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

特許第5877533号
(P5877533)

(45)発行日　平成28年3月8日(2016.3.8)　(24)登録日　平成28年2月5日(2016.2.5)

(51)Int.Cl.		F I		
G06F 11/30	(2006.01)	G06F	11/30	A
G06F 9/445	(2006.01)	G06F	9/06	610K
G06F 9/48	(2006.01)	G06F	9/46	311G

請求項の数 9　(全 22 頁)

(21)出願番号　特願2014-559432 (P2014-559432)
(86)(22)出願日　平成25年1月31日 (2013.1.31)
(86)国際出願番号　PCT/JP2013/052205
(87)国際公開番号　WO2014/118940
(87)国際公開日　平成26年8月7日 (2014.8.7)
審査請求日　平成26年9月17日 (2014.9.17)

(73)特許権者　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目７番３号
(74)代理人　100099461
弁理士　溝井　章司
(74)代理人　100152881
弁理士　山地　博人
(72)発明者　徳永　寿郎
日本国東京都千代田区丸の内二丁目７番３号　三菱電機株式会社内
(72)発明者　攝津　敦
日本国東京都千代田区丸の内二丁目７番３号　三菱電機株式会社内

審査官　多胡　滋

最終頁に続く

(54)【発明の名称】計算機装置及び計算機装置の制御方法

(57)【特許請求の範囲】
【請求項１】
割込みを検知する割込み検知機構が含まれるＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）と、
前記割込み検知機構が割り込みを検知した際に前記割込み検知機構により呼び出され、前記割込み検知機構が検知した割り込みがＣＰＵ例外であるか否かが判定される割込み判定部が含まれるＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）とを備える計算機装置であって、
前記計算機装置に、ＣＰＵ例外に対する処理を行うプログラムであるＲＡＳ（Ｒｅｌｉａｂｉｌｉｔｙ　Ａｖａｉｌａｂｉｌｉｔｙ　Ｓｅｒｖｉｃｅａｂｉｌｉｔｙ）モジュールが追加された場合に、
前記ＣＰＵが、
前記計算機装置の起動時に、前記ＲＡＳモジュールに含まれる第１の初期化手順を呼び出し、前記第１の初期化手順を実行して、前記ＲＡＳモジュールが用いる資源を初期化し、
前記ＲＡＳモジュールの前記第１の初期化手順の実行後に、前記ＯＳに含まれる初期化手順を呼び出し、前記初期化手順を実行して、前記ＯＳが用いる資源を初期化し、
前記ＯＳの前記初期化手順の実行後に、前記ＲＡＳモジュールに含まれる第２の初期化手順を呼び出し、前記第２の初期化手順を実行して、前記ＯＳに含まれる前記割込み判定部を前記ＲＡＳモジュールにコピーし、前記割込み検知機構が割込みを検知した際に前記

ＯＳの割込み判定部ではなく前記ＲＡＳモジュールにコピーされた割込み判定部を呼び出すように前記割込み検知機構を設定することを特徴とする計算機装置。
【請求項２】
前記計算機装置は、更に、
前記ＯＳのプログラムコード及び前記ＲＡＳモジュールのプログラムコードが格納されるメモリを備え、
前記ＯＳの前記初期化手順に対応するプログラムコードの最後に、前記初期化手順の呼び出し元のプログラムコードへのジャンプ命令が記述されており、
前記ＣＰＵは、
前記ＲＡＳモジュールの前記第１の初期化手順を実行して、前記メモリに格納されている、前記ＯＳの前記初期化手順に対応するプログラムコードの最後の記述を前記ＲＡＳモジュールの前記第２の初期化手順のプログラムコードへのジャンプ命令に変更することを特徴とする請求項１に記載の計算機装置。
【請求項３】
前記ＣＰＵは、
前記ＲＡＳモジュールの前記第１の初期化手順を実行して、前記メモリに格納されている、前記ＲＡＳモジュールの前記第２の初期化手順に対応するプログラムコードの最後の記述を前記ＯＳの前記初期化手順の呼び出し元のプログラムコードへのジャンプ命令に変更することを特徴とする請求項２に記載の計算機装置。
【請求項４】
前記ＲＡＳモジュールには、ＣＰＵ例外に対する処理を行うＣＰＵ例外処理部が含まれ、
前記割込み検知機構が割込みを検知した場合に、前記割込み検知機構により、前記ＲＡＳモジュールにコピーされた割込み判定部が呼び出され、
前記ＲＡＳモジュールにコピーされた割込み判定部により、前記割込み検知機構により検知された割込みがＣＰＵ例外であると判定された場合に、前記ＣＰＵ例外処理部によりＣＰＵ例外に対する処理が行われることを特徴とする請求項１～３のいずれかに記載の計算機装置。
【請求項５】
前記割込み検知機構が割込みを検知した場合に、前記割込み検知機構により、前記ＲＡＳモジュールにコピーされた割込み判定部が呼び出され、
前記ＲＡＳモジュールにコピーされた割込み判定部により、前記割込み検知機構により検知された割込みがＣＰＵ例外であると判定された場合に、前記ＣＰＵ例外処理部によりＣＰＵ例外の発生時に動作していたＯＳが特定され、前記ＣＰＵ例外処理部により、特定されたＯＳから情報が収集されてＣＰＵ例外に対する処理が行われることを特徴とする請求項４に記載の計算機装置。
【請求項６】
前記ＲＡＳモジュールには、ＣＰＵ例外以外の割込みの発生時に動作していたＯＳを特定するＯＳ特定部が含まれ、
前記割込み検知機構が割込みを検知した場合に、前記割込み検知機構により、前記ＲＡＳモジュールにコピーされた割込み判定部が呼び出され、
前記ＲＡＳモジュールにコピーされた割込み判定部により、前記割込み検知機構により検知された割込みがＣＰＵ例外以外であると判定された場合に、前記ＯＳ特定部により割込みの発生時に動作していたＯＳが特定され、特定されたＯＳにより割込みに対する処理が行われることを特徴とする請求項１～５のいずれかに記載の計算機装置。
【請求項７】
前記計算機装置は、更に、
ハイパーバイザーを備え、
前記ＣＰＵは、
前記ＲＡＳモジュールの前記第１の初期化手順の実行後に、前記ハイパーバイザーに含

まれる初期化手順を呼び出し、前記初期化手順を実行して、前記ハイパーバイザーが用いる資源を初期化し、
前記ハイパーバイザーの前記初期化手順の実行後に、前記ＯＳに含まれる初期化手順を呼び出し、前記初期化手順を実行して、前記ＯＳが用いる資源を初期化し、
前記ＯＳの前記初期化手順の実行後に、前記ＲＡＳモジュールに含まれる第２の初期化手順を呼び出し、前記第２の初期化手順を実行することを特徴とする請求項１～６のいずれかに記載の計算機装置。
【請求項８】
前記ＲＡＳモジュールには、ＣＰＵ例外に対する処理を行うＣＰＵ例外処理部が含まれ、
前記割込み検知機構が割込みを検知した場合に、前記割込み検知機構により、前記ＲＡＳモジュールにコピーされた割込み判定部が呼び出され、
前記ＲＡＳモジュールにコピーされた割込み判定部により、前記割込み検知機構により検知された割込みがＣＰＵ例外であると判定された場合に、前記ＣＰＵ例外処理部によりＣＰＵ例外の発生時に動作していたハイパーバイザーが特定され、前記ＣＰＵ例外処理部により、特定されたハイパーバイザーから情報が収集されてＣＰＵ例外に対する処理が行われることを特徴とする請求項７に記載の計算機装置。
【請求項９】
割込みを検知する割込み検知機構が含まれるＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）と、
前記割込み検知機構が割り込みを検知した際に前記割込み検知機構により呼び出され、前記割込み検知機構が検知した割り込みがＣＰＵ例外であるか否かが判定される割込み判定部が含まれるＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）とを備える計算機装置に、
ＣＰＵ例外に対する処理を行うプログラムであるＲＡＳ（Ｒｅｌｉａｂｉｌｉｔｙ　Ａｖａｉｌａｂｉｌｉｔｙ　Ｓｅｒｖｉｃｅａｂｉｌｉｔｙ）モジュールが追加された場合に、
前記ＣＰＵが、
前記計算機装置の起動時に、前記ＲＡＳモジュールに含まれる第１の初期化手順を呼び出し、前記第１の初期化手順を実行して、前記ＲＡＳモジュールが用いる資源を初期化し、
前記ＲＡＳモジュールの前記第１の初期化手順の実行後に、前記ＯＳに含まれる初期化手順を呼び出し、前記初期化手順を実行して、前記ＯＳが用いる資源を初期化し、
前記ＯＳの前記初期化手順の実行後に、前記ＲＡＳモジュールに含まれる第２の初期化手順を呼び出し、前記第２の初期化手順を実行して、前記ＯＳに含まれる前記割込み判定部を前記ＲＡＳモジュールにコピーし、前記割込み検知機構が割込みを検知した際に前記ＯＳの割込み判定部ではなく前記ＲＡＳモジュールにコピーされた割込み判定部を呼び出すように前記割込み検知機構を設定することを特徴とする計算機装置の制御方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）が実装されている計算機装置に、ＯＳのモジュールの修正なしに、ＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）例外に対処する異常対処機能（＝ＲＡＳ機能（ＲＡＳ：Ｒｅｌｉａｂｉｌｉｔｙ，　Ａｖａｉｌａｂｉｌｉｔｙ，　Ｓｅｒｖｉｃｅａｂｉｌｉｔｙ）を実現するプログラムであるＲＡＳモジュールを追加する技術に関する。
また、ＯＳとともにＨｙｐｅｒｖｉｓｏｒが実装されている計算機装置に、ＯＳ及びＨｙｐｅｒｖｉｓｏｒのモジュールの修正なしに、ＲＡＳモジュールを追加する技術に関する。
【０００２】
なお、Ｈｙｐｅｒｖｉｓｏｒは、計算機装置の仮想化を実現するソフトウェアである。

Ｈｙｐｅｒｖｉｓｏｒは、ＯＳと計算機ハードウェアの間に位置し、計算機装置の動作をエミュレートするソフトウェアであり、１つの計算機装置上で複数のＯＳを同時に動作させ、複数のＯＳ間での通信や資源共有の仲介などを行う。

また、ＣＰＵ例外は、ＣＰＵが通常の処理を続行できない状態になった場合（例えばゼロ除算など）の例外である。

ＣＰＵ例外発生時には、前もって設定しておいた別のプログラムを呼び出すことができる仕組みとなっている。

なお、本明細書では、ＣＰＵ例外と、ＣＰＵ例外以外の割込みの双方を「割込み」と呼ぶ。

また、「割込み」のうち、ＣＰＵ例外以外の割込みは「標準割込み」と呼び、ＣＰＵ例外とは区別する。

【背景技術】

【０００３】

従来技術では、ＯＳ（一つまたは複数存在してもよい）やＨｙｐｅｒｖｉｓｏｒ（Ｈｙｐｅｒｖｉｓｏｒはない構成でもよい）が、それぞれ独立したモジュール構成で動作している計算機装置において、ＣＰＵ例外に対応したＲＡＳ機能を実現する場合、ＯＳやＨｙｐｅｒｖｉｓｏｒにＣＰＵ例外に対応する処理を追加するなどの方法で実現していた。

例えば、特許文献１では、本体系障害（＝本明細書におけるＣＰＵ例外にあたる）に遭遇した仮想計算機で実行状態にあったプロセスの障害情報を障害情報格納領域から取り出す手段をＶＭモニタ（＝本明細書におけるＨｙｐｅｒｖｉｓｏｒにあたる）に設けた構成が開示されている。

また、例えば、特許文献２では、Ｈｙｐｅｒｖｉｓｏｒによって複数のＯＳを動作させる仮想計算機において例外を解決する技術が開示されている。

具体的には、例外発生時にＯＳが実行していた処理部分のメモリイメージをＨｙｐｅｒｖｉｓｏｒ側にコピーし、Ｈｙｐｅｒｖｉｓｏｒが、例外発生時にＯＳが実行していた処理のうちの特権命令をエミュレートすることで例外を解決する技術が開示されている。

【先行技術文献】

【特許文献】

【０００４】

【特許文献１】特開平０１－０５３２３８号公報

【特許文献２】特開２００６－１５５２７２号公報

【発明の概要】

【発明が解決しようとする課題】

【０００５】

従来、ＯＳ（一つまたは複数存在してもよい）とＨｙｐｅｒｖｉｓｏｒ（Ｈｙｐｅｒｖｉｓｏｒは存在しなくてもよい）が独立したモジュールで構成されている計算機システムに、そのＯＳの例外処理に対応したＲＡＳ機能を追加する場合は、そのＯＳやＨｙｐｅｒｖｉｓｏｒやＣＰＵが持つ割込み検知機構に対して、修正・変更を加える必要があった。

例えば、特許文献１及び特許文献２の技術においても、Ｈｙｐｅｒｖｉｓｏｒ側にＣＰＵ例外に対応する機能を予め持たせてＲＡＳ機能を実現している。

このため、ＯＳやＨｙｐｅｒｖｉｓｏｒのモジュールに修正を加えることが困難な場合（技術上の難易度が高くコストがかかる場合や、ライセンス上修正が加えられない場合や、品質保持の観点から修正を行いたくない場合なども含む）は、ＲＡＳ機能を追加することが困難であるという課題がある。

【０００６】

この発明は、上記のような課題を解決することを主な目的とする。

つまり、この発明は、ＯＳへの修正なしに、計算機装置にＲＡＳモジュールを追加でき、適切にＲＡＳ機能が実現されるようにすることを主な目的とする。

【課題を解決するための手段】

【０００７】

本発明に係る計算機装置は、
割込みを検知する割込み検知機構が含まれるＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）と、
前記割込み検知機構が割り込みを検知した際に前記割込み検知機構により呼び出され、前記割込み検知機構が検知した割り込みがＣＰＵ例外であるか否かが判定される割込み判定部が含まれるＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）とを備える計算機装置であって、
前記計算機装置に、ＣＰＵ例外に対する処理を行うプログラムであるＲＡＳ（Ｒｅｌｉａｂｉｌｉｔｙ　Ａｖａｉｌａｂｉｌｉｔｙ　Ｓｅｒｖｉｃｅａｂｉｌｉｔｙ）モジュールが追加された場合に、
前記ＣＰＵが、
前記計算機装置の起動時に、前記ＲＡＳモジュールに含まれる第１の初期化手順を呼び出し、前記第１の初期化手順を実行して、前記ＲＡＳモジュールが用いる資源を初期化し、
前記ＲＡＳモジュールの前記第１の初期化手順の実行後に、前記ＯＳに含まれる初期化手順を呼び出し、前記初期化手順を実行して、前記ＯＳが用いる資源を初期化し、
前記ＯＳの前記初期化手順の実行後に、前記ＲＡＳモジュールに含まれる第２の初期化手順を呼び出し、前記第２の初期化手順を実行して、前記ＯＳに含まれる前記割込み判定部を前記ＲＡＳモジュールにコピーし、前記割込み検知機構が割込みを検知した際に前記ＯＳの割込み判定部ではなく前記ＲＡＳモジュールにコピーされた割込み判定部を呼び出すように前記割込み検知機構を設定することを特徴とする。

【発明の効果】

【０００８】

本発明によれば、計算機装置に実装されているＯＳへの修正なしに、ＲＡＳモジュールを計算機装置に追加することができ、また、ＣＰＵの割込み検知機構が割込みを検知した場合に適切にＲＡＳモジュールが呼び出されて、ＲＡＳ機能が実現される。

【図面の簡単な説明】

【０００９】

【図１】実施の形態１に係る計算機装置の構成例を示す図。
【図２】実施の形態１に係る計算機装置における初期化処理の概要を示すフローチャート図。
【図３】実施の形態１に係るＲＡＳモジュールの第１の初期化手順の詳細を示すフローチャート図。
【図４】実施の形態１に係るＯＳの初期化手順の詳細を示すフローチャート図。
【図５】実施の形態１に係るＲＡＳモジュールの第２の初期化手順の詳細を示すフローチャート図。
【図６】実施の形態１に係る計算機装置におけるＣＰＵ例外発生時の動作例を示す図。
【図７】実施の形態１に係る計算機装置におけるＣＰＵ例外発生時の動作例を示すフローチャート図。
【図８】実施の形態２に係る計算機装置の構成例を示す図。
【図９】実施の形態２に係る計算機装置における初期化処理の概要を示すフローチャート図。
【図１０】実施の形態２に係るＲＡＳモジュールの第１の初期化手順の詳細を示すフローチャート図。
【図１１】実施の形態２に係る計算機装置におけるＣＰＵ例外発生時の動作例を示す図。
【図１２】実施の形態２に係る計算機装置におけるＣＰＵ例外発生時の動作例を示す図。
【図１３】実施の形態２に係る計算機装置におけるＣＰＵ例外発生時の動作例を示すフローチャート図。
【図１４】実施の形態３に係る計算機装置における割込み発生時の動作例を示す図。
【図１５】実施の形態３に係る計算機装置における割込み発生時の動作例を示すフローチ

ャート図。
【図１６】実施の形態１に係る計算機装置のＲＡＳモジュール追加前の構成例を示す図。
【図１７】実施の形態２に係る計算機装置のＲＡＳモジュール追加前の構成例を示す図。
【発明を実施するための形態】
【００１０】
以下の実施の形態１～３では、１つまたは複数のＯＳやＨｙｐｅｒｖｉｓｏｒが動作する計算機装置を説明する。
より具体的には、ＯＳやＨｙｐｅｒｖｉｓｏｒのモジュールの修正なしに、ＣＰＵ例外に対処するＲＡＳモジュールを追加でき、また、ＣＰＵ例外が発生した際に、適切にＲＡＳモジュールが呼び出され、ＲＡＳモジュールによってＣＰＵ例外に対する処理が実施される計算機装置及び計算機装置の制御方法を説明する。
【００１１】
また、実施の形態１～３では、ＣＰＵ例外の発生に重ねて、ＯＳやＨｙｐｅｒｖｉｓｏｒに障害が発生した場合でも、ＲＡＳ機能が実行される計算機装置を説明する。
ＯＳやＨｙｐｅｒｖｉｓｏｒ自身がＲＡＳ機能を実現するように構成されている場合は、ＣＰＵ例外発生時に動作していたＯＳやＨｙｐｅｒｖｉｓｏｒ自身の障害の発生が重なったときには、ＲＡＳ機能が実行できないという課題がある。
実施の形態１～３では、このような課題を解決する計算機装置を説明する。
【００１２】
また、実施の形態１～３では、ＣＰＵ例外発生時に、どのＯＳ（またはＨｙｐｅｒｖｉｓｏｒ）が動作していたかを判定することにより、該当するＯＳ（またはＨｙｐｅｒｖｉｓｏｒ）の障害情報を取得できるＲＡＳモジュールを説明する。
また、ＣＰＵ例外ではなく、標準割込みが発生した場合に、ＯＳ側の割込み判定部を参照して対応する割込み処理部を呼び出すことにより、ＯＳ側で割込み登録内容が変更されていた場合（計算機装置の立上げから当該割込みの発生時までの間に動的に割込み登録内容が変更されていた場合も含む）でも、正しく割込み処理部を呼び出すことができるＲＡＳモジュールを説明する。
【００１３】
実施の形態１.
［実施の形態１：構成の説明］
図１は、実施の形態１に係る計算機装置（１００）の構成例を示すブロック図である。
計算機装置（１００）の構成は、ハードウェアとソフトウェアに大別される。
【００１４】
計算機装置（１００）には、ハードウェアとして、ＣＰＵ（１０１）、メモリ（１０３）、二次記憶装置（１０４）が存在する。
ＣＰＵ（１０１）は、１つでも複数存在する構成（マルチコア、マルチＣＰＵ、マルチプロセッサなど）でもよい。
ＣＰＵ（１０１）には、割込み検知機構（１０２）が存在する。
割込み検知機構（１０２）は、割込み（ＣＰＵ例外及び標準割込み）を検知する。
メモリ（１０３）は、ＲＡＭ（Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）である。
また、二次記憶装置（１０４）は、例えば、ＲＯＭ（Ｒｅａｄ Ｏｎｌｙ Ｍｅｍｏｒｙ）、ＨＤＤ（Ｈａｒｄ Ｄｉｓｋ Ｄｒｉｖｅ）、ＳＳＤ（Ｓｏｌｉｄ Ｓｔａｔｅ Ｄｒｉｖｅ）である。
後述するソフトウェアは、二次記憶装置（１０４）に記憶されており、実行の際に、二次記憶装置（１０４）からメモリ（１０３）にロードされ、順次メモリ（１０３）からＣＰＵ（１０１）に読み込まれ、実行される。
また、後述するソフトウェアの実行により得られた情報、データ、変数値等がメモリ（１０３）やＣＰＵ（１０１）内のレジスタに記憶される。
また、図１では、図示を省略しているが、計算機装置（１００）には、入出力装置や通信装置を含む各種のデバイスが装備されている。

【００１５】
　また、計算機装置（１００）のソフトウェアとして、ＯＳ（１１０）、ＲＡＳモジュール（１３０）、ブートプログラム（１４０）がそれぞれ別モジュールとして存在する。
　ブートプログラム（１４０）は、計算機装置（１００）が起動されるときに実行される。
　ＯＳ（１１０）は1つでも複数存在してもよい。
　ＯＳ（１１０）のモジュールには、初期化手順（１１１）、割込み判定部（１１２）、ＣＰＵ例外処理部（１１３）、割込み処理部（１１５）が存在する。
【００１６】
　初期化手順（１１１）は、ＯＳ（１１０）が用いる資源を初期化するためのプログラムである。
　ＯＳ（１１０）が用いる資源には、ハードウェア資源、ソフトウェア資源の両者が含まれる。
【００１７】
　割込み判定部（１１２）は、割込み検知機構（１０２）が割り込みを検知した際に割込み検知機構（１０２）により呼び出されるプログラムである。
　割込み判定部（１１２）では、割込み検知機構（１０２）が検知した割り込みがＣＰＵ例外であるか標準割込みであるかが判定される。
　なお、後述するように、計算機装置（１００）の起動時に、割込み判定部（１１２）はＲＡＳモジュール（１３０）にコピーされて割込み判定部（１３４）となり、また、割込み検知機構（１０２）の設定が変更されて、割込み検知機構（１０２）が割り込みを検知した際に割込み検知機構（１０２）は割込み判定部（１３４）を呼び出す。
　このため、計算機装置（１００）にＲＡＳモジュール（１３０）が追加された後は、割込み判定部（１１２）は割込み検知機構（１０２）から呼び出されることはない。
【００１８】
　ＣＰＵ例外処理部（１１３）及び割込み処理部（１１５）は、ＣＰＵ例外ではない標準割込みが発生した場合に実行されるプログラムである。
　ＣＰＵ例外処理部（１１３）及び割込み処理部（１１５）の動作の詳細は、実施の形態３において説明する。
【００１９】
　ＲＡＳモジュール（１３０）は、ＣＰＵ例外に対する処理を行うプログラムである。
　ＲＡＳモジュール（１３０）には、第１の初期化手順（１３２）、第２の初期化手順（１３３）、割込み判定部（１３４）、ＯＳ特定部（１３５）、故障検知部（１３６）、故障情報収集部（１３７）、故障同定部（１３８）、故障対処部（１３９）が存在する。
　なお、ＲＡＳモジュール（１３０）は、単にＲＡＳともいう。
【００２０】
　第１の初期化手順（１３２）は、ＯＳ（１１０）の初期化手順（１１１）が実行される前に実行されるプログラムである。
　第１の初期化手順（１３２）では、ＲＡＳモジュール（１３０）が用いる資源が初期化される。
　ＲＡＳモジュール（１３０）が用いる資源には、ハードウェア資源、ソフトウェア資源の両者が含まれる。
　また、第１の初期化手順（１３２）では、後述の第２の初期化手順（１３３）が実行されるように、ＯＳ（１１０）の初期化手順（１１１）のプログラムコードの最後の部分を書き換える処理が行われる。
【００２１】
　第２の初期化手順（１３３）は、ＯＳ（１１０）の初期化手順（１１１）の実行後に実行されるプログラムである。
　第２の初期化手順（１３３）では、ＯＳ（１１０）に含まれる割込み判定部（１１２）をＲＡＳモジュール（１３０）にコピーし、割込み検知機構（１０２）が割込みを検知し

た際にＯＳ（１１０）の割込み判定部（１１２）ではなくＲＡＳモジュール（１３０）にコピーされた割込み判定部（１３４）を呼び出すように割込み検知機構（１０２）を設定する処理が行われる。

【００２２】

割込み判定部（１３４）は、前述したように、ＲＡＳモジュール（１３０）にコピーされた割込み判定部（１１２）である。

このため、割込み判定部（１３４）では、割込み判定部（１１２）と同じ処理が行われる。

すなわち、割込み判定部（１３４）では、割込み検知機構（１０２）が検知した割り込みがＣＰＵ例外であるか標準割込みであるかが判定される。

【００２３】

ＯＳ特定部（１３５）は、割込み判定部（１３４）により、割込み検知機構（１０２）が検知した割り込みが標準割込みであると判定された場合に実行されるプログラムである。

ＯＳ特定部（１３５）では、割込みが発生したときに動作していたＯＳ（１１０）が特定される。

ＯＳ特定部（１３５）の動作の詳細は、実施の形態３で説明する。

【００２４】

故障検知部（１３６）、故障情報収集部（１３７）、故障同定部（１３８）及び故障対処部（１３９）は、それぞれ、割込み判定部（１３４）により、割込み検知機構（１０２）が検知した割り込みがＣＰＵ例外であると判定された場合に実行されるプログラムである。

故障検知部（１３６）では、ＣＰＵ例外の原因となった故障が特定される。

故障情報収集部（１３７）では、ＣＰＵ例外が発生したときに動作していたＯＳ（１１０）が特定され、特定されたＯＳ（１１０）から故障に関する情報が収集される。

故障同定部（１３８）では、故障情報収集部（１３７）により収集された情報から、故障に対応する故障対処方法が同定される。

故障対処部（１３９）では、故障同定部（１３８）で同定された故障対処方法が実施される。

故障検知部（１３６）、故障情報収集部（１３７）、故障同定部（１３８）及び故障対処部（１３９）は、それぞれ、ＣＰＵ例外処理部の例に相当する。

【００２５】

なお、上記の割込み判定部（１１２）及び割込み判定部（１３４）は、ＣＰＵ例外および標準割込みに対処する処理（プログラム）のメモリ（１０３）上のアドレスを保持する。

また、上記の割込み処理部（１１５）は、割込み発生時に対処するための処理（プログラム）を保持する。

【００２６】

上述したように、ＯＳ（１１０）、ＲＡＳモジュール（１３０）、ブートプログラム（１４０）は、それぞれプログラムであり、計算機装置（１００）では、ＣＰＵ（１０１）がこれらのプログラムを読込み、プログラムに記述されている内容に従って処理を行う。

以下では、「ＣＰＵ（１０１）が～する」又は「ＣＰＵ（１０１）により～される」という説明を行うとともに、理解のしやすさ、文脈を考慮して、ＯＳ（１１０）、ＲＡＳモジュール（１３０）、ブートプログラム（１４０）が「～する」、または、これらに含まれる要素（例えば、第１の初期化手順（１３２））が「～する」という表現を用いることもある。

本明細書では、ソフトウェアが動作主体として記述されていても、その記述は、ＣＰＵ（１０１）によるプログラムの実行により処理が行われていることを表している。

【００２７】

［実施の形態１：動作の概要説明（初期化時の全体的な動作の概要）］

まず、本実施の形態に係る計算機装置（１００）の初期化時の全体的な動作の概要について説明する。
実施の形態１に係る計算機装置（１００）を起動する際の、ＲＡＳモジュール（１３０）、ＯＳ（１１０）の初期化処理の全体的なフローチャートを図２に示す。
【００２８】
最初に、計算機装置（１００）が起動されると、ＣＰＵ（１０１）により、ブートプログラム（１４０）が実行され、ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）が呼び出され（Ｓ２０１）、ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）が実行される（Ｓ２０２）。
ブートプログラム（１４０）やＲＡＳモジュール（１３０）の「第１の初期化手順」（Ｓ２０２）の詳細については後述する。
【００２９】
次に、ＯＳ（１１０）の初期化手順（１１１）が呼び出され、ＯＳ（１１０）の初期化手順（１１１）が実行される（Ｓ２０４）。
ＯＳの「初期化手順」（Ｓ２０４）の詳細については後述する。
【００３０】
最後に、ＲＡＳモジュール（１３０）の第２の初期化手順（１３３）が呼び出され、ＲＡＳモジュール（１３０）の第２の初期化手順（１３３）が実行される（Ｓ２０５）。
ＲＡＳモジュール（１３０）の「第２の初期化手順」（Ｓ２０５）の詳細については後述する。
【００３１】
また、計算機装置（１００）に複数のＯＳ（１１０）が存在する場合は、ＯＳ（１１０）の「初期化手順」（Ｓ２０４）では、それぞれのＯＳ（１１０）の初期化手順（１１１）が順番に実行される。
【００３２】
［実施の形態１：動作の概要説明（ＲＡＳモジュールの「第１の初期化手順」の動作）］
次に、実施の形態１における初期化時の動作の詳細について説明する。
前述のＲＡＳモジュール（１３０）の「第１の初期化手順」（Ｓ２０２）の詳細フローチャートを図３に示す。
【００３３】
「第１の初期化手順」（Ｓ２０２）では、まず、ＣＰＵ（１０１）が第１の初期化手順（１３２）を実行して、ＲＡＳモジュール（１３０）の初期化を実施する（Ｓ３０１）。
Ｓ３０１では、主にＲＡＳモジュール（１３０）が使用する資源の初期化処理が実施される。
その後、ＣＰＵ（１０１）は、「ＯＳの「初期化手順」実行後にＲＡＳモジュールの「第２の初期化手順」を呼び出す」という処理を、ＯＳの「初期化手順」の最後に付け加える処理を実施する（Ｓ３０２）。
具体的には、ＲＡＳモジュール（１３０）の第２の初期化手順（１３３）はメモリ（１０３）上に配置されたプログラムであり、ＯＳ（１１０）の初期化手順（Ｓ２０４）の最後で、このＲＡＳモジュール（１３０）の第２の初期化手順（１３３）のプログラムのメモリ（１０３）上のアドレスを呼び出すように、ＯＳの初期化手順（１１１）の最後の部分を書換える。
つまり、ＣＰＵ（１０１）は、メモリ（１０３）に格納されている、ＯＳ（１１０）の初期化手順（１１１）のプログラムコードの最後の記述をＲＡＳモジュール（１３０）の第２の初期化手順（１３３）のプログラムコードへのジャンプ命令に変更する。
この処理のより具体的な実現方法の例を、以下の（１）～（４）に示す。
【００３４】
（１）ＯＳ（１１０）のコンパイル後の実行ファイルのシンボル情報から、ＯＳ（１１０）の初期化手順（１１１）のコード位置（＝メモリ上のプログラムのアドレス）やサイズを、ＲＡＳモジュール（１３０）側で把握できるようにする。

例えば、ＯＳ（１１０）のシンボル情報をメモリ（１０３）上にロードしておきＲＡＳモジュール（１３０）側からシンボル情報を参照できるようにする方法や、予めＯＳ（１１０）のシンボル情報をＲＡＳモジュール（１３０）に（ハードコーディングするなどして）取り込んでおく方法により、ＲＡＳモジュール（１３０）がＯＳ（１１０）の初期化手順（１１１）のコード位置やサイズを把握できるようにする。

（２）ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）が、前述（１）で把握可能としたＯＳ（１１０）の初期化手順（１１１）のコード位置とサイズ情報より、ＯＳ（１１０）の初期化手順（１１１）の最後のコード位置を把握する。

（３）この最後のコード位置の部分には、ＯＳ（１１０）の初期化手順（１１１）の呼び出し元の位置に戻るというコード（ジャンプ命令）が書かれている。

ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）は、この呼び出し元の位置を記録しておき、この最後のコード部分を、ＲＡＳモジュール（１３０）の第２の初期化手順（１３３）のコード位置へのジャンプ命令に修正する。

（４）ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）は、第２の初期化手順（１３３）のコードの最後の部分を、前述（３）で記録したＯＳ（１１０）の初期化手順（１１１）の呼び出し元の位置へのジャンプ命令に修正する。

【００３５】

次に、ＯＳ（１１０）の初期化手順（１１１）が呼び出される（Ｓ３０３）。

具体的には、ＯＳ（１１０）の初期化手順（１１１）のプログラムのメモリ上のアドレスが呼び出される。

【００３６】

［実施の形態１：動作の概要説明（ＯＳの「初期化手順」の動作）］

次に、前述のＯＳ（１１０）の「初期化手順」（Ｓ２０４）の詳細フローチャートを図４に示す。

【００３７】

ここでは、ＣＰＵ（１０１）は、ＯＳ（１１０）の初期化を実施する（Ｓ４０１）。

Ｓ４０１では、主にＯＳ（１１０）自身が使用する資源の初期化処理が実施される。

その後、ＣＰＵ（１０１）は、前述のＲＡＳモジュール（１３０）の「第１の初期化手順」（Ｓ２０２）にて書換えられた通り、ＲＡＳモジュール（１３０）の第２の初期化手順（１３３）を呼び出す（Ｓ４０２）。

具体的には、ＣＰＵ（１０１）は、ＲＡＳモジュール（１３０）の第２の初期化手順（１３３）のプログラムのメモリ上のアドレスを呼び出すことにより実現する。

【００３８】

［実施の形態１：動作の概要説明（ＲＡＳモジュールの「第２の初期化手順」の動作）］

最後に、前述のＲＡＳモジュール（１３０）の「第２の初期化手順」（Ｓ２０５）の詳細フローチャートを図５に示す。

ここでは、ＣＰＵ（１０１）は、ＯＳ（１１０）の割込み判定部（１１２）のプログラムコードを、ＲＡＳモジュール（１３０）にコピーする（Ｓ５０１）。

ＲＡＳモジュール（１３０）にコピーされた割込み判定部（１１２）は、割込み判定部（１３４）となる。

その後、ＣＰＵ（１０１）は、割込み発生時にＲＡＳモジュール（１３０）の割込み判定部（１３４）を呼び出すように割込み検知機構（１０２）を設定する処理を実施する（Ｓ５０２）。

【００３９】

［実施の形態１：動作の概要説明（ＲＡＳモジュールを追加する方法）］

以上では、既にＲＡＳモジュール（１３０）が追加された構成での、起動時の初期化動作を説明したが、以下では、ＲＡＳモジュール（１３０）を計算機装置（１００）に追加する方法を説明する。

【００４０】

ＲＡＳモジュール（１３０）を追加する前の計算機装置（１００）の構成例を図１６に

示す。
ＲＡＳモジュール（１３０）を追加する前の計算機装置（１００）は、ＯＳ（一つでも複数のＯＳが存在する構成でもよい）が独立したモジュールで構成されている。
ＲＡＳモジュール（１３０）が追加される前は、計算機装置（１００）は、図１６のようにＲＡＳモジュール（１３０）がない状態で、通常動作を行っている。
図１６の計算機装置（１００）の起動時は、まず計算機装置（１００）上のブートプログラム（１４０）が立ち上がり、ブートプログラム（１４０）からＯＳ（１１０）の初期化手順（１１１）が呼び出される。
【００４１】
この図１６の計算機装置（１００）に対し、本実施の形態のＲＡＳモジュール（１３０）を追加する場合、まず、追加するＲＡＳモジュール（１３０）のプログラムを、計算機装置（１００）の二次記憶装置（１０４）上の空き領域に配置する。
また、計算機装置（１００）のブートプログラム（１４０）を、計算機装置（１００）の起動時にＲＡＳモジュール（１３０）の第１の初期化手順（１３２）を呼び出すよう変更する。
具体的には、ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）のプログラムのメモリ（１０３）上のアドレスを、ブートプログラム（１４０）から呼び出すよう変更する。
以上がＲＡＳモジュール（１３０）を計算機装置（１００）に追加する方法の概要である。
以上のようにＲＡＳモジュール（１３０）を計算機装置（１００）に追加した後（図１の状態）に、計算機装置（１００）を起動した場合、前述した図２のフローチャートのような動作が行われる。
【００４２】
以上のような構成、動作により、ＯＳ（一つでも複数のＯＳが存在する構成でもよい）が独立したモジュールで構成されている計算機装置に、ＯＳのモジュールへの修正なしに、ＯＳが所有する割込み判定部に対応したＲＡＳ機能を、容易に追加することが可能である。
【００４３】
［実施の形態１：動作の概要説明（割込み発生時の動作概要）］
次に、割込みが発生した場合の実施の形態１に係る計算機装置（１００）動作の概要を説明する。
図６は、ＯＳ（１１０）の動作時に割込みが発生した場合の処理の流れを示す。
ここで、ＯＳは複数存在していてもよい。
なお、図６では、説明に直接関係のないメモリ（１０３）、二次記憶装置（１０４）、ブートプログラム（１４０）の図示は省略している。
また、図６の実線の矢印は処理の流れを示しており、破線の矢印は処理の内容を示している。
【００４４】
割込みが発生した場合に、ＣＰＵ（１０１）の割込み検知機構（１０２）が、ＲＡＳモジュール（１３０）を呼び出す。
ＲＡＳモジュール（１３０）は、発生した割込みがＣＰＵ例外であるかどうかを判定する。
そして、発生した割込みがＣＰＵ例外であれば、ＲＡＳモジュール（１３０）は、どのＯＳが動作していたかを、計算機装置（１００）のプログラムカウンタ（ＣＰＵの実行アドレスを保持しているレジスタ）により判定（どのＯＳが動作中だったかを、プログラムカウンタが保持する実行中のコードの場所より判定）し、該当するＯＳを特定し、特定したＯＳから故障情報を収集する。
この動作により、ＯＳが複数の構成の場合、ＣＰＵ例外発生時に、どのＯＳが動作していたかを判定することが可能となり、該当のＯＳから障害情報を収集することができる。

【００４５】
［実施の形態１：動作の概要説明（割込み発生時のフローチャート）］
割込み発生時のフローチャートを図７に示す。
【００４６】
最初に、割込み検知機構（１０２）が割込みを検知する（Ｓ７０１）。
次に、割込み検知機構（１０２）は、ＲＡＳモジュール（１３０）の割込み判定部（１３４）を呼び出す（Ｓ７０２）。
割込み判定部（１３４）は、割込み検知機構（１０２）が検知した割込みがＣＰＵ例外か標準割込みかの判定を行う（Ｓ７０３）。
【００４７】
割込み検知機構（１０２）が検知した割込みが標準割込みである場合（Ｓ７０３でＮＯ）は、処理が図１５のＳ１００１へ移る。
図１５のフローチャートの詳細は、実施の形態３において説明する。
一方、割込み検知機構（１０２）が検知した割込みがＣＰＵ例外である場合（Ｓ７０３でＹＥＳ）は、ＲＡＳモジュール（１３０）の故障検知部（１３６）がＣＰＵ例外の原因の故障を特定する（Ｓ７０５）。
次に、ＲＡＳモジュール（１３０）の故障情報収集部（１３７）が、ＣＰＵ例外発生時に動作していたＯＳ（１１０）を特定し、該当するＯＳ（１１０）から故障情報を収集する（Ｓ７０６－１）。
なお、ＲＡＳモジュール（１３０）の故障情報収集部（１３７）は、前述の通り計算機装置（１００）のプログラムカウンタにより、ＣＰＵ例外発生時に動作していたＯＳ（１１０）を特定する。
次に、ＲＡＳモジュール（１３０）の故障同定部（１３８）が、Ｓ７０５で特定された故障に対応する故障対処方法を同定する（Ｓ７０７）。
最後に、ＲＡＳモジュール（１３０）の故障対処部（１３９）が、Ｓ７０７で同定された故障対処方法に従って、故障に対処する処理を行う（Ｓ７０８）。
【００４８】
以上、実施の形態１によれば、ＣＰＵ例外発生時にＲＡＳモジュール（１３０）内の割込み判定部（１３４）、故障検知部（１３６）等が実行されることにより、ＣＰＵ例外とＯＳの障害の発生が重なった場合でも、ＲＡＳ機能を実行することができる。
【００４９】
以上、本実施の形態では、ＯＳ（一つでも複数のＯＳが存在する構成でもよい）が独立したモジュールで構成されている計算機装置に、ＯＳのモジュールへの修正なしに、ＯＳが所有する割込み判定部に対応したＲＡＳ機能を追加することを可能とするＲＡＳ方式を説明した。
より具体的には、本実施の形態では主に以下の（１）～（４）を説明した。
（１）計算機装置の起動時に呼び出されるＲＡＳモジュールの「初期化手順」は、ＯＳの「初期化手順」よりも前に実行される「第１の初期化手順」と、ＯＳの「初期化手順」が完了した後に実行される「第２の初期化手順」の２つに分かれる。
（２）ＲＡＳモジュールの「第１の初期化手順」では、ＯＳの「初期化手順」の最後の部分が、ＯＳの「初期化手順」の後にＲＡＳモジュールの「第２の初期化手順」を呼び出すように書換えられる。
（３）ＲＡＳモジュールの「第２の初期化手順」では、ＯＳの「割込み判定部」がＲＡＳモジュールにコピーされる。
（４）ＲＡＳモジュールの「第２の初期化手順」では、割込み発生時にＯＳの「割込み判定部」ではなくＲＡＳモジュールの「割込み判定部」を呼び出すように、「割込み検知機構」が設定される。
【００５０】
また、本実施の形態では、ＣＰＵ例外発生時にＲＡＳモジュール内の割込み判定部、故障検知部等が実行されることにより、ＣＰＵ例外とＯＳの障害の発生が重なった場合でも

、ＲＡＳ機能が実行されるＲＡＳ方式を説明した。
【００５１】
また、本実施の形態では、ＲＡＳモジュール内の故障情報収集部が実行されることにより、ＣＰＵ例外発生時に、どのＯＳが動作していたかを、ＣＰＵ例外発生時のプログラムカウンタから判定し、該当のＯＳから障害情報を収集することが可能なＲＡＳ方式を説明した。
【００５２】
実施の形態２.
［実施の形態２：構成の説明］
図８は、実施の形態２に係る計算機装置（１００）の構成例を示すブロック図である。
実施の形態１の構成（図１）との違いは、Ｈｙｐｅｒｖｉｓｏｒ（１２０）およびＨｙｐｅｒｖｉｓｏｒ（１２０）の初期化手順（１２１）が存在することのみであり、他は実施の形態１の構成（図１）と同じある。
本実施の形態でも、ＯＳ（１１０）は１つでも複数存在する構成でもよい。
ＣＰＵ（１０１）も、１つでも複数存在する構成（マルチコア、マルチＣＰＵ、マルチプロセッサなど）でもよい。
【００５３】
［実施の形態２：動作の概要説明（初期化時の全体的な動作の概要）］
実施の形態２における動作の説明として、まず、初期化時の全体的な動作の概要について説明する。
実施の形態２における計算機装置（１００）を起動する際の、ＲＡＳモジュール（１３０）、ＯＳ（１１０）、Ｈｙｐｅｒｖｉｓｏｒ（１２０）のモジュールの初期化処理の全体的なフローを図９に示す。
【００５４】
ここで、実施の形態１の初期化時の動作（図２）との違いは、Ｓ２０２とＳ２０４の処理の間に、Ｈｙｐｅｒｖｉｓｏｒの「初期化手順」（Ｓ２０３）が追加されている部分であり、他は実施の形態１のフロー（図２）と同じである。
このため、ここではＳ２０３の処理とその前後の処理を主に説明する。
【００５５】
まず、図９のＲＡＳモジュールの「第１の初期化手順」（Ｓ２０２）までは、実施の形態１と同じである。
次に、Ｈｙｐｅｒｖｉｓｏｒ（１２０）の初期化手順（１２１）が呼び出されて実行される（Ｓ２０３）。
Ｓ２０３では、ＣＰＵ（１０１）は、主にＨｙｐｅｒｖｉｓｏｒ（１２０）が使用する資源の初期化などを実施する。
次に、図９のＯＳの「初期化手順」（Ｓ２０４）以降は、実施の形態１と同じである。
また、Ｓ２０５の詳細は、実施の形態１の図５に示すとおりである。
また、ＯＳが複数存在する構成の場合には、図９のフローチャートのＯＳの「初期化手順」（Ｓ２０４）で、それぞれのＯＳ（１１０）の初期化手順（１１１）が順番に実行される。
【００５６】
［実施の形態２：動作の概要説明（ＲＡＳモジュールの「第１の初期化手順」の動作）］
次に、実施の形態２における初期化時の動作の詳細について説明する。
前述のＲＡＳモジュールの「第１の初期化手順」（Ｓ２０２）の詳細フローを図１０に示す。
ここで、実施の形態１のＲＡＳモジュールの「第１の初期化手順」（図３）との違いは、最後にＯＳの「初期化手順」（図３のＳ３０３）を呼び出すのではなく、Ｈｙｐｅｒｖｉｓｏｒの「初期化手順」を呼び出す（図１０のＳ３０４）部分のみであるため、この部分のみ説明する。
【００５７】

Ｓ３０４では、ＣＰＵ（１０１）は、Ｈｙｐｅｒｖｉｓｏｒ（１２０）の初期化手順（１２１）を呼び出す。
具体的には、ＣＰＵ（１０１）は、Ｈｙｐｅｒｖｉｓｏｒ（１２０）の初期化手順（１２１）のプログラムのメモリ（１０３）上のアドレスを呼び出すことにより実現する。
【００５８】
［実施の形態２：動作の概要説明（ＲＡＳモジュールを追加する方法）］
以上では、既にＲＡＳモジュール（１３０）が追加された構成での、起動時の初期化動作を説明したが、以下では、ＲＡＳモジュール（１３０）を計算機装置（１００）に追加する方法を説明する。
【００５９】
ＲＡＳモジュール（１３０）を追加する前の計算機装置（１００）の構成例を図１７に示す。
ＲＡＳモジュール（１３０）を追加する前の計算機装置（１００）は、ＯＳ（一つでも複数のＯＳが存在する構成でもよい）と、Ｈｙｐｅｒｖｉｓｏｒが独立したモジュールで構成されている。
ＲＡＳモジュール（１３０）が追加される前は、計算機装置（１００）は、図１７のようにＲＡＳモジュール（１３０）がない状態で、通常動作を行っている。
図１７の計算機装置（１００）の起動時は、まず計算機装置（１００）上のブートプログラム（１４０）が立ち上がり、ブートプログラム（１４０）からＨｙｐｅｒｖｉｓｏｒ（１２０）の初期化手順（１２１）が呼び出され、Ｈｙｐｅｒｖｉｓｏｒ（１２０）の初期化手順（１２１）からＯＳ（１１０）の初期化手順（１１１）が呼び出される。
【００６０】
この図１７の計算機装置（１００）に対し、本実施の形態のＲＡＳモジュール（１３０）を追加する場合、まず、追加するＲＡＳモジュール（１３０）のプログラムを、計算機装置（１００）の二次記憶装置（１０４）上の空き領域に配置する。
また、計算機装置（１００）のブートプログラム（１４０）を、計算機装置（１００）の起動時にＲＡＳモジュール（１３０）の第１の初期化手順（１３２）を呼び出すよう変更する。
具体的には、ＲＡＳモジュール（１３０）の第１の初期化手順（１３２）のプログラムのメモリ（１０３）上のアドレスを、ブートプログラム（１４０）から呼び出すよう変更する。
以上がＲＡＳモジュール（１３０）を計算機装置（１００）に追加する方法の概要である。
以上のようにＲＡＳモジュール（１３０）を計算機装置（１００）に追加した後（図８の状態）に、計算機装置（１００）を起動した場合、前述した図９のフローチャートのような動作が行われる。
【００６１】
以上のような構成、動作により、ＯＳ（一つでも複数のＯＳが存在する構成でもよい）と、Ｈｙｐｅｒｖｉｓｏｒが独立したモジュールで構成されている計算機装置に、ＯＳやＨｙｐｅｒｖｉｓｏｒのモジュールへの修正なしに、ＯＳが所有する割込み判定部に対応したＲＡＳ機能を、容易に追加することが可能である。
【００６２】
［実施の形態２：動作の概要説明（割込み発生時の動作概要）］
次に、割込みが発生した場合の実施の形態２に係る計算機装置（１００）動作の概要を説明する。
図１１は、ＯＳ（１１０）の動作時に割込みが発生した場合の処理の流れを示す。
ここで、ＯＳは複数存在していてもよい。
図１２は、Ｈｙｐｅｒｖｉｓｏｒ（１２０）の動作時に割込みが発生した場合の処理の流れを示す。
図１１と図１２の違いはＲＡＳモジュール（１３０）の故障情報収集部（１３７）が、

ＯＳの故障情報を収集するか、Ｈｙｐｅｒｖｉｓｏｒの故障情報を収集するかの違いのみである。

また、図１１及び図１２では、説明に直接関係のないメモリ（１０３）、二次記憶装置（１０４）、ブートプログラム（１４０）の図示は省略している。

また、図１１及び図１２の実線の矢印は処理の流れを示しており、破線の矢印は処理の内容を示している。

【００６３】

割込みが発生した場合に、ＣＰＵ（１０１）の割込み検知機構（１０２）が、ＲＡＳモジュール（１３０）を呼び出す。

ＲＡＳモジュール（１３０）は、発生した割込みがＣＰＵ例外であるかどうかを判定する。

そして、発生した割込みがＣＰＵ例外であれば、ＲＡＳモジュール（１３０）は、ＣＰＵ例外が発生した時にどのＯＳまたはＨｙｐｅｒｖｉｓｏｒが動作していたかを、計算機装置（１００）のプログラムカウンタ（ＣＰＵの実行アドレスを保持しているレジスタ）により判定（どのＯＳまたはＨｙｐｅｒｖｉｓｏｒが動作中だったかを、プログラムカウンタが保持する実行中のコードの場所より判定）し、該当するＯＳまたはＨｙｐｅｒｖｉｓｏｒを特定し、特定したＯＳまたはＨｙｐｅｒｖｉｓｏｒから故障情報を収集する。

この動作により、ＣＰＵ例外発生時に、どのＯＳ（ＯＳは複数の構成でもよい）、またはＨｙｐｅｒｖｉｓｏｒが動作していたかを判定することが可能となり、該当のＯＳ、またはＨｙｐｅｒｖｉｓｏｒから障害情報を収集することができる。

【００６４】

［実施の形態２：動作の概要説明（割込み発生時のフローチャート）］

割込み発生時のフローチャートを図１３に示す。

【００６５】

このフローチャートと実施の形態１の図７のフローチャートとの違いは、Ｓ７０６－１の処理がＳ７０６－２の処理となっていることのみである。

このため、以下では、Ｓ７０６－２の処理のみを説明する。

【００６６】

ＲＡＳモジュール（１３０）の故障検知部（１３６）で該当の故障を特定（Ｓ７０５）した後、ＲＡＳモジュール（１３０）の故障情報収集部（１３７）が、ＣＰＵ例外発生時に動作していたＯＳ（またはＨｙｐｅｒｖｉｓｏｒ）を特定し、該当するＯＳ（またはＨｙｐｅｒｖｉｓｏｒ）から故障情報を収集する（Ｓ７０６－２）。

なお、ＲＡＳモジュール（１３０）の故障情報収集部（１３７）は、前述の通り計算機装置（１００）のプログラムカウンタにより、ＣＰＵ例外発生時に動作していたＯＳ（またはＨｙｐｅｒｖｉｓｏｒ）を特定する。

以降の処理（Ｓ７０７以降）は、実施の形態１の図７と同じである。

【００６７】

以上、実施の形態２によれば、ＣＰＵ例外発生時にＲＡＳモジュール（１３０）内の割込み判定部（１３４）、故障検知部（１３６）等が実行されることにより、ＣＰＵ例外とＯＳの障害の発生が重なった場合でも、ＲＡＳ機能を実行することができる。

また、ＣＰＵ例外発生時にＲＡＳモジュール（１３０）内の割込み判定部（１３４）、故障検知部（１３６）等が実行されることにより、ＣＰＵ例外とＨｙｐｅｒｖｉｓｏｒの障害の発生が重なった場合でも、ＲＡＳ機能を実行することができる。

【００６８】

以上、本実施の形態では、ＯＳ（一つでも複数のＯＳが存在する構成でも良い）と、Ｈｙｐｅｒｖｉｓｏｒが独立したモジュールで構成されている計算機装置に、ＯＳやＨｙｐｅｒｖｉｓｏｒのモジュールへの修正なしに、ＯＳが所有する割込み判定部に対応したＲＡＳ機能を追加することを可能とするＲＡＳ方式を説明した。

より具体的には、本実施の形態では以下の（１）～（４）を説明した。

（１）計算機装置の起動時に呼び出されるＲＡＳモジュールの「初期化手順」は、ＯＳ

やＨｙｐｅｒｖｉｓｏｒの「初期化手順」よりも前に実行される「第１の初期化手順」と、ＯＳやＨｙｐｅｒｖｉｓｏｒの「初期化手順」が完了した後に実行される「第２の初期化手順」の２つに分かれる。

（２）ＲＡＳモジュールの「第１の初期化手順」では、ＯＳの「初期化手順」の最後の部分が、ＯＳの「初期化手順」の後にＲＡＳモジュールの「第２の初期化手順」を呼び出すように書換えられる。

（３）ＲＡＳモジュールの「第２の初期化手順」では、ＯＳの「割込み判定部」がＲＡＳモジュールにコピーされる。

（４）ＲＡＳモジュールの「第２の初期化手順」では、割込み発生時にＯＳの「割込み判定部」ではなくＲＡＳモジュールの「割込み判定部」を呼び出すように、「割込み検知機構」が設定される。

【００６９】

また、本実施の形態では、ＣＰＵ例外発生時にＲＡＳモジュール内の割込み判定部、故障検知部等が実行されることにより、ＣＰＵ例外発生とＨｙｐｅｒｖｉｓｏｒの障害の発生が重なった場合でも、ＲＡＳ機能が実行されるＲＡＳ方式を説明した。

【００７０】

また、本実施の形態では、ＲＡＳモジュール内の故障情報収集部が実行されることにより、ＣＰＵ例外発生時に、どのＯＳ（ＯＳは複数の構成でもよい）、またはＨｙｐｅｒｖｉｓｏｒが動作していたかを、ＣＰＵ例外発生時のプログラムカウンタから判定し、該当のＯＳ、またはＨｙｐｅｒｖｉｓｏｒから障害情報を収集することが可能なＲＡＳ方式を説明した。

【００７１】

実施の形態３．

本実施の形態では、割込み検知機構（１０２）で検知した割込みが、ＣＰＵ例外ではなく標準割込みであった場合の動作の概要を説明する。

計算機装置（１００）の構成は、実施の形態１の構成例（図１）でも、実施の形態２の構成図（図８）でもよい。

ここでは実施の形態２の構成（Ｈｙｐｅｒｖｉｓｏｒがある構成）にて説明するが、実施の形態１の構成（Ｈｙｐｅｒｖｉｓｏｒがない構成）でも以下の動作が行われる。

【００７２】

［実施の形態３：動作の概要説明（割込み発生時の動作）］

図１４に割込みが発生した場合の処理の流れを示す。

図１４では、説明に直接関係のないメモリ（１０３）、二次記憶装置（１０４）、ブートプログラム（１４０）の図示は省略している。

また、図１４の実線の矢印は処理の流れを示しており、破線の矢印は処理の内容を示している。

【００７３】

図１４において、Ｈｙｐｅｒｖｉｓｏｒ（１２０）は割込みをマスクする処理をしており、割込みを受付けないものとする。

このため、Ｈｙｐｅｒｖｉｓｏｒ（１２０）の処理中に割込みが発生した場合には、Ｈｙｐｅｒｖｉｓｏｒ（１２０）からＯＳ（１１０）側に処理が移ったタイミングで、割込み検知機構（１０２）がＲＡＳモジュール（１３０）の割込み判定部（１３４）を呼び出す。

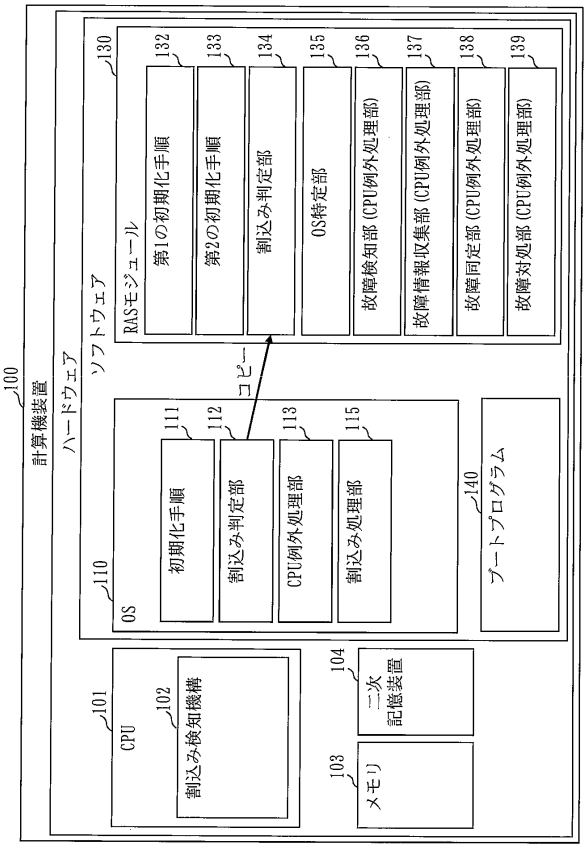
従って、Ｈｙｐｅｒｖｉｓｏｒ（１２０）からＯＳ（１１０）側に処理が戻った時が割込み発生のタイミングとなり、割込み発生時のプログラムカウンタはＯＳの処理実行中となる（プログラムカウンタがＨｙｐｅｒｖｉｓｏｒの処理中を指すことはない）。

実施の形態１（Ｈｙｐｅｒｖｉｓｏｒがない構成）では、Ｈｙｐｅｒｖｉｓｏｒ（１２０）自体がないため、このようなＨｙｐｅｒｖｉｓｏｒ（１２０）の割込みのマスク処理は存在しない。

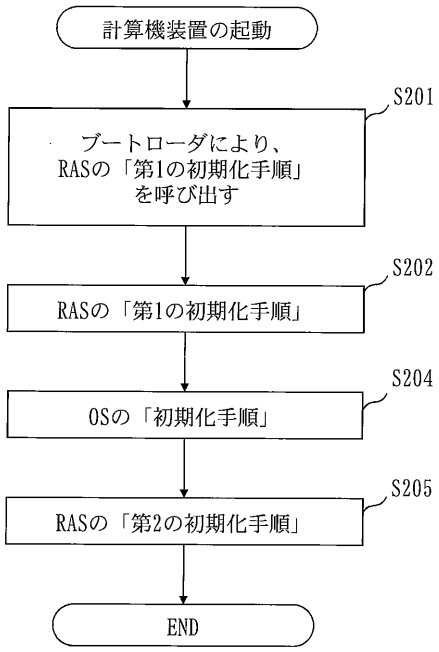
【００７４】

［実施の形態３：動作の概要説明（割込み発生時の動作）］
図１５は、図７又は図１３のＳ７０３でＮＯと判定された後の動作を示す。
つまり、図７又は図１３のＳ７０３で割込み判定部（１３４）により割込みがＣＰＵ例外ではなく標準割込みと判定された場合に、図１５のＳ１００１が行われる。
【００７５】
図１５において、まず、ＲＡＳモジュール（１３０）のＯＳ特定部（１３５）が、標準割込み発生時に動作していたＯＳ（１１０）を特定する（Ｓ１００１）。
ＯＳ特定部（１３５）は、実施の形態１及び実施の形態２で説明した故障情報収集部（１３７）と同様に、計算機装置（１００）のプログラムカウンタにより、標準割込み発生時に動作していたＯＳ（１１０）を特定する。
次に、ＯＳ特定部（１３５）は、該当するＯＳ（１１０）の割込み判定部（１１２）を参照して、該当するＯＳ（１１０）の割込み処理部（１１５）のプログラムのアドレスを特定して呼び出す（Ｓ１００２）。
次に、ＯＳ（１１０）の割込み処理部（１１５）が実行され、以降、ＯＳ（１１０）側に処理が戻る（Ｓ１００３）。
【００７６】
実施の形態１または実施の形態２により追加されたＲＡＳ機能において、実施の形態３で説明した動作を実施することにより、割込み検知機構で検知した割込みがＣＰＵ例外ではなく標準割込みであった場合に、正しくＯＳ側の割込み処理部を呼び出すことができる。
例えば、ＯＳ側で初期化時から割込み発生時までの間にユーザ操作などによって、割込み動作の登録が動的に追加・変更されていた場合でも、実施の形態３で説明した動作により、ＯＳに適正に割込みを処理させることができる。
【００７７】
以上、本実施の形態では、ＣＰＵ例外ではなく標準割込みが発生した場合に、ＯＳ側の割込み判定部を参照し、対応するＯＳの割込み処理部を呼び出すことにより、正しくＯＳ側の割込み処理部を呼び出すことができるＲＡＳ方式を説明した。
【００７８】
以上、本発明の実施の形態について説明したが、これらの実施の形態のうち、２つ以上を組み合わせて実施しても構わない。
あるいは、これらの実施の形態のうち、１つを部分的に実施しても構わない。
あるいは、これらの実施の形態のうち、２つ以上を部分的に組み合わせて実施しても構わない。
なお、本発明は、これらの実施の形態に限定されるものではなく、必要に応じて種々の変更が可能である。
【符号の説明】
【００７９】
１００　計算機装置、１０１　ＣＰＵ、１０２　割込み検知機構、１０３　メモリ、１０４　二次記憶装置、１１０　ＯＳ、１１１　初期化手順、１１２　割込み判定部、１１３　ＣＰＵ例外処理部、１１５　割込み処理部、１２０　Ｈｙｐｅｒｖｉｓｏｒ、１２１　初期化手順、１３０　ＲＡＳモジュール、１３２　第１の初期化手順、１３３　第２の初期化手順、１３４　割込み判定部、１３５　ＯＳ特定部、１３６　故障検知部、１３７　故障情報収集部、１３８　故障同定部、１３９　故障対処部、１４０　ブートプログラム。

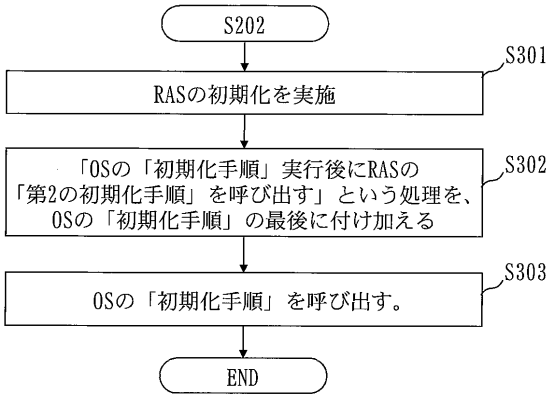
【図１】


100
計算機装置
ハードウェア
ソフトウェア
130
RASモジュール
132
第1の初期化手順
133
第2の初期化手順
134
割込み判定部
135
OS特定部
136
故障検知部(CPU例外処理部)
137
故障情報収集部(CPU例外処理部)
138
故障同定部(CPU例外処理部)
139
故障対処部(CPU例外処理部)
コピー
110
OS
111
初期化手順
112
割込み判定部
113
CPU例外処理部
115
割込み処理部
140
ブートプログラム
101
CPU
102
割込み検知機構
104
二次記憶装置
103
メモリ

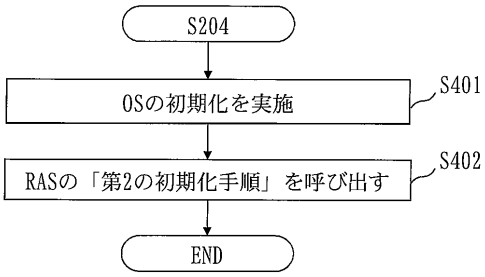
【図２】


計算機装置の起動
S201
ブートローダにより、
RASの「第1の初期化手順」
を呼び出す
S202
RASの「第1の初期化手順」
S204
OSの「初期化手順」
S205
RASの「第2の初期化手順」
END

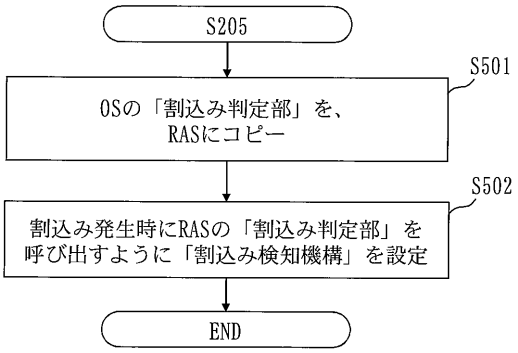
【図３】


S202
S301
RASの初期化を実施
「OSの「初期化手順」実行後にRASの
「第2の初期化手順」を呼び出す」という処理を、
OSの「初期化手順」の最後に付け加える
S302
S303
OSの「初期化手順」を呼び出す。
END

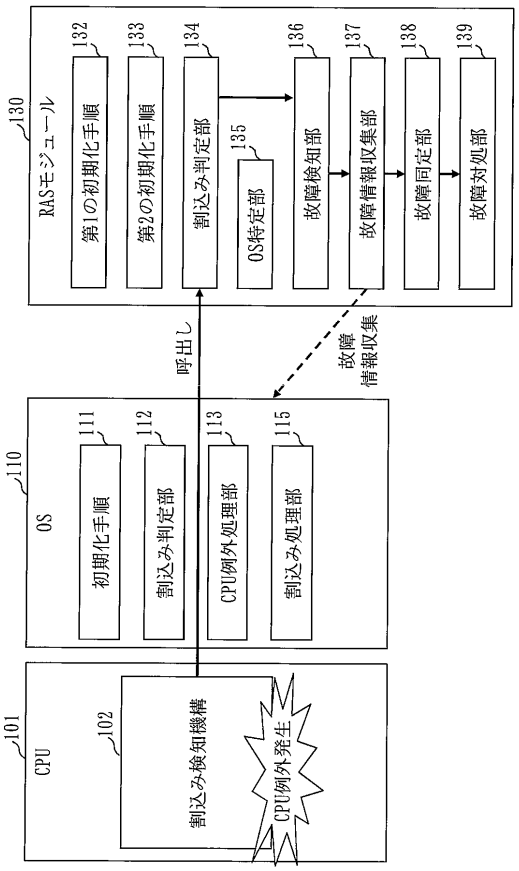
【図４】


S204
S401
OSの初期化を実施
S402
RASの「第2の初期化手順」を呼び出す
END

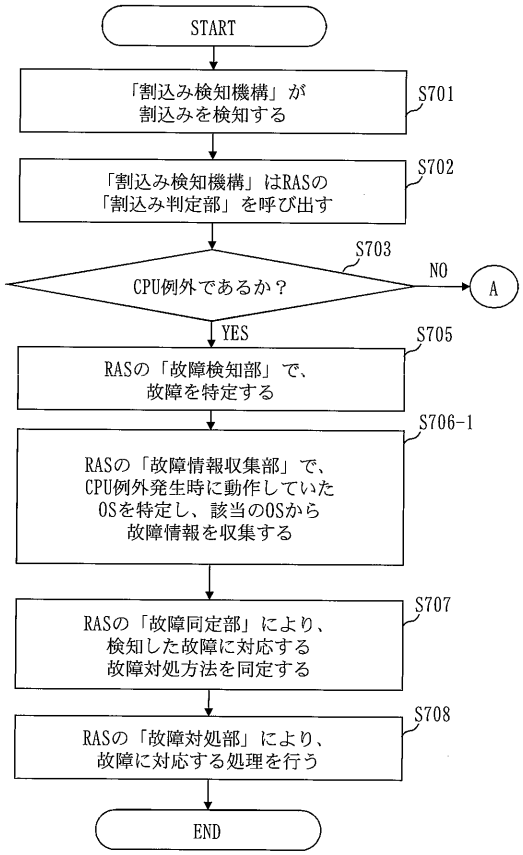
【図５】


S205
S501
OSの「割込み判定部」を、
RASにコピー
S502
割込み発生時にRASの「割込み判定部」を
呼び出すように「割込み検知機構」を設定
END

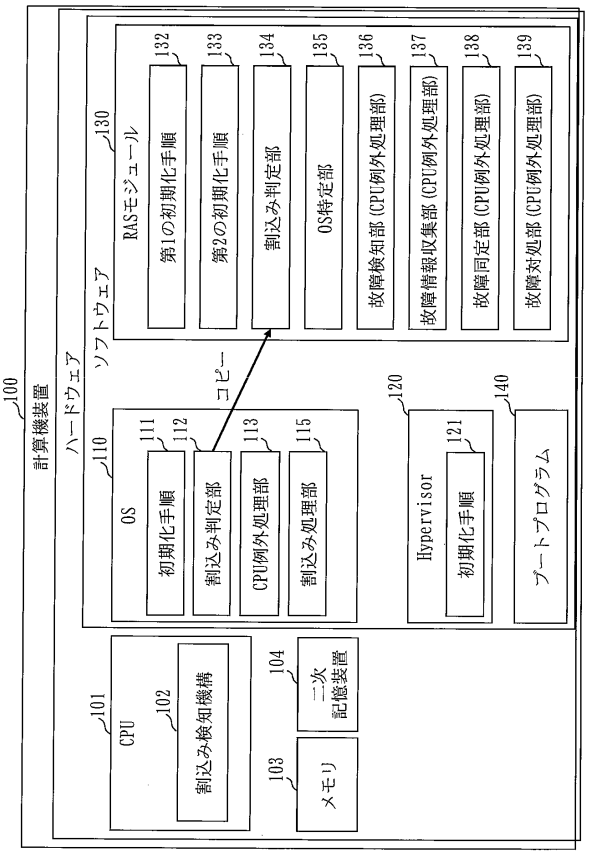
【図６】


RASモジュール
130
第1の初期化手順
132
第2の初期化手順
133
割込み判定部
134
OS特定部
135
故障検知部
136
故障情報収集部
137
故障同定部
138
故障対処部
139
呼出し
故障情報収集
OS
110
初期化手順
111
割込み判定部
112
CPU例外処理部
113
割込み処理部
115
CPU
101
割込み検知機構
102
CPU例外発生

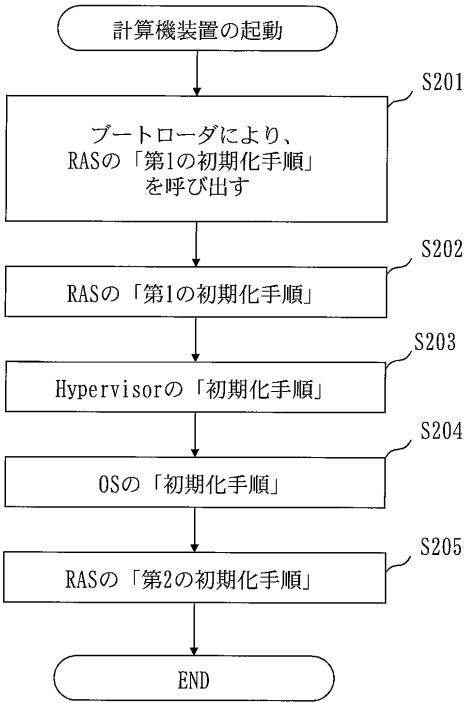
【図７】


START
「割込み検知機構」が
割込みを検知する
S701
「割込み検知機構」はRASの
「割込み判定部」を呼び出す
S702
CPU例外であるか？
S703
NO
A
YES
RASの「故障検知部」で、
故障を特定する
S705
RASの「故障情報収集部」で、
CPU例外発生時に動作していた
OSを特定し、該当のOSから
故障情報を収集する
S706-1
RASの「故障同定部」により、
検知した故障に対応する
故障対処方法を同定する
S707
RASの「故障対処部」により、
故障に対応する処理を行う
S708
END

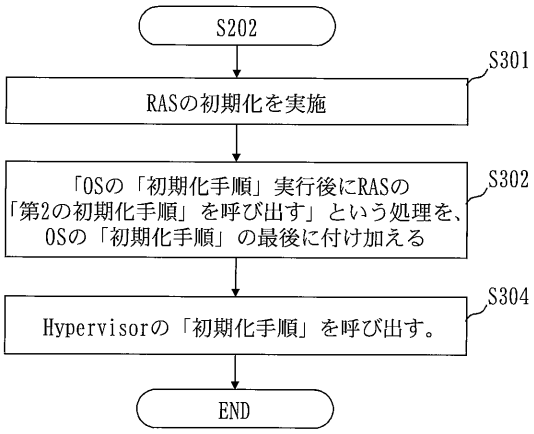
【図８】


計算機装置
100
ハードウェア
ソフトウェア
RASモジュール
130
第1の初期化手順
132
第2の初期化手順
133
割込み判定部
134
OS特定部
135
故障検知部(CPU例外処理部)
136
故障情報収集部(CPU例外処理部)
137
故障同定部(CPU例外処理部)
138
故障対処部(CPU例外処理部)
139
コピー
OS
110
初期化手順
111
割込み判定部
112
CPU例外処理部
113
割込み処理部
115
Hypervisor
120
初期化手順
121
ブートプログラム
140
CPU
101
割込み検知機構
102
二次
記憶装置
104
メモリ
103

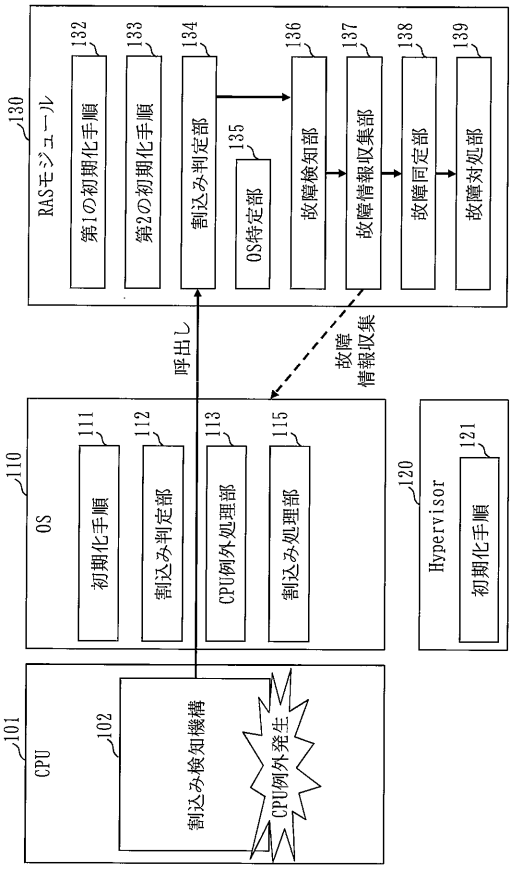
【図９】


計算機装置の起動
ブートローダにより、
RASの「第1の初期化手順」
を呼び出す
S201
RASの「第1の初期化手順」
S202
Hypervisorの「初期化手順」
S203
OSの「初期化手順」
S204
RASの「第2の初期化手順」
S205
END

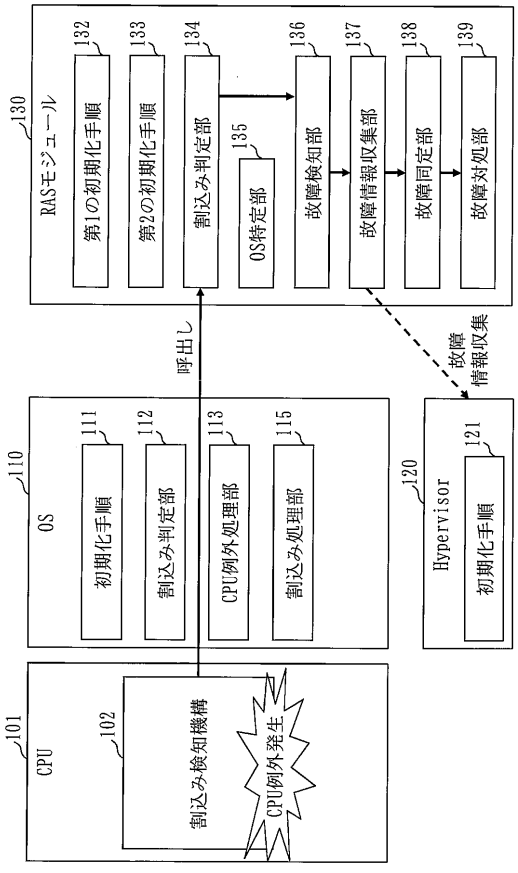
【図１０】



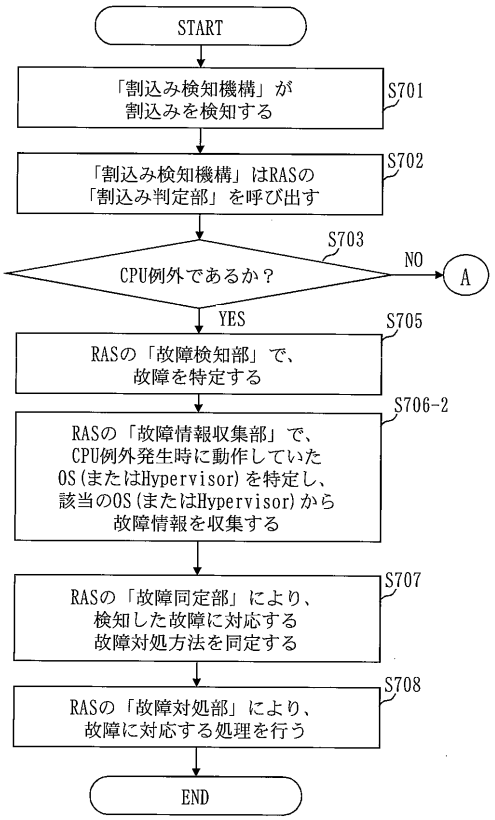
【図１１】



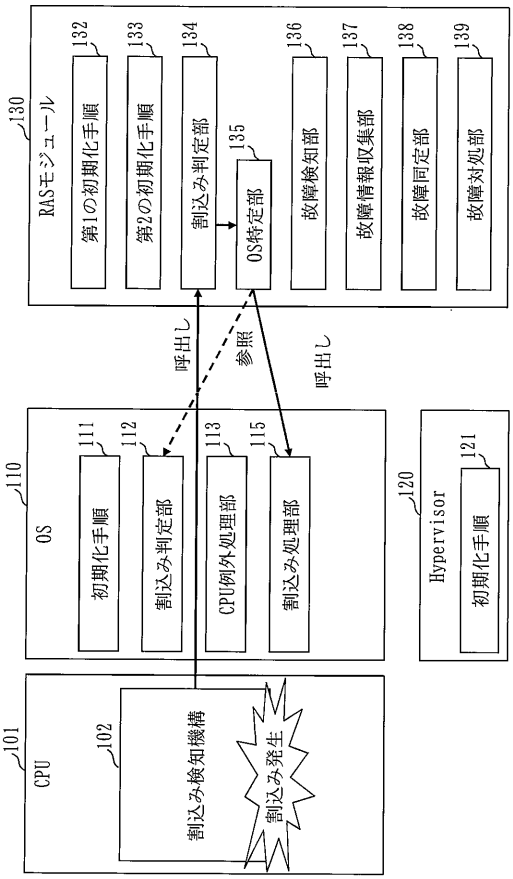
【図１２】



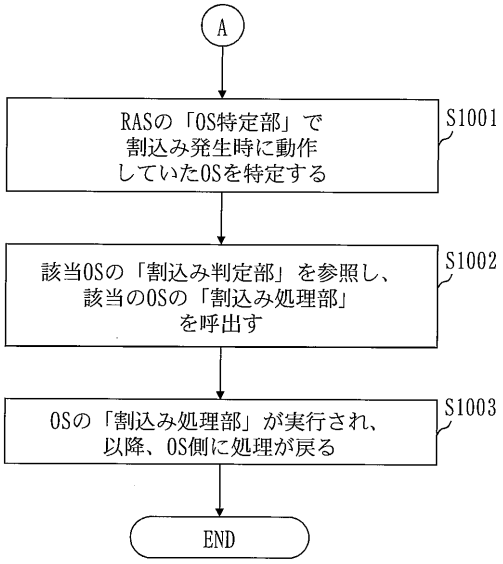
【図１３】



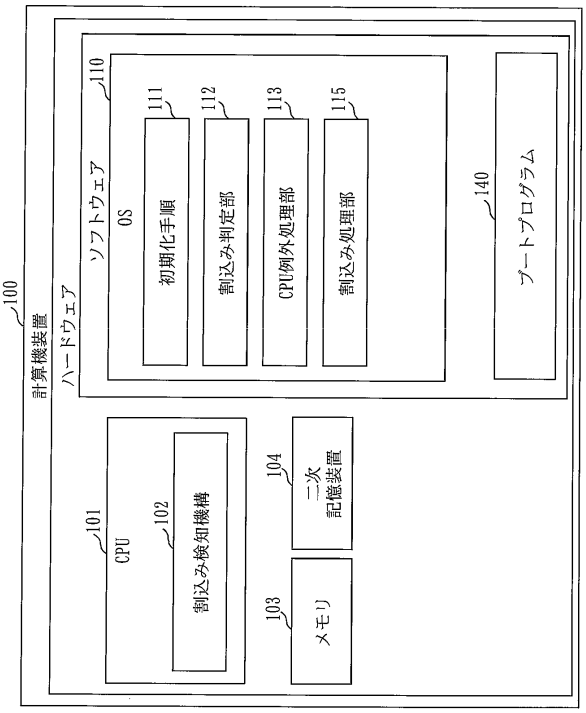
【図１４】



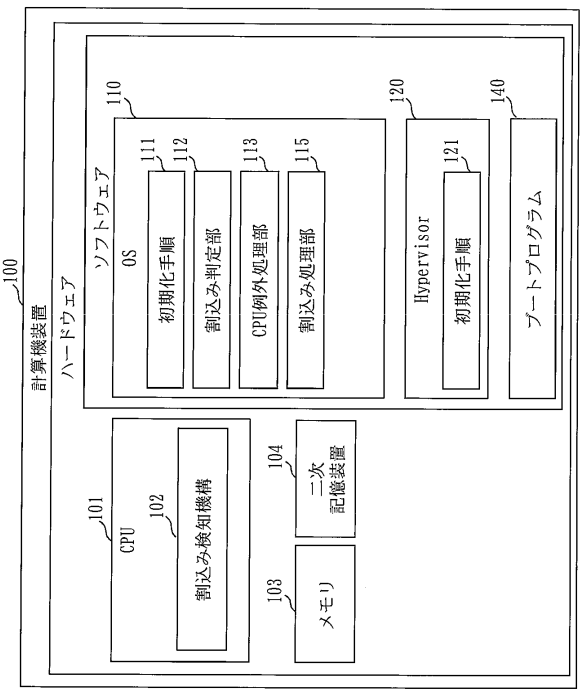
【図１５】



【図１６】



【図１７】



フロントページの続き

(56)参考文献 特開平０８－０１６４２０（ＪＰ，Ａ）
特開２００４－０１３２４０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １１／３０
Ｇ０６Ｆ ９／４４５
Ｇ０６Ｆ ９／４８