

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4810282号
(P4810282)

(45) 発行日 平成23年11月9日(2011.11.9)

(24) 登録日 平成23年8月26日(2011.8.26)

(51) Int.Cl.

F I

G 0 6 F 3/12 (2006.01)

G 0 6 F 3/12 C

G 0 6 F 13/10 (2006.01)

G 0 6 F 13/10 3 3 0 B

請求項の数 10 (全 34 頁)

(21) 出願番号 特願2006-100194 (P2006-100194)
 (22) 出願日 平成18年3月31日(2006.3.31)
 (65) 公開番号 特開2007-272763 (P2007-272763A)
 (43) 公開日 平成19年10月18日(2007.10.18)
 審査請求日 平成21年3月24日(2009.3.24)

(73) 特許権者 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 100145827
 弁理士 水垣 親房
 (72) 発明者 山道 雅樹
 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
 審査官 横山 佳弘

最終頁に続く

(54) 【発明の名称】 モジュールインストール方法、モジュールインストール装置、及びプログラム

(57) 【特許請求の範囲】

【請求項 1】

情報処理装置にインストール済みのプログラムに対してモジュールの追加が指示されると、該インストール済みのプログラムを構成する1又は複数のモジュールを検出する検出ステップと、

該検出された前記1又は複数のモジュールと前記追加が指示されたモジュールとを動作させるべき順番を判定する判定ステップと、

該判定された動作の順番で前記複数のモジュールが処理を実行するように、前記追加が指示されたモジュールを前記情報処理装置にインストールするモジュールインストールステップと、を有し、

前記インストール済みのプログラムはプリンタドライバプログラムであって、

改ざん検知モジュールと暗号化モジュールとの追加が指示された際に、

前記判定ステップでは、前記動作の順番として、暗号化モジュールより後に改ざん検知モジュールが動作されるべきと判定され、

前記モジュールインストールステップでは、前記暗号化モジュールがデータが処理を実行した後、該処理されたデータに対して前記改ざん検知モジュールが改ざん検知処理を実行するようにインストール順序を変更して、前記暗号化モジュールと前記改ざん検知モジュールをインストールされることを特徴とするモジュールインストール方法。

【請求項 2】

前記モジュールインストールステップでは、前記追加が指示されたモジュールが前記情

報処理装置の記憶領域に記憶され、該追加モジュールが前記動作の順番で処理を実行するように、前記情報処理装置の記憶領域に記憶される前記インストール済みのプログラムの構成情報への該追加モジュールの登録が制御されることを特徴とする請求項1に記載のモジュールインストール方法。

【請求項3】

前記モジュールインストールステップでは、前記検出ステップで検出された前記1又は複数のモジュールが前記動作の順番で処理を実行するように、前記インストール済みのプログラムの構成情報の前記1又は複数のモジュールの登録が変更制御されることを特徴とする請求項2に記載のモジュールインストール方法。

【請求項4】

前記判定ステップにおいて、前記検出ステップで検出された前記1つまたは複数のモジュールと前記追加が指示されたモジュールとを動作させるべき順番が存在しないと判定された場合、前記動作の順番が存在しない旨を通知する通知ステップを有することを特徴とする請求項1乃至3のいずれか1項に記載のモジュールインストール方法。

【請求項5】

前記判定ステップにおいて、前記検出ステップで検出された前記1つまたは複数のモジュールと前記追加が指示されたモジュールとを動作させるべき順番が存在しないと判定された場合、前記追加が指示されたモジュールのインストール処理を中止するインストール中止ステップを有することを特徴とする請求項4に記載のモジュールインストール方法。

【請求項6】

前記モジュールインストールステップにおいて、前記動作の順番で前記検出ステップで検出された前記1又は複数のモジュールと前記追加が指示されたモジュールとが処理を実行するようにモジュールのインストールが完了された際に、該インストールの完了を表示するインストール結果表示ステップを有することを特徴とする請求項1乃至5のいずれか1項に記載のモジュールインストール方法。

【請求項7】

前記モジュールインストールステップでインストールされた複数のモジュールを、前記プログラムの構成情報に基づいて、前記動作の順番で呼び出すことを特徴とする請求項2又は3に記載のモジュールインストール方法。

【請求項8】

追加指示されるモジュールの順序を変更してもよいかどうかを表示する表示ステップと、
前記表示ステップによる表示後、前記モジュールインストールステップでは、前記暗号化モジュールがデータが処理を実行した後、該処理されたデータに対して前記改ざん検知モジュールが改ざん検知処理を実行するように、前記暗号化モジュールと前記改ざん検知モジュールをインストールされることを特徴とする請求項1に記載のモジュールインストール方法。

【請求項9】

情報処理装置にインストール済みのプログラムに対してモジュールの追加が指示されると、該インストール済みのプログラムを構成する1又は複数のモジュールを検出する検出手段と、

該検出された前記1又は複数のモジュールと前記追加が指示されたモジュールとを動作させるべき順番を判定する判定手段と、

該判定された動作の順番で前記複数のモジュールが処理を実行するように、前記追加が指示されたモジュールを前記情報処理装置にインストールするモジュールインストール手段と、を有し、

前記インストール済みのプログラムはプリンタドライバプログラムであって、改ざん検知モジュールと暗号化モジュールとの追加が指示された際に、

前記判定手段が、前記動作の順番として、暗号化モジュールより後に改ざん検知モジュールが動作されるべきと判定し、

10

20

30

40

50

前記モジュールインストール手段が、前記暗号化モジュールがデータが処理を実行した後、該処理されたデータに対して前記改ざん検知モジュールが改ざん検知処理を実行するようにインストール順序を変更して、前記暗号化モジュールと前記改ざん検知モジュールをインストールすることを特徴とするモジュールインストール装置。

【請求項 10】

請求項 1 乃至 8 のいずれか 1 項に記載された工程をコンピュータに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】

10

本発明は、モジュールインストール方法、モジュールインストール装置、及びプログラムに関するものである。

【背景技術】

【0002】

近年、印刷結果の付加価値を高めるために、例えばコピーガード機能、フォームオーバーレイ、電子透かし、ジョブ追跡機能等の高度な新規付加機能をプリンタドライバに追加する動きがある。多くの場合、これらの付加機能は、通常印刷以上の付加価値を有するために、基本プリンタドライバセットに同梱せずに有償オプションとして提供されることが想定される。

【0003】

20

また、複数のプリンタドライバのバージョンや印刷装置のモデルとの依存性を極力抑えるために、これらの機能は共通拡張インタフェースをもつ追加のアドイン（プラグイン）モジュールとして提供される。

【0004】

この共通拡張インタフェースに対応していれば、過去にリリースした基本プリンタドライバセットとの組み合わせでも、アドインモジュールを提供しさえすれば、新規付加機能を実現可能という利点が存在する。言い換えると、新規機能をアドインで提供することにより、印刷装置の発売当初には想定されていない、印刷装置が認識できない機能も、付加的に後からプリンタドライバに追加することが可能である。

【0005】

30

特許文献 1 においては、前述のような共通拡張インタフェースに対応したプリンタドライバに対してプログラムファイルやデータファイルなどの機能拡張モジュールを追加、削除するための詳細な方法が提示されている。

【0006】

また、特許文献 2 には、ファイル検索プログラム 7 はレジストリファイル 12 に登録されているプリンタドライバを検索し、チェックプログラム 6 はレジストリファイル 12 に第 2 のプリンタ 11 のドライバと同一ラインナップのプリンタドライバが既に登録されているか否かを確認する。そして、新たに登録する第 2 のプリンタ 11 のドライバのリソースと、既に登録されている第 1 のプリンタのドライバのリソースとの競合の有無を判断する。すなわち、第 2 のプリンタ 11 に割り当てられる I/O ポートアドレスと、第 1 のプリンタに割り当てられている I/O ポートアドレスとが競合した場合はステップ 106 へ移行し、競合しなかった場合はステップ 105 へ移行する。そして、ファイル書き込みプログラム 5 によって第 2 のプリンタ 11 のドライバをレジストリファイル 12 に登録し、このプリンタドライバをコンピュータのハードディスク等の補助記憶装置に書き込むことによりインストールを実施する。また、I/O ポートアドレスが競合した場合はステップ 106 において、インストール順序が正しくないことを使用者に知らせるため、アラーム表示プログラム 9 によってモニタ 13 上に所定の警告を表示させる。そして、ドライバを消去し、インストール順序を変更するという技術が記載されている。

40

【特許文献 1】特開 2005 - 208895 号公報

【特許文献 2】特開平 10 - 124435 号公報

50

【発明の開示】

【発明が解決しようとする課題】

【0007】

しかしながら従来の技術においてはモジュールを追加するときに既にインストールされているモジュールとの動作順番について考慮されていない。そのため、既存のプリンタドライバを構成するモジュールに対して追加で機能拡張モジュールを追加した際に、動作をしない場合や、正常な動作をしない場合があるという問題があった。

【0008】

さらに、特許文献2は、一度インストールしてみて、ポートの設定が競合する場合に、インストール順序を入れ換えるものである。

10

【0009】

なお、上記の問題は、プリンタドライバに限られるものではなく、モジュールを追加可能なプログラム全体に起こり得るものであった。

【0010】

本発明は、上記の問題点を解決するためになされたものである。本発明の目的は、プリンタドライバ等のプログラムにモジュールを追加する際に、該追加するモジュールとプログラムを構成する既存のモジュールとの動作順番を正しく制御し、印刷処理の実行時に、プログラムを構成する各モジュールが適切に動作するようにする仕組みを提供することである。

【0011】

20

さらに本発明の別の目的として、本願発明は、前記の動作順番に応じて適切なインストール手順を事前に決定してインストールする仕組みを提供するという別の側面がある。

【課題を解決するための手段】

【0012】

情報処理装置にインストール済みのプログラムに対してモジュールの追加が指示されると、該インストール済みのプログラムを構成する1又は複数のモジュールを検出する検出ステップと、該検出された前記1又は複数のモジュールと前記追加が指示されたモジュールとを動作させるべき順番を判定する判定ステップと、該判定された動作の順番で前記複数のモジュールが処理を実行するように、前記追加が指示されたモジュールを前記情報処理装置にインストールするモジュールインストールステップと、を有し、前記インストール済みのプログラムはプリンタドライバプログラムであって、改ざん検知モジュールと暗号化モジュールとの追加が指示された際に、前記判定ステップでは、前記動作の順番として、暗号化モジュールより後に改ざん検知モジュールが動作されるべきと判定され、前記モジュールインストールステップでは、前記暗号化モジュールがデータが処理を実行した後、該処理されたデータに対して前記改ざん検知モジュールが改ざん検知処理を実行するようにインストール順序を変更して、前記暗号化モジュールと前記改ざん検知モジュールをインストールされることを特徴とする。

30

【発明の効果】

【0016】

本発明によれば、プリンタドライバ等のプログラムにモジュールを追加する際に、該追加するモジュールとプログラムを構成する既存のモジュールとの動作順番を正しく設定制御することが可能になる。

40

【0017】

従って、プリンタドライバ等のプログラムを構成する各モジュールを適切な順番で適用し、適切に動作させることができる等の効果を奏する。

【発明を実施するための最良の形態】

【0018】

〔第1実施形態〕

以下、本発明を適用するのに好適である実施形態について説明を行う。

【0019】

50

図 1 は、本発明を適用可能な情報処理装置の一例を示すブロック図である。

【 0 0 2 0 】

図 1 において、100 は情報処理装置としてのコンピュータ（例えば、パーソナルコンピュータ）である。本発明の印刷システムでは、コンピュータ100 とプリンタ30 はセントロニクス等のインタフェースあるいはイーサネット（登録商標）等のネットワークによって接続されている。なお、図 1 においては、コンピュータ100 とプリンタ30 がネットワーク20 によって接続されている場合を示しているが、図示しないインタフェース（例えば、USB，セントロニクスインタフェース等）を介して接続される構成であってもよい。

【 0 0 2 1 】

1 は CPU で、ROM 3 のプログラム用 ROM，外部メモリ 12 に記憶されたプログラム（BIOS，OS，アプリケーションプログラム，後述する、アドインインストーラを含む各種インストーラ等）に基づいてシステムバス 4 に接続された各デバイスを制御し、装置全体を統括制御する。

【 0 0 2 2 】

また、ROM 3 のフォント用 ROM にはフォントデータ等を記憶し、ROM 3 のデータ用 ROM には各種データを記憶する。2 は RAM で、CPU 1 の主メモリ，ワークエリア等として機能する。

【 0 0 2 3 】

5 はキーボードコントローラ（KBC）で、ポインティングデバイス9（例えばマウス）やキーボード（KB）10 等からの入力を制御する。6 はディスプレイコントローラ（DC）で、ディスプレイ11 の表示を制御する。なお、ディスプレイ11 は、CRT であってもLCD 等のその他の形式の表示装置であってもよい。

【 0 0 2 4 】

7 はメモリコントローラ（MC）で、外部メモリ 12（ハードディスク（HD），図示しないフレキシブルディスク（FD），CD-ROM 等）とのアクセスを制御する。8 はネットワークインタフェース（NWIF）で、ネットワーク20 を介してプリンタ30 との通信を制御する。

【 0 0 2 5 】

< システムの構成 >

以下、図 1 に示したコンピュータ100 を例にして、プリンタドライバおよびアドインモジュール、および本発明を実現するために必要なその他の構成について説明する。

【 0 0 2 6 】

図 2 は、本発明の第 1 実施形態に係る印刷システムの構成例を示すブロック図であり、図 1 と同一のものには同一の符号を付してある。

【 0 0 2 7 】

図 2 において、101 はプリンタドライバである。プリンタドライバ101 は、コンピュータ100 からプリンタ30 へ印刷を行うためのプログラムで、コンピュータ100 のアプリケーション108 等からの印刷命令をOS 内の描画エンジン109 を介して入力する。そして、プリンタドライバ101 は、入力した印刷命令をプリンタ30 が認識可能な制御コマンドに変換し、印刷サービス105 を介してプリンタに転送する。

【 0 0 2 8 】

なお、プリンタドライバ101 は、プリンタドライバインストーラ107 またはOS が提供する図示しないプリンタドライバインストール方法によってインストールされ、利用可能となる。

【 0 0 2 9 】

アドインモジュール102 は、例えばDLL（Dynamic Link Library）などのプログラムファイルを含むファイル群である。アドインモジュール102 は、アドインインストーラ106 にもしくはプリンタドライバ101 によってインストールされることによって、プリンタドライバの共通インタフェースを用いて通信可能となり、機能拡張を実現する。

10

20

30

40

50

即ち、アドインインストーラ 106 は、プリンタドライバ 101 がインストール済みの状態で実行可能なプログラムであり、アドインモジュール 102 はプリンタドライバの一部として動作する構成となっている。

【0030】

次に、アドインモジュール情報ファイル 103 について説明する。

【0031】

アドインモジュール情報ファイル 103 は、アドインモジュールを構成するファイル群の 1 ファイルであり、図 3 に例示するような情報を有する。なお、アドインモジュール情報ファイル 103 のファイル形式は可読性のあるテキスト形式や XML 形式であってもかまわないし、バイナリファイルのような非可読形式であってもよい。ここで、図 3 を参照して、アドインモジュール情報ファイル 103 のデータ構成について説明する。

10

【0032】

図 3 は、図 2 に示したアドインモジュール情報ファイル 103 のデータ構成を示すデータ構成図である。

【0033】

図 3 において、201 はアドインモジュール構成ファイル情報であり、アドインモジュール 102 のファイル構成が記述されている領域である。202 はアドインモジュール 102 のバージョン情報が記述されている領域である。203 はアドインモジュール 102 のリリース日が記述されている領域である。204 はアドインモジュール 102 の製品名情報が記述されている領域である。205 はアドインモジュール 102 の詳細情報が記述されている領域である。

20

【0034】

206 は PriorityNumber (プライオリティナンバ) であり、動作の順番を決定する一意の番号が記述されている領域である。207 はアドインモジュール 102 の Copyright 情報が記述されている領域である。

【0035】

以下、図 2 の説明に戻る。

【0036】

プリンタドライバファイル情報 104 は、OS が印刷サービス 105 を用いてプリンタドライバを利用するための一管理情報 (構成情報) であり、外部メモリ 12 の所定領域 (OS により管理される所定の記憶領域) に登録されている。このプリンタドライバファイル情報 104 は、例えば Windows (登録商標) ではレジストリ中の「DependentFiles」が該当する。なお、プリンタドライバが複数インストールされている場合には、各プリンタドライバ毎に、プリンタドライバファイル情報 104 が外部メモリ 12 に登録される。

30

【0037】

図 2 においては、アドインモジュール 102 がインストール済であり、プリンタドライバファイル情報 104 に、アドインモジュールファイル情報 111 が登録されている。このアドインモジュールファイル情報 111 には、アドインモジュール情報ファイルを特定する情報 (例えば、アドインモジュール情報ファイル名称) 112 が含まれる。

【0038】

40

また、図示しないがアドインモジュール 102 とは異なるアドインモジュールが別途インストールされていても、プリンタドライバ 101 は動作可能である。なお、プリンタドライバ 101 に複数のアドインモジュールがインストールされている場合には、各アドインモジュール毎にアドインモジュールファイル情報 111 (アドインモジュール情報ファイルを特定する情報 112 を含む) が登録される。

【0039】

印刷サービス 105 は、オペレーティングシステム (OS) 上で印刷やプリンタドライバ 101 のインストールを制御するサービスプログラムである。

【0040】

アドインインストーラ 106 は、本発明の目的を達成するための中心となるものであり

50

、インストール済のプリンタドライバ１０１にアドインモジュールを追加インストールするプログラムである。なお、追加インストールするアドインモジュールファイルは、アドインインストーラ１０６内又は外部メモリ１２内の他の記憶領域に格納されているものとする。

【００４１】

なお、アドインインストーラ１０６は、主に以下の（１）～（５）の機能を有する。

【００４２】

（１）１つまたは複数のプリンタドライバを構成する１つまたは複数のモジュールを検出する構成モジュール検出機能

（２）追加するアドインモジュールと、プリンタドライバ１０１を構成する１つまたは複数のモジュール（インストール済みのアドインモジュール１０２）との適切な動作の順番（該複数のアドインモジュールを動作させるべき順番）を判定するモジュール適正動作順番判定機能

（３）アドインモジュールファイルを追加するファイル追加機能

（４）上述のモジュール適正動作順番判定機能で判定された動作順番に従って追加された追加ファイルをプリンタドライバファイル情報１０４に登録するファイル登録機能

（５）印刷サービス１０５にモジュール構成の変更を認識させる認識機能

なお、アドインインストーラ１０６における処理の流れについては、後述する図４以降を用いて詳細に説明する。

【００４３】

プリンタドライバインストーラ１０７は、プリンタドライバ１０１をコンピュータ１００にインストールするプログラムである。

【００４４】

なお、アドインインストーラ１０６，プリンタドライバインストーラ１０７は、図１に示したＣＰＵ１が外部メモリ１２に格納されたプログラムをＲＡＭ２上で実行することにより実現されるものである。

【００４５】

<アドインインストーラ１０６における処理の流れ>

図４は、本発明における第１の制御処理手順の一例を示すフローチャートであり、アドインインストーラ１０６の処理の流れに対応する。即ち、このフローチャートの処理は、ＣＰＵ１が外部メモリ１２に格納されたアドインインストーラ１０６に対応するプログラムをＲＡＭ２上で実行することにより実現されるものである。なお、図中、Ｓ３０００～Ｓ３４００は各ステップを示す。

【００４６】

まず、ステップＳ３０００において、アドインインストーラ１０６は、アドインモジュールの追加対象プリンタドライバについて、インストール済みのアドインモジュールの検出を行う。なお、このアドインモジュールの追加対象プリンタドライバは、コンピュータ１００にインストール済みのプリンタドライバであり、ユーザによってアドインモジュールの追加対象として選択されたものが対応する。ここでは、図２のプリンタドライバ１０１とする。また、このステップＳ３０００の処理の詳細は図５において説明する。

【００４７】

次に、ステップＳ３１００において、アドインインストーラ１０６は、アドインモジュール順番情報を作成する。この処理は、ステップＳ３０００で検出されたインストール済みのアドインモジュール１０２と、アドインインストーラ１０６により追加するアドインモジュールを合わせた印刷処理において、適正な順番情報（前記複数のアドインモジュールを動作させるべき順番）を保持するアドインモジュール順番情報を作成する。なお、このアドインモジュール順番情報は、適正に動作可能な順番（前記複数のアドインモジュールを動作させるべき順番）が存在した場合にのみ作成される。

【００４８】

また、このアドインモジュール順番情報の保持形態としては、ファイル形式やＲＡＭ２

10

20

30

40

50

上に保持する等形態は問わないが、ここではRAM 2上に保持するものとして説明する。このアドインモジュール順番情報は、最小の情報構成として順番を示す番号とモジュール名称を保持している。なお、このステップS 3 1 0 0の処理の詳細は図6において説明する。

【0049】

次に、ステップS 3 2 0 0において、アドインインストーラ106は、アドインモジュール順番情報がRAM 2上に存在するか否かを判定する。そして、アドインモジュール順番情報がRAM 2上に存在しないと判定した場合には、アドインインストーラ106は、ステップS 3 4 0 0に処理を進める。

【0050】

ステップS 3 4 0 0では、アドインインストーラ106は、アドインモジュールインストール処理を中止し、本フローチャートの処理を終了する。

【0051】

一方、ステップS 3 2 0 0で、アドインモジュール順番情報がRAM 2上に存在すると判定した場合には、アドインインストーラ106は、ステップS 3 3 0 0に処理を進める。

【0052】

ステップS 3 3 0 0では、アドインインストーラ106は、アドインモジュールインストール処理を行い、本フローチャートの処理を終了する。なお、このステップS 3 3 0 0の処理の詳細は図7において説明する。

【0053】

図5は、本発明における第2の制御処理手順の一例を示すフローチャートであり、図4のステップS 3 0 0 0に示したアドインモジュール検出処理の詳細な流れに対応する。即ち、このフローチャートの処理は、CPU 1が外部メモリ12に格納されたアドインインストーラ106に対応するプログラムをRAM 2上で実行することにより実現されるものである。なお、図中、S 4 0 0 0～S 4 6 0 0は各ステップを示す。

【0054】

まず、ステップS 4 0 0 0において、アドインインストーラ106は、プリンタドライバファイル情報104を参照して、プリンタドライバ101についての構成情報を取得する。

【0055】

次に、ステップS 4 1 0 0において、アドインインストーラ106は、アドインモジュール情報ファイル103が存在するか否かを判定する。なお、この判定は、ステップS 4 0 0 0によって取得された構成情報内にアドインモジュール情報ファイル103に関するファイル情報（アドインモジュールファイル情報111）が1つでも含まれている場合にアドインモジュール情報ファイル103が存在すると判定する。一方、構成情報内にアドインモジュールファイル情報111が1つも含まれていない場合にはアドインモジュール情報ファイル103が存在しないと判定する。

【0056】

ステップS 4 1 0 0で、アドインモジュール情報ファイル103が存在しないと判定した場合には、アドインインストーラ106は、そのまま本フローチャートの処理を終了して、図4のステップS 3 1 0 0に処理を進める。

【0057】

一方、ステップS 4 1 0 0で、アドインモジュール情報ファイル103が存在すると判定した場合には、アドインインストーラ106は、ステップS 4 2 0 0に処理を進める。

【0058】

ステップS 4 2 0 0では、アドインインストーラ106は、ステップS 4 0 0 0で取得した構成情報内に登録された全てのアドインモジュール情報ファイル103の参照が終了したか否かを判定する。

【0059】

10

20

30

40

50

そして、ステップS 4 2 0 0で、まだ全てのアドインモジュール情報ファイル1 0 3の参照が終了していないと判定した場合には、アドインインストーラ1 0 6は、ステップS 4 3 0 0に処理を進める。

【0 0 6 0】

ステップS 4 3 0 0では、アドインインストーラ1 0 6は、ステップS 4 0 0 0で取得した構成情報内に登録されており且つ未だ参照していないアドインモジュール情報ファイル1 0 3を1つ選択し、該アドインモジュール情報ファイル1 0 3内の情報を参照する。

【0 0 6 1】

次に、ステップS 4 4 0 0において、アドインインストーラ1 0 6は、ステップS 4 3 0 0でアドインモジュール情報ファイル1 0 3内の情報が正常に参照されたか否かを判定する。

10

【0 0 6 2】

ステップS 4 4 0 0において、ステップS 4 3 0 0でアドインモジュール情報ファイル1 0 3内の情報が正常に参照されなかったと判定した場合には、アドインインストーラ1 0 6は、ステップS 4 5 0 0に処理を進める。

【0 0 6 3】

ステップS 4 5 0 0では、アドインインストーラ1 0 6は、このアドインモジュールは正常に動作していない、即ちアドインされていないとみなし、ステップS 4 6 0 0に処理を進める。

【0 0 6 4】

20

一方、ステップS 4 4 0 0において、ステップS 4 3 0 0でアドインモジュール情報ファイル1 0 3内の情報が正常に参照されたと判定した場合には、アドインインストーラ1 0 6は、ステップS 4 6 0 0に処理を進める。

【0 0 6 5】

ステップS 4 6 0 0では、アドインインストーラ1 0 6は、次のアドインモジュール情報ファイル1 0 3の参照へ処理を進め、ステップS 4 2 0 0に処理を戻す。

【0 0 6 6】

そして、ステップS 4 2 0 0で、構成情報内に登録された全てのアドインモジュール情報ファイル1 0 3の参照が終了したと判定した場合には、アドインインストーラ1 0 6は、本フローチャートの処理を終了して、図4のステップS 3 1 0 0に処理を進める。

30

【0 0 6 7】

以上が、図4のステップS 3 0 0 0に示したアドインモジュール検出処理の詳細動作となり、アドインを追加するプリンタドライバが複数選択されている場合には、選択されているプリンタドライバの数だけ同様の処理を繰り返す。

【0 0 6 8】

図6は、本発明における第3の制御処理手順の一例を示すフローチャートであり、図4のステップS 3 1 0 0に示したアドインモジュール順番情報作成処理の詳細な流れに対応する。即ち、このフローチャートの処理は、CPU 1が外部メモリ1 2に格納されたアドインインストーラ1 0 6に対応するプログラムをRAM 2上で実行することにより実現されるものである。なお、図中、S 5 0 0 0～S 5 8 0 0は各ステップを示す。

40

【0 0 6 9】

まず、ステップS 5 0 0 0において、アドインインストーラ1 0 6は、アドインモジュール順番情報の新規作成など初期化処理を行う。

【0 0 7 0】

ステップS 5 1 0 0において、アドインインストーラ1 0 6は、アドインの追加対象プリンタドライバに対し、図4のステップ3 0 0 0で検出された及びこれから追加する全てのアドインモジュールのアドインモジュール情報ファイル取得を終了したか否かを判定する。即ち、アドインモジュール順番情報の作成処理を完了したか否かを判定する。

【0 0 7 1】

ステップS 5 1 0 0で、図4のステップS 3 0 0 0で検出された及びこれから追加する

50

アドインモジュールの全てのアドインモジュール情報ファイル取得を未だ終了していないと判定した場合には、アドインインストーラ 106 は、ステップ S 5 2 0 0 に処理を進める。

【 0 0 7 2 】

ステップ S 5 2 0 0 では、アドインインストーラ 106 は、図 4 のステップ S 3 0 0 0 で検出された及びこれから追加するアドインモジュール内のまだ取得していないアドインモジュール情報ファイル 103 を 1 つ取得する。以下、このアドインモジュール情報ファイル 103 に対応するアドインモジュールを現在注目しているアドインモジュールと呼ぶ。

【 0 0 7 3 】

ステップ S 5 3 0 0 では、アドインインストーラ 106 は、ステップ S 5 2 0 0 で取得したアドインモジュール情報ファイル 103 から PriorityNumber 206 を取得する。なお、PriorityNumber とは例えば 1 以上の整数などであり、数値の大小によって動作する順番を定義するものである。この PriorityNumber によって新規にモジュールを追加した場合にも対応が可能となる。

【 0 0 7 4 】

ステップ S 5 4 0 0 では、アドインインストーラ 106 は、本ステップまでにアドインモジュール順番情報内に書き込み済みのアドインモジュールの PriorityNumber と現在注目しているアドインモジュールの PriorityNumber とを比較する。そして、アドインインストーラ 106 は、PriorityNumber の大きい順となるように順番（アドインモジュール順番情報への書き込み位置）を決定する。

【 0 0 7 5 】

次に、ステップ S 5 5 0 0 において、アドインインストーラ 106 は、ステップ S 5 4 0 0 で決定された書込位置に書き込みが可能か否かを判定する。なお、書き込みが不可能な場合としては、PriorityNumber が同一のものや共存が不可能なモジュールの競合等が考えられる。

【 0 0 7 6 】

ステップ S 5 5 0 0 で、書き込みが不可能であると判定した場合、アドインインストーラ 106 は、ステップ S 5 8 0 0 に処理を進める。

【 0 0 7 7 】

ステップ S 5 8 0 0 では、アドインインストーラ 106 は、適正な順番（予め追加されているアドインモジュールとこれから追加するアドインモジュールとを動作させるべき順番）が存在しないと判断して、現在 RAM 2 に保持しているアドインモジュール順番情報を削除する。そして、アドインインストーラ 106 は、図示しないステップにおいて、PriorityNumber が同一であったり、共存が不可能なモジュールの競合等が発生しモジュールを適用する適正な順番（動作させるべき順番）が存在しない旨の通知をディスプレイ 11 に表示する。そして、本フローチャートの処理を終了し、図 4 のステップ S 3 2 0 0 に処理を進める。

【 0 0 7 8 】

一方、ステップ S 5 5 0 0 で、書き込みが可能であると判定した場合には、アドインインストーラ 106 は、ステップ S 5 6 0 0 に処理を進める。

【 0 0 7 9 】

ステップ S 5 6 0 0 では、アドインインストーラ 106 は、現在注目しているアドインモジュールのアドインモジュール名称を、ステップ S 5 4 0 0 で決定したアドインモジュール順番情報内の書込位置に書込処理する。本実施形態では、順番を示す番号とアドインモジュール名称を書込処理する。

【 0 0 8 0 】

次に、ステップ S 5 7 0 0 において、アドインインストーラ 106 は、次のアドインモジュール情報ファイルへ処理を進め、ステップ S 5 1 0 0 に処理を戻す。

【 0 0 8 1 】

10

20

30

40

50

そして、ステップ S 5 1 0 0 で、図 4 のステップ 3 0 0 0 で検出された及びこれから追加するアドインモジュールの全てのアドインモジュール情報ファイル取得を終了したと判定した場合には、本フローチャートの処理を終了し、図 4 のステップ S 3 2 0 0 に処理を進める。

【 0 0 8 2 】

以上が図 4 のステップ S 3 1 0 0 に示したアドインモジュール順番情報作成処理の詳細動作となり、アドインを追加するプリンタドライバが複数選択されている場合には選択されているプリンタドライバの数だけ同様の処理を繰り返す。

【 0 0 8 3 】

なお、図 6 では各アドインモジュールが PriorityNumber を保持することでアドインモジュール順番情報を作成したが、アドインモジュール順番情報をファイル情報として外部から提供するようにしてもよい。例えば、アドインモジュール順番情報をファイル情報として Web 上で公開し、該 Web 上でから取得して提供するように構成してもよい。

【 0 0 8 4 】

また、アドインモジュール順番情報は、XML のようなテキスト形式でも非可読のバイナリ形式でもよくデータ形式は問わない。

【 0 0 8 5 】

図 7 は、本発明における第 4 の制御処理手順の一例を示すフローチャートであり、図 4 のステップ S 3 3 0 0 に示したアドインモジュールインストール処理の詳細な流れに対応する。即ち、このフローチャートの処理は、CPU 1 が外部メモリ 1 2 に格納されたアドインインストーラ 1 0 6 に対応するプログラムを RAM 2 上で実行することにより実現されるものである。なお、図中、S 6 0 0 0 ~ 6 5 0 0 は各ステップを示す。

【 0 0 8 6 】

まず、ステップ S 6 0 0 0 において、アドインインストーラ 1 0 6 は、プリンタドライバファイル情報 1 0 4 内に登録されたアドインモジュールファイル情報 1 1 1 を初期化する。

【 0 0 8 7 】

次に、ステップ S 6 1 0 0 において、アドインインストーラ 1 0 6 は、ステップ S 3 1 0 0 で作成されたアドインモジュール順番情報を取得する。

【 0 0 8 8 】

次に、ステップ S 6 2 0 0 において、アドインインストーラ 1 0 6 は、ステップ S 6 1 0 0 で取得したアドインモジュール順番情報に記載されているアドインモジュールのインストール処理が全て完了したか否かを判定する。

【 0 0 8 9 】

ステップ S 6 2 0 0 で、ステップ S 6 1 0 0 で取得したアドインモジュール順番情報に記載されているアドインモジュールのインストール処理がまだ全て完了していないと判定した場合には、アドインインストーラ 1 0 6 は、ステップ S 6 3 0 0 に処理を進める。

【 0 0 9 0 】

ステップ S 6 3 0 0 では、アドインインストーラ 1 0 6 は、アドインモジュール順番情報に従い動作の順番が早いものから 1 つアドインモジュール（アドインモジュール構成ファイル）を取得する。

【 0 0 9 1 】

次に、ステップ S 6 4 0 0 において、アドインインストーラ 1 0 6 は、ステップ S 6 3 0 0 で取得したアドインモジュール構成ファイルを外部メモリ 1 2 内の所定領域（OS の記憶領域）に追加する。なお、このステップは、今回新たに追加インストールするアドインモジュールに対してのみ行い、予め追加済みであったアドインモジュールに関してはスキップするものとする。

【 0 0 9 2 】

次に、ステップ S 6 5 0 0 において、アドインインストーラ 1 0 6 は、プリンタドライバ構成情報（プリンタドライバファイル情報 1 0 4 ）内にアドインモジュールファイル情

10

20

30

40

50

報 1 1 1 (アドインモジュール情報ファイルを特定する情報 1 1 2 を含む) を登録する。

【 0 0 9 3 】

次に、ステップ S 6 6 0 0 において、アドインインストーラ 1 0 6 は、アドインモジュール順番情報に次の記載されているアドインモジュールに処理を移し、ステップ S 6 2 0 0 へ処理を戻す。

【 0 0 9 4 】

そして、ステップ S 6 2 0 0 で、アドインモジュール順番情報に記載されているアドインモジュールのインストール処理が全て完了したと判定した場合には、本フローチャートの処理を終了する。

【 0 0 9 5 】

以上のようにインストール処理することで、予め追加されているアドインモジュールと新たに追加したアドインモジュールとが、図 4 のステップ S 3 1 0 0 で判定された動作の順番で処理を実行するように、新たなアドインモジュールを登録することができる。

【 0 0 9 6 】

< アドインインストーラ UI について >

図 8 は、本発明における第 5 の制御処理手順の一例を示すフローチャートであり、アドインインストーラ 1 0 6 によるアドインインストール時の UI 表示の流れに対応する。即ち、このフローチャートの処理は、C P U 1 が外部メモリ 1 2 に格納されたアドインインストーラ 1 0 6 に対応するプログラムを R A M 2 上で実行することにより実現されるものである。なお、図中、S 7 0 0 0 ~ S 7 4 0 0 は各ステップを示す。

【 0 0 9 7 】

また、図 9 ~ 図 1 3 は、アドインインストーラ 1 0 6 によるアドインインストール時の UI 表示処理時の表示画面の一例を示す模式図である。

【 0 0 9 8 】

まず、アドインセットアップウィザードが開始されると、アドインインストーラ 1 0 6 は、ステップ S 7 0 0 0 において、図 9 に示すようなスタート画面をディスプレイ 1 1 に表示する。

【 0 0 9 9 】

図 9 に示すスタート画面において、「次へ」ボタン 8 0 1 が指示されると (P D 9 でクリックされると) 、アドインインストーラ 1 0 6 は、図 8 のステップ S 7 1 0 0 に処理を進める。

【 0 1 0 0 】

なお、図 8 のフローチャートには示していないが、図 9 に示すスタート画面において、「キャンセル」ボタン 8 0 2 が指示されると (P D 9 でクリックされると) 、アドインインストーラ 1 0 6 は、図 8 のフローチャートの処理を中止する。

【 0 1 0 1 】

次に図 8 のステップ S 7 1 0 0 では、アドインインストーラ 1 0 6 は、図 1 0 (a) のようなドライバ機種選択画面をディスプレイ 1 1 に表示する。

【 0 1 0 2 】

図 1 0 (a) に示すドライバ機種選択画面では、9 0 1 に示すように、現在アドインモジュールのインストール対象となっているプリンタドライバの一覧が表示されている。

【 0 1 0 3 】

このドライバ機種選択画面で、「追加と削除」ボタン 9 0 2 が指示されると (P D 9 でクリックされると) 、アドインインストーラ 1 0 6 は、図 1 0 (b) のような画面をディスプレイ 1 1 に表示する。

【 0 1 0 4 】

図 1 0 (b) に示す画面では、9 0 6 に示すように、アドインモジュールがインストール可能なプリンタドライバの一覧が表示される。

【 0 1 0 5 】

図 1 0 (b) に示す画面で、「インストールするドライバ一覧へ追加」ボタン 9 0 7 が

10

20

30

40

50

指示されると（P D 9でクリックされると）、アドインインストーラ106は、906において選択されているプリンタドライバをアドインモジュールのインストール対象となっているプリンタドライバの一覧を表す910に追加処理する。

【0106】

また、図10（b）に示す画面で、「削除」ボタン908が指示されると（P D 9でクリックされると）、アドインインストーラ106は、910で選択されているプリンタドライバをインストール対象から削除する。

【0107】

また、図10（b）に示す画面で、「OK」ボタン909が指示されると（P D 9でクリックされると）、アドインインストーラ106は、図10（b）を閉じ、910の内容を901に反映させて図10（a）の画面に戻す。

10

【0108】

図10（b）に示す画面で、「キャンセル」ボタン910が指示されると（P D 9でクリックされると）、アドインインストーラ106は、図10（b）を閉じ、図10（a）の画面に戻す。

【0109】

図10（a）の画面で、「戻る」ボタン903が指示されると（P D 9でクリックされると）、アドインインストーラ106は、図8のステップS7000に処理を戻す。

【0110】

図10（a）の画面で、「開始」ボタン904が指示されると（P D 9でクリックされると）、アドインインストーラ106は、図8のステップS7200に処理を進める。

20

【0111】

なお、図8のフローチャートには示していないが、図10（a）の画面で、「キャンセル」ボタン905が指示されると（P D 9でクリックされると）、アドインインストーラ106は、図8のフローチャートの処理を中止する。

【0112】

図8のステップS7200において、アドインインストーラ106は、図11のようなインストール警告表示画面をディスプレイ11に表示する。

【0113】

図11の画面で、「いいえ」ボタン1005が指示されると（P D 9でクリックされると）、図8のフローチャートには示していないが、アドインインストーラ106は、図8のフローチャートの処理を中止する。

30

【0114】

また、図11の画面で、「はい」ボタン1001が指示されると（P D 9でクリックされると）、アドインインストーラ106は、ステップS7300に処理を進める。

【0115】

次に、図8のステップS7300において、アドインインストーラ106は、図12のようなステータス画面をディスプレイ11に表示し、図4のフローチャートに示した処理（インストール実処理）を実行する。

【0116】

40

そして図4に示したインストール実処理の完了後、アドインインストーラ106は、図8のステップS7400に処理を進める。

【0117】

図8のステップS7400では、アドインインストーラ106は、図13のようなインストール結果画面をディスプレイ11に表示することで、アドインインストールが成功したか否かを表示する。図13に示した例では、図4に示した処理でアドインのインストールが成功（完了）した場合に対応する。

【0118】

なお、アドインインストーラ106は、図4に示した処理で、動作において適切なインストール順番が存在しない場合等は、このインストール結果画面で、アドインのインスト

50

ールが失敗した旨を表示する。

【0119】

図13のインストール結果画面において、PD9で1201が選択されて「終了」ボタン1203が指示されると(PD9でクリックされると)、アドインインストーラ106は、図13の画面を閉じ、コンピュータ100の再起動を行う。これにより、アドインインストーラ106によって行われたプリンタドライバ101の変更をコンピュータ100に反映させる。

【0120】

また、図13のインストール結果画面において、PD9で1202が選択されて「終了」ボタン1203が指示されると(PD9でクリックされると)、アドインインストーラ106は、図13を閉じる。

10

【0121】

<アドインモジュール102の呼び出し処理の流れ>

本実施形態によりインストールされたアドインモジュールを備えたプリンタドライバの印刷処理の詳細を図1、図2、および図14を用いて説明する。

【0122】

アプリケーション108、グラフィックエンジン109、プリンタドライバ101、および印刷サービス105は、外部メモリ12に保存されたファイルとして存在する。そして、これらは、CPU1により実行されるOSやそのモジュールを利用するモジュールによってRAM2にロードされて実行されるプログラムモジュールである。

20

【0123】

また、アプリケーション108およびプリンタドライバ101は、外部メモリ12のFDやCD-ROM、あるいは不図示のネットワーク20を経由して外部メモリ12のHDに追加することが可能となっている。

【0124】

外部メモリ12に保存されているアプリケーション108は、CPU1によりRAM2にロードされて実行される。しかし、このアプリケーション108からプリンタ30に対して印刷を行う際には、同様にRAM2にロードされ実行可能となっているグラフィックエンジン109を利用して出力(描画)を行う。

【0125】

30

グラフィックエンジン109は、プリンタなどの印刷装置毎に用意されたプリンタドライバ101を外部メモリ12からRAM2にロードし、アプリケーション108の出力をプリンタドライバ101に設定する。また、グラフィックエンジン202は、アプリケーション108から受け取るGDI(Graphic Device Interface)関数をDDI(Device Driver Interface)関数に変換して、プリンタドライバ101へ出力する。

【0126】

プリンタドライバ101は、グラフィックエンジン109から受け取ったDDI関数に基づいて、プリンタが認識可能な制御コマンド、例えばPDL(Page Description Language)に変換する。該変換されたプリンタ制御コマンドは、OSによってRAM2にロードされた印刷サービス105を経てプリンタ30へ印刷データとして出力される仕組みとなっている。

40

【0127】

本実施形態の印刷システムは、プリンタドライバ101内にアドインモジュール処理部110を有する。

【0128】

プリンタドライバ101は、アドインモジュール102の実行に際し、アドインモジュール処理部110を通してアドインモジュール102を呼び出し、処理を行う。アドインモジュール102の呼び出し処理の詳細に関しては図14を用いて示す。

【0129】

図14は、本発明における第6の制御処理手順の一例を示すフローチャートであり、プ

50

リントドライバ101によるアドインモジュールの呼び出し処理の流れに対応する。即ち、このフローチャートの処理は、CPU1が外部メモリ12に格納されたアドインインストーラ106に対応するプログラムをRAM2上で実行することにより実現されるものである。なお、図中、S13000～S13700は各ステップを示す。

【0130】

まず、ステップS13000において、プリントドライバ101は、プリントドライバファイル情報104を参照して、プリントドライバ101についての構成情報を取得する。

【0131】

次に、ステップS13100において、プリントドライバ101は、アドインモジュール情報ファイル103が存在するか否かを判定する。なお、この判定は、ステップS13000によって取得された構成情報内にアドインモジュール情報ファイル103に関するファイル情報（アドインモジュールファイル情報111）が1つでも含まれている場合にアドインモジュール情報ファイル103が存在すると判定する。一方、構成情報内にアドインモジュールファイル情報111が1つも含まれていない場合にはアドインモジュール情報ファイル103が存在しないと判定する。なお、本実施形態において図7の処理が実行されている場合には、アドインモジュールファイル情報111は、プリントドライバファイル情報104において動作順番の早いものから順番に登録されている。

10

【0132】

ステップS13100で、アドインモジュール情報ファイル103が存在しないと判定した場合には、プリントドライバ101は、のまま本フローチャートの処理を終了する。

20

【0133】

一方、ステップS13100で、アドインモジュール情報ファイル103が存在すると判定した場合には、プリントドライバ101は、ステップS13200に処理を進める。

【0134】

ステップS13200では、プリントドライバ101は、ステップS13000で取得した構成情報内に登録された全てのアドインモジュール情報ファイルの参照が終了したか否かを判定する。

【0135】

そして、ステップS13200で、まだ全てのアドインモジュール情報ファイル103の参照が終了していないと判定した場合には、プリントドライバ101は、ステップS13300に処理を進める。

30

【0136】

ステップS13300では、プリントドライバ101は、ステップS13000で取得した構成情報内に登録されており且つ未だ参照していないアドインモジュール情報ファイル103を1つ選択し、該アドインモジュール情報ファイル103内の情報を参照する。

【0137】

次に、ステップS13400において、プリントドライバ101は、ステップS13300でアドインモジュール情報ファイル内の情報が正常に参照されたか否かを判定する。

40

【0138】

ステップS13400において、ステップS13300でアドインモジュール情報ファイル内の情報が正常に参照されなかったと判定した場合には、プリントドライバ101は、ステップS13500に処理を進める。

【0139】

そして、ステップS13500では、プリントドライバ101は、このアドインモジュールは正常に動作していない、即ちアドインされていないとみなし、ステップS13700に処理を進める。

【0140】

一方、ステップS13400において、ステップS13300でアドインモジュール情報ファイル内の情報が正常に参照されたと判定した場合には、プリントドライバ101は、

50

ステップ S 1 3 6 0 0 に処理を進める。

【 0 1 4 1 】

そして、ステップ S 1 3 6 0 0 では、プリンタドライバ 1 0 1 は、ステップ S 1 3 3 0 0 で参照したアドインモジュールを呼び出す。

【 0 1 4 2 】

なお、プリンタドライバ 1 0 1 は、ステップ S 1 3 6 0 0 において呼び出された対象アドインモジュールの動作にとって、不都合なその他のアドインモジュールがないかの確認を行う。そして、不都合がある場合には、警告画面をディスプレイ 1 1 に表示することでさらに順番制御の精度を高めることも可能である。例えば、機能制限を行うことを目的とするモジュールが対象モジュールとして呼び出された場合、対象モジュールは、後段モジュールからの書き換えによる機能制限を無効にするのを防止するために改ざん検知としてハッシュ値をデータに追加する。なお、ハッシュ値とは、ある規則に従って算出されたデータを代表する値である。例えば、ホスト側で埋め込んだハッシュ値とデバイス側で得られるハッシュ値を比較することで改ざんの検知を行うことができる。

10

【 0 1 4 3 】

ここで、後段の呼び出しモジュールとして暗号化モジュールのような、さらに印刷データに対して暗号表となる鍵を使用して暗号化を行うことで書き換えを行うような暗号化モジュールをアドインモジュール情報ファイル 1 0 3 において発見した場合を考える。このように、上述のようなハッシュ値が算出された後で、上述のような暗号化モジュールが実行されると、暗号化することで得られるハッシュ値も異なったものとなるため、改ざんされたとして検知されてしまう。そのため印刷を中止し、該当モジュールをアンインストールまたは暗号化モジュールと改ざん検知を含む機能制限モジュールの順番を換えてインストールする旨などを表示する。

20

【 0 1 4 4 】

つまり、ジョブの改ざん検知モジュールの処理の後に、暗号化モジュールが動作して、ジョブを暗号化すると、暗号化される前のデータは改ざんされていないにもかかわらず、改ざんされてしまっていると検知される場合がある。係る場合、暗号化処理の後にジョブの改ざん検知モジュールの処理を行うようにするほうがよい。

【 0 1 4 5 】

以下、フローチャートの説明に戻る。

30

【 0 1 4 6 】

ステップ S 1 3 6 0 0 の処理が終了すると、プリンタドライバ 1 0 1 は、ステップ S 1 3 7 0 0 に処理を進める。

【 0 1 4 7 】

ステップ S 1 3 7 0 0 では、プリンタドライバ 1 0 1 は、次のアドインモジュール情報ファイルの参照へ処理を進め、ステップ S 1 3 2 0 0 に処理を戻す。

【 0 1 4 8 】

そして、ステップ S 1 3 2 0 0 で、全てのアドインモジュール情報ファイルの参照が終了したと判定した場合には、プリンタドライバ 1 0 1 は、本フローチャートの処理を終了する。

40

【 0 1 4 9 】

以上説明したように、本実施形態によれば、プリンタドライバにアドインモジュールを追加する際に、該追加するアドインモジュールとプリンタドライバを構成する既存のアドインモジュールとの動作順番（適用順）が正しくなるように設定制御することが可能になる。

【 0 1 5 0 】

従って、印刷処理の実行時に、プリンタドライバは、プリンタドライバを構成する各アドインモジュールを適切な順序（動作させるべき順序）で適用して適切に動作させることが可能となる。

【 0 1 5 1 】

50

〔第2実施形態〕

上記第1実施形態では、プリンタドライバにアドインモジュールを追加する場合について説明したが、本実施形態では、Windows（登録商標）VistaのFilter Pipelineにおいてフィルタ構成で提供されるフィルタモジュールをプリンタドライバに追加する場合について説明する。なお、ハードウェア構成については第1実施形態と同様とする。

【0152】

<システムの構成>

図15は、本発明の第2実施形態に係る印刷システムの構成例を示すブロック図であり、図2と同一のものには同一の符号を付してある。

【0153】

図15において、1401はプリンタドライバ101においてフィルタ構成で提供されるフィルタモジュールの配置情報であり、フィルタモジュール1404の呼び出し順番を規定している。フィルタモジュール配置情報ファイル1401は、Windows（登録商標）VistaにおいてはFilter Configuration File対応する。

【0154】

1404はプリンタドライバを構成しているフィルタモジュールであり、フィルタモジュール情報ファイル1405を含む。ここで、図16を参照して、フィルタモジュール情報ファイル1405のデータ構成について説明する。

【0155】

図16は、図15に示したフィルタモジュール情報ファイル1405のデータ構成を示すデータ構成図である。

【0156】

図16において、1501はフィルタモジュール1404のファイル構成が記述されている領域である。1502はフィルタモジュール1404のバージョン情報が記述されている領域である。1503はフィルタモジュール1404のリリース日が記述されている領域である。1504はフィルタモジュール1404の製品名情報が記述されている領域である。1505はフィルタモジュール1404の詳細情報が記述されている領域である。

【0157】

1506はPriorityNumber（プライオリティナンバ）であり、動作の順番を決定する一意の番号が記述されている領域である。1507はフィルタモジュール1404のCopyright情報が記述されている領域である。

【0158】

なお、プリンタドライバファイル情報104はOSが印刷サービス105を用いてプリンタドライバを利用するための一管理情報で、例えばWindows（登録商標）ではレジストリ中のDependentFilesが該当する。

【0159】

以下、図15の説明に戻る。

【0160】

プリンタドライバ101は、0個または1つ以上のフィルタモジュールを保持することが可能である。フィルタモジュール1404は、例えばDLL（Dynamic Link Library）などのプログラムファイルを含むファイル群である。

【0161】

図15においては、フィルタモジュール1404がインストール済みであり、プリンタドライバファイル情報104に、フィルタモジュール配置情報ファイル1401が登録されている。このフィルタモジュール配置情報ファイル1401には、フィルタモジュール情報ファイルを特定する情報（例えば、フィルタモジュール情報ファイル名称）1402が含まれる。

【0162】

なお、プリンタドライバ101に複数のフィルタモジュールがインストールされている

10

20

30

40

50

場合には、フィルタモジュール配置情報ファイル 1 4 0 1 内には、各フィルタモジュール毎にフィルタモジュール情報ファイルを特定する情報 1 4 0 2 が含まれる（呼出順）。

【 0 1 6 3 】

フィルタアドインインストーラ 1 4 0 3 は、本発明の目的を達成するための中心となるものであり、フィルタモジュール 1 4 0 4 をインストール済みのプリンタドライバ 1 0 1 に追加インストールするプログラムである。なお、追加インストールするフィルタモジュールファイルは、フィルタアドインインストーラ 1 4 0 3 内又は外部メモリ 1 2 内の他の記憶領域に格納されているものとする。

【 0 1 6 4 】

なお、フィルタアドインインストーラ 1 4 0 3 は、主に以下の（１）～（６）の機能を有する。

【 0 1 6 5 】

（１）１つまたは複数のプリンタドライバを構成する１つまたは複数のフィルタモジュールを検出する構成フィルタモジュール検出機能

（２）フィルタモジュールとプリンタドライバを構成する１つまたは複数のフィルタモジュールとの適切な動作の順番（該複数のフィルタモジュールを動作させるべき順番）を判定するフィルタモジュール適正動作順番判定機能

（３）フィルタモジュールファイルを追加するファイル追加機能

（４）上述の動作順番に従って（上述の動作順番で動作するように）フィルタモジュール配置情報を変更するフィルタモジュール配置情報変更機能

（５）追加された追加ファイルをプリンタドライバファイル情報 1 0 4 に登録するファイル登録機能

（６）印刷サービスにモジュール構成の変更を認識させる認識機能

なお、フィルタアドインインストーラ 1 4 0 3 における処理の流れについては、後述する図 1 7 以降を用いて詳細に説明する。

【 0 1 6 6 】

<フィルタアドインインストーラ 1 4 0 3 における処理の流れ>

図 1 7 は、本発明における第 7 の制御処理手順の一例を示すフローチャートであり、フィルタアドインインストーラ 1 4 0 3 の処理の流れに対応する。即ち、このフローチャートの処理は、CPU 1 が外部メモリ 1 2 に格納されたアドインインストーラ 1 0 6 に対応するプログラムを RAM 2 上で実行することにより実現されるものである。なお、図中、S 1 6 0 0 0 ~ S 1 6 4 0 0 は各ステップを示す。

【 0 1 6 7 】

まず、ステップ S 1 6 0 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュールの追加対象プリンタドライバについて、インストール済みのフィルタモジュールの検出を行う。なお、このフィルタモジュールの追加対象プリンタドライバは、コンピュータ 1 0 0 にインストール済みのプリンタドライバであり、ユーザによってフィルタモジュールの追加対象として選択されたものが対応する。ここでは、図 1 5 のプリンタドライバ 1 0 1 とする。また、このステップ S 1 6 0 0 0 の処理の詳細は図 1 8 において説明する。

【 0 1 6 8 】

次に、ステップ S 1 6 1 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュール順番情報を作成する。この処理は、ステップ S 1 6 0 0 0 で検出されたインストール済みのフィルタモジュール 1 4 0 4 と、フィルタアドインインストーラ 1 4 0 3 により追加するフィルタモジュールを合わせた印刷処理において、適正な順番情報（前記複数のフィルタモジュールを動作させるべき順番）を保持するフィルタモジュール順番情報を作成する。なお、このフィルタモジュール順番情報は、適正に動作可能な順番（前記複数のフィルタモジュールを動作させるべき順番）が存在した場合にのみ作成される。

【 0 1 6 9 】

また、このフィルタモジュール順番情報の保持形態としては、ファイル形式や RAM 2

10

20

30

40

50

上に保持する等形態は問わないが、ここではRAM 2上に保持するものとして説明する。このフィルタモジュール順番情報は、最小の情報構成として順番を示す番号とモジュール名称を保持している。なお、このステップS 1 6 1 0 0の処理の詳細は図19において説明する。

【0170】

次に、ステップS 1 6 2 0 0において、フィルタアドインインストーラ1403は、フィルタモジュール順番情報がRAM 2上に存在するか否かを判定する。そして、フィルタモジュール順番情報がRAM 2上に存在しないと判定した場合には、フィルタアドインインストーラ1403は、ステップS 1 6 4 0 0に処理を進める。

【0171】

ステップS 1 6 4 0 0では、フィルタアドインインストーラ1403は、フィルタモジュールインストール処理を中止し、本フローチャートの処理を終了する。

【0172】

一方、ステップS 1 6 2 0 0で、アドインモジュール順番情報がRAM 2上に存在すると判定した場合には、フィルタアドインインストーラ1403は、ステップS 1 6 3 0 0に処理を進める。

【0173】

ステップS 1 6 3 0 0では、フィルタアドインインストーラ1403は、フィルタモジュールインストール処理を行い、本フローチャートの処理を終了する。なお、このステップS 1 6 3 0 0の処理の詳細は図20において説明する。

【0174】

図18は、本発明における第8の制御処理手順の一例を示すフローチャートであり、図17のステップS 1 6 0 0 0に示したフィルタモジュール検出処理の詳細な流れに対応する。即ち、このフローチャートの処理は、CPU 1が外部メモリ12に格納されたフィルタアドインインストーラ1403に対応するプログラムをRAM 2上で実行することにより実現されるものである。なお、図中、S 1 7 0 0 0～S 1 7 6 0 0は各ステップを示す。

【0175】

まず、ステップS 1 7 0 0 0において、フィルタアドインインストーラ1403は、プリンタドライバファイル情報104を参照して、プリンタドライバ101についての構成情報を取得する。

【0176】

次に、ステップS 1 7 1 0 0において、フィルタアドインインストーラ1403は、ステップS 1 7 0 0 0によって取得された構成情報内にフィルタモジュール配置情報ファイル1401が存在するか否かを判定する。

【0177】

ステップS 1 7 1 0 0で、フィルタモジュール配置情報ファイル1401が存在しないと判定した場合には、フィルタアドインインストーラ1403は、そのまま本フローチャートの処理を終了して、図17のステップS 1 6 1 0 0に処理を進める。

【0178】

一方、ステップS 1 7 1 0 0で、フィルタモジュール配置情報ファイル1401が存在すると判定した場合には、フィルタアドインインストーラ1403は、ステップS 1 7 2 0 0に処理を進める。

【0179】

ステップS 1 7 2 0 0では、フィルタアドインインストーラ1403は、フィルタモジュール配置情報ファイル1401内に登録された全てのフィルタモジュール情報ファイル1405の参照が終了したか否かを判定する。

【0180】

そして、ステップS 1 7 2 0 0で、まだ全てのフィルタモジュール情報ファイル1405の参照が終了していないと判定した場合には、フィルタアドインインストーラ1403

10

20

30

40

50

は、ステップ S 1 7 3 0 0 に処理を進める。

【 0 1 8 1 】

ステップ S 1 7 3 0 0 では、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュール配置情報ファイル 1 4 0 1 内に登録されており且つ未だ参照していないフィルタモジュール情報ファイル 1 4 0 5 を 1 つ選択し、該フィルタモジュール情報ファイル 1 4 0 5 内の情報を参照する。

【 0 1 8 2 】

次に、ステップ S 1 7 4 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 1 7 3 0 0 でフィルタモジュール情報ファイル 1 4 0 5 内の情報が正常に参照されたか否かを判定する。

10

【 0 1 8 3 】

ステップ S 1 7 4 0 0 において、ステップ S 1 7 3 0 0 でフィルタモジュール情報ファイル 1 4 0 5 内の情報が正常に参照されなかったと判定した場合には、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 1 7 5 0 0 に処理を進める。

【 0 1 8 4 】

ステップ S 1 7 5 0 0 では、フィルタアドインインストーラ 1 4 0 3 は、このフィルタモジュールは正常に動作していない、即ちインストールされていないとみなし、ステップ S 1 7 6 0 0 に処理を進める。

【 0 1 8 5 】

一方、ステップ S 1 7 4 0 0 において、ステップ S 1 7 3 0 0 でフィルタモジュール情報ファイル 1 4 0 5 内の情報が正常に参照されたと判定した場合には、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 1 7 6 0 0 に処理を進める。

20

【 0 1 8 6 】

ステップ S 1 7 6 0 0 では、フィルタアドインインストーラ 1 4 0 3 は、次のフィルタモジュール情報ファイル 1 4 0 5 の参照へ処理を進め、ステップ S 1 7 2 0 0 に処理を戻す。

【 0 1 8 7 】

そして、ステップ S 1 7 2 0 0 で、フィルタモジュール配置情報ファイル 1 4 0 1 内に登録された全てのフィルタモジュール情報ファイル 1 4 0 5 の参照が終了したと判定した場合には、フィルタアドインインストーラ 1 4 0 3 は、本フローチャートの処理を終了して、図 1 7 のステップ S 1 6 1 0 0 に処理を進める。

30

【 0 1 8 8 】

以上が、図 1 7 のステップ S 1 6 0 0 0 に示したフィルタモジュール検出処理の詳細動作となり、フィルタモジュールを追加するプリンタドライバが複数選択されている場合には、選択されているプリンタドライバの数だけ同様の処理を繰り返す。

【 0 1 8 9 】

図 1 9 は、本発明における第 9 の制御処理手順の一例を示すフローチャートであり、図 1 7 のステップ S 1 6 1 0 0 に示したフィルタモジュール順番情報作成処理の詳細な流れに対応する。即ち、このフローチャートの処理は、CPU 1 が外部メモリ 1 2 に格納されたフィルタアドインインストーラ 1 4 0 3 に対応するプログラムを RAM 2 上で実行することにより実現されるものである。なお、図中、S 1 8 0 0 0 ~ S 1 8 8 0 0 は各ステップを示す。

40

【 0 1 9 0 】

まず、ステップ S 1 8 0 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュール順番情報の新規作成など初期化処理を行う。

【 0 1 9 1 】

ステップ S 1 8 1 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュールの追加対象プリンタドライバに対し、図 1 7 のステップ S 1 6 0 0 0 で検出された及びこれから追加する全てのフィルタモジュールのフィルタ情報ファイル取得を終了したか否かを判定する。即ち、フィルタモジュール順番情報の作成処理を完了したか否かを

50

判定する。

【0192】

ステップS18100で、図17のステップS16000で検出された及びこれから追加するアドインモジュールの全てのフィルタモジュール情報ファイル取得を未だ終了していないと判定した場合には、フィルタアドインインストーラ1403は、ステップS18200に処理を進める。

【0193】

ステップS18200では、フィルタアドインインストーラ1403は、図17のステップS16000で検出された及びこれから追加するフィルタモジュール内のまだ取得していないフィルタモジュール情報ファイル1405を1つ取得する。以下、このフィルタモジュール情報ファイル1405に対応するアドインモジュールを現在注目しているアドインモジュールと呼ぶ。

10

【0194】

ステップS18300では、フィルタアドインインストーラ1403は、ステップS18200で取得したフィルタモジュール情報ファイル1405からPriorityNumber1506を取得する。なお、PriorityNumberとは例えば1以上の整数などであり、数値の大小によって動作する順番を定義するものである。このPriorityNumberによって新規にモジュールを追加した場合にも対応が可能となる。

【0195】

ステップS18400では、フィルタアドインインストーラ1403は、本ステップまでにフィルタモジュール順番情報内に書き込み済みのフィルタモジュールのPriorityNumberと現在注目しているフィルタモジュールのPriorityNumberとを比較する。そして、フィルタアドインインストーラ1403は、PriorityNumberの大きい順となるように順番（フィルタモジュール順番情報への書き込み位置）を決定する。

20

【0196】

次に、ステップS18500において、フィルタアドインインストーラ1403は、ステップS18400で決定された書込位置に書き込みが可能か否かを判定する。なお、書き込みが不可能な場合としては、PriorityNumberが同一のものや共存が不可能なモジュールの競合等が考えられる。

【0197】

ステップS18500で、書き込みが不可能であると判定した場合、フィルタアドインインストーラ1403は、ステップS18800に処理を進める。

30

【0198】

ステップS18800では、フィルタアドインインストーラ1403は、適正な順番（予め追加されているフィルタモジュールとこれから追加するフィルタモジュールとを動作させるべき順番）が存在しないと判断して、現在RAM2に保持しているフィルタモジュール順番情報を削除する。そして、フィルタアドインインストーラ1403は、図示しないステップにおいて、PriorityNumberが同一であったり、共存が不可能なモジュールの競合等が発生しモジュールを適用する適正な順番（動作させるべき順番）が存在しない旨の通知をディスプレイ11に表示する。そして、本フローチャートの処理を終了し、図17のステップS16200に処理を進める。

40

【0199】

一方、ステップS18500で、書き込みが可能であると判定した場合には、フィルタアドインインストーラ1403は、ステップS18600に処理を進める。

【0200】

ステップS18600では、フィルタアドインインストーラ1403は、現在注目しているフィルタモジュールのフィルタモジュール名称を、ステップS18400で決定したフィルタモジュール順番情報内の書込位置に書込処理する。本実施形態では、順番を示す番号とフィルタモジュール名称を書込処理する。

【0201】

50

次に、ステップ S 1 8 7 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、次のフィルタモジュール情報ファイルへ処理を進め、ステップ S 1 8 1 0 0 に処理を戻す。

【 0 2 0 2 】

そして、ステップ S 1 8 1 0 0 で、図 1 7 のステップ 1 6 0 0 0 で検出された及びこれから追加するフィルタモジュールの全てのフィルタモジュール情報ファイル取得を終了したと判定した場合には、本フローチャートの処理を終了し、図 1 7 のステップ S 1 6 2 0 0 に処理を進める。

【 0 2 0 3 】

以上が図 1 7 のステップ S 1 6 1 0 0 に示したフィルタモジュール順番情報作成処理の詳細動作となり、フィルタモジュールを追加するプリンタドライバが複数選択されている場合には選択されているプリンタドライバの数だけ同様の処理を繰り返す。

10

【 0 2 0 4 】

なお、図 1 9 では各フィルタモジュールが PriorityNumber を保持することでフィルタモジュール順番情報を作成したが、フィルタモジュール順番情報をファイル情報として外部から提供するようにしてもよい。例えば、フィルタモジュール順番情報をファイル情報として Web 上で公開し、該 Web 上でから取得して提供するように構成してもよい。

【 0 2 0 5 】

また、フィルタモジュール順番情報は、XML のようなテキスト形式でも非可読のバイナリ形式でもよくデータ形式は問わない。

【 0 2 0 6 】

20

図 2 0 は、本発明における第 1 0 の制御処理手順の一例を示すフローチャートであり、図 1 7 のステップ S 1 6 3 0 0 に示したフィルタモジュールアドインインストール処理の詳細な流れに対応する。即ち、このフローチャートの処理は、CPU 1 が外部メモリ 1 2 に格納されたフィルタアドインインストーラ 1 4 0 3 に対応するプログラムを RAM 2 上で実行することにより実現されるものである。なお、図中、S 1 9 0 0 0 ~ 1 9 7 0 0 は各ステップを示す。

【 0 2 0 7 】

まず、ステップ S 1 9 0 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、プリンタドライバファイル情報 1 0 4 内に登録されたフィルタモジュール配置情報ファイル 1 4 0 1 を初期化する。ここでは一度、フィルタモジュール配置情報ファイル 1 4 0 1 に記載されている各フィルタモジュール情報ファイルを特定する情報 1 4 0 2 についての消去を行う。

30

【 0 2 0 8 】

次に、ステップ S 1 9 1 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 1 6 1 0 0 で作成されたフィルタモジュール順番情報を取得する。

【 0 2 0 9 】

次に、ステップ S 1 9 2 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 1 9 1 0 0 で取得したフィルタモジュール順番情報に記載されているフィルタモジュールのアドインインストール処理が全て完了したか否かを判定する。

【 0 2 1 0 】

40

ステップ S 1 9 2 0 0 で、ステップ S 1 9 1 0 0 で取得したフィルタモジュール順番情報に記載されているフィルタモジュールのアドインインストール処理がまだ全て完了していないと判定した場合には、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 1 9 3 0 0 に処理を進める。

【 0 2 1 1 】

ステップ S 1 9 3 0 0 では、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュール順番情報に従い動作の順番が早いものから 1 つフィルタモジュール（フィルタモジュール構成ファイル）を取得する。

【 0 2 1 2 】

次に、ステップ S 1 9 4 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、ス

50

ステップ S 1 9 3 0 0 で取得したフィルタモジュール構成ファイルを外部メモリ 1 2 内の所定領域 (O S の記憶領域) に追加する。なお、このステップは、今回新たに追加インストールするフィルタモジュールに対してのみ行い、予め追加済みであったフィルタモジュールに関してはスキップするものとする。

【 0 2 1 3 】

次に、ステップ S 1 9 5 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、ステップ S 9 4 0 0 で取得したプリンタドライバ構成情報 (プリンタドライバファイル情報 1 0 4) 内にフィルタモジュールを登録する。

【 0 2 1 4 】

次に、ステップ S 1 9 6 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、追加したフィルタモジュールのフィルタモジュール情報ファイルを特定する情報 1 1 2 を現在のフィルタモジュール配置情報ファイルのフィルタ構成の終端に追加する。

【 0 2 1 5 】

次に、ステップ S 1 9 7 0 0 において、フィルタアドインインストーラ 1 4 0 3 は、フィルタモジュール順番情報に次の記載されているフィルタモジュールに処理を移し、ステップ S 1 9 2 0 0 へ処理を戻す。

【 0 2 1 6 】

そして、ステップ S 1 9 2 0 0 で、フィルタモジュール順番情報に記載されているフィルタモジュールのインストール処理が全て完了したと判定した場合には、本フローチャートの処理を終了する。

【 0 2 1 7 】

以上のようにインストール処理することで、予め追加されているフィルタモジュールと新たに追加したフィルタモジュールとが、図 1 6 のステップ S 1 3 1 0 0 で判定された動作の順番で処理を実行するように、新たなフィルタモジュールを登録することができる。

【 0 2 1 8 】

< フィルタアドインインストーラ UI について >

フィルタアドインインストーラ UI については、上述した第 1 実施形態におけるアドインインストーラ UI と同様である。なお、アドインインストール処理自体は、図 4 の処理が図 1 7 の処理に置き換わるものとする。

【 0 2 1 9 】

< フィルタモジュール 1 4 0 4 の呼び出し処理の流れ >

本実施形態によりインストールされたフィルタモジュールを備えたプリンタドライバの印刷処理の詳細を、図 1、図 1 5、および図 2 1 を用いて説明する。なお、第 1 実施形態において用いた図と番号が等しいものは同様のものを示している。

【 0 2 2 0 】

尚、Windows (登録商標) Vista の XPS (XML Paper Specification) Print Path においてグラフィックエンジン 1 0 9 は、アプリケーション 1 0 8 から受け取る描画命令または X P S ドキュメントを X P S ドキュメントとしてプリンタドライバ 1 0 1 へ出力する。

【 0 2 2 1 】

プリンタドライバ 1 0 1 は、グラフィックエンジン 1 0 9 から受け取った X P S ドキュメントをプリンタが認識可能な制御コマンド、例えば P D L (Page Description Language) に変換する。本実施形態の印刷システムは、プリンタドライバ 1 0 1 内のフィルタモジュール 1 4 0 4 を O S が呼び出すことで印刷が実行される。プリンタドライバ 1 0 1 は、フィルタモジュール 1 4 0 4 の実行に際し O S を通してフィルタモジュール 1 4 0 4 を呼び出し、処理を行う。フィルタモジュールの呼び出し処理の詳細に関しては図 2 1 を用いて示す。

【 0 2 2 2 】

図 2 1 は、本発明における第 1 1 の制御処理手順の一例を示すフローチャートであり、プリンタドライバ 1 0 1 によるフィルタモジュールの呼び出し処理の流れに対応する。即ち、このフローチャートの処理は、C P U 1 が外部メモリ 1 2 に格納されたアドインイン

10

20

30

40

50

ストーリー 106 に対応するプログラムを RAM 2 上で実行することにより実現されるものである。なお、図中、S 20000 ~ S 20700 は各ステップを示す。

【0223】

まず、ステップ S 20000 において、プリンタドライバ 101 は、プリンタドライバファイル情報 104 を参照して、プリンタドライバ 101 についての構成情報を取得する。

【0224】

次に、ステップ S 20100 において、プリンタドライバ 101 は、ステップ S 20000 によって取得された構成情報内にフィルタモジュール配置情報ファイル 1401 が存在するか否かを判定する。なお、本実施形態では、図 20 に示した処理によりフィルタモジュール配置情報ファイル 1401 は動作順番の早いものから順番にフィルタモジュール情報ファイルを特定する情報 1402 を登録している。

10

【0225】

ステップ S 20100 で、フィルタモジュール配置情報ファイル 1401 が存在しないと判定した場合には、プリンタドライバ 101 は、そのまま本フローチャートの処理を終了する。

【0226】

一方、ステップ S 20100 で、フィルタモジュール配置情報ファイル 1401 が存在すると判定した場合には、プリンタドライバ 101 は、ステップ S 20200 に処理を進める。

20

【0227】

ステップ S 20200 では、プリンタドライバ 101 は、フィルタモジュール配置情報ファイル 1401 内に登録された全てのフィルタモジュール 1404 の呼出が終了したか否かを判定する。

【0228】

そして、ステップ S 20200 で、まだ全てのフィルタモジュール 1404 の呼出が終了していないと判定した場合には、プリンタドライバ 101 は、ステップ S 20300 に処理を進める。

【0229】

ステップ S 20300 では、プリンタドライバ 101 は、フィルタモジュール配置情報ファイル 1401 内に登録されており且つ未だ呼び出されていないフィルタモジュール 1404 を 1 つ選択し、該フィルタモジュール 1404 を呼出処理する。

30

【0230】

以上説明したように、本実施形態によれば、プリンタドライバにフィルタモジュールを追加する際、該追加するフィルタモジュールとプリンタドライバを構成する既存のフィルタモジュールとの印刷処理時の動作順番（適用順）が正しくなるように設定制御することが可能になる。

【0231】

従って、印刷処理の実行時に、プリンタドライバは、プリンタドライバを構成する各フィルタモジュールを適切な順序（動作させるべき順番）で適用して適切に動作させることが可能となる。

40

【0232】

なお、第 1 , 2 実施形態では、プリンタドライバに新たなモジュールを追加する際、該追加するモジュールとプリンタドライバを構成する既存のモジュールとの印刷処理時の動作順番（適用順）が正しくなるように新たなモジュールを追加するとともに、既存のモジュールの登録状態を変更制御する構成について説明した。

【0233】

しかし、新たに追加するモジュールがない状態で図 4 ~ 図 8 又は図 17 ~ 図 20 に示したフローチャートの処理を実行することにより、既存のモジュールの印刷処理時の動作順番（適用順）が正しくなるように既存のモジュールの登録を変更制御することが可能に

50

なる。

【 0 2 3 4 】

例えば、図 4 ~ 図 8 又は図 1 7 ~ 図 2 0 に示したフローチャートの処理を実行することなくユーザが従来の方法で新たなモジュールを追加した場合、該モジュールは既にインストールされているモジュールとの動作順番について考慮されず、正常な動作をしない場合がある。

【 0 2 3 5 】

このような場合、モジュール自体は追加済みであるので、新たなに追加するモジュールがない状態で図 1 7 ~ 図 2 0 に示したフローチャートの処理を実行する。これにより、インストーラは、新たなモジュールは追加しないが、既にインストールされているモジュールの動作順番（適用順）が正しくなるように、既存のモジュールの登録を変更制御する。

10

【 0 2 3 6 】

なお、上記各実施形態では、コンピュータにインストールされた 1 つまたは複数のプリンタドライバに 1 又は複数のアドインモジュール（フィルタモジュール）を追加するインストーラ及びプリンタドライバにおけるモジュールの適用制御について説明した。

【 0 2 3 7 】

しかしながら、アドインモジュール等を追加する対象は、プリンタドライバでなくても、アドインモジュール等を追加可能なアプリケーションプログラムであればどのようなアプリケーションプログラムであってもよい。即ち、コンピュータにインストールされた 1 つまたは複数のアプリケーションプログラムに 1 又は複数のアドインモジュール（フィルタモジュール）を追加するインストーラ及びアプリケーションプログラムにおけるモジュールの適用制御にも本発明は適用可能である。

20

【 0 2 3 8 】

なお、本実施形態では、上述の各種データの構成及びその内容はこれに限定されるものではなく、用途や目的に応じて、様々な構成や内容で構成されることは言うまでもない。

【 0 2 3 9 】

以上、一実施形態について示したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記憶媒体等としての実施態様をとることが可能である。具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

30

【 0 2 4 0 】

以下、図 2 1 に示すメモリマップを参照して、本発明に係る情報処理装置（コンピュータ 1 0 0 ）で読み取り可能な各種データ処理プログラムを格納する記憶媒体のメモリマップの構成について説明する。

【 0 2 4 1 】

図 2 1 は、本発明に係る情報処理装置（コンピュータ 1 0 0 ）で読み取り可能な各種データ処理プログラムを格納する記憶媒体（記録媒体）のメモリマップを説明する図である。

【 0 2 4 2 】

なお、特に図示しないが、記憶媒体に記憶されるプログラム群を管理する情報、例えばバージョン情報、作成者等も記憶され、かつ、プログラム読み出し側の OS 等に依存する情報、例えばプログラムを識別表示するアイコン等も記憶される場合もある。

40

【 0 2 4 3 】

さらに、各種プログラムに従属するデータも上記ディレクトリに管理されている。また、各種プログラムをコンピュータにインストールするためのプログラムや、インストールするプログラムが圧縮されている場合に、解凍するプログラム等も記憶される場合もある。

【 0 2 4 4 】

本実施形態における図 4 , 図 5 , 図 6 , 図 7 , 図 8 , 図 1 4 , 図 1 7 , 図 1 8 , 図 1 9 , 図 2 0 , 図 2 1 に示す機能が外部からインストールされるプログラムによって、ホスト

50

コンピュータにより遂行されていてもよい。そして、その場合、CD-ROMやフラッシュメモリやFD等の記憶媒体により、あるいはネットワークを介して外部の記憶媒体から、プログラムを含む情報群を出力装置に供給される場合でも本発明は適用されるものである。

【0245】

以上のように、前述した実施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録した記憶媒体を、システムあるいは装置に供給する。そして、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）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、本発明の目的が達成されることは言うまでもない。

【0246】

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は本発明を構成することになる。

10

【0247】

従って、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等、プログラムの形態を問わない。

【0248】

プログラム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、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモリカード、ROM、DVDなどを用いることができる。

20

【0249】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の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は本発明を構成することになる。

【0250】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続し、該ホームページから本発明のプログラムそのものをハードディスク等の記憶媒体にダウンロードすることによっても供給できる。また、該ホームページから圧縮され自動インストール機能を含むファイルをハードディスク等の記憶媒体にダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバやFTPサーバ等も本発明の請求項に含まれるものである。

30

【0251】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布する。さらに、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせる。さらに、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

40

【0252】

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、前述した実施形態の機能が実現されるだけでなく、以下のような構成も含まれることは言うまでもない。例えば、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働している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。

【0253】

さらに、記憶媒体から読み出されたプログラムコードを、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む。そして、該メモリに書き込まれたプログラムコードの指示に基づき、その機能拡張ボード

50

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。

【0254】

また、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器からなる装置に適用してもよい。また、本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適用できることは言うまでもない。この場合、本発明を達成するためのソフトウェアによって表されるプログラムを格納した記憶媒体を該システムあるいは装置に読み出すことによって、そのシステムあるいは装置が、本発明の効果を享受することが可能となる。

【0255】

本発明は上記実施形態に限定されるものではなく、本発明の趣旨に基づき種々の変形（各実施形態の有機的な組み合わせを含む）が可能であり、それらを本発明の範囲から排除するものではない。

【0256】

本発明の様々な例と実施形態を示して説明したが、当業者であれば、本発明の趣旨と範囲は、本明細書内の特定の説明に限定されるのではない。

【0257】

なお、上述した各実施形態およびその変形例を組み合わせた構成も全て本発明に含まれるものである。

【0258】

以上説明したように、本発明は、情報処理装置にインストールされたプリンタドライバに対してモジュールを追加することが可能なインストーラであって、モジュールを追加するときに現在インストールされているモジュールを検出し、追加するモジュールとインストールされているモジュールとの動作順番を考慮してインストールを行う。よって、プリンタドライバにモジュールを追加する際に、該追加するモジュールとプリンタドライバを構成する既存のモジュールとの動作順番を正しくなるように設定制御することが可能になる。

【0259】

また、本発明は、印刷実行時には前記インストール方法に基づき適切な順番（動作させるべき順番）でインストールされているモジュールを動作させるプリンタドライバも含む。従って、印刷処理の実行時に、プリンタドライバは、プリンタドライバを構成する各タモジュールを適切に適用して動作させることが可能となる。

【0260】

以上のように新規機能をアドインモジュールやフィルタモジュールで提供することにより、印刷装置の発売当初には想定されていない機能や、付加的に後から追加したい機能したい機能を、動作順序正しく追加することが可能なフレキシブルなプリンタドライバの新機能提供環境を構築することができる。

【0261】

なお、上述した各実施形態及びその変形例のいずれか又は全てを組み合わせた構成も本発明に含まれるものである。

【図面の簡単な説明】

【0262】

【図1】本発明を適用可能な情報処理装置の一例を示すブロック図である。

【図2】本発明の第1実施形態に係る印刷システムの構成例を示すブロック図である。

【図3】図2に示したアドインモジュール情報ファイルのデータ構成を示すデータ構成図である。

【図4】本発明における第1の制御処理手順の一例を示すフローチャートである。

【図5】本発明における第2の制御処理手順の一例を示すフローチャートである。

【図6】本発明における第3の制御処理手順の一例を示すフローチャートである。

【図7】本発明における第4の制御処理手順の一例を示すフローチャートである。

【図 8】本発明における第 5 の制御処理手順の一例を示すフローチャートである。

【図 9】アドインインストーラによるアドインインストール時の UI 表示処理時の表示画面の一例を示す模式図である。

【図 10】アドインインストーラによるアドインインストール時の UI 表示処理時の表示画面の一例を示す模式図である。

【図 11】アドインインストーラによるアドインインストール時の UI 表示処理時の表示画面の一例を示す模式図である。

【図 12】アドインインストーラによるアドインインストール時の UI 表示処理時の表示画面の一例を示す模式図である。

【図 13】アドインインストーラによるアドインインストール時の UI 表示処理時の表示画面の一例を示す模式図である。

10

【図 14】本発明における第 6 の制御処理手順の一例を示すフローチャートである。

【図 15】本発明の第 2 実施形態に係る印刷システムの構成例を示すブロック図である。

【図 16】図 15 に示したフィルタモジュール情報ファイルのデータ構成を示すデータ構成図である。

【図 17】本発明における第 7 の制御処理手順の一例を示すフローチャートである。

【図 18】本発明における第 8 の制御処理手順の一例を示すフローチャートである。

【図 19】本発明における第 9 の制御処理手順の一例を示すフローチャートである。

【図 20】本発明における第 10 の制御処理手順の一例を示すフローチャートである。

【図 21】本発明における第 11 の制御処理手順の一例を示すフローチャートである。

20

【図 22】本発明に係る情報処理装置（コンピュータ）で読み取り可能な各種データ処理プログラムを格納する記憶媒体（記録媒体）のメモリマップを説明する図である。

【符号の説明】

【 0 2 6 3 】

1 CPU

2 RAM

3 ROM

1 2 外部メモリ

3 0 プリンタ

1 0 0 コンピュータ

30

1 0 1 プリンタドライバ

1 0 2 アドインモジュール

1 0 3 アドインモジュール情報ファイル

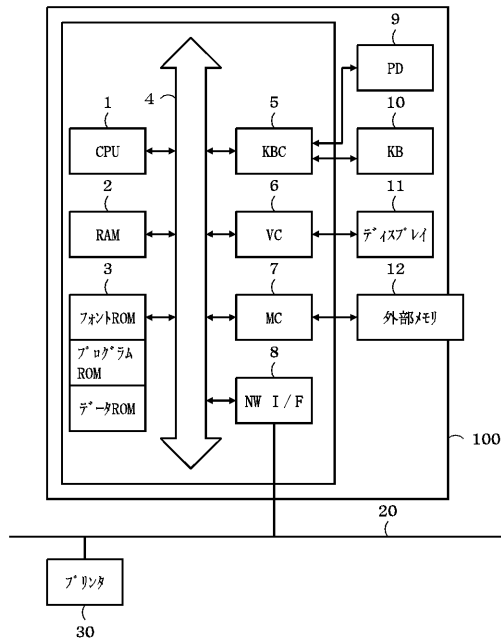
1 0 4 プリンタドライバファイル情報

1 0 5 印刷サービス

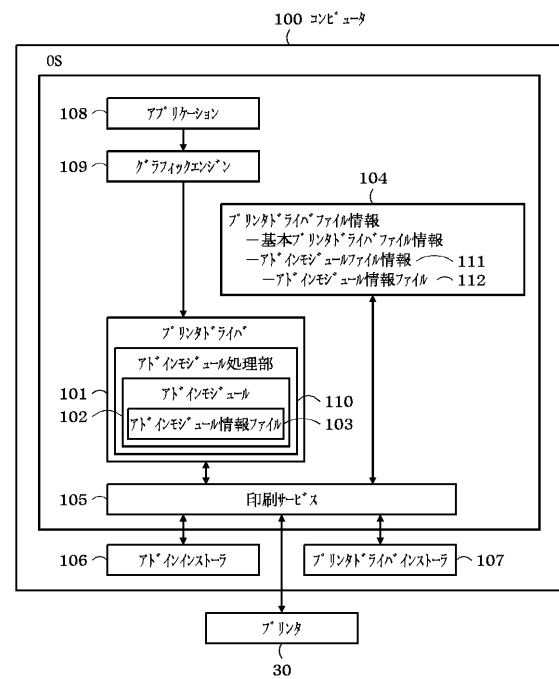
1 0 6 アドインインストーラ

1 0 7 プリンタドライバインストーラ

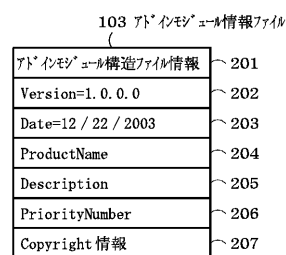
【図 1】



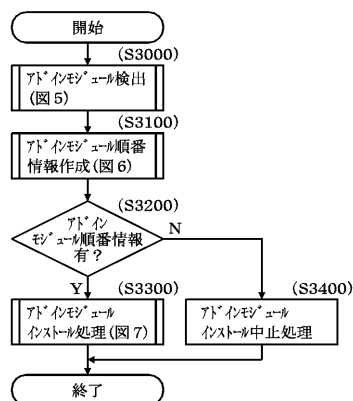
【図 2】



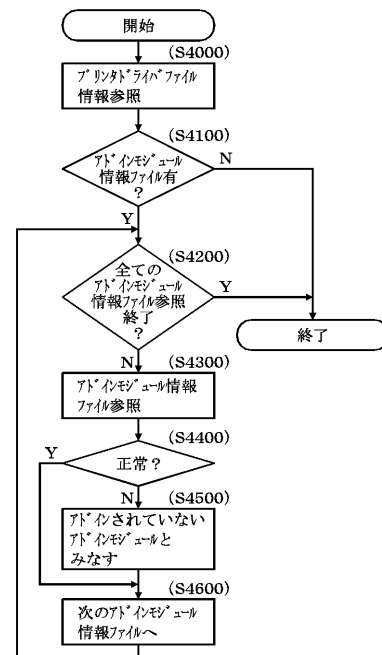
【図 3】



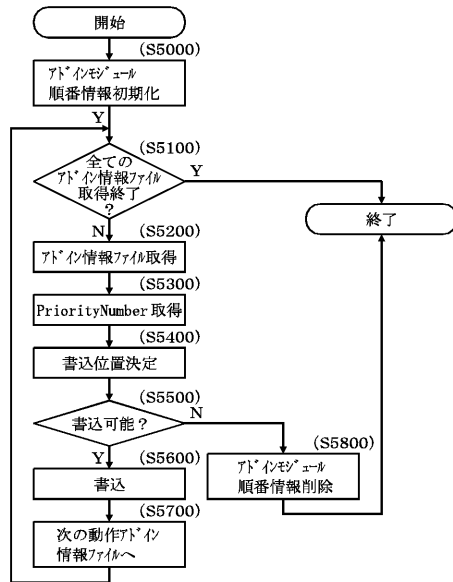
【図 4】



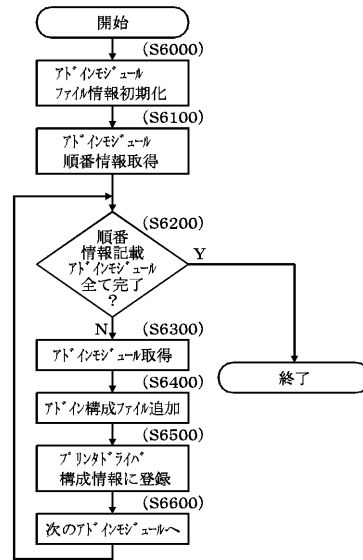
【図 5】



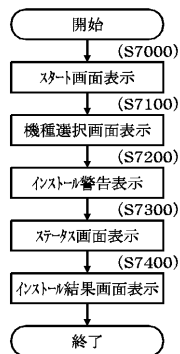
【図 6】



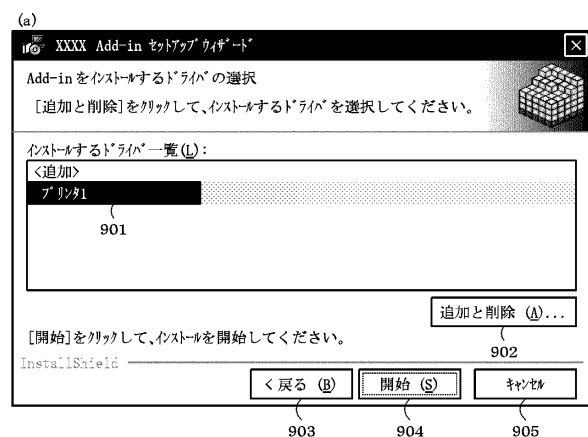
【図 7】



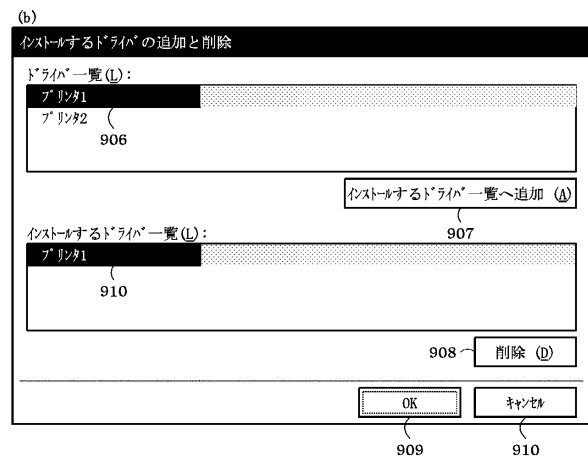
【図 8】



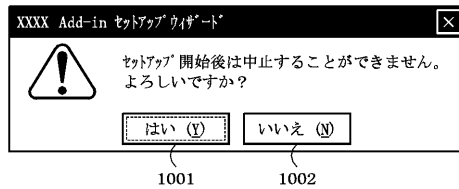
【図 10】



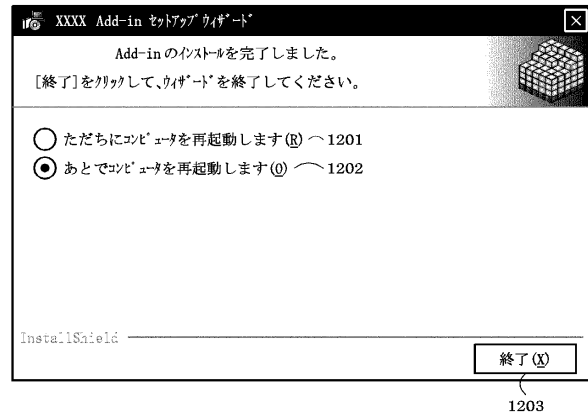
【図 9】



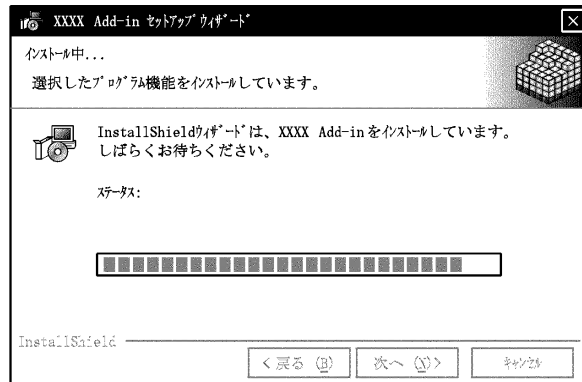
【図 1 1】



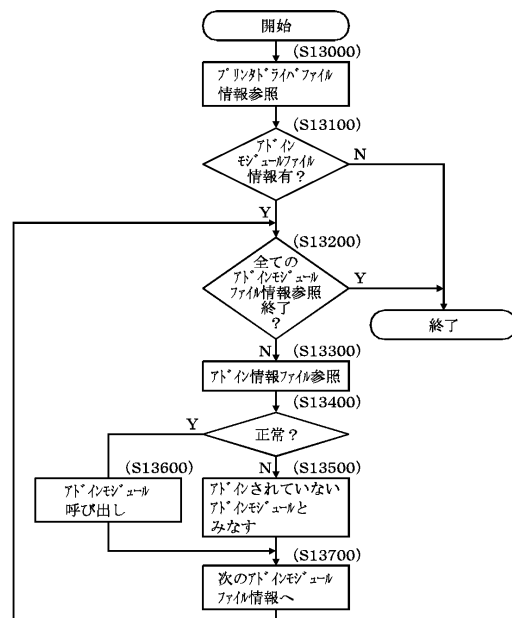
【図 1 3】



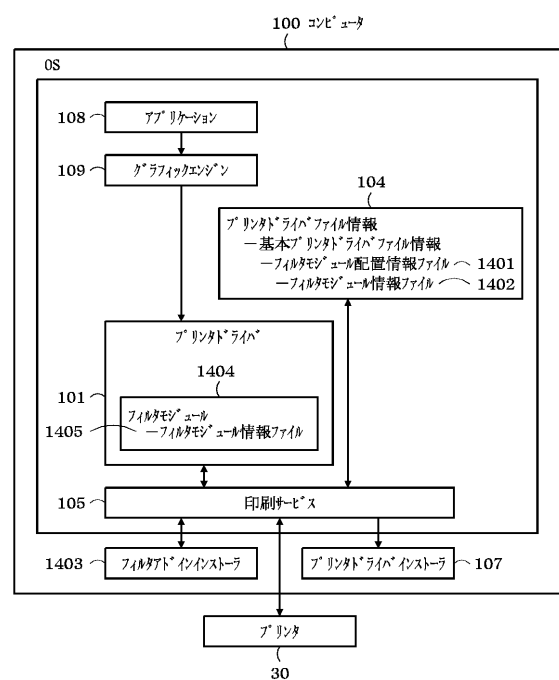
【図 1 2】



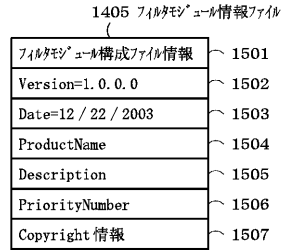
【図 1 4】



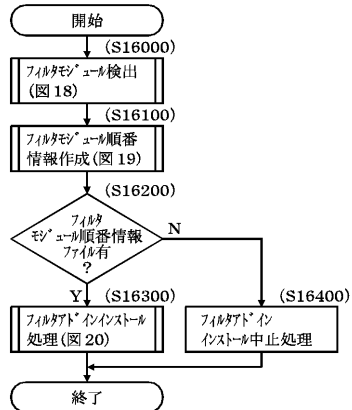
【図 1 5】



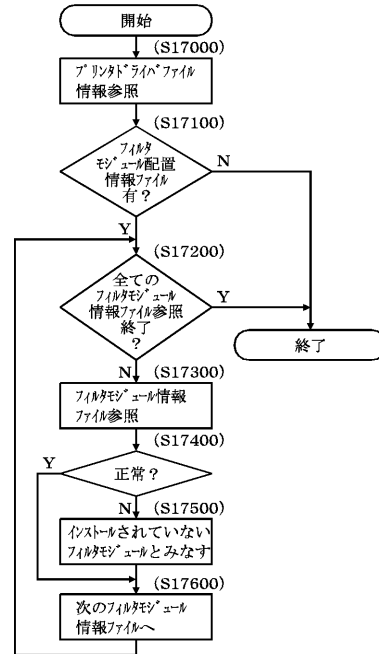
【図 16】



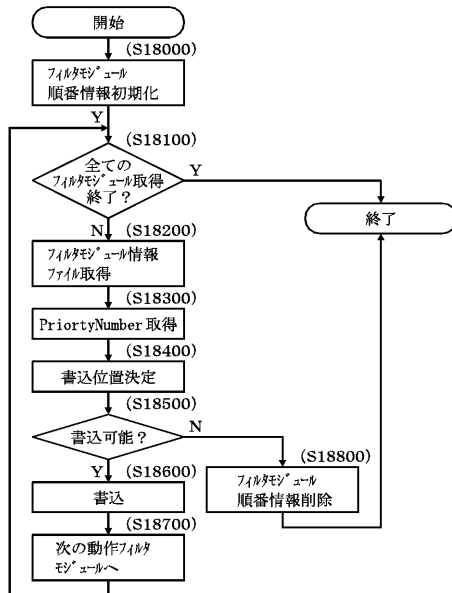
【図 17】



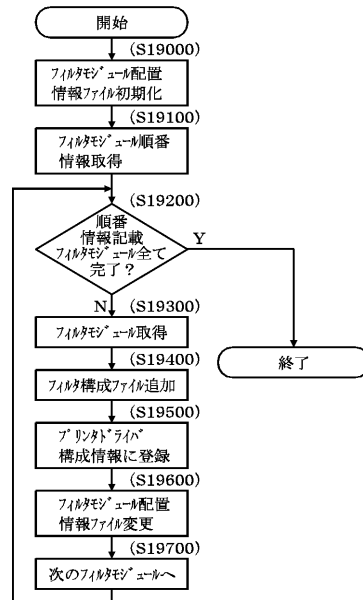
【図 18】



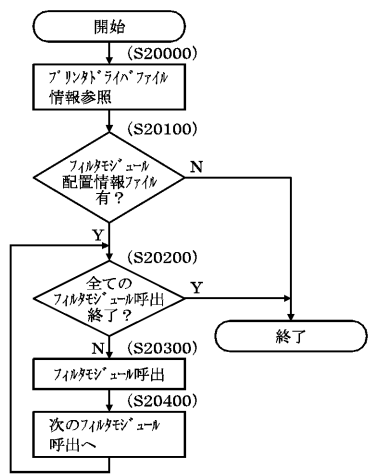
【図 19】



【図 20】



【図 2 1】



【図 2 2】

FD/CD-ROM等の記憶媒体	
ディレクトリ情報	
第1のデータ処理プログラム	図4に示すフローチャートのステップに対応するプログラムコード群
第2のデータ処理プログラム	図5に示すフローチャートのステップに対応するプログラムコード群
第3のデータ処理プログラム	図6に示すフローチャートのステップに対応するプログラムコード群
第4のデータ処理プログラム	図7に示すフローチャートのステップに対応するプログラムコード群
第5のデータ処理プログラム	図8に示すフローチャートのステップに対応するプログラムコード群
第6のデータ処理プログラム	図14に示すフローチャートのステップに対応するプログラムコード群
第7のデータ処理プログラム	図17に示すフローチャートのステップに対応するプログラムコード群
第8のデータ処理プログラム	図18に示すフローチャートのステップに対応するプログラムコード群
第9のデータ処理プログラム	図19に示すフローチャートのステップに対応するプログラムコード群
第10のデータ処理プログラム	図20に示すフローチャートのステップに対応するプログラムコード群
第11のデータ処理プログラム	図21に示すフローチャートのステップに対応するプログラムコード群

記憶媒体のメモリマップ

フロントページの続き

(56)参考文献 特開2005-208895(JP,A)
特開2006-085529(JP,A)
特開昭63-268031(JP,A)
特開2005-050062(JP,A)
特開平10-124435(JP,A)
特開平06-274328(JP,A)
特開2003-058378(JP,A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F	3 / 1 2
G 0 6 F	1 3 / 1 0
G 0 6 F	9 / 4 4 5