(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2017-156907
(P2017-156907A)
(43)公開日　平成29年9月7日(2017.9.7)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 9/50	(2006.01)	G06F	9/46	465C	
		G06F	9/46	465A	

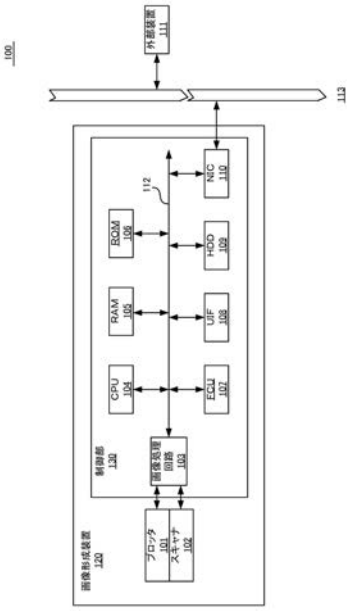
審査請求 未請求　請求項の数 9　ＯＬ　（全 13 頁）

(21)出願番号　特願2016-38571(P2016-38571)
(22)出願日　平成28年3月1日(2016.3.1)

(71)出願人　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号
(74)代理人　100110607
弁理士　間山　進也
(72)発明者　山▲崎▼　智広
東京都大田区中馬込１丁目３番６号　株式会社リコー内

(54)【発明の名称】情報処理装置、情報処理方法およびプログラム

(57)【要約】
【課題】　情報処理装置、情報処理方法およびプログラムを提供すること。
【解決手段】　本実施形態の画像形成装置１２０は、並列実行を行う複数のコアを有するＣＰＵ１０４を含む制御部１３０を含んでおり、制御部１３０のＣＰＵ１０４は、並列実行するべき処理を、並列実行するべき処理の優先度または負荷に応じて、複数のコア全部を対象として複数の処理を発行するか、または処理を専ら実行する固定したコアを選択して処理を発行することで負荷分散を行っている。
【選択図】　図１



【特許請求の範囲】
【請求項１】
並列実行を行う情報処理装置であって、
複数のコアを有するＣＰＵを含む制御手段と、
前記並列実行するべき処理を、前記複数のコアに分散して発行する手段と
を含み、前記並列実行するべき処理を、前記複数のコアに分散して発行する手段は、
前記並列実行するべき処理の優先度または負荷に応じて、前記複数のコア全部を対象として複数の処理を発行するか、または処理を専ら実行する固定したコアを選択して処理を発行する
情報処理装置。
【請求項２】
前記情報処理装置は、ユーザにより要求される処理を前記優先度の高い処理として、前記固定したコアを割り当てる、請求項１に記載の情報処理装置。
【請求項３】
前記情報処理装置は、ＨＤＤ装置のデータ消去処理および画像描画処理のための負荷を、それぞれ異なる前記固定したコアに処理させる、請求項１または２に記載の情報処理装置。
【請求項４】
前記情報処理装置は、さらに、
前記複数のコアに依頼する処理の識別値を、先入れ先出し方式で管理するとともに、前記識別値に応じて処理先のコアを固定するか否かを判断するためのリスト手段と、
前記リスト手段に登録された前記処理を、前記判断に応じて予め空にされた当該処理を実行するべき前記コアに対応する実行待ち手段に送付する実行コア指定手段とを含む、請求項１～３のいずれか１項に記載の情報処理装置。
【請求項５】
前記情報処理装置は、組込装置である、請求項１～４のいずれか１項に記載の情報処理装置。
【請求項６】
並列実行を行う情報処理装置が実行する装置実行可能な情報処理方法であって、
前記並列実行するべき処理の優先度または負荷を決定するステップと、
前記複数のコアに前記並列実行するべき処理を分散して発行するステップと、
前記複数のコアに前記並列実行するべき処理を発行するステップと
を含み、
前記並列実行するべき処理を、前記複数のコアに分散して発行するステップは、
前記並列実行するべき処理の優先度または負荷に応じて、前記複数のコア全部を対象として複数の処理を発行するか、または処理を専ら実行する固定したコアを選択して処理を発行するするステップを含む、情報処理方法。
【請求項７】
前記並列実行するべき処理を、前記複数のコアに分散して発行するステップは、ユーザにより要求される処理を前記優先度の高い処理に前記固定したコアを割り当てるステップを含む、請求項６に記載の情報処理方法。
【請求項８】
さらに、
前記複数のコアに依頼する処理の識別値を、先入れ先出し方式で管理するとともに、前記識別値に応じて処理先のコアを固定するか否かを判断するステップと
前記判断に応じて予め空にされた当該処理を実行するべき前記コアに対応する実行待ち手段に前記処理を送付するステップと
を含む請求項６または７に記載の情報処理方法。
【請求項９】
処理を並列実行するための情報処理装置実行可能なプログラムであって、前記プログラ

ムは、情報処理装置が、
前記並列実行するべき処理の優先度または負荷を決定するステップと、
前記複数のコアに前記並列実行するべき処理を分散して発行するステップと、
前記複数のコアに前記並列実行するべき処理を発行するステップと
を実行し、
前記並列実行するべき処理を、前記複数のコアに分散して発行するステップは、
前記並列実行するべき処理の優先度または負荷に応じて、前記複数のコア全部を対象として複数の処理を発行するか、または処理を専ら実行する固定したコアを選択して処理を発行するするステップを含む、
プログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、情報処理装置、情報処理方法、およびプログラムに関する。
【背景技術】
【０００２】
マルチコアプロセッサシステムの動作形態として、ＳＭＰ（Symmetrical Multi Processing）と、ＡＭＰ（Asymmetrical Multi Processing）がある。ＳＭＰは、複数のプログラムを複数のＣＰＵコアに動的に分散させ、各コアの負荷を分散させる。一方、ＡＭＰは特定のプログラムを特定のＣＰＵコアに割り付けることでプログラムのリアルタイム性を改善させている。
【０００３】
組込装置などでは、精密な動作タイミングが必要なプログラム（例えば、ドライバなどのようにハードウェアを直接操作するプログラム）が存在する場合、ＳＭＰ制御を行うと、装置の動作中にＣＰＵが変更される場合も発生し、キャッシュを有効利用できないこと、および処理のリアルタイム性を保証できないこと、などからＡＭＰ制御を利用する場合も知られている。また、マルチコアプロセッサ・システムではＳＭＰ制御と、ＡＭＰ制御を混在させる構成もまたすでに知られている。
【０００４】
しかし、従来のＳＭＰ制御の場合、プログラムは、ＣＰＵコアに対して均等に割り振られる。一方、ＡＭＰ制御の場合、ＣＰＵコアが遊休状態となる期間が発生することも想定される。また、ＳＭＰ制御の場合には、プログラムの優先度を考慮して、プログラムの実行先を制御することも可能である。しかしながら、画像形成装置のような組込装置の場合、画像処理やセキュリティ性向上のためのＨＤＤ（ハードディスク・ドライブ）データ消去など高負荷の処理が存在する。ＳＭＰ制御の場合、上述した高負荷の処理や優先度の高い処理を、他の処理を後回しにしてでも優先的に実行させたい場合でも充分対応できないという問題がある。
【０００５】
一方、特定の処理の優先度を上げると、その他の処理に影響が出てしまい、例えばユーザの操作レスポンスが低下するなどの問題もある。これまで、マルチコア・プロセッサを使用する組込装置におけるプロセス制御について検討されている。例えば、特開２０１１‐１００２７７号公報(特許文献１）では、特定のプロセッサに処理負荷が偏ることを抑制するため、システム起動時に前回起動時に記録しておいた起動コア番号からＡＭＰ制御のコアを、順次変えることにより特定のコアに負荷が偏ることを抑制する。
【０００６】
しかしながら、特許文献１では、特定プログラムを実行するコアを事前に固定していること、および起動時にコア固定するプログラムを固定的に割り当てているため、優先度の高い処理や高負荷処理によってユーザの操作性が低下する点を改善することはできなかった。
【発明の概要】

【発明が解決しようとする課題】
【０００７】
本発明は、組込装置でのプログラム実行において、マルチコア・プロセッシングにおけるユーザ操作性を改善する技術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】
【０００８】
本発明によれば、
並列実行を行う情報処理装置であって、
複数のコアを有するＣＰＵを含む制御手段と、
前記並列実行するべき処理を、前記複数のコアに分散して発行する手段と
を含み、前記並列実行するべき処理を、前記複数のコアに分散して発行する手段は、
前記並列実行するべき処理の優先度または負荷に応じて、前記複数のコア全部を対象として複数の処理を発行するか、または処理を専ら実行する固定したコアを選択して処理を発行する
情報処理装置が提供される。
【発明の効果】
【０００９】
本発明によれば、組込装置でのプログラム実行において、マルチコア・プロセッシングにおけるユーザ操作性を改善する技術を提供することが可能となる。
【図面の簡単な説明】
【００１０】
【図１】画像形成装置１２０のハードウェアブロック１００を示す図。
【図２】本実施形態のＣＰＵ１０４の詳細な構造を示す図。
【図３】本実施形態の画像形成装置１２０のソフトウェアブロック３００を示す図。
【図４】本実施形態のプログラムの並列実行を説明する概念図。
【図５】本実施形態の負荷分散処理のフローチャート。
【図６】実施形態で、アプリの機能を提供するプログラムのプログラム識別値を特定のコアに割り当てる情報の実施形態を示す図。
【図７】本実施形態において、優先／高負荷ジョブが終了した後に、第１のポリシーによる負荷分散処理に復帰する際の処理のフローチャート。
【図８】本実施形態において、優先／高負荷アプリの実行中に、負荷分散ポリシーの切替を行う実施形態のフローチャート。
【発明を実施するための形態】
【００１１】
以下、本発明を実施形態により説明するが、本発明は、後述する実施形態に限定されるものではない。図１は、本実施形態の情報処理装置が、組込装置、具体的には、画像形成装置であるものとして、画像形成装置１２０のハードウェアブロック１００を示す。なお、図１には、説明の便宜上、画像形成装置１２０が外部アクセスする際の要素も記述した。本実施形態の画像形成装置１２０は、組込装置として構成されており、プロッタ１０１といった画像を出力する機器、スキャナ１０２といった原稿をデジタル化する機器、および画像形成装置１２０の全体の処理を司る制御部１３０を含んでいる。制御部１３０が、本実施形態における制御手段に相当する。
【００１２】
制御部１３０は、例えば特定用途集積回路（ＡＩＳＣ）を備える画像処理回路１０３を含んでおり、画像処理回路１０３は、プロッタ１０１およびスキャナ１０２の機能を提供するために、画像処理を実行する。さらに制御部１３０は、ＣＰＵ１０４、ＲＡＭ１０５、ＲＯＭ１０６を含んでいる。
【００１３】
ＣＰＵ１０４は、本実施形態ではマルチコア・プロセッサであり、ＣＰＵコアへの負荷分散を行いながら複数のプログラム並列実行する。ＲＡＭ１０５は、オペレーティング・

システム（ＯＳ）といったプログラムを読み込んでプログラムをＣＰＵ１０４が実行するための実行空間を提供する。その他、ＣＰＵ１０４がプログラムを実行するためのデータなどを格納する記憶空間を提供する。

【００１４】

ＲＯＭ１０６は、ＢＩＯＳ（Bacic Input Output System)、ブートストラップ（Bootstrap）プログラム、その他、ＣＰＵ１０４が機能を提供するためのプログラムを記憶しており、ＣＰＵ１０４の起動時にプログラムをＣＰＵ１０４に読み込ませ、ハードウェアの初期設定、ＯＳ起動、実行モニタなどの機能を可能とする。以上のハードウェア・ブロックは、システムバス１１２により相互接続されていて、システムクロックに従ってその動作が制御されている。

【００１５】

また、制御部１３０は、ＰＣＩｅといった周辺バスを介して接続されたＦＣＵ１０７、ＵＩＦ１０８、ＨＤＤ１０９およびＮＩＣ１１０を備えている。ＦＣＵ１０７は、Ｇ３、Ｇ４といった規格のファクシミリ通信を制御するための回路である。ＵＩＦ１０８は、ディスプレイ装置、タッチパネル、テンキー／キーボードなどのユーザインタフェースを制御するための回路である。ハードディスク・ドライブ（以下、ＨＤＤ装置１０９として参照する）１０９は、例えばＡＴＡ、ＳＡＴＡ、ＵＳＢなどの通信プロトコルを使用して、外付け記録媒体を制御するための回路である。

【００１６】

ＮＩＣ（ネットワークインタフェース・カード）１１０は、ネットワーク１１３へと画像形成装置１２０を接続させることで、ウェブ・サーバ、ストレージ・サーバ、認証サーバ、クラウド・サーバといった外部装置１１１との情報通信を可能としている。本実施形態のネットワーク１１３は、イーサネット（登録商標）、ＦＴＨ、ＩＥＥＥ８０２．ｘなどの有線または無線プロトコルを使用してＬＡＮ、インターネットを適宜含んで構成することができ、特に通信プロトコルには限定はない。

【００１７】

本実施形態で使用するＣＰＵ１０４は、例えば、ＰＥＮＴＩＵＭ（登録商標）ＤＵＡＬ ＣＯＲＥ(登録商標）、ＣＯＲＥ２ ＤＵＯ（登録商標）、ＣＯＲＥ２ ＱＵＡＤ（登録商標）、ＣＥＬＥＲＯＮ（登録商標） ＤＵＡＬ ＣＯＲＥ、ＡＴＯＭ(登録商標）、ＣＯＲＥ２ＤＵＯ（登録商標）、ＣＯＲＥ２ＱＵＡＤ（登録商標）、ＣＯＲＥｉ（登録商標）シリーズなどの他、ＸＥＯＮ(登録商標）、マルチコア構成を備えるＰＥＮＴＩＵＭ（登録商標）互換ＣＰＵ、ＰＯＷＥＲ ＰＣ（登録商標）などを挙げることができるがこれらに限定されるものではない。

【００１８】

使用するオペレーティング・システム（ＯＳ）としては、ＷｉｎｄｏｗｓＳｅｒｖｅｒ（登録商標）、ＵＮＩＸ（登録商標）、ＬＩＮＵＸ（登録商標）、ＯＰＥＮＢＳＤ、ＣｅｎｔＯＳ、Ｕｂｎｔｕまたはそれ以外の適切なＯＳを挙げることができる。さらに、ＣＰＵ１０４は、上述したＯＳ上で動作する、アセンブラ言語、Ｃ、Ｃ＋＋、Ｖｉｓｕａｌ Ｃ＋＋、ＶｉｓｕａｌＢａｓｉｃ、Ｊａｖａ（登録商標）、ＪａｖａＳｃｒｉｐｔ（登録商標）、Ｐｅｒｌ、Ｒｕｂｙ、Ｐｙｔｈｏｎなどのプログラミング言語により記述されたアプリケーション・プログラムを格納し、実行することができる。

【００１９】

図２は、本実施形態のＣＰＵ１０４の詳細な構造を示す。なお、図２には、ＣＰＵ１０４の機能を説明する上で、ＲＡＭ１０５の適切なアドレス領域に構成される、実行キュー２０１～２０４および実行プログラムリスト２１８も示す。実行プログラムリスト２１８が、本実施形態におけるプログラムの識別値に応じて処理先のコアを固定するか否かを判断するためのリスト手段に相当する。

【００２０】

ＣＰＵ１０４は、図２に示すように複数のコア２０５、コア２０６、コア２０７、コア２０８を備えるマルチコア構成を採用する。各コア２０５～２０７は、それぞれのために

用意された実行キュー２０１～２０４に格納されたプログラムを、先入れ先出し順に読み出してプログラムを実行する。本実施形態における実行キューが、本実施形態における実行待ち手段に相当する。

【００２１】

また、本実施形態では、画像形成装置１２０の機能を提供するためのプログラムの実行要求があると、プログラムまたはプログラムの呼び出しを検出したＯＳは、プログラム識別値を実行プログラムリスト２１８に一旦登録する。そして、ＯＳ３２０の実行コア指定部３２０ａによるコア指定処理を経て、当該プログラムが指定されたコアに対応づけされた実行キュー２０１～２０４に送られ、指定されたコアによる実行が可能とされる。

【００２２】

本実施形態の実行キュー２０５～２０８が、プログラムの実行順を制御する手段に相当する。実行コア指定部３２０ａが、コア指定処理を完了した後、実行プログラムリスト２１８の処理済みの項目は、実行プログラムリスト２１８から削除される。以上の処理を使用して実行プログラムリスト２１８は、プログラムが各キューに割り当てられる前に実行コア指定部３２０ａが、実行予定のプログラムを管理する機能を提供し、次にプログラムを実行するべきコア２０５～２０８にプログラムを動的に割り当てることを可能としている。

【００２３】

ＣＰＵ１０４は、コア２０５～２０８それぞれが使用するＬ１キャッシュ２０９～２１２を備えていて、コア２０５～２０８のアクセス効率を改善している。さらに、ＣＰＵ１０４は、コア２０５～２０８それぞれに対し、いわゆるアプリケーション・キャッシュとしても参照されるＬ２キャッシュを提供し、アプリケーションレベルで使用するデータなどへのアクセス効率を改善する。さらに、ＣＰＵ１０４は、図示する具体例では、Ｌ３キャッシュ２１７を備えている。Ｌ３キャッシュは、コア２０５～２０８が排他的に使用するデータではなく、コア２０５～２０８が共有することができるデータを格納する機能を提供し、コア２０５～２０８が使用する可能性のある共有データへのアクセス効率を改善させている。

【００２４】

なお、本実施形態で使用するＣＰＵ１０４は、４コア構成に限定されるものではなく、２コア、８コア、１６コアなど、処理上の要求に応じて増加させることができる。また、他の実施形態では、マルチコアのＣＰＵをデュアルに実装する、デュアルＣＰＵ構成を使用することもできる。

【００２５】

図３は、本実施形態の画像形成装置１２０のソフトウェアブロック３００を示す。画像形成装置１２０は、画像形成装置１２０が各種機能を提供するためのアプリケーション・プログラム（以下、単に「アプリ」として参照する。）を実装している。これらのアプリとしては、例示的にコピーアプリ３１０ａ、プリンタアプリ３１０ｂ、スキャナアプリ３１０ｃ、ＦＡＸアプリ３１０ｄ、およびＨＤＤ消去アプリ３１０ｅを挙げることができるが、これらに限定されるものではない。この他、Ｆｉｒｅｆｏｘ（登録商標）、Ｍｏｚｉｌａ（登録商標）、Ｉｎｔｅｒｎｅｔ　Ｅｘｐｌｏｒｅｒ（登録商標）、Ａｐａｃｈｅ（登録商標）といった各種のアプリを、提供するべき機能に応じて実装することができることは言うまでもないことである。

【００２６】

各アプリ３１０ａ～３１０ｅは、ＯＳ３２０の管理下で動作しており、ＯＳにより、各アプリ３１０ａ～３１０ｅが呼び出し、動作の開始・終了、実行時エラーなどの例外処理などが行われる。本実施形態では、ＯＳ３２０は、さらに実行コア指定部３２０ａを備えていて、本実施形態に従い、アプリを実行する際のコアを指定する機能を提供する。このような実行コア指定部３２０ａとしては、例えばＯｐｅｎＭＰのコマンドを解釈実行することができＯＳのモジュール、ＵＮＩＸ（登録商標）、ＬＩＮＵＸ（登録商標）のｔａｓｋｓｅｔコマンドを実行するシェルコマンド領域を挙げることができる。本実施形態の実

行コア指定部３２０ａが、並列実行するべき処理を、複数のコアに分散して発行する手段および実行コア指定手段に相当する。

【００２７】

さらに、画像形成装置１２０は、ハードウェアとのインタフェースおよび制御を行うためのＢＩＯＳおよびＣＰＵの稼働状態、温度、ファン速度などをモニタすることで、コアの負荷を監視および管理するＢＩＯＳ／モニタプログラム３３０を備えている。ＯＳ３２０は、ＢＩＯＳ／モニタプログラム３３０からの情報を取得し、コア２０５～２０８に割り当てるべき、プログラム／スレッドを制御する。

【００２８】

一般に、特定のコアの負荷が設定された閾値を超えた場合、ＯＳ３２０は、対応する実行キューに登録されたプログラム識別値を、特定のポリシーを使用して他のコアのための実行キューに移動させることで、負荷の高いコアの負荷を軽減して負荷をバランスする。例えば、移転先のコアは、その時点で最も稼働率の低いコアとすることができる。また例えば、一定時間ごとに実行キューの内容を他の実行キューに移転する態様を使用することができる。

【００２９】

本実施形態においては、コアの処理負荷を分散させる際、実行キュー単位でコアを移転するのではなく、ＯＳが、アプリの実行を特定のコアに固定する場合には、実行コアが指定されたアプリが、実行コア２０５～２０８に対応する実行キュー２０１～２０４に残され、他のアプリが他のコアに移動される。この結果、コアを固定して実行させるべきアプリが、特定のコアで実行され、他のアプリが、負荷分散ポリシーで規定される他のコアで実行される。すなわち、本実施形態では、特定のコアに対応する実行キューの入れ替えを、コアが指定されたプログラムを残して他のアプリを、他のコアに対応する実行キューに移動させるものである。

【００３０】

このため、実行コアが指定されていないアプリは、一定時間毎に行われる実行キューの入れ替え対象とされる。実行キュー２０１～２０４に、コアを固定するべきアプリが登録されていなければ、当該実行キューの内容を、一定時間で移転させることにより、コア２０５～２０８をＯＳに設定されたポリシーにより負荷分散が可能となる。一方、実行キュー２０１～２０４にコアを固定するべきアプリが登録された場合、当該実行キューは、当該アプリを残して、他のアプリが他の残りの実行キューに移転され、そして当該実行キューは、アプリが終了するまで、他のアプリが割り当てられることがない。

【００３１】

図４は、本実施形態のプログラムの並列実行を説明する概念図である。図４（ａ）は従来の処理分散ポリシーに従う実行制御のシーケンスであり、図４（ｂ）は、本実施形態による処理分散ポリシーに従う実行制御のシーケンスである。図４（ａ）の従来例では、画像形成装置１２０において、図３に示したコピーアプリ３１０ａを、コア２０７で、スキャナアプリ３０１ｃをコア２０８で実行するように固定する。そして、その他のプログラムをコア２０５、２０６で並列実行するようにコアを割り当てる。

【００３２】

従来例では、次回に画像形成装置１２０が動作開始する場合、コピーアプリ３１０ａをコア２０８に割り当て、コア２０７を、スキャナアプリ３１０ｃに割り当てることで、負荷分散を行う。

【００３３】

しかしながら、図４（ａ）に示すように、ユーザがスキャナ機能をあまり使用しない場合、コア２０８は、アイドル状態が続いてしまい、また再起動後にスキャナアプリ３１０ｃをコア２０７に固定したとしても、コア２０８は、依然としてアイドル状態を続けてしまうことは同じである。このため、アプリを単純に特定のコアに割り当てるポリシーでは、システム全体で無駄が生じてしまうことに対応できない。

【００３４】

一方、本実施形態では、画像形成装置１２０の起動時は、アプリを特定の実行コアに対して割り当てることはせず、全コアをアプリ用に開放する。そして、実行コアの指定があるアプリが実行される場合に、当該コアで実行中のアプリを他のコアに移転させ、実行コアが指定されたアプリにコアの容量を占有させる負荷分散ポリシーを採用し、ＣＰＵ資源を有効利用するものである。

【００３５】

図４（ｂ）の実施形態を使用して具体的に本実施形態の作用を説明する、コピーアプリ３１０ａの起動をトリガとして、それまでコア２０７で実行されていたアプリは、別のコア２０８に移動される。また、コア２０８は、図４（ａ）と同様に、スキャナアプリ３１０ｃに割り当てられているものの、本実施形態では、コア２０８は、スキャナアプリ３１０ｃが実行されない限り、他のコア２０５、２０６と同様に、その他のアプリの処理１－ａ、１－ｂ、２－ａ、２－ｂなどを実行することが可能である。

【００３６】

すなわち、本実施形態で採用するポリシーによれば、優先性の高い、または高負荷を要求するプログラムが起動されるまでは、コア全部２０５～２０８が均等にプログラムを実行する第１のポリシーで負荷分散を行う。また、優先性が高いか、高負荷を要求するアプリが要求された場合、実行するコアを固定し、それ以外のアプリにはコアを固定することなく実行させる第２のポリシーで負荷分散を達成する。この結果、ユーザ要求による割り込みの発生を優先させつつ、コアの負荷分散を達成することが可能となり、コアの利用性が改善される。

【００３７】

なお、本実施形態において、優先度の高いプログラムとは、ユーザ要求による割り込みを契機として動作開始する、コピーアプリ３１０ａ、スキャナアプリ３１０ｃ、ＦＡＸアプリ３１０ｄの他、検索アプリ、ネットワーク・アクセスアプリなどを挙げることができる。また、本実施形態で高負荷を要求するアプリまたは処理としては、ＨＤＤ消去アプリ３１０ｅによるデータ消去処理および画像描画処理などを挙げることができるがこれらに限定されるものではない。

【００３８】

図５は、本実施形態の負荷分散処理のフローチャートである。図５の処理は、ステップＳ５００から開始し、ステップＳ５０１で優先／高負荷処理が起動されると、ステップＳ５０２で、優先／高負荷アプリ実行コア要求を、実装形式に応じて当該アプリまたはＯＳが発行する。ステップＳ５０３では、ＯＳ３０２の実行コア指定部３２０ａが、優先／高負荷アプリ実行コア要求に対応して実行固定コアを確保する処理を実行する。

【００３９】

この処理は、例えば、対応する実行キュー２０１～２０４に登録されているプログラム識別値を他のコアに対応する実行キューに移動させる処理を伴う。さらにこの処理においては、その時点で固定予定のコアが他のアプリを実行している場合、例えば処理負荷の低いコアに処理を移動することで、アプリに対してコアを固定的に確保することにより即時的なアプリの実行可能とする。

【００４０】

その後、ステップＳ５０４では、実行プログラムリスト２１８を検索し、先頭に登録されているアプリが、すでに優先／高負荷アプリか否かを判断する。この判断は、アプリに固有に割り当てられているプログラム識別値を使用して実行プログラムリスト２１８を検索することで実装することができる。実行プログラムリスト２１８の先頭に登録されているアプリが、現在実行コア指定要求が発行されている優先／高負荷アプリである場合（ｙｅｓ）、ステップＳ５０６で当該優先／高負荷アプリのためすでに確保したコアに対応する実行キューにプログラムを登録し、アプリを指定したコアに発行する。この時点ではステップＳ５０３の処理で対応する実行キューはからの状態とされているので、実行キューに登録された後、アプリは、即時的実行される。

【００４１】

一方、ステップＳ５０５の判断で、実行プログラムリスト２１８の先頭に登録されているアプリが、その時点で実行コア指定要求を発行した優先／高負荷アプリでない場合（ｎｏ）、ステップＳ５０７で固定するべき他のコアへとアプリの処理を発行する。この際の処理は、ステップＳ５０６で説明したと同様にし実行される。

【００４２】

ステップＳ５０６、Ｓ５０７で処理を発行した後、ステップＳ５０８で、実行プログラムリスト２１８が空か否かを判断する。実行プログラムリスト２１８が空の場合（ｙｅｓ）、コア指定を行わなければならないアプリはもうないので、処理をステップＳ５０９に進めて処理を終了する。一方、実行プログラムリスト２１８が空でない場合（ｎｏ）、処理をステップＳ５０４に戻し、アプリを実行するコアを割り当てて処理を発行させ、ステップＳ５０８で実行プログラムリスト２１８が空になるまで、処理を反復させる。なお、実行プログラムリスト２１８の先頭に登録されているアプリが、コア指定を要しないアプリである場合、実行コア指定部３２０ａは、第１のポリシーに従って、排他使用されていない、いずれかのコアにアプリを実行させる。

【００４３】

以上の処理により、優先／高負荷アプリの起動割り込みがあるまでは、コアの処理負荷に応じてアプリに対してコアを割り当てながら、優先／高負荷アプリの要求があった時にはプログラムに指定したコアを割り当てて処理を可能とする、本実施形態のポリシーに従い。マルチコア間の負荷分散を可能としている。

【００４４】

図６は、本実施形態で、アプリの機能を提供するプログラムのプログラム識別値を特定のコアに割り当てる情報の実施形態を示す。図６の情報は、プログラムの実装形式に応じてＯＳ３２０の実行コア指定部３２０ａの実行時データとして作成することもできるし、アプリを提供するプログラムのコア指定情報として行ごとに分散させてアプリのプログラム・コード中に記述することもできる。図６に示したアプリは、ユーザ要求により優先度を高くするべきアプリの他、例えば、ＨＤＤデータ消去といった高負荷を要求するアプリである。

【００４５】

図６に示したアプリは、単なる例示であり、この他にもユーザによる割り込みに応じて処理を開始するべきアプリまたは処理は、図６の情報に追加することができる。これら以外の、例えば、機器チェック、ファクシミリ受信、稼働状況モニタ、その他の非ユーザ要求のアプリ、サービスについては、特定の用途および目的に応じて図６の情報に追加できるが、本実施形態の要旨においては、「その他のアプリ」として分類されるべきものである。

【００４６】

図７は、本実施形態において、優先／高負荷ジョブが終了した後に、第１のポリシーによる負荷分散処理に復帰する際の処理のフローチャートである。図７の処理は、ステップＳ７００から開始し、ステップＳ７０１で、図６に示した情報～優先／高負荷アプリのプログラム識別値を取得し、ステップＳ７０２で実行プログラムリスト２１８を検索する。

【００４７】

ステップＳ７０３で、実行プログラムリスト２１８先頭に優先／高負荷アプリのプログラム識別値が登録されていない場合（ｎｏ）、ステップＳ７０４で実行コア指定を解除し、ステップＳ７０５で、実行プログラムリスト２１８が空か否かを判断する。実行プログラムリスト２１８が空でない場合、ステップＳ７０２に処理を戻し、再度、先頭に優先／高負荷処理プログラムが登録されているか否かについて検索する。

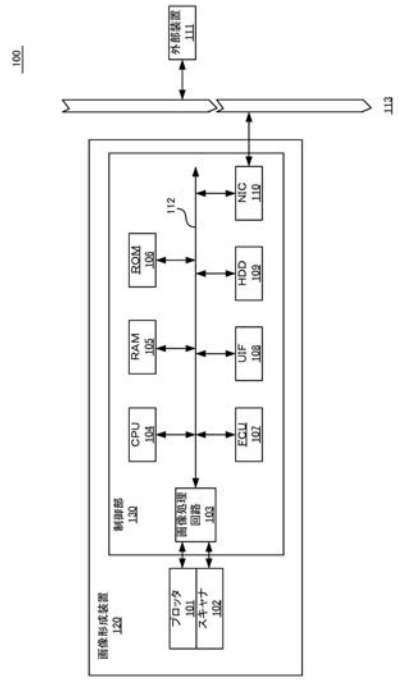
【００４８】

一方、ステップＳ７０３の判断で実行プログラムリスト２１８に優先／高負荷アプリのプログラム識別値が登録されている場合（ｙｅｓ）、実行コアを確保・実行するための図５の処理に処理を委ね、図７の処理はステップＳ７０６に進んで終了する。また、ステップＳ７０５の判断で、実行プログラムリスト２１８が空と判断された場合（ｙｅｓ）、処

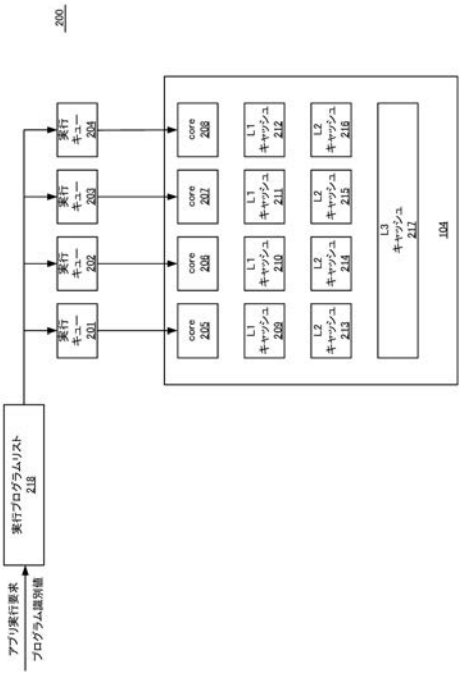
理はステップＳ７０６に進み終了する。
【００４９】
なお、図７の処理は、図５の処理に後続して呼び出されることが好ましいが、図５の処理と統合して、一体として負荷分散処理のプログラムモジュールとして実装することができる。
【００５０】
図８は、本実施形態において、優先／高負荷アプリの実行中に、負荷分散ポリシーの切替を行う実施形態のフローチャートである。図８の処理は、ステップＳ８００から開始し、ステップＳ８０１で優先／高負荷アプリが呼び出されると、例えばプロセス管理リストに対応するアプリを登録し、当該アプリに実行フラグを設定する。その後、ステップＳ８０２で、図５のステップＳ５０６またはステップＳ５０７を呼び出して、コアを固定した第２のポリシーの下での処理を開始させる。
【００５１】
次いでステップＳ８０３で新規アプリの起動が要求されると、ステップＳ８０４で、新規アプリが優先／高負荷アプリであるか否かを判断し、優先／高負荷アプリでない場合（ｎｏ）、処理をステップＳ８１０に分岐させ、図７のステップＳ７０２～Ｓ７０５の処理を適用して、利用可能な固定されていないコアの中から第１のポリシーに従い、コアを割り当て、アプリを該当する実行キュー２０１～２０４に発行する。
【００５２】
一方、ステップＳ８０４の判断で、優先／高負荷アプリが要求されていると判断された場合（ｙｅｓ）、ステップＳ８０５で、図５のステップＳ５０７を呼び出して、コアを固定した第２のポリシーの下での処理を開始させる。その後、ステップＳ８０６で、先行して動作していた優先／高負荷アプリが終了すると、ステップＳ８０７で図７のステップ７０２～Ｓ７０５を呼び出して、実行コアの解除を行う。その後、ステップＳ８０８でプロセス管理リストの該当するアプリの動作フラグを解除し、処理はステップＳ８０９に進んで終了する。
【００５３】
以上の処理によって、優先／高負荷アプリの実行中に起動されたプログラムにおいても、優先／高負荷アプリとの競合による意図しない性能低下を防ぐことができる。なお、本実施形態の負荷分散プログラムは、アプリの機能を提供するプログラム自体に、コア指定、コア解除を行うＯＳコマンドを記述しておくことができる。
【００５４】
また、他の実施形態では、ＳｈｅｌｌＳｃｒｉｐｔといったＯＳコマンドをアプリを提供するためのプログラムとは別に用意し、アプリの実行状態をＯＳプログラムが監視して第１のポリシーおよび第２のポリシーを切り替え制御することができる。また、本実施形態の負荷分散処理を実行するためのプログラムは、Ｃ＋＋、Ｊａｖａ（登録商標）、ＳｈｅｌｌＳｃｒｉｐｔなどいかなる言語で記述することができ、特に限定されるものではない。
【００５５】
これまで本発明を、実施形態をもって説明してきたが、本発明は、実施形態に限定されるものではなく、他の実施形態、追加、変更、削除など、当業者が想到することができる範囲内で変更することができ、いずれの態様においても本発明の作用・効果を奏する限り、本発明の範囲に含まれるものである。
【符号の説明】
【００５６】
１００　：ハードウェアブロック
１０１　：プロッタ
１０２　：スキャナ
１０３　：画像処理回路
１０４　：ＣＰＵ

１０５　　：ＲＡＭ
１０６　　：ＲＯＭ
１０９　　：ＨＤＤ装置
１１１　　：外部装置
１１３　　：ネットワーク
１２０　　：画像形成装置
１３０　　：制御部
２０１～２０４　　：実行キュー
２０５～２０８　　：コア
２０９～２１２　　：Ｌ１キャッシュ
２１３～２１６　　：Ｌ２キャッシュ
２１７　　：Ｌ３キャッシュ
２１８　　：実行プログラムリスト
３００　　：ソフトウェアブロック
３０１ｃ　：スキャナアプリ
３１０ａ　：コピーアプリ
３１０ｂ　：プリンタアプリ
３１０ｃ　：スキャナアプリ
３１０ｄ　：ＦＡＸアプリ
３１０ｅ　：ＨＤＤ消去アプリ
３２０ａ　：実行コア指定部
３３０　　：モニタプログラム
【先行技術文献】
【特許文献】
【００５７】
【特許文献１】特開２０１１－１００２７７号公報

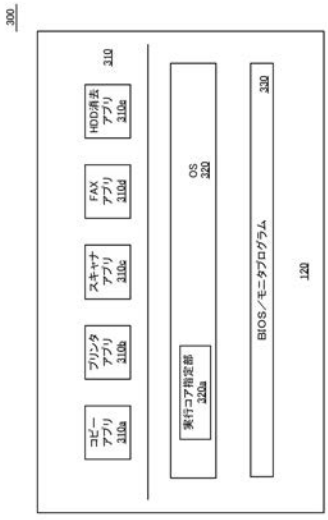
【図１】


100
外部装置
111
113
画像形成装置
120
制御部
130
CPU
104
RAM
105
ROM
106
HDD
109
UIF
108
NIC
110
ECU
107
112
画像処理回路
103
プロッタ
101
スキャナ
102

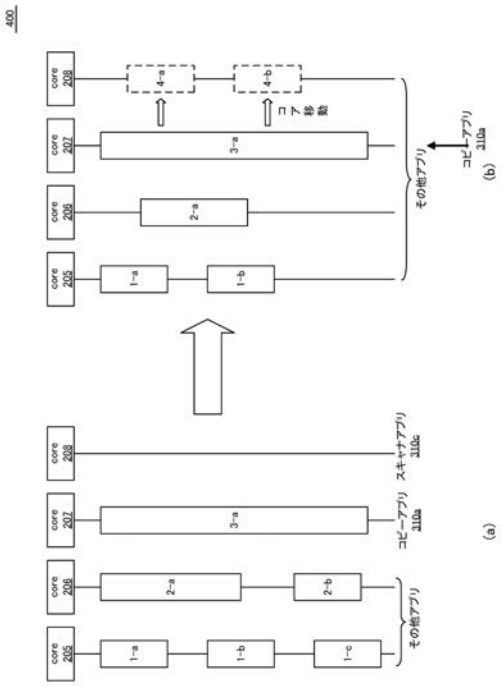
【図２】


200
実行キュー
201
実行キュー
202
実行キュー
203
実行キュー
204
core
205
core
206
core
207
core
208
L1キャッシュ
209
L1キャッシュ
210
L1キャッシュ
211
L1キャッシュ
212
L2キャッシュ
213
L2キャッシュ
214
L2キャッシュ
215
L2キャッシュ
216
L3キャッシュ
217
104
実行プログラムリスト
219
アプリ実行要求
プログラム識別値

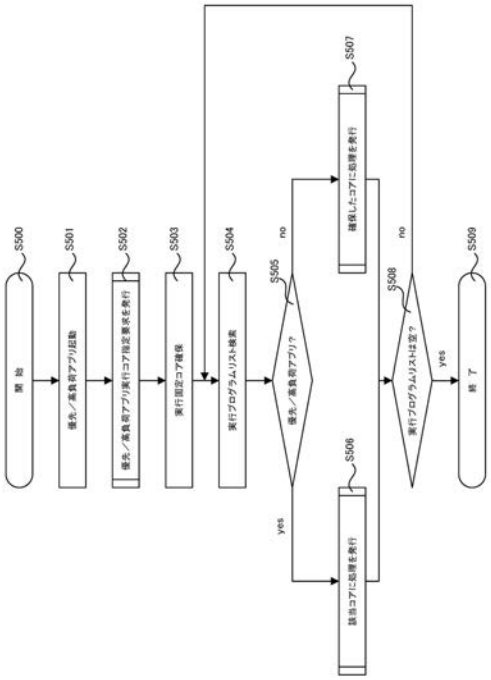
【図３】


300
310
コピーアプリ
310a
プリンタアプリ
310b
スキャナアプリ
310c
FAXアプリ
310d
HDD消去アプリ
310e
実行コア指定部
320a
OS
320
BIOS／モニタプログラム
330
120

【図４】


400
core
205
core
206
core
207
core
208
1-a
1-b
1-c
2-a
2-b
3-a
4-a
4-b
コア移動
コピーアプリ
310a
スキャナアプリ
310c
その他アプリ
(a)
(b)

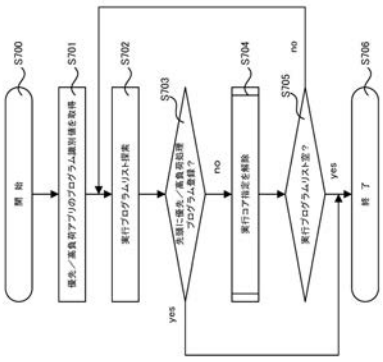
【図５】


開始
優先／高負荷アプリ起動
優先／高負荷アプリ実行コア指定要求を発行
実行指定コア確保
実行プログラムリスト検索
優先／高負荷アプリ？
no
yes
確保したコアに処理を発行
該当コアに処理を発行
実行プログラムリストは空？
no
yes
終了
S500
S501
S502
S503
S504
S505
S506
S507
S508
S509

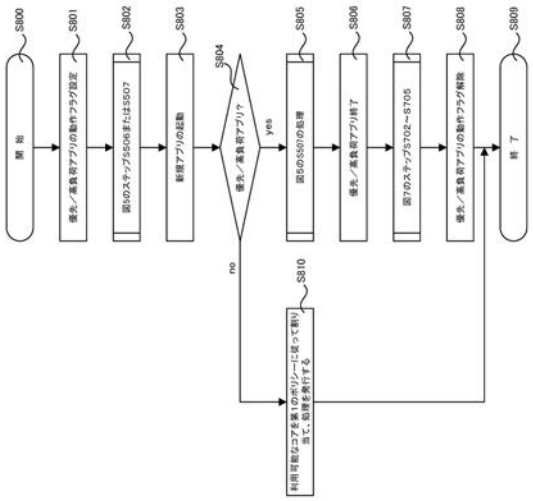
【図６】


600
プログラム識別値
実行させるコア
HDDデータ消去
1
プリンタ
2
コピー
3
スキャナ
4

【図７】


開始
優先／高負荷アプリのプログラム識別値を取得
実行プログラムリスト検索
実装に優先／高負荷処理プログラム登録？
yes
no
実行コア指定を解除
実行プログラムリスト完？
no
yes
終了
S700
S701
S702
S703
S704
S705
S706

【図８】


開始
優先／高負荷アプリの動的フラグ設定
図5のステップS506または図5507
新規アプリの起動
優先／高負荷アプリ？
no
yes
図6のS801の処理
優先／高負荷アプリ終了
図7のステップS702～S705
優先／高負荷アプリの動的フラグ解除
終了
利用可能なコアを第1のポリシーに従って割り当て、処理を実行する
S800
S801
S802
S803
S804
S805
S806
S807
S808
S809
S810