

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5866316号
(P5866316)

(45) 発行日 平成28年2月17日(2016.2.17)

(24) 登録日 平成28年1月8日(2016.1.8)

(51) Int.Cl.		F I			
G06F	9/54	(2006.01)	G06F	9/06	640C
G06F	9/44	(2006.01)	G06F	9/06	620A
G06F	21/51	(2013.01)	G06F	21/51	

請求項の数 2 (全 13 頁)

(21) 出願番号	特願2013-136317 (P2013-136317)	(73) 特許権者	000006150
(22) 出願日	平成25年6月28日 (2013.6.28)		京セラドキュメントソリューションズ株式会社
(65) 公開番号	特開2015-11518 (P2015-11518A)		大阪府大阪市中央区玉造1丁目2番28号
(43) 公開日	平成27年1月19日 (2015.1.19)	(74) 代理人	100097113
審査請求日	平成27年4月20日 (2015.4.20)		弁理士 堀 城之
		(74) 代理人	100162363
			弁理士 前島 幸彦
		(72) 発明者	荒木 俊史
			大阪府大阪市中央区玉造1丁目2番28号
			京セラドキュメントソリューションズ株式会社内
		審査官	坂庭 剛史

最終頁に続く

(54) 【発明の名称】 アプリケーション実行時間短縮方法

(57) 【特許請求の範囲】

【請求項1】

基本コンポーネントであるネイティブ関数を有するソフトウェア構造を有する画像形成装置によるアプリケーション実行時間短縮方法であって、

アプリケーションは、前記画像形成装置が提供する機能以外の機能を実現するJava(登録商標)で作成されたアプリケーションであり、

前記ソフトウェア構造は、アプリケーション、アプリケーションフレームワーク、及びネイティブ関数提供機能から構成され、

前記アプリケーションフレームワークは、インターフェイス、ミドルウェア、及びアプリケーションライブラリーから構成され、

前記アプリケーション及び前記インターフェイスは、前記Java(登録商標)に適用しており、前記ミドルウェア、前記アプリケーションライブラリー、及び前記ネイティブ関数提供機能は、ネイティブ関数に適用しており、

前記アプリケーション実行時間短縮方法は、(1)前記アプリケーションが前記ミドルウェアを経由して前記ネイティブ関数を実行する方法と、(2)前記アプリケーションが前記ミドルウェアを経由しないで前記ネイティブ関数を実行する方法を含み、

(1)前記アプリケーションが前記ミドルウェアを経由して前記ネイティブ関数を実行する方法は、

前記アプリケーションにより、前記インターフェイスを呼び出すステップと、

前記インターフェイスにより、前記ミドルウェアを呼び出すステップと、

前記ミドルウェアにより、前記ネイティブ関数提供機能呼び出すステップと、
前記ミドルウェアにより、前記ネイティブ関数提供機能からネイティブ関数の情報を取り出すと、前記ネイティブ関数の情報により前記ミドルウェアが前記ネイティブ関数を実行するステップと、

前記ミドルウェアにより、前記インターフェイスに終了を通知するステップと、
前記インターフェイスにより、前記アプリケーションに終了を通知するステップと、を備え、

前記ミドルウェアは、前記アプリケーションから実行されるときに、前記アプリケーションの認証を行い、

(2) 前記アプリケーションが前記ミドルウェアを経由しないで前記ネイティブ関数を実行する方法は、アプリケーションに前記ミドルウェアを経由せずに前記ネイティブ関数を実行する設定が行われている場合に実施され、

(A) 前記アプリケーションの処理を開始するとき、前記ネイティブ関数の情報を前記ネイティブ関数が実行できるプラットフォームに登録するネイティブ関数登録工程と、

(B) 前記ネイティブ関数の情報により前記ネイティブ関数を前記プラットフォームで実行するネイティブ関数工程と、

(C) 前記アプリケーションの処理を終了するとき、前記ネイティブ関数の情報を前記プラットフォームから削除するネイティブ関数削除工程とを備え、

(A) 前記ネイティブ関数登録工程は、

前記アプリケーションにより、前記アプリケーションが予め保持するデジタル署名、及び実行するネイティブ関数の情報に対する登録要求を、前記アプリケーションフレームワークに出力するステップと、

前記アプリケーションフレームワークにより、前記デジタル署名、及び前記ネイティブ関数の情報の登録要求が入力されると、前記画像形成装置が有する補助記憶部に保存された認証データを取り出し、前記認証データにより前記デジタル署名が認証されたか否かを判定するステップと、

前記デジタル署名が認証されたとき、前記アプリケーションフレームワークにより、前記ネイティブ関数の情報を前記アプリケーションライブラリーのプラットフォームに登録するステップと、

前記アプリケーションフレームワークにより、前記アプリケーションに前記ネイティブ関数の登録処理の終了通知を出力するステップと、

前記デジタル署名が認証されないとき、前記アプリケーションフレームワークにより、前記アプリケーションに実行不可の通知を出力し、前記ネイティブ関数の登録処理を終了するステップと、

前記アプリケーションにより、前記実行不可の通知が入力されると実行を停止するステップと、を備え、

(B) 前記ネイティブ関数工程は、

前記アプリケーションにより、前記ネイティブ関数の登録処理の終了通知が入力されると、前記アプリケーションからハードウェアのインターフェイスに関連するコントローラ機能の呼出要求を前記アプリケーションフレームワークに出力するステップと、

前記アプリケーションフレームワークにより、前記インターフェイスに関連するコントローラ機能呼び出すステップと、

前記アプリケーションにより、ネイティブ関数呼び出すステップと、

前記アプリケーションフレームワークにより、前記アプリケーションが呼び出している前記ネイティブ関数を前記アプリケーションライブラリーのプラットフォームで実行するステップと、

前記アプリケーションフレームワークにより、前記アプリケーションにネイティブ関数処理の終了通知を出力し、前記ネイティブ関数処理を終了するステップと、を備え、

(C) 前記ネイティブ関数削除工程は、

10

20

30

40

50

前記アプリケーションフレームワークにより、前記デジタル署名、及び前記ネイティブ関数の情報の削除要求が入力されると、前記補助記憶部から前記認証データを取り出し、前記認証データによりデジタル署名が認証されたかを判定するステップと、

前記デジタル署名が認証されたとき、前記アプリケーションフレームワークにより、前記アプリケーションライブラリーの前記プラットフォームに登録されている前記ネイティブ関数の情報を削除するステップと、

前記アプリケーションフレームワークにより、前記アプリケーションに前記ネイティブ関数の削除処理の終了通知を出力し、前記ネイティブ関数の削除処理を終了するステップと、

前記デジタル署名が認証されないとき、前記アプリケーションフレームワークにより、前記アプリケーションの実行を停止するステップと、

前記アプリケーションフレームワークにより、前記アプリケーションライブラリーの前記プラットフォームに登録されている前記ネイティブ関数の情報を削除するステップと、を備え

ることを特徴とするアプリケーション実行時間短縮方法。

【請求項 2】

前記ミドルウェアを経由せずに前記ネイティブ関数を実行する設定が行われているアプリケーションは、データのアクセスが頻繁に行われる処理のアプリケーション、または繰り返し計算が行われる処理のアプリケーションを含むことを特徴とする請求項 1 に記載のアプリケーション実行時間短縮方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、デバイスに搭載されるアプリケーションの実行時間を短縮するための方法に関する。

【背景技術】

【0002】

最近の技術によれば、プリンタ、多機能プリンタ、多機能周辺装置、又は複合機などの MFP (Multi function Peripheral) である画像形成装置は、Java (登録商標) の実行環境を備え、Java 言語によって作成されたアプリケーション (以下、「Java アプリケーション」という) を実行できる仕組みが構築されている。Java アプリケーションは、画像形成装置から提供される機能のインターフェイスを使用することで、画像形成装置の機能を利用して様々な処理を実行することができる。これにより画像形成装置では、ユーザーが機能のインターフェイスを使用して、画像形成装置が提供していない機能を実現できる Java アプリケーションを作成し、画像形成装置に搭載することができる。このような Java アプリケーションの搭載が可能な画像形成装置などのデバイスでは、アプリケーションの実行が容易となるような対策も行われている。例えば、特許文献 1 のアプリケーションが実行可能な情報処理装置では、ユーザーがアプリケーション毎に異なる入力操作を行うことなく、アプリケーションの機能を容易に呼び出すことを可能にしている。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2005 - 293407 号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

しかし、特許文献 1 の情報処理装置では、アプリケーションの機能を容易に呼び出すことはできるが、アプリケーションの処理時間を速くすることができない。また、画像形成

10

20

30

40

50

装置などのデバイスが提供する機能のインターフェイスを使用してアプリケーションを作成することは容易になったが、デバイスが提供する機能のインターフェイスを使用して作成されたアプリケーションを実行すると、処理に時間がかかるという問題があった。

【 0 0 0 5 】

本発明はこのような状況に鑑みてなされたものであり、上記課題を解決できるアプリケーション処理時間短縮方法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】

【 0 0 0 6 】

本発明のアプリケーション実行時間短縮方法は、基本コンポーネントであるネイティブ関数を有するソフトウェア構造を有する画像形成装置によるアプリケーション実行時間短縮方法であって、アプリケーションは、前記画像形成装置が提供する機能以外の機能を実現する J a v a (登録商標) で作成されたアプリケーションであり、前記ソフトウェア構造は、アプリケーション、アプリケーションフレームワーク、及びネイティブ関数提供機能から構成され、前記アプリケーションフレームワークは、インターフェイス、ミドルウェア、及びアプリケーションライブラリーから構成され、前記アプリケーション及び前記インターフェイスは、前記 J a v a (登録商標) に適用しており、前記ミドルウェア、前記アプリケーションライブラリー、及び前記ネイティブ関数提供機能は、ネイティブ関数に適用しており、前記アプリケーション実行時間短縮方法は、(1) 前記アプリケーションが前記ミドルウェアを経由して前記ネイティブ関数を実行する方法と、(2) 前記アプリケーションが前記ミドルウェアを経由しないで前記ネイティブ関数を実行する方法を含み、(1) 前記アプリケーションが前記ミドルウェアを経由して前記ネイティブ関数を実行する方法は、前記アプリケーションにより、前記インターフェイスを呼び出すステップと、前記インターフェイスにより、前記ミドルウェアを呼び出すステップと、前記ミドルウェアにより、前記ネイティブ関数提供機能を呼び出すステップと、前記ミドルウェアにより、前記ネイティブ関数提供機能からネイティブ関数の情報を取り出すと、前記ネイティブ関数の情報により前記ミドルウェアが前記ネイティブ関数を実行するステップと、前記ミドルウェアにより、前記インターフェイスに終了を通知するステップと、前記インターフェイスにより、前記アプリケーションに終了を通知するステップと、を備え、前記ミドルウェアは、前記アプリケーションから実行されるときに、前記アプリケーションの認証を行い、(2) 前記アプリケーションが前記ミドルウェアを経由しないで前記ネイティブ関数を実行する方法は、アプリケーションに前記ミドルウェアを経由せずに前記ネイティブ関数を実行する設定が行われている場合に実施され、(A) 前記アプリケーションの処理を開始するとき、前記ネイティブ関数の情報を前記ネイティブ関数が実行できるプラットフォームに登録するネイティブ関数登録工程と、(B) 前記ネイティブ関数の情報により前記ネイティブ関数を前記プラットフォームで実行するネイティブ関数工程と、(C) 前記アプリケーションの処理を終了するとき、前記ネイティブ関数の情報を前記プラットフォームから削除するネイティブ関数削除工程とを備え、(A) 前記ネイティブ関数登録工程は、前記アプリケーションにより、前記アプリケーションが予め保持するデジタル署名、及び実行するネイティブ関数の情報に対する登録要求を、前記アプリケーションフレームワークに出力するステップと、前記アプリケーションフレームワークにより、前記デジタル署名、及び前記ネイティブ関数の情報の登録要求が入力されると、前記画像形成装置が有する補助記憶部に保存された認証データを取り出し、前記認証データにより前記デジタル署名が認証されたか否かを判定するステップと、前記デジタル署名が認証されたとき、前記アプリケーションフレームワークにより、前記ネイティブ関数の情報を前記アプリケーションライブラリーのプラットフォームに登録するステップと、前記アプリケーションフレームワークにより、前記アプリケーションに前記ネイティブ関数の登録処理の終了通知を出力するステップと、前記デジタル署名が認証されないとき、前記アプリケーションフレームワークにより、前記アプリケーションに実行不可の通知を出力し、前記ネイティブ関数の登録処理を終了するステップと、前記アプリケーションにより、前記実行不可の通知が入力されると実行を停止するステップと、を備え、(B) 前記ネイティブ関数工

10

20

30

40

50

程は、前記アプリケーションにより、前記ネイティブ関数の登録処理の終了通知が入力されると、前記アプリケーションからハードウェアのインターフェイスに関連するコントローラ機能の呼出要求を前記アプリケーションフレームワークに出力するステップと、前記アプリケーションフレームワークにより、前記インターフェイスに関連するコントローラ機能呼び出すステップと、前記アプリケーションにより、ネイティブ関数を呼び出すステップと、前記アプリケーションフレームワークにより、前記アプリケーションが呼び出している前記ネイティブ関数を前記アプリケーションライブラリの前記プラットフォームで実行するステップと、前記アプリケーションフレームワークにより、前記アプリケーションにネイティブ関数処理の終了通知を出力し、前記ネイティブ関数処理を終了するステップと、を備え、(C)前記ネイティブ関数削除工程は、前記アプリケーションフレームワークにより、前記デジタル署名、及び前記ネイティブ関数の情報の削除要求が入力されると、前記補助記憶部から前記認証データを取り出し、前記認証データによりデジタル署名が認証されたかを判定するステップと、前記デジタル署名が認証されたとき、前記アプリケーションフレームワークにより、前記アプリケーションライブラリの前記プラットフォームに登録されている前記ネイティブ関数の情報を削除するステップと、前記アプリケーションフレームワークにより、前記アプリケーションに前記ネイティブ関数の削除処理の終了通知を出力し、前記ネイティブ関数の削除処理を終了するステップと、前記デジタル署名が認証されないとき、前記アプリケーションフレームワークにより、前記アプリケーションの実行を停止するステップと、前記アプリケーションフレームワークにより、前記アプリケーションライブラリの前記プラットフォームに登録されている前記ネイティブ関数の情報を削除するステップと、を備えることを特徴としている。

10

20

また、本発明のアプリケーション実行時間短縮方法における前記ミドルウェアを経由せずに前記ネイティブ関数を実行する設定が行われているアプリケーションは、データのアクセスが頻繁に行われる処理のアプリケーション、または繰り返し計算が行われる処理のアプリケーションを含むことを特徴としている。

【発明の効果】

【0007】

本発明は、デバイスが提供する機能を使用して作成されるアプリケーションにおいて、アプリケーションの実行時間を短縮するアプリケーション実行時間短縮方法を提供する。

【図面の簡単な説明】

30

【0008】

【図1】本発明の実施形態に係る画像形成装置の機能構成を示す図である。

【図2】本発明の実施形態に係る画像形成装置におけるアプリケーションを実行するためのソフトウェア構造を示す図である。

【図3】本発明の実施形態に係る画像形成装置におけるネイティブ関数登録処理のフローチャートである。

【図4】本発明の実施形態に係る画像形成装置におけるネイティブ関数処理のフローチャートである。

【図5】本発明の実施形態に係る画像形成装置におけるネイティブ関数削除処理のフローチャートである。

40

【発明を実施するための形態】

【0009】

以下、本発明を実施するための形態(以下、「実施形態」という)を、図面を参照して説明する。本発明は、画像形成装置から提供される機能であるOS(Operating System)とアプリケーションの中間に位置するソフトウェア(以下、「ミドルウェア」という)を毎回使用せずに、画像形成装置におけるソフトウェアの基本コンポーネント(以下、「ネイティブ関数」という)を直接使用することで、Javaアプリケーション(以降からは単に「アプリケーション」という)の実行時間を短縮するものである。

【0010】

実施形態における画像形成装置100の機能構成について、図1を用いて説明する。図

50

1に示す画像形成装置100は、制御部110、補助記憶部120、記憶部130、操作パネル140、操作パネル処理部150、原稿読取部160、画像形成部170、ネットワーク通信部180、及び画像印刷部190を備えている。これらの各部は、バスなどにより接続される構成となっている。制御部110には、アプリケーション実行処理部110aが設けられている。補助記憶部120には、アプリケーション保存エリア120a、及び認証データ保存エリア120bが設けられている。

【0011】

制御部110は、RAMやROM等の主記憶手段、及びMPU(Micro Processing Unit)やCPU(Central Processing Unit)等の制御手段を備えている。また、制御部110は、各種I/O、USB(ユニバーサル・シリアル・バス)、バス、その他ハードウェア等のインターフェイスに関連するコントローラ機能を備え、画像形成装置100全体を制御する。

10

アプリケーション実行処理部110aは、ユーザーが操作パネル140またはPC(パーソナルコンピュータ)200から画像形成装置100に搭載されているアプリケーション510を指定すると、指定されたアプリケーションを実行する。

【0012】

補助記憶部120は、フラッシュメモリー等からなる補助記憶装置で、制御部110が実行する処理のプログラムやデータを記憶する。

アプリケーション保存エリア120aは、画像形成装置100が提供するプログラム以外のユーザーが作成するアプリケーションのプログラム及びデータを保存する。

20

認証データ保存エリア120bは、ユーザーが作成するアプリケーションを認証するためのデータを保存する。

【0013】

記憶部130は、ハードディスクドライブからなる記憶装置で、制御部110が実行する処理のプログラムやデータが記憶される。

原稿画像データ保存エリア130aは、原稿読取部160から読み取った原稿画像データと、ネットワーク通信部180が受信した原稿画像データなどを一時的に保存する。

【0014】

操作パネル140は、操作画面の表示とユーザーによる操作を受け付ける。操作パネル処理部150は、ユーザーにより選択される操作項目を操作パネル140に表示する処理、操作パネル140からユーザーの操作を入力する処理を行う。

30

【0015】

原稿読取部160は、画像形成装置100の原稿台にセットされた原稿の読み取りを行うもので、ユーザーが操作パネル140から読み取り要求を行うことにより、原稿台にセットされた原稿の読み取りを行うことができる。

【0016】

画像形成部170は、原稿読取部160で読み取られた原稿画像データを印刷可能なイメージやFAXやメールで送信可能なイメージの画像データに変換する。

【0017】

ネットワーク通信部180は、ネットワーク300に接続するための着脱可能なLANインターフェイスを備えている。また、画像形成装置100は、ネットワーク300に接続されているPC200などの装置とネットワーク通信部180により通信を行うことができる。

40

【0018】

画像印刷部190は、ユーザーから印刷要求のあった画像データを用紙に印刷する機能を備えている。

【0019】

次に、本発明の実施形態におけるアプリケーションを実行するためのソフトウェア構造について、図2を用いて説明する。図2に示すように、アプリケーションを実行するためのソフトウェア構造は、アプリケーション510、アプリケーションフレームワーク52

50

0、及びネイティブ関数提供機能530から構成されている。アプリケーションフレームワーク520は、インターフェイス520a、ミドルウェア520b、及びアプリケーションライブラリー520cから構成されている。このようなソフトウェア構造において、アプリケーション510及びインターフェイス520aは、Java（登録商標）に適用しているのでJava層に含まれる部分となる。また、ミドルウェア520b、アプリケーションライブラリー520c、及びネイティブ関数提供機能530は、ネイティブ関数に適用しているのでネイティブ層に含まれる部分となる。

【0020】

アプリケーション510は、画像形成装置100が提供する機能以外で、ユーザーが作成するプログラムである。

アプリケーションフレームワーク520は、アプリケーション510が制御部110により実行されるときにアプリケーション510が使用するソフトウェア、プログラム、及びライブラリーである。

インターフェイス520aは、アプリケーション510とミドルウェア520bとを接続するプログラムである。アプリケーション510は、インターフェイス520aを經由してミドルウェア520bを実行する。

ミドルウェア520bは、OSを直接動作させ、アプリケーションを実行するための機能を提供する。ミドルウェア520bは、アプリケーション510から実行されるときに、アプリケーション510の認証を行う。

アプリケーションライブラリー520cは、アプリケーションがミドルウェア520bを經由せずに、ネイティブ関数を実行するとき、アプリケーションが使用するライブラリーである。アプリケーションライブラリー520cには、ネイティブ関数を実行できる環境（以下、「プラットフォーム」という）が設けられ、プラットフォームにネイティブ関数を実行するためのネイティブ関数の情報が登録され、このネイティブ関数の情報によりネイティブ関数を実行することができる。

ネイティブ関数提供機能530は、ミドルウェア520bが実行するネイティブ関数の情報を提供する。ミドルウェア520bは、ネイティブ関数の情報によりネイティブ関数を実行することができる。

【0021】

このようなアプリケーションを実行するための図2に示すソフトウェア構造において、まず、矢印(1)に示すようにアプリケーション510からインターフェイス520aを呼び出す。次に、矢印(2)に示すようにインターフェイス520aからミドルウェア520bを呼び出す。次に、矢印(3)に示すようにミドルウェア520bからネイティブ関数提供機能530を呼び出し、矢印(4)に示すようにミドルウェア520bがネイティブ関数提供機能530からネイティブ関数の情報を取り出すと、ネイティブ関数の情報によりミドルウェア520bがネイティブ関数を実行する。次に、矢印(5)に示すようにミドルウェア520bからインターフェイス520aに終了を通知する。次に、矢印(6)に示すようにインターフェイス520aからアプリケーション510に終了を通知する。以上のように、通常は矢印(1)から矢印(6)を繰り返すことで、アプリケーションが実行されるが、アプリケーション510がアプリケーションライブラリー520cにネイティブ関数の実行を指示するときには、矢印(7)に示すようにアプリケーションライブラリー520cに登録されているネイティブ関数の情報により、アプリケーションライブラリー520cがネイティブ関数を実行する。アプリケーション510がミドルウェア520bを經由せずにネイティブ関数を実行する処理の詳細については、後述する。

【0022】

次に、本発明の実施形態に係るアプリケーション実行処理について、図2及び図3を用いて説明する。ユーザーが操作パネル140またはPC200から画像形成装置100に搭載されているアプリケーション510を指定して起動する操作を行うと、制御部110に通知され、制御部110がアプリケーション実行処理部110aを起動する。起動されたアプリケーション実行処理部110aはアプリケーション保存エリア120aから指定

10

20

30

40

50

されたアプリケーション 5 1 0 のプログラムを取り出し、アプリケーション 5 1 0 を開始する。

また、アプリケーション 5 1 0 にミドルウェア 5 2 0 b を経由せずにネイティブ関数を実行する設定が行われていると、アプリケーション 5 1 0 は、処理を開始するときにネイティブ関数登録処理を実行し、次にネイティブ関数処理を実行し、処理を終了するときにネイティブ関数削除処理を実行する。

【 0 0 2 3 】

まず、ネイティブ関数登録処理について、図 3 に示すネイティブ関数登録処理のステップ順に説明する。

【 0 0 2 4 】

(ステップ S 1 1 0)

まず、アプリケーション 5 1 0 は、デジタル署名、実行するネイティブ関数の情報に対する登録要求をアプリケーションフレームワーク 5 2 0 へ出力する。デジタル署名は、ミドルウェア 5 2 0 b を経由せずにネイティブ関数を実行する設定が行われているアプリケーション 5 1 0 が予め保持している。

【 0 0 2 5 】

(ステップ S 1 2 0)

次いで、アプリケーションフレームワーク 5 2 0 は、デジタル署名、ネイティブ関数の情報の登録要求を入力すると、認証データ保存エリア 1 2 0 b から認証データを取り出し、認証データによりデジタル署名の確認を行う。

【 0 0 2 6 】

(ステップ S 1 3 0)

次いで、アプリケーションフレームワーク 5 2 0 は、デジタル署名が認証されたかを判定する。デジタル署名が認証されたとき(ステップ S 1 3 0 の Y e s)は、ステップ S 1 4 0 へ進む。デジタル署名が認証されないとき(ステップ S 1 3 0 の N o)は、ステップ S 1 5 0 へ進む。

【 0 0 2 7 】

(ステップ S 1 4 0)

ステップ S 1 3 0 の Y e s において、アプリケーションフレームワーク 5 2 0 は、ネイティブ関数の情報をアプリケーションライブラリー 5 2 0 c のプラットフォームに登録し、アプリケーション 5 1 0 にネイティブ関数登録処理の終了通知を出力すると、ネイティブ関数登録処理を終了する。

【 0 0 2 8 】

(ステップ S 1 5 0)

ステップ S 1 3 0 の N o において、アプリケーションフレームワーク 5 2 0 は、アプリケーション 5 1 0 に実行不可の通知を出力し、ネイティブ関数登録処理を終了する。アプリケーション 5 1 0 は、実行不可の通知を入力すると実行を停止する。

【 0 0 2 9 】

次に、ネイティブ関数処理について、図 4 に示すネイティブ関数処理のステップ順に説明する。

【 0 0 3 0 】

(ステップ S 2 1 0)

まず、アプリケーション 5 1 0 は、ネイティブ関数登録処理の終了通知を入力すると、アプリケーション 5 1 0 からハードウェアのインターフェイスに関連するコントローラ機能の呼出要求をアプリケーションフレームワーク 5 2 0 へ出力する。

【 0 0 3 1 】

(ステップ S 2 2 0)

次いで、アプリケーションフレームワーク 5 2 0 は、インターフェイスに関連するコントローラ機能呼び出す。

【 0 0 3 2 】

10

20

30

40

50

(ステップS230)

次いで、アプリケーション510は、ネイティブ関数を呼び出す。

【0033】

(ステップS240)

次いで、アプリケーションフレームワーク520は、アプリケーション510が呼び出しているネイティブ関数をアプリケーションライブラリー520cのプラットフォームで実行し、アプリケーション510にネイティブ関数処理の終了通知を出力すると、ネイティブ関数処理を終了する。

【0034】

次に、ネイティブ関数削除処理について、図5に示すネイティブ関数削除処理のステップ順に説明する。

10

【0035】

(ステップS310)

まず、アプリケーション510は、ネイティブ関数処理の終了通知を入力すると、デジタル署名、呼び出すネイティブ関数の情報に対する削除要求をアプリケーションフレームワーク520に出力する。

【0036】

(ステップS320)

次いで、アプリケーションフレームワーク520は、デジタル署名、ネイティブ関数の情報の削除要求を入力すると、認証データ保存エリア120bから認証データを取り出し、認証データによりデジタル署名の確認を行う。

20

【0037】

(ステップS330)

次いで、アプリケーションフレームワーク520は、デジタル署名が認証されたかを判定する。デジタル署名が認証されたとき(ステップS330のYes)は、ステップS340に進む。デジタル署名が認証されないとき(ステップS330のNo)は、ステップS350に進む。

【0038】

(ステップS340)

ステップS330のYesにおいて、アプリケーションフレームワーク520は、アプリケーションライブラリー520cのプラットフォームに登録されているネイティブ関数の情報を削除し、アプリケーション510にネイティブ関数削除処理の終了通知を出力すると、ネイティブ関数削除処理を終了する。

30

【0039】

(ステップS350)

ステップS330のNoにおいて、アプリケーションフレームワーク520は、アプリケーション510の実行を停止し、アプリケーションライブラリー520cのプラットフォームに登録されているネイティブ関数の情報を削除する。

【0040】

以上により、画像形成装置100に搭載されているアプリケーション510において、ミドルウェア520bを経由せずにネイティブ関数を実行することで、ミドルウェア520bによる処理の実行時間が削除される。これにより、アプリケーション510の実行時間を短縮することができる。また、アプリケーション510によりネイティブ関数が実行されるときには、アプリケーションフレームワーク520は、アプリケーション510が保持しているデジタル署名の確認を行い、認証が行われたときにアプリケーション510に対してネイティブ関数の実行を許可する。これにより、許可されていないアプリケーション510がミドルウェア520bを経由せずにネイティブ関数を実行することで、不適切な処理が行われることによるセキュリティの脅威を、ミドルウェア520bと同様に防止することができる。従って、データのアクセスが頻繁に行われる処理、または繰り返し計算が行われる処理などのアプリケーション510では、処理時間が従来に比べてとても

40

50

短くなるので、ユーザーの待ち時間に対するストレスを軽減できる。また、本発明のアプリケーション 510 はセキュリティが確保され、このような複数のアプリケーション 510 を画像形成装置 100 に搭載することで、ユーザーの要求に対処する付加価値の高い画像形成装置 100 を提供できる。

【0041】

なお、実施形態においては、画像形成装置 100 に搭載するアプリケーション 510 の例で説明したが、ソフトウェアを備え、ソフトウェアにより機能が実行されるデバイス一般に適用可能である。

また、実施形態では、Javaアプリケーションの例で説明したが、Javaアプリケーションに限定されず、デバイスにおいて実行環境が整えられている言語によって作成されたアプリケーションであれば、同様に適用することができる。

10

【0042】

このような本発明のアプリケーション実行時間短縮方法は、デバイスが提供する機能を使用して作成されるアプリケーションにおいて、アプリケーションの実行時間を短縮できる。

【0043】

以上、具体的な実施の形態により本発明を説明したが、上記実施の形態は本発明の例示であり、この実施の形態に限定されないことは言うまでもない。

【産業上の利用可能性】

【0044】

本発明は、アプリケーションの搭載が可能な画像形成装置などのデバイスに好適であるが、デバイスに限られるものではなく、ネイティブ関数を有しているソフトウェアが搭載されるシステムにも適用できる。

20

【符号の説明】

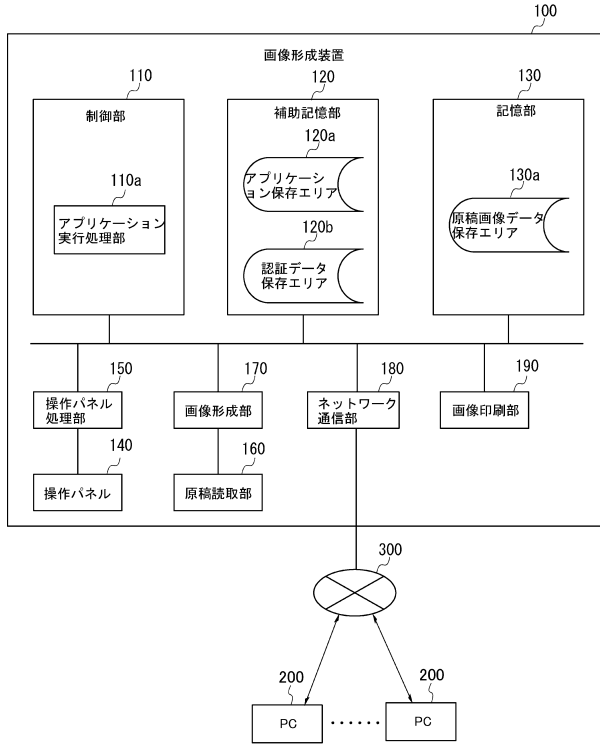
【0045】

100・・・画像形成装置
 110・・・制御部
 110a・・・アプリケーション実行処理部
 120・・・補助記憶部
 120a・・・アプリケーション保存エリア
 120b・・・認証データ保存エリア
 130・・・記憶部
 130a・・・原稿画像データ保存エリア
 140・・・操作パネル
 150・・・操作パネル処理部
 160・・・原稿読取部
 170・・・画像形成部
 180・・・ネットワーク通信部
 190・・・画像印刷部
 200・・・PC
 300・・・ネットワーク
 510・・・アプリケーション
 520・・・アプリケーションフレームワーク
 520a・・・インターフェイス
 520b・・・ミドルウェア
 520c・・・アプリケーションライブラリー
 530・・・ネイティブ関数提供機能

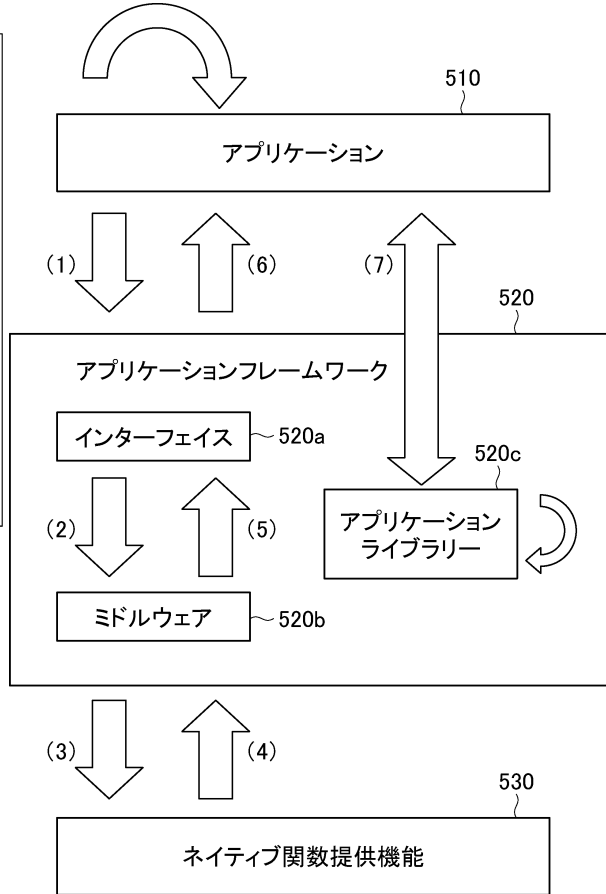
30

40

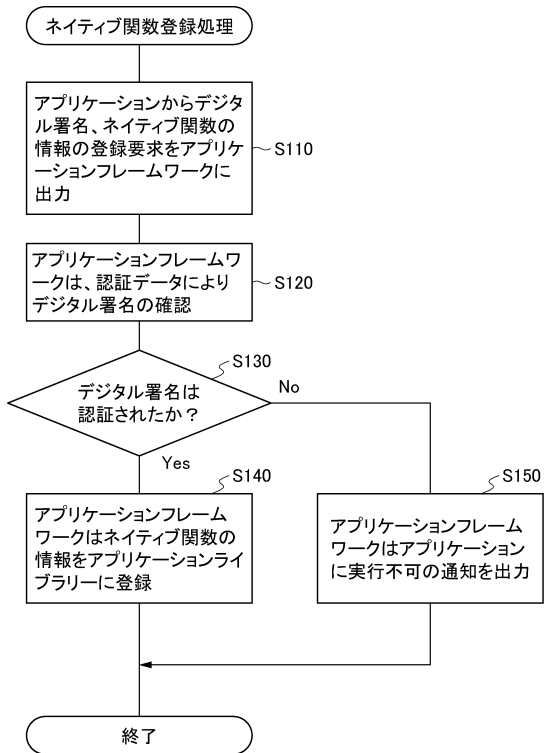
【図1】



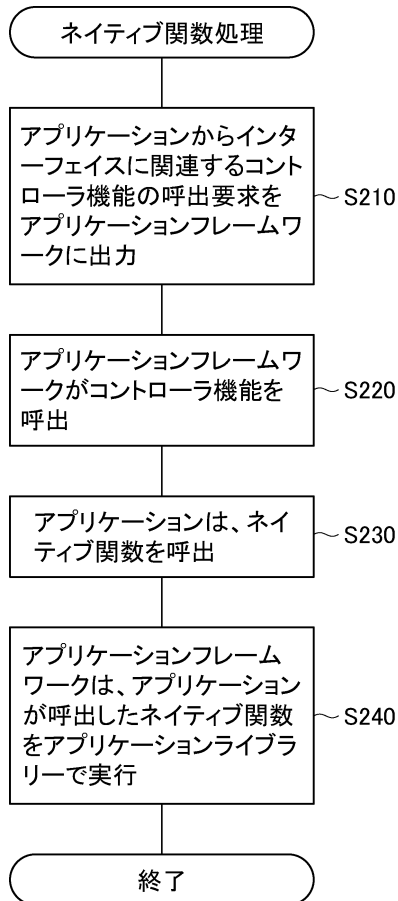
【図2】



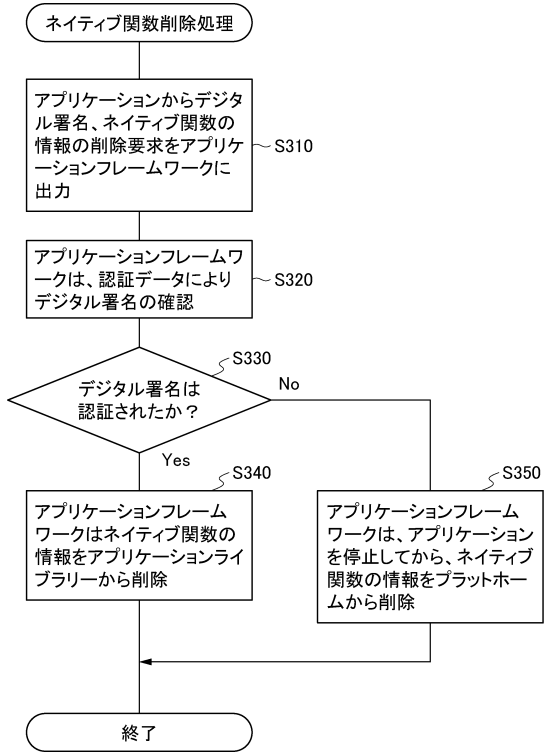
【図3】



【図4】



【図5】



フロントページの続き

(56)参考文献 特開2009-294773(JP,A)

特開2007-213349(JP,A)

特開2013-097745(JP,A)

特開2010-009149(JP,A)

特表2009-528596(JP,A)

特開2009-110357(JP,A)

特開2008-191901(JP,A)

特開2003-076558(JP,A)

大関江利子、浅井真生、戸部章子、土屋二郎，“2008年秋冬モデル搭載アプリケーション機能(2)次世代JAVAアプリ”，NTT DOCOMOテクニカル・ジャーナル，社団法人電気通信協会，2009年 1月 1日，Vol.16，No.4，pp.52-53

出村成和，“よくわかるAndroid NDKのディープな話：最終回 JavaからC/C++ライブラリを呼び出すJNI”，日経Linux，日本，日経BP社，2013年 3月 8日，第15巻，第4号(通巻163号)，p.124，ISSN 1345-0182

(58)調査した分野(Int.Cl.，DB名)

G06F 9/54

G06F 9/44

G06F 21/51