(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7102524号
(P7102524)

(45)発行日　令和4年7月19日(2022.7.19)

(24)登録日　令和4年7月8日(2022.7.8)

(51)国際特許分類		FI	
G06F	*9/4401(2018.01)*	G06F	9/4401
G06F	*8/61 (2018.01)*	G06F	8/61
G06F	*9/445(2018.01)*	G06F	9/445

請求項の数　11　（全12頁）

(21)出願番号	特願2020-531623(P2020-531623)
(86)(22)出願日	平成30年12月11日(2018.12.11)
(65)公表番号	特表2021-507353(P2021-507353A)
(43)公表日	令和3年2月22日(2021.2.22)
(86)国際出願番号	PCT/CA2018/051575
(87)国際公開番号	WO2019/113686
(87)国際公開日	令和1年6月20日(2019.6.20)
審査請求日	令和3年9月17日(2021.9.17)
(31)優先権主張番号	62/598,095
(32)優先日	平成29年12月13日(2017.12.13)
(33)優先権主張国・地域又は機関	米国(US)
(31)優先権主張番号	62/598,319
(32)優先日	平成29年12月13日(2017.12.13)

最終頁に続く

(73)特許権者　511015984
アブソリュート ソフトウェア コーポレイション
カナダ国 ヴィ７エックス １ケイ８ ブリティッシュコロンビア州 バンクーバー ダンスミュア ストリート １４００－１０５５

(74)代理人　100078880
弁理士　松岡 修平

(72)発明者　ホルジェンコ， ユージーン
カナダ国 ヴィ７エックス １ケイ８ ブリティッシュコロンビア州， バンクーバー， ダンスミュア ストリート １４００－１０５５ アブソリュート ソフトウェア コーポレイション内

最終頁に続く

(54)【発明の名称】　複数のバイナリイメージのファームウェア公開

(57)【特許請求の範囲】

【請求項１】

電子デバイスのファームウェアから、該電子デバイスのオペレーティングシステムに複数のバイナリイメージを公開する方法であって、

前記電子デバイスの起動中に実行される、

前記ファームウェアに記録されている第１バイナリイメージを、前記電子デバイスの構成テーブルにインストールするステップと、

前記ファームウェアに記録されている第２バイナリイメージを、前記構成テーブルにインストールするステップと、

前記第１バイナリイメージのコピーを、前記オペレーティングシステムのファイルシステムに保存するステップと、

前記オペレーティングシステムのロードが開始した後に実行される、

前記第１バイナリイメージのコピーを実行して、

前記ファイルシステムに前記第２バイナリイメージのコピーを保存し、

第３バイナリイメージを前記構成テーブルから前記ファイルシステムにコピーする、

ステップと、

を有する方法。

【請求項２】

前記ファームウェアに記録されている前記第３バイナリイメージを、前記構成テーブルにインストールするステップを有する、

請求項１に記載の方法。
【請求項３】
前記第１バイナリイメージの前記コピーは、前記ファイルシステムに前記第２バイナリイメージの前記コピーを保存するステップ、および、前記第３バイナリイメージを前記構成テーブルから前記ファイルシステムにコピーするステップの前に実行される、
前記構成テーブルを検索して、前記第２および前記第３バイナリイメージを見つけるステップと、
前記第２および前記第３バイナリイメージの完全性を検証するステップと、
を有する、請求項１に記載の方法。
【請求項４】
前記構成テーブル内の前記第２および前記第３バイナリイメージのハッシュを計算するステップと、
前記ハッシュを読み取り専用のＵＥＦＩ（Unified Extensible Firmware Interface）変数に保存するステップと、
前記第１バイナリイメージのコピーを用い、前記ハッシュにアクセスすることによって、前記第２および前記第３バイナリイメージの完全性を検証するステップと、
を有する、請求項１に記載の方法。
【請求項５】
前記第２および前記第３のバイナリイメージが、それぞれ異なるエンティティから提供され、
前記構成テーブルが、ＡＣＰＩ（Advanced Configuration and Power Interface table）テーブルである、
請求項１に記載の方法。
【請求項６】
前記第１バイナリイメージは、前記構成テーブル内の、特定のオペレーティングシステムプラットフォームバイナリテーブルにインストールされ、
前記第２および前記第３バイナリイメージはどちらも、前記構成テーブル内の、共通プラットフォームバイナリテーブルに、ノードとしてインストールされる、
請求項１に記載の方法。
【請求項７】
前記第１バイナリイメージの前記コピーは、実行されると、
前記第２および前記第３バイナリイメージが前記オペレーティングシステムのロードの完了後に自動的に実行されるよう、前記オペレーティングシステム内のレジストリを変更するステップを実行する、
請求項１に記載の方法。
【請求項８】
複数のバイナリイメージを公開する電子デバイスであって、
プロセッサと、
オペレーティングシステムと、
コンピュータで読み取り可能な指示を記憶するファームウェアと、
を含み、
前記コンピュータで読み取り可能な指示は、前記プロセッサによって実行されると、
前記電子デバイスの起動中に、該電子デバイスに、
前記ファームウェアに記憶された第１バイナリイメージの、前記電子デバイスの構成テーブルへのインストールと、
前記ファームウェアに記憶された第２バイナリイメージの、前記構成テーブルへのインストールと、
前記ファームウェアに記憶された第３バイナリイメージの、前記構成テーブルへのインストールと、
前記第１バイナリイメージのコピーの、前記オペレーティングシステムのファイルシステ

ムへの保存と、
を実行させ、
前記オペレーティングシステムのロードが開始した後に、該電子デバイスに前記第１バイナリイメージの前記コピーを実行させることにより、
前記ファイルシステムへの前記第２バイナリイメージのコピーの保存と、
前記第３バイナリイメージの前記構成テーブルから前記ファイルシステムへのコピーと、
を実行させる、
電子デバイス。
【請求項９】
前記構成テーブルが、ＡＣＰＩ（Advanced Configuration and Power Interface table）テーブルである、
請求項８に記載の電子デバイス。
【請求項１０】
前記第１バイナリイメージは、前記構成テーブル内の、特定のオペレーティングシステムプラットフォームバイナリテーブルにインストールされ、
前記第２および前記第３バイナリイメージはそれぞれ、前記構成テーブル内の、別々のプラットフォームバイナリテーブルにインストールされる、
請求項８に記載の電子デバイス。
【請求項１１】
コンピュータで読み取り可能な指示を記録した非遷移的記録媒体であって、
該コンピュータで読み取り可能な指示は、電子デバイス内のプロセッサによって実行されると、
前記電子デバイスの起動中に、該電子デバイスに、
前記電子デバイスのファームウェアに記憶された第１バイナリイメージの、前記電子デバイスの構成テーブルへのインストールと、
前記ファームウェアに記憶された第２バイナリイメージの、前記構成テーブルへのインストールと、
前記ファームウェアに記憶された第３バイナリイメージの、前記構成テーブルへのインストールと、
前記第１バイナリイメージのコピーの、前記電子デバイスのオペレーティングシステムのファイルシステムへの保存と、
を実行させ、
前記電子デバイスに前記第１バイナリイメージの前記コピーを実行させることにより、
前記ファイルシステムへの前記第２バイナリイメージのコピーの保存と、
前記ファイルシステムへの前記第３バイナリイメージのコピーの保存と、
を実行させる、
非遷移的記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本出願は、ファームウェアからバイナリイメージを公開する分野に関連する。特に、ファームウェアからオペレーティングシステム環境への複数のバイナリイメージの公開に関連する。
【背景技術】
【０００２】
現在のWindows（登録商標）Platform Binary Table（ＷＰＢＴ）の実装の形態では、実行のためにファームウェアからWindows（登録商標）に公開できるバイナリイメージは1つだけである。ファームウェアは、ブート中にシステム情報のテーブルを構築する。このテーブルは、オペレーティングシステムが、例えば、どのハードウェアがインストールされているかを判断するために使用される。テーブルの１つには、ファームウェアに組み

込まれている実行ファイルに関する情報が含まれている。オペレーティングシステムは、ブート中にこのテーブルを探し、存在する場合は実行ファイルをファイルシステムにコピーして実行する。

【発明の概要】

【０００３】

本発明は、一つの公開可能な実行ファイルに関する情報を保持するように構成されたバイナリテーブルが、オペレーティングシステムのブート中に、複数の公開されるバイナリを生成することを可能にする。特に、本明細書は、ＷＰＢＴの拡張を開示し、それによってWindows（登録商標）および他のオペレーティングシステムにより実行される複数のバイナリイメージを、ファームウェアが公開するためのサポートを追加する。この拡張を使用することにより、ＯＥＭ（Original Equipment Manufacturer）は独自のバイナリイメージを、Absolute（登録商標）Persistence（登録商標）Agentまたは任意の他のソフトウェアエージェントのイメージとともに含むことができる。エンドカスタマは、拡張内にリストされているエージェントまたはバイナリイメージの何れか又は全てを利用する、或いは、何れも利用しないクライアントソフトウェアを購入するオプションを有している。

【図面の簡単な説明】

【０００４】

【図１】図１は、本発明の一実施形態による、複数のバイナリを公開するための処理の主なステップの概要である。

【図２】図２は、本発明の一実施形態による、複数のバイナリイメージを公開するための実行およびコードのフローを示すシステムの概略ブロック図である。

【図３】図３は、本発明の一実施形態による、ＸＰＢＴ及び他のＡＣＰＩテーブルのデータ構造及びそれらの相互関係を示す概略図である。

【図４】図４は、本発明の一実施形態による、複数のバイナリを公開するための処理のステップのフローチャートである。

【発明を実施するための形態】

【０００５】

Ａ．用語

【０００６】

ＡＢＴ：Absolute Software（アブソリュートソフトウェア）

【０００７】

ＡＣＰＩ：Advanced Configuration and Power Interfaceは、デスクトップおよびモバイルコンピュータの電力消費を効率的に制御するための業界仕様である。ＡＣＰＩは、コンピュータの基本入出力システム（Basic Input/Output System）、オペレーティングシステム、および周辺装置が、電力使用量に関して互いにどのように通信するかを指定する。ＡＣＰＩは、ＡＣＰＩ準拠のオペレーティングシステムとシステムファームウェアとの間のインタフェースを提供するテーブルを定義する。これらのテーブルは、プラットフォームに依存しない方法でシステムハードウェアの記述を可能にし、固定フォーマットのデータ構造またはＡＭＬ（ＡＣＰＩマシン言語）のいずれかとして表される。

【０００８】

ＡＰＩ：Application Programming Interface（アプリケーション・プログラミング・インターフェース）。

【０００９】

ＢＩＯＳ：Basic Input/Output Systemは、電子デバイスのブートプロセス中のハードウェアの初期化、およびオペレーティングシステムとプログラムへのランタイムサービスの提供に使用されるファームウェアである。

【００１０】

ＤＸＥ：Driver Execution Environment（ドライバ実行環境）

【００１１】

ＦＡＴ：File Allocation Table（ファイル割り当てテーブル）

【００１２】
ＮＴＦＳ：New Technology File System
【００１３】
ＯＥＭ：Original Equipment Manufacturer
【００１４】
ＯＳ：Operating System（オペレーティングシステム）
【００１５】
Platform Extension Manager：ＷＢＰＴによって指定されるWindows（登録商標）ネイティブユーザーモードアプリケーション（例えば、PlatExtMgr.exe、wpbbin.exe）。このネイティブユーザモードアプリケーションは、PlatExtMgr.exeとしてＳＤＫに組み込まれる。これはＷＰＢＴによって公開されるため、名前のないメモリ内のバイナリバッファになる。Windows（登録商標）がＷＰＢＴを検出し、次いで、このバイナリバッファを検出すると、Windows（登録商標）はこのバッファを、MicrosoftのＷＰＢＴ仕様に従ってwpbbin.exeファイルとして保存する。これが実行されると、このアプリケーションはＸＰＢＴテーブルを検出し、それらのテーブルによって参照されるバイナリをインストールする。
【００１６】
ＲＳＤＴ：Root System Description Table（ルートシステム記述テーブル）
【００１７】
ＳＤＫ：Software Development Kit（ソフトウェア開発キット）
【００１８】
ＳＨＡ：Secure Hash Algorithm（セキュアハッシュアルゴリズム）
【００１９】
ＵＥＦＩ：Unified Extensible Firmware Interfaceは、オペレーティングシステムとプラットフォームファームウェアとの間のソフトウェアインタフェースを定義する仕様である。ＵＥＦＩは、不揮発性メモリにファームウェアとして保存される。
【００２０】
ＷＰＢＴ：Windows Platform Binary Table（Windowsプラットフォームバイナリテーブル）
【００２１】
ＸＰＢＴ：Extended Platform Binary Table（拡張プラットフォームバイナリテーブル）。ＸＰＢＴは複数のノードに対応できる。
【００２２】
ＸＳＤＴ：Extended System Description Table（拡張システム記述テーブル）
【００２３】
Ｂ．実施例
図１を参照すると、本プロセスにおける主なステップがフローチャートの形で示されている。ステップ２では、コンピュータのブート中に、複数のバイナリイメージがＡＣＰＩテーブルにインストールされる。ここで、バイナリイメージの１つは、マネージャバイナリである。ステップ４では、まだブート中であり、マネージャバイナリがオペレーティングシステムのファイルシステムに保存される。ある時点の、ＯＳのロード処理の早い段階におけるある時点で、ＯＳはマネージャバイナリを実行する。これにより、ステップ６で他のバイナリイメージがＯＳファイルシステムに保存される。以下、より詳細なフローチャートについて説明する。
【００２４】
図２には、ＵＥＦＩ ＢＩＯＳのようなファームウェア１２からコンピューティングデバイスのオペレーティングシステムに複数のバイナリを公開するためのシステム１０が示されている。
【００２５】
第１再生モジュール（例えば、ABT Persistence（登録商標））２０が、ファームウェア

１２に格納されている。この再生モジュール２０は、ＤＸＥドライバであるＡｂｔＤｘｅ２２を含み、これは、コンピューティングデバイスから遠隔にあるサーバにコンタクトすることができるセキュリティモジュールである。また、再生モジュール２０には、インストーラ２４（例えば、AbtAgentInstaller）が含まれており、インストーラ２４には、プラットフォーム拡張マネージャバイナリイメージ２６（例えば、PlatExtMgr.exe）およびエージェント２８（例えば、AbtAgent）が含まれている。ドライバ２２は、デバイスのブート時に実行され、インストーラ２４の動作を開始させる。

【００２６】

第２再生モジュールである、ＯＥＭ再生モジュール４０もファームウェア１２に格納されている。ＯＥＭ再生モジュール４０は、第２インストーラ４１、および、バイナリであるOem.exe４２を含む。また、ＯＥＭ再生モジュール４０は、デバイスの起動時に自動的に動作する。

【００２７】

マネージャバイナリイメージ２６は、インストーラ２４（例えば、AbtAgentInstaller）によってＷＰＢＴ５０にインストールされ、マネージャバイナリイメージ２６のコピー２６Ａとして示される。

【００２８】

ＸＰＢＴ（Extended Platform Binary Table）５３は、ＷＰＢＴ５０の作成とともに、ＯＳエージェント２８Ａ用のＸＰＢＴノード５４と共にインストーラ２４によって作成される。その他のすべてのＸＰＢＴノードは、ＵＥＦＩ ＢＩＯＳ１２で実行される、個別のＯＥＭのまたはサードパーティのインストーラを使用して作成される。例えば、ＡＢＴ再生モジュール２０には１つのインストーラ２４があり、ＯＥＭ再生モジュール４０には別のインストーラ４１がある。ＡＢＴ再生モジュール２０は、ＯＥＭ再生モジュール４０がOem.exeエージェントをインストールできるようにするために必要である。したがって、異なるエンティティがＸＰＢＴ５３を更新および／または拡張して、それぞれの部分を挿入することが可能である。

【００２９】

Windows（登録商標）の起動中、ＷＰＢＴによって指定されるマネージャバイナリイメージ２６Ａは、プラットフォーム拡張マネージャバイナリイメージ２６の追加コピー２６Ｂ（wpbbin.exe）として保存され、通常どおりWindows（登録商標）によって実行される。この方法では、追加のバイナリイメージをWindows（登録商標）で実行することはできない。

【００３０】

ＯＳエージェント２８ＡおよびＯＥＭエージェントOem.exe４２Ａを含む、それぞれの有効なＸＰＢＴペイロードバイナリは、ＯＳファイルシステム６０に保存される。プラットフォーム拡張マネージャ２６Ｂは、ペイロードバイナリ２８Ａ、４２Ａが特定のＯＳ用であることを確認し、それを処理する前にＸＰＢＴ５３の完全性を検証する。ＸＰＢＴ５３のＸＰＢＴ ＳＨＡ２５６ハッシュ５７は、ＵＥＦＩ ＢＩＯＳ１２内のインストーラ２４によって計算され、読み取り専用ＵＥＦＩ変数５９に保存される。プラットフォーム拡張マネージャ２６ＢがＸＰＢＴの完全性を検証するためにＵＥＦＩ変数５９を読み出す。ハッシュ５７の計算は、様々なエンティティがそれらの特定のＸＰＢＴノード５４、５６をＸＰＢＴ５３に挿入した後、ＡｂｔＤｘｅドライバ２２によって開始される。これにより、ＸＰＢＴの完全性が、プラットフォーム拡張マネージャ２６Ｂによって後から検証可能となる。

【００３１】

ＡＰＩは、さまざまなエンティティがＸＰＢＴノード５４、５６をＸＰＢＴ５３に挿入できるようにするＡｂｔＤｘｅドライバコードで提供される。したがって、ＯＥＭインストーラ４１は、インストーラ２４が起動した後に実行を開始するか、あるいは、もし事前に起動していたなら、インストーラ２４によるＸＰＢＴ５３の作成完了を待たなければならない。エージェント２８Ａはエージェント２８Ｂとして保存され、Oem.exe４２ＡはOem

.exe４２Ｂとして保存される。ＸＰＢＴインストールデータは、ＯＳレジストリ６２を更新するために使用され、その結果、ＸＰＢＴバイナリ、すなわちエージェント２８ＢおよびOem.exe４２Ｂが実行される。

【００３２】

図３は、ＷＰＢＴ、ＸＰＢＴおよび他のＡＣＰＩテーブルのデータ構造を示す。ＡＣＰＩルートシステム記述ポインタ７０は、標準ＡＣＰＩヘッダ７６およびポインタ７８を含む、ＲＳＤＴ（ルートシステム記述テーブル）またはＸＳＤＴ（拡張システム記述テーブル）７２を指す。ポインタ７８は、ＷＢＰＴ５０と１つ以上の異なる名前のＸＢＰＴ１１０、１２０とを含むＡＣＰＩテーブルを指す。ＷＢＰＴ５０は、プラットフォーム拡張マネージャバイナリイメージ２６Ｂを実行するための、標準ＡＣＰＩヘッダ８４、ハンドオフサイズ８６、ハンドオフアドレス８８、コマンド長９０、および、コマンドライン引数９２を含む。ハンドオフアドレスは、プラットフォーム拡張マネージャバイナリイメージ２６Ａを指す。

【００３３】

ＸＰＢＴ１１０、１２０の２つの例が示されている。一般に、複数のノードを有するため、ＸＰＢＴ１２０のみが使用されるが、実施形態に応じて、各タイプのテーブルが無くてもよく、あるいは、各タイプのテーブルが１またはそれ以上あってもよい。１つのバイナリを有するＸＰＢＴ１１０は、標準ＡＣＰＩヘッダ１０６、インストールデータ１１６、およびエージェント１１８のバイナリイメージを含む。インストールデータ１１６は、プラットフォーム拡張マネージャ２６Ｂによって、対応するＸＰＢＴペイロードバイナリ１１８を起動するために使用される。複数のバイナリ用のＸＰＢＴ１２０は、インストールデータ１２２、エージェント１２４のバイナリイメージ、さらにインストールデータ１２６、および別のエージェント１２８のさらなるバイナリイメージを含む。

【００３４】

ＸＰＢＴインストールデータ１１６は、バイナリ１１８が保存される位置と、バイナリ１１８が通常のＯＳブートプロセスの一部として実行されるようにレジストリに対して加えられる必要な変更とを含む。ＯＳがバイナリ１１８を実行する前の署名の検証は、ＯＳに依存している。ペイロードバッファ（例えば、バイナリイメージ１１８）は、ＸＰＢＴ１１０に直接続くように作成され、テーブルサイズは、ペイロードバッファを含むように調整される。このようにして、ＡＰＩは、全体、すなわち、テーブル１１０およびそれに対応するバイナリイメージ１１８を返す。

【００３５】

同様に、インストールデータ１２２は、バイナリ１２４が保存される位置と、バイナリ１２４が通常のＯＳブートプロセスの一部として実行されるようにレジストリに対して加えられる必要な変更とを含む。インストールデータ１２６とバイナリ１２８についても同様である。第１ペイロードバッファ（例えば、バイナリイメージ１２４）は、ＸＰＢＴ１２０に直接続くように作成され、第２インストールデータ１２６およびペイロードバッファ（例えば、バイナリイメージ１２８）は、第１ペイロードバッファ１２４に直接続くように作成される。テーブルサイズは、ペイロードバッファを含むように調整される。これにより、ＡＰＩはテーブル１２０とそれに対応するバイナリイメージ１２４、１２８を返す。

【００３６】

表１に、ＸＰＢＴ５３の例を示す。ＡＣＰＩテーブル５２内のＸＰＢＴエントリ５３は、ＷＰＢＴのものと同様である。複数のＸＰＢＴノードエントリ５４、５６がＡＣＰＩテーブル５２に存在し得る。表１～５は、例示的なＸＰＢＴエントリの詳細な定義を提供する。表２に、ＸＰＢＴに存在するイメージフラグの例を示す。表３に、ＣＰＵ Ｔｙｐｅとラベル付けされたイメージフラグの値の例を示す。表４に、ＯＳ Ｔｙｐｅとラベル付けされたイメージフラグの値の例を示す。表５に、イメージタイプとラベル付けされたイメージフラグの値の例を示す。

【表１】

フィールド	バイト長	バイトオフセット	記述
ACPI Header			
Signature	4	0	‘XPBT’署名
Length	4	4	バイト長
Revision	1	8	1
Checksum	1	9	テーブル全体の合計をゼロにする必要がある
OEMID	6	10	OEM ID
OEM Table ID	8	16	製造モデル ID
OEM Revision	4	24	OEM 改訂
Creator ID	4	28	テーブルを作成するユーティリティのベンダ ID
Creator Revision	4	32	テーブルを作成するユーティリティの改訂数
XPBT Specific			
Image Size	4	36	バイナリイメージサイズ
Image Address	8	40	バイナリイメージのアドレス
Image Flags	2	48	イメージフラグ（表２参照）
Install Data Len	2	50	インストールデータの長さ
Install Data	Variable	52	インストールデータ

【表２】

XPBT – フラグ	ビット長	ビットオフセット	記述
Binary Valid	1	0	バイナリイメージは有効であり、オブジェクトタイプが OS によってサポートされている場合に実行される必要がある。
Reserved	3	1	
CPU Type	4	4	表３参照
OS Type	4	8	表４参照
Image Type	4	12	表５参照

【表３】

フラグ– CPU タイプ	記述
0	x86
1	x86_64
2	Arm Arch32
3	Arm Arch64
4-15	Reserved

【表４】

フラグ– OS タイプ	記述
0	Windows
1	Linux
2	Android
3	Mac OS
4	iOS
5-15	Reserved

【表５】

XPBT – コンテンツタイプ	記述
0x01	PE COFF 32 bit
0x02	PE COFF 64 bit
0x03	ELF 32 bit
0x04	ELF 64 bit
0x05 – 0x0F	Reserved

【００３７】
表１～５は、複数のバイナリイメージがどのようにサポートされかを示す実装の一例にすぎず、他の実装の形態も可能である。
【００３８】
図４を参照すると、電子デバイスにおいて複数のバイナリを公開するために行われるプロセスが示されている。このプロセスは、電子デバイス内のコンピュータで読み取り可能なメモリに記憶されたコンピュータで読み取り可能な命令を実行することにより、電子デバイス内の１つまたは複数のプロセッサによって実行される。まず、コンピュータまたは他の電子デバイスが起動され、ステップ１４０で、電子デバイスがブートを開始する。ステップ１４５において、この起動プロセスの間に、電子デバイスのファームウェアは、複数のバイナリイメージのセットの中の第１バイナリイメージをＡＣＰＩテーブルにインストールする。第１バイナリイメージは、この場合、マネージャイメージ２６であり、ＡｂｔＤｘｅドライバ２２によって起動されるインストーラ２４により、ＷＰＢＴ ＡＣＰＩテーブルにインストールされる。起動プロセス中のステップ１５０において、ファームウェア

は、第２バイナリイメージ２８をＡＣＰＩテーブルにインストールする。まだ起動プロセス中のステップ１５５において、ファームウェアは、第３バイナリイメージ４２をＡＣＰＩテーブルにインストールする。

【００３９】

ステップ１６０で、ファームウェアはバイナリイメージのハッシュ５７をＵＥＦＩ読み取り専用変数として保存する。ステップ１６２で、ＯＳがロードを開始する。ステップ１６５では、ＯＳのロード中に、ＡＣＰＩテーブルにインストールされているマネージャイメージ２６Ａのコピーが、オペレーティングシステムのファイルシステムに保存される。ＯＳブート処理の初期の段階のある時点で、ステップ１７０において、ＯＳファイルシステムに保存されたマネージャ２６Ｂが実行される。マネージャ２６Ｂは、実行されると、ステップ１７５で第２バイナリ２８ＢをＯＳファイルシステムに保存し、ステップ１８０で第３バイナリ４２ＢをＯＳファイルシステムに保存する。ステップ１８５で、ＯＳはドライバ、サービスおよびアプリケーションのロードを開始し、最終的にはデスクトップが表示可能な状態になる。

【００４０】

その結果、バイナリ２８Ｂと４２Ｂの両方が実行され、ファームウェアからＯＳに公開される。

【００４１】

以上説明したように、ＸＰＢＴは既存のすべてのバージョンのWindows（登録商標）８／１０で動作し、ＸＰＢＴ専用モデルへの移行を支援する。

【００４２】

Ｃ．変形例

本発明は、ＡＢＴおよびＯＥＭ再生モジュールに関連して説明されているが、同様に機能するのであれば、別の供給元からの他のモジュールが使用されてもよい。

【００４３】

ＸＰＢＴ署名とＸＰＢＴ ＡＣＰＩテーブルレイアウトは実装に依存する。ＸＰＢＴ署名は、プラットフォーム拡張マネージャが何を検索するかを知っている限り、どのようなものであってもよい。ＸＰＢＴレイアウトは、プラットフォーム拡張マネージャが解析方法を認識している限り、どのようなものであってもよい。したがって、他の実施形態におけるＡＣＰＩ署名は、表１に関連して定義されたテーブルレイアウトを有するＸＰＢＴとは異なる。

【００４４】

全体的な目標は、ＡＣＰＩテーブルから複数のバイナリイメージをロードすることをサポートすることである。これは、ＸＰＢＴ内に複数のバイナリイメージを持つこと、または、各々が１つのイメージを持つ複数のＸＰＢＴテーブルを持つこと、あるいは、それら２つの組み合わせ（すなわち、各々が１つまたは複数のバイナリイメージを有する複数のＸＰＢＴテーブルを持つこと）、によって達成される。

【００４５】

他の実施形態では、ＡＣＰＩテーブルの代わりに他の構成テーブルを使用することができる。

【００４６】

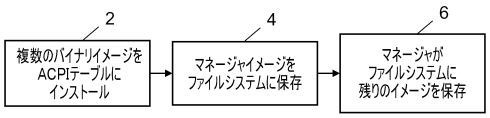
バイナリイメージペイロードは、サービス、アプリケーション、またはドライバでもよい。

【００４７】

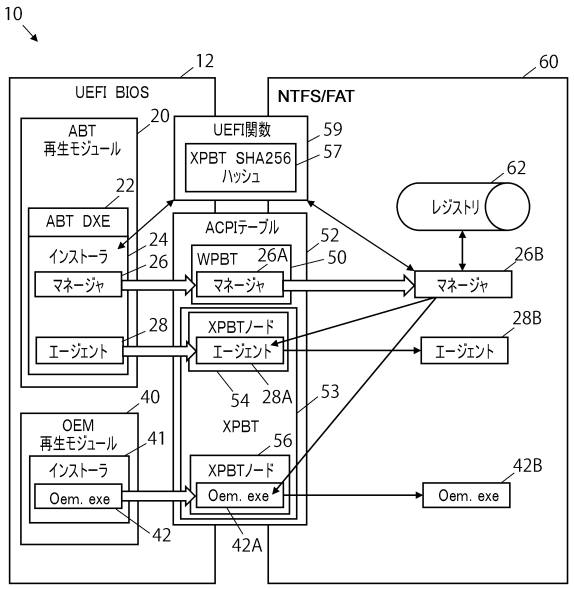
また、本発明は、他のオペレーティングシステムに関連して実施することができる。すなわち、ＸＰＢＴの仕様を、ＯＳに依存させず、Windows（登録商標）に加えて他の種類のＯＳをサポートさせることが可能である。

【図面】

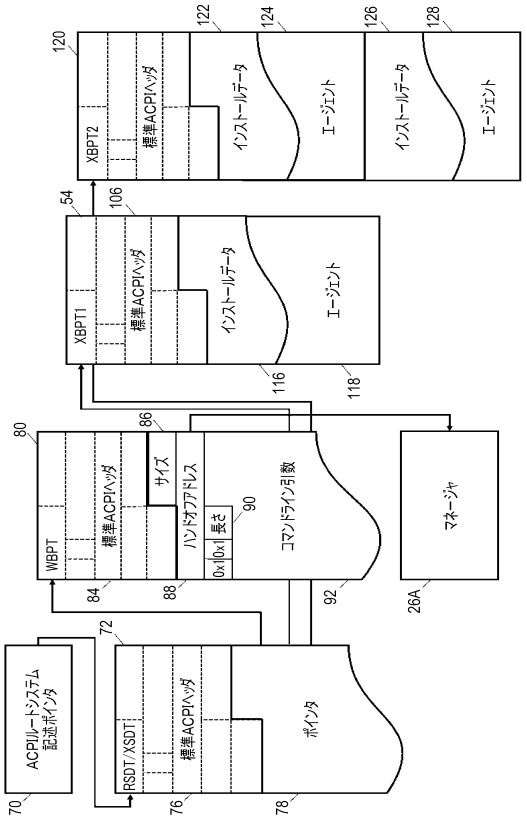
【図１】


2
4
6
複数のバイナリイメージをACPIテーブルにインストール
マネージャイメージをファイルシステムに保存
マネージャがファイルシステムに残りのイメージを保存

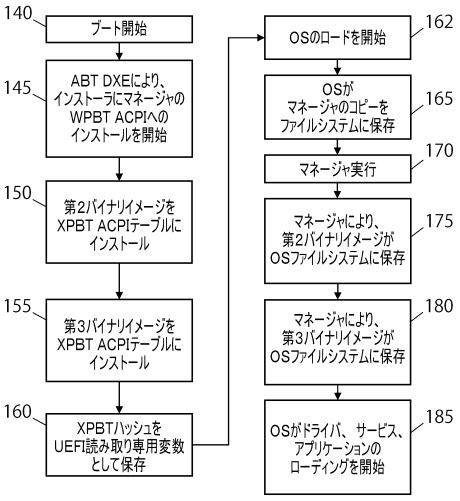
【図２】


10
12
UEFI BIOS
60
NTFS/FAT
ABT再生モジュール
20
UEFI関数
59
XPBT SHA256ハッシュ
57
62
レジストリ
22
ABT DXE
ACPIテーブル
52
インストーラ
24
WPBT
26A
50
26
マネージャ
マネージャ
マネージャ
26B
XPBTノード
28
28B
エージェント
エージェント
エージェント
54
28A
53
40
OEM再生モジュール
41
XPBT
56
インストーラ
XPBTノード
42B
Oem. exe
Oem. exe
Oem. exe
42
42A

【図３】


120
122
124
126
128
XBPT2
標準ACPIヘッダ
インストールデータ
エージェント
インストールデータ
エージェント
54
106
XBPT1
標準ACPIヘッダ
インストールデータ
エージェント
116
118
80
86
WBPT
標準ACPIヘッダ
サイズ
ハンドラアドレス
0x□□0x□
長さ
コマンドライン引数
マネージャ
84
88
90
92
26A
72
ACPIルートシステム記述ポインタ
RSDT/XSDT
標準ACPIヘッダ
ポインタ
70
76
78

【図４】


140
ブート開始
145
ABT DXEにより、インストーラにマネージャのWPBT ACPIへのインストールを開始
150
第2バイナリイメージをXPBT ACPIテーブルにインストール
155
第3バイナリイメージをXPBT ACPIテーブルにインストール
160
XPBTハッシュをUEFI読み取り専用変数として保存
162
OSのロードを開始
165
OSがマネージャのコピーをファイルシステムに保存
170
マネージャ実行
175
マネージャにより、第2バイナリイメージがOSファイルシステムに保存
180
マネージャにより、第3バイナリイメージがOSファイルシステムに保存
185
OSがドライバ、サービス、アプリケーションのローディングを開始

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
早期審査対象出願
(72)発明者 ブッシュ， ジェフリー マイケル
アメリカ合衆国 テキサス州 ７８７５８， オースティン， センチュリー オークス テラス ４３０－１１４０１ アブソリュート ソフトウェア インコーポレイテッド内
(72)発明者 ガードナー， フィリップ ビー
カナダ国 ヴィ７エックス １ケイ８ ブリティッシュコロンビア州， バンクーバー， ダンスミュア ストリート １４００－１０５５ アブソリュート ソフトウェア コーポレイション内
審査官 北川 純次
(56)参考文献 特開２００５－１８２７９０（ＪＰ，Ａ）
特開２０００－２７６３５９（ＪＰ，Ａ）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ８／６１
Ｇ０６Ｆ ９／４４０１
Ｇ０６Ｆ ９／４４５