

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5819350号
(P5819350)

(45) 発行日 平成27年11月24日 (2015.11.24)

(24) 登録日 平成27年10月9日 (2015.10.9)

(51) Int.Cl.

F I

G 0 6 F 9 / 4 4 5 (2006.01)

G 0 6 F 9 / 0 6 6 1 0 J

請求項の数 8 (全 24 頁)

(21) 出願番号	特願2013-124391 (P2013-124391)	(73) 特許権者	000005108
(22) 出願日	平成25年6月13日 (2013.6.13)		株式会社日立製作所
(65) 公開番号	特開2015-1757 (P2015-1757A)		東京都千代田区丸の内一丁目6番6号
(43) 公開日	平成27年1月5日 (2015.1.5)	(74) 代理人	110001678
審査請求日	平成27年5月7日 (2015.5.7)		特許業務法人藤央特許事務所
		(72) 発明者	早川 典充
			神奈川県秦野市堀山下1番地 株式会社日立製作所 ITプラットフォーム事業本部内
		(72) 発明者	吉田 圭佑
			神奈川県横浜市西区みなとみらい二丁目3番3号 株式会社日立情報通信エンジニアリング内

最終頁に続く

(54) 【発明の名称】 計算機システム及び起動方法

(57) 【特許請求の範囲】

【請求項 1】

複数のオペレーティングシステムが稼働する計算機を複数備える計算機システムであって、

前記複数の計算機の各々は、プロセッサ、前記プロセッサに接続される第1のメモリ、前記プロセッサに接続される第2のメモリ、前記プロセッサに接続されるI/Oデバイス、及び前記プロセッサに接続される記憶装置を有し、

前記複数のオペレーティングシステムは、第1のオペレーティングシステム及び第2のオペレーティングシステムを含み、

前記第2のメモリは、前記第1のオペレーティングシステムの起動時に、前記計算機が備えるハードウェアを制御し、前記第1のオペレーティングシステムの起動処理を実行する第1のハードウェア制御部を格納し、

前記記憶装置は、

前記第1のオペレーティングシステムの実行イメージである第1のオペレーティングシステムイメージと、

前記第1のオペレーティングシステムを起動する第1のオペレーティングシステムブートロードと、

前記第2のオペレーティングシステムの実行イメージである第2のオペレーティングシステムイメージと、

前記第2のオペレーティングシステムを起動する第2のオペレーティングシステムブー

10

20

トローダと、

前記第2のオペレーティングシステムの起動時に、前記計算機が備えるハードウェアを制御し、前記第2のオペレーティングシステムの起動処理を実行する第2のハードウェア制御部と、

前記第2のハードウェア制御部が参照する情報が格納される前記第1のメモリにおける記憶領域の絶対アドレスであるアドレスデータを書き換えるアドレス書換部と、を格納し、

前記第2のハードウェア制御部は、当該第2のハードウェア制御部を起動するための起動部を含み、

前記プロセッサは、電源が投入された場合、前記第2のメモリに格納される前記第1のハードウェア制御部を前記第1のメモリに書き込み、前記第1のハードウェア制御部を実行し、

前記第1のハードウェア制御部は、

前記記憶装置から前記第2のハードウェア制御部、及び前記アドレス書換部を読み出して、前記第1のメモリの所定の記憶領域に書き込み、

前記第2のハードウェア制御部に含まれる前記起動部が格納される記憶領域の先頭アドレスを取得して、ベースアドレスデータとして前記第1のメモリに書き込み、

前記記憶装置から前記第1のオペレーティングシステムブートローダを読み出して、前記第1のメモリに書き込み、

前記第1のメモリに書き込まれた第1のオペレーティングシステムブートローダを起動し、

前記第1のオペレーティングシステムブートローダは、

前記記憶装置から前記第1のオペレーティングシステムイメージを読み出して、前記第1のメモリに書き込み、

前記第1のメモリに書き込まれた第1のオペレーティングシステムイメージを実行することによって前記第1のオペレーティングシステムを起動し、

前記第1のオペレーティングシステムは、前記アドレス書換部を起動し、

前記アドレス書換部は、

前記第1のメモリに書き込まれたベースアドレスデータを取得し、

書換対象のアドレスデータが格納される記憶領域のアドレスである対象アドレスを取得し、

前記取得されたベースアドレスデータに基づいて、前記取得された対象アドレスに対応する記憶領域に格納される前記アドレスデータを書き換え、

前記起動部を起動し、

前記起動部は、前記書き換えられたアドレスデータを用いて、前記第2のハードウェア制御部を起動し、

前記第2のハードウェア制御部は、

前記記憶装置から前記第2のオペレーティングシステムブートローダを読み出して、前記第1のメモリに書き込み、

前記第1のメモリに書き込まれた第2のオペレーティングシステムブートローダを起動し、

前記第2のオペレーティングシステムブートローダは、

前記記憶装置から前記第2のオペレーティングシステムイメージを読み出して、前記第1のメモリに書き込み、

前記第1のメモリに書き込まれた第2のオペレーティングシステムイメージを実行することによって前記第2のオペレーティングシステムを起動することを特徴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であって、

前記アドレス書換部は、

10

20

30

40

50

前記第 2 のハードウェア制御部から前記対象アドレスに関する情報を取得し、
前記取得された対象アドレスに対応する記憶領域に格納されるアドレスデータに前記ベースアドレスデータを加算することによって、当該アドレスデータを書き換えることを特徴とする計算機システム。

【請求項 3】

請求項 2 に記載の計算機システムであって、
前記アドレス書換部は、
前記第 2 のハードウェア制御部の実行形式のファイルを参照することによって前記対象アドレスに関する情報を取得することの特徴とする計算機システム。

【請求項 4】

請求項 3 に記載の計算機システムであって、
前記記憶装置は、第 1 の記憶装置及び第 2 の記憶装置を含み、
前記第 1 の記憶装置は、前記第 1 のオペレーティングシステムイメージ、前記第 1 のオペレーティングシステムブートローダ、前記第 2 のハードウェア制御部、及び前記アドレス書換部を格納し、
前記第 2 の記憶装置は、前記第 2 のオペレーティングシステムイメージ、及び前記第 2 のオペレーティングシステムブートローダを格納することの特徴とする計算機システム。

【請求項 5】

複数のオペレーティングシステムが稼働する計算機における起動方法であって、
前記計算機は、プロセッサ、前記プロセッサに接続される第 1 のメモリ、前記プロセッサに接続される第 2 のメモリ、前記プロセッサに接続される I/O デバイス、及び前記プロセッサに接続される記憶装置を備え、
前記複数のオペレーティングシステムは、第 1 のオペレーティングシステム及び第 2 のオペレーティングシステムを含み、
前記第 2 のメモリは、前記第 1 のオペレーティングシステムの起動時に、前記計算機が備えるハードウェアを制御し、前記第 1 のオペレーティングシステムの起動処理を実行する第 1 のハードウェア制御部を格納し、
前記記憶装置は、
前記第 1 のオペレーティングシステムの実行イメージである第 1 のオペレーティングシステムイメージと、
前記第 1 のオペレーティングシステムを起動する第 1 のオペレーティングシステムブートローダと、
前記第 2 のオペレーティングシステムの実行イメージである第 2 のオペレーティングシステムイメージと、
前記第 2 のオペレーティングシステムを起動する第 2 のオペレーティングシステムブートローダと、
前記第 2 のオペレーティングシステムの起動時に、前記計算機が備えるハードウェアを制御し、前記第 2 のオペレーティングシステムの起動処理を実行する第 2 のハードウェア制御部と、
前記第 2 のハードウェア制御部が参照する情報が格納される前記第 1 のメモリにおける記憶領域の絶対アドレスであるアドレスデータを書き換えるアドレス書換部と、を格納し、

前記第 2 のハードウェア制御部は、当該第 2 のハードウェア制御部を起動するための起動部を含み、

前記方法は、

前記プロセッサが、電源が投入された場合、前記第 2 のメモリに格納される前記第 1 のハードウェア制御部を前記第 1 のメモリに書き込み、前記第 1 のハードウェア制御部を実行する第 1 のステップと、

前記プロセッサによって実行される前記第 1 のハードウェア制御部が、前記記憶装置から前記第 2 のハードウェア制御部、及び前記アドレス書換部を読み出して、前記第 1 のメ

10

20

30

40

50

モリの所定の記憶領域に書き込む第2のステップと、

前記プロセッサによって実行される第1のハードウェア制御部が、前記第2のハードウェア制御部に含まれる前記起動部が格納される記憶領域の先頭アドレスを取得して、ベースアドレスデータとして前記第1のメモリに書き込む第3のステップと、

前記プロセッサによって実行される前記第1のハードウェア制御部が、前記記憶装置から前記第1のオペレーティングシステムブートローダを読み出して、前記第1のメモリに書き込む第4のステップと、

前記プロセッサによって実行される前記第1のハードウェア制御部が、前記第1のメモリに書き込まれた第1のオペレーティングシステムブートローダを起動する第5のステップと、

10

前記プロセッサによって実行される前記第1のオペレーティングシステムブートローダが、前記記憶装置から前記第1のオペレーティングシステムイメージを読み出して、前記第1のメモリに書き込む第6のステップと、

前記プロセッサによって実行される前記第1のオペレーティングシステムブートローダが、前記第1のメモリに書き込まれた第1のオペレーティングシステムイメージを実行することによって前記第1のオペレーティングシステムを起動する第7のステップと、

前記プロセッサによって実行される前記第1のオペレーティングシステムが、前記アドレス書換部を起動する第8のステップと、

前記プロセッサによって実行される前記アドレス書換部が、前記第1のメモリに書き込まれたベースアドレスデータを取得する第9のステップと、

20

前記プロセッサによって実行されるアドレス書換部が、書換対象のアドレスデータが格納される記憶領域のアドレスである対象アドレスを取得する第10のステップと、

前記プロセッサによって実行されるアドレス書換部が、前記取得されたベースアドレスデータに基づいて、前記取得された対象アドレスに対応する記憶領域に格納される前記アドレスデータを書き換える第11のステップと、

前記プロセッサによって実行されるアドレス書換部が、前記起動部を起動する第12のステップと、

前記プロセッサによって実行される前記起動部は、前記書き換えられたアドレスデータを用いて、前記第2のハードウェア制御部を起動する第13のステップと、

前記プロセッサによって実行される前記第2のハードウェア制御部が、前記記憶装置から前記第2のオペレーティングシステムブートローダを読み出して、前記第1のメモリに書き込む第14のステップと、

30

前記プロセッサによって実行される前記第2のハードウェア制御部が、前記第1のメモリに書き込まれた第2のオペレーティングシステムブートローダを起動する第15のステップと、

前記プロセッサによって実行される前記第2のオペレーティングシステムブートローダが、前記記憶装置から前記第2のオペレーティングシステムイメージを読み出して、前記第1のメモリに書き込む第16のステップと、

前記プロセッサによって実行される前記第2のオペレーティングシステムブートローダが、前記第1のメモリに書き込まれた第2のオペレーティングシステムイメージを実行することによって前記第2のオペレーティングシステムを起動する第17のステップと、を含むことを特徴する起動方法。

40

【請求項6】

請求項5に記載の起動方法であって、

前記第10のステップでは、前記第2のハードウェア制御部から前記対象アドレスに関する情報を取得し、

前記第11のステップでは、前記取得された対象アドレスに対応する記憶領域に格納されるアドレスデータに前記ベースアドレスデータを加算することによって、当該アドレスデータを書き換えることを特徴とする起動方法。

【請求項7】

50

請求項 6 に記載の起動方法であって、

前記第 10 のステップでは、前記第 2 のハードウェア制御部の実行形式のファイルを参照することによって前記対象アドレスに関する情報を取得することを特徴とする起動方法。

【請求項 8】

請求項 7 に記載の起動方法であって、

前記記憶装置は、第 1 の記憶装置及び第 2 の記憶装置を含み、

前記第 1 の記憶装置は、前記第 1 のオペレーティングシステムイメージ、前記第 1 のオペレーティングシステムブートローダ、前記第 2 のハードウェア制御部、及び前記アドレス書換部を格納し、

前記第 2 の記憶装置は、前記第 2 のオペレーティングシステムイメージ、及び前記第 2 のオペレーティングシステムブートローダを格納することを特徴とする起動方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一つの計算機上に複数の OS が稼働する計算機システム、及び計算機の起動方法に関する。

【背景技術】

【0002】

近年、物理計算機上のハードウェアリソースを有効に活用するために、一台の物理計算機上で二つ以上の OS（オペレーティングシステム）を起動する技術が注目されている（例えば、特許文献 1 及び特許文献 2 参照）。

【0003】

特許文献 1 には、「マルチオペレーティングシステム計算機は、優先度が設定された複数のタスクを優先度順に実行する第 1 の OS（オペレーティングシステム）と、前記第 1 の OS と異なる第 2 の OS とを動作 OS として交互に動作させるマルチオペレーティングシステム計算機であって、前記第 1 の OS が動作 OS として動作しているときに前記複数のタスクのうち所定の優先度が設定された所定のタスクであって動作 OS の切替え契機を特定するための切替え契機タスクとして用いられるタスクが実行された場合に、動作 OS を前記第 1 の OS から前記第 2 の OS に切り替える OS 切替え部」を備えることが記載されている。

【0004】

また、特許文献 2 には、「第 1 の OS と第 2 の OS との少なくとも 2 つの OS を起動するマルチ OS 起動装置において、（1）メモリ領域を有し、前記メモリ領域に対してメモリ空間が定義される 1 次記憶部と、（2）セカンドブートローダと、前記第 2 の OS とを記憶する 2 次記憶部と、（3）CPU に対する制御情報を示すコンテキストであって前記第 1 の OS 用のコンテキストである第 1 のコンテキストで動作する前記第 1 の OS のもとで動作するファーストブートローダを前記第 1 のコンテキストで動作中の前記第 1 の OS のもとで動作させることにより、前記第 1 の OS が管理する第 1 のメモリ空間として前記第 1 のコンテキストにより前記 1 次記憶装置に対して定義された前記 1 次記憶部のメモリ領域に前記 2 次記憶部から前記セカンドブートローダと前記第 2 の OS とを前記ファーストブートローダにロードさせる OS 実行部と、（4）前記第 1 のメモリ空間として定義されたメモリ領域にロードされた前記セカンドブートローダを前記第 1 のコンテキストで動作中の前記第 1 の OS のもとで実行することにより前記第 2 の OS が管理する第 2 のメモリ空間として定義されるメモリ領域と前記セカンドブートローダ及び前記第 2 の OS がロードされたメモリ領域とを含む第 3 のメモリ空間を前記 1 次記憶部に対して定義する前記セカンドブートローダ用のコンテキストを前記セカンドブートローダに生成させると共に生成された前記セカンドブートローダ用のコンテキストへ前記第 1 のコンテキストから切り替えさせ、前記セカンドブートローダ用のコンテキストのもとで前記セカンドブートローダを実行することにより前記第 3 のメモリ空間に含まれる前記第 2 のメモリ空間として

10

20

30

40

50

定義された前記１次記憶部のメモリ領域に前記ファーストブートローダによって前記１次記憶部のメモリ領域にロードされた前記第２のＯＳを前記セカンドブートローダにロードさせると共に前記第２のＯＳ用のコンテキストを生成させ、生成された前記第２のＯＳ用のコンテキストへ前記セカンドブートローダ用のコンテキストから切り替えさせ、前記第２のＯＳ用のコンテキストのもとで前記セカンドブートローダに前記第２のＯＳの起動を実行させるロード実行部と」を備えることが記載されている。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開平１１－１４９３８５号公報

10

【特許文献２】国際公開第２００９／１１３３９４号

【発明の概要】

【発明が解決しようとする課題】

【０００６】

特許文献１及び特許文献２では、ＯＳ１のデバイスドライバが、外部記憶装置からＯＳ２のブートローダ及びイメージファイルを読み出し、ＯＳ２用のメモリ領域に書き込むことによって、ＯＳ２を起動させることを特徴とする。

【０００７】

しかし、セキュリティ上の理由で、ＯＳ１が、ＯＳ２のブートローダ及びイメージファイルが格納される記憶領域に対するアクセスが許可されていない場合がある。また、ＯＳ１のイメージファイル及びＯＳ２のイメージファイルが異なる外部記憶装置に格納されている場合、ＯＳ１のカーネルが古い等の理由によって、ＯＳ１が、ＯＳ２のイメージファイルが格納される外部記憶装置のデバイスドライバを使用できない場合がある。前述のように、セキュリティ及びハードウェア構成の制限によって、ＯＳ２を起動することができない場合が存在する。

20

【０００８】

本発明は、ＯＳ１が直接関与することなく、通常の起動処理と同様の手順に従ってＯＳ２を起動するためのシステム及び起動方法を提供することを目的とする。

【課題を解決するための手段】

【０００９】

30

本願において開示される発明の代表的な一例を示せば以下の通りである。すなわち、複数のオペレーティングシステムが稼働する計算機を複数備える計算機システムであって、前記複数の計算機の各々は、プロセッサ、前記プロセッサに接続される第１のメモリ、前記プロセッサに接続される第２のメモリ、前記プロセッサに接続されるＩ／Ｏデバイス、及び前記プロセッサに接続される記憶装置を有し、前記複数のオペレーティングシステムは、第１のオペレーティングシステム及び第２のオペレーティングシステムを含み、前記第２のメモリは、前記第１のオペレーティングシステムの起動時に、前記計算機が備えるハードウェアを制御し、前記第１のオペレーティングシステムの起動処理を実行する第１のハードウェア制御部を格納し、前記記憶装置は、前記第１のオペレーティングシステムの実行イメージである第１のオペレーティングシステムイメージと、前記第１のオペレーティングシステムを起動する第１のオペレーティングシステムブートローダと、前記第２のオペレーティングシステムの実行イメージである第２のオペレーティングシステムイメージと、前記第２のオペレーティングシステムを起動する第２のオペレーティングシステムブートローダと、前記第２のオペレーティングシステムの起動時に、前記計算機が備えるハードウェアを制御し、前記第２のオペレーティングシステムの起動処理を実行する第２のハードウェア制御部と、前記第２のハードウェア制御部が参照する情報が格納される前記第１のメモリにおける記憶領域の絶対アドレスであるアドレスデータを書き換えるアドレス書換部と、を格納し、前記第２のハードウェア制御部は、当該第２のハードウェア制御部を起動するための起動部を含み、前記プロセッサは、電源が投入された場合、前記第２のメモリに格納される前記第１のハードウェア制御部を前記第１のメモリに書き込み

40

50

、前記第 1 のハードウェア制御部を実行し、前記第 1 のハードウェア制御部は、前記記憶装置から前記第 2 のハードウェア制御部、及び前記アドレス書換部を読み出して、前記第 1 のメモリの所定の記憶領域に書き込み、前記第 2 のハードウェア制御部に含まれる前記起動部が格納される記憶領域の先頭アドレスを取得して、ベースアドレスデータとして前記第 1 のメモリに書き込み、前記記憶装置から前記第 1 のオペレーティングシステムブートローダを読み出して、前記第 1 のメモリに書き込み、前記第 1 のメモリに書き込まれた第 1 のオペレーティングシステムブートローダを起動し、前記第 1 のオペレーティングシステムブートローダは、前記記憶装置から前記第 1 のオペレーティングシステムイメージを読み出して、前記第 1 のメモリに書き込み、前記第 1 のメモリに書き込まれた第 1 のオペレーティングシステムイメージを実行することによって前記第 1 のオペレーティングシステムを起動し、前記第 1 のオペレーティングシステムは、前記アドレス書換部を起動し、前記アドレス書換部は、前記第 1 のメモリに書き込まれたベースアドレスデータを取得し、書換対象のアドレスデータが格納される記憶領域のアドレスである対象アドレスを取得し、前記取得されたベースアドレスデータに基づいて、前記取得された対象アドレスに対応する記憶領域に格納される前記アドレスデータを書き換え、前記起動部を起動し、前記起動部は、前記書き換えられたアドレスデータを用いて、前記第 2 のハードウェア制御部を起動し、前記第 2 のハードウェア制御部は、前記記憶装置から前記第 2 のオペレーティングシステムブートローダを読み出して、前記第 1 のメモリに書き込み、前記第 1 のメモリに書き込まれた第 2 のオペレーティングシステムブートローダを起動し、前記第 2 のオペレーティングシステムブートローダは、前記記憶装置から前記第 2 のオペレーティングシステムイメージを読み出して、前記第 1 のメモリに書き込み、前記第 1 のメモリに書き込まれた第 2 のオペレーティングシステムイメージを実行することによって前記第 2 のオペレーティングシステムを起動することを特徴する。

10

20

【発明の効果】

【 0 0 1 0 】

本発明によれば、第 1 の OS が直接関与することなく第 2 の OS を起動することができる。また、アドレス書換部がアドレスデータを書き換えることによって、第 2 のハードウェア制御部がどの記憶領域に書き込まれても正常に動作することができる。

【 0 0 1 1 】

前述した以外の課題、構成及び効果は、以下の実施形態の説明によって明らかにされる。

30

【図面の簡単な説明】

【 0 0 1 2 】

【図 1】本発明の実施例 1 における計算機システムの構成の一例を示す説明図である。

【図 2】本発明の実施例 1 における第 2 の HW 制御プログラムの構成例を示す説明図である。

【図 3】本発明の実施例 1 における第 1 の HW 制御プログラムの構成例を示す説明図である。

【図 4】本発明の実施例 1 におけるアドレスデータ書換モジュールの構成例を示す説明図である。

40

【図 5】本発明の実施例 1 におけるリソース範囲定義情報の一例を示す説明図である。

【図 6】本発明の実施例 1 の書換対象アドレス情報の一例を示す説明図である。

【図 7 A】本発明の実施例 1 における計算機システムの起動処理を説明するシーケンス図である。

【図 7 B】本発明の実施例 1 における計算機システムの起動処理を説明するシーケンス図である。

【図 7 C】本発明の実施例 1 における計算機システムの起動処理を説明するシーケンス図である。

【図 7 D】本発明の実施例 1 における計算機システムの起動処理を説明するシーケンス図である。

50

【図 8】本発明の実施例 1 における第 2 の制御部用ローダのアドレスデータの変化の状態を示す説明図である。

【発明を実施するための形態】

【0013】

以下、図面を用いて実施例について説明する。

【実施例 1】

【0014】

図 1 は、本発明の実施例 1 における計算機システムの構成の一例を示す説明図である。

【0015】

本実施例の計算機システムは、物理計算機 10、第 1 の外部記憶装置 20、及び第 2 の外部記憶装置 30 から構成される。物理計算機 10 は、第 1 の外部記憶装置 20 及び第 2 の外部記憶装置 30 と直接、又は、ネットワークを介して接続される。ネットワークとしては、WAN、及びLAN等が考えられる。なお、物理計算機 10 は、装置内部に記憶装置を備えていてもよい。

10

【0016】

物理計算機 10 は、所定の処理を実行する OS が稼働する計算機である。本実施例では、第 1 の OS 270 及び第 2 の OS 280 の二つの OS が稼働するものとする。なお、第 1 の OS 270 及び第 2 の OS 280 は、一般的な OS 及び仮想化機能を提供するハイパバイザのいずれを適用してよい。以下の説明では、第 1 の OS 270 を一般的な OS、第 2 の OS 280 をハイパバイザであるものとする。

20

【0017】

また、本実施例では、第 1 の OS 270 及び第 2 の OS 280 を並列に稼働させる。具体的には、第 1 の OS 270 の起動後、第 2 の OS 280 を起動させる起動方式を採用する。また、第 2 の OS 280 は、一般的な起動手順に沿って起動するようにする。

【0018】

これによって、第 1 の OS 270 によって第 2 の OS 280 の OS イメージが扱われないため、セキュリティ面の課題を解決することができる。また、第 1 の OS 270 のデバイスドライバに依存することなく第 2 の OS 280 を起動することができる。

【0019】

物理計算機 10 は、ハードウェアリソース 100 として複数のプロセッサ、複数のメモリ、複数の I/O デバイス、及び ROM 108 を備える。

30

【0020】

プロセッサは、メモリに格納されるプログラムを実行する。プロセッサがプログラムを実行することによって、OS 等の機能が実現される。以下、プログラムを主体に処理を説明する場合、当該プログラムがプロセッサによって実行されていることを示す。

【0021】

メモリは、プロセッサによって実行されるプログラム及び当該プログラムの実行に必要な情報を格納する。また、メモリは、各プログラムのワークエリアを含む。

【0022】

I/O デバイスは、外部装置と接続し、外部から情報を入力し、外部に情報出力するためのデバイスである。例えば、I/O デバイスとしては、ネットワークインタフェース、及びディスクインタフェース等が考えられる。

40

【0023】

ROM 108 は、OS の起動時に用いられる第 1 の HW (ハードウェア) 制御プログラム 210 を格納するための専用の記憶媒体である。第 1 の HW 制御プログラム 210 は、第 1 の外部記憶装置 20 等に格納されてもよい。なお、第 1 の HW 制御プログラム 210 としては、BIOS (Basic Input/Output System) が考えられる。

【0024】

第 1 の HW 制御プログラム 210 は、OS の起動処理時に、物理計算機 10 のハードウ

50

ェアリソース１００を制御し、ＯＳブートローダ及びＯＳイメージ等、ＯＳの起動に必要なプログラムを外部記憶装置から読み出し、読み出されたプログラムをメモリに書き込む。

【００２５】

第１の外部記憶装置２０及び第２の外部記憶装置３０は、物理計算機１０がＯＳを起動するために必要なプログラム及び情報を格納する装置である。

【００２６】

第１の外部記憶装置２０及び第２の外部記憶装置３０は、例えば、ＨＤＤ（Hard Disk Drive）及びＳＳＤ（Solid State Drive）等の記憶媒体が考えられる。また、第１の外部記憶装置２０及び第２の外部記憶装置３０は、コントローラ及び複数の記憶媒体を備えるストレージシステムであってもよい。

10

【００２７】

物理計算機１０は、第１の外部記憶装置２０及び第２の外部記憶装置３０からプログラム及び情報を読み出し、複数のＯＳを起動させる。ここで、第１の外部記憶装置２０及び第２の外部記憶装置３０に格納されるプログラム及び情報について説明する。

【００２８】

第１の外部記憶装置２０は、プログラムとして、第１のＯＳブートローダ２４０、第１のＯＳイメージ２５０、アドレスデータ書換モジュール２２０、及び第２のＨＷ制御プログラム２３０を格納し、また、情報としてリソース範囲定義情報２６０を格納する。

【００２９】

第１のＯＳイメージ２５０は、第１のＯＳ２７０のＯＳイメージである。

20

【００３０】

第１のＯＳブートローダ２４０は、第１のＯＳイメージ２５０を用いて第１のＯＳ２７０を起動する。本実施例では、第１のＯＳブートローダ２４０は、第１の外部記憶装置２０から第１のＯＳイメージ２５０を読み出し、第１のＯＳイメージ２５０を第１の分割メモリ１０２に書き込み、第１のＯＳ２７０を起動する。

【００３１】

第２のＨＷ制御プログラム２３０は、ＯＳの起動処理時に、物理計算機１０のハードウェアリソース１００を制御し、ＯＳブートローダ及びＯＳイメージ等、ＯＳの起動に必要なプログラムを外部記憶装置から読み出し、読み出されたプログラムをメモリに書き込む。

30

【００３２】

アドレスデータ書換モジュール２２０は、第２のＨＷ制御プログラム２３０の起動時にアドレスデータを書き換え、その後、第２のＨＷ制御プログラム２３０を起動する。ここで、アドレスデータは、第２のＨＷ制御プログラム２３０の起動時に参照される関数等が格納される記憶領域の先頭アドレス（ポインタ）である。

【００３３】

リソース範囲定義情報２６０は、ハードウェアリソース１００の割り当てに関する情報を格納する。なお、リソース範囲定義情報２６０の詳細については、図５を用いて後述する。

40

【００３４】

第２の外部記憶装置３０は、プログラムとして、第２のＯＳブートローダ２４５及び第２のＯＳイメージ２５５を格納する。

【００３５】

第２のＯＳイメージ２５５は、第２のＯＳ２８０のＯＳイメージである。

【００３６】

第２のＯＳブートローダ２４５は、第２のＯＳイメージ２５５を用いて第２のＯＳ２８０を起動する。本実施例では、第２のＯＳブートローダ２４５は、第２の外部記憶装置３０から第２のＯＳイメージ２５５を読み出し、第２のＯＳイメージ２５５を第２の分割メモリ１０５に書き込み、第２のＯＳ２８０を起動する。

50

【 0 0 3 7 】

なお、第 2 の OS ブートローダ 2 4 5 及び第 2 の OS イメージ 2 5 5 は、第 1 の外部記憶装置 2 0 に格納されてもよい。

【 0 0 3 8 】

ハードウェアリソース 1 0 0 は、リソース範囲定義情報 2 6 0 に基づいて割り当てられる。本実施例では、OS 等の各プログラムが排他的に使用するハードウェアリソースと、各プログラムが共有して使用するハードウェアリソースとに分けられる。

【 0 0 3 9 】

ここで、排他的に使用するハードウェアリソースは、当該ハードウェアリソースが割り当てられたプログラムが占有的に使用し、他のプログラムが使用できないハードウェアリソースを示す。

10

【 0 0 4 0 】

以下の説明では、リソース範囲定義情報 2 6 0 に基づいて分割されたハードウェアリソース 1 0 0 を分割ハードウェアリソース 1 0 0 と記載する。本実施例では、分割ハードウェアリソース 1 0 0 には、分割プロセッサ、分割メモリ、分割共有メモリ、及び分割 I / O デバイスが含まれる。

【 0 0 4 1 】

また、以下の説明では、分割ハードウェアリソース、及び分割ハードウェアリソース上の稼働するプログラムをまとめて OS セグメントと記載する。本実施例では、一つの OS セグメントに一つの OS が稼働するものとする。

20

【 0 0 4 2 】

本実施例では、物理計算機 1 0 には、第 1 の OS セグメント 3 0 1 及び第 2 の OS セグメント 3 0 2 の二つの OS セグメントが存在する。

【 0 0 4 3 】

第 1 の OS セグメント 3 0 1 には、第 1 の分割ハードウェアリソース 1 1 1 が割り当てられる。第 1 の分割ハードウェアリソース 1 1 1 は、第 1 の分割プロセッサ 1 0 1、第 1 の分割メモリ 1 0 2、第 1 の分割 I / O デバイス 1 0 3、及び分割共有メモリ 1 0 7 を含む。また、第 1 の分割ハードウェアリソース 1 1 1 上では、第 1 の HW 制御プログラム 2 1 0、第 1 の OS ブートローダ 2 4 0、及び第 1 の OS 2 7 0 が稼働する。

【 0 0 4 4 】

第 2 の OS セグメント 3 0 2 には、第 2 の分割ハードウェアリソース 1 1 2 が割り当てられる。第 2 の分割ハードウェアリソース 1 1 2 は、第 2 の分割プロセッサ 1 0 4、第 2 の分割メモリ 1 0 5、第 2 の分割 I / O デバイス 1 0 6、及び分割共有メモリ 1 0 7 を含む。また、アドレスデータ書換モジュール 2 2 0、第 2 の HW 制御プログラム 2 3 0、第 2 の OS ブートローダ 2 4 5、及び第 2 の OS 2 8 0 が稼働する。

30

【 0 0 4 5 】

前述したように、各 OS セグメントに含まれるプログラムは、当該 OS セグメントに割り当てられた分割ハードウェアリソースのみを使用し、他の OS セグメントに割り当てられた分割ハードウェアリソースは使用しない。

【 0 0 4 6 】

ただし、後述するように、第 2 の OS セグメント用起動モジュール 2 7 1 を実行する第 1 の分割プロセッサ 1 0 1 は、第 2 の OS 2 8 0 の起動時に、第 2 の分割プロセッサ 1 0 4 に起動命令を発行することによって、当該第 2 の分割プロセッサ 1 0 4 を起動させる。ただし、当該処理は、第 2 の OS セグメント 3 0 2 の初期起動時にのみ許可される処理である。また、当該処理は、単に、分割プロセッサを起動させる処理であり、第 1 の OS セグメント 3 0 1 が、占有的に使用するわけではないため、セキュリティ及びハードウェアリソースの割当ポリシー等の問題は発生しない。

40

【 0 0 4 7 】

ここで、OS セグメント上で稼働するプログラムについて説明する。なお、第 1 の外部記憶装置 2 0 と同一のプログラムについては説明を省略する。

50

【 0 0 4 8 】

第 1 の OS セグメント 3 0 1 上では、第 1 の HW 制御プログラム 2 1 0、第 1 の OS ブートローダ 2 4 0、及び第 1 の OS 2 7 0 が稼働する。

【 0 0 4 9 】

第 1 の OS 2 7 0 は、第 1 の分割ハードウェアリソース 1 1 1 上で稼働する OS である。第 1 の OS 2 7 0 は、第 2 の OS セグメント用起動モジュール 2 7 1 を含む。

【 0 0 5 0 】

第 2 の OS セグメント用起動モジュール 2 7 1 は、第 2 の OS 2 8 0 を起動させるための命令を発効する。具体的には、第 2 の OS セグメント用起動モジュール 2 7 1 は、第 2 の分割プロセッサ 1 0 4 の起動を指示する。

10

【 0 0 5 1 】

なお、第 2 の OS セグメント用起動モジュール 2 7 1 を実行する第 1 の分割プロセッサ 1 0 1 が、第 2 の分割プロセッサ 1 0 4 を操作（起動）することになる。しかし、第 2 の OS セグメント 3 0 2 に割り当てられた第 2 の分割ハードウェアリソース 1 1 2 の初期起動時に限り許可される処理であり、セキュリティ面の問題も発生しない。

【 0 0 5 2 】

第 2 の OS セグメント 3 0 2 上では、アドレスデータ書換モジュール 2 2 0、第 2 の HW 制御プログラム 2 3 0、第 2 の OS ブートローダ 2 4 5、及び第 2 の OS 2 8 0 が稼働する。

【 0 0 5 3 】

20

本実施例の第 2 の OS 2 8 0 は、ハイパバイザに対応する。ハイパバイザは、第 2 の分割ハードウェアリソース 1 1 2 を用いて複数の L P A R 2 9 0 を生成し、生成された L P A R 2 9 0 上で複数のゲスト OS 2 9 5 を稼働させる。本実施例のハイパバイザは、リソース管理部 2 8 1 及びプロセススケジュール部 2 8 2 を有する。

【 0 0 5 4 】

リソース管理部 2 8 1 は、リソース範囲定義情報 2 6 0 に基づいて、第 2 の分割ハードウェアリソース 1 1 2 を管理する。プロセススケジュール部 2 8 2 は、第 2 の分割ハードウェアリソース 1 1 2 を用いて生成された L P A R 2 9 0 上にゲスト OS 2 9 5 を稼働させる。

【 0 0 5 5 】

30

図 2 は、本発明の実施例 1 における第 2 の HW 制御プログラム 2 3 0 の構成例を示す説明図である。

【 0 0 5 6 】

第 2 の HW 制御プログラム 2 3 0 は、第 2 の OS 2 8 0 の起動時に実行されるプログラムである。第 2 の HW 制御プログラム 2 3 0 は、第 2 の制御部用ローダ 2 3 1、第 2 の制御部 2 3 2 及び書換対象アドレス情報 2 3 3 を含む。

【 0 0 5 7 】

第 2 の制御部用ローダ 2 3 1 は、第 2 の制御部 2 3 2 を起動する。具体的には、第 2 の制御部用ローダ 2 3 1 は、第 2 の分割メモリ 1 0 5 の空き領域に第 2 の制御部 2 3 2 を書き込み、当該第 2 の制御部 2 3 2 を起動する。

40

【 0 0 5 8 】

本実施例では、後述するアドレスデータ書換モジュール 2 2 0 によって、第 2 の制御部用ローダ 2 3 1 が保持するアドレスデータに関する情報が変換される。

【 0 0 5 9 】

第 2 の制御部 2 3 2 は、第 2 の制御部用ローダ 2 3 1 によって起動された後、第 2 の分割ハードウェアリソース 1 1 2 を制御する。また、第 2 の制御部 2 3 2 は、第 2 の外部記憶装置 3 0 から第 2 の OS ブートローダ 2 4 5 を読み出し、第 2 の分割メモリ 1 0 5 に読み出された第 2 の OS ブートローダ 2 4 5 を書き込む。その後、第 2 の制御部 2 3 2 は、第 2 の分割メモリ 1 0 5 に書き込まれた第 2 の OS ブートローダ 2 4 5 を起動する。

【 0 0 6 0 】

50

書換対象アドレス情報 2 3 3 は、書換対象のアドレスデータが格納される記憶領域に関する情報を格納する。書換対象アドレス情報 2 3 3 の詳細については、図 6 を用いて後述する。

【 0 0 6 1 】

一般的な計算機には、計算機のハードウェアリソースを制御する H W 制御プログラムが存在する。H W 制御プログラムは、O S の起動時に、O S イメージより先にメモリに書き込まれ、実行される。

【 0 0 6 2 】

図 2 に示すように、一般的な H W 制御プログラムは、制御部用ロードを含む。制御部用ロードは、計算機の起動時に最初に行われるプログラムである。そのため、制御部用ロードは、関数等の一部のデータの読み出しについては、絶対アドレスで表記されたアドレスデータを参照するように設定されている。

10

【 0 0 6 3 】

二つの O S を独立に起動させるために、一つの計算機上に二つの異なる H W 制御プログラムを実行する場合、アドレスデータが重複することによって、制御部用ロードが正常に動作しない可能性がある。また、H W 制御プログラムをそれぞれ異なる記憶領域に書き込んだ場合であっても、同様の問題が発生する。

【 0 0 6 4 】

すなわち、H W 制御プログラムは、分割メモリに書き込まれた場合、格納される記憶領域が通常の記憶領域と異なる。そのため、アドレスデータが示す参照先のアドレスが不適切なアドレスとなり、制御部用ロードが正常に動作しない可能性がある。

20

【 0 0 6 5 】

本発明では、第 2 の H W 制御プログラム 2 3 0 が書き込まれる記憶領域に依存することなく、第 2 の制御部用ロード 2 3 1 が正常に動作するための機能を提供する。

【 0 0 6 6 】

具体的には、アドレスデータ書換モジュール 2 2 0 が、第 2 の H W 制御プログラム 2 3 0 が書き込まれた記憶領域に対応させて、第 2 の制御部用ロード 2 3 1 に設定されるアドレスデータを書き換える。これによって、参照先のアドレスが適切なアドレスに変更されるため、第 2 の制御部用ロード 2 3 1 は、正常に関数等を呼び出し、処理を実行することができる。

30

【 0 0 6 7 】

なお、第 2 の制御部 2 3 2 は、第 2 の制御部用ロード 2 3 1 によって第 2 の分割メモリ 1 0 5 の空き領域に書き込まれ、起動される。したがって、第 2 の制御部用ロード 2 3 1 が正常に稼働することによって、第 2 の制御部 2 3 2 が第 1 の分割メモリ 1 0 2 上で稼働するという問題は発生しない。

【 0 0 6 8 】

図 3 は、本発明の実施例 1 における第 1 の H W 制御プログラム 2 1 0 の構成例を示す説明図である。

【 0 0 6 9 】

第 1 の H W 制御プログラム 2 1 0 は、第 1 の O S 2 7 0 の起動時、すなわち、システム起動時に実行されるプログラムである。第 1 の H W 制御プログラム 2 1 0 は、第 1 の制御部用ロード 2 1 1、第 1 の制御部 2 1 2、及びベースアドレスデータ 2 1 3 を含む。

40

【 0 0 7 0 】

第 1 の制御部用ロード 2 1 1 は、第 1 の制御部 2 1 2 を起動する。具体的には、第 1 の制御部用ロード 2 1 1 は、第 1 の分割メモリ 1 0 2 の空き領域に第 1 の制御部 2 1 2 を書き込み、当該第 1 の制御部 2 1 2 を起動する。

【 0 0 7 1 】

第 1 の制御部 2 1 2 は、第 1 の制御部用ロード 2 1 1 によって起動された後、第 1 の分割ハードウェアリソース 1 1 1 を制御する。本実施例の第 1 の制御部 2 1 2 は、第 1 の O S 2 7 0 を起動する前に、以下のような処理を実行する。

50

【 0 0 7 2 】

第 1 の制御部 2 1 2 は、アドレスデータ書換モジュール 2 2 0、及び第 2 の HW 制御プログラム 2 3 0 を第 1 の外部記憶装置 2 0 から読み出し、第 2 の分割メモリ 1 0 5 に読み出されたアドレスデータ書換モジュール 2 2 0、及び第 2 の HW 制御プログラム 2 3 0 を書き込む。

【 0 0 7 3 】

第 1 の制御部 2 1 2 は、HW 制御プログラム 2 に含まれる第 2 の制御部用ローダ 2 3 1 が格納される記憶領域の先頭アドレスを取得し、取得されたアドレスをベースアドレスデータ 2 1 3 として保持する。さらに、第 1 の制御部 2 1 2 は、ベースアドレスデータ 2 1 3 を、第 2 の分割メモリ 1 0 5 の特定の記憶領域に書き込む。

10

【 0 0 7 4 】

なお、当該記憶領域は、予め、定義された記憶領域であるものとする。また、アドレスデータ書換モジュール 2 2 0 は、当該記憶領域を参照するように予め設定されているものとする。

【 0 0 7 5 】

前述した処理が完了した後、第 1 の制御部 2 1 2 は、第 1 の外部記憶装置 2 0 から第 1 の OS ブートローダ 2 4 0 を読み出し、第 1 の分割メモリ 1 0 2 に読み出された第 1 の OS ブートローダ 2 4 0 を書き込む。その後、第 1 の制御部 2 1 2 は、第 1 の分割メモリ 1 0 2 に書き込まれた第 1 の OS ブートローダ 2 4 0 を起動する。

【 0 0 7 6 】

ベースアドレスデータ 2 1 3 は、第 2 の制御部用ローダ 2 3 1 が格納される記憶領域の先頭アドレスである。

20

【 0 0 7 7 】

図 4 は、本発明の実施例 1 におけるアドレスデータ書換モジュール 2 2 0 の構成例を示す説明図である。

【 0 0 7 8 】

アドレスデータ書換モジュール 2 2 0 は、第 2 の制御部用ローダ 2 3 1 が保持するアドレスデータを変換する。アドレスデータ書換モジュール 2 2 0 は、第 1 の OS 2 7 0 起動前に、第 1 の HW 制御プログラム 2 1 0 によって、第 1 の外部記憶装置 2 0 から読み出され、第 2 の分割メモリ 1 0 5 に書き込まれる。また、アドレスデータ書換モジュール 2 2 0 は、第 2 の OS セグメント用起動モジュール 2 7 1 によって起動される。

30

【 0 0 7 9 】

アドレスデータ書換モジュール 2 2 0 は、アドレスデータ書換部 2 2 1、及び第 2 の HW 制御プログラム用ローダ 2 2 2 を含む。

【 0 0 8 0 】

アドレスデータ書換部 2 2 1 は、第 2 の制御部用ローダ 2 3 1 におけるアドレスデータを変換する。具体的には、以下のような処理が実行される。

【 0 0 8 1 】

アドレスデータ書換部 2 2 1 は、第 2 の HW 制御プログラム 2 3 0 のリロケーションセクションを参照することによって、アドレスデータが格納される記憶領域のアドレスを取得する。

40

【 0 0 8 2 】

アドレスデータ書換部 2 2 1 は、取得されたアドレスに対応する記憶領域に格納されるアドレスデータに、ベースアドレスデータ 2 1 3 を加算することによって、当該アドレスデータを書き換える。

【 0 0 8 3 】

なお、予め、第 2 の HW 制御プログラム 2 3 0 にアドレスデータが格納される記憶領域のアドレスの情報が含まれる場合、アドレスデータ書換モジュール 2 2 0 は、第 2 の HW 制御プログラム 2 3 0 に含まれる情報を参照することによって、当該アドレスデータが含まれる記憶領域のアドレスを取得してもよい。

50

【 0 0 8 4 】

第2のHW制御プログラム用ローダ222は、第2のHW制御プログラム230に含まれる第2の制御部用ローダ231のアドレスデータが書き換えられた後、当該第2の制御部用ローダ231を起動する。

【 0 0 8 5 】

書き換えられたアドレスデータは、第2の分割メモリ105の所定の記憶領域に対応するアドレスデータに変換されているため、第2の制御部用ローダ231は、書き換えられたアドレスデータに基づいて、関数等を正しく参照することができる。したがって、第2の制御部用ローダ231は、正常に動作することができる。

【 0 0 8 6 】

図5は、本発明の実施例1におけるリソース範囲定義情報260の一例を示す説明図である。

【 0 0 8 7 】

リソース範囲定義情報260は、分割ハードウェアリソースとして割り当てるハードウェアリソース100の範囲（割当量、割当数等）に関する情報を格納する。本実施例では、リソース範囲定義情報260は、プロセッサの定義情報261、メモリの定義情報262、及びI/Oデバイスの定義情報263を含む。

【 0 0 8 8 】

プロセッサの定義情報261は、プロセッサ番号2611及び属性2612を含む。プロセッサ番号2611は、物理計算機10が備えるプロセッサを一意に識別するための識別番号である。属性2612は、プロセッサの割り当て属性を示す情報である。

【 0 0 8 9 】

各OSセグメントが共有して使用するプロセッサの場合、属性2612には共有して使用する旨の情報が格納される。例えば、属性2612には「共有」が格納される。

【 0 0 9 0 】

また、OSセグメントが占有的に使用するプロセッサの場合、属性2612にはOSセグメントを特定するための情報が格納される。例えば、第1のOSセグメント301に割り当てられるプロセッサの場合、属性2612には「第1の分割プロセッサ」が格納される。これによって、各OSセグメントが占有的に使用するプロセッサを定義することができる。

【 0 0 9 1 】

図5に示す例では、プロセッサ番号2611が奇数の識別番号であるプロセッサは、第1のOSセグメント301に割り当てられる第1の分割プロセッサ101として定義され、プロセッサ番号2611が偶数の識別番号であるプロセッサは、第2のOSセグメント302に割り当てられる第2の分割プロセッサ104として定義される。

【 0 0 9 2 】

メモリの定義情報262は、範囲始点2621、範囲終点2622、及び属性2623を含む。

【 0 0 9 3 】

範囲始点2621及び範囲終点2622は、分割メモリとして割り当てる記憶領域の始点及び終点のアドレスである。属性2623は、記憶領域の割り当て属性を示す情報である。なお、範囲始点2621及び範囲終点2622の代わりに、記憶領域の先頭アドレス及び記憶領域の長さを、記憶領域を特定する情報として用いてもよい。

【 0 0 9 4 】

各OSセグメントが共有して使用する記憶領域、すなわち、分割共有メモリの場合、属性2623には「共有」が格納される。

【 0 0 9 5 】

また、OSセグメントが占有的に使用する記憶領域の場合、属性2623にはOSセグメントを特定するための情報が格納される。例えば、第1のOSセグメント301に割り当てられる記憶領域の場合、属性2623には「第1の分割メモリ」が格納される。これ

10

20

30

40

50

によって、各OSセグメントが占有的に使用する記憶領域、すなわち、分割メモリを定義することができる。

【0096】

図5に示す例では、物理アドレス1から物理アドレス2の範囲の記憶領域は、第1のOSセグメント301に割り当てられる第1の分割メモリ102として定義され、物理アドレス3から物理アドレス4の範囲の記憶領域は、第2のOSセグメント302に割り当てられる第2の分割メモリ105として定義される。また、物理アドレス2から物理アドレス3の範囲の記憶領域は、第1のOSセグメント301及び第2のOSセグメント302に割り当てられる分割共有メモリ107として定義される。

【0097】

I/Oデバイスの定義情報263は、I/Oデバイス番号2631及び属性2632を含む。I/Oデバイス番号2631は、物理計算機10が備えるI/Oを一意に識別するための識別番号である。属性2632は、I/Oデバイスの割り当て属性を示す情報である。

【0098】

各OSセグメントが共有して使用するI/Oデバイスの場合、属性2632には共有して使用する旨の情報が格納される。例えば、属性2632には「共有」が格納される。

【0099】

また、OSセグメントが占有的に使用するI/Oデバイスの場合、属性2632にはOSセグメントを特定するための情報が格納される。例えば、第1のOSセグメント301に割り当てられるI/Oデバイスの場合、属性2632には「第1の分割I/Oデバイス」が格納される。これによって、各OSセグメントが占有的に使用するI/Oデバイスを定義することができる。

【0100】

図5に示す例では、I/Oデバイス番号2631が奇数の識別番号であるI/Oデバイスは、第1のOSセグメント301に割り当てられる第1の分割I/Oデバイス103として定義され、I/Oデバイス番号2631が偶数の識別番号のI/Oデバイスは、第2のOSセグメント302に割り当てられる第2の分割I/Oデバイス106として定義される。

【0101】

図6は、本発明の実施例1の書換対象アドレス情報233の一例を示す説明図である。

【0102】

書換対象アドレス情報233は、第2の制御部用ローダ231が格納される記憶領域の先頭アドレスからのオフセットを格納する。

【0103】

図7A、図7B、図7C、及び図7Dは、本発明の実施例1における計算機システムの起動処理を説明するシーケンス図である。

【0104】

物理計算機10に電源が投入されると、まず、ROM108に格納される第1のHW制御プログラム210の第1の制御部用ローダ211が起動する。第1の制御部用ローダ211は、第1のHW制御プログラム210をメモリに書き込み、第1の制御部212を起動する(ステップS101)。その後、第1の制御部用ローダ211は、処理を終了する(ステップS102)。

【0105】

本実施例では、第1の分割プロセッサ101に含まれる一つのプロセッサが、第1の制御部用ローダ211を実行し、また、第1の制御部用ローダ211によって起動された第1の制御部212を実行する。また、第1の制御部212は、第1の分割メモリ102の記憶領域に書き込まれる。

【0106】

なお、第1の制御部用ローダ211及び第1の制御部212を実行するプロセッサ、及

10

20

30

40

50

び第1のHW制御プログラム210が書き込まれる記憶領域は、前述したものに限定されず、システム構成に合わせて適宜変更できる。

【0107】

以下の説明では、第2の制御部用ローダ231を実行する一つの第1の分割プロセッサ101を第1の起動用プロセッサとも記載する。

【0108】

第1の制御部212は、起動後、第1の外部記憶装置20から、リソース範囲定義情報260を読み出し、読み出されたリソース範囲定義情報260を分割共有メモリ107に書き込む(ステップS111)。

【0109】

第1の制御部212は、第1の外部記憶装置20からアドレスデータ書換モジュール220及び第2のHW制御プログラム230を読み出し、リソース範囲定義情報260に基づいて、読み出されたアドレスデータ書換モジュール220及び第2のHW制御プログラム230を第2の分割メモリ105に書き込む(ステップS112)。

【0110】

第1の制御部212は、第2の分割メモリ105に書き込まれた第2のHW制御プログラム230に含まれる第2の制御部用ローダ231が格納される記憶領域の先頭アドレスを取得する(ステップS113)。第1の制御部212は、取得された先頭アドレスをベースアドレスデータ213として保持する。

【0111】

第1の制御部212は、リソース範囲定義情報260に基づいて、第2の分割メモリ105にベースアドレスデータ213を書き込む(ステップS114)。本実施例では、第1の制御部212は、第2の分割メモリ105の記憶領域のうち、予め、定義された記憶領域にベースアドレスデータ213を書き込む。

【0112】

第1の制御部212は、第1のOSブートローダ240を第1の外部記憶装置20から読み出し、第1の分割メモリ102に読み出された第1のOSブートローダ240を書き込み、第1のOSブートローダ240を起動する(ステップS115)。その後、第1の制御部212は、処理を終了する。第1のOSブートローダ240は、第1の起動用プロセッサによって実行される。

【0113】

なお、第1の制御部212は、第2の分割メモリ105へのアクセスを行っているが、第1のOS270起動前の処理であり、第1のOS270によって第2のOS280のデータを直接扱われることはない。したがって、セキュリティ上の問題は発生しない。

【0114】

第1のOSブートローダ240は、起動後、第1の外部記憶装置20から第1のOSイメージ250を読み出し、リソース範囲定義情報260に基づいて、読み出された第1のOSイメージ250を第1の分割メモリ102に書き込む(ステップS121)。

【0115】

第1のOSブートローダ240は、第1のOSイメージ250を用いて、第1のOS270を起動する(ステップS122)。その後、第1のOSブートローダ240は、処理を終了する。第1の起動用プロセッサが、第1のOSイメージ250を用いて、第1のOS270を起動する。

【0116】

第1のOS270は、起動後、リソース範囲定義情報260に基づいて、第1のOS270が使用する第1の分割プロセッサ101を起動する(ステップS131)。これ以降、第1のOS270は、第1の分割プロセッサ101を用いて稼働する。

【0117】

第1のOS270は、第2のOSセグメント用起動モジュール271を起動する(ステップS132)。その後、第1のOS270は、通常の処理を実行する。なお、本発明は

10

20

30

40

50

、第1のOS 270が実行する処理の種類に限定されない。

【0118】

第2のOSセグメント用起動モジュール271は、起動後、第2の分割プロセッサ104を起動させ、アドレスデータ書換モジュール220の起動を指示する(ステップS141)。その後、第2のOSセグメント用起動モジュール271は、処理を終了する。

【0119】

具体的には、第2のOSセグメント用起動モジュール271は、第2の分割プロセッサ104のうちの少なくとも一つのプロセッサの起動命令を発行する。以下、第2のOSセグメント用起動モジュール271によって起動される第2の分割プロセッサ104を第2の起動用プロセッサとも記載する。

10

【0120】

なお、ステップS141の処理は、第2のOSセグメント302の初期起動時にのみ許可される処理である。また、当該処理は、単に、分割プロセッサを起動させる処理であり、第1のOSセグメント301が、占有的に使用するわけではないため、セキュリティ及びハードウェアリソースの割当ポリシ等の問題は発生しない。

【0121】

第2のOSセグメント用起動モジュール271によって起動された第2の起動用プロセッサは、リソース範囲定義情報260に基づいて、第2の分割メモリ105に書き込まれるアドレスデータ書換モジュール220を実行する。これによって、最初に、アドレスデータ書換部221が起動される。

20

【0122】

アドレスデータ書換部221は、起動後、第2の制御部用ローダ231に設定されるアドレスデータを書き換える(ステップS201)。具体的には、以下のような処理が実行される。

【0123】

まず、アドレスデータ書換部221は、第2の分割メモリ105に書き込まれた第2のHW制御プログラム230から書換対象アドレス情報233を取得する。これによって、書換対象のアドレスデータが特定される。

【0124】

アドレスデータ書換部221は、第2の分割メモリ105に書き込まれるベースアドレスデータ213を取得する。

30

【0125】

アドレスデータ書換部221は、書換対象アドレス情報233に基づいて、書換対象のアドレスデータが格納される記憶領域のアドレスデータに、ベースアドレスデータ213の値を加算することによって、アドレスデータを書き換える。例えば、第2の制御部用ローダ231のバイナリコードが書き換えられる。

【0126】

以上がステップS201の処理の説明である。

【0127】

次に、アドレスデータ書換部221は、第2のHW制御プログラム用ローダ222を起動する(ステップS202)。その後、アドレスデータ書換部221は、処理を終了する。第2のHW制御プログラム用ローダ222は、第2の起動用プロセッサによって実行される。

40

【0128】

第2のHW制御プログラム用ローダ222は、起動後、第2の制御部用ローダ231を起動する(ステップS211)。その後、第2のHW制御プログラム用ローダ222は、処理を終了する。第2の制御部用ローダ231は、第2の起動用プロセッサによって実行される。

【0129】

第2の制御部用ローダ231は、起動後、第2の制御部232を起動する(ステップS

50

2 2 1)。その後、第2の制御部用ローダ2 3 1は、処理を終了する。第2の制御部2 3 2は、第2の起動用プロセッサによって実行される。

【0 1 3 0】

本実施例では、第2の制御部用ローダ2 3 1は、書き換えられたアドレスデータを用いて処理を実行するため、不正なアドレスに基づくエラーは発生しない。なお、第2の制御部用ローダ2 3 1が実行する処理は、公知のものであるため詳細な説明は省略する。

【0 1 3 1】

第2の制御部2 3 2は、第2のOSブートローダ2 4 5を第2の外部記憶装置3 0から読み出し、第2の分割メモリ1 0 5に読み出された第2のOSブートローダ2 4 5を書き込み、第2のOSブートローダ2 4 5を起動する(ステップS 2 3 1)。その後、第2の制御部2 3 2は、処理を終了する。第2のOSブートローダ2 4 5は、第2の起動用プロセッサによって実行される。

10

【0 1 3 2】

第2のOSブートローダ2 4 5は、起動後、第2の外部記憶装置3 0から第2のOSイメージ2 5 5を読み出し、リソース範囲定義情報2 6 0に基づいて、読み出された第2のOSイメージ2 5 5を第2の分割メモリ1 0 5に書き込む(ステップS 2 4 1)。

【0 1 3 3】

第2のOSブートローダ2 4 5は、第2のOSイメージ2 5 5を用いて、第2のOS 2 8 0を起動する(ステップS 2 4 2)。その後、第2のOSブートローダ2 4 5は、処理を終了する。第2のプロセッサが、第2のOSイメージ2 5 5を用いて、第2のOS 2 8 0を起動する。第2のOS 2 8 0は、第2の起動用プロセッサによって実行される。

20

【0 1 3 4】

第2のOS 2 8 0は、起動後、リソース範囲定義情報2 6 0に基づいて、第2のOS 2 8 0が使用する第2の分割プロセッサ1 0 4を起動する(ステップS 2 5 1)。これ以降、第2のOS 2 8 0は、第2の分割プロセッサ1 0 4を用いて稼働する。

【0 1 3 5】

図8は、本発明の実施例1における第2の制御部用ローダ2 3 1のアドレスデータの変化の状態を示す説明図である。

【0 1 3 6】

記憶領域4 0 0は、物理計算機1 0が備えるメモリによって提供されるメモリの記憶領域を示す。本実施例では、P__A d 1(物理アドレス1)からP__A d 2(物理アドレス2)までの記憶領域が第1の分割メモリ1 0 2に対応し、P__A d 2(物理アドレス2)からP__A d 3(物理アドレス3)までの記憶領域が分割共有メモリ1 0 7に対応し、P__A d 3(物理アドレス3)からP__A d 4(物理アドレス4)までの記憶領域が第2の分割メモリ1 0 5に対応する。

30

【0 1 3 7】

また、メモリマップ4 0 1は、アドレスデータが書き換えられる前の第2の制御部用ローダ2 3 1のメモリマップを示し、メモリマップ4 0 2は、アドレスデータが書き換えられた後の第2の制御部用ローダ2 3 1のメモリマップを示す。

【0 1 3 8】

40

図8に示す例では、メモリマップ4 0 1、4 0 2の太枠の範囲、すなわち、A d__1からA d__2 9までの範囲に含まれる記憶領域が書換対象のアドレスデータが格納される記憶領域である。

【0 1 3 9】

図8に示すように、アドレスデータ書換部2 2 1によって、各記憶領域に格納されるアドレスデータは、元のアドレスデータにベースアドレスデータ2 1 3の値が加算された値に書き換えられる。

【0 1 4 0】

これによって、第2の制御部用ローダ2 3 1は、第2の分割メモリ1 0 5内の所定のアドレスを参照することによって、正常に動作することができる。すなわち、第2のHW制

50

御プログラム 230 がどの記憶領域に書き込まれても、当該第 2 の HW 制御プログラム 230 を正常に動作させることが可能となる。

【0141】

以上で説明したように、本発明によれば、第 2 の HW 制御プログラム 230 が第 2 の OS 280 の起動処理を行うため、第 1 の OS 270 が直接関与することなく、第 2 の OS 280 を起動することができる。

【0142】

また、第 2 の HW 制御プログラム 230 に含まれる第 2 の制御部用ローダ 231 のアドレスデータを予め書き換えることによって、第 2 の HW 制御プログラム 230 がどの記憶領域に書き込まれても起動させることができる。

【0143】

したがって、第 1 の OS 270 が第 2 の OS 280 のデータを扱うことができないシステム構成の場合、第 1 の HW 制御プログラム 210 のデバイスドライバが、第 2 の外部記憶装置 30 に対応していない場合であっても、第 2 の OS 280 を起動することができる。

【0144】

(変形例 1)

実施例 1 では、第 2 の HW 制御プログラム 230 が書換対象アドレス情報 233 を保持していたが、本発明はこれに限定されない。この場合、ステップ S201 の処理は以下のようになる。

【0145】

アドレスデータ書換部 221 は、第 2 の制御部用ローダ 231 の実行形式のファイルを参照し、当該実行形式のファイルに含まれるリロケーションセクション(リロケーション情報)に基づいて、書換対象のアドレスデータが含まれる記憶領域を特定する。

【0146】

なお、第 2 の制御部用ローダ 231 の実行形式のファイルは、予め、第 1 の外部記憶装置 20 等に格納し、第 1 の制御部 212 が第 2 の分割メモリ 105 に書き込む方法が考えられる。

【0147】

その他の処理は、実施例 1 と同一であるため説明を省略する。

【0148】

(変形例 2)

実施例 1 では、第 1 の制御部 212 が、アドレスデータ書換モジュール 220 及び第 2 の HW 制御プログラム 230 を第 2 の分割メモリ 105 に書き込んでいたが、本発明はこれに限定されない。

【0149】

一つの方法としては、第 1 の OS セグメント 301 上で稼働する任意のプログラムが、アドレスデータ書換モジュール 220 及び第 2 の HW 制御プログラム 230 を第 2 の分割メモリ 105 に書き込む。この場合、第 2 の分割メモリ 105 への一時的なアクセスを可能とするために、以下のような処理が必要となる。

【0150】

任意のプログラムは、第 2 の分割メモリ 105 の一部の記憶領域に対して、書き込み可能なページテーブルを生成し、当該ページテーブルを参照することによって、第 1 の OS 270 用の仮想メモリ空間に第 2 の分割メモリ 105 の一部の記憶領域をマッピングする。

【0151】

また、アドレスデータ書換モジュール 220 及び第 2 の HW 制御プログラム 230 が第 2 の分割メモリ 105 に書き込まれた後、任意のプログラムは、第 1 の OS 270 用の仮想メモリ空間にマップされた第 2 の分割メモリ 105 の一部の記憶領域のマッピングを解除する。これによって、第 2 の分割メモリ 105 へのアクセスが禁止される。

10

20

30

40

50

【 0 1 5 2 】

他の方法としては、第 1 の制御部 2 1 2 が、第 2 の分割メモリ 1 0 5 ではなく、分割共有メモリ 1 0 7 に、アドレスデータ書換モジュール 2 2 0 及び第 2 の H W 制御プログラム 2 3 0 を書き込む。この場合、第 2 の分割プロセッサ 1 0 4 が、分割共有メモリ 1 0 7 にアクセスするように設定すればよい。

【 0 1 5 3 】

各実施例では、ソフトウェアによる制御を用いた例について説明したが、その一部をハードウェアによって実現することも可能である。

【 0 1 5 4 】

以上、本発明を添付の図面を参照して詳細に説明したが、本発明はこのような具体的構成に限定されるものではなく、添付した請求の範囲の趣旨内における様々な変更及び同等の構成を含むものである。

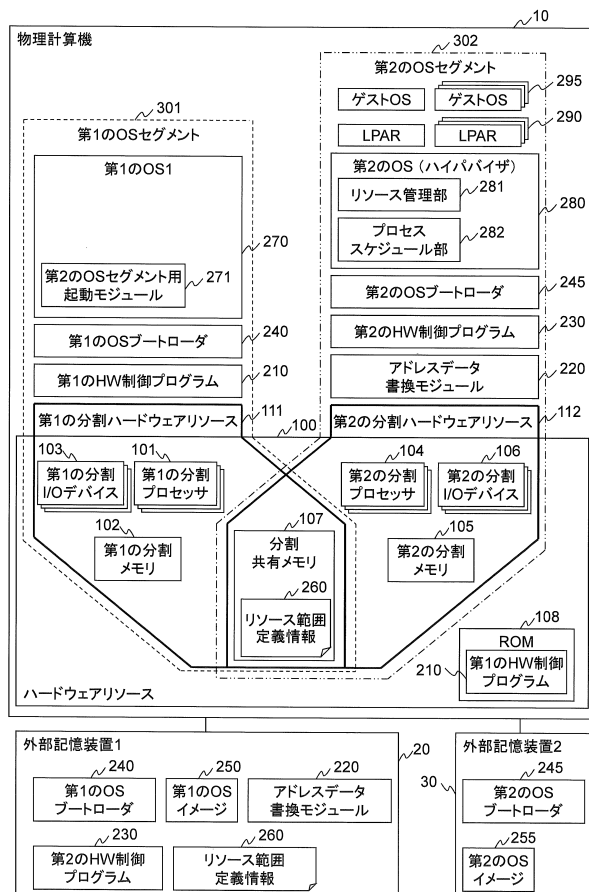
【符号の説明】

【 0 1 5 5 】

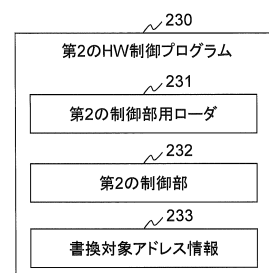
1 0	物理計算機	
2 0	第 1 の外部記憶装置	
3 0	第 2 の外部記憶装置	
1 0 0	ハードウェアリソース	
1 0 1	第 1 の分割プロセッサ	
1 0 2	第 1 の分割メモリ	20
1 0 3	第 1 の分割 I / O デバイス	
1 0 4	第 2 の分割プロセッサ	
1 0 5	第 2 の分割メモリ	
1 0 6	第 2 の分割 I / O デバイス	
1 0 7	分割共有メモリ	
1 0 8	R O M	
1 1 1	第 1 の分割ハードウェアリソース	
1 1 2	第 2 の分割ハードウェアリソース	
2 1 0	第 1 の H W 制御プログラム	
2 1 1	第 1 の制御部用ローダ	30
2 1 2	第 1 の制御部	
2 1 3	ベースアドレスデータ	
2 2 0	アドレスデータ書換モジュール	
2 2 1	アドレスデータ書換部	
2 2 2	第 2 の H W 制御プログラム用ローダ	
2 3 0	第 2 の H W 制御プログラム	
2 3 1	第 2 の制御部用ローダ	
2 3 2	第 2 の制御部	
2 3 3	書換対象アドレス情報	
2 4 0	第 1 の O S ブートローダ	40
2 4 5	第 2 の O S ブートローダ	
2 5 0	第 1 の O S イメージ	
2 5 5	第 2 の O S イメージ	
2 6 0	リソース範囲定義情報	
2 6 1	プロセッサの定義情報	
2 6 2	メモリの定義情報	
2 6 3	I / O デバイスの定義情報	
2 7 0	第 1 の O S	
2 7 1	第 2 の O S セグメント用起動モジュール	
2 8 0	第 2 の O S	50

- 2 8 1 リソース管理部
- 2 8 2 プロセススケジュール部
- 2 9 0 L P A R
- 2 9 5 ゲストOS
- 3 0 1 第1のOSセグメント
- 3 0 2 第2のOSセグメント
- 4 0 0 記憶領域
- 4 0 1 メモリマップ
- 4 0 2 メモリマップ

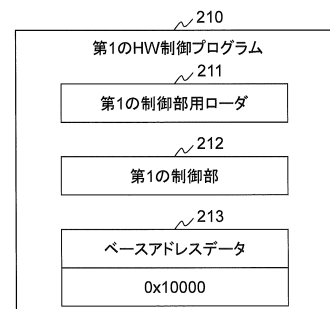
【図1】



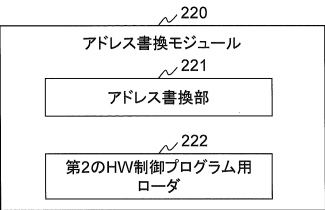
【図2】



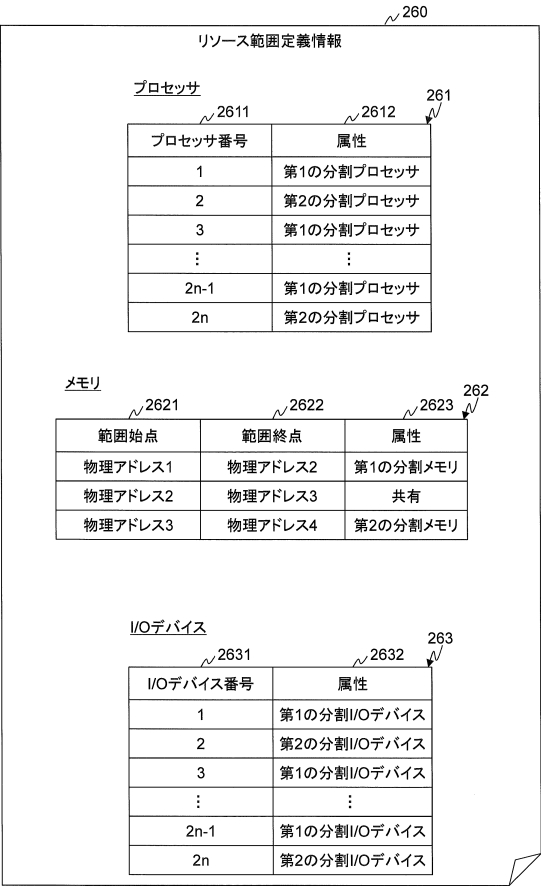
【図3】



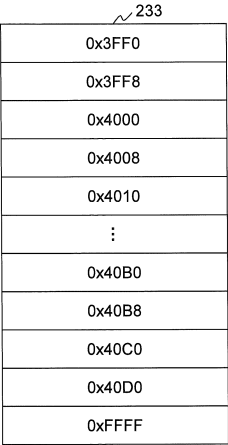
【図 4】



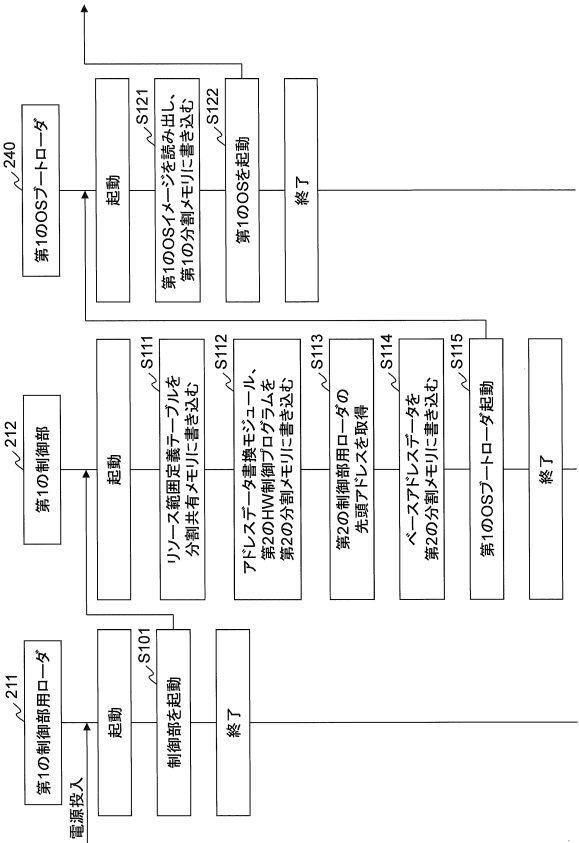
【図 5】



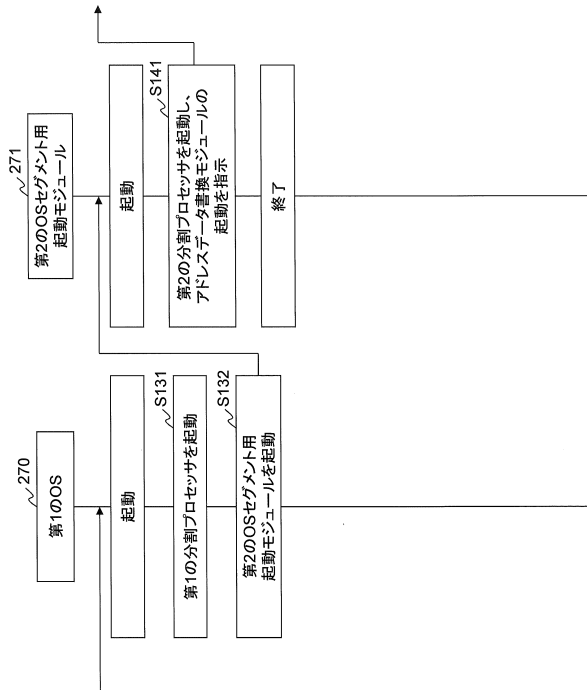
【図 6】



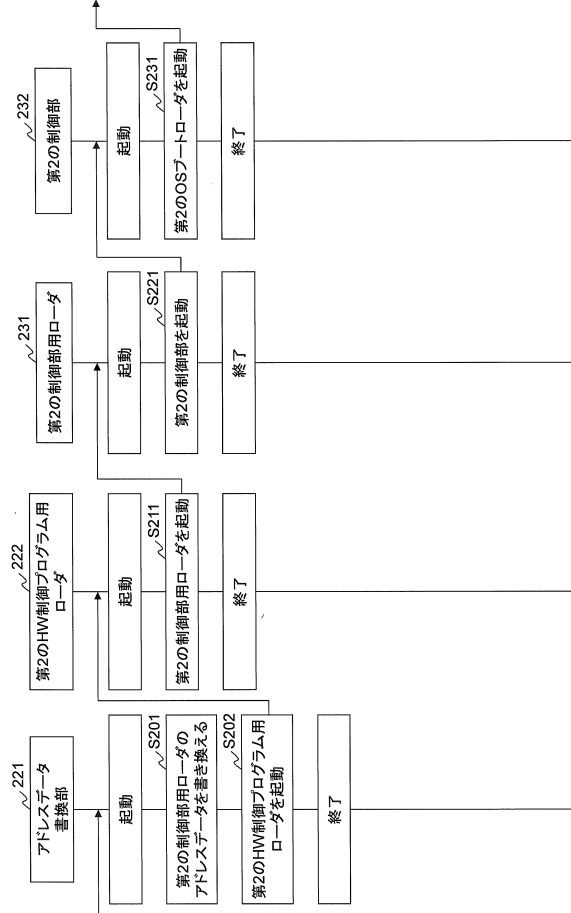
【図 7 A】



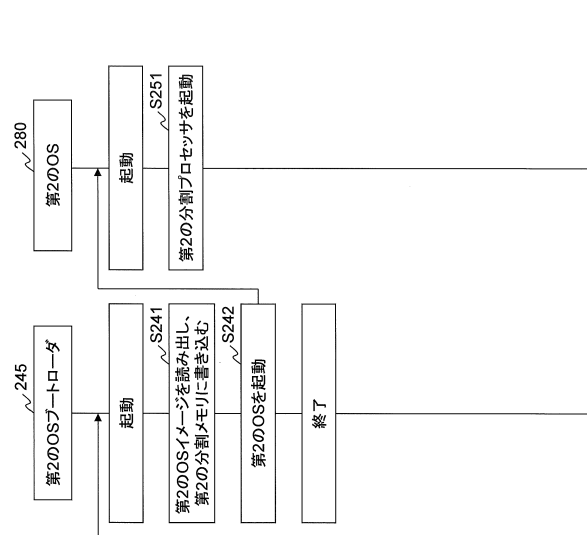
【 図 7 B 】



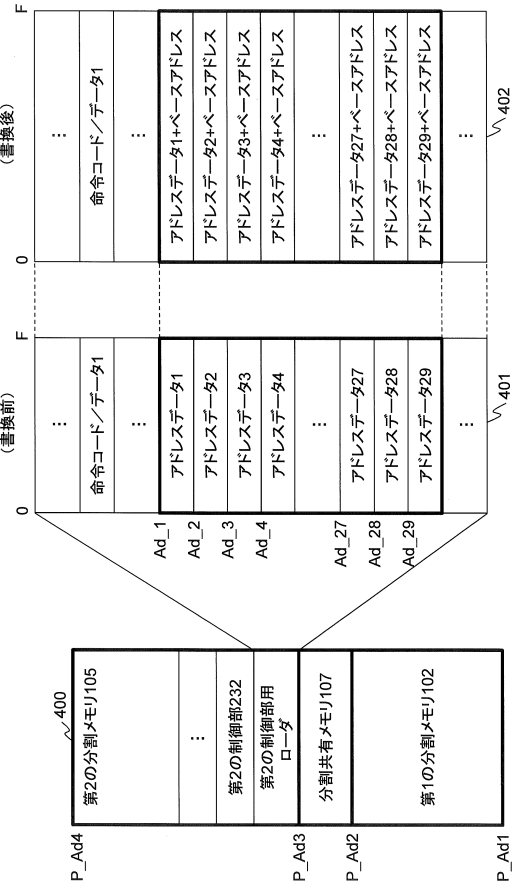
【 図 7 C 】



【圖 7 D】



【圖 8】



フロントページの続き

- (72)発明者 下城 孝
神奈川県横浜市西区みなとみらい二丁目3番3号 株式会社日立情報通信エンジニアリング内
- (72)発明者 小長谷 昌利
神奈川県横浜市西区みなとみらい二丁目3番3号 株式会社日立情報通信エンジニアリング内
- (72)発明者 中川 義仁
神奈川県小田原市中里322番2号 株式会社日立製作所 ITプラットフォーム事業本部内
- (72)発明者 森木 俊臣
東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内

審査官 石川 亮

- (56)参考文献 特開2001-290665(JP, A)
特許第4916576(JP, B2)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/445