

研究環境向けLANの試みーUNIUNET : I. システムの 概念と基本機能の実現

宇都宮, 孝一
九州大学大学院総合理工学研究科情報システム学専攻

吉岡, 和幸
九州大学大学院工学研究科情報工学専攻 | 日本電気株式会社

西野, 浩明
九州大学大学院総合理工学研究科情報システム学専攻

清水, 泰行
九州大学大学院総合理工学研究科情報システム学専攻

<https://doi.org/10.15017/17614>

出版情報 : 九州大学大学院総合理工学報告. 6 (2), pp.251-257, 1985-01-01. 九州大学大学院総合理工学研究科
バージョン :
権利関係 :

研究環境向け LAN の試み—UNIQUET

I. システムの概念と基本機能の実現

宇津宮 孝一* ・吉岡 和幸**
西野 浩明*** ・清水 泰行***

(昭和59年9月29日)

A Local Area Network for Research Computing Environment—UNIQUET

I. System Design Principles and Fundamental Facilities

Kouichi UTSUMIYA, Kazuyuki YOSHIOKA,
Hiroaki NISHINO and Yasuyuki SHIMIZU

This paper describes system design principles, operating system extensions, and an implementation of fundamental facilities for a prototype of an integrated research computing environment. This environment, called UNIQUET, is based on a local area network of powerful personal computers and consists of interconnected UNIX systems. It can provide scientific researchers with facilities such as a uniform style of user interfaces, system-transparent resource sharing, distributed computing, and good interuser communication and cooperation.

1. はじめに

近年の高性能個人用計算機の普及に伴い、特に科学研究者の計算環境は非常に改善されてきた。しかしながら、その使用形態を詳しく眺めてみると、依然として大型計算機の端末の使用や研究者固有の計算利用が多く、研究という知的作業全体を支援する環境には程遠い。研究環境における計算機の利用は、一般のオフィスと異なり、個別的、非定型的で、少量多品種的性格が強い。そのため、一般のオフィス・オートメーション(OA)の手法がそのままの形で適用できない。

我々は、計算機システムにおけるヒューマン・インターフェース研究の一環として、日本人になじむ、科学研究者のための研究支援環境の試作研究を行っている¹⁾。これは、異種の高性能個人用計算機を研究者用作業台(ワークステーション)として、ローカルエリ

ア・ネットワーク(LAN)で有機的に結合した分散処理指向のシステムである。異種ワークステーションを制御するオペレーティング・システム(OS)としては、研究機関の間で評価の高いUNIX[†]を採用し²⁾、この上に透過性の高いユーザ・インターフェースとネットワーク機能等の基本機能を実現した。

このようなシステムは、既に幾つか開発されているが、そのほとんどがUNIXを用いて同種の計算機をネットワークで結合することに力点が置かれており、統合化した研究支援環境として成熟した形態を整えたものではない^{3)~6)}。唯一の例外は、米国カーネギーメロン大学で進行中のSPICE計画であるが、日本人にもなじむという点に関しては問題がある⁷⁾。

本論文では、UNIXを基盤とする研究環境向けLANとして開発を進めているUNIQUET(UNIX-based Qualified NETwork)のシステム設計概念と有用な応用がその上に構築される基本機能の実現について述べる。

[†] UNIX オペレーティング・システムは AT & T のベル研究所が開発したものである。

* 情報システム学専攻

** 工学研究科情報工学専攻修士課程
(現在日本電気(株))

*** 情報システム学専攻修士課程

2. 研究支援環境の分散処理モデル

研究者の研究活動を、計算機により支援する（研究支援環境）として考慮すべき点は次のとおりである。

- (1) 人がすべき作業と機械にさせる作業の明確化。
- (2) 機械にさせる作業の大半が身近で行える能力と自律性を有する計算設備があること。
- (3) 研究者間で、計算機を介して情報のやりとりができること。更に、やりとりした情報の管理が円滑に行えること。また、高価な設備や他の研究者が有する有用な資源を共用できること。

上記の3点は、それぞれ次の接近法が必要なることを示唆している。

(1) 研究活動の活性化と効率化のため、研究環境という一種のオフィス環境における OA 的接近法。

(2) 種々の研究者に適用できる研究者用ワークステーションの提供。

(3) 研究者・研究室間を結ぶネットワークの構築。

従来の研究支援環境は、集中処理の典型である大型計算機の TSS 環境に代表される。しかしながら、上述した条件を満たすモデルとしては、Fig. 1 に示すようにワークステーション及び大型計算機などの共用設備を LAN で有機的に結合した分散処理モデルが最適である⁹⁾。これは、研究者または研究グループを単位としてワークステーション及びその端末が割り当てられ、プログラム開発、科学技術計算、実験データ処理、文書処理、研究情報管理などほとんどの処理は其中で自律的に行われる。一方、研究グループ間の情報のやりとりや共有資源の利用は、LAN を介してなされるというものである。幾つかの研究グループからなる1つの研究組織が、1つの LAN システムを構成する。従って、研究組織間の情報のやりとりは、LAN 間を接続するゲートウェイ・プロセッサを介して行われる。このモデルを研究支援環境として実現する際には、以下の点に注意しなければならない。

(1) 研究者は計算機の非専門家なので、使いやすいユーザ・インターフェースの提供、ネットワークを意識させない透過性の確保などが重要である。

(2) 通常、ワークステーションは異機種である。ただし、知的財産であるソフトウェアの流通と共用、及びワークステーションの高い機能などの点から、OS は柔軟で、標準的であることが不可欠である。

(3) ネットワークは、標準的かつ十分高速性を有

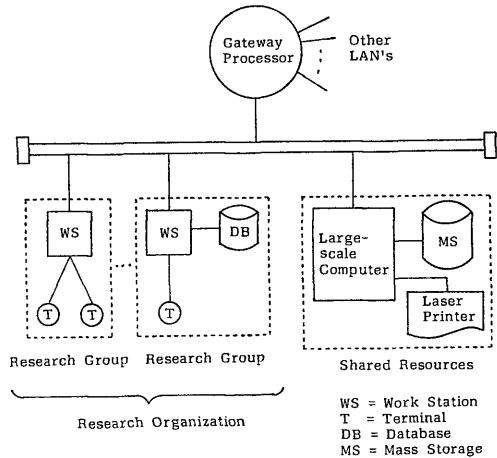


Fig. 1 A distributed computing model for research environment.

し、拡張性がなければならない。

3. UNIQUET システムの概念

前章の分散処理モデルを具現化するシステムの設計においては、下記の方針と概念を基本とした。

(1) 同種 OS 下の異種計算機

多様性（分野の異なる研究者に適したワークステーションの必要性）の許容とシステム全体の一樣性の確保という観点から、異種ハードウェアと同種ソフトウェアの組み合わせを、我々が目指すシステムの基本形とした。ソフトウェアの中心である OS は、機能、流通性及び拡張性の点から、研究機関の間で評価の高い UNIX が最適だと判断してこれを採用した。

(2) LAN の伝送媒体

LAN の物理的な伝送媒体等については、将来性のある媒体・方式を見極めるという立場から、試作段階では特定することをしなかった。

(3) UNIX を核とするネットワーク OS

UNIX が有する利点、すなわち、有用なソフトウェア資源と国際流通性、木構造ファイル・システム、仮想ファイルやパイプの概念、プロセス間通信などを最大限取り入れながら、ネットワーク OS (NOS) としてまとめあげる⁹⁾。

(4) 1 台の仮想的計算機としての見方

システムの透過性を確保し、利用者が LAN 上の計算機群を1台の仮想的計算機と見なすことができるようにするため、UNIX が有する下記の概念をネットワ

ーク全体で機能するよう自然な拡張を図る¹⁰⁾。

- ・取り外し可能なファイル・システム

他系ファイル・システムを、必要なときに自系ファイル・システムの一部として機能するよう動的に結合する。これによって、各 UNIX が管理する資源のネットワークにまたがる統一的な利用を可能にする。

- ・コマンド・パイプライン

コマンド入出力のためのパイプを自系外のシステム間に拡張することにより、他系上でコマンド（プロセス）の実行が容易に行える。

(5) 対象指向分散処理

データ（対象）は、本来そこに存在するのが自然であると思われる場所に置かれるものであるという経験則に基づいて、データの存在場所に操作内容（コマンド）をメッセージとして送ることによって処理を進める。これにより、データの保護や完全性の保持を容易にし、データ転送効率を高めることが可能となる。

4. UNIQUET の構成と基本機能

UNIQUET のハードウェア及びソフトウェア構成と、利用者から見た UNIQUET の基本機能について述べる。

4.1. UNIQUET の構成

4.1.1. ハードウェア構成

Fig. 2 に示すように、試作システムのハードウェア構成は、相互に自律的な UNIX システム3台、すなわち、UX-300、TOWER 1632 及び OA-8140 と、大容量ディスク、高品質プリンタ等を有する共有資源としての大型計算機 M-240H をバス型 LAN で接続したものである。これは LAN システムとしての最小構成条件 (minimum-subset) を満足している。

4.1.2. ソフトウェア構成

分散処理環境を実現するために、各 UNIX システムに付加するネットワーク用プロセス群の構成及びこれらと OSI 参照モデルとの対応づけを Fig. 3 に示す⁹⁾。

以下に各プロセスの機能の概略について述べる。

- ・NM (Network Manager) システム間でのメッセージ（データ）転送の管理。
- ・MT (Message Transmitter) セッション間でのメッセージ（データ）転送の管理。
- ・Nsh (Network shell) UNIX の標準コマンド・インタプリタである shell にネットワーク・サポ

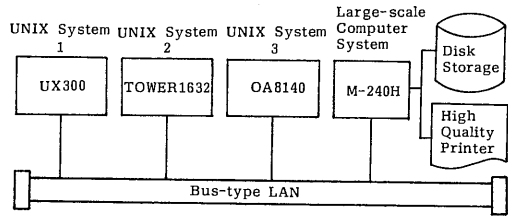


Fig. 2 Hardware configuration of a prototype system.

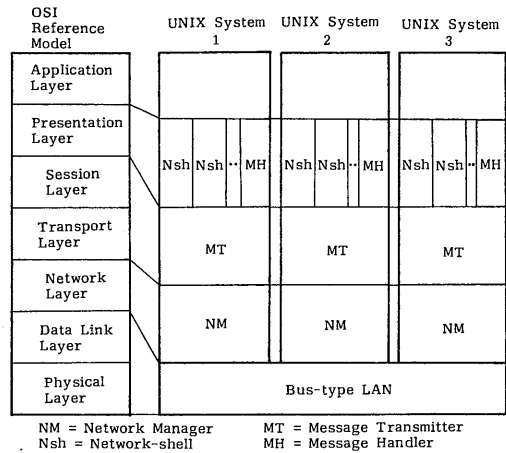


Fig. 3 Layered structure of software system.

ート機能 (4.2 参照) を付加したもの。

- ・MH (Message Handler) 他系資源アクセスのために転送されてきたメッセージのインタプリタで、すべての他系利用者によって共用される。

4.2. UNIQUET の基本機能

分散処理環境を実現し、しかも UNIQUET に付加した機能が利用者自然に受け入れられるために、特に重要と思われる3点、すなわち、(1) システム間の接続、(2) カレント・ディレクトリ (current directory) の管理及び (3) 他系資源 (テキスト/プログラム、コマンド等) の利用について述べる。これらは、前章で説明したように、UNIX の諸概念を十分に反映した概念モデルに基づくものである。

(1) システム間の接続

UNIX のファイル・システムは木構造を基本としている。UNIX は、特殊入出力装置などをマウント (mount) コマンドにより、ディレクトリの下に木構造の一部として結合する仮想ファイルの概念を導入している。UNIQUET におけるシステムの接続は、この

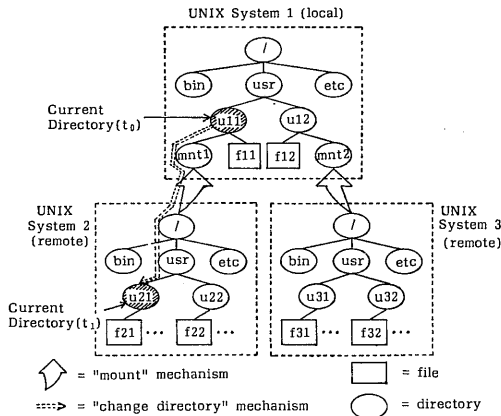


Fig. 4 File system concept of UNIQUET.

概念をシステム間に拡張したものである。つまり、Fig. 4 に示すように、他系のファイルを利用する場合は、他系ファイル・システムを自系システムのファイル木の、ある1つのディレクトリの下に動的に結合（動的マウント）することによって他系と接続するものである。

(2) カレント・ディレクトリの管理

UNIX では、利用者はファイル木の節（ノード）に当たるディレクトリを自由に移動して処理を行う。UNIQUET においては、Fig. 4 に示すように、他系をマウントした後に、従来のディレクトリ変更機構を用いて、他系上のディレクトリに移動し、処理を進めることができるように拡張を行った。

(3) 他系資源の利用

処理の分散、資源の共用という分散処理の大原則を満足させるために、UNIX におけるコマンド・パイプラインの概念を、分散処理環境にまで発展させたシステム間パイプとして実現した（詳細は第5章参照）。研究環境においては、ファイルに関する処理も大きな比重を占めている。既に述べたように、UNIQUET では、各ファイルを1つの対象（オブジェクト）と考え、これらの対象へ操作内容（コマンド）をメッセージとして送ることによって処理を進める対象指向の考え方を導入している。第6章で詳述するように、この考え方をシステム間パイプを用いて容易に実現することができる。

5. UNIQUET の資源利用方式

4.2 (3) で述べた 他系資源（各種ハードウェア資

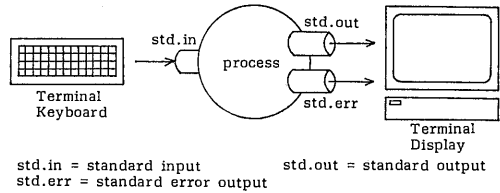


Fig. 5 Standard I/O for process.

源、文書テキスト/ソース・プログラム・ファイル、コマンド等）利用のためのシステム間パイプについて詳述する。まず、UNIX におけるコマンド・パイプラインの概念を述べてから、UNIQUET のシステム間パイプについて説明する。

5.1. UNIX のパイプライン処理

UNIX の標準コマンド、あるいは、実行形式のアプリケーション・プログラムがコマンド・インタプリタ shell によって起動されると、その実体であるプロセスが CPU 上で活動状態となる。その際、Fig. 5 に示すように、プロセスには標準入力（通常は端末キーボード）と標準出力/標準エラー出力（通常は端末ディスプレイ）の3つの入出力が存在する。ある1つの作業を行うのに、複数のプロセスによる連続処理を必要とする場合がある。このとき、ある1つのプロセスの標準出力を次のプロセスの標準入力とするために、パイプラインを用いて両者を直接つなぎ合わせることが可能である（Fig. 6 参照）。このパイプラインの機能が、UNIX を強力な OS としているといっても過言ではない。

5.2. UNIQUET のシステム間パイプ

システム間パイプは、Fig. 7（ただし、破線で囲んだ部分は除く）に示すプロセス群によって実現される。すなわち、利用者が起動したプロセス（以下ユーザ・プロセスと略）への入力を制御する2つのプロセス（DD 及び SDD）、ユーザ・プロセスからの標準出力を他系上の次のユーザ・プロセスへの入力するためにデータ転送を制御するプロセス（SOFM）、更に他系の場合、エラーメッセージを自系のディスプレイ上に表示させるためのプロセス（SEFM）などによって構成される。この状態で、利用者が次のようなコマンド行を入力したと考える（ただし、“|” はシステム間パイプを表す。また、command 2, command 1, command 0 はそれぞれ他系 2, 他系 1, 自系で起動されるコマンド（プロセス）とする）。

```
command 2 | command 1 | command 0
```

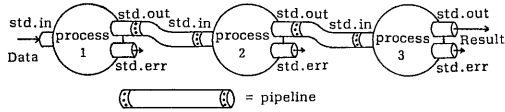


Fig. 6 Process-to-process connection with pipelines.

以下この例を用いて、システム間パイプの敷設、実行の2つのフェーズについて説明する (Fig. 7 参照).

(1) フェーズ1 システム間パイプの敷設

Nsh (他系ではMH) は、各ユーザ・プロセスの標準出力 (他系ならば標準エラー出力も) の行き先の情報を管理するプロセス (SOF (他系ならばSEFも)) を生成し、SDD, SOFM (,SEFM) との接続のためのパイプラインを決定する。ここで、SOF 2 及び SOF 1 が保持するデータ転送経路情報の内容は、それぞれ SOFM 2→MT 2→NM 2→NM 1→MT 1→DD 1→SDD 1→ [process 1] と SOFM 1→MT 1→NM 1→NM 0→MT 0→DD 0→SDD 0→[process 0] となっている (SEF2, SEF 1 の内容は双方とも SEFM 2 (1)→MT 2 (1)→NM 2 (1)→NM 0→MT 0→DD 0→Nsh→(自系ディスプレイ) となっている).

(2) フェーズ2 実行

フェーズ1がすべての系で終了した後に、Nsh はユーザ・プロセスの起動を行う。この後のデータの流れは次のようになる。[process 2]→SOF 2 ⇒ [process 1]→SOF 1 ⇒ [process 0] (ただし、⇒の部分 はそれぞれフェーズ1で記述した経路である)。

6. UNIQNET の基本機能の利用

分散処理環境において、前章で述べたシステム間パイプ及び標準 UNIX のコマンド群を有効に利用するために、UNIQNETではコマンド展開という手法を使って、利用者の入力したコマンドを処理している。Nsh は、利用者が入力したコマンド列を受け取って、ファイル名の解釈を行い、そのコマンド列を幾つかの標準 UNIX コマンドに展開する。そして、システム間パイプを用いて、ファイル及びディレクトリ (オブジェクト) が存在するシステム上で、それぞれのコマンドを起動する。ここでは、このコマンドの展開について述べる。

UNIX においては、標準入力から入力したデータを標準出力に出力するコマンド (フィルタ型コマンドと

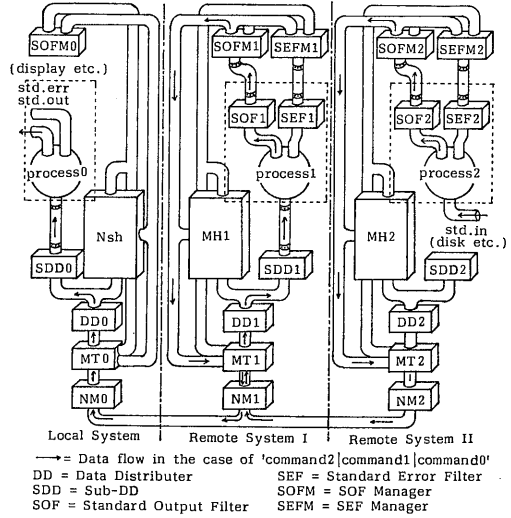


Fig. 7 Network-wide pipeline system.

呼ぶ) を、コマンド・パイプラインを用いて組み合わせることによって、ファイル及びディレクトリの操作を行うことが基本である。このフィルタ型コマンドの中で、cat コマンドは、標準入力から得たデータをそのまま標準出力に出力するものなので、特に基本的なコマンドであると考えられる。UNIQNET では、この cat コマンドを最大限に利用することにより、コマンドの展開を行い、ネットワーク間にまたがる機能の実現をしている。

一般に他系資源のアクセスを行う場合、分散処理を考慮に入れると、次の3つの機能が不可欠である。

- (1) システム間のファイル転送。
- (2) 他系システム上でのコマンドの実行。
- (3) (1), (2) を並行処理, 例えばバック・グラウンド処理すること。

これらは、特に基本となる機能であり、それぞれを組み合わせることにより、UNIQNET 上に種々の機能を実現することが可能となる。また、これらの機能は、単一システム内における UNIX の機能の、ネットワークへの自然な拡張であるため、利用者はネットワークを意識することなく他系資源をアクセスすることができる。

次に、上記 (1), (2) の処理について、コマンド展開の実例を挙げて説明する。

- (1) システム間のファイル転送
システム間のファイル転送は、システム間パイプを

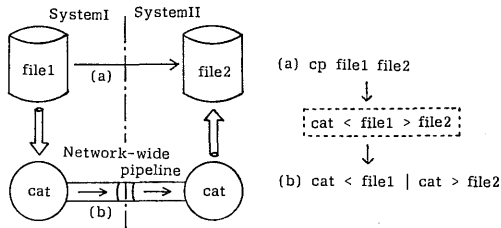


Fig. 8 An example of intersystem file copying.

用いてファイルの内容を送るものである。各システムでは、システム間パイプに対して、cat コマンドがデータの送受を行う。従って、システム I の file1 からシステム II の file2 へ転送を行う複写コマンド (cp) は、Fig. 8 に示すようにコマンド展開される。すなわち、

```
cat < [file 1] | cat > [file 2]
```

のようにコマンドの展開を行った後、Nsh はシステム間パイプの左側の cat コマンドをシステム I に対して、右側の cat コマンドをシステム II に対して起動することを要求する。起動後、左側の cat コマンドが file1 の内容を読み込んで、システム間パイプに書き出す。そして、右側の cat コマンドはシステム間パイプを読み込んで、その内容を file2 へ書き出す。この繰り返しによって、ファイルが複写される。

(2) 他系システム上でのコマンドの実行

他系システム上でコマンドを実行する場合には、そのコマンドの他系上での実行結果を自系システムに出力するために、自系システムで cat コマンドを起動することが必要となる。例えば、他系システムにカレント・ディレクトリがある場合に、ディレクトリ一覧を表示させるコマンド (ls) を実行する際には、次のように展開される。

```
ls
↓
ls | cat
```

この処理のあと、Nsh は、他系システムに対し ls の起動を要求し、自系では cat コマンドを起動する。このように他系上でのコマンドの実行結果を、システム間パイプを通して cat コマンドで表示することができる。

7. UNIQUET の検討

この章では、UNIQUET の検討の1つとして、本

システムの接近法に対する利点及び欠点について考察する。

主な利点は次の2点である。

(1) 保守性・移植性が良い。

他系システムへ転送する要求の単位として、標準 UNIX のコマンド・レベルのメッセージを採用したため、UNIX のカーネル部の改造及びネットワーク用コマンドの新たな作成が不要である^{3)・4)}。

(2) 分散処理への移行に対し、利用者はごく自然に対応が可能である。

UNIQUET は、UNIX をその特徴を生かしながら、分散処理環境に自然に拡張したものである。このため、コマンド・パイプライン等の機能はそのままであり、従来の UNIX コマンドと同じ手続きで他系資源をアクセスすることができる。

しかしながら、次のような欠点もある。

(1) 並行的に相当数のプロセスが動作することによる効率の低下。

第5章で述べたように、システム間パイプの構築には、ヘッダ処理、ネットワーク管理などのために、多くのプロセスが動作する。その結果、メッセージが幾つものプロセス間を通過するので、実質的転送速度が低下することになる。

(2) 利用者自身が作成したコマンドで、直接他系資源をアクセスすることが困難である。

UNIQUET では、システム・コールのレベルではなく、コマンド・レベルのメッセージ転送を採用した。そのため、Nsh が、コマンドの展開やファイル名の処理等を行うので、コマンド名、コマンドの構文及びその展開法をあらかじめ知っている必要がある。その結果、Nsh は、利用者が作成したコマンドを処理することが困難である。

しかしながら UNIX では、標準コマンド (フィルタ型コマンド) をパイプラインを用いて組み合わせることにより、多くの処理を行うことが可能である。従って、ネットワークに拡張したコマンド・パイプライン上で、標準コマンドを組み合わせることによって、ほとんどの処理を実質的に行うことができるものと思われる。

以上、UNIQUET について利点及び欠点を述べたが、UNIQUET は、将来、分散型オペレーティング・システムを設計・構築する際の1つのプロトタイプとし、十分その機能を備えているものと考えられる。

8. おわりに

異種の高性能個人用計算機からなる科学研究者用ワークステーションをローカルエリア・ネットワークで接続し、機能が高く普及の著しい UNIX オペレーティング・システムを基盤にして、研究支援環境の構築に必要な基本機能を実現した。LAN のどのワークステーションからも同じ手続きで、あたかも全体が1つのシステムであるかのように使用できるメリットは想像以上に大きい。

現在この基本機能を用いて、使いやすいユーザ・インターフェース及び研究者間の情報授受・管理用の応用を作成している。これらを統合して、研究支援環境のプロトタイプとして完成することが急務である。また、LAN の伝送媒体、秘密保護機構などに関する検討は今後の課題である。こうした点に関しては改めて報告したい。

参 考 文 献

- 1) 西野ほか：UNIX を基盤とする研究室向け LAN のプロトタイプ UNIQUET (1), (2), 第37回電気関係学会九支連大論文集 (1984).
- 2) Ritchie, D. M. and Thompson, K.: The UNIX Time-Sharing System, Bell Sys. Tech. J, 57, 6, pp. 1905-1929 (1978).
- 3) Rowe, L. A. and Birman, K. P.: A Local Network Based on the UNIX Operating System, IEEE Trans., Software Eng., SE-8, 2, pp. 137-146 (1982).
- 4) Hwang, K. et al.: A Unix-Based Local Computer Network with Load Balancing, IEEE Computer, 15, 4, pp. 55-65 (1982).
- 5) Brownbridge, D. R., Marshall, L. F. and Randell, B.: The Newcastle Connection or UNIXes of the World Unite!, Software P & E, 12, 12, pp. 1147-1162 (1982).
- 6) Walker, B., Popek, G. et al.: The LOCUS Distributed Operating System, Proc. of 9th ACM Symposium on Operating Systems Principles, pp. 49-70 (1983).
- 7) Barstow, D. R., Shrobe, H. E. and Sandwell, E.: Interactive Programming Environment-The Spice Project, pp. 546-557, McGraw-Hill (1984).
- 8) Davies, D. W. et al.: Distributed Systems-Architecture and Implementation, Lecture Notes in Computer Science 105, p. 510, Springer-Verlag (1983).
- 9) Saito, N., Murai, J. and Nakamura, O.: Design Principle of Operating System for Local Area Network, 情報処理学会ローカルエリアネットワークシンポジウム報告集, pp.51-58 (1983).
- 10) グロウ, J. R.: ネットワーク通信機能を組み込んだ Unix システム, 日経エレクトロニクス, 1-30, 335, pp. 167-181 (1984).