

(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2009-70208
(P2009-70208A)

(43) 公開日 平成21年4月2日(2009.4.2)

(51) Int.Cl.			F I	テーマコード (参考)		
G06F	12/08	(2006.01)	G06F 12/08	523C	5B005	
H04N	1/00	(2006.01)	H04N 1/00	C	5B060	
G06F	9/50	(2006.01)	G06F 9/46	462A	5C062	
G06F	12/10	(2006.01)	G06F 12/10	553B		
G06F	12/00	(2006.01)	G06F 12/10	559		

審査請求 未請求 請求項の数 6 O L (全 16 頁) 最終頁に続く

(21) 出願番号 特願2007-238984 (P2007-238984)
(22) 出願日 平成19年9月14日 (2007.9.14)

(71) 出願人 00006747
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 100089118
弁理士 酒井 宏明
(72) 発明者 矢崎 洋平
東京都大田区中馬込1丁目3番6号 株式会社リコー内
Fターム(参考) 5B005 JJ22 LL15 MM31 RR01
5B060 AA08 AC13
5C062 AA02 AA05 AB42 AC22 AF00
AF14

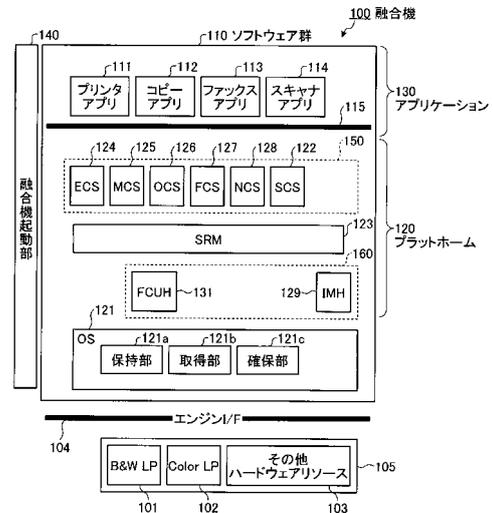
(54) 【発明の名称】 画像処理装置およびメモリ確保方法

(57) 【要約】

【課題】メモリの使用効率を改善する画像処理装置およびメモリ確保方法を提供すること。

【解決手段】データを記憶する物理メモリと、搭載可能な物理メモリの容量と、アプリケーションおよび制御手段の少なくとも一方が管理する仮想メモリ領域にマッピングする物理メモリ内の領域である直接マップ領域の容量とを対応づけた容量情報を保持する保持部121aと、保持された容量情報から、搭載されている物理メモリの容量に対応する直接マップ領域の容量を取得する取得部121bと、取得された容量の直接マップ領域を、搭載されている物理メモリ内に確保する確保部121cとを備えた。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

ハードウェア資源を利用した画像処理にかかるアプリケーションと、オペレーティングシステムと、前記オペレーティングシステム上で動作し、複数の前記アプリケーションからアクセスされて前記ハードウェア資源の制御を行う制御手段とを備えた画像処理装置であって、

データを記憶する物理メモリと、

搭載可能な前記物理メモリの容量と、前記アプリケーションおよび前記制御手段の少なくとも一方が管理する仮想メモリ領域にマッピングする前記物理メモリ内の領域である直接マップ領域の容量とを対応づけた容量情報を保持する保持手段と、

前記保持手段に保持された前記容量情報から、搭載されている前記物理メモリの容量に対応する前記直接マップ領域の容量を取得する取得手段と、

取得された前記容量の前記直接マップ領域を、搭載されている前記物理メモリ内に確保する確保手段と、

を備えたことを特徴とする画像処理装置。

【請求項 2】

前記保持手段は、複数の異なる前記容量情報と、前記容量情報のそれぞれを識別する識別子とを対応づけて保持し、

複数の前記容量情報のうち、指定された前記容量情報の前記識別子を記憶する識別子記憶手段をさらに備え、

前記取得手段は、前記識別子記憶手段に記憶されている前記識別子に対応する前記容量情報から、搭載されている前記物理メモリの容量に対応する前記直接マップ領域の容量を取得すること、

を特徴とする請求項 1 に記載の画像処理装置。

【請求項 3】

前記識別子記憶手段に記憶されている前記識別子を、ユーザの指定に応じて他の前記識別子に変更する変更手段をさらに備えたこと、

を特徴とする請求項 2 に記載の画像処理装置。

【請求項 4】

前記確保手段は、搭載されている前記物理メモリの予め定められたアドレスを先頭アドレスとして、取得した前記容量の前記直接マップ領域を、搭載されている前記物理メモリ内に確保すること、

を特徴とする請求項 1 に記載の画像処理装置。

【請求項 5】

前記確保手段は、前記物理メモリ内の前記直接マップ領域以外の領域に、前記オペレーティングシステムが管理する仮想メモリ領域にマッピングする前記物理メモリ内の領域である汎用領域を確保すること、

を特徴とする請求項 1 に記載の画像処理装置。

【請求項 6】

ハードウェア資源を利用した画像処理にかかるアプリケーションと、オペレーティングシステムと、前記オペレーティングシステム上で動作し、複数の前記アプリケーションからアクセスされて前記ハードウェア資源の制御を行う制御手段とを備えた画像処理装置で行われるメモリ確保方法であって、

前記画像処理装置は、

データを記憶する物理メモリと、

搭載可能な前記物理メモリの容量と、前記アプリケーションおよび前記制御手段の少なくとも一方が管理する仮想メモリ領域にマッピングする前記物理メモリ内の領域である直接マップ領域の容量とを対応づけた容量情報を保持する保持手段と、を備え、

取得手段が、前記保持手段に保持された前記容量情報から、搭載されている前記物理メモリの容量に対応する前記直接マップ領域の容量を取得する取得ステップと、

10

20

30

40

50

確保手段が、取得された前記容量の前記直接マップ領域を、搭載されている前記物理メモリ内に確保する確保ステップと、

を備えたことを特徴とするメモリ確保方法。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、プロセスが直接マッピングするメモリ領域を確保する画像処理装置およびメモリ確保方法に関するものである。

【背景技術】

【0002】

近年、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像処理装置が一般的に知られている。このような複合型の画像処理装置は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応する3種類のアプリケーションを設け、アプリケーションの切替えによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させる。

【0003】

このような従来の複合型の画像処理装置は、プリンタ、コピーおよびファクシミリにそれぞれ対応する別個のアプリケーション(OS(オペレーティングシステム)を含む)を備えている。そして、通常、各アプリケーションを実行して生成されるプロセスが必要とするメモリはそれぞれ別個に獲得され、また、個別に管理される。

【0004】

一方、このような複合型の画像処理装置では、例えば、読込んだ画像データをコピーすると同時にファクシミリ送信する場合のように、同一のデータを複数のアプリケーションのプロセスで利用する場合がある。このため、各アプリケーションのプロセスで共通の画像データ等を利用可能とするためのさまざまなメモリ管理技術が提案されている。

【0005】

例えば、特許文献1では、画像形成処理を行うアプリケーションのプロセス等によって直接マップ可能な直接マップ領域を獲得する技術が提案されている。これにより、多種の画像処理機能それぞれに対応するアプリケーションのプロセス間でのデータの共有が可能となる。

【0006】

また、例えば、特許文献2では、直接マップ領域用の物理アドレスと、直接マップ領域用の仮想アドレスとを対応づけておくことにより、アドレス変換時の例外エラーの発生を減少させ、直接マップ領域に対するメモリアクセスのオーバーヘッドを軽減する技術が提案されている。この方法では、連続する仮想アドレスを連続する物理アドレスに割り当てるため、ダイレクト・メモリ・アクセスを利用した場合のメモリアクセスの際に発生するオーバーヘッドを軽減しつつ、メモリの使用効率を向上させることができる。

【0007】

昨今では、ユーザの使用形態に応じて画像処理装置の機器構成を変更可能とすることが求められている。このため、例えば、画像処理装置に搭載されるメモリのメモリサイズが使用形態によってそれぞれ異なる場合がある。また、搭載されるアプリケーション構成を動的に変更するシステムも存在する。さらに、アプリケーションの機能拡張・規模増大に伴い、使用するメモリサイズも増大化の傾向にある。

【0008】

【特許文献1】特開2003-337714号公報

【特許文献2】特開2003-316646号公報

【発明の開示】

【発明が解決しようとする課題】

【0009】

10

20

30

40

50

しかしながら、特許文献 1 および 2 の方法では、直接マップ領域のサイズとして事前に定められた固定サイズを割り当てるため、画像処理装置の使用形態によってはメモリの使用効率が悪化する場合があった。例えば、利用されるアプリケーションに対して十分なメモリサイズのメモリを搭載しているにもかかわらず、直接マップ領域に割り当てられるメモリサイズが過大であるため、実際にアプリケーションが利用可能なメモリサイズが不足する場合があった。

【 0 0 1 0 】

本発明は、上記に鑑みてなされたものであって、装置に搭載されるメモリのサイズに応じて特定プロセス用に割り当てるメモリ領域のサイズを変更可能とすることにより、メモリの使用効率を改善することができる画像処理装置およびメモリ確保方法を提供することを目的とする。

10

【課題を解決するための手段】

【 0 0 1 1 】

上述した課題を解決し、目的を達成するために、請求項 1 にかかる発明は、ハードウェア資源を利用した画像処理にかかるアプリケーションと、オペレーティングシステムと、前記オペレーティングシステム上で動作し、複数の前記アプリケーションからアクセスされて前記ハードウェア資源の制御を行う制御手段とを備えた画像処理装置であって、データを記憶する物理メモリと、搭載可能な前記物理メモリの容量と、前記アプリケーションおよび前記制御手段の少なくとも一方が管理する仮想メモリ領域にマッピングする前記物理メモリ内の領域である直接マップ領域の容量とを対応づけた容量情報を保持する保持手段と、前記保持手段に保持された前記容量情報から、搭載されている前記物理メモリの容量に対応する前記直接マップ領域の容量を取得する取得手段と、取得された前記容量の前記直接マップ領域を、搭載されている前記物理メモリ内に確保する確保手段と、を備えたことを特徴とする。

20

【 0 0 1 2 】

また、請求項 2 にかかる発明は、請求項 1 にかかる発明において、前記保持手段は、複数の異なる前記容量情報と、前記容量情報のそれぞれを識別する識別子とを対応づけて保持し、複数の前記容量情報のうち、指定された前記容量情報の前記識別子を記憶する識別子記憶手段をさらに備え、前記取得手段は、前記識別子記憶手段に記憶されている前記識別子に対応する前記容量情報から、搭載されている前記物理メモリの容量に対応する前記

30

【 0 0 1 3 】

また、請求項 3 にかかる発明は、請求項 2 にかかる発明において、前記識別子記憶手段に記憶されている前記識別子を、ユーザの指定に応じて他の前記識別子に変更する変更手段をさらに備えたこと、を特徴とする。

【 0 0 1 4 】

また、請求項 4 にかかる発明は、請求項 1 にかかる発明において、前記確保手段は、搭載されている前記物理メモリの予め定められたアドレスを先頭アドレスとして、取得した前記容量の前記直接マップ領域を、搭載されている前記物理メモリ内に確保すること、を特徴とする。

40

【 0 0 1 5 】

また、請求項 5 にかかる発明は、請求項 1 にかかる発明において、前記確保手段は、前記物理メモリ内の前記直接マップ領域以外の領域に、前記オペレーティングシステムが管理する仮想メモリ領域にマッピングする前記物理メモリ内の領域である汎用領域を確保すること、を特徴とする。

【 0 0 1 6 】

また、請求項 6 にかかる発明は、ハードウェア資源を利用した画像処理にかかるアプリケーションと、オペレーティングシステムと、前記オペレーティングシステム上で動作し、複数の前記アプリケーションからアクセスされて前記ハードウェア資源の制御を行う制御手段とを備えた画像処理装置で行われるメモリ確保方法であって、前記画像処理装置は

50

、データを記憶する物理メモリと、搭載可能な前記物理メモリの容量と、前記アプリケーションおよび前記制御手段の少なくとも一方が管理する仮想メモリ領域にマッピングする前記物理メモリ内の領域である直接マップ領域の容量とを対応づけた容量情報を保持する保持手段と、を備え、取得手段が、前記保持手段に保持された前記容量情報から、搭載されている前記物理メモリの容量に対応する前記直接マップ領域の容量を取得する取得ステップと、確保手段が、取得された前記容量の前記直接マップ領域を、搭載されている前記物理メモリ内に確保する確保ステップと、を備えたことを特徴とする。

【発明の効果】

【0017】

本発明によれば、装置に搭載されるメモリのサイズに応じて特定プロセス用に割り当てるメモリ領域のサイズを変更可能とすることにより、メモリの使用効率を改善することができるという効果を奏する。

10

【0018】

また、本発明によれば、搭載メモリサイズに応じて特定プロセス用に割り当てるメモリ領域のサイズを、複数のサイズの候補から指定可能とすることにより、使用形態に応じてメモリの使用効率をさらに改善することができるという効果を奏する。

【発明を実施するための最良の形態】

【0019】

以下に添付図面を参照して、この発明にかかる画像処理装置およびメモリ確保方法の最良な実施の形態を詳細に説明する。

20

【0020】

(第1の実施の形態)

第1の実施の形態にかかる画像処理装置は、搭載されているメモリの容量に応じて、プロセスが直接マッピング可能な領域である直接マップ領域の容量を変更するものである。

【0021】

以下では、一実施の形態として、画像処理装置であってコピー機能、ファクシミリ(FAX)機能、プリント機能、スキャナ機能及び入力画像(スキャナ機能による読み取り原稿画像やプリンタあるいはFAX機能により入力された画像)を配信する機能等を融合したいわゆるMFP(Multi Function Peripheral)と称される融合機100に適用した例を示す。

30

【0022】

図1は、第1の実施の形態にかかる融合機100の機能ブロック図である。図1に示すように、融合機100は、ハードウェア資源105と、ソフトウェア群110と、融合機起動部140とを含むように構成される。

【0023】

ハードウェア資源105は、白黒レーザープリンタ(B&W LP)101と、カラーレーザープリンタ(Color LP)102と、スキャナ、ファクシミリなどのその他ハードウェアリソース103などを有する。

【0024】

また、ソフトウェア群110は、UNIX(登録商標)などのオペレーティングシステム(以下、OSという)121上に起動されているプラットフォーム120とアプリケーション130とから構成される。

40

【0025】

融合機起動部140は、融合機100の電源投入時に先ず始めに実行され、プラットフォーム120やアプリケーション130を起動する。プラットフォーム120は、アプリケーション130からの処理要求を解釈して、ハードウェア資源の獲得要求を発生させる制御部である下記に示すコントロールサービス150と、一または複数のハードウェア資源の管理をおこない、コントロールサービス150からの獲得要求を調停するシステムリソースマネージャー(SRM(System Resource Manager)123)と、SRM123からの獲得要求に応じてハードウェア資源105の管理を行うハンドラ層160とを有する

50

。

【0026】

このコントロールサービス150は、複数のサービスモジュールにより形成され、具体的には、SCS (System Control Service) 122と、ECS ((Engine Control Service) 124と、MCS (Memory Control Service) 125と、OCS (Operationpanel Control Service) 126と、FCS (FAX Control Service) 127と、NCS (Network Control Service) 128とがある。なお、このプラットフォーム120は、あらかじめ定義された関数により前記アプリケーション130からの処理要求を受信可能とするアプリケーションプログラムインターフェース (API) 115を有する。

【0027】

OS121は、UNIX (登録商標) などのオペレーティングシステムであり、プラットフォーム120並びにアプリケーション130の各ソフトウェアをそれぞれプロセスとして並列実行する。オープンソースのUNIX (登録商標) を用いることにより、プログラムの安全性を確保できるとともに、ネットワーク対応可能となり、ソースコードの入手も容易となる。さらに、OS、TCP/IPのロイヤリティが不要であり、アウトソーシングも容易となる。

【0028】

OS121は、保持部121aと、取得部121bと、確保部121cと、を備えている。保持部121aは、搭載可能なメモリの容量と、直接マップ領域の容量とを対応づけた容量情報を保持するものである。図2は、第1の実施の形態の容量情報のデータ構造の一例を示す説明図である。図2に示すように、容量情報は、搭載メモリサイズと、直接マップ領域サイズとを含んでいる。同図では、例えば搭載メモリサイズが128MB (メガバイト) のときは、直接マップ領域として12MBを確保することを意味する容量情報の例が示されている。

【0029】

取得部121bは、搭載されているメモリのメモリサイズを取得するとともに、取得したメモリサイズに対応する直接マップ領域サイズを、容量情報を参照して取得するものである。確保部121cは、取得部121bによって取得されたサイズの直接マップ領域を、メモリ内に確保するものである。また、確保部121cは、物理メモリ内の直接マップ領域以外の領域を対象として、汎用的に管理される汎用領域を物理メモリ内に確保する。

【0030】

SRM123は、SCS122とともにシステムの制御およびリソースの管理をおこなうものであり、スキャナやプロッタなどのエンジン部、メモリ、HDDファイル、ホストI/O (セントロI/F、ネットワークI/F、IEEE1394I/F、RS232CI/Fなど) のハードウェア資源を利用する上位層からの要求にしたがって調停をおこない、実行制御する。具体的には、このSRM123は、要求されたハードウェア資源が利用可能であるかどうか (他の要求により利用されていないかどうか) を判断し、利用可能であれば要求されたハードウェア資源が利用可能である旨を上位層に伝える。また、上位層からの要求に対してハードウェア資源の利用スケジューリングをおこない、要求内容 (たとえば、プリンタエンジンによる紙搬送と作像動作、メモリ確保、ファイル生成など) を直接実施するようにしてもよい。

【0031】

SCS122は、アプリ管理 (機能1)、操作部制御 (機能2)、システム画面表示 (ジョブリスト画面、カウンタ表示画面など) (機能3)、LED表示 (機能4)、リソース管理 (機能5)、割り込みアプリ制御 (機能6) 等の複数の機能を行なう。

【0032】

具体的には、アプリ管理 (機能1) では、アプリの登録と、その情報を他のアプリに通知する処理をおこなう。操作部制御 (機能2) では、アプリの操作部使用権の排他制御をおこなう。システム画面表示 (機能3) では、操作部使用権を持つアプリからの要求内容に応じて、エンジン部の状態に対応する警告画面の表示をおこなう。LED表示 (機能4

10

20

30

40

50

)では、警告LED、アプリキーなどのシステムLEDの表示制御をおこなう。リソース管理(機能5)では、アプリ(ECS)がジョブを実行するにあたって、排他しなければならないエンジンリソース(スキャナ、ステーブルなど)の排他制御のためのサービスをおこなう。割り込みアプリ制御(機能6)では、特定のアプリを優先動作させるための制御及びサービスをおこなう。

【0033】

ECS124は、白黒レーザプリンタ(B&W LP)101、カラーレーザプリンタ(Color LP)102、その他ハードウェアリソース103などのエンジン部を制御するものであり、画像読み込みと印刷動作、状態通知、ジャムリカバリなどをおこなう。

10

【0034】

MCS125は、メモリ制御をおこなうものであり、具体的には、画像メモリの取得および開放、ハードディスク装置(HDD)の利用、画像データの圧縮および伸張などをおこなう。

【0035】

OCS126は、オペレータと本体制御間の情報伝達手段となる操作パネルを制御するモジュールであり、オペレータのキー操作イベントを本体制御に通知する処理、各アプリがGUIを構築するためのライブラリ関数を提供する処理、構築されたGUI情報をアプリ別に管理する処理、操作パネル上への表示反映処理などをおこなう。

20

【0036】

FCS127は、システムコントローラの各アプリ層からPSTN/ISDN網を使ったファクシミリ送受信、BKM(バックアップSRAM)で管理されている各種ファクシミリデータの登録/引用、ファクシミリ読み取り、ファクシミリ受信印刷、融合送受信をおこなうためのAPI(Application Program Interface)を提供する。

【0037】

NCS128は、ネットワークI/Oを必要とするアプリケーションに対して共通に利用できるサービスを提供するためのモジュール群であり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介をおこなう。

30

【0038】

また、ハンドラ層160は、後述するFCU(FAX Control Unit)の管理を行うFCUH(FAX Control Unit Handler)131と、各プロセスに対するメモリの割り振り、各プロセスに割り振ったメモリの管理、直接マップ領域の管理などを行うIMH(Imaging Memory Handler)129とを含む。また、IMH129はイメージデータを仮想メモリ領域(ユーザ仮想空間)から物理メモリへマップする。プロセスの起動に応じて、システムコールを行ない、プロセス用の仮想メモリ領域をマップしたり、マップした仮想メモリ領域をプロセスの終了時に開放する処理等を行なう。IMH129は、後述されるCPUによって直接アクセス可能なMEM-P領域と、プロセスによって直接アクセス可能なMEM-P0領域へのマップを行なう。

40

【0039】

SRM123、FCUH131およびIMH129は、予め定義されている関数によりハードウェア資源105に対する処理要求を送信するエンジンI/F133を利用してハードウェア資源105に対する処理要求を行う。

【0040】

アプリケーション130は、ページ記述言語(PDL)、PCLおよびポストスクリプト(PS)を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキャナ用アプリケーションであるスキャナアプリ114とを有する。各アプリケーション111~114は、プラットフォーム120上の各プロセスを利用して動作実行し得るため、画面制御、キー操作制御およびジョブ生成などを

50

おこなう画面表示制御プログラムがその主体となる。なお、NCS 128により接続されたネットワークを介して新たなアプリケーションをネットワーク経由で搭載することもできる。また、各アプリケーションはアプリケーションごとに追加または削除することができる。

【0041】

このように、融合機100は、各アプリで共通的に必要となる処理をプラットフォーム120で一元的に処理する。

【0042】

次に、融合機100のハードウェア構成について説明する。図3は、図1に示した融合機100のハードウェア構成を示すブロック図である。図3に示すように、この融合機100は、オペレーションパネル310、ファックスコントロールユニット(FCU)320、USB(Universal Serial Bus)デバイス330、IEEE1394デバイス340およびエンジン部350とコントローラ300のASIC301とをPCI(Peripheral Component Interconnect)バス等で接続した構成となる。

10

【0043】

コントローラ300は、ASIC301にMEM-C302、HD303、ネットワークI/Fコントローラ309などを接続するとともに、このASIC301とCPU304とをCPUチップセットのNB305を介して接続している。このように、NB305を介して接続する理由は、CPU304自体のインターフェースが公開されていないためである。ここで、このASIC301とNB305は、単にPCIを介して接続されているのではなく、AGP308を介して接続されている。このようにAGP308を介して接続することとした理由は、この融合機100が図1に示したプラットフォーム120やアプリケーション130を形成する複数のプロセスを実行制御する関係上、これらを低速のPCIで接続したのでは、パフォーマンスが低下するからである。

20

【0044】

CPU304は、融合機100の全体制御をおこなうものであり、具体的には、OS121上でプラットフォーム120を形成するSCS122、SRM123、ECS124、MCS125、OCS126、FCS127、NCS128をそれぞれプロセスとして起動して実行させるとともに、アプリケーション130を形成するプリンタアプリ111、コピーアプリ112、ファックスアプリ113、スキャナアプリ114を起動して実行させる。

30

【0045】

NB305は、CPU304とMEM-P306a、NV-RAM306b、SB307、ASIC301とを接続するためのブリッジである。

【0046】

MEM-P306aは、融合機100の描画用メモリなどとして用いるシステムメモリであり、NV-RAM306bは、電源断時でもデータが消えない不揮発メモリであり、SB307は、NB305とROM、PCIデバイス、周辺デバイスとを接続するためのブリッジである。MEM-C302は、コピー用画像バッファ、符号バッファとして用いるローカルメモリであり、ASIC301は、画像処理用のハードウェア要素を有する画像処理用途向けのICである。

40

【0047】

HDD303は、画像データの蓄積、プログラムの蓄積、フォントデータの蓄積、フォームの蓄積を行うためのストレージであり、オペレーションパネル310は、操作者からの入力操作の受け付け並びに操作者に向けた表示をおこなう操作部である。

【0048】

したがって、ASIC301には、MEM-C302を接続するためのRAMインターフェースと、HDD303を接続するためのハードディスクインターフェースが設けられ、これらの記憶部に対して画像データの入出力をおこなう場合には、入出力先がRAMインターフェースまたはハードディスクインターフェースに切り替えられる。

50

【 0 0 4 9 】

A G P 3 0 8 は、グラフィック処理を高速化するために提案されたグラフィックスアクセラレーターカード用のバスインターフェースであり、システムメモリに高スループットで直接アクセスすることにより、グラフィックスアクセラレーターカードを高速にする。

【 0 0 5 0 】

U N I X (登録商標)等のOS 1 2 1上にプラットフォーム 1 2 0及びアプリケーション 1 3 0が構築される融合機 1 0 0の場合、仮想メモリ領域は、OS 1 2 1のカーネルによって解釈可能な仮想アドレス空間と、ソフトウェアによって生成されたプロセスによって解釈可能な仮想アドレス空間とによって制御されている。

【 0 0 5 1 】

そのため、OS 1 2 1独自で仮想メモリ領域(以下、カーネル用仮想メモリ領域とする)へのアクセスを制御すると共に、ソフトウェアによって生成されたプロセスによって展開される仮想メモリ領域(以下、プロセス毎仮想メモリ領域とする)へのアクセスを制御する必要がある。また、OS 1 2 1がカーネル用仮想メモリ領域からプロセス毎仮想メモリ領域をアクセスする際には、OS 1 2 1からアクセス可能な仮想メモリ領域に変換される。

【 0 0 5 2 】

先ず、MEM - C 3 0 2、MEM - P 3 0 6 a及びHDD 3 0 3等にて展開される物理メモリと仮想メモリ領域との対応について図4で説明する。図4を参照すると、物理メモリ 1 0には、カーネルが使用するカーネル領域 1 1と、OS 1 2 1で汎用的に管理される汎用領域 1 2と、MEM - P 3 0 6 aの領域の一部であって、アプリケーション 1 3 0又はコントロールサービス 1 5 0によって生成されるプロセスから仮想メモリ領域へマップされるMEM - P 0領域 1 3と、CPU 3 0 4から高速にアクセスされるMEM - P 3 0 6 aとしての実質的なMEM - P領域 1 4と、白黒レーザプリンタ(B & W LP) 1 0 1、カラーレーザプリンタ(Color LP) 1 0 2、その他ハードウェアリソース 1 0 3がエンジン I / F 1 3 3を介して高速にアクセス可能なMEM - C 3 0 2としての実質的なMEM - C領域 1 5とを有する。

【 0 0 5 3 】

領域 1 1から領域 1 4は、MEM - P 3 0 6 a側のRAMが使用され、領域 1 5は、MEM - C 3 0 2側のRAMが使用される。

【 0 0 5 4 】

この物理メモリ 1 0に対応させて展開される仮想メモリ領域 2 0は、OS 1 2 1によって汎用的に管理されるユーザプログラム領域 2 1と、プロセスが該プロセスの仮想メモリ領域 2 0から物理メモリ 1 0へ直接マップ可能な直接マップ領域 2 2と、OS 1 2 1によって汎用的に管理され必要に応じてデータがスタックされるスタック領域 2 3とを有する。

【 0 0 5 5 】

ユーザプログラム領域 2 1は、更に、アプリケーション 1 3 0をCPU 3 0 4が実行するためのプログラムコードを格納するプログラムコード領域 2 1 1と、アプリケーション 1 3 0をCPU 3 0 4が実行する際に必要となるプログラムデータを格納するプログラムデータ領域 2 1 2と、動的にデータ領域の割付けを行なうヒープ領域 2 1 3とを有する。

【 0 0 5 6 】

ユーザプログラム領域 2 1は、物理メモリ 1 0の汎用領域 1 2にマップされる。スタック領域 2 3もまた、物理メモリ 1 0の汎用領域 1 2にマップされる。直接マップ領域 2 2は、MEM - P 3 0 6 aの一部と、MEM - C 3 0 2とに対応する領域である。直接マップ領域 2 2は、物理メモリ 1 0のMEM - P 0領域 1 3、MEM - P領域 1 4及びMEM - C領域 1 5とに夫々対応するMEM - P 0領域 2 2 1、MEM - P領域 2 2 2及びMEM - C領域 2 2 3とを有する。

【 0 0 5 7 】

次に、このように構成された第1の実施の形態にかかる融合機 1 0 0によるメモリ確保

10

20

30

40

50

処理について説明する。図5は、第1の実施の形態におけるメモリ確保処理の全体の流れを示すフローチャートである。

【0058】

まず、取得部121bは、搭載可能なメモリの容量と直接マップ領域の容量とを対応づけた容量情報を保持部121aから取得する(ステップS501)。なお、容量情報は、カーネルビルド時に読み込まれるビルド時の設定情報を保存したconfigファイルから取込まれ、ビルドの結果生成されたOS内の保持部121aで保持される。

【0059】

次に、取得部121bは、容量情報を取得できたか否かを判断する(ステップS502)。容量情報が取得できた場合(ステップS502: YES)、取得部121bは、搭載メモリサイズを取得するシステムコールなどによって、融合機100に搭載されている物理メモリのメモリサイズを取得する(ステップS503)。

10

【0060】

次に、取得部121bは、取得した物理メモリのメモリサイズに対応する直接マップ領域サイズを、容量情報から取得する(ステップS504)。例えば、物理メモリのメモリサイズとして128MBを取得した場合は、取得部121bは、図2のような容量情報から、対応する直接マップ領域サイズとして12MBを取得する。

【0061】

ステップS502で、容量情報が取得できなかった場合(ステップS502: NO)、取得部121bは、直接マップ領域サイズのデフォルト値を設定する(ステップS505)。これは、例えば、configファイル内に容量情報に対応する情報が設定されていなかった場合等が該当する。取得部121bは、例えば、直接マップ領域サイズのデフォルト値としてカーネルのソースコードに記載され、ビルドの結果生成されたOS内に保持しているデフォルト値を取得する。

20

【0062】

次に、取得部121bは、取得した直接マップ領域サイズが搭載されている物理メモリのメモリサイズ以下であるか否かを判断する(ステップS506)。直接マップ領域サイズが物理メモリのメモリサイズ以下の場合は(ステップS506: YES)、確保部121cは、直接マップ領域サイズ分の領域を物理メモリ内に直接マップ領域として確保する(ステップS507)。次に、確保部121cは、確保した直接マップ領域以外の領域を対象として、汎用領域を確保する(ステップS508)。

30

【0063】

ステップS506で、取得した直接マップ領域サイズが搭載されている物理メモリのメモリサイズ以下でない場合は(ステップS506: NO)、直接マップ領域サイズの指定が不適切であるため、エラー処理を行ってメモリ確保処理を終了する(ステップS509)。

【0064】

このように、第1の実施の形態にかかる画像処理装置では、搭載されているメモリの容量に応じて、プロセスが直接マッピング可能な直接マップ領域の容量を変更することができる。これにより、搭載するメモリのメモリサイズを変更した場合であっても、直接マップ領域を適切なサイズに変更できるため、メモリの使用効率を改善することが可能となる。

40

【0065】

(第2の実施の形態)

第2の実施の形態にかかる画像処理装置は、搭載されているメモリの容量に応じた直接マップ領域の容量を、複数の容量の候補から指定可能とするものである。

【0066】

図6は、第2の実施の形態にかかる融合機600の機能ブロック図である。同図に示すように、融合機600は、ハードウェア資源105と、ソフトウェア群610と、融合機起動部140とを含むように構成される。

50

【 0 0 6 7 】

第 2 の実施の形態では、OS 6 2 1 内の保持部 6 2 1 a および取得部 6 2 1 b の構成と、ソフトウェア群 6 1 0 内にアプリケーション 6 3 0 の 1 つとして変更部 6 1 6 を追加したことから、第 1 の実施の形態と異なっている。その他の構成および機能は、第 1 の実施の形態にかかる融合機 1 0 0 の構成を表すブロック図である図 1 と同様であるので、同一符号を付し、ここでの説明は省略する。

【 0 0 6 8 】

保持部 6 2 1 a は、複数の異なる容量情報を保持可能とした点が、第 1 の実施の形態の保持部 1 2 1 a と異なっている。図 7 は、第 2 の実施の形態の容量情報のデータ構造の一例を示す説明図である。図 7 に示すように、第 2 の実施の形態の保持部 6 2 1 a は、容量情報を識別する識別情報としてのフラグと、各容量情報とを対応づけて保持している。同図では、例えば、フラグ = 1 および 2 の容量情報の例として、搭載メモリサイズが 1 2 8 MB のときに、それぞれ異なるサイズ (1 2 MB および 1 8 MB) の直接マップ領域サイズが指定されていることが示されている。

10

【 0 0 6 9 】

なお、図 6 では省略しているが、第 2 の実施の形態では、NV - RAM 3 0 6 b などの不揮発メモリに、複数の容量情報のうち適用する容量情報のフラグを記憶する。例えば、NV - RAM 3 0 6 b の予め定められたアドレスを当該フラグの設定用に用意し、後述する変更部 6 1 6 のみを変更できるように、このアドレスにフラグを記憶する。

【 0 0 7 0 】

取得部 6 2 1 b は、NV - RAM 3 0 6 b からフラグを読み込み、読み込んだフラグに対応する容量情報を参照して、物理メモリサイズに対応する直接マップ領域サイズを取得する。

20

【 0 0 7 1 】

変更部 6 1 6 は、NV - RAM 3 0 6 b の所定のアドレスに設定された容量情報のフラグを、ユーザによって変更可能とするためのアプリケーションである。変更部 6 1 6 は、例えば、NV - RAM 3 0 6 b の所定のアドレスからフラグを読み出し、オペレーションパネル 3 1 0 に変更可能に表示する。そして、変更部 6 1 6 は、オペレーションパネル 3 1 0 からユーザが指定したフラグによって、NV - RAM 3 0 6 b のフラグを更新する。このように、変更部 6 1 6 は、NV - RAM 3 0 6 b 上のフラグを書き換えるだけのものであり、使用する汎用領域の容量は、その他のアプリケーションと比較して十分に小さいものである。

30

【 0 0 7 2 】

次に、このように構成された第 2 の実施の形態にかかる融合機 6 0 0 によるメモリ確保処理について説明する。図 8 は、第 2 の実施の形態におけるメモリ確保処理の全体の流れを示すフローチャートである。

【 0 0 7 3 】

まず、取得部 6 2 1 b は、NV - RAM 3 0 6 b の所定のアドレスから容量情報のフラグを読み込む (ステップ S 8 0 1) 。次に、取得部 6 2 1 b は、読み込んだフラグに対応する容量情報を保持部 6 2 1 a から取得する (ステップ S 8 0 2) 。

40

【 0 0 7 4 】

ステップ S 8 0 3 からステップ S 8 1 0 までの、取得判定処理、サイズ取得処理、および領域確保処理は、第 1 の実施の形態にかかる融合機 1 0 0 におけるステップ S 5 0 2 からステップ S 5 0 9 までと同様の処理なので、その説明を省略する。

【 0 0 7 5 】

このように、第 2 の実施の形態にかかる画像処理装置では、搭載されているメモリの容量に応じた直接マップ領域の容量を、複数の容量の候補から指定することができる。これにより、融合機の使用形態等に応じて、起動時に獲得する汎用領域と直接マップ領域のサイズを切替えることができる。すなわち、使用形態等に応じてメモリの使用効率をさらに改善することが可能となる。

50

【産業上の利用可能性】

【0076】

以上のように、本発明にかかる画像処理装置およびメモリ確保方法は、プロセスから物理メモリ内の領域を直接マップ可能なメモリ管理方法を採用した装置に適している。

【図面の簡単な説明】

【0077】

【図1】第1の実施の形態にかかる融合機の機能ブロック図である。

【図2】第1の実施の形態の容量情報のデータ構造の一例を示す説明図である。

【図3】図1に示した融合機のハードウェア構成を示すブロック図である。

【図4】物理メモリと仮想メモリ領域との対応例を示す図である。

10

【図5】第1の実施の形態におけるメモリ確保処理の全体の流れを示すフローチャートである。

【図6】第2の実施の形態にかかる融合機の機能ブロック図である。

【図7】第2の実施の形態の容量情報のデータ構造の一例を示す説明図である。

【図8】第2の実施の形態におけるメモリ確保処理の全体の流れを示すフローチャートである。

【符号の説明】

【0078】

- 100 融合機
- 101 B & W LP
- 102 Color LP
- 103 その他ハードウェアリソース
- 105 ハードウェア資源
- 110 ソフトウェア群
- 111 プリンタアプリ
- 112 コピーアプリ
- 113 ファックスアプリ
- 114 スキャナアプリ
- 120 プラットフォーム
- 121 a 保持部
- 121 b 取得部
- 121 c 確保部
- 130 アプリケーション
- 140 融合機起動部
- 150 コントロールサービス
- 160 ハンドラ層
- 300 コントローラ
- 301 ASIC
- 302 MEM - C
- 303 HDD
- 304 CPU
- 305 NB
- 306 a MEM - P
- 306 b NV - RAM
- 307 SB
- 308 AGP
- 309 ネットワークI/Fコントローラ
- 310 オペレーションパネル
- 320 FCU
- 330 USBデバイス

20

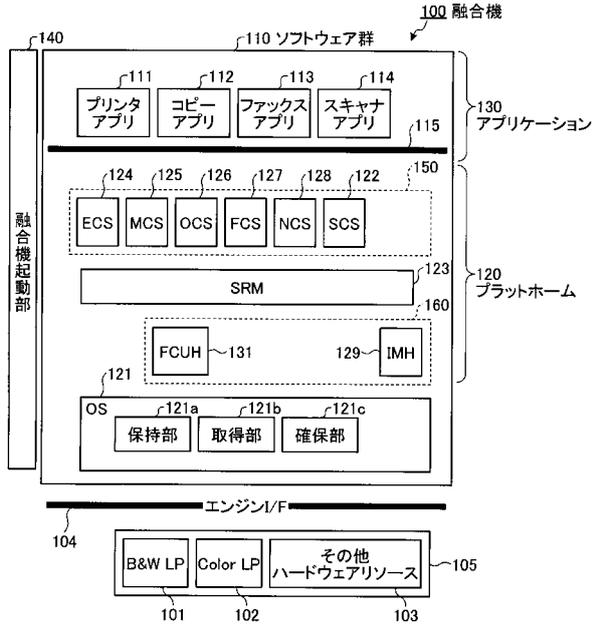
30

40

50

3 4 0	IEEE 1394 デバイス	
3 5 0	エンジン部	
1 0	物理メモリ	
1 1	カーネル領域	
1 2	汎用領域	
1 3	MEM - P 0 領域	
1 4	MEM - P 領域	
1 5	MEM - C 領域	
2 0	仮想メモリ領域	
2 1	ユーザプログラム領域	10
2 2	直接マップ領域	
2 3	スタック領域	
2 1 1	プログラムコード領域	
2 1 2	プログラムデータ領域	
2 1 3	ヒープ領域	
2 2 1	MEM - P 0 領域	
2 2 2	MEM - P 領域	
2 2 3	MEM - C 領域	
6 0 0	融合機	
6 1 0	ソフトウェア群	20
6 1 6	変更部	
6 2 1 b	取得部	
6 2 1 a	保持部	
6 3 0	アプリケーション	

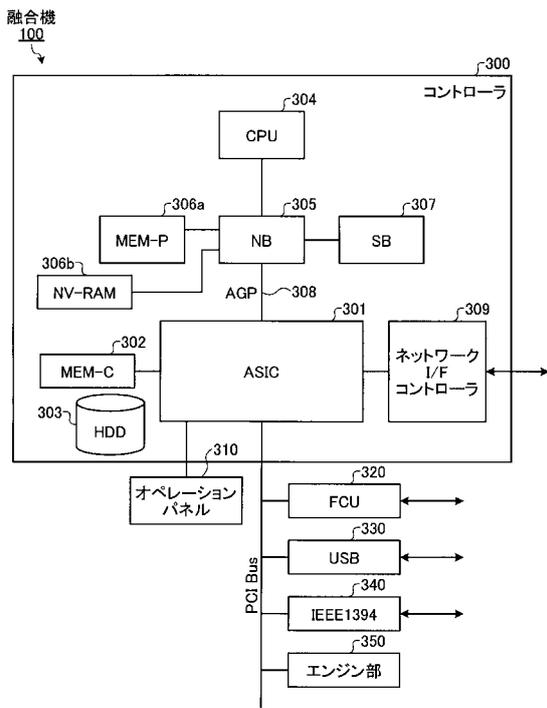
【 図 1 】



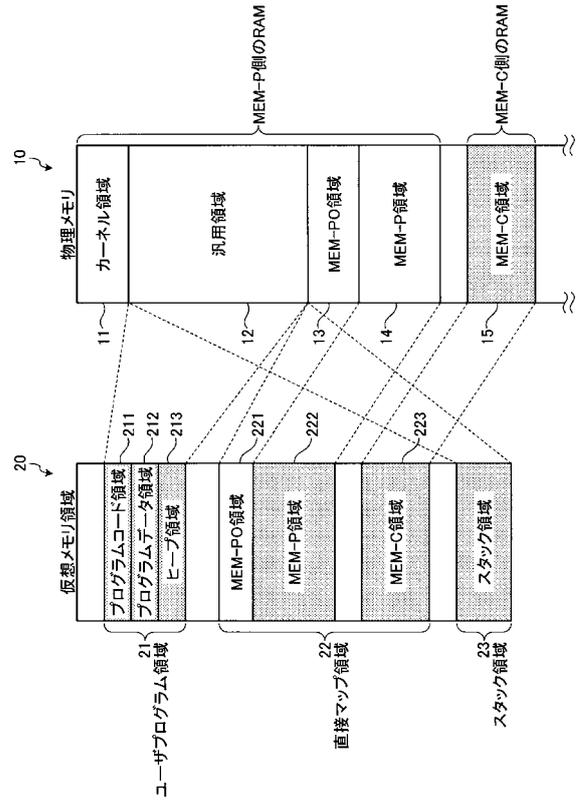
【 図 2 】

搭載メモリサイズ(MB)	直接マップ領域サイズ(MB)
128	12
384	36
⋮	⋮

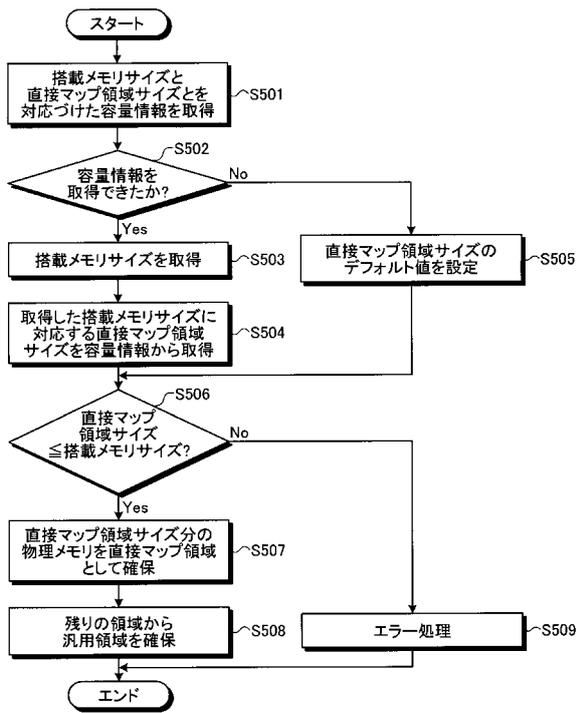
【 図 3 】



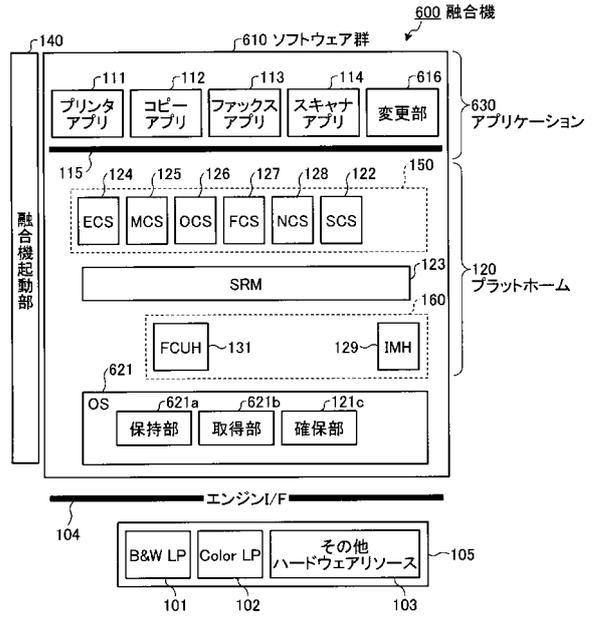
【 図 4 】



【図5】



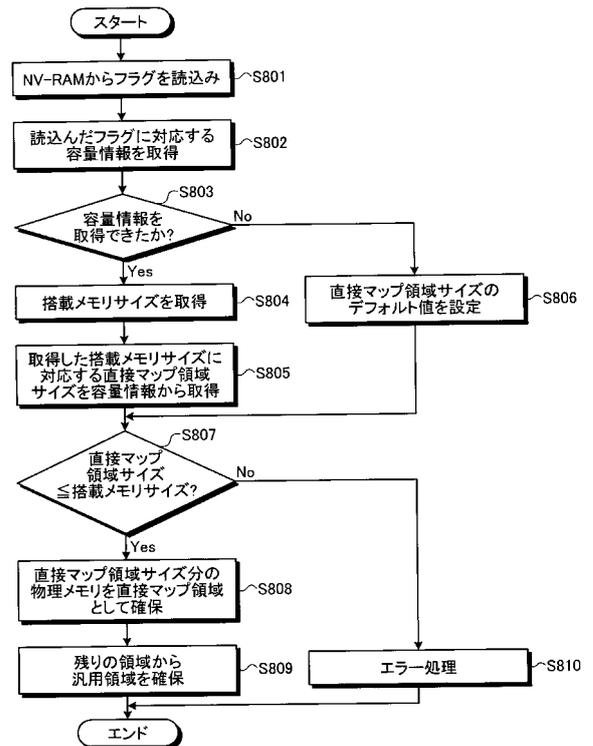
【図6】



【図7】

フラグ	搭載メモリサイズ(MB)	直接マップ領域サイズ(MB)
1	128	12
	384	36
	⋮	⋮
2	128	18
	384	54
	⋮	⋮
⋮	⋮	⋮

【図8】



フロントページの続き

(51) Int.Cl.		F I		テーマコード(参考)
G 0 6 F	12/02		(2006.01)	
		G 0 6 F	12/08	5 5 9 D
		G 0 6 F	12/00	5 8 0
		G 0 6 F	12/02	5 7 0 D