

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号  
特許第5962829号  
(P5962829)

(45) 発行日 平成28年8月3日 (2016. 8. 3)

(24) 登録日 平成28年7月8日 (2016. 7. 8)

(51) Int. Cl.	F I
HO 4 N 1/00 (2006. 01)	HO 4 N 1/00 C
GO 6 F 3/12 (2006. 01)	GO 6 F 3/12 3 7 9
B 4 1 J 29/38 (2006. 01)	GO 6 F 3/12 3 2 0
GO 3 G 21/00 (2006. 01)	B 4 1 J 29/38 Z
	GO 3 G 21/00 5 0 2

請求項の数 6 (全 18 頁)

(21) 出願番号	特願2015-171374 (P2015-171374)	(73) 特許権者	000006747
(22) 出願日	平成27年8月31日 (2015. 8. 31)		株式会社リコー
(62) 分割の表示	特願2014-253501 (P2014-253501) の分割		東京都大田区中馬込 1 丁目 3 番 6 号
原出願日	平成13年5月16日 (2001. 5. 16)	(74) 代理人	100089118 弁理士 酒井 宏明
(65) 公開番号	特開2016-27711 (P2016-27711A)	(72) 発明者	森田 哲也
(43) 公開日	平成28年2月18日 (2016. 2. 18)		東京都大田区中馬込 1 丁目 3 番 6 号 株式 会社リコー内
審査請求日	平成27年9月30日 (2015. 9. 30)		
(31) 優先権主張番号	特願2000-204257 (P2000-204257)		
(32) 優先日	平成12年7月5日 (2000. 7. 5)	審査官	橘爪 正樹
(33) 優先権主張国	日本国 (JP)		

最終頁に続く

(54) 【発明の名称】 画像形成装置、画像形成方法およびプログラム

(57) 【特許請求の範囲】

【請求項 1】

表示部、印刷部および撮像部を含む複数のハードウェア資源を有し、プリンタ、コピーまたはファクシミリの画像形成処理に係るユーザーサービスをそれぞれ提供する画像形成装置であって、

プリンタアプリケーション、コピーアプリケーション、及びファクシミリアプリケーションを少なくとも含む複数のアプリケーションと、

前記複数のアプリケーションそれぞれからの要求を受信するアプリケーションプログラムインターフェースを有し、該アプリケーションプログラムインターフェースを介して受信した要求に基づいて、前記ハードウェア資源の管理、実行制御、画像形成処理並びにジョブ生成を行うプラットフォームと、を有し、

前記プラットフォームは、

前記アプリケーションプログラムインターフェースを介して受信した要求に基づいたジョブ生成を行うアプリサービスと、

前記ハードウェア資源の管理、実行制御並びに画像形成処理を行う共通システムサービスであって、前記複数のアプリケーションがそれぞれ仮想的な画面メモリを複数持つことができる図面管理機能を有して当該複数のアプリケーションにそれぞれの画面を生成させて前記表示部に表示させるオペレーションパネルコントロールサービスを少なくとも含む共通システムサービスと、

を備えたことを特徴とする画像形成装置。

## 【請求項 2】

前記オペレーションパネルコントロールサービスは、複数の前記仮想的な画面メモリに生成した前記それぞれの画面のうち一の画面を前記表示部に表示させることを特徴とする請求項 1 に記載の画像形成装置。

## 【請求項 3】

前記オペレーションパネルコントロールサービスは、複数の前記仮想的な画面メモリに生成した前記それぞれの画面を合成した画面を前記表示部に表示させることを特徴とする請求項 1 に記載の画像形成装置。

## 【請求項 4】

前記複数の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ことを特徴とする請求項 1 ～ 3 のいずれか一つに記載の画像形成装置。

10

## 【請求項 5】

表示部、印刷部および撮像部を含む複数のハードウェア資源を用いて、プリンタ、コピーまたはファクシミリの画像形成処理に係るユーザーサービスをそれぞれ提供する画像形成装置で実行される画像形成方法であって、

前記画像形成装置は、プリンタアプリケーション、コピーアプリケーション、及びファクシミリアプリケーションを少なくとも含む複数のアプリケーションを有し、

前記複数のアプリケーションそれぞれからの要求を受信するアプリケーションプログラムインターフェースを有するプラットフォームによって、該アプリケーションプログラムインターフェースを介して受信した要求に基づいて、前記ハードウェア資源の管理、実行制御、画像形成処理並びにジョブ生成を行い、

20

前記プラットフォームは、

アプリサービスによって、前記アプリケーションプログラムインターフェースを介して受信した要求に基づいたジョブ生成を行い、

前記複数のアプリケーションがそれぞれ仮想的な画面メモリを複数持つことができる図面管理機能を有して当該複数のアプリケーションにそれぞれの画面を生成させて前記表示部に表示させるオペレーションパネルコントロールサービスを少なくとも含む共通システムサービスによって、前記ハードウェア資源の管理、実行制御並びに画像形成処理を行うことを特徴とする画像形成方法。

## 【請求項 6】

30

表示部、印刷部および撮像部を含む複数のハードウェア資源を有し、プリンタ、コピーまたはファクシミリの画像形成処理に係るユーザーサービスをそれぞれ提供し、プリンタアプリケーション、コピーアプリケーション、及びファクシミリアプリケーションを少なくとも含む複数のアプリケーションを有するコンピュータに、

前記複数のアプリケーションそれぞれからの要求を受信するアプリケーションプログラムインターフェースを有するプラットフォームによって、該アプリケーションプログラムインターフェースを介して受信した要求に基づいて、前記ハードウェア資源の管理、実行制御、画像形成処理並びにジョブ生成を行い、

前記プラットフォームは、

アプリサービスによって、前記アプリケーションプログラムインターフェースを介して受信した要求に基づいたジョブ生成を行い、

40

前記複数のアプリケーションがそれぞれ仮想的な画面メモリを複数持つことができる図面管理機能を有して当該複数のアプリケーションにそれぞれの画面を生成させて前記表示部に表示させるオペレーションパネルコントロールサービスを少なくとも含む共通システムサービスによって、前記ハードウェア資源の管理、実行制御並びに画像形成処理を行うことを実行させるためのプログラム。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

この発明は、表示部、印刷部および撮像部などのハードウェア資源を有し、プリンタ、

50

コピーまたはファクシミリなどの複合サービスをおこなう画像形成装置、画像形成方法およびプログラムに関し、特に、プリンタ、コピーおよびファクシミリ装置などに対応する各ソフトウェア（アプリケーション）を効率良く開発するとともに装置全体としての生産性を高めることができる画像形成装置、画像形成方法およびプログラムに関する。

【背景技術】

【0002】

従来、プリンタ、コピー、ファクシミリは、それぞれ別筐体として配設されるのが一般的であったが、最近では、これら各装置の機能を1つの筐体内に収納した画像形成装置（以下「複合機」と言う）が知られている。

【0003】

10

この複合機は、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種類のソフトウェアを設け、ソフトウェアの切り替えによって、該装置をプリンタ、コピーまたはファクシミリ装置として動作させるものである。

【0004】

かかる複合機を用いることにより、室内にプリンタ、コピーおよびファクシミリをそれぞれ別個に設ける必要がなくなるので、トータルな低コスト化および省スペース化を図ることができる。

【発明の概要】

【発明が解決しようとする課題】

20

【0005】

しかしながら、かかる複合機内にプリンタ、コピーおよびファクシミリ装置に対応するソフトウェア（専用OSを含む）をそれぞれ別個に設けることとしたのでは、各ソフトウェアの開発に多大の時間を要する。

【0006】

すなわち、かかる従来の複合機は、単に筐体を1つにまとめたものにすぎず、従来と同様に、プリンタ用ソフトウェア、コピー用ソフトウェアおよびファクシミリ用ソフトウェアを別個に開発せねばならない。

【0007】

もともと、プリンタのソフトウェア、コピーのソフトウェアおよびファクシミリのソフトウェアは、同種の画像を取り扱う性質上様々な面でアルゴリズムが共用できるため、ソフトウェアを別個に作成していたのでは、メモリ容量の累増などを招く問題もある。

30

【0008】

このため最近では、3種類の専用OSを別個に設けた場合の重複処理を省くために、UNIX（登録商標）などの汎用OSをかかると複合機に採用されることも多いが、単にOS部分を共通化するだけでは、ソフトウェアの開発効率をさほど高めることはできない。

【0009】

なお、特許文献1には、サーバダイアログおよび分散型アプリケーションなどからなるアプリケーション層と、フォントマネージャーやネットワークマネージャーなどからなる機能層と、オペレーティングシステムなどからなる制御層とで文書サービスアーキテクチャーを階層化する電子印刷システムが開示されているが、この従来技術のものは、フォントなどの一部の機能を共通化したものにすぎず、各ソフトウェアの開発効率を高めるものではない。

40

【0010】

また、複合機やプリンタなどに接続するパソコンには、複数のアプリケーションを搭載することができるが、このパソコンは、複合機に係るハードウェア資源を管理するものではない。本発明は、パソコンそのものではなく、パソコンが接続される複合機などの画像形成装置自体に関するものである。

【0011】

この発明は、上述した従来技術による問題点を解消するためになされたものであり、プ

50

リント、コピーおよびファクシミリ装置などに対応する各ソフトウェア（アプリケーション）を効率良く開発するとともに装置全体としての生産性を高めることができる画像形成装置、画像形成方法およびプログラム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】

【0012】

上述した課題を解決し、目的を達成するため、請求項1の発明に係る画像形成装置は、表示部、印刷部および撮像部を含む複数のハードウェア資源を有し、プリンタ、コピーまたはファクシミリの画像形成処理に係るユーザーサービスをそれぞれ提供する画像形成装置であって、プリンタアプリケーション、コピーアプリケーション、及びファクシミリアプリケーションを少なくとも含む複数のアプリケーションと、前記複数のアプリケーションそれぞれからの要求を受信するアプリケーションプログラムインターフェースを有し、該アプリケーションプログラムインターフェースを介して受信した要求に基づいて、前記ハードウェア資源の管理、実行制御、画像形成処理並びにジョブ生成を行うプラットフォームと、を有し、前記プラットフォームは、前記アプリケーションプログラムインターフェースを介して受信した要求に基づいたジョブ生成を行うアプリサービスと、前記ハードウェア資源の管理、実行制御並びに画像形成処理を行う共通システムサービスであって、前記複数のアプリケーションがそれぞれ仮想的な画面メモリを複数持つことができる画面管理機能を有して当該複数のアプリケーションにそれぞれの画面を生成させて前記表示部に表示させるオペレーションパネルコントロールサービスを少なくとも含む共通システムサービスと、を備えたことを特徴とする。

【0013】

この請求項1の発明によれば、各ユーザーサービスのアプリケーションを作成する場合には、画面表示制御やキー操作部分を作成すれば足りる。

【発明の効果】

【0038】

請求項1の発明によれば、各ユーザーサービスのアプリケーションを作成する場合には、画面表示制御やキー操作部分を作成すれば足りる。

【図面の簡単な説明】

【0051】

【図1】この発明の実施の形態に係る複合機概念を説明するための説明図である。

【図2】図1に示したプラットフォームを説明するための説明図である。

【図3】図1に示した複合機の具体的なソフトウェア構成を示す構成図である。

【図4】図3に示したコピーアプリを用いたコピー動作を説明するための説明図である。

【図5】図3に示したプリンタアプリを用いたプリント動作を説明するための説明図である。

【図6】図3に示したスキャナアプリを用いたスキャナ動作を説明するための説明図である。

【図7】コピーアプリ、プリンタアプリ、スキャナアプリの3つのアプリケーションの起動時処理と、それぞれのアプリ画面を生成し表示する処理と、アプリ画面選択キーが押下されたときの処理を説明するための説明図である。

【図8】ファックスアプリまたはプリンタアプリを用いたファックス送信動作を説明するための説明図である。

【図9】図1に示した複合機のハードウェア構成を示す構成図である。

【図10】図9に示したASICの細部構成を示すブロック図である。

【発明を実施するための形態】

【0052】

以下に添付図面を参照して、この発明に係る画像形成装置、画像形成方法およびプログラムの好適な実施の形態を詳細に説明する。なお、本実施の形態では、本発明を複合機に適用した場合を示すこととする。

【0053】

まず最初に、本実施の形態に係る複合機概念について図1および図2を用いて説明する。図1は、本実施の形態に係る複合機概念を説明するための説明図であり、図2は、図1に示したプラットフォームを説明するための説明図である。

【0054】

図1(a)に示すように、従来のプリンタ100は、描画・印刷機能並びにエンジン制御をおこなう専用OS101上にプリンタ用アプリケーション(プリンタアプリ)が搭載されていた。また、ファクシミリやコピーについても別個の筐体で構成されていた。また、これらを1つの筐体上にまとめる複合機も登場してきたが、単にプリンタ、コピーおよびファクシミリの機能を独立に設けたのでは効率的ではない。

【0055】

このため、同図(b)に示すように、従来の専用OS101を汎用OS部分111とエンジン制御部分112で形成し、これらをエンジンインターフェース(I/F)で連結するとともに、該汎用OS111上にプリンタアプリ113、コピーアプリ114および各種アプリ115を搭載する装置構成が採用されてきた。

【0056】

かかる複合機110では、たとえばUNIX(登録商標)などの汎用OSを採用しているので、プリンタアプリ113、コピーアプリ114、各種アプリ115をそれぞれプロセスとして簡単に並列実行することができる。

【0057】

しかしながら、この複合機110についても、プリンタアプリ113、コピーアプリ114、各種アプリ115が、それぞれ独立に開発される必要があるため、各ソフトウェアの開発負担をさほど軽減できるわけではない。

【0058】

そこで、本実施の形態に係る複合機120では、図1(c)に示すように、各アプリケーションの共通部分を共通システムサービス121aおよびアプリサービス121bとして括りだし、この共通システムサービス121a、アプリサービス121bおよび汎用OS111によりプラットフォーム122を形成する。

【0059】

特に、この複合機120では、アプリサービス121bが各アプリ123~125と共通システムサービス121aとの間に介在するよう構成することにより、本来各アプリ123~125がそれぞれ独立しておこなうべきジョブの生成やデータ通信をアプリサービス121bが一括しておこなうこととしている。このため、このプラットフォーム122上に搭載するプリンタアプリ123、コピーアプリ124および各種アプリ125の開発労力軽減並びにアプリケーションのスリム化を図ることができる。

【0060】

たとえば、図2(a)に示すように、コピーアプリが130,000ステップ、ファックスアプリが125,000ステップ、プリンタアプリが100,000ステップのコードからなる場合に、これらを別個のアプリケーションとして形成すると、合計で $130,000 + 125,000 + 100,000 = 355,000$ ステップのコードを要する。

【0061】

ここで、各アプリケーションに共通に利用できる部分が180,000ステップであるならば、この部分をプラットフォームとして集約することにより、生産性が向上する。

【0062】

たとえば、同図(b)に示すように、コピーアプリが40,000ステップ、ファックスアプリが100,000ステップ、プリンタアプリが35,000ステップ、プラットフォームが90,000ステップのコードで形成できるとすると、合計で $40,000 + 100,000 + 35,000 + 90,000 = 265,000$ ステップとなり、装置全体の生産性が $134\%$ ( $355,000 / 265,000$ )向上する。

【0063】

また、プラットフォーム部分を考えると、生産性が $200\%$ ( $180,000 / 90,000$ )向上し、さらにコピーアプリ、ファックスアプリ、プリンタアプリの開発効率についても大幅に

10

20

30

40

50

向上する。

【 0 0 6 4 】

このように、本実施の形態に係る複合機 1 2 0 では、アプリサービス 1 2 1 b、共通システムサービス 1 2 1 a および汎用 OS 1 1 1 からなるプラットフォーム 1 2 2 上に、プリンタアプリ 1 2 3、コピーアプリ 1 2 4 および各種アプリ 1 2 5 を搭載するように構成しているので、装置全体の生産性を高めるとともに、各アプリケーションの開発効率を高めることができる。

【 0 0 6 5 】

次に、図 1 に示した複合機 1 2 0 のソフトウェア構成についてさらに詳細に説明する。図 3 は、図 1 に示した複合機 1 2 0 の具体的なソフトウェア構成を示す構成図である。

10

【 0 0 6 6 】

同図に示すように、この複合機 1 2 0 は、白黒ラインプリンタ (B&W LP) 3 0 1、カラーラインプリンタ (Color LP) 3 0 2、その他ハードウェアリソース 3 0 3 などとを有するとともに、ソフトウェア群 3 1 0 は、プラットフォーム 3 2 0 およびアプリケーション 3 4 0 からなる。

【 0 0 6 7 】

プラットフォーム 3 2 0 は、汎用 OS 3 2 1 と、共通システムサービス 3 3 0 と、アプリサービス 3 2 9 とで形成される。なお、このプラットフォーム 3 2 0 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を有する。

20

【 0 0 6 8 】

汎用 OS 3 2 1 は、UNIX (登録商標) などの汎用オペレーティングシステムであり、プラットフォーム 3 2 0 並びにアプリケーション 3 4 0 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。

【 0 0 6 9 】

共通システムサービス 3 3 0 は、アプリケーション 3 4 0 に対して基本的な共通サービスを提供するものであり、アプリケーション 3 3 0 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下記に示すコントロールサービスと、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) 3 2 3 ) とを有する。

30

【 0 0 7 0 】

このコントロールサービスは、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) 3 2 2 と、ECS (Engine Control Service) 3 2 4 と、MCS (Memory Control Service) 3 2 5 と、OCS (Operation panel Control Service) 3 2 6 と、FCS (FAX Control Service) 3 2 7 と、NCS (Network Control Service) 3 2 8 とがある。

40

【 0 0 7 1 】

SRM 3 2 3 は、SCS 3 2 2 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、スキャナ部やプリンタ部などのエンジン、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、IEEE 1394 I/F、RS 232 C I/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。

【 0 0 7 2 】

具体的には、この SRM 3 2 3 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

50

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。

【 0 0 7 3 】

S C S 3 2 2 は、（ 1 ）アプリ管理、（ 2 ）操作部制御、（ 3 ）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）、（ 4 ）L E D 表示、（ 5 ）リソース管理、（ 6 ）割り込みアプリ制御をおこなう。具体的には、（ 1 ）アプリ管理では、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。登録されたアプリに対しては、システムの設定やアプリからの要求設定に応じてエンジン状態を通知する。また、登録済みのアプリに対しては、電力モード移行の問い合わせ、割り込みモードなど、システムの状態遷移のための可否問い合わせをおこなう。

10

【 0 0 7 4 】

また、（ 2 ）操作部制御では、アプリの操作部使用権の排他制御をおこなう。そして、操作部の使用権を持つアプリへ操作部ドライバ（O C S）からのキー情報を排他的に通知する。このキー情報は、アプリ切替中などのシステムの状態遷移に応じて一時的に通知を停止するマスク制御をおこなう。

【 0 0 7 5 】

また、（ 3 ）システム画面表示では、操作部使用権を持つアプリからの要求内容に応じて、エンジン状態に対応する警告画面の表示をおこなう。これらのなかには、利用者制限画面などアプリの状態に応じて警告表示をオン／オフするものもある。エンジン状態以外では、ジョブの予約・実行状況を表示するためのジョブリスト画面、トータルカウンタ類を表示するためのカウンタ画面、C S S の通報中を示す画面の表示制御をおこなう。これらのシステム画面表示に関しては、アプリへ操作部使用権の解放を要求せず、アプリ画面を覆うシステム画面として描画をおこなう。

20

【 0 0 7 6 】

また、（ 4 ）L E D 表示では、警告L E D、アプリキーなどのシステムL E D の表示制御をおこなう。アプリ固有のL E D については、アプリが直接表示用ドライバを使用して制御する。

【 0 0 7 7 】

また、（ 5 ）リソース管理では、アプリ（E C S）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い、（ 6 ）割り込みアプリ制御では、特定のアプリを優先動作させるための制御・サービスをおこなう。

30

【 0 0 7 8 】

E C S 3 2 4 は、白黒ラインプリンタ（B&W LP）3 0 1、カラーラインプリンタ（Color LP）3 0 2、その他ハードウェアリソース3 0 3などのエンジン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。

【 0 0 7 9 】

具体的には、アプリケーション3 4 0 から受け取ったジョブモードの指定にしたがい、印刷要求をS R M 3 2 3 に順次発行していくことで、一連のコピー／スキャン／印刷動作を実現する。このE C S 3 2 4 が取り扱う対象のジョブは、画像入力デバイスにスキャナ（SCANNER）が指定されているか、または、画像出力デバイスにプロッタ（PLOTTER）が指定されているものとする。

40

【 0 0 8 0 】

たとえば、コピー動作の場合には「SCANNER PLOTTER」と指定され、ファイル蓄積の場合には「SCANNER MEMORY」と指定され、ファクシミリ送信の場合には「SCANNER FAX\_\_IN」と指定される。また、蓄積ファイル印刷またはプリンタアプリ3 1 1からの印刷の場合には「MEMORY PLOTTER」と指定され、ファクシミリ受信の場合には「FAX\_\_OUT PLOTTER」と指定される。

【 0 0 8 1 】

50

なお、ジョブの定義はアプリケーションによって異なるが、ここでは利用者が扱う 1 セットの画像群に対する処理動作を 1 ジョブと定義する。たとえば、コピーの A D F ( Automatic Document Feeder ) モードの場合は、原稿台に置かれた 1 セットの原稿を読み取る動作が 1 ジョブとなり、圧板モードは最終原稿が確定するまでの読み取り動作が 1 ジョブとなる。また、コピーアプリ 3 1 2 の場合には、一束の原稿をコピーする動作が 1 ジョブとなり、ファックスアプリ 3 1 3 の場合には、1 文書の送信動作または 1 文書の受信動作が 1 ジョブとなり、プリンタアプリの場合には、1 文書の印刷動作が 1 ジョブとなる。

#### 【 0 0 8 2 】

M C S 3 2 5 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 ( H D D ) の利用、画像データの圧縮および伸張などをおこなう。

#### 【 0 0 8 3 】

ここで、ハードディスク装置に蓄積される画像データファイルとして必要な情報を管理するために必要な機能としては、( 1 ) ファイルアクセス ( 生成 / 削除 / オープン / クローズ ) 機能 ( 排他処理を含む )、( 2 ) ファイル名称 / I D 管理 ( ファイル / ユーザ ) / パスワード管理 / 蓄積時刻管理 / ページ数 / データフォーマット ( 圧縮方式など ) / アクセス制限 / 作成アプリ / 印刷条件管理などの各種ファイル属性管理 ( 物理的なページ単位の画像データのファイルとしての管理 )、( 3 ) ファイル単位およびページ単位での結合 / 挿入 / 切断機能、( 4 ) ファイルソート機能 ( 蓄積時刻順 / ユーザ I D 順など )、( 5 ) 全ファイル情報の通知 ( 表示 / 検索用 )、( 6 ) リカバリ機能 ( 破損ファイルのファイル / ページ破棄 )、( 7 ) ファイルの自動削除機能などがある。

#### 【 0 0 8 4 】

また、R A M などのメモリへ画像データを保持しアクセスするための機能としては、( 1 ) アプリケーション 3 4 0 からのファイルおよびページ / バンド属性情報を取得する機能、( 2 ) アプリケーション 3 4 0 からの画像データ領域の確保、解放、リード ( Read )、ライト ( Write ) 機能などがある。

#### 【 0 0 8 5 】

O C S 3 2 6 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 G U I 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 G U I 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。

#### 【 0 0 8 6 】

この O C S 3 2 6 は、( 1 ) G U I 構築のためのライブラリの提供機能、( 2 ) 操作部ハードウェア資源管理機能、( 3 ) V R A M 描画 / L C D 表示機能 ( ハードウェア表示、表示アプリ切替、表示言語切替、ウィンドウ暗色表示、メッセージ / アイコンブリンク表示、メッセージの連結表示 )、( 4 ) ハードキー入力検出機能、( 5 ) タッチパネルキー入力検出機能、( 6 ) L E D 出力機能、( 7 ) ブザー出力機能などを有する。

#### 【 0 0 8 7 】

F C S 3 2 7 は、システムコントローラの各アプリ層から P S T N / I S D N 網を使ったファクシミリ送受信、B K M ( バックアップ S R A M ) 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 A P I を提供するものである。

#### 【 0 0 8 8 】

具体的には、この F C S 3 2 7 は、( 1 ) アプリ層から送信依頼されたドキュメントを P S T N / I S D N 網を使ってファクシミリ受信機に送信をおこなう送信機能、( 2 ) P S T N / I S D N 網から受信したファクシミリ受信画面、各種レポート類を各アプリ層に転送、印刷をおこなう受信機能、( 3 ) ファックスボードに記憶されている電話帳、グループ情報などのファクシミリ管理項目の引用や登録をおこなう電話帳引用・登録機能、( 4 ) ファックスボードに搭載されている B K M に記憶されている送受信結果履歴情報など

10

20

30

40

50

を必要としているアプリに通知するファックスログ通知機能、(5)ファックスボードの状態変化があったときにFCSに登録してあるアプリに変化のあったイベントを通知するイベント通知機能などを有する。

【0089】

NCSS328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。具体的には、ftpd、httpd、lpd、snmpd、telnetd、smtpdなどのサーバデーモンや、同プロトコルのクライアント機能などを有する。

10

【0090】

アプリサービス329は、プラットフォーム320を形成する共通サービスの一つであるが、上記共通システムサービス330を形成するECS324、MCS325、OCS326、FCS327、NCSS328、SRM323およびSCS322とは異なり、アプリケーション340側に立ったサービスを提供するものである。

【0091】

言い換えると、このアプリサービス329は、アプリケーション340と共通システムサービス330との間に介在し、両者の間の橋渡しを担う役割を果たしている。

【0092】

具体的には、このアプリサービス329は、コピーアプリ312、ファックスアプリ313、スキャナアプリ314などが、本来おこなうべきジョブの生成やデータ通信の機能を一括して代行する。このため、コピーアプリ312、ファックスアプリ313、スキャナアプリ314などは、画面やキー操作を対象とすれば足りるので、アプリの開発効率が向上する。

20

【0093】

アプリケーション340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311と、コピー用アプリケーションであるコピーアプリ312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313と、スキャナ用アプリケーションであるスキャナアプリ314と、ネットファイル用アプリケーションであるネットファイルアプリ315と、工程検査用アプリケーションである工程検査アプリ316とを有する。

30

【0094】

各アプリケーション311～316は、プラットフォーム320上の各プロセスを利用して動作実行し得るため、画面制御およびキー操作制御などをおこなう画面表示制御プログラムがその主体となる。特に、アプリサービス329がプラットフォーム320上に設けられているので、ジョブの生成やデータ通信の機能を設ける必要がない。なお、NCSS328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。

【0095】

次に、図3に示したコピーアプリ312を用いたコピー動作、プリンタアプリ311を用いたプリント動作、スキャナアプリ314を用いたスキャナ動作についてさらに具体的に説明する。

40

【0096】

図4は、図3に示したコピーアプリ312を用いたコピー動作を説明するための説明図である。同図に示すように、コピーアプリ312はコピー画面・キー操作モジュール312aを有し、オペパネからコピー条件が指定され、スタートキーが押下されると、コピー画面・キー操作モジュール312aがアプリサービス329のアプリジョブ生成モジュール329aに対してコピー条件を転送する(ステップS401)。なお、このコピー条件には、紙サイズ、部数、両面、ソート、ステープルなどがある。

50

## 【 0 0 9 7 】

その後、アプリジョブ生成モジュール 3 2 9 a は、E C S 3 2 4 の A P I を用いてエンジンジョブ実行制御モジュール 3 2 4 a に対してエンジンジョブ生成関数を呼び出し、ジョブモードの設定をおこなう。なお、かかるジョブモードとは、スキャナ、プロッタ、フィニッシャなどを動作させるために必要なパラメータ群であり、上記コピー条件から生成される。さらに、アプリジョブ生成モジュール 3 2 9 a は、E C S 3 2 4 の A P I を用いてエンジンジョブ実行制御モジュール 3 2 4 a に対してジョブ実行開始関数を呼び出し、ジョブ実行を指示する（ステップ S 4 0 2 ）。

## 【 0 0 9 8 】

エンジンジョブ実行制御モジュール 3 2 4 a は、M C S 3 2 5 の A P I を用いて画像メモリハンドリングモジュール 3 2 5 a に対して画像メモリ確保関数を呼び出し、画像メモリの確保を指示する（ステップ S 4 0 3 ）。

## 【 0 0 9 9 】

画像メモリハンドリングモジュール 3 2 5 a は、S R M 3 2 3 のメモリ資源管理モジュール 3 2 3 b に対してメモリ取得関数を呼び出しメモリを取得する（ステップ S 4 0 4 ）。

エンジンジョブ実行制御モジュール 3 2 4 a は、S R M 3 2 3 のエンジン資源管理モジュールに対して資源取得関数を呼び出し、スキャナおよびプロッタの動作を開始し、コピージョブを実行する。

## 【 0 1 0 0 】

次に、図 3 に示したプリンタアプリ 3 1 1 を用いたプリント動作についてさらに具体的に説明する。図 5 は、図 3 に示したプリンタアプリ 3 1 1 を用いたプリント動作を説明するための説明図である。同図に示すように、プリンタアプリ 3 1 1 は、R I P 処理モジュール 3 1 1 a と、プリンタ画面・キー操作モジュール 3 1 1 b とを有する。

## 【 0 1 0 1 】

P C などのホストからセントロ I / F、U S B I / F、ネットワーク I / F などを経由して S R M 3 2 3 のホスト I / O 資源管理モジュール 3 2 3 d に印刷データコマンドが入力されると（ステップ S 5 0 1 ）、このホスト I / O 資源管理モジュール 3 2 3 d は、印刷データを N C S 3 2 8 のホスト通信 I / F 制御モジュール 3 2 8 a に転送する。ここで、このホスト通信 I / F 制御モジュール 3 2 8 a は、入力された印刷データを解析して送り先アプリケーションを判定する。ホスト通信 I / F 制御モジュール 3 2 8 a は、複数の通信プロトコルまたはホスト I / F からデータを受け取り、送り先を調停する（ステップ S 5 0 2 ）。

## 【 0 1 0 2 】

印刷データが、アプリサービス 3 2 9 内のアプリデータ通信モジュール 3 2 9 b に転送されると（ステップ S 5 0 3 ）、さらにアプリジョブ生成モジュール 3 2 9 a に入力され、新たなプリントジョブが生成される（ステップ S 5 0 4 ）。

## 【 0 1 0 3 】

アプリジョブ生成モジュール 3 2 9 a は、印刷データの言語種類に応じてプリンタアプリ 3 1 1 の R I P 処理モジュール 3 1 1 a にデータを転送し（ステップ S 5 0 5 ）、この R I P 処理モジュール 3 1 1 a は、印刷データの画像展開をおこなった後に、アプリジョブ生成モジュール 3 2 9 a に戻される（ステップ S 5 0 6 ）。

## 【 0 1 0 4 】

そして、このアプリジョブ生成モジュール 3 2 9 a で生成された画像データは、M C S 3 2 5 の画像メモリハンドリングモジュール 3 2 5 a に出力される（ステップ S 5 0 7 ）。

画像メモリには複数のアプリからの要求が発生するため、メモリ資源の調停をおこなう S R M 3 2 3 のメモリ資源管理モジュール 3 2 3 b に対してメモリ確保要求を発行し、取得した後に展開された画像データがメモリ上に書き込まれる（ステップ S 5 0 8 ）。

## 【 0 1 0 5 】

画像データの印刷準備が完了すると、アプリジョブ生成モジュール 3 2 9 a が E C S 3 2 4 の A P I を用いてエンジンジョブ実行制御モジュール 3 2 4 a に対してエンジンジョ

ブ生成関数を呼び出し、ジョブモード設定をおこなった後に、ジョブ開始関数を呼び出し、ジョブ実行を指示する（ステップS509）。

【0106】

エンジンジョブ実行制御モジュール324aは、SRM323のエンジン資源管理モジュール323aに対してエンジン資源取得関数を呼び出し、プロッタの動作を開始し、プリントジョブを実行する（ステップS510）。

【0107】

次に、図3に示したスキャナアプリ314を用いたスキャナ動作についてさらに具体的に説明する。図6は、図3に示したスキャナアプリ314を用いたスキャナ動作を説明するための説明図である。同図に示すように、スキャナアプリ314は、スキャナ画面・キー操作モジュール314aを有する。

【0108】

オペパネからスキャナアプリ314を選択するとともにスキャン条件が入力され、スタートキーが押下されると、スキャナ画面・キー操作モジュール314aがアプリサービス329のアプリジョブ生成モジュール329aに対してスキャン条件を転送すると（ステップS601）、アプリジョブ生成モジュール329aは、要求された画像サイズにしたがってスキャンに必要なメモリを確保するために、MCS325の画像メモリハンドリングモジュール325aが、メモリ資源の調停をおこなうSRM323のメモリ資源管理モジュール323bに対してメモリ確保要求を発行し、メモリが取得された後に画像データがメモリ上に書き込まれる（ステップS602～S603）。

【0109】

画像データのスキャン準備が完了すると、アプリジョブ生成モジュール329aが、ECS324のエンジンジョブ実行制御モジュール324aに対してエンジンジョブ生成関数を呼び出し、ジョブモード設定をおこなった後にジョブ開始関数を呼び出し、スキャンジョブ実行を指示する（ステップS604）。

【0110】

エンジンジョブ実行制御モジュール324aは、SRM323のエンジン資源管理モジュール323aに対してエンジン資源取得関数を呼び出し、スキャナの動作を開始し、スキャナジョブを実行する（ステップS605）。

【0111】

アプリジョブ生成モジュール329aが、ECS324からスキャン完了のイベントを受け取ると、読み取り画像をネットワーク他のホストI/O資源を利用して外部のPCに転送をおこなうため、アプリデータ通信モジュール329bがNCS328のホスト通信I/F制御モジュール328aに対してURLなどの転送先PCのアドレスおよびスキャンした画像データ（またはメモリ上のアドレス）を送る（ステップS606～S607）。ここでは、ftpやhttpなどのファイル転送をおこなうプロトコル機能を使って画像データ通信を制御することとする。

【0112】

ホスト通信I/F制御モジュール328aは、ネットワーク、IEEE1394、USB、SCSIなどのホストI/O資源を確保するために、SRM323のホストI/O資源管理モジュール323dの資源確保関数を呼び出す（ステップS608）。ホストI/O資源管理モジュール323dは、要求されたホストI/O資源が利用可能となった時点で画像データ通信をおこなう（ステップS609）。

【0113】

次に、コピーアプリ312、プリンタアプリ311、スキャナアプリ314の3つのアプリケーションの起動時処理と、それぞれのアプリ画面を生成し表示する処理と、アプリ画面選択キーが押下されたときの処理について説明する。図7は、コピーアプリ312、プリンタアプリ311、スキャナアプリ314の3つのアプリケーションの起動時処理と、それぞれのアプリ画面を生成し表示する処理と、アプリ画面選択キーが押下されたときの処理を説明するための説明図である。

10

20

30

40

50

## 【 0 1 1 4 】

アプリケーションの起動時処理； 電源オンやシステムリセット時には少なくとも S C S 3 2 2 と O C S 3 2 6 が起動された後にアプリケーションが起動される。コピーアプリ 3 1 2、プリンタアプリ 3 1 1 およびスキャナアプリ 3 1 4 は、S C S 3 2 2 のアプリ登録管理モジュール 3 2 2 b に対して起動された旨を登録する（ステップ S 7 0 1 ~ S 7 0 3 ）。

## 【 0 1 1 5 】

画面生成処理； O C S 3 2 6 では、複数のアプリケーションがそれぞれ仮想的な画面メモリを複数持つことができる図面管理機能を有する。このため、コピーアプリ 3 1 2、プリンタアプリ 3 1 1 およびスキャナアプリ 3 1 4 は、それぞれウインドウ制御ライブラリモジュール 3 2 6 a により画面生成を実行することができる（ステップ S 7 0 4 ~ S 7 0 8 ）。

## 【 0 1 1 6 】

表示処理； 複数の画面メモリのうち一つが表示パネル上に描画される。また、複数の画面メモリの内容を合成した画面が表示されるようにしても良い。たとえば、コピー動作中にファックス受信した場合には、アプリ共通の画面領域にファックス受信中である旨を表示することができる。

## 【 0 1 1 7 】

画面切換処理； 表示パネルに描画される画面メモリは、S C S 3 2 2 のシステム画面表示制御モジュールによって切り換えられる。たとえば、操作パネル上のアプリ選択キーが押下された場合には、ステップ S 7 0 9 ~ S 7 1 4 にしたがって選択されたアプリ画面が描画される。また、S C S 3 2 2 には、機器内部ハードウェアやエンジンの状態情報を保持する機器状態管理モジュール 3 2 2 c があるため、ジャムや紙なしなどのエラーや異常時の表示画面などを切り換えることができる。

## 【 0 1 1 8 】

次に、ファックスアプリ 3 1 3 またはプリンタアプリ 3 1 1 を用いたファックス送信動作について説明する。図 8 は、ファックスアプリ 3 1 3 またはプリンタアプリ 3 1 1 を用いたファックス送信動作を説明するための説明図である。

## 【 0 1 1 9 】

同図に示すように、ファックスアプリ 3 1 3 を用いたファックス送信動作については、ファックス操作画面において送信宛先番号が入力され、スタートキーが押下されると、ファックス画面・キー操作モジュール 3 1 3 a がアプリサービス 3 2 9 のアプリジョブ生成モジュール 3 2 9 a に対してジョブ生成関数を呼び出し（ステップ S 8 0 1 ）、このファックスジョブ生成モジュール 3 1 3 b が、F C S 3 2 7 のファックスジョブ実行制御モジュール 3 2 7 a に対して送信開始関数を呼び出す（ステップ S 8 0 2 ）。

## 【 0 1 2 0 】

すると、このファックスジョブ実行制御モジュール 3 2 7 a は、E C S 3 2 4 の A P I を用いてエンジンジョブ実行制御モジュール 3 2 4 a に対してエンジンジョブ生成関数を呼び出し、ジョブモードの設定をおこなう（ステップ S 8 0 3 ）。このエンジンジョブ実行制御モジュール 3 2 4 a は、M C S 3 2 5 の A P I を用いて画像メモリハンドリングモジュール 3 2 5 a に対して画像メモリ確保関数を呼び出し、画像メモリの確保を指示する（ステップ S 8 0 4 ）。

## 【 0 1 2 1 】

画像メモリハンドリングモジュール 3 2 5 a は、S R M 3 2 3 に対してメモリ取得関数を呼び出し、ファックス原稿のスキャンに必要なメモリを取得し（ステップ S 8 0 5 ）、メモリが確保されると、エンジンジョブ実行制御モジュール 3 2 4 a は、S R M 3 2 3 のエンジン資源管理モジュール 3 2 3 a に対してスキャナ資源の確保を要求する関数を呼び、確保できた後にスキャナ動作を開始する（ステップ S 8 0 6 ）。

## 【 0 1 2 2 】

スキャナから原稿画像データがメモリ上に保持されると、E C S 3 2 4 から F C S 3 2

10

20

30

40

50

7 に対してスキャン完了イベントが通知され、ファクシミリジョブ実行制御モジュール 3 2 7 a が、ファックスボード資源管理モジュール 3 2 3 g の送信開始関数を呼び出すと、P S T N、I S D N 回線などを使ってファックス送信手順が開始される (ステップ S 8 0 7 ~ S 8 0 8)。

【0 1 2 3】

次に、プリンタアプリ 3 1 1 を用いたファックス送信動作については、通常のプリント動作と同様に、印刷データがホスト I / O 資源管理モジュール 3 2 3 d に入力され、その際に送信宛先番号などのファクシミリ送信に必要な情報を含むコマンド情報が付加される (ステップ S 8 0 9)。

【0 1 2 4】

ファックス送信用の印刷データが、N C S 3 2 8 のホスト通信 I / F 制御モジュールからアプリサービス 3 2 9 のアプリデータ通信モジュール 3 2 9 b を経て、アプリジョブ生成モジュール 3 2 9 a に転送されると (ステップ S 8 1 0 ~ S 8 1 2)、新たなファックス送信用プリントジョブが生成される。

【0 1 2 5】

そして、印刷データが R I P 処理モジュール 3 1 1 a に転送され、画像データが生成されると、この画像データは、ファクシミリ送信に必要な情報とともに、F C S 3 2 7 のファックスジョブ実行制御モジュール 3 2 7 a に入力され (ステップ S 8 1 3 ~ 8 1 5)、以後ファックス送信動作と同様の処理がなされる (ステップ S 8 0 7 ~ S 8 0 8)。

【0 1 2 6】

なお、上記ステップ S 8 1 3 ~ S 8 1 4 による R I P 処理をおこなうのではなく、P S T N や I S D N 回線などを使ってバイナリーデータの転送をおこなうプロトコル (B F T P ; Binary File Transfer Protocol) によって印刷データを送信することもできる。

【0 1 2 7】

また、ここでは説明の便宜上、送信動作のみを説明したが、たとえば、ステップ S 8 0 8、S 8 0 7、S 8 0 2、S 8 0 3、S 8 0 4、S 8 0 5、S 8 0 6 の順に処理することにより、ファックス受信や印刷が可能となる。なお、B F T P を用いた印刷データの受信および印刷は、ステップ S 8 0 8、S 8 0 7、S 8 0 2、S 8 1 5、S 8 1 3、S 8 1 4、S 8 1 5、S 8 0 3、S 8 0 4、S 8 0 5、S 8 0 6 の順に処理することで実現できる。

【0 1 2 8】

次に、図 1 に示した複合機 1 2 0 のハードウェア構成について説明する。図 9 は、図 1 に示した複合機 1 2 0 のハードウェア構成を示す構成図である。同図に示すように、この複合機 1 2 0 は、C P U 9 0 2、S D R A M 9 0 3、フラッシュメモリ 9 0 4 および H D 9 0 5 などを A S I C 9 0 1 に接続したコントローラボード 9 0 0 と、オペレーションパネル 9 1 0 と、ファックスコントロールユニット (F C U) 9 2 0 と、U S B 9 3 0 と、I E E E 1394 9 4 0 と、プリンタ 9 5 0 とからなる。

【0 1 2 9】

そして、オペレーションパネル 9 1 0 は A S I C 9 0 1 に直接接続され、F C U 9 2 0、U S B 9 3 0、I E E E 1394 9 4 0 およびプリンタ 9 5 0 は、P C I バスを介して A S I C 9 0 1 に接続されている。

【0 1 3 0】

図 1 0 は、図 9 に示した A S I C 9 0 1 の細部構成を示すブロック図である。同図に示すように、この A S I C 9 0 1 は、C P U インターフェース (C P U I / F)、S D R A M インターフェース (S D R A M I / F)、ローカルバスインターフェース (Local BUS I / F)、P C I インターフェース (P C I I / F)、1 2 8 4、M A C (Media Access Controller)、I / O、O P E インターフェース (O P E I / F)、H D インターフェース (H D I / F)、Comp / de-comp、Rotate1によって形成されている。

【0 1 3 1】

かかるハードウェア構成を採用することにより、デバイスの共有化による低コスト設計

10

20

30

40

50

が可能となるとともに、アプリ間融合が容易となる。また、低速機から高速機までスケラブルなアーキテクチャとなり、各アプリで使用するハード/ソフトが共通化され、開発効率を向上させることができる。また、新規機能に対する対応が容易となる。

【0132】

上述してきたように、本実施の形態では、各アプリケーションの共通部分をアプリサービス121bおよび共通システムサービス121aとして括りだし、このアプリサービス121b、共通システムサービス121aおよび汎用OS111によりプラットフォーム122を形成し、このプラットフォーム122上に、プリンタアプリ123、コピーアプリ124および各種アプリ125を搭載するよう構成したので、各アプリケーションの開発労力軽減並びにアプリケーションのスリム化を図ることができる。

10

【符号の説明】

【0133】

- 100 プリンタ
- 101 専用OS
- 102 プリンタアプリ
- 110 複合機
- 111 汎用OS
- 112 エンジン制御部
- 113 プリンタアプリ
- 114 コピーアプリ
- 115 各種アプリ
- 120 複合機
- 121a 共通システムサービス
- 121b アプリサービス
- 122 プラットホーム
- 123 プリンタアプリ
- 124 コピーアプリ
- 125 各種アプリ

20

【先行技術文献】

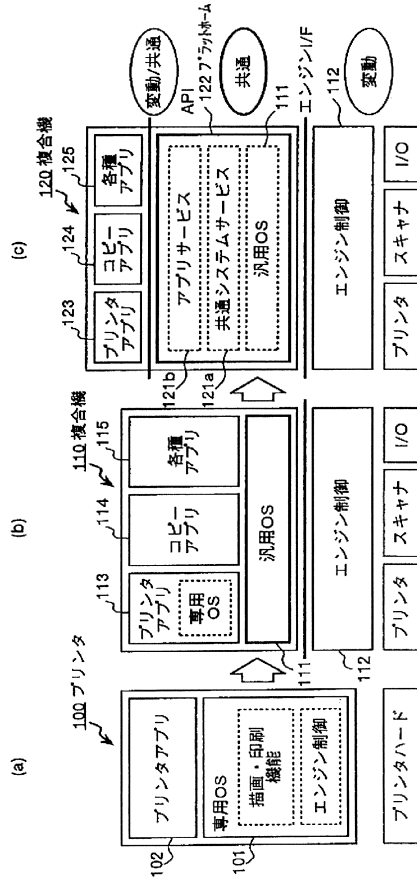
【特許文献】

30

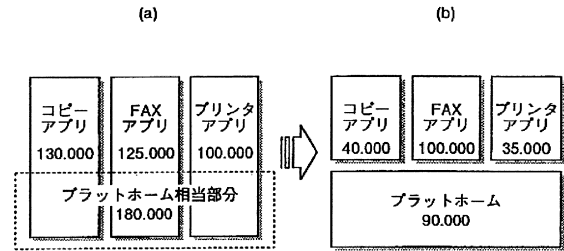
【0134】

【特許文献1】特公平7-79368号公報

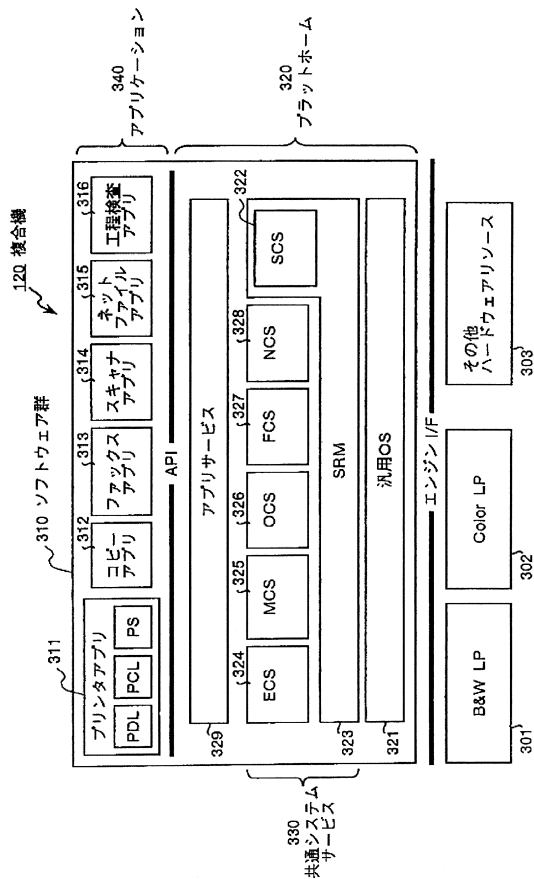
【図 1】



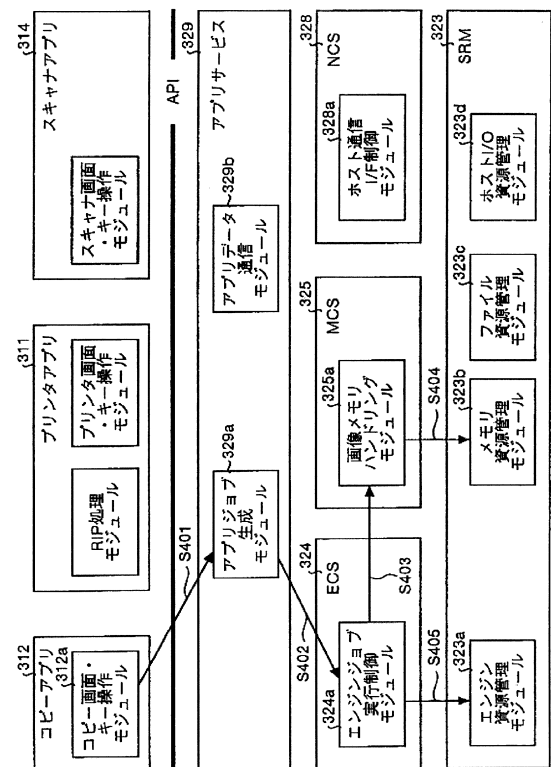
【図 2】



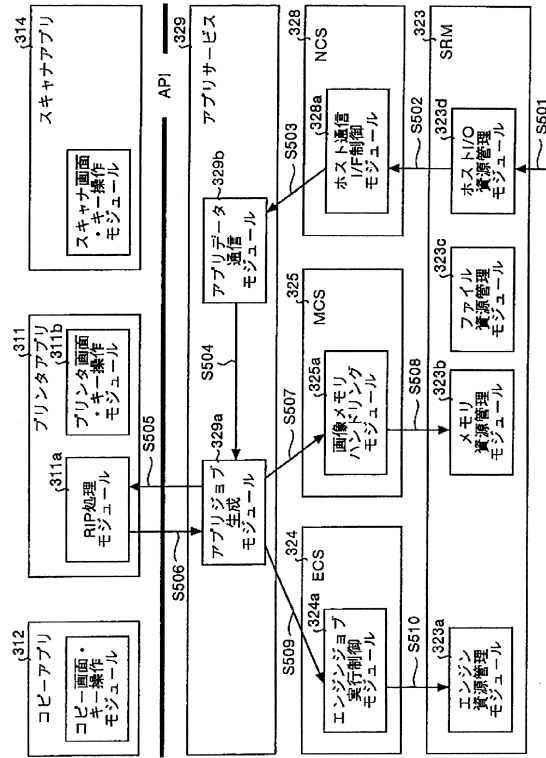
【図 3】



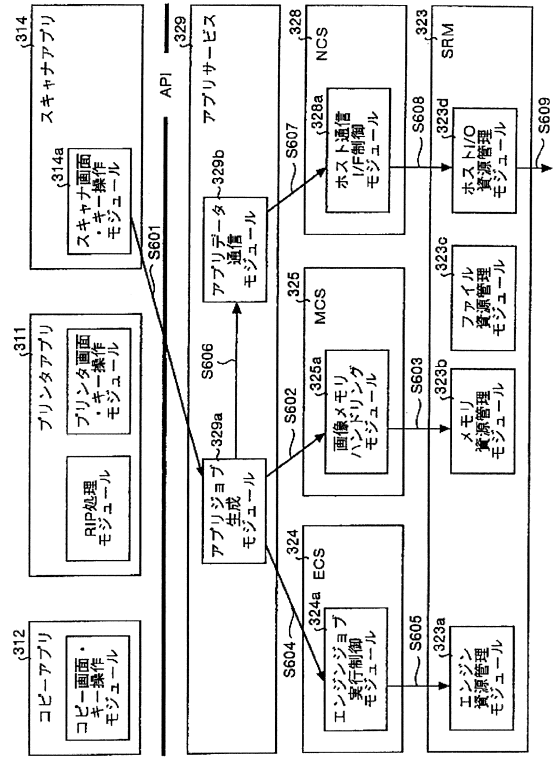
【図 4】



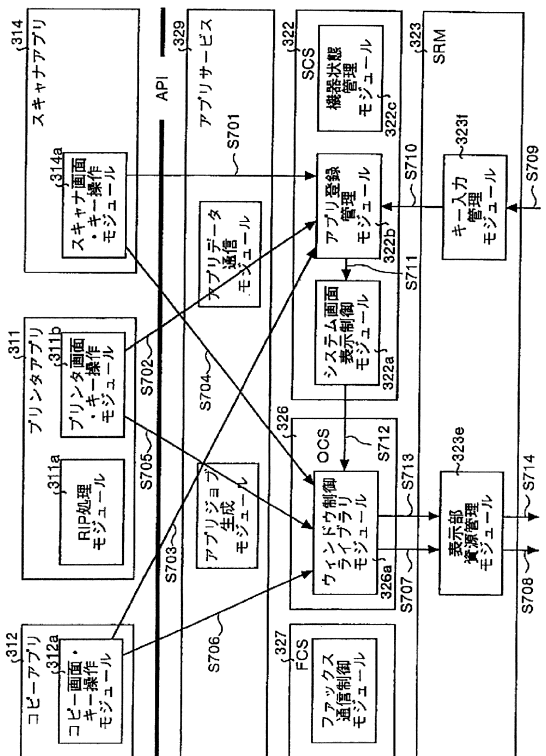
【図 5】



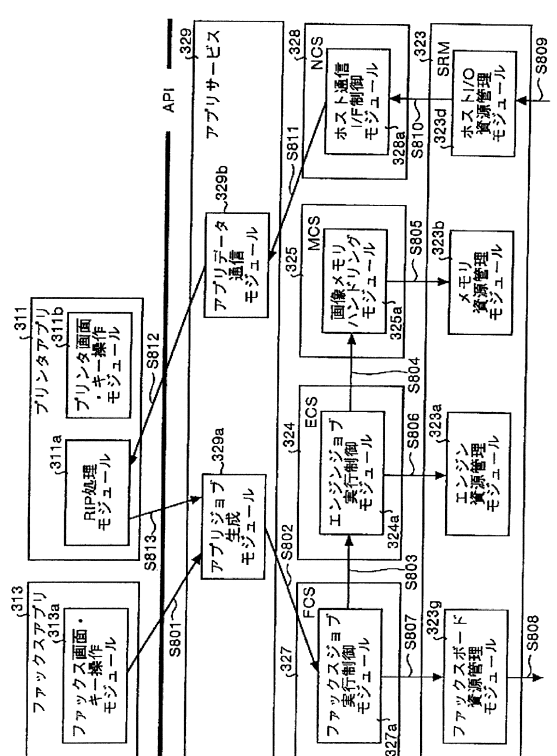
【図 6】



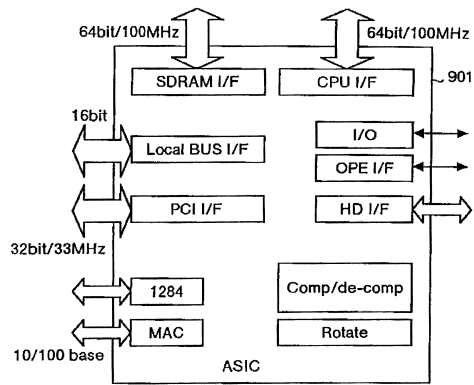
【図 7】



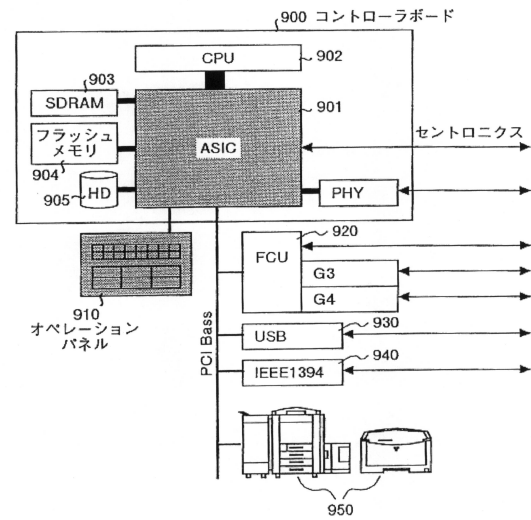
【図 8】



【図 10】



【図 9】



---

フロントページの続き

(56)参考文献 特開平09-051398(JP,A)  
特開平09-293134(JP,A)  
特開2000-056625(JP,A)  
特開平07-093230(JP,A)

(58)調査した分野(Int.Cl., DB名)  
H04N 1/00  
B41J29/38  
G03G21/00  
G06F 3/12