Vol. 27 No. 1 情報処理学会論文誌 Jan. 1986

ファイル管理機能のネットワーク化による分散処理OSの構成法†

谷口秀夫†† 鈴木達郎†† 瀬々正治††

LSIを中心とする急速なハードウェアの進歩により，LAN (Local Area Network) のような高速伝送路で処理装置を有機的に結合したシステムの構築が進められている．このようなシステムに必要な分散処理OS (Operating System) には，リモートのファイルに簡単にアクセスでき，既存のローカル処理用プログラムをそのままネットワーク処理にも利用できる高度な通信機能が必要とされている．このような機能を有するリモートアクセス方式による分散処理OSは，すでにいくつか実現されているが，高速性は生かされているものの放送機能は生かされておらず，十分とは言えない．本論文では，高速性と放送機能を生かしたリモートアクセス方式による分散処理OSが提案された．具体的には，①分散したファイルを統一管理する方式，②一つのファイルだけではなくファイルの集合に対する一括アクセスを従来の応用プログラムインタフェースを保存して実現する方式，③リモートアクセスの処理要求と処理結果を通信する制御方式，が示された．また，UNIXでのインプリメント例により，リモートアクセスはローカルアクセスに比べ数倍のアクセス時間になることが示された．

1. はじめに

現在，ローカルエリアネットワーク (LAN) 等の通信路で結ばれた処理装置（以降ノードと呼ぶ）を有機的に結合したシステムの構築が行われている．このようなシステムで，通信路の特徴を生かしたサービスを提供するには，各ノードで走行するオペレーティングシステム (OS) がネットワーク機能をもつ分散処理OSである必要がある．

一方，LANを利用した分散処理OSを構築する上での基本OSとしては，①入出力機器を仮想化しファイルと統合して一つの木構造で管理しており，②シェル等により優れたプログラム開発環境を提供しているため蓄積されたソフトウェアが多い，等の利点を有するUNIX* が有効と思われる．

本論文では，LANの高速性と放送機能を利用したサービス（例えば，複数のノードで同じ画面を見ながら作業を行うことができる電子会議サービス）を実現するために，ファイル管理機能をネットワーク化したリモートアクセス機能について述べる．リモートアクセス機能とは，他ノードにあるリモートファイルに自ノードのローカルファイルアクセスと同じインタフェースでアクセスできる機能である．具体的には，本機能を実現する上での基本メカニズムである，①分散しているファイルを統一管理する方式，②資源の集合（グループファイルと呼ぶ）へのアクセス方式，③アクセス要求の転送方式，の構成法，およびこれらをUNIXに付加する際の実現法や評価について報告する．

2. 本分散処理OSの特徴

分散処理OSを構築するには，従来から，①HOSTによるmaster-slave方式や，②新規にネットワークOSとして設計する方式などがある．しかし，「既存のアプリケーションプログラム (AP) をそのまま利用できる」ためには，各ノード上のOSに自律性をもたせたままリモートへの要求のみを他ノードへ転送するという機構を追加する方式が良い．リモートアクセスの動作例を**図1**に示す．

この方式による分散処理OSのリモートアクセス機能の実現においては，サービスプログラムの作成を容易にする等のために，以下の条件を満足しなければならない．

（1） 対APインタフェースがローカルにあるファイルアクセスと同形式であること．

（2） グループファイルへのアクセスがローカルと同様に一つの命令で一度に可能なこと．

（3） プロセス制御などのファイルアクセス以外への拡張性があること．

（4） アクセス速度が速いこと．

これらの要求を満足するためには，①ネットワーク内に分散しているファイルを統一管理する方式，②グ

† A Distributed Operating System Based on Networking the File Management Mechanism by HIDEO TANIGUCHI, TATUO SUZUKI and MASAJI ZEZE (NTT Electrical Communications Laboratories).

†† NTT 電気通信研究所

* UNIX は AT&T のベル研究所が開発した OS.

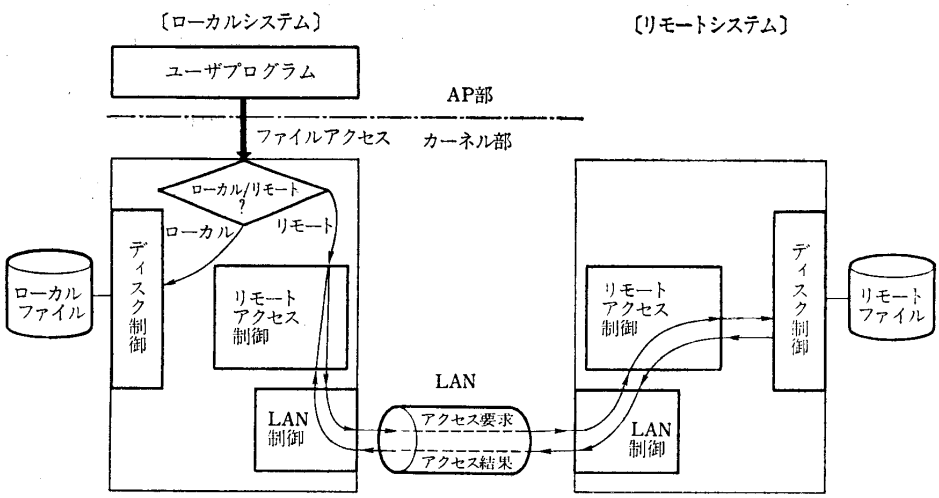


図 1　リモートアクセスの動作例（ディスク I/O の場合）
Fig. 1　Example of remote access execution for disk.

ループファイルへの同時アクセス方式，③リモートへのアクセス要求転送方式，を実現する必要がある．

UNIX をベースとして，上記方式の実現を図ったものとしては，COCANET[1]，ALTOS-NET II[2]，LOCUS[3]，Newcastle Connection[4]，NMSU[5]などがあり，これらの比較は文献4)が詳しい．これらの OS では，LAN の高速性を生かした一対一のリモートアクセスは実現されているが，放送機能を生かした一対複数のリモートアクセスは実現されていない．

本 OS の特徴は，LAN の一つの特徴である放送機能を利用して，一対複数のリモートアクセスも実現したことである．具体的には，UNIX のディレクトリファイルの延長として「グループネットワークディレクトリ」の概念を導入し，グループファイルに対するリモートアクセス手段を実現した．これにより，ローカルな環境を想定して作成されたプログラムに手を加えずに，そのまま電子会議などの放送形機能を必要とするサービス[6]にも適用できる．

3. リモートアクセス機能の構成法

3.1 分散したファイルの統一管理方式

各ノード上で作成されたファイルについて，対 AP インタフェースを保存してネットワーク内で一意に管理するため，ファイルの先頭に，①ネットワーク化したことを示す識別子（グローバル識別子と呼ぶ）と，②そのファイルが存在するノードに対応した特定の名前（ノード識別子と呼ぶ）を付加する．

これにより，ネットワーク内の全ファイルは，グローバル識別子の下で一つに管理できる．AP からのファイルアクセス要求に対し，OS 内部で AP が指定したファイル名に基づきそのファイルが存在するノードを識別できる．

3.2 グループファイルへのアクセス方式

グループファイルへのリモートアクセス機能を実現するため，複数のノードからなるグループに対応するノード識別子を導入する．これをグループノード識別子と呼ぶ．一方，一つのノードに対応するノード識別子は単一ノード識別子と呼ぶ．グループノード識別子の導入により，AP はグループファイルへのリモートアクセス処理を単一ファイルへのリモートアクセス処理の場合と同じ形で行える．

グループファイルへの出力系のアクセスは，すべて放送形のアクセスとなり，グループに属する全ノードにアクセス内容を転送し，実行する．つまり，一つのアクセス要求で複数のファイルにアクセスできる．アクセス結果の返送は，以下の問題があるため，行わない．

（1） アクセスを要求したプロセスは，アクセス結果を受け取るため，グループ内のノード数がわからないと終了時点が判断できない．

（2） 通信のプロトコルが複雑化して，アクセス速度の低下を招く．

一方，グループファイルからの入力系アクセスは，アクセス要求をグループ全員に転送して実行すると，グループ全員から入力結果が返送されてくる．これを積極的に用いることも考えられるが，この場合には上記（1）（2）に加え以下の問題がある．

（3） 対 AP インタフェースが，グループファイル

へのリモートアクセスと単一ファイルへのリモートアクセスで異なる.

これらの問題を解決するためには,特定の一つのノード(入力ノードと呼ぶ)のみがアクセスを実行して結果を返すようにした方がよい.したがって,入力ノードの指定が必要になり,その方式としては「既存の対APインタフェースを守って,放送形サービスのプログラム作成を容易にする」ために,アクセスとは独立に入力ノードを指定する独立指定方式を採用する.

3.3 アクセス要求の転送単位と通信方式

〈転送単位〉 他ノードへ転送するアクセス要求の単位として,OS内の情報(例えば,ファイルの識別できる管理テーブルのid),や対APインタフェースとなるシステムコール,さらにはライブラリ関数,などいくつかのレベルが考えられる[2]が,次の理由で「システムコール転送」のレベルが優れている.

(1) ローカルな環境で作成したソフトウェアを完全にそのままネットワーク環境でも実行できる.

(2) 通信のオーバヘッド(回数)が小さい.

(3) 既存のOSの改造量が小さい.

具体的には,OS内で各システムコールがローカルかリモートかを判断した後,リモートアクセス制御を用いて相手ノードにアクセス要求を転送する.相手ノードでは,要求に基づいてアクセスを代行実行後,その結果を返送する.

〈通信方式〉 リモートアクセスの通信制御手順は,通信速度の向上と機能の拡張性の点から,次のようにする.

(1) 機能拡張が容易な次の二レベル構成とする.

(A) リモートアクセス制御レベル:アクセスを代行するプロセス(代行プロセスと呼ぶ)とのパスを制御する.

(B) リモートアクセス転送レベル:アクセスするファイル対応のパスを制御し,リモートアクセス内容を転送(システムコール転送)する.

(2) 構成が容易な同期型(1次局・2次局の構成)とする.

(3) LANの物理的信頼度が高くパケット異常は少ないことから,正常系の速度向上を重視して,単一リモートアクセスではコマンド(アクセス要求のパケット)に対応するレスポンス(アクセス結果のパケット)をタイマ監視して通信での異常を検出し,グループリモートアクセスではレスポンスはなく通信での異常はない方式とする.

(4) パケット数を減らし,かつ代行プロセスのI/O回数も減らすために,リモートアクセス制御レベルでセグメンティング機能をもつ.

4. UNIXにおける実現法[7),8)]

4.1 グローバルトリーの導入

UNIXのファイル[9)]はトリー構造で管理されており,以下のように分類される.

(1) パス名をもつディレクトリ

(2) 入出力機器を仮想化した特殊ファイル

(3) データをもつ通常ファイル

ファイルはすべてテーブルで管理されており,ディレクトリや通常ファイルについては,管理テーブルがそのファイル実体へのポインタをもっている.

UNIXファイルシステムをネットワークに拡張するため,分散処理OSの下では,ネットワーク内の全資源を,①グローバル識別子としての「ネットワークルートディレクトリ("//"と記述する)」,②単一ノード識別子としての「単一ネットワークディレクトリ」,③グループノード識別子としての「グループネットワークディレクトリ」,の導入により,**図2**に示すような一つのグローバルトリーで管理する.

図2は,単一ネットワークディレクトリ n_0 に対応するノードにおけるグローバルトリーの例を示したものである.図2において,n_0, n_1, n_2 は単一ネットワークディレクトリであり,対応する各ノードにおけるUNIXファイルシステムのルートディレクトリにあたる.

グローバルトリー導入におけるディレクトリのファイル実体管理は,次の理由から,「各ノードが,自ノー

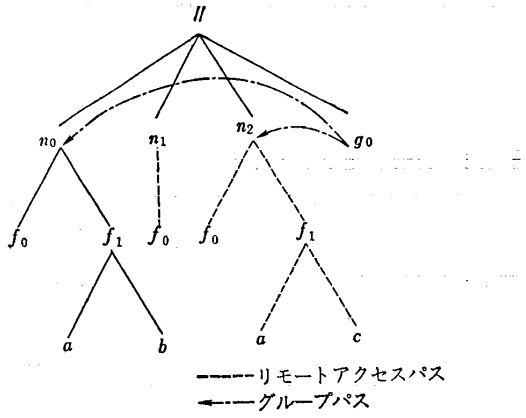


図2 グローバルトリーの例(ノード n_0 上)
Fig. 2 Example of global tree at node n_0.

ドにある通常／特殊ファイルへのパスとなっているディレクトリをもつ」自己管理方式が優れている．

（1） 更新が容易．

（2） ローカルファイルへのアクセスは，従来のアクセス速度が維持できる．

（3） ファイルを管理する特別なノードが不要．

図2において，実線で結ばれたファイルの実体が，単一ネットワークディレクトリ n_0 に対応するノード上に存在する．単一ネットワークディレクトリ n_1, n_2 は，対応するノードにあるファイルへのアクセスパスとなる．なお，グループネットワークディレクトリ g_0 に関しては，次節で述べる．

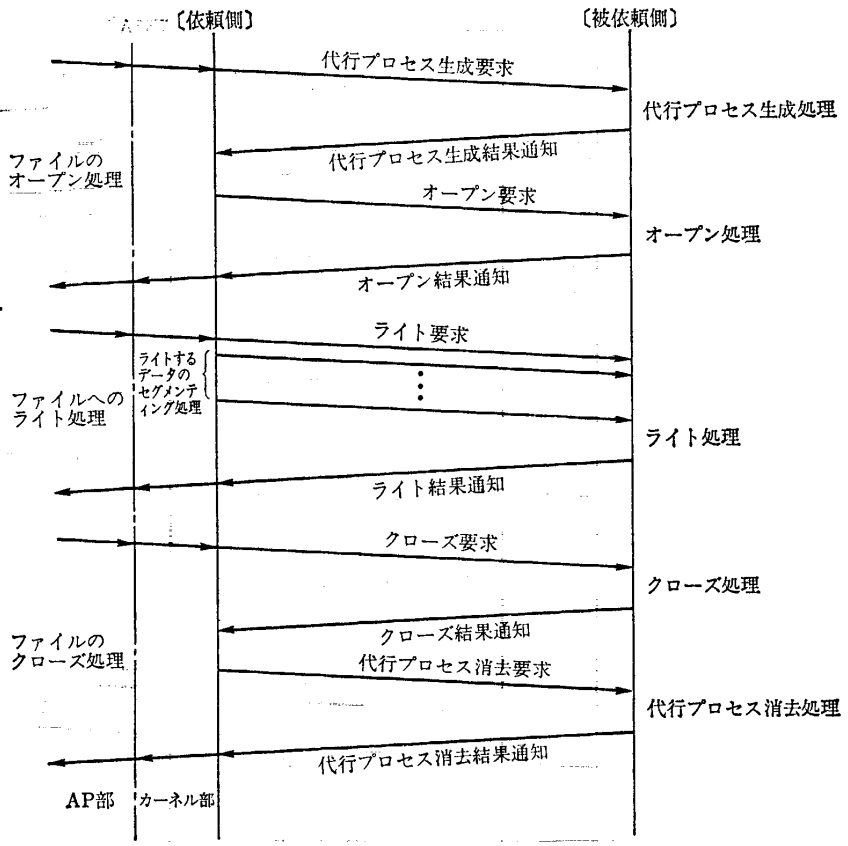


図 3 単一リモートアクセスのシーケンス例
Fig. 3 Example sequence of single remote access.

4.2 グループリモートアクセス機能の実現

グループファイルへのリモートアクセス機能を実現するため導入したグループネットワークディレクトリは，LAN で定義されるグループアドレスと一対一に対応させ，グループネットワークディレクトリを作成することにより，“自ノードがそのグループに属する” という機能も兼ねる．

例えば図2において，g_0 は n_0 と n_2 に対応するノードからなるグループネットワークディレクトリとすると，g_0 を介した出力系のアクセスは，“$//g_0/f_1/a$” のパス名指定により，n_0 と n_2 に対応する両ノード上のファイル “f_1/a” に「一度に」アクセスできる．

read システムコールなどの入力系アクセスは，入力ノードの指定状態に基づき，“$//g_0/f_1/a$”のパス名指定により，n_0 または n_2 に対応するノード上のファイル “f_1/a” にアクセスできる．

入力ノードを指定するシステムコールの例を**表1**に示す．

表 1 入力ノード指定のシステムコール例
Table 1 Example of systemcall to set the input node.

形　式	setnod (path 1, path 2)
パラメータ	char *path 1; グループネットワークディレクトリ名へのポインタ char *path 2; 単一ネットワークディレクトリ名へのポインタ
機　能	path 1 で指定されるグループネットワークディレクトリがもつグループの入力ノードを path 2 で指定される単一ネットワークディレクトリに対応するノードに設定する．

4.3 アクセス要求の転送方式

リモートアクセスの通信制御処理は，処理速度向上のため OS 内で実現した．

単一リモートアクセスによるオープン/ライト（セグメンティングあり）/クローズのシーケンス例を**図 3** に示し，グループリモートアクセスによるオープン/ライト（セグメンティングなし）/入力ノード設定/リード（セグメンティングあり）/クローズのシーケンス例を**図 4** に示す．

各レベルの内容を以下に示す．

〈リモートアクセス制御レベル〉 相手ノード上の代行プロセスとのパスを制御する．本レベルのパケット形式を**図 5** に示し，構成要素について以下に説明す

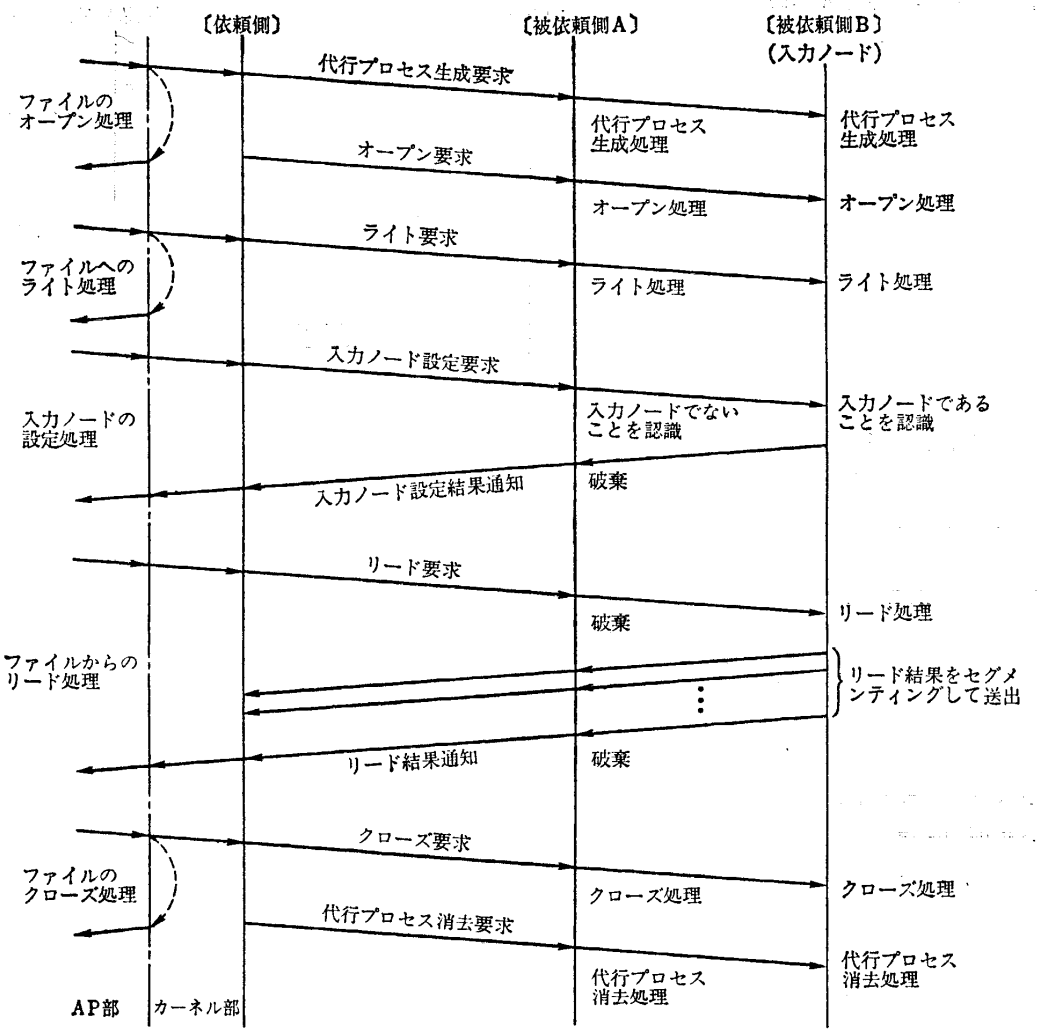
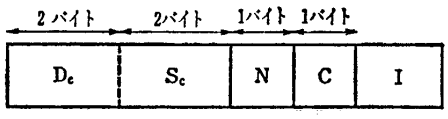
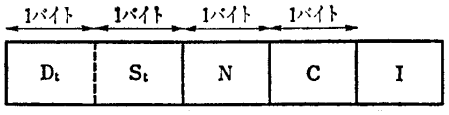


図 4 グループリモートアクセスのシーケンス例
Fig. 4 Example sequence of group remote access.



Dc, Sc：パス番号
N ：シーケンス番号
C ：制御部
I ：情報部

図 5 リモートアクセス制御レベルのパケット形式
Fig. 5 Packet format of remote access control level.



Dt, St：パス番号
N ：シーケンス番号
C ：制御部
I ：情報部

図 6 リモートアクセス転送レベルのパケット形式
Fig. 6 Packet format of remote access transfer level.

る．

（1） パス番号：通信パスを識別する．

（2） シーケンス番号：コマンドとレスポンスの対応づけやパケット欠落を監視する．

（3） 制御部：セグメンティング機能をサポートするチェイン情報をもつ．さらに，コマンドとレスポンスを識別し，①代行プロセスの設定／解放，②リモートアクセス中プロセスの fork に伴う通信パスの分離，③情報転送，④グループに対する入力ノードの設定を指示する．

（4） 情報部：コマンドやレスポンスで必要となる情報をもつ．

〈リモートアクセス転送レベル〉 AP からのリモートアクセス要求に基づき，アクセスするファイル対応のパスを制御して，アクセス情報の転送を行う．本レベルのパケット形式を**図 6** に示し，構成要素について以下に説明する．

（1） パス番号：通信パスを識別する．

（2） シーケンス番号：コマンドとレスポンスの対応づけやパケットの欠落を監視する．

（3） 制御部：コマンドやレスポンスを識別し，open や write などの 14 種類のシステムコールに対応したアクセス要求や結果を示す．

（4） 情報部：コマンドやレスポンスで必要となる情報をもつ．

5. 評価と考察

UNIX システムⅢをベースとした，リモートアクセス機能をもつ分散処理 OS を，MC 68000 (8 MHz) を CPU とするノード上で走行させ，評価した結果について述べる．伝送路は，伝送速度 10 Mbps のイーサネット形 LAN を利用した．

5.1 アクセス速度

ローカルファイルアクセスとリモートファイルアクセスのアクセス速度を比較する．アクセス速度比を次のように定義する．

$$\text{アクセス速度比}=\frac{\left(\begin{array}{c}\text{リモートファイルアクセス}\\ \text{に要する時間}\end{array}\right)}{\left(\begin{array}{c}\text{ローカルファイルアクセス}\\ \text{に要する時間}\end{array}\right)}$$

ファイルアクセスに関するシステムコールを，その処理内容から次の 4 種類に分ける．

（A） ファイルへのアクセス開始を示すもの．例えばファイルのオープン処理．

（B） ファイルへデータの入出力を行うもの．例えばファイルへの出力処理．

（C） ファイルへのアクセス終了を示すもの．例えばファイルのクローズ処理．

（D） ファイルの状態を制御するもの．例えばファイルのアクセス権変更処理．

測定は，測定ツールプログラムのみが走行する状態で行い，「time」コマンドを利用した．各種別のアクセス速度比を**表 2** に示す．表 2 から，次のことがわかる．

（1） 項目(D)から，リモートアクセスの処理は，代行プロセスの生成や消去の処理にかなりの時間を費やしている．項目(D)の場合は，代行プロセスの生成や消去の処理が，処理全体の約 3/4 を占めている．

（2） リモートファイルへの出力は，出力データ長が短いほどオーバヘッドは小さい．これは，パケットの送受信処理を伴うため，ローカル処理に比べ出力データのメモリ間転送回数が多いことによる．

（3） ディスク I/O を伴わない処理（アクセスするファイルのデータがすでにメモリ上にある場合やファイルのクローズ処理）は，ローカル処理時間が短いため，オーバヘッドが全体に占める割合は大きい．

5.2 セグメンティング機能の効果

リモートファイルに対する入力と出力に関して，セグメンティング機能の効果について述べる．LAN のパケットサイズは最大 1.5 キロバイトであり，セグメンティングを必要としない 1 キロバイトの入出力に要する時間と，セグメンティングを必要とする n キロバイト（$n=2, 3, \cdots, 10$）の入出力に要する時間の比を**図 7** に示す．

図 7 から次のことがわかる．

（1） 4 キロバイトまではセグメンティング機能の効果がある．しかし，4 キロバイト以上については，セグメンティング処理によるオーバヘッドとパケット送出処理の削減が相殺され，それ以上の効果はない．

（2） 出力処理より入力処理の方が効果が大きい．

表 2 アクセス速度比
Table 2 Access speed ratio.
（ローカルアクセスを 1 としたときのリモートアクセスの相対比）

項目 / 場合	(A)	(B)		(C)	(D)	
		ファイルへの出力処理			ファイルのアクセス権変更処理	
	ファイルのオープン処理	アクセスデータ長が 128 バイト	アクセスデータ長が1024バイト	ファイルのクローズ処理	代行プロセス生成が必要	代行プロセス生成が不必要
ディスク I/O を伴う場合	5	3	10	—	20	5
ディスク I/O を伴わない場合	20	10	35	200	20	6

—：存在しない

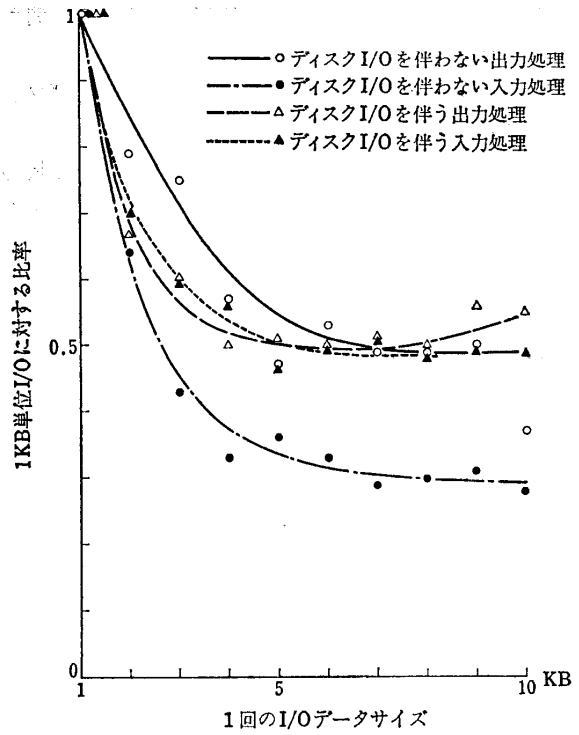


図 7 セグメンティング機能の効果
Fig. 7 Effect of segmenting process.

これは，セグメンティングのブロック化処理が，出力処理では異なった論理メモリ空間（AP 空間と OS 空間）の間で行われ，入力処理では同一の論理メモリ空間（OS 空間）の中で行われるため，前者は後者に比べメモリ間転送に時間を要することによる．

5.3 グループリモートアクセス機能の有効性

複数のファイルに対して，グループリモートアクセスを利用した場合と単一リモートアクセスをくり返し利用した場合の処理時間比を，次のように定義する．

$$\text{処理時間比}=\frac{\left(\begin{array}{l}\text{グループリモートアクセスを利用}\\\text{した場合の所要時間}\end{array}\right)}{\left(\begin{array}{l}\text{単一リモートアクセスを利用した}\\\text{場合の所要時間}\times\text{ノード数}\end{array}\right)}$$

リモートファイルへの出力処理について，結果を図8に示す．図8に示すパケット送出制御時間とは，LAN 制御がパケット送出の割合を制御するパラメータであり，具体的には，パケット送信要求とパケット伝送路送出の間に行う遅延処理の時間である．この時間は，LAN 制御の受信処理能力が送信処理能力より劣るために必要である．したがって，コマンドとレスポンスが対になっている単一リモートアクセスでは不要な処理である．図8から，次のことがわかる．

（1） リモートアクセスに要する時間が短いほど，グループリモートアクセス機能は有効である．

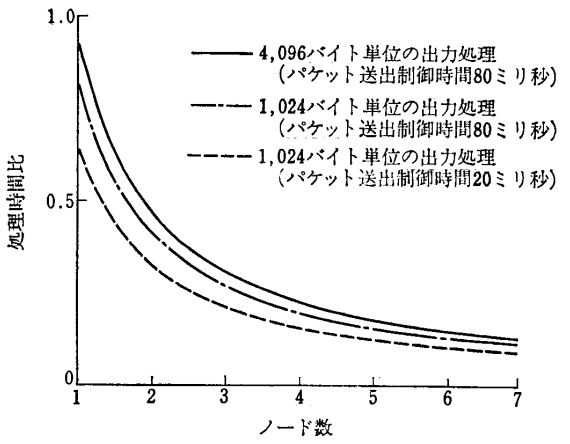


図 8 グループアクセス機能の有効性
Fig. 8 Effect of group access process.

（2） LAN 制御の性能が，グループリモートアクセス機能の有効性に大きな影響を与える．

伝送路の信頼性が高い LAN などの場合，グループリモートアクセスの信頼性は LAN 制御の性能に大きく影響される．例えば，1024 バイトデータのリモートファイルへの出力処理では，パケット送出制御時間が 20 ミリ秒の場合，パケット欠落は発生しないが，10 ミリ秒の場合，約1割のパケット欠落が発生した．したがって，レスポンスのないグループリモートアクセス機能の利用においては，次の点を考慮する必要がある．

（1） LAN 制御の性能が高いこと．

（2） パケット欠落を他の手段で発見できること．

例えば，ファイルへのアクセスでは単一リモートアクセスと組み合せて再確認し，ディスプレイ表示では目視確認できるサービスに適用する．

例えば，グループリモートアクセス機能を利用して電子会議サービスを実現すると，単一リモートアクセス機能を利用した場合に比べ数倍の応速速度が参加人数に関係なく保証できる．ただし，信頼性が低いため，会議参加者が異常を確認できるディスプレイ表示などに適用し，電子会議サービスの機能として，再表示機能が必要である．

6. ま と め

分散したファイルを統一管理し，リモートファイルへのアクセス依頼をシステムコール転送の形で行うリモートアクセス方式による分散処理 OS の構築を提案した．本方式は，LAN の放送機能を用いたグループファイルへのリモートアクセス機能が特徴である．

UNIX システムⅢをベースに実現した分散処理 OS

の場合，単一リモートアクセスはローカルアクセスに比べ数倍のアクセス時間が必要である．一方，グループリモートアクセスの性能は，下位の通信モジュールの性能に左右されるが，単一リモートアクセスに比べ約5倍の能力をもつ．評価は，伝送路として伝送速度10 Mbps のイーサネット形 LAN を利用したが，実用システムにおいては提供サービスに応じた伝送路の選定が必要である．

ローカルアクセスの利用環境と同じ形態でリモートアクセスの利用環境を提供するため必要な技術を，以下にまとめる．

（1） ファイル名の先頭に，ネットワーク化したことを示すグローバル識別子と，そのファイルが存在するノードに対応する単一ノード識別子を付加して，ローカルのファイル構成をそのままネットワーク化し，分散しているファイルを統一管理した．

（2） LAN の放送機能を利用して，複数のノードに対応するグループノード識別子の導入と，アクセスとは独立して入力処理を行うノードを指定する機能により，アクセス速度が速いグループリモートアクセス機能を自然な形で実現した．

（3） リモートへのアクセス要求の転送単位をシステムコールとし，機能拡張性のあるリモートアクセスの通信制御手順を確立した．

参考文献

1) Rome, L. A. and Birman, K. P.: A Local Network Based on the UNIX Operating System, *IEEE Trans. Softw. Eng.*, Vol. SE-8, No. 2, pp. 137-146 (1982).
2) Kavaler, P. and Greenspan, A.: Extending UNIX to Local-area Networks, *Mini-micro Systems*, pp. 197-202 (Sep. 1983).
3) Walker, B., Popek, G. et al.: The LOCUS Distributed Operating System, Proc. of 9th ACM Symposium on Operating Systems Principle, pp. 49-70 (1983).
4) Brownbridge, D. R. et al.: The Newcastle Connection, *Softw. Pract. Exper.*, Vol. 12, pp. 1147-1162 (1982).
5) Karshmer, A. I. et al.: The New Mexico State University Ring-Star System: A Distributed UNIX Environment, *Softw. Pract. Exper.*, Vol. 13, pp. 1157-1168 (1983).
6) 坂本，鈴木：統合構内サービスの一提案，昭57信学総全大，1464 (1982).
7) 谷口，鈴木：リモートアクセスによる UNIX の分散処理 OS 化，情報処理学会分散処理システム研究会，23-3 (1984).
8) 坂本，鈴木，谷口，横山：EINS における分散処理システムの構成，情報処理学会「LAN/マルチメディアの応用と分散処理」シンポジウム，pp. 151-158 (1984).
9) Thompson, K.: UNIX Implementation, *Bell Syst. Tech. J.*, Vol. 57, No. 6 (1978).

（昭和60年4月 1 日受付）
（昭和60年7月18日採録）