AnT におけるOSサーバ入れ替え機能

藤原康行[†] 岡本幸大[††] 田端利宏[††]
乃村能成[††] 谷口秀夫[††]

AnT オペレーティングシステムは，高い適応性と堅牢性を実現するために，マイクロカーネル構造を持ち，OS機能の大半をOSサーバとして実現する．これにより，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となる．ここでは，***AnT*** 上でのOSサーバ入れ替えについて述べる．設計方針は，OSサーバが保有する情報の最小化，および新旧OSサーバ間における処理の継続である．

Dynamic OS Server Replacement Scheme for *AnT*

YASUYUKI FUJIWARA,[†] KOUTA OKAMOTO,[††] TOSHIHIRO TABATA,[††]
YOSHINARI NOMURA[††] and HIDEO TANIGUCHI [††]

The ***AnT*** operating system is based on microkernel architecture to achieve high adaptability and robustness. It realizes most of OS functions as a set of OS servers. This architecture makes it possible for service to continue without stopping a system by replacing OS server at the time of extension or malfunction in OS server. This paper describes dynamic OS server replacement scheme for ***AnT***. This scheme is based on a design policy: Reduce information OS server holds; Processing Continuation between the old and the new OS server.

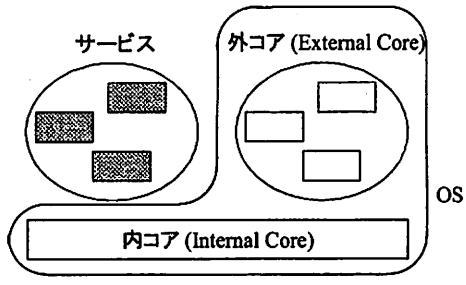
1. はじめに

近年，計算機が提供するサービスの高度化が著しい．これに伴い，基盤ソフトウェアの一つであるオペレーティングシステム (以降，OSと略す) に対し，高い適応性と堅牢性が求められている．これは，機能拡張や不具合発生時の復旧が容易に行えることを意味する．適応性と堅牢性を実現するための手法として，マイクロカーネル構造[1)2)3)] の適用が挙げられる．マイクロカーネル構造は，割り込み処理や例外処理といった最小限のOS処理をカーネル (マイクロカーネル) として実現し，ファイル管理処理や通信制御処理などの処理をプロセスとして実現するプログラム構造である．つまり，多くのOS機能は，カーネル外にプロセスとして実現する．以降，これをOSサーバと呼ぶ．これにより，OS機能拡張時，およびOSサーバ不具合発生時には，対象となるOSサーバ (旧OSサーバ) を新たなOSサーバ (新OSサーバ) と入れ替える[4)5)] 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となる．

OSサーバに対する処理依頼は頻繁に要求される．例えば，外部計算機との通信を行う場合，通信制御処理を行うOSサーバへの処理依頼は頻繁に行われる．この通信制御OSサーバに不具合が見つかった場合，サービスを継続しつつ新たな通信制御OSサーバに入れ替える必要がある．このように，OSサーバ入れ替え途中に発行された処理依頼への対処法は重要な課題である．また，OSサーバは多くの情報を有しているため，サービス利用者に意識させないでOSサーバを入れ替えるには，旧OSサーバから新OSサーバへの情報移行に工夫が必要である．

我々は，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[6)] (An operating system with adaptability and toughness) を開発している．***AnT*** はマイクロカーネル構造を持ち，OS機能の大半をOSサーバとして実現している．ここでは，***AnT*** 上でのOSサーバ入れ替えについて述べる．設計方針は，OSサーバが保有する情報の最小化，および新旧OSサーバ間における処理の継続である．

† 岡山大学工学部情報工学科
Faculty of Engineering, Okayama University
†† 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University



(1) OS: 内コア + 外コア
・内コア: 最小システムの動作を保証する部分
・外コア: 適応したシステムに必須な部分(動的再構成構造)
(2) サービス: サービスを提供する部分

図 1 プログラムの基本構造

2. *AnT* オペレーティングシステム

2.1 適応性と堅牢性

AnT オペレーティングシステムは，計算機システムの開発者と利用者の両者に「使いやすい」かつ「壊れにくい」という特徴を有する基盤ソフトウェアであることを目指している．

高い適応性を有する基盤ソフトウェアの構成法として，具体的には，サービス (機能) の組込みや取外しが簡単なプログラム構成法，および利用者の利用履歴を考慮したプログラム実行制御法が必要である．また，計算機システムの性能低下の大きな要因であるメモリ間データ複写をゼロにする高速なプログラム間データ通信機構も必要である．

高い堅牢性を有する基盤ソフトウェアの構成法として，具体的には，組み込んだプログラムに不具合が発生しても他のプログラムやシステム全体に悪影響を与えない仕組み，およびネットワークを介したセキュリティ低下を防ぐ仕組みが必要である．

2.2 プログラム構造

プログラムは，OS とサービスからなる．OS は，内コアとプロセスとして動作する外コアからなる．サービスは，プロセスからなる．この様子を図 1 に示す．

内コアは，最小のシステムの動作を保証するプログラム部分である．外コアは，適応したシステムに必須なプログラム部分であり，動的に再構成な構造を有する．サービスは，サービスを提供するプログラム部分である．

2.3 サーバプログラム間通信制御[7)8)]

OS サーバ間の通信は，コア間通信データ域[9)](ICA: Inter-core Communication Area) を利用する．図 2 に示すように，ICA は仮想空間のマッピング表の書き換えによりプロセス間でのゼロコピー通信を支援しており，高速な通信が可能となる．

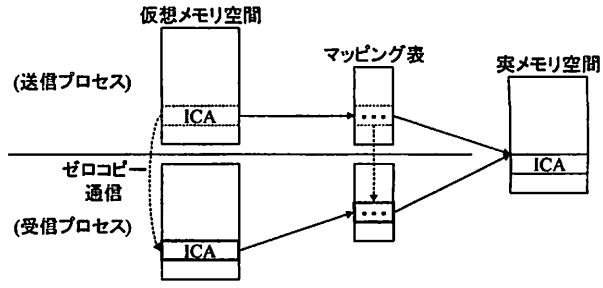


図 2 ICA の基本機構

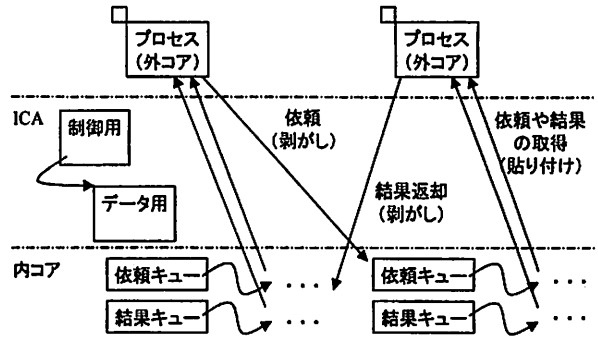


図 3 サーバプログラム間通信の基本機構

また，呼び出す手続きの内容に関する情報と扱うデータを別の ICA に格納する．ここで，手続きの内容に関する情報を格納した ICA を制御用 ICA と呼び，扱うデータを格納した ICA をデータ用 ICA と呼ぶ．これにより，OS サーバ間でのデータの持ち回りを複写レスで行うことが可能となる．

プログラム間の呼び出し制御の基本機構を図 3 に示す．内コアは，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．

3. 設計方針

3.1 契　　機

マイクロカーネル OS において，OS サーバを入れ替える契機は以下の 2 つの場合が考えられる．

(1) OS サーバの機能を拡張する場合 (機能拡張時)
(2) OS サーバの不具合発生に対処する場合 (不具合発生時)

機能拡張時とは，ドライバの更新やOS機能の拡張を行う場合である．Linuxに代表されるモノリシックカーネル構造を有するOSは，機能拡張時にOSの再起動を必要とする．一方，マイクロカーネル構造を有するOSでは，OSサーバ入れ替えを利用することでOSを再起動することなくサービスを継続することが可能である．これは，動作中の旧OSサーバを新OSサーバと入れ替えることで実現できる．

不具合発生時とは，内部矛盾発見によるOSサーバの停止，または無限ループによるOSサーバの無応答が発生した場合である．モノリシックカーネル構造を有するOSは，不具合発生時にはOSが停止してしまう．一方，マイクロカーネル構造を有するOSでは，OSサーバ入れ替えを利用することでOSを停止させることなくサービスを継続することが可能である．これは，停止状態または無応答状態の旧OSサーバを新OSサーバと入れ替えることで実現できる．

3.2 方　　針

OSサーバ入れ替えにおいては，OSサーバが提供するサービスを利用する応用プログラム(AP)への影響を最小限にすることが非常に重要である．このため，OSサーバ入れ替え機能の設計方針として，以下がある．

(方針1) OSサーバが保有する情報の最小化

(方針2) 新旧OSサーバ間における処理の継続

(方針1)により，OSサーバに不具合が発生しても，その影響を最小化できる．また，OSサーバ入れ替え処理の時間を短縮化できる．また，(方針2)により，OSサーバ入れ替えのAPへの影響を最小化できる．処理を継続するためには，新旧OSサーバ間における情報の引継ぎが必要である．ここで，引継ぎ対象の情報には，処理依頼に関する情報(依頼情報)とOSサーバの内部状態に関する情報(状態情報)がある．

次章より，上記の設計方針に基づいた***AnT***におけるOSサーバ入れ替えの機構について述べる．

4. OSサーバ保有情報の処理方式

4.1 依頼情報の扱い

4.1.1 基　　本

依頼情報の扱いを図4に示す．OSサーバに対する処理依頼は制御用ICAに格納され発行される．そこで，基本的には，OSサーバは一度の処理に1つの処理依頼(制御用ICA)を保持する(方針1)．また，機能拡張時は，処理依頼をOSサーバ内に保持していない時にOSサーバ入れ替え行う．不具合発生時は，現在処理中の処理依頼をエラーとしてAPに返却し，

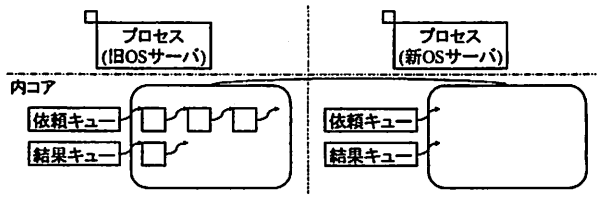


図4　依頼情報の扱い

OSサーバ内に処理依頼を保持しない状態へ移行させてOSサーバ入れ替えを行う．これにより，APへの影響を最小化できる．

また，OSサーバ入れ替え時に，旧OSサーバが保有するすべてのICAを新OSサーバに移行させる．具体的には，内コアのメモリ管理部のマッピング表の書き換えにより，旧OSサーバプロセスの仮想空間に貼り付いているICAを新OSサーバプロセスの仮想空間に貼り替える．さらに，内コアが各OSサーバに用意している依頼キューおよび結果キューに繋がれている制御用ICAを旧OSサーバから新OSサーバへ繋ぎかえる．これにより，処理の継続を可能にする(方針2)．

4.1.2 特　　例

バッファキュッシュ制御のようにキャッシュ機能を持つOSサーバは，複数の処理依頼を保持することがある．複数の処理依頼を保持したままOSサーバ入れ替えを行うと，キャッシュしていた依頼情報を損失してしまう．

そこで，キャッシュ機能を持つOSサーバには，いかなる時点でOSサーバ入れ替えを行っても保持中の依頼を復元可能なように，状態情報ICA(4.2節で後述する)に保持中の依頼に関する情報を保存する．

4.2 状態情報の扱い

4.2.1 保存と復元

4.1節で述べたように，依頼情報はICAに格納され旧OSサーバから新OSサーバへ移行される．そこで，状態情報についても同様な扱いとして，処理の効率化を図る．つまり，状態情報をICA(状態情報ICA)に保存する．OSサーバ入れ替え時には，状態情報ICAを旧OSサーバから新OSサーバへ移行し，新OSサーバの処理開始時に状態情報ICAから状態を復元する．

4.2.2 登録と取得

OSサーバ入れ替えにより，全てのICAは旧OSサーバから新OSサーバへ移行される．しかし，いずれのICAが状態情報ICAであるかを特定するには，新OSサーバに状態情報ICAの位置情報(アドレス)を通知する必要がある．そこで，状態情報ICAのアドレスを内コアへ登録し，OSサーバ入れ替え時に，新

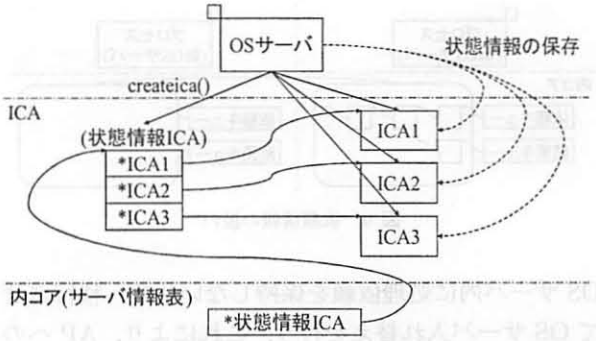


図 5　状態情報を保存している ICA を追跡可能にする場合

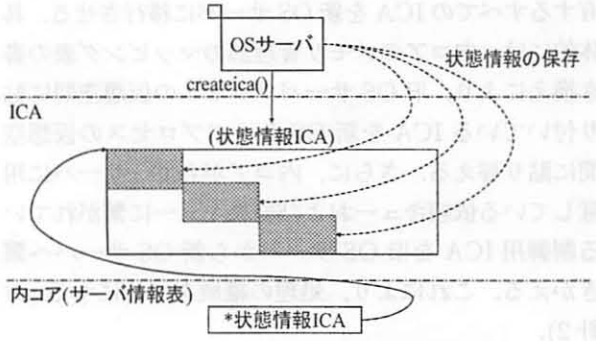


図 6　1 つの ICA にすべての情報を保存する場合

OS サーバへ通知する．

状態情報 ICA を内コアに登録，および内コアから状態情報 ICA のアドレスを取得する機能について以下で述べる．

全ての OS サーバが状態情報の移行を必要とするとは限らない．この点を考慮し，OS サーバ内で作成した ICA を状態情報 ICA として内コアに登録する機能を実現する．

OS サーバ入れ替え時に，新 OS サーバが内コアから状態情報 ICA のアドレスを取得する機能について以下の案がある．

(案 1)　OS サーバ登録の処理結果として状態情報 ICA のアドレスを取得する．

(案 2)　状態情報 ICA のアドレス取得用の機能を新たに作成する．

OS サーバを引継いだ場合，OS サーバ登録後は状態情報 ICA のアドレス取得を必ず行う必要がある．(案 1) は OS サーバ登録時に状態情報 ICA のアドレスの取得を行う方法である．しかし，個々のインタフェース機能を簡素化する観点から，OS サーバ登録の結果に状態情報 ICA のアドレスを取得する機能を含めない．したがって，(案 2) を採用する．

4.2.3　保存形式

OS サーバごとに，移行する状態情報の量は異なる．一方，本 OS サーバ入れ替え機能では，OS サーバが保有する 1 つの状態情報 ICA のアドレスを旧 OS サーバから新 OS サーバへ通知し，OS サーバ内で状態情報の復元を行う．このため，各 OS サーバは，状態情報の保存形式を自由に決定できる．

状態情報の保存形式の例として以下の 2 つを示す．

(手法 1)　1 つの ICA をマスタとし，状態情報を保存している ICA を追跡可能にする．

(手法 2)　1 つの ICA にすべての情報を保存する．

図 5 に示す (手法 1) は，状態情報 ICA として内コアに登録する ICA に，状態情報を保存した ICA のアドレスを格納する．OS サーバ入れ替え後に，この状態情報 ICA に格納されている ICA のアドレスを辿ることで，状態情報の復元が可能となる．本手法は，任意に ICA の作成および削除が可能なため，メモリの有効活用が可能となる長所がある．一方，ICA の仮想空間からの剥がし，および仮想空間への貼り付け処理を行う回数が増加するため，OS サーバ入れ替えに要するオーバヘッドが増加する短所がある．

図 6 に示す (手法 2) は，状態情報 ICA として内コアに登録する ICA に，すべての状態情報を保存する．本手法は，1 つの状態情報 ICA の授受により，状態情報の移行が可能となるため，OS サーバ入れ替えに要するオーバヘッドの増加量を抑える長所がある．一方，ICA は作成時にサイズを指定する[9] が，全ての状態情報を保存可能な十分な大きさの ICA を取得する必要があるため，メモリの有効利用率が低下する短所がある．

5.　機能拡張時の OS サーバ入れ替え法

5.1　要求条件

OS サーバ入れ替え機能の発生契機には，機能拡張時および不具合発生時の 2 通りが存在する．しかし，入れ替え処理の開始が決定した後の入れ替え処理の大半は同様であるため，ここでは機能拡張時の OS サーバ入れ替え法について述べる．機能拡張 OS サーバ入れ替えの要求条件として以下の 3 つを挙げる

(条件 1)　機能拡張 OS サーバ入れ替え機能が呼び出されると速やかに OS サーバを入れ替える．

(条件 2)　OS サーバ入れ替えが成功したか失敗したかを機能呼び出し元のプロセスへ通知する．

(条件 3)　OS サーバ入れ替えが失敗した場合，入れ替え対象となった OS サーバは現状のサービスを維持する．

5.2　処理の流れ

AP に OS サーバの入れ替わりを意識させないため，

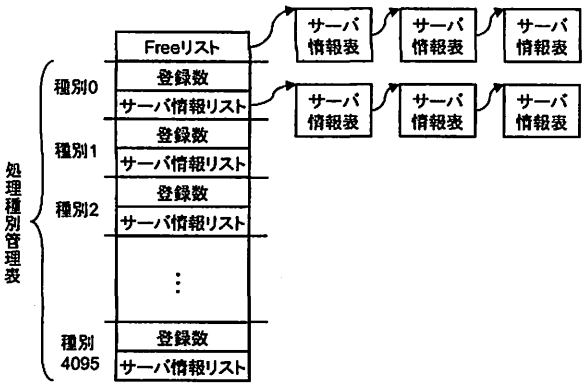


図 7　サーバプログラム間通信制御の管理構造

旧 OS サーバと新 OS サーバは，同一の OS サーバ識別子 (処理種別) となる．処理内容が異なる可能性があり，同一の処理種別を持つ OS サーバが複数存在することは好ましくない．このため，旧 OS サーバを OS サーバ登録から解除し，新 OS サーバを新たに OS サーバ登録させる．しかし，本処理内容では一時的にも OS サーバが存在しなくなる期間が発生する．この間に，旧 OS サーバに対して発行された依頼は，新 OS サーバへの依頼として引き渡すことが不可能となる．そこで，OS サーバ入れ替えにおいて，旧 OS サーバの OS サーバ登録情報をそのまま新 OS サーバへ移行させる．

AnT のサーバプログラム間通信では，OS サーバ登録情報をサーバ情報表に格納し管理している．OS サーバの管理の様子を図 7 に示す．OS サーバとサーバ情報表は，OS サーバ 1 つに対してサーバ情報表が 1 つ割り当てられ，処理種別毎に管理される．AP から OS サーバへの依頼処理では，サーバ情報表を参照し，対象の OS サーバプロセスを特定している．

OS サーバ入れ替え時には，旧 OS サーバで利用していたサーバ情報表を新 OS サーバに引継ぐことで，OS サーバ入れ替え途中に旧 OS サーバに対して発行された依頼を，新 OS サーバへの依頼として引き渡すことを可能とする．また，状態情報 ICA のアドレスをサーバ情報表に保持させることで，状態情報の移行を可能とする．

機能拡張時の OS サーバ入れ替え処理の流れを図 8 に示す．

制御 AP が機能拡張 OS サーバ入れ替え機能を呼び出すと，対象の OS サーバのサーバ情報表の OS サーバ入れ替え用のフラグを立てる．対象 OS サーバは依頼を取得する際に，サーバ情報表の OS サーバ入れ替え用のフラグを確認し，速やかに OS サーバ入れ替え処理を開始する．これにより，(条件 1) を満足できる．

新 OS サーバを起動し，サーバ情報表の設定をした後，新 OS サーバに処理を譲渡する．新 OS サーバは処理を開始し，OS サーバ登録処理中に自プロセスが OS サーバ入れ替え対象であることを知り，旧 OS サーバに処理を返却する．旧 OS サーバは自プロセスに対して発行されていた依頼および状態情報を新 OS サーバに引き渡す．成功したことを制御 AP に通知し，旧 OS サーバは終了する．

なお，入れ替え処理中に失敗した場合は，制御 AP にその旨を通知し，従来の依頼取得を再開する．これにより，(条件 2) と (条件 3) を満足できる．

6. おわりに

マイクロカーネル構造を持ち，OS 機能の大半を OS サーバとして実現している ***AnT*** オペレーティングシステムについて，OS サーバ入れ替え機能について述べた．

マイクロカーネル OS において，OS サーバを入れ替える契機について説明した．また，OS サーバ入れ替えの設計方針を，OS サーバが保有する情報の最小化，および新旧 OS サーバ間における処理の継続と定めた．これらの方針を受け，依頼情報および状態情報の処理方式を述べた．

また，機能拡張時の OS サーバ入れ替え法の設計方針として「機能拡張 OS サーバ入れ替え機能が呼び出されると速やかに OS サーバを入れ替える」，「OS サーバ入れ替えが成功したか失敗したかを機能呼び出し元のプロセスへ通知する」，「OS サーバ入れ替えが失敗した場合，入れ替え対象となった OS サーバは現状のサービスを維持する」と定めた．これに基づき，機能拡張時の OS サーバ入れ替え処理の流れについて述べた．

残された課題として，不具合発生時の OS サーバ入れ替え法の明確化，および実装と性能評価がある．

謝辞　本研究の一部は、科学研究費補助金 基盤研究 (B)「適応性と頑健性を有する基盤ソフトウェアのカーネル開発」(課題番号：18300010) による。

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, H. Bos, "Can we make operating systems reliable and se-

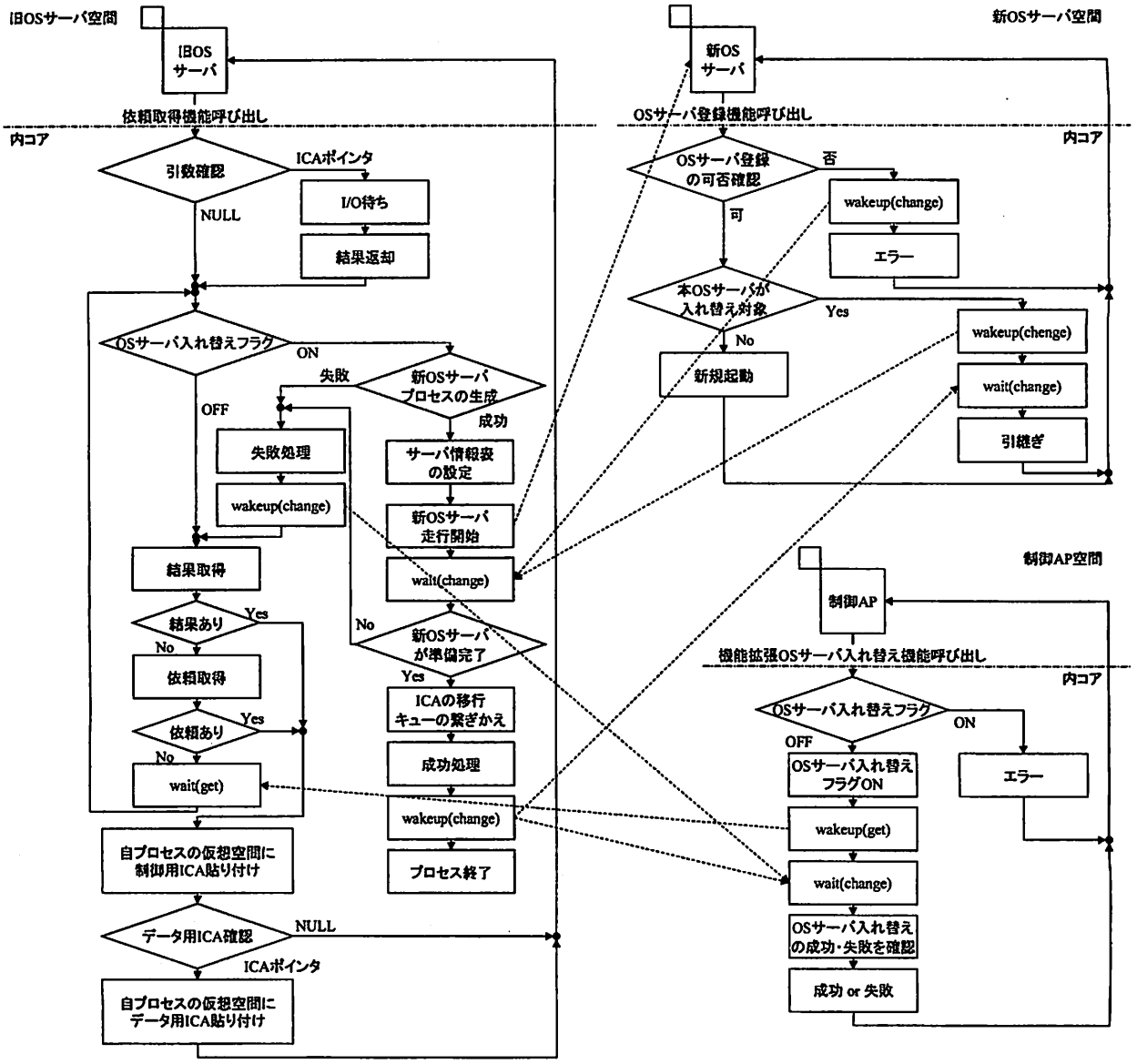


図 8 機能拡張時の OS サーバ入れ替え処理の流れ

cure?," 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(19920315), 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," マルチメディアと分散処理ワークショップ論文集, Vol.2007, No.9, pp.61-66, 2007.

8) 岡本幸大, 谷口秀夫, "*AnT* オペレーティングシステムにおけるサーバプログラム間通信機構の評価," 電子情報通信学会技術研究報告, Vol.107, No.558, pp.49-54, 2008.

9) 梅本昌典, 田端利宏, 乃村能成, 谷口秀夫, "*AnT* オペレーティングシステムにおけるメモリ領域管理の設計と実現," 情報処理学会研究報告, 2007-OS-104, Vol.2007, No.10, pp.33-40, 2007.