

(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2008-102838
(P2008-102838A)

(43) 公開日 平成20年5月1日(2008.5.1)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 3/12 (2006.01)	G06F 3/12 C	2C061
B41J 29/00 (2006.01)	B41J 29/00 T	5B021
G06F 21/22 (2006.01)	G06F 3/12 K	5B276
	G06F 9/06 660J	

審査請求 未請求 請求項の数 4 O L (全 17 頁)

(21) 出願番号 特願2006-286407 (P2006-286407)
(22) 出願日 平成18年10月20日 (2006.10.20)

(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 名取 英夫
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) 2C061 AP01 CQ09 CQ12 CQ34 CQ41
5B021 CC00
5B276 FC00

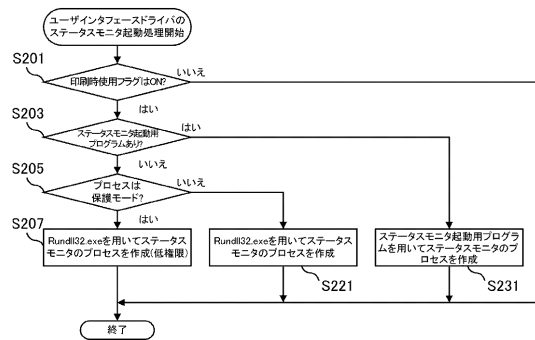
(54) 【発明の名称】 情報処理方法およびプログラム

(57) 【要約】

【課題】 保護モードで動作するプロセスから呼び出されたプリンタドライバが、プリンタの状態を表示するステータスマニタを起動する際に、保護モードにより表示される警告ダイアログの頻度を軽減する。

【解決手段】 ステータスマニタ起動部はプロセスが保護モードかどうかを判定する。保護モードと判定した場合、ステータスマニタ起動用プログラムの有無を判定し、ステータスマニタ起動用プログラムが存在しなければ低い権限でステータスマニタを起動し、逆に、ステータスマニタ起動用プログラムが存在すればこれを用いてステータスマニタを起動する。

【選択図】 図8



【特許請求の範囲】**【請求項 1】**

第一のプロセスから第二のプロセスを作成するためのプロセス作成方法であって、前記第一のプロセスが保護モードで動作しているかどうかを判定するステップと、前記第二のプロセスを作成するための起動用プログラムの存在の有無を判定するステップとを含み、

前記起動用プログラムが存在する場合は前記起動用プログラムを用いて前記第二のプロセスを作成し、

前記第一のプロセスが保護モードであり且つ前記起動用プログラムが存在しない場合は、低権限で動作するように前記第二のプロセスを作成するステップを含むことを特徴とするプロセス作成方法。

10

【請求項 2】

アプリケーションの描画命令を印刷コマンドとしてプリンタに供給するプリンタドライバであって、

前記プリンタドライバは少なくとも、

プリンタのステータスを表示するためのステータスマニタと、

前記アプリケーションの印刷要求に従って呼び出されるユーザインタフェースドライバとを含み、

前記ユーザインタフェースドライバは前記ステータスマニタのプロセスを作成するにあたり、

20

前記ユーザインタフェースドライバを呼び出したプロセスが保護モードかどうかの判定と、

前記ステータスマニタのプロセスを作成するための起動用プログラムの有無の判定を行い、

前記起動用プログラムが存在する場合は前記起動用プログラムを用いて前記ステータスマニタのプロセスを作成し、

前記ユーザインタフェースドライバを呼び出したプロセスが保護モードであり且つ前記起動用プログラムが存在しない場合は、低権限で動作するようにステータスマニタのプロセスを作成することを特徴とするプリンタドライバ。

【請求項 3】

30

前記ユーザインタフェースドライバは、

前記起動用プログラムを用いて前記ステータスマニタのプロセスを作成する前に、ユーザに操作方法を案内することを特徴とする請求項 2 記載のプリンタドライバ。

【請求項 4】

アプリケーションが印刷を開始した際に前記ステータスマニタを起動するかどうかを、前記ユーザインタフェースドライバで判定することを特徴とする請求項 2 あるいは請求項 3 記載のプリンタドライバ。

【発明の詳細な説明】**【技術分野】****【0001】**

40

本発明は、例えばプリンタにより印刷を行わせた場合等に、そのプリンタの状態を監視する機能を有する印刷制御方法及びプログラムに関する。

【背景技術】**【0002】**

プリンタによる印刷処理過程において、印刷を開始したものの紙詰まりで印刷が途中で停止したり、インク切れにより出力結果に何も印字されていなかったりといった状況は多々起こり得る。こういった場合、ユーザが印刷に失敗したプリンタを眺めて印刷失敗の原因を特定するのは困難である。そこでインク切れや紙詰まりといった異常なプリンタの状態（以下ステータス）は、LEDの点滅やブザーなどでユーザに知らされていた。

【0003】

50

しかし、プリンタが異常状態をユーザに通知する機能を有していても、ユーザがそのパターンを全て覚えておくか、マニュアル等を逐一参照しなければ、実際にどのようなステータス異常が起こっているかを理解することは出来なかった。また、プリンタの能力向上に伴い検知すべきステータスが増えたことで、全てのステータスを単純なパターンで表すのは困難となった。

【 0 0 0 4 】

そこで様々なプリンタステータスをユーザの手元で、グラフィカルに、かつリアルタイムに提示するツールとして、プリンタとコンピュータの双方向通信を利用したステータスマニタが開発された。

【 0 0 0 5 】

ステータスマニタによるプリンタの様々なステータスの表示の仕方には幾つかの実現手法がある。例えば、ステータスマニタが自らプリンタとの双方向通信をし、プリンタから返却された情報を元に行うものがある。また、マイクロソフト（登録商標）社のオペレーティングシステム（以下OS）の一つであるWindows（登録商標）では、プリントデータをスプールしてから出力する印刷サポート機能を有している。このOSの印刷サポート機能により、ランゲージモニタと呼ばれるモジュールが動作し、IEEE 1284 や USB 1.1 などに準拠した双方向通信によってコンピュータに接続された印刷装置との間で通信を行う。従って、各印刷ジョブの排紙完了や印刷装置内で発生したエラー情報などを知ることが可能となる。このランゲージモニタを介してプリンタから返却された情報をステータスマニタが得て、ステータスを表示することができる。

【 0 0 0 6 】

通常、ステータスマニタはプリンタドライバを構成するモジュールの一つとして実現される。

【 0 0 0 7 】

印刷中のプリンタのステータスを表示するために、通常はアプリケーションが印刷を実行した時に、プリンタドライバにより自動的にステータスマニタが起動される。また、プリンタドライバによっては、ユーザがステータスマニタを起動できるようなユーザインタフェースを備えているものもある。アプリケーションが印刷を実行した時に自動的にステータスマニタが起動される場合、実際にステータスマニタを起動するプリンタドライバのモジュールは、前述したランゲージモニタか、ユーザインタフェースを提供するユーザインタフェースドライバであることが多い。

【 0 0 0 8 】

ところが、ステータスマニタが自動的に起動されるとアプリケーションのフォーカスが移ってしまったり、ステータスマニタのダイアログが突然表示されたりするため、ユーザによってはステータスマニタが自動的に起動されることを嫌うこともあり得る。その対策として、印刷時にステータスマニタを起動するかどうかをユーザを指定させるための設定項目を持つステータスマニタが考えられている。例えば「印刷時にステータスマニタを使用する」という設定項目がOFFに設定されているのであれば印刷開始時にステータスマニタが表示されなくなる。

【 0 0 0 9 】

ここで、印刷を実行するアプリケーションであるが、印刷以外にも様々な機能を有する。通常、印刷を行うアプリケーションは印刷を行うための文書を作成するか、あるいはコンピュータやネットワーク上の記憶装置から文書を取得し印刷を行う。ここでアプリケーションにより取得される文書としては、ユーザ自身が過去に作成した文書や、他者が作成した文書が考えられる。仮にアプリケーションにセキュリティ上の欠陥があり、他者が作成した文書に悪意のある仕掛がしてあると、ユーザが使用するコンピュータは甚大な被害を被りかねない。特にインターネットと通信するアプリケーションは、不明なソースから信頼性の低いコンテンツをダウンロードするため、他のプログラムよりも悪用されるリスクが高くなっている。このためOSとアプリケーションの双方にセキュリティを向上させ、安全性を高めるための工夫がなされている。

10

20

30

40

50

【 0 0 1 0 】

例えばマイクロソフト（登録商標）社のOSであるWindows（登録商標） Vista（登録商標）で動作するアプリケーションの中には保護モードでの動作をサポートしているものがある。保護モードのプロセスは整合性レベルと呼ばれるOSのセキュリティ機能に基づいて動作する。保護モードは、権限を大幅に制限してプロセスを実行することによって、プロセス外部からの攻撃からユーザーやシステムを保護するための動作モードである。ここでプロセス外部からの攻撃とは、コンピュータ上でのデータの書き込み、変更、破壊や、あるいは悪意のある実行コードのインストールといったユーザに不利益をもたらす可能性のあるものを指す。保護モードで動作するプロセスの権限は著しく低いため、プロセス外部から攻撃された場合でも、システムやユーザのデータ領域が破壊されたり改竄される可能性は大幅に低減し、ユーザやシステムの重要なデータは守られることになる。

10

【 0 0 1 1 】

図1は保護モードと整合性レベルに関する概念図である。OSのセキュリティ機能として提供される整合性レベルは、プロセスごとにセキュリティオブジェクトへのアクセスを制限する。セキュリティオブジェクトとは書き換え可能な論理資源を指しており、具体的にはファイルやレジストリが挙げられる。

【 0 0 1 2 】

整合性レベルは「高」「中」「低」の3段階に分かれており、それぞれの段階でプロセスに与えられる特権と、書き込みが行える領域が異なる。この関係を図1(A)に示す。高い整合性レベルで実行されるプロセスは管理者特権が与えられ、システム管理等に使用される重要なセキュリティオブジェクト領域に書き込みが可能となる。中間の整合性レベルで実行されるプロセスはユーザ権限が与えられ、ユーザ専用のセキュリティオブジェクト領域に書き込みが可能となる。低い整合性レベルで実行されるプロセスは信頼性の低い権限が与えられ、整合性の低いプロセス専用のセキュリティオブジェクト領域にのみ書き込みが可能となる。

20

【 0 0 1 3 】

ここで高い整合性レベルで実行されるプロセスは、中間の整合性レベルまたは低い整合性レベルのセキュリティオブジェクト領域に書き込みが可能である。同様に中間の整合性レベルで実行されるプロセスは低い整合性レベルのセキュリティオブジェクト領域に書き込みが可能である。

30

【 0 0 1 4 】

保護モードと整合性レベルの関係を図1(B)に示す。保護モードで動作するプロセス100には、低い整合性レベルが得られている。つまり保護モードのプロセス100は低い権限で動作しているため、低い整合性レベルのセキュリティオブジェクト領域にのみ書き込み110が可能である。

【 0 0 1 5 】

一方、保護モードのプロセス100が中間の整合性レベルや高い整合性レベルのセキュリティオブジェクトには書き込みを行う場合は、ブローカープロセスと呼ばれる外部のプロセスを仲介する。ユーザ専用の領域へ書き込み112を行う場合は中間の整合性レベルが与えられたユーザブローカー122を介してアクセスを行う。この際にプロセス100は低い整合性レベルであるため、ユーザブローカー122にアクセスを委託する場合にユーザ権限が必要となる。同様に、システム管理等の重要な領域へ書き込み114を行う場合は高い整合性レベルが与えられた管理ブローカー124を介してアクセスを行う。この際にプロセス100は低い整合性レベルであるため、管理ブローカー124にアクセスを委託する場合に管理者権限が必要となる。

40

【特許文献1】特開2003-167718号公報

【特許文献2】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】

50

【 0 0 1 6 】

ところで、前記のような従来の技術には次のような解決すべき課題があった。

【 0 0 1 7 】

前述したように、通常、ステータスマニタは、アプリケーションが印刷を開始した時にアプリケーションのプロセスから呼び出されたユーザインタフェースドライバにより自動的に起動される。また、ユーザインタフェースドライバがステータスマニタを起動するためのユーザインタフェースを備えている場合は、ユーザが任意のタイミングでステータスマニタを起動することも可能である。つまり、ステータスマニタはプリンタドライバを呼び出したプロセスとは別のプロセスとして作成される。

【 0 0 1 8 】

しかし、ユーザインタフェースドライバを呼び出したプロセスが保護モードで動作している場合、ステータスマニタのプロセスに与えられる整合性レベルによっては、警告ダイアログが表示される場合がある。

【 0 0 1 9 】

保護モードで動作するアプリケーションが印刷を開始すると、OSの印刷サポート機能を介して、ユーザインタフェースドライバが呼び出される。そしてプリンタドライバはステータスマニタを起動する。例えばここで、プリンタドライバが中間の整合性レベルをステータスマニタのプロセスに与えると、ステータスマニタが起動される前に警告ダイアログが表示される。これは保護モードで動作するプロセスには低い整合性レベルが与えられているが、このプロセスから起動される別のプロセスの整合性レベルの方が高く、より広いセキュリティオブジェクトにアクセスできるためである。つまり、保護モードで動作するプロセスから呼び出されたユーザインタフェースドライバが、より高い権限をもつステータスマニタを起動するにあたり、ユーザの許可が必要になるためである。

【 0 0 2 0 】

図7(A)は警告ダイアログの一例である。プリンタドライバにより起動されるステータスマニタのモジュールが直接実行可能なファイル形式、すなわちファイル名の拡張子がEXEとなる実行ファイルである場合に、警告ダイアログ700が表示される。警告ダイアログ700の起動許可ボタン720をユーザが押下することで、ステータスマニタが起動される。起動許可ボタン720が押下される時にチェックボックス721がONになっていると、以後、ステータスマニタが起動される時に警告ダイアログ700が表示されなくなる。

【 0 0 2 1 】

図7(B)も警告ダイアログの一例である。プリンタドライバにより起動されるステータスマニタのモジュールが直接実行不可能な共有ライブラリ形式、すなわちファイル名の拡張子がDLLとなる実行ファイルをRundll32.exeで起動する場合に、警告ダイアログ750が表示される。警告ダイアログ750の起動許可ボタン770をユーザが押下することで、ステータスマニタが起動される。

【 0 0 2 2 】

警告ダイアログ750には、警告ダイアログ700のチェックボックス712に相当するものが存在しない。これは標準のDLL起動用プログラムRundll32.exeを用いてDLL形式の実行ファイルを起動する場合、DLL形式の実行ファイルの安全性をOSが保証できないためである。

【 0 0 2 3 】

以上のように、保護モードで動作するアプリケーションからOSの印刷サポート機能を介して呼び出されたユーザインタフェースドライバが、ステータスマニタを起動する場合、警告ダイアログが表示される。前述したようにステータスマニタはアプリケーションが印刷を行う度に起動されるため、印刷を行う度に図7で例示した警告ダイアログが表示される。従って、警告ダイアログの起動許可ボタンあるいは起動不許可ボタンを押下するまで印刷処理が開始されず、印刷処理の手順が複雑になってしまうことが問題となる。

【 0 0 2 4 】

10

20

30

40

50

また、印刷開始時にステータスマニタを使用しないように設定すれば警告ダイアログは表示されないと、ユーザは予想する可能性がある。しかし、ユーザインタフェースドライバにより起動されたステータスマニタ自身が、印刷時にステータスマニタを使用しない設定を参照している場合には問題となる。折角ユーザがステータスマニタを使用しないように設定を行っても、印刷開始時にステータスマニタが起動されてしまうため警告ダイアログが表示されてしまう欠点がある。

【0025】

本発明は上記従来の問題点に鑑み、保護モードで動作するアプリケーションにより表示される警告ダイアログの出現頻度を抑制することを考慮した、ステータスマニタを有するプリンタドライバを提供することを目標とする。

10

【課題を解決するための手段】

【0026】

上記目的を達成するために、本発明のプリンタドライバは以下のような構成を有する。

【0027】

アプリケーションの描画命令を印刷コマンドとしてプリンタに供給するプリンタドライバであって、

前記プリンタドライバは少なくとも、

プリンタのステータスを表示するためのステータスマニタと、

アプリケーションの印刷要求に従ってOSの印刷サポート機能を介して呼び出されるユーザインタフェースドライバとを含み、

20

前記ユーザインタフェースドライバは前記ステータスマニタのプロセスを作成するにあたり、

前記ユーザインタフェースドライバを呼び出したプロセスが保護モードかどうかの判定と、

前記ステータスマニタのプロセスを作成するための起動用プログラムの有無の判定を行い、

前記起動用プログラムが存在する場合は前記起動用プログラムを用いて前記ステータスマニタのプロセスを作成し、

前記ユーザインタフェースドライバを呼び出したプロセスが保護モードであり且つ前記起動用プログラムが存在しない場合は、低権限で動作するようにステータスマニタのプロセスを作成ようにした。

30

【0028】

また、前記ユーザインタフェースドライバは、前記起動用プログラムを用いて前記ステータスマニタのプロセスを作成する前に、ユーザインタフェースドライバを呼び出した保護モードで動作するプロセスにより上記警告ダイアログが表示された場合の操作方法を、ユーザに案内するようにした。

【0029】

また、前記ユーザインタフェースドライバは、アプリケーションが印刷を開始した際に前記ステータスマニタを起動するかどうかを判定するようにした。

【発明の効果】

40

【0030】

上記構成により本発明によれば以下のような効果が得られる。

【0031】

(1) 保護モードで動作するプロセスから呼び出されたユーザインタフェースドライバから、保護モードの外で動作するステータスマニタのプロセスを作成する際に、保護モードのプロセスにより表示される警告ダイアログを抑制することが可能となり、快適な印刷環境をユーザに提供できる。

【0032】

(2) 保護モードのプロセスにより警告ダイアログが表示される場合には、警告ダイアログの操作方法を事前にユーザに案内することで、警告ダイアログが表示されないように

50

ユーザを誘導することができ、快適な印刷環境をユーザに提供できる。

【0033】

(3)印刷時にステータスマニタを使用するかどうかの判定を、ユーザインタフェースドライバがステータスマニタを起動する前に行うことで、印刷開始時のステータスマニタの起動を最低限に抑えることが可能になり、警告ダイアログが表示される頻度を抑えることが出来る。

【発明を実施するための最良の形態】

【0034】

[実施例1]

以下、図面に示す実施例に基づき本発明を詳細に説明する。

10

【0035】

<プリンタドライバの動作>

図2は、本発明におけるプリンタドライバの実施形態を説明するために、印刷システムをOSの印刷サポート機能とプリンタドライバを中心に表したブロック図である。

【0036】

図2において、本発明に関するプリンタドライバの中で特に重要となるモジュールは、グレーの網掛けがされたモジュールである。それらは、ユーザインタフェースドライバ205とステータスマニタ207、そしてステータスマニタ起動用プログラム233である。

【0037】

20

本実施例におけるステータスマニタ207のモジュールは直接実行不可能な共有ライブラリ形式、つまりDLL形式である実行ファイルを想定している。DLL形式の実行ファイルは直接実行できないため、独立したプロセスとして実行するためには、DLL形式の実行ファイルを呼び出すためのDLL起動用プログラムが必要となる。OSがWindows(登録商標)である場合は、OS標準の汎用的なDLL起動用プログラムとしてRundll32.exeというプログラムが提供されており、図2の231で示される。ステータスマニタ起動用プログラム233もDLL形式の実行ファイルを呼び出すためのものであるが、ステータスマニタ専用のDLL起動用プログラムとなっている。

【0038】

30

図2においてプリンタドライバとして提供されるモジュールはユーザインタフェースドライバ205、ステータスマニタ207、印刷データ編集処理部213、イメージ生成処理部215、そしてランゲージモニタ219となっている。ここで、ステータスマニタ起動用プログラム233はプリンタドライバに含めてもよいし、プリンタドライバには含めずに、プリンタドライバとは別のソフトウェアとしてインストールし、プリンタドライバに追加する形式にしてもよい。本実施例ではプリンタドライバには含めずに、プリンタドライバとは別のソフトウェアとしてインストールすることを想定している。

【0039】

ステータスマニタ207は様々なプリンタのステータスをユーザの手元で、グラフィカルに、かつリアルタイムに提示する機能を有している。ステータスマニタ207の提示機能の例については図4を用いて後述する。前述したようにステータスマニタ207はDLL形式の実行ファイルであるから、231で示されるRundll32.exeやステータスマニタ起動用プログラム233を介して、ユーザインタフェースドライバ205によって起動される。ステータスマニタ207は起動された後、後述するランゲージモニタ219と通信を行うことでプリンタのステータスを取得し、ユーザにステータスの提示を行う。

40

【0040】

ユーザインタフェースドライバ205は、一般に印刷に使用する用紙サイズや印刷方向、その他の属性を設定する機能を提供する。そして、これらの属性と設定値を印刷設定情報としてアプリケーションに返却する。同時に本発明におけるユーザインタフェースドライバ205はステータスマニタを起動するための機能を有している。そして、アプリケー

50

ションの印刷実行時にステータスマニタを起動したり、ユーザが任意のタイミングでステータスマニタを起動できるようにするためのユーザインタフェースを提供する。ユーザインタフェースドライバ205が提供するユーザインタフェースの例については図3を用いて後述する。

【0041】

アプリケーション201が作成した文書の各ページの印刷データは、OSの印刷サポート機能203を通じてスプールファイル209に一時的に蓄えられる。そして、これをプリンタドライバのモジュール群がイメージデータ(すなわちビットマップデータ)に展開し、プリンタが解釈可能な印刷コマンドに変換してプリンタ218に供給して印刷するよう構成されている。

10

【0042】

アプリケーション201は作成した任意の文書の印刷にあたる。通常、ユーザが望む印刷結果を得るために、ユーザインタフェースドライバ205のユーザインタフェースを介して印刷設定情報を作成する。そして、アプリケーション201はOSの印刷サポート機能203に対して、印刷開始や印刷終了、文書の各ページの描画開始や描画終了を要求する。ここで、OSの印刷サポート機能203はアプリケーション201により上記印刷に関する要求が行われる度に、印刷イベントをユーザインタフェースドライバ205に通知する。

【0043】

ユーザインタフェースドライバ205は印刷開始の印刷イベントを受け取ると、ステータスマニタ207を起動する。上述したようにステータスマニタ207は直接実行できないため、ステータスマニタ起動用プログラム233が存在すれば、これを用いてステータスマニタ207を起動する。ステータスマニタ起動用プログラム233が存在しなければ231で示されるRundll32.exeを用いてステータスマニタ207を起動する。ユーザインタフェースドライバ205がステータスマニタ207を起動する詳細な手順は図8を用いて後述する。

20

【0044】

アプリケーション201が印刷するために描画したデータはOSの印刷サポート機能により一旦スプールファイル209にスプールされる。そして、デスプール時にOSの印刷サポート機能によって、プリンタドライバのデスプールモジュール群211が呼び出される。デスプールモジュール群211はスプールファイル209に格納された描画データからプリンタ218が解釈可能な印刷コマンドに変換し、プリンタ218に供給する機能を有する。

30

【0045】

本実施例によけるプリンタドライバのデスプールモジュール群211は、印刷データ編集処理部213とイメージ生成処理部215、そしてランゲージモニタ219で構成されている。

【0046】

印刷データ編集処理部213は、スプールファイル209から印刷すべき文書ページの印刷データと印刷設定情報を読み出し、逆順印刷のように印刷するページ順序を入れ換えたり、割付け印刷(N-up)のように複数の文書ページを1枚の印刷用紙に合成したりする編集処理機能を有する。そして、印刷データ編集処理部213は、編集処理した後の印刷データをイメージ生成処理部215に供給する。

40

【0047】

イメージ生成処理部215は、まず、印刷データ編集処理部213から供給された編集処理済みの印刷データをもとにイメージデータに展開する。続いて、イメージ生成処理部215は、展開されたイメージデータをプリンタ218が解釈可能な印刷コマンドに変換し、変換した印刷コマンドをランゲージモニタ219に供給する。

【0048】

ランゲージモニタ219は、イメージ生成処理部215から供給された印刷コマンドを

50

解釈しつつ、プリンタ 2 1 8 に印刷コマンドを供給する。そして、プリンタ 2 1 8 のステータスを取得し、ステータスマニタ 2 0 7 と通信を行い、プリンタ 2 1 8 のステータス情報として送信する。

【 0 0 4 9 】

出力デバイスであるプリンタ 2 1 8 は、供給された印刷コマンドを解釈し、可視画像 2 2 3 を印刷用紙 2 2 1 に対して形成する機能を持つ。

【 0 0 5 0 】

<ユーザインタフェース>

図 3 は本実施形態におけるユーザインタフェースドライバが提供する印刷設定ダイアログのうち、プリンタのメンテナンスを行う機能を選択するためのユーティリティ画面の一例を示す図である。ここではメンテナンス機能として、ヘッドの調整やクリーニングなどの、プリンタの機能を維持したり印刷時の動作音を調整したりするような機能や、プリンタのステータスを確認するためにステータスマニタを起動させるための機能を想定している。

10

【 0 0 5 1 】

印刷設定ダイアログ 3 0 1 は、表示エリア 3 0 2 ~ 3 1 1、3 4 0 ~ 3 4 8 から構成される。印刷設定では表示する項目が多いため、タブシートを使用して設定項目を内容ごとに分けて見やすい構成にするのが通例である。本実施形態でもタブを使用する。

【 0 0 5 2 】

基本設定タブ 3 0 2 では、印刷の基本に関する内容を表示する。

20

【 0 0 5 3 】

ページ設定タブ 3 0 3 は、用紙の大きさ、印刷方向、割付け印刷 (N - u p) などの各種ページ設定に関する内容を表示する。

【 0 0 5 4 】

特殊効果タブ 3 0 4 は、各種画像補整やセピア調など特殊な効果を画像に与える場合に使用する機能の選択画面を表示する。

【 0 0 5 5 】

お気に入りタブ 3 0 5 は頻繁に使用する印刷設定などの保存と呼び出し機能を選択させる画面を表示する。

【 0 0 5 6 】

ユーティリティタブ 3 0 6 は、上述したようにステータスマニタを起動する機能や、ヘッドの調整やクリーニングなどの機能の選択画面を表示する。

30

【 0 0 5 7 】

ボタン 3 4 0 ~ 3 4 7 は、各々にプリンタを操作するための機能が割り当てられている。ユーザが各ボタンを押下することにより所定の印刷コマンドをスプールファイルを介してプリンタに送信し、ボタンで表現される各機能を実施することが可能になっている。クリーニングボタン 3 4 0 は印刷結果にかすれやスジ等が見られる場合に押下されることを想定しており、インクを吐出するオリフィスをクリーニングする機能を提供する。強力クリーニングボタン 3 4 1 はクリーニングボタン 3 4 0 よりも強力でオリフィスをクリーニングする機能を提供する。ヘッド位置調整ボタン 3 4 2 は吐出されたインクが印刷用紙で像を形成する際の精度を調整するために、プリンタヘッドの位置を調整する機能を提供する。ノズルチェックパターン印刷ボタン 3 4 3 はオリフィスにインクが固着し吐出不良が発生していないか確認するためのパターンを印刷する機能を提供する。電源オフボタン 3 4 4 はプリンタの電源をオフにする機能を提供する。自動電源設定ボタン 3 4 5 はプリンタの電源がオフであっても印刷コマンドが送信された場合自動的に電源をオンにする機能や、一定時間印刷コマンドが送信されない場合自動的に電源をオフにする機能を提供する。サイレント設定ボタン 3 4 6 はプリンタの印刷動作時の動作音を抑える設定するための機能を提供する。特殊設定ボタン 3 4 7 はプリンタのその他の動作モードの設定を行う機能を提供する。プリンタ状態の確認ボタン 3 4 8 を押下すると、ユーザインタフェースドライバはステータスマニタのプロセスを作成し、ステータスマニタの画面が表示されプリ

40

50

ンタのステータスを確認することができる。ステータスマニタの画面については図 4 を用いて後述する。

【 0 0 5 8 】

OK ボタン 3 0 8 またはキャンセルボタン 3 0 9 を押下すると印刷設定ダイアログ 3 0 1 が閉じる。ヘルプボタン 3 1 1 はユーティリティタブ 3 0 6 の各メンテナンス項目に関する説明文を別ウィンドウで表示させることができる。

【 0 0 5 9 】

< 印刷開始時におけるステータスマニタ起動 >

本実施例では、アプリケーションが印刷を開始する際に、ユーザインタフェースドライバがステータスマニタを起動する構成になっている。この手順を図 5 のタイミングチャートを用いて説明する。図 5 はアプリケーションの印刷開始時にユーザインタフェースドライバがステータスマニタを起動するタイミングチャートの一例である。

10

【 0 0 6 0 】

アプリケーションが印刷開始を OS の印刷サポート機能に要求すると (5 0 1)、OS の印刷サポート機能はユーザインタフェースドライバに印刷開始のイベントを通知する (5 0 3)。印刷開始のイベントを受け取ったユーザインタフェースドライバはステータスマニタを起動し、OS の印刷サポート機能に制御を返却する (5 0 5)。続けて OS の印刷サポート機能は実際の印刷開始の処理を行い、アプリケーションに制御を返却する (5 0 7)。このようにアプリケーションが印刷を実行する度にユーザインタフェースドライバはステータスマニタを起動する。

20

【 0 0 6 1 】

以上のように OS の印刷サポート機能がユーザインタフェースドライバに通知する印刷イベントを利用することで、ユーザインタフェースドライバがステータスマニタを起動することができる。ここでユーザインタフェースドライバはステータスマニタを起動する際の引数として、アプリケーションの印刷ジョブに関する情報を渡す。こうすることで、ステータスマニタはアプリケーションの印刷開始時に起動されたのか、あるいは、ユーザインタフェースのメンテナンス画面から起動されたのかを判定できるようになる。

【 0 0 6 2 】

なお、前述したように本実施例においてはステータスマニタは DLL 形式の実行ファイルであるから、 Rundll32.exe やステータスマニタ起動用プログラムを介して、ユーザインタフェースドライバによって起動される。ユーザインタフェースドライバによるステータスマニタ起動の手順は図 8 を用いて後述する。

30

【 0 0 6 3 】

< ステータスマニタ >

図 4 を用いてステータスマニタが表示する画面について説明する。ステータスマニタダイアログ 4 0 0 は表示エリア 4 0 1 ~ 4 1 3 で構成される。

【 0 0 6 4 】

メニュー 4 0 1 はステータスマニタの設定や表記機能を切替るためのメニューである。図示していないが、オプションメニューからは、アプリケーションの印刷時にステータスマニタを使用するかどうかを設定することができる。インクについてメニューを選択すると、表示中のプリンタで使用されるインクタンクやインクカートリッジの製品型番が、図示しない別ウィンドウにグラフィカルに表示される。ヘルプメニューを選択するとステータスマニタについてのヘルプ画面が表示される。

40

【 0 0 6 5 】

プリンタ情報タブ 4 0 3 にはプリンタに関するステータスが表示される。表示エリア 4 0 5 にはプリンタのステータスとインクの残量等がグラフィカルに表示される。表示エリア 4 0 7 は、プリンタのステータスとインクの残量等が文章で説明される。表示エリア 4 0 9 は印刷中のジョブがあれば、ジョブの所有者とジョブの名称、現在印刷中のページ数と総ページ数が表示される。印刷中止ボタン 4 1 1 を押下すると、印刷中のジョブがあれば印刷を中止することができる。印刷待ち一覧を表示ボタン 4 1 3 を押下すると、OS が

50

標準機能として提供する図示しないキューウィンドウを呼び出すことができる。

【 0 0 6 6 】

なお、オプションメニューから印刷時にステータスマニタを使用するかどうかを設定することができるのは、アプリケーションが印刷を開始する度にステータスマニタが起動することをユーザが嫌う可能性があるためである。そのようなユーザはステータスマニタのオプションメニューから「印刷時にステータスマニタを使用する」設定をOFFにすれば良い。この「印刷時にステータスマニタを使用する」設定に着目したステータスマニタの処理手順について図6を用いて説明する。

【 0 0 6 7 】

図6は「印刷時にステータスマニタを使用する」設定に着目したステータスマニタの処理のフローチャートである。ステータスマニタは起動すると、まずS101で起動時にユーザインタフェースドライバから渡された引数に基づいて、アプリケーションの印刷開始時における起動かどうかを判定する。アプリケーションの印刷開始時における起動ならばS103に移り、アプリケーションの印刷開始時における起動でないならばS105に移る。

10

【 0 0 6 8 】

S103では共通記憶領域から印刷時使用フラグを読み込み、印刷時使用フラグがONかどうかを判定する。もしONであればS105に移り、ONでなければステータスマニタの処理を終了する。印刷時使用フラグは上述した「印刷時にステータスマニタを使用する」設定の設定内容に対応する。なお、プリンタドライバインストール時における印刷時使用フラグの初期値がONであることは言うまでもない。

20

【 0 0 6 9 】

S105からS115がステータスマニタの処理ループとなっている。ここでは「印刷時にステータスマニタを使用する」設定に関してのみの処理を示す。S107ではユーザにより、「印刷時にステータスマニタを使用する」がOFFに設定されたかどうかを判定する。OFFに設定された場合はS109に移り、それ以外の場合はS111に移る。S109では印刷時使用フラグをOFFにする。

【 0 0 7 0 】

続いてS111では「印刷時にステータスマニタを使用する」がONに設定されたかを判定する。ONに設定された場合はS113に移り、それ以外の場合はS115に移る。S113では印刷時使用フラグをONにする。

30

【 0 0 7 1 】

S115はステータスマニタの処理ループの終端であり、終了条件はステータスマニタが起動された条件により異なる。アプリケーションの印刷開始時にステータスマニタが起動された場合は、ユーザインタフェースドライバにより引数で指定された印刷ジョブの印刷処理が完了したことが終了条件となる。ユーザインタフェースドライバのメンテナンス画面から起動された場合は、ユーザによって明示的にステータスマニタの終了が指示されたことが終了条件となる。

【 0 0 7 2 】

続いてS117では印刷時使用フラグを、ファイルやレジストリ等の共通記憶領域に保存する。

40

【 0 0 7 3 】

このように、印刷時にステータスマニタを使用するかどうかの設定を、印刷時使用フラグとして保存することで、ステータスマニタだけではなくユーザインタフェースドライバも参照することができるようになる。なお、本実施例のステータスマニタは、S101とS103で印刷時にステータスマニタを使用するかどうかを判定しているが、この処理は必須ではない。なぜならば図8を用いて後述するように、本実施例ではユーザインタフェースドライバがステータスマニタを起動する前に判定を行うためである。ただし、従来のステータスマニタの処理を強調するために、図6のフローチャートもS101とS103の判定処理を入れ、従来の手順のままに示した。

50

【 0 0 7 4 】

< ステータスマニタ起動処理 >

さて、以上説明したように、本実施例ではユーザインタフェースドライバがステータスマニタを起動する。ここで、ユーザインタフェースドライバを呼び出したプロセスが保護モードで動作している場合は、上述したようにステータスマニタを起動した時点で警告ダイアログが表示される。この警告ダイアログが表示される頻度を抑制するための、ユーザインタフェースドライバのステータスマニタの起動手順を図 8 のフローチャートに示す。ただし、図 8 ではアプリケーションが印刷を開始した際に着目したフローチャートとなっている。

【 0 0 7 5 】

まず、S 2 0 1 で印刷時にステータスマニタを使用する設定がされているかどうかを見るために、共通記憶領域から印刷時使用フラグを読み出し ON かどうかを判定する。印刷時使用フラグが ON であれば S 2 0 3 に移り、ON でなければ終了する。

【 0 0 7 6 】

S 2 0 3 では、ステータスマニタ起動用プログラムが印刷システムの記憶領域の中に存在するかどうかを判定する。ステータスマニタ起動用プログラムが存在しなければ S 2 0 5 に移り、存在すれば S 2 3 1 に移る。S 2 3 1 ではステータスマニタ起動用プログラムを介して、ステータスマニタのプロセスを作成した後に終了する。S 2 3 1 で作成するステータスマニタのプロセスには、ユーザ権限を与えると良い。ここで、ユーザインタフェースドライバを呼び出したプロセスが保護モードで動作している場合は、ユーザ権限で動作するステータスマニタのプロセスは保護モード（つまり低権限）ではないため、保護モードで動作するプロセスにより図 7 (A) に例示される警告ダイアログが表示される。逆にユーザインタフェースドライバを呼び出したプロセスが保護モードで動作していない場合は、警告ダイアログは表示されない。

【 0 0 7 7 】

S 2 0 5 では、ユーザインタフェースドライバを呼び出したプロセスが保護モードかどうかを判定する。保護モードでなければ S 2 0 7 に移り、保護モードであれば S 2 2 1 に移る。S 2 2 1 では OS 標準の D L L 起動用プログラムの R u n d 1 1 3 2 . e x e を介して、ステータスマニタのプロセスをユーザインタフェースドライバを呼び出したプロセスと同じ権限で作成する。ここではユーザインタフェースドライバを呼び出したプロセスは保護モードではないから、警告ダイアログは表示されない。S 2 2 1 でステータスマニタを起動したあとは、ユーザインタフェースドライバのステータスマニタ起動処理を終了する。

【 0 0 7 8 】

S 2 0 7 では上述した R u n d 1 1 3 2 . e x e を介して、ステータスマニタのプロセスを低権限で作成した後に、ユーザインタフェースドライバのステータスマニタ起動処理を終了する。ここで低権限とは保護モードと等価であるから、警告ダイアログは表示されない。

【 0 0 7 9 】

このように、ユーザインタフェースドライバがステータスマニタのプロセスを作成するにあたり、ユーザインタフェースドライバを呼び出したプロセスが保護モードかどうかを判定し、また、ステータスマニタ起動用プログラムの存在を判定し、条件に応じて適切な権限を与えてステータスマニタを起動することにより、警告ダイアログの表示を最低限に抑えることができる。また、S 2 3 1 により図 7 (A) で例示される警告ダイアログが表示されたとしても、上述したようにチェックボックス 7 1 2 を ON にして許可するボタン 7 2 0 を押下した後は、本警告ダイアログは表示されなくなる。

【 0 0 8 0 】

ここで、ユーザが図 3 のボタンを押下してステータスマニタを起動する場合においても、保護モードのプロセスによる警告ダイアログの表示頻度を抑制することが、快適な印刷環境をユーザに提供するうえで重要である。この場合は、図 8 の S 2 0 3 から処理を開始

10

20

30

40

50

することで、表示頻度の抑制が実現できる。

【0081】

さて、S207において低権限のプロセスとして作成されたステータスマニタは、書き込みが可能なセキュリティオブジェクト領域が限られているため、その動作の一部に制限が発生することがある。具体的には図4で例示したステータスマニタダイアログの印刷待ち一覧を表示ボタン413を押下することによって表示されるOS標準のキューウィンドウもまた低権限で動作するため、キューウィンドウからの印刷中止の操作ができないといった副作用がある。しかし、本実施例ではステータスマニタの画面の印刷中止ボタン411を押下すれば、印刷中止要求をステータスマニタからランゲージモニタに通知し、ランゲージモニタが印刷中止を行えるので、上記の動作制限は問題にならないことは言うまでもない。

10

【0082】

[実施例2]

以下、図9と図10に示す実施例にもとづき本発明を詳細に説明する。本実施例は、ユーザインタフェースドライバがステータスマニタを起動する前に警告ダイアログの操作方法の説明を表示し、ユーザを正しく導くことで、警告ダイアログが表示される頻度を更に下げることの特徴とする。なお、図8以外の図は実施例1と同一であるため説明を省く。

【0083】

<ステータスマニタ起動処理>

本実施例におけるユーザインタフェースドライバのステータスマニタ起動処理のフローチャートを図9に示す。実施例1の図8と図9(A)の差異はS231におけるステータスマニタのプロセス作成の前に、S301でユーザに警告ダイアログの操作案内をすることだけである。S301の処理については図9(B)を用いて後述する。図9(A)の他の説明は図8と同一であるため、ここでは記載を省略する。

20

【0084】

図9(B)はユーザへの警告ダイアログの操作方法を案内するためのフローチャートであり、図9(A)のS301の処理を詳細に示している。S311では、ユーザインタフェースドライバを呼び出したプロセスが保護モードかどうかを判定する。保護モードであればS313に移り、保護モードでなければ終了する。S313では、過去に図7(A)の警告ダイアログのチェックボックス712がONの状態ですべて許可ボタン712が押下されたかどうかを調べる。警告ダイアログは、保護モードで動作するユーザインタフェースドライバを呼び出したプロセスで表示されており、チェックボックスがONの状態ですべてプロセスの作成が許可されたかどうかは、ユーザインタフェースドライバから参照なのである。上述したように、ユーザが一度、チェックボックスをONにした状態で許可ボタン720を押下していれば、警告ダイアログは二度と表示されないのであるから、ユーザへ警告ダイアログの操作方法を案内する必要は無い。従ってS313で、チェックボックス712がONで許可ボタン720が押されていたと判断された場合は操作案内処理を終了し、チェックボックス712がONで許可ボタン720が押されていたと判断されなかった場合はS314に移る。S314では、これから表示される、警告ダイアログでの操作方法を表示した後、操作案内処理を終了する。

30

40

【0085】

S314の操作案内は、具体的には、次に表示される警告ダイアログで「常に許可する」チェックボックス712をONにした状態で許可ボタンを押下するように促すメッセージを表示する。このメッセージの一例を図10のダイアログに示す。操作案内ダイアログ1000はメッセージ表示領域を持ち、文章で操作方法の説明を行う。OKボタン1002を押下すると操作案内ダイアログ1000が閉じる。また、操作案内ダイアログ1000では文章で操作説明を行っているが、グラフィカルに図7(A)の模式図を表示して、より解りやすくユーザに説明することもできる。

【0086】

ユーザは図10のダイアログのメッセージを読みOKボタン1002を押下すると、処

50

理がS 2 3 1に移り、ユーザインタフェースドライバはステータスマニタを起動しようとする。ステータスマニタが起動される前に、図7(A)の警告ダイアログが表示され、ユーザは図10で説明されたように警告ダイアログを操作することが可能になる。

【0087】

以上説明したように、本実施例では、ユーザインタフェースドライバがステータスマニタを起動する前に警告ダイアログの操作方法の説明を表示し、ユーザを正しく導くことで、警告ダイアログが表示される頻度を更に下げることができる。

【0088】

以上説明したこれらの実施形態によれば、保護モードのプロセスから呼び出されるプリンタドライバがステータスマニタを起動する場合において、次のような効果が得られる。

10

【0089】

(1) 保護モードで動作するプロセスから呼び出されたユーザインタフェースドライバから、保護モードの外で動作するステータスマニタのプロセスを作成する際に、保護モードのプロセスにより表示される警告ダイアログを抑制することが可能となり、快適な印刷環境をユーザに提供できる。

【0090】

(2) 保護モードのプロセスにより警告ダイアログが表示される場合には、警告ダイアログの操作方法を事前にユーザに案内することで、警告ダイアログが表示されないようにユーザを誘導することができ、快適な印刷環境をユーザに提供できる。

【0091】

(3) 印刷時にステータスマニタを使用するかどうかの判定を、ユーザインタフェースドライバがステータスマニタを起動する前に行うことで、印刷開始時のステータスマニタの起動を最低限に抑えることが可能になり、警告ダイアログが表示される頻度を抑えることが出来る。

20

【図面の簡単な説明】

【0092】

【図1】保護モードと整合性レベルに関する概念図。

【図2】本発明の実施形態に係わるプリンタドライバのブロック図。

【図3】印刷設定ダイアログの一例を示す図。

【図4】ステータスマニタのダイアログの一例を示す図。

30

【図5】ステータスマニタ起動に関するタイミングチャート。

【図6】ステータスマニタの処理の一例を示すフローチャート。

【図7】保護モードのプロセスにより表示される警告ダイアログの一例を示す図。

【図8】本発明の実施形態に係わるステータスマニタ起動手順を示すフローチャート。

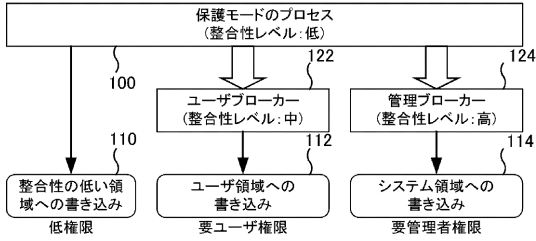
【図9】本発明の実施形態に係わるステータスマニタ起動手順を示すフローチャート。

【図10】本発明の実施形態に係わる操作説明を行うダイアログの一例を示す図。

【図1】

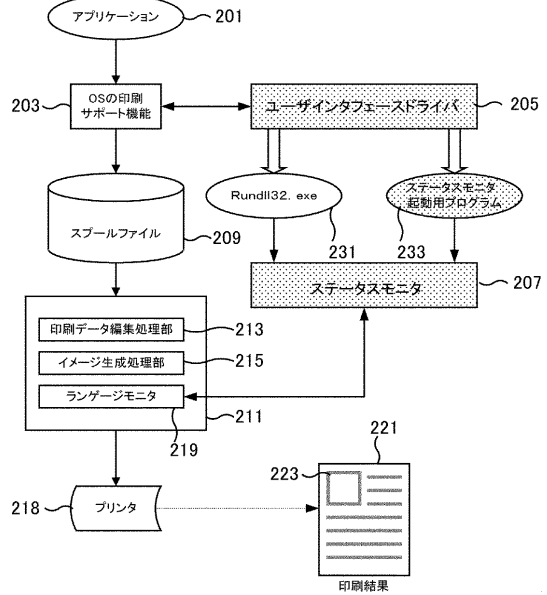
整合性レベル	プロセスに与えられる特権と書き込み可能な領域	
高	管理者特権	システム管理等の重要なファイル領域 システム管理等の重要なレジストリ領域
中	ユーザ特権	ユーザ専用のファイル領域 ユーザ専用のレジストリ領域
低	信頼性の低い権限	整合性が低いプロセス専用の ファイル領域やレジストリ領域

(A)

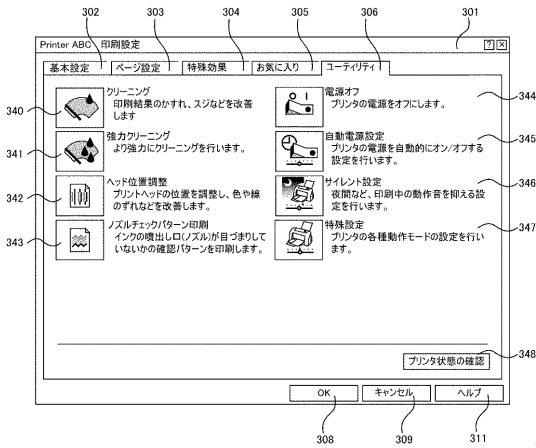


(B)

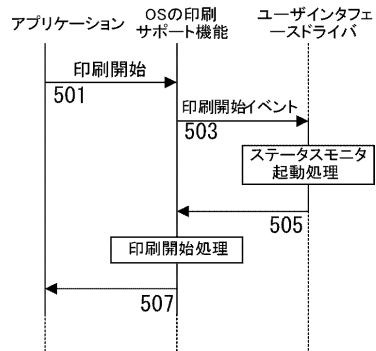
【図2】



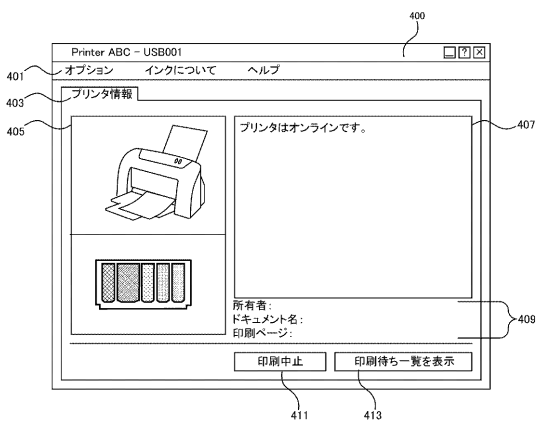
【図3】



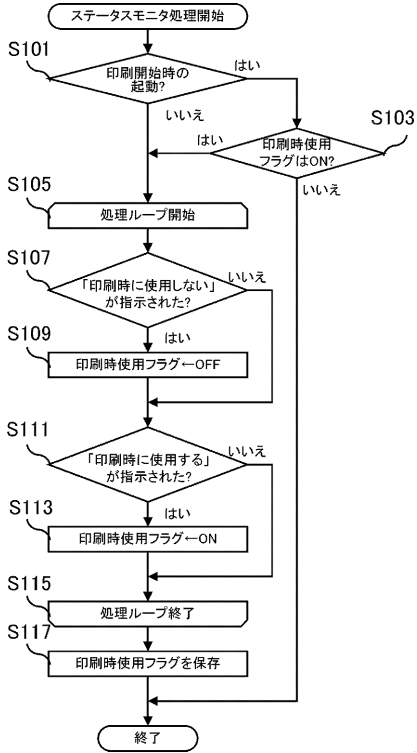
【図5】



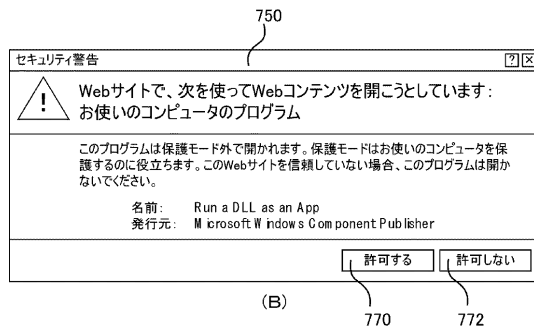
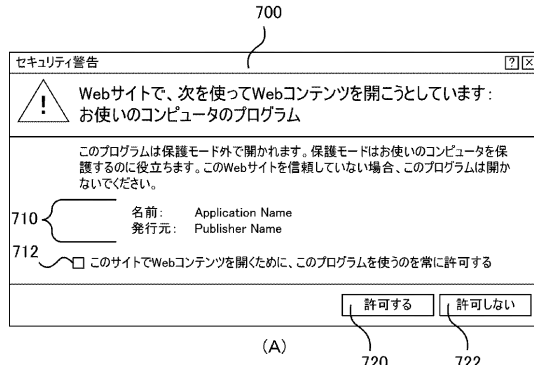
【図4】



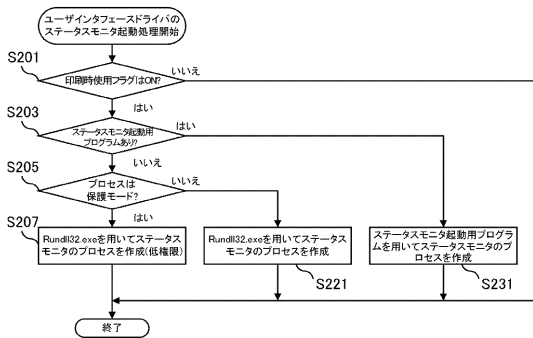
【図6】



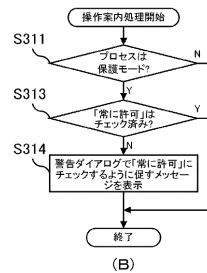
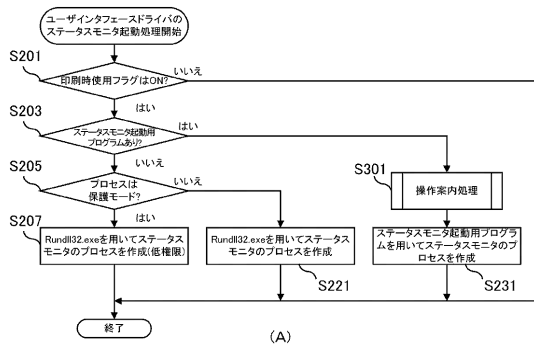
【図7】



【図8】



【図9】



【図 10】

