

(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2009-199419

(P2009-199419A)

(43) 公開日 平成21年9月3日(2009.9.3)

(51) Int.Cl.	F 1	テーマコード (参考)
G 0 6 F 3/12 (2006.01)	G 0 6 F 3/12 C	5 B 0 2 1
G 0 6 F 17/21 (2006.01)	G 0 6 F 3/12 N	5 B 1 0 9
	G 0 6 F 17/21 5 6 6 R	

審査請求 有 請求項の数 7 O L (全 12 頁)

(21) 出願番号	特願2008-41368 (P2008-41368)	(71) 出願人	000005267
(22) 出願日	平成20年2月22日 (2008.2.22)		ブラザー工業株式会社
			愛知県名古屋市瑞穂区苗代町 1 5 番 1 号
		(74) 代理人	110000578
			名古屋国際特許業務法人
		(72) 発明者	宮田 優治
			愛知県名古屋市瑞穂区苗代町 1 5 番 1 号
			ブラザー工業株式会社内
		F ターム (参考)	5B021 PP08
			5B109 NA14 RC13 TA06

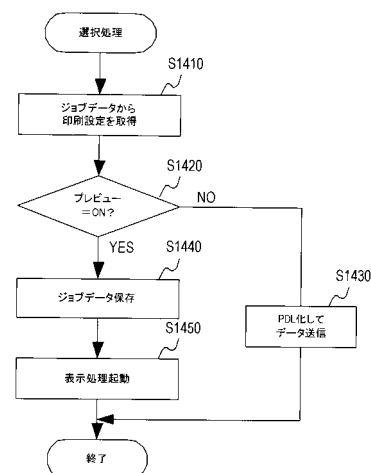
(54) 【発明の名称】 コンピュータ、印刷制御プログラム及び印刷制御装置

(57) 【要約】

【課題】 プレビューの表示がされていてもアプリが操作できる印刷制御のためのコンピュータの提供。

【解決手段】 すでに印刷のための加工が済んだジョブデータから印刷設定を取得する (S 1 4 1 0)。そして、取得した印刷設定において、プレビューの表示をするように設定されているか否かを判断する (S 1 4 2 0)。プレビューの表示をするように設定されていないと判断すると (S 1 4 2 0 で No)、ジョブデータを PDL 化すると共に、プリンタポート I / F 4 7 を通じてプリンタ 1 0 に送信する (S 1 4 3 0)。一方、プレビューの表示をするように設定されていると判断すると (S 1 4 2 0 で Yes)、ジョブデータを RAM 3 5 に保存する (S 1 4 4 0)。そして、表示処理を別タスクで起動して (S 1 4 5 0)、この処理を終える。そして、表示処理によってプレビューが表示される。

【選択図】 図 4



【特許請求の範囲】

【請求項 1】

マルチタスクOS（オペレーティングシステム）によってタスクを管理し、当該マルチタスクOSの下で実行したアプリケーションプログラムによって起動されるプリンタドライバを、当該プリンタドライバによる一連の処理が終了するまで、起動元のアプリケーションプログラムに基づく処理を一時停止した状態で、前記起動元のアプリケーションプログラムと同一タスクでCPUにより実行するコンピュータであって、

前記プリンタドライバは、

前記起動元が保持する印刷対象のデータを、プリンタに印刷させる様式に加工する加工手順と、

前記加工手順による加工後のデータを、前記プリンタに出力する出力手順と、

表示プログラムを、当該プリンタドライバとは異なるタスクとして起動する表示起動手順と、

前記出力手順及び前記表示起動手順の少なくともいずれか一方を選択して実行する選択手順と、

前記表示起動手順により前記表示プログラムが起動されると、当該プリンタドライバを終了する終了手順と、

を、前記CPUに実行させるためのプログラムとして構成され、

前記表示プログラムは、

プレビュー画像として、前記加工手順による加工後のデータに基づいてプリンタが印刷する画像を、当該コンピュータが表示画面を制御する表示装置に、表示させる表示手順

を、前記CPUに実行させるためのプログラムとして構成される

ことを特徴とするコンピュータ。

【請求項 2】

前記表示手順は、印刷指令を入力可能なユーザインタフェースとして機能する画面を、前記プレビュー画像と共に表示する手順であり、

前記表示プログラムは、

前記表示手順により表示された画面を通じて前記印刷指令が入力されると、前記プリンタドライバを起動する印刷起動手順

を、前記CPUに実行させるためのプログラムとして構成されており、

前記印刷起動手順によって前記プリンタドライバが起動された場合、前記出力手順では、前記表示装置に表示された前記プレビュー画像に対応する、前記加工手順にて加工されたデータを前記プリンタに出力する

ことを特徴とする請求項 1 に記載のコンピュータ。

【請求項 3】

前記印刷起動手順では、前記表示手順により表示された画面を通じて前記印刷指令が入力されなくても、前記表示手順にて前記プレビュー画像が表示されてから所定時間が経過すると、前記プリンタドライバを起動する

ことを特徴とする請求項 2 に記載のコンピュータ。

【請求項 4】

前記表示プログラムは、

前記加工手順による加工後のデータを、前記記憶媒体に記憶させる記憶手順

を、前記CPUに実行させるためのプログラムとして構成されている

ことを特徴とする請求項 2 又は請求項 3 に記載のコンピュータ。

【請求項 5】

前記プリンタドライバは、

前記印刷起動手順によって前記プリンタドライバが起動された場合には、前記加工手順の実行を禁止する禁止手順

を、前記CPUに実行させるためのプログラムとして構成される

ことを特徴とする請求項 2～請求項 4 の何れかに記載のコンピュータ。

10

20

30

40

50

【請求項 6】

マルチタスクOS（オペレーティングシステム）によってタスクを管理し、当該マルチタスクOSの下で実行したアプリケーションプログラムによって起動されるプリンタドライバを、当該プリンタドライバによる一連の処理が終了するまで、起動元のアプリケーションプログラムに基づく処理を一時停止した状態で、前記起動元のアプリケーションプログラムと同一タスクでCPUにより実行するコンピュータに、実行させるための印刷プログラムであって、

前記印刷プログラムは、前記プリンタドライバと表示プログラムとから成り、

前記プリンタドライバは、

前記起動元が保持する印刷対象のデータを、プリンタに印刷させる様式に加工する加工手順と、

前記加工手順による加工後のデータを、前記プリンタに出力する出力手順と、

前記表示プログラムを、当該プリンタドライバとは異なるタスクとして起動する表示起動手順と、

前記出力手順及び前記表示起動手順の少なくともいずれか一方を選択して実行する選択手順と、

前記表示起動手順により前記表示プログラムが起動されると、当該プリンタドライバを終了する終了手順と、

を、前記CPUに実行させるためのプログラムとして構成され、

前記表示プログラムは、

プレビュー画像として、前記加工手順による加工後のデータに基づいてプリンタが印刷する画像を、前記コンピュータが表示画面を制御する表示装置に、表示させる表示手順

を、前記CPUに実行させるためのプログラムとして構成されている

ことを特徴とする印刷プログラム。

【請求項 7】

印刷制御手段と表示制御手段とを備える印刷制御装置であって、

前記印刷制御手段は、

印刷対象のデータを、プリンタに印刷させる様式に加工する加工手段と、

前記加工手段による加工後のデータを、前記プリンタに出力する出力手段と、

前記表示制御手段を、前記印刷制御手段と並列的に処理が可能のように起動する表示起動手段と、

前記出力手段及び前記表示起動手段の少なくともいずれか一方を選択して起動する選択手段と、

前記表示起動手段により前記表示制御手段が起動されると、前記印刷制御手段の動作を終了する終了手段と、

を備え、

前記表示制御手段は、前記加工手段による加工後のデータに基づいて前記プリンタが印刷する画像を表示する

ことを特徴とする印刷制御装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、印刷を制御するコンピュータ、プログラム及び装置に関する。

【背景技術】

【0002】

従来、プリンタドライバに組み込まれたプレビュー処理実行用のプログラムに従って、プリンタにより印刷されるデータに対応するプレビュー画像を、表示装置に表示する方法が知られている（例えば特許文献1）。この種の方法によれば、アプリケーションプログラム（以下、アプリと呼ぶ。）がプレビュー機能を備えていなくても、プレビュー表示を実行できる。ここで言うプレビュー表示とは、アプリによって生成されたデータがどのよ

10

20

30

40

50

うに印刷されるかを、印刷の前に表示することである。

【特許文献1】特開平7-325693号公報

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、周知のWINDOWS（登録商標）等のOS（オペレーティングシステム）により動作するコンピュータでは、アプリから呼び出されるプリンタドライバがアプリと同一タスクで実行されるため、プリンタドライバが実行されている期間、アプリによる処理が停止して、ユーザからのアプリの操作を受け付けることができなくなってしまう。従って、従来技術では、プリンタドライバによって表示されたプレビュー画像を、ユーザが、アプリを操作して、アプリの表示画面と見比べることができないといった欠点があった。

10

【0004】

本発明は先述した課題に鑑み、プレビューの表示とアプリによる表示とを見比べることが可能なコンピュータ、プログラム及び印刷制御装置の提供を目的とする。

【課題を解決するための手段】

【0005】

先述した課題を解決するためになされた請求項1に記載のコンピュータは、マルチタスクOSによってタスクを管理する。そして、マルチタスクOSの下で実行したアプリによって起動されるプリンタドライバを、プリンタドライバによる一連の処理が終了するまで、起動元のアプリに基づく処理を一時停止した状態で、起動元のアプリと同一タスクでCPUにより実行する。

20

【0006】

そして、このコンピュータは、プリンタドライバと表示プログラムとを実行するように構成されている。また、プリンタドライバは、加工手順、出力手順、表示起動手順、選択手順および終了手順をCPUに実行させるためのプログラムとして構成されている。

【0007】

具体的に、当該コンピュータは、プリンタドライバにより、加工手順で、起動元が保持する印刷対象のデータを、プリンタに印刷させる様式に加工する。一方、選択手順では、出力手順及び表示起動手順の少なくともいずれか一方を選択して実行する。また、出力手順では、加工手順による加工後のデータを、プリンタに出力する。この他、表示起動手順では、表示プログラムを、プリンタドライバとは異なるタスクとして起動する。そして、表示起動手順により表示プログラムが起動されると、終了手順により、プリンタドライバを終了する。

30

【0008】

一方、表示プログラムは、表示手順をCPUに実行させるためのプログラムとして構成されている。具体的に、表示手順では、プレビュー画像として、加工手順による加工後のデータに基づいてプリンタが印刷する画像を、コンピュータが表示画面を制御する表示装置に、表示させる。

【0009】

40

請求項1に記載のコンピュータによれば、ユーザは、プレビューの表示とアプリの表示とを見比べることができる。なぜなら、本発明のプリンタドライバは、コンピュータが表示プログラムを別タスクとして起動した後にプリンタドライバを終了するように構成されているからである。つまり、本発明によれば、プリンタドライバによる一連の処理が終了するまで、起動元のアプリに基づく処理を一時停止するように構成されたコンピュータにおいても、プレビューを表示している間に、プリンタドライバを終了することができるので、アプリがユーザインタフェースを通じた操作を受け付けることができる。

【0010】

ところで、請求項1に記載のコンピュータは、請求項2に記載のように構成されるとよい。請求項2に記載の表示プログラムにより実現される表示手順では、印刷指令を入力可

50

能なユーザインタフェースとして機能する画面を、プレビュー画像と共に表示させる。

【0011】

また、この表示プログラムは、表示手順により表示された画面を通じて印刷指令が入力されると、プリンタドライバを起動する印刷起動手順を、コンピュータが実行可能な構成にされている。

【0012】

このコンピュータは、印刷起動手順によってプリンタドライバが起動された場合、出力手順では、表示装置に表示されたプレビュー画像に対応する、加工手順にて加工されたデータをプリンタに出力する。

【0013】

請求項2に記載のコンピュータによれば、ユーザは、プレビューで印刷の具合を確認してから印刷をさせることができる。つまり、プレビュー画面を通じて、プレビュー画像通りの印刷が実行できる。

【0014】

請求項2に記載のコンピュータは、請求項3に記載のように構成されるとよい。請求項3に記載のコンピュータが実行する印刷起動手順では、表示手順により表示された画面を通じて印刷指令が入力されなくても、表示手順にてプレビュー画像が表示されてから所定時間が経過すると、プリンタドライバを起動する。

【0015】

請求項3に記載のコンピュータによれば、ユーザがプレビューをただけで印刷を指令するのを忘れたとしても、印刷が実行される。従って、プレビューを表示させただけなのに印刷を指令したとユーザが勘違いして印刷を待っていても印刷が実行されない、という事態を防ぐことができる。

【0016】

この他、請求項2又は請求項3に記載のコンピュータは、請求項4に記載のように構成されるとよい。請求項4に記載のコンピュータが実行する表示プログラムは、加工手順による加工後のデータを、記憶媒体に記憶させる記憶手順をコンピュータが実行可能な構成にされている。

【0017】

請求項4に記載のコンピュータによれば、一旦プレビューの表示をしたものの印刷をすることなくプレビューを終えて、そのプレビューに対応する印刷を後からする場合に、記憶したデータと呼び出せば直ぐに印刷できて便利である。

【0018】

請求項2～請求項4に記載のコンピュータは、請求項5に記載のように構成されるとよい。請求項5に記載のコンピュータが実行するプリンタドライバは、印刷起動手順によってプリンタドライバが起動された場合には、加工手順の実行を禁止する禁止手順をコンピュータが実行可能な構成にされている。

【0019】

請求項5に記載のコンピュータによれば、加工手順を重複して実行することを防ぐことができ、無駄な処理をCPUが実行することを防ぐことができる。

請求項6に記載の印刷プログラムは、請求項1に記載のコンピュータが実行するプログラムである。従って、請求項1に記載のコンピュータと同様な効果を奏することができる。

【0020】

また、請求項1に記載のコンピュータが実行する処理は、ハードウェアによっても実現できる。

請求項7に記載の印刷制御装置は、印刷制御手段と表示制御手段とを備える印刷制御装置である。この印刷制御装置が備える印刷制御手段は、加工手段と、出力手段と、表示起動手段と、選択手段と、終了手段とを有する。

【0021】

10

20

30

40

50

加工手段は、印刷対象のデータを、プリンタに印刷させる様式に加工する。一方、選択手段は、出力手段及び表示起動手段の少なくともいずれか一方を選択して起動する。また、出力手段は、加工手段による加工後のデータを、プリンタに出力し、表示起動手段は、表示制御手段を、印刷制御手段と並列的に処理が可能のように起動する。この他、終了手段は、表示起動手段により表示制御手段が起動されると、印刷制御手段の動作を停止する。

【0022】

また、表示制御手段は、加工手段による加工後のデータに基づいてプリンタが印刷する画像を表示する。従って、この印刷制御装置によれば、請求項1記載のコンピュータと同様の効果を奏することができる。

【発明を実施するための最良の形態】

【0023】

以下、図面と共に説明する。図1は、プリンタ10と、プリンタ10を制御する、本発明が適用されたPC（パーソナルコンピュータ）30との構成を表すブロック図である。

プリンタ10は、各種プログラムを実行して装置内各部を統括制御するCPU11と、CPU11により実行される各種プログラムを記憶するROM13と、CPU11によるプログラム実行時に作業領域として使用されるRAM15と、各種設定データを記憶するデータ書き換え可能な不揮発性メモリとしてのNVRAM17と、CPU11に制御されてPC30から入力されたジョブデータに基づく画像を用紙に印刷する印刷部21と、表示部（液晶モニタ）及びユーザが操作可能な操作部（キー群）を備える表示操作部23と、LANに接続されたプリンタポートインタフェース（いわゆるLANインタフェース）27とを備える。

【0024】

また、PC30は、各種プログラムを実行して装置内各部を統括制御するCPU31と、CPU31により実行されるブートプログラム等を記憶するROM33と、CPU31によるプログラム実行時に作業領域として使用されるRAM35と、オペレーティングシステムやアプリケーションプログラム、プリンタドライバ等のプログラム及び各種データを記憶するハードディスク装置（HDD）37と、液晶モニタ等から構成される情報表示用の表示部41と、キーボードやポインティングデバイス等から構成されるユーザが操作可能な入力部43と、LANに接続されたプリンタポートインタフェース（いわゆるLANインタフェース）47とを備える。

【0025】

また、ハードディスク装置37には、オペレーティングシステムとして、複数のタスクを並列処理可能なマルチタスクOS（例えば、WINDOWS（登録商標））が記憶されており、CPU31は、このマルチタスクOSによってタスクを管理する。

【0026】

この他、本実施例のプリンタドライバは、複数のアプリによって共有されるプログラム部品（いわゆるDLL（Dynamic Link Library））として構成されている。即ち、当該プリンタドライバは、CPU31が上記マルチタスクOSの下で実行したアプリに対し、ユーザが印刷指示を入力することによって、当該アプリから呼び出されて起動される。

【0027】

そして、この際、CPU31は、プリンタドライバを、起動元のアプリと同一タスクで実行する。従って、PC30では、プリンタドライバによる一連の処理が終了するまで、起動元のアプリに基づく処理が一時停止することになる。

【0028】

次に、図2を用いて、印刷処理を説明する。図2は、印刷処理を表すフローチャートである。この処理は、CPU31が主体となって実行するものであり、入力部43を通じて印刷指示を受けることにより開始される。なお、この処理は、CPU31が、上記プリンタドライバの記述に従って実行するものである。従って、この処理を実行している間には、アプリに基づく処理が一時停止することになる。

【0029】

この処理では、ウォーターマーク加工処理（S110）、スケーリング加工処理（S120）、マルチページ加工処理（S130）、選択処理（S140）が、CPU31により順に実行される。なお、ウォーターマーク加工処理、スケーリング加工処理およびマルチページ加工処理は、加工手順および加工手段の一例である。

【0030】

まず、図3を用いて、ウォーターマーク加工処理（S110）の詳細を説明する。CPU31は、この処理を開始すると初めに、ジョブデータをアプリから受信する（S1110）。このジョブデータは、印刷設定および印刷対象のデータである印刷データを含む。そして、受信したジョブデータに含まれる印刷設定を読み出して取得する（S1120）

10

【0031】

なお、印刷設定には、ウォーターマーク加工、スケーリング加工およびマルチページ加工のそれぞれについて、加工済みであるか否かを示すフラグ、並びに、プレビューの表示をするか否かを示すフラグが含まれる。

【0032】

次は、取得した印刷設定に含まれる、ウォーターマーク加工処理に対応したフラグが済みになっているか否かを判断する（S1130）。最初にデータを受け取ったときは、済みにはなっていない。しかし、後の処理によって、いったん加工したデータが再びウォーターマーク処理の対象になることがある。S1130は、そのときに重複して加工を実行しないようにするための判断である。

20

【0033】

ウォーターマークの加工処理に対応したフラグが済みになっていると判断すると（S1130でYes）、この処理を終える。一方、ウォーターマークの加工処理に対応したフラグが済みになっていないと判断すると（S1130でNo）、取得した印刷設定に基づいて、ウォーターマーク加工処理を実行するか否かを判断する（S1140）。ウォーターマーク加工処理を実行しないと判断すると（S1140でNo）、S1160に進む。ウォーターマーク加工処理を実行すると判断すると（S1140でYes）、ジョブデータが示す印刷設定により指定されたウォーターマーク（「社外秘」マーク等）を、印刷データに付加することで、印刷データを加工して、印刷データの印刷処理時にウォーターマークが用紙に印刷されるようにする（S1150）。その後、S1160に進む。

30

【0034】

S1160に進むと、ジョブデータに対して、ウォーターマーク加工処理に対応したフラグを済みに設定して、この処理を終える。なお、S1130及びS1160は、禁止手順の一例である。

【0035】

次は、ウォーターマーク加工処理によって加工されたジョブデータに対して、スケーリング加工処理（S120）を実行する。スケーリング加工とは、印刷倍率を設定することである。例えば、A4としてのデータをA3用紙に拡大して印刷する際に行うものである。

40

【0036】

なお、この処理の詳細は説明しない。理由は、ウォーターマーク加工処理を適宜、読み替えばよいからである。ここで実行するスケーリング加工処理は、図3において、複数表現されている「ウォーターマーク加工処理」をそれぞれ「スケーリング加工処理」に、「ウォーターマーク付加」を「スケーリング加工」に、それぞれ読み替えて理解される内容に等しい。

【0037】

次は、スケーリング加工処理によって加工されたジョブデータに対して、マルチページ加工処理（S130）を実行する。マルチページ加工とは、複数頁分の画像を、一枚の用紙に印刷するために行うデータ加工のことである。なお、マルチページ加工処理もスケー

50

リング加工処理と同じように、ウォーターマーク加工処理からの読み替え方だけを説明する。即ち、マルチページ加工処理は、図3において、複数表現されている「ウォーターマーク加工処理」をそれぞれ「マルチページ加工処理」に、「ウォーターマーク付加」を「マルチページ加工」に、それぞれ読み替えて理解される内容に等しい。

【0038】

次は、マルチページ加工処理によって加工されたジョブデータに対して、選択処理（S140）を実行する。この処理は、印刷設定に応じて、プレビューの表示を実行可能な表示処理を起動するか、ジョブデータをPDL化処理すると共にプリンタに送信するか、どちらかを選択して実行する処理である。なお、PDL（Page Description Language）化処理とはプリンタに印刷指令をする際に、プリンタに印刷処理の内容を理解させるためのデータを生成する処理である。

10

【0039】

ここから、図4を用いて説明する。まず、ジョブデータから印刷設定を取得する（S1410）。そして、取得した印刷設定において、プレビューに対応するフラグが、プレビュー表示すべき設定になっているか否かを判断する（S1420）。なお、S1420は、選択手順および選択手段の一例である。

【0040】

そして、上記フラグが、プレビュー表示すべき設定になっていないと判断すると（S1420でNo）、ジョブデータをPDL化すると共に、プリンタポートI/F47を通じてプリンタ10に送信する（S1430）。なお、S1430は、出力手順および出力手段の一例である。そして、選択処理を終了すると共に、印刷処理を終了する。なお、この終了が、終了手順および終了手段の一例である。

20

【0041】

一方、上記フラグが、プレビュー表示すべき設定になっていると判断すると（S1420でYes）、ジョブデータをRAM35に保存する（S1440）。そして、表示処理を印刷処理と並列処理可能なように、当該印刷処理とは別タスクで表示処理を起動する（S1450）。なお、S1450は、表示起動手順および表示起動手段の一例である。その後、図4に示す選択処理を終了すると共に、図2に示す印刷処理を終了する。なお、この終了も、終了手順および終了手段の一例である。

【0042】

印刷設定において、プレビュー表示すべき設定になっているか否かは、印刷処理の起動以前での、入力部43を通じて入力される指示に従うことになる。つまり、プレビューの表示なしで印刷するような指示であれば、プレビューに対応するフラグがOFFに設定されるので、S1420でNoと判断することになる。一方、プレビューの表示が指示されれば、プレビューに対応するフラグがONに設定されるので、S1420でYesと判断することになる。

30

【0043】

先述したように、プリンタドライバとしての印刷処理を実行している間、アプリに基づく処理は一時停止することになる。言い換えれば、CPU11は、この処理を終えればアプリに基づく処理を実行できるようになる。

40

【0044】

次は、図5を用いて、表示処理を説明する。この表示処理は、プレビュー表示用の表示プログラムに従って、CPU31が実行するものであり、先述したように選択処理によってS1450にて起動されることを契機に開始される。なお、この表示プログラムは、本実施例のPC30において、プリンタドライバと共にハードディスク装置37に記憶されている。この表示プログラムは、予め、上述した印刷処理を行うためのプリンタドライバとセットで、ユーザに、通信やCD-ROM等の記録媒体を媒介にして提供される。

【0045】

ところで、先述したように、当該PC30においては、表示処理を、アプリを動作させているタスクとは、別のタスクで動作させる。従って、以下に説明する表示処理の実行中

50

には、当該表示処理を実行する原因となったプリンタドライバ起動元のアプリに基づく処理が停止し続けることがない。この点が本発明の大きな特徴である。

【0046】

この表示処理を開始すると、まず、当該表示処理を実行する原因となった印刷処理におけるウォーターマーク加工処理等において加工された印刷データに基づいて、当該印刷データに基づく画像がプリンタ10にて印刷されるときに用紙に形成される画像を、プレビューとして、表示部41に表示する(S210)。なお、S210は、表示手順および表示制御手段の一例である。ここで図6に移る。図6は、表示部41に表示されるプレビューの具体例である。図6に表されているように、印刷される様子のプレビュー表示60に加えて、印刷アイコン62、中止アイコン64、保存アイコン66の各アイコンが表示されている。これらのアイコンは、入力部43を通じてユーザが選択できるように、グラフィカルユーザインタフェースとして機能する。

10

【0047】

図5に戻る。次は、先述した保存アイコン66が選択されたかを判断する(S220)。保存アイコン66が選択されたと判断すると(S220でYes)、ジョブデータを所定のファイル形式(例えば、TIFF形式やPDF形式)にてHDD37に保存する(S230)なお、S230は、記憶手順の一例である。そして、この処理を終える。ここで保存したジョブデータは、後で印刷処理に入力することができる。

【0048】

一方、保存アイコン66が選択されていなかったと判断すると(S220でNo)、中止アイコン64が選択されたかを判断する(S240)。中止アイコン64が選択されたと判断すると(S240でYes)、この表示処理を終える。

20

【0049】

一方、中止アイコン64が選択されていなかったと判断すると(S240でNo)、印刷アイコン62が選択されたかを判断する(S250)。印刷アイコン62が選択されていなかったと判断すると(S250でNo)、この処理を開始してから所定時間が経過したかを判断する(S260)。この処理を開始してから所定時間が経過していないと判断すると(S260でNo)、S220に戻る。一方、印刷アイコン62が選択されと判断すると(S250でYes)、S270に進む。

【0050】

つまり、印刷アイコン62、中止アイコン64、保存アイコン66の何れかが選択されていない状況下では、所定時間が経過するまで、待機することになる。一方、印刷アイコン62、中止アイコン64、保存アイコン66の何れも選択されることなく、この処理が開始してから所定時間が経過したと判断すると(S260でYes)、S270に進む。このように構成することで、本実施例では、ユーザが印刷を選択したものと勘違いして、プリンタに印刷物を取りに行ってしまった場合でも印刷が実行されて、ユーザがプリンタの前で待ちぼうけになることを防ぐようにしている。

30

【0051】

S270に進むと、印刷設定のプレビューに対応したフラグをOFFに変更する。そして、印刷処理を起動する(S280)。なお、S280の処理は、印刷起動手順の一例である。そして、この処理を終える。

40

【0052】

また、先述したように、プリンタドライバとしての印刷処理を起動元のアプリと同じタスクで起動するように、本実施例におけるマルチタスクOSは構成されている。従って、この場合は表示プログラムが起動元のアプリに相当するので、印刷処理を起動するタスクは、表示処理と同じタスクである。

【0053】

ところで、S280にて起動された印刷処理においては、S1130でYes、S1420でNo、と判断することになる。言い換えると、そのように判断するために、S1160とS270の処理によってフラグを設定した上で、印刷処理を再び起動するように、

50

印刷処理および表示処理は構成されている。このように構成されることで、ウォーターマーク付加の処理（S 1 1 5 0）などが重複して実行されたり、再び表示処理が起動（S 1 4 2 0でY e sの後のS 1 4 5 0）されたりすることを防ぎつつ、S 1 4 3 0が実行されるようにしている。

【0054】

従って、S 2 8 0で印刷処理が起動された後には、プレビューに用いられた印刷データを含むジョブデータがP D L化されて、プリンタ10に送信され、この印刷データに基づく画像がプリンタ10により用紙に印刷される。

【0055】

ここで、本実施例と従来技術との違いを、概念的に説明する。以下では、一度プレビューを表示して、その後に印刷を実行する際の処理を、具体例として説明する。図7（a）は、従来技術によって印刷処理が実行される様子を概念的に表したものである。印刷処理が開始される（A）と、アプリを実行するタスクと同じタスクで、プリンタドライバを実行する。そして、アプリから受け取った印刷対象のデータを、印刷する形式に加工する（B）。そして、プレビューを表示（C）した後に、P D L化処理を実行すると共にプリンタにジョブデータを送信して（D）、印刷処理を終了する（E）。 10

【0056】

一方、本実施例の場合を図7（b）を用いて説明する。アプリと同一タスクで、プリンタドライバに従う印刷処理を開始して（A'）、印刷対象のデータを加工する（B'）までは、従来技術と同じである。従来技術と異なるのは、その後に、別タスクで動作するように表示処理を起動（F）し、P D L化すると共にジョブデータをプリンタに送信することなく（G）、印刷処理を終える（H）ことである。このようにしてプリンタドライバとしての印刷処理を終えれば、アプリに基づく処理の実行を再開できる。 20

【0057】

そして、別タスクで起動された表示処理（J）によって、プレビューの表示が実行される（K）。この状態では、アプリと表示処理とが別タスクなので、両者を見比べるために表示を切り換えることができる。

【0058】

そして、プレビューの表示を終える、つまり表示処理を終えると、印刷処理が同じタスクで起動される（L）。ウォーターマーク付加などの加工は既に済んでいるのでスキップし（M）、プレビューの表示もスキップして（N）、P D L化処理をすると共にプリンタに送信して（P）、印刷処理を終える（Q）。 30

【0059】

特に、プレビュー起動（F）、フローチャートで言うと選択処理のS 1 4 2 0によって、先述した本発明の効果がもたらされる。

この他の効果を二つ述べる。一つ目は次の通りである。従来ではプレビューを表示する間は、他のデータを印刷することができなかった。理由は、プリンタドライバは一つずつしかデータを処理しないように構成されているからである。それに対して本発明での印刷処理は、プレビューの表示中には終了している。従って、プレビューを表示しながら、他のデータの印刷を実行できる。 40

【0060】

二つ目は次の通りである。ネットワークを通じた印刷の場合、印刷処理開始時にC P U 3 1は、プリンタポートI / F 4 7を通じてプリンタ10に通信開始を指示する。しかし実際に印刷対象のデータが送られるのは、ユーザがプレビューを見終わってから印刷を指示した後である。ところが、その間に所定時間が経過してしまうと、タイムアウトになりプリンタが接続を切ってしまうことがある。この場合、プレビューの表示の後にユーザが印刷を指令したとしても、印刷できない。それに対して本発明では、プレビューが終わってから（S 2 8 0の後）プリンタと接続するようにすればよいので、タイムアウトすることがなくなり、先述した理由によって印刷に失敗することを予防できる。

【0061】

10

20

30

40

50

なお、特許請求の範囲に記載された技術範囲は、本実施例に限定されない。例えば、ハードウェアによって実現してもよい。

【図面の簡単な説明】

【0062】

【図1】プリンタ10及び本発明が適用されたPC30の概略構成を示すブロック図である。

【図2】印刷処理を表すフローチャートである。

【図3】ウォーターマーク加工処理を表すフローチャートである。

【図4】選択処理を表すフローチャートである。

【図5】表示処理を表すフローチャートである。

【図6】プレビュー表示の具体例である。

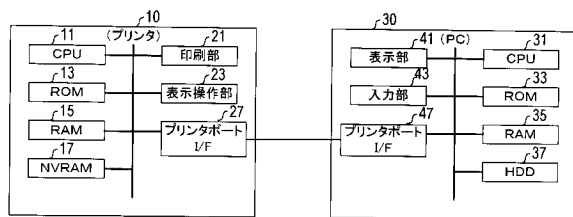
【図7】各処理とタスクとの関係を表した概念図である。

【符号の説明】

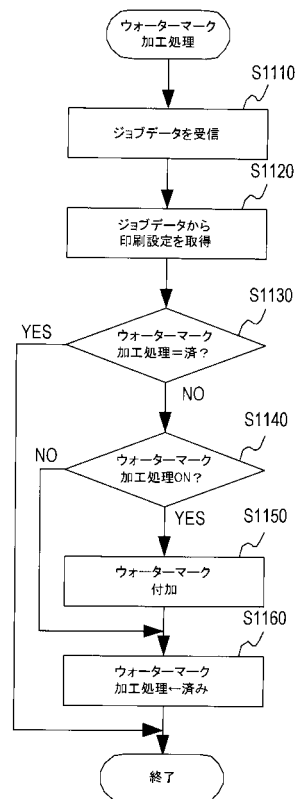
【0063】

10…プリンタ、11、31…CPU、13、33…ROM、15、35…RAM、17…NVRAM、21…印刷部、23…表示操作部、27、47…プリンタポートI/F、30…PC、37…ハードディスク装置、41…表示部、43…入力部、60…プレビュー表示、62…印刷アイコン、64…中止アイコン、66…保存アイコン

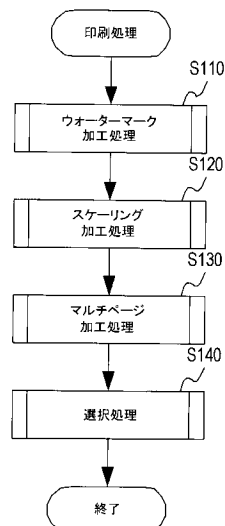
【図1】



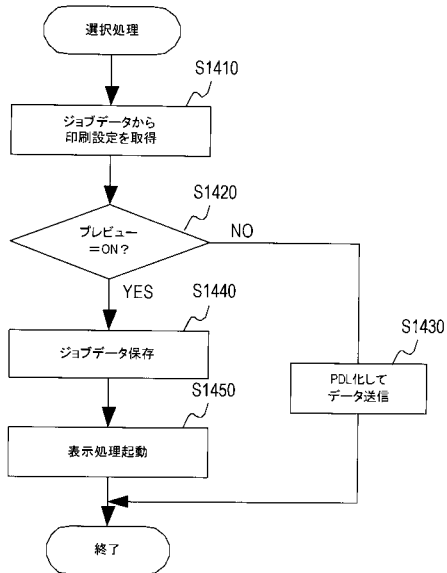
【図3】



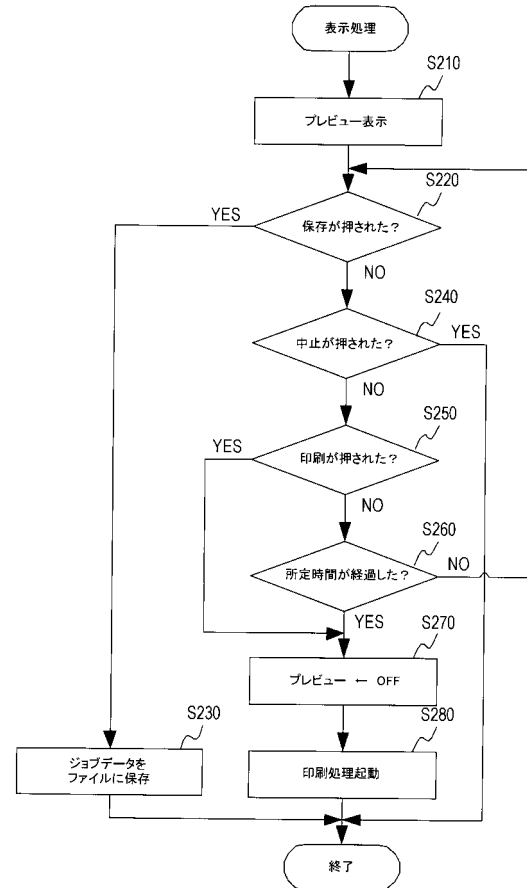
【図2】



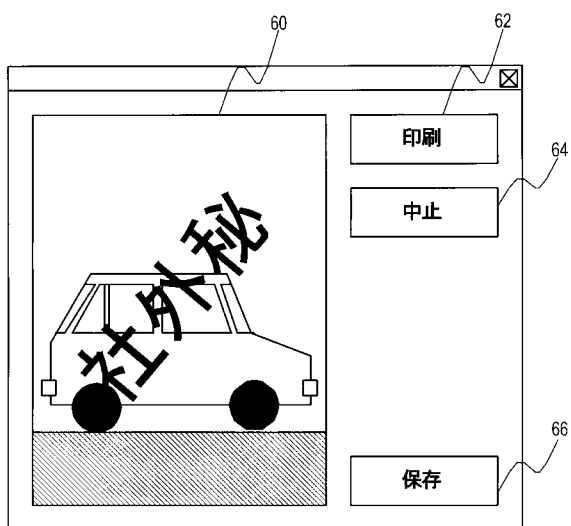
【図 4】



【図 5】



【図 6】



【図 7】

