UNIX上の有編成ファイル構築

3F－2

山川　直巳*，井手　俊一*，松井　浩二*
*（株）東芝

１．はじめに

近年の計算機システムのオープン化の流れの中でＵＮＩＸ†が標準的なＯＳとして定着しつつあるが、ビジネス用途に適用するには、従来システムからの移行性・耐障害性の機能強化が必要である。

ファイルシステムにおいては、有編成ファイル（索引順編成，相対編成，順編成）アクセス制御機能と、トランザクション機能が重要である。これをわれわれは、ミドルウェアとしてＵＮＩＸ上に実現した。

２．システム概要

有編成ファイルは、ＵＮＩＸファイルデータにサーバがレコード管理構造を作成・管理する方式で実現した。このため有編成ファイルとＵＮＩＸファイルの対応がつき、ファイル名の管理、コピー・セーブなどのファイル操作はＯＳの機能がそのまま使える。

図１にシステム全体図を記す。クライアントサーバ方式によりユーザプログラム異常動作からのデータ保護を図った。このときオーバヘッドの大きいクライアントサーバプロセス通信部には独自の方式を採った。これについて３．（プロセス間通信方式）に述べる。

またファイルシステム上に有編成ファイルを構築したために、カーネルとのキャッシングの重複と、同期書き込み時のINODE更新多発が問題となる。これについての対策を４．（バッファ管理方式）で述べる。

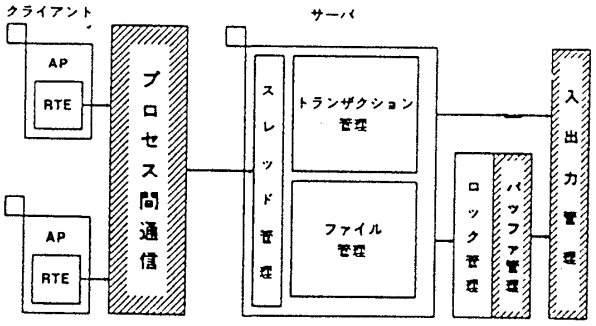

図１　システム全体図

３．プロセス間通信方式

クライアントサーバ通信部は、処理要求ＩＰＣ(1)メッセージ・終了ＩＰＣセマフォと、共有メモリにより実現した。

図２により、動作を説明する：

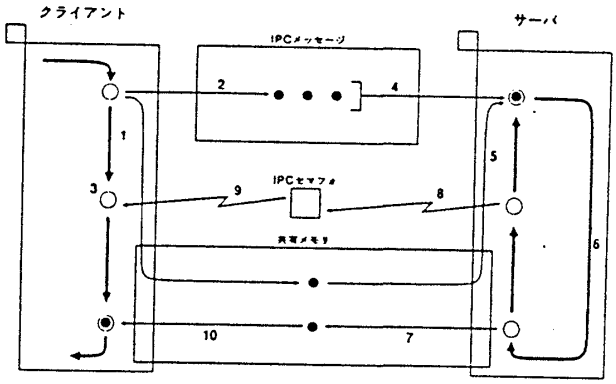

図２　プロセス間通信方式

［クライアント］
(1)要求データを共有メモリへ書き込む
(2)メッセージキューに要求をつなぐ
(3)セマフォを待つ
［サーバ］
(4)メッセージキューから要求を取り出す
(5)共有メモリから要求データを読む
(6)要求に従って処理を行う
(7)結果を共有メモリに書き込む
(8)セマフォを操作する
［クライアント］
(9)セマフォ待ちが解除される
(10)共有メモリのデータを読み取る

要求，応答データはクライアント毎に作成した共有メモリを介して交換する。これにより、データのコピー回数を減らせた。また他のクライアントからデータが保護でき、信頼性が向上する。

Developing Organized File System on UNIX
Naomi YAMAKAWA, Shun'ichi IDE, Kouji MATSUI
TOSHIBA CORPORATION

---

†ＵＮＩＸは UNIX SOFTWARE LABORATORY が開発しライセンスしているオペレーティングシステムです。

処理要求後クライアントは処理終了までブロックする。クライアントの異常終了は、ＩＰＣセマフォサービスのＵＮＤＯ機能[2]を用いて、サーバが検出できるようにした。

注釈（１）SystemVのﾌﾟﾛｾｽ間同期ｼｽﾃﾑｻｰﾋﾞｽ
（２）ﾌﾟﾛｾｽ終了時にｾﾏﾌｫ値を変化する機能

４．バッファ管理方式

トランザクション制御を行うためには，更新データをコミットまでメモリ上に保持し，コミット時に一度にディスク上へ反映するといった同期制御が必要である。

採用した方式では、Ｉ/Ｏ要求のためのＩ/Ｏバッファと、コミット前データを配置する更新バッファの２つのバッファを使用する。

Ｉ/Ｏバッファは、ＭＭＡＰ[3]による対象ファイルとのマップ空間である。ページ単位で管理され、ＬＲＵ制御される。マップ空間のサイズは、カーネル空間に制限されることなく十分大きく取れる。ファイルＩ/ＯはＩ/Ｏバッファへ該当ページのマッピング、データ転送というステップで行う。ＲＥＡＤ/ＷＲＩＴＥに比べて、ファイル制御部のＩ/Ｏなどのオーバヘッドが少なく高速となる。このバッファの内容はＯＳのメモリ制御により不定期にディスク上へ反映されるので、コミット前のデータは配置できない。その未確定データのためのバッファが更新バッファである。

このバッファ方式を図３に記した。これにより動作概要を説明する。

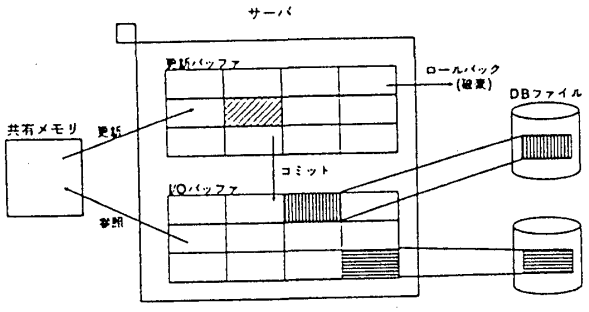

図３　トランザクションの更新処理

アプリケーションプログラムがファイルを更新する場合，まず更新バッファ上に領域を確保し，その領域へＩ/Ｏバッファからデータをコピーし、その後更新バッファ上の必要なデータを更新していく。

更新が終わりコミットが発行されると，ログを採取した後、更新バッファ上のデータをＩ/Ｏバッファへ転送し，該当領域のＳＹＮＣを行う。この際、不要なINODEの書き込みは起こらない。

更新をロールバックする場合は更新バッファ上のデータ破棄による。

システム障害発生時は、コミット処理時に取られるログにより対象ファイルの復元が可能である。このため、Ｉ/Ｏバッファから更新バッファへのデータ転送・ＳＹＮＣは遅延させることができる。ただしこの場合ロールバックは更新データの破棄ではできない。これに備えて更新ログをＳＹＮＣまで保持しておき、ロールフォワードにより復元する。

参照する場合はＩ/Ｏバッファを直接参照するので、COPY-ON-WRITE になっている。

なおトランザクション排他制御は、別のロック機構により行っている。

注釈（３）ファイルを仮想空間にマップする
システムサービス

５．　おわりに

本報告では，ＵＮＩＸ上に有編成ファイルアクセス法、トランザクション機構を実現する上で重要な、プロセス間通信方式、バッファ管理方式について紹介した。