# *AnT* におけるOSサーバ入れ替え機能の評価

谷口秀夫[†1] 藤原康行[†1] 後藤佑介[†1]
田端利宏[†1] 乃村能成[†1]

高い適応性と堅牢性の実現を目指し，マイクロカーネル構造を有する ***AnT*** オペレーティングシステムについて，OS サーバの入れ替え機能が提案されている．本機能の特徴は，OS サーバの保有情報を最小化し，新旧 OS サーバ間における処理の継続を可能にしている点である．本機能により，OS 機能拡張時および OS サーバ不具合発生時には，OS サーバを入れ替えることで，システムを停止させることなくサービスの継続が可能になる．ここでは，本機能の評価結果を報告する．

# Evaluation of Dynamic OS Server Replacement Mechanism for *AnT*

Hideo Taniguchi,[†1] Yasuyuki Fujiwara,[†1]
Yusuke Gotoh,[†1] Toshihiro Tabata[†1]
and Yoshinari Nomura [†1]

To increase the adaptability and robustness of software systems, we have proposed a dynamic OS server replacement mechanism in ***AnT*** operating system based on micro kernel architecture. Using our mechanism, information on OS server can be minimized and processing on the old system can be handed over to the new one. When OS functions are extended or OS server is malfunctioned, this architecture allows us to continue the service without stopping the system by replacing the OS server. In this paper, we evaluate the effectiveness of our mechanism.

†1 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University



## 1. はじめに

計算機システムの重要性が増大するにつれ，高い適応性と堅牢性がシステムに求められている．特に，システムの基盤ソフトウェア（オペレーティングシステム（OS））に対し，この要望が強くなっている．この要望を満足する手法として，マイクロカーネル構造 OS[1)–3)] がある．マイクロカーネル構造 OS は，OS 機能の必須部分をマイクロカーネルとして構築し，他の大半の OS 機能をプロセス（OS サーバ）として実現する．これにより，サービスが必要とする OS 機能を提供する OS サーバのみを動作させることで，システムへの高い適応性を実現できる．また，OS 機能の不具合発生時の対処として，プログラム入れ替えの機能[4),5)] が必要である．これは，モノリシックカーネル構造 OS では，不具合が生じた OS プログラムモジュールの入れ替え（交換）になる．多くの場合，OS プログラムモジュール相互の関係は複雑であるため，OS プログラムモジュールの入れ替え機能の実現は容易ではない．一方，マイクロカーネル構造 OS では，不具合が生じた OS サーバの入れ替え，つまりプロセスの入れ替えで対処できる．多くの場合，プロセス相互の関係はプログラムモジュール相互の関係に比べ簡易であるため，マイクロカーネル構造 OS は，モノリシックカーネル構造 OS に比べ高い堅牢性を実現できる．

OS サーバの入れ替え機能においては，入れ替えによるサービスへの影響を最小限とすることが重要である．我々は，マイクロカーネル構造を有する ***AnT*** オペレーティングシステム[6),7)] について，OS サーバの入れ替え機能を提案した[8)]．本機能の特徴は，OS サーバの保有情報を最小化し，新旧 OS サーバ間における処理の継続を可能にしている点である．

ここでは，提案した入れ替え機能の評価結果について報告する．

## 2. OS サーバ入れ替え機能

### 2.1 サーバ間通信制御機構

***AnT*** オペレーティングシステムにおけるプログラム間の呼び出し制御の基本機構を**図 1** に示す．内コア（カーネル）は，各 OS サーバに依頼キューと結果キューを用意する．処理の呼び出しは，制御用 ICA とデータ用 ICA を授受 (仮想空間上の剥がしと貼り付け) することで行われる．基本的な呼び出しの処理を以下に述べる．

(1) 依頼元プロセスは依頼先プロセスの依頼キューに依頼を登録する．
(2) 依頼先プロセスは登録された情報を取得し処理を実行する．
(3) 依頼先プロセスは依頼元プロセスの結果キューに結果を登録して返却する．

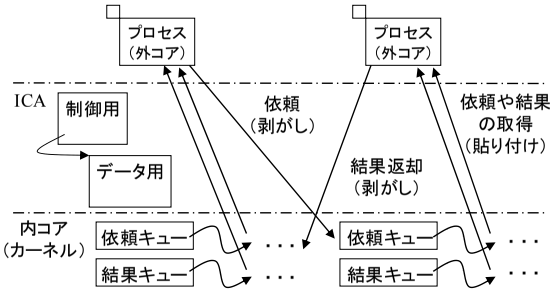



**図 1**　OS サーバに対する通信制御の基本機構

### 2.2　OS サーバ入れ替え機能実現時の課題と対処

#### 2.2.1　通信先 OS サーバの特定法

OS サーバはプロセス識別子をもつため，OS サーバと通信を行う場合，プロセス識別子で通信相手を特定できる．しかし，OS サーバ入れ替えでは，新旧 OS サーバを入れ替えるため，プロセス識別子が変化する問題がある．

このため，以下の対処を行う．

**（対処 1）** OS サーバが提供する OS 機能に識別子（機能識別子）を設ける．

**（対処 2）** 機能識別子をもとに通信相手を特定する通信制御機構を設ける．

**（対処 3）** OS サーバのプロセス識別子と機能識別子を関連付ける管理構造を設ける．

**（対処 1）**により，OS サーバ入れ替えの前後で OS サーバの機能識別子が変化することはない．

**（対処 2）**により，他のプログラムは，機能識別子をもとに OS サーバとの通信を行う．なお，マイクロカーネル構造をもつ OS では，OS サーバとの通信が頻発するため，Linux に代表されるモノリシックカーネル構造の OS に比べ，性能低下が懸念される．そこで，OS サーバの機能識別子を利用した性能低下を抑制する通信制御機構を設ける必要がある．

**（対処 3）**により，カーネルが提供する資源を機能識別子と関連付けることができる．これにより，新旧 OS サーバ間での資源移譲を簡易化できる．

#### 2.2.2　新旧 OS サーバ間の情報移譲法

OS サーバは，内部状態情報や他のプログラムとの通信情報などの情報を保有している．例えば，内部状態情報として，ファイル管理 OS サーバでは，サービス提供先プログラムが利用しているファイルとプログラムとの対応付け情報がある．また，他のプログラムとの通信情報として，磁気ディスク入出力 OS サーバでは，ファイル管理 OS サーバからのファイル読み込み処理依頼を格納した通信情報がある．このため，入れ替えの前後でサービスに影響を与えないためには，新旧 OS サーバ間で情報を移譲する必要がある．

このため，以下の対処を行う．

**（対処 1）** OS サーバ入れ替えにより移譲する情報を OS サーバ外部に保持する．具体的には，ICA に保持する．

**（対処 2）** OS サーバ入れ替え処理では移譲する情報の位置情報を新 OS サーバに通知する．

上記の対処により，移譲する情報が明確化され，新旧 OS サーバ間での情報移譲を複写レスで行うことができ，高速な入れ替えを可能にする．

#### 2.2.3　入れ替え処理時間の短縮法

入れ替え処理時間の短縮は，入れ替え時のサービス継続提供とともに重要な事項である．

このため，以下の対処を行う．

**（対処 1）** OS サーバとの通信情報を格納する領域をカーネルに実現する．

**（対処 2）** OS サーバの処理に原子性をもたせる．

**（対処 1）**により，OS サーバに対する通信情報を OS サーバプロセスから分離できる．具体的には，カーネル内に OS サーバに対する依頼を格納する依頼キューと他の OS サーバからの処理結果を格納する結果キューを設ける．OS サーバはシステムコールを発行することで，カーネル内にある依頼キューや結果キューから依頼や結果を取得し，サービスを行う．

**（対処 2）**により，OS サーバの入れ替え契機を明確化できる．つまり，OS サーバが不可分操作を行っていない場合に OS サーバ入れ替えを行う．

## 3.　考　　察

### 3.1　入れ替え処理時間

入れ替え処理時間は，入れ替え処理時の他プロセスの影響により変化する．OS サーバ入れ替え処理の様子を**図 2** に示す．最初に，処理の流れを説明する．AP プロセスからの処理の依頼は，OS サーバに対して行われる．OS サーバは，処理の依頼を受け取ると依頼内容実行処理を行い，ドライバプロセスへ処理を依頼する．ドライバプロセスは，処理の依頼に基づき入出力制御処理を行い，結果を OS サーバへ返却する．OS サーバは，結果を取得すると結果内容返却処理を行い，結果を AP プロセスへ返却する．AP プロセスは，結果を取得し，処理を終了する．このような処理の流れにおいて，入れ替え制御プロセスが OS サー

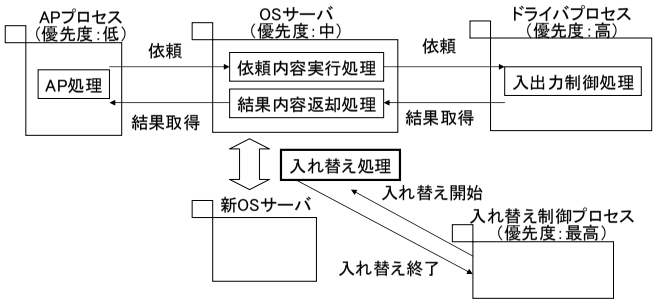



**図 2**　OS サーバ入れ替え処理

バを新 OS サーバに入れ替える際の処理について説明する．なお，各プロセスの優先度は，入れ替え制御プロセス，ドライバプロセス，OS サーバ，AP プロセスの順で高いと仮定する．また，新 OS サーバの優先度は，OS サーバと同じとする．

入れ替え処理は，入れ替え制御プロセス，OS サーバ，および新 OS サーバのコンテキストで実行される．このため，OS サーバや新 OS サーバより高優先度なプロセス（ドライバプロセス）が動作するか否か，つまりドライバプロセスへの制御移行が起こるか否かにより，処理の流れが変化する．ドライバプロセスへの制御移行が起こらない場合の処理の流れを**図 3** に示し，以下に説明する．

(1) 入れ替え制御プロセスは，システムコールにより入れ替え開始を指示する．
(2) 上記の指示は，OS サーバに伝えられ，OS サーバは新 OS サーバをプロセスとして生成する．
(3) 生成された新 OS サーバは，システムコールにより自分自身を新サーバとして登録することを要求する．
(4) 新サーバの登録が要求されると，その旨が OS サーバに伝えられ，OS サーバは終了する．
(5) OS サーバの終了は入れ替え制御プロセスに伝えられ，入れ替えシステムコールが終了する．これを受け，入れ替え制御プロセスは処理を終了する．
(6) 新サーバの登録が終了すると，新サーバは起動を開始し，サーバとしての処理を開始する．

上記の処理流れは，入れ替え処理時間が最短になる場合である．以降では，入れ替え処理時間の長大化について議論する．

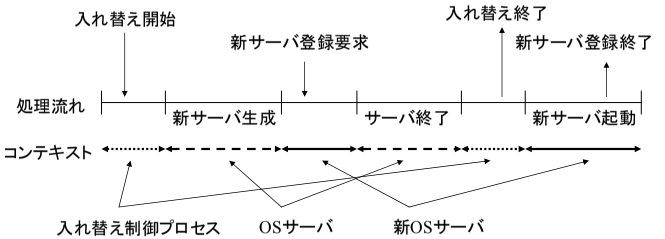



**図 3**　OS サーバ入れ替え処理の流れ（ドライバプロセスへの制御移行なしの場合）

入れ替え処理時間の長大化は，入れ替え可能になるまでに待ちが発生する場合，および入れ替え処理中に他プロセスに制御が移行し入れ替え処理が待たされる場合に発生する．前者の場合は，OS サーバの動作中（依頼内容実行処理または結果内容返却処理の実行中）に入れ替え開始を指示した場合である．後者の場合として，大きく２つある．ひとつは，OS サーバより高優先度なプロセスによりプリエンプションが発生した場合である．もうひとつは，入れ替え処理での新サーバ生成時に発生する入出力による待ちの間に，OS サーバより高優先度なプロセスが走行を開始し，待ちが解除され入れ替え処理が実行可能になっても処理を開始できない場合である．したがって，図 2 と図 3 の場合，図 3 の処理において，OS サーバもしくは新 OS サーバのコンテキスト実行時にドライバプロセスへの制御移行が発生すると入れ替え処理時間が長大化する．この現象は，図 2 において，OS サーバの依頼内容実行処理あるいはドライバプロセスの入出力制御処理の途中で，入れ替え制御プロセスが入れ替え開始を指示した場合に発生する．以上より，入れ替え開始を指示した契機と入れ替え時間の長大化の関係は，以下のようになる．

(1) 依頼内容実行処理中に入れ替え開始を指示した場合，残存する依頼内容実行処理時間（入れ替え待ち時間），および入れ替え処理中のドライバプロセス走行時間の影響を受ける．
(2) 入出力制御処理中に入れ替え開始を指示した場合，入れ替え処理中のドライバプロセス走行時間の影響を受ける．
(3) 結果内容返却処理中に入れ替え開始を指示した場合，残存する結果内容返却処理時間（入れ替え待ち時間）の影響を受ける．

ここで，結果内容返却処理時間が他の処理に比べ長くないと仮定すると，上記 (1), (2), (3) の順で入れ替え時間の長大化が発生するといえる．

マイクロカーネル構造 OS では，OS 機能を提供する OS サーバが複数存在し，また多くのドライバプロセスが存在する．一方，OS サーバ入れ替え時間は，OS サーバ処理時間と入れ替えたい OS サーバの優先度以上の優先度を持つプロセスに影響される．したがって，入れ替え時間の長大化を防ぐためには，OS サーバ処理時間の短縮とともに，プロセスの優先度設定を工夫して高優先度プロセスを必要最小限とすることが重要である．

### 3.2 応答時間

AP プロセスの処理の依頼から結果取得までの時間（応答時間）について考察する．図 2 と図 3，および入れ替え処理時間の考察から，入れ替え処理により応答時間が長大化する場合として，以下の場合がある．なお，応答時間が長大化した場合の最大増加量は，入れ替え処理時間であることはいうまでもない．

(1) OS サーバが依頼内容実行処理を開始する前に入れ替え開始が指示され，入れ替え処理を実行するために依頼内容実行処理の開始が遅れる場合である．AP プロセスは優先度が低いため，プリエンプションにより入れ替え制御プロセスに制御が移行し入れ替え開始の指示が行われる．入れ替え処理においては待ちが発生するため，その間に AP プロセスが走行し，処理を依頼する．

(2) OS サーバが結果内容返却処理を開始する前に入れ替え開始が指示され，入れ替え処理を実行するために結果内容返却処理の開始が遅れる場合である．ドライバプロセスに比べ入れ替え制御プロセスは優先度が高いため，プリエンプションにより入れ替え制御プロセスに制御が移行し入れ替え開始の指示が行われる．ドライバプロセスは OS サーバより優先度が高いため，入れ替え制御プロセスのコンテキストで走行する入れ替え処理部分が実行されるが，OS サーバのコンテキストでの実行は抑制される．このため，入出力制御処理への影響は少ない．しかし，ドライバプロセスが処理の結果を通知しても，OS サーバのコンテキストでは結果取得よりも入れ替え処理を優先して実行する．この場合は，長大化の影響が最も大きい．

(3) AP プロセス が結果取得を行う直前に入れ替え開始が指示され，入れ替え処理を実行するために結果取得の開始が遅れる場合である．これは，AP プロセス起床後の走行直後から結果取得のシステムコール処理が終了する直前までの短い時間内に，プリエンプションにより入れ替え制御プロセスに制御が移行し入れ替え開始の指示が行われた場合であり，その確率は少ないといえる．

## 4. 実測評価

### 4.1 評価環境

Intel® Celeron プロセッサ（2.0GHz）を搭載した計算機上で ***AnT*** を走行させ，入れ替え処理時間と応答時間を測定した．各プロセスの処理内容は図 2 に示すものである．なお，AP 処理を 1000 ミリ秒の WAIT 処理，入出力制御処理を 50 ミリ秒の PU 処理と 100 ミリ秒の WAIT 処理とした．また，OS サーバの各処理は PU 処理とした．ここで，PU 処理は，特定の領域のインクリメントを繰り返すプロセッサ処理であり，WAIT 処理は，指定時間だけ実行権を放棄するシステムコールを用いた処理である．

### 4.2 入れ替え処理時間

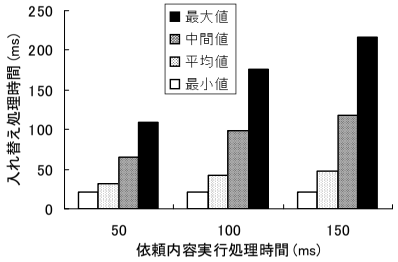
OS サーバの依頼内容実行処理または結果内容返却処理の時間と入れ替え処理時間の関係を明らかにする．測定結果を**図 4** に示す．図 4 から，以下のことがわかる．

(1) いずれの場合も，最小値は 21 ミリ秒程度である．これは先に示した入れ替え処理時間が最短になる場合である．ここで，AP プロセスから処理を依頼しない状態，つまり高優先度のドライバプロセスが動作しない状態で入れ替え処理を行った際，その処理時間は，最小 21.04 ミリ秒，平均 26.69 ミリ秒，最大 32.21 ミリ秒であった．先の最小値は，この最小の値に相当する．なお，この時間のバラツキは磁気ディスク装置の回転待ちやシーク待ちが原因と推察できる．

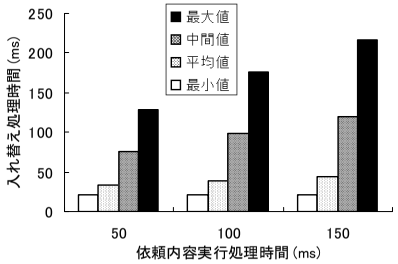
(2) (A)〜(C) と (D)〜(F) を比較することにより，入れ替え処理時間の平均値や最大値は，結果内容返却処理時間よりも依頼内容実行処理時間の変化に大きく影響されていることがわかる．これは，各処理時間を長くすると，その処理中に入れ替え開始を指示する確率が増加することに起因し，先の考察を裏付けるものである．

(3) (D) では，結果内容返却処理時間を長くすると，入れ替え処理時間の最大値も長くなっている．これに対し，(F) では，結果内容返却処理時間を長くしても，入れ替え処理時間の最大値に変化が見られない．これは，(D) の場合，結果内容返却処理時間（50〜150 ミリ秒）は，依頼内容実行処理時間（50 ミリ秒）以上の長さであるため，最大値は結果内容返却処理時間の変化に依存する．一方，(F) の場合，結果内容返却処理時間（50〜150 ミリ秒）は，依頼内容実行処理時間（150 ミリ秒）以下の長さであるため，最大値は依頼内容実行処理時間の変化に依存する．

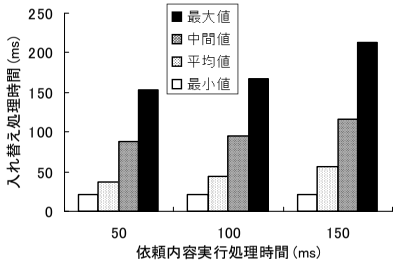
(4) いずれの場合も，平均値が最小値に近くなっている．これは，AP 処理と入出力制御処理の時間の総和が 1150 ミリ秒であるのに対し，OS サーバの処理（依頼内容実行処理



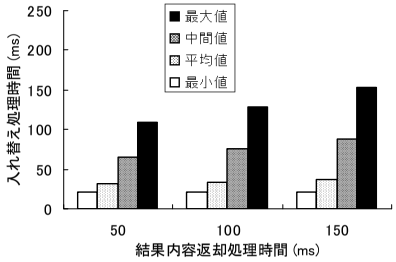

(A) 結果内容返却処理時間 = 50 ms



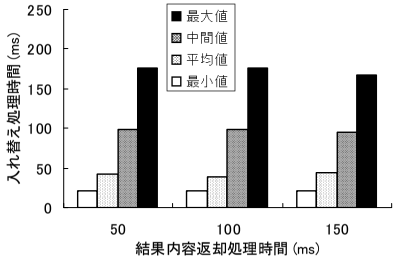

(B) 結果内容返却処理時間 = 100 ms



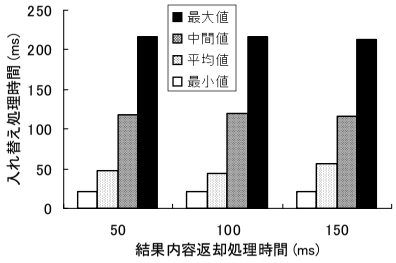

(C) 結果内容返却処理時間 = 150 ms




(D) 依頼内容実行処理時間 = 50 ms




(E) 依頼内容実行処理時間 = 100 ms




(F) 依頼内容実行処理時間 = 150 ms

**図 4** 入れ替え処理時間

と結果内容返却処理の時間の和）が 100〜300 ミリ秒と短いため，入れ替え処理時間が長大化する確率が大きくないことに起因する．

### 4.3 応答時間

入れ替え処理が応答時間に与える影響を明らかにする．依頼内容実行処理時間と結果内容返却処理時間がいずれも 50 ミリ秒の場合について，ランダム間隔で入れ替えを行った（100

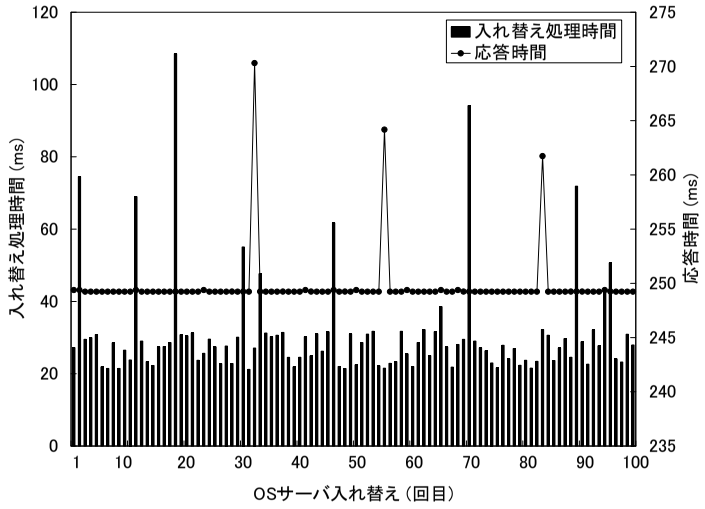



**図 5**　入れ替え処理時間と応答時間

回）ときの入れ替え処理時間と応答時間の関係を**図 5** に示す．つまり，図 4 (A) の依頼内容実行処理時間 50 ミリ秒の場合の入れ替え処理時間は，図 5 の入れ替え処理時間に関する値である．図 5 から，以下のことがわかる．

(1) 応答時間の大半は約 250 ミリ秒である．これは，OS サーバの処理時間（50＋50＝100 ミリ秒）とドライバプロセスの処理時間（150 ミリ秒）の和であり，入れ替え処理が応答時間に影響を与えない箇所で実行されたことを意味する．

(2) 応答時間が増加する回数は，3 回程度（3 %）であり，非常に少ない．なお，この増加量の最大値は 21 ミリ秒程度であり，3.2 節で述べたように，これは入れ替え処理時間に相当する．

(3) 入れ替え処理時間が増加した際でも，応答時間の増加に影響していない．これは，3 章で述べたように，両者の時間の増加要因に直接的な関係がないためである．

## 5. ま と め

***AnT*** オペレーティングシステムに実現した OS サーバ入れ替え機能について，入れ替え処理時間と応答時間に関する性能を考察し，評価した．

入れ替え処理時間は，依頼内容実行処理中に入れ替え開始を指示した場合，入出力制御処理中に入れ替え開始を指示した場合，および結果内容返却処理中に入れ替え開始を指示した場合に長くなることを示した．入れ替え処理時間の長大化を防ぐためには，OS サーバ処理時間の短縮とともに，プロセスの優先度設定を工夫して高優先度プロセスを必要最小限とすることが重要である．また，応答時間は，OS サーバが依頼内容実行処理を開始する前，OS サーバが結果内容返却処理を開始する前，AP が結果取得を行う直前のいずれかにおいて入れ替え開始が指示された場合に長くなることを示した．

実測評価により，入れ替え処理時間は，依頼内容実行処理または結果内容返却処理の処理時間の影響を大きく受け，これらの処理の影響を受けない場合でも 21 〜 32 ミリ秒程度であることを明らかにした．また，応答時間の増加量の最大値は，21 ミリ秒程度であることを明らかにした．

残された課題として，より詳細な評価がある．

## 参 考 文 献

1) J. Liedtke, “Toward Real Microkernels,” Commnuications of The ACM, Vol.39, Issue 9, pp.70-77, 1996.
2) S. Tanenbaum, N. Herder, and H. Bos, “Can we make operating systems reliable and secure?,” IEEE Computer Magazine, Vol.39, No.5, pp.44-51, 2006.
3) D.L. Black, D.B. Golub, D.P. Julin, R.F. Rashid, R.P. Draves, R.W. Dean, A. Forin, J. Barrera, H. Tokuda, G. Malan, and D. Bohman, “Microkernel Operating System Architecture and Mach,” Journal of Information Processing, Vol.14, No.4, pp.442-453, 1992.
4) 谷口 秀夫, 伊藤 健一, 牛島 和夫, “プロセス走行時におけるプログラムの部分入替え法,” 信学論 (D-I), Vol.J78-D-I, No.5, pp.429-499, 1995.
5) Linux Journal Staff, “Kernel Korner: Dynamic Kernels - Modularized Device Drivers,” Linux Journal, Issue 23, No.7, 1996.
6) 谷口 秀夫, 乃村 能成, 田端 利宏, 安達 俊光, 野村 裕佑, 梅本 昌典, 仁科 匡人, “適応性と堅牢性をあわせ持つ ***AnT*** オペレーティングシステム,” 情報処理学会研究報告, 2006-OS-103, Vol.2006, No.86, pp.71-78, 2006.
7) 岡本 幸大, 谷口 秀夫, “***AnT*** におけるサーバ間の高速なプログラム間通信機構,” マルチメディア通信と分散処理ワークショップ論文集, Vol.2007, No.9, pp.61-66, 2007.
8) 藤原 康行, 岡本 幸大, 田端 利宏, 乃村 能成, 谷口 秀夫, “***AnT*** における OS サーバ入れ替え機能,” マルチメディア通信と分散処理ワークショップ論文集, Vol.2008, No.14, pp.201-206, 2008.