(19)日本国特許庁(JP)

# (12)特許公報(B2)

(11)特許番号
特許第7084826号
(P7084826)

(45)発行日　令和4年6月15日(2022.6.15)

(24)登録日　令和4年6月7日(2022.6.7)

(51)国際特許分類
*Ｇ０６Ｆ　21/12　(2013.01)*
*Ｇ０６Ｆ　21/56　(2013.01)*

ＦＩ
Ｇ０６Ｆ　21/12
Ｇ０６Ｆ　21/56　３５０

請求項の数　10　（全18頁）

(21)出願番号　特願2018-159476(P2018-159476)
(22)出願日　平成30年8月28日(2018.8.28)
(65)公開番号　特開2020-35078(P2020-35078A)
(43)公開日　令和2年3月5日(2020.3.5)
審査請求日　令和3年8月24日(2021.8.24)

(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　110003281
特許業務法人大塚国際特許事務所
(72)発明者　安川 朱里
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　平井 誠

最終頁に続く

(54)【発明の名称】　情報処理装置、その制御方法、およびそのプログラム

(57)【特許請求の範囲】
【請求項１】
アプリケーションプログラムの検証を行う検証手段と、
前記検証手段による検証が失敗した場合には、前記アプリケーションプログラムの種別に基づき、前記アプリケーションプログラムの復旧を行うか否かを判定し、
復旧を行うと判定した場合には前記アプリケーションプログラムを復旧し、
復旧を行わないと判定した場合には前記アプリケーションプログラムの実行を許さず、
前記検証手段による検証が成功したアプリケーションプログラム又は前記復旧したアプリケーションプログラムの実行を許す制御手段と
を有することを特徴とする情報処理装置。
【請求項２】
請求項１に記載の情報処理装置であって、
前記検証手段は、前記アプリケーションプログラムのハッシュ値を計算し、
前記制御手段は、計算した前記ハッシュ値と予め記憶したハッシュ値とが一致した場合に検証が成功したと判定することを特徴とする情報処理装置。
【請求項３】
請求項２に記載の情報処理装置であって、
前記制御手段は、前記アプリケーションプログラムが実行可能な形式で再インストール可能であれば、前記アプリケーションプログラムの復旧を行うと判定し、前記アプリケーションプログラムを再インストールし、前記アプリケーションプログラムのハッシュ値を計

算し、計算した前記ハッシュ値により、前記予め記憶したハッシュ値を更新する
ことを特徴とする情報処理装置。
【請求項４】
請求項１乃至３のいずれか一項に記載の情報処理装置であって、
基本入出力システムに含まれたローダを初段とし、前記アプリケーションプログラムをロードするためのローダを最終段として、前段のローダにより次段のローダを検証し、ローディング対象のプログラムをロードする複数段のローダ手段と、
前記基本入出力システムを検証する、改ざんできない第２の検証手段とを更に有し、
前記第２の検証手段により前記基本入出力システムを検証し、前記複数段のローダのそれぞれによって読み込まれたプログラム及び次段のローダを検証する
ことを特徴とする情報処理装置。
【請求項５】
請求項４に記載の情報処理装置であって、
前記第２の検証手段はハードウェアにより前記基本入出力システムを検証する
ことを特徴とする情報処理装置。
【請求項６】
請求項１乃至５のいずれか一項に記載の情報処理装置であって、
ユーザインターフェイス手段を更に有し、
前記制御手段により前記アプリケーションプログラムの実行を許さない場合には、前記ユーザインターフェイス手段により前記アプリケーションプログラムの実行が許されていないことを表示する
ことを特徴とする情報処理装置。
【請求項７】
請求項６に記載の情報処理装置であって、
実行を禁止される前記アプリケーションプログラムがログインアプリケーションの場合には、前記情報処理装置による機能の提供を停止する
ことを特徴とする情報処理装置。
【請求項８】
請求項１乃至７のいずれか一項に記載の情報処理装置であって、
プリンタ手段と、
スキャナ手段と
を更に有することを特徴とする情報処理装置。
【請求項９】
アプリケーションプログラムの検証を行い、
前記検証が失敗した場合には、前記アプリケーションプログラムの種別に基づき、前記アプリケーションプログラムの復旧を行うか否かを判定し、
復旧を行うと判定した場合には前記アプリケーションプログラムを復旧し、
復旧を行わないと判定した場合には前記アプリケーションプログラムの実行を許さず、
前記検証が成功したアプリケーションプログラム又は前記復旧したアプリケーションプログラムの実行を許す
ことを特徴とする情報処理装置の制御方法。
【請求項１０】
アプリケーションプログラムの検証を行う検証手段と、
前記検証手段による検証が失敗した場合には、前記アプリケーションプログラムの種別に基づき、前記アプリケーションプログラムの復旧を行うか否かを判定し、
復旧を行うと判定した場合には前記アプリケーションプログラムを復旧し、
復旧を行わないと判定した場合には前記アプリケーションプログラムの実行を許さず、
前記検証手段による検証が成功したアプリケーションプログラム又は前記復旧したアプリケーションプログラムの実行を許す制御手段と
してコンピュータを機能させるためのプログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、例えばソフトウェア改ざん検知機能等を有する情報処理装置、その制御方法、およびそのプログラムに関する。
【背景技術】
【０００２】
機器を制御するソフトウェアを第三者が改ざんし、不正利用する攻撃が問題になっている。不正利用されてしまうと、情報資産を盗まれたり、他のシステムを攻撃する踏み台に利用されたりすることで、機器の所有者に甚大な被害を引き起こす危険性がある。そういった攻撃に対し、ソフトウェアが改ざんされていないことを機器利用時に検証する手段が考案されている（特許文献１）。
【０００３】
また、特許文献２では複合機に接続された拡張ユニットを起動する際に、拡張ユニットのソフトウェアの正当性を確認し、正当性が確認された後に拡張ユニットを起動する手段が考案されている。正当性確認に失敗した場合は、拡張ユニットの利用を停止し、不正なソフトウェアが動作しないように制御する。
【先行技術文献】
【特許文献】
【０００４】
【文献】特開２０１７－１５３０４４号公報
特開２００８－１７１０４１号公報
【発明の概要】
【発明が解決しようとする課題】
【０００５】
特許文献２では、ソフトウェアの正当性確認に失敗した場合、拡張ユニットの利用を一律停止するため、拡張ユニットが利用できずシステムの可用性が低下するという問題がある。
【０００６】
本発明は上記従来例に鑑みてなされたもので、システムの可用性を担保しつつ、機器をセキュアに利用させることを目的とする。
【課題を解決するための手段】
【０００７】
上記目的を達成するために本発明は以下の構成を有する。
【０００８】
本発明の一側面によれば、アプリケーションプログラムの検証を行う検証手段と、
前記検証手段による検証が失敗した場合には、前記アプリケーションプログラムの種別に基づき、前記アプリケーションプログラムの復旧を行うか否かを判定し、
復旧を行うと判定した場合には前記アプリケーションプログラムを復旧し、
復旧を行わないと判定した場合には前記アプリケーションプログラムの実行を許さず、
前記検証手段による検証が成功したアプリケーションプログラム又は前記復旧したアプリケーションプログラムの実行を許す制御手段とを有することを特徴とする情報処理装置が提供される。
【発明の効果】
【０００９】
本発明によれば、システムの可用性を担保しつつ、ユーザは機器をセキュアに利用することができる。
【図面の簡単な説明】
【００１０】
【図１】本発明に係わるMFPとクライアントPCの接続形態を示すブロック構成図である。
【図２】MFPのコントローラ部の内部構成図である。

【図３】MFPのコントローラ内で実行されるソフトウェアのブロック構成図である。
【図４】起動時検証用正解値リストの例を示す図である。
【図５】アプリケーションの構成図である。
【図６】アプリケーションの管理ファイル及び正解値リストの例を示す図である。
【図７】本発明のMFP側の処理を実施するフロー図である。
【図８Ａ】本発明のMFP側の処理を実施するフロー図である。
【図８Ｂ】本発明のMFP側の処理を実施するフロー図である。
【図９】設定に係る画面構成図である。
【図１０】設定に係る画面構成図である。
【図１１】本発明のMFP側の処理を実施するフロー図である。
【図１２】設定に係る画面構成図である。
【発明を実施するための形態】
【００１１】
［実施形態１］
以下、本発明の実施の形態を図面に基づいて解説する。本実施形態では、ソフトウェア起動時のソフトウェア検証処理について説明する。ここではMFP（Multi-Function Peripheral：複合機）を例に実施形態を説明するが、本発明は複合機以外の任意の情報処理装置に適用可能な技術である。
【００１２】
図１は本発明に係るMFPとクライアントPCの接続形態を示すブロック図である。MFP１００とクライアントPC１２０はLAN１５０を介して接続されている。MFP１００はユーザとの入出力を行う操作部１０２を有する。操作部１０２はユーザインターフェイスを提供する。MFP１００は電子データを紙媒体に出力するプリンタ部１０３を有する。MFP１００は紙媒体を読み込み電子データに変換するスキャナ部１０４を有する。操作部１０２とプリンタ部１０３とスキャナ部１０４はコントローラ部１０１に接続され、コントローラ部１０１の制御に従い複合機としての機能を実現する。クライアントPC１２０はMFP１００に対してプリントジョブの送信といった処理を行う。
【００１３】
●コントローラ部の構成
図２はMFP１００のコントローラ部１０１の詳細を示すブロック図である。CPU２０１はコントローラ内の主な演算処理を行うプロセッサーである。CPU２０１はバスを介してDRAM２０２と接続される。DRAM２０２はCPU２０１が演算する過程で演算命令を表すプログラムデータや、処理対象のデータを一時的に配置するための作業メモリとしてCPU２０１によって使用される。CPU２０１はバスを介してI/Oコントローラ２０３と接続される。I/Oコントローラ２０３はCPU２０１の指示に従い各種デバイスに対する入出力を行う。I/Oコントローラ２０３にはSATA(Serial Advanced Technology Attachment)I/F２０５が接続され、その先にFlash（登録商標）ROM２１１が接続される。CPU２０１はFlash ROM２１１を、MFPの機能を実現するためのプログラム、およびドキュメントファイルを永続的に記憶するために使用する。Flash ROM２１１に格納されるプログラムには、追加的にインストールされたアプリケーションプログラム（アプリケーション）が含まれる。I/Oコントローラ２０３にはネットワークI/F２０４が接続され。ネットワークI/F２０４の先には、有線LANデバイス２１０が接続される。
【００１４】
CPU２０１はネットワークI/F２０４を介して有線LANデバイス２１０を制御することで、LAN１５０上の通信を実現する。I/Oコントローラ２０３にはパネルI/F２０６が接続され、CPU２０１はパネルI/F２０６を介して操作部１０２に対するユーザ向けの入出力を実現する。I/Oコントローラ２０３にはプリンタI/F２０７が接続され、CPU２０１はプリンタI/F２０７を介してプリンタ部１０３を利用した紙媒体の出力処理を実現する。I/Oコントローラ２０３にはスキャナI/F２０８が接続され、CPU２０１はスキャナI/F２０８を介してスキャナ部１０４を利用した原稿の読み込み処理を実現する。I/Oコントローラ２

０３にはUSB I/F２０９が接続され、USB I/Fに接続された任意の機器の制御を行う。ROM２２０はCPU２０１とバスで接続されていて、BIOS(Basic Input Output System：基本入出力システム)を実現する制御プログラムが記憶している。

【００１５】

BIOS検証ユニット２２１はROM２２０およびCPU２０１とバスで接続されていて、ROM２２０に記憶されたBIOSデータの検証と、CPUへのBIOS起動指示を行う。ここで、BIOS検証ユニット２２１はハードウェアであり、BIOS検証がハードウェア検証であることを確認しておく。BIOS検証ユニット２２１とCPU２０１を繋ぐバスは悪意のある第三者に細工をされないために、同一チップ、またはそれに準ずる構成で実現され外部から物理的に確認できない形態になっている。本実施形態では、BIOS検証ユニット２２１の制御機構は集積回路としてハードウェアで実現されている構成を想定するが、専用のCPU、制御ソフトを記憶したROMといった要素を同一チップ内に実装し、製造後に変更できない構成であっても良い。BIOS検証ユニット２２１はCPU２０１にバスで接続されているものとしたが、I/Oコントローラ２０３を介して接続されていてもよい。

【００１６】

コピー機能を実施する場合は、CPU２０１がSATA I/F２０５を介してFlash ROM２１１からコピー機能のためのプログラムとデータをDRAM２０２に読み込む。CPU２０１がDRAM２０２に読み込まれたプログラムを実行してパネルI/F２０６を介して操作部１０２に対するユーザからのコピー指示を検出する。CPU２０１はコピー指示を検出するとスキャナI/F２０８を介してスキャナ部１０４から原稿を電子データとして受け取りDRAM２０２に格納する。CPU２０１はDRAM２０２に格納した画像データに対して出力に適した色変換処理などを実施する。CPU２０１はDRAM２０２に格納した画像データをプリンタI/F２０７を介してプリンタ部１０３に転送し、紙媒体への出力処理を実施する。

【００１７】

PDL（ページ記述言語）印刷を実施する場合は、クライアントPC１２０がLAN１５０を介して印刷指示を行う。CPU２０１はSATA I/F２０５を介してFlash２１１からPDL印刷のプログラムとデータをDRAM２０２に読み込み、DRAM２０２に読み込まれたプログラムを実行してネットワークI/F２０４を介して印刷指示を検出する。CPU２０１はPDL送信指示を検出するとネットワークI/F２０４を介して印刷データを受信し、SATA I/F２０５を介してFlash ROM２１１に印刷データを保存する。CPU２０１は印刷データの保存が完了すると、Flash ROM２１１に保存した印刷データをDRAM２０２に画像データとして展開する。CPU２０１はDRAM２０２に格納した画像データに対して出力に適した色変換処理などを実施する。CPU２０１はDRAM２０２に格納した画像データをプリンタI/F２０７を介してプリンタ部１０３に転送し、紙媒体への出力処理を実施する。

【００１８】

●ソフトウェアの構成

図３はMFPのコントローラ部１０１で実行されるソフトウェアの構造をあらわすブロック図である。コントローラ部１０１で実行されるソフトウェアは全て、CPU２０１が実行する。CPU２０１は、ROM２２０に記憶されたBIOS３６０を実行する。CPU２０１は、Flash ROM２１１に記憶された、ローダ３７０、Initrd３８０、コントローラソフト３００をDRAM２０２に読み込んだ後に実行する。BIOS３６０はI/Oコントローラ２０３やDRAM２０２をCPU２０１が制御するための基本処理を実行する。

【００１９】

BIOS３６０は内部的にBIOSとしての制御ソフトと制御ソフトに対応する署名データとを含む。ローダ読み込み検証部３６１はBIOS３６０の制御ソフトに含まれ、ローダを検証する処理コードとローダに付与された署名に対応する公開鍵を含む。さらにBIOS３６０はFlash ROM２１１からローダ３７０を読み込み、開始する処理を含む。ローダ３７０はFlash ROM２１１からカーネル３９０、Initrd３８０を読み込み、開始する処理を実行する。ローダ３７０は内部的にローダとしての制御ソフトと制御ソフトに対応する署名データとを含む。

【００２０】
カーネル、Initrd読み込み検証部３７１はローダ３７０に含まれ、カーネル、Initrdを検証する処理とカーネル、Initrdに付与された署名に対する公開鍵を含む。Initrd３８０はFlash ROM２１１からコントローラソフト３００を読み込み、開始する処理を実行する。Initrd３８０は内部的にInitrdとしての制御ソフトと制御ソフトに対する署名データとを含む。起動時検証部３８１はInitrd３８０に含まれ、コントローラソフト３００を構成する全てのプログラムファイルを起動時に検証する処理と、付与された署名に対する公開鍵を含む。ここで、全ての署名データに対する秘密鍵はソフトウェアの開発時のみ利用され一般に流通することはない。
【００２１】
操作制御部３０１は操作部１０２にユーザ向けの画面イメージを表示し、ユーザ操作を検知して、操作された画面上に表示したボタン等の画面部品に紐づけられた処理を実行する。データ記憶部３０２は他の制御部からの要求でデータをFlash ROM２１１に記憶、および読み出しを行う。例えば、ユーザが何らかの機器設定を変更したい場合は、操作部１０２にユーザが入力した内容を操作制御部３０１が検知し、操作制御部３０１からの要求でデータ記憶部３０２が設定値としてFlash ROM２１１に保存する。ネットワーク制御部３０７はデータ記憶部３０２に記憶された設定値に従い、システム起動時や、設定変更検出時にIPアドレスなどネットワーク設定をTCP/IP制御部３０８に行う。
【００２２】
TCP/IP制御部３０８は他の制御からの指示に従い、ネットワークI/F２０４を介して、ネットワークパケットの送受信処理を行う。ジョブ制御部３０３は他の制御部からの指示に従って、ジョブ実行の制御を行う。画像処理部３０４はジョブ制御部３０３からの指示に従って、画像データを用途ごとに適した形式に加工する。印刷処理部３０５はジョブ制御部３０３からの指示に従い、プリンタI/F２０７を介して、紙媒体に画像を印刷し出力する。読み取り制御部３０６はジョブ制御部３０３からの指示に従い、スキャナI/F２０８を介して、設置された原稿を読み込む。USB制御部３１１はUSB I/F２０９を制御し、USB接続された任意の機器の制御を行う。起動時検証用正解値リスト３２１は起動時検証部３８１が検証処理に利用する正解値のリストである。図４に起動時検証用正解値リスト３２１のデータ形式のサンプルを示す。起動時検証用正解値リスト３２１はコントローラソフト３００に含まれる全てのプログラムファイルに対して、ファイル名３００１とハッシュ３００２の組み合わせをリスト化したものである。プログラムファイルにはOSのプログラムおよび機能を提供するプログラムまたはそのいずれかが含まれる。データの内容としては、少なくともファイル名称、ファイルの配置場所（ディレクトリ上の位置）、ファイルから計算したハッシュ値を含むものとし、ハッシュ値等の情報がリスト化される。
【００２３】
アプリケーション制御部３３２は、MFP１００上で動作するアプリケーションの動作状態やアプリケーションの起動/停止を管理する。本実施形態では、Java(登録商標)モジュールの動的インストールや実行を管理する、OSGI(Open Services Gateway Initiative)フレームワークを用いて動作するアプリケーションの管理を行う。アプリケーション検証部３３１はアプリケーション制御部３３２が起動するアプリケーションの検証を実施する。
【００２４】
バンドルアプリケーション記憶部３３５はコントローラソフト３００に予め包含されるアプリケーション（バンドルアプリケーション）を記憶する。図３のバンドルアプリケーション記憶部３３５では、認証アプリケーション３３３とコピーアプリケーション３３４が記憶されている。認証アプリケーション３３３は、MFP１００を利用するユーザを認証する処理を行うためのアプリケーションである。コピーアプリケーション３３４は操作部１０２にコピーを実行するための設定等を行うための画面を表示し、ユーザの指示を受け付けてコピーを実行するためのアプリケーションである。尚、MFP１００には操作部１０２を用いてコントローラソフト３００に含まれないアプリケーションを更に追加することが可能である。アプリケーションの管理の詳細については後述する。

【００２５】
例えば、コピー機能を実行する場合は、アプリケーション制御部３３２がコピーアプリケーション３３４を起動して操作制御部３０１にコピー画面の表示を指示する。操作制御部３０１がコピー機能の開始要求を検知し、ジョブ制御部３０３にコピーを指示する。ジョブ制御部３０３は読み取り制御部３０６に原稿読み取りを指示し、スキャン画像を取得する。ジョブ制御部３０３は画像処理部３０４に指示し、スキャン画像を印刷に適した形式に変換する。ジョブ制御部３０３は印刷処理部３０５に印刷を指示し、コピー結果を出力する。実行される機能や、機能に応じたユーザインターフェイスに相違はあっても、他の機能についてもこれと同様の手順でアプリケーションは実行される。
【００２６】
●アプリケーション制御部による処理
図５を用いてアプリケーション制御部３３２の処理について詳細に説明する。図５はFlash ROM２１１内のアプリケーションに関連するディレクトリ構成の一例を示す図である。
【００２７】
Flash ROM２１１は、バンドルディレクトリ５１０、キャッシュディレクトリ５２０を含む。バンドルディレクトリ５１０には、図３のバンドルアプリケーション（例えば認証アプリケーション３３３とコピーアプリケーション３３４）が格納されている。MFP１００にコントローラソフト３００がインストールもしくはアップデートされたタイミングでバンドルアプリケーションがバンドルディレクトリ５１０に格納される。キャッシュディレクトリ５２０には、インストールされたアプリケーション（５２１～５２４）、そのアプリケーションが利用するデータ、インストールされたアプリケーションの起動順や状態を管理する管理ファイル５２５、アプリケーション検証のための正解値リスト５２６が格納されている。インストールされたアプリケーションとは、バンドルアプリケーションに加え、MFP１００の操作部１０２を用いて後からインストールされた、コントローラソフト３００に含まれないアプリケーションも含む。図５において、カード認証アプリケーション５２３、送信アプリケーション５２４が後からインストールされたアプリケーションである。すなわち、バンドルアプリケーションは、バンドルディレクトリ５１０に元のプログラムファイルが格納されており、それが展開されて例えば実行可能な形式でキャッシュディレクトリ５２０に格納されている。そのため、キャッシュディレクトリ５２０内のプログラムやデータが損なわれても、バンドルディレクトリ５１０からキャッシュディレクトリ５２０へと再インストールすることで復旧できる。
【００２８】
図６（Ａ）に管理ファイル５２５、図６（Ｂ）にアプリケーション検証用正解値リスト５２６のデータ形式のサンプルを示す。
【００２９】
図６（Ａ）の管理ファイル５２５は、キャッシュディレクトリ５２０に含まれる全てのアプリケーションに対して、アプリケーション名６００１、バンドルディレクトリパス６００２、アプリケーション種別６００３、アプリケーションの状態６００４、起動順６００５の組み合わせをリスト化したものである。図６（Ａ）において、それぞれ認証アプリケーション５２１がlogin_app、コピーアプリケーション５２２がcopy_app、カード認証アプリケーション５２３がcard_app、送信アプリケーション５２４がsend_appに対応する。アプリケーション種別６００３とは、バンドルアプリケーションであるか否かやアプリケーションタイプを示す。アプリケーションタイプとは、ログインアプリケーションやコピーアプリケーションであるかを示す情報である。ログインアプリケーションとは、MFPを利用するユーザを認証する処理を行うアプリケーションであり、例えば操作部１０２に図１２のような認証入力画面を表示し、入力された認証情報に対してユーザ認証を実施する。ユーザ認証を実施することでMFPの不正利用を制限したり、ユーザ毎にカスタマイズしたメニュー画面を表示したりすることが可能となる。図５において、認証アプリケーション５２１及びカード認証アプリケーション５２３がログインアプリケーションに分類される。

【００３０】
図６（Ｂ）のアプリケーション検証用正解値リスト５２６は、キャッシュディレクトリ５２０に含まれる全てのアプリケーションに対して、アプリケーション名６１０１とハッシュ６１０２の組み合わせをリスト化したものである。尚、アプリケーション検証用正解値リスト５２６は不正に改ざんされないよう暗号化や署名を付与してFlash ROM２１１に格納してもよい。
【００３１】
アプリケーション制御部３３２は、MFP１００にアプリケーションが追加でインストールされると、キャッシュディレクトリ５２０内にアプリケーションを格納し、アプリケーションが利用するデータを格納するディレクトリを生成する。更に、アプリケーションとデータの関連、アプリケーション種別等のアプリケーションの情報を管理ファイル５２５に格納する。また、アプリケーション制御部３３２はコントローラソフト３００のインストールに伴い、バンドルディレクトリ５１０のアプリケーションが追加されると、キャッシュディレクトリ５２０内にアプリケーションを格納し、アプリケーションが利用するデータを格納するディレクトリを生成する。バンドルディレクトリ５１０のアプリケーションが更新された場合は、バンドルディレクトリ５１０のアプリケーションを用いてキャッシュディレクトリ５２０のアプリケーションを更新する。例えば、コントローラソフト３００の更新に伴い、バンドルディレクトリ５１０の認証アプリケーション３３３が更新されると、認証アプリケーション３３３を用いてキャッシュディレクトリ５２０内の認証アプリケーション５２１の更新を行う。
【００３２】
アプリケーション制御部３３２はキャッシュディレクトリ５２０のアプリケーションを追加もしくは更新する際に、アプリケーション検証部３３１にアプリケーション検証用の正解値リスト５２６の更新を依頼する。アプリケーション検証部３３１はアプリケーション制御部３３２からの指示に基づきキャッシュディレクトリ５２０に格納されたアプリケーションのハッシュ値を算出し、アプリケーション用正解値リスト５２６を更新する。アプリケーション制御部３３２は、MFP１００起動時に管理ファイル５２５を読み込み、起動順に従ってキャッシュディレクトリ５１０内のアプリケーションを順に起動する。また、アプリケーション制御部３３２は、アプリケーションの状態の変更等により管理情報に変更が発生すると、それに合わせて管理ファイル５２５を更新する。
【００３３】
●ソフトウェアの検証処理
図７を用いて、MFP１００が起動時にソフトウェアを検証する処理フローを説明する。この処理は、MFP１００が起動するたびに一度行われる。ここで図７（A）のMFP１００が実施する処理は、BIOS検証ユニット２２１が実施するものである。以下の説明で、図７（A）の検証処理をハードウェア検証と呼ぶ。図７（B）のMFP１００が実施する処理は、Flash ROM２１１に格納されたプログラムをCPU２０１がDRAM２０２に読み込んだのち、CPU２０１の演算処理として実施するものである。以下の説明で、図７（B）の検証処理をソフトウェア検証と呼ぶ。夫々の検証処理は同じMFP１００による検証処理であっても、検証主体が異なること、ハードウェア検証はCPU２０１の実行するソフトウェアの検証処理ではないことに留意されたい。
【００３４】
電源が供給され、起動処理が開始されるとBIOS検証ユニット２２１が起動され、S１００１としてBIOSの検証処理を開始する。S１００２でMFP１００はBIOS３６０の検証処理を実施し、成功したかどうか確認する。成功した場合はS１００３を実行し、失敗した場合はS１００５を実行する。検証処理としてはBIOS検証ユニット２２１がROM２２０から読み込んだBIOS３６０の署名に対して、BIOS検証ユニット２２１に配置された公開鍵を用いて署名検証を行う。本実施形態の起動時検証は、起動順序を考慮した署名検証であり、署名検証では、次に起動する主体の署名検証を行うことでセキュリティ性を担保する。本例では初段のBIOS検証のみが専用ハードウェアで行われ、これ以降の検証はソフトウェアで

行われるので、ここでは検証ソフトウェアをひとつのオブジェクトとして見ることで検証処理の主体として扱っている。BIOS検証の次段の検証主体は、本例ではBIOS３６０に含まれたローダ読み込み検証部３６１である。

【００３５】

Ｓ１００３でMFP１００はCPU２０１に指示することでBIOS３６０を起動する。Ｓ１００５でMFP１００はBIOSの起動を行わず、起動シーケンスをこのステップで中止することでシステムを停止する。ここで、BIOS検証ユニット２２１はユーザ通知に関するデバイスを持たないため、通知は行わないが、LED(Light Emitting Diode)を接続して、発光させることで通知をおこなっても良い。Ｓ１００４でMFP１００はBIOS３６０の検証処理を終了する。ハードウェア検証は、ハードウェアで実装された検証方法であり、この検証処理を改ざんするためには集積回路の改ざんが必要であり、極めて堅牢な検証方法である。

【００３６】

BIOS３６０が起動されると、Ｓ１０１１としてFlashROM２１１に配置されたソフトウェアの検証処理を開始する。すなわち、検証の成功を契機にして、検証済みの主体により、次の段の検証主体となるソフトウェアの検証が行われる。

【００３７】

Ｓ１０１２でMFP１００はローダ読み込み検証部３６１を利用して、ローダ３７０の検証処理を実施し、成功したかどうか確認する。成功した場合はＳ１０１３を実行し、失敗した場合はＳ１０２２を実行する。検証処理としてはFlashROM２１１から読み込んだ、次の起動対象であるローダ３７０の署名に対して、ローダ読み込み検証部３６１が持つ公開鍵を用いて署名検証を行う。

【００３８】

Ｓ１０１３でMFP１００はローダを起動する。Ｓ１０１４でMFPはカーネル、Initrd読み込み検証部３７１を利用して、カーネル３９０の検証処理を実施し、成功したかどうか確認する。成功した場合はＳ１０１５を実行し、失敗した場合はＳ１０２２を実行する。検証処理としてはFlashROM２１１から読み込んだ、次の起動対象であるカーネル３９０の署名に対して、Initrd読み込み検証部３７１が持つカーネル３９０の署名に対する公開鍵を用いて署名検証を行う。Ｓ１０１５でMFP１００はカーネルを起動する。

【００３９】

Ｓ１０１６でMFPはカーネル、Initrd読み込み検証部３７１を利用して、Initrd３８０の検証処理を実施し、成功したかどうか確認する。成功した場合はＳ１０１７を実行し、失敗した場合はＳ１０２２を実行する。検証処理としてはFlashROM２１１から読み込んだ、次の起動対象であるInitrd３８０の署名に対して、Initrd読み込み検証部３７１が持つInitrd３８０の署名に対する公開鍵を用いて署名検証を行う。Ｓ１０１７でMFP１００はInitrd３８０を起動する。

【００４０】

Ｓ１０１８でMFP１００は起動時検証部３８１を利用して、コントローラソフト３００の検証を実施し、成功したかどうか確認する。成功した場合はＳ１０１９を実行し、失敗した場合はＳ１０２２を実行する。検証処理としてはFlash ROM２１１から読み込んだ次の起動対象である起動時検証用正解値リスト３２１に記載された、コントローラソフト３００に含まれる全プログラムファイルのハッシュ値を取得する。そして、Flash ROM２１１を読み込んで再計算した全プログラムファイルのハッシュ値と、ファイル毎に比較する処理が行われる。Ｓ１０１９でMFP１００はコントローラソフト３００の起動を開始する。コントローラソフト３００は複数のプログラムファイルに分割されているため、システムの起動のために必要なプログラムファイルが順次起動される。Ｓ１０２０でMFPはアプリケーション制御部３３２を利用して、アプリケーションの起動処理を実施する。アプリケーションの起動処理については後述する。Ｓ１０２２でMFP１００は改ざんを検知したことを、操作部１０２にエラー画面を表示することでユーザに通知する。Ｓ１０２３でMFP１００は起動シーケンスをこのステップで中止することでシステムを停止する。Ｓ１０２１でMFP１００はFlashROM２１１に配置されたソフトウェアの検証処理を終了する。

【００４１】
一般的にソフトウェア検証はソフトウェアであるプログラムによって実装された検証方法であるため、記憶部のソフトウェアを書き換えることで改ざんすることができる。上記フローの様に、検証を行うソフトウェアをあらかじめ別の構成部によって検証しておくことで、改ざんされていないことを保証できる。そして、プログラム内の各プログラムは順番にソフトウェア検証を行うが、１つ前のソフトウェア検証を起点にシステム全体が改ざんされていないことを保証できる。
【００４２】
●アプリケーション起動処理
図８Ａ、図８Ｂを用いて、ＭＦＰ１００がアプリケーション制御部３３２によりアプリケーションを起動する処理フローを説明する。図８のＭＦＰ１００が実施する処理は、Flash ROM２１１に格納されたプログラムをCPU２０１がDRAM２０２に読み込んだのち、CPU２０１の演算処理として実施するものである。
【００４３】
ＭＦＰ１００はアプリケーション制御部３３２を利用して、Ｓ２００１にて、Flash ROM２１１のキャッシュディレクトリ５２０から管理ファイル５２５を読み込み起動するアプリケーションを決定する。アプリケーション制御部３３２はアプリケーションを起動する前に当該アプリケーションの検証をアプリケーション検証部３３１に要求する（Ｓ２００２）。ここで、アプリケーション制御部３３２は、アプリケーション検証要求に、起動するアプリケーション名及びアプリケーション種別を含める。アプリケーション名、アプリケーション種別は、それぞれ図６（Ａ）の管理ファイル５２５に含まれるアプリケーション名６００１、アプリケーション種別６００３の情報を示す。アプリケーション検証部３３１はアプリケーションの検証要求を受信するとＳ２１００で検証処理を実行する。その詳細は図８Ｂを参照して後で説明する。
【００４４】
アプリケーション検証が終了すると、アプリケーション検証部３３１はその結果を通知し（Ｓ２１０１）、Ｓ２００３にてアプリケーション制御部３３２はアプリケーション検証部３３１から結果情報を受信する。アプリケーション制御部３３２は受信した結果情報を確認する（Ｓ２００４）。結果情報が「アプリケーション起動許可」であると判断した場合、アプリケーション制御部３３２は当該アプリケーションの起動処理を実施する（Ｓ２００５）。結果情報が「アプリケーション起動禁止」であると判断した場合、アプリケーション制御部３３２は当該アプリケーションの起動処理は実施せず、停止アプリケーションとして管理ファイルの状態を更新する（Ｓ２００６）。結果情報が「アプリケーション復旧要求」であると判断した場合、アプリケーション制御部３３２はアプリケーションの復旧処理を実施する（Ｓ２００７）。
【００４５】
アプリケーションの復旧処理として、アプリケーション制御部３３２はまずキャッシュディレクトリ５２０内の当該アプリケーションを削除し、バンドルディレクトリ５１０内に含まれる当該アプリケーションをキャッシュディレクトリ５２０に展開して、アプリケーションを復旧する。すなわち、バンドルディレクトリ５１０に含まれるプログラムファイルを用いて、アプリケーションをキャッシュディレクトリ５２０に再インストールするということもできる。例えば、図５のコピーアプリケーション５２２の検証結果が「アプリケーション復旧要求」であった場合、アプリケーション制御部３３２はキャッシュディレクトリ５２０内のコピーアプリケーション５２２を削除し、管理ファイル５２５を参照してバンドルディレクトリ５１０内のコピーアプリケーション３３４を取得して、キャッシュディレクトリ５２０のコピーアプリケーションの領域に展開することでコピーアプリケーション５２２を復旧する。そして、アプリケーション制御部３３２はアプリケーション検証部３３１にアプリケーション検証用の正解値リスト５２６の更新を依頼する（Ｓ２００８）。アプリケーション検証部３３１はそれを受信すると（Ｓ２１０２）、キャッシュディレクトリ５２０の更新されたアプリケーションを用いてハッシュ値を算出し、アプリ

ケーション用正解値リスト５２６を更新する（Ｓ２１０３）。そして正解値リスト５２６の更新の完了をアプリケーション制御部３３２に通知する（Ｓ２１０４）。アプリケーション制御部３３２は、正解値リストの更新完了の通知を受けると（Ｓ２００９）、キャッシュディレクトリ５２０の当該アプリケーションを起動する（Ｓ２０１０）。アプリケーション制御部３３２は、管理ファイル５２５を参照して次に起動するアプリケーションの起動処理を、Ｓ２００２から繰り返す。なお図８Ａにおいて台形のブロックは、それに挟まされたブロックを、条件に従って繰り返すことを示す。ここでその条件はキャッシュディレクトリ５２０内のアプリケーションすべてを対象とすることである。

【００４６】

ここで、バンドルディレクトリ５１０内のアプリケーションは起動時検証部３８１によりコントローラソフトウェア３００の一部としてS１０１８において検証されており、改ざんされていないことが保証されている。そのため、Ｓ２００７にてバンドルディレクトリ５１０のアプリケーションを用いてアプリケーションを復旧する際、バンドルディレクトリ５１０のアプリケーションの検証は実施しない。また、復旧されたキャッシュディレクトリ５２０内のアプリケーションについても検証を実施せず、そのままアプリケーションの起動を行う。

【００４７】

次に図８Ｂのアプリケーション検証処理について説明する。アプリケーション検証部３３１はアプリケーションの検証要求を受信すると（Ｓ２２００）、検証すべきアプリケーション名とアプリケーション種別の情報を取得する（Ｓ２２０１）。そして、Ｓ２２０２にてアプリケーション検証部３３１はFlash ROM２１１のキャッシュディレクトリ５２０からアプリケーション検証用正解値リスト５２６を読み込む。次に、Ｓ２２０３にてアプリケーション検証部３３１はアプリケーションの検証処理を実施し、検証が成功したかどうか確認する。アプリケーションの検証処理では、Flash ROM２１１のキャッシュディレクトリ５２０に格納されたアプリケーションのデータを読み込んでハッシュ値を算出し、正解値リスト５２６に記載された当該アプリケーションのハッシュ値と比較する処理が行われる。比較した値が一致してアプリケーションの検証に成功した場合は、結果情報として「アプリケーション起動許可」を設定する（Ｓ２２０５）。アプリケーションの検証に成功しなかった場合は、Ｓ２２０６において、アプリケーション検証部３３１はＳ２２０１で取得したアプリケーションの種別を確認し、バンドルアプリケーションか否かを判断する。バンドルアプリケーションではないと判断した場合は、結果情報として「アプリケーション起動禁止」を設定する（Ｓ２２０８）。バンドルアプリケーションであると判断した場合、結果情報として「アプリケーション復旧要求」を設定する（Ｓ２２０７）。Ｓ２１０１にてアプリケーション検証部３３１は結果情報をアプリケーション制御部３３２に通知する。このように、本実施形態の情報処理装置は、ＢＩＯＳに含まれたローダを初段とし、アプリケーションプログラムをロードするためのInitrdを最終段として、ローディング対象のプログラムをロードするとともに前段のローダにより次段のローダをロードする複数段のローダを備えている。これによって、ローダを含むＢＩＯＳを改ざんできないハードウェア等により検証し、検証されたローダによってプログラムを読み、プログラムと次段のローダを検証する。この繰り返しによって、ロードされるプログラムの正当性が検証される。

【００４８】

以上、実施形態１により、アプリケーションの正当性を確認するシステムにおいて、正当性の検証に失敗した際、アプリケーション種別に応じて復旧できるか否かを判断し、復旧可能（すなわち再インストール可能ということもできる）と判断した場合はアプリケーションを自動で復旧して起動することで、システムの可用性を担保しつつ、ユーザは機器をセキュアに利用することができる。

【００４９】

［実施形態２］

以下、本発明の第２の実施の形態を図面に基づいて解説する。本実施形態では、アプリケーション検証に失敗した場合にアプリケーションの種別に応じて、ユーザへの通知方法を

制御する方法について説明する。

【００５０】

図９は操作部１０２に表示されるメニュー画面９０１であり、複合機が持つさまざまな機能の実行をユーザが指示するためのものである。ボタン９０２はコピー機能をユーザが指示するために利用される。ボタン９０３はスキャンして保存する機能をユーザが指示するために利用される。ボタン９０４はスキャンして送信する機能をユーザが指示するために利用される。ボタン９０５は機器の設定変更をユーザが指示するために利用される。また、画面下部にメッセージ表示領域９０６があり、機器の動作中に発生したさまざまなユーザ向けのメッセージを表示することができる。図９では、検証に失敗したアプリケーションがある旨をユーザに通知している。このメッセージは、検証に失敗したアプリケーションプログラムの使用が許されないことも同時に示している。

【００５１】

図１０は操作部１０２に表示されるエラー画面１００１である。図８では検証に失敗したアプリケーションがあるため、システムを停止したことを通知している。また、この画面から通常の機能実行画面に遷移することはできず、ユーザはMFP１００を利用することはない。

【００５２】

図１１に本実施形態におけるアプリケーション検証処理を示す。これは実施形態１の図８Ｂに代えて実行される手順を示す。図８Ｂとの差異は、バンドルアプリケーションであると判定された場合に実行されるS２３０１～S２３０３であり、それ以外は実施形態１と同じ構成および処理である。以下に図１１を説明するが図８Ｂと共通するステップについては説明を省略した部分がある。

【００５３】

S２２０４でアプリケーションの検証に失敗した場合、アプリケーション検証部３３１はアプリケーション種別を確認する。S２２０６にてS２２０１で取得したアプリケーション種別を確認しバンドルアプリケーションか否かを判断する。バンドルアプリケーションではないと判断した場合、アプリケーション検証部３３１はS２３０１において更に、アプリケーション種別がログインアプリケーションであるか否かを判断する。S２３０１において、アプリケーション検証部３３１はアプリケーション種別がログインアプリケーションであると判断した場合、図１０に示すようなエラー画面１００１を表示し、MFP１００のシステム全体を停止する（S２３０３）。システム全体の停止とは、MFP１００は起動しているがユーザはMFP１００の機能を利用できない状態を示す。MFP１００は操作部１０２に図１０のエラー画面を表示し、通常の機能実行画面への遷移を禁止するため、例えばコピー機能などのMFP１００の機能は利用できないように制限される。また、MFP１００は、図１０のエラー画面表示に伴い、ネットワークI/F２０４を介したMFP１００の機能についても、利用できないように制限する。例えば、図１０のエラー画面を表示中にネットワークI/F２０４を介してPDL印刷指示を受信しても、MFP１００は印刷の実行は行わずエラーとして印刷を終了する。S２３０２にて、アプリケーション検証部３３１はアプリケーション種別がログインアプリケーションでないと判断した場合、図９に示すようにメニュー画面のメッセージ領域にエラーメッセージ９０６を表示する。それにより、ユーザは不正なアプリケーション以外のMFP１００の利用を継続できる。

【００５４】

ログインアプリケーションの起動を停止すると、ユーザは認証されずにMFPを利用できるようになるため、MFPのセキュリティレベルが低下する。そのため、S２３０３でエラー画面を表示し、MFP１００が利用できないようにすることでログインアプリケーションが改ざんされ、不正にMFP１００を利用されることを防止することができる。

【００５５】

以上、実施形態２により、アプリケーションの正当性を確認するシステムにおいて、正当性の検証に失敗した場合に、アプリケーションの種別に応じてシステムを停止することで、検証失敗時に、機器を不正に利用されることを防止し、かつシステムの可用性を担保し

つつ、ユーザは機器をセキュアに利用することができる。

【００５６】

［その他の実施例］

本発明は、上述の実施形態の１以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサーがプログラムを読出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

【００５７】

また上記実施形態では、ＢＩＯＳ検証ユニット２２１をハードウェアで実現しているが、たとえば書き換え不可能なＲＯＭに固定的に記憶したプログラムなどにより、内容を変更できない（或いは改ざんできない）よう構成したＢＩＯＳ検証ユニット２２１を実現してもよい。

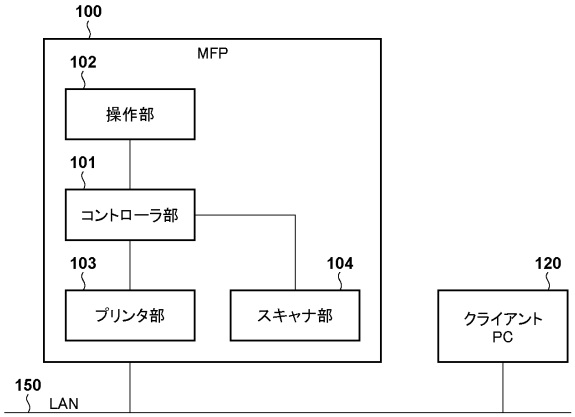
【符号の説明】

【００５８】

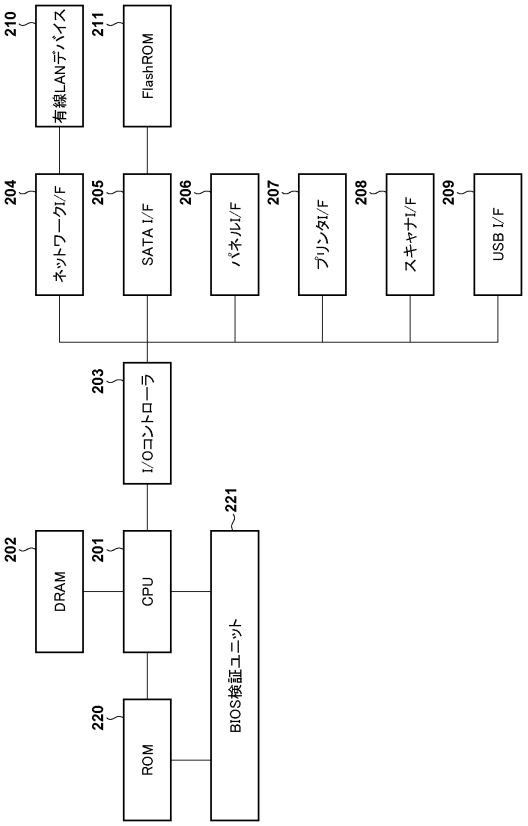
１００　MFP、２２０　CPU、３３１　アプリケーション検証部、３３２　アプリケーション制御部、５１０　バンドルディレクトリ、５２０　キャッシュディレクトリ

【図面】

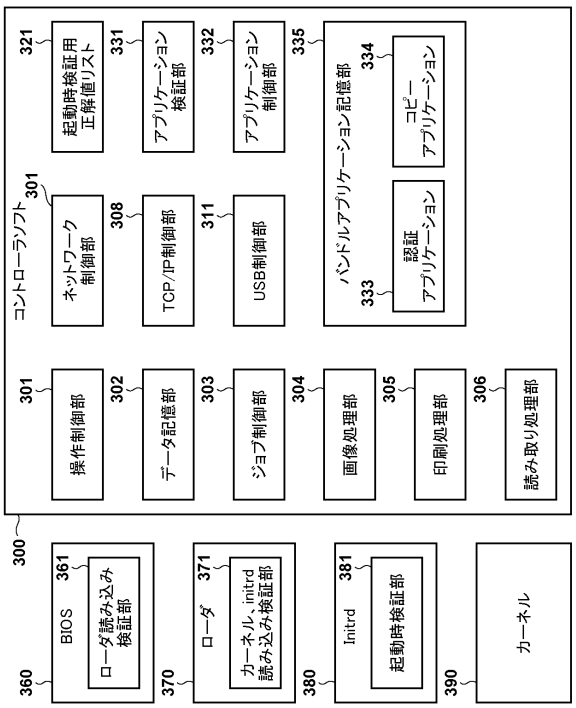
【図１】




【図２】



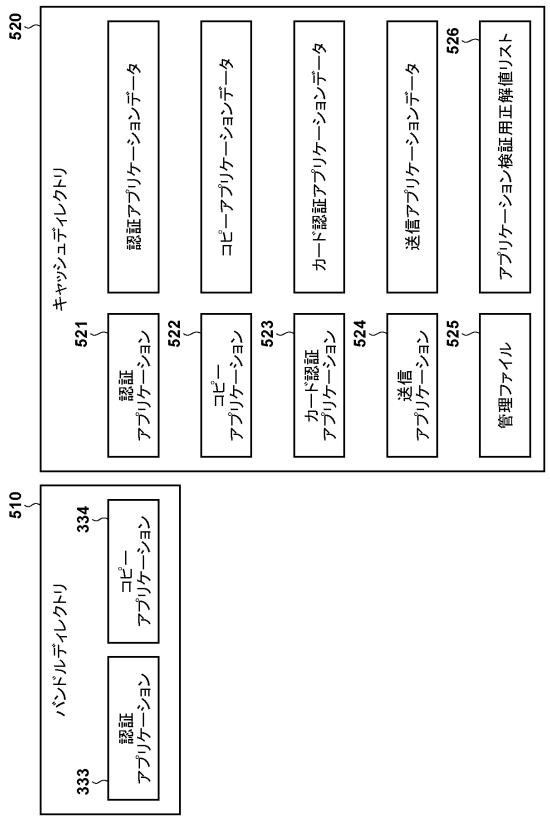

【図３】




【図４】

| 3001 | 3002 |
|---|---|
| /bin/cat | 270c2468bd4fdbc9e89ae71 |
| /lib/libc.so | e26b6d7fdd32a3e50cfb5cb |
| /lib/libm.so | 887xb9019a76dcdbbfefe25 |

【図５】




【図６】

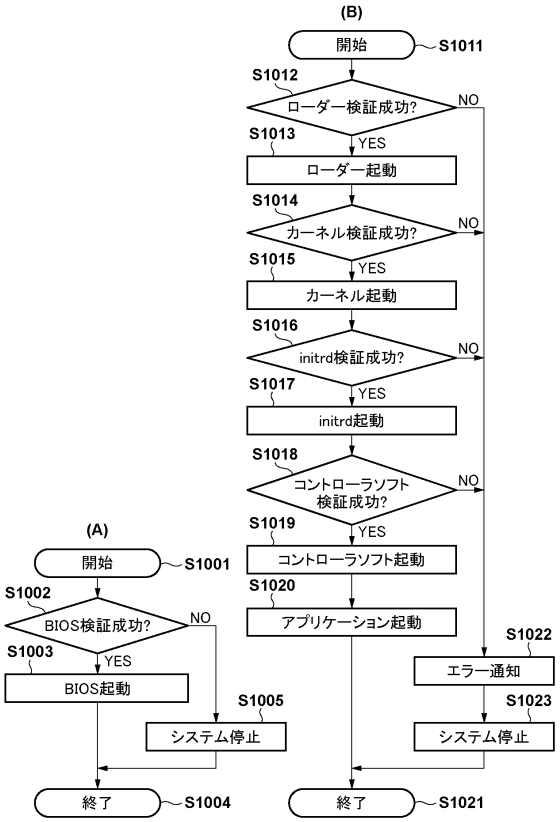
(A)

| 6001 | 6002 | 6003 | 6004 | 6005 |
|---|---|---|---|---|
| login_app | /bin/bundles/login_app.jar | bundle, login | Start | 1 |
| copy_app | /bin/bundles/copy_app.jar | bundle, copy | Start | 2 |
| card_app | – | –, login | Stop | 1 |
| send_app | – | –, send | Start | 3 |

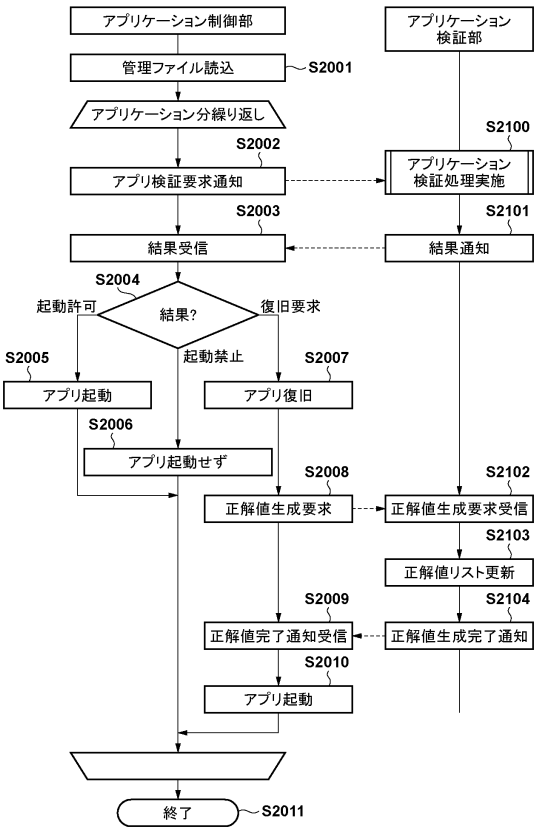
(B)

| 6101 | 6102 |
|---|---|
| login_app | 99f7c540abd20f794d7374fb |
| copy_app | 5784bdc430a7dbe320f24d53 |
| card_app | 35782bfdea93bb5fea39fbe2 |
| send_app | bd6e9f8306743baef218444c |

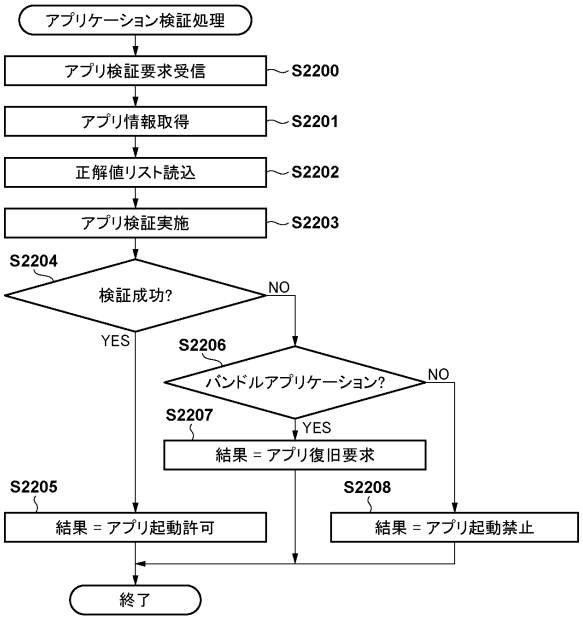
【図７】



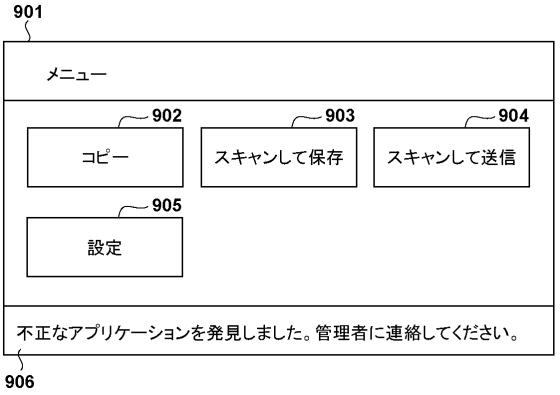

【図８Ａ】



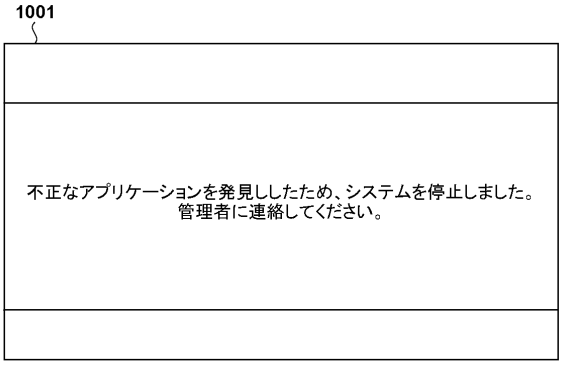
【図８Ｂ】



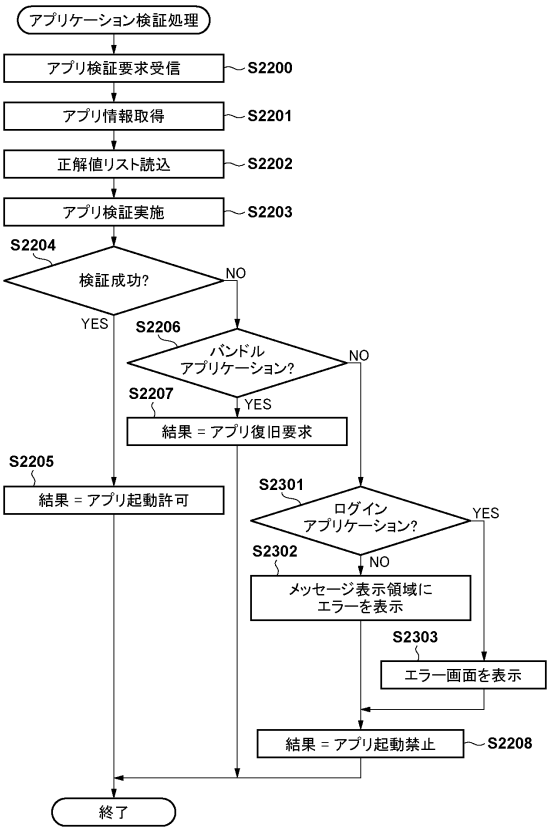

【図９】



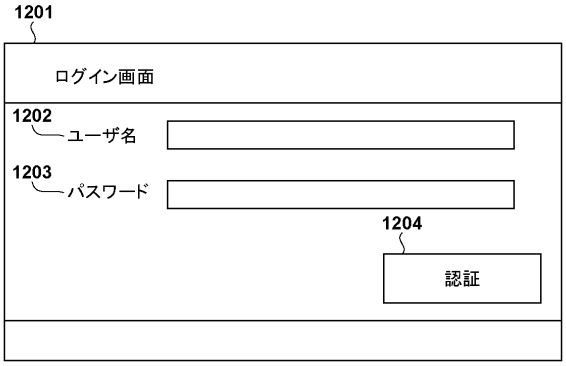

【図１０】




【図１１】



【図１２】



フロントページの続き

(56)参考文献 特開２０１６－００６６５９（ＪＰ，Ａ）
特開２００８－０１８６６６（ＪＰ，Ａ）
特開２０１７－１５３０４４（ＪＰ，Ａ）
米国特許出願公開第２０１２／０２９０８７０（ＵＳ，Ａ１）

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／００－８８