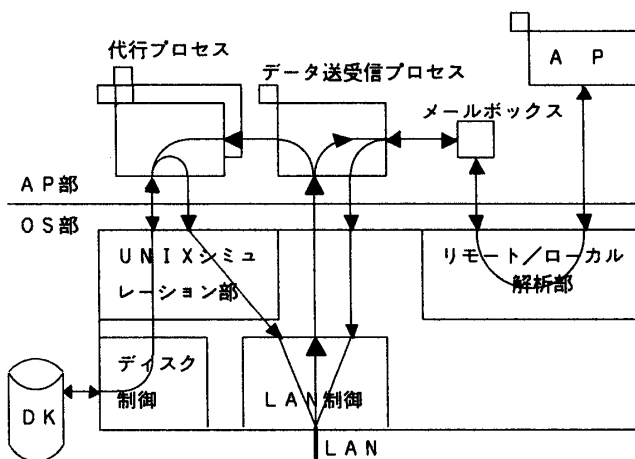


図1. 通信メカニズムの概要

- (3) ①複数のリモートファイルアクセス方式を同時にサポートし、②性能が許せば本機能実現時のデバグを容易にするため、できるだけOS核外で実現したい。

図1からわかるように、①代行プロセスは処理の要求と処理の結果取得が1つになった完了型のUNIXシステムコールを発行するため依頼元のユーザプロセス対応に存在するが、②データ送受信プロセスはROSの非完了システムコールを利用することにより1つのプロセスで実現して、多数のアクセス/被アクセス時のプロセス増加を最小限に押えた。

5. あとがき

異OS間の分散ファイルシステム構築例として、UNIXが走行するマシンとトランザクション処理用OSが走行するマシンをLANで結合する方式について報告した。

本機能により、UNIXが走行するマシンから、lsやcpなどのUNIXコマンドを用いてトランザクション処理用OSが走行するマシン上のファイルを操作できる。特に、スクリーンエディタが利用できるため大変便利である。

<参考文献>

- [1] Standberg, R., et al.: " Design and Implementation of the Sun Network Filesystem ", Proceedings of USENIX Summer Conference, USENIX Association, 1985
- [2] Rifkin, A. P., et al.: " RFS Architectual Overview ", Proceedings of USENIX Summer Conference, USENIX Association, 1986
- [3] 谷口、鈴木、瀬々: ファイル管理のネットワーク化による分散処理OSの構成法, 情処論文誌第27巻第1号, pp56-63, 1986
- [4] 谷口: 分散型リアルタイムOSの一構成法, 情処第34回全大, 4B-4, 1987
- [5] 遠城: 異種OS上におけるUNIXインタフェースの実現法, 情処第33回全大, 2V-2, 1986