(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2006-31538
(P2006-31538A)
(43)公開日　平成18年2月2日(2006.2.2)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 3/12	(2006.01)	G06F 3/12	C	5B021

審査請求 有　請求項の数 9　ＯＬ　（全 11 頁）

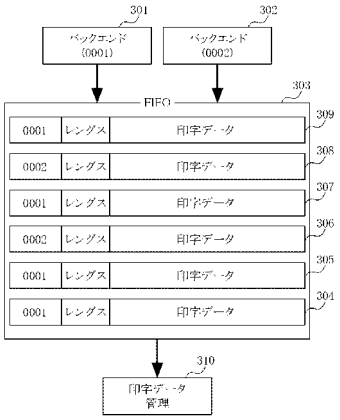
(21)出願番号　特願2004-211692 (P2004-211692)
(22)出願日　平成16年7月20日 (2004.7.20)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100090538
弁理士　西山　恵三
(74)代理人　100096965
弁理士　内尾　裕一
(72)発明者　小柴　晃良
東京都大田区下丸子３丁目３０番２号キヤノン株式会社内

Ｆターム(参考)　5B021 AA01　BB00　CC04

(54)【発明の名称】情報処理装置及び制御方法及びプログラム並びに記憶媒体

(57)【要約】
【課題】　プリンタにデータを出力する処理部の入り口が１つであるため、データ展開処理がマルチプロセスで動作すると、複数のジョブの印字データが混在した状態で、データ出力部にデータが送られてくる場合がある。
【解決手段】　プリンタへのデータ出力部に対してデータを送り出すバックエンド部にて、データをパケット化し、ジョブＩＤを付加してデータ出力部にデータを送り、データ出力部では、受信したパケットデータのジョブＩＤ部を判定し、データの扱いを決定する手段を設ける。
【選択図】　図３



【特許請求の範囲】
【請求項１】
印刷装置と通信可能な情報処理装置であって、
印刷データを複数に分割して複数のパケットにパケット化し、パケット毎にジョブ識別子を付加してデータ出力するパケット印刷データ生成手段と、
印刷のキャンセル指示に応じて、記憶手段に記憶される前記パケット印刷データ生成部により生成された前記印刷装置に転送前の複数のパケットを前記ジョブ識別子に基づいて破棄する破棄手段とを有することを特徴とする情報処理装置。
【請求項２】
印刷装置と通信可能な情報処理装置であって、
印刷ジョブのジョブ投入の監視、フィルタ及びレンダリング処理の起動およびジョブのキャンセル制御を行うジョブ管理手段と、
レンダリング処理され生成された印刷データを分割してパケット化し、パケット毎にジョブＩＤを付加してデータ出力する生成手段と、
レンダリング処理により生成された印刷データを取得しプリンタへデータ出力を行い、プリンタの状態監視を行うプリンタインターフェース処理手段と、
前記印刷ジョブに対する操作を行う為の操作手段と、
レンダリング処理により生成されたデータのジョブＩＤを判定する印字データ管理手段と、
印刷がキャンセルされたことをジョブ管理手段に通知する通知手段とを有することを特徴とする情報処理装置。
【請求項３】
前記ジョブＩＤに基づき前記パケット化されたデータを破棄する破棄手段を有することを特徴とする請求項２に記載の情報処理装置。
【請求項４】
処理中のジョブＩＤと異なるジョブＩＤのデータを取得した場合、取得したデータを有効として、これまで処理中としていたジョブＩＤを印刷キャンセルされたジョブＩＤのデータであるとする手段を有することを特徴とする請求項２に記載の情報処理装置。
【請求項５】
ユーザー操作により印刷キャンセルされた場合、キャンセル操作された時のジョブＩＤを保持しておく保持手段を有し、プリンタに送るデータのジョブＩＤがキャンセルされたジョブＩＤと同一であるか判断する判断手段を有することを特徴とする請求項２に記載の情報処理装置。
【請求項６】
処理対象でないジョブＩＤのデータを破棄しないで保持しておく保持手段を有することを特徴とする請求項２に記載の情報処理装置。
【請求項７】
印刷装置と通信可能な情報処理装置であって、
印刷データを複数に分割して複数のパケットにパケット化し、パケット毎にジョブ識別子を付加してデータ出力するパケット印刷データ生成ステップと、
印刷のキャンセル指示に応じて、記憶手段に記憶される前記パケット印刷データ生成部により生成された前記印刷装置に転送前の複数のパケットを前記ジョブ識別子に基づいて破棄する破棄ステップとを有することを特徴とする制御方法。
【請求項８】
請求項７に記載された制御方法をコンピュータに実行させるためのプログラム。
【請求項９】
請求項８に記載されたプログラムをコンピュータが読込み実行可能な形態で記憶したコンピュータ可読の記憶媒体。
【発明の詳細な説明】
【技術分野】

【０００１】
本発明は、マルチプロセスシステムにおける、印刷ジョブの制御方法の仕組みに関する。
【背景技術】
【０００２】
従来、ＵＮＩＸ（登録商標）系のＯＳにおけるアプリケーションでは、印刷スプーラへのジョブ投入までを印刷処理として処理している。また、印字データがプリンタまでの出力がされたことを管理するため、プリンタモニタ処理を設け、ジョブの完了を監視する処理を用意していた。
【０００３】
また、複数のジョブが投入された場合、シーケンシャルにジョブの完了を待って、データ出力処理を起動していた。
【特許文献１】特開２００４－０３８６５９号公報
【発明の開示】
【発明が解決しようとする課題】
【０００４】
しかしながら、マルチプロセスで動作するシステムにおいて、印刷ジョブ管理処理、印字データ展開処理、プリンタ管理処理などの各処理がそれぞれ別のプロセスとして動作するようになっている場合において、複数の印刷ジョブがスプーラに投入され、１つ目の印刷処理がキャンセルされた場合、印字データ出力用のＦＩＦＯ（Ｆｉｒｓｔ－Ｉｎ Ｆｉｒｓｔ－Ｏｕｔ）メモリに複数のプロセスから出力された印刷ジョブのデータが、混在して格納されてしまう場合があり、キャンセルされたデータだけでなく２つ目のデータの印刷も正しく行うことができないという問題あった。
【０００５】
本願発明は、上記問題点を鑑みてなされたものであり、上記ＦＩＦＯメモリにプリンタへの送信前の複数の印刷ジョブのデータが混在する状況下で特定の印刷ジョブのキャンセル指示がはいった場合でも、特定の印刷ジョブに関連するデータを確実に破棄できる仕組み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】
【０００６】
上記目的を達成するために、印刷装置と通信可能な情報処理装置であって、印刷データを複数に分割して複数のパケットにパケット化し、パケット毎にジョブ識別子を付加してデータ出力するパケット印刷データ生成手段と、印刷のキャンセル指示に応じて、記憶手段に記憶される前記パケット印刷データ生成部により生成された前記印刷装置に転送前の複数のパケットを前記ジョブ識別子に基づいて破棄する破棄手段とを有することを特徴とする仕組みを提供する。
【発明の効果】
【０００７】
上記ＦＩＦＯメモリにプリンタへ送信前の複数の印刷ジョブのデータが混在する状況下で、特定の印刷ジョブのキャンセル指示がはいった場合でも、キャンセル対象となっている特定の印刷ジョブに関連するデータを確実に破棄できる。
【発明を実施するための最良の形態】
【０００８】
（第１の実施形態）
図１は本発明のシステム全体の構成を示す構成図である。ＵＮＩＸ（登録商標）やＬｉｎｕｘ（登録商標）システムにおいて、印刷ジョブのキャンセル制御を司るジョブ管理１０３、フィルタ・レンダリング１０４、バックエンド部１０５は、ＣＵＰＳ（Ｃｏｍｍｏｎ Ｕｎｉｘ（登録商標） Ｐｒｉｎｔｉｎｇ Ｓｙｓｔｅｍ）と呼ばれる、印刷システムの構成要素であり、一般的に、ＣＵＰＳでは、スプーラを監視し、スプーラに投入されたデータを、印刷用データに展開して、バックエンド部１０５からプリンタに出力する処

理を行っている。

【０００９】

一般的に、バックエンド部１０５の先には、プリンタ１１１が接続されていることが多いが、ＵＮＩＸ（登録商標）やＬｉｎｕｘシステムの印刷処理において、プリンタの状態表示処理や、印刷完了を監視するため、プリンタモニタと呼ばれる処理が、間にはいる場合がある。このプリンタモニタ処理は、プリンタに対してのデータ出力の管理と共に、プリンタの状態を取得する処理も行っている。また、プリンタの状態を表示するアプリケーションとのインターフェース処理も持つことが多い。

【００１０】

アプリケーション１０１が印刷操作を行うと、印刷スプーラ１０２にデータが投入される。印刷スプーラ１０２は、印刷処理のジョブ管理部１０３により、データ投入が監視されており、印刷スプーラ１０２にデータが投入されると、ジョブ管理部１０３が、印刷スプーラ１０２からデータを取得し、印刷ジョブを開始する。取得したデータの種類に合わせて、フィルタ・レンダリング処理部１０４が起動され、データが印字用のデータに変換される。

【００１１】

印字用に変換されたデータはバックエンド部１０５に引き渡され、バックエンド部１０５からプリンタにデータ出力がされる。

【００１２】

先に述べたように、本実施例では、バックエンド部１０５からプリンタ１１１に直接印字データが送られるのではなく、バックエンド部１０５とプリンタ１１１の間に、プリンタデータの管理およびプリンタステータス取得等を行う処理が入る。

【００１３】

この処理部分は、印字データ管理部１０７、ＵＩインターフェース部１０９、プリンタインターフェース処理部１１０を含むプリンタモニタと呼ぶ部分である。

【００１４】

プリンタモニタは、通常ＣＵＰＳシステムとは別のプロセスとして動作する。そのため、印字データインターフェースとしてＦＩＦＯ方式を採用したＦＩＦＯメモリを用いることが多い。

【００１５】

ＦＩＦＯメモリ１０６は、Ｆｉｒｓｔ　Ｉｎ　Ｆｉｒｓｔ　Ｏｕｔ　の規則でデータの入出力を行う名前付きパイプの総称であり、入力された順番に、データ出力がされる。１０５バックエンドから出力されたデータは、このＦＩＦＯメモリ１０６に入り、印字データ管理部１０７にて読み出される。

【００１６】

印字データ管理部１０７では、ＦＩＦＯメオリ１０６からのデータ入力を監視し、印字データが入力されてくると、順次データを読み出し、プリンタインターフェース部１１０にデータ出力をする。

【００１７】

ＵＩインターフェース部１０９は、プリンタステータス表示や、プリンタの停止、再開、中止などの操作をユーザーに提供するＵＩアプリケーション部１０８からのリクエストを監視し、各リクエストに応じて、プリンタインターフェース部１１０にコマンドの発行、ステータスの取得を行い、ＵＩアプリケーション部１０８に結果を返す処理を行っている。

【００１８】

プリンタインターフェース部１１０では、印字データ管理部１０７およびＵＩインターフェース部１０９からのデータおよびコマンドに応じて、プリンタ１１１へのデータ出力およびプリンタ１１１からのデータの読み出し処理を行っている。

【００１９】

バックエンド処理部１０５について、図４のフローチャートを用いて説明する。フィル

タ・レンダリング処理部１０４にて、印字データに展開されたデータが、バックエンド処理部１０５にて、パケット単位に分解され、ＦＩＦＯメモリ１０６に出力している。

【００２０】

まずステップＳ４０１では、印刷するデータのジョブＩＤの取得処理を行っている。次にステップＳ４０２では、処理の終了のシグナルを受信していないかをチェックしており、シグナル受信があった場合は、バックエンド処理を終了する。終了シグナルの受信がなかった場合は、ステップＳ４０３に進み、印字データの読み込み処理を行う。

【００２１】

ステップＳ４０４では、印字データの読み込み処理にて、データ終了が検出されたかのチェックを行っている。ここでデータ終了を検出した場合、バックエンド処理を終了する。データ終了が検出されず、正常にデータ読み込みが行われた場合、ステップＳ４０５に進む。ステップＳ４０５では、読み込んだデータの先頭にヘッダ情報を付加している。

【００２２】

ヘッダ情報とは、図２のジョブＩＤ２０５とレングス２０６を含み、ジョブＩＤ２０５は、ステップＳ４０１で取得したジョブＩＤをセットし、レングス２０６には、ステップＳ４０３で読み込んだデータサイズをセットする。

【００２３】

次にステップＳ４０６にて、ＦＩＦＯメオリ１０６にデータ出力処理を行う。データ出力処理を完了すると、ステップＳ４０２に戻り前述の処理を繰り返し処理する。

【００２４】

図３は、バックエンド処理部１０５にてパケット化されたデータがＦＩＦＯメオリ１０６経由で印字データ管理部１０７に流れる様子を示している。

【００２５】

３０１は、ジョブＩＤが０００１に対応するバックエンドのプロセスであり、３０２はジョブＩＤが０００２に対応するバックエンドのプロセスを示している。３０３はＦＩＦＯメオリであり、ＦＩＦＯメモリ内にジョブＩＤが０００１のパケット３０４、３０５、３０７、３０９があり、ジョブＩＤが０００２のパケット３０６、３０８もＦＩＦＯメモリ内にある。ＦＩＦＯメオリ３０３にパケットデータが入力された順番は、３０４～３０９の昇順であり、３０４から順次データが取り出される事になる。

【００２６】

通常１つの印刷ジョブの場合、３０２のバックエンドは無く、１つのバックエンドからのデータがＦＩＦＯメモリ３０３に出力され、印字データ管理部３１０にて、データの読み出しが行われる。

【００２７】

図３のＦＩＦＯメモリ３０３のような状態になるのは、マルチプロセスシステムにおいて、複数の印刷要求が発行され、処理中のジョブがキャンセルされ、次のジョブの印刷が開始されるタイミングにおいて、プロセスの処理順によって、ＦＩＦＯメモリ３０３内にキャンセルされたジョブのデータが残っている状態で、次のジョブが動作し始めることにより発生する。

【００２８】

また、一般的には、ジョブ管理処理は、１つのジョブのみを動作させ、ジョブ完了を待って、次のジョブを起動するようにしているが、ジョブ管理処理が、データ展開処理等をマルチジョブで動作可能なように制御する場合にも、データ出力処理の入り口であるＦＩＦＯメモリに複数ジョブのデータが混在して入力されてくる場合がある。

【００２９】

ＦＩＦＯメモリにて混在してしまった印字データに対しての処理について、図５を用いて印字データ管理処理について説明する。

【００３０】

ステップＳ５０１では、初期設定処理を行っている。初期処理では、処理中のジョブＩＤであるカレントＪ（Ｊｏｂ）ＩＤおよびキャンセルされたジョブＩＤであるキャンセル

Ｊ（Ｊｏｂ）ＩＤ格納領域の初期化を行っている。

【００３１】

ステップＳ５０２では、ＦＩＦＯメモリからデータ読み込みが可能であるまで処理を待機している。この待機処理では、シグナル受信した場合も、待機状態が解除されるので、次のステップＳ５０３において、プロセスの終了に関するシグナル受信があったか否かのチェックを行っている。プロセス終了シグナルを受信していた場合は、ステップＳ５１１にて確保領域の開放などの後処理を行い、プロセスを終了する。プロセス終了シグナル受信が無かった場合は、ステップＳ５０４に進み、ＦＩＦＯメモリからパケットデータの読み込みを行う。ステップＳ５０４では、パケットデータのヘッダ部分を読み込んでおり、ヘッダ内のレングス情報分の印字データを続くステップＳ５０５にて読み込んでいる。

【００３２】

ステップＳ５０４およびステップＳ５０５の読み込み処理にて、データ終了が検出されたかのチェックをステップＳ５０６にて行っている。データ終了が検出された場合、ステップＳ５１０にてＦＩＦＯメモリの初期化、カレントジョブＩＤ領域の初期化を行ってから、ステップＳ５０２に戻り、データ入力の待機状態に戻る。

【００３３】

ステップＳ５０６にて、データの終了が検出されなかった場合、ステップＳ５０７に進み、読み込んだパケットデータの妥当性のチェックを行っている。

【００３４】

パケットデータの妥当性チェックに関しては、図８を用いて説明する。図８のステップＳ８０１では、カレントジョブＩＤがセットされているかチェックしている。セットされていない場合は、ステップＳ８０７にて、現在のジョブＩＤをカレントジョブＩＤ領域にセットして、ステップＳ８０６に進む。ステップＳ８０６では、パケットデータ有効として、本ルーチンをリターンする。

【００３５】

ステップＳ８０１にてカレントジョブＩＤがセットされていると判定された場合、ステップＳ８０２に進み、受信したパケットデータのジョブＩＤと、キャンセルジョブＩＤ領域のジョブＩＤが同じであるかのチェックを行っている。キャンセルジョブＩＤと同一のジョブＩＤである場合は、読み込んだパケットデータは、キャンセルされたジョブのデータであると判断し、無効なものとして、ステップＳ８０８に進み、プリンタへ転送前のパケット化されたデータをデータ無効としてリターンする。

【００３６】

ステップＳ８０２にてキャンセルジョブＩＤと同一でない場合、ステップＳ８０３に進み、読み込んだパケットデータのジョブＩＤとカレントジョブＩＤが同じであるか否かを判定している。カレントジョブＩＤと同じ場合は、ステップＳ８０６に進み、有効データであるとしてリターンする。

【００３７】

ステップＳ８０３にてカレントジョブＩＤと同じでないと判断された場合、ステップＳ８０４に進む。

【００３８】

読み込んだデータがカレントジョブＩＤと異なり、かつ、キャンセルジョブＩＤとも異なっている場合、処理していたジョブのデータが、切れてしまい、別のジョブのデータを読み込んだと判断し、このデータを有効データとして、また、ここまで処理していたジョブのＩＤのデータを、無効データとして処理するため、まず、ステップＳ８０４にて、ここまで処理対象としていたカレントジョブＩＤをキャンセルジョブＩＤ領域にセットし、そしてステップＳ８０５にて、読み込んだパケットデータのジョブＩＤをカレントジョブＩＤにセットする。読み込んだパケットデータは、ステップＳ８０６にて有効データとしてリターンする。

【００３９】

図５の説明に戻ると、ステップＳ５０７のパケットデータチェック処理におけるデータ

が有効であるか否かの出力に対して、ステップＳ５０８にて、判定を行う。無効データである場合は、ステップＳ５０２に戻り、データ入力の待機状態に戻る。

【００４０】

ステップＳ５０８にてパケットデータが有効であると判定された場合、ステップＳ５０９にて、パケットデータの印字データ部分の出力処理を行う。本実施例では、印字データを１１１プリンタに直接出力するのではなく、プリンタインターフェース部１１０に対してデータ出力し、プリンタインターフェース部１１０では、ＵＩインターフェースからのコマンドと、印字データそれぞれに対して、プリンタとのデータ入出力制御を行っている。

【００４１】

次にＵＩインターフェース部１０９に関して、図６を用いて説明する。

【００４２】

まずステップＳ６０１では、ＵＩインターフェース処理で使用するデータ領域の初期化を行っている。

【００４３】

次にステップＳ６０２にて、ＵＩアプリケーションからのリクエストコマンドの待機状態になる。このコマンド待機状態では、シグナルを受信した場合も待機状態が解除されるので、ステップＳ６０３にて、プロセス終了シグナルの受信があったか否かのチェックを行っている。プロセス終了シグナル受信があった場合は、ステップＳ６１３に進み、ＵＩインターフェース処理の後処理を行って、処理を終了する。

【００４４】

ステップＳ６０３にてプロセス終了シグナルの受信が無い場合は、ステップＳ６０４にてＵＩアプリケーションからのリクエストコマンドの受信を行う。

【００４５】

受信したリクエストコマンドの種類のチェックを、ステップＳ６０５、Ｓ６０７、Ｓ６０９、Ｓ６１１にて行っており、ステップＳ６０５では、プリンタステータス取得要求コマンドであるかをチェックしている。プリンタステータス取得要求コマンドである場合は、ステップＳ６０６にて、プリンタインターフェース処理部１１０を介してプリンタステータスの取得を行い、取得したステータス情報のレスポンスをＵＩアプリケーション部１０８に返却する。

【００４６】

ステップＳ６０７では、リクエストコマンドが印刷停止要求コマンドであるかのチェックを行っている。印刷停止要求コマンドである場合、ステップＳ６０８にて、プリンタ停止コマンドをプリンタインターフェース処理部１１０を介してプリンタに送信する。

【００４７】

ステップＳ６０９では、リクエストコマンドが印刷再開要求コマンドであるかのチェックを行っている。印刷再開要求コマンドである場合、ステップＳ６１０にて、プリンタ動作再開コマンドをプリンタインターフェース処理部１１０を介してプリンタに送信する。

【００４８】

ステップＳ６１１では、リクエストコマンドが印刷中止要求コマンドであるかのチェックを行っている。印刷中止要求コマンドである場合、ステップＳ６１２にて、ジョブキャンセル処理を行う。各リクエストコマンドの種類に応じた処理を行ったら、ステップＳ６０２に戻り、次のＵＩリクエストコマンドの受信待機状態に戻る。

【００４９】

ステップＳ６１２のジョブキャンセル処理について、図７を用いて説明する。ジョブキャンセル操作がユーザーになされた場合、ジョブ管理側とプリンタ側の双方に印刷キャンセルに関する通知を行う必要がある。

【００５０】

まず、ステップＳ７０１では、ジョブ管理部１０３に、ジョブキャンセル要求を発行している。次にステップＳ７０２では、キャンセルジョブＩＤ領域に、カレントジョブＩＤ

の値をセットしている。次にステップＳ７０３では、プリンタインターフェース処理部１１０を介して、印刷キャンセル要求をプリンタに送信する。

【００５１】

以上説明してきたように、上記実施の形態によれば、ＦＩＦＯメモリに、プリンタへ送信前の複数の印刷ジョブのデータが混在する状況下で、特定の印刷ジョブのキャンセル指示が入った場合でも、キャンセル対象となっている特定の印刷ジョブに関連するデータを確実に破棄できる。

【００５２】

［その他の実施の形態］

以上が本発明の実施の形態の説明であるが、本発明は、これら実施の形態に限られるものではなく、特許請求の範囲で示した機能、または実施の形態の構成が持つ機能を達成できる構成であれば、どのようなものであっても適用可能である。

【００５３】

また、本発明の目的は、前述した実施の形態の機能を実現するソフトウェアのプログラムコードを記憶した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（またはＣＰＵやＭＰＵ等）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体及びプログラムは本発明を構成することになる。

【００５４】

また、プログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード、ＲＯＭ等を用いることができる。

【００５５】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているＯＳ（オペレーティングシステム）等が実際の処理の一部または全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【００５６】

更に、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵ等が実際の処理の一部または全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【図面の簡単な説明】

【００５７】

【図１】印刷システム構成図

【図２】印字データのパケット化

【図３】ＦＩＦＯ経由でのパケットデータの流れ

【図４】バックエンド処理フローチャート

【図５】印字データ管理処理フローチャート

【図６】ＵＩインターフェース処理フローチャート

【図７】ジョブキャンセル処理フローチャート

【図８】ジョブＩＤチェック処理フローチャート

【符号の説明】

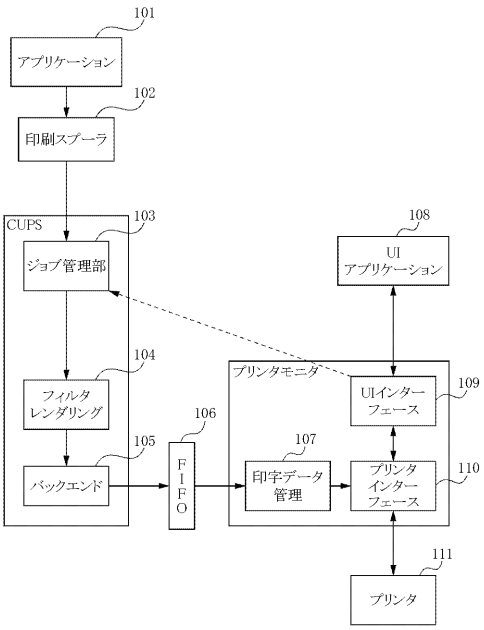
【００５８】

１０１　（印刷を行う）アプリケーション

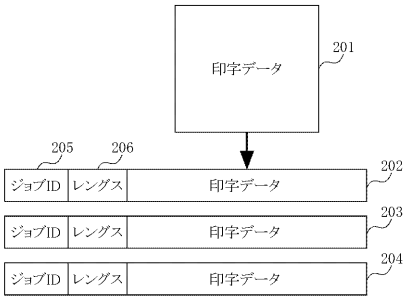
１０２　印刷スプーラ

１０３　ジョブ管理部
１０４　フィルタ・レンダリング処理部
１０５　バックエンド
１０６　ＦＩＦＯ
１０７　印字データ管理
１０８　ＵＩアプリケーション
１０９　ＵＩインターフェース処理
１１０　プリンタインターフェース処理
１１１　プリンタ

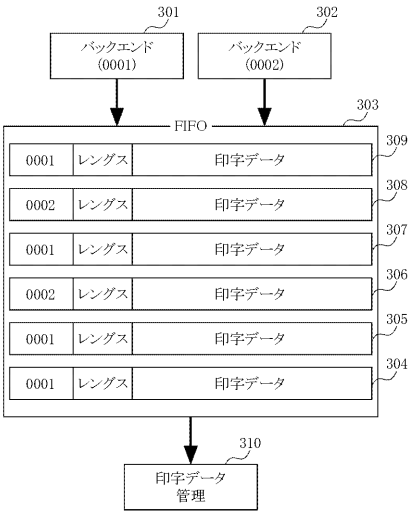
【図１】



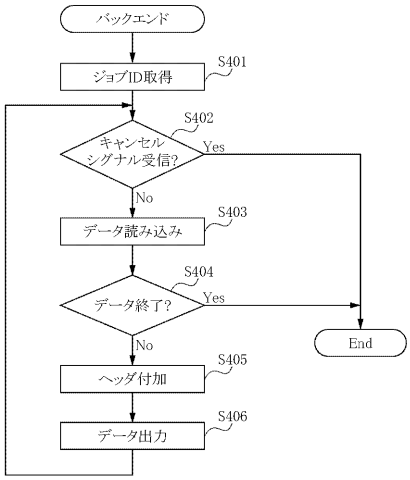
【図２】



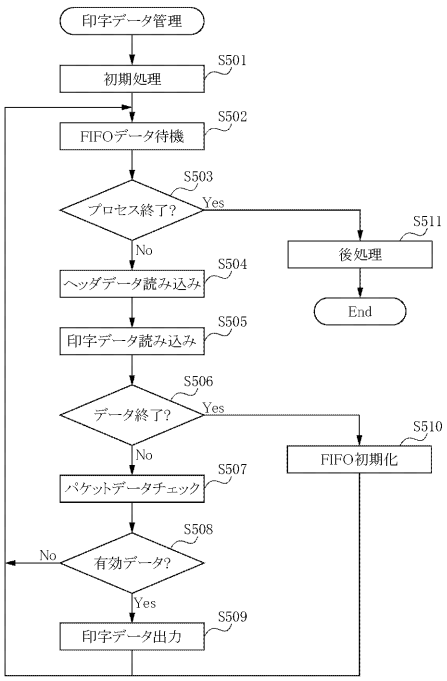
【図３】


301
302
バックエンド
(0001)
バックエンド
(0002)
303
FIFO
0001 レングス 印字データ
309
0002 レングス 印字データ
308
0001 レングス 印字データ
307
0002 レングス 印字データ
306
0001 レングス 印字データ
305
0001 レングス 印字データ
304
310
印字データ
管理

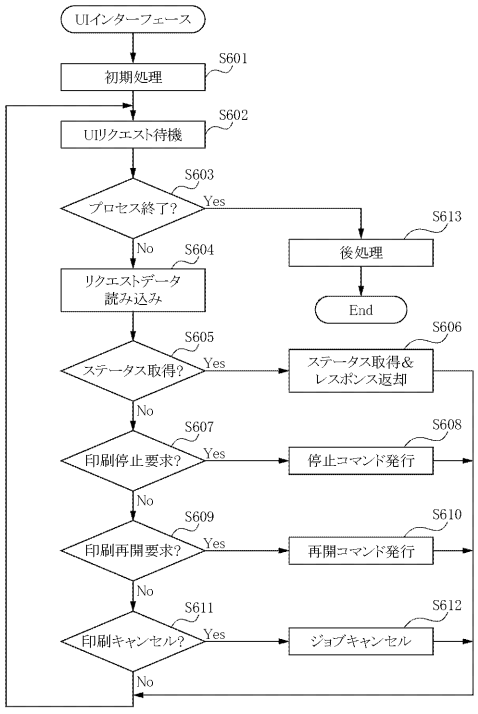
【図４】


バックエンド
S401
ジョブID取得
S402
キャンセル
シグナル受信?
Yes
No
S403
データ読み込み
S404
データ終了?
Yes
End
No
S405
ヘッダ付加
S406
データ出力

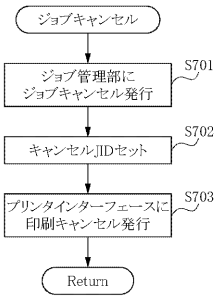
【図５】


印字データ管理
S501
初期処理
S502
FIFOデータ待機
S503
プロセス終了?
Yes
S511
後処理
End
No
S504
ヘッダデータ読み込み
S505
印字データ読み込み
S506
データ終了?
Yes
S510
FIFO初期化
No
S507
パケットデータチェック
S508
No
有効データ?
Yes
S509
印字データ出力

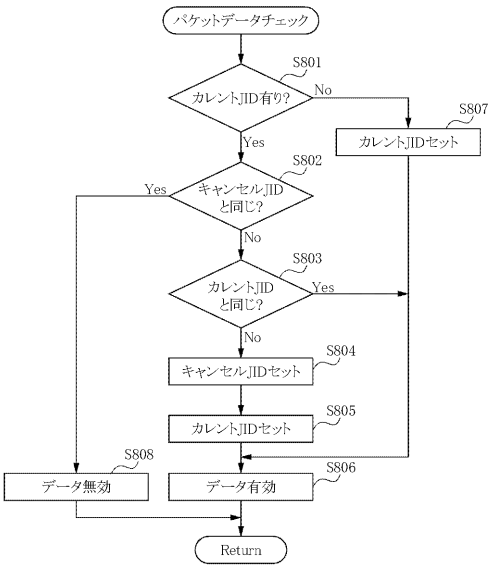
【図６】


UIインターフェース
S601
初期処理
S602
UIリクエスト待機
S603
プロセス終了?
Yes
S613
後処理
End
No
S604
リクエストデータ
読み込み
S605
ステータス取得?
Yes
S606
ステータス取得&
レスポンス返却
No
S607
印刷停止要求?
Yes
S608
停止コマンド発行
No
S609
印刷再開要求?
Yes
S610
再開コマンド発行
No
S611
印刷キャンセル?
Yes
S612
ジョブキャンセル
No

【図７】


ジョブキャンセル
ジョブ管理部に
ジョブキャンセル発行
S701
キャンセルJIDセット
S702
プリンタインターフェースに
印刷キャンセル発行
S703
Return

【図８】


パケットデータチェック
S801
カレントJID有り?
No
Yes
S807
カレントJIDセット
S802
キャンセルJID
と同じ?
Yes
No
S803
カレントJID
と同じ?
Yes
No
S804
キャンセルJIDセット
S805
カレントJIDセット
S808
データ無効
S806
データ有効
Return