(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-258698
(P2004-258698A)
(43)公開日　平成16年9月16日(2004.9.16)

(51)Int.Cl.[7]	F I	テーマコード（参考）
G06F 9/46	G06F 9/46 350	5B098

審査請求 未請求　請求項の数 1　ＯＬ　（全 8 頁）

(21)出願番号　特願2003-45260(P2003-45260)
(22)出願日　平成15年2月24日(2003.2.24)

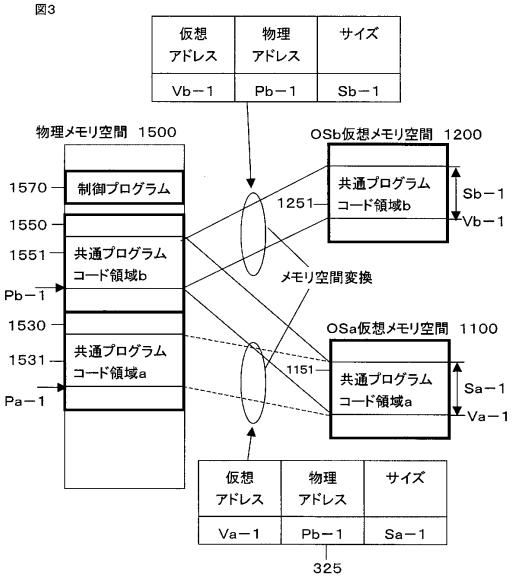
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(74)代理人　100075096
弁理士　作田　康夫
(72)発明者　森　明子
神奈川県秦野市堀山下１番地　株式会社日立製作所エンタープライズサーバ事業部内
(72)発明者　今田　豊寿
神奈川県秦野市堀山下１番地　株式会社日立製作所エンタープライズサーバ事業部内

Ｆターム(参考)　5B098 AA03　GA04　HH01　HH07

(54)【発明の名称】マルチＯＳ計算機システムの制御方法

(57)【要約】
【課題】制御プログラムの制御下で複数のＯＳが同時に動作する計算機システムのメモリ消費量の削減及びメモリの有効利用を図る。
【解決手段】同じＯＳが複数動作することが可能な計算機システムにおいて、制御プログラムが、ある１つのＯＳの持つ各ＯＳに共通なプログラムコード領域のマッピング先物理メモリ空間を共有メモリ化し、他のＯＳは各ＯＳに共通なプログラムコード領域を共有メモリ化された前述の物理メモリ空間にマッピングする。
【選択図】　図３

図3



【特許請求の範囲】
【請求項１】
制御プログラムの制御下で少なくとも２個の同一ＯＳを含む複数のＯＳが動作する計算機システムにおいて、同一ＯＳ上で動作する各同一ＯＳに共通なプログラムコードが配置される仮想メモリ空間を同一の物理メモリ空間に変換するマルチＯＳ計算機システムの制御方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、制御プログラムの制御下で複数のＯＳが同時に動作することが可能な計算機システムの制御方法に関する。
【０００２】
【従来の技術】
一台の計算機上の制御プログラム下で複数のＯＳが同時に動作する計算機システムにおける共有メモリを用いた従来技術としては、制御プログラムが、複数のＯＳ上で動作する異なるプロセスの仮想メモリ空間を同一の物理メモリ空間に変換することにより、複数のプロセス相互間で共有する共有メモリ構築方法が知られている（例えば、特許文献１参照）。
【０００３】
【特許文献１】
特開２００２－１５７１３３号公報
【０００４】
【発明が解決しようとする課題】
動作させるＯＳが同じものである場合、各ＯＳに存在するカーネル等の一部のプログラムコードは各ＯＳで同じものを持つことになる。したがって、一台の計算機システム上で複数のＯＳを同時に動作させる場合には、各ＯＳがそれぞれ重複したプログラムコード領域をメモリ空間に持つことになり、メモリの消費に無駄が生じている。
【０００５】
本発明では前述した問題を解決し、一台の計算機上で複数のＯＳが同時に動作する計算機システムにおいて、メモリの有効利用を提供するものである。ただし、ここでいうプログラムコードとは、アプリケーションプログラムのように生成、消滅が頻繁に生じるようなものではなく、メモリに占有かつ固定的に配置されるようなもの（カーネル等）である。
【０００６】
【課題を解決するための手段】
本発明によれば前記課題は、制御プログラムの制御下で少なくとも２個の同一ＯＳを含む複数のＯＳが動作する計算機システムにおいて、同一ＯＳ上で動作する各同一ＯＳに共通なプログラムコードが配置される仮想メモリ空間を同一の物理メモリ空間に変換することにより達成される。
【０００７】
各ＯＳが制御プログラムに対して、共通プログラムコードの共有化を要求すると、要求を受け取った制御プログラムは、ＯＳが管理するメモリ空間変換テーブル内のメモリマッピングを、共有メモリ空間に指定された共通プログラムコード領域に対応する物理メモリ空間に変更する。この変更はＯＳに非通知行われるため、ＯＳが関与することなく共通プログラムコード領域の共有メモリ化を構築することができる。
【０００８】
共有メモリの構築および共有メモリ空間の使用に至る一連の処理において、ＯＳにはメモリ変換テーブルが書き換えられたことは通知せず、ＯＳによってメモリ空間変換テーブルの書き換えを検知されることもない。
【０００９】
ただし共有メモリ構築において、制御プログラムは、ＯＳが共有メモリとして使用したい

共通プログラムコード領域をメモリ空間に配置するおよその位置及びサイズを知る必要がある。制御プログラムは、共有メモリ化要求を発行したＯＳのメモリ空間をサーチしてバイナリコードの完全に一致する領域を見つけるため、偶然に一致するデータ領域の共有メモリ化を避けなければならない。そのために共有メモリ化要求の際に、共通プログラムコードの位置及びサイズを渡す必要がある。したがって、本発明が適用可能なＯＳは、共通プログラムコードがどのメモリ位置に配置されるか、及び共通プログラムコードのサイズを知ることの出来るＯＳが対象となる。

【００１０】

本発明はＯＳの介入を受けないため、前述の注意点を除けばＯＳの有する機能に制約されることがなくＯＳの選択が自由に行える。また、ＯＳ介入がない分、計算機システムの性能向上を図ることができる。

【００１１】

【発明の実施の形態】

以下に、本発明による計算機システムの実施形態を図面により詳細に説明する。

【００１２】

図１は本発明の一実施形態による計算機システムの構成を示すブロック図、図２は共有メモリによって共通プログラムコード領域を共有していない状態のメモリマップの構成を示す図、図３は共有メモリによって共通プログラムコード領域を共有している状態のメモリマップの構成を示す図、図４は共有メモリ制御プログラムが制御する共有メモリ制御テーブルの構成を示す図である。図１～４において、１００はハードウェア、２００はハイパバイザ（制御プログラム）、２１０は共有メモリ制御プログラム、２２０は共有メモリ制御テーブル、３００はＯＳａ、４００はＯＳｂ（ただしＯＳａとＯＳｂは同一のＯＳである）、３１０は共有メモリ化要求アプリケーションａ、４１０は共有メモリ化要求アプリケーションｂ、３２０はメモリ空間変換テーブルａ、４２０はメモリ空間変換テーブルｂ、１１００はＯＳａの仮想メモリ空間、１２００はＯＳｂの仮想メモリ空間、１５００は物理メモリ空間である。

【００１３】

本発明の一実施形態による計算システムは、図１に示すように、計算機システムを構成するハードウェア１００と、一台の計算機上で複数の異種または同種ＯＳが同時に動作することを可能とする制御プログラムであるハイパバイザ２００と、ハイパバイザ２００上で動作するＯＳａ３００、ＯＳｂ４００と、ＯＳａ３００、ＯＳｂ４００上で動作する共有メモリ化要求アプリケーションａ３１０、共有メモリ化要求アプリケーションｂ４１０とにより構成されている。そしてハイパバイザ２００は共有メモリ制御プログラム２１０を備え、この共有メモリプログラム２１０は、共有メモリ制御テーブル２２０を有している。またＯＳａ３００、ＯＳｂ４００は各ＯＳが管理するメモリ空間変換テーブルａ３２０、メモリ空間変換テーブルｂ４２０を有している。

【００１４】

なお、図１に示す計算機システムの例は、それぞれ１つの共有メモリ化要求アプリケーションが動作する２つのＯＳを有すると示しているが、ＯＳはさらに多数であってよくて、対象ＯＳ同士が同じＯＳであれば、同じＯＳでないものが混在していても構わない。

【００１５】

図２において、ＯＳａ３００の仮想メモリ空間１１００内では、共通プログラムコード領域ａが仮想メモリ空間の１１５１で動作し、ＯＳｂ４００の仮想メモリ空間１２００内では、共通プログラムコード領域ｂが仮想メモリ空間の１２５１で動作している。またＶａ－１及びＳａ－１は、仮想メモリ空間１１５１の開始アドレス及びサイズを示している。Ｖｂ－１及びＳｂ－１は、仮想メモリ空間１２５１の開始アドレス及びサイズを示している。

【００１６】

物理メモリ空間１５００内では、ハイパバイザ（制御プログラム）２００が物理メモリ空間１５７０で動作している。物理メモリ空間１５３０は、ハイパバイザ２００によってＯ

Ｓａのメモリ空間１１００に割り当てられた物理メモリ空間を示し、物理メモリ空間１５５０は、ハイパバイザ２００によってＯＳｂのメモリ空間１２００に割り当てられた物理メモリ空間を示している。共有メモリ構築前、物理メモリ空間１５３１は、仮想メモリ空間１１５１のマッピング先とされ、物理メモリ空間１５５１は、仮想メモリ空間１２５１のマッピング先とされている。

【００１７】

前述の状態で、物理メモリ空間１５３１は、共通プログラムコード領域１１５１に占有使用され、物理メモリ空間１５５１は、共通プログラムコード領域１２５１に占有使用されている。Ｐａ－１は、物理メモリ空間１５３１の開始物理アドレスを、Ｐｂ－１は、物理メモリ空間１５５１の開始物理アドレスをそれぞれ示している。

【００１８】

図２内に示しているテーブル３２１、４２１は、ハードウェアのメモリ空間変換を示すものであり、テーブル３２１は、ＯＳａ３００が管理するメモリ空間変換テーブルａ３２０の、仮想メモリ空間１１５１の変換に関する部分を示したものである。このテーブル３２１におけるＰａ－１（３２５）は、共通プログラムコード領域である仮想メモリ空間１１５１のマッピング先の物理メモリ空間の開始物理アドレスを示している。またテーブル４２１は、ＯＳｂ４００が管理するメモリ空間変換テーブルｂ４２０の、仮想メモリ空間１２５１の変換に関する部分を示したものである。このテーブル４２１におけるＰｂ－１（４２５）は、共通プログラムコード領域である仮想メモリ空間１２５１のマッピング先の物理メモリ空間の開始物理アドレスを示している。

【００１９】

図３は、図１に示す計算機システム動作時において、ＯＳａ３００とＯＳｂ４００のそれぞれの共通プログラムコード領域が共有メモリ空間を構築している状態でのメモリマップを示している。図３において、図２と異なるのは、仮想メモリ空間１１５１のマッピング先が物理メモリ空間１５３１ではなく、物理メモリ空間１５５１になっている点である。共有メモリ構築の際、メモリ空間変換テーブルａ３２０内の仮想メモリ空間１１５１のマッピング先の物理アドレス３２５がＰａ－１からＰｂ－１に変更される。この結果、物理メモリ空間１２５１のマッピング先が共有メモリ空間となり、物理メモリ１５３１は、どの仮想アドレス空間もマッピングされない状態となる。

【００２０】

図４は、共有メモリ制御プログラム２１０が制御する共有メモリ制御テーブル２２０の構成を示している。共有メモリ制御テーブル２２０は、エントリ名２２１、開始物理アドレス２２２、共有メモリサイズ２２３、アクセスフラグ２２４の各フィールドにより構成される。エントリ名２２１には、その共有メモリ空間を使用するエントリの名前が格納される。開始物理アドレス２２２には、共有メモリ空間として使用する物理メモリ空間の開始アドレスが格納される。

【００２１】

図５が示すフローを参照して、ＯＳｂ上の共有メモリ化要求アプリケーションｂ４１０が共有メモリ制御プログラムに対して共有メモリ空間の登録を要求したときの処理動作を説明する。

（１）ＯＳｂ４００上の共有メモリ化要求アプリケーションｂ４１０が共有メモリ制御テーブルへの登録要求を行うことによって処理が開始される。ＯＳｂの共通プログラムコードは、現在使用している仮想メモリ空間１２５１を共有メモリとして使用するため、共有メモリ制御プログラム２１０に対して共有メモリ制御テーブル２２０に仮想メモリ空間１２５１を共有メモリとして登録するよう要求する。なお、共有メモリ化要求アプリケーションが登録要求を行う際には、仮想メモリ空間１２５１の開始仮想アドレスＶｂ－１、サイズＳｂ－１、エントリ名及びアクセスフラグを共有メモリ制御プログラムに渡す必要がある（ステップ２０１０、２０２０）。

（２）この要求に対して、共有メモリ制御プログラム２１０は、共有メモリ制御テーブル２２０内をサーチし、指定されたエンリがエントリ名２２１に登録されているか否かを調

べ、同名のエントリがエントリ２２１に存在する場合、登録不可能なため、エラーリターンとなる（ステップ２０３０、２１５０）。
（３）ステップ２０３０で、エントリ名２２１に同一のエントリが存在しない場合、エントリ名２２１へ指定エントリを格納し、次にメモリ空間変換テーブルｂ４２０内を検索して、ＯＳｂの共有プログラムコードが使用している仮想メモリ空間１２５１に対応する物理アドレスＰｂ－１を求め、共有メモリ制御テーブル２２０内の開始物理アドレス２２２へ格納する（ステップ２０４０、２０５０）。
（４）次に共有メモリサイズＳｂ－１を２２３に格納し、さらにアクセスフラグ２２４へＯＳｂとそのバージョンを識別するための識別子を格納し、処理を終了する（ステップ２０６０、２０７０）。
【００２２】
前述の処理が終了した時点で、共通プログラムコード領域のマッピング先である物理アドレスＰｂ－１から大きさＳｂ－１までの物理メモリ空間１５５１は、共有メモリ空間として共有メモリ制御テーブルに登録されたことになる。
【００２３】
次に、図６が示すフローを参照して、ＯＳａ上の共有メモリ化要求アプリケーションａが共有メモリ制御テーブルに登録された共有可能メモリ空間の使用を要求したときの処理動作について説明する。
（１）共有処理が開始されると、ＯＳａの共通プログラムコードが使用している仮想メモリ空間１１５１のマッピングを現在のマッピング先である物理アドレスＰａ－１からサイズＳａ－１を持つ物理メモリ空間１５３１から、共有メモリテーブル２２０に登録されている共有メモリ空間にマッピングを変更するように共有メモリ制御プログラム２１０に要求する。この際、ＯＳａ上の共有メモリ化要求アプリケーションは、ＯＳｂ上の共有メモリ化要求アプリケーションが共有メモリテーブル２２０に登録したエントリ名と同一のエントリ名を使用して要求を発行する。また、仮想メモリ空間１１５１の開始仮想アドレスＶａ－１、サイズＳａ－１及びアクセスフラグを共有メモリ制御プログラムに渡す必要がある（ステップ２２１０、２２２０）。
（２）この要求に対して、共有メモリ制御プログラム２１０は、共有メモリ制御テーブル２２０内のエントリ名２２１に、要求されたエントリ名が登録されているか否かをサーチする。エントリ名２２１に同一のエントリ名がなければ、要求を実行することができないのでエラーリターンとする（ステップ２２３０、２３５０）。
（３）ステップ２２３０で、エントリ名２２１に同一エントリ名が登録されていた場合、次に、共有メモリサイズ２２３をチェックする。このチェックの結果、要求された共有メモリ空間のサイズが登録されている共有メモリのサイズと等しくなければ、すなわち、Ｓａ－１がＳｂ－１に等しくなければ要求を満たすことはできないためエラーリターンとする（ステップ２２４０、２３５０）。
（４）ステップ２２４０のチェックの結果、要求された共有メモリ空間サイズが登録されている共有メモリサイズと等しかった場合、次にアクセスフラグ２２４をチェックする。このチェックの結果、要求されたＯＳａのアクセスフラグが共有メモリ制御テーブル２２０に登録されているアクセスフラグと等しくなければ、ＯＳａとＯＳｂは異なり、要求を満たすことはできないため、エラーリターンとする（ステップ２２５０、２３５０）。
（５）ステップ２２５０のチェックの結果、要求されたアクセスフラグが登録されているアクセスフラグと等しかった場合、次にＯＳａの共通プログラムコードが使用する仮想メモリ空間１１５１のマッピング先である物理アドレスＰａ－１からサイズＳａ－１分のコードデータと、共有メモリ制御テーブル２２０に登録されているＰｂ－１からサイズＳｂ－１のコードデータと等しいか否かのチェックを行う。物理アドレスＰａ－１は、ＯＳａが管理するメモリ変換テーブルａ３２０内から仮想アドレスＶａ－１からサイズＳａ－１を持つ仮想メモリ空間１１５１に対応する物理メモリ空間１５３１の先頭アドレスである。このチェックの結果、等しくなければ要求を満たすことができないため、エラーリターンとする（ステップ２２６０、２３５０）。

（６）ステップ２２６０のチェックで、ＯＳａの共通プログラムコードデータと共有メモリ制御テーブルに登録された物理メモリ空間のデータが等しかった場合は、次に、開始物理アドレス３２５を、現在のマッピング先の物理アドレスＰａ－１から共有メモリ制御テーブル２２０内の該当エントリの開始物理アドレス２２２が示す共有メモリ空間１１５１の物理アドレスＰｂ－１に書き換える。このときＯＳａには、メモリ空間変換テーブルａ３２０が書き換えられたことは通知しない。以上で処理を正常終了する（ステップ２２７０）。

【００２４】

前述した処理が終了した時点で、ＯＳａ上で動作する共通プログラムコードａが使用する仮想メモリ空間１１５１とＯＳｂ上で動作する共通プログラムコードｂが使用する仮想メモリ空間１５２１が、同一の物理メモリ空間１５５１へマッピングされ、共有メモリとして使用可能となる。そのため、本来ＯＳａの共通プログラムコードが使用する物理メモリ空間１５３１がどのＯＳにもマッピングされていない状態となり、物理メモリ空間１５３１の使用が可能となる。したがってＯＳ同士の共通コードのマッピングを共有化することにより、メモリ消費量の削減が可能、メモリの有効利用が実現できる。

【００２５】

【発明の効果】

以上に説明したように本発明によれば、一台の計算機上で制御プログラムの制御下で複数のＯＳが同時に動作する計算機システムにおいて、同一のＯＳ上で動作するそれぞれのＯＳに共通なプログラムコードが使用する仮想メモリ空間のマッピング先を共有メモリによって共有化することで、計算機システム全体でメモリ消費量が削減され、メモリの有効利用が実現可能となる。

【図面の簡単な説明】

【図１】本発明の一実施形態による計算機システムの構成を示すブロック図である。

【図２】ＯＳａとＯＳｂの共通プログラムコードが動作するそれぞれのプロセスが共有メモリを構築していない状態のメモリマップの構成を示す図である。

【図３】ＯＳａとＯＳｂの共通プログラムコードが動作するそれぞれのプロセスが共有メモリを構築している状態のメモリマップの構成を示す図である。

【図４】共有メモリの制御プログラムが制御する共有メモリ制御テーブルの構成を示す図である。

【図５】ＯＳｂ上の共有メモリ化要求アプリケーションが共有メモリ空間の登録要求をしたときの処理動作を説明するフローチャートである。

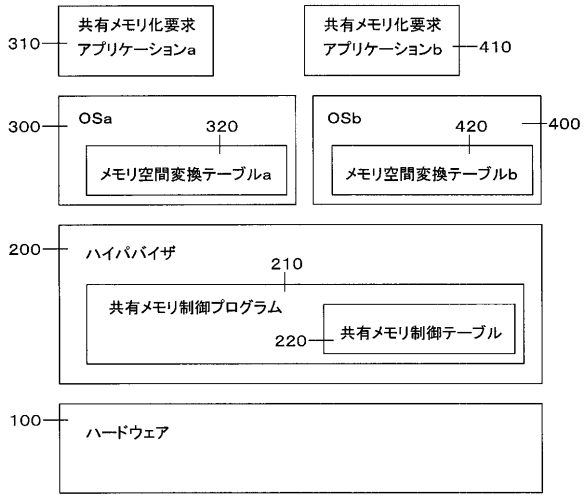
【図６】ＯＳａ上の共有メモリ化要求アプリケーションが共有メモリ空間の使用要求をしたときの処理動作を説明するフローチャートである。

【符号の説明】

１００　ハードウェア
２００　ハイパバイザ
２１０　共有メモリ制御プログラム
２２０　共有メモリ制御テーブル
３００　ＯＳａ
４００　ＯＳｂ
３１０　共有メモリ化要求アプリケーションａ
４２０　共有メモリ化要求アプリケーションｂ
１１００　ＯＳａの仮想メモリ空間
１５００　ＯＳｂの仮想メモリ空間

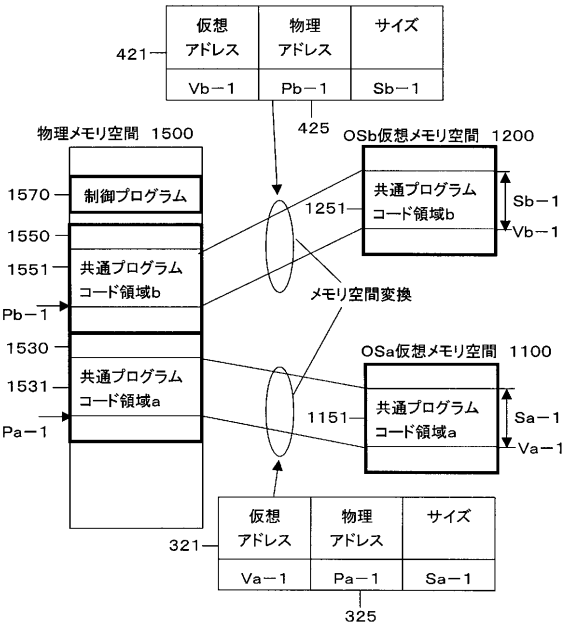
【図１】

図1



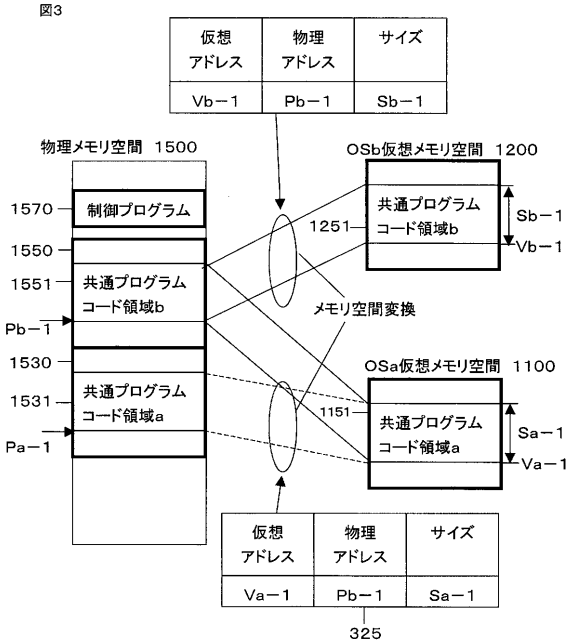
【図２】

図2



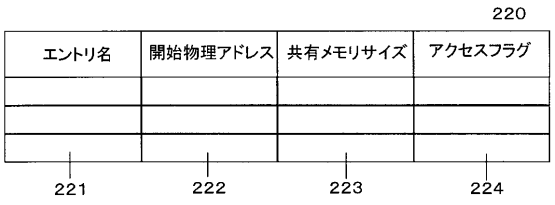
【図３】

図3



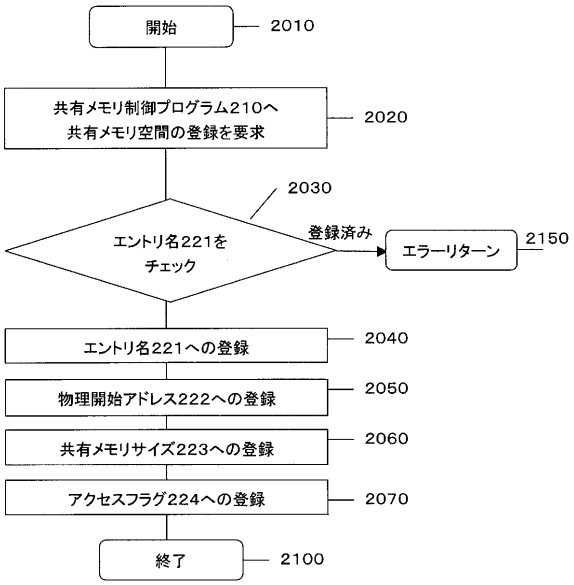
【図４】

図4



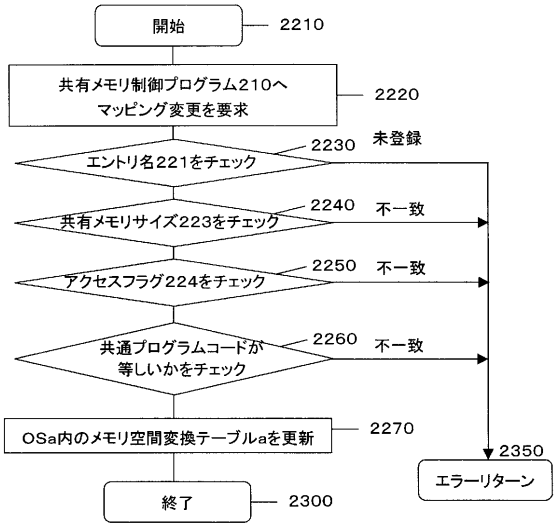
【図５】

図5


開始
2010
共有メモリ制御プログラム210へ
共有メモリ空間の登録を要求
2020
2030
エントリ名221を
チェック
登録済み
エラーリターン
2150
エントリ名221への登録
2040
物理開始アドレス222への登録
2050
共有メモリサイズ223への登録
2060
アクセスフラグ224への登録
2070
終了
2100

【図６】

図6


開始
2210
共有メモリ制御プログラム210へ
マッピング変更を要求
2220
エントリ名221をチェック
2230
未登録
共有メモリサイズ223をチェック
2240
不一致
アクセスフラグ224をチェック
2250
不一致
共通プログラムコードが
等しいかをチェック
2260
不一致
OSa内のメモリ空間変換テーブルaを更新
2270
終了
2300
エラーリターン
2350