(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3659062号

(P3659062)

(45)発行日　平成17年6月15日(2005.6.15)　(24)登録日　平成17年3月25日(2005.3.25)

(51)Int.Cl.[7]	F I		
G06F 9/46	G06F	9/46	350
G06F 9/48	G06F	13/10	330A
G06F 13/10	G06F	9/46	311G

請求項の数 3　(全 24 頁)

(21)出願番号　特願平11-140914
(22)出願日　平成11年5月21日(1999.5.21)
(65)公開番号　特開2000-330806(P2000-330806A)
(43)公開日　平成12年11月30日(2000.11.30)
審査請求日　平成13年9月13日(2001.9.13)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100075096
弁理士　作田　康夫
(72)発明者　遠藤　芳則
茨城県日立市大みか町七丁目１番１号
株式会社　日立製作所　日立研究所内
(72)発明者　中村　浩三
茨城県日立市大みか町七丁目１番１号
株式会社　日立製作所　日立研究所内

最終頁に続く

(54)【発明の名称】計算機システム

(57)【特許請求の範囲】

【請求項１】

複数個のオペレーティングシステムと、複数個のオペレーティングシステムを切り替えるＯＳ切替手段と、複数のオペレーティングシステムで共有する周辺デバイスとを有する計算機システムにおいて、

割込み要因毎に起動するオペレーティングシステムを記憶する優先割込みテーブルと、前記周辺デバイスからの割込み内容と起動するオペレーティングシステムを対応付けた割込み内容判定テーブルを備え、

前記周辺デバイスのデータ入出力を行うデータ入出力サーバを一つのオペレーティングシステムに備え、

前記ＯＳ切換手段は、割込み要因から優先割込みテーブルを参照し、該割込み要因に対応するオペレーティングシステムに動作を切り替えると共に、該オペレーティングシステムが備える割込み処理手段を呼び出し、該割込み要因が複数のオペレーティングシステムで共有する周辺デバイスからの割込みである場合には、前記一つのオペレーティングシステムの割込み処理手段を呼び出し、

当該一つのオペレーティングシステムの割込み処理手段は、

割込み要因が複数のオペレーティングシステムで共有する割込みか否かを判定し、

共有する割込みと判断したときは、割込みを生成した周辺デバイスにアクセスして割込みの内容を解析し、

該解析の結果と前記割込み内容判定テーブルを比較して割込み対象とするオペレーティ

ングシステムを判定して、該オペレーティングシステムが備える割込み処理手段を呼び出し、
前記一つのオペレーティングシステム以外に備えられた前記周辺デバイスとデータを入出力するデータ入出力クライアントは、前記データ入出力サーバにデータ入出力を依頼すること
を特徴とする計算機システム。
【請求項２】
請求項１に記載の計算機システムにおいて、
前記データ入出力サーバは最も優先順位の高いオペレーティングシステムが管理する領域に実装することを特徴とする計算機システム。
【請求項３】
請求項１又は２に記載の計算機システムにおいて、
複数のオペレーティングシステムの間で互いに通信するＯＳ間通信手段とを備え、
該ＯＳ間通信手段を用い前記データ入出力サーバと前記データ入出力クライアントとの間で通信することを特徴とする計算機システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は複数個のオペレーティングシステムを単一のプロセッサで動作させる計算機システムにおける周辺デバイスとのデータ入出力方法を対象とし、特に、複数個のオペレーティングシステムが同一の周辺デバイスを共有に使用するための制御方法に関連する。
【０００２】
【従来の技術】
通常の計算機では、１つのオペレーティングシステムが動作し、それが計算機のプロセッサ，メモリ、および２次記憶装置といった計算機資源を管理し、計算機が効率よく動作できるように資源スケジュールを実施している。ところで、オペレーティングシステムには様々な種類がある。バッチ処理に優れるものや、事務処理といったＧＵＩに優れるもの、実時間処理に優れているものなど様々である。これら複数の特徴を引き出すため、複数あるオペレーティングシステムを１台の計算機で同時に実行したいというニーズがある。例えば、大型計算機においては、実際の業務に伴うオンライン処理を実行するオペレーティングシステムと、開発用のオペレーティングシステムを１台の計算機で動作させたいという要求がある。あるいはＧＵＩが整っているオペレーティングシステムと実時間性に優れているオペレーティングシステムを同時に稼働させたいという要求もある。
【０００３】
１台の計算機上で複数のオペレーティングシステムを動作させる機構としては、大型計算機で実現されている仮想計算機方式（ＯＳシリーズ第１１巻，ＶＭ，岡崎世雄著，共立出版株式会社）がある。仮想計算機方式では、仮想計算制御プログラムが全ハードウェアを占有して管理し、それを仮想化して仮想計算機を構成する。仮想計算機を構成する制御部は、物理メモリ，入出力機器装置，外部割り込み等を仮想化する。例えば、分割された物理メモリは、仮想計算機に対してはあたかも０番地から始まる物理メモリのように振る舞い、入出力装置を識別する装置番号も同様に仮想化されている。更に、磁気ディスクの記憶領域も分割して磁気ディスク装置の仮想化まで実現している。
【０００４】
一方、１台の計算機で複数のオペレーティングシステムのインタフェースを提供する技術としてマイクロカーネルがある。マイクロカーネルでは、マイクロカーネルの上にユーザに見せるオペレーティングシステム機能を提供するオペレーティングシステムサーバを構築し、ユーザはそのサーバを経由して計算機資源を利用する。オペレーティングシステム毎のサーバを用意すれば、ユーザに様々のオペレーティングシステム環境を提供することができる。
【０００５】

【発明が解決しようとする課題】
例えば、車載ナビゲーションシステムでは、外部環境変化に対して即座に応答するというリアルタイム性と、いかなる場合でもシステムが停止しないという信頼性が重要視されている。このため、割込み応答性が高く、かつ、コンパクトでモジュール構成を有するリアルタイム用オペレーティングシステム（リアルタイムＯＳ）が用いられることが多い。しかしながら、リアルタイムＯＳは、リアルタイム性・信頼性を重要視する反面、人間とのインタフェースに優れているとは言い難い。一方、一般のパソコン（ＰＣ）に用いられている事務処理用オペレーティングシステム（事務処理ＯＳ）は、ＧＵＩといった直接画面を操作する環境が準備され、対人間インタフェースに優れている。このため、従来はリアルタイムＯＳを使用していた分野にも事務処理ＯＳのユーザインタフェースを使用したいという要求が高まっている。しかし、事務処理ＯＳは人間との対話を主な処理としているため、割込み応答性よりも処理のスループットを重要視し、比較的長時間割込み禁止状態のまま処理を実行することもある。また、コンパクトな構成を有するリアルタイムＯＳに対し、信頼性において匹敵するとは言い難い。

【０００６】
しかしながら、大型計算機上で複数の仮想計算機(オペレーティングシステム)を並行して動作させる方式と同様に、組み込みシステムにおいて同一計算機システム上で事務処理ＯＳとリアルタイムＯＳとを動作させ、必要に応じてこれらのオペレーティングシステムを切り替えることができれば、優れたユーザインタフェースとリアルタイム性・信頼性とを両立させることができる。マイクロプロセッサの性能向上を考慮すれば、複数のオペレーティングシステムを一つの計算機システム上で動作させることは、大型計算機だけに許された技術ではなくなってきている。

【０００７】
大型計算機ではマルチユーザに対応する必要があるが、組み込み機器では最低限シングルユーザに対応できればよい。そこで、リアルタイムＯＳと事務処理
ＯＳを一つの計算機システムで動作させるとき、それぞれのオペレーティングシステムの重要性を考慮し、リアルタイムＯＳが実行すべき処理がある場合はリアルタイムＯＳを優先的に動作させ、リアルタイムＯＳ上で実行するタスクがなくなったとき事務処理ＯＳを実行するという最も単純なオペレーティングシステム切替え方式が適用可能である。このようなオペレーティングシステムの切替え方法では、周辺デバイスとの入出力において、入出力装置毎に割込み要求を受けるオペレーティングシステムを静的に決定し、発生した割込み番号を判定することでオペレーティングシステムを切り替える方法が用いられる。例えば、前記手法を車載ナビゲーションシステムに適用したとき、位置決定に必要なセンサからの割込みはリアルタイムＯＳの、人間とのインタフェースに用いるスイッチやディスプレイコントローラからの割込みは事務処理ＯＳの割込みハンドラを起動するようにオペレーティングシステムを切り替える。しかしながら事務処理ＯＳの信頼性は必ずしも高くないため、本適用例では事務処理ＯＳの処理が停止すると同時にユーザインタフェースも使用不可能になり、車載ナビゲーションシステム全体の信頼性をも低下させてしまう。前記課題を解決するには、絶対に停止させたくない処理はリアルタイムＯＳ側に、停止を許容する処理は事務処理ＯＳにそれぞれ分離すればよいが、例えばユーザインタフェースのための入出力処理をリアルタイムＯＳ側で処理してしまうと、ユーザインタフェースに優れた事務処理
ＯＳを導入した意味がなくなってしまう。よって、入出力デバイスを複数のオペレーティングシステム、即ち事務処理ＯＳとリアルタイムＯＳの両方から使用できることが必須となり、上記説明した割込み番号から入出力処理を実行するオペレーティングシステムを静的に決定する方法では不十分であり、実用に耐えないという課題を有する。

【０００８】
本発明の目的は、複数の相異なるオペレーティングシステムを一つの計算機システム上で動作させるマルチオペレーティングシステム制御装置において、計算機に備えられた入出力デバイスを複数のオペレーティングシステムで共有する仕組みを提供することにある。

【０００９】
【課題を解決するための手段】
本発明では、上記課題を解決するため、第１の発明では、割込み要因毎に起動するオペレーティングシステムを記憶する優先割込みテーブルと、優先割込みテーブルを書き換える優先割込みテーブル書き換え手段とを備え、ＯＳ切換手段は割込み要因から優先割込みテーブルを参照し対応するオペレーティングシステムに切り替えると共に、該オペレーティングシステムが備える割込み処理手段を呼び出すことで計算機に備えられた入出力デバイスを複数のオペレーティングシステムで共有する。
【００１０】
第２の発明では、複数のオペレーティングシステムから共有する周辺デバイスと、前記共有する周辺デバイスとデータ入出力するデータ入出力サーバを該一つのオペレーティングシステムに備え、前記オペレーティングシステム以外に備えられた前記共有する周辺デバイスとデータ入出力するデータ入出力クライアントは、データ入出力サーバにデータ入出力を依頼し、データ入出力サーバが周辺デバイスとデータ入出力した結果を受信しデータ入出力処理することで計算機に備えられた入出力デバイスを複数のオペレーティングシステムで共有する。
【００１１】
【発明の実施の形態】
本発明の実施例を図面で説明する。
【００１２】
図１に本発明の第一の実施例における全体構成を示す。通常、計算機システムは、プロセッサ１００，メモリ１０１，入出力制御装置１０２，ディスプレイ
１０４，スイッチ１０５，センサ１０６などから構成される。プロセッサ１００，メモリ１０１，入出力制御装置１０２はプロセッサバス１０３によって接続される。プロセッサ１００は複数個のオペレーティングシステムを動作させるためのマイクロプロセッサである。メモリ１０１は、オペレーティングシステム116,１１７，各オペレーティングシステム上で動作するタスクプログラム１１０～
１１５，各オペレーティングシステムのための入出力ドライバプログラム１２０，１２１，割込みハンドラプログラム１２２，１２３，ＯＳ間制御機能プログラム１２４を保持する。これらのプログラムはプロセッサ１００によって読み出されて実行される。
【００１３】
入出力制御装置１０２は、画面表示用デバイスであるディスプレイ１０４，ユーザからの指示を受けるスイッチ１０５，周囲環境の変化を測定するセンサ106等を接続する。また、工場／プラント制御用、あるいは、組込み向け計算機システムを実現する場合、入出力制御装置１０２にはネットワーク１０９が接続されることがある。ネットワーク１０９には、通信機器などの入出力機器が接続される。なお、入出力制御装置１０２に接続される入出力装置１０４～１０６のうち、いずれかまたは全ては、システム構成によって省略されることもあり、さらに多様な入出力装置が接続されることもある。入出力制御装置１０２は、割込み信号線１０８によってプロセッサ１００と接続され、入出力動作完了などを通知する。図１では説明のため割込み信号線１０８とプロセッサバス１０３が別の装置であるように記載しているが、実際には、割込み信号線１０８はプロセッサバス１０３の一部である。プロセッサ１００内部にはタイマ装置１０７が設けられており、一定周期毎に内部割込みを発生させる。タイマ装置１０７からの割込みは、オペレーティングシステムの計時などに使用される。
【００１４】
プロセッサ１００は割込み信号線１０８によって通知される外部割込みやタイマ装置１０７などからの内部割込みをマスクできる機能を装備するものとする。割込みマスクとは、プログラムが割込みマスクを解除するまで、特定の割込みが入ることを遅延させる機能である。一般に、割込みマスク機能には、下記の三種類が存在する。
【００１５】

（１）全割込みマスク：全ての割込みをマスクする。
【００１６】
（２）個別割込みマスク：個々の割込みをそれぞれマスクできるようにする。
（３）割込みレベルマスク：各割込みにレベルを設定し、指定レベル以下の割込みをマスクする。
【００１７】
プロセッサ１００の種類によって、上記（１）（２）の組合せ、または（１）（３）の組合せのいずれかを装備することが多い。後者の組合せを装備するプロセッサを採用する場合、対応する入出力装置の重要性や最小応答時間に応じ割込みレベルを割り当てることになる。例えば、短い応答時間が要求されるネットワーク１０９からの割込みを、ディスプレイ１０４，スイッチ１０５および記憶装置１０６などからの割込みより高いレベルに設定する。
【００１８】
本実施例では、計算機システム内に二つのオペレーティングシステム１１６，１１７が存在する場合を説明する。オペレーティングシステム１１６，１１７は、各自に割り当てられたメモリ、及びプロセッサ資源を用い、タスク１１０～１１５を実行する。図１では、オペレーティングシステム数が２個，全タスク数が６個の例を示しているが、これらの数値よりも多い、または、少ないオペレーティングシステム，タスクを実装することも可能である。本実施例では、オペレーティングシステム数の動的な変更は想定しないが、各オペレーティングシステムが動的にタスク生成・削除を行うことは可能である。さらに本実施例では、オペレーティングシステム１１６が事務処理ＯＳ，オペレーティングシステム117がリアルタイムＯＳであると仮定するが、実際に、各オペレーティングシステムがいかなる種類であっても、本発明で述べる技術は適用可能である。タスク110～１１２は事務処理ＯＳで実行される事務処理タスクであり、タスク１１３～１１５はリアルタイムＯＳで実行されるリアルタイムタスクである。さらに、リアルタイムＯＳでの応答時間を保証するため、事務処理ＯＳよりリアルタイムＯＳの割込み優先順位が高いと仮定し以下説明する。前記仮定では、リアルタイムＯＳのいずれかのタスクが実行状態の場合はリアルタイムＯＳ１１６がプロセッサ資源が使用し、すべてのタスクがアイドル状態またはウェイト状態の場合は事務処理ＯＳにコンテクストが切り替えプロセッサ資源が使用する。
【００１９】
事務処理ＯＳ１１６及びリアルタイムＯＳ１１７を一つのプロセッサ上で連携しながら動作させるため、ＯＳ間制御機能１２４を備える。ＯＳ間制御機能124は、事務処理ＯＳとリアルタイムＯＳ間の両者からアクセス可能な共有メモリ１２５，事務処理ＯＳとリアルタイムＯＳ間でメッセージ転送を行うＯＳ間通信機能１２６，事務処理ＯＳとリアルタイムＯＳ実行環境を切り替えるＯＳコンテクスト切換１２７，発生した割込みを各オペレーティングシステムに振り分ける共通割込みハンドラ１２８，割込み要求に対応するオペレーティングシステムや割込み開始アドレスを記憶する優先管理テーブル１２９とを備える。
【００２０】
共有メモリ１２５は、オペレーティングシステム間で高速なデータ交換を実現することを目的とし、事務処理ＯＳとリアルタイムＯＳの両者から読み書きが可能である。ここで、データ競合を防止するためセマフォ機能を用い排他的に共有メモリをアクセスすることが望ましい。ＯＳ間通信機能１２６は、各オペレーティングシステムに対応するメッセージキューを準備し、各オペレーティングシステム間でメッセージを受け渡す。共通割込みハンドラ１２８は、割込み要求が発生したとき最初に呼び出され、優先割込みテーブル１２９に記憶された割込み番号毎のオペレーティングシステムを判定し、対応する割込みハンドラ１２２，１２３を呼び出す。ＯＳコンテクスト切換１２７は、割込み要求やＯＳ間制御機能の呼び

出しによりオペレーティングシステムの切換が必要と判断されたときコンテクストを切り替える。割込み管理テーブル１２９は、割込み番号毎に呼び出される割込みハンドラが所属するオペレーティングシステム情報や、各オペレーティングシステム毎の割込みハンドラの開始アドレス情報を保持する。従来の方法では、割込み管理テーブル１２９をシステム起動時に静的に決定していたため、システム動作中にその内容を書き換えることはできなかった。本システムでは、システム動作中でも割込み管理テーブル１２９を書き換える手段を提供することで、一つの入出力装置を複数のオペレーティングシステムが共用することを可能としたことを特徴とする。

【００２１】

オペレーティングシステム１１６，１１７は、入出力装置との間でデータ入出力を処理する入出力ドライバ１２０，１２１，入出力装置からの割込み要求を受ける割込みハンドラ１２２，１２３を備える。割込みハンドラ１２２，１２３は、共通割込みハンドラ１２８から呼び出され、ユーザが定義した割込み処理プログラムを実行する。入出力ドライバ１２０，１２１は、各タスクから入出力装置を制御するためのインタフェースを提供し、入出力装置を読み書きすることでデータ入出力したり、入出力装置を制御するといった機能を提供する。リスケジューラ１１８，１１９は、タスクの生成・削除・停止・再開や、外部割込み・内部割込みに伴ってタスク切り替えを行わなければならない場合に起動される。リスケジューラは直前に実行していたタスクの実行環境（プログラムカウンタ，ステータスレジスタ，汎用レジスタなど）をタスク管理テーブルに格納し、新たに実行すべきタスクを決定し、そのタスクの実行環境をタスク管理テーブルから取り出し各レジスタに設定することで、選択したタスクを実行する。

【００２２】

図２に、本発明が前提とするプロセッサ１００の内部構成例を示す。キャッシュメモリ１３０はメモリ１０１上のデータまたは命令を一時的に格納するバッファ記憶装置である。ＣＰＵ１３１は演算回路であり、メモリ１０１若しくはキャッシュメモリ１３０上に存在する命令を順次読み出して実行する。命令実行に際して、演算結果を一時的に保持するための汎用レジスタ１３２、実行命令アドレスを保持するプログラムカウンタ１３３，実行状態を保持するステータスレジスタ１３４を用いる。キャッシュメモリ１３０，ＣＰＵ１３１，汎用レジスタ132,プログラムカウンタ１３３、ステータスレジスタ１３４は、互いに、データ転送を行うデータバス１３５、アドレス指定を行うアドレスバス１３６によって接続されている。

【００２３】

割込み信号線１０８とタイマ装置１０７は割込みコントローラ１３７に接続される。割込みコントローラ１３７は、ＣＰＵ１３０に対して割込み状態信号138を生成する役割を有する。割込み状態信号１３８は、プロセッサ１００に対して現在如何なる種類の割込みが発生しているかを示す信号線である。通常、ステータスレジスタ１３４は現在の割込みマスクに関する情報を有しており、割込み状態信号１３８で指定される割込みを受け付けるか否かを決定する。割込みを受け付ける場合、割込みコントローラ１３７は、プログラムカウンタ１３３，ステータスレジスタ１３４などの値を書き換え、対応する割込み処理プログラムを実行させる。

【００２４】

ステータスレジスタ１３４の構成例を図３に示す。ここでは、プロセッサ100が全割込みマスク機能と割込みレベルマスク機能とを装備している場合の例を示す。図３の例では、ステータスレジスタ１３４が割込みブロックビット１４０と割込みマスクレベルフィールド１４１を有する。割込みブロックビット１４０がＯＮである場合、プロセッサ１００に対する全ての割込みがマスクされる。割込みマスクレベルフィールド１４１は現在の割込みマスクレベル値を示し、これ以下の割込みレベルは受け付けられない。図３の例では、割込みマスクレベルフィールド１４１は４ビット長である。このため、合計１６種類のマスクレベルを指定可能であるが（通常、割込みレベル０は「割込みが発生していない」ことを意味し、割込みマスクレベルを０とすることは「割込みマスクを行わない」ことを意

味するため、実質的には１５種類となる）、割込みマスクレベルフィールド
１４１のビット数を変更することにより、受理できる割込みレベルの種類を増減させることができる。

【００２５】

図４は、プロセッサ１００が全割込みマスク機能と個別割込みマスク機能とを装備している場合のステータスレジスタ１３４である。この例では、ステータスレジスタ１３４は実際には２つのレジスタ（実行状態レジスタ１４２と割込みマスクレジスタ１４３）から構成される。図３と同様、実行状態レジスタ１４２内に割込みブロックビット１４０が設けられている。割込みマスクレジスタ１４３内の割込みマスクビット１４４～１４７はそれぞれ別々の割込みに対応しており、該割込みマスクビットのいずれかをＯＮとした場合、対応する割込みが受け付けられなくなる。図３に示すステータスレジスタの例は、図４のステータスレジスタの特殊例である。例えば、割込みマスクビット１４４のみＯＮとなっている状態をレベル１とし、割込みマスクビット１４４，１４５の二つがＯＮとなっている状態をレベル２，割込みマスクビット１４４～１４６の三つがＯＮとなっている状態をレベル３，…、のように対応させることができる。このため、以降、ステータスレジスタ１３４は図４に示す構成を有するとして説明する。

【００２６】

一般的なプロセッサでは、割込みを受理すると、ハードウェアによって自動的に割込みブロックビット１４０がＯＮに書き換えられる。必要に応じて、割込み処理プログラムが割込みブロックビット１４０をＯＦＦに書き換えて割込み受付けを許可することができる。また、オペレーティングシステムおよびタスクが、一時的に割込みブロックビット１４０や割込みマスクレジスタ１４３の内容を書き換え、特定割込みの受付けを待たせることもできる。割込みマスク機能は、排他制御の実現や、割込み処理実行中に再び同一割込みが発生することを避けるために用いられる。

【００２７】

図５に割込みハンドラ１２２，１２３，共通割込みハンドラ１２８，ＯＳコンテクスト切替１２７，割込み管理テーブル１２９の詳細を示す。

【００２８】

割込みハンドラ１２２，１２３は、割込み発生時のレジスタなどの保存，一時変数の保持を行うための領域として、それぞれ、割込みスタック１５１，１５３を所有する。割込み発生時には、割込み発生直前のレジスタ値を保存し、割込み処理終了後にはこのレジスタ値を元に復帰するため、割込みスタック１５１，
１５３を備える。使用するプロセッサ１００の種類によっては、割込み発生時に自動的にレジスタが切り替わり、かつ、割込み処理終了後に切替え前のレジスタに戻す機能を備えているものもある。しかしながら、多重割込みが可能なシステムを考慮した場合には、このようなハードウェアを使用していても、割込みスタックが必要となる（割込み処理中により高い緊急性を有する割込みが発生した場合、新たに発生した割込み処理を優先的に実行し、次いで、元の割込み処理に戻る必要がある）。割込みハンドラ１２２，１２３は、割込みスタック１５１，
１５３において、どの領域まで使用したかを示すためのポインタとして、割込みスタックポインタ１５０，１５２を所有する。割込みハンドラ１２２，１２３は、割込みスタック、および、割込みスタックポインタを用いて割込み発生前の実行環境（レジスタなど）を保存し、必要な割込み処理を実行する。割込み処理終了後、割込み発生前の実行環境を復帰して元々動作していたプログラムの実行を継続するように動作する。

【００２９】

共通割込みハンドラ１２８は、発生した割込みを前記割込みハンドラ１２２，１２３に対して振り分ける役割を有する。このため、共通割込みハンドラ１２８は、現在実行しているオペレーティングシステムが事務処理ＯＳ１１６であるかリアルタイムＯＳ１１７であるかを記憶する領域として、実行ＯＳ記憶変数154を所有する。例えば、図５に示した時点で事務処理ＯＳが処理を実行しているものと仮定すると、この場合、実行ＯＳ記憶変数

１５４には、「事務処理ＯＳ」が記憶されている。勿論、実行ＯＳ記憶変数１５４が文字列「事務処理ＯＳ」を記憶することは非常に非効率的であるため、例えば、事務処理ＯＳであれば０，リアルタイムＯＳであれば１といった整数形式で記憶するとよい。

【００３０】

割込み管理テーブル１２９は、発生した割込みがいずれのオペレーティングシステムの割込みハンドラを起動するかを示す優先割込みテーブル１５６、および各オペレーティングシステムが備える割込みハンドラの開始アドレスを示す割込みアドレステーブル１５５を有する。優先割込みテーブル１５６は、例えば図７に示したような個々の割込みがどちらのオペレーティングシステムで処理されるべきかを示す対応表で構成され、フラグ１が立っているオペレーティングシステムが備える割込みハンドラに処理を依頼する。さらに、複数のオペレーティングシステムで共有する入出力装置からの割込みにはフラグ１を、共有しない割込みにはフラグ０を記憶する。本実施例では、１６種類の割込み要因が存在すると仮定すると、割込み対応テーブル１５６も１６個のエントリ（ＩＲＱ＃０～　　ＩＲＱ＃１５）の構成になる。なお、本説明では優先ドライバテーブル１６０にフラグを記録する方法を想定したが、何らかの識別可能な情報を記録すればいかなる方法でも実現可能である。割込みアドレステーブル１５５は、例えば図６に示すような各々のオペレーティングシステムが備える割込みハンドラの開始アドレスを保持する表である。

【００３１】

本発明では、一つの入出力機器を複数のオペレーティングシステムから共用するという目的を達成するため、共通割込みハンドラ１２８が受けた割込み要求から起動するオペレーティングシステムを動的に変更する仕組みを提供する。優先割込みテーブル１５６の内容は、ユーザオペレーションや入出力機器の使用状況によって更新され、共通割込みハンドラ１２８は優先割込みテーブル１５６の内容に従って割込みハンドラを呼び出すように動作する。さらに詳細を説明すると、共通割込みハンドラ１２８は、実行ＯＳ記憶変数１５４を参照し、現在実行しているオペレーティングシステムの種類を判定する。オペレーティングシステムの種類が優先割込みテーブル１５６から得られた振り分け先オペレーティングシステムに一致しなければ、ＯＳコンテクスト切替えモジュール１２７にオペレーティングシステム切替えを依頼する。前記オペレーティングシステム切替えが終了した場合、ないしオペレーティングシステムが一致し切り替えが不要な場合は、該当するオペレーティングシステムの割込みハンドラに処理を依頼する。なお、実行ＯＳ記憶変数１５４はＯＳコンテクスト切替モジュール１２７からも参照される。このように、ＯＳ間制御機能プログラム内の各モジュールが有する内部構造は、個々のモジュールによって占有されるものとは限らず、各モジュールが必要に応じて共用できるものとする。

【００３２】

ＯＳコンテクスト切替えモジュール１２７は、ＯＳ間制御機能の呼び出しや割込み発生の結果、オペレーティングシステムを切り替えなければならない場合に起動される。二つのオペレーティングシステムを切り替えるため、これらの実行環境（すなわち、レジスタ値）を保存しておくための領域を所有する。図中では、事務処理ＯＳの実行環境を保存するための領域として、事務処理ＯＳ用保存コンテクスト１５７，リアルタイムＯＳの実行環境を保存するための領域として、リアルタイムＯＳ用保存コンテクスト１５８を用意する。ＯＳコンテクスト切替モジュール１２７は、コンテクストを切り替える際に現在実行しているオペレーティングシステムに対応する保存コンテクストに実行環境を保存し、次に、もう一方の保存コンテクストから実行環境を読み出して、プロセッサ１００のレジスタに設定する。これにより、オペレーティングシステムの切替えを実施できる。以降、図５で示したモジュールのうち、ＯＳコンテクスト切替モジュール１２７，共通割込みハンドラ１２８，割込みハンドラ１２２の処理フローの説明を行う。割込みハンドラ１２３に関しては、割込みハンドラ１２２と同様の処理フローを有するため、説明を省略する。

【００３３】

図８は、ＯＳコンテクスト切替モジュール１２７の処理フローである。ＯＳコンテクスト切替モジュール１２７は、実行するオペレーティングシステムを切り替えなければならな

い場合にのみ呼び出され、切替えが必要か否かのチェックは呼出し前に行われている。従って、実行ＯＳ記憶変数１５４を参照し、どちらのオペレーティングシステムへ切り替えなければならないかをチェックする（処理１６０）。ここで、事務処理ＯＳ１１６からリアルタイムＯＳ１１７への切替えを行う場合、事務処理ＯＳ用保存コンテクスト１５７内に使用中レジスタの値を退避する（処理１６１）。次に、リアルタイムＯＳ用保存コンテクスト１５８から実行環境を復帰して、レジスタに設定することにより（処理１６２）、以前退避した状態からリアルタイムＯＳ１１７を再実行させることができる。リアルタイムＯＳ１１７から事務処理ＯＳ１１６への切替えを行う場合、逆に、リアルタイムＯＳ用保存コンテクスト１５８に使用中レジスタの値を退避し(処理１６３)、次に、事務処理ＯＳ用保存コンテクスト１５７からレジスタを復帰する（処理１６４）。いずれの場合においても、最後に、切替え後のオペレーティングシステムの種類を実行ＯＳ記憶変数１５４に書き込む（処理１６５）。

【００３４】

図９は、共通割込みハンドラ１２８の処理フローを示している。一般に、単一のオペレーティングシステムによって制御される計算機システムでは、一旦、全ての割込みを割込みハンドラと呼ばれるモジュールが処理し、次いで、各プログラムに振り分ける。しかしながら、本実施例で説明するような複数個のオペレーティングシステムを動作させる計算機システムでは、この更に前に、共通割込みハンドラが全ての割込みを受付け、これを対応するオペレーティングシステムの割込みハンドラに振り分ける形となる。各オペレーティングシステムに割込みを振り分けるに当たって、割込み発生時に、割込み対象とは別のオペレーティングシステムが処理を実行していることもある。この場合、割込みに対応するオペレーティングシステムに切り替える必要があり、共通割込みハンドラはこのような役割も有することになる。

【００３５】

共通割込みハンドラ１２８は、割込み発生後、第一に、実行ＯＳ記憶変数154の内容を取り出して、現在実行中のオペレーティングシステムが事務処理0S116であるかリアルタイムＯＳ１１７であるかをチェックする(処理１７０)。次に、優先割込みテーブル１５６を参照し、発生した割込み要求がいずれのオペレーティングシステムの割込みハンドラを起動すべきかを入手する(処理１７１)。図６に示す優先割込みテーブル１５６に記載された値を用いた場合を想定すると、割込みＩＲＱ＃０が発生した場合にはリアルタイムＯＳ１１７，割込みＩＲＱ＃１が発生した場合には事務処理ＯＳ１１６，…、割込みＩＲＱ＃１５が発生した場合にはリアルタイムＯＳ１１７、というように対応するオペレーティングシステムを入手することができる。次に、得られた割込み対象オペレーティングシステムが実行中オペレーティングシステムに等しいか否かをチェックする（処理172)。割込み要求が現在実行中のオペレーティングシステムに対するものでなければ、一旦、オペレーティングシステムの切替えを行わなければならない。この切替え処理は、ＯＳコンテクスト切替モジュール１２７に依頼することによって行う(処理１７３)。次に割込み対象オペレーティングシステムが事務処理ＯＳ１１６であるかリアルタイムＯＳ１１７であるかをチェックし（処理１７４）、対象が事務処理ＯＳ１１６であれば、割込みアドレステーブル１５５を参照し事務処理ＯＳの割込みハンドラ１２２を起動する(処理１７５)。リアルタイムＯＳ１１７への割込みが発生していれば、同様に割込みアドレステーブル１５５を参照しリアルタイムＯＳの割込みハンドラ１２３を起動することになる（処理１７６）。一般に、割込みハンドラ１２２，１２３は、割込み処理でタスク切替えを行わなければならないとき、リスケジューラ１１８，１１９を呼び出し、共通割込みハンドラ１２８に制御を戻すことはない。しかし、タスク切替えが発生しない場合、そのまま割込みハンドラの処理を終了する。このとき、割込みハンドラ１２２，１２３は、共通割込みハンドラへ制御を戻し(図１０で後述)、共通割込みハンドラは処理１７７から動作を再開する。処理１７７は、割込み発生時にオペレーティングシステムを切り替えたかどうかをチェックする処理である。処理１７３によってオペレーティングシステムを切り替えた場合、ここで再び、オペレーティングシステムを切り替え、元の実行環境に戻す必要がある。このため、

もう一度ＯＳコンテクスト切替モジュール１２７に依頼して、切替え処理を実行させる（処理１７８）。

【００３６】

図１０は共通割込みハンドラ１２８から呼び出される事務処理ＯＳの割込みハンドラ１２２の処理フローを示したものである。第一に、使用中レジスタを割込みスタック１５１に退避し（処理１８０）、割込み処理を実行する(処理１８１)。ここで、オペレーティングシステムがリスケジューリング中か否か（すなわち、リスケジューラ１１８の処理実行中か否か）をチェックする（処理１８２）。リスケジューリング中とは、次に実行すべきタスクを選択している最中であり、実際にはタスクを実行していないことを意味する。したがって、オペレーティングシステムの処理を単純化させるならば、この場合、再びリスケジューリングを最初からやり直せば良いことになる。すなわち、リスケジューリング中であると判定した場合、割込みスタックの退避レジスタを破棄し（処理１８７）、リスケジューラ１１８を最初から起動する（処理１８８）。なお、リスケジューリング中に割込みが発生しても、実行すべきタスクを新たに選択し直す必要がない場合もある。本方式は、この状況でもリスケジューリングを最初から行うことになり、効率的でないこともある。このため、リスケジューリング中に発生し、リスケジューラ１１８の実行に影響を与える処理（例えば、タスク起動・終了など）を待ち行列などに登録しておく方法を用いることもできる。この場合、リスケジューラ１１８の終了前で該待ち行列に登録しておいた処理を纏めて実行する。この方式を採用すると、割込み発生のたびに途中まで実行していたリスケジューリングをやり直す必要が無い。しかしながら、当該待ち行列に登録していた処理を纏めて実行した後、これによってリスケジューリングが再び必要となるか否かを再チェックしなければならない。なお、本実施例では、単純化のため、前者の方式で説明を行う。しかし、後者の方式を用いる場合、リスケジューリング中か否かを示すフラグ、ないし処理待ち行列を用意すればよい。オペレーティングシステムが提供するシステムコールなどにおいて、リスケジューラ１１８の実行に影響を与える処理を、リスケジューリング中であれば、該待ち行列に登録する。更に、リスケジューラ１１８の処理フロー中に、該待ち行列に登録された処理を一括して実行するモジュールを挿入する。

【００３７】

割込みハンドラ１２２がリスケジューリング中でないと判定した場合、当該オペレーティングシステムはその時点で何らかのタスクを実行している。このため、第一に、タスク切替えが必要か否かを判定する（処理１８３）。タスク切替えが必要なければ(現実行タスクが最高優先順位であり、かつ、実行可能であれば)、このまま、割込みハンドラ１２２の処理を終了する。このため、割込みスタック１５１上に退避しておいたレジスタ値を復帰し（処理１８４）、共通割込みハンドラへ制御を戻す（処理１８５）。タスク切替えが必要であると判定した場合、割込みスタック１５１上に退避したレジスタの値をタスク管理テーブルにコピーし(処理１８６）、割込みスタック上の退避レジスタを破棄する(処理１８７）。この後、リスケジューラ１１８を起動する（処理１８８）。なお、プロセッサ１００に、メモリ１０１上のデータの参照を行うと共に割込みスタックポインタ１５０の値を増減できる機能が備わっていれば、処理１８６と処理１８７を纏めて実行することができる。

【００３８】

以上、ＯＳ間制御機能１２４に備えられた優先割込みテーブル１５６を参照し、動作させるオペレーティングシステムを決定した後、割込みハンドラを起動することで、入出力装置を複数のオペレーティングシステムで共用する方法を説明したが、入出力装置によっては割込み番号から一意に動作させるオペレーティングシステムを決定できない場合も発生する。例えば図１１に示す車載ナビゲーションシステムの入出力パネルには、ユーザが指示を入力するためのスイッチ105と、結果を表示するディスプレイ１０４とを備える。スイッチ１０５は、それぞれのリアルタイムＯＳ１１７に備えられたタスクのみが使用するリアルタイム
ＯＳ専用スイッチ１９０，事務処理専用スイッチ１９１，双方のオペレーティングシステ

ムに備えられたタスクから共用可能な共用スイッチ１９２を備えることが望ましい。このように、オペレーティングシステム毎に専用スイッチを設けることで、頻繁に使用する機能を固定のスイッチに割り当てることが可能になり、操作性が向上するという特徴が得られる。さらに固定のスイッチに機能を割り当てることで、ボタンに機能名称を表示可能になり識別性が向上する。しかし、これらスイッチ毎に異なる割込み番号を付与することは、割込み信号線数の制約やハードウェアの簡略化の観点より不利である。一方、スイッチに同一の割込み番号を付与すると、スイッチが押下されたときいずれのオペレーティングシステムが備える割込みハンドラを起動すべきか判定できない。この問題を解決するため、ＯＳ間制御機能１２４に備えられた共有メモリ１２５に、スイッチテーブル２００を設ける。スイッチテーブルには、各スイッチ毎にいずれのオペレーティングシステムで使用するスイッチかの情報を記憶する。図１２では、スイッチテーブルには「リアルタイムＯＳ」，「事務処理ＯＳ」，「共用」が記憶される。勿論、テーブルに文字列を記憶することは非常に非効率的であるため、例えばリアルタイムＯＳであれば０，事務処理ＯＳであれば１，共用であれば２といった正数形式で情報を記憶するとよい。

【００３９】

次に、上記スイッチテーブル２００を用い複数のオペレーティングシステムで入出力装置を共有する共通割込みハンドラ１２８の処理フローを図１３を用い説明する。なお、ここで説明する処理はスイッチのみならず、割込みの内容を判断することで起動するオペレーティングシステムを判定することが必要な他の入出力装置に適用することも可能である。

【００４０】

共通割込みハンドラ１２８は、割込み発生後、実行ＯＳ記憶変数１５４の内容を参照し、現在実行中のオペレーティングシステムが事務処理ＯＳ１１６であるかリアルタイムＯＳ１１７であるかをチェックする（処理２１０）。次に、優先割込みテーブル１５６を参照し、共有フラグが立っていない場合は、優先割込みテーブル１５６を参照し、発生した割込みがいずれのオペレーティングシステムに対応しているかを入手する（処理２１２）。共有フラグが立っている場合は、入出力装置にアクセスし割込み内容を判定することで割込み対象オペレーティングシステムを判定する（処理２１３）。ここでは、割込み内容とスイッチテーブル２００を比較し、リアルタイムＯＳ専用スイッチが押されたと判定した場合はリアルタイムＯＳ１１７を、事務処理ＯＳ専用スイッチが押されたと判定した場合は事務処理ＯＳ１１６を割込み対象ＯＳとしてそれぞれ記憶する。また共用スイッチが押されたと判定した場合は、優先割込みテーブル１５６を参照し、発生した割込みがいずれのオペレーティングシステムに対応しているかを入手する。ここで図７の優先割込みテーブル１５６と、図１２のスイッチテーブル２００に記憶された値を用い処理した場合を想定すると、スイッチ割込みにＩＲＱ＃２が割り当てられていれば、スイッチ＃０が入力された場合はリアルタイムＯＳ、スイッチ＃２が入力された場合は事務処理ＯＳ、スイッチ＃３が入力された場合は共用なので再度優先割込みテーブル１５６を参照しリアルタイムＯＳというように対応する割込み対象オペレーティングシステムを入手する。

【００４１】

次に、割込み対象オペレーティングシステムが実行中オペレーティングシステムに等しいか否かをチェックする（処理２１４）。発生した割込みが現在実行中のオペレーティングシステムに対するものでなければ、一旦、オペレーティングシステムの切替えを行わなければならない。この切替え処理は、ＯＳコンテクスト切替モジュール１２７に依頼することによって行う（処理２１５）。次に割込み対象オペレーティングシステムが事務処理ＯＳ１１６であるかリアルタイムＯＳ１１７であるかをチェックし（処理２１６）、対象が事務処理ＯＳ１１６であれば、割込みアドレステーブル１５５を参照し事務処理ＯＳの割込みハンドラ１２２を起動する（処理２１７）。リアルタイムＯＳ１１７への割込みが発生していれば、同様に割込みアドレステーブル１５５を参照しリアルタイムＯＳの割込みハンドラ１２３を起動することになる（処理２１８）。一般に、割込みハンドラ１２２，１２３は、割込み処理によってタスク切替えを行わなければならないとき、リスケジューラ１１８，１１９を呼び出し、

共通割込みハンドラ１２８に制御を戻すことはない。しかし、タスク切替えが発生しない場合、そのまま割込みハンドラの処理を終了する。このとき、割込みハンドラ１２２，１２３は、共通割込みハンドラ１２８へ制御を戻し、共通割込みハンドラ１２８は処理219から動作を再開する。処理２１９は、割込み発生時にオペレーティングシステムを切り替えたかどうかをチェックする処理である。処理２１５によってオペレーティングシステムを切り替えた場合、ここで再び、オペレーティングシステムを切り替え、元の実行環境に戻す必要がある。このため、もう一度ＯＳコンテクスト切替モジュール１２７に依頼して、切替え処理を実行させる（処理２２０）。以上、ＯＳ間制御機能１２４に備えられた共通割込みハンドラから割込み管理テーブル１２９に記憶された優先割込みテーブル１５６を参照し、複数のオペレーティングシステムを１つの計算機で動作させるシステムにおいて入出力装置を共有する方法を説明したが、前記説明した方法で全てのケースに対応することはできない。例えば、グラフィックスアクセラレータやシリアルコミュニケーションといった入出力装置では、入出力ドライバから動作条件や各種動作パラメータを設定し、ハードウェア内部のレジスタで前記パラメータや動作状態を保持する。このような入出力装置に対し単純に前記処理を適用すると、各々のオペレーティングシステム内の入出力ドライバ、及び割込みハンドラは、オペレーティングシステムが切り替わったことを認識することができないため、以前のオペレーティングシステムの入出力ドライバが設定した状態から処理を継続しようとする。しかし、設定しようとするパラメータは、自オペレーティングシステムの入出力ドライバが最後に設定した値から継続したときのみ正常に動作し、オペレーティングシステムが切り替えられる前に何らかのパラメータが入出力装置に書き込まれていれば希望に沿った動作をしないのは明確である。

【００４２】

これら問題を解決するため、本発明では図１４に示すように各々のオペレーティングシステムに備えられた入出力ドライバ１２０，１２１、及び割込みハンドラ１２２，１２３といった入出力処理にクライアントーサーバモデルを適用したことを特徴とする。即ち、複数のオペレーティングシステムの入出力処理の一つでサーバ側入出処理２２２を実行し、その他のオペレーティングシステムの入出力処理でクライアント側入出力処理２２１を実行する。サーバ側入出力処理222は、複数のオペレーティングシステムで共有する入出力装置２２６を管理するように動作し、入出力ドライバ１２１から入出力装置２２６へのデータ入出力225を実行したり、入出力装置２２６からの割込み要求を割込みハンドラ１２３で受けるように動作する。さらに、クライアント側入出力処理２２１からの依頼通知２２３を受け、その内容を判断し入出力装置２２６とデータ入出力２２５し、その結果を結果通知２２４としてクライアント側入出力装置２２１に返信するように動作する。一方、クライアント側入出力処理２２１は、入出力装置２２６へのデータ入出力をサーバ側入出力装置２２２に依頼通知２２３を送信し、返信された結果通知２２４を用い入出力処理を実行する。従って、クライアント側入出力処理２２１から入出力装置２２６のハードウェアを直接制御することはなく、常にサーバ側入出力装置が入出力装置を管理するため、動作状態に矛盾は生じない。

【００４３】

なお、サーバ側入出力処理２２２は優先順位の高いオペレーティングシステムに設定することが望ましい。本実施例では事務処理ＯＳよりリアルタイムＯＳの優先順位が高いと仮定しているため、サーバ側入出力処理２２２をリアルタイムＯＳに備える。優先順位の高いオペレーティングシステムにサーバ側入出力処理２２２を備えることで、割込み要求といった処理がマスクされることを防止できる。従って、割込み要求発生での応答時間が不当に遅くなることを防ぎ、データの取りこぼしを無くすことができるという特徴が得られる。

【００４４】

以降、図１４で示したモジュールのうち、共通割込みハンドラ１２８，割込みハンドラ１２２，１２３の処理フローの説明を行う。

【００４５】

図１４に示すクライアントーサーバモデルを入出力処理に適用したとき、共通割込みハンドラ１２８の動作フローを図１５を用い説明する。なお、基本的なシステム構成は図１、及び図５と同様であり、さらに優先割込みテーブル１５６では、複数のオペレーティングシステムで共有する入出力装置の割込み番号には共有フラグ１が、特定のオペレーティングシステムが使用する入出力装置の割込み番号には０がそれぞれ記憶されているものとする。さらに、図１４に示したように、サーバ側入出力処理２２２をリアルタイムＯＳ１１７に、クライアント側入出力処理を事務処理ＯＳにそれぞれ備えると仮定する。

【００４６】

共通割込みハンドラ１２８は、割込み発生後、第一に、実行ＯＳ記憶変数154の内容を取り出して、現在実行中のオペレーティングシステムが事務処理OS116であるかリアルタイムＯＳ１１７であるかをチェックする（処理２３０）。次に、優先割込みテーブル１５６を参照し（処理２３１）、共有フラグが立っていない場合は優先割込みテーブル１５６を参照し、発生した割込みがいずれのオペレーティングシステムに対応しているかを入手する（処理２３２）。共有フラグ１が立っている場合は、リアルタイムＯＳを割込み対象オペレーティングシステムと判定する（処理２３３）。

【００４７】

次に、割込み対象オペレーティングシステムが実行中オペレーティングシステムに等しいか否かをチェックする（処理２３４）。発生した割込みが現在実行中のオペレーティングシステムに対するものでなければ、一旦、オペレーティングシステムの切替えを行わなければならない。この切替え処理は、ＯＳコンテクスト切替モジュール１２７に依頼することによって行う（処理２３５）。次に割込み対象オペレーティングシステムが事務処理ＯＳ１１６であるかリアルタイム
ＯＳ１１７であるかをチェックし（処理２３６）、対象が事務処理ＯＳ１１６であれば、割込みアドレステーブル１５５を参照し事務処理ＯＳの割込みハンドラ１２２を起動する（処理２３７）。リアルタイムＯＳ１１７への割込みが発生していれば、同様に割込みアドレステーブル１５５を参照しリアルタイムＯＳの割込みハンドラ１２３を起動することになる（処理２３８）。一般に、割込みハンドラ１２２，１２３は、割込み処理によってタスク切替えを行わなければならないとき、リスケジューラ１１８，１１９を呼び出し、共通割込みハンドラ１２８に制御を戻すことはない。しかし、タスク切替えが発生しない場合、そのまま割込みハンドラの処理を終了する。このとき、割込みハンドラ１２２，１２３は、共通割込みハンドラへ制御を戻し、共通割込みハンドラは処理２３９から動作を再開する。処理２３９は、割込み発生時にオペレーティングシステムを切り替えたかどうかをチェックする処理である。処理２３５によってオペレーティングシステムを切り替えた場合、ここで再び、オペレーティングシステムを切り替え、元の実行環境に戻す必要がある。このため、もう一度ＯＳコンテクスト切替モジュール１２７に依頼して、切替え処理を実行させる（処理２４０）。

【００４８】

図１６は共通割込みハンドラ１２８から呼び出されるサーバ側入出力装置222の一部であるリアルタイムＯＳの割込みハンドラ１２３の処理フローを示したものである。

【００４９】

第一に、使用中レジスタを割込みスタック１５３に退避し（処理２５０）、優先先割込みテーブル１５６を参照することで割込み要求した入出力装置が複数のオペレーティングシステムから共有されているか判定する（処理２５１）。共有していない場合は、リアルタイムＯＳに対する割込み要求であり、対応する割込み処理を実行する（処理２５３）。共有している場合は、再度優先先割込みテーブル１５６を参照し、発生した割込み要求がいずれのオペレーティングシステムに対応しているかを判定する（処理２５２）。リアルタイムＯＳ側のフラグが立っている場合、ないし共有フラグが立っていない場合は、リアルタイムＯＳ割込みに対する割込み処理を実行する（処理２５３）。事務処理ＯＳ割込み側のフラグが立っている場合は、事務処理ＯＳ割込みに対する割込み処理を実施する（処理２５４）。ここで、事務処理ＯＳ割込みにおける割込み処理２５４は、割込みが発生した

ことをクライアント側の割込みハンドラ１２２に通知するため、事務処理ＯＳ１１６へのソフトウェア割込みを生成する。さらに必要があれば前記ソフトウェア割込み処理で入手した情報を、共有メモリ１２５に書き込む。なお生成するソフトウェア割込みは、該実行中の割込み処理より優先順位を低くする必要がある。事務処理ＯＳ割込みに対する割込み処理実行では、何らリアルタイムＯＳに影響を与えないためタスク切替判定２５５は必要としないため、このまま割込みハンドラ１２３の処理を終了する。このため、割込みスタック１５３上に退避しておいたレジスタ値を復帰し（処理２５９）、共通割込みハンドラへ制御を戻す（処理２６０）。なお、処理２５５を省くのは処理高速化のためであり、処理２５５を実行するように動作しても何ら問題はない。

【００５０】

一方リアルタイムＯＳ割込みに対応する割込み処理実行後（処理２５３）、タスク切替えが必要か否かを判定する（処理２５５）。タスク切替えが必要なければ、割込みハンドラ１２３の処理を終了する。このため、割込みスタック１５１上に退避しておいたレジスタ値を復帰し（処理２５９）、共通割込みハンドラへ制御を戻す（処理２６０）。タスク切替えが必要であると判定した場合、割込みスタック１５１上に退避したレジスタの値をタスク管理テーブルにコピーし（処理２５６）、割込みスタック上の退避レジスタを破棄する（処理２５７）。この後、リスケジューラ１１９を起動する（処理２５８）。なお、プロセッサ１００に、メモリ１０１上データ参照と同時に割込みスタックポインタ１５２の値を増減できる機能が備わっていれば、処理２５６と処理２５７を纏めて実行することができる。

【００５１】

一方、割込みハンドラ１２３で生成したソフトウェア割込みにより起動される事務処理ＯＳの割込みハンドラ１２２は、その大部分を図１０で説明した動作フローと同内容で処理を実行する。なお、一部動作が異なるのが処理１８１に示した割込みに対応する割込み処理の実行である。本動作例では、図１６の処理254において事務処理ＯＳ割込みに対する割込み処理を実行し、ここで入手した情報は、共有メモリ１２５に書き込まれているものとする。従って、割込みハンドラ１２２内の処理１８１は共有メモリ１２５に記録された前記情報を読み出し、割込みに対応する処理を実行するように動作することで、直接入出力装置をアクセスすることなく割込み処理が実現される。

【００５２】

また、図１４で説明したようにクライアントーサーバモデルを適用し複数のオペレーティングシステムで入出力装置を共用する場合においても、図１１～図
１３の説明同様、割込みの内容を診断しないといずれのオペレーティングシステムの割込み処理を実行してよいか判断できないケースが発生する。本ケースでのリアルタイムＯＳ１１７に備えられた割込みハンドラ１２３の処理フローを図
１７を用い説明する。なお、割込み内容を判定するためのテーブルは前記説明同様にスイッチテーブル２００を用いた場合を想定するが、ここで説明する処理はスイッチのみならず、他の入出力装置に適用することも可能である。

【００５３】

第一に、使用中レジスタを割込みスタック１５３に退避し（処理２７０）、優先先割込みテーブル１５６を参照することで割込み要求した入出力装置が複数のオペレーティングシステムから共有されているか判定する（処理２７１）。共有していない場合は、リアルタイムＯＳに対する割込み要求であり、対応する割込み処理を実行する（処理２７４）。共有フラグが立っている場合は、割込み内容を判定することで割込み対象オペレーティングシステムを判定する(処理２７２)。割込み内容とスイッチテーブル２００を比較し、リアルタイムＯＳ専用スイッチが押されたと判定した場合はリアルタイムＯＳ１１７を、事務処理ＯＳ専用スイッチが押されたと判定した場合は事務処理ＯＳ１１６を割込み対象ＯＳとしてそれぞれ記憶する。また共用スイッチが押されたと判定した場合は、優先割込みテーブル１５６を参照し、フラグが立っているＯＳを割込み対象ＯＳとして記憶する。図７に示す優先割込みテーブル１５６と、図１２のスイッチテーブル200を用い、かつスイッ

チ割込みにＩＲＱ＃２が割り当てられている場合を想定すると、スイッチ＃０が入力された場合はリアルタイムＯＳ，スイッチ＃２が入力された場合は事務処理ＯＳ，スイッチ＃３が入力された場合は共用なのでさらに優先割込みテーブル１５６を参照しリアルタイムＯＳというように対応する割込み対象オペレーティングシステムを入手することができる。

【００５４】

次に割込み対象ＯＳを参照し（処理２７３）、リアルタイムＯＳ１１７であればリアルタイムＯＳ割込みに対する割込み処理を実行する（処理２７４）。事務処理ＯＳ１１６であれば事務処理ＯＳ割込みに対する割込み処理２７４をそれぞれ実行する（処理２７５）。ここで、なお、事務処理ＯＳ割込みに対する割込み処理２７５は、割込みが発生したことをクライアント側の割込みハンドラ１２２に知らせるため、事務処理ＯＳ１１６へのソフトウェア割込みを生成する。さらに必要があれば前記ソフトウェア割込み処理で入手した情報を、共有メモリ125に書き込む。なお生成するソフトウェア割込みは、該実行中の割込み処理より優先順位を低くする必要がある。

【００５５】

事務処理ＯＳ割込みに対応する割込み処理実行では、何らリアルタイムＯＳに影響を与えないためタスク切替判定２７６は必要としない。従って、このまま割込みハンドラ１２３の処理を終了する。そこで、割込みスタック１５３上に退避しておいたレジスタ値を復帰し（処理２８０）、共通割込みハンドラへ制御を戻す（処理２８１）。なお、処理２７６を省くのは処理高速化のためであり、処理２７６を実行するように動作しても何ら問題はない。

【００５６】

一方リアルタイムＯＳ割込みに対応する割込み処理実行後、タスク切替えが必要か否かを判定する（処理２７６）。タスク切替えが必要なければ、割込みハンドラ１２３の処理を終了する。このため、割込みスタック１５３上に退避しておいたレジスタ値を復帰し（処理２８０）、共通割込みハンドラへ制御を戻す（処理２８１）。タスク切替えが必要であると判定した場合、割込みスタック１５１上に退避したレジスタの値をタスク管理テーブルにコピーし（処理２７７）、割込みスタック上の退避レジスタを破棄する（処理２７８）。この後、リスケジューラ１１９を起動する（処理２７９）。

【００５７】

以上、複数のオペレーティングシステムから同一の入出力装置を共有するための方法としてクライアントーサーバモデルを適用する方法について説明した。この中でサーバ側のオペレーティングシステムから他方のオペレーティングシステムへ割込み発生をソフトウェア割込みで知らせる方法につき説明したが、何らかの通信手段を備えていれば割込み発生をサーバ側入出力処理２２２からクライアント側入出力処理２２１へ通知することが可能であり、前記説明したソフトウェア割込みと置き換えることができる。本実施例では、２つのオペレーティングシステムの間で通信するため、図１に示すようにＯＳ間通信機能１２６を備える。そこでＯＳ間通信機能１２６を用い、複数のオペレーティングシステムから同一の入出力装置を共有するための方法を図１８を用い説明する。

【００５８】

図１８における構成要素は、その大部分が図１および図５で説明した構成要素と同一であるが、事務処理ＯＳ１１６に備えられていた入出力ドライバ１２０，割込みハンドラ１２２に代えて仮想入出力ドライバ２９０，仮想割込みハンドラ２９１を備えたところが異なる。仮想入出力ドライバ２９０は、アプリケーションタスクから入出力装置をアクセスするためのインタフェースを提供するが、直接入出力装置に読み書きしたり、直接入出力装置を制御することはしない。ここでは、ＯＳ間通信機能１２６が提供するメッセージ通信を用い、入出力ドライバ１２１に入出力装置への読み書きや制御を依頼するとともに、その結果もメッセージ通信を用い入出力ドライバ１２１から受け取るように動作する。仮想割込みハンドラ２９１は、割込みハンドラ１２３から送られるメッセージで起動され、割込み処理を実行する。

【００５９】
次に、双方のオペレーティングシステムが共有する入出力装置割込み要求が発生したときの処理の流れを図１８を用い説明する。説明の都合上、ここで扱う入出力装置が生成する割込み番号の優先割込みテーブルには、共有フラグ１がセットされ、さらに事務処理ＯＳ側にフラグが立っているものとする。
【００６０】
入出力装置で発生した割込み要求は、一旦共通割込みハンドラ１２８にトラップされ、図１５のフローに従って処理を実行する。ここでは両方のオペレーティングシステムが共有する入出力装置からの割込みであると仮定しているため、リアルタイムＯＳに備えられた割込みハンドラ１２３を起動する。割込みハンドラ１２３は、図１６で説明したフローに従って処理を実行する。ここでは、優先割込みテーブル１５６に共有フラグ、及び事務処理ＯＳにフラグが立っていると仮定しているため、事務処理ＯＳ割込みに対応する割込み処理２５４が実行される。ここで、事務処理ＯＳ割込みに対する割込み処理２５４は、割込みが発生したことをクライアント側の仮想割込みハンドラ２９１に知らせるため、ＯＳ間通信機能１２６を用い仮想割込みハンドラ２９１へメッセージを送る。なお、割込み処理で入手した情報は、メッセージに添付して送信してもよいし、共有メモリ１２５に書き込み仮想割込みハンドラ２９１から参照することで情報転送してもよい。さらに、仮想割込みハンドラ２９１にメッセージを受信するためのメッセージキューを設けられない時は、アプリケーションタスクとして備えられたサーバタスク１１２に一旦メッセージを送り、該サーバタスク１１２から仮想割込みハンドラ２９１を呼び出すことで仮想割込みハンドラ２９１を起動してもよい。仮想割込みハンドラ２９１は、メッセージに添付された割込み処理の処理結果、ないし共有メモリ１２５を参照することで割込み処理を実行される。
【００６１】
さらに、事務処理ＯＳが備えるタスクから依頼された入出力処理を実行する処理の流れを図１９を用い説明する。事務処理ＯＳ内の仮想入出力ドライバ２９０は、事務処理ＯＳから入出力装置をアクセスするための唯一のインタフェースであり、依頼された入出力処理はＯＳ間通信機能１２６を用い入出力ドライバ121へ転送される。なお、仮想入出力ドライバ２９０で入手した情報は、メッセージに添付して送信してもよいし、共有メモリ１２５に書き込み入出力ドライバ121から参照することで情報転送してもよい。さらに、入出力ドライバ１２１にメッセージを受信するためのメッセージキューを設けられない時は、アプリケーションタスクとして備えられたサーバタスク１１３に一旦メッセージを送り、サーバタスク１１３から入出力ドライバ１２１を呼び出すことで入出力ドライバ１２１を起動してもよい。入出力ドライバ１２１は、メッセージに添付された依頼内容、ないし共有メモリ１２５を参照することで入出力処理を実行する。
【００６２】
次に、ＯＳ間制御機能１２４に備えられた優先割込みテーブル１５６の書き換えフローを図２０を用い説明する。優先割込みテーブル１５６は、ユーザ要求を管理する管理タスク１１５等によって書き換える。しかし、優先割込みテーブル１５６を書き換えている最中に、割込み処理が優先割込みテーブル１５６を参照すると不整合が発生するおそれがある。そこで最初に全割込み要求をマスクし、割込みが発生しないようにする（処理３０１）。なお書き換えを実施する割込み番号のみ割込み要求をマスクしてもよい。割込みマスクの設定後、優先割込みテーブル１５６の内容を更新し（処理３０２）、設定した割込みマスクを解除する（処理３０３）ことで優先割込みテーブル１５６の書き換えが実施できる。
【００６３】
以上、複数のオペレーティングシステムから入出力装置を共有する方法を説明したが、実際の応用例として車載ナビゲーション装置といったユーザインタフェースを備える組み込み装置のグラフィックス表示に本発明を適用した例を図２１を用い説明する。なお、グラフィックス表示システム上で動作するオペレーティングシステムは、図１同様に事務処理

ＯＳ１１６、及びリアルタイムＯＳ１１７で、リアルタイムＯＳ上で動作するタスク／スレッドは事務処理ＯＳ上で動作するタスク／スレッドより優先順位が高いと仮定する。

【００６４】

グラフィックス表示装置は、描画や表示といったコマンドを解釈しグラフィックスハードウェア３１４を制御するグラフィックスドライバ３１０，３１２，グラフィックスハードウェア３１４が生成する割込み要求で起動するグラフィックス割込みハンドラ３１１，３１３、及び画素に展開されたイメージデータをディスプレイに表示するグラフィックスハードウェア３１４で構成する。グラフィックスハードウェア３１４は、各画素の輝度値を保持するフレームメモリ３１６、及び垂直同期周波数毎にフレームメモリ３１６から各画素の輝度値を取り出し、ディスプレイ１０４に出力する表示制御３１５とを備える。ここで、フレームメモリ３１６では、事務処理ＯＳのグラフィックスドライバ３１０が作成したイメージを蓄える事務処理ＯＳ用描画領域３１７と、リアルタイムＯＳのグラフィックスドライバ３１２が作成したイメージを蓄えるリアルタイムＯＳ用描画領域３１８とを独立・分離することで、リソースの衝突を防ぐ。これにより、オペレーティングシステムを切り替えた場合でも、各々の描画イメージは保持されるため、再度オペレーティングシステムが切り替わったときでも、切り替わり前の状態から処理を続行できるという利点を有する。また、リアルタイムＯＳに備えられたグラフィックスドライバ３１２、及びグラフィックス割込みハンドラ３１３はサーバとして動作し、事務処理ＯＳに備えられたグラフィックスドライバ310 、及びグラフィックス割込みハンドラ３１１はクライアントとして動作すると仮定する。

【００６５】

リアルタイムＯＳ１１７に組み込まれたアプリケーションタスクからの描画，表示要求における動作フローを最初に説明する。アプリケーションタスクは、グラフィックスドライバ３１２に備えられた描画コマンドを呼び出し、描画処理を実行する。描画コマンドは、線や多角形の描画するためのコマンドや線幅や塗りつぶしパターンといった属性を指定するコマンドで構成され、グラフィックスドライバ３１２は描画コマンドを解釈し、リアルタイムＯＳ用描画領域３１８に画素展開する。グラフィックスハードウェア３１４が、画素展開をハードウェアで実行するアクセラレーション機能を有していれば、それを活用してもよい。描画が終了すると、次にグラフィックスドライバ３１２に備えられた表示コマンドを実行する。表示コマンドは、フレームメモリ３１６における表示開始アドレスや画面サイズを指定するコマンドで構成され、表示制御３１５内のレジスタを制御することで指定された画像を表示する。指定された画面の表示が開始されると、表示制御３１５は割込み要求を生成する。割込み要求は、一旦共通割込みハンドラ１２８にトラップされるが、直ちに共通デバイスからの割込み要求と判定し、グラフィックス割込みハンドラ３１３を呼び出す。これによりグラフィックス割込みハンドラ３１３は、画面が表示されたことを識別する。

【００６６】

次に、事務処理ＯＳ１１６に組み込まれたアプリケーションタスクからの描画，表示要求における動作フローを次に説明する。アプリケーションタスクは、グラフィックスドライバ３１０に備えられた描画コマンドを呼び出し、事務処理ＯＳ用描画領域３１７に画素展開する。描画が終了すると、次にグラフィックスドライバ３１０に備えられた表示コマンドを実行する。表示コマンドは、フレームメモリ３１６における表示開始アドレスや画面サイズを指定するメッセージをＯＳ間通信機能１２６を用いグラフィックスドライバ３１２に転送する。その要求を受けたグラフィックスドライバ３１２は、表示制御３１５内のレジスタを制御することで指定された画像を表示する。なお、グラフィックスドライバ３１２は、いずれのオペレーティングシステムからの表示要求が優先であるかを判定し、常に優先的に表示すべきフレームメモリを表示する。指定された画面の表示が開始されると、表示制御３１５は割込み要求を生成する。割込み要求は、一旦共通割込みハンドラ１２８でトラップされるが、直ちに共通デバイスからの割込み要求と判定し、グラフィックス割込みハンドラ３１３を起動する。グラフィックス割込み

ハンドラ３１３は、ＯＳ間通信機能１２６を用い、割込み要求を受けたことをグラフィックス割込みハンドラ３１１に伝える。これによりグラフィックス割込みハンドラ３１１は、画面が表示されたことを識別する。

【００６７】

以上、本発明を適用したグラフィックス表示装置の具体的な実施例を説明したが、本発明を適用することで例えばシリアル／ネットワーク通信やＣＤ－ＲＯＭ／ＤＶＤ－ＲＯＭ記憶装置といった様々な周辺装置を複数のオペレーティングシステムから共有することが可能になる。

【００６８】

【発明の効果】

本発明によれば、複数個のオペレーティングシステムを単一のプロセッサで動作させる計算機システムにおいて、各オペレーティングシステムが実行するアプリケーションタスクから同一の入出力装置を共有することが可能になるため、入出力装置の総数を削減することが可能になる。さらに、本発明によれば、複数個のオペレーティングシステムが共有する入出力装置とデータ入出力するための入出力ドライバ、及び割込みハンドラを該一つのオペレーティングシステムに実装し、それ以外のオペレーティングシステムから該入出力ドライバ及び割込みハンドラに入出力処理を依頼することで、入出力装置をリセットすることなく入出力処理を継続することが可能になり、操作性が向上する。

【図面の簡単な説明】

【図１】本発明の第１の実施例におけるシステム構成を示す図である。

【図２】計算機システムのハードウェア構成を示す図である。

【図３】割込みレベルマスク機能を装備する場合のステータスレジスタを示す図である。

【図４】個別割込みマスク機能を装備する場合のステータスレジスタを示す図である。

【図５】割込み処理プログラムの詳細を示す第１の図である。

【図６】割込みアドレステーブルの構成を示す図である。

【図７】優先割込みテーブルの構成を示す図である。

【図８】ＯＳコンテクスト切換モジュールの処理フローを示す図である。

【図９】共通割込みハンドラの処理フローを示す第１の図である。

【図１０】割込みハンドラの処理フローを示す第１の図である。

【図１１】車載ナビゲーション装置の入出力パネルの構成を示す図である。

【図１２】スイッチテーブルの構成を示す図である。

【図１３】共通割込みハンドラの処理フローを示す第２の図である。

【図１４】本発明の第２の実施例におけるシステム構成を示す図である。

【図１５】共通割込みハンドラの処理フローを示す第３の図である。

【図１６】割込みハンドラの処理フローを示す第２の図である。

【図１７】割込みハンドラの処理フローを示す第３の図である。

【図１８】入出力処理プログラムの詳細を示す第１の図である。

【図１９】入出力処理プログラムの詳細を示す第２の図である。

【図２０】優先割込みテーブル書き換えフローを示す図である。

【図２１】本発明をグラフィックス表示装置に適用したシステムの構成を示す図である。

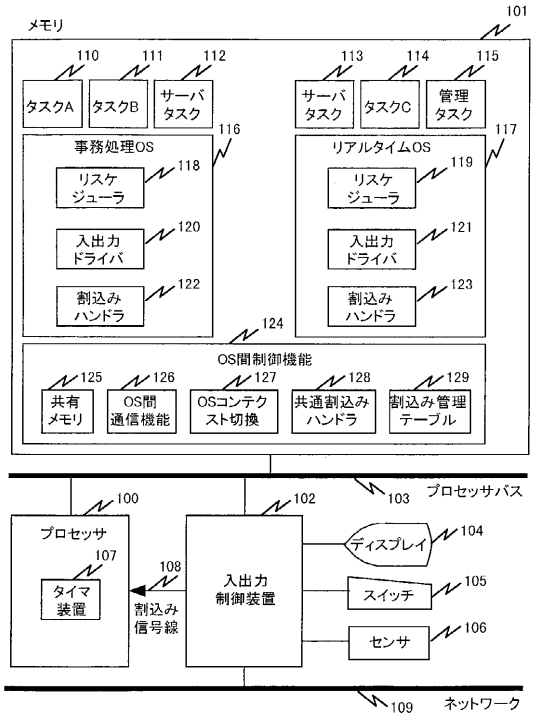
【符号の説明】

１００…プロセッサ、１０１…メモリ、１０２…入出力制御装置、１０３…プロセッサバス、１０４…ディスプレイ、１０５…スイッチ、１０６…センサ、１０７…タイマ装置、１０８…割込み信号線、１０９…ネットワーク、１１０…タスクＡ、１１１…タスクＢ、１１２，１１３…サーバタスク、１１４…タスクＣ、１１５…管理タスク、１１６…事務処理ＯＳ、１１７…リアルタイムＯＳ、１１８，１１９…リスケジューラ、１２０，１２１…入出力ドライバ、１２２，１２３…割込みハンドラ、１２４…ＯＳ間制御機能、１２５…共有メモリ、126…ＯＳ間通信機能、１２７…ＯＳコンテクスト切換、１２８…共通割込みハンドラ、１２９…割込み管理テーブル、１３０…キャッシュメモリ、１３１…ＣＰＵ、１３２…汎用レジスタ、１３３…ＰＣ、１３４…ＳＲ、１３５…データバス、１３

６…アドレスバス、１３７…割込みコントローラ、１４０…割込みブロックビット、１４１…割込みマスクレベル、１４２…実行状態レジスタ、１４３…割込みマスクレジスタ、１４４～１４７…割込みマスクビット、１５０，１５２…割込みスタックポインタ、１５１，１５３…割込みスタック、１５４…実行ＯＳ記憶、１５５…割込みアドレステーブル、１５６…優先割込みテーブル、１５７…事務処理ＯＳ用保存コンテクスト、１５８…リアルタイムＯＳ用保存コンテクスト、１９０…リアルタイムＯＳ専用スイッチ、１９１…事務処理ＯＳ専用スイッチ、１９２…共用スイッチ、２００…スイッチテーブル、２２１…クライアント側入出力処理、２２２…サーバ側入出力処理、２２６…入出力装置、２９０…仮想入出力ドライバ、２９１…仮想割込みハンドラ、３１０，３１２…グラフィックスドライバ、３１１，３１３…グラフィックス割込みハンドラ、３１４…グラフィックスハードウェア、３１５…表示制御、３１６…フレームメモリ、317…事務処理ＯＳ用描画領域、３１８…リアルタイムＯＳ用描画領域。

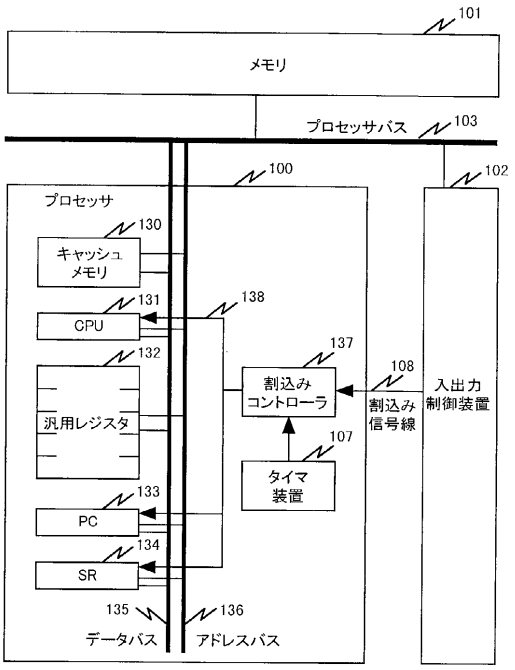
【図１】

図 1



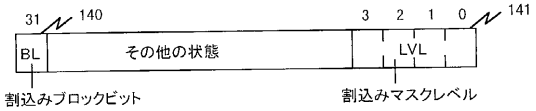
【図２】

図 2



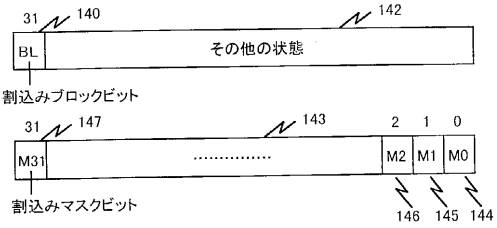
【図３】

図 3



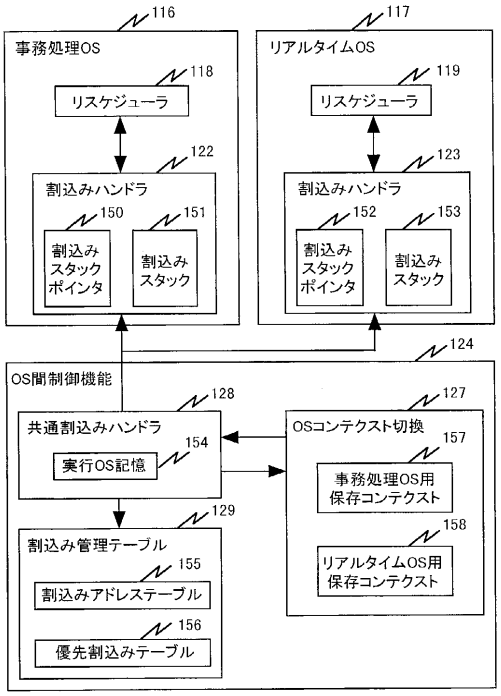
【図４】

図 4



【図５】

図 5



【図６】

図 6

割込みアドレステーブル 155

割込み番号	割込み開始アドレス	
	事務処理OS	リアルタイムOS
IRQ 0	0x00000400	0x00000600
IRQ 1	0x00000420	0x00000620
IRQ 2	0x00000440	0x00000640
IRQ 3	0x00000480	0x00000680
IRQ 4	0x000004A0	0x000006A0
:	:	:
IRQ 15	0x000005C0	0x000007C0

【図７】

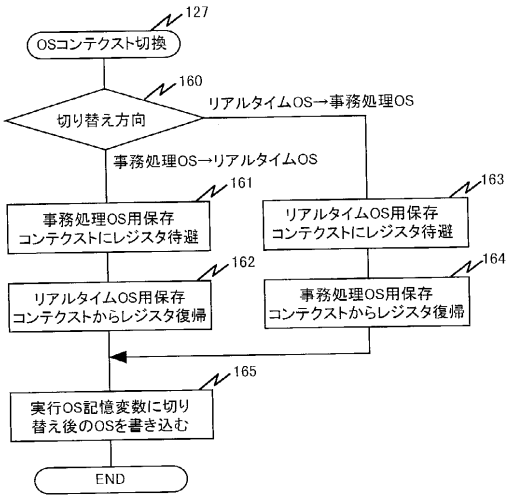
図 7

優先割込みテーブル 156

割込み番号	事務処理OS	リアルタイムOS	共有
IRQ 0	0	1	0
IRQ 1	1	0	0
IRQ 2	0	1	1
IRQ 3	1	0	1
IRQ 4	0	1	0
:	:	:	:
IRQ 15	0	1	0

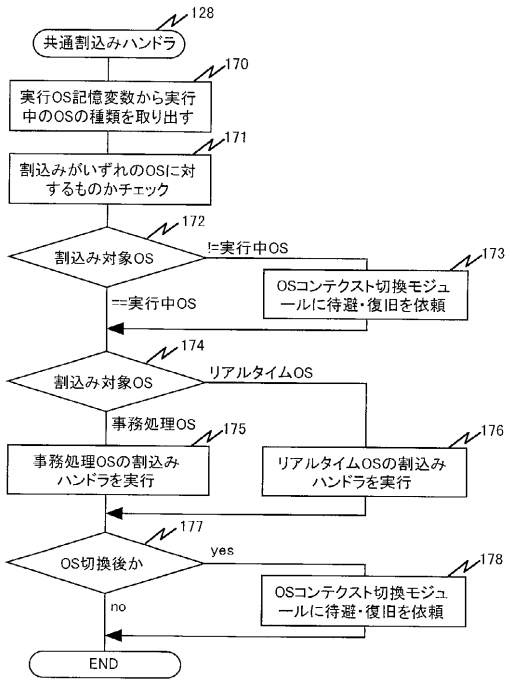
【図８】

図 8



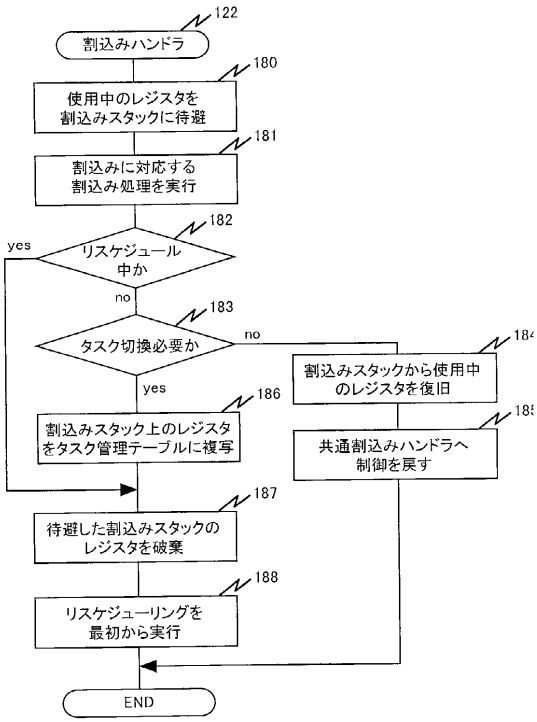
【図９】

図 9



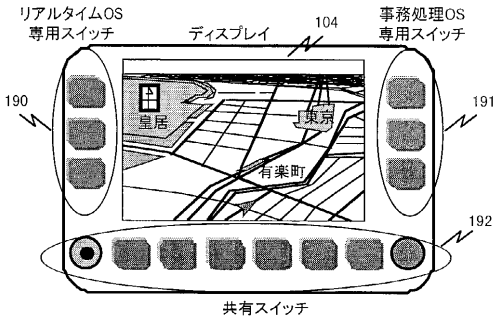
【図１０】

図 10



【図１１】

図 11



【図１２】

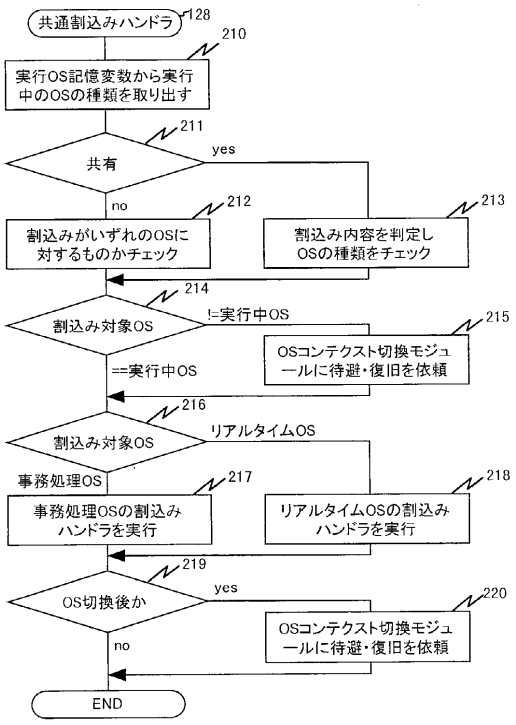
図 12

スイッチテーブル 200

スイッチ番号	属性
# 0	リアルタイムOS
# 1	リアルタイムOS
# 2	事務処理OS
# 3	共有
# 4	共有
:	:
# n	共有

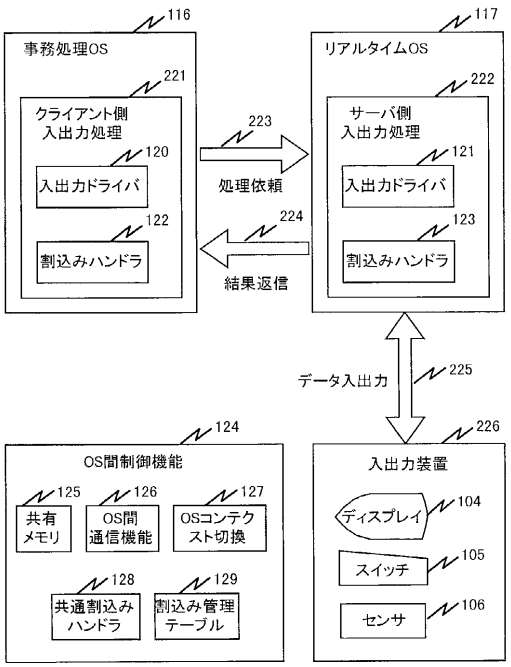
【図１３】

図 13



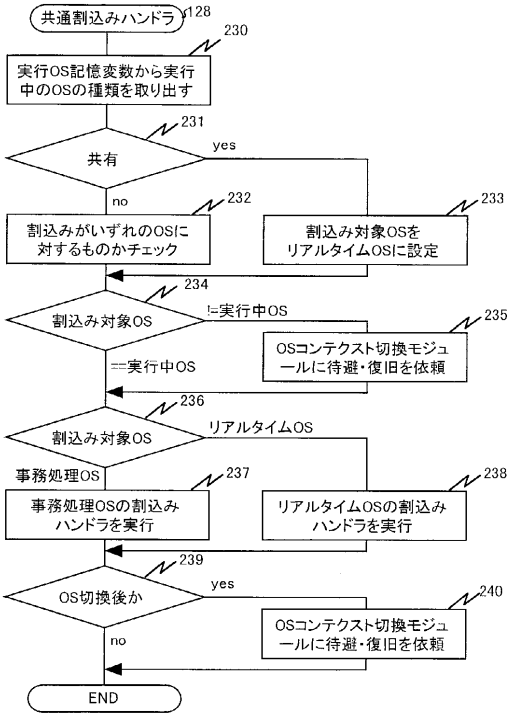
【図１４】

図 14



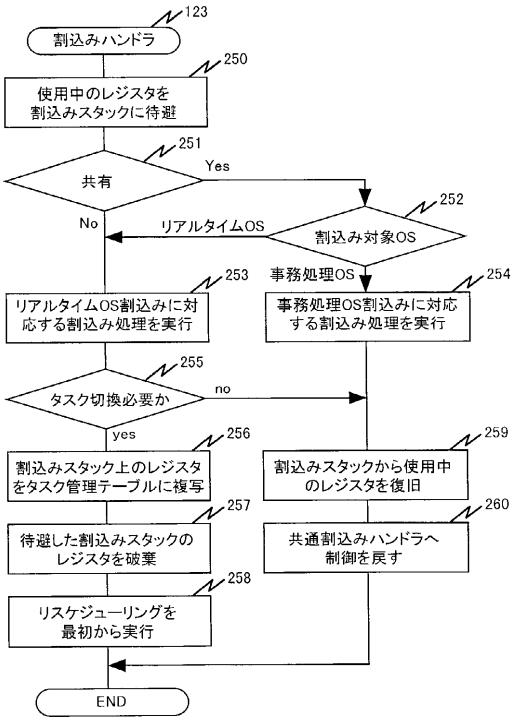
【図１５】

図 15



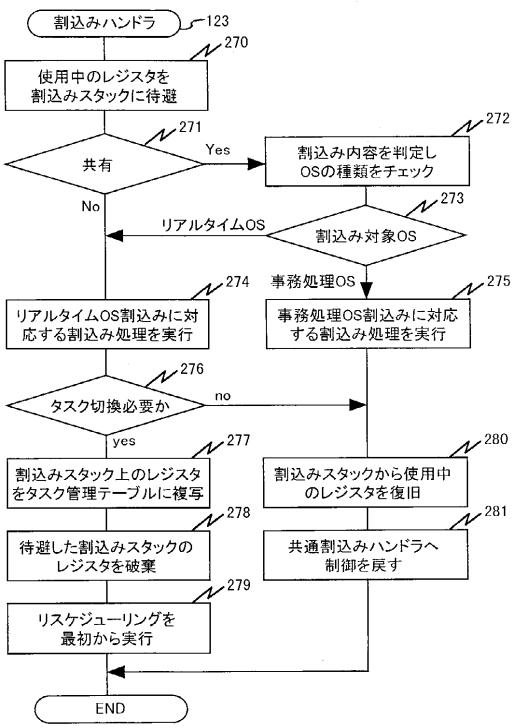
【図１６】

図 16



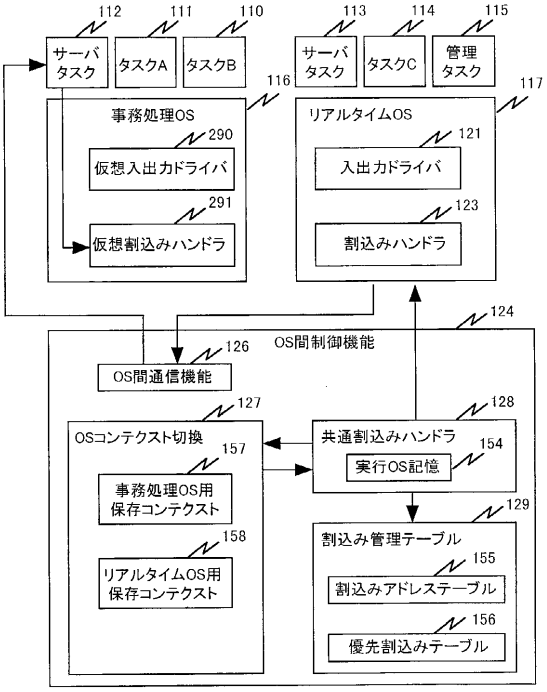
【図１７】

図 17



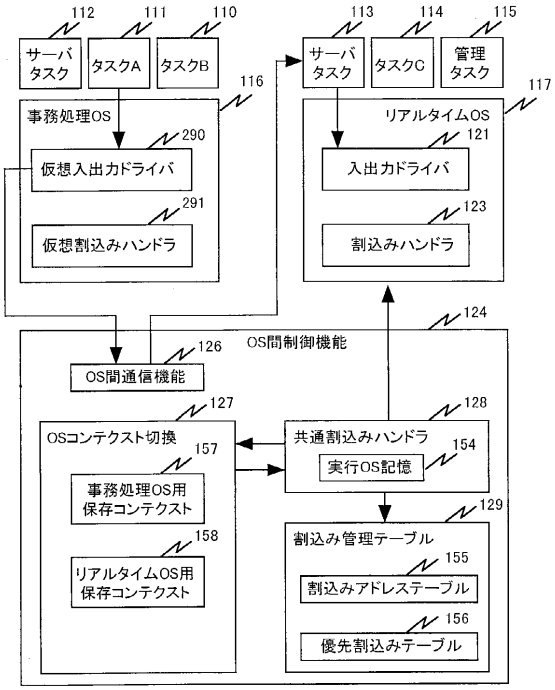
【図１８】

図 18



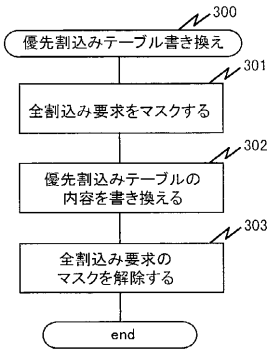
【図１９】

図 19



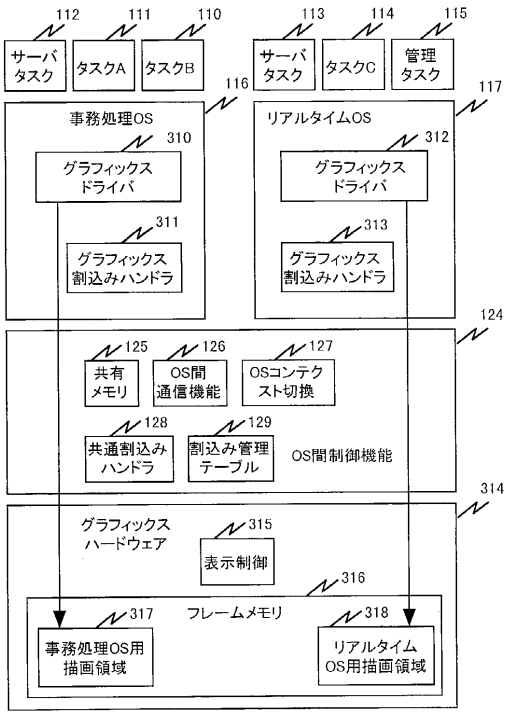
【図２０】

図 20



【図２１】

図 21



フロントページの続き

(72)発明者 上脇 正
茨城県日立市大みか町七丁目１番１号 株式会社 日立製作所 日立研究所内
(72)発明者 齋藤 雅彦
茨城県日立市大みか町七丁目１番１号 株式会社 日立製作所 日立研究所内
(72)発明者 大野 洋
東京都千代田区神田駿河台四丁目６番地 株式会社 日立製作所 新事業推進本部内
(72)発明者 中村 智明
東京都千代田区神田駿河台四丁目６番地 株式会社 日立製作所 新事業推進本部内

審査官 殿川 雅也

(56)参考文献 特表平０３－５０５３８３（ＪＰ，Ａ）
特開平０１－０９３８３０（ＪＰ，Ａ）
特開平１１－０８５５４７（ＪＰ，Ａ）
特開平０３－０４８９３７（ＪＰ，Ａ）
国際公開第９９／０１２０９５（WO，Ａ１）
米国特許第４７４７０４０（ＵＳ，Ａ）
KAHNG, R. Y., Dual System Operation, IBM TECHNICAL DISCLOSURE BULLETIN, 米国, IBM, １９７０年 4月, Vol. 12, No. 11, pp. 1899 - 1900
平成１１年度の日立技術の展望, 日立評論, 日本, 日立評論社, １９９９年 1月, 第８１巻、第１号, 第２７頁乃至第２９頁
三宅常之, Ｗｉｎｄｏｗｓ ＣＥとμＩＴＲＯＮ, 複合技術を日立が製品化, 日経エレクトロニクス, 日本, 日経ＢＰ社, １９９９年１１月, 第７５６号, 第１３４頁乃至第１３５頁
新井利明、外5名, ナノカーネル方式による異種ＯＳ共存技術「ＤＡＲＭＡ」の提案, 情報処理学会第５９回（平成１１年後期）全国大会講演論文集（１）, 日本, 社団法人情報処理学会, １９９９年 9月３０日, pp. 1-139 - 1-140
佐藤雅英、外6名, ナノカーネル方式による異種ＯＳ共存技術「ＤＡＲＭＡ」の実装, 情報処理学会第５９回（平成１１年後期）全国大会講演論文集（１）, 日本, 社団法人情報処理学会, １９９９年 9月３０日, pp. 1-141 - 1-142

(58)調査した分野(Int.Cl.[7], ＤＢ名)
G06F 9/46 - 9/54
G06F 13/10