

# マルチ OS におけるファイル共有方式

6M-1

八木橋 信一\*、森 加奈子\*、山内 秀明\*\*、伊波 通晴\*

\* 日本電気(株)C&Cシステムインタフェース技術本部

\*\* 日本電気マイコンテクノロジー(株)

## 1. はじめに

単一マシン上で複数の OS を共存させるシステムにおいて、OS 間での情報共有化機能は不可欠である。すでに実現した MS-DOS と UNIX の同時稼働環境<sup>[1][2]</sup>においては、キャラクタデバイスインタフェースによるデータ交換機能を用意したが、大量の情報を共有する汎用的な手段としてはファイル共有がもっとも望ましい。ここでは、MS-DOS と UNIX の同時稼働環境に適用したファイル共有方式について述べる。

## 2. ファイル共有方式

ファイル共有を適用する環境を図 1. に示す。ここではホスト OS として UNIX System V 3.0 が稼働し、ゲスト OS として MS-DOS 3.1 以降の版が稼働している。ゲスト OS は、Intel 80386 の仮想 86 モードによってエミュレートされており、ゲスト OS 環境による周辺装置へのアクセスは UNIX カーネルが監視している。この環境をエミュレーション環境と呼ぶ。

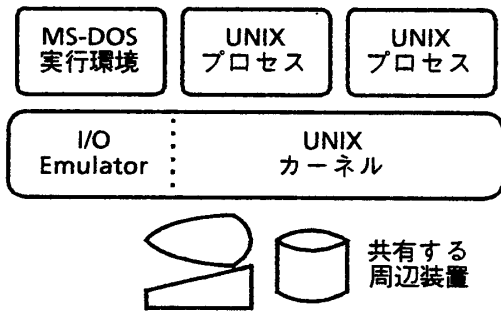


図 1 エミュレーション環境

ここでは、UNIX ファイルシステムの一部を MS-DOS のドライブとしてマウントすることによりファイル共有を実現し、どちらの OS からでも、共有ファイルに対して独自のアクセスができるようにする。

MS-DOS のファイルシステムに外部ドライブを接続するためには、ブロックデバイスドライバを記述する方法とシステムコールのリダイレクタを組み込む方法の 2 つがあるが、論理的なファイルアクセスを把握できるのは後者の方法である。

多くのネットワークシステムでは、このリダイレクタ方式が使われている。完全なリダイレクタを作成するためには MS-DOS の非公開機能を知らなければならないので、今回は MS-NETWORKS のリダイレクタを使用して開発を行った。

MS-DOS におけるネットワークのインプリメンテーションでは、ネットワークプログラム自身のオーバーヘッドにより、大量のメモリを必要とするアプリケーションには対応できないものが多い。この開発においては、占有メモリを小さくすることを設計上の課題とした。

UNIX 側は、MS-DOS ファイルサーバとして機能するデーモンを用意することで対応する。このデーモンの主な機能は、ネットワーク管理、システムコールマッピング、およびファイル名管理である。それぞれについて以下に述べる。

## 3. ネットワーク管理機能

占有メモリを最小化するため、エミュレーション環境では一般的な MS-NETWORKS のインプリメンテーションとは違った構成になっている。もっとも大きな違いは、トランスポート層を持たないことである。次に、OSI 参照モデルに対応する各層の機能について述べる。(図 2. 参照)

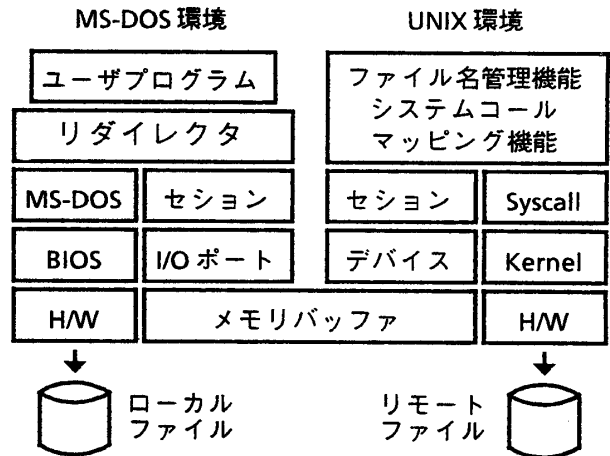


図 2 システム階層

A File Sharing System for Multiple Operating System Environment  
Shin-ichi YAGIHASHI\*, Kanako MORI\*, Hideaki YAMAUCHI\*\*,  
and Michiharu IHA\*

\* C&C Systems Interface Engineering Laboratory, NEC Corporation

\*\* NEC Microcomputer Technology Corporation

### 3.1 物理層とデータリンク層

エミュレーション環境における2つのノードは同一のシステム内に存在するため、物理層はメモリバッファで実現している。データリンク層は、UNIX側はスペシャルデバイスインタフェースを用いて構築されており、MS-DOS側は入出力ポートを介して転送制御するプログラムとして実現している。この入出力ポートへのアクセスをUNIXカーネルがトラップし、メモリバッファとの間の仮想的なDMA機能をエミュレートしている。

### 3.2 セッション層

MS-NETWORKSのリダイレクタを使用するため、セッション層プロトコルはMS-NETWORKSと同じものを用いている。ただし、MS-NETWORKSが仮定しているトランスポート層プロトコルは使用しないため、MS-DOS側のセッション層プログラムは独自のものを作成した。

UNIX側は、セッション層以上の機能をデーモンとして実現している。このデーモンは、セッション開始時に子プロセスを起動する。起動されたプロセスは、自分自身のUID、GID、およびUMASKの値をMS-DOSのエミュレーション環境を実現しているプロセスと同じ値に設定する。このためデーモンは、MS-DOSを起動しているユーザプロセスと同じアクセス権でファイルを操作できる。

### 3.3 アプリケーション層

MS-DOS側のアプリケーション層は、MS-NETWORKSのリダイレクタを使用する。UNIX側のデーモンは、リダイレクタが用いるファイル共有プロトコルに従い、サーバとしての機能を果たす。

## 4. システムコールマッピング機能

リダイレクタから渡されたファイルアクセス要求は、デーモンによって解釈され、UNIXシステムコールの発行によってアクセスが実現される。ただしファイル属性については、OS間に非互換な点が多く、すべての機能をサポートすることはできない。MS-DOSファイルのレコードロック機能については、UNIX SystemV 3.0でサポートされるロック機能を用いて実現しているが、一部の点が非互換である。

## 5. ファイル名管理機能

UNIXのファイル命名規則とMS-DOSの命名規則の違いを吸収するため、デーモンはファイル名

の変換機能や管理機能を持つ。たとえばUNIXのtest.cというファイル名はMS-DOSのTEST.Cという名前に対応づけられる。しかし、対応がつかないような名前については独自のファイル名を生成して対処する。このファイル名生成機能はダイナミックに稼働するため、あるファイル名が常に同じ名前にマッピングされるとは限らない。

## 6. おわりに

現在のファイル共有機能および他のネットワークファイルシステムの占有メモリ量を表1.に示した。最小限の機能だけをMS-DOS環境に残し、多くの部分をUNIX環境で実現したため、占有メモリ量を大幅に減らすことができた。

また、ローカルファイルとリモートファイルを比較した転送性能を表2.に示した。データ転送性能も実用的なレベルであることがわかる。

エミュレーション環境のファイル共有システム自身は単一のシステムに閉じているが、NFSがインストールされたシステムでは、実際のネットワークを介したりリモートファイルにアクセスすることもできる。

## 参考文献

- [1] 八木橋, 真鍋: V60 マルチ OS の実現, 第 36 回 情報処理学会全国大会, 7D-4, 1988.
- [2] 八木橋, 真鍋: 流通ソフトウェアを考慮したマルチ OS, 情報処理学会オペレーティングシステム研究会, 39-1, 1988.

表 1. 占有メモリ量 (PC-9801 で測定)

MS-NETWORKS (BRANCH 4670)	87.5 K バイト
PC-NFS (BRANCH 4680)	89.7 K バイト
本システム	26.3 K バイト

表 2. 転送性能 (PC-9801 で測定)

① 1M バイトファイルのコピー	
ローカル間 (リアルモード)	25 秒
ローカル間 (エミュレーション)	22 秒
ローカル → リモート	29 秒
リモート → ローカル	37 秒
② アプリケーション (一太郎 Ver.3) の起動	
ローカル (リアルモード)	7 秒
ローカル (エミュレーション)	10 秒
リモート	13 秒

UNIX は米国 AT&T で開発され、ライセンスされている。  
MS-DOS および MS-NETWORKS は米国マイクロソフト社の登録商標である。  
NFS および PC-NFS は米国サンマイクロシステムズ社の登録商標である。  
Intel は米国インテル社の登録商標である。  
一太郎は株式会社ジャストシステムの登録商標である。